สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# การบีบอัดเสียงโดยใช้เอดีพีซีเอ็มแล้วส่งผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี

# SPEECH COMPRESSION BY USING ADPCM THEN TRANSFERRING VIA TCP/IP PROTOCOL TO NETWORK

โดย

นาย วรุตม์ บุญธรรม

นางสาว อังคณา โชติเวชชกุล

เลขหมู่.....
เลขทะเบียน 42202
วัน, เดือน, ปี 15 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การบีบอัดเสียงโดยใช้เอดีพีซีเอ็มแล้วส่งผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี

# SPEECH COMPRESSION BY USING ADPCM THEN TRANSFERRING VIA TCP/IP PROTOCOL TO NETWORK


โดย

นาย วรุตม์ บุญธรรม 40010697

นางสาว อังคณา โชติเวชชกุล 40010981

อาจารย์ที่ปรึกษา

ดร.สุทธิชัย นพนาคีพงษ์

อาจารย์ กฤษณ์ วงศ์รุจิระ


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b19 0655x

# การบีบอัดเสียงโดยใช้เอดีพีซีเอ็มแล้วส่งผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี

# SPEECH COMPRESSION BY USING ADPCM THEN TRANSFERRING VIA TCP/IP PROTOCOL TO NETWORK


โดย นาย วรุตม์ บุญธรรม 40010697

นางสาว อังคณา โชติเวชชกุล 40010981

อาจารย์ที่ปรึกษา ดร. สุทธิชัย นพนาคีพงษ์

อาจารย์ กฤษณ์ วงรุจิระ



**บทคัดย่อ**

ในยุคของข่าวสารเช่นปัจจุบัน การติดต่อสื่อสารได้ถูกพัฒนามากขึ้น ทำให้เกิดปริมาณของข้อมูลข่าวสารเพิ่มมากขึ้น การบีบอัดข้อมูลข่าวสารจึงได้รับการพัฒนาและเข้ามามีบทบาท และในยุคนี้ระบบอินเตอร์เน็ตได้เข้ามามีบทบาทในชีวิตประจำวันของคนเรามากขึ้น เพราะปราศจากข้อจำกัดทางด้านเวลาและระยะทาง ซึ่งระบบอินเตอร์เน็ต เป็นระบบที่เชื่อมต่อกันโดยใช้โปรโตคอลทีซีพี/ไอพี

ปริญญานิพนธ์ฉบับนี้ จึงนำการบีบอัดเสียงโดยใช้เอดีพีซีเอ็ม มาทำงานร่วมกับการส่งข้อมูลผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี โดยผู้ใช้สามารถเลือกได้ว่าจะส่งข้อมูลไปที่ไอพีใด ซึ่งจะทำให้สามารถใช้งานได้หลากหลายมากขึ้น

**ABSTRACT**

In the information technology generation such as nowaday . The development of communication causes a lot of data . So, the data compression was developed order to have high efficiency . Internet has an influent to the routine of people because internet doesn't depend on time and distance . Internet is the system that used TCP/IP protocol to work on.

So , this thesis combines speech compression by using ADPCM and data transmission to network via TCP/IP protocol . By the way , the users can choose where the data goes to by using IP address to identify , so can make the many ways that we can use.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การบีบอัดเสียงโดยใช้เอดีพีซีเอ็มแล้วส่งผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี**

**SPEECH COMPRESSION BY USING ADPCM THEN**

**TRANSFERRING VIA TCP/IP PROTOCOL TO NETWORK**


ผู้จัดทำ

1. นาย วรุตม์ บุญธรรม 40010697
2. นางสาว อังคณา โชติเวชชกุล 40010981


..... อาจารย์ที่ปรึกษา

( ดร. สุทธิชัย นพนาคีพงษ์ )

Krit Wongmjit ..... อาจารย์ที่ปรึกษา

( อาจารย์ กฤษณ์ วงรุจิระ )


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| เรื่อง | | | หน้า |
|---|---|---|---|
| สารบัญรูป | | | |
| สารบัญตาราง | | | |
| บทที่ 1 | บทนำ | | 1 |
| บทที่ 2 | ทฤษฎีหรือหลักการ | | |
| | 2.1 | การเข้ารหัสของสัญญาณเสียง | 2 |
| | 2.2 | การแซมปลิ้ง (sampling) | 3 |
| | 2.3 | การจัดระดับ (quantizing) | 3 |
| | 2.4 | คอมแพนดิ้ง (companding) | 4 |
| | 2.5 | การเข้ารหัส (coding) | 6 |
| | 2.6 | การเข้ารหัสสัญญาณเสียงแบบอแด็พทิฟว์ดิฟเฟอเรนเชียลพัลส์โค๊ดมอดูเลชั่น (Adaptive Differential Pulse Code Modulation : ADPCM) | 12 |
| | | 2.6.1 การจัดระดับแบบอแด็พทิฟว์ (Adaptive quantizing) | 14 |
| | | 2.6.2 การเข้ารหัสเอดีพีซีเอ็ม (ADPCM encoder) | 14 |
| | | 2.6.3 การถอดรหัสเอดีพีซีเอ็ม (ADPCM decoder) | 16 |
| | 2.7 | โปรโตคอลทีซีพี/ไอพี | 16 |
| | | 2.7.1 เชื่อมโยงเครือข่าย | 16 |
| | | 2.7.2 ความหมายของโปรโตคอล | 17 |
| | | 2.7.3 โปรโตคอลทีซีพี/ไอพี | 17 |
| | | 2.7.4 ทีซีพี/ไอพีและอินเตอร์เน็ต | 18 |
| | 2.8 | สถาปัตยกรรมทีซีพี/ไอพี | 18 |
| | | 2.8.1 แบบอ้างอิงทีซีพี/ไอพี | 18 |
| | | 2.8.2 โปรโตคอลสแตค | 19 |
| | | 2.8.3 การส่งถ่ายข้อมูลระหว่างชั้น | 20 |
| | | 2.8.4 ไอพีแอดเดรส | 21 |
| | | 2.8.5 ระบบชื่อโดเมน | 21 |
| | | 2.8.6 โปรโตคอลในทีซีพี/ไอพี | 23 |
| | | 2.8.7 ทีซีพี/ไอพี ไคลเอ็นต์-เซอร์ฟเวอร์ | 26 |
| | | 2.8.8 ไอพียุคใหม่ | 28 |
| | 2.9 | ทีซีพี | 28 |
| | | 2.9.1 โปรโตคอลทีซีพี | 28 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เรื่อง | | | หน้า |
|---|---|---|---|
| | 2.9.2 | บริการรับประกันความเชื่อถือของทีซีพี | 30 |
| | 2.9.3 | ทีซีพีเฮดเดอร์ | 30 |
| | 2.9.4 | หน้าที่ของฟิลด์ที่สำคัญในเฮดเดอร์ | 32 |
| | 2.9.5 | กลไกการทำงานของทีซีพี | 35 |
| | 2.9.6 | การคำนวณผลรวมตรวจสอบ | 38 |
| | 2.9.7 | ทีซีพีออพชั่น | 39 |
| | 2.9.8 | การควบคุมกระแสข้อมูล | 41 |
| บทที่ 3 | การคำนวณและการสร้าง | | |
| | 3.1 | ส่วน Sound Recording | 39 |
| | 3.2 | ส่วน Compression (Encoding) | 39 |
| | 3.3 | ส่วน Send&Receive File Via TCP/IP | 43 |
| | 3.4 | ส่วน Decompression (Decoding) | 44 |
| | 3.5 | ส่วน Playback Sound | 45 |
| บทที่ 4 | การทดลองและผลการทดลอง | | |
| | 4.1 | การทดลอง | 53 |
| | 4.1.1 | การทดลองเพื่อหาค่าเริ่มต้นของโปรแกรมที่ดีที่สุด | 64 |
| | 4.1.2 | การหาค่าเฉลี่ยของความผิดพลาดที่เกิดขึ้นในการทำงานของโปรแกรม | 64 |
| | 4.1.3 | การนำเอากรณีที่ค่าเริ่มต้นของโปรแกรมให้ค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดมาเก็บผลการทดลอง | 65 |
| | 4.2 | ผลการทดลอง | |
| | 4.2.1 | ผลการทดลองของ 4.1.1 การทดลองหาค่าเริ่มต้นที่ดีที่สุด | 66 |
| | 4.2.2 | ผลการทดลองของ 4.1.3 การนำเอากรณีที่ค่าเริ่มต้นของโปรแกรมให้ค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดมาใช้ | 83 |
| | 4.2.3 | แสดงในรูปแบบของกราฟแสดงการวิเคราะห์ทางความถี่ที่ความถี่ต่างๆ | 102 |
| | 4.2.4 | ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณที่ด้านรับ | 121 |
| | 4.2.5 | ค่า Minimum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 124 |
| | 4.2.6 | ค่า Maximum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 126 |
| | 4.2.7 | ค่า Average RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 128 |
| บทที่ 5 | บทวิจารณ์และบทสรุป | | |
| | 5.1 | บทวิจารณ์ | 130 |
| | 5.2 | บทสรุป | 131 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป


| รูป | | หน้า |
|---|---|---|
| รูปที่ 2.1 | กระบวนการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม | 2 |
| รูปที่ 2.2 | การแซมปลิ้งสัญญาณเสียง | 3 |
| รูปที่ 2.3 | การจัดระดับสัญญาณพีเอเอ็ม | 4 |
| รูปที่ 2.4 | การแทรกขั้นตอนของการอัดสัญญาณและการยืดสัญญาณลงในระบบพีซีเอ็ม | 5 |
| รูปที่ 2.5 | คุณลักษณะคอมเพรสชั่นของไดโอด | 5 |
| รูปที่ 2.6 | แบบอย่างคุณลักษณะของการคอมเพรสชั่น | 6 |
| รูปที่ 2.7 | คุณลักษณะของคอมแพนดิ้งเมื่อ A=87.6 | 7 |
| รูปที่ 2.8 | บล็อกไดอะแกรมของระบบเอดีพีซีเอ็ม | 13 |
| รูปที่ 2.9 | การเชื่อมโยง (ก) ในเครือข่ายเดียวกัน (ข) ต่างเครือข่ายกัน | 17 |
| รูปที่ 2.10 | แบบอ้างอิงทีซีพี/ไอพี | 19 |
| รูปที่ 2.11 | โปรโตคอลสแตคของทีซีพี/ไอพี | 19 |
| รูปที่ 2.12 | การห่อหุ้มข้อมูลตามลำดับโปรโตคอลสแตค | 21 |
| รูปที่ 2.13 | โครงสร้างลำดับชั้นของระบบโดเมน | 22 |
| รูปที่ 2.14 | เอฟทีพีไคลเอ็นต์-เซอร์ฟเวอร์ | 27 |
| รูปที่ 2.15 | ตัวอย่างแบบจำลองไคลเอ็นต์-เซอร์ฟเวอร์ในทีซีพี/ไอพี | 27 |
| รูปที่ 2.16 | การเอนแคปซูเลตทีซีพี | 29 |
| รูปที่ 2.17 | แบบจำลองการลำเลียงข้อมูลในทีซีพี | 29 |
| รูปที่ 2.18 | ทีซีพีเฮดเดอร์ | 30 |
| รูปที่ 2.19 | การส่งโดยใช้เลขลำดับ | 33 |
| รูปที่ 2.20 | การตอบรับด้วยเลขลำดับ | 34 |
| รูปที่ 2.21 | การส่งซ้ำหากไม่ได้รับการตอบรับ | 34 |
| รูปที่ 2.22 | การสถาปนาการเชื่อมโยงด้วยทีซีพี | 36 |
| รูปที่ 2.23 | ขั้นตอนการถ่ายโอนข้อมูล | 37 |
| รูปที่ 2.24 | การถ่ายโอนเซกเมนต์ทั้งสองทิศทาง | 37 |
| รูปที่ 2.25 | การปิดการเชื่อมต่อ | 38 |
| รูปที่ 2.26 | ฟิลด์ที่ใช้คำนวณค่าผลรวมตรวจสอบในทีซีพี | 39 |
| รูปที่ 2.27 | ทีซีพีออพชัน | 40 |
| รูปที่ 2.28 | การส่งถ่ายแบบฮาล์ฟดูเพล็กซ์ | 41 |
| รูปที่ 2.29 | ฟิลด์ในทีซีพีเฮดเดอร์ที่ใช้ควบคุมกระแสข้อมูล | 42 |
| รูปที่ 2.30 | หน้าต่างของฝ่ายส่ง | 42 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป หน้า


รูปที่ 2.31 การเลื่อนช่องหน้าต่าง 43
รูปที่ 2.32 หน้าต่างฝ่ายรับ 43
รูปที่ 2.33 การเลื่อนหน้าต่างฝ่ายรับ 44
รูปที่ 2.34 การประกาศขนาดหน้าต่าง 45
รูปที่ 2.35 การส่งเซกเมนต์อย่างต่อเนื่อง 46
รูปที่ 3.1 บล็อกไดอะแกรมโดยรวมของโปรแกรม 47
รูปที่ 3.2 บล็อกไดอะแกรมการเข้ารหัสแบบเอดีพีซีเอ็ม 49
รูปที่ 3.3 แผนผังบล็อกการเข้ารหัส (Encoder) 51
รูปที่ 3.4 บล็อกไดอะแกรมการถอดรหัสแบบเอดีพีซีเอ็ม 54
รูปที่ 3.5 แสดงฟอร์มของ unit NetVoiceP 56
รูปที่ 3. 6 แสดงฟอร์มของ unit NetVoiceP เมื่อพร้อมทำการส่ง 57
รูปที่ 3. 7 แสดงฟอร์มของ unit NetVoicePlayP 57
รูปที่ 3.8 แผนผังแสดงการเข้ารหัสแบบเอดีพีซีเอ็ม 60
รูปที่ 3.9 แผนผังแสดงการถอดรหัสแบบเอดีพีซีเอ็ม 63
รูปที่ 4.1 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1 66
รูปที่ 4.2 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2 66
รูปที่ 4.3 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3 67
รูปที่ 4.4 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4 67
รูปที่ 4.5 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5 68
รูปที่ 4.6 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6 68
รูปที่ 4.7 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7 69
รูปที่ 4.8 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8 69
รูปที่ 4.9 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9 70
รูปที่ 4.10 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างที่ความถี่ 200 เฮิรตซ์ 71
รูปที่ 4.11 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1 71
รูปที่ 4.12 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2 72
รูปที่ 4.13 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3 72
รูปที่ 4.14 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4 73
รูปที่ 4.15 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5 73
รูปที่ 4.16 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6 74
รูปที่ 4.17 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7 74
รูปที่ 4.18 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8 75



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป หน้า


| | | |
|---|---|---|
| รูปที่ 4.19 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9 | 75 |
| รูปที่ 4.20 | กราฟแสดงค่าเปอร์เซนต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 2000 เฮิรตซ์ | 76 |
| รูปที่ 4.21 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1 | 77 |
| รูปที่ 4.22 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2 | 77 |
| รูปที่ 4.23 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3 | 78 |
| รูปที่ 4.24 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4 | 78 |
| รูปที่ 4.25 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5 | 79 |
| รูปที่ 4.26 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6 | 79 |
| รูปที่ 4.27 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7 | 80 |
| รูปที่ 4.28 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8 | 80 |
| รูปที่ 4.29 | แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9 | 81 |
| รูปที่ 4.30 | กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 3800 เฮิรตซ์ | 82 |
| รูปที่ 4.31 | กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆเฉลี่ยของความถี่ทั้งสาม | 82 |
| รูปที่ 4.32 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ | 83 |
| รูปที่ 4.33 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 400 เฮิรตซ์ | 84 |
| รูปที่ 4.34 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 600 เฮิรตซ์ | 85 |
| รูปที่ 4.35 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 800 เฮิรตซ์ | 86 |
| รูปที่ 4.36 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1000 เฮิรตซ์ | 87 |
| รูปที่ 4.37 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1200 เฮิรตซ์ | 88 |
| รูปที่ 4.38 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1400 เฮิรตซ์ | 89 |
| รูปที่ 4.39 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1600 เฮิรตซ์ | 90 |
| รูปที่ 4.40 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1800 เฮิรตซ์ | 91 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| รูป | | หน้า |
|---|---|---|
| รูปที่ 4.41 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ | 92 |
| รูปที่ 4.42 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2200 เฮิรตซ์ | 93 |
| รูปที่ 4.43 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2400 เฮิรตซ์ | 94 |
| รูปที่ 4.44 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2600 เฮิรตซ์ | 95 |
| รูปที่ 4.45 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2800 เฮิรตซ์ | 96 |
| รูปที่ 4.46 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3000 เฮิรตซ์ | 97 |
| รูปที่ 4.47 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3200 เฮิรตซ์ | 98 |
| รูปที่ 4.48 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3400 เฮิรตซ์ | 99 |
| รูปที่ 4.49 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3600 เฮิรตซ์ | 100 |
| รูปที่ 4.50 | เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ | 101 |
| รูปที่ 4.51 | เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ | 102 |
| รูปที่ 4.52 | เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 400 เฮิรตซ์ | 103 |
| รูปที่ 4.53 | เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 600 เฮิรตซ์ | 104 |
| รูปที่ 4.54 | เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 800 เฮิรตซ์ | 105 |
| รูปที่ 4.55 | เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1000 เฮิรตซ์ | 106 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป หน้า


รูปที่ 4.56 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1200 เฮิรตซ์ 107

รูปที่ 4.57 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1400 เฮิรตซ์ 108

รูปที่ 4.58 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1600 เฮิรตซ์ 109

รูปที่ 4.59 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1800 เฮิรตซ์ 110

รูปที่ 4.60 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ 111

รูปที่ 4.61 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2200 เฮิรตซ์ 112

รูปที่ 4.62 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2400 เฮิรตซ์ 113

รูปที่ 4.63 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2600 เฮิรตซ์ 114

รูปที่ 4.64 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2800 เฮิรตซ์ 115

รูปที่ 4.65 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3000 เฮิรตซ์ 116

รูปที่ 4.66 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3200 เฮิรตซ์ 117

รูปที่ 4.67 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3400 เฮิรตซ์ 118

รูปที่ 4.68 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3600 เฮิรตซ์ 119

รูปที่ 4.69 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ 120

รูปที่ 4.70 แสดงขนาดของไฟล์ที่ด้านส่งก่อนทำการเข้ารหัส 122

รูปที่ 4.71 แสดงขนาดของไฟล์ที่ด้านส่งหลังจากทำการเข้ารหัสแล้ว 122

รูปที่ 4.72 แสดงขนาดของไฟล์ที่ด้านรับหลังจากทำการถอดรหัสแล้ว 123

รูปที่ 4.73 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณที่ด้านรับ 123



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป หน้า


| | | |
|---|---|---|
| รูปที่ 4.74 | กราฟแสดงค่า Minimum RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง | 125 |
| รูปที่ 4.75 | กราฟแสดงค่า Minimum RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ | 125 |
| รูปที่ 4.76 | กราฟแสดงค่า Maximum RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง | 127 |
| รูปที่ 4.77 | กราฟแสดงค่า Maximum RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ | 127 |
| รูปที่ 4.78 | กราฟแสดงค่า Average RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง | 129 |
| รูปที่ 4.79 | กราฟแสดงค่า Average RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ | 129 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| ตาราง | | หน้า |
|---|---|---|
| ตารางที่ 2.1 | ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม A=87.6 เฉพาะด้านบวก | 8 |
| ตารางที่ 2.2 | ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม A=87.6 เฉพาะด้านลบ | 9 |
| ตารางที่ 2.3 | ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม μ=255 เฉพาะด้านบวก | 10 |
| ตารางที่ 2.4 | ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม μ=255 เฉพาะด้านลบ | 11 |
| ตารางที่ 2.5 | โปรโตคอลมาตรฐานบางส่วนในอินเทอร์เน็ต | 24 |
| ตารางที่ 3.1 | ตารางแสดงค่าอินเด็กซ์แอดจัสต์เมนต์ | 52 |
| ตารางที่ 3.2 | ตารางที่ใช้ในการคำนวณค่าสเต็บไซส์ | 53 |
| ตารางที่ 4.1 | ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 200 เฮิรตซ์ | 70 |
| ตารางที่ 4.2 | ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 2000 เฮิรตซ์ | 76 |
| ตารางที่ 4.3 | ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 3800 เฮิรตซ์ | 81 |
| ตารางที่ 4.4 | ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณเสียงที่ด้านรับ | 121 |
| ตารางที่ 4.5 | ตารางแสดงค่า Minimum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 124 |
| ตารางที่ 4.6 | ตารางแสดงค่า Maximum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 126 |
| ตารางที่ 4.7 | ตารางแสดงค่า Maximum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง | 128 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

การส่งข้อมูลข่าวสารในปัจจุบันนิยมใช้การส่งสัญญาณแบบดิจิตอล เพราะสามารถแก้ปัญหาการลดทอนและการพร่าเพี้ยนได้ และวิธีที่ใช้กันอย่างแพร่หลายก็คือ พัลส์โค๊ดมอดูเลชั่น (Pulse Code Modulation : PCM) ซึ่งถูกนำไปใช้อย่างกว้างขวางโดยเฉพาะกับสัญญาณเสียง แต่ยังมีการเข้ารหัสอีกแบบหนึ่งที่ทำการส่งด้วยรหัสที่น้อยกว่า 8 บิทคือ ระบบการเข้าสัญญาณแบบอแด็พทิฟว์ดิฟเฟอร์เรนเชียลพัลส์โค๊ดมอดูเลชั่น (Adaptive Differential Pulse Code Modulation : ADPCM) วิธีนี้จะใช้ผลต่างระหว่างแซมเปิ้ลที่อยู่ข้างเคียงกันทำให้ประหยัดจำนวนบิทในการส่งสัญญาณและใช้อัตราการส่งที่ต่ำลงได้

นอกจากนั้นการสื่อสารผ่านเครือข่ายอินเตอร์เน็ตก็ทวีความสำคัญเพิ่มขึ้นเป็นลำดับ เพราะอินเตอร์เน็ตช่วยให้การติดต่อสื่อสารระหว่างซีกโลกหนึ่งกับซีกโลกหนึ่งเป็นไปอย่างรวดเร็วโดยแทบปราศจากข้อจำกัดเรื่องเวลาและระยะทาง อินเตอร์เน็ตมีรูปแบบการทำงานเฉพาะตัวภายใต้โปรโตคอล ซึ่งเป็นข้อตกลงร่วมที่กำหนดวิธีสื่อสารระหว่างคอมพิวเตอร์และเชื่อมเครือข่ายเข้าด้วยกัน คอมพิวเตอร์ทุกเครื่องในอินเตอร์เน็ตใช้โปรโตคอล ทีซีพี/ไอพี สื่อสารระหว่างกันหรือกล่าวได้ว่าอินเตอร์เน็ตเป็นเครือข่ายคอมพิวเตอร์ที่เชื่อมกันด้วยทีซีพี/ไอพี

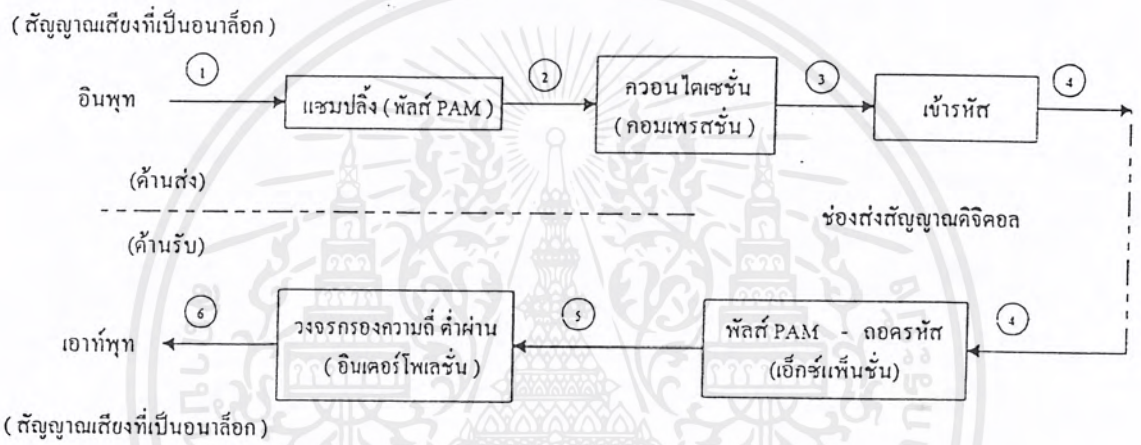
ในปริญญานิพนธ์นี้ได้ทำการศึกษาการบีบอัดเสียงโดยวิธีเอดีพีซีเอ็ม ให้ทำงานร่วมกับการสื่อสารผ่านเครือข่ายอินเตอร์เน็ตโดยใช้โปรโตคอลทีซีพี/ไอพี ซึ่งจะทำให้สามารถทำงานได้หลากหลายมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีหรือหลักการ

## 2.1 การเข้ารหัสของสัญญาณเสียง

วิธีการเข้ารหัสสัญญาณเสียงในระบบสื่อสารแบบดิจิตอลที่ได้รับการพัฒนาและนำมาใช้อย่างกว้างขวางก็คือ พัลส์โค๊ดมอดูเลชั่น (Pulse Code Modulation : PCM) ซึ่งระบบนี้เป็นวิธีการหนึ่ง ในการเปลี่ยนสัญญาณเสียงจากอนาล็อกให้เป็นดิจิตอล โดยจะประกอบด้วยขั้นตอนต่างๆ ดังนี้



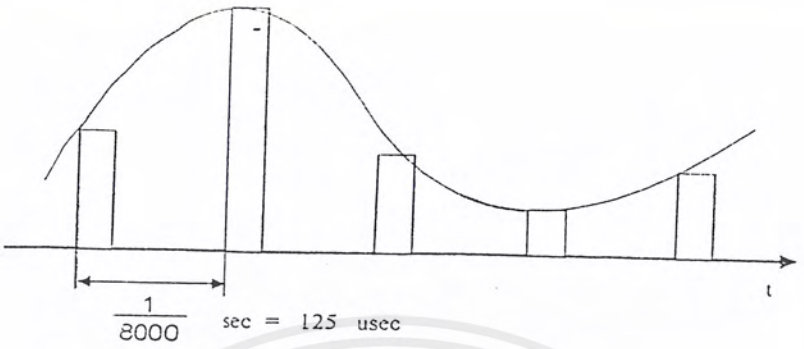

รูปที่ 2.1 กระบวนการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม

จากรูปที่ 2.1 แสดงขั้นตอนการประมวลสัญญาณเพื่อให้ได้สัญญาณพีซีเอ็ม กล่าวอย่างกว้างๆ ก็คือ ระบบพีซีเอ็ม เป็นการจัดการกับสัญญาณพัลส์แอมปลิจูดมอดูเลชั่น (Pulse Amplitude Modulation : PAM) โดยการนำสัญญาณพีเอเอ็ม ไปทำการเข้ารหัส (coding) เป็นสัญญาณดิจิตอล แล้วจึงนำสัญญาณดิจิตอลที่ได้นั้นส่งผ่านระบบต่อไปและทางด้านรับก็จะทำการถอดรหัส (decoding) เป็นสัญญาณพีเอเอ็ม แล้วนำสัญญาณพีเอเอ็ม นั้นไปดีมอดูเลตเพื่อให้ได้รับสัญญาณเดิมกลับคืนมา

ในการส่งโดยระบบพีซีเอ็ม นั้นต้องส่งด้วยจำนวนบิท 8 บิท แต่ยังมีการเข้ารหัสอีกแบบหนึ่งที่ทำการส่งด้วยรหัสน้อยกว่า 8 บิทคือ การเข้ารหัสสัญญาณเสียงแบบอแด็พทิฟว์ดิฟเฟอร์เรนเชียลพัลส์โค๊ดมอดูเลชั่น ซึ่งมีข้อได้เปรียบกว่าระบบพีซีเอ็ม หลายประการดังจะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การแซมปลิ้ง (sampling)

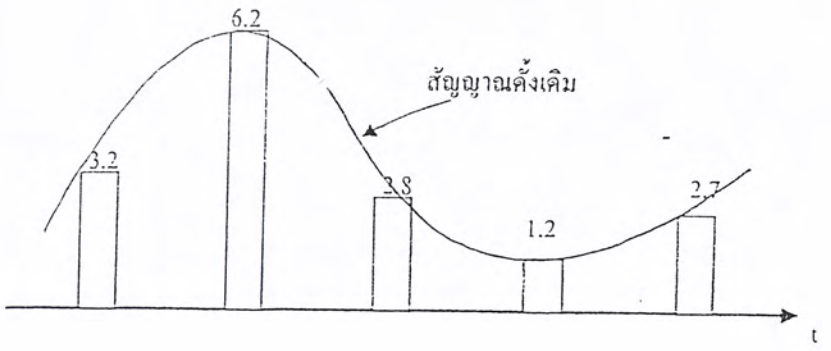



รูปที่ 2.2 การแซมปลิ้งสัญญาณเสียง

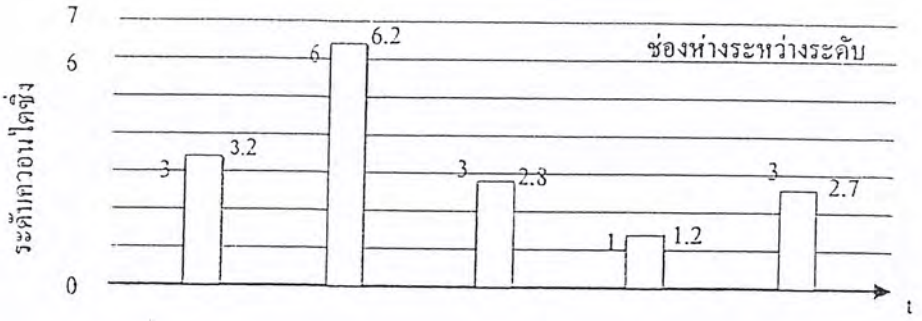
แซมปลิ้ง (sampling) คือ การทำสัญญาณซึ่งมีค่าต่อเนื่องให้เป็นแบบดีสครีทในช่วงเวลาที่เท่าๆ กัน จากทฤษฎีการแซมปลิ้ง ถ้าเก็บแซมเปิ้ล (sample) ด้วยอัตรา 2 เท่าหรือมากกว่าความถี่สูงสุดของสัญญาณอนาล็อกแล้ว จะทำให้สัญญาณเดิมกลับคืนมาได้ เนื่องจากสัญญาณเสียงที่ใช้ในระบบโทรศัพท์นั้นถูกจำกัดให้มีความถี่ระหว่าง 0.3-3.4 kHz. ดังนั้นอัตราการแซมปลิ้งต่ำสุดจะต้องเท่ากับ 6.8 kHz. สำหรับในทางปฏิบัติจะใช้ 8 kHz. คือแซมปลิ้งทุกๆ 125 ไมโครวินาที

## 2.3 การจัดระดับ (quantizing)

ขบวนพัลส์พีเอเอ็ม ที่ผ่านการแซมปลิ้งมาแล้ว ยังถือว่าเป็นชนิดอนาล็อกอยู่ คือจะมีแอมปลิจูดที่เปลี่ยนแปลงอย่างต่อเนื่องไปกับเวลาที่เป็นช่วงๆ การจัดระดับคือ กระบวนการที่เปลี่ยนแอมปลิจูดของพัลส์พีเอเอ็มเหล่านั้นให้เป็นค่าตัวเลขแบบดีสครีท ตามที่แสดงไว้ในรูปที่ 2.3




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.3 การจัดระดับสัญญาณพีเอเอ็ม

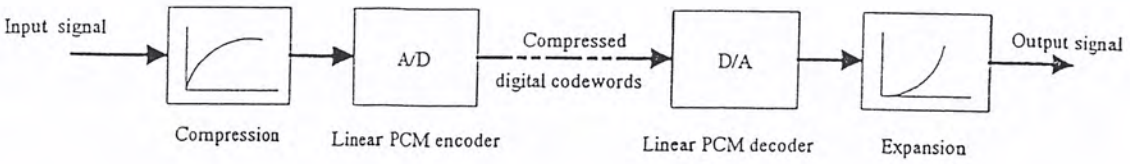
จากรูปที่ 2.3 แอมปลิจูดของแซมเปิ้ลทุกตัวของพีเอเอ็ม จะถูกจัดให้เป็นระดับซึ่งเรียกว่า ระดับการควอนไตซ์ (quantizing level) โดยมีระยะห่างระหว่างระดับข้างเคียง เรียกว่า ควอนไตซ์ซิ่งอินเทอร์วัล (quantizing interval) เท่ากัน กรณีนี้เรียกว่าเป็นการจัดแบบยูนิฟอร์ม (uniform quantizing) ขนาดของแซมเปิ้ลทุกตัวจะแสดงด้วยค่าระดับการควอนไตซ์ที่ใกล้เคียงที่สุด เช่น ขนาดของแซมเปิ้ลที่ $t = t_1$ คือ 3.2 จะจัดให้เป็นระดับ 3 หรือค่าแซมเปิ้ลที่ $t = t_2$ มีขนาด 6.2 จะจัดให้เป็น 6 เป็นต้น จะเห็นได้ว่าสัญญาณพีเอเอ็ม ที่ถูกจัดระดับแล้วนี้จะเป็นเพียงค่าโดยประมาณของสัญญาณอนาล็อกเท่านั้น ดังนั้นส่วนเกินและส่วนขาดจากการจัดระดับจึงเป็นค่าผิดพลาดระหว่างสัญญาณเดิมและค่าที่ได้จัดระดับ ซึ่งค่าผิดพลาดนี้เรียกว่า ควอนไตซ์ซิ่งนอยส์ (quantizing noise) หรือค่าความผิดเพี้ยนจากการควอนไตซ์ (quantizing distortion)

จากหลักการที่กล่าวมานี้ ในทางปฏิบัติจะไม่สามารถหลีกเลี่ยงควอนไตซ์ซิ่งนอยส์ได้ แต่เพื่อรักษาคุณภาพเสียงในการสนทนาให้ดี จึงจำเป็นต้องทำควอนไตซ์ซิ่งนอยส์นี้ให้ลดลง ในเบื้องต้นคือลดควอนไตซ์ซิ่งอินเทอร์วัลให้แคบลง ก็สามารถลดควอนไตซ์ซิ่งนอยส์ได้ในระดับหนึ่ง

## 2.4 คอมแพนดิ้ง (companding)

ตามที่ได้กล่าวมาแล้วว่าเราไม่สามารถหลีกเลี่ยงควอนไตซ์ซิ่งนอยส์ได้ แต่เราสามารถทำให้มันลดลงได้โดยการลดควอนไตซ์ซิ่งอินเทอร์วัล หรือการเพิ่มจำนวนระดับนั่นเอง แต่เมื่อจำนวนระดับเพิ่มขึ้นแล้ว จำนวนบิทที่ใช้ก็จะเพิ่มขึ้นด้วย จึงจำเป็นต้องใช้ความเร็วในการส่งสัญญาณดิจิตอลสูงขึ้น ตามปกติควอนไตซ์ซิ่งนอยส์จะเกิดขึ้นอย่างสม่ำเสมอในทุกอินเทอร์วัล โดยไม่เกี่ยวข้องกับแอมปลิจูดของสัญญาณเดิม ในกรณีที่สัญญาณมีระดับสูง คุณภาพของการเข้ารหัสของสัญญาณเสียงจะดีกว่ากรณีของสัญญาณซึ่งมีระดับต่ำ ดังนั้นจึงจำเป็นต้องพิจารณาควอนไตซ์ซิ่งนอยส์ในบริเวณที่สัญญาณมีระดับต่ำ การจัดระดับแบบนอนยูนิฟอร์ม (non-uniform quantizing) คือบริเวณที่สัญญาณมีแอมปลิจูดต่ำจะใช้ควอนไตซ์ซิ่งอินเทอร์วัลแคบๆ และในทางตรงกันข้ามบริเวณที่สัญญาณมีแอมปลิจูดจะใช้ควอนไตซ์ซิ่งอินเทอร์วัลกว้างๆ ซึ่งการทำให้เป็นแบบนอนยูนิฟอร์มนั้นจะใช้หลักการคอมแพนดิ้งเข้าช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

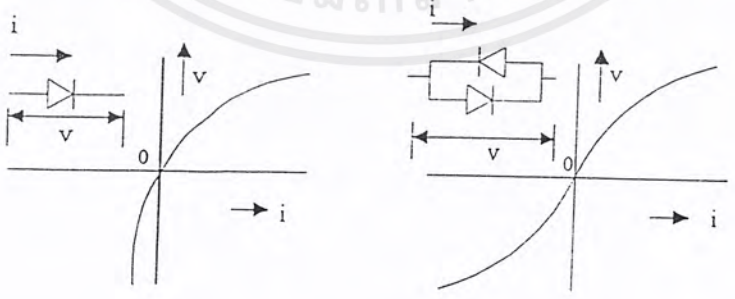



รูปที่ 2.4 การแทรกขั้นตอนของการอัดสัญญาณและการยืดสัญญาณลงในระบบพีซีเอ็ม

การทำคอมแพนดิ้งเมื่อนำมาใช้ในระบบพีซีเอ็ม จะทำให้ประสิทธิภาพในการใช้จำนวนบิทในการเข้ารหัสสูงขึ้น กรณีที่จัดระดับแบบยูนิฟอร์มนั้นจะใช้ประมาณ 2,000 ระดับ จึงจะสามารถรักษาคุณภาพของเสียงให้ดีในการเข้ารหัสจะต้องใช้ถึง 11 บิทต่อแซมเปิ้ล 1 ตัว แต่ถ้าใช้แบบนอนยูนิฟอร์มแล้วจะใช้เพียง 7 บิท ซึ่งมีระดับเพียง 128 ระดับเท่านั้น ก็เพียงพอที่จะทำให้อัตราส่วนของสัญญาณต่อควอนไตซ์ซิ่งนอยส์ใกล้เคียงกับการจัดระดับแบบยูนิฟอร์ม CCITT กำหนดว่าให้ใช้ 8 บิทต่อแซมเปิ้ล 1 ตัว และระดับการควอนไตซ์เท่ากับ 256 ระดับก็จะเป็นการรับรองว่าเสียงพูดจะมีคุณภาพที่ดี

สำหรับคุณลักษณะของการคอมเพรสเซอร์นั้นจะเป็นแบบลอการิทึม รูปแบบโดยทั่วไปจะใช้คุณสมบัติของ V-I ของไดโอด ตามรูปที่ 2.5 กรณีที่ใช้เป็นคอนเพรสเซอร์จะมีกระแส i เป็นอินพุท โวลเตจ v เป็นเอาท์พุท สำหรับกรณีที่ใช้เป็นเอกซ์แพนเดอร์จะมีโวลเตจเป็นอินพุท และกระแสเป็นเอาท์พุท

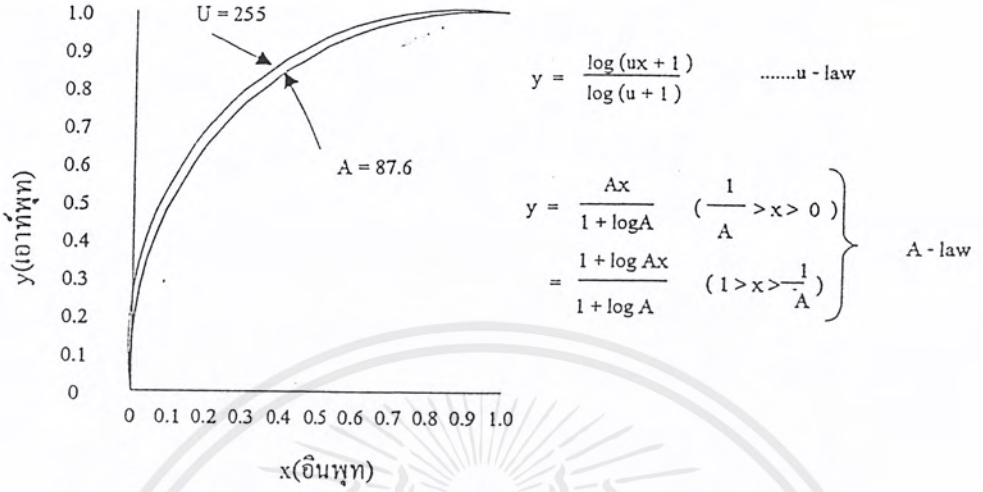
คุณลักษณะของคอมเพรสเซอร์ที่ใช้สำหรับประกอบการเข้ารหัสสัญญาณเสียงในปัจจุบันคือ $\mu$-law
ซึ่งใช้กันในทวีปอเมริการเหนือและประเทศญี่ปุ่นเป็นหลัก โดยที่ค่า $\mu$ ที่ใช้คือ 255 สำหรับในทวีปยุโรปและในประเทศไทยจะใช้ A-law ซึ่งโดยหลักการแล้วจะเหมือนกับกฎ $\mu$-law แต่จะแตกต่างกันในรายละเอียด โดยค่า A ที่ใช้คือ 87.6 คุณลักษณะทั้งสองแบบนี้แสดงไว้ในรูปที่ 2.6 เฉพาะกรณี $\mu$=255 และ A=87.6




(ก) ไดโอดตัวเดียว (ข) ไดโอดหลายตัว

รูปที่ 2.5 คุณลักษณะคอมเพรสชั่นของไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



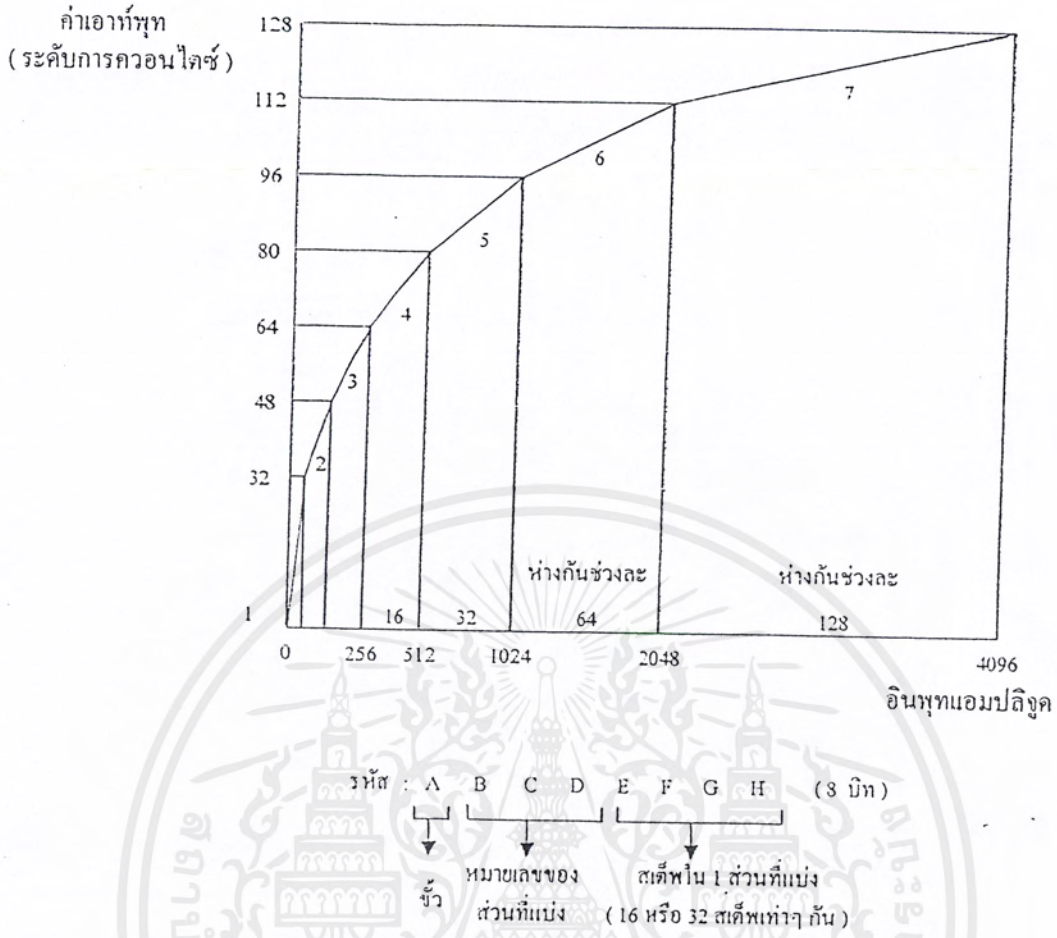

รูปที่ 2.6 แบบอย่างคุณลักษณะของการคอมเพรสชั่น

กรณีที่ $\mu$=100 จะใช้วงจรคอมเพรสเซอร์ตามรูปที่ 2.5 แต่กรณีที่ $\mu$= 255 และ A=87.6 จะใช้วงจรคอมเพรสเซอร์ที่มีคุณลักษณะเป็นเส้นตรง โดยแยกเป็นส่วนๆเรียกว่า เซกเมนต์ (segment)

## 2.5 การเข้ารหัส (coding)

ในการจัดแบ่งเซกเมนต์ในกรีที่ใช้ $\mu$=255 จะแบ่งเป็น 15 เซกเมนต์ สำหรับกรณีที่ใช้ A=87.6 จะแบ่งเป็น 13 เซกเมนต์ โดยเซกเมนต์ที่ 1 จะเป็นเส้นตรงผ่านจุดเริ่มต้นไปทั้งทางบวกและทางลบ สำหรับคุณลักษณะของ A-law นั้นแสดงไว้ในรูปที่ 2.7 ซึ่งแสดงคุณลักษณะเฉพาะด้านบวกเท่านั้น ดังนั้นจะต้องใช้สัญญาณไบนารี 4 บิท มาแสดงตำแหน่งของเซกเมนต์นั้น เมื่อทำการแบ่งเป็นเซกเมนต์แล้วการควอนไตซ์สัญญาณภายในแต่ละเซกเมนต์ก็จะสามารถใช้ควอนไตซ์เป็นเชิงเส้นได้ เนื่องจากระบบพีซีเอ็มใช้การเข้ารหัสด้วยสัญญาณไบนารี 8 บิท 4 บิทบนต้องใช้แสดงตำแหน่งของเซกเมนต์ จึงเหลือเพียง 4 บิทล่างไว้แสดงระดับภายในแตละเซกเมนต์ นั่นคือในแต่ละเซกเมนต์ก็จะแบ่งออกเป็น 16 ขั้นได้ และเมื่อใช้ควอนไตซ์เซอร์แบบเป็นเชิงเส้นแต่ละขั้นมีค่าเท่ากัน การแสดงตำแหน่งของเซกเมนต์นั้นจะใช้บิทบนสุดแสดงว่าเซกเมนตีนั้นอยู่ในซีกบวกหรือซีกลบ และใช้ 3 บิทถัดไปแสดงลำดับของเซกเมนต์ในซีกนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.7 คุณลักษณะของคอมแพนดิ้งเมื่อ A=87.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Segment number | Number of intervals * interval size | Value at segment end points | Decision value number n | Decision value $x_n$ (see Note 1) | Character signal before inversion of the even bits<br>Bit number 1 2 3 4 5 6 7 8 | Quantized value(value at decoder output ) $y_n$ | Decoder output value number |
| | | 4096 | (128) | (4096) | | | |
| 7 | 16 * 128 | | 127<br>113 | 3968<br>2176 | 1 1 1 1 1 1 1 1<br>see Note 2<br>1 1 1 1 0 0 0 0 | 4032 | 128 |
| 6 | 16 * 64 | 2048 | 112<br>97 | 2048<br>1088 | see Note 2<br>1 1 1 0 0 0 0 0 | 2112 | 113 |
| 5 | 16 * 32 | 1024 | 96<br>81 | 1024<br>544 | see Note 2<br>1 1 0 1 0 0 0 0 | 1056 | 97 |
| 4 | 16 * 16 | 512 | 80<br>65 | 512<br>272 | see Note 2<br>1 1 0 0 0 0 0 0 | 528 | 81 |
| 3 | 16 * 8 | 256 | 64<br>49 | 256<br>136 | see Note 2<br>1 0 1 1 0 0 0 0 | 264 | 65 |
| 2 | 16 * 4 | 128 | 48<br>33 | 128<br>68 | see Note 2<br>1 0 1 0 0 0 0 0 | 132 | 49 |
| 1 | 32 * 2 | 64 | 32<br>1 | 64<br>2 | see Note 2<br>1 0 0 0 0 0 0 0 | 66<br>1 | 33<br>1 |
| | | | 0 | 0 | | | |

Note 1 - 4096 normalized value units correspond to $T_{max}$ = 3.14 dBm0

Note 2 - The character signals are obtained by inverting the even bits of the signals of column 6. Before this inversion, the character signal corresponding to positive input values between two successive decision values numbered n and n+1 (see column 4) is (128 + n) expressed as a binary number

Note 3 - The value at the decoder output is $y_n = \frac{x_{n-1} + x_n}{2}$ for n = 1,....,127,128.

Note 4 - $x_{128}$ is a virtual decision value.

**ตารางที่ 2.1 ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม A=87.6 เฉพาะด้านบวก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Segment number | Number of intervals * interval size | Value at segment end points | Decision value number n | Decision value $x_n$ (see Note 1) | Character signal before inversion of the even bits<br>Bit number<br>1 2 3 4 5 6 7 8 | Quantized value(value at decoder output) $y_n$ | Decoder output value number |
| | | | - | | | | |
| | | | 0 | 0 | 0 0 0 0 0 0 0 0 | -1 | 1 |
| ↑ | | | 1 | -2 | | | |
| 1 | 32 * 2 | | | | see Note 2 | | |
| | | -64 | 32 | -64 | 0 0 1 0 0 0 0 0 | -66 | 33 |
| | | | 33 | -68 | | | |
| 2 | 16 * 4 | | | | see Note 2 | | |
| | | -128 | 48 | -128 | 0 0 1 1 0 0 0 0 | -132 | 49 |
| | | | 49 | -136 | | | |
| 3 | 16 * 8 | | | | see Note 2 | | |
| | | -256 | 64 | -256 | 0 1 0 0 0 0 0 0 | -264 | 65 |
| | | | 65 | -272 | | | |
| 4 | 16 * 16 | | | | see Note 2 | | |
| | | -512 | 80 | -512 | 0 1 0 1 0 0 0 0 | -528 | 81 |
| | | | 81 | -544 | | | |
| 5 | 16 * 32 | | | | see Note 2 | | |
| | | -1024 | 96 | -1024 | 0 1 1 0 0 0 0 0 | -1056 | 97 |
| | | | 97 | -1088 | | | |
| 6 | 16 * 64 | | | | see Note 2 | | |
| | | -2048 | 112 | -2048 | 0 1 1 1 0 0 0 0 | -2112 | 113 |
| | | | 113 | -2176 | | | |
| 7 | 16 * 128 | | | | see Note 2 | | |
| | | | 127 | -3968 | 0 1 1 1 1 1 1 1 | -4032 | 128 |
| | | -4096 | (128) | (-4096) | | | |

Note 1 - 4096 normalized value units correspond to $T_{max}$ = 3.14 dBm0

Note 2 - The character signals are obtained by inverting the even bits of the signals of column 6. Before this inversion, the character signal corresponding to positive input values between two successive decision values numbered n and n+1 (see column 4) is (128 + n) expressed as a binary number

Note 3 - The value at the decoder output is $y_n = \frac{x_{n-1} + x_n}{2}$ for n = 1,...,127,128.

Note 4 - $x_{128}$ is a virtual decision value.

**ตารางที่ 2.2 ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม A=87.6 เฉพาะด้านลบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Segment number | Number of intervals * interval size | Value at segment end points | Decision value number n | Decision value $x_n$ (see Note 1) | Character signal before inversion of the even bits<br>Bit number 1 2 3 4 5 6 7 8 | Quantized value(value at decoder output) $y_n$ | Decoder output value number |
| | | 8159 | (128) | (8159) | | | |
| 8 | 16 * 256 | | 127 | 7903 | 1 0 0 0 0 0 0 0 | 8031 | 127 |
| | | | | | see Note 2 | | |
| | | | 113 | 4319 | | | |
| 7 | 16 * 128 | 4063 | 112 | 4063 | 1 0 0 0 1 1 1 1 | 4191 | 112 |
| | | | | | see Note 2 | | |
| | | | 97 | 2143 | | | |
| 6 | 16 * 64 | 2015 | 96 | 2015 | 1 0 0 1 1 1 1 1 | 2079 | 96 |
| | | | | | see Note 2 | | |
| | | | 81 | 1055 | | | |
| 5 | 16 * 32 | 991 | 80 | 991 | 1 0 1 0 1 1 1 1 | 1023 | 80 |
| | | | | | see Note 2 | | |
| | | | 65 | 511 | | | |
| 4 | 16 * 16 | 479 | 64 | 479 | 1 0 1 1 1 1 1 1 | 495 | 64 |
| | | | | | see Note 2 | | |
| | | | 49 | 239 | | | |
| 3 | 16 * 8 | 223 | 48 | 223 | 1 1 0 0 1 1 1 1 | 231 | 48 |
| | | | | | see Note 2 | | |
| | | | 33 | 103 | | | |
| 2 | 16 * 4 | 95 | 32 | 95 | 1 1 0 1 1 1 1 1 | 99 | 32 |
| | | | | | see Note 2 | | |
| | | | 17 | 35 | | | |
| 1 | 15 * 2 | 31 | 16 | 31 | 1 1 1 0 1 1 1 1 | 33 | 16 |
| | | | | | see Note 2 | | |
| | | | 2 | 3 | | | |
| | 1 * 1 | | 1 | 1 | 1 1 1 1 1 1 1 0 | 2 | 1 |
| | | | 0 | 0 | 1 1 1 1 1 1 1 1 | 0 | 0 |

Note 1 - 8159 normalized value units correspond to $T_{max}$ = 3.17 dBm0

Note 2 - The character signals correspondind to negative input values between two successive decision values numbered n and n+1 (see column 4) is (127 - n)expressed as a binary number for n = 0,1,...,127

Note 3 - The value at the decoder output is $y_0 = x_0 = 0$ for n = 0,and $y_n = \frac{x_n + x_{n-1}}{2}$ for n = 1,2,...,127.

Note 4 - $x_{128}$ is a virtual decision value.

**ตารางที่ 2.3 ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม μ=255 เฉพาะด้านบวก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Segment number | Number of intervals * interval size | Value at segment end points | Decision value number n | Decision value $x_n$ (see Note 1) | Character signal before inversion of the even bits<br>Bit number<br>1 2 3 4 5 6 7 8 | Quantized value(value at decoder output) $y_n$ | Decoder output value number |
| | | | 0 | 0 | | 0 | 0 |
| | 1 * 1 | | 1 | -1 | 0 1 1 1 1 1 1 1 | | |
| ▲ | | | | | 0 1 1 1 1 1 1 0 | -2 | 1 |
| 1 | 15 * 2 | | | -3 | see Note 2 | | |
| | | -31 | 16 | -31 | 0 1 1 0 1 1 1 1 | -33 | 16 |
| 2 | 16 * 4 | | 17 | -35 | see Note 2 | | |
| | | -95 | 32 | -95 | 0 1 0 1 1 1 1 1 | -99 | 32 |
| 3 | 16 * 8 | | 33 | -103 | see Note 2 | | |
| | | -223 | 48 | -223 | 0 1 0 0 1 1 1 1 | -231 | 48 |
| 4 | 16 * 16 | | 49 | -239 | see Note 2 | | |
| | | -479 | 64 | -479 | 0 0 1 1 1 1 1 1 | -495 | 64 |
| 5 | 16 * 32 | | 65 | -511 | see Note 2 | | |
| | | -991 | 80 | -991 | 0 0 1 0 1 1 1 1 | -1023 | 80 |
| 6 | 16 * 64 | | 81 | -1055 | see Note 2 | | |
| | | -2015 | 96 | -2015 | 0 0 0 1 1 1 1 1 | -2079 | 96 |
| 7 | 16 * 128 | | 97 | -2143 | see Note 2 | | |
| | | -4063 | 112 | -4063 | 0 0 0 0 1 1 1 1 | -4191 | 112 |
| | | | 113 | -4319 | see Note 2 | | |
| 8 | 16 * 256 | | 126 | -7647 | 0 0 0 0 0 0 0 1 | -7775 | 126 |
| | | | 127 | -7903 | 0 0 0 0 0 0 0 0 | -8031 | 127 |
| | | -8159 | (128) | -8159 | | | |

Note 1 - 8159 normalized value units correspond to $T_{max}$ = 3.17 dBm0

Note 2 - The character signals correspondind to negative input values between two successive decision values numbered n and n+1 (see column 4) is (127 - n)expressed as a binary number for n = 0,1,...,127

Note 3 - The value at the decoder output is $y_0 = x_0 = 0$ for n = 0,and $y_n = \frac{x_n + x_{n+1}}{2}$ for n = 1,2,...,127.

Note 4 - $x_{128}$ is a virtual decision value.

**ตารางที่ 2.4 ตารางการเข้ารหัสและถอดรหัสของระบบพีซีเอ็ม μ=255 เฉพาะด้านลบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การเข้ารหัสสัญญาณเสียงแบบอแด็พทิฟว์ดิฟเฟอร์เรนเชียลพัลส์โค๊ดมอดูเลชั่น (Adaptive Differential Pulse Code Modulation : ADPCM)

ในการใช้งานพีซีเอ็ม นั้นมีอัตราการส่งมาตรฐานคือ 64 Kbit/s ซึ่งต้องการช่องสัญญาณในการส่งที่มีแบนด์วิดธ์กว้างในการนำไปใช้งาน เช่น งานเกี่ยวกับความปลอดภัยของสัญญาณเสียงในการส่งไปในสายส่งบนช่องสัญญาณวิทยุซึ่งมีความจุต่ำ การประยุกต์ใช้งานแบบนี้จะเป็นที่ต้องการสำหรับการเข้ารหัสสัญญาณเสียงที่มีอัตราการส่งต่ำ ในขณะที่สามารถดูแลการรับส่งและได้สัญญาณเสียงที่มีคุณภาพกลับออกมา สำหรับการเข้ารหัสสัญญาณที่มีอัตราการส่งต่ำนั้นต้องอาศัยสถิติของคุณลักษณะของรูปคลื่นของสัญญาณเสียงและคุณสมบัติการได้ยิน การออกแบบนี้เพื่อจุดประสงค์ 2 ประการ คือ

1. เพื่อเคลื่อนสัญญาณเสียงไปในระยะทางที่ไกลเท่าที่จะทำได้
2. เพื่อกำหนดบิทในการเข้ารหัสสัญญาณเสียงให้สามารถเข้าใจได้เมื่อถอดรหัส

จากการศึกษาได้พบว่าการเข้ารหัสสัญญาณแบบดีพีซีเอ็ม เป็นวิธีการบีบอัดสัญญาณวิธีหนึ่งที่ถูกนำมาใช้ในการลดจำนวนบิทของการเข้ารหัสสัญญาณ แต่ความต้องการในการลดจำนวนบิทในการเข้ารหัสสัญญาณนั้นยังคงมีอยู่จึงได้เกิดระบบการเข้ารหัสสัญญาณเสียงอีกแบบหนึ่งขึ้นมาซึ่งก็คือ ระบบการเข้ารหัสสัญญาณเสียงแบบอแด็พทิฟว์ดิฟเฟอร์เรนเชียลพัลส์โค๊ดมอดูเลชั่น ที่มีอัตราการส่งข้อมูล 32 Kbit/s ซึ่งสามารถเพิ่มความจุของช่องสัญญาณเป็น 2 เท่า เมื่อเทียบกับระบบพีซีเอ็ม ในปัจจุบันที่มีอัตราการส่งข้อมูล 64 Kbit/s

หลักการของระบบเอดีพีซีเอ็ม นั้นจะคล้ายคลึงกับระบบดีพีซีเอ็ม แต่จะมีการเพิ่มคุณสมบัติในการปรับขั้นของการควอนไตซ์ และวิธีการเปรียบเทียบสัญญาณเข้าไปในระบบดีพีซีเอ็ม ซึ่งทำให้สามารถลดจำนวนบิทในการเข้าสัญญาณลงได้มากขึ้น มาตรฐานของระบบเอดีพีซีเอ็ม จะอธิบายถึงการคอมเพรซซิ่ง และเอ็กซ์แพนดิ้ง ค่าความยาวของแซมเปิ้ลจาก 8 บิท จะลดลงเหลือ 3, 4 หรือ 5 บิท โดยมีอัตราการบีบอัดเป็น 2.67, 2 และ 1.6 ตามลำดับ และมีอัตราการส่งสัญญาณเท่ากับ 24, 32 และ 40 Kbit/s ตามลำดับ เมื่อมีอัตราการแซมปลิ้งเท่ากับ 8 kHz ในระบบเอดีพีซีเอ็ม จะเพิ่มคุณสมบัติในการปรับขั้นของการควอนไตซ์โดยใช้ควอนไตซ์เซอร์ เป็นแบบอแด็พทิฟว์ ควอนไตซ์เซอร์(adaptive quantizer) แทนการใช้แบบนอนยูนิฟอร์ม ควอนไตซ์เซอร์ (non-uniform quantizer) ธรรมดา ซึ่งแบบอแด็พทิฟว์ นอกจากค่าระดับการควอนไตซ์ จะเปลี่ยนตามค่าระดับของสัญญาณแบบนอนยูนิฟอร์มแล้ว ค่าสเต็บไซส์ของระดับของการควอนไตซ์ยังปรับได้ตามค่าระดับของสัญญาณผลต่างที่เข้ามาอีกด้วย คือเมื่อระดับของสัญญาณผลต่างมีค่าสูง ค่าสเต็บไซส์ ก็จะปรับเป็นค่าสูง แต่ถ้าระดับของผลต่างมีค่าต่ำ ค่าสเต็บไซส์ก็จะปรับเป็นค่าต่ำด้วย เพื่อให้จำนวนบิทที่ใช้ในการเข้ารหัสสัญญาณสามารถครอบคลุมค่าระดับของผลต่างของสัญญาณได้ทั้งหมด ส่วนการเปรียบเทียบสัญญาณจะใช้อแด็พทิฟว์ พรีดิกเตอร์ (adaptive predictor) เป็นตัวหาค่าประมาณของสัญญาณอินพุทเพื่อนำมาใช้เปรียบเทียบกับสัญญาณอินพุทที่เข้ามาตัวถัดไป

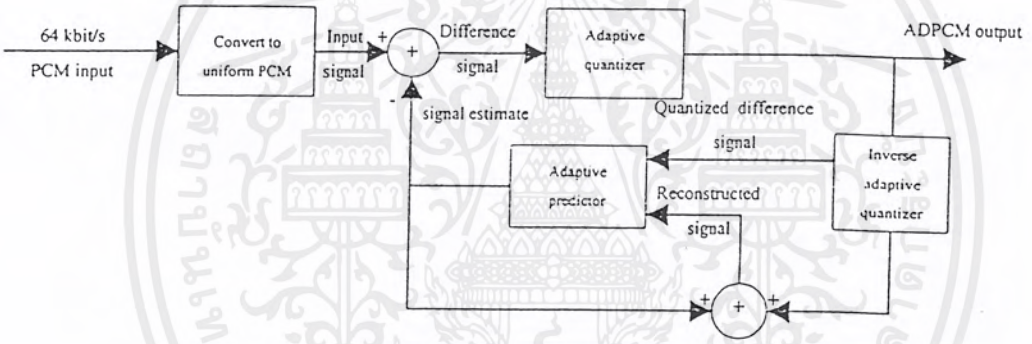
ข้อได้เปรียบของระบบการเข้ารหัสสัญญาณแบบเอดีพีซีเอ็ม ที่เหนือกว่าระบบการเข้ารหัสสัญญาณแบบพีซีเอ็ม และแบบดีพีซีเอ็ม คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

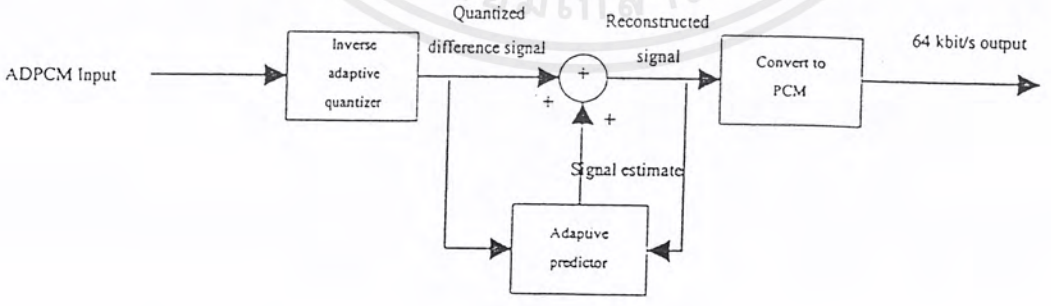
1. สามารถลดจำนวนบิทที่ใช้ในการเข้ารหัสสัญญาณได้มากกว่า เนื่องจากว่าจะใช้การเข้ารหัสจากค่าผลต่างของสัญญาณที่อยู่ข้างเคียงกันซึ่งมีค่าน้อย และยังเพิ่มคุณสมบัติในการปรับขั้นระดับของการควอนไตซ์ตามขนาดของสัญญาณผลต่างเข้าไปด้วย จึงสามารถลดจำนวนบิทที่ใช้เข้ารหัสลงได้มากกว่าระบบพีซีเอ็ม และดีพีซีเอ็ม ซึ่งก็เท่ากับว่าเป็นการลดขนาดของข้อมูลลงด้วย จึงมีประโยชน์ที่ต้องมีการเก็บข้อมูลลงในหน่วยความจำ ทำให้สามารถประหยัดหน่วยความจำที่ใช้ในการเก็บข้อมูลลงได้ และเป็นการลดค่าใช้จ่ายลงอีกด้วย ระบบเอดีพีซีเอ็ม ปัจจุบันได้มีการพัฒนามาใช้กับเสียง ภาพ และข้อมูล

2 มีคุณสมบัติในการใช้สายส่งสัญญาณได้อย่างมีประสิทธิภาพมากกว่าระบบพีซีเอ็ม และระบบดีพีซีเอ็ม เนื่องจากอัตราเร็วที่ใช้ในการส่งสัญญาณจะลดลงเมื่อขนาดของข้อมูลลดลง จึงสามารถเพิ่มจำนวนของสัญญาณที่จะส่งไปในสายส่งได้มากขึ้น

บล็อกไดอะแกรมของระบบการเข้ารหัสสัญญาณแบบเอดีพีซีเอ็ม แสดงได้ดังรูปที่ 2.8




(ก) การเข้ารหัสเอดีพีซีเอ็ม




(ข) การถอดรหัสเอดีพีซีเอ็ม

รูปที่ 2.8 บล็อกไดอะแกรมของระบบเอดีพีซีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.1 การจัดระดับแบบอแด็พทิฟว์ (Adaptive quantizing)

การจัดระดับแบบนอนยูนิฟอร์ม ควอนไตซ์ซิ่ง ธรรมดา ขนาดสเต็บไซส์ ที่ใช้ในการจัดระดับจะถูกกำหนดเอาไว้อย่างแน่นอนจะไม่ขึ้นกับขนาดของผลต่างของสัญญาณที่เข้ามา ไม่ว่าสัญญาณผลต่างจะมีค่าสูงหรือค่าต่ำก็ตามขนาดสเต็บไซส์ก็จะยังคงเท่าเดิมเสมอแต่การจัดระดับแบบอแด็พทิฟว์ ควอนไตซ์ซิ่ง ขนาดของสเต็บไซส์ จะปรับได้ตามขนาดของผลต่างของสัญญาณ เมื่อระดับสัญญาณผลต่างมีค่าสูงขึ้น ขนาดของสเต็บไซส์ ก็จะถูกปรับให้ห่างขึ้น และเมื่อระดับของสัญญาณผลต่างมีค่าต่ำลง ขนาดของสเต็บไซส์ ก็จะถูกปรับให้ต่ำลงตามด้วย เพื่อให้สามารถครอบคลุมระดับของสัญญาณผลต่างได้ทั้งหมด วิธีการนี้จะเป็นการหลีกเลี่ยงการเพิ่มของควอนไตซิ่งนอยส์ได้อย่างมีประสิทธิภาพ และในขณะเดียวกันก็สามารถลดอัตราการส่งข้อมูลลงได้ด้วยแต่ก็ยังใช้ได้ผลดี

### 2.6.2 การเข้ารหัส เอดีพีซีเอ็ม (ADPCM encoder)

ขั้นตอนการทำงานของกระบวนการเข้ารหัสสัญญาณแบบเอดีพีซีเอ็ม แสดงได้ดังรูปที่ 2.8 (ก) สามารถแบ่งขั้นตอนการทำงานออกได้ดังนี้คือ เริ่มแรกต้องทำการแปลงสัญญาณเสียงอินพุท 64 Kbit/s A-law หรือ μ-law พีซีเอ็ม ให้ไปเป็นสัญญาณแบบยูนิฟอร์มพีซีเอ็ม ซึ่งจะได้เป็นค่าแอมปลิจูดของสัญญาณ แล้วจะใช้ค่าแอมปลิจูดนี้เป็นสัญญาณอินพุทของภาคเข้ารหัสสัญญาณ จากนั้นหาค่าผลต่างของสัญญาณ (difference signal) ซึ่งได้จากการลบกันระหว่างค่าคาดคะเนสัญญาณอินพุทปัจจุบัน (signal estimate) กับค่าสัญญาณอินพุทปัจจุบัน

เมื่อกำหนดให้

$d(n)$ = สัญญาณผลต่าง (difference signal or prediction error)

$x(n)$ = สัญญาณอินพุท (input signal)

$x_p(n)$ = ค่าคาดคะเนสัญญาณอินพุทปัจจุบัน (signal estimate)

จะได้ว่า

$$d(n) = x(n) - x_p(n) \qquad \text{.....2.1}$$

ค่าสัญญาณผลต่างที่ได้นี้จะถูกนำไปทำการจัดระดับ (quantization) ซึ่งในระบบเอดีพีซีเอ็ม นี้ จะใช้การจัดระดับแบบอแด็พทิฟว์ ควอนไตซ์ซิ่ง ที่สามารถปรับขนาดของสเต็บไซส์ ได้ตามขนาดของสัญญาณผลต่างทำให้สามารถลดจำนวนบิทที่ใช้ในการเข้ารหัสสัญญาณลงได้ และจำนวนบิทที่ใช้สามารถครอบคลุมขนาดของสัญญาณผลต่างได้ทั้งหมด ซึ่งในโครงงานนี้ได้ทำการศึกษาถึงระบบการเข้ารหัสสัญญาณเสียงแบบเอดีพีซีเอ็ม ที่มีอัตราการบีบอัดข้อมูลเป็น 2:1 คือ จะใช้จำนวนบิทในการเข้ารหัสสัญญาณเป็น 4 บิทต่อ 1 แซมเปิ้ล ที่ความถี่ของการแซมปลิ้ง 8 kHz จะได้อัตราการส่งข้อมูลเท่ากับ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kbit/s เนื่องจากใช้การเข้ารหัสขนาด 4 บิท จึงทำการจัดระดับของการควอนไตซ์ไว้ทั้งหมด 16 ระดับ ค่ารหัสของค่าสัญญาณผลต่างที่ได้นี้ก็คือ รหัสเอดีพีซีเอ็ม ซึ่งเป็นเอาท์พุทของภาคเข้ารหัส เพื่อส่งไปยังปลายทางให้ทำการถอดรหัสให้ได้สัญญาณเสียงที่ใกล้เคียงกับสัญญาณเสียงเดิมกลับมา

ค่ารหัสเอดีพีซีเอ็ม ขนาด 4 บิทนี้จะถูกป้อนเข้าไปยังอินเวอร์ส อแด็พทิฟว์ ควอนไตซ์เซอร์ (inverse adaptive quantizer)เพื่อสร้างสัญญาณผลต่างที่ถูกควอนไตซ์ (quantized difference signal) ที่ใกล้เคียงกับสัญญาณผลต่างตัวเดิมกลับออกมา

เมื่อกำหนดให้

$v(n)$ = quantized difference signal

$q(n)$ = quantization error

จะได้ว่า

$$v(n) = d(n) + q(n) \qquad \text{.....2.2}$$

และเนื่องจากการจัดระดับเป็นแบบอแด็พทิฟว์ การหาค่าสัญญาณผลต่างที่ถูกควอนไตซ์ จึงต้องเป็นแบบอแด็พทิฟว์ด้วย จากนั้นสัญญาณผลต่างที่ถูกควอนไตซ์ นี้ก็จะถูกรวมเข้ากับค่าคาดคะเนสัญญาณอินพุทตัวก่อน (previous signal estimate) เพื่อสร้างสัญญาณที่ถูกสร้างกลับคืนมา (reconstructed signal) ป้อนเข้าไปเป็นอินพุทให้กับอแด็พทิฟว์ พรีดิกเตอร์ ร่วมกับสัญญาณผลต่างที่ถูกควอนไตซ์ เพื่อทำนายค่าคาดคะเนสัญญาณอินพุทตัวถัดไป (signal estimate) ออกมา

เมื่อกำหนดให้

$u(n)$ = reconstructed signal or predictor input

$x_p(n)$ = previous signal estimate

จะได้ว่า

$$u(n) = x_p(n-1) + v(n) \qquad \text{.....2.3}$$

เมื่อแทนสมการที่ 2.2 ลงในสมการที่ 2.3 จะได้ว่า

$$u(n) = x_p(n-1) + d(n) + q(n) \qquad \text{.....2.4}$$

ค่าคาดคะเนสัญญาณอินพุทที่ได้จะนำไปเปรียบเทียบกับสัญญาณอินพุทปัจจุบันที่เข้ามาใหม่ เพื่อหาค่าสัญญาณผลต่างตัวถัดไป แล้วนำไปสร้างเป็นรหัสเอดีพีซีเอ็ม ตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.3 การถอดรหัสเอดีพีซีเอ็ม (ADPCM decoder)

ขั้นตอนการทำงานของกระบวนการถอดรหัสสัญญาณแบบเอดีพีซีเอ็ม แสดงได้ดังรูปที่ 2.8 (ข) โครงสร้างของภาคถอดรหัสสัญญาณจะเหมือนกับส่วนป้อนกลับ (feedback loop) ของสัญญาณของภาคเข้ารหัสสัญญาณ ร่วมกับส่วนที่ทำการแปลงยูนิฟอร์มพีซีเอ็ม ให้เป็น A-law หรือ μ-law พีซีเอ็มซึ่งก็จะมีกระบวนการทำงานเหมือนกับภาคเข้ารหัสสัญญาณนั่นเอง ส่วนที่ทำการแปลงยูนิฟอร์มพีซีเอ็ม ให้เป็น A-law หรือ μ-law พีซีเอ็มจะทำหน้าที่แปลงค่าแอมปลิจูดของสัญญาณให้กลับไปเป็นค่าระดับของการควอนไตซ์ของสัญญาณเพื่อให้ได้สัญญาณในรูปของ 64 Kbit/s พีซีเอ็มซึ่งมีลักษณะใกล้เคียงกับสัญญาณเสียงเดิมกลับคืนมา

## 2.7 โปรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพีเป็นโปรโตคอลที่ได้รับการออกแบบให้เป็นอิสระจากชนิดของคอมพิวเตอร์และระบบปฏิบัติการ ตัวโปรโตคอลมีความเชื่อถือได้สูงและสามารถปรับเปลี่ยนการทำงานตามสภาพเครือข่ายได้ในบางกรณีที่บางเส้นทางชำรุด ซึ่งความรู้เบื้องต้นเกี่ยวกับทีซีพี/ไอพีมีดังนี้

### 2.7.1 การเชื่อมโยงเครือข่าย

จุดประสงค์การเชื่อมโยงคอมพิวเตอร์เข้าเป็นเครือข่ายคือต้องการให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลระหว่างกันได้ เครือข่ายคอมพิวเตอร์เริ่มจากเครือข่ายขนาดเล็กภายในองค์กรที่เชื่อมโยงกันภายใต้สภาพพื้นที่จำกัดซึ่งเรียกว่า เครือข่ายเฉพาะที่ (LAN : Local Area Network) เมื่อเชื่อมเครือข่ายย่อยเข้าด้วยกันและขยายขอบเขตครอบคลุมพื้นที่ระหว่างเมือง หรือระหว่างประเทศ ก็จะเรียกเครือข่ายนั้นว่า เครือข่ายพื้นที่กว้าง (WAN : Wide Area Network)

เครือข่ายยุคเริ่มต้นมีสถานีที่ใช้ฮาร์ดแวร์ประเภทเดียวกันและสามารถทำงานร่วมกันได้อย่างกลมกลืน ต่อมาเมื่อมีเทคโนโลยีเครือข่ายเพิ่มขึ้นเช่น อีเทอร์เน็ตและโทเค็นริง ปัญหาที่เกิดขึ้นคือจะเชื่อมเครือข่ายต่างเทคโนโลยีเข้าด้วยกันได้อย่างไรโดยไม่จำกัดคอมพิวเตอร์จะอยู่ในเครือข่ายเดียวกันดังรูปที่ 2.9 (ก) หรือต่างเครือข่ายกันโดยใช้ฮาร์ดแวร์เครือข่ายต่างกันดังรูป (ข)

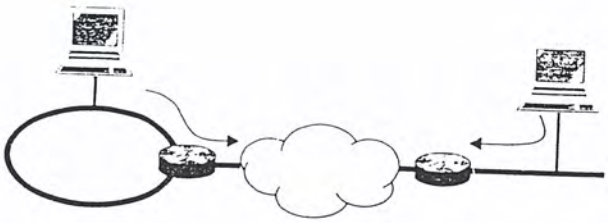
(ก)

(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง





(ข)

(ข)

รูปที่ 2.9 การเชื่อมโยง (ก) ในเครือข่ายเดียวกัน (ข) ต่างเครือข่ายกัน

### 2.7.2 ความหมายของโปรโตคอล

การเชื่อมเครือข่ายต่างฮาร์ดแวร์จำเป็นต้องกำหนดข้อตกลงร่วม หรือ โปรโตคอล (protocol) เพื่อให้คอมพิวเตอร์สื่อสารกันตามข้อกำหนด ทีซีพี/ไอพีจัดเป็นโปรโตคอลหนึ่งที่ออกแบบมาเพื่อแก้ปัญหาการเชื่อมโยงดังกล่าว

โปรโตคอลในความหมายของระบบเครือข่ายคือข้อกำหนดการสื่อสาร คอมพิวเตอร์หรืออุปกรณ์เครือข่ายจะมีซอฟต์แวร์ที่ปฏิบัติงานตามโปรโตคอลที่กำหนดพร้อมทั้งมีกรรมวิธีแก้ไขปัญหาที่เกิดขึ้น เช่นหากข้อมูลที่ขนถ่ายมีข้อผิดพลาด คอมพิวเตอร์จะดำเนินการตามแบบแผนในโปรโตคอลเช่นส่งข้อมูลซ้ำใหม่

ในระบบเครือข่ายขนาดใหญ่อาจมีเส้นทางเชื่อมโยงระหว่างกันได้เป็นจำนวนมากข้อมูลที่ส่งออกไปอาจไม่ได้ใช้เส้นทางเดียวกันตลอด ข้อมูลที่ส่งออกไปก่อนอาจไปถึงปลายทางช้ากว่า กรณีนี้เครื่องหมายปลายทางจำเป็นต้องจัดลำดับข้อมูลใหม่

กรณีที่คอมพิวเตอร์ต้นทางสามารถส่งข้อมูลได้เร็วเกินกว่าปลายทางจะรับได้ทัน โปรโตคอลจะกำหนดกรรมวิธีควบคุมการลำเลียงข้อมูลระหว่างต้นทางและปลายทางให้สัมพันธ์กัน

### 2.7.3 โปรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพีเป็นโปรโตคอลที่ใช้งานอย่างแพร่หลายในแทบทุกเครือข่ายไม่ว่าจะเป็นเครือข่ายเฉพาะที่หรือเครือข่ายในบริเวณกว้าง ทีซีพี/ไอพีเชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ หรือ อินเตอร์เน็ต (Internet)

ทีซีพี/ไอพีผ่านการออกแบบให้เป็นอิสระจากชนิดคอมพิวเตอร์ ฮาร์ดแวร์ และระบบปฏิบัติการ กลไกของโปรโตคอลมีความเชื่อถือได้สูงและทำงานได้แม้บางภาวะที่การสื่อสารมีความผิดปกติ รวมทั้งสามารถเลือกเส้นทางส่งข้อมูลตามสภาพเครือข่ายได้ในกรณีที่บางเส้นทางชำรุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
42202

ชื่อทีซีพี/ไอพีมีที่มาจากโปรโตคอลสองโปรโตคอลคือ ทีซีพี (TCP : Transmission Control Protocol) และไอพี (IP : Internet Protocol) ไอพีทำหน้าที่กำหนดแอดเดรส จัดแบ่งขนาดข้อมูลให้พอเหมาะและเลือกเส้นทางส่งข้อมูล ส่วนทีซีพีมีหน้าที่รับประกันความถูกต้องในการลำเลียงข้อมูล

ทีซีพีและไอพีไม่ได้เป็นเพียงสองโปรโตคอลที่มีอยู่เท่านั้น หากแต่ยังมีโปรโตคอลสนับสนุนอีกเป็นจำนวนมากและจัดรวมกันเป็น ชุดโปรโตคอลทีซีพี/ไอพี (TCP/IP protocol suite)

### 2.7.4 ทีซีพี/ไอพีและอินเตอร์เน็ต

อินเตอร์เน็ตเป็นทั้งเครือข่ายของคอมพิวเตอร์ และเครือข่ายของเครือข่าย เพราะอินเตอร์เน็ตเป็นสังคมเครือข่ายขนาดใหญ่ที่ประกอบด้วยเครือข่ายย่อยจำนวนมากต่อเชื่อมกันด้วยทีซีพี/ไอพี คอมพิวเตอร์ในอินเตอร์เน็ตทุกเครื่องจึงใช้โปรโตคอลทีซีพี/ไอพีเพื่อสื่อสารระหว่างกัน

## 2.8 สถาปัตยกรรมทีซีพี/ไอพี

### 2.8.1 แบบอ้างอิงทีซีพี/ไอพี

ระบบการสื่อสารข้อมูลในเครือข่ายคอมพิวเตอร์ประกอบด้วยทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ซับซ้อน การมองภาพของระบบโดยรวมทั้งหมดเป็นหน่วยใหญ่ย่อมยากต่อการทำความเข้าใจ การใช้แบบอ้างอิงที่แบ่งระบบออกเป็นส่วนย่อยจะช่วยลดความซับซ้อนและสร้างความเข้าใจได้ง่ายกว่า

เครือข่ายคอมพิวเตอร์มีแบบอ้างอิงที่ใช้เป็นมาตรฐานคือ แบบอ้างอิงโอเอสไอ (OSI : Open Systems Interconnection Reference Model) ในขณะที่ทีซีพี/ไอพีเป็นโปรโตคอลที่กำเนิดก่อนโอเอสไอและมีแบบอ้างอิงเฉพาะตามรูปที่ 2.10 แบบอ้างอิงทีซีพี/ไอพีมีระดับชั้นจากล่างขึ้นบนและมีลักษณะสมบัติประจำชั้นต่างๆดังต่อไปนี้

**ฟิสิคัล** ชั้นของการกำหนดคุณสมบัติฮาร์ดแวร์ เช่นคุณสมบัติทางกล (หัวต่อและชนิดสายสื่อสาร) และคุณสมบัติทางไฟฟ้า (ลักษณะสัญญาณและอัตราเร็ว) กล่าวโดยรวมแล้วระดับชั้นฟิสิคัลกำหนดวิธีการถ่ายโอนข้อมูลในระดับบิต ตัวอย่างของการเชื่อมต่อที่ตรงกับระดับชั้นฟิสิคัลได้แก่ RS232 และ X.21 เป็นต้น

**เดทาลิงค์** ชั้นของซอฟต์แวร์ (ดีไวซ์ไดรเวอร์) และฮาร์ดแวร์ซึ่งทำงานด้านการเชื่อมโยงเข้ากับสายสื่อสาร ตัวอย่างมาตรฐานในระดับชั้นนี้ได้แก่ อีเทอร์เน็ตและโทเค็นริง เป็นต้น

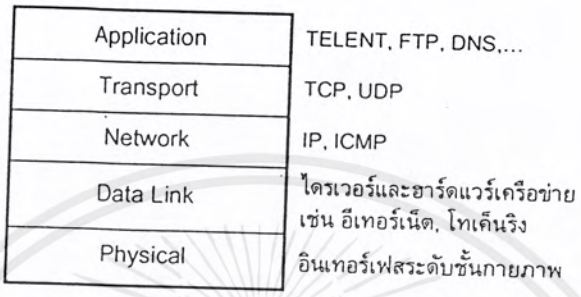
**เน็ตเวอร์ค** ชั้นที่ทำหน้าที่เลือกเส้นทางเพื่อส่งข้อมูลระหว่างระหว่างสถานีต้นทางและสถานีปลายทาง ตัวอย่างโปรโตคอลในระดับชั้นนี้ได้แก่ ไอพี

**ทรานสปอร์ต** ชั้นที่ทำหน้าที่จัดเตรียมการส่งข้อมูลระหว่างสถานีต้นทางและปลายทางโดยสถาปนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อและรักษาสภาพการเชื่อมต่อ ตลอดจนยกเลิกการเชื่อมต่อเมื่อสิ้นสุดกระบวนการ และอาจมีหน้าที่เพิ่มเติมในการรับประกันความถูกต้องของข้อมูลที่จัดส่ง ทีซีพี/ไอพีมีโปรโตคอลประจำชั้นนี้จำนวนสองโปรโตคอลคือ ทีซีพีและยูดีพี

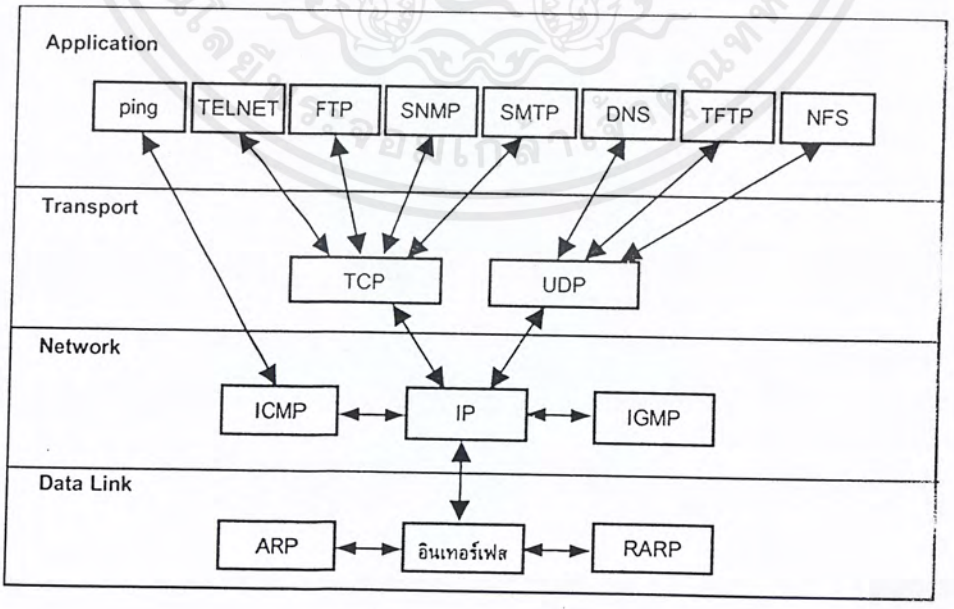
**แอพพลิเคชัน** ระดับชั้นนี้กำหนดการทำงานของโปรโตคอลประยุกต์ ตัวอย่างโปรโตคอลในระดับชั้นนี้ได้แก่ เอฟทีพี (FTP) เอสเอ็มทีพี (STMP) หรือเทลเน็ต (TELNET) เป็นต้น




รูปที่ 2.10 แบบอ้างอิงทีซีพี/ไอพี

### 2.8.2 โปรโตคอลสแตค

การทำงานตามโปรแกรมประยุกต์หนึ่งๆ ไม่ได้ใช้โปรโตคอลพร้อมกันทั้งชุด หากแต่ใช้เพียงโปรโตคอลที่สัมพันธ์กันไปในแต่ละชั้นของแบบอ้างอิง ตัวอย่างเช่น เทลเน็ตจะอาศัยทีซีพีและไอพีตามลำดับ การซ้อนทับของโปรโตคอลจากระดับชั้นบนไปชั้นล่างเรียกว่า โปรโตคอลสแตค (protocol stack)




รูปที่ 2.11 โปรโตคอลสแตคของทีซีพี/ไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวอร์คตามรูปที่ 2.11 เป็นแกนสำคัญของโปรโตคอลสแตค เนื่องจากทั้งทีซีพีและยูดีพีต้องใช้ไอพีเพื่อเลือกเส้นทางส่งแพ็กเก็ต ในระดับชั้นเน็ตเวอร์คยังมีไอซีเอ็มพีสนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดปกติที่เกิดขึ้นจากการส่งแพ็กเก็ตและมีไอจีเอ็มพีดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัลติคาสต์

ระดับชั้นทรานสปอร์ตมีสองโปรโตคอลที่สำคัญคือทีซีพีและยูดีพี แอพลิเคชั่นจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงาน

โปรโตคอลระดับล่างถัดจากไอพีได้แก่ โปรโตคอลระดับเดทาลิงค์ซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งานเช่นโปรโตคอล ซีเอสเอ็มเอ/ซีดี (CSMA/CD) ตามมาตรฐานอีเทอร์เน็ต ในระดับชั้นนี้มีโปรโตคอลในชุดของทีซีพี/ไอพีทำหน้าที่สนับสนุนการทำงานอยู่สองโปรโตคอลคือเออาร์พีและอาร์เออาร์พี ทั้งสองโปรโตคอลทำหน้าที่แปลงค่าระหว่างไอพีแอดเดรสกับฮาร์ดแวร์แอดเดรส

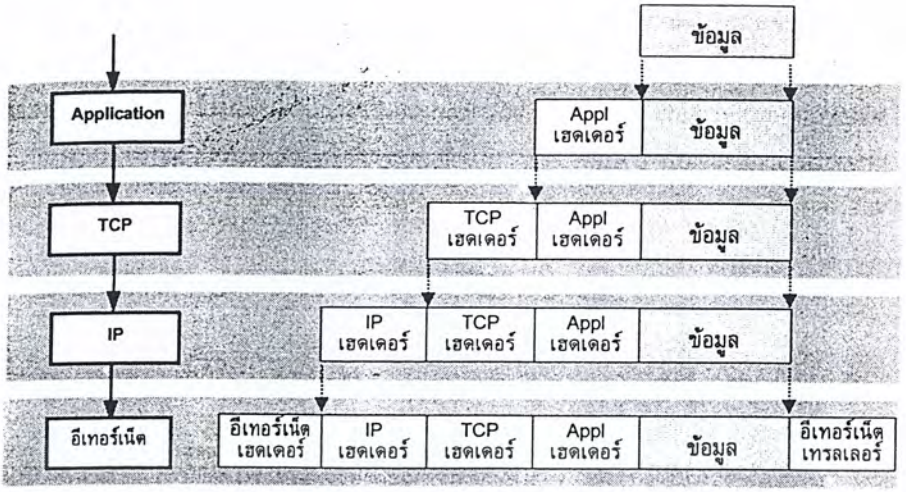
### 2.8.3 การส่งถ่ายข้อมูลระหว่างชั้น

โปรโตคอลในแต่ละชั้นล้วนมีหน้าที่เกี่ยวข้องในการส่งผ่านข้อมูลจากสถานีต้นทางไปยังสถานีปลายทาง ข้อมูลจะถูกส่งผ่านจากโปรโตคอลระดับบนสุดจากสถานีต้นทางไปยังระดับล่างจนกระทั่งข้อมูลถูกแปลงให้อยู่ในรูปของสัญญาณไฟฟ้าแล้วเดินทางผ่านสถานีเครือข่ายไปยังสถานีปลายทาง โปรโตคอลระดับล่างสุดที่สถานีปลายทางจะรับสัญญาณและส่งผ่านขึ้นไปยังโปรโตคอลระดับบนต่อไป

เมื่อข้อมูลผ่านแต่ละระดับชั้น โปรโตคอลในชั้นนั้นจะผนวกข่าวสารกำกับการทำงานประจำโปรโตคอลซึ่งเรียกว่า โปรโตคอลเฮดเดอร์ (protocol header) เข้ากับข้อมูลเฮดเดอร์และตัวข้อมูลจากระดับบนจะถูกส่งผ่านไปยังระดับล่าง โปรโตคอลระดับล่างจะมองเฮดเอดร์และตัวข้อมูลรวมเป็นเสมือนข้อมูลและเพิ่มเฮดเดอร์ประจำชั้นเข้าไปข้อมูลเดิมจึงมีแค่เฮดเดอร์หุ้มเป็นชั้นๆ กระบวนการนี้เรียกว่า การเอ็นแคปซูเลต (encapsulation) ตัวอย่างในรูปที่ 2.12 แสดงการเอ็นแคปซูเลตแพ็กเก็ตทีซีพี/ไอพีในอีเทอร์เน็ต

เมื่อสถานีปลายทางได้รับแพ็กเก็ตก็จะดำเนินการส่งไปตามลำดับชั้น โปรโตคอลประจำชั้นจะทำการถอดเฮดเดอร์ออกและส่งส่วนที่เหลือไปยังชั้นถัดไป เฮดเดอร์จถูกถอดออกเหลือเฉพาะข้อมูลเมื่อถึงชั้นบนสุด กระบวนการนี้เรียกว่า การดีแคปซูเลต (decapsulation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.12 การห่อหุ้มข้อมูลตามลำดับโปรโตคอลสแตค

### 2.8.4 ไอพีแอดเดรส

อินเตอร์เน็ตแยกแยะเครื่องโดยใช้ ไอพีแอดเดรส ประจำฮาร์ดแวร์อินเทอร์เฟสที่เชื่อมเข้าเครือข่าย ตัวอย่างของฮาร์ดแวร์อินเทอร์เฟสได้แก่การ์ดเครือข่าย คอมพิวเตอร์ที่มีมากกว่าหนึ่งอินเทอร์เฟสสามารถมีไอพีแอดเดรสได้ตามจำนวนอินเทอร์เฟส แต่คอมพิวเตอร์โดยทั่วไปมักมีเพียงอินเทอร์เฟสเดียว จึงมักเรียกว่าไอพีแอดเดรสเป็นแอดเดรสประจำเครื่อง ส่วนอุปกรณ์ฮาร์ดแวร์เช่น เราเตอร์ มักมีอินเทอร์เฟสจำนวนมากเพื่อใช้โยงเครือข่าย เราเตอร์จึงมีไอพีแอดเดรสหลายค่าตามจำนวนอินเทอร์เฟส

อุปกรณ์ที่มีตั้งแต่สองอินเทอร์เฟสขึ้นไปเรียกว่าอุปกรณ์ มัลติโฮม (multihomed) เราเตอร์จึงเป็นอุปกรณ์มัลติโฮม เราเตอร์อาจสร้างมาจากฮาร์ดแวร์เฉพาะ หรือนำโฮสต์ที่มีหลายอินเทอร์เฟสมาทำหน้าที่เป็นเราเตอร์ แต่โฮสต์ที่มีหลายอินเทอร์เฟสไม่จำเป็นต้องเป็นเราเตอร์หากไม่ได้ติดตั้งและกำหนดหน้าที่ให้ส่งต่อแพ็กเก็ตตามหน้าที่ของเราเตอร์ ในกรณีนี้โฮสต์จะเป็นเพียงมัลติโฮมโฮสต์เท่านั้น

ไอพีแอดเดรสมีขนาด 32 บิต เขียนแยกออกเป็นเลขฐานสิบ 4 ส่วนๆละ 8 บิตและ เขียนเรียงต่อกันไปโดยใช้จุดคั่นตัวเลข เช่น 158.108.2.71 หน่วยงานที่ต้องการเชื่อมต่อเข้าอินเทอร์เน็ตจะต้องขอหมายเลขเครือข่ายอย่างเป็นทางการจากศูนย์บริการ เช่น อินเทอร์นิกหรือจากผู้ให้บริการอินเทอร์เน็ตที่เครือข่ายนั้นขอเชื่อมไปสู่อินเทอร์เน็ต

### 2.8.5 ระบบชื่อโดเมน

การใช้ไอพีแอดเดรสไม่สะดวกต่อการจดจำ อินเทอร์เน็ตจึงใช้การตั้งชื่อเครื่องและมีระบบ ดีเอ็นเอส (DNS : Domain Name System) ให้บริการแปลงชื่อไปสู่ไอพีแอดเดรส เช่น nontri.ku.ac.th เป็นชื่อแทนเครื่องที่มีไอพีแอดเดรส 158.108.2.71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโดเมนหรือเรียกว่า FQDN (Fully-qualified Domain Name) มีวิธีเขียนเป็นมาตรฐานและสื่อความหมายได้เช่น nuclues.nectec.or.th เป็นชื่อเครื่อง nuclues ที่เนคเทค หรือ nontri.ku.ac.th เป็นชื่อเครื่อง nontri ที่มหาวิทยาลัยเกษตรศาสตร์ ชื่อจะคั่นด้วยจุดเพื่อบอกถึงองค์กรที่สังกัดเป็นลำดับ อักษรตัวเล็กหรือตัวใหญ่ในชื่อไม่แตกต่างกัน

#### 2.8.5.1 โดเมน

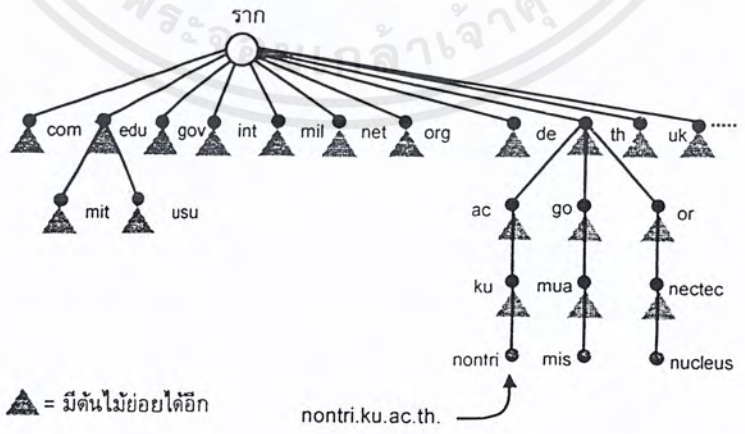
การตั้งชื่อในอินเทอร์เน็ตมีหลักเกณฑ์สากลเรียกว่า ระบบชื่อโดเมน ซึ่งเป็นระบบจัดแบ่งเครือข่ายเป็นลำดับชั้น เช่นภายในบริษัทอาจแบ่งออกเป็นแผนกย่อย แต่ละแผนกอาจแบ่งย่อยอีกเป็นฝ่าย แต่ละส่วนย่อยนี้เรียกว่า โดเมนย่อย (Sub-Domain)

ตัวอย่างเช่น nontri.ku.ac.th ประกอบด้วยโดเมนย่อย 4 ระดับ ชื่อโดเมนระดับบนสุดคือ th หมายถึงประเทศไทย โดเมนย่อยระดับถัดมาคือ ac (academic) หมายถึงโดเมนในเครือข่ายสถาบันการศึกษา โดเมนถัดมาคือ ku หมายถึงมหาวิทยาลัยเกษตรศาสตร์ และ nontri เป็นชื่อเครื่องคอมพิวเตอร์

#### 2.8.5.2 โดเมนระดับบนสุด

อินเทอร์เน็ตมีกำเนิดมาจากเครือข่ายในประเทศสหรัฐอเมริกา ชื่อโดเมนระดับบนสุดในยุคแรกเริ่มมีอยู่ 6 กลุ่มคือ com, edu, gov, mil, net และ org เมื่ออินเทอร์เน็ตขยายตัวออกไปทั่วโลกจึงมีโดเมนประจำประเทศเพิ่มขึ้น

การแยกแยะเครื่องด้วยระบบโดเมนจึงมีโครงสร้างตามลำดับชั้น โดยเริ่มที่จุดยอดซึ่งเรียกว่า ราก (root) และแตกออกตามโดเมนระดับบนสุด จากนั้นจะแตกย่อยเป็นลำดับจนถึงชื่อเครื่อง โครงสร้างระบบชื่อโดเมนจึงเหมือนต้นไม้หัวกลับ เช่นเดียวกับโครงสร้างไดเรกทอรีของแฟ้มข้อมูลดังรูปที่ 2.13




รูปที่ 2.13 โครงสร้างลำดับชั้นของระบบโดเมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.8.5.3 ตรวจสอบชื่อด้วย nslookup

ระบบปฏิบัติการทั่วไปมีโปรแกรมให้บริการแปลงระหว่างชื่อเครื่องและไอพีแอดเดรส ตัวอย่างเช่นในยูนิกซ์มีโปรแกรม nslookup สำหรับสืบค้นฐานข้อมูลดีเอ็นเอสและใช้ตรวจหาชื่อเครื่องและไอพีแอดเดรส ตัวอย่างต่อไปนี้แสดงการใช้ nslookup สอบถามแอดเดรส

```
% nslookup                              เรียกโปรแกรม
Default Server:  ns.ku.ac.th            แสดงเซอร์ฟเวอร์ที่ให้บริการ
Address:  158.108.2.67

> www                                   ถามไอพีแอดเดรสของ www
Server:  ns.ku.ac.th
Address:  158.108.2.67
Name:    www.ku.ac.th
Address:  158.108.2.72                  ได้แอดเดรสของ www

> 158.108.2.71                          ถามชื่อของแอดเดรส 158.108.2.71
Server:  ns.ku.ac.th
Address:  158.108.2.67

Name:    nontri.ku.ac.th                ได้ชื่อเครื่องของ 158.108.2.71
Address:  158.108.2.71

> www.nectec.or.th.                     ถามแอดเดรสของเว็บเซอร์ฟเวอร์ที่เนคเทค
Server:  ns.ku.ac.th
Address:  158.108.2.67

Non-authoritative answer:
Name:    www.nectec.or.th
Address:  202.44.204.33
>quit                                   ออกจาก nslookup
```

### 2.8.6 โปรโตคอลในทีซีพี/ไอพี

ตามที่กล่าวไว้แล้วว่าทีซีพี/ไอพีประกอบด้วยโปรโตคอลย่อยจำนวนมาก ตารางที่ 2.5 แสดงส่วนหนึ่งของโปรโตคอลมาตรฐานในอินเทอร์เน็ต ในที่นี้จะแนะนำเฉพาะโปรโตคอลที่สำคัญโดยสรุปดังนี้

**ไอพี (IP : Internet Protocal)**

ไอพีเป็นโปรโตคอลแกนของทีซีพี/ไอพี ไอพีทำหน้าที่กำหนดรูปแบบของแอดเดรสประจำเครื่องเพื่อใช้ในการลำเลียงข้อมูลจากเครื่องต้นทางไปยังเครื่องปลายทาง นอกจากนี้ยังทำหน้าที่เลือกเส้นทางส่งข้อมูล ตลอดจนแบ่งขนาดข้อมูลให้เหมาะกับฮาร์ดแวร์ระดับล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ไอซีเอ็มพี (ICMP: Internet Control Message Protocal)**

ไอซีเอ็มพีเป็นโปรโตคอลซึ่งใช้รายงานสถานะความผิดพลาดที่เกิดขึ้น ตัวอย่างเช่นในกรณีที่เราเตอร์ไม่สามารถนำข้อมูลส่งไปถึงปลายทางได้ เราเตอร์จะใช้ไอซีเอ็มพีแจ้งสาเหตุของปัญหากลับไปยังฝ่ายส่ง

**ทีซีพี (TCP : Transmission Control Protocal)**

ทีซีพีทำหน้าที่นำส่งข้อมูลโดยรับประกันความเชื่อถือ ทีซีพีด้านส่งต้องส่งแพ็กเก็ตซ้ำใหม่หากแพ็กเก็ตสูญหาย ทีซีพีด้านรับมีหน้าที่จัดแพ็กเก็ตให้ถูกต้องตามลำดับและกำจัดแพ็กเก็ตซ้ำซ้อน ทีซีพีเป็นโปรโตคอลแบบ "connection oriented" คือต้องสถาปนาการเชื่อมต่อระหว่างสถานีต้นทางและปลายทางก่อนการส่งข้อมูล

ทีซีพีต้นทางจัดแบบข้อมูลเพื่อส่งให้ไอพีดำเนินการ ทีซีพีปลายทางเมื่อรับแพ็กเก็ตจากไอพีก็จะส่งต่อให้โปรโตคอลประยุกต์ โปรโตคอลประยุกต์ที่ใช้บริการผ่านทีซีพีได้แก่ เทลเน็ต เอสเอ็มทีพี หรือเอฟทีพี เป็นต้น

ตารางที่ 2.5 โปรโตคอลมาตรฐานบางส่วนในอินเทอร์เน็ต

| โปรโตคอล | ชื่อ | RFC | STD |
|---|---|---|---|
| IP | Internet Protocol | 791 | 5 |
| ICMP | Internet Control Message Protocol | 792 | 5 |
| IGMP | Internet Group Multicast Protocol | 1112 | 5 |
| UDP | User Datagram Protocol | 768 | 6 |
| TCP | Transmission Control Protocol | 793 | 7 |
| TELNET | Telnet Protocol | 854,855 | 8 |
| FTP | File Transfer Protocol | 959 | 9 |
| SMTP | Simple Mail Transfer Protocol | 821 | 10 |
| MAIL | Format of Electronic Mail Messages | 822 | 11 |
| NTPV2 | Network Time Protocol (Version 2) | 1119 | 12 |
| DOMAIN | Domain Name System | 1034,1035 | 13 |
| DNS-MX | Mail Routing and the Domain System | 974 | 14 |
| SNMP | Simple Network Management Protocol | 1157 | 15 |
| SMI | Structure of Management Information | 1155 | 16 |
| MIB-II | Management Information Base-II | 1213 | 17 |
| NETBIOS | NetBIOS Service Protocols | 1001,1002 | 19 |
| ECHO | Echo Protocol | 862 | 20 |
| DISCARD | Discard Protocol | 863 | 21 |
| CHARGEN | Character Generator Protocol | 864 | 22 |
| QUOTE | Quote of the Day Protocol | 865 | 23 |
| USERS | Active Users Protocol | 866 | 24 |
| DAYTIME | Daytime Protocol | 867 | 25 |
| TIME | Time Server Protocol | 868 | 26 |
| TFTP | Trivial File Transfer Protocol | 1350 | 33 |
| TP-TCP | ISO Transport Service on top of the TC | 1006 | 35 |
| PPP | Point-to-Point Protocol (PPP) | 1661 | 51 |
| PPP-HDLC | PPP in HDLC Framing | 1662 | 51 |
| IP-SMDS | IP Datagrams over the SMDS Service | 1209 | 52 |
| POP3 | Post Office Protocol, Version 3 | 1939 | 53 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ยูดีพี (UDP : User Datagram Protocal)**

ยูดีพีเป็นโปรโตคอลแบบ "connectionless" คือไม่ต้องสถาปนาการเชื่อมต่อระหว่างสถานีรับและส่ง ยูพีดีเป็นโปรโตคอลระดับชั้นเดียวกับทีซีพีแต่ว่าไม่มีกลไกรับประกันความเชื่อถือในการขนถ่ายข้อมูล หากข้อมูลสูญหาย ซ้ำซ้อน หรือลำดับไม่ถูกต้อง ยูดีพีจะปล่อยให้โปรโตคอลที่เรียกใช้งานดำเนินการกับปัญหาเหล่านี้เอง

ลักษณะเด่นของยูดีพีคือใช้การประมวลผลต่ำกว่าทีซีพี เนื่องจากเฮดเดอร์มีขนาดเล็กและไม่ต้องสถาปนาการเชื่อมต่อ แอพลิเคชันที่ทำงานโดยรับส่งคำถามและคำตอบเป็นรายการ (transaction) เช่น ดีเอ็นเอส หรือทีเอฟทีพีจะเหมาะกับการใช้บริการผ่านยูดีพี

**เอสเอ็มทีพี (STMP : Simple Mail Transfer Protocal)**

บริการพื้นฐานที่มีในทุกเครือข่ายได้แก่ บริการไปรษณีย์อิเล็กทรอนิกส์ เอสเอ็มทีพีเป็นโปรโตคอลทำหน้าที่รับส่งจดหมายอิเล็กทรอนิกส์ระหว่างโฮสต์

**เอฟทีพี (FTP : File Transfer Protocal)**

เอฟทีพีให้บริการถ่ายโอนแฟ้มข้อมูลระหว่างเครื่อง เอฟทีพีช่วยให้ผู้ใช้เข้าถึงโฮสต์และจำกัดขอบเขตการทำงานเฉพาะที่เกี่ยวข้องกับแฟ้มข้อมูลเช่น สำเนาแฟ้ม ลบแฟ้ม หรือสร้างไดเรกทอรี เป็นต้น

**ทีเอฟทีพี (TFTP : Trivial Transfer Protocal)**

ทีเอฟทีพีทำหน้าที่เป็นโปรโตคอลถ่ายโอนแฟ้มเช่นเดียวกับเอฟทีพี แต่ทีเอฟทีพีใช้บริการผ่านยูพีดี ประโยชน์หนึ่งของการใช้ทีเอฟทีพีได้แก่การใช้ไน สถานีไร้ดิสก์ (diskless workstation) ซึ่งไม่มีดิสก์ประจำตัว สถานีไร้ดิสก์จะบูตระบบด้วยโปรโตคอลบูตพีและใช้ทีเอฟทีพีเพื่อถ่ายโอนระบบปฏิบัติการจากทีเอฟทีพีเซอร์ฟเวอร์

**เทลเน็ต (TELNET : Telecommunication Network)**

เทลเน็ตเป็นโปรโตคอลสำหรับขอเข้าใช้โฮสต์ระยะไกล หรือเรียกว่า รีโมตล็อกอิน (remote login) เทลเน็ตให้บริการเข้าใช้คอมพิวเตอร์ในเครือข่ายโดยเสมือนกับว่ากำลังทำงานอยู่ที่เทอร์มินัลของคอมพิวเตอร์เครื่องนั้น เทงเน็ตเซอร์ฟเวอร์ที่คอมพิวเตอร์ปลายทางจะรอรับการขอบริการจากเทลเน็ตไคลเอ็นต์ ผูขอใช้เทลเน็ตจะต้องมีบัญชีประจำเครื่องที่ให้บริการเทลเน็ต

**ดีเอ็นเอส (DNS : Domain Name System)**

ดีเอ็นเอสเป็นโปรโตคอลที่ให้บริการสอบถามไอพีแอดเดรสหรือชื่อโดเมน ก่อนการติดต่อไปยังโฮสต์ใดๆ ชื่อโฮสต์จะถูกส่งไปสอบถามผ่านเซอร์ฟเวอร์ที่ให้บริการดีเอ็นเอสเพื่อขอไอพีแอดเดรสกลับมา นอกจากนี้ยังให้บริการเกี่ยวข้องกับฐานข้อมูลประจำเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เอสเอ็นเอ็มพี (SNMP : Simple Network Management Protocal)**

เอสเอ็นเอ็มพีทำหน้าที่เป็นโปรโตคอลช่วยบริหารระบบ เช่น เก็บรวบรวมข้อมูลการทำงานของอุปกรณ์และคอมพิวเตอร์ภายในเครือข่าย ตรวจสอบปริมาณข้อมูลที่ไหลเวียนหรือช่วยวิเคราะห์หาข้อผิดปกติในระบบ เป็นต้น

**เอ็นเอฟเอส (NFS : Network File System)**

เอ็นเอฟเอสเป็นโปรโตคอลที่พัฒนาขึ้นโดยบริษัทซันไมโครซิสเต็ม เอ็นเอสเอฟใช้งานในระบบรีโมตดิสค์เพื่อให้คอมพิวเตอร์เครื่องหนึ่งสามารถใช้แฟ้มข้อมูลที่เก็บอยู่ในดิสค์ของอีกเครื่องหนึ่งเสมือนว่าเป็นแฟ้มที่อยู่ในดิสค์ของเครื่องนั้น

**เอ็นเอ็นทีพี (NNTP : Network News Tranfer Protocal)**

เอ็นเอ็นทีพีเป็นโปรโตคอลสำหรับใช้แลกเปลี่ยนข่าวตามบริการยูสเน็ต คอมพิวเตอร์แม่ข่ายที่ให้บริการข่าวจะใช้เอ็นเอ็นทีพีแลกเปลี่ยนข่าวระหว่างกัน ส่วนคอมพิวเตอร์ลูกข่ายจะใช้เอ็นเอ็นทีพีสำหรับอ่านและส่งข่าวกับเครื่องแม่ข่าย

**เอชทีทีพี (HTTP : Hypertext Transfer Protocal)**

เอชทีทีพีเป็นโปรโตคอลกำหนดการส่งข้อมูลระหว่างบราวเซอร์และเว็บเซอร์ฟเวอร์ในระบบเว็บ และจัดเป็นโปรโตคอลหนึ่งที่ใช้แพร่หลายในอินเทอร์เน็ต

**บูตพี (BOOTP : Bootstrap Protocal)**

ให้บริการบูตระบบสำหรับสถานีที่ไม่มีดิสค์ สถานีที่เป็นบูตพีไคลเอ็นต์จะติดต่อกับบูตพีเซอร์ฟเวอร์เพื่อขอถ่ายโอนระบบปฏิบัติการ

**ดีเอชซีพี (DHCP : Dynamic Host Configuration Protocal)**

ดีเอชซีพีบริการติดตั้งค่าแบบไดนามิกให้โฮสต์ในเครือข่าย การทำงานของดีเอชซีพีเป็นแบบไคลเอ็นต์-เซอร์ฟเวอร์ ดีเอชซีพีเซอร์ฟเวอร์จะให้ค่าแบบไม่ตายตัวกับไคลเอ็นต์ที่ขอบริการเช่น การให้ไอพีแอดเดรส หรือค่าอื่นประจำไคลเอ็นต์

### 2.8.7 ทีซีพี/ไอพี ไคลเอ็นต์-เซอร์ฟเวอร์

โปรโตคอลในทีซีพี/ไอพีอาศัยหลักการทำงานตามแบบไคลเอ็นต์-เซอร์ฟเวอร์ ซึ่งมีองค์ประกอบ 3 ส่วนคือ

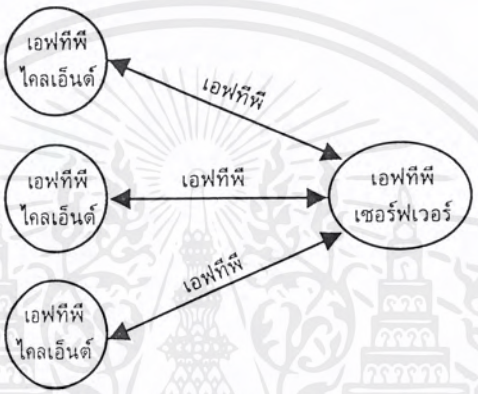
1. โปรเซสไคลเอ็นต์ซึ่งทำหน้าที่ขอบริการ
2. โปรเซสเซอร์ฟเวอร์ซึ่งทำหน้าที่ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรโตคอลที่ใช้สื่อสารระหว่างไคลเอ็นต์และเซอร์ฟเวอร์

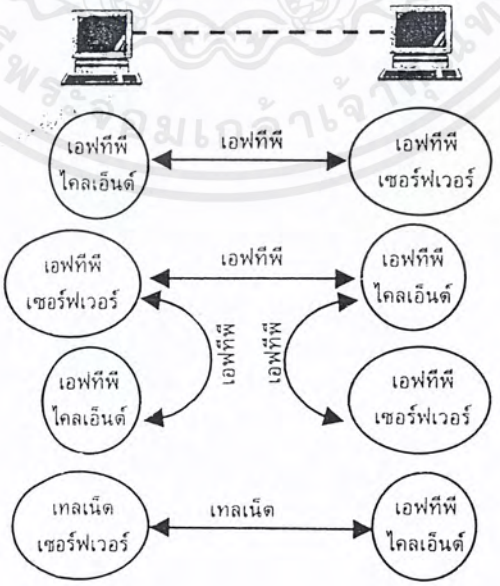
รูปที่ 2.14 แสดงการเชื่อมโยงระหว่างไคลเอ็นต์ 3 โปรเซส ซึ่งขอบริการถ่ายโอนแฟ้มจากเซอร์ฟเวอร์ผ่านเอฟทีพี โปรเซสไคลเอ็นต์และเซอร์ฟเวอร์ไม่จำเป็นต้องอยู่ต่างเครื่องกัน เนื่องจากระบบปฏิบัติการส่วนใหญ่ที่ใช้ทีซีพี/ไอพีมักทำงานแบบผู้ใช้หลายคนและพร้อมกันหลายงาน แต่ละไคลเอ็นต์จึงอาจเป็นโปรเซสของผู้ใช้ต่างบุคคลที่ทำงานภายในเครื่องเดียวกันและขอบริการไปยังเซอร์ฟเวอร์ต่างเครื่องหรือเครื่องเดียวกับไคลเอ็นต์ก็ได้

แบบจำลองไคลเอ็นต์-เซอร์ฟเวอร์ในทีซีพี/ไอพีนี้แตกต่างไปจากหลักการไคลเอ็นต์-เซอร์ฟเวอร์ในบางระบบปฏิบัติการเครือข่ายซึ่งกำหนดหน้าที่การทำงานของคอมพิวเตอร์อย่างเจาะจงว่าเครื่องใดเป็นเซอร์ฟเวอร์และเครื่องใดเป็นไคลเอ็นต์โดยสลับหน้าที่กันไม่ได้




รูปที่ 2.14 เอฟทีพีไคลเอ็นต์-เซอร์ฟเวอร์

ในขณะที่คอมพิวเตอร์แต่ละเครื่องตามแบบของทีซีพี/ไอพี สามารถทำหน้าที่เป็นไคลเอ็นต์เพื่อขอบริการจากเครื่องอื่น หรือทำหน้าที่เป็นเซอร์ฟเวอร์เพื่อให้บริการเครื่องอื่นได้พร้อมกันดังรูปที่ 2.15




รูปที่ 2.15 ตัวอย่างแบบจำลองไคลเอ็นต์-เซอร์ฟเวอร์ในทีซีพี/ไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.8 ไอพียุคใหม่

เนื่องจากอินเทอร์เน็ตเติบโตและขยายตัวอย่างรวดเร็วจนมีแนวโน้มว่าอาจจะมีปัญหาขาดแคลนไอพีแอดเดรส และตารางการเลือกเส้นทางประจำเราเตอร์หลักของอินเทอร์เน็ตก็มีขนาดใหญ่ขึ้นจนเราเตอร์อาจรองรับไม่ได้ในอนาคต ไออีทีเอฟเห็นปัญหาและวางแนวทางแก้ไขมาตั้งแต่พ.ศ. 2524 และได้กำหนดให้ ไอพีรุ่นหก (IP Version 6 Ipv6) เป็นไอพีที่นำมาใช้แทนไอพีรุ่นที่สี่

ไอพีรุ่นหกไม่ได้มีเพียงแต่แอดเดรสที่รองรับจำนวนโฮสต์ได้มากขึ้นเท่านั้น หากแต่ปรับปรุงจุดด้อยที่มีอยู่เดิมในรุ่นสี่ และตัดทอนส่วนที่ไม่ได้ใช้งาน รวมทั้งเพิ่มการทำงานแบบใหม่ที่สอดคล้องกับความต้องการทั้งปัจจุบันและอนาคต ไอพีรุ่นหกขยายแอดเดรสขึ้นเป็น 128 บิต สนับสนุนการทำงานแบบเวลาจริง และมีระบบติดตั้งแอดเดรสแบบอัตโนมัติ มีระบบรักษาความปลอดภัยที่กำหนดเป็นมาตรฐานประจำภายในโปรโตคอล รวมทั้งโครงสร้างการจัดสรรแอดเดรสที่ช่วยให้ตารางเลือกเส้นทางไม่ขยายขนาดอย่างรวดเร็วเกินไป

ไออีทีเอฟคาดการณ์ไว้ว่าไอพีรุ่นสี่จะประสบปัญหาขาดแคลนแอดเดรสอย่างมากระหว่างพ.ศ. 2548 ถึง 2553 ปัจจุบันชุดโปรโตคอลในไอพีรุ่นหกยังคงอยู่ระหว่างการจัดทำมาตรฐานและทดลองใช้งาน

## 2.9 ทีซีพี

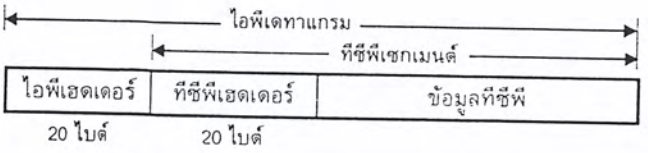
ทีซีพีเพิ่มความน่าเชื่อถือการส่งเดทาแกรมให้กับไอพี และให้บริการโปรโตคอลประยุกต์ระดับบนผ่านหมายเลขพอร์ตที่กำหนด ฟังก์ชันของไอพีมีความเรียบง่ายเพื่อการส่งเดทาแกรมด้วยสมรรถนะสูง ความเชื่อถือและรับประกันความถูกต้องเป็นงานของทีซีพีในโฮสต์ปลายทางแต่ละด้าน เนื้อหาในหัวข้อจะกล่าวถึงหน้าที่พื้นฐานของทีซีพีซึ่งประกอบด้วย

ทีซีพีเฮดเดอร์
แฟล็กและออพชันกำหนดการทำงาน
การส่งข้อมูลโดยใช้เลขลำดับ
การสถาปนาและการยกเลิกการเชื่อมต่อ
การควบคุมกระแสข้อมูล

### 2.9.1 โปรโตคอลทีซีพี

ทีซีพีเป็นโปรโตคอลที่ให้บริการชนิดที่ต้องมีการเชื่อมต่อ และรับประกันความเชื่อถือในการลำเลียงข้อมูล ทีซีพีรับประกันความเชื่อถือโดยทำหน้าที่ตรวจสอบเซกเมนต์ที่ผิดปกติและจัดส่งเซกเมนต์ซ้ำใหม่ รวมทั้งจัดลำดับให้ถูกต้องก่อนส่งไปยังโปรแกรมประยุกต์ระดับบนเฮดเดอร์และข้อมูลของทีซีพีเรียกว่า เซกเมนต์ การเอ็นแคปซูลเลตทีซีพีเซกเมนต์ในไอพีเดทาแกรมแสดงได้รูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

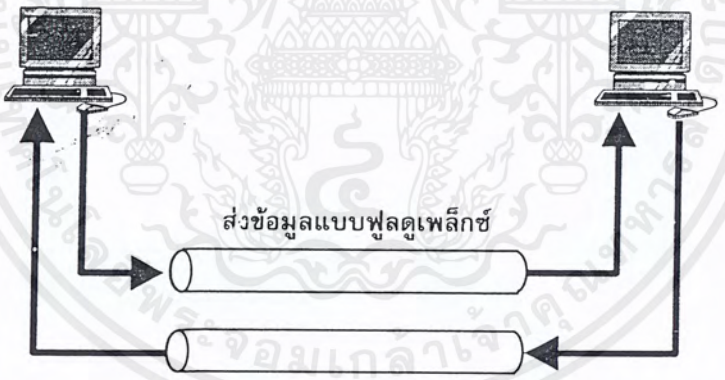



รูปที่ 2.16 การเอนแคปซูเลตทีซีพี

สถานีต้นทางจะต้องสถาปนาการเชื่อมโยงกับสถานีปลายทางก่อนส่งทีซีพีเซกเมนต์ การสถาปนาการเชื่อมโยงใช้ประโยชน์เพื่อให้มั่นใจว่าปลายทางพร้อมจะสื่อสารด้วย และเมื่อเสร็จสิ้นการส่งผ่านข้อมูลแล้วก็จะปิดการเชื่อมโยง

การสถาปนาการเชื่อมโยงของทีซีพีอาจเปรียบเทียบได้กับการติดต่อทางโทรศัพท์กล่าวคือเมื่อหมุนหมายเลขปลายทางแล้ว ผู้เรียกต้องรอให้ปลายทางรับสาย เมื่อทักทายและแจ้งให้ทราบว่าใครเป็นผู้เรียกสายแล้วจึงเริ่มการสนทนา กระบวนการสถาปนาของทีซีพีมีขั้นตอนเฉพาะ

ทีซีพีทำงานตามแบบไคลเอ็นต์-เซอร์เวอร์ ไคลเอ็นต์จะเป็นผู้ร้องขอบริการและขอสถาปนาการเชื่อมโยง ส่วนเซอร์ฟเวอร์รับการร้องขอและให้บริการต่อไคลเอ็นต์ การขนถ่ายข้อมูลระหว่างไคลเอ็นต์และเซอร์ฟเวอร์เป็นแบบฟูลดูลเพล็กซ์ เสมือนมีท่อลำเลียงสองท่อต่อเชื่อมระหว่างไคลเอ็นต์เซอร์ฟเวอร์ ดังรูปที่ 2.17




รูปที่ 2.17 แบบจำลองการลำเลียงข้อมูลในทีซีพี

ทีซีพีลำเลียงข้อมูลในรูปของไบต์ขนาด 8 บิต หรือมักเรียกว่าออกเต็ต (octet) และเรียกบริการลำเลียงแบบนี้ว่า บริการสายข้อมูลแบบไบต์ (byte stream service) โปรโตคอลประยุกต์ปลายทางไม่จำเป็นต้องรับข้อมูลมาดำเนินการครั้งละไบต์เสมอไป หากแต่จะดำเนินการไปตามการทำงานของแต่ละโปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

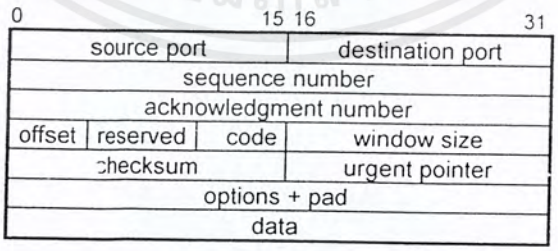
### 2.9.2 บริการรับประกันความเชื่อถือของทีซีพี

ทีซีพีให้บริการจัดการด้านความเชื่อถือการลำเลียงเซกเมนต์ที่สำคัญคือ การตรวจจับและแก้ไขข้อผิดพลาด, การควบคุมปริมาณการไหลข้อมูล, การจัดลำดับ, และการกำจัดเซกเมนต์ซ้ำ โดยมีรายละเอียดดังต่อไปนี้

- การตรวจจับและแก้ไขข้อผิดพลาดมุ่งเน้นแก้ปัญหาเซกเมนต์ที่ผิดปกติจากปัญหาของสายสื่อสารหรือโปรโตคอลระดับล่าง หรือเซกเมนต์เดินทางไปไม่ถึงปลายทาง ทีซีพีจัดการกับปัญหานี้โดยใช้ผลรวมตรวจสอบเช่นเดียวกับยูดีพี หากค่าผลรวมตรวจสอบไม่ถูกต้อง ทีซีพีจะส่งเซกเมนต์ซ้ำใหม่
- เมื่อทีซีพีได้รับเซกเมนต์จะตอบรับกลับไปต้นทาง ทีซีพีใช้การตอบรับเพื่อยืนยันว่าได้รับข้อมูลอย่างถูกต้อง
- ทีซีพีจะตั้งเวลาเมื่อส่งเซกเมนต์เพื่อรอการตอบรับจากปลายทาง หากไม่มีการตอบรับภายในเวลาที่กำหนดทีซีพีจะส่งเซกเมนต์ซ้ำ วิธีนี้เรียกว่า การตอบรับแบบบวกพร้อมกับการส่งซ้ำ (positive acknowledge with retransmission)
- ทีซีพีฝ่ายรับสามารถกำหนดให้ฝ่ายส่งจัดส่งข้อมูลเป็นปริมาณเท่าที่จะรับได้จริงตามขนาดบัฟเฟอร์ที่มีอยู่ การควบคุมปริมาณการไหลข้อมูลเป็นการป้องกันไม่ให้ส่งเซกเมนต์ไปเกินความสามารถของฝ่ายรับ
- หากเซกเมนต์มาถึงไม่เป็นลำดับ ทีซีพีฝ่ายรับต้องจัดลำดับเซกเมนต์ให้ถูกต้อง
- ทีซีพีมีหน้าที่กำจัดเซกเมนต์ซ้ำซ้อน

### 2.9.3 ทีซีพีเฮดเดอร์

ทีซีพีเฮดเดอร์ประกอบด้วยฟิลด์จำนวนมากทำหน้าที่ให้บริการตามฟังก์ชันที่กล่าวข้างต้น รูปที่ 2.18 แสดงเฮดเดอร์ของทีซีพี แต่ละฟิลด์มีความหมายดังต่อไปนี้




รูปที่ 2.18 ทีซีพีเฮดเดอร์

source port ขนาด 16 บิต : หมายเลขพอร์ตของสถานีต้นทาง

destination port ขนาด 16 บิต : หมายเลขพอร์ตของสถานีปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

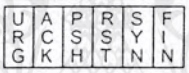
sequence number ขนาด 32 บิต : ทีซีพีใช้ เลขลำดับ เป็นตัวนับจำนวนไบต์ที่ส่งทุกครั้งที่สถาปนาการเชื่อมโยงทีซีพีจะเลือกเลขลำดับเริ่มต้นสำหรับชี้ตำแหน่งข้อมูลไบต์แรกที่จะจัดส่ง หมายเลขเริ่มต้นไม่จำเป็นต้องเริ่มด้วย 1 แต่อาจเริ่มด้วยค่าใดๆ ก็ได้ ข้อมูลในเซกเมนต์ถัดไปจะมีเลขลำดับที่สัมพัทธ์เลขลำดับในเซกเมนต์ก่อนหน้า

acknowledgement number ขนาด 32 บิต : ค่ากำหนด เลขตอบรับ ซึ่งใช้ตอบกลับไปว่าได้รับข้อมูลแล้ว เลขตอบรับจะมีค่าเท่ากับเลขลำดับประจำเซกเมนต์บวกด้วยจำนวนไบต์ข้อมูลและบวกด้วยหนึ่ง เช่นเซกเมนต์หนึ่งมีเลขลำดับเท่ากับ 21 และมีข้อมูล 20 ไบต์ เลขตอบรับที่ต้องส่งกลับไปจะเท่ากับ 21+20+1=42 ซึ่งแจ้งว่าได้รับข้อมูลตั้งแต่ต้นถึงไบต์ลำดับที่ 41 แล้ว และคาดว่าไบต์ถัดไปคือไบต์ที่ 42

offset (data offset) ขนาด 4 บิต : ขอกถึงตำแหน่งเริ่มต้นของไบต์ข้อมูลหรืออีกนัยหนึ่งใช้บอกขนาดเฮดเดอร์ ตัวเลขนี้มีหน่วยเป็นจำนวนเท่าของ 4 ไบต์เช่นเดียวกับที่ใช้ในไอพีเดทาแกรม เฮดเดอร์ของทีซีพีมีความยาวขึ้นกับฟิลด์ option ตัวเลขในฟิลด์ offset จะเท่ากับ 5 ซึ่งเท่ากับ 20 ไบต์ (5*4=20) หากไม่ใช้ออพชันใด

reserved (RSV) ขนาด 4 บิต : สำรองไว้ใช้ในอนาคต

code ประกอบด้วย 6 ฟิลด์ย่อย แต่ละฟิลด์ย่อยมีขนาด 1 บิต ทำหน้าที่เป็นแฟล็กเรียงลำดับจากซ้ายไปขวาต่อไปนี้




- URGent ถ้าบิตนี้เป็น “1” หมายความว่า Urgent pointer บรรจุตำแหน่งข้อมูลที่ต้องรีบดำเนินการเร่งด่วนก่อน
- ACKnowledgement ถ้าบิตนี้เป็น “1” หมายถึงเป็นเซกเมนต์ตอบรับ โดยตอบอ้างอิงเลขลำดับตามที่กำหนดในฟิลด์ acknowledgement number
- PuSH ถ้าบิตนี้เป็น “1” หมายความว่าทันทีที่สถานีปลายทางได้รับเซกเมนต์ต้องรีบส่งข้อมูลไปยังโปรโตคอลประยุกต์ทันทีโดยไม่ต้องรอให้บัฟเฟอร์เต็ม
- ReSeT ถ้าบิตนี้เป็น “1” หมายถึงให้ยกเลิกการเชื่อมต่อนี้เนื่องจากอาจมีความผิดปกติเกิดขึ้นระหว่างคู่สถานีที่ติดต่อกันอยู่ หากจำเป็นต้องส่งข้อมูลระหว่างกันอีกก็ต้องเริ่มต้นสถาปนาการเชื่อมต่อใหม่
- SYNchronize ถ้าบิตนี้เป็น “1” หมายถึงขอเริ่มต้นสถาปนาการเชื่อมต่อและเมื่อการสถาปนาเสร็จสิ้น บิตนี้จะถูกกำหนดให้เป็น “0” หลังจากนั้นจึงสามารถส่งผ่านข้อมูลระหว่างกันได้
- FINish ถ้าบิตนี้เป็น “1” หมายถึงขอจบการเชื่อมต่อ

window size ขนาด 16 บิต : สถานีปลายทางใช้ฟิลด์นี้แจ้งขนาดบัฟเฟอร์ที่มีอยู่ (หน่วยเป็นไบต์) สถานีที่ติดต่อด้วยต้องไม่ส่งข้อมูลเกินค่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

checksum ขนาด 16 บิต : ผลรวมตรวจสอบความถูกต้องของเซกเมนต์โดยคำนวณทั้งเฮดเดอร์และข้อมูล (ใช้เฮดเดอร์เทียมเช่นเดียวกับยูดีพี)

urgent pointer ขนาด 16 บิต : พอยเตอร์ชี้ตำแหน่งไบต์ข้อมูลที่ต้องดำเนินการเร่งด่วนที่ต้องการให้โปรแกรมประยุกต์ดำเนินการทันที ค่าที่บรรจุในฟิลด์นี้จะมีความหมายก็ต่อเมื่อแฟล็ก URG ถูกเซ็ตเป็น "1"

options ขนาดแปรเปลี่ยนได้ : ใช้กำหนดงานเพิ่มเติมให้กับทีซีพีซึ่งจะมีหรือไม่ก็ได้ หากฟิลด์ offset หากมีค่าเป็น 5 แสดงว่ามีเฮดเดอร์มีขนาด 20 ไบต์ซึ่งหมายถึงไม่ใช้ออพชัน ออพชันที่มีให้ใช้งานจะกล่าวในหัวข้อต่อไป

pad ขนาด 0 ถึง 24 บิต : ใช้เป็นส่วนที่ทำให้ขนาดของออพชันเป็นจำนวนเท่าของ 32 บิต (เพื่อให้เฮดเดอร์ลงตัวที่ค่าจำนวนเท่าของ 32)

### 2.9.4 หน้าที่ของฟิลด์ที่สำคัญในเฮดเดอร์

แฟล็กในทีซีพีมีความสำคัญต่อการทำงานของทีซีพีทั้งฝ่ายรับและฝ่ายส่ง ในส่วนนี้จะอธิบายถึงแฟล็กที่สำคัญคือ PSH, URG, SYN และ ACK รวมทั้งฟิลด์อื่นที่ใช้ประกอบกับแฟล็กดังกล่าว

#### 2.9.4.1 แฟล็ก PSH

ทีซีพีรับผิดชอบการจัดขนาดของเซกเมนต์ให้เหมาะสมเพื่อนำส่ง ทีซีพีรับข้อมูลจากโปรโตคอลประยุกต์มาเก็บไว้ในบัฟเฟอร์ แล้วจึงทยอยนำข้อมูลขนาดที่เหมาะสมมาสร้างเซกเมนต์ก่อนส่งต่อไป วิธีนี้แตกต่างจากยูดีพีซึ่งโปรโตคอลประยุกต์จะเป็นผู้กำหนดขนาดข้อมูลในยูพีดีเดทาแกรมเองโดยที่ยูพีดีไม่มีหน้าที่จัดการหรือเลือกขนาดได้

ความสามารถของทีซีพีส่วนนี้มีประโยชน์สำหรับงานบางประเภทที่ต้องนำส่งข้อมูลขนาดเล็กในทันทีโดยไม่ต้องรอให้บัฟเฟอร์เต็ม ตัวอย่างเช่นบริการเทลเน็ต เมื่อผู้ใช้พิมพ์คำสั่งตามด้วยการเคาะแป้น Enter ย่อมหมายถึงผู้ใช้ต้องการให้ทีซีพีนำคำสั่งนี้ส่งไปยังรีโมตเซอร์ฟเวอร์โดยทันที ทีซีพีมีฟังก์ชันการทำงานที่เรียกว่า "push" ผ่านแฟล็ก PSH เพื่อจัดให้ส่งข้อมูลทันที เมื่อฝ่ายรับได้รับเซกเมนต์จะต้องรีบส่งให้กับโปรโตคอลประยุกต์ทันทีเช่นกัน

#### 2.9.4.2 แฟล็ก URG

ฟังก์ชันอีกประการหนึ่งที่มีในทีซีพีคือ การกำหนดให้ข้อมูลในทีซีพีเซกเมนต์มีความสำคัญเร่งด่วนเป็นกรณีพิเศษที่ต้องรีบดำเนินการก่อน ฟังก์ชันนี้เรียกว่า "urgent" และทำงานผ่านแฟล็ก URG ตัวอย่างการใช้งานฟังก์ชันอย่างเช่นกรณีของของบริการรีโมตล็อกอิน หากผู้ใช้กดปุ่มขัดจังหวะการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วย Ctrl-C หรือ Ctrl-Break ขณะที่มีการส่งข้อมูลอยู่ ทีซีพีจะจัดส่งข้อมูลด่วนนี้ไปเพื่อให้ปลายทางรีบดำเนินการก่อนเพื่อหยุดการส่งข้อมูลในขณะนั้น

#### 2.9.4.3 แฟล็ก SYN และ ACK สำหรับการเชื่อมโยง

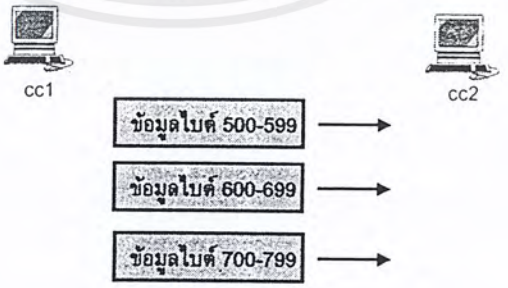
แฟล็กที่อยู่ในฟิลด์ code เป็นค่าสำคัญที่กำหนดสภาวะการทำงานต่างๆ ซึ่งได้แก่ การสถาปนาการเชื่อมโยง การส่งข้อมูล และการปิดการเชื่อมโยง

เซกเมนต์แรกที่ส่งจากไคลเอ็นต์เพื่อขอสถาปนาการเชื่อมโยงจะกำหนดให้แฟล็ก SYN เป็น 1 และ ACK เป็น 0 ฟิลด์ sequence number ในเซกเมนต์แรกนี้จะบรรจุ เลขลำดับเริ่มต้น (initial sequence number) และค่าที่จำเป็นอื่นๆ เช่น ขนาดบัฟเฟอร์ที่รับได้ในฟิลด์ window หรือกำหนดขนาดเซกเมนต์สูงสุดไว้ในฟิลด์ options

เซอร์ฟเวอร์ที่ได้รับเซกเมนต์ขอสถาปนา หากพร้อมก็จะตอบกลับโดยเซ็ตแฟล็ก SYN และ ACK ให้เป็น 1 ทั้งคู่ และกำหนดค่าในฟิลด์อื่นๆ ที่จำเป็นเช่นกัน

#### 2.9.4.4 การถ่ายโอนโดยใช้เลขลำดับ

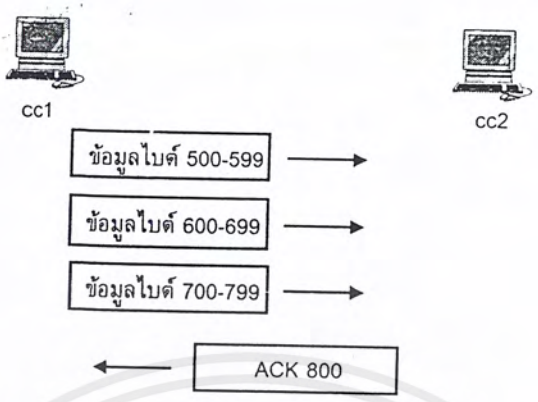
ทีซีพีอาศัยเลขลำดับและเลขตอบรับเพื่อยืนยันการถ่ายโอน ตามที่กล่าวมาแล้วว่าเลขลำดับที่ใช้ในเฮดเดอร์ไม่ใช่เลขนับลำดับเซกเมนต์ หากแต่เป็นเลขชี้ตำแหน่งไบต์ข้อมูลที่กำลังจัดส่ง ในขั้นตอนแรกของการสถาปนาการเชื่อมต่อแต่ละครั้ง ทีซีพีด้านไคลเอ็นต์จะเลือกเลขลำดับเริ่มต้นขนาด 32 บิต แล้วส่งเซกเมนต์ที่กำกับด้วยเลขลำดับเริ่มต้นเพื่อขอสถาปนาการเชื่อมต่อ เลขลำดับนี้จะเพิ่มค่าไปตามจำนวนไบต์ที่ส่งในแต่ละครั้งตามรูปที่ 2.13 ซึ่งแสดงลำดับเลขเริ่มต้นที่ 500 และเซกเมนต์ถัดไปกำกับด้วยเลขลำดับ 600 และ 700 (ทีซีพีไม่ได้ส่งตัวเลขกำกับขนาดไปในเซกเมนต์โดยตรงเหมือนในรูปนี้ ตัวเลข 599, 699 และ 799 เป็นเพียงตัวเลขที่แสดงให้ผู้อ่านทราบจำนวนไบต์ข้อมูลที่ส่งในแต่ละเซกเมนต์)




รูปที่ 2.19 การส่งโดยใช้เลขลำดับ

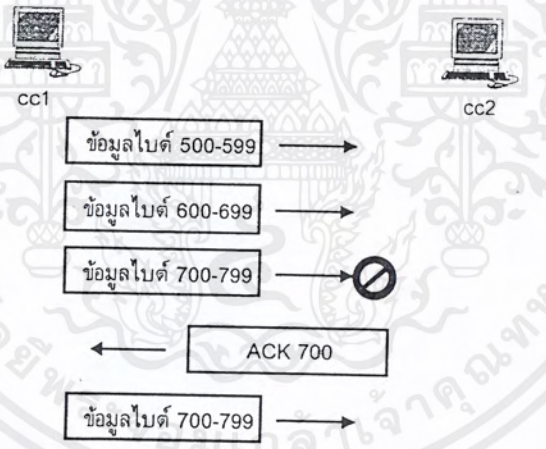
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทีซีพีที่ได้รับเซกเมนต์จะตอบกลับด้วยเลขตอบรับดังตัวอย่างในรูปที่ 2.20 สถานี cc2 ส่งเซกเมนต์กลับมาด้วยค่าเลขตอบรับ 800 ซึ่งหมายถึงได้รับข้อมูลตั้งแต่ต้นถึงเลขลำดับ 799 แล้ว และคาดว่าจะรอรับไบต์ลำดับ 800 ต่อไป




รูปที่ 2.20 การตอบรับด้วยเลขลำดับ

หากไม่มีการตอบรับภายในระยะเวลาที่กำหนดก็จะนำส่งเซกเมนต์นั้นซ้ำใหม่ดังรูปที่ 2.21 เนื่องจากข้อมูลเซกเมนต์ที่สามเกิดสูญหาย ทีซีพีจะส่งเซกเมนต์ซ้ำใหม่




รูปที่ 2.21 การส่งซ้ำหากไม่ได้รับการตอบรับ

เลขลำดับมีขนาด 32 บิต ค่าบิตสูงสุดที่เป็นไปได้คือ $2^{32}$-1 เมื่อเลขลำดับเพิ่มถึงค่าสูงสุดแล้วจะวนกลับมาที่ 0 ใหม่อีกครั้งหนึ่ง แต่ในเครือข่ายความเร็วสูงที่ใช้ในปัจจุบันถึงแม้มีการถ่ายโอนข้อมูลอย่างต่อเนื่องก็จไม่พบปัญหาเลขลำดับวนกลับมาว้ำค่าเดิมในระยะเวลาสั้นๆ ด้วยเหตุนี้หากมีเซกเมนต์ที่มีเลขลำดับซ้ำหับที่เคยได้ตอบรับไปแล้ว ทีซีพีถือว่าเกิดจากปัญหาการส่งเซกเมนต์ซ้ำซึ่งฝ่ายรับจะกำจัดเซกเมนต์นั้นทิ้งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.9.4.5 การเลือกลำดับเลขเริ่มต้น

การเลือกลำดับเลขเริ่มต้นที่ไม่เหมาะสมอาจสร้างปัญหาได้ เช่นกรณีที่ไคลเอ็นต์ส่งเซกเมนต์ออกไปและเครื่องขัดข้องจนต้องปิดเครื่อง หลังจากรีบูตใหม่แล้วหากผู้ใช้สถาปนาการเชื่อมต่อกับเซอร์ฟเวอร์ซ้ำในทันที ไคลเอ็นต์อาจใช้ซ็อกเก็ตชุดเดิมอีกครั้งเพราะสถานะเดิมสูญหายไปเมื่อรีบูตใหม่ ถ้าบางเซกเมนต์ของการเชื่อมโยงเดิมถูกหน่วงเวลาอยู่ในเครือข่ายและเพิ่งเดินทางมาถึงเซอร์ฟเวอร์ซึ่งมีการเชื่อมโยงใหม่ภายใต้ซ็อกเก็ตชุดเดียวกันและมีเลขลำดับเข้ากันได้กับที่ใช้ในการเชื่อมโยงใหม่ เซอร์ฟเวอร์ย่อมไม่มีทางแยกเซกเมนต์ของการเชื่อมโยงเก่าที่ถูกหน่วงเวลาไว้กับเซกเมนต์ของการเชื่อมโยงใหม่ได้

เพื่อแก้ปัญหาเลขลำดับดังกล่าว ทีซีพีเลือกใช้ลำดับเริ่มต้นขนาด 32 บิต มาจากจังหวะของนาฬิกาประจำเครื่องในขณะนั้น เลขลำดับของการเชื่อมโยงเก่าจึงไม่อยู่ในช่วงการเชื่อมโยงใหม่ ยกเว้นในกรณีที่มีการใช้เลขลำดับอย่างรวดเร็วจนกระทั้งตัวเลขวนกลับมาหาค่าเดิมอีกครั้ง

ทีซีพีแก้ปัญกานี้โดยให้เซกเมนต์หนึ่งๆคงอยู่ในเครือข่ายได้ไม่เกินค่าๆหนึ่งเรียกว่า ช่วงชีวิตนานสุดของเซกเมนต์ (Maximum Segment Lifetime : MSL) ค่าโดยปกติที่กำหนดให้ใช้คือ 120 วินาที เลขลำดับที่ทีซีพีใช้จะกำหนดตำแหน่งไบต์ข้อมูลได้ $2^{32}$ ไบต์หรือ $2^{35}$ บิต ทีซีพีจึงไม่สามารถใช้เลขลำดับได้เร็วเกินกว่า $2^{35}$ บิต/120 วินาที หรือประมาณ 286 ล้านบิตต่อวินาที

### 2.9.5 กลไกการทำงานของทีซีพี

ทีซีพีมีกลไกการลำเลียงข้อมูลระหว่างต้นทางและปลายทางหลายส่วน ในที่นี้จะกล่าวเฉพาะกลไกที่สำคัญ 3 ประการก่อนได้แก่ การสถาปนาการเชื่อมโยง การถ่ายโอนข้อมูล และการยกเลิกการเชื่อมโยงตามหัวข้อต่อไปนี้

#### 2.9.5.1 การสถาปนาการเชื่อมโยง

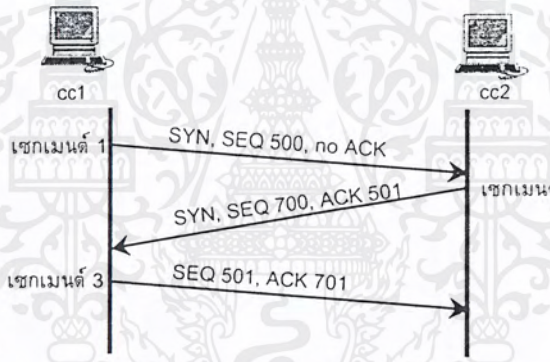
เพื่อให้ผู้อ่านมองเห็นภาพขั้นตอนการทำงานของทีซีพี ขอยกตัวอย่างการสถาปนาการเชื่อมต่อระหว่างสถานีสองเครื่องคือ cc1 จะเป็นไคลเอ็นต์ และโปรโตคอลประยุกต์ด้าน cc2 โดยให้ cc1 เป็นฝ่ายขอบริการจาก cc2 ดังนั้นโปรโตคอลประยุกต์ด้าน cc1 จะเป็นไคลเอ็นต์ และโปรโตคอลประยุกต์ด้าน cc2 จะเป็นเซอร์ฟเวอร์

cc2 จะอยู่ในสภาพที่เรียกว่า การเปิดแบบพาสซีฟ (passive open) คือเซอร์ฟเวอร์สั่งให้ทีซีพีรอรับการเชื่อมต่อ ส่วน cc1 เริ่มการติดต่อเมื่อไคลเอ็นต์สั่งงาน การเปิดการเชื่อมต่อจากไคลเอ็นต์เรียกว่า การเปิดแบบแอกทีฟ (active open) กระบวนการนี้ประกอบด้วยการส่งเซกเมนต์ 3 เซกเมนต์ดังรูปที่ 2.16 และมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทีซีพีของ cc1 เลือกเลขลำดับเริ่มต้น (ในที่นี้คือ 500) แล้วส่งเซกเมนต์ที่บรรจุเลขลำดับนี้ไปยัง cc2 พร้อมทั้งเซ็ตแฟล็ก SYN ให้เป็น "1" ขอให้สังเกตว่าเซกเมนต์นี้ (เซกเมนต์ 1) แฟล็ก ACK จะมีค่าเป็น "0"
2. ทีซีพีของ cc2 ได้รับเซกเมนต์จาก cc1 ก็จะเลือกเลขลำดับเริ่มต้นประจำตัวเช่นเดียวกัน (ในที่นี้คือ 700) แล้วตอบกลับด้วยเซกเมนต์ SYN (เซกเมนต์ 2) พร้อมทั้งเซ็ตแฟล็ก ACK เพื่อแจ้งให้ทราบว่าได้รับเซกเมนต์ 1 โดยใช้เลขลำดับที่ได้รับจากทาง cc1 บวกด้วยหนึ่ง (ในที่นี้คือ 501)
3. ทีซีพีของ cc1 จะส่งเซกเมนต์ตอบรับกลับไป (เซกเมนต์ 3) โดยเซ็ตแฟล็ก ACK และใช้เลขลำดับที่ได้รับจาก cc2 บวกด้วยหนึ่ง

ต่อจากนั้นทีซีพีของ cc1 จะแจ้งไปยังโปรโตคอลระดับบนว่าเชื่อมต่อแล้ว ส่วนทีซีพีของ cc2 เมื่อได้รับ เซกเมนต์ตอบรับ (เซกเมนต์ 3) ก็จะแจ้งขึ้นไปยังโปรโตคอบระดับบนว่าเชื่อมต่อแล้วเช่นกัน เมื่อสิ้นสุดขั้นตอนนี้การสถาปนาทั้งสองด้านก็จะสมบูรณ์และพร้อมจะส่งข้อมูลได้ กระบวนการสถาปนาแบบนี้เรียกว่า "three way handshake" เพราะใช้ 3 เซกเมนต์โต้ตอบระหว่างกัน เซกเมนต์ทั้งสามนี้นิยมเรียกว่า SYN, SYN และ ACK ตามลำดับ



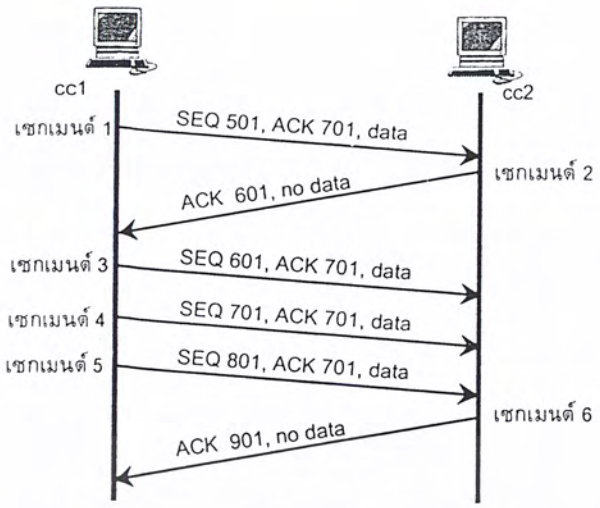

รูปที่ 2.22 การสถาปนาการเชื่อมโยงด้วยทีซีพี

**2.9.5.2 การถ่ายโอนข้อมูล**

การขนถ่ายข้อมูลเริ่มได้หลังจากการสถาปนาเสร็จสิ้นแล้ว รูปที่ 2.23 แสดงขั้นตอนโดยสมมติให้ cc1 ส่งข้อมูลไปยัง cc2 ครั้งละ 100 ไบต์ จำนวน 4 ครั้ง

เซกเมนต์แรกที่ cc1 ส่งไปยัง cc2 บรรจุข้อมูลไบต์ 501 ถึง 600 และใช้ตอบรับ 701 เมื่อ cc2 รับข้อมูลเซกเมนต์ 1 แล้ว ก็จะตอบกลับด้วยเซกเมนต์ 2 พร้อมเลขตอบรับ กำหนดข้อมูลที่คาดว่าจะได้รับต่อไปคือ 601

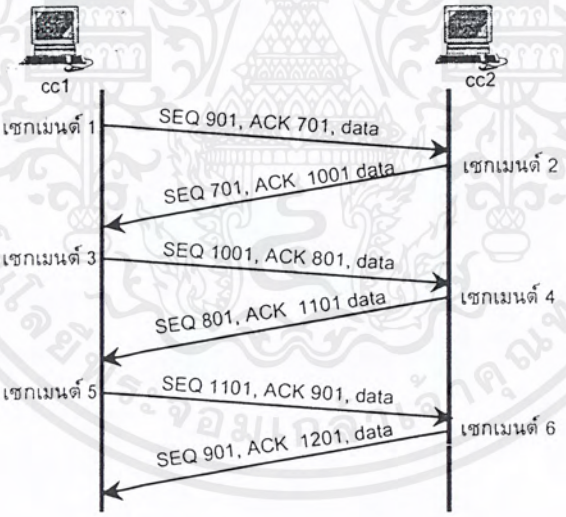
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.23 ขั้นตอนการถ่ายโอนข้อมูล

cc1 สามารถส่งเซกเมนต์ไปอย่างต่อเนื่องได้ เช่นในแผนภาพมีการส่ง 3 เซกเมนต์คือไบต์ 601, 701 และ 801 ทาง cc2 สามารถตอบกลับในคราวเดียวกันได้ (เซกเมนต์ 6)

ในกรณีที่มีการส่งข้อมูลทั้งสองทิศทางระหว่าง cc1 และ cc2 หลักการทั่วไปยังคงเป็นเช่นเดียวกัน เพียงจะมีการตอบรับพร้อมกับส่งข้อมูลระหว่างกันดังรูปที่ 2.24



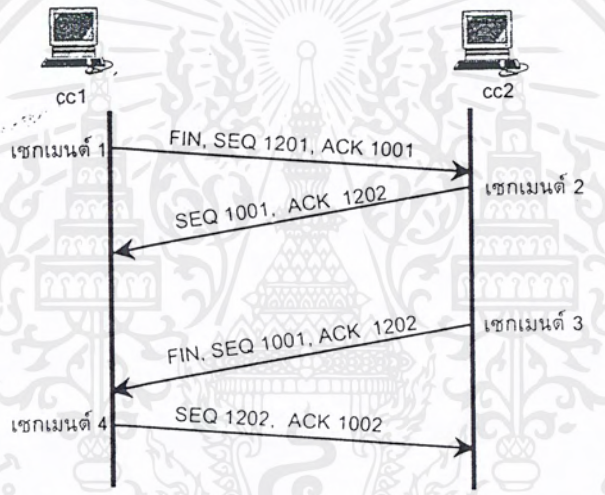

รูปที่ 2.24 การถ่ายโอนเซกเมนต์ทั้งสองทิศทาง

**2.9.5.3 การยกเลิกการเชื่อมต่อ**

เนื่องจากการถ่ายโอนในทีซีพีเป็นแบบฟูลดูเพล็กซ์ การยกเลิกการเชื่อมต่อจึงต้องมีขั้นตอนเกิดขึ้นทั้งสองด้านดังรูปที่ 2.25 ในที่นี้สมมติว่า cc1 ไม่มีข้อมูลส่งอีกต่อไปจึงต้องเป็นฝ่ายขอปิดการเชื่อมต่อ โปรโตคอลประยุกต์ของ cc1 จะแจ้งไปยังทีซีพีว่าส่งข้อมูลหมดแล้ว ถัดจากนั้นจะมีการแลกเปลี่ยนเซกเมนต์จำนวน 4 เซกเมนต์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทีซีพีของ cc1 ส่งเซกเมนต์พร้อมที่มีเลขลำดับและเลขตอบรับตามปกติแต่เซ็ตแฟล็ก FIN เป็น "1"
2. เมื่อทีซีพีของ cc2 ได้รับเซกเมนต์ FIN จะส่งเซกเมนต์ตอบรับ (เซกเมนต์ 2) และแจ้งไปขึ้นยังโปรแกรมประยุกต์ว่า cc1 ขอปิดการเชื่อมต่อ โปรแกรมประยุกต์ของ cc2 จะแจ้งกลับมายังทีซีพีว่าปิดการเชื่อมต่อได้
3. ทีซีพี cc2 ส่งเซกเมนต์ FIN ไปยัง cc1 (เซกเมนต์ 3)
4. เมื่อ cc1 ได้รับเซกเมนต์ FIN จะตอบรับกลับไป (เซกเมนต์ 4) และแจ้งไปยังโปรแกรมประยุกต์ว่าการเชื่อมโยงปิดลงแล้ว

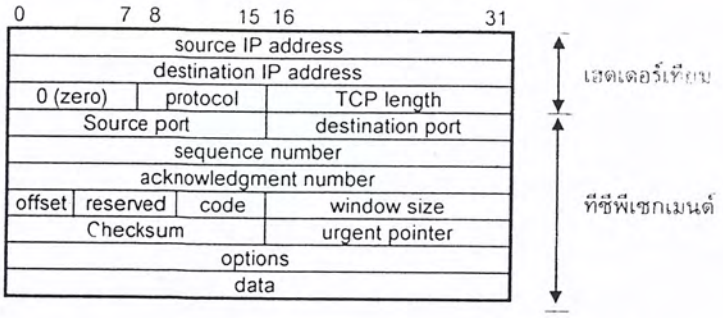



รูปที่ 2.25 การปิดการเชื่อมต่อ

### 2.9.6 การคำนวณผลรวมตรวจสอบ

ค่าผลรวมตรวจสอบในฟิลด์ checksum จะครอบคลุมทั้งส่วนหัวและข้อมูลทีซีพีพร้อมทั้งเฮดเดอร์เทียมดังรูปที่ 2.26 เฮดเดอร์เทียมเป็นเพียงส่วนที่นำมาใช้คำนวณโดยไม่ส่งไปกับทีซีพีเซกเมนต์เช่นเดียวกับที่ใช้ในยูดีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.26 ฟิลด์ที่ใช้คำนวณค่าผลรวมตรวจสอบในทีซีพี

### 2.9.7 ทีซีพีออพชัน

ทีซีพีเฮดเดอร์อาจมีค่าออพชันกำหนดการทำงานเผื่อเลือกเพิ่มเติมได้ ฟิลด์ออพชันของทีซีพีมีขนาดไม่ตายตัวและจัดแบ่งออกได้เป็น 2 รูปแบบคือ

- ออพชันไบต์เดี่ยวที่บรรจุเฉพาะค่าออพชัน
- ออพชันหลายไบต์ที่บรรจุชนิด ความยาว และค่าออพชัน

ในเซกเมนต์หนึ่งๆอาจมีออพชันได้หลายค่าเรียงต่อเนื่องเป็นรายการ อาร์เอฟซี 793 กำหนดให้ใช้ได้ 3 ออพชันคือ ออพชันจบรายการ ออพชันไม่ต้องดำเนินการใด และออพชันขนาดเซกเมนต์สูงสุด ในอาร์เอฟซี 1072, 1185 และ 1323 มีออพชันให้ใช้เพิ่มขึ้น

รูปที่ 2.27 แสดงฟอร์แมตของบางออพชันที่กำหนดในอาร์เอฟซีออพชันทุกแบบจะเริ่มต้นด้วยค่ากำหนดชนิดออพชันขนาด 1 ไบต์ เฉพาะออพชันชนิด 0 และ 1 เท่านั้นที่เป็นออพชันไบต์เดี่ยว ส่วนออพชันอื่นจะมีไบต์ที่สองกำหนดความยาวออพชันและค่าที่ใช้ ในที่นี้จะอธิบายเพียงการใช้งานเบื้องต้นบางออพชัน

#### 2.9.7.1 จบรายการออพชัน

ออพชัน 0 เป็นออพชันไบต์เดี่ยว ใช้เป็นตัวบ่งบอกจุดสิ้นสุดรายการออพชันที่มีในเฮดเดอร์

#### 2.9.7.2 ออพชันไม่ต้องดำเนินการใด

ออพชัน 1 เป็นออพชันแฝงที่ทีซีพีไม่ต้องดำเนินการใด ออพชันนี้ใช้เพื่อให้ขนาดรวมของออพชันทั้งหมดลงตัวเป็นจำนวนเท่าของ 32 บิต เช่นหากใช้ออพชัน 3 ต้องแทรกออพชัน 1 อีกหนึ่งไบต์ (ให้ครบ 4 ไบต์) หากใช้ออพชัน 8 ต้องแทรกออพชัน 1 อีกสองไบต์ (เพื่อให้ครบ 12 ไบต์)

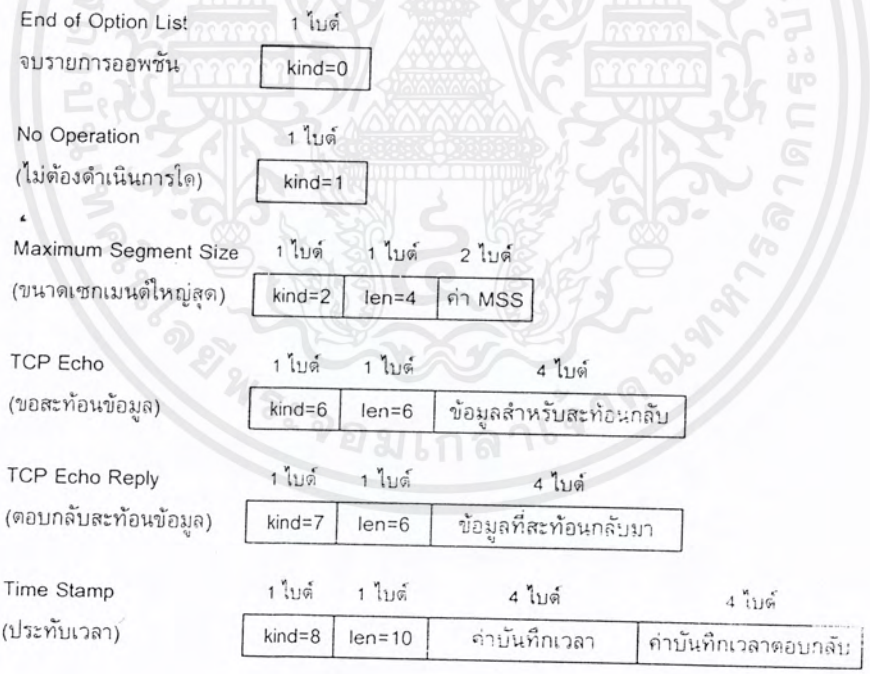
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.7.3 ออพชันขนาดเซกเมนต์ใหญ่สุด

เพื่อให้ทีซีพีสามารถแจ้งถึงขีดจำกัดของขนาดเซกเมนต์ที่รับได้ (ไม่รวมเฮดเดอร์) ทีซีพีจะใช้ออพชันในช่วงการสถาปนาเพื่อแจ้งขนาดใหญ่สุดของข้อมูลในแต่ละเซกเมนต์ที่จะรับได้และเรียกค่านี้ว่าขนาดเซกเมนต์ใหญ่สุด (Maximum Segment Size : MSS)

หากขั้นตอนในการสถาปนาไม่ได้ใช้ออพชันกำหนดค่านี้ ทีซีพีจะใช้ขนาดเซกเมนต์ใหญ่สุดเท่ากับ 536 โดยคำนวณจากขนาดไอพีเดทาแกรมโดยปริยาย (576 ไบต์) ลบด้วยขนาดเฮดเดอร์ต่ำสุดของไอพี (20 ไบต์) และขนาดเฮดเดอร์ต่ำสุดของทีซีพี (20 ไบต์)

เซกเมนต์ขนาดใหญ่ให้ประสิทธิภาพการส่งดีกว่าเซเมนต์ขนาดเล็กโดยเฉพาะการส่งข้อมูลต่อเนื่อง ทั้งนี้เพราะเปอร์เซ็นต์ของโอเวอร์เฮดจากเฮดเดอร์มีค่าน้อยกว่า แต่สถานีและอุปกรณ์เครือข่ายที่มีอยู่อย่างหลากหลายมักรองรับขนาดเซกเมนต์ได้ไม่เท่ากัน การเลือกขนาดเซกเมนต์ยังต้องคำนึงถึงค่าเอ็มทียูเพื่อหลีกเลี่ยงการเกิดไอพีแฟรกเมนต์




รูปที่ 2.27 ทีซีพีออพชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

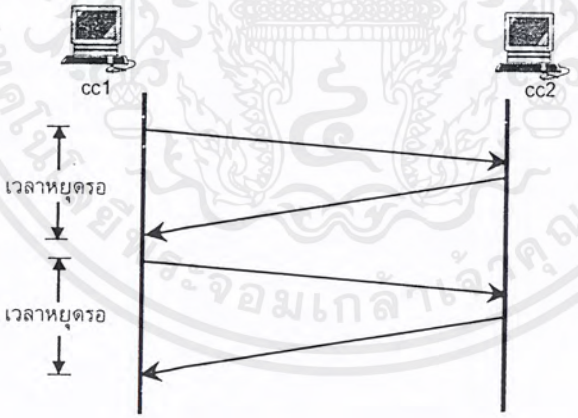
### 2.9.8 การควบคุมกระแสข้อมูล

ทีซีพีอาศัยเลขลำดับเพื่อการจัดส่งข้อมูล เลขลำดับจึงเป็นกุญแจสำคัญในการควบคุมกระแสข้อมูลระหว่างฝ่ายรับและฝ่ายส่ง ไม่ให้ส่งข้อมูลเกินกว่าที่จะรับได้ การควบคุมกระแสข้อมูล (flow control) ของทีซีพีทำงานภายใต้วิธี การเลื่อนหน้าต่าง (sliding windows)

เทคนิคการเลื่อนช่องหน้าต่างไม่ได้เป็นเทคนิคที่ใช้เฉพาะทีซีพีเท่านั้น หากแต่เป็นเทคนิคทั่วไปที่นำมาใช้ในที่โปรโตคอลระดับชั้นทรานสพอร์ตอื่นๆด้วย เพื่อให้ผู้อ่านทำความเข้าใจได้ทั้งกระบวนการทบทวนถึงหลักพื้นฐานและประโยชน์ของเทคนิคดังกล่าวก่อน

#### 2.9.8.1 การเลื่อนหน้าต่าง

ลองพิจารณาถึงสภาวะการถ่ายโอนระหว่างไคลเอ็นต์ cc1 และเซอร์ฟเวอร์ cc2 ดังรูปที่ 2.28 หาก cc1 ใช้หลักการตอบรับเป็นเซกเมนต์คือทุกครั้งที่ส่งเซกเมนต์ใดๆจะต้องรอการตอบจาก cc2 ก่อนที่จะส่งเซกเมนต์ต่อไปได้ ช่วงที่ cc1 รอการตอบรับก็จะเป็นช่วงหยุดที่ไม่สามารถนำส่งข้อมูลได้และเท่ากับเป็นการใช้สายสื่อสารแบบฮาล์ฟดูเพล็กซ์ซึ่งสูญเสียแบนด์วิดธ์ไปครึ่งหนึ่ง เนื่องจากต้องรอให้อีกด้านหนึ่งส่งเสร็จสิ้นก่อน การตอบรับโดยไม่มีเทคนิคการเลื่อนช่องหน้าต่างจึงทำให้เสียประสิทธิภาพการถ่ายโอนข้อมูล

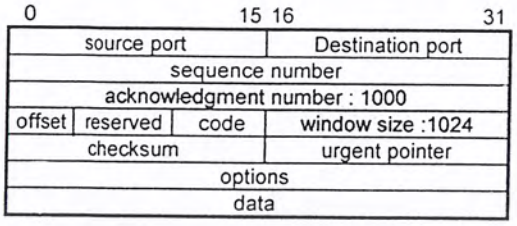



รูปที่ 2.28 การส่งถ่ายแบบฮาล์ฟดูเพล็กซ์

ทีซีพีแก้ปัญหานี้โดยทำงานแบบฟูลดูเพล็กซ์ ฝ่ายส่งสามารถส่งเซกเมนต์ออกไปได้หลายเซกเมนต์โดยไม่ต้องรอการตอบรับทุกๆเซกเมนต์ แต่ทีซีพีต้องดำเนินการควบคุมปริมาณการส่งนี้เพราะบัฟเฟอร์ฝ่ายรับย่อมมีขนาดจำกัด ทีซีพีฝ่ายรับจะต้องแจ้งให้ทางฝ่ายส่งถึงขนาดข้อมูลที่รับได้โดยใช้เลขตอบรับ (ฟิลด์ acknowledgement number) และขนาดหน้าต่าง (ฟิลด์ window) สองค่านี้จึงใช้เป็นตัวกำหนดขนาดการรับส่งข้อมูลแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

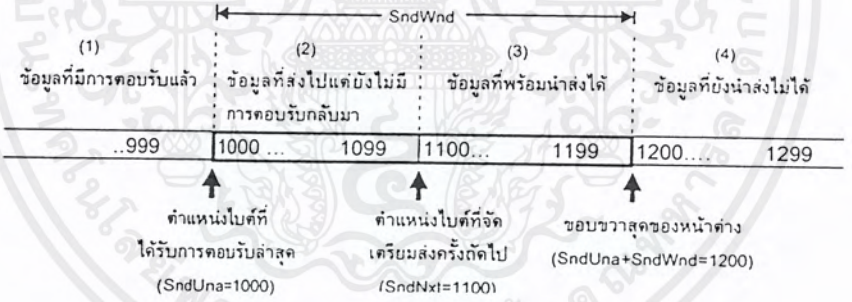
รูปที่ 2.29 เป็นตัวอย่างทีซีพีเฮดเดอร์ซึ่งประกาศหน้าต่างขนาด 1024 ไบต์ ค่าในฟิลด์บ่งบอกว่าสถานีอีกด้านหนึ่งได้รับข้อมูลตั้งแต่ถึงลำดับ 999 ครบถ้วนแล้ว และพร้อมรับข้อมูลไม่เกินเลขลำดับ 2023 (คำนวณจาก 1000-1+1024)




รูปที่ 2.29 ฟิลด์ในทีซีพีเฮดเดอร์ที่ใช้ควบคุมกระแสข้อมูล

### 2.9.8.2 หน้าต่างฝ่ายส่ง

ทีซีพีควบคุมกระแสข้อมูลที่ฝ่ายส่งโดยใช้ค่า 3 ค่าคือ ตัวแปรชี้ตำแหน่งไบต์ที่ตอบรับ (SndUna), ตัวแปรชี้ตำแหน่งไบต์ที่ต้องส่งครั้งถัดไป (SndNxt) และ ตัวแปรกำหนดขนาดหน้าต่างที่ส่งได้ (SndWnd) ค่าทั้ง 3 จะแบ่งบัฟเฟอร์ของฝ่ายส่งออกเป็น 4 ส่วนดังรูปที่ 2.30




รูปที่ 2.30 หน้าต่างของฝ่ายส่ง

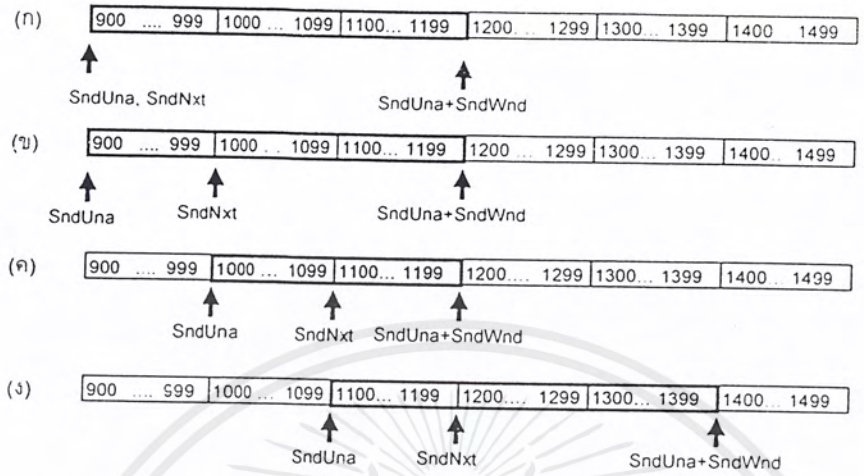
เพื่อให้เห็นจึงมีการปรับหน้าต่าง ขอให้พิจารณารูปที่ 2.31 โดยให้ในสภาวะเริ่มต้นทางฝ่ายส่งมีขนาดของหน้าต่างเท่ากับ 300 ไบต์ และหมายเลขลำดับเริ่มต้นที่ 900 ดังรูปที่ 2.31 (ก)

เมื่อส่งข้อมูลออกไป 100 ไบต์แรก SndNxt จะเลื่อนไปทางขวา 100 ไบต์ ตัวแปรจะมีลักษณะดังรูปที่ 2.31 (ข) ความจำเป็นที่ต้องมีตัวแปรชี้ตำแหน่งไบต์ที่ได้รับการตอบรับล่าสุดก็เพื่อใช้เก็บตำแหน่งข้อมูลที่ส่งไปแล้วแต่ยังไม่ได้ตอบรับเนื่องจากอาจส่งซ้ำใหม่

ในขั้นตอนถัดไปหากมีการส่งข้อมูล 100 ไบต์ SndNxt จะขยับไปอยู่ที่ตำแหน่ง 1100 และเมื่อมีการตอบรับข้อมูล 100 ไบต์แรกกลับมา SndUna จะเลื่อนไปที่ตำแหน่ง 1000 ตามรูปที่ 2.31 (ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นถัดไปหากมีการตอบรับกลับมาพร้อมกับการกำหนดขนาดหน้าต่าง 300 ไบต์ หน้าต่างจะเป็นดังรูปที่ 2.31 (ง)

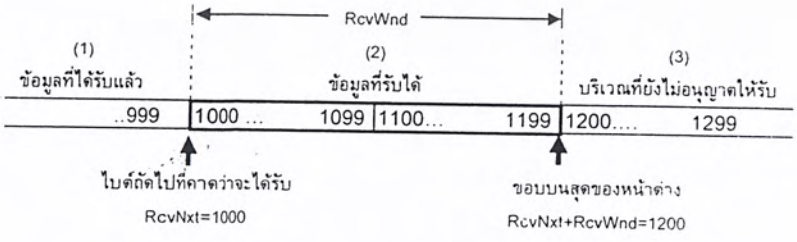



รูปที่ 2.31 การเลื่อนช่องหน้าต่าง

โปรดพิจารณาว่ากรอบเข้มเป็นเสมือนกรอบหน้าต่างซึ่งเลื่อนไปตามลักษณะการส่งข้อมูล โดยจะลดขนาดลงเมื่อส่งข้อมูลออกไปแต่ไม่ได้รับการประกาศจากฝ่ายรับให้ขยายขนาด และจะขยายขนาดเมื่อรับเซกเมนต์ที่กำหนดขนาดหน้าต่าง

### 2.9.8.3 หน้าต่างฝ่ายรับ

ทีซีพีฝ่ายรับจัดการบัฟเฟอร์โดยให้มี ตัวแปรบอกขนาดหน้าต่าง (RcvWnd) และตัวแปรชี้ตำแหน่งข้อมูลถัดไปที่คาดว่าจะได้รับ (RcvNxt) ตัวแปรทั้งสองนี้จะแบ่งบัฟเฟอร์ออกเป็น 3 ส่วนดังรูปที่ 2.32



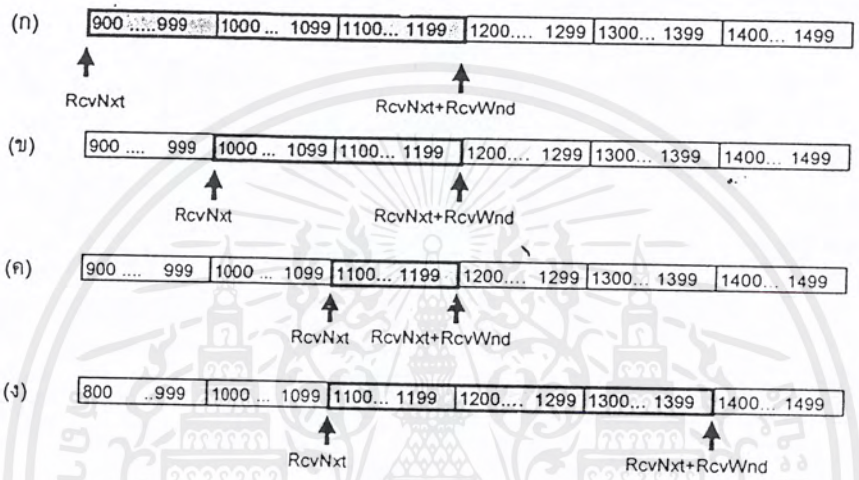

รูปที่ 2.32 หน้าต่างฝ่ายรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากพิจารณาการเลื่อนหน้าต่างของฝ่ายรับโดยกำหนดให้มีขนาดของหน้าต่างเริ่มต้น 300 ไบต์ดังรูปที่ 2.33 (ก) เมื่อข้อมูล 100 ไบต์แรกเดินทางมาถึง หน้าต่างข้อมูลที่รับได้จะลดลงเหลือ 200 ไบต์ รูปที่ 2.33 (ข)

ในขั้นต่อมาหากทีซีพีได้รับข้อมูลอีก 100 ไบต์ ขนาดของหน้าต่างข้อมูลก็จะลดลงเหลือ 100 ไบต์ ดังรูปที่ 2.33(ค)

ในขั้นถัดมาเมื่อโปรโตคอลประยุกต์นำข้อมูลไปใช้ หน้าต่างก็จะเลื่อนและขยายขนาดขึ้นดังรูปที่ 2.33 (ง)

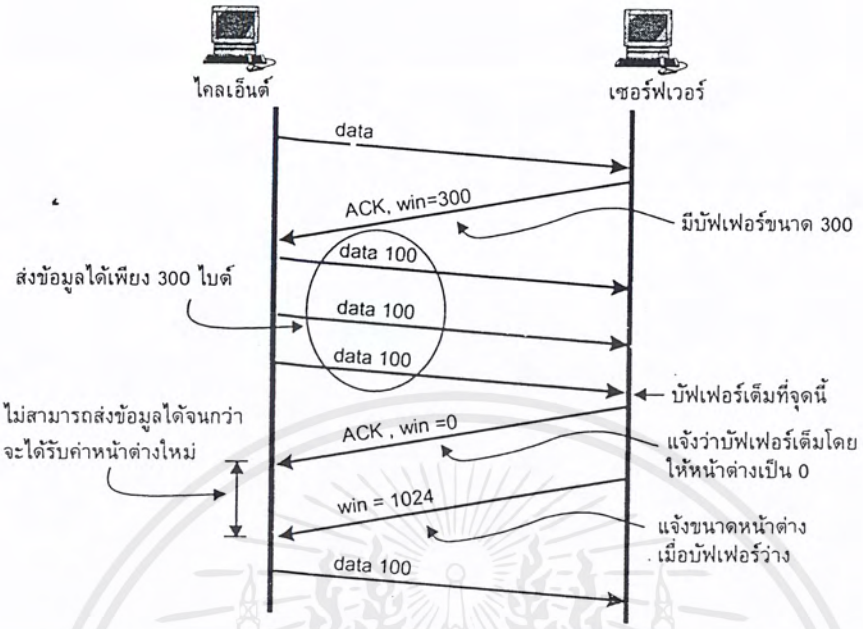



รูปที่ 2.33 การเลื่อนหน้าต่างฝ่ายรับ

กรณีที่บัฟเฟอร์ฝ่ายรับเต็มจนไม่สามารถรับเซกเมนต์เพิ่มได้ ทีซีพีจะส่งเซกเมนต์กำหนดขนาดหน้าต่างโดยให้ฟิลด์ window size มีค่าเท่ากับ 0 ทีซีพีอีกฝ่ายจะหยุดส่งเซกเมนต์จนกว่าจะได้รับเซกเมนต์ที่กำหนดขนาดหน้าต่างใหม่ดังรูปที่ 2.34

การรอประกาศค่าหน้าต่างใหม่เป็นกรณีหนึ่งที่ทีซีพีต้องดำเนินการเป็นกรณีพิเศษเพราะหากเซกเมนต์ประกาศค่าหน้าต่างใหม่เกิดสูญหาย จะทำให้การสื่อสารไม่สามารถดำเนินต่อไปได้ ทีซีพีแก้ปัญหากรณีนี้โดยใช้ตัวจับเวลาเพื่อให้รับประกันว่าต้องส่งเซกเมนต์กำหนดค่าหน้าต่างใหม่ภายในเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.34 การประกาศขนาดหน้าต่าง

### 2.9.8.4 ขนาดหน้าต่างเพื่อการส่งอย่างต่อเนื่อง

หากฝ่ายรับมีบัฟเฟอร์จำนวนมากย่อมสามารถจะประกาศค่าหน้าต่างขนาดใหญ่ได้ และให้ฝ่ายส่งสามารถรักษาสภาพการส่งอย่างต่อเนื่องโดยไม่ถูกขัดจังหวะ สำหรับเครือข่ายที่มีเวลาหน่วงสูงจากการส่งข้อมูลไปและรอการตอบรับหากให้มีการส่งได้อย่างต่อเนื่องแล้ว ฝ่ายรับจะต้องมีขนาดของบัฟเฟอร์สอดคล้องกับทั้งแบนด์วิดธ์และเวลาหน่วง

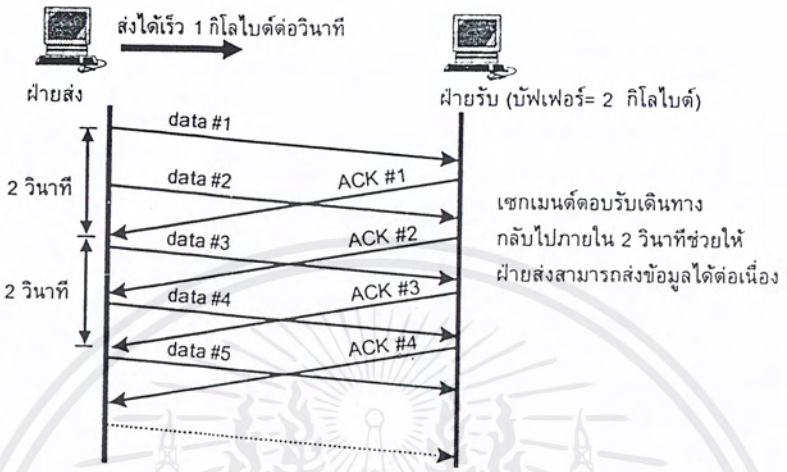
ตัวอย่างเช่นหากฝ่ายส่งสามารถส่งข้อมูลได้ 1 กิโลไบต์ต่อวินาที และการตอบรับใช้เวลา 2 วินาที ฝ่ายรับจะต้องมีบัฟเฟอร์ขนาด 2 กิโลไบต์เพื่อให้อีกฝ่ายส่งได้อย่างต่อเนื่องดังรูปที่ 2.35 โดยสมมติว่าโปรโตคอลประยุกต์สามารถนำข้อมูลออกจากบัฟเฟอร์ไปในทันที

จากรูปที่ 2.35 หากฝ่ายรับมีบัฟเฟอร์เพียง 1 กิโลไบต์ ฝ่ายส่งจะถูกบังคับให้หยุดรอทุกๆ 1 วินาที ก่อนที่จะส่งข้อมูลถัดไป ดังนั้นขนาดบัฟเฟอร์ที่เพียงพอต่อการรักษาการส่งอย่างต่อเนื่องจะต้องมีขนาดไม่น้อยกว่าผลคูณระหว่างแบนด์วิดธ์กับเวลาหน่วงตามสูตร

window = bandwidth * delay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่นให้หากมีแบนด์วิดธ์ขนาด 1 เมกะไบต์ต่อวินาที และค่าเวลาหน่วงคือ 2 มิลลิวินาทีผลคูณที่ได้คือขนาดบัฟเฟอร์ 2 กิโลไบต์ ค่านี้เรียกโดยทั่วไปว่า ผลคูณแบนด์วิดธ์เวลาหน่วง (Bandwidth-delay product)



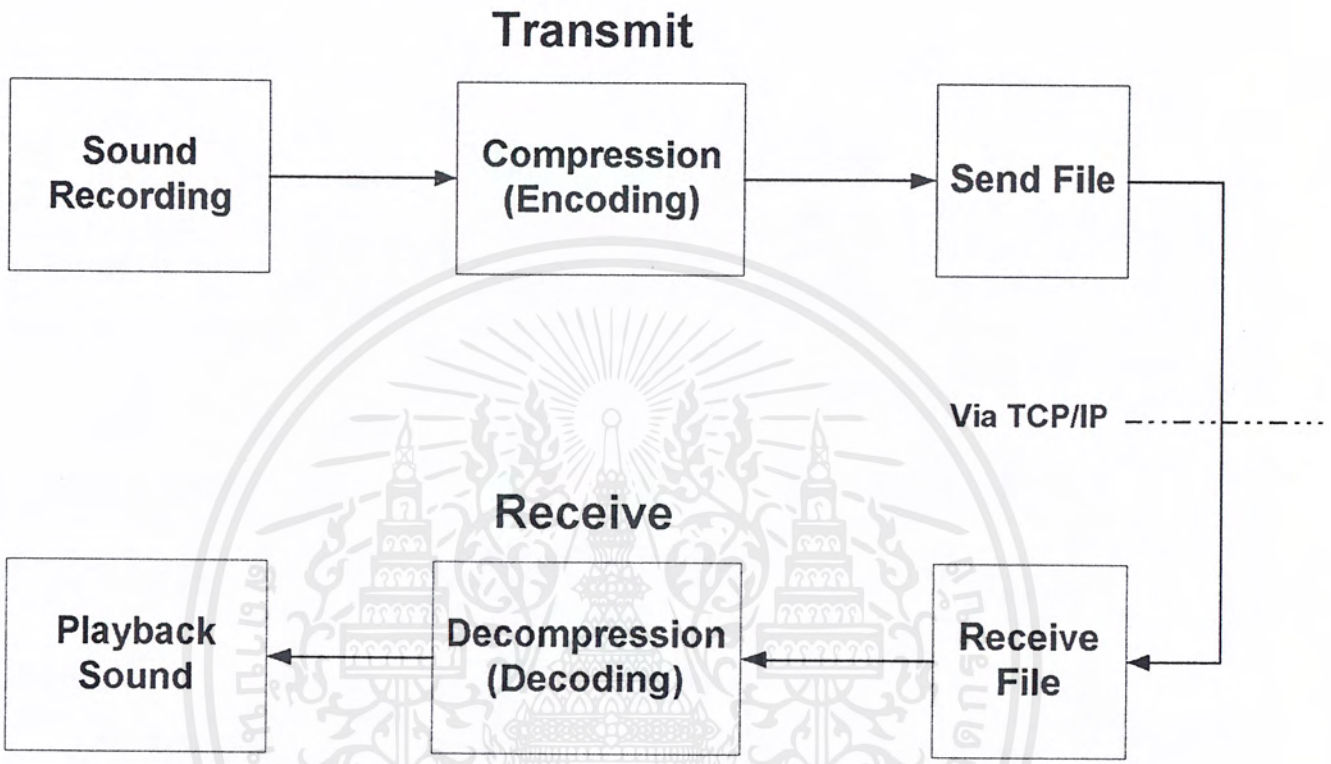

รูปที่ 2.35 การส่งเซกเมนต์อย่างต่อเนื่อง

การส่งข้อมูลต่อเนื่องอาจทำไม่ได้เสมอไป เครือข่ายที่มีแบนด์วิดธ์หนึ่งเมกะไบต์ต่อวินาทีและค่าเวลาหน่วงเท่ากับ 100 มิลลิวินาที ขนาดบัฟเฟอร์ควรมีค่าเท่ากับ 100 กิโลไบต์ซึ่งเกินกว่าขนาดหน้าต่างในฟิลด์ window ที่มีได้ (ค่า window มากสุดเท่ากับ $2^{16}-1$ หรือ 64 กิโลไบต์) ปัญหานี้แก้ไขได้โดยใช้ออพชันขยายขนาดหน้าต่าง เพื่อให้สามารถส่งข้อมูลด้วยขนาดหน้าต่างที่ใหญ่ขึ้นโดยไม่ต้องแก้ไขทีซีพีเฮดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การคำนวณและการสร้าง

บล็อกไดอะแกรมโดยรวมของโปรแกรมมีดังนี้




รูปที่ 3.1 บล็อกไดอะแกรมโดยรวมของโปรแกรม

โปรแกรมโดยรวมทั้งหมดจะแบ่งออกเป็นทั้งหมด 5 ส่วนหลัก คือ

3.1 ส่วน Sound Recording

3.2 ส่วน Compression(Encoding)

3.3 ส่วน Send&Recieve File Via TCP/IP

3.4 ส่วน Decompression(Decoding)

3.5 ส่วน Playback Sound

โดยที่ทั้ง 5 ส่วนนี้ จะอยู่ใน 4 unit ของโปรแกรม ได้แก่

- unit NetVoiceP
- unit NetVoiceEncodeLib
- unit NetVoiceDecodeLib
- unit NetVoicePlayP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแต่ละส่วนของบล็อกไดอะแกรมมีรายละเอียดดังนี้

**3.1 ส่วน Sound Recording**

ในส่วนนี้เราจะใช้คอมโพเนนท์(component) ของเดลไฟเข้ามาช่วยซึ่งก็คือ คอมโพเนนท์ TmediaPlayer โดยที่ในส่วนนี้จะมีโพรซีเยอร์ (procedure) ต่างๆที่เกี่ยงข้องดังนี้

3.1.1 procedure TNetVoiceF.CloseMedia

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์ปิดไฟล์

3.1.2 procedure TNetVoiceF.OpenMedia

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์เปิดไฟล์

3.1.3 procedure TNetVoiceF.RecordMedia

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์ทำการบันทึกเสียง

3.1.4 procedure TNetVoiceF.SaveMedia

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์บันทึกไฟล์

3.1.5 procedure TNetVoiceF.SetWaveParameters

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์กำหนดค่าพารามิเตอร์ของไฟล์ .wav

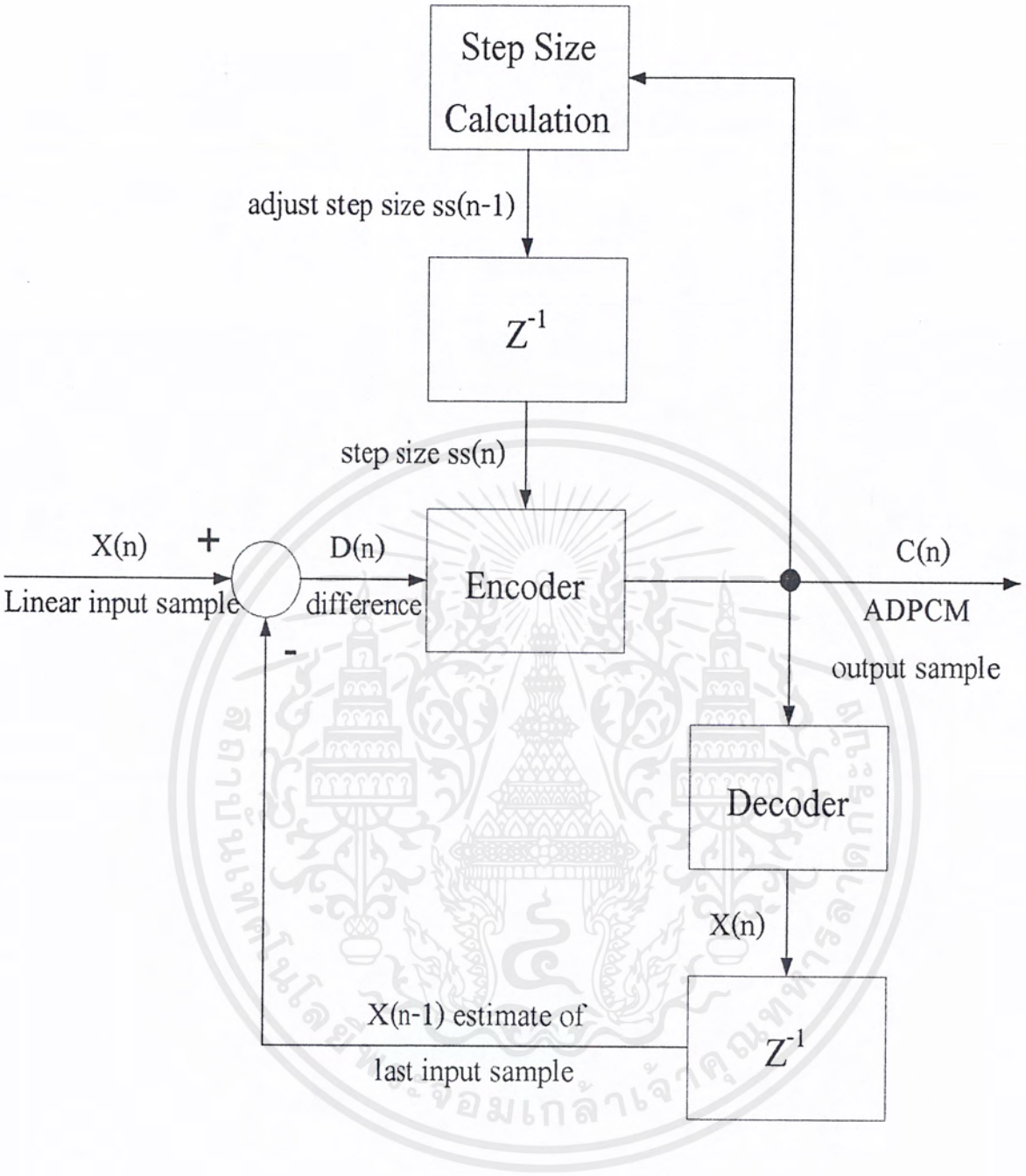
3.1.6 procedure TNetVoiceF.StopMedia

ทำหน้าที่สั่งให้มัลติมีเดียของวินโดว์หยุดการทำงาน

**3.2 ส่วน Compression (Encoding)**

ในส่วนของกระบวนการเข้ารหัสแบบเอดีพีซีเอ็ม สามารถแสดงด้วยบล็อกไดอะแกรม ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.2 บล็อกไดอะแกรมการเข้ารหัสแบบเอดีพีซีเอ็ม

จากรูปที่ 3.2 กำหนดให้

| | | |
|---|---|---|
| X(n) | = | สัญญาณอินพุทปัจจุบัน (Input signal) |
| X(n-1) | = | ค่าประมาณของสัญญาณอินพุทตัวก่อน (Previous predicted sample) |
| d(n) | = | ค่าผลต่างของสัญญาณ (Difference signal) |
| ss(n) | = | ค่าสเต็บไซส์ของสัญญาณอินพุทปัจจุบัน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ss(n-1) = ค่าสเต็บไซส์ของสัญญาณอินพุทตัวก่อน

C(n) = รหัสไบนารี 4 บิท (ADPCM output) ซึ่งบิทที่4 จะเป็นบิทเครื่องหมาย "0" แทนค่าบวก "1" แทนค่าลบ

X(n) จะต้องถูกนำไปเปรียบเทียบเทียบกับ X(n-1) เพื่อหาค่าผลต่างของสัญญาณ d(n) จากนั้นเราจะใช้ค่าผลต่างนี้ร่วมกับค่า ss(n) เพื่อหารหัสไบนารี 4 บิท ซึ่งก็คือค่าเอดีพีซีเอ็มที่ต้องการ รหัสเอดีพีซีเอ็มนี้จะถูกป้อนกลับมาใช้ในการหาค่าสเต็บไซส์ตัวถัดไป และยังนำมาใช้ในการคำนวณค่าประมาณของสัญญาณอินพุทตัวถัดไปเพื่อนำไปใช้การหาค่าผลต่างกับสัญญาณอินพุทตัวถัดไปอีกด้วย

ขั้นตอนการคำนวณ สามารถทำได้ดังนี้

1.) สภาวะเริ่มต้น (Initial condition)

ที่สภาวะเริ่มต้น เราสามารถที่จะกำหนดค่าได้ 2 ค่า คือ

- ค่าประมาณของสัญญาณอินพุทตัวก่อน ซึ่งมีค่าได้ตั้งแต่ –4096 ถึง 4096
- ค่าอินเด็กซ์ ซึ่งเป็นค่าที่ใช้ในการปรับระดับสเต็บไซส์ ซึ่งมีค่าได้ตั้งแต่ 0-48

2.) หาค่าผลต่างของสัญญาณจากสมการ

$$d(n) = X(n) - X(n-1) \qquad .....(3.1)$$

3.) หาค่ารหัสไบนารี 4 บิท (ADPCM output) โดยการนำค่า d(n) ไปทำการเข้ารหัสซึ่งทำได้ดังนี้

3.1) เริ่มต้นกำหนดให้รหัสไบนารี 4 บิท $B_3B_2B_1B_0$ : 0000

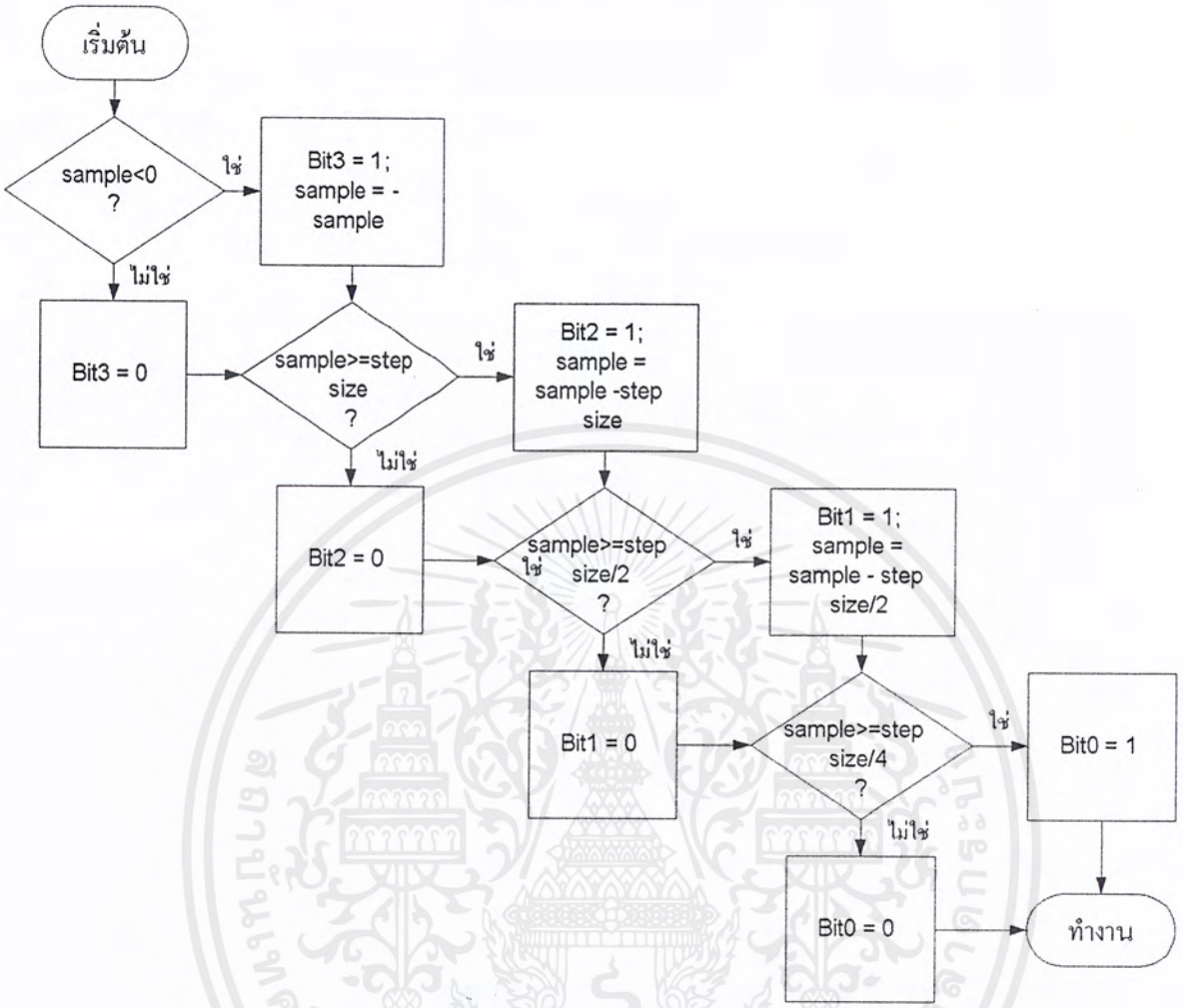
3.2) พิจารณาค่าผลต่างของสัญญาณ D[n]

- ถ้า $d(n) < 0$ ให้ $B_3 = 1$ จากนั้น $d(n) = |d(n)|$
- ถ้า $d(n) >=$ step size ให้ $B_2 = 1$ จากนั้น $d(n) = d(n) -$ step size
- ถ้า $d(n) >=$ step size/2 ให้ $B_1 = 1$ จากนั้น $d(n) = d(n) -$ step size/2
- ถ้า $d(n) >=$ step size/4 ให้ $B_0 = 1$

จะได้ค่ารหัสไบนารี $C(n) = B_3B_2B_1B_0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการเข้ารหัส(Encoder) จากบล็อกไดอะแกรม สามารถแสดงด้วยแผนผังการทำงานดังรูป




รูปที่ 3.3 แผนผังบล็อกการเข้ารหัส (Encoder)

4.) การคำนวณหาค่าสเต็บไซส์

การคำนวณหาค่าสเต็บไซส์จะอาศัยตารางที่ 3.1 และ 3.2 มีหลักการดังนี้

4.1) จากรหัสไบนารี 4 บิทเอดีพีซีเอ็ม ตรวจสอบค่า B2-B0 (magnitude) โดยใช้ตารางที่ 3.1 เทียบหาค่าอินเด็กซ์ แอดจัสต์เมนต์ (Index Adjustment)

4.2) นำค่าอินเด็กซ์ แอดจัสต์เมนท์ ที่หาได้ไปบวกกับค่าอินเด็กซ์ แอดจัสต์เมนท์ ที่ตรงกับค่าสเต็บไซส์เดิม จะได้ค่าอินเด็กซ์ แอดจัสต์เมนต์ ตัวใหม่

4.3) โดยใช้ตารางที่ 3.2 หาค่าสเต็บไซส์ ตัวใหม่ได้โดยเลือกเอาค่าสเต็บไซส์ตัวที่ตรงกับค่าอินเด็กซ์ แอดจัสต์เมนทต์ ที่คำนวณได้จากข้อ 4.2) จะได้ค่าสเต็บไซส์ ไปใช้ในการคำนวณกับค่าสัญญาณอินพุทตัวถัดไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 4 Bits Binary Code ( C[n] ) | Index Adjustment |
|---|---|
| 1111 or 0111 | +8 |
| 1110 or 0110 | +6 |
| 1101 or 0101 | +4 |
| 1100 or 0100 | +2 |
| 1011 or 0011 | -1 |
| 1010 or 0010 | -1 |
| 1001 or 0001 | -1 |
| 1000 or 0000 | -1 |

ตารางที่ 3.1 ตารางแสดงค่าอินเด็กซ์แอดจัสต์เมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Index | Step Size | Index | Step Size |
|---|---|---|---|
| 1 | 16 | 25 | 157 |
| 2 | 17 | 26 | 173 |
| 3 | 19 | 27 | 190 |
| 4 | 21 | 28 | 209 |
| 5 | 23 | 29 | 230 |
| 6 | 25 | 30 | 253 |
| 7 | 28 | 31 | 279 |
| 8 | 31 | 32 | 307 |
| 9 | 34 | 33 | 337 |
| 10 | 37 | 34 | 371 |
| 11 | 41 | 35 | 408 |
| 12 | 45 | 36 | 449 |
| 13 | 50 | 37 | 494 |
| 14 | 55 | 38 | 544 |
| 15 | 60 | 39 | 598 |
| 16 | 66 | 40 | 658 |
| 17 | 73 | 41 | 724 |
| 18 | 80 | 42 | 796 |
| 19 | 88 | 43 | 876 |
| 20 | 97 | 44 | 963 |
| 21 | 107 | 45 | 1060 |
| 22 | 118 | 46 | 1166 |
| 23 | 130 | 47 | 1282 |
| 24 | 143 | 48 | 1411 |
| | | 49 | 1552 |

ตารางที่ 3.2 ตารางที่ใช้ในการคำนวณค่าสเต็บไซส์

### 3.3 ส่วน Send&Receive File Via TCP/IP

ในส่วนนี้เราจะใช้คอมโพเนนท์ของเดลไฟเข้ามาช่วยซึ่งก็คือ คอมโพเนนท์ NMStrm1 กับคอมโพเนนท์ NMStrmServ1 โดยที่ในส่วนนี้จะมีโพรซีเยอร์ ต่างๆที่เกี่ยงข้องดังนี้

3.3.1 procedure TNetVoiceF.NMStrm1Connect

ให้ status bar แสดงข้อความ "Connected"

3.3.2 procedure TNetVoiceF.NMStrm1ConnectionFailed

ให้ status bar แสดงข้อความ "Connection Failed"

3.3.3 procedure TNetVoiceF.NMStrm1Disconnect

ให้ status bar แสดงข้อความ "Disconnected"

3.3.4 procedure TNetVoiceF.NMStrm1HostResolved

ให้ status bar แสดงข้อความ "Host Resolved"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 procedure TNetVoiceF.NMStrm1InvalidHost

ให้ status bar แสดงข้อความ "Invalid Host , Specify a new host :"

3.3.6 procedure TNetVoiceF.NMStrm1MessageSent

ให้ status bar แสดงข้อความ "Send complete"

3.3.7 procedure TNetVoiceF.NMStrm1PacketSent

ให้ status bar แสดงจำนวนไบต์ที่ส่ง

3.3.8 procedure TNetVoiceF.NMStrmServ1ClientContact

ให้ status bar แสดงข้อความ "Client Connected"

3.3.9 procedure TNetVoiceF.NMStrmServ1MSG

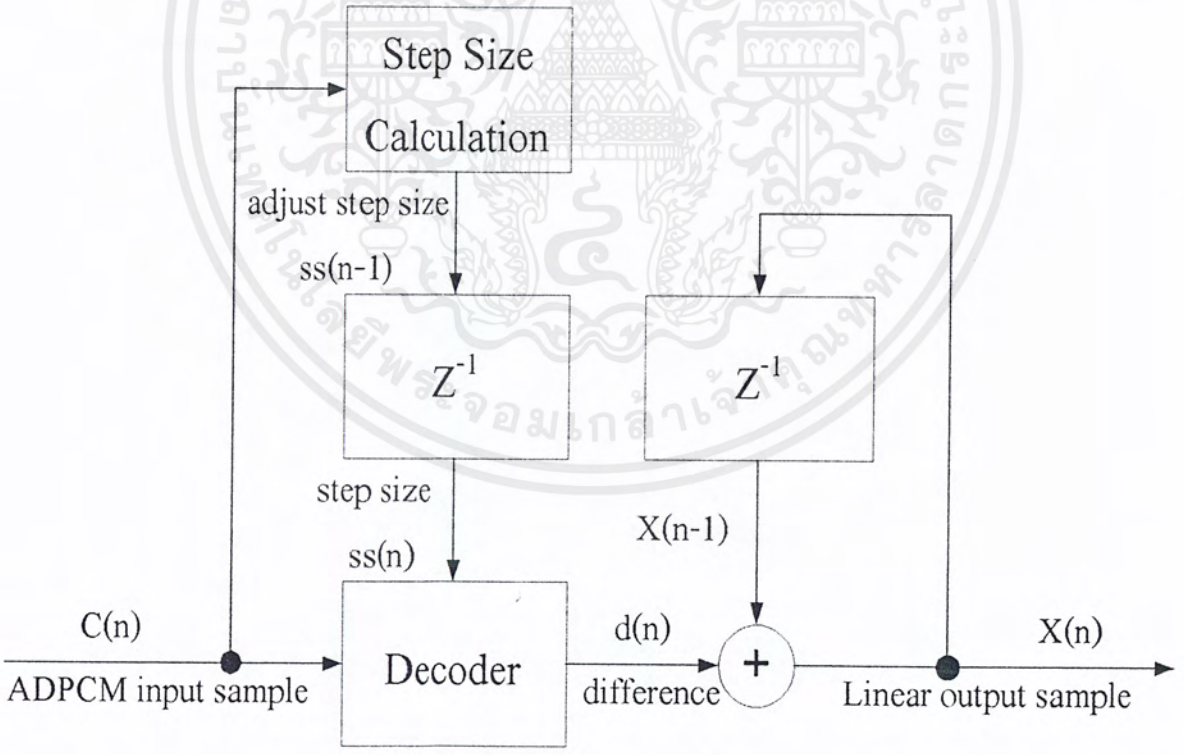
ส่วนนี้ทำหน้าที่เช็คไฟล์ที่รับมาว่ามีรึไม่

3.3.10 procedure TNetVoiceF.NMStrmServ1Disconnect

ให้ status bar แสดงข้อความ "Client Disconnected"

### 3.4.1 ส่วน Decompression (Decoding)

ในส่วนของกระบวนการถอดรหัสแบบเอดีพีซีเอ็ม สามารถแสดงด้วยบล็อกไดอะแกรม ดังรูป




รูปที่ 3.4 บล็อกไดอะแกรมการถอดรหัสแบบเอดีพีซีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.4 กำหนดให้

| | | |
|---|---|---|
| X(n) | = | สัญญาณอินพุทปัจจุบัน (Input signal) |
| X(n-1) | = | ค่าประมาณของสัญญาณอินพุทตัวก่อน (Previous predicted sample) |
| d(n) | = | ค่าผลต่างของสัญญาณ (Difference signal) |
| ss(n) | = | ค่าสเต็บไซส์ของสัญญาณอินพุทปัจจุบัน |
| ss(n-1) | = | ค่าสเต็บไซส์ของสัญญาณอินพุทตัวก่อน |
| C(n) | = | รหัสไบนารี 4 บิท (ADPCM output) ซึ่งบิทที่4 จะเป็นบิทเครื่องหมาย "0" แทนค่าบวก "1" แทนค่าลบ |

จากรหัสไบนารี 4 บิท เอดีพีซีเอ็ม C(n) นำมาใช้คำนวณหาค่าสเต็บไซส์และนำค่าสเต็บไซส์ที่ได้ไปทำการคำนวณเพื่อให้ได้ค่าผลต่างของสัญญาณและนำไปเปรียบเทียบกับค่าประมาณของสัญญาณอินพุทตัวก่อน

ขั้นตอนการคำนวณ สามารถทำได้ดังนี้

1.) หาค่าผลต่างของสัญญาณกลับคืนมาโดยใช้สมการ

$d(n) = ( \text{step size} * B_2 ) + ( \text{step size}/2 * B_1 ) + (\text{step size}/4 * B_0 ) + \text{step size}/8$ .....3.2

ตรวจสอบค่าของ $B_3$

ถ้า $B_3 = 1$ $d(n) = d(n) * (-1)$ .....3.3

ถ้า $B_3 = 0$ $d(n) = d(n)$ .....3.4

2.) หาค่าประมาณของสัญญาณอินพุทตัวถัดไปจากค่าประมาณของสัญญาณอินพุทตัวก่อน ดังสมการ

$X(n) = X(n-1) + d(n)$ .....3.5

### 3.5 ส่วน Playback Sound

ในส่วนนี้เราจะใช้คอมโพเนนท์(component) ของเดลไฟเข้ามาช่วยซึ่งก็คือ คอมโพเนนท์ TmediaPlayer โดยที่ในส่วนนี้จะมีโพรซีเยอร์ (procedure) ต่างๆที่สำคัญดังนี้

- procedure TNetVoicePlayP.OpenWave

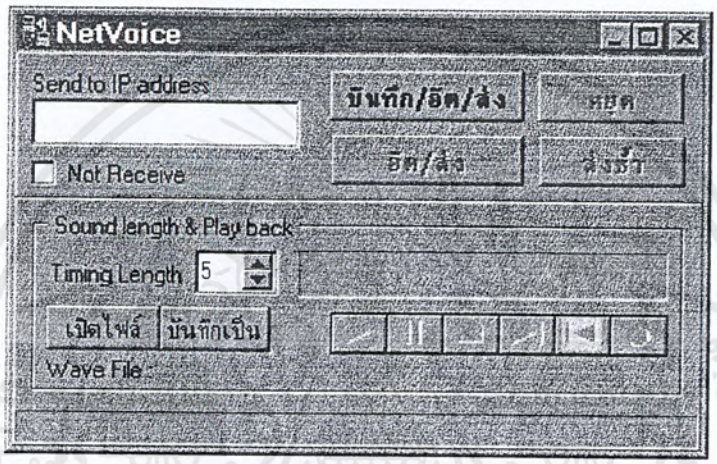
เป็นส่วนที่ทำหน้าที่เล่นไฟล์ที่รับมาจากด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในโปรแกรมนี้แบ่งออกเป็น 4 ยูนิต ดังที่ได้กล่าวไปแล้ว รายละเอียดของแต่ละยูนิตมีดังนี้

**-unit NetVoiceP**

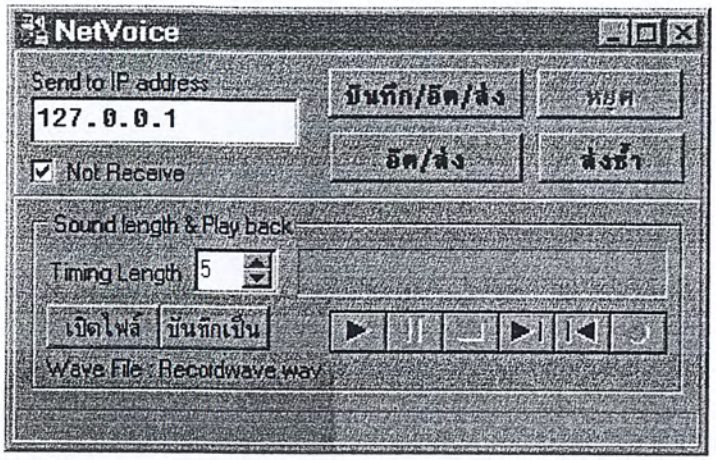
ยูนิตนี้จะเป็นส่วนของฟอร์มหลักของโปรแกรมเป็นส่วนของบล็อก Sound Recording และบล็อก Send&Receive file via TCP/IP และยูนิตนี้ยังเรียกใช้ยูนิตที่เหลืออีก 3 ยูนิตอีกด้วย ซึ่งฟอร์มของยูนิตนี้เป็นดังนี้




รูปที่ 3.5 แสดงฟอร์มของ unit NetVoiceP

โดยที่เมื่อพร้อมจะส่งฟอร์มจะมีลักษณะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3. 6 แสดงฟอร์มของ unit NetVoiceP เมื่อพร้อมทำการส่ง

**- unit NetVoicePlayP**

ยูนิตนี้เป็นอีกฟอร์มของโปรแกรม เป็นส่วนของบล็อก Playback Sound และยูนิตนี้ยังเรียกใช้ unit NetVoiceP และ unit NetVoiceDecodeLib อีกด้วย ซึ่งฟอร์มของยูนิตนี้มีลักษณะเป็นดังนี้

รูปที่ 3. 7 แสดงฟอร์มของ unit NetVoicePlayP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **unit NetVoiceEncodeLib**

ยูนิตนี้ไม่มีฟอร์มแต่เป็นไลบรารี(library) ที่ให้ยูนิตอื่นเรียกใช้ โดยที่ยูนิตนี้มีฟังก์ชันและโพรซีเยอร์ต่างๆดังนี้

1.) function check_wav_file

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะเข้ารหัสว่าเป็นไฟล์ .wav หรือไม่ ซึ่งสามารถตรวจสอบได้จากส่วนเฮดเดอร์(header) ของไฟล์.wav ไบต์ที่ 8,9,10,11 จะเก็บอักขระ W,A,V,E ตามลำดับ

2.) function check_type_of_record

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะเข้ารหัสในส่วนของรูปแบบ ซึ่งโปรแกรมนี้จะทำการเข้ารหัสได้เฉพาะกับไฟล์ที่เป็น PCM A-law เท่านั้น ซึ่งจะสามารถทำการตรวจสอบได้โดยในกรณีที่เป็น PCM A-law จะได้ว่าไบต์ที่ 20 ในส่วนของเฮดเดอร์ของไฟล์.wav จะมีค่าเป็น 1

3.) function check_channel

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะเข้ารหัสในส่วนของการบันทึกเป็นแบบ mono channel หรือ stereo channel ซึ่งจะสามารถตรวจสอบได้จากไบต์ที่ 22 ในเฮดเดอร์ของไฟล์ .wav โดยกรณีบันทึกแบบ mono channel จะเก็บค่าเป็น '1' ส่วนกรณีบันทึกแบบ stereo channel จะเก็บค่าเป็น '2' ในกรณีของ mono channel การบันทึกค่าแซมปลิ้งแต่ละค่าจะบันทึกขนาดของสัญญาณทีละไบต์เรียงต่อๆกันไป แต่ในการบันทึกค่าแซมปลิ้งแบบ stereo channel ขนาดของแต่ละแซมปลิ้งจะบันทึกโดย channel ซ้ายบันทึกลงในไบต์คู่ channel ขวาบันทึกลงในไบต์คี่ ในโครงงานนี้จะเลือกทำการเข้ารหัสเฉพาะกรณีที่บันทึกแบบ mono channel เท่านั้น

*หมายเหตุ* ส่วนเฮดเดอร์ ของไฟล์ .wav ที่ใช้แสดงการบันทึกแบบ mono channel หรือ stereo channel จะได้แก่ไบต์ที่ 22 และ 23 ไบต์แรกของแต่ละกรณีจะมีค่าต่างกัน จึงใช้ไบต์เดียวในการตรวจสอบได้

4) function check_sampling_rate

เป็นฟังก์ชั่นที่ทำหน้าที่ตรวจสอบอัตราการแซมปลิ้งของไฟล์ที่ต้องการจะทำการเข้ารหัส จากการทดลองทำการบันทึกเสียงและตรวจสอบไบต์ที่ 24 ในส่วนเฮดเดอร์ของไฟล์ .wav ซึ่งจะเก็บค่า 64 , 17 , 34 , 68 ตามลำดับ ในโครงงานนี้จะเลือกทำการเข้ารหัสเฉพาะกรณีที่มีอัตราการแซมปลิ้ง 8 kHz

*หมายเหตุ* ส่วนเฮดเดอร์ของไฟล์ .wav ที่ใช้ในการเก็บอัตราการแซมปลิ้ง จะประกอบด้วยจำนวนไบต์ทั้งหมด 4 ไบต์ คือ ไบต์ที่ 24-27 แต่เนื่องจากไบต์แรกของแต่ละกรณีมีค่าแตกต่างกัน จึงใช้ไบต์แรกไบต์เดียวในการตรวจสอบได้

5) function check_byte_per_sample

เป็นฟังก์ชั่นที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะเข้ารหัส ในส่วนของจำนวนไบต์ที่เก็บต่อหนึ่งแซมเปิ้ล โดยตรวจสอบในไบต์ที่ 32 ในส่วนของเฮดเดอร์ของไฟล์ .wav ต้องมีค่าเป็น 1

6) function check_bit_per_sample

เป็นฟังก์ชั่นที่ทำหน้าที่ตรวจสอบจำนวนบิทต่อหนึ่งแซมเปิ้ล ซึ่งมีอยู่ 2 กรณี คือ บันทึกแบบ 8 บิทต่อแซมเปิ้ล และ 16 บิทต่อแซมเปิ้ล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนบิทต่อแซมเปิ้ลจะเก็บไว้ในส่วนเฮดเดอร์ของไฟล์ .wav ที่ตำแหน่งไบต์ที่ 34-38 จากการลองทำการบันทึกและอ่านค่าที่ได้เปรียบเทียบกัน จะสามารถใช้ไบต์ที่ 34 ในการตรวจสอบจำนวนบิทต่อแซมเปิ้ลได้โดย ถ้าเป็นการบันทึกแบบ8 บิทต่อแซมเปิ้ล ไบต์ที่ 34 จะเก็บค่าเป็น 8 ส่วนกรณีที่เก็บค่าเป็น 16 บิทต่อแซมเปิ้ล ไบต์ที่ 34 จะเก็บค่าเป็น 16 ซึ่งในโครงงานนี้จะให้เป็นการบันทึกแบบ 8 บิทต่อแซมเปิ้ล

7) procedure copy_all_to_DOC

จะเป็นส่วนที่ทำการแสดงค่ารายละเอียดในไฟล์เสียงที่เป็นอินพุทเข้ามาโดยจะเก็บไว้ในชื่อของ EncodeIn.doc ซึ่งจะนำไปใช้ในการเปรียบเทียบกับไฟล์เสียงที่เป็นเอาท์พุทที่ได้จากภาคการถอดรหัส เพื่อดูว่าการทำงานของโปรแกรมตามที่ออกแบบไว้มีความผิดพลาดมากน้อยเพียงใด

8) procedure open_output_wav_code_file

จะเป็นส่วนที่ทำหน้าที่เก็บไฟล์ .wav :ซึ่งเป็นไฟล์ที่เก็บค่ารหัส ADPCM 4 บิท ซึ่งเป็นเอาท์พุทของภาคเข้ารหัส โดยเก็บไว้ในชื่อ Encode.wav

9) procedure write_header_to_code_file

เป็นส่วนที่ทำหน้าที่ในการกำหนดเฮดเดอร์ให้กับ Encode.wav เพื่อให้เป็นไฟล์เสียงแบบ ADPCM ซึ่งจะทำการกำหนดใหม่ในส่วนของ 60 ไบต์แรก

10) procedure adpcm_algorithm

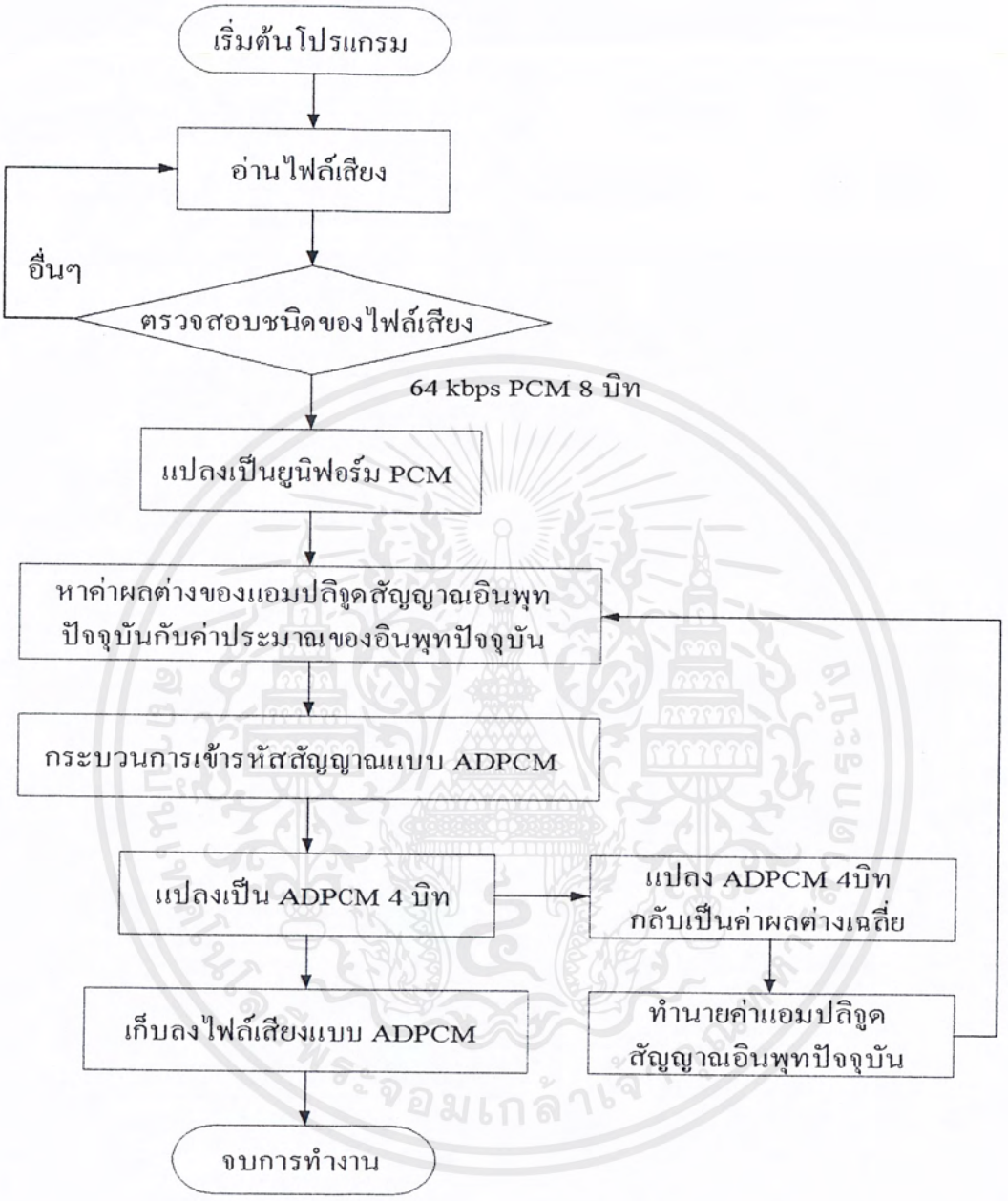
เป็นส่วนที่ทำหน้าที่ตามหลักการของการเข้ารหัสแบบ ADPCM ที่ได้แสดงไว้ข้างต้น เพื่อให้ได้รหัส ADPCM 4 บิท แล้วเก็บเอาท์พุทที่ได้ลงใน Encode.wav

11) procedure open_output_DOC_code_file

เป็นส่วนที่ทำการเขียนแสดงรายละเอียดของไฟล์ .Encode.wav ลงในไฟล์ EncodeOut.doc เพื่อแสดงรายละเอียด และนำมาเปรียบเทียบค่าความผิดพลาดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในยูนิตนี้มีแผนผังการทำงานดังนี้




รูปที่ 3.8 แผนผังแสดงการเข้ารหัสแบบเอดีพีซีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- unit NetVoicedecodeLib**

ยูนิตนี้ไม่มีฟอร์มแต่เป็นไลบรารี(library) ที่ให้ยูนิตอื่นเรียกใช้ โดยที่ยูนิตนี้มีฟังก์ชันและโพรซีเยอร์ต่างๆดังนี้

1.) function check_wav_file

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะถอดรหัสว่าเป็นไฟล์ .wav หรือไม่ ซึ่งสามารถตรวจสอบได้จากส่วนเฮดเดอร์(header) ของไฟล์.wav ไบต์ที่ 8,9,10,11 จะเก็บอักขระ W,A,V,E ตามลำดับ

2.) function check_type_of_record

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะถอดรหัสในส่วนของรูปแบบ ซึ่งโปรแกรมนี้จะทำการถอดรหัสได้เฉพาะกับไฟล์ที่เป็น PCM A-law เท่านั้น ซึ่งจะสามารถทำการตรวจสอบได้โดยในกรณีที่เป็น PCM A-law จะได้ว่าไบต์ที่ 20 ในส่วนของเฮดเดอร์ของไฟล์.wav จะมีค่าเป็น 1

3.) function check_channel

เป็นฟังก์ชันที่ทำหน้าที่ตรวจสอบไฟล์ที่ต้องการจะถอดรหัสในส่วนของการบันทึกเป็นแบบ mono channel หรือ stereo channel ซึ่งจะสามารถตรวจสอบได้จากไบต์ที่ 22 ในเฮดเดอร์ของไฟล์ .wav โดยกรณีบันทึกแบบ mono channel จะเก็บค่าเป็น '1' ส่วนกรณีบันทึกแบบ stereo channel จะเก็บค่าเป็น '2'

ในกรณีของ mono channel การบันทึกค่าแซมปลิ้งแต่ละค่าจะบันทึกขนาดของสัญญาณทีละไบต์เรียงต่อๆกันไป แต่ในการบันทึกค่าแซมปลิ้งแบบ stereo channel ขนาดของแต่ละแซมปลิ้งจะบันทึกโดย channel ซ้ายบันทึกลงในไบต์คู่ channel ขวาบันทึกลงในไบต์คี่ ในโครงงานนี้จะเลือกทำการถอดรหัสเฉพาะกรณีที่บันทึกแบบ mono channel เท่านั้น

*หมายเหตุ* ส่วนเฮดเดอร์ ของไฟล์ .wav ที่ใช้แสดงการบันทึกแบบ mono channel หรือ stereo channel จะได้แก่ไบต์ที่ 22 และ 23 ไบต์แรกของแต่ละกรณีจะมีค่าต่างกัน จึงใช้ไบต์เดียวในการตรวจสอบได้

4) function check_sampling_rate

เป็นฟังก์ชั่นที่ทำหน้าที่ตรวจสอบอัตราการแซมปลิ้งของไฟล์ที่ต้องการจะทำการถอดรหัส จากการทดลองทำการบันทึกเสียงและตรวจสอบไบต์ที่ 24 ในส่วนเฮดเดอร์ของไฟล์ .wav ซึ่งจะเก็บค่า 64 , 17 , 34 , 68 ตามลำดับ ในโครงงานนี้จะเลือกทำการถอดรหัสเฉพาะกรณีที่มีอัตราการแซมปลิ้ง 8 kHz

*หมายเหตุ* ส่วนเฮดเดอร์ของไฟล์ .wav ที่ใช้ในการเก็บอัตราการแซมปลิ้ง จะประกอบด้วยจำนวนไบต์ทั้งหมด 4 ไบต์ คือ ไบต์ที่ 24-27 แต่เนื่องจากไบต์แรกของแต่ละกรณีมีค่าแตกต่างกัน จึงใช้ไบต์แรกไบต์เดียวในการตรวจสอบได้

5.) function check_bit_per_sample

เป็นฟังก์ชั่นที่ทำหน้าที่ตรวจสอบจำนวนบิทต่อหนึ่งแซมเปิ้ล ซึ่งมีอยู่ 2 กรณี คือ บันทึกแบบ 8 บิทต่อแซมเปิ้ล และ 16 บิทต่อแซมเปิ้ล

จำนวนบิทต่อแซมเปิ้ลจะเก็บไว้ในส่วนเฮดเดอร์ของไฟล์ .wav ที่ตำแหน่งไบต์ที่ 34-38 จากการลองทำการบันทึกและอ่านค่าที่ได้เปรียบเทียบกัน จะสามารถใช้ไบต์ที่ 34 ในการตรวจสอบจำนวนบิทต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แซมเปิ้ลได้โดย ถ้าเป็นการบันทึกแบบ8 บิทต่อแซมเปิ้ล ไบต์ที่ 34 จะเก็บค่าเป็น 8 ส่วนกรณีที่เก็บค่าเป็น 16 บิทต่อแซมเปิ้ล ไบต์ที่ 34 จะเก็บค่าเป็น 16 ซึ่งในโครงงานนี้จะให้เป็นการบันทึกแบบ 8 บิทต่อแซมเปิ้ล

6.) procedure copy_all_to_DOC

จะเป็นส่วนที่ทำการแสดงค่ารายละเอียดในไฟล์เสียงที่รับเป็นอินพุทเข้ามาจากด้านส่งโดยจะเก็บไว้ในชื่อของ DecodeIn.doc ซึ่งจะเป็นไฟล์เดียวกับ EncodeOut.doc เพียงแต่อยู่กันคนละที่คือด้านส่งและด้านรับ

7.) procedure open_output_wav_code_file

จะเป็นส่วนที่ทำหน้าที่เก็บไฟล์ .wav :ซึ่งเป็นไฟล์ที่ถอดรหัสมาจาก ADPCM 4 บิท มาเป็นไฟล์เสียง PCM 8 บิทโดยเก็บไว้ในชื่อ Decode.wav

8.) procedure write_header_to_code_file

เป็นส่วนที่ทำหน้าที่ในการกำหนดเฮดเดอร์ให้กับ Decode.wav เพื่อให้เป็นไฟล์เสียงแบบ PCM 8 บิทซึ่งจะทำการกำหนดใหม่ในส่วนของ 60 ไบต์แรก

10) procedure adpcm_algorithm

เป็นส่วนที่ทำหน้าที่เรียกใช้งานฟังก์ชันอื่น ในกระบวนการถอดรหัสแบบ ADPCM

11.) function map_to_output_level และ function map_to_level

เป็นฟังก์ชันที่ทำหน้าที่ในการแปลงค่าแอมปลิจูดที่ได้จากการถอดรหัส ให้กลับไปอยู่ในรูปของค่าระดับการควอนไตซ์แล้วเขียนลงในไฟล์ Decode.wav

12.) function step_adjust

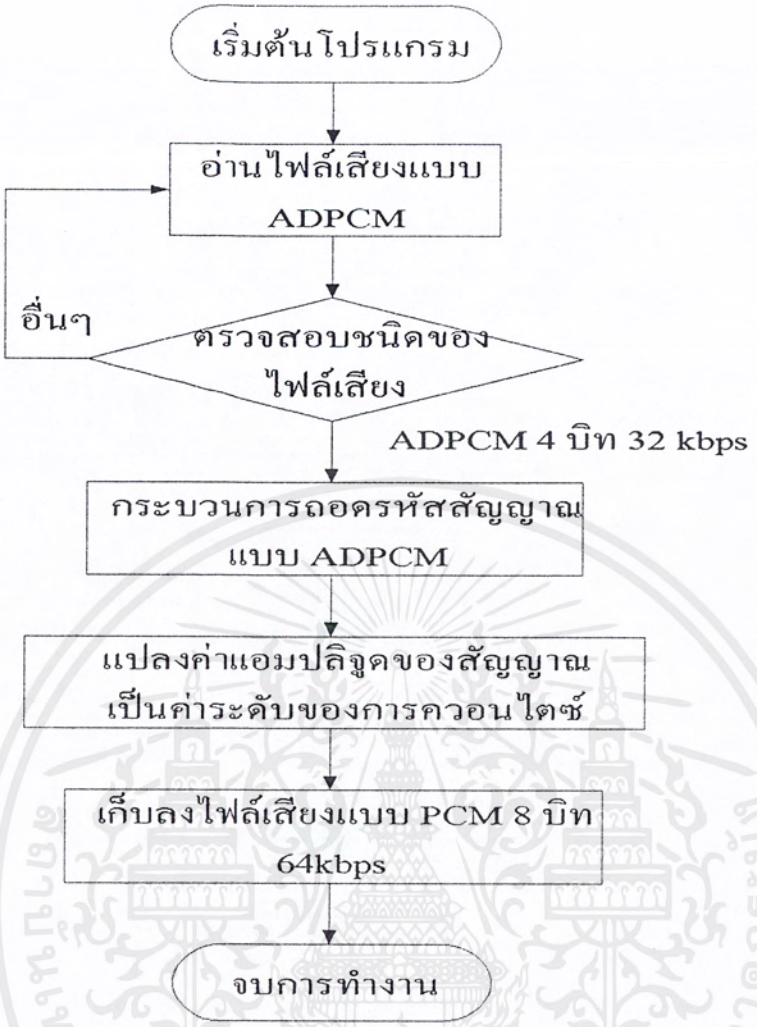
เป็นฟังก์ชันที่ใช้ในการปรับขั้นเพื่อใช้ในการหาค่าอินเด็กซ์และค่าสเต็บไซส์ตัวต่อไป ที่จะนำไปใช้ในการหาค่าแอมปลิจูดของสัญญาณตัวต่อไป

13) procedure open_output_DOC_code_file

เป็นส่วนที่ทำการเขียนแสดงรายละเอียดของไฟล์ .Decode.wav ลงในไฟล์ DecodeOut.doc เพื่อแสดงรายละเอียด และนำมาเปรียบเทียบค่าความผิดพลาดของโปรแกรม

ในยูนิตนี้มีแผนผังการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.9 แผนผังแสดงการถอดรหัสแบบเอดีพีซีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# การทดลองและผลการทดลอง

## 4.1 การทดลอง

4.1.1 การทดลองเพื่อหาค่าเริ่มต้นของโปรแกรมที่ดีที่สุด

ในการทดลองนี้ในขั้นแรกจะทำการทดลองเพื่อหาค่าเริ่มต้นที่เหมาะสม โดยค่าเริ่มต้นในที่นี้มี 2 ค่า คือ ค่าอินเด็กซ์และค่าประมาณของสัญญาณอินพุทตัวก่อน ซึ่งค่าอินเด็กซ์นั้นจะมีค่าตั้งแต่ 16 ถึง1552 ตามตารางที่ 3.2 และค่าประมาณของสัญญาณอินพุทตัวก่อนจะมีค่าตั้งแต่ -4096 ถึง 4096 จากค่าสองค่านี้ เราสามารถกำหนดค่าเริ่มต้นให้โปรแกรมได้ทั้งหมด 9 กรณี ดังนี้คือ

4.1.1.1 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 0 และค่าอินเด็กซ์ = 48

4.1.1.2 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 0 และค่าอินเด็กซ์ = 24

4.1.1.3 ค่าประมาณของสัญญาณอินพุทตัวก่อน = -4096 และค่าอินเด็กซ์ = 0

4.1.1.4 ค่าประมาณของสัญญาณอินพุทตัวก่อน = -4096 และค่าอินเด็กซ์ = 24

4.1.1.5 ค่าประมาณของสัญญาณอินพุทตัวก่อน = -4096 และค่าอินเด็กซ์ = 48

4.1.1.6 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 4096 และค่าอินเด็กซ์ = 0

4.1.1.7 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 4096 และค่าอินเด็กซ์ = 24

4.1.1.8 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 4096 และค่าอินเด็กซ์ = 48

4.1.1.9 ค่าประมาณของสัญญาณอินพุทตัวก่อน = 0 และค่าอินเด็กซ์ = 0

แล้วหาค่าเปอร์เซ็นต์ความผิดพลาดของแต่ละกรณีว่ากรณีไหนมีค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดเพื่อที่จะได้นำมาใช้ โดยใช้สัญญาณฟังก์ชันไซน์และความถี่ที่นำมาใช้ในการทดลองอยู่ในช่วง 200 ถึง 3800 เฮิรตซ์ (Heartz) โดยในการหากรณีที่เหมาะสมที่สุด จะใช้ความถี่ 3 ค่าคือ 200 , 2000 และ 3800 เฮิรตซ์

4.1.2 การหาค่าเฉลี่ยของความผิดพลาดที่เกิดขึ้นในการทำงานของโปรแกรม

เมื่อกำหนดให้

Lt = ระดับของการควอนไตซ์ของสัญญาณเสียงทางด้านส่ง

Lr = ระดับของการควอนไตซ์ของสัญญาณเสียงทางด้านรับ

N = จำนวนไบต์ของไฟล์เสียง

จะได้ว่า

$$\text{เปอร์เซ็นต์ความผิดพลาดเฉลี่ย} = [(\sum_{n=1}^{N} ( \left| Lt - Lr \right| / 256 )) * 100\%] / N \quad .....4.1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การนำเอากรณีที่ค่าเริ่มต้นของโปรแกรมให้ค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดมาเก็บผลการทดลอง

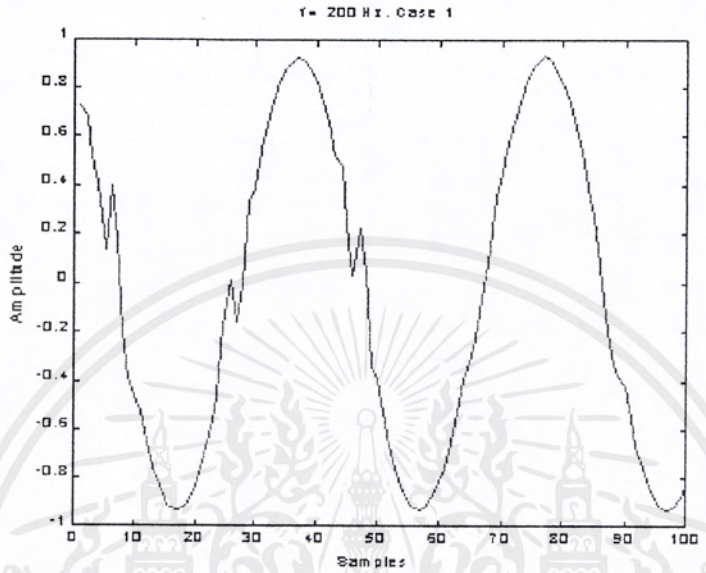
โดยจะนำค่าเริ่มต้นของโปรแกรมที่ให้ค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดมาเก็บผลการทดลอง โดยจะเก็บผลการทดลองเป็นรูปเวฟฟอร์มและกราฟการวิเคราะห์ทางความถี่ที่ความถี่ต่างๆ ดังนี้ 200 ,400 , 600 , 800 , 1000 , 1200 , 1400 , 1600 , 1800 , 2000 , 2200 , 2400 , 2600 , 2800 , 3000 , 3200 , 3400 , 3600 และ 3800 เฮิรตซ์ และได้มีการหาค่าเปอร์เซ็นต์ความผิดพลาด , ค่าMinimum RMS Power , ค่า Maximum RMS Power และค่า Average RMS Power ที่ความถี่ต่างๆอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

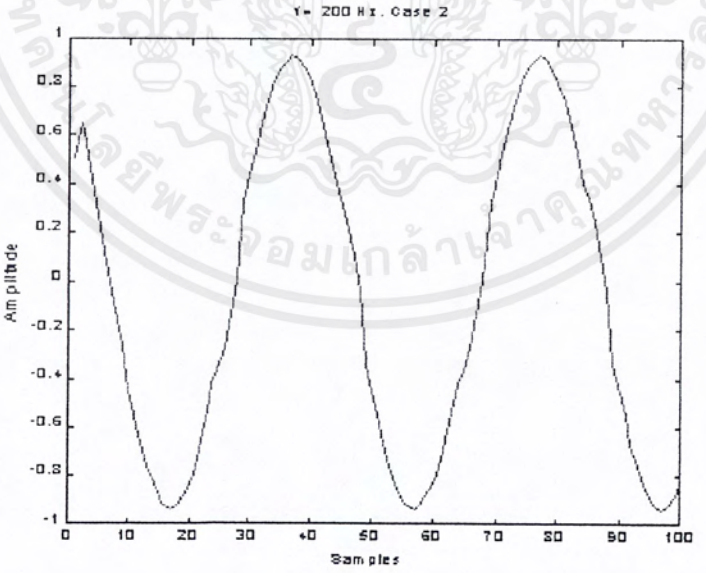
## 4.2 ผลการทดลอง

### 4.2.1 ผลการทดลองของ 4.1.1 การทดลองหาค่าเริ่มต้นที่ดีที่สุด

#### 4.2.1.1 แสดงรูปเวฟฟอร์มที่ความถี่ 200 เฮิรตซ์ ของกรณีต่างๆ

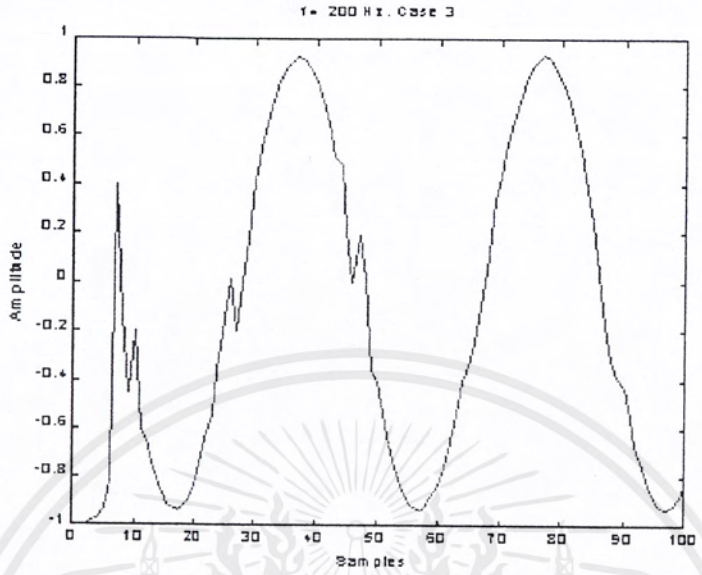



รูปที่ 4.1 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1

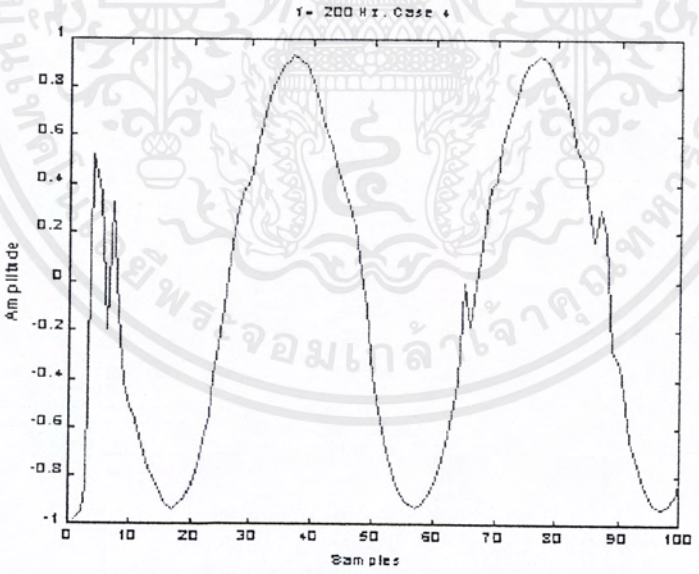



รูปที่ 4.2 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

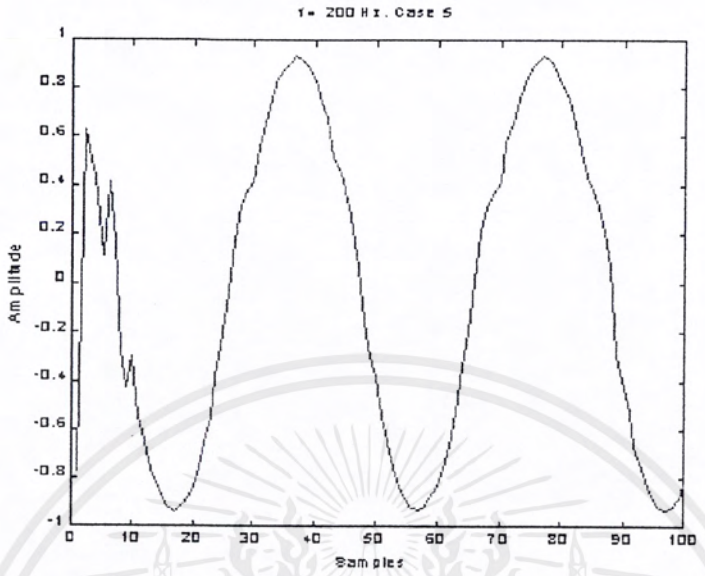



รูปที่ 4.3 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3

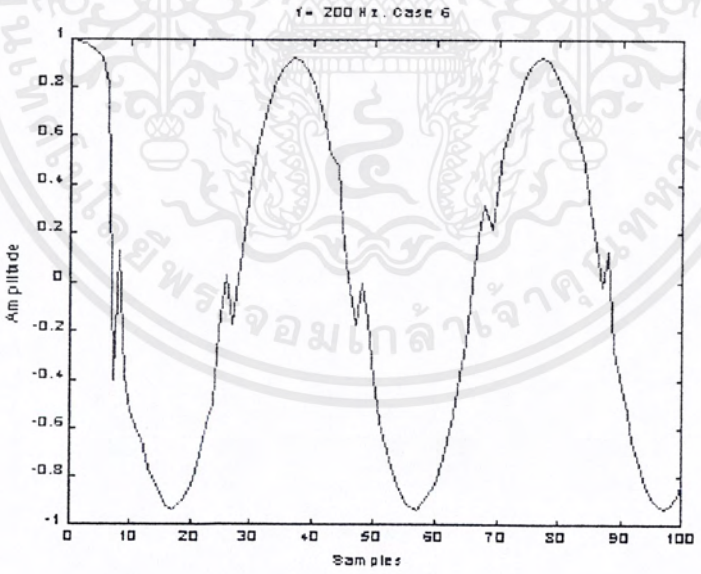



รูปที่ 4.4 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

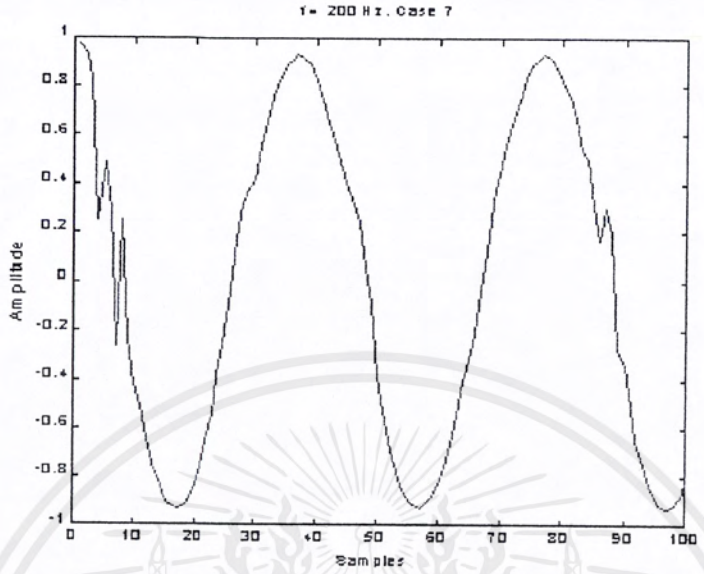



รูปที่ 4.5 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5

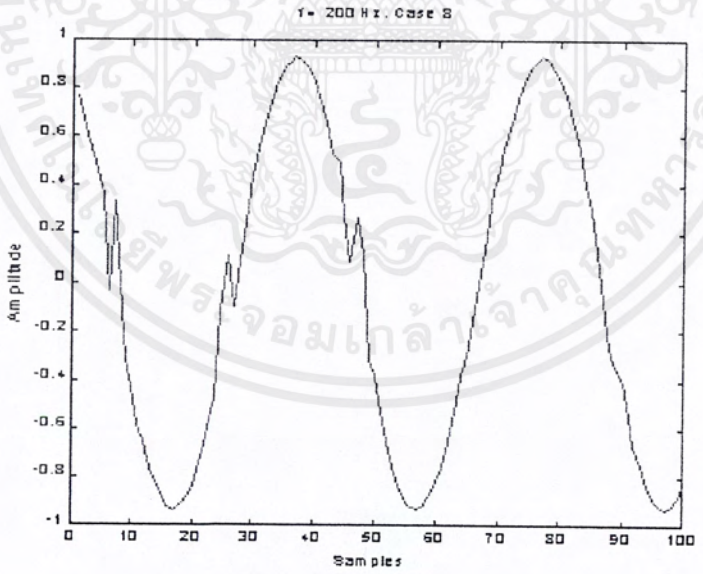



รูปที่ 4.6 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

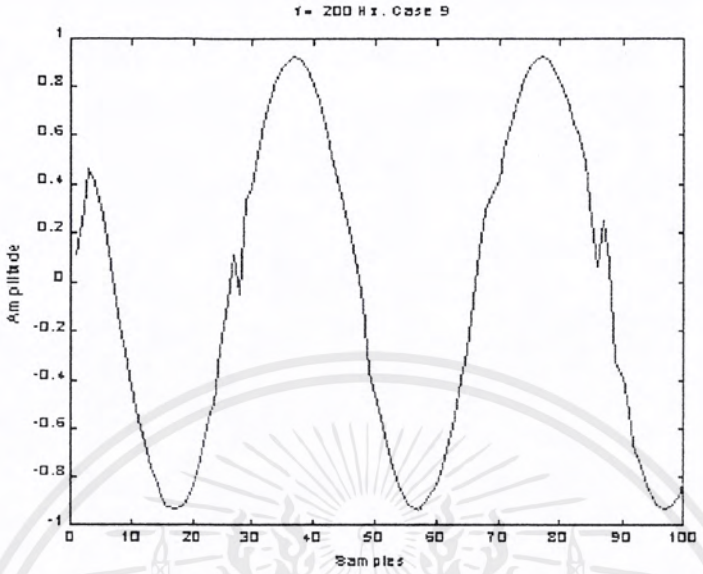



รูปที่ 4.7 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7




รูปที่ 4.8 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



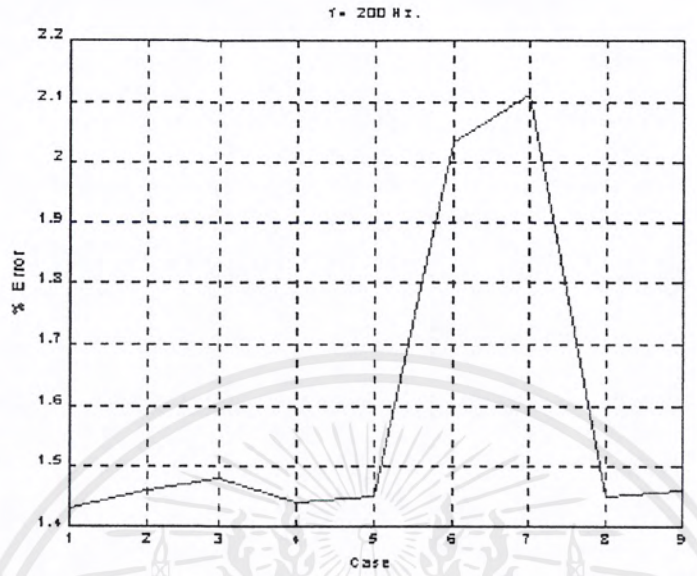

รูปที่ 4.9 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 200 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9

4.2..1.2 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 200 เฮิรตซ์

| กรณีที่ | ขนาดข้อมูลของไฟล์ที่ด้านส่ง (ไบต์) | ขนาดข้อมูลของไฟล์เมื่อบีบอัดแล้ว (ไบต์) | ขนาดข้อมูลของไฟล์ที่ด้านรับ (ไบต์) | เปอร์เซ็นต์ความผิดพลาด |
|---|---|---|---|---|
| 1 | 40,000 | 19,968 | 39,986 | 1.43 |
| 2 | 40,000 | 19,968 | 39,986 | 1.46 |
| 3 | 40,000 | 19,968 | 39,986 | 1.48 |
| 4 | 40,000 | 19,968 | 39,986 | 1.44 |
| 5 | 40,000 | 19,968 | 39,986 | 1.45 |
| 6 | 40,000 | 19,968 | 39,986 | 2.03 |
| 7 | 40,000 | 19,968 | 39,986 | 2.11 |
| 8 | 40,000 | 19,968 | 39,986 | 1.45 |
| 9 | 40,000 | 19,968 | 39,986 | 1.46 |

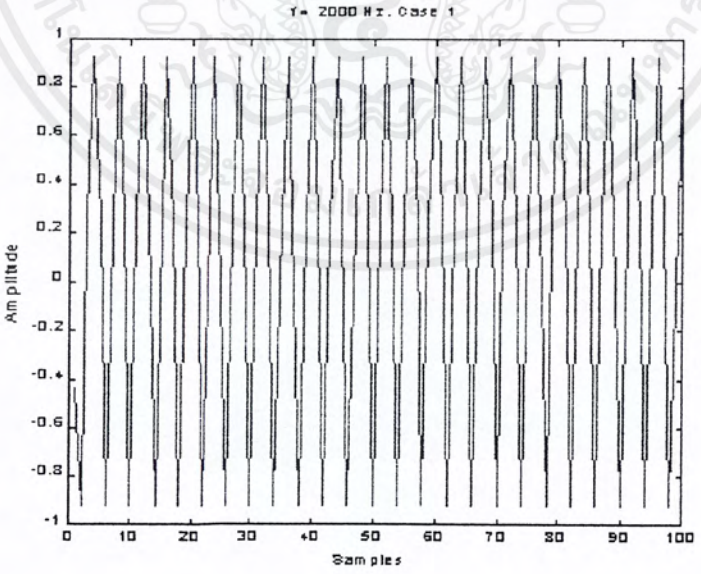
ตารางที่ 4.1 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



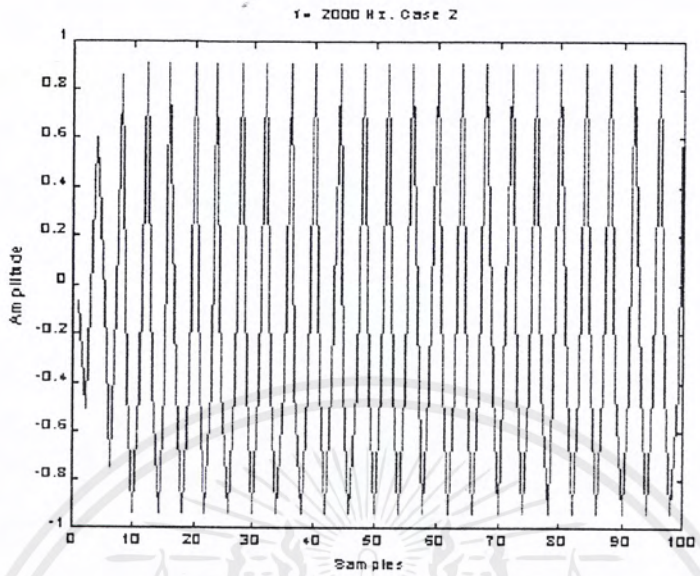

รูปที่ 4.10 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆ ที่ความถี่ 200 เฮิรตซ์

4.2.1.3 แสดงรูปเวฟฟอร์มที่ความถี่ 2000 เฮิรตซ์ ของกรณีต่างๆ

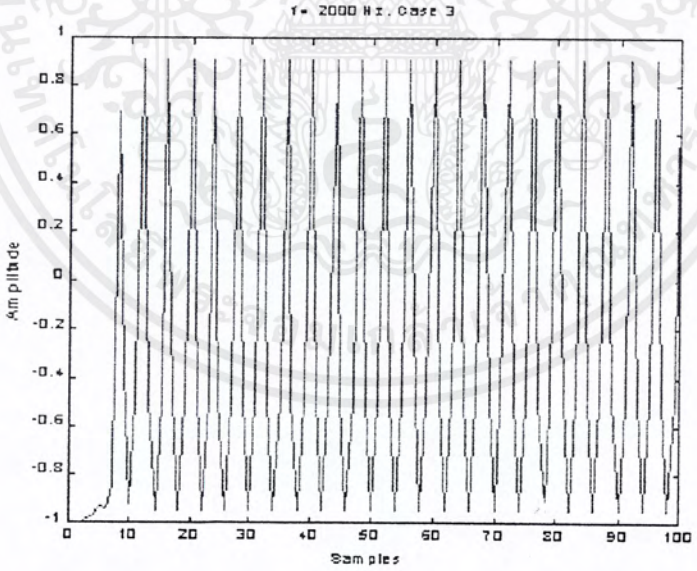



**รูปที่ 4.11 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

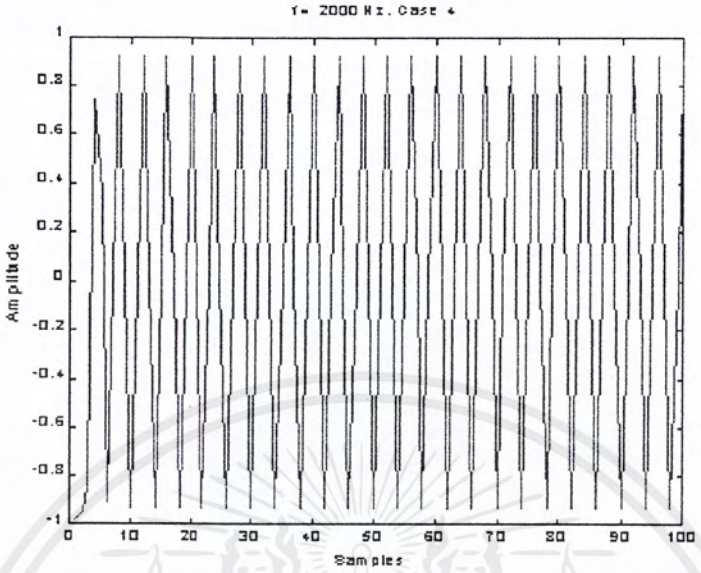



รูปที่ 4.12 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2

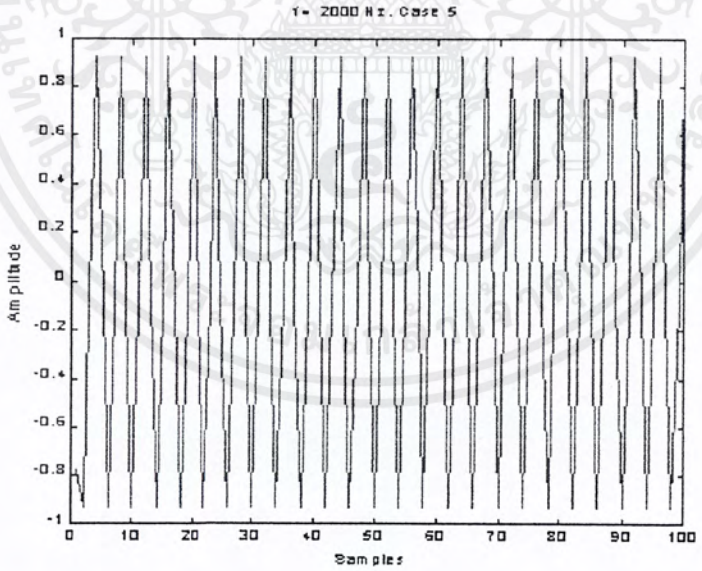



รูปที่ 4.13 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

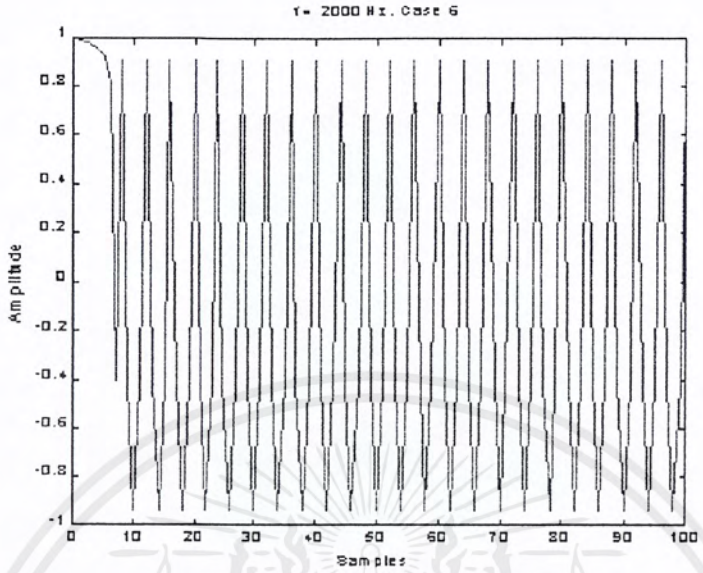



รูปที่ 4.14 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4

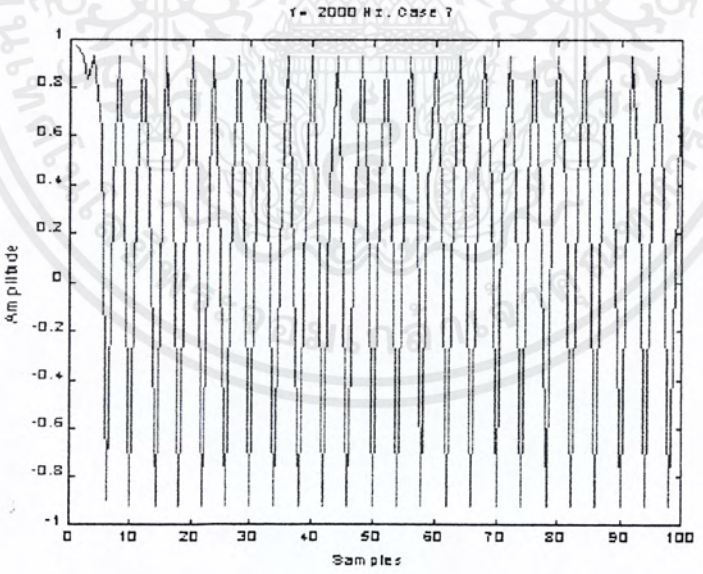



รูปที่ 4.15 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

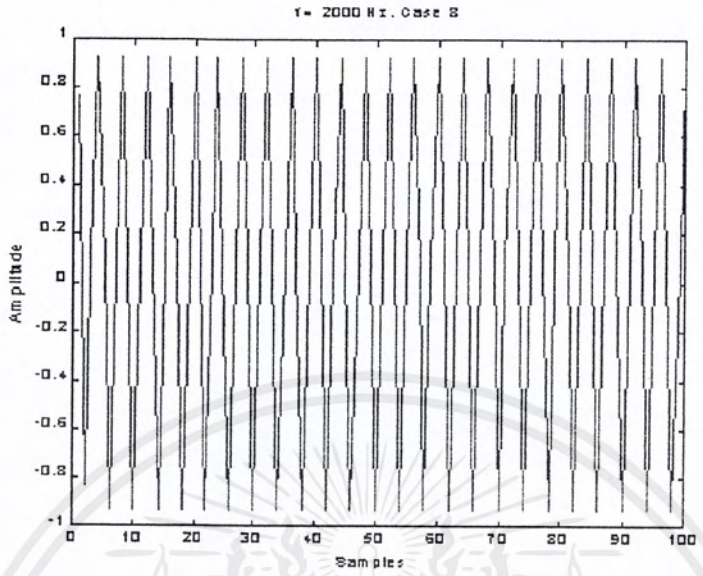



รูปที่ 4.16 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6

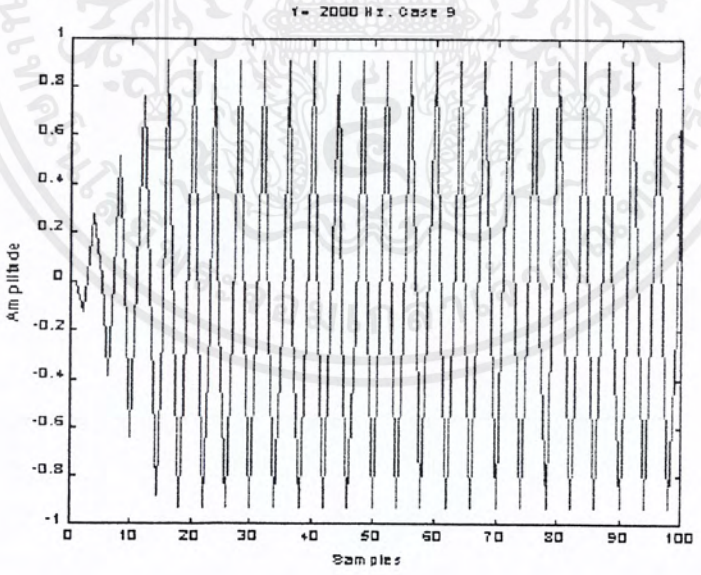



รูปที่ 4.17 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.18 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8



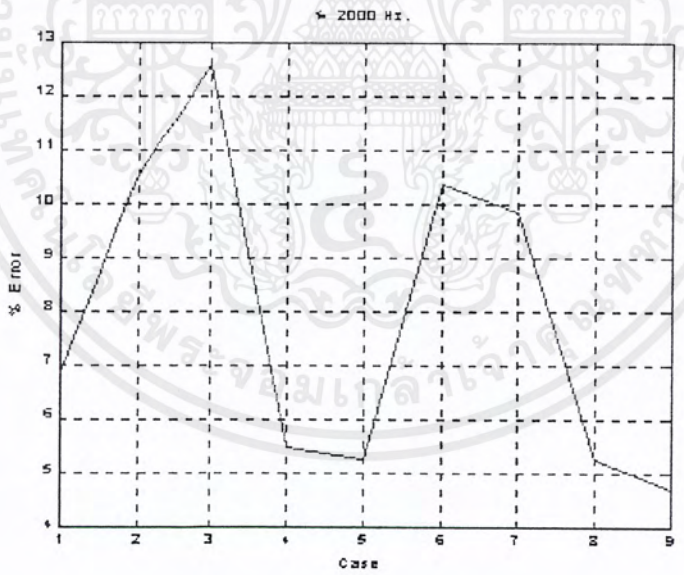

รูปที่ 4.19 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.4 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 2000 เฮิรตซ์

| กรณีที่ | ขนาดข้อมูลของไฟล์ที่ด้านส่ง (ไบต์) | ขนาดข้อมูลของไฟล์เมื่อบีบอัดแล้ว (ไบต์) | ขนาดข้อมูลของไฟล์ที่ด้านรับ (ไบต์) | เปอร์เซ็นต์ความผิดพลาด |
|---|---|---|---|---|
| 1 | 40,000 | 19,968 | 39,986 | 6.85 |
| 2 | 40,000 | 19,968 | 39,986 | 10.55 |
| 3 | 40,000 | 19,968 | 39,986 | 12.61 |
| 4 | 40,000 | 19,968 | 39,986 | 5.51 |
| 5 | 40,000 | 19,968 | 39,986 | 5.26 |
| 6 | 40,000 | 19,968 | 39,986 | 10.36 |
| 7 | 40,000 | 19,968 | 39,986 | 9.87 |
| 8 | 40,000 | 19,968 | 39,986 | 5.26 |
| 9 | 40,000 | 19,968 | 39,986 | 4.70 |

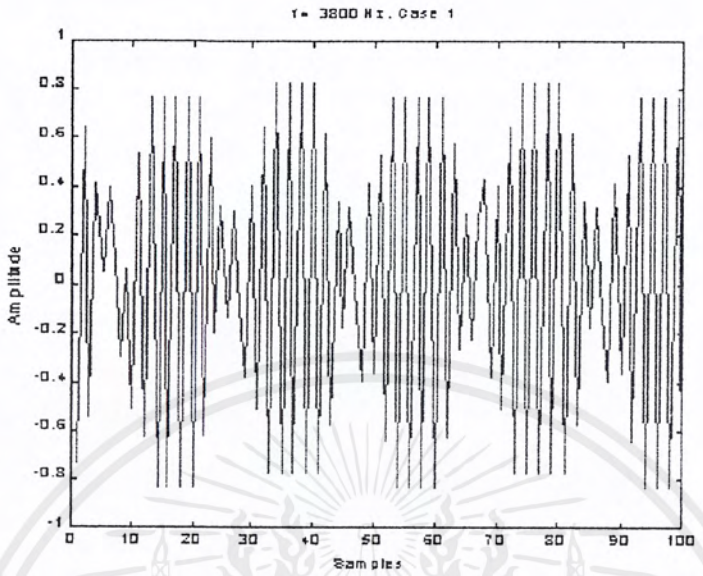
ตารางที่ 4.2 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 2000 เฮิรตซ์



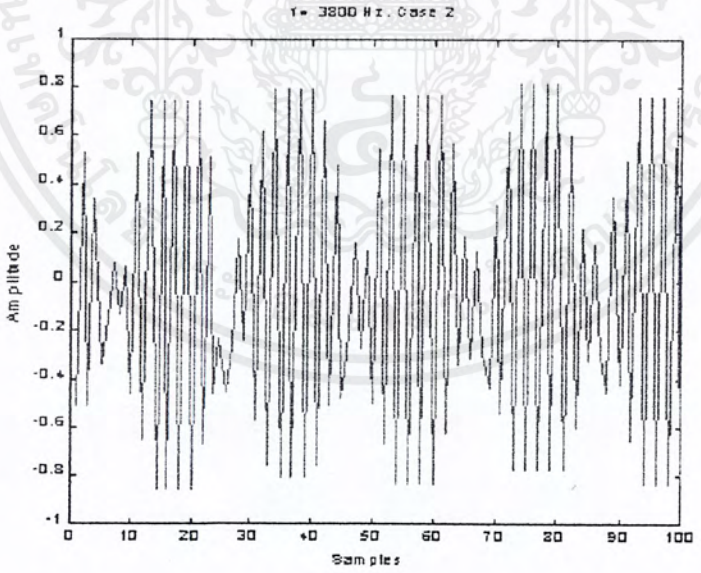

รูปที่ 4.20 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 2000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.5 แสดงรูปเวฟฟอร์มที่ความถี่ 3800 เฮิรตซ์ ของกรณีต่างๆ

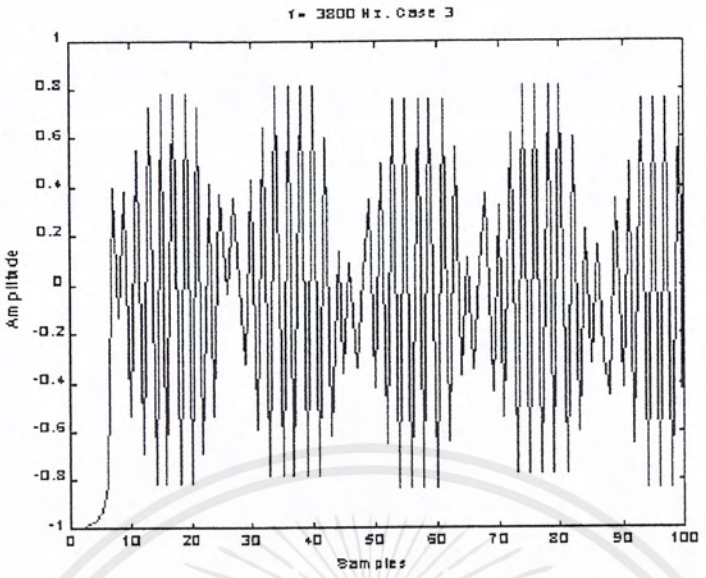



รูปที่ 4.21 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 1

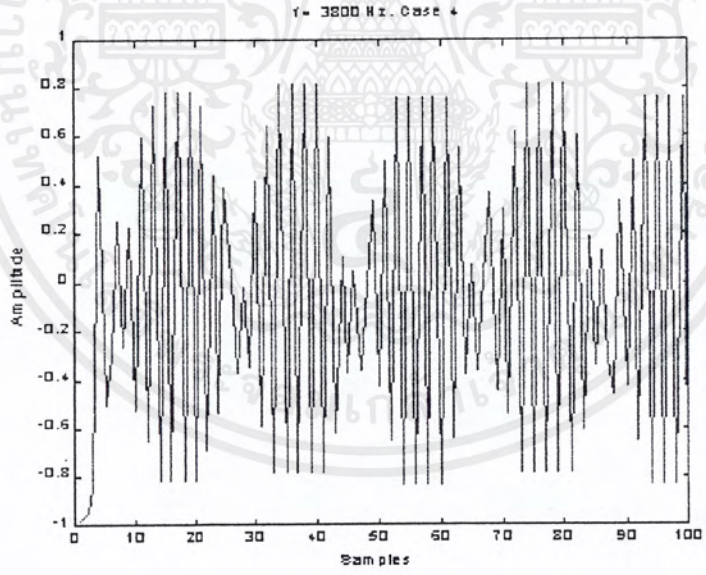



รูปที่ 4.22 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

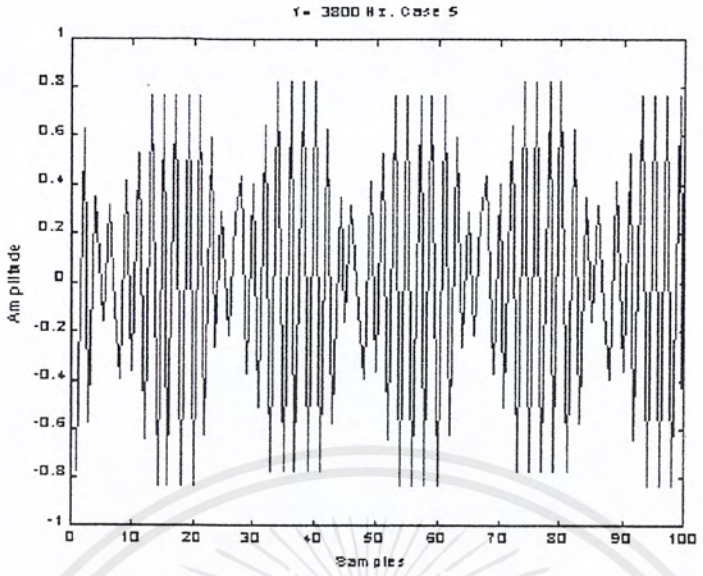



รูปที่ 4.23 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 3

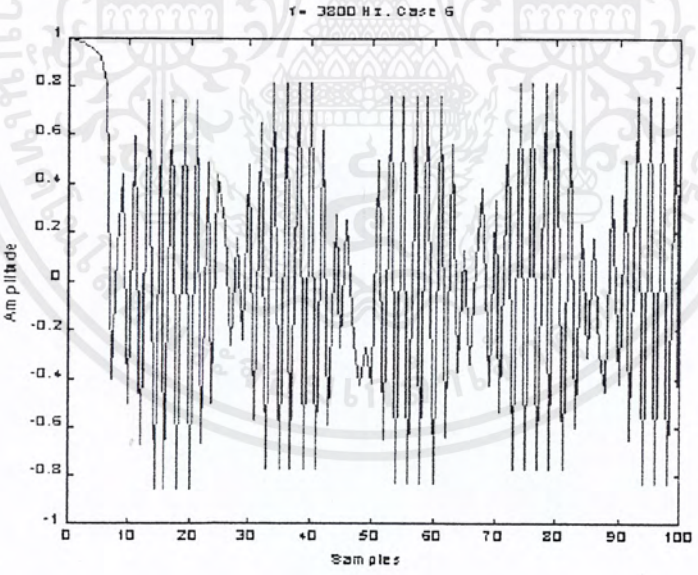



รูปที่ 4.24 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

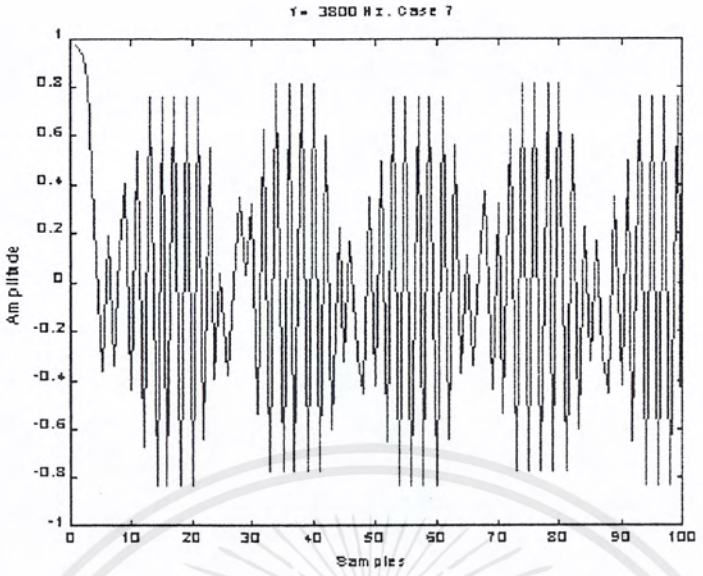



รูปที่ 4.25 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 5

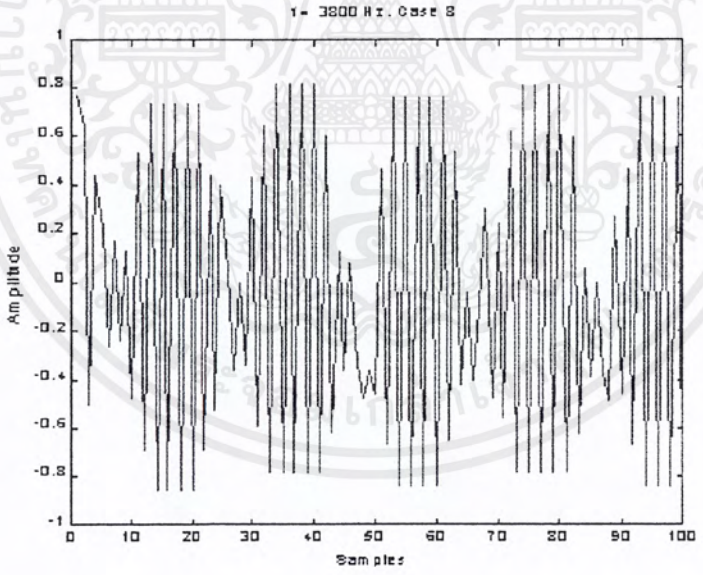



รูปที่ 4.26 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

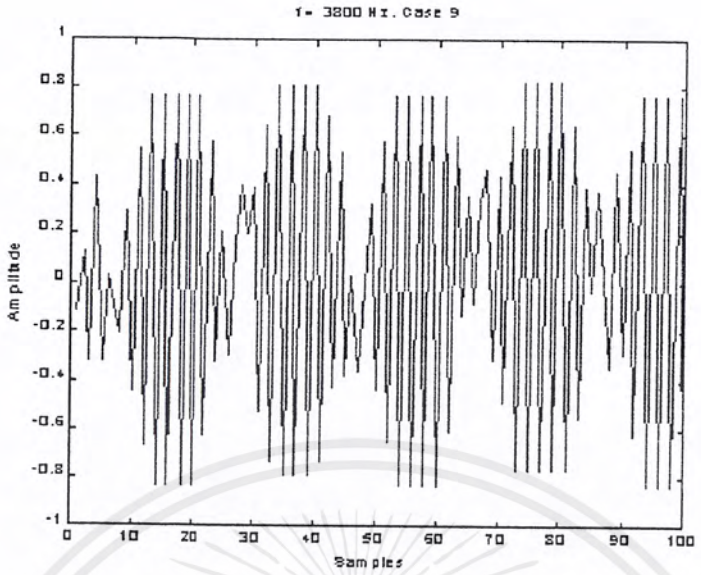



รูปที่ 4.27 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 7




รูปที่ 4.28 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



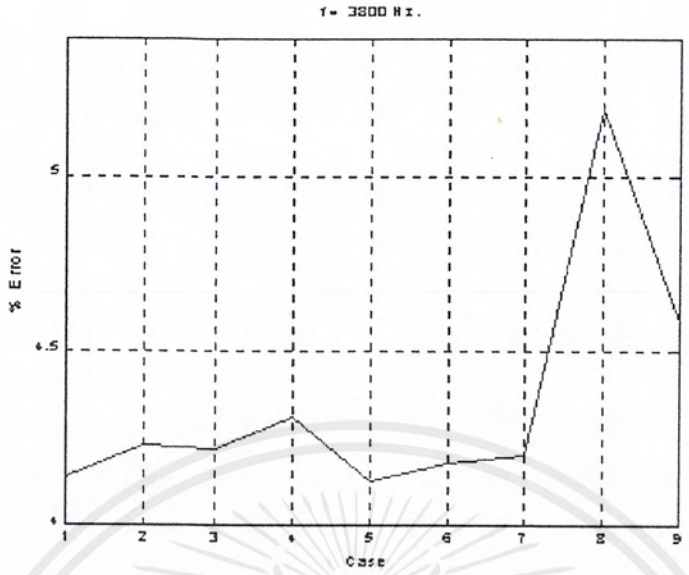

รูปที่ 4.29 แสดงเวฟฟอร์มของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์ เมื่อใช้ค่าเริ่มต้นในกรณีที่ 9

4.2.1.6 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 3800 เฮิรตซ์

| กรณีที่ | ขนาดข้อมูลของไฟล์ที่ด้านส่ง (ไบต์) | ขนาดข้อมูลของไฟล์เมื่อบีบอัดแล้ว (ไบต์) | ขนาดข้อมูลของไฟล์ที่ด้านรับ (ไบต์) | เปอร์เซ็นต์ความผิดพลาด |
|---|---|---|---|---|
| 1 | 40,000 | 19,968 | 39,986 | 4.14 |
| 2 | 40,000 | 19,968 | 39,986 | 4.23 |
| 3 | 40,000 | 19,968 | 39,986 | 4.22 |
| 4 | 40,000 | 19,968 | 39,986 | 4.31 |
| 5 | 40,000 | 19,968 | 39,986 | 4.13 |
| 6 | 40,000 | 19,968 | 39,986 | 4.18 |
| 7 | 40,000 | 19,968 | 39,986 | 4.20 |
| 8 | 40,000 | 19,968 | 39,986 | 5.20 |
| 9 | 40,000 | 19,968 | 39,986 | 4.59 |

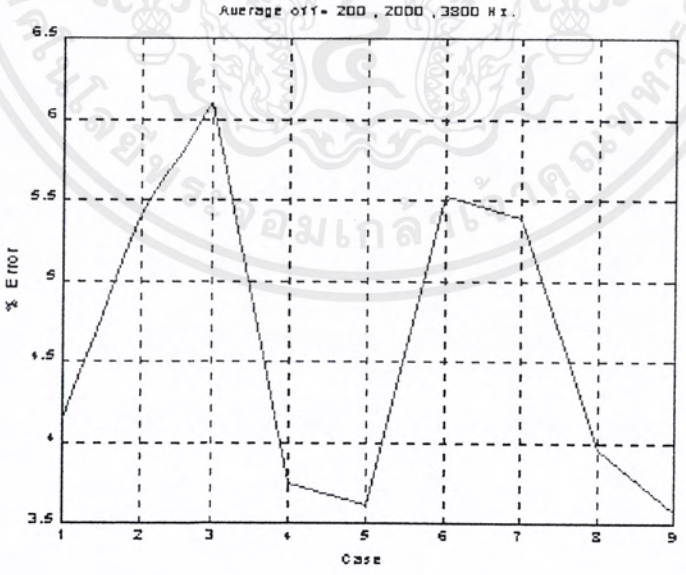
ตารางที่ 4.3 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.30 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆที่ความถี่ 3800 เฮิรตซ์

และเมื่อนำค่าเปอร์เซ็นต์ความผิดพลาดของแต่ละกรณีและแต่ละความถี่มาหาค่าเฉลี่ยจะได้กราฟดังนี้



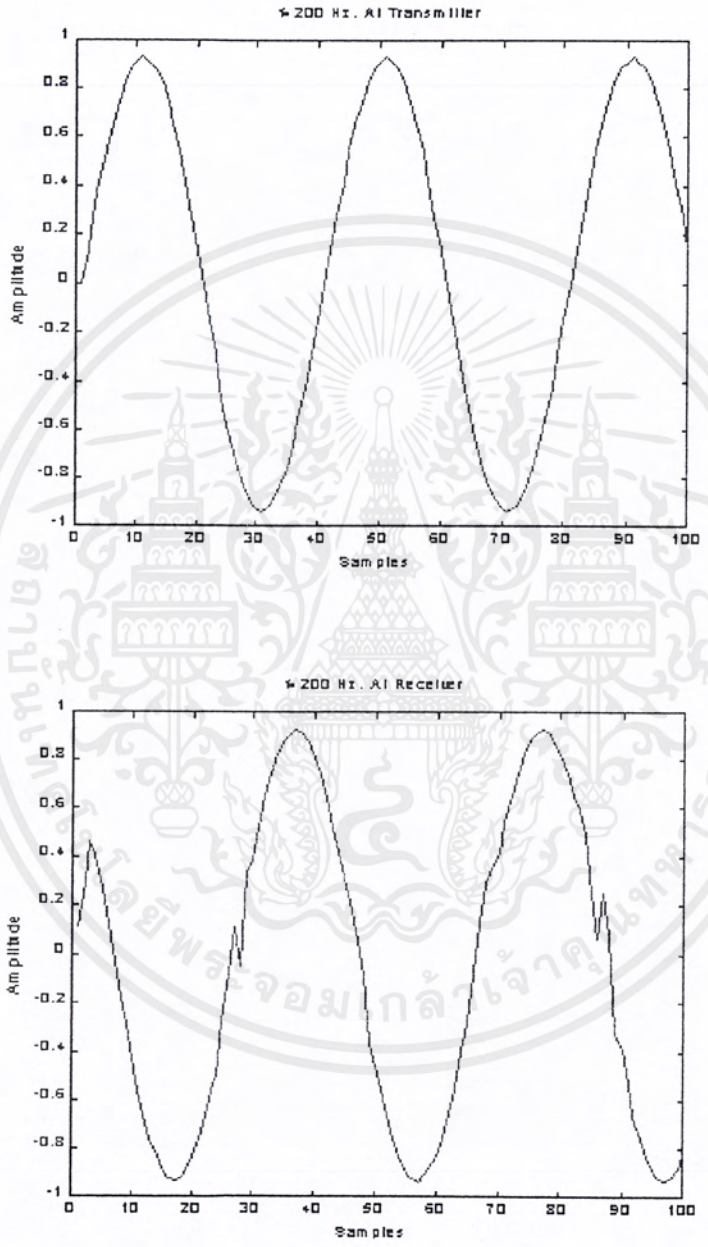

รูปที่ 4.31 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของกรณีต่างๆเฉลี่ยของความถี่ทั้งสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองของ 4.1.3 การนำเอากรณีที่ค่าเริ่มต้นของโปรแกรมให้ค่าเปอร์เซ็นต์ความผิดพลาดน้อยที่สุดมาใช้

4.2.2.1 แสดงในรูปแบบของเวฟฟอร์มที่ความถี่ต่างๆโดยแสดงตัวอย่าง 100 แซมเปิ้ล

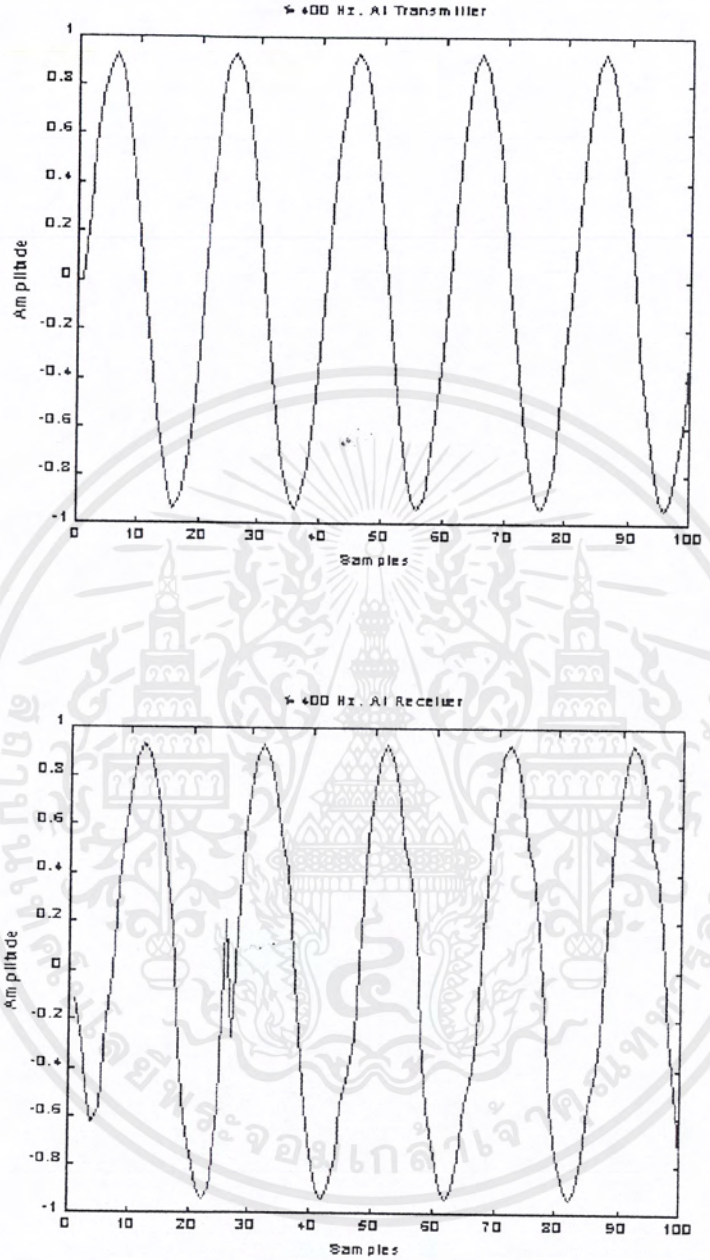
- ความถี่ 200 เฮิรตซ์




รูปที่ 4.32 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

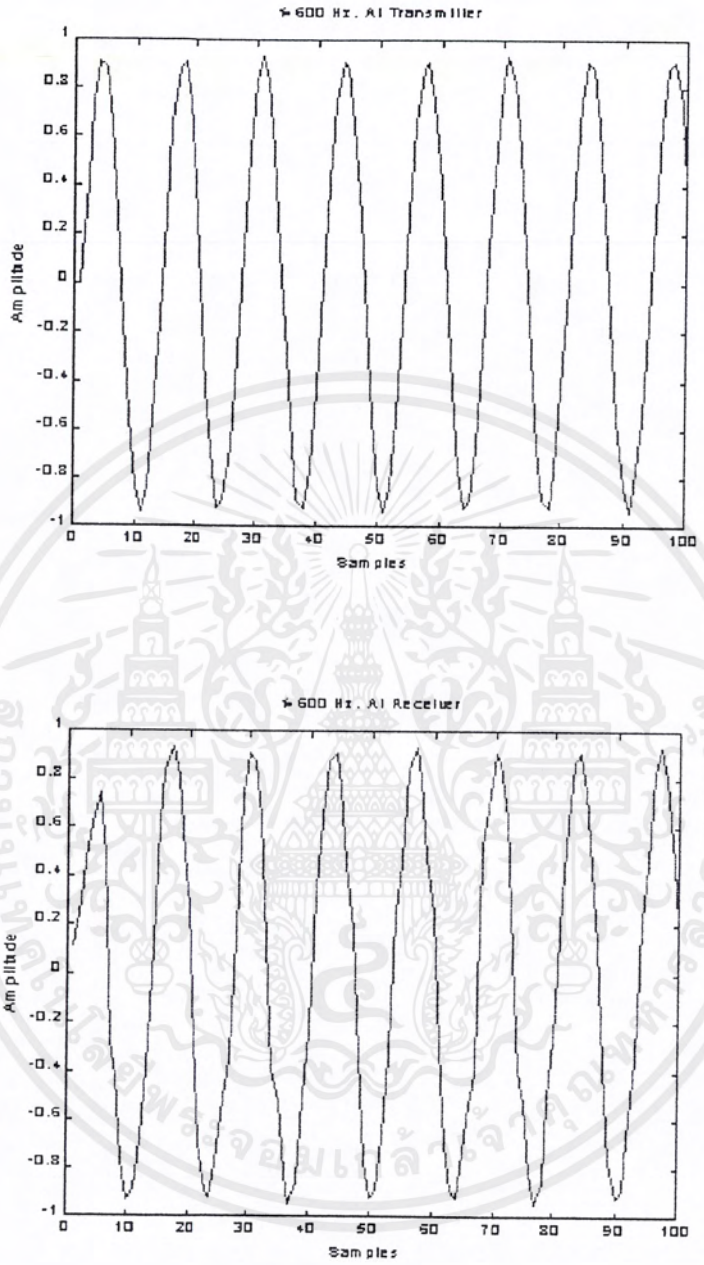
- ความถี่ 400 เฮิรตซ์




รูปที่ 4.33 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

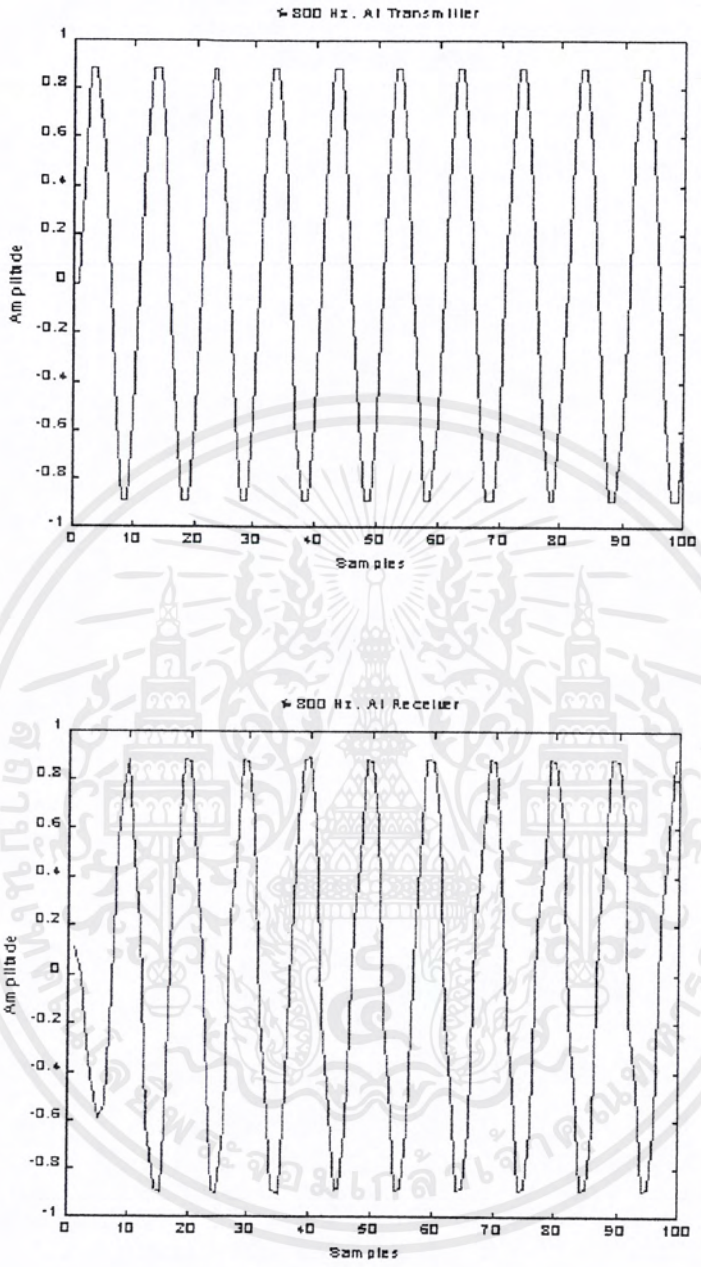
- ความถี่ 600 เฮิรตซ์




รูปที่ 4.34 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

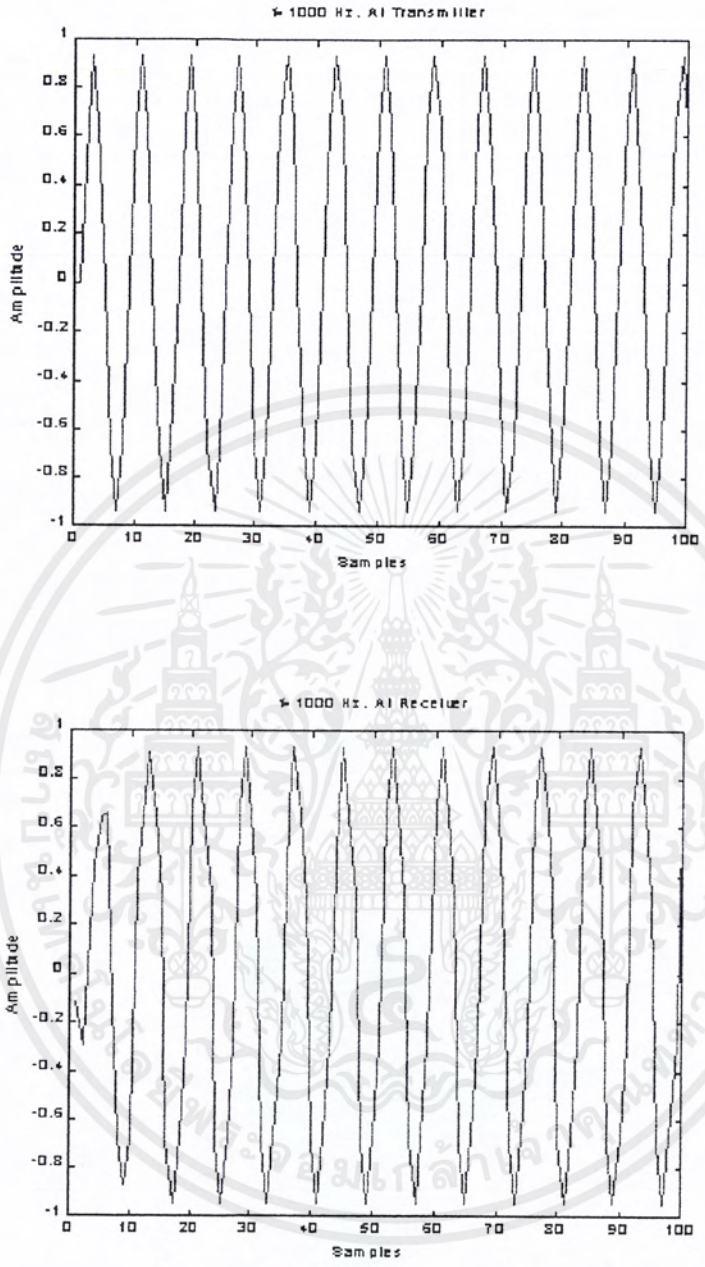
- ความถี่ 800 เฮิรตซ์




รูปที่ 4.35 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

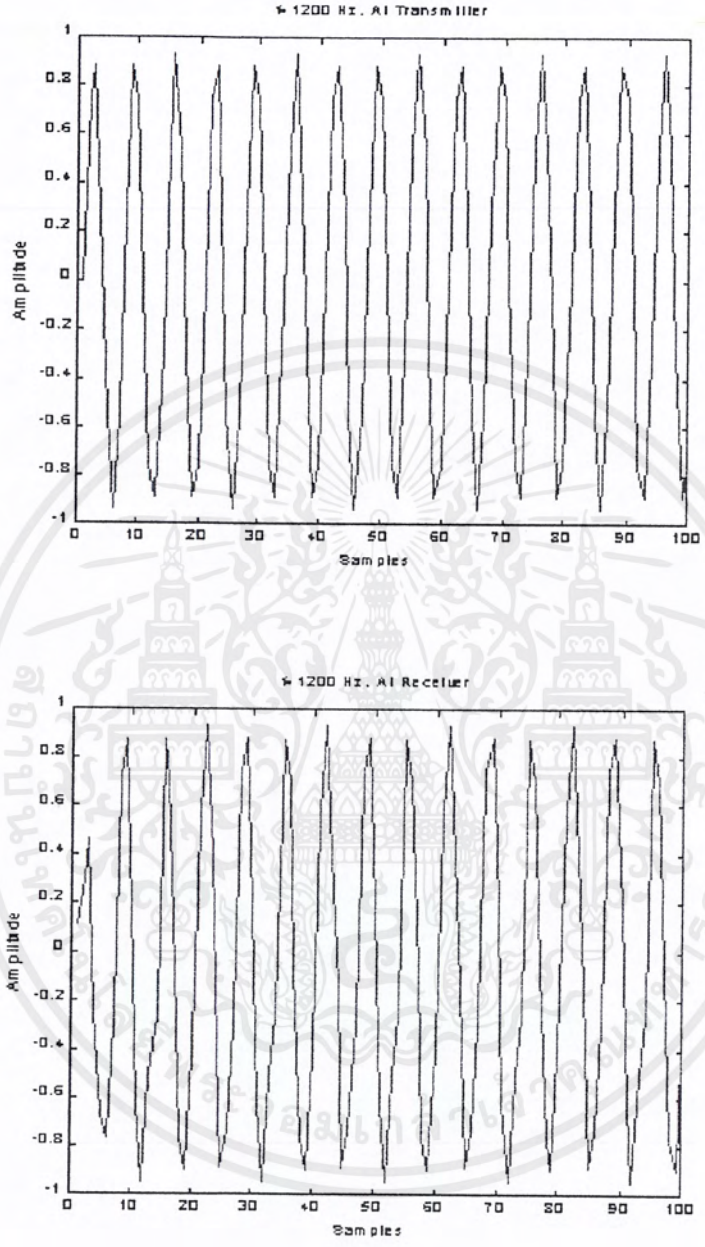
- ความถี่ 1000 เฮิรตซ์




รูปที่ 4.36 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

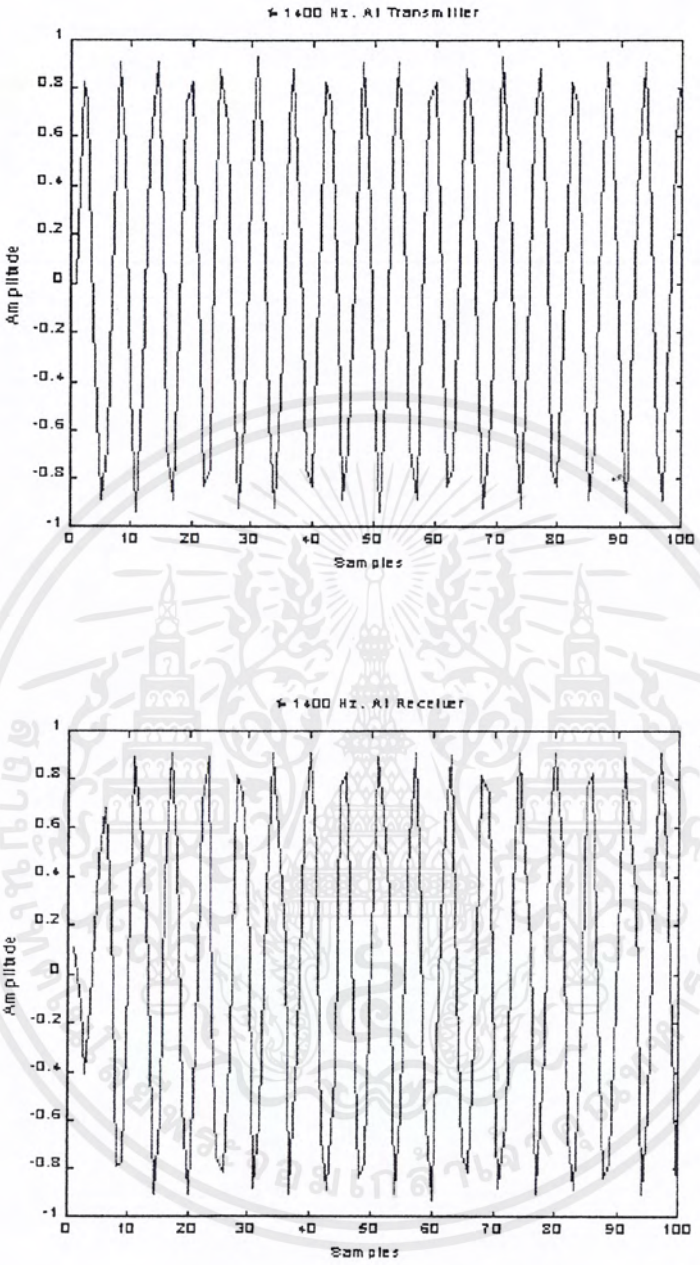
- ความถี่ 1200 เฮิรตซ์




รูปที่ 4.37 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

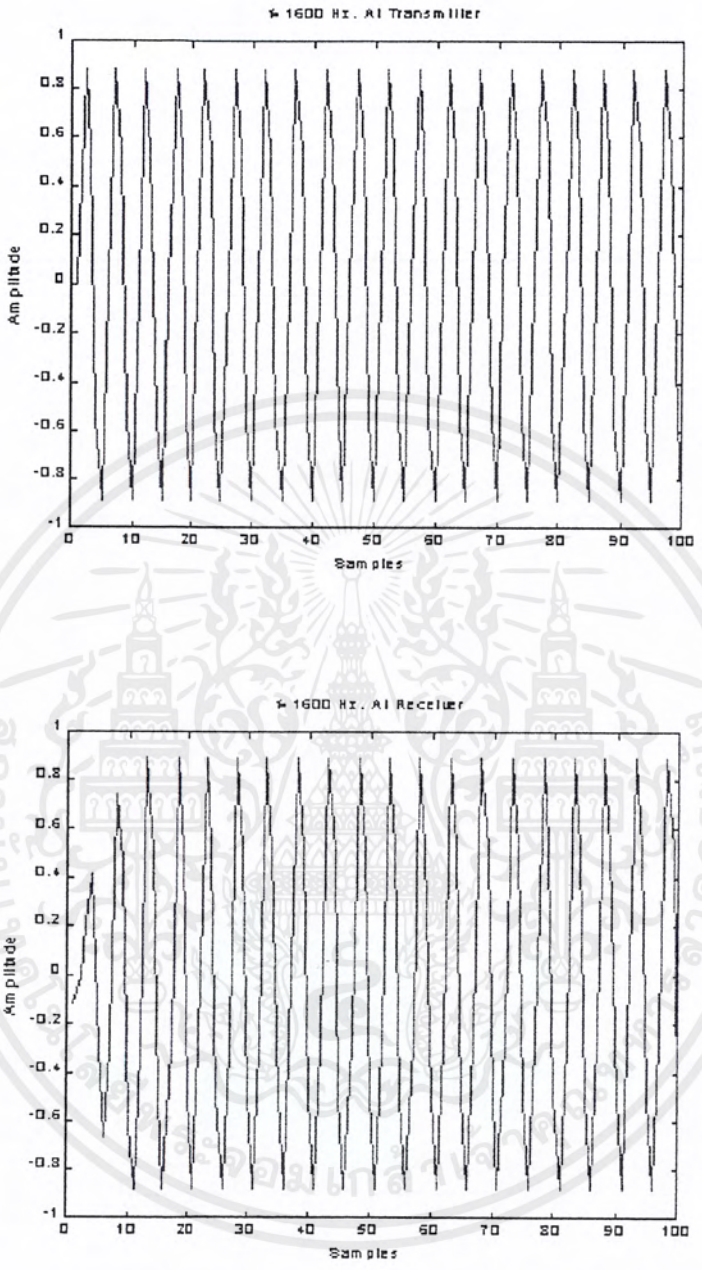
- ความถี่ 1400 เฮิรตซ์




รูปที่ 4.38 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

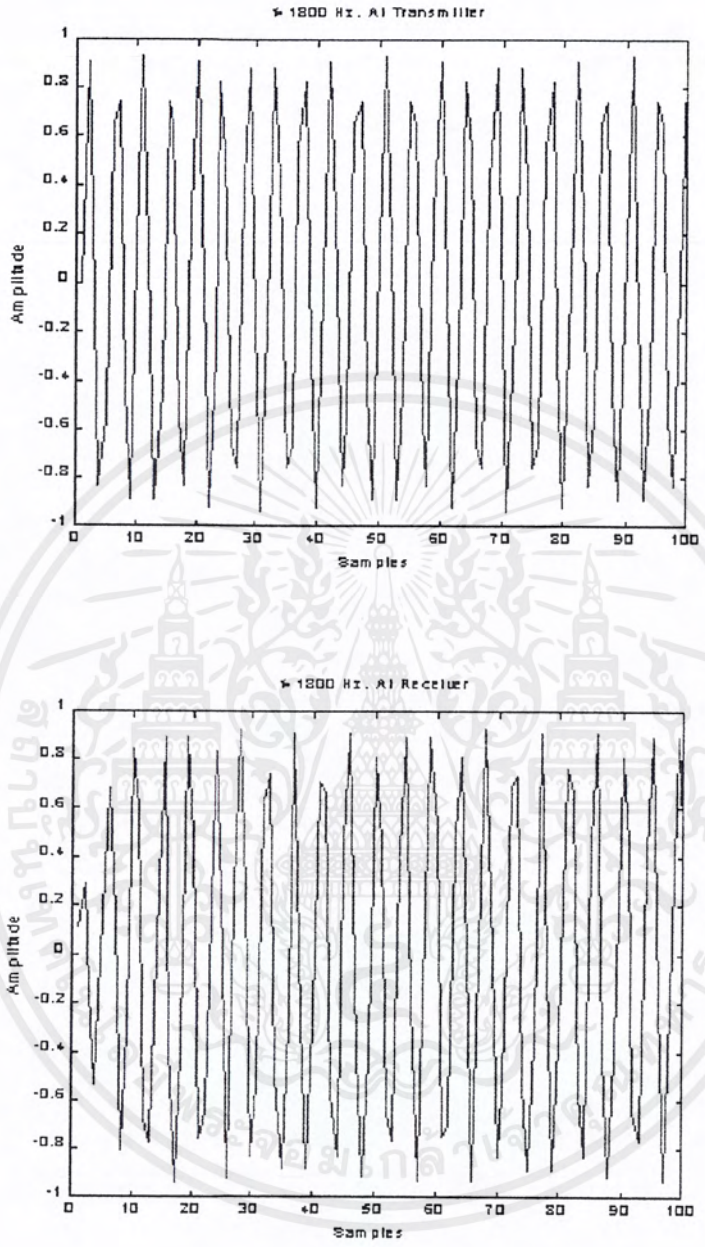
- ความถี่ 1600 เฮิรตซ์




รูปที่ 4.39 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

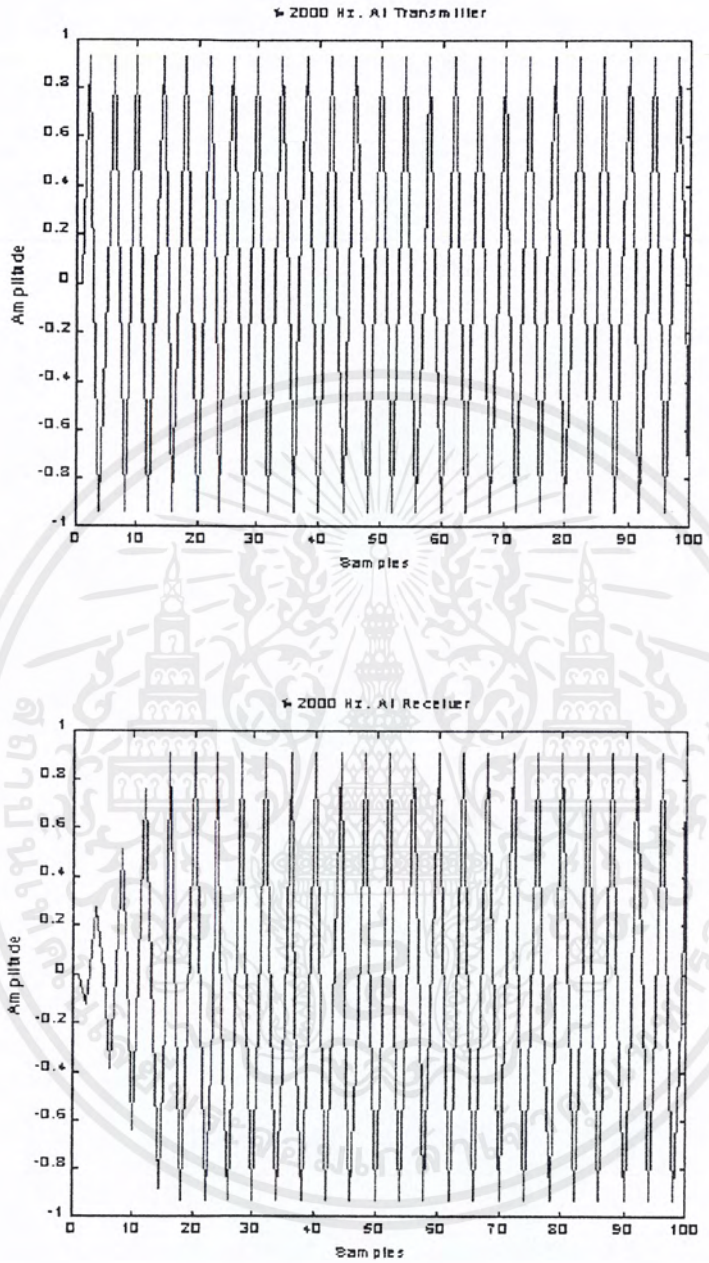
- ความถี่ 1800 เฮิรตซ์




รูปที่ 4.40 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

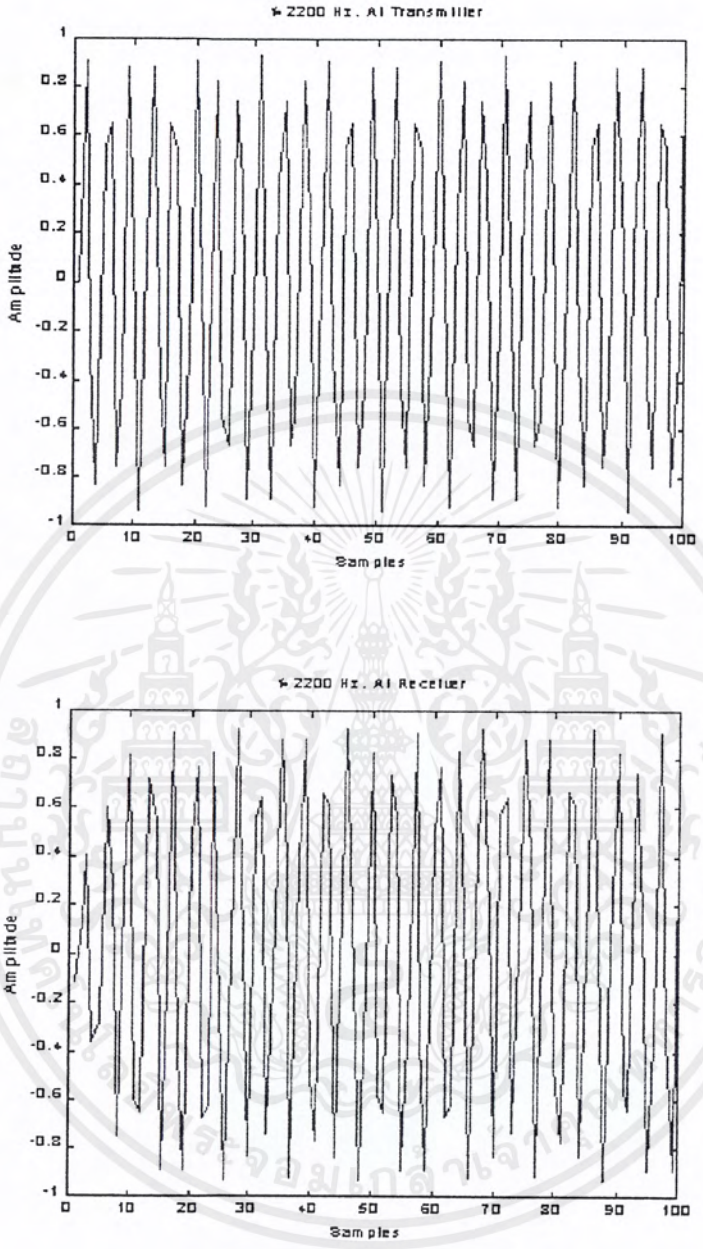
- ความถี่ 2000 เฮิรตซ์




รูปที่ 4.41 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

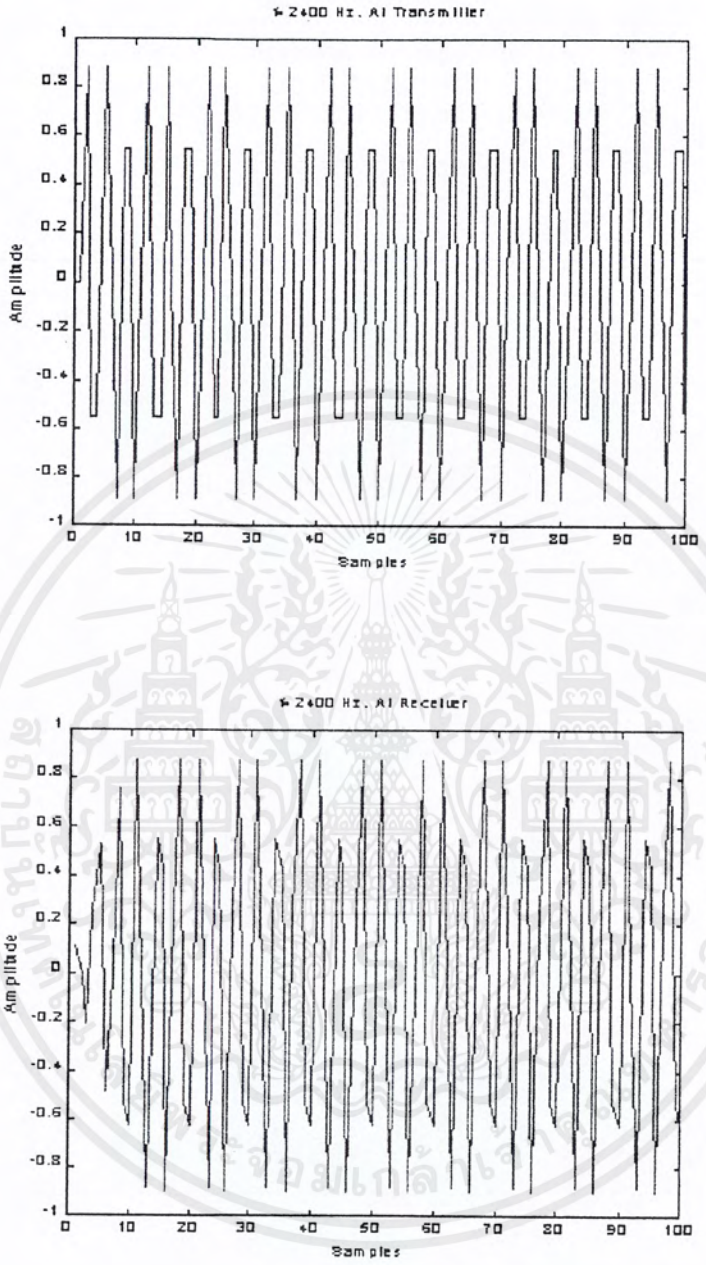
- ความถี่ 2200 เฮิรตซ์




รูปที่ 4.42 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

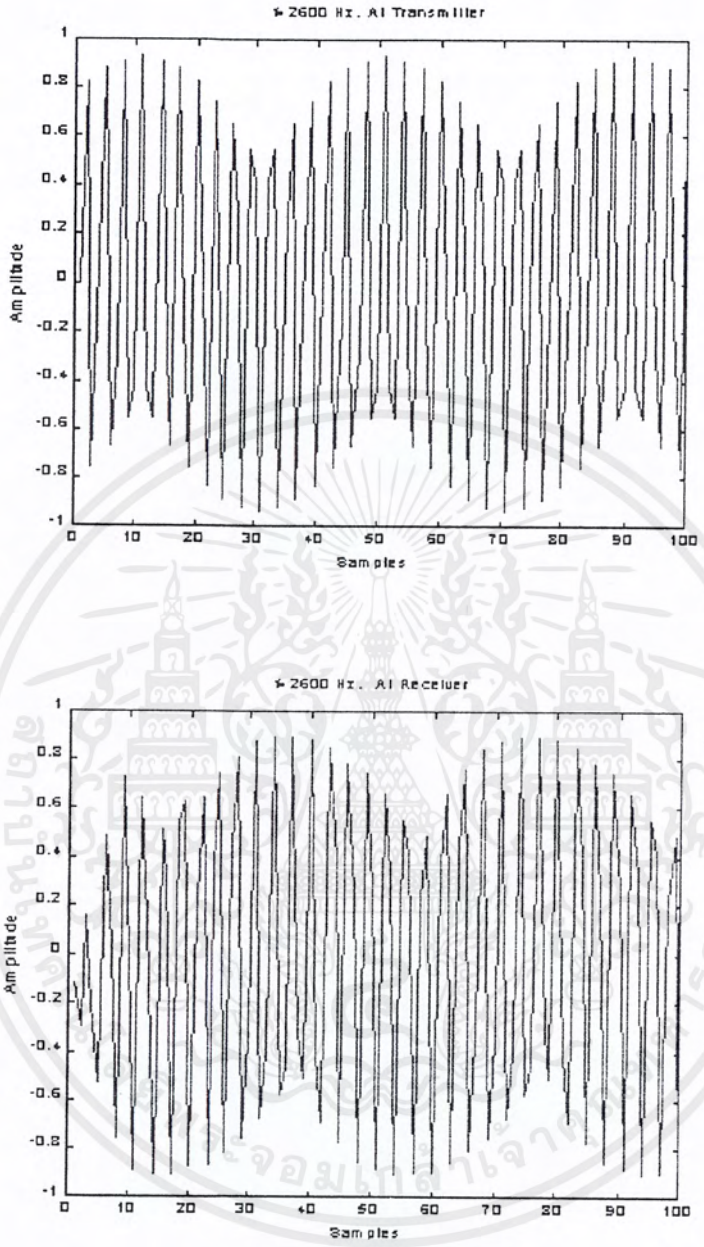
- ความถี่ 2400 เฮิรตซ์




รูปที่ 4.43 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

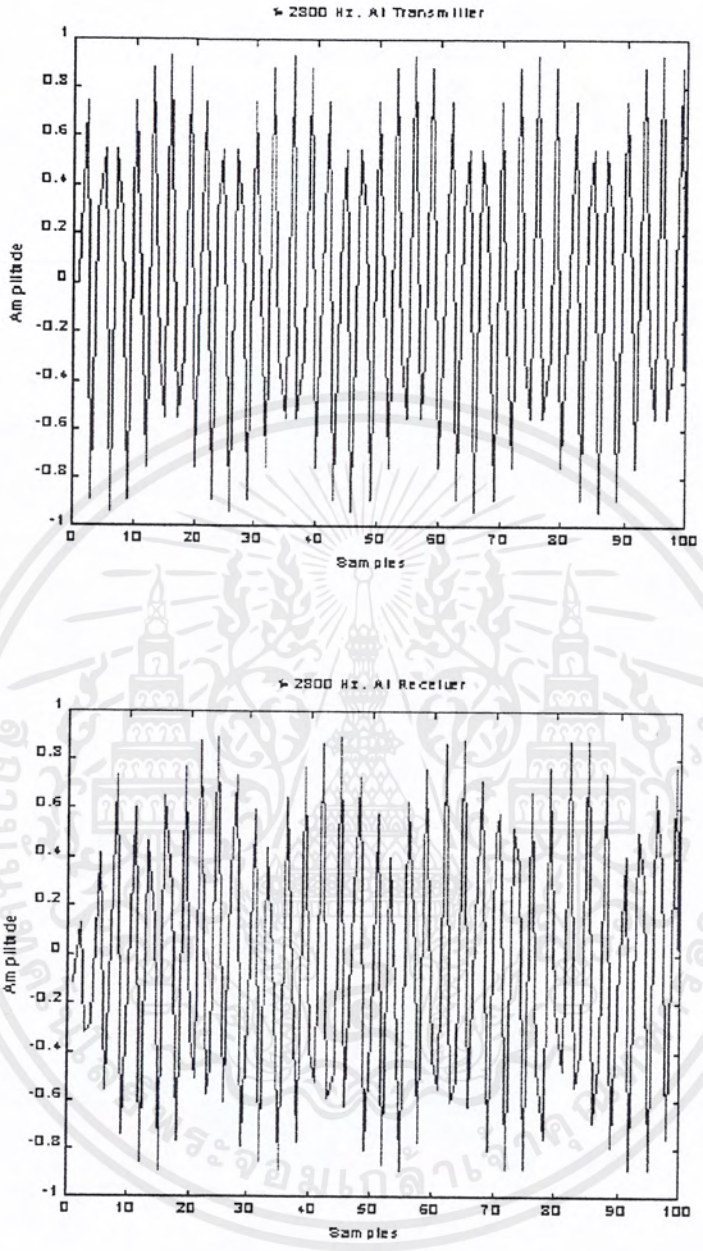
- ความถี่ 2600 เฮิรตซ์




รูปที่ 4.44 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

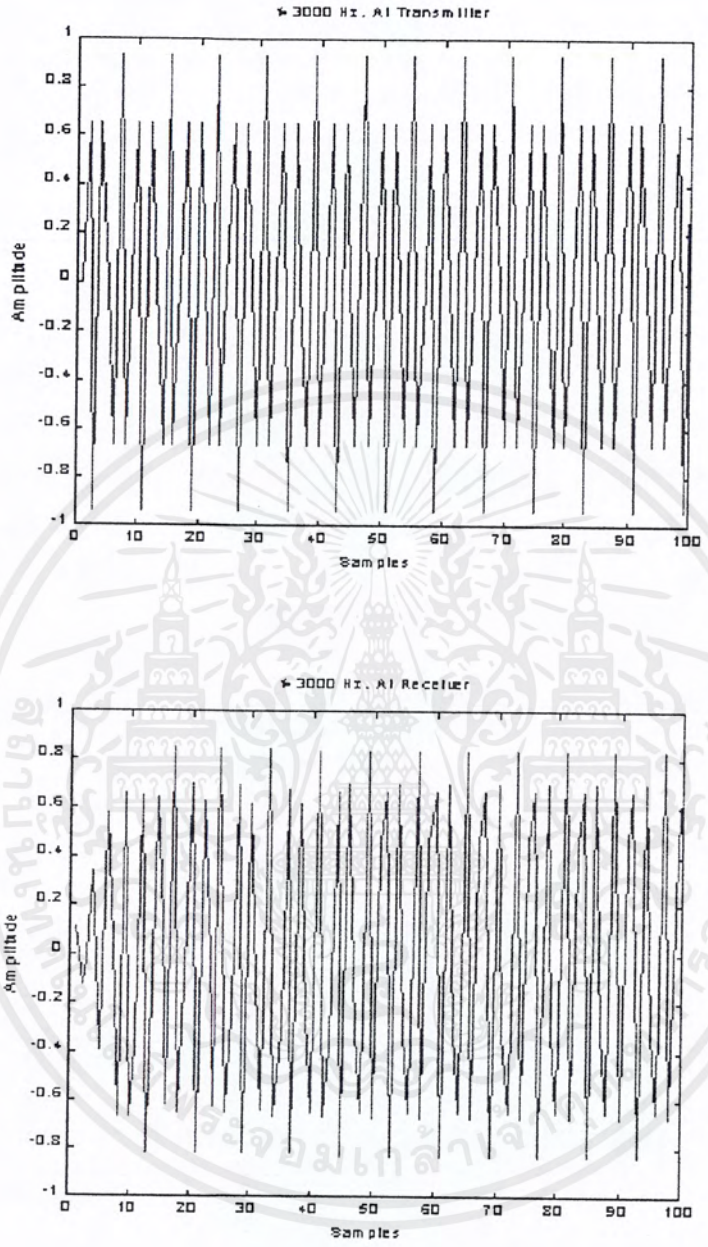
- ความถี่ 2800 เฮิรตซ์




รูปที่ 4.45 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

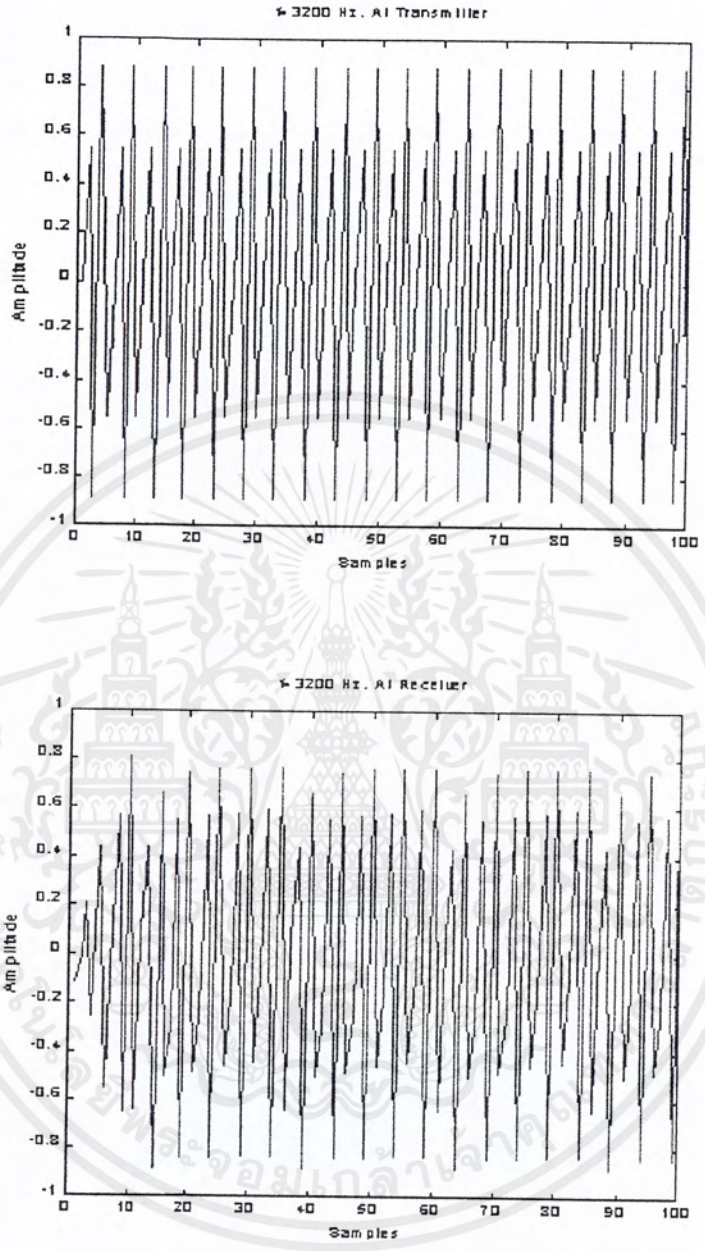
- ความถี่ 3000 เฮิรตซ์




รูปที่ 4.46 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

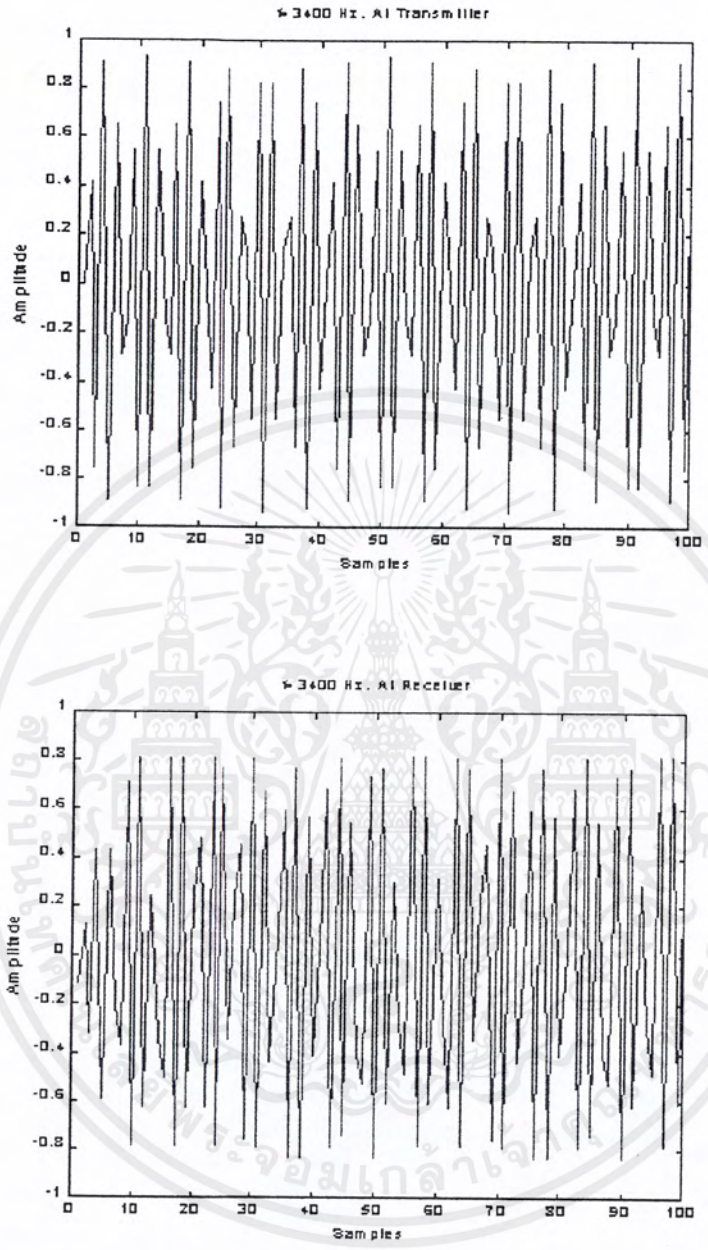
- ความถี่ 3200 เฮิรตซ์




รูปที่ 4.47 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

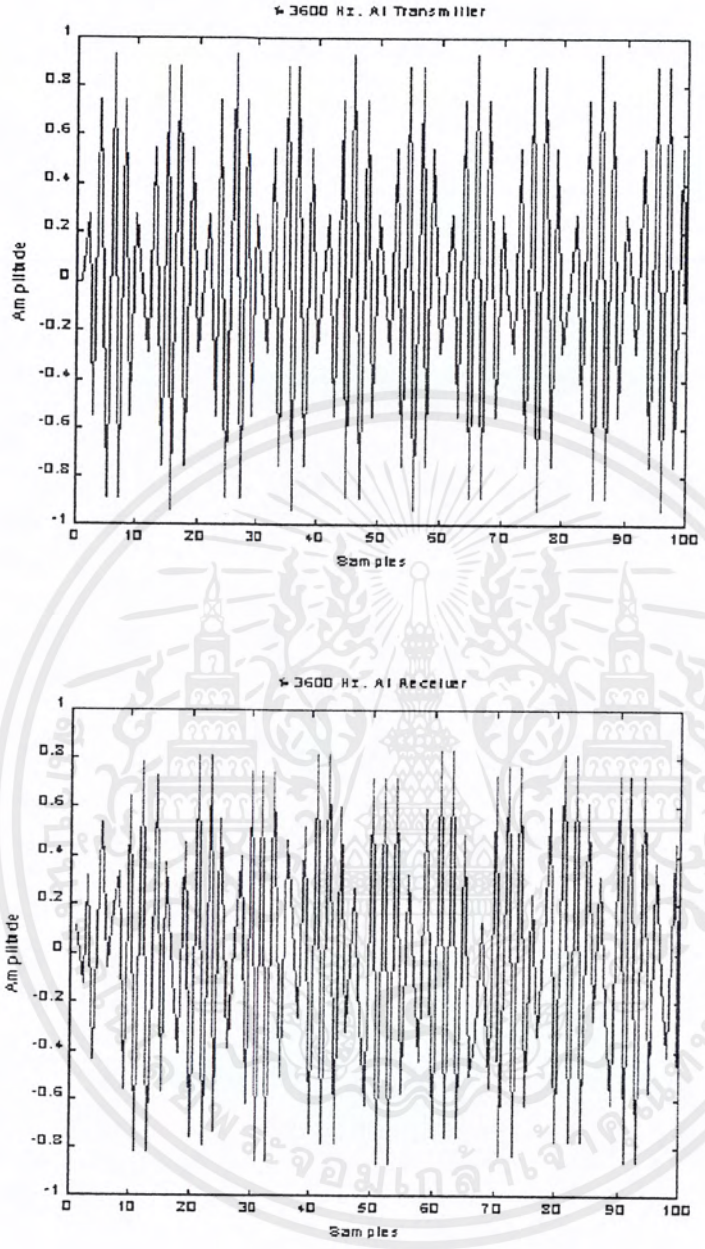
- ความถี่ 3400 เฮิรตซ์




รูปที่ 4.48 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

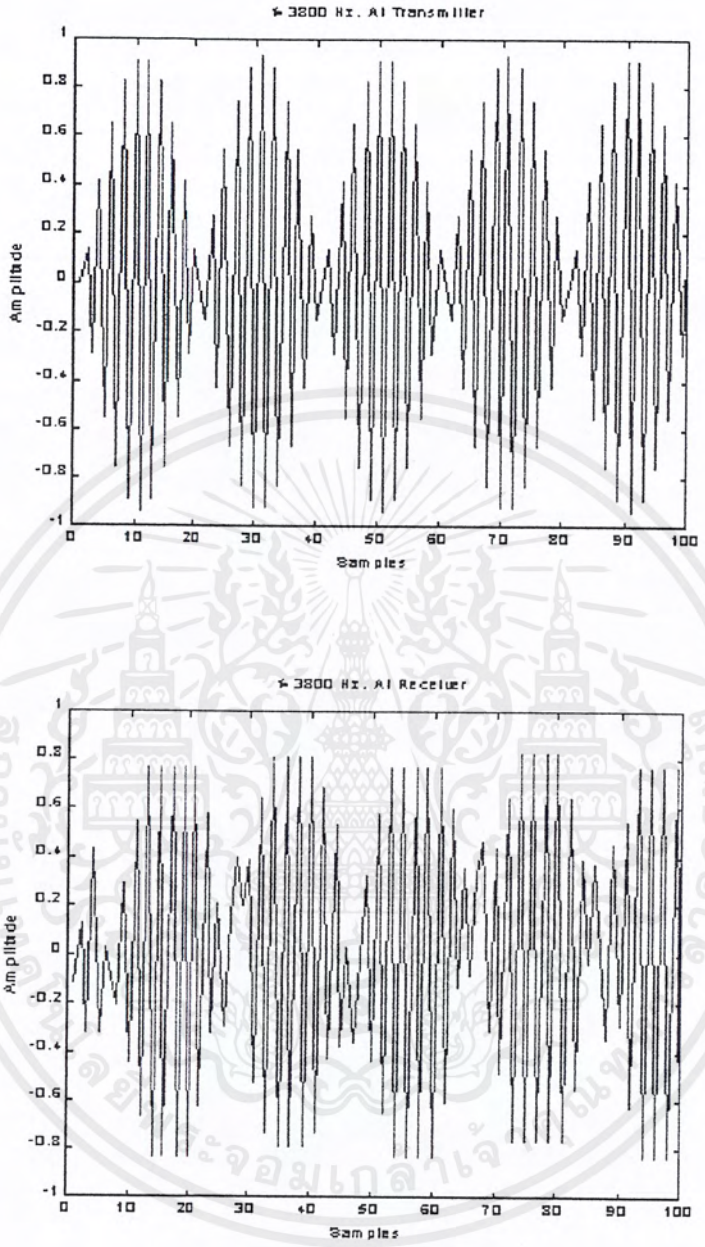
- ความถี่ 3600 เฮิรตซ์




รูปที่ 4.49 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ 3800 เฮิรตซ์

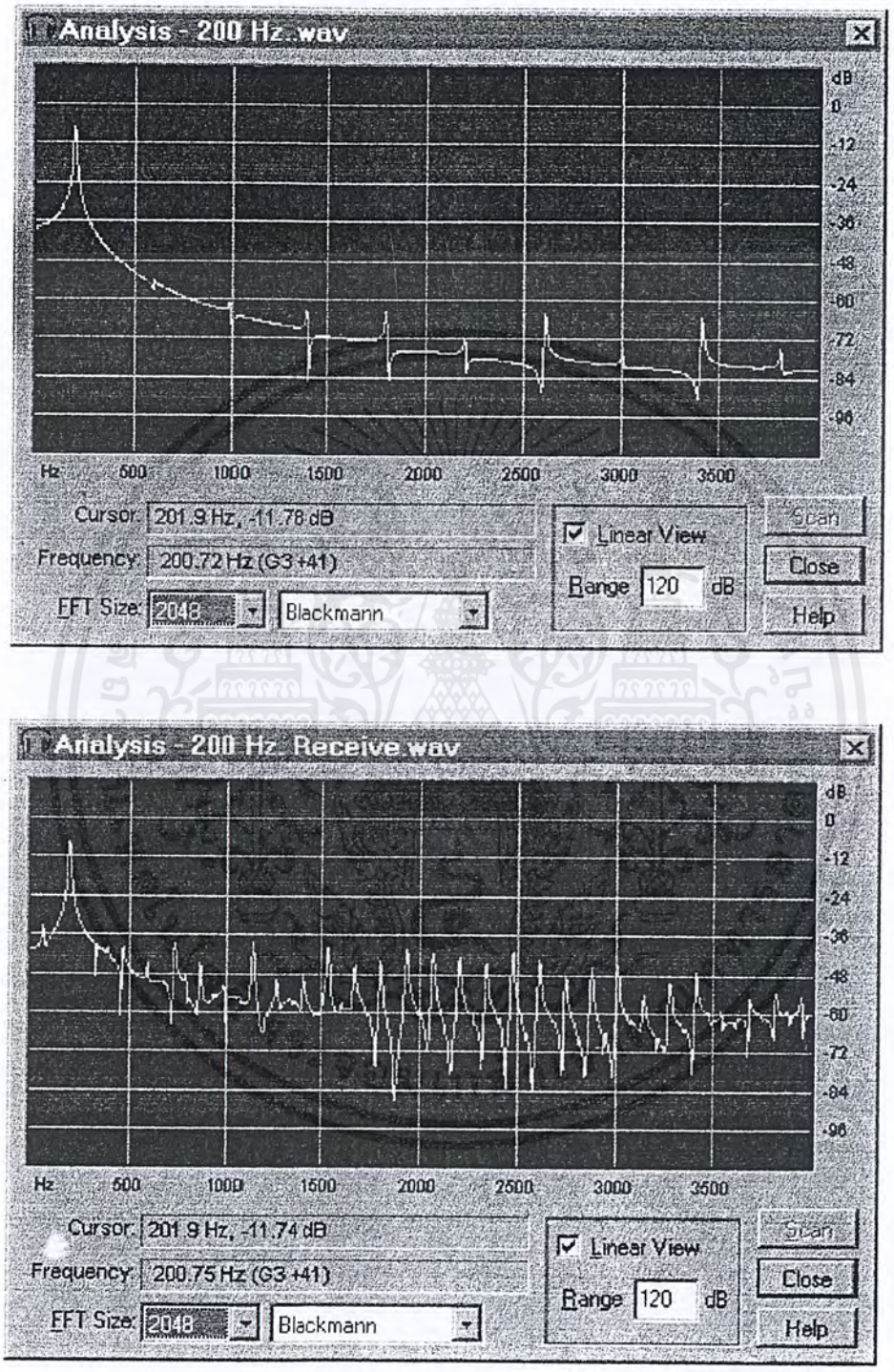



รูปที่ 4.50 เปรียบเทียบเวฟฟอร์มทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 แสดงในรูปแบบของกราฟแสดงการวิเคราะห์ทางความถี่ที่ความถี่ต่างๆ

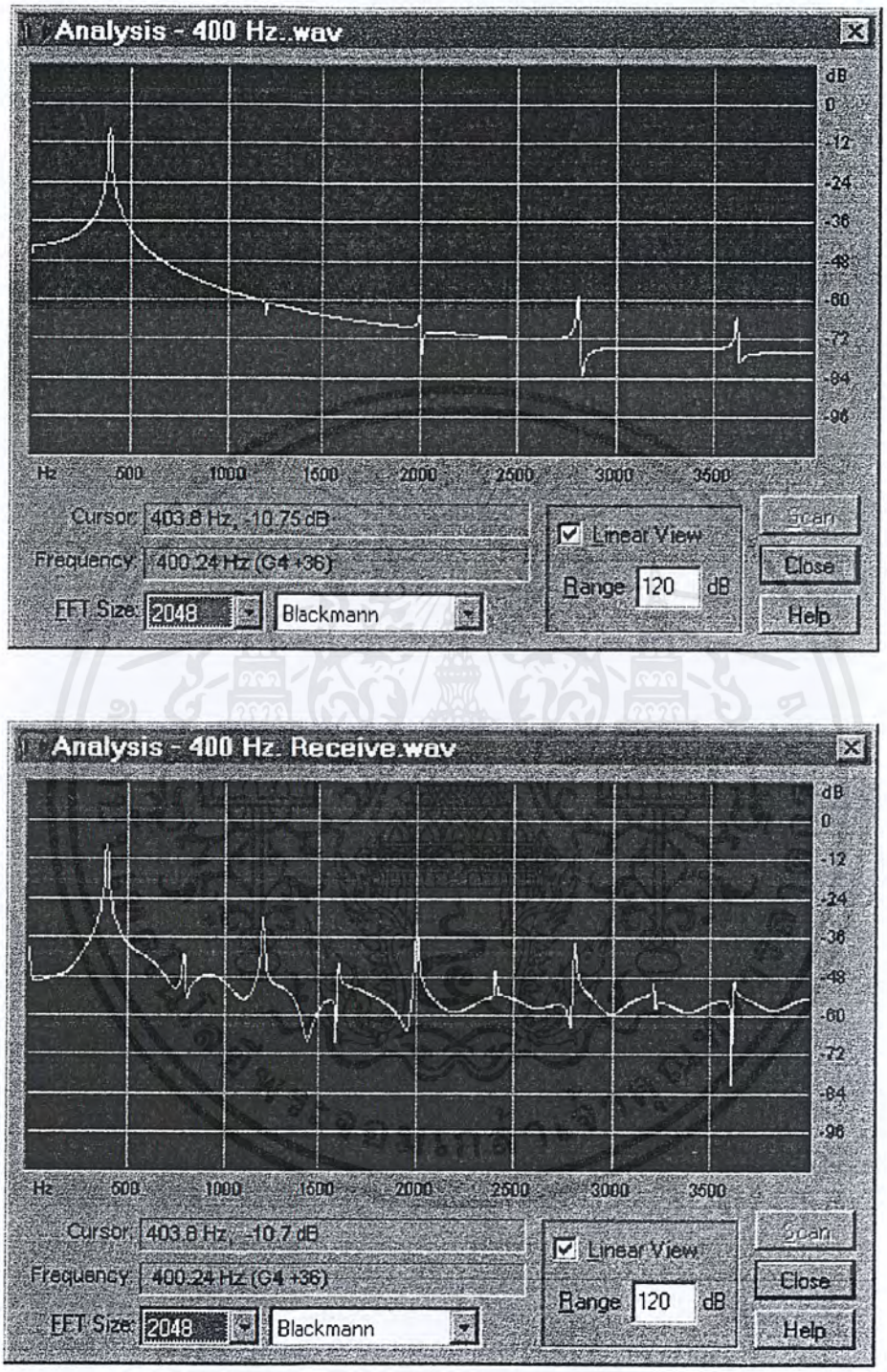
- ความถี่ 200 เฮิรตซ์




รูปที่ 4.51 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

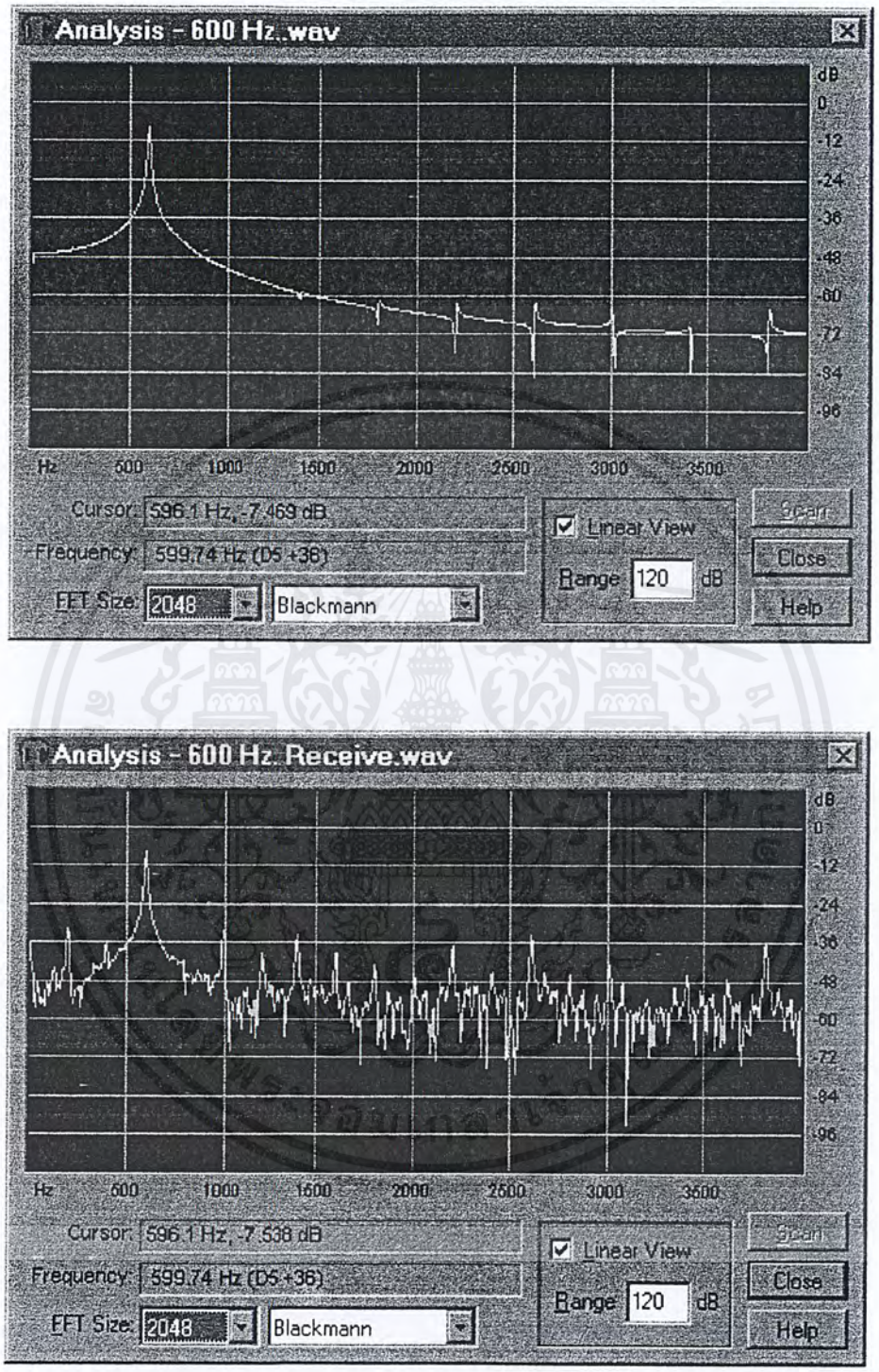
- ความถี่ 400 เฮิรตซ์




รูปที่ 4.52 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

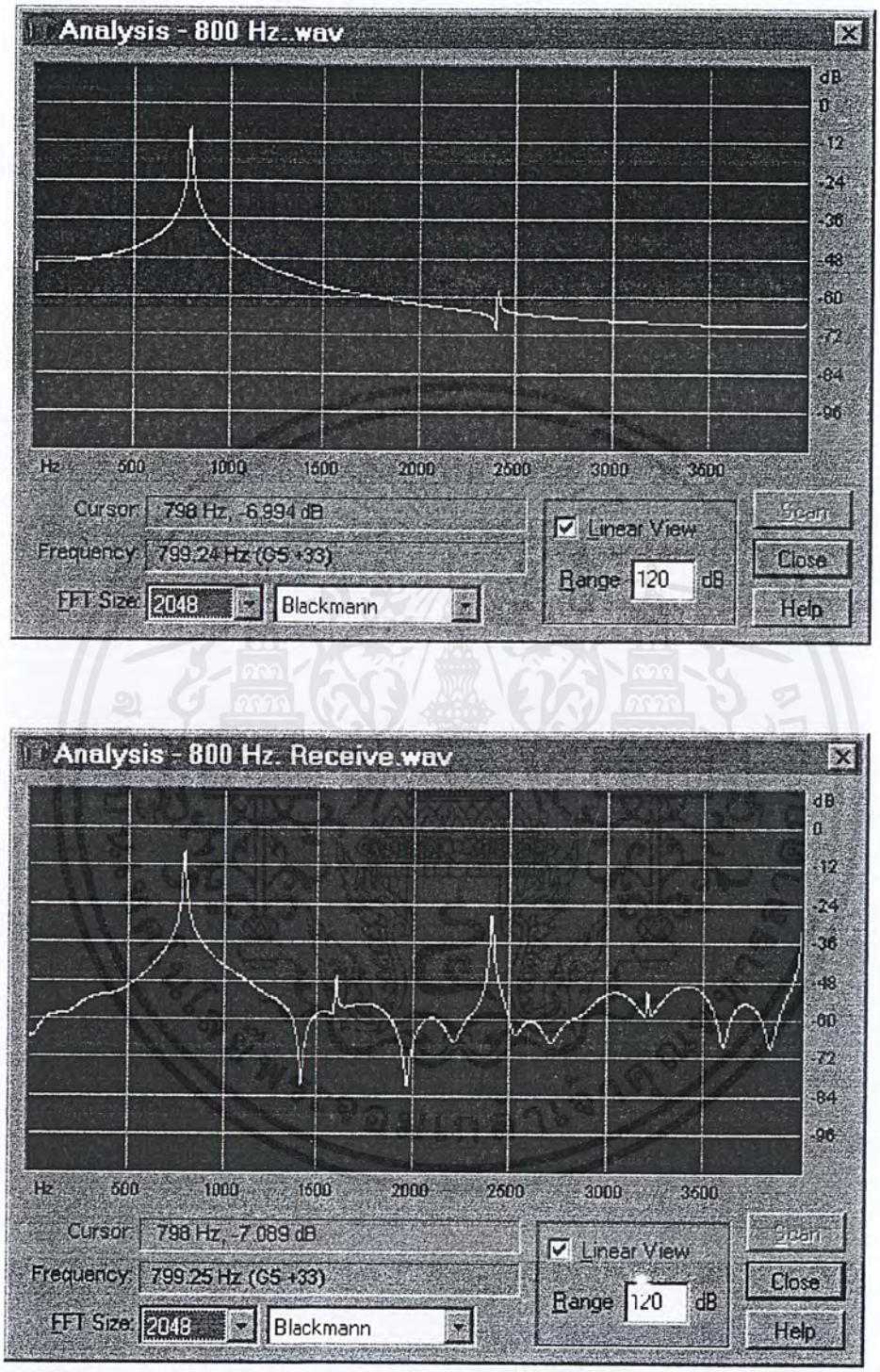
- ความถี่ 600 เฮิรตซ์




รูปที่ 4.53 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

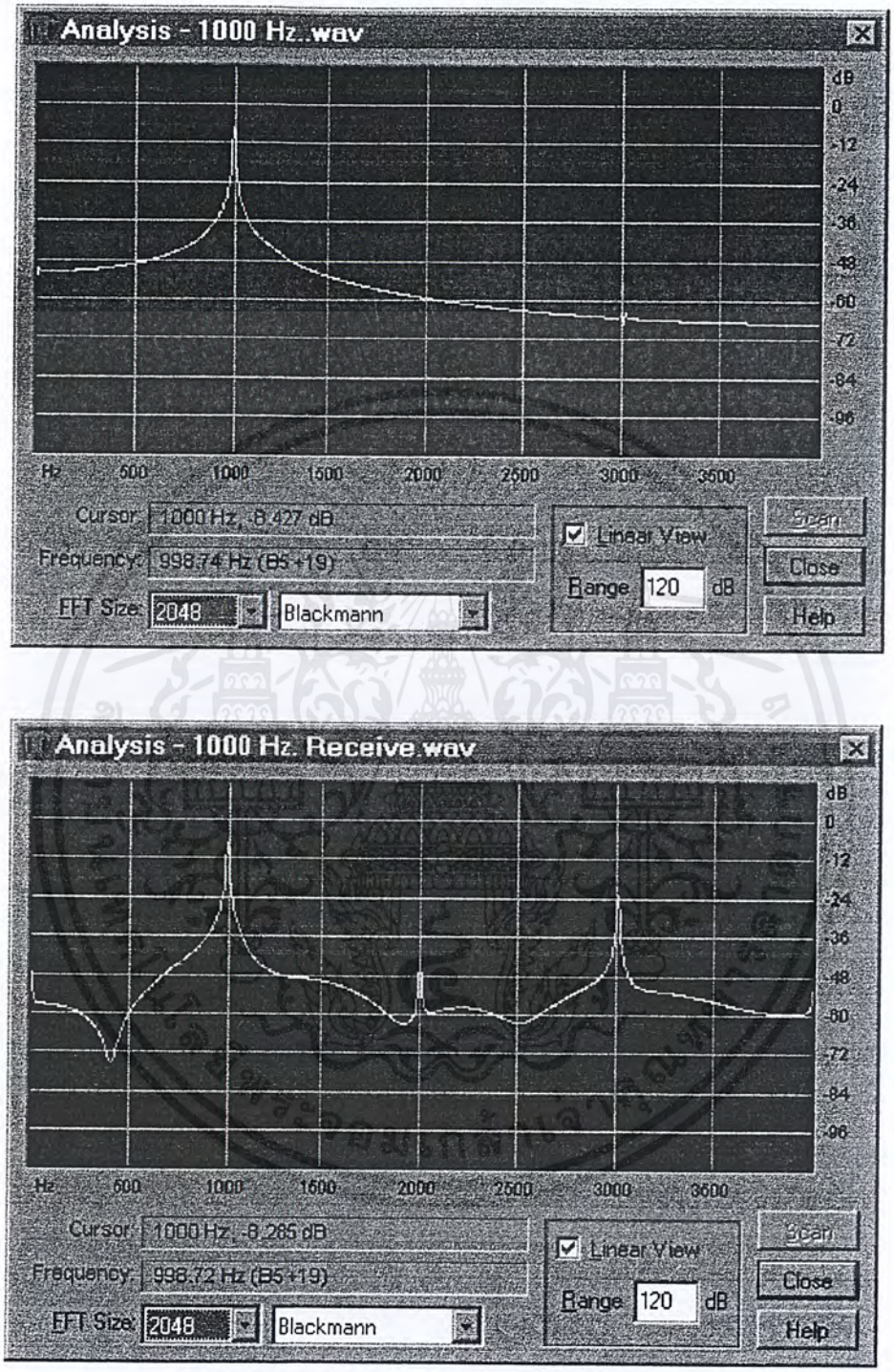
- ความถี่ 800 เฮิรตซ์




รูปที่ 4.54 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

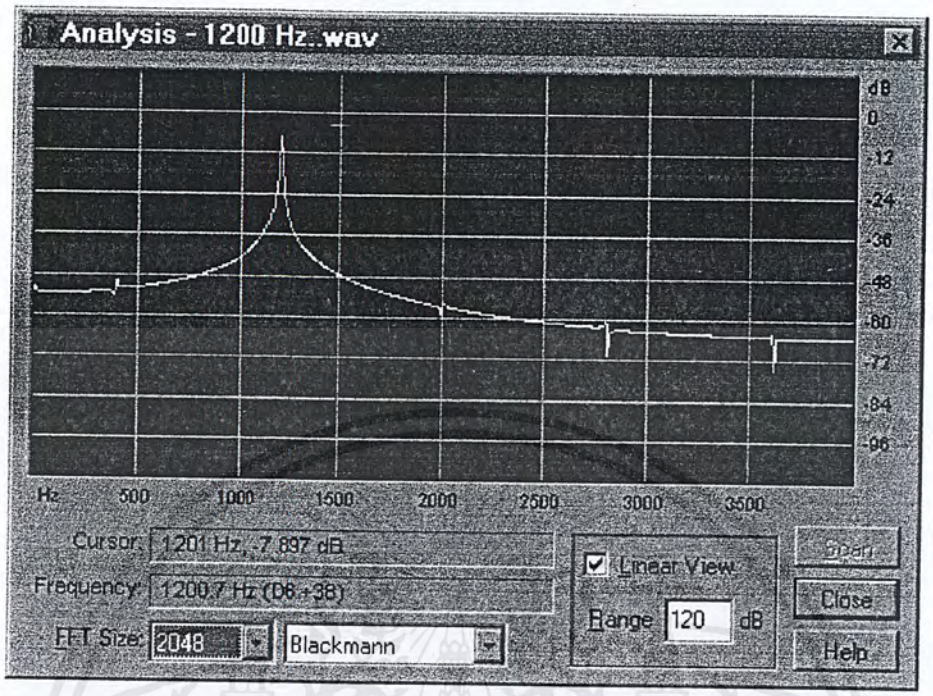
- ความถี่ 1000 เฮิรตซ์

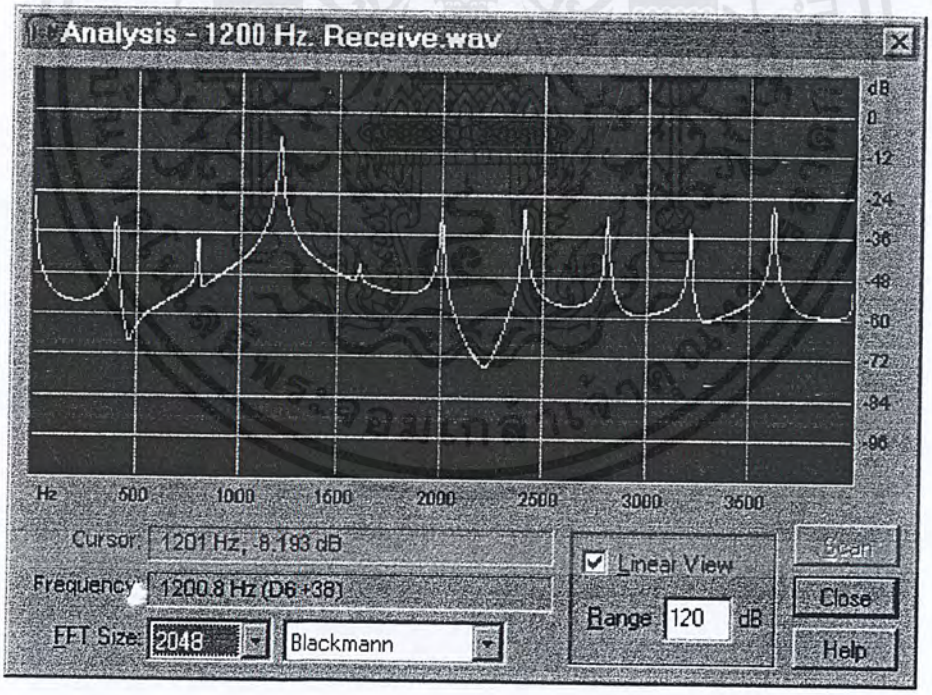



รูปที่ 4.55 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ 1200 เฮิรตซ์

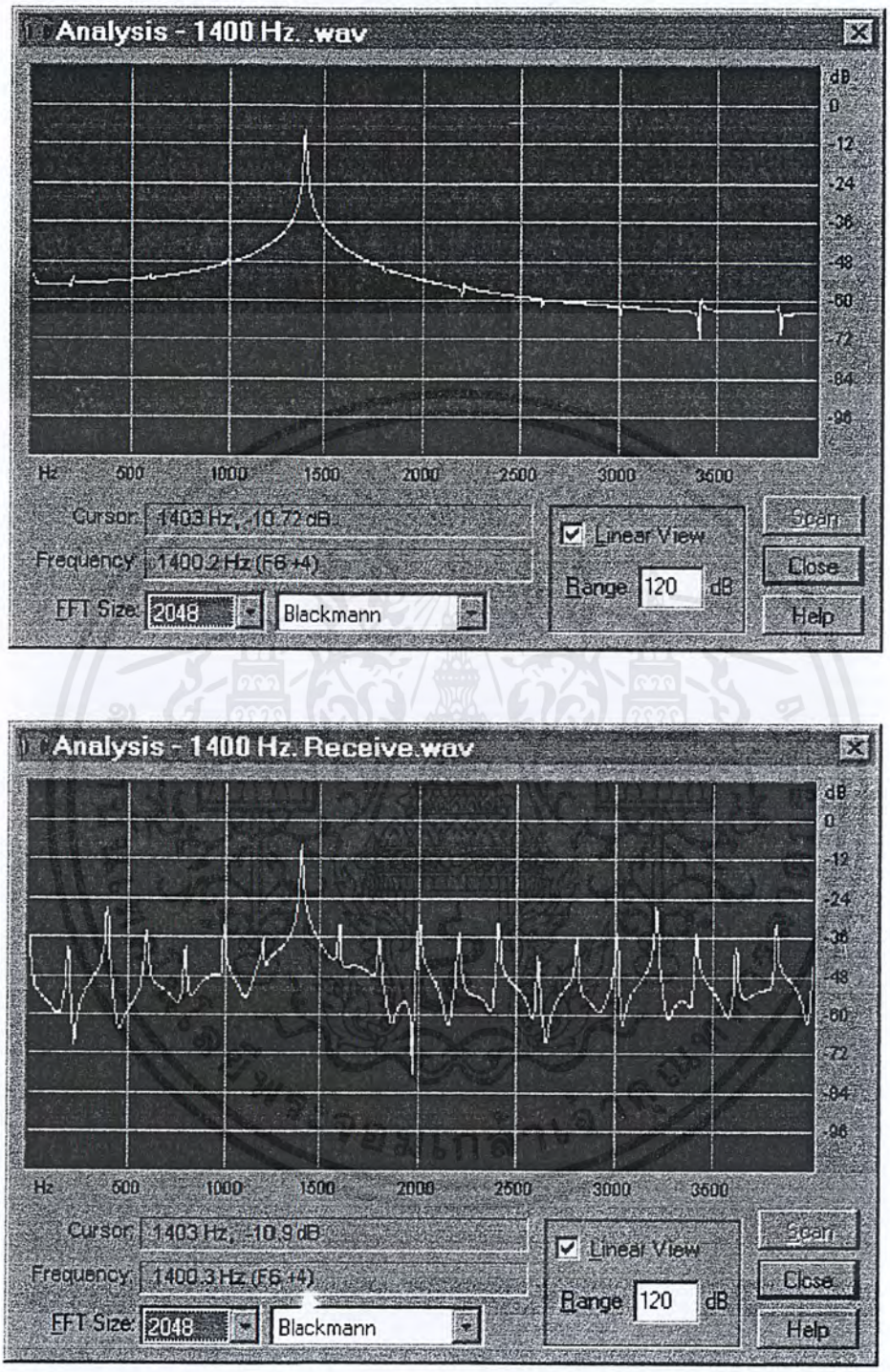



รูปที่ 4.56 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณชายน์ความถี่ 1200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

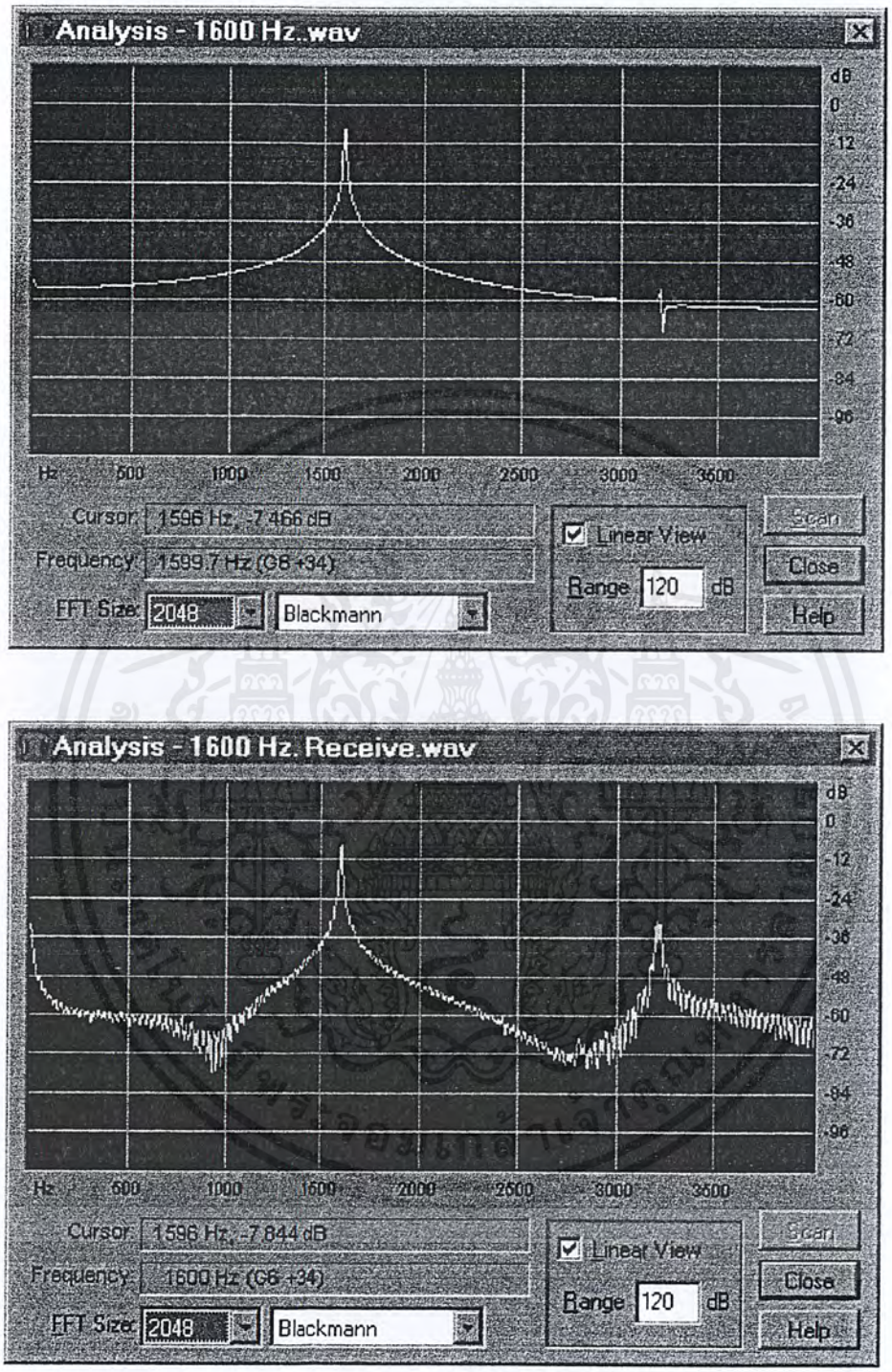
- ความถี่ 1400 เฮิรตซ์




รูปที่ 4.57 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

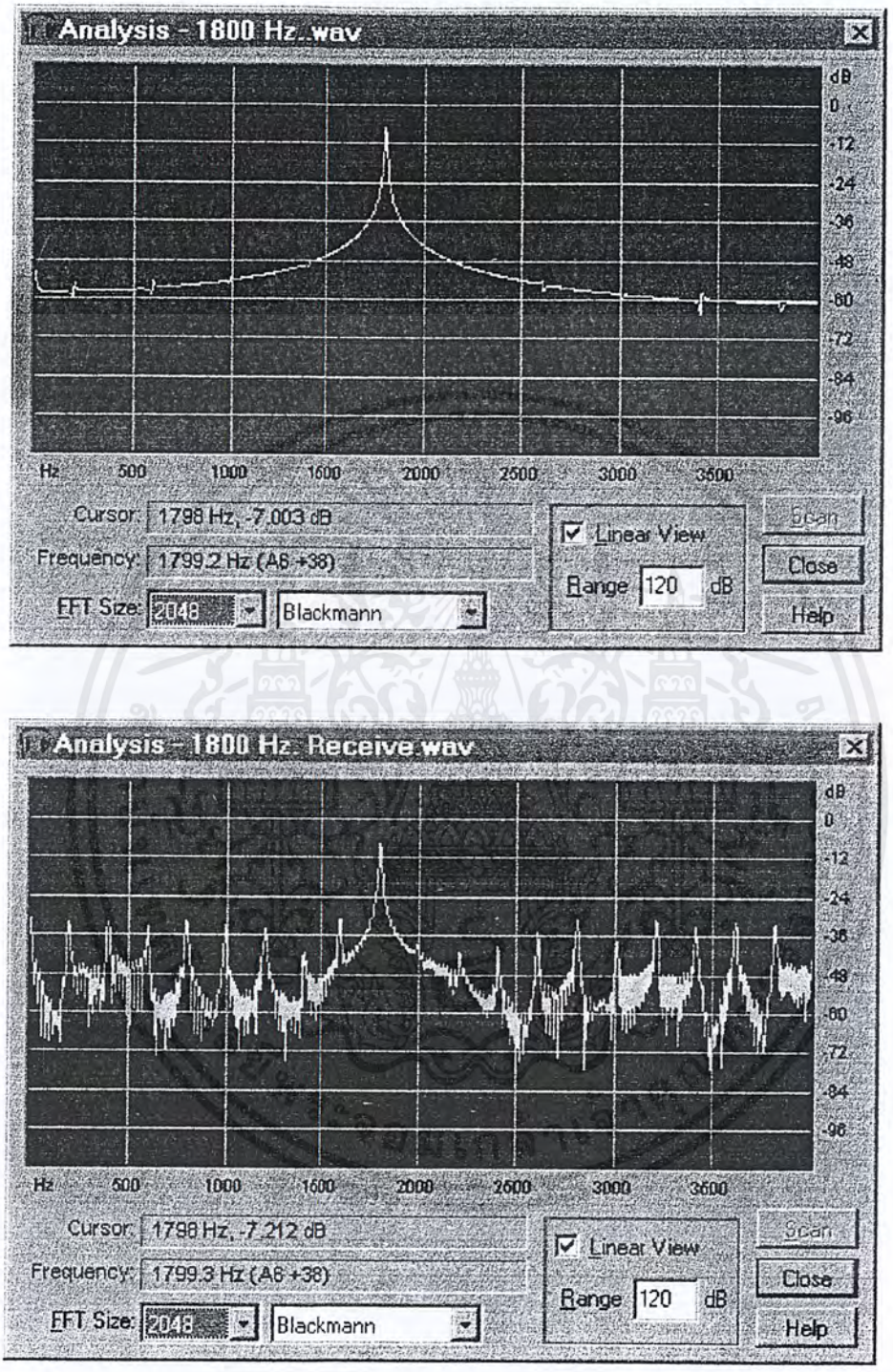
- ความถี่ 1600 เฮิรตซ์




รูปที่ 4.58 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

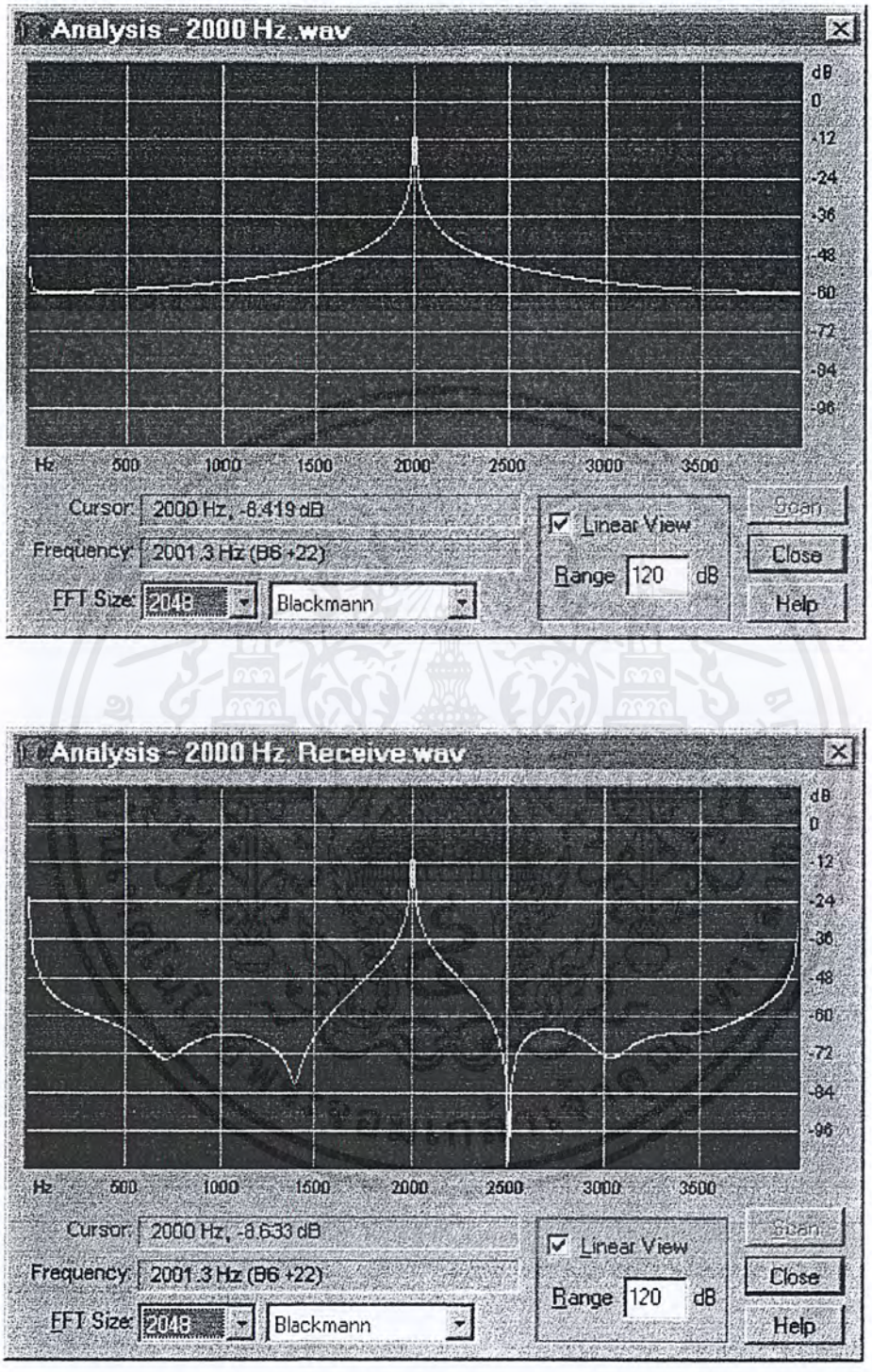
- ความถี่ 1800 เฮิรตซ์




รูปที่ 4.59 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 1800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

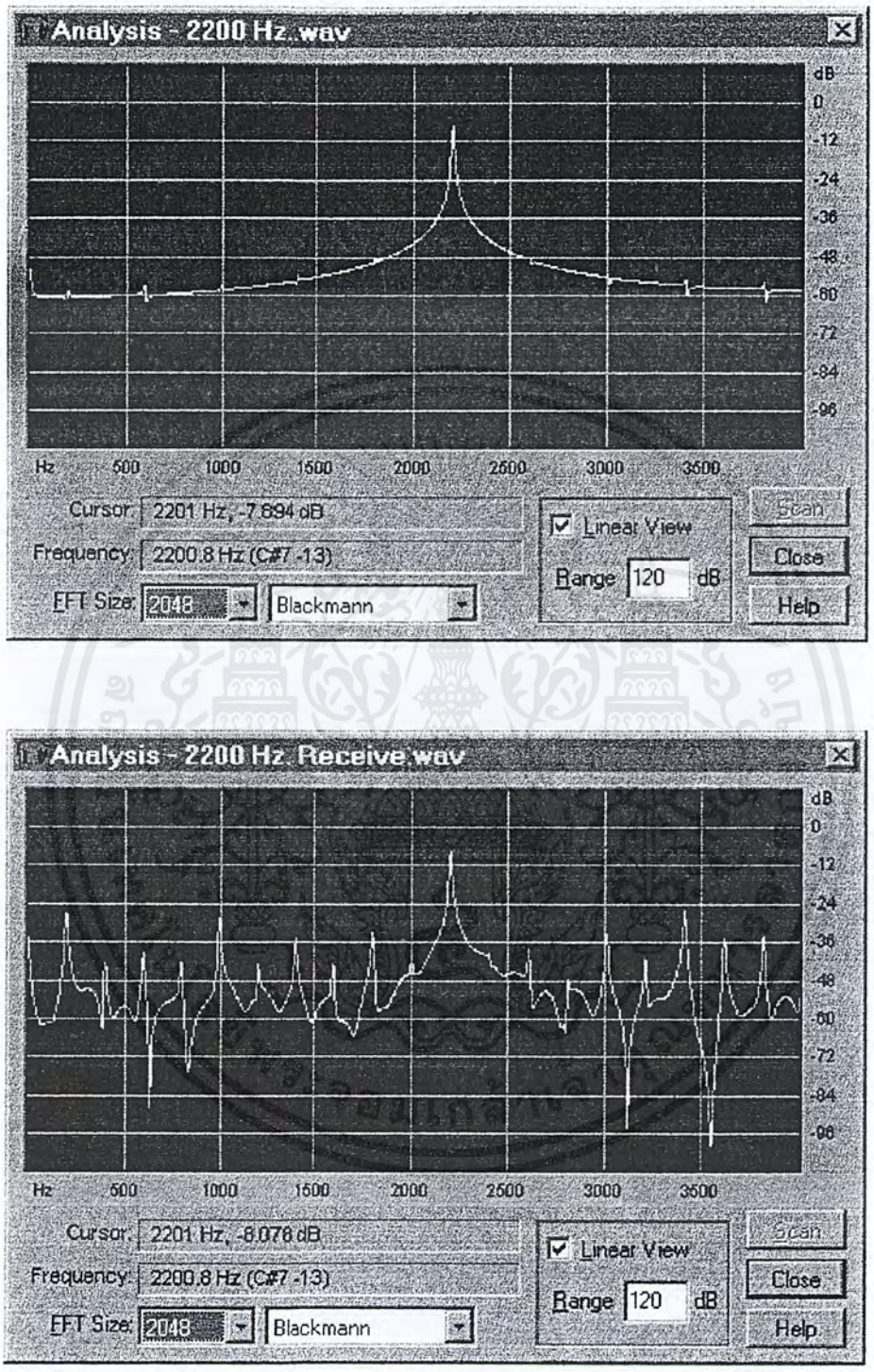
- ความถี่ 2000 เฮิรตซ์




รูปที่ 4.60 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

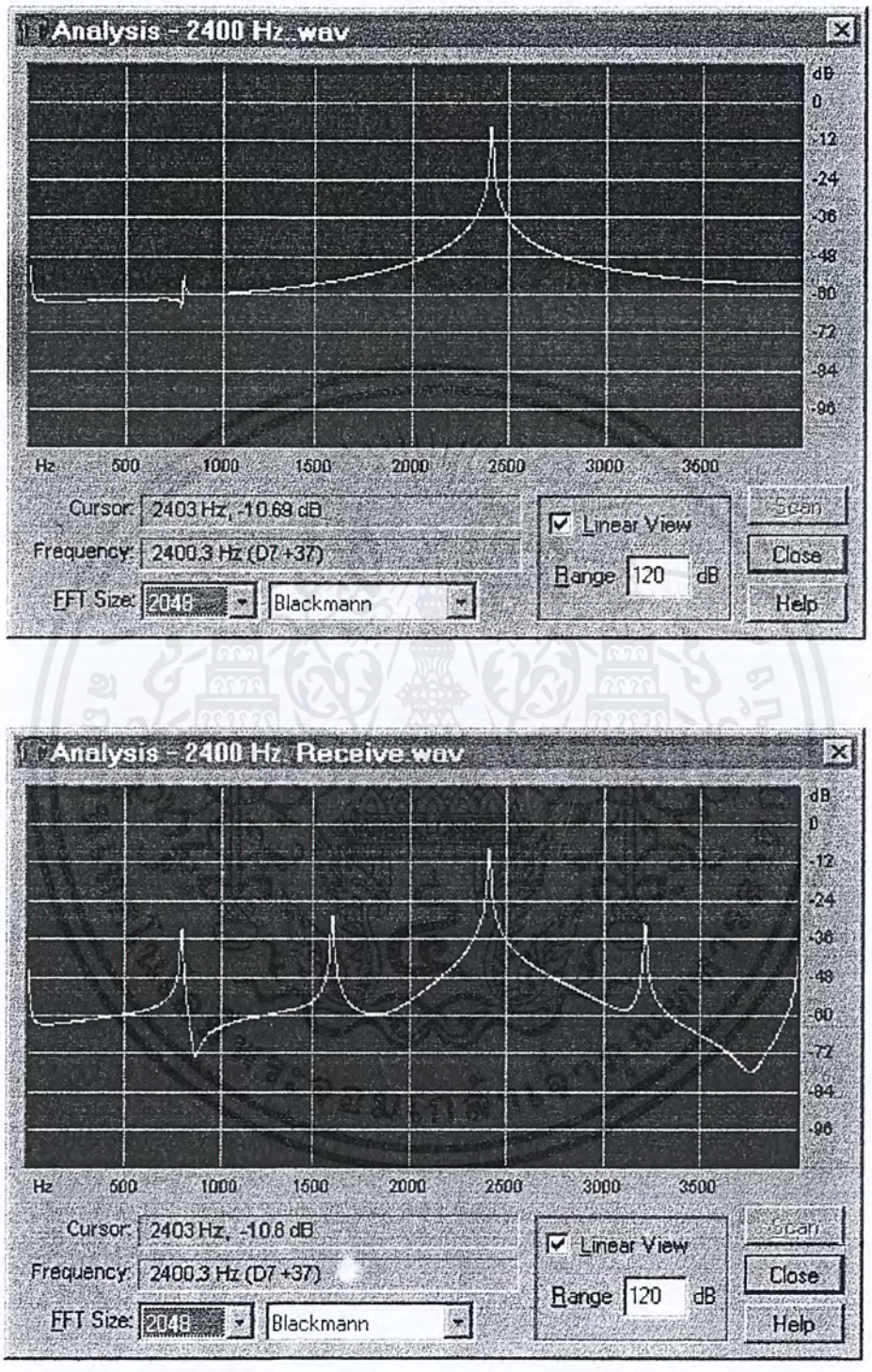
- ความถี่ 2200 เฮิรตซ์




รูปที่ 4.61 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

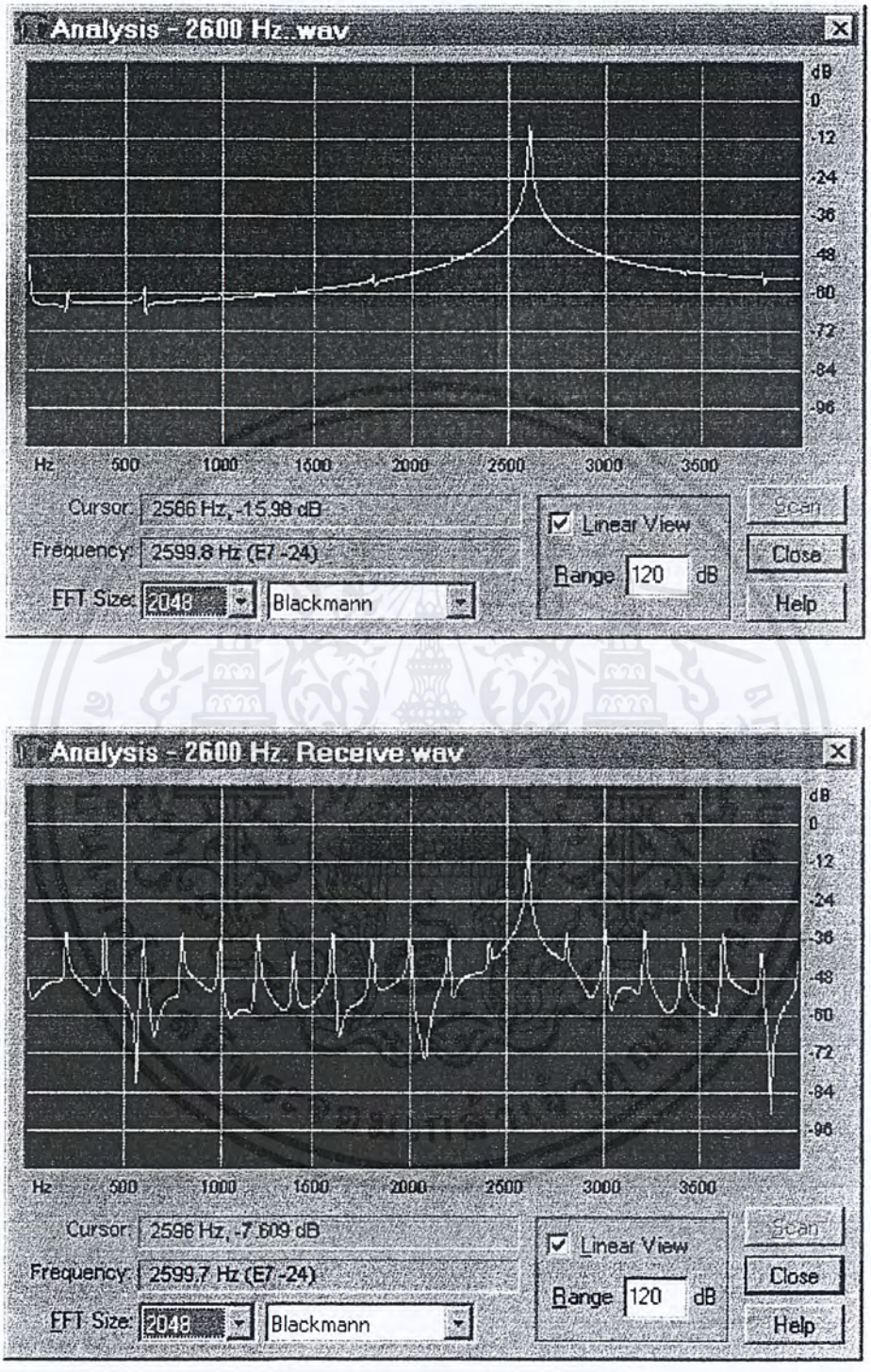
- ความถี่ 2400 เฮิรตซ์




รูปที่ 4.62 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

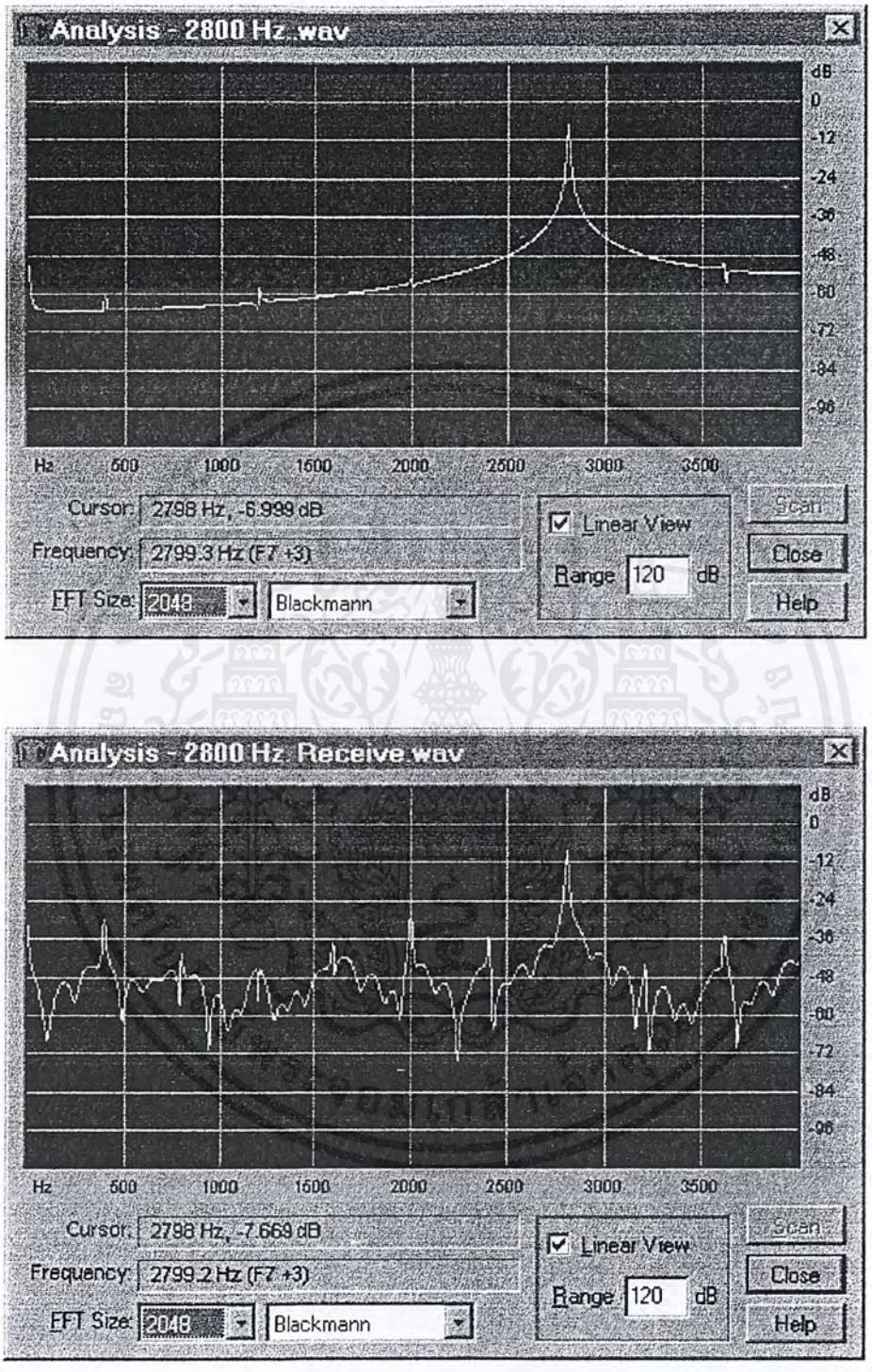
- ความถี่ 2600 เฮิรตซ์




รูปที่ 4.63 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

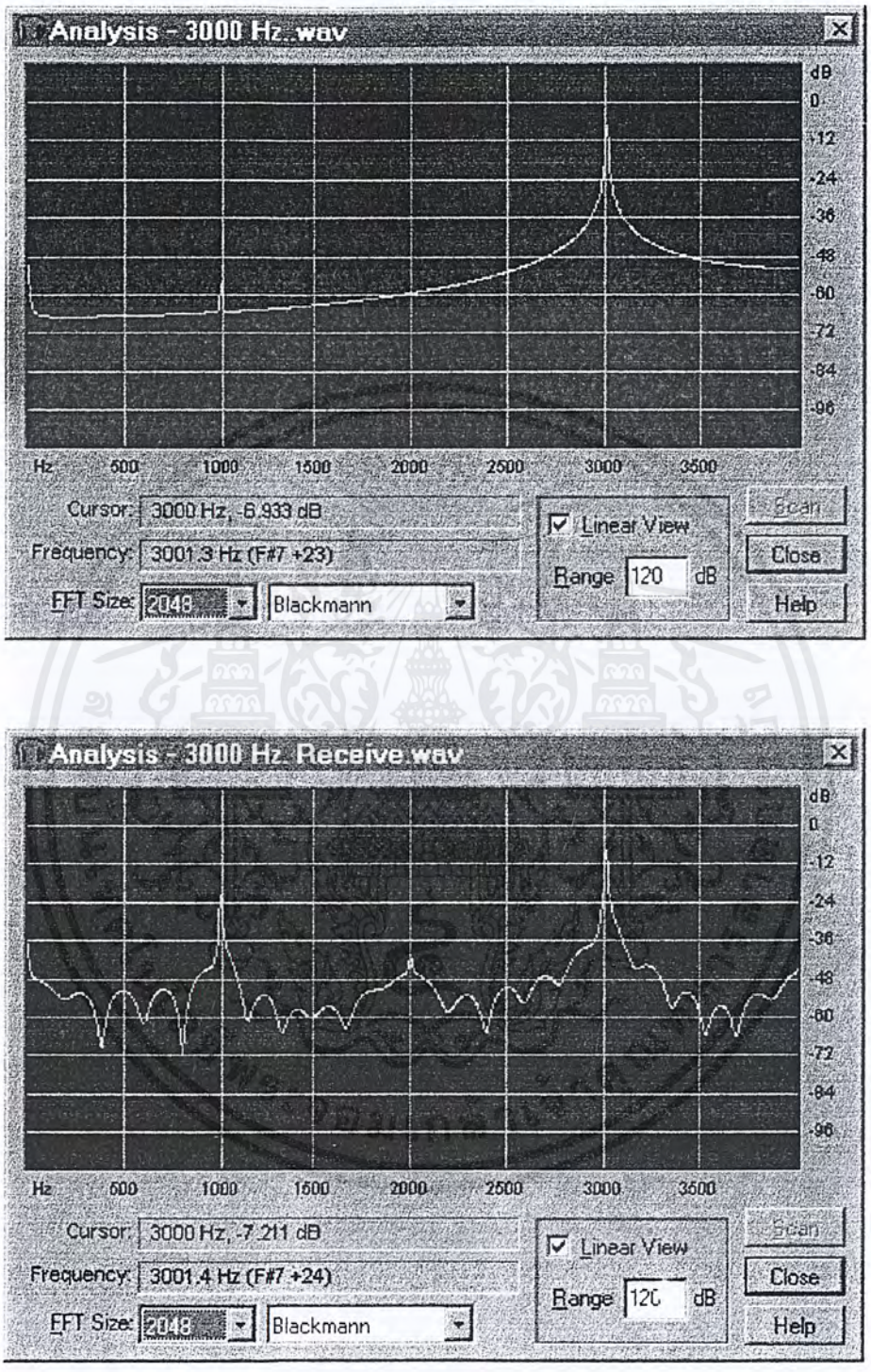
- ความถี่ 2800 เฮิรตซ์




รูปที่ 4.64 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 2800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

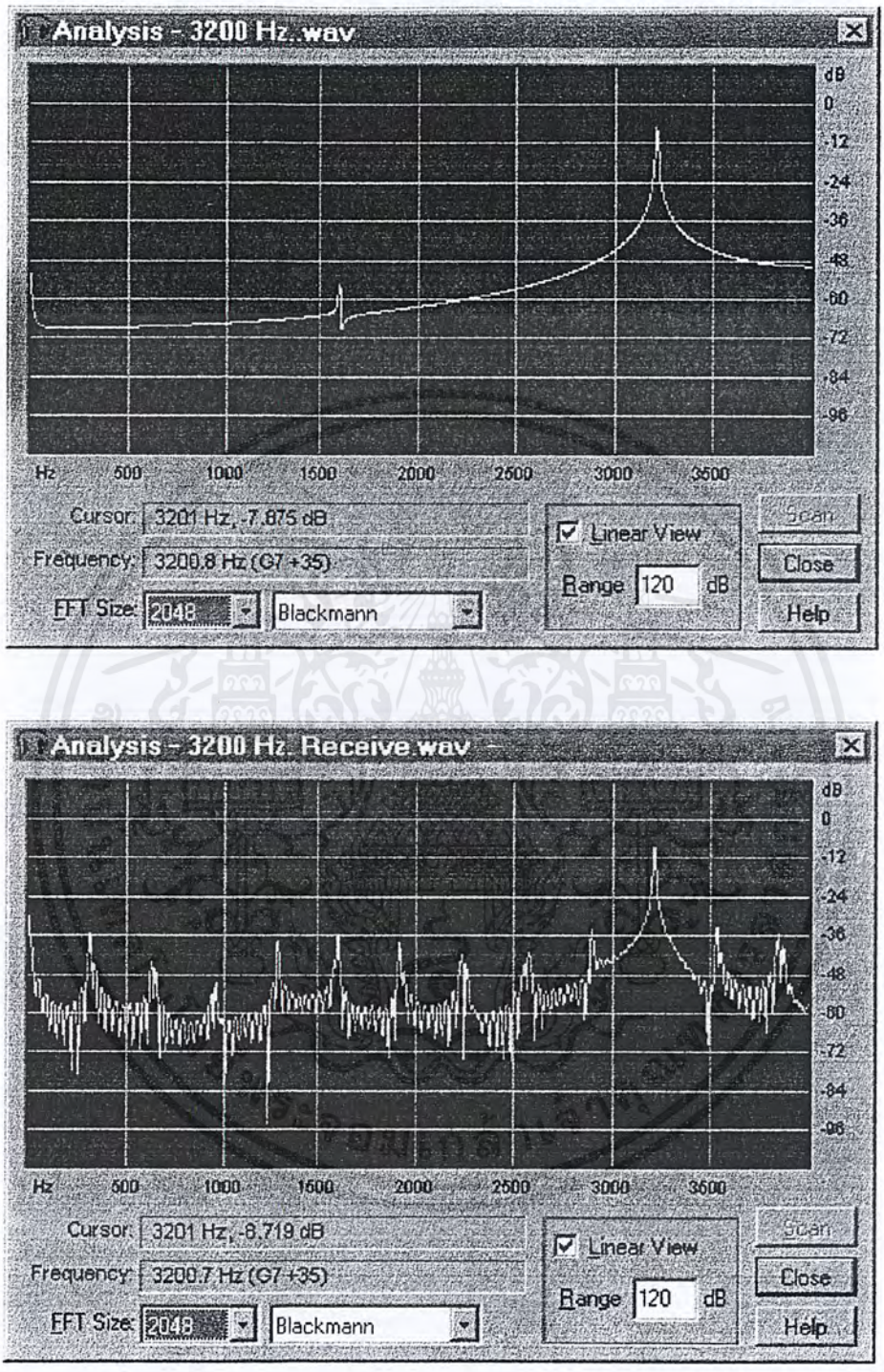
- ความถี่ 3000 เฮิรตซ์




รูปที่ 4.65 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3000 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

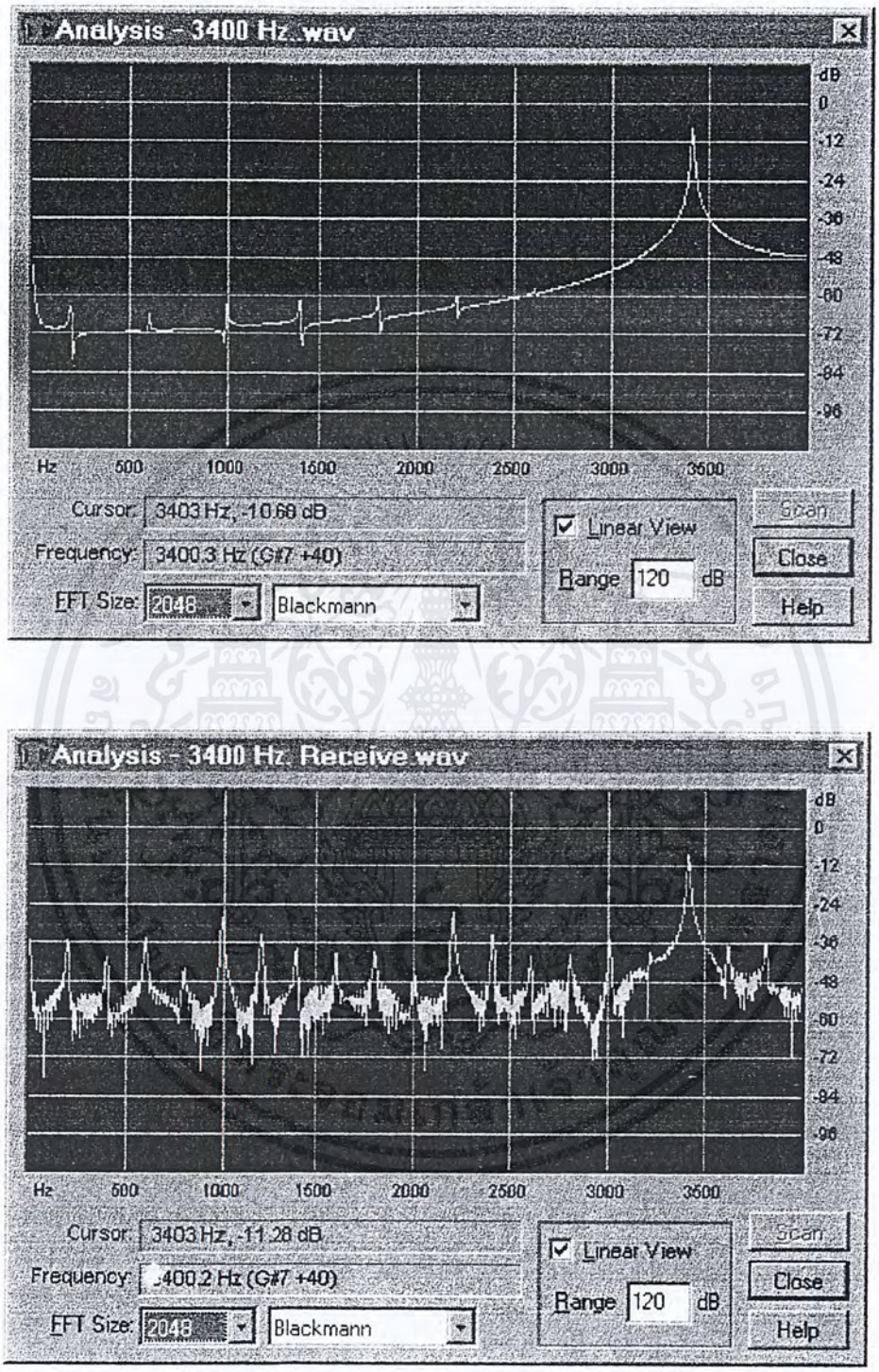
- ความถี่ 3200 เฮิรตซ์




รูปที่ 4.66 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3200 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

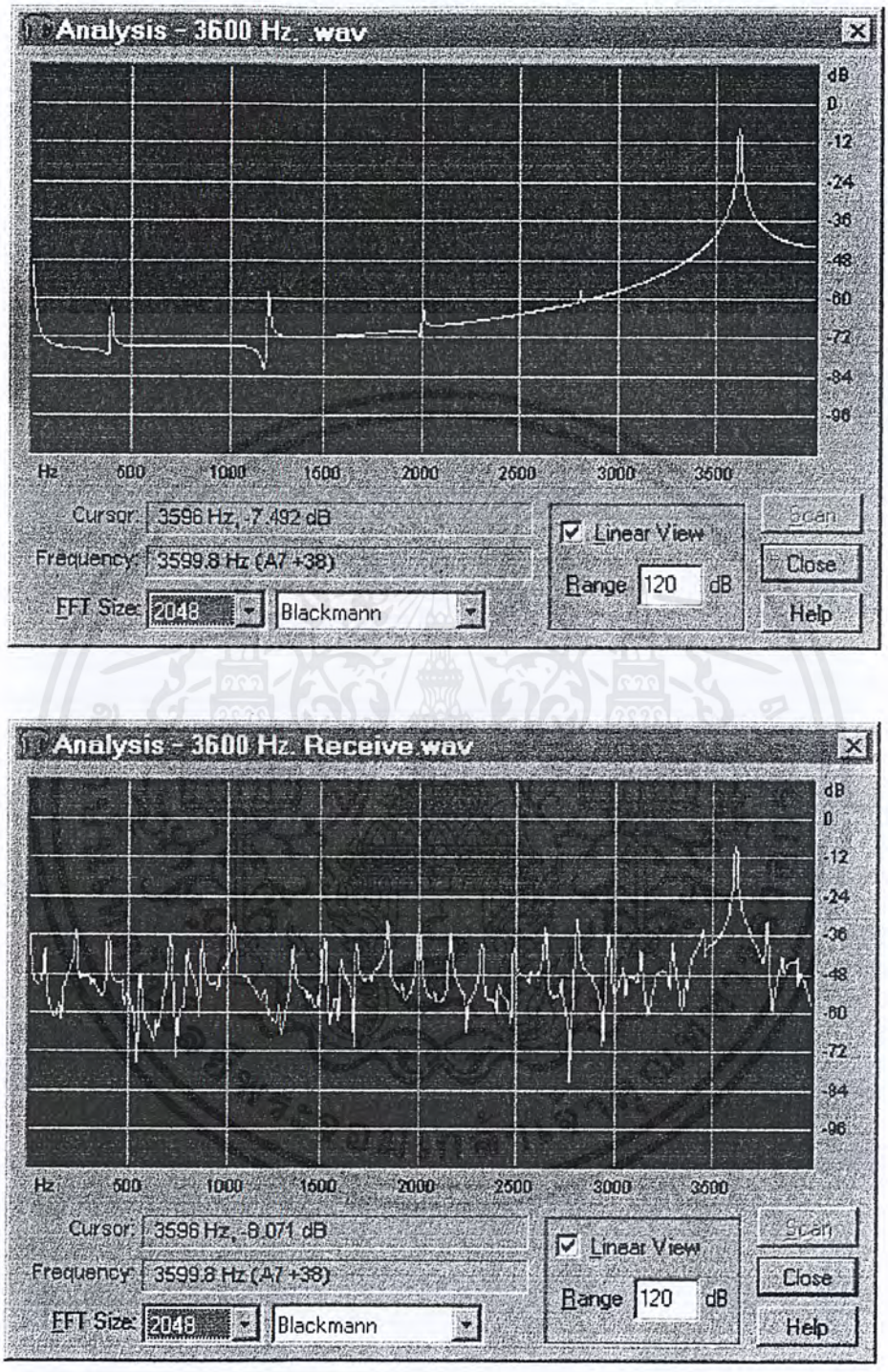
- ความถี่ 3400 เฮิรตซ์




รูปที่ 4.67 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3400 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

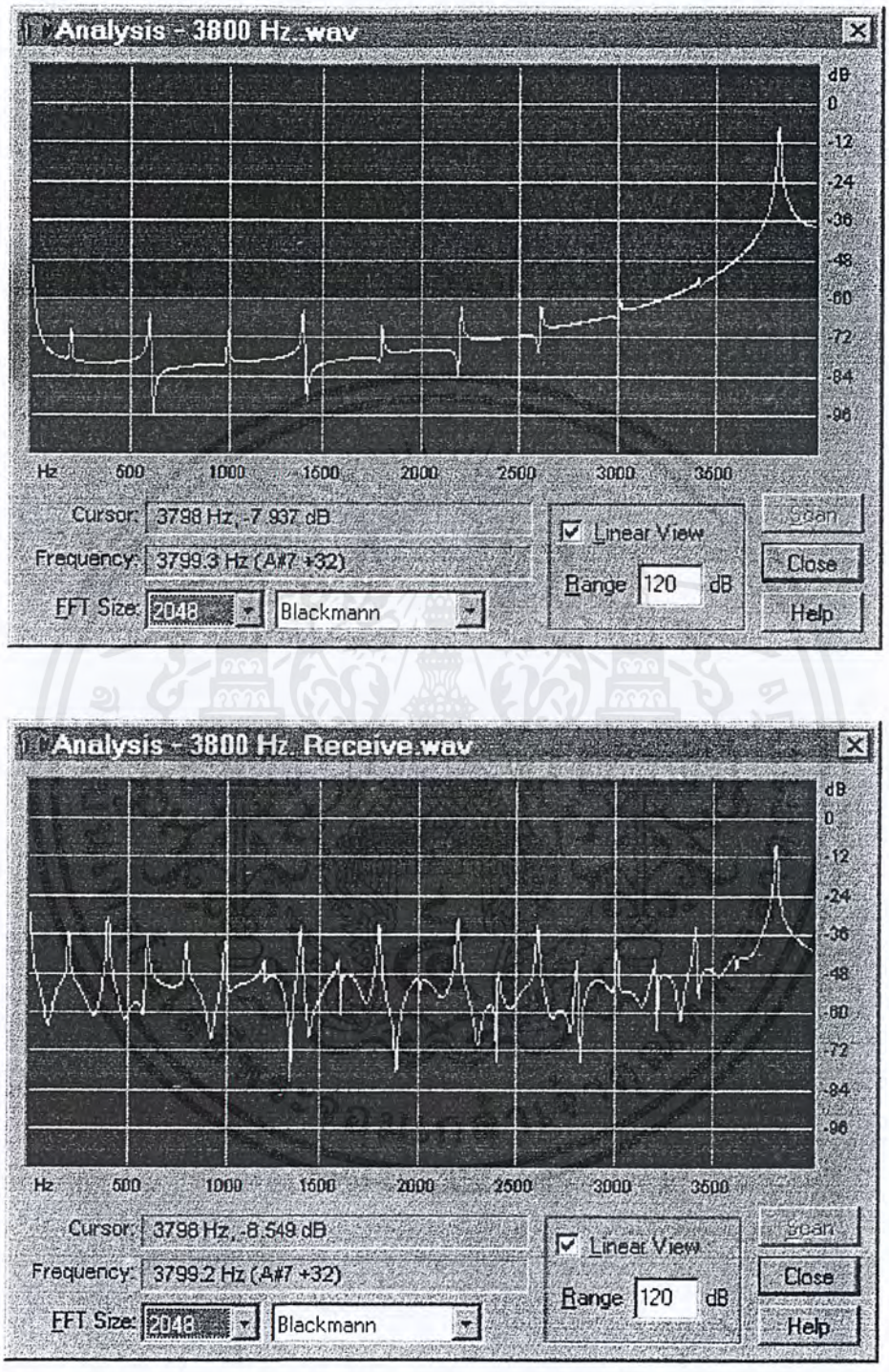
- ความถี่ 3600 เฮิรตซ์




รูปที่ 4.68 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3600 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ 3800 เฮิรตซ์




รูปที่ 4.69 เปรียบเทียบกราฟการวิเคราะห์ทางความถี่ทางด้านส่งและด้านรับของสัญญาณซายน์ความถี่ 3800 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

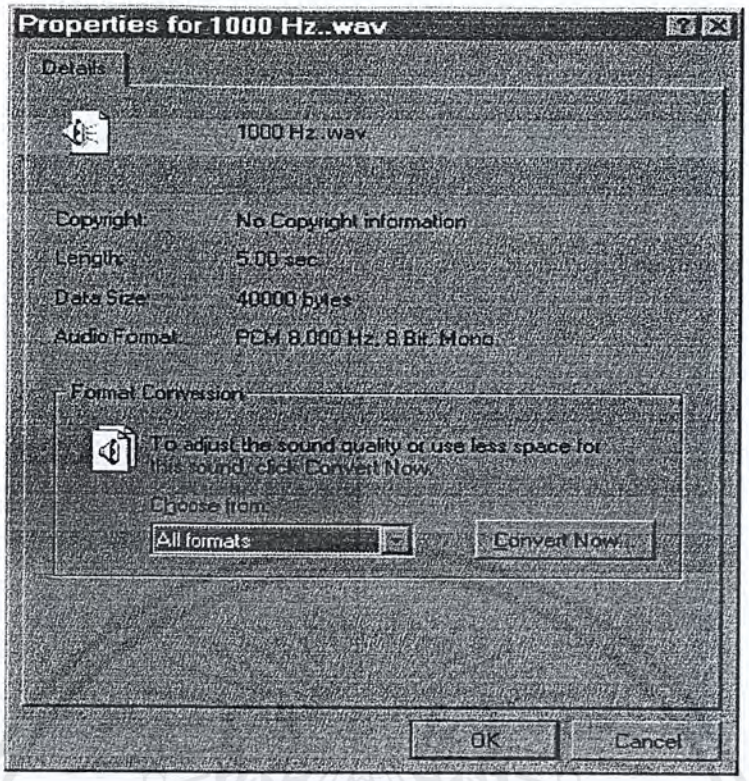
4.2.4 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณที่ด้านรับ

การคำนวณค่าเปอร์เซ็นต์ความผิดพลาด ทำได้โดยใช้สมการที่ 4.1

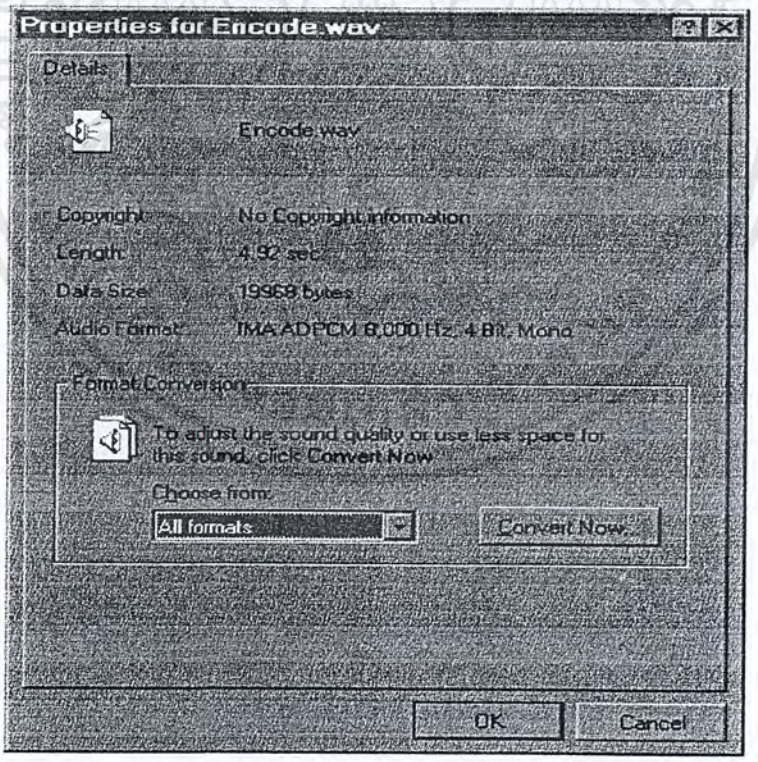
| ความถี่ของไฟล์สัญญาณซายน์ (เฮิรตซ์) | ขนาดข้อมูลของไฟล์ที่ด้านส่ง (ไบต์) | ขนาดข้อมูลของไฟล์เมื่อบีบอัดแล้ว (ไบต์) | ขนาดข้อมูลของไฟล์ที่ด้านรับ (ไบต์) | เปอร์เซ็นต์ความผิดพลาด |
|---|---|---|---|---|
| 200 | 40,000 | 19,968 | 39,986 | 1.46 |
| 400 | 40,000 | 19,968 | 39,986 | 2.42 |
| 600 | 40,000 | 19,968 | 39,986 | 2.77 |
| 800 | 40,000 | 19,968 | 39,986 | 3.61 |
| 1000 | 40,000 | 19,968 | 39,986 | 4.21 |
| 1200 | 40,000 | 19,968 | 39,986 | 4.19 |
| 1400 | 40,000 | 19,968 | 39,986 | 4.37 |
| 1600 | 40,000 | 19,968 | 39,986 | 4.24 |
| 1800 | 40,000 | 19,968 | 39,986 | 4.54 |
| 2000 | 40,000 | 19,968 | 39,986 | 4.70 |
| 2200 | 40,000 | 19,968 | 39,986 | 4.35 |
| 2400 | 40,000 | 19,968 | 39,986 | 2.59 |
| 2600 | 40,000 | 19,968 | 39,986 | 3.83 |
| 2800 | 40,000 | 19,968 | 39,986 | 5.25 |
| 3000 | 40,000 | 19,968 | 39,986 | 6.69 |
| 3200 | 40,000 | 19,968 | 39,986 | 4.84 |
| 3400 | 40,000 | 19,968 | 39,986 | 5.54 |
| 3600 | 40,000 | 19,968 | 39,986 | 4.93 |
| 3800 | 40,000 | 19,968 | 39,986 | 4.59 |

ตารางที่ 4.4 ตารางแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณที่ด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

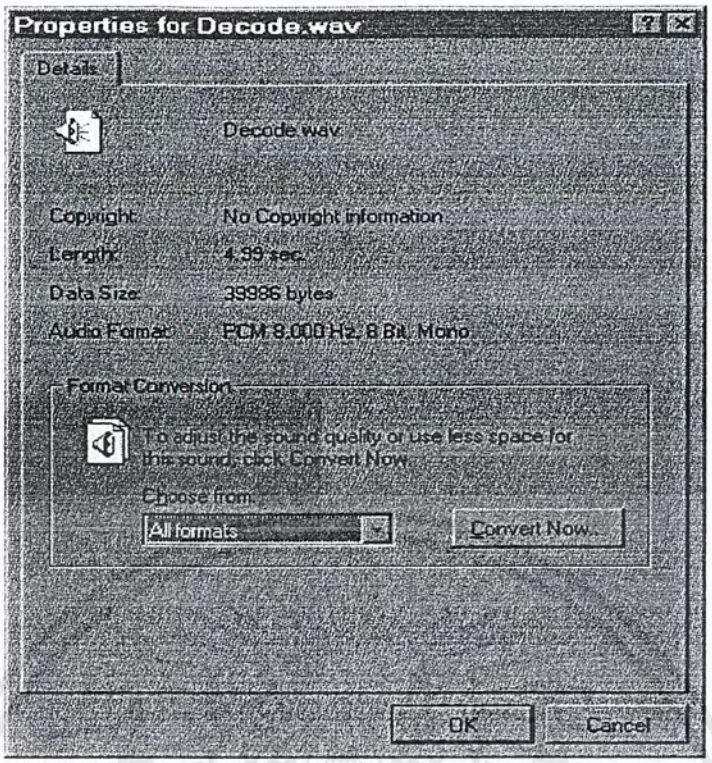



รูปที่ 4.70 แสดงขนาดของไฟล์ที่ด้านส่งก่อนทำการเข้ารหัส

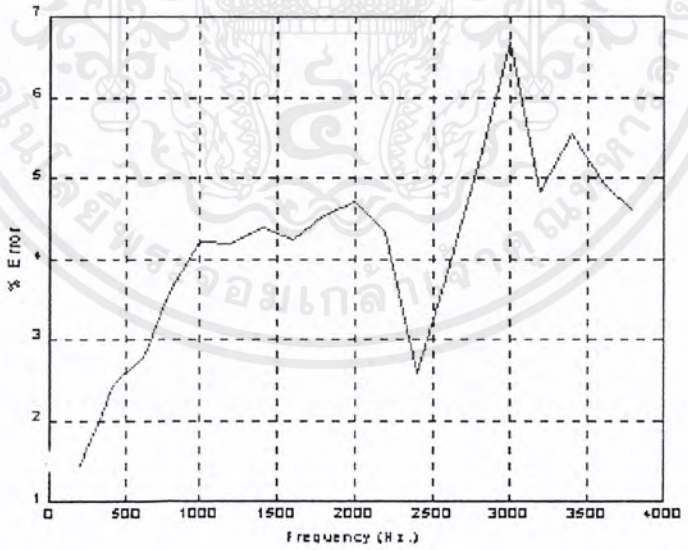



รูปที่ 4.71 แสดงขนาดของไฟล์ที่ด้านส่งหลังจากทำการเข้ารหัสแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.72 แสดงขนาดของไฟล์ที่ด้านรับหลังจากทำการถอดรหัสแล้ว




รูปที่ 4.73 กราฟแสดงค่าเปอร์เซ็นต์ความผิดพลาดของสัญญาณที่ด้านรับ

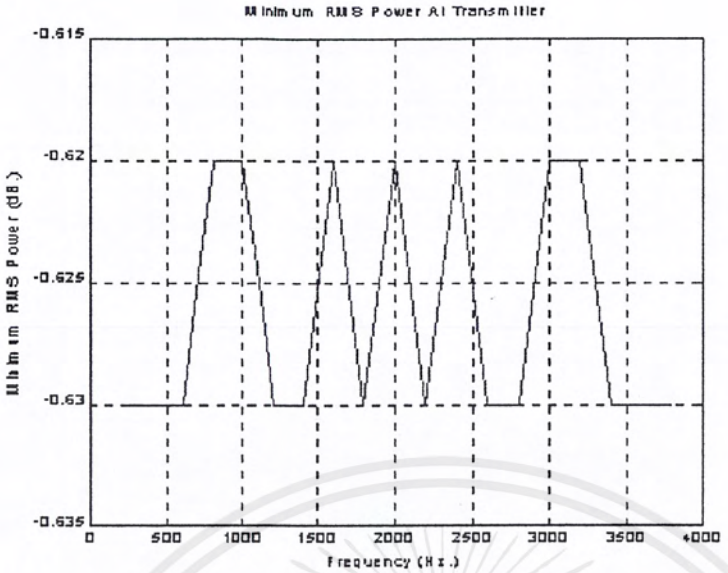
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 ค่า Minimum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

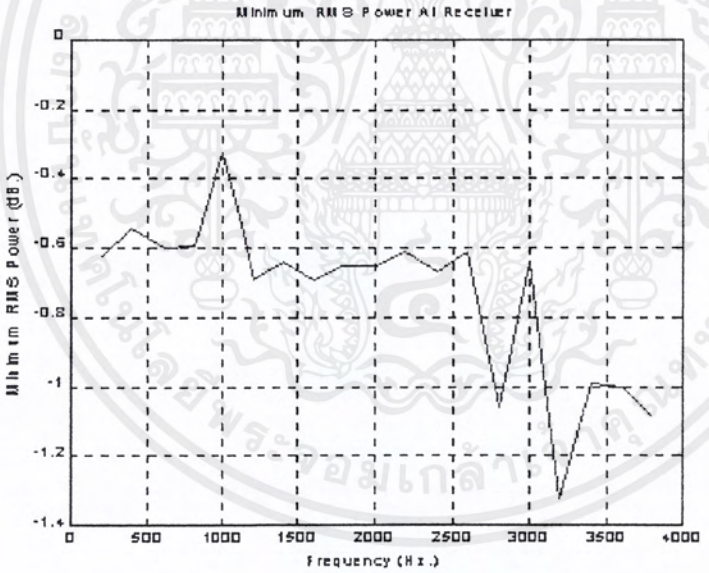
| ความถี่ของไฟล์สัญญาณซายน์ (เฮิรตซ์) | ค่า Minimum RMS Power (เดซิเบล) | |
|---|---|---|
| | ทางด้านส่ง | ทางด้านรับ |
| 200 | -0.63 | -0.62 |
| 400 | -0.63 | -0.54 |
| 600 | -0.63 | -0.60 |
| 800 | -0.62 | -0.59 |
| 1000 | -0.62 | -0.32 |
| 1200 | -0.63 | -0.69 |
| 1400 | -0.63 | -0.64 |
| 1600 | -0.62 | -0.69 |
| 1800 | -0.63 | -0.65 |
| 2000 | -0.62 | -0.65 |
| 2200 | -0.63 | -0.61 |
| 2400 | -0.62 | -0.67 |
| 2600 | -0.63 | -0.61 |
| 2800 | -0.63 | -1.06 |
| 3000 | -0.62 | -0.64 |
| 3200 | -0.62 | -1.32 |
| 3400 | -0.63 | -0.99 |
| 3600 | -0.63 | -1.00 |
| 3800 | -0.63 | -1.09 |

ตารางที่ 4.5 ตารางแสดงค่า Minimum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.74 กราฟแสดงค่า Minimum RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง




รูปที่ 4.75 กราฟแสดงค่า Minimum RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ

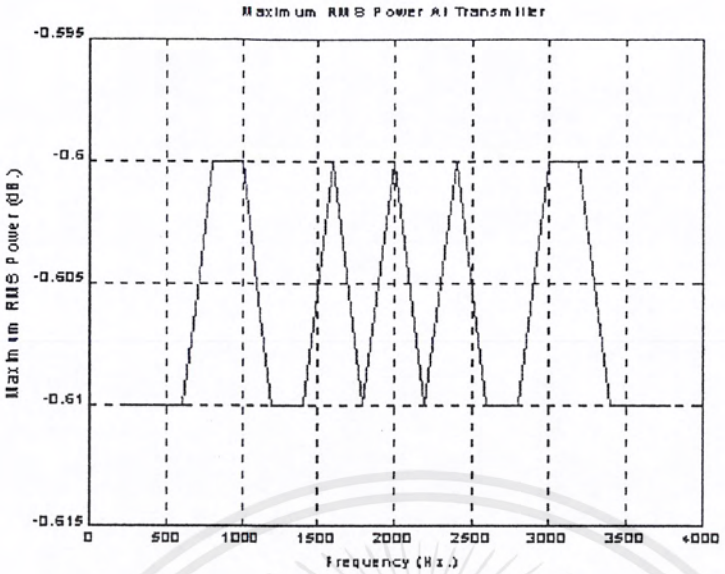
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 ค่า Maximum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

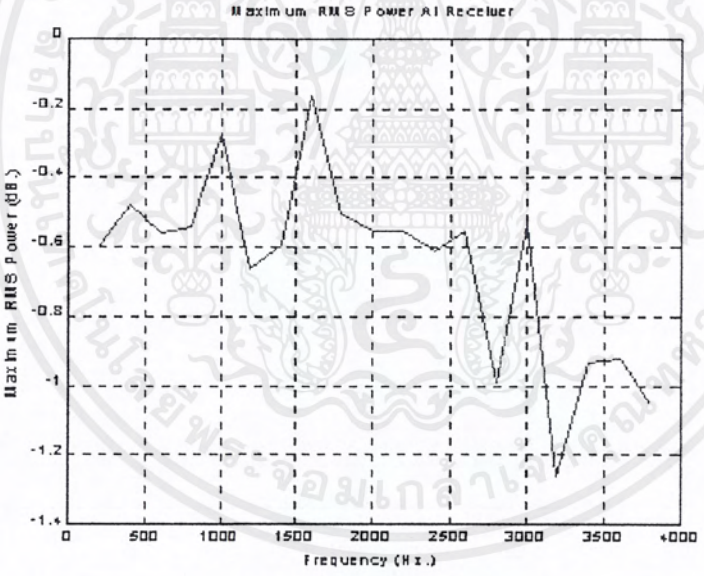
| ความถี่ของไฟล์สัญญาณซายน์ (เฮิรตซ์) | ค่า Maximum RMS Power (เดซิเบล) | |
|---|---|---|
| | ทางด้านส่ง | ทางด้านรับ |
| 200 | -0.61 | -0.59 |
| 400 | -0.61 | -0.48 |
| 600 | -0.61 | -0.56 |
| 800 | -0.60 | -0.54 |
| 1000 | -0.60 | -0.27 |
| 1200 | -0.61 | -0.66 |
| 1400 | -0.61 | -0.59 |
| 1600 | -0.60 | -0.16 |
| 1800 | -0.61 | -0.50 |
| 2000 | -0.60 | -0.55 |
| 2200 | -0.61 | -0.55 |
| 2400 | -0.60 | -0.61 |
| 2600 | -0.61 | -0.55 |
| 2800 | -0.61 | -0.99 |
| 3000 | -0.60 | -0.53 |
| 3200 | -0.60 | -1.26 |
| 3400 | -0.61 | -0.93 |
| 3600 | -0.61 | -0.92 |
| 3800 | -0.61 | -1.05 |

ตารางที่ 4.6 ตารางแสดงค่า Maximum RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.76 กราฟแสดงค่า Maximum RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง




รูปที่ 4.77 กราฟแสดงค่า Maximum RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ

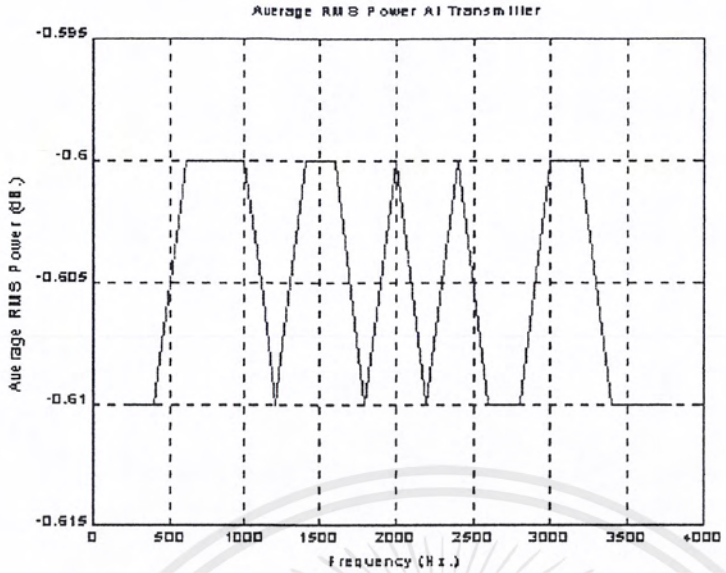
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 ค่า Average RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

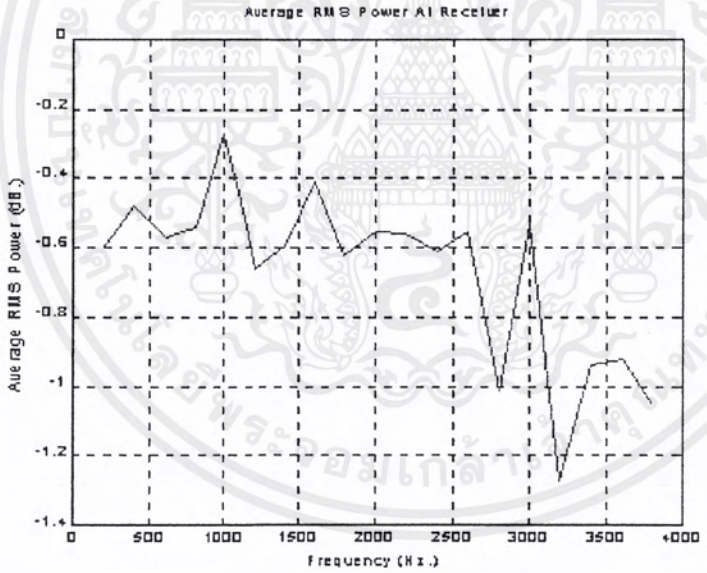
| ความถี่ของไฟล์สัญญาณซายน์ (เฮิรตซ์) | ค่า Average RMS Power (เดซิเบล) | |
|---|---|---|
| | ทางด้านส่ง | ทางด้านรับ |
| 200 | -0.61 | -0.59 |
| 400 | -0.61 | -0.48 |
| 600 | -0.60 | -0.57 |
| 800 | -0.60 | -0.54 |
| 1000 | -0.60 | -0.27 |
| 1200 | -0.61 | -0.66 |
| 1400 | -0.60 | -0.59 |
| 1600 | -0.60 | -0.41 |
| 1800 | -0.61 | -0.62 |
| 2000 | -0.60 | -0.55 |
| 2200 | -0.61 | -0.56 |
| 2400 | -0.60 | -0.61 |
| 2600 | -0.61 | -0.55 |
| 2800 | -0.61 | -1.01 |
| 3000 | -0.60 | -0.53 |
| 3200 | -0.60 | -1.27 |
| 3400 | -0.61 | -0.94 |
| 3600 | -0.61 | -0.92 |
| 3800 | -0.61 | -1.05 |

ตารางที่ 4.7 ตารางแสดงค่า Average RMS Power ของไฟล์สัญญาณทางด้านรับและด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.78 กราฟแสดงค่า Average RMS Power ของไฟล์สัญญาณซายน์ทางด้านส่ง




รูปที่ 4.79 กราฟแสดงค่า Average RMS Power ของไฟล์สัญญาณซายน์ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# บทวิจารณ์และบทสรุป

## 5.1 บทวิจารณ์

จากการทดลองในตอนแรก ซึ่งเป็นการทดลองเพื่อหาค่าเริ่มต้นที่ดีที่สุดสำหรับโปรแกรม เราจะพบว่า เมื่อเราเปลี่ยนค่าเริ่มต้นรูปเวฟฟอร์มที่ได้จะแตกต่างกันออกไป ซึ่งจากการทดลองพบว่าค่าเริ่มต้นที่ดีที่สุดคือ ค่าอินเด็กซ์ เท่ากับ 0 และค่าประมาณของสัญญาณอินพุทตัวก่อน เท่ากับ 0 เราจึงได้นำค่าเริ่มต้นนี้มาทดลองต่อในตอนที่2

การทดลองตอนที่2 ได้ใช้สัญญาณฟังก์ชันซายน์ที่ความถี่ 200 เฮิรตซ์ ถึง 3800 เฮิรตซ์ ซึ่งสามารถครอบคลุมช่วงความถี่ของเสียงของมนุษย์ ซึ่งมีช่วงความถี่ประมาณ 300 เฮิรตซ์ ถึง 3400 เฮิรตซ์ โดยเพิ่มความถี่ช่วงละ 200 เฮิรตซ์ แล้วแสดงรูปเวฟฟอร์มของทางด้านรับและด้านส่งเทียบกัน แล้วทำการหาค่าเปอร์เซ็นต์ความผิดพลาด นำมาพล็อตเป็นกราฟ จากกราฟอาจกล่าวได้ว่าที่ช่วงความถี่ต่ำกว่า 2600 เฮิรตซ์ ค่าของเปอร์เซ็นต์ความผิดพลาดจะมีค่าน้อยกว่าที่ความถี่สูงกว่า 2600 เฮิรตซ์ และเมื่อพิจารณาจากค่าของ RMS Power ก็จะได้ผลที่ตรงกันคือ ค่า Minimum RMS Power , ค่า Maximum RMS Power และ ค่า Average RMS Power ที่ช่วงความถี่สูงกว่า 2600 เฮิรตซ์จะมีค่าสูงกว่าที่ช่วงความถี่ต่ำกว่า 2600 เฮิรตซ์

เพราะฉะนั้น อาจกล่าวได้ว่าถ้าต้องการนำโปรแกรมนี้ไปประยุกต์ใช้กับเสียงของคน ถ้าต้องการให้มีค่าเปอร์เซ็นต์ความผิดพลาดต่ำๆ นั้น ก็อาจต้องใช้เสียงที่มีความถี่ไม่สูงนักหรืออาจจะเป็นเสียงของผู้ชาย เป็นต้น

และในการทดลองยังได้มีการแสดงกราฟของการวิเคราะห์ทางความถี่ไว้ด้วย เพื่อดูองค์ประกอบทางความถี่ จากกราฟแสดงให้เห็นว่าทางด้านรับ แม้อาจจะมีความถี่ต่างๆที่ไม่ใช่ความถี่ที่ส่งมาจากด้านส่งเข้ามาผสมด้วยแต่ความถี่เหล่านั้นก็ยังต่ำกว่าความถี่หลักๆอยู่มากกว่า -15 เดซิเบล

## 5.2 บทสรุป

จากการทดลองนี้ เราสามารถทำการเข้ารหัสข้อมูลแบบเอดีพีซีเอ็มและทำการส่งผ่านเครือข่ายโดยใช้โปรโตคอลทีซีพี/ไอพี แล้วทางด้านรับก็สามารถรับไฟล์จากทางด้านส่งแล้วทำการถอดรหัสแบบเอดีพีซีเอ็ม ซึ่งไฟล์ที่ได้ใกล้เคียงจากต้นฉบับทางด้านส่งมาก แม้ว่าอาจจะมีความผิดพลาดอยู่บ้างแต่ก็อยู่ในระดับที่ยอมรับได้ เพราะข้อมูลเสียงไม่ต้องการความถูกต้อง 100 % และจากการทดลองนี้ทำให้ได้ทราบถึงปัญหาต่างๆ ไม่ว่าจะเป็นทั้งทางด้านซอฟท์แวร์หรือฮาร์ดแวร์

ปัญหาที่เกิดขึ้นทางด้านซอฟท์แวร์ ได้แก่

1. ในการเขียนโปรแกรมได้ใช้โปรแกรมเดลไฟในการเขียนโปรแกรม ซึ่งต้องใช้เครื่องคอมพิวเตอร์ที่มีหน่วยความจำ(RAM) ที่มากพอสมควร เพราะโปรแกรมเดลไฟจะกินหน่วยความจำของเครื่องคอมพิวเตอร์ไปเรื่อยๆ ทำให้คอมพิวเตอร์ช้าลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมเดลไฟ มีพื้นฐานมาจากโปรแกรมปาสคาล ซึ่งถ้าในตัวโปรแกรมมีการวนลูป (loop) เป็นจำนวนมาก อาจเกิดการโอเวอร์โฟลว์(overflow) ของข้อมูลขึ้นได้ ทำให้เกิดความผิดพลาด
3. ในการทดลองไม่สามารถหาโปรแกรมสำเร็จรูปใดที่มีความสามารถในการแสดงเวฟฟอร์มของไฟล์ที่ทำการเข้ารหัสแบบเอดีพีซีเอ็มได้ ไม่ว่าจะเป็น MATLAB , Cool Edit , Sound Recorder ก็ไม่สามารถแสดงได้ ในรายงานนี้จึงไม่สามารถแสดงให้เห็นได้

ในส่วนของฮาร์ดแวร์ก็พบปัญหาต่างๆ ดังนี้

1. ต้องใช้อุปกรณ์ในการทดสอบโปรแกรมค่อนข้างมาก เพราะเครื่องคอมพิวเตอร์ที่นำมาทดสอบโปรแกรมนี้จำเป็นต้องมีการ์ดเสียง และต้องมีไมโครโฟนเพื่อใช้ในการอัดเสียง และต้องมีลำโพงเพื่อใช้ในการเล่นไฟล์เสียง
2. ถ้าต้องการทดสอบส่งไฟล์ผ่านโปรโตคอลทีซีพี/ไอพี ถึงแม้จะสามารถทำการทดสอบโดยการส่งไฟล์วนลูปกลับมาเข้าเครื่องตัวเองได้ก็ตาม แต่ถ้าต้องการส่งระหว่างคอมพิวเตอร์ก็ต้องใช้คอมพิวเตอร์ 2 ตัว และทั้ง 2 ตัวต้องมีไอพีแอดเดรส และถ้าคอมพิวเตอร์ทั้ง 2 ตัวอยู่ภายในเครือข่ายเดียวกันแล้ว คอมพิวเตอร์ทั้ง 2 ตัว ต้องมองเห็นกัน (ทดสอบได้โดยใช้คำสั่ง ping) จึงจะสามารถทำการส่งได้ แต่ถ้าเป็นระหว่างเครือข่ายทั้ง 2 ตัวต้องสามารถใช้งานอินเตอร์เน็ตได้ โดยอาจจะใช้โมเด็มหรือการ์ดแลน (LAN) ก็ได้

ในส่วนของแนวทางในการพัฒนาโปรแกรมนี้ให้มีความสมบูรณ์หรือให้สามรถใช้งานได้หลากหลายมากขึ้นอาจจะทำได้โดย

1. พัฒนาในส่วนของการเข้ารหัสและถอดรหัสแบบเอดีพีซีเอ็มให้ใช้งานได้กับไฟล์รูปแบบต่างๆมากขึ้น โดยอาจพัฒนาให้สามารถเข้ารหัสไฟล์ที่เป็นสเตอริโอได้ หรืออาจจะสามารถเข้ารหัสไฟล์ที่มีอัตราการแซมปลิ้ง ต่างๆนอกจาก 8 kHz.
2. พัฒนาในส่วนของอัตราส่วนของการบีบอัดข้อมูล ให้สามรถบีบอัดข้อมูลได้มากขึ้น โดยอาจจะพัฒนาอัตราส่วนของการบีบอัดให้เป็น 4:1
3. อาจเปลี่ยนวิธีในการเข้ารหัสและถอดรหัสข้อมูลให้มีประสิทธิภาพมากขึ้น โดยอาจใช้วิธี CELP ซึ่งสามารถบีบอัดข้อมูลได้เล็กกว่าวิธีเอดีพีซีเอ็ม เป็นต้น
4. ในส่วนของการส่งข้อมูลผ่านเครือข่าย อาจจะใช้โปรโตคอลยูดีพีแทนโปรโตคอลทีซีพี ซึ่งจะทำให้สามารถส่งข้อมูลได้เร็วขึ้น แต่ความถูกต้องของข้อมูลอาจจะไม่ดีเท่าใช้โปรโตคอลทีซีพี
5. ในส่วนของโปรแกรมอาจพัฒนาให้มีการใช้งานได้ดีขึ้น โดยอาจจะเพิ่มความสามารถในการส่งให้เป็นแบบฟูลดูเพล็กซ์

เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Source code ของส่วน Main program**

```
program NetVoice;

uses
  Forms,
  NetVoiceP in 'NetVoiceP.pas' {NetVoiceF},
  NetVoiceDecodeLib in 'NetVoiceDecodeLib.pas',
  NetVoiceEncodeLib in 'NetVoiceEncodeLib.pas',
  NetVoicePlayP in 'NetVoicePlayP.pas' {NetVoicePlayF};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TNetVoiceF, NetVoiceF);
  Application.CreateForm(TNetVoicePlayF, NetVoicePlayF);
  Application.Run;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Source code ของโปรแกรมในส่วนของ unit NetVoiceP**

```
unit NetVoiceP;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, ExtCtrls, Buttons, Psock, NMSTRM, ComCtrls, MPlayer, Spin,
  ScktComp, MMSystem;

const
  Recordwave    = 'Recordwave.wav';
  EncodeRevwave = 'EncodeReceive.wav'; // Receive Encdoe Wave From Client

type
  TNetVoiceF = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    ReceiveIP: TMaskEdit;
    Panel2: TPanel;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    TimeLength: TSpinEdit;
    pgBar: TProgressBar;
    MediaPlayer1: TMediaPlayer;
    StatusBar1: TStatusBar;
    NMStrm1: TNMStrm;
    NMStrmServ1: TNMStrmServ;
    Timer1: TTimer;
    SaveDialog1: TSaveDialog;
    StartBtn: TSpeedButton;
    ReSendBtn: TSpeedButton;
    StopBtn: TSpeedButton;
    OpenWaveBtn: TSpeedButton;
    SaveAsWaveBtn: TSpeedButton;
    GroupBox6: TGroupBox;
    Panel6: TPanel;
    Panel7: TPanel;
    rbMono: TRadioButton;
    rbStereo: TRadioButton;
    Panel12: TPanel;
    Panel17: TPanel;
    rb11: TRadioButton;
    rb22: TRadioButton;
    rb44: TRadioButton;
    rb8k: TRadioButton;
    Panel18: TPanel;
    rb8: TRadioButton;
    rb16: TRadioButton;
    OpenDialog1: TOpenDialog;
    Label3: TLabel;
    WaveFile: TLabel;
    EncodeSendBtn: TSpeedButton;
    NotReceiveCheck: TCheckBox;
    Timer2: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure NMStrmServ1ClientContact(Sender: TObject);
    procedure NMStrmServ1MSG(Sender: TComponent; const sFrom: String; strm:
TStream);
    procedure NMStrmServ1Status(Sender: TComponent; Status: String);
    procedure NMStrm1Connect(Sender: TObject);
    procedure NMStrm1ConnectionFailed(Sender: TObject);
    procedure NMStrm1Disconnect(Sender: TObject);
    procedure NMStrm1HostResolved(Sender: TComponent);
    procedure NMStrm1InvalidHost(var Handled: Boolean);
    procedure NMStrm1MessageSent(Sender: TObject);
    procedure NMStrm1PacketSent(Sender: TObject);
    procedure NMStrm1Status(Sender: TComponent; Status: String);
    procedure OpenWaveBtnClick(Sender: TObject);
    procedure SaveAsWaveBtnClick(Sender: TObject);
    procedure StartBtnClick(Sender: TObject);
    procedure StopBtnClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure ReSendBtnClick(Sender: TObject);
    procedure EncodeSendBtnClick(Sender: TObject);
    procedure TimeLengthChange(Sender: TObject);
    procedure NMStrmServ1Disconnect(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure NotReceiveCheckClick(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
    FDeviceID: Word;
    ReceiveWave : boolean;
  public
    { Public declarations }
    procedure OpenMedia;
    procedure RecordMedia;
    procedure StopMedia;
    procedure SaveMedia;
    procedure CloseMedia;
    procedure SetWaveParameters;
    procedure SendEncodeWave;
  end;

var
  NetVoiceF: TNetVoiceF;

implementation

uses NetVoiceEncodeLib, NetVoicePlayP, NetVoiceDecodeLib;

{$R *.DFM}

var
  MyError, Flags: Longint;

procedure TNetVoiceF.CloseMedia;
var
  MyGenParms: TMCI_Generic_Parms;
begin
  if FDeviceID <> 0 then begin
    Flags:=0;
    MyGenParms.dwCallback:=Handle; // TNetVoiceF.Handle
    MyError:=mciSendCommand(FDeviceID, mci_Close, Flags, Longint(@MyGenParms));
    if MyError = 0 then
      FDeviceID:=0;
  end;
end;

procedure TNetVoiceF.OpenMedia;
var
  MyOpenParms: TMCI_Open_Parms;
begin
  Flags:=mci_Wait or mci_Open_Element or mci_Open_Type;
  with MyOpenParms do begin
    dwCallback:=Handle; // TNetVoiceF.Handle
    lpstrDeviceType:=PChar('WaveAudio');
    lpstrElementName:=PChar('');
  end;
  MyError:=mciSendCommand(0, mci_Open, Flags, Longint(@MyOpenParms));
  if MyError = 0 then
    FDeviceID:=MyOpenParms.wDeviceID;
end;

procedure TNetVoiceF.RecordMedia;
var
  MyRecordParms: TMCI_Record_Parms;
begin
  Flags:=mci_Notify;
  with MyRecordParms do begin
    dwCallback:=Handle;  // TNetVoiceF.Handle
    dwFrom:=0;
    dwTo:=10000;
  end;
  MyError:=mciSendCommand(FDeviceID, mci_Record, Flags,
Longint(@MyRecordParms));
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TNetVoiceF.SaveMedia;
  type    // not implemented by Delphi
    PMCI_Save_Parms = ^TMCI_Save_Parms;
    TMCI_Save_Parms = record
    dwCallback: DWord;
    lpstrFileName: PAnsiChar;  // name of file to save
  end;
var
  MySaveParms: TMCI_Save_Parms;
begin
  if FDeviceID <> 0 then begin
      // save the file...
    Flags:=mci_Save_File or mci_Wait;
    with MySaveParms do
    begin
      dwCallback:=Handle;
      lpstrFileName:= Recordwave;
      WaveFile.Caption := Recordwave;
    end;
    MyError:=mciSendCommand(FDeviceID, mci_Save, Flags, Longint(@MySaveParms));
  end;
end;

procedure TNetVoiceF.SetWaveParameters;
const
  cMONO     = 1;
  cSTEREO   = 2;
  sRATE8    = 8000;  // Supported by PCM wave format (as long as I've known)
  sRATE11   = 11025;
  sRATE22   = 22050;
  sRATE44   = 44100;
  bBIT8     = 8;
  bBIT16    = 16;
var
  MyWaveParms : TMCI_Wave_Set_Parms;
  SampPerSec     : DWORD;
  Channels       : WORD;
  BitsSample     : WORD;
begin
  if rbMono.checked then Channels := cMONO else Channels := cSTEREO;
  if rb8k.checked then SampPerSec := sRATE8;
  if rb11.checked then SampPerSec := sRATE11;
  if rb22.checked then SampPerSec := sRATE22;
  if rb44.checked then SampPerSec := sRATE44;
  if rb8.checked then BitsSample := bBIT8 else BitsSample := bBIT16;
  if FDeviceId <> 0 then begin // Channel must be already open
    Flags := MCI_WAVE_SET_ANYINPUT or
    MCI_WAVE_SET_ANYOUTPUT or
    MCI_WAVE_SET_BITSPERSAMPLE or
    MCI_WAVE_SET_AVGBYTESPERSEC or
    MCI_WAVE_SET_BLOCKALIGN or
    MCI_WAVE_SET_SAMPLESPERSEC or
    MCI_WAVE_SET_CHANNELS or
    MCI_WAVE_SET_FORMATTAG;
    with MyWaveParms do begin
      wFormatTag := WAVE_FORMAT_PCM;
      dwCallback := Handle;  // TNetVoiceF.Handle
      nChannels   := Channels;
      nSamplesPerSec := SampPerSec;
      nAvgBytesPerSec := (BitsSample * Channels) div 8 * SampPerSec;
      wBitsPerSample := BitsSample;
      nBlockAlign := (BitsSample * Channels) div 8;
    end;
    MyError := mciSendCommand(fDeviceId, MCI_SET, Flags, longint(@MyWaveParms));
  end;
end;

procedure TNetVoiceF.StopMedia;
var
  MyGenParms: TMCI_Generic_Parms;
begin
  if FDeviceID <> 0 then begin
    Flags:=mci_Wait;
    MyGenParms.dwCallback:=Handle;  // TNetVoiceF.Handle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MyError:=mciSendCommand(FDeviceID, mci_Stop, Flags, Longint(@MyGenParms));
  end;
end;

procedure TNetVoiceF.Timer1Timer(Sender: TObject);
var EncodeResult : string;
begin
  with pgBar do
  begin
    if (position >= max) then
    begin
      StopBtnClick(self);
      EncodeSendBtnClick(self);
      Exit;
    end;
    position := position + 1;
  end;
end;

procedure TNetVoiceF.OpenWaveBtnClick(Sender: TObject);
begin
  if OpenDialog1.Execute then
    if OpenDialog1.FileName <> '' then
    begin
      WaveFile.Caption := OpenDialog1.FileName;
      MediaPlayer1.Close;
      MediaPlayer1.FileName := WaveFile.Caption;
      MediaPlayer1.Open;

      EncodeSendBtn.Enabled := True;
      ReSendBtn.Enabled := True;
    end;
end;

procedure TNetVoiceF.SaveAsWaveBtnClick(Sender: TObject);
begin
  if WaveFile.Caption <> '' then
    if SaveDialog1.Execute then
      if SaveDialog1.FileName <> '' then
        if
Not(CopyFile(PChar(WaveFile.Caption),PChar(SaveDialog1.FileName),True)) then
          Showmessage('Cannot save file');
end;

procedure TNetVoiceF.StartBtnClick(Sender: TObject);
begin
  OpenMedia;
  SetWaveParameters;
  RecordMedia;
  Timer1.Enabled := True;
  StartBtn.Enabled := False;
  EncodeSendBtn.Enabled := False;
  ReSendBtn.Enabled := False;
  StopBtn.Enabled := True;
end;

procedure TNetVoiceF.StopBtnClick(Sender: TObject);
begin
  StopMedia;
  SaveMedia;
  CloseMedia;

  StartBtn.Enabled := True;
  EncodeSendBtn.Enabled := True;
  ReSendBtn.Enabled := True;
  StopBtn.Enabled := False;

  Timer1.Enabled:= False;
  pgBar.Position := 0;
  with MediaPlayer1 do
  begin
    FileName := Recordwave;
    Enabled := true;
    Open;
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure TNetVoiceF.SendEncodeWave;
var
  MyFStream: TFileStream;
  TmpStr: String;
begin
  if ReceiveIP.Text <> '' then
  begin
    NMStrm1.Host := ReceiveIP.Text;
    NMStrm1.FromName := 'Client';
    MyFStream := TFileStream.Create(EncodewaveOut, fmOpenRead);
    try
      NMStrm1.PostIt(MyFStream);
    finally
      MyFStream.Free;
    end;
  end;
end;

procedure TNetVoiceF.EncodeSendBtnClick(Sender: TObject);
var EncodeResult : string;
begin
  if WaveFile.Caption <> '' then
  begin
    StartBtn.Enabled := False;
    EncodeSendBtn.Enabled := False;
    ReSendBtn.Enabled := False;
    StopBtn.Enabled := False;

    MediaPlayer1.Close;
    Screen.Cursor := crHourGlass;

    if not(EncodeWave(WaveFile.Caption, EncodeResult)) then
      ShowMessage(EncodeResult)
    else
      SendEncodeWave;

    MediaPlayer1.Open;
    Screen.Cursor := crDefault;

    StartBtn.Enabled := True;
    EncodeSendBtn.Enabled := True;
    ReSendBtn.Enabled := True;
  end;
end;

procedure TNetVoiceF.ReSendBtnClick(Sender: TObject);
begin
  StartBtn.Enabled := False;
  EncodeSendBtn.Enabled := False;
  ReSendBtn.Enabled := False;
  StopBtn.Enabled := False;

  Screen.Cursor := crHourGlass;
  SendEncodeWave;
  Screen.Cursor := crDefault;

  StartBtn.Enabled := True;
  EncodeSendBtn.Enabled := True;
  ReSendBtn.Enabled := True;
end;

procedure TNetVoiceF.NMStrm1Connect(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Connected';
end;

procedure TNetVoiceF.NMStrm1ConnectionFailed(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Connection Failed';
end;

procedure TNetVoiceF.NMStrm1Disconnect(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  StatusBar1.SimpleText := 'Disconnected';
end;

procedure TNetVoiceF.NMStrm1HostResolved(Sender: TComponent);
begin
  StatusBar1.SimpleText := 'Host Resolved';
end;

procedure TNetVoiceF.NMStrm1InvalidHost(var Handled: Boolean);
var
  TmpStr: String;
begin
  If InputQuery('Invalid Host!', 'Specify a new host:', TmpStr) then
  Begin
    NMStrm1.Host := TmpStr;
    Handled := TRUE;
  End;
end;

procedure TNetVoiceF.NMStrm1MessageSent(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Send complete';
end;

procedure TNetVoiceF.NMStrm1PacketSent(Sender: TObject);
begin
  StatusBar1.SimpleText := IntToStr(NMStrm1.BytesSent)+' of '+IntToStr
(NMStrm1.BytesTotal)+' send';
end;

procedure TNetVoiceF.NMStrm1Status(Sender: TComponent; Status: String);
begin
  If StatusBar1 <> nil then
    StatusBar1.SimpleText := status;
end;

procedure TNetVoiceF.TimeLengthChange(Sender: TObject);
begin
  pgBar.Max := TimeLength.Value;
end;

procedure TNetVoiceF.NMStrmServ1ClientContact(Sender: TObject);
begin
  NMStrmServ1.ReportLevel := Status_Basic;
  NMStrmServ1.TimeOut := 90000;
  StatusBar1.SimpleText := 'Client connected';
  ReceiveWave := False;
end;

procedure TNetVoiceF.NMStrmServ1MSG(Sender: TComponent; const sFrom: String;
strm: TStream);
var MyFStream: TFileStream;
begin
  If FileExists(EncodeRevwave) then DeleteFile(EncodeRevwave);
  MyFStream := TFileStream.Create(EncodeRevwave, fmCreate);
  try
    MyFStream.CopyFrom(strm, strm.size);
    ReceiveWave := True;
    Timer2.Enabled := True;
  finally
    MYFStream.Free;
  end;
end;

procedure TNetVoiceF.NMStrmServ1Status(Sender: TComponent; Status: String);
begin
  If StatusBar1 <> nil then
    StatusBar1.SimpleText := status;
end;

procedure TNetVoiceF.NMStrmServ1Disconnect(Sender: TObject);
begin
  StatusBar1.SimpleText := 'Client Disconnected';
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TNetVoiceF.FormCreate(Sender: TObject);
begin
  StartBtn.Enabled := True;
  EncodeSendBtn.Enabled := False;
  ReSendBtn.Enabled := False;
  StopBtn.Enabled := False;
end;

procedure TNetVoiceF.NotReceiveCheckClick(Sender: TObject);
begin
  if NotReceiveCheck.Checked then
    NMStrmServ1.Port := 6712
  else
    NMStrmServ1.Port := 6711;
end;

procedure TNetVoiceF.Timer2Timer(Sender: TObject);
var DecodeResult : string;
begin
  Timer2.Enabled := False;
  if ReceiveWave then
  begin
    Screen.Cursor := crHourGlass;

    if not(DecodeWave(EncodeRevwave, DecodeResult)) then
      ShowMessage(DecodeResult)
    else
    begin
      Screen.Cursor := crDefault;
      NetVoicePlayF.OpenWave;
      NetVoicePlayF.Show;
    end;

    Screen.Cursor := crDefault;
  end;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Source code ของโปรแกรมในส่วนของ unit NetVoiceEncodeLib**

```
unit NetVoiceEncodeLib;

interface

uses Windows, Sysutils;

const
  EncodeInDoc     = 'EncodeIn.doc';
  EncodeOutDoc    = 'EncodeOut.doc';
  EncodewaveOut   = 'Encode.wav';

var
  input_wavp,output_wav_codep:file of byte;
  input_docp,output_doc_codep:textfile;
  input_wav_filename,input_DOC_filename:string;
  output_wav_code_filename,output_DOC_code_filename:string;
  samp,ss,value,index,valpred:integer;
  level:byte;
  code,code1,code2:integer;
  data_byte:integer;
  map_table:array[0..255] of integer;
  StepSizeTable:array[0..49] of integer;
  ETotalWaveByte,ETotalWaveByteData : integer;
  ETotalADPCMByte,ETotalADPCMByteData : integer;

  function EncodeWave(wavefilename : string; var resultstr : string): boolean;

implementation


function check_wav_file() : boolean;
var ch:byte;
begin
  reset(input_wavp);
  seek(input_wavp,8);
  read(input_wavp,ch);
  if char(ch) = 'W' then
  begin
    read(input_wavp,ch);
    if char(ch) = 'A' then
    begin
      read(input_wavp,ch);
      if char(ch) = 'V' then
      begin
        read(input_wavp,ch);
        if char(ch) = 'E' then
        begin
          result := true;
        end
        else begin
          closefile(input_wavp);
          result := false;
        end;
      end
      else begin
        closefile(input_wavp);
        result := false;
      end;
    end
    else begin
      closefile(input_wavp);
      result := false;
    end;
  end {first if}
  else begin
    closefile(input_wavp);
    result := false;
  end;
end;

function check_type_of_record():boolean;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var ch:byte;
begin
  reset(input_wavp);
  seek(input_wavp,20);
  read(input_wavp,ch);
  case ch of
    1 : result := true;
    else begin
      result := false;
      closefile(input_wavp);
    end;
  end;
end;

function check_channel():integer;
var ch:byte;
begin
  reset(input_wavp);
  seek(input_wavp,22);
  read(input_wavp,ch);
  case ch of
    1 : result := 1;
    2 : begin
          result := 2;
          closefile(input_wavp);
        end;
    else begin
           result := 3;
           closefile(input_wavp);
    end;
  end;  {case}
end;

function check_sampling_rate():integer;
var ch:byte;
begin
  reset(input_wavp);
  seek(input_wavp,24);
  read(input_wavp,ch);
  case ch of
    64 : result := 64;
    17 : begin
           result := 17;
           closefile(input_wavp);
         end;
    34 : begin
           result := 34;
           closefile(input_wavp);
         end;
    68 : begin
           result := 68;
           closefile(input_wavp);
         end;
  end;
end;

function check_byte_per_sample():boolean;
var ch:byte;
begin
  reset(input_wavp);
  seek(input_wavp,32);
  read(input_wavp,ch);
  if ch = 1 then
    result := true
  else begin
    result := false;
    closefile(input_wavp);
  end;
end;

function check_bit_per_sample():boolean;
var ch : byte;
begin
  reset(input_wavp);
  seek(input_wavp,34);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  read(input_wavp,ch);
  if ch = 8 then
    result := true
  else begin
    result := false;
    closefile(input_wavp);
  end;
end;

procedure copy_all_to_DOC;
var ch:byte;
    i:string;
begin
  reset(input_wavp);
  assignfile(input_docp,EncodeInDoc);
  rewrite(input_docp);
  closefile(input_docp);
  assignfile(input_docp,EncodeInDoc);
  append(input_docp);
  while not eof(input_wavp) do
  begin
    read(input_wavp,ch);
    i := inttostr(ch)+' ';
    write(input_docp,i);
  end;
  closefile(input_docp);

  ETotalWaveByte := filesize(input_wavp);
  ETotalWaveByteData := filesize(input_wavp)-60;
  data_byte := filesize(input_wavp)-58;
end;

procedure open_output_wav_code_file;
begin
  assignfile(output_wav_codep,EncodewaveOut);
  rewrite(output_wav_codep);
  closefile(output_wav_codep);
end;

procedure write_header_to_code_file;
var i:integer;
    RIFF_size,data_size,fact_size:integer;
    size1,size2,size3,size4,data1,data2,data3,data4,header:integer;
    ch:byte;
begin
  i:=0;
  header := 0;
  reset(output_wav_codep);
  while i<=59 do
  begin
    write(output_wav_codep,header);
    i:=i+1;
  end; {while}

  seek(output_wav_codep,0);
  ch := ord('R');
  write(output_wav_codep,ch);
  ch := ord('I');
  write(output_wav_codep,ch);
  ch := ord('F');
  write(output_wav_codep,ch);
  ch := ord('F');
  write(output_wav_codep,ch);

  seek(output_wav_codep,filepos(output_wav_codep)+4);
  ch := ord('W');
  write(output_wav_codep,ch);
  ch := ord('A');
  write(output_wav_codep,ch);
  ch := ord('V');
  write(output_wav_codep,ch);
  ch := ord('E');
  write(output_wav_codep,ch);

  ch := ord('f');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write(output_wav_codep,ch);
ch := ord('m');
write(output_wav_codep,ch);
ch := ord('t');
write(output_wav_codep,ch);
ch := ord(' ');
write(output_wav_codep,ch);
ch := 20;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+3);
ch := 17;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+1);
ch := 1;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+1);
ch := 64;
write(output_wav_codep,ch);
ch := 31;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+2);
ch := 215;
write(output_wav_codep,ch);
ch := 15;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+3);
ch := 1;
write(output_wav_codep,ch);
ch := 4;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+1);
ch := 2;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+1);
ch := 249;
write(output_wav_codep,ch);
ch := 1;
write(output_wav_codep,ch);
ch := ord('f');
write(output_wav_codep,ch);
ch := ord('a');
write(output_wav_codep,ch);
ch := ord('c');
write(output_wav_codep,ch);
ch := ord('t');
write(output_wav_codep,ch);
ch := 4;
write(output_wav_codep,ch);

seek(output_wav_codep,filepos(output_wav_codep)+7);
ch := ord('d');
write(output_wav_codep,ch);
ch := ord('a');
write(output_wav_codep,ch);
ch := ord('t');
write(output_wav_codep,ch);
ch := ord('a');
write(output_wav_codep,ch);

if (data_byte mod 2) = 0 then
begin
  RIFF_size := (data_byte div 2)+52;
  size1 := RIFF_size shr 24;
  size2 := (RIFF_size shr 16)and($FF);
  size3 := (RIFF_size shr 8)and($FF);
  size4 := (RIFF_size and $FF);
  seek(output_wav_codep,4);
  ch := size4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    write(output_wav_codep,ch);
    ch := size3;
    write(output_wav_codep,ch);
    ch := size2;
    write(output_wav_codep,ch);
    ch := size1;
    write(output_wav_codep,ch);
  end;

  if (data_byte mod 2) = 1 then
  begin
    RIFF_size := ((data_byte+1) div 2)+52;
    size1 := RIFF_size shr 24;
    size2 := (RIFF_size shr 16) and $FF;
    size3 := (RIFF_size shr 8) and $FF;
    size4 := (RIFF_size and $FF);
    seek(output_wav_codep,4);
    ch := size4;
    write(output_wav_codep,ch);
    ch := size3;
    write(output_wav_codep,ch);
    ch := size2;
    write(output_wav_codep,ch);
    ch := size1;
    write(output_wav_codep,ch);
  end;

  if (data_byte mod 2)=0 then //To calculate data chunk size when data_byte is
even
  begin
    data_size := data_byte div 2;
    data1 := data_size shr 24;
    data2 := (data_size shr 16) and $FF;
    data3 := (data_size shr 8) and $FF;
    data4 := (data_size and $FF);
    seek(output_wav_codep,56);
    ch := data4;
    write(output_wav_codep,ch);
    ch := data3;
    write(output_wav_codep,ch);
    ch := data2;
    write(output_wav_codep,ch);
    ch := data1;
    write(output_wav_codep,ch);
    fact_size := data_byte;
    data1 := fact_size shr 24;
    data2 := (fact_size shr 16) and $FF;
    data3 := (fact_size shr 8) and $FF;
    data4 := (fact_size and $FF);
    seek(output_wav_codep,48);
    ch := data4;
    write(output_wav_codep,ch);
    ch := data3;
    write(output_wav_codep,ch);
    ch := data2;
    write(output_wav_codep,ch);
    ch := data1;
    write(output_wav_codep,ch);
  end;

  if (data_byte mod 2)=1 then
  begin
    data_size := (data_byte+1) div 2;
    data1 := data_size shr 24;
    data2 := (data_size shr 16) and $FF;
    data3 := (data_size shr 8) and $FF;
    data4 := (data_size and $FF);
    seek(output_wav_codep,56);
    ch := data4;
    write(output_wav_codep,ch);
    ch := data3;
    write(output_wav_codep,ch);
    ch := data2;
    write(output_wav_codep,ch);
    ch := data1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    write(output_wav_codep,ch);
    fact_size := data_byte;
    data1 := fact_size shr 24;
    data2 := (fact_size shr 16) and $FF;
    data3 := (fact_size shr 8) and $FF;
    data4 := (fact_size and $FF);
    seek(output_wav_codep,48);
    ch := data4;
    write(output_wav_codep,ch);
    ch := data3;
    write(output_wav_codep,ch);
    ch := data2;
    write(output_wav_codep,ch);
    ch := data1;
    write(output_wav_codep,ch);
  end;
  closefile(output_wav_codep);
end;

procedure adpcm_encode(samp:integer);
var E,diff:integer;
begin
  E := 0;
  SS := StepSizeTable[index];
  code := $00;
  diff := samp - valpred;

  if diff < 0 then
  begin
    code := $08;
    E := -diff;
  end
  else E := diff;

  if E >= SS then
  begin
    code := code or $04;
    E := E-SS;
  end;

  if E >= (SS div 2) then
  begin
    code := code or $02;
    E := E-(SS div 2);
  end;

  if E >= (SS div 4) then
  begin
    code := code or $01;
  end;
end;

procedure ADPCM_code(code:integer;i,data_byte:integer);
begin
  case (data_byte mod 2) of
    0 : begin
          if (i mod 2) = 0 then code1 := code shl 4;
          if (i mod 2) <> 0 then
          begin
            code2 := code;
            value := code1 or code2;
          end;
        end;
    1 : begin
          if i < (data_byte-1) then
          begin
            if (i mod 2) = 0 then code1 := code shl 4;
            if (i mod 2) <> 0 then
            begin
              code2 := code;
              value := code1 or code2;
            end;
          end;

          if i = (data_byte-1) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            begin
              code1 := code shl 4;
              code2 := $00;
              value := code1 or code2;
            end;
        end;
    end; {case}
end;

procedure write_adpcm_code(i:integer);
var ch : byte;
begin
  if i < (data_byte-1) then
  begin
    case (i mod 2) of
      0 : ;
      1 : begin
            ch := value;
            write(output_wav_codep,ch);
          end;
    end; {case}
  end;
  if i = (data_byte-1) then
  begin
    ch := value;
    write(output_wav_codep,ch);
  end;
end;

procedure adpcm_decode(code:integer);
var diff,E:integer;
    temp:integer;
begin
  SS := StepSizeTable[index];
  E := SS div 8;
  temp := code and $01;
  if temp <> 0 then
    E := E + (SS div 4);
  temp := code and $02;
  if temp <> 0 then
    E := E + (SS div 2);
  temp := code and $04;
  if temp <> 0 then
    E := E + SS;
  temp := code and $08;
  if temp <> 0 then
    diff := -E
  else diff := E;
  samp:=valpred+diff;
  valpred := samp;

  //To clip valpred value

  if (valpred < -4096) then
    valpred := -4096;
  if (valpred > 4096) then
    valpred := 4096;
end;

function step_adjust(code:integer):integer;
begin
  case (code and $07) of
    $00 : result := -1;
    $01 : result := -1;
    $02 : result := -1;
    $03 : result := -1;
    $04 : result := 2;
    $05 : result := 4;
    $06 : result := 6;
    $07 : result := 8;
  end;
end;

procedure ADPCM_algorithm;
var i : integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ch : byte;
begin
  i := 0;
  seek(input_wavp,58);
  reset(output_wav_codep);
  seek(output_wav_codep,60);
  while not eof(input_wavp) do
  begin
    read(input_wavp,level);
    samp := map_table[level];
    adpcm_encode(samp);
    ADPCM_code(code,i,data_byte);
    write_adpcm_code(i);
    adpcm_decode(code);
    index := index + step_adjust(code);

    //To clip index value

    if index < 0 then index := 0;
    if index > 48 then index := 48;

    SS := StepSizeTable[index];
    i := i+1;
  end; {while}
  closefile(output_wav_codep);
end;

procedure open_output_DOC_code_file;
var ch : byte;
    i : string;
begin
  assignfile(output_wav_codep,EncodewaveOut);
  reset(output_wav_codep);
  assignfile(output_doc_codep,EncodeOutDoc);
  rewrite(output_doc_codep);
  closefile(output_doc_codep);
  append(output_doc_codep);
  while not eof(output_wav_codep) do
  begin
    read(output_wav_codep,ch);
    i := inttostr(ch)+' ';
    write(output_doc_codep,i);
  end;
  closefile(output_doc_codep);

  ETotalADPCMByte := filesize(output_wav_codep);
  ETotalADPCMByteData := filesize(output_wav_codep)-60;
end;

procedure initialtable;
begin
  map_table[0] := -4096;
  map_table[1] := -3968;
  map_table[2] := -3840;
  map_table[3] := -3712;
  map_table[4] := -3584;
  map_table[5] := -3456;
  map_table[6] := -3328;
  map_table[7] := -3200;
  map_table[8] := -3072;
  map_table[9] := -2944;
  map_table[10] := -2816;
  map_table[11] := -2688;
  map_table[12] := -2560;
  map_table[13] := -2432;
  map_table[14] := -2304;
  map_table[15] := -2176;
  map_table[16] := -2048;
  map_table[17] := -1984;
  map_table[18] := -1920;
  map_table[19] := -1856;
  map_table[20] := -1792;
  map_table[21] := -1728;
  map_table[22] := -1664;
  map_table[23] := -1600;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
map_table[24] := -1536;
map_table[25] := -1472;
map_table[26] := -1408;
map_table[27] := -1344;
map_table[28] := -1280;
map_table[29] := -1216;
map_table[30] := -1152;
map_table[31] := -1088;
map_table[32] := -1024;
map_table[33] := -992;
map_table[34] := -960;
map_table[35] := -928;
map_table[36] := -896;
map_table[37] := -864;
map_table[38] := -832;
map_table[39] := -800;
map_table[40] := -768;
map_table[41] := -736;
map_table[42] := -704;
map_table[43] := -672;
map_table[44] := -640;
map_table[45] := -608;
map_table[46] := -576;
map_table[47] := -544;
map_table[48] := -512;
map_table[49] := -496;
map_table[50] := -480;
map_table[51] := -464;
map_table[52] := -448;
map_table[53] := -432;
map_table[54] := -416;
map_table[55] := -400;
map_table[56] := -384;
map_table[57] := -368;
map_table[58] := -352;
map_table[59] := -336;
map_table[60] := -320;
map_table[61] := -304;
map_table[62] := -288;
map_table[63] := -272;
map_table[64] := -256;
map_table[65] := -248;
map_table[66] := -240;
map_table[67] := -232;
map_table[68] := -224;
map_table[69] := -216;
map_table[70] := -208;
map_table[71] := -200;
map_table[72] := -192;
map_table[73] := -184;
map_table[74] := -176;
map_table[75] := -168;
map_table[76] := -160;
map_table[77] := -152;
map_table[78] := -144;
map_table[79] := -136;
map_table[80] := -128;
map_table[81] := -124;
map_table[82] := -120;
map_table[83] := -116;
map_table[84] := -112;
map_table[85] := -108;
map_table[86] := -104;
map_table[87] := -100;
map_table[88] := -96;
map_table[89] := -92;
map_table[90] := -88;
map_table[91] := -84;
map_table[92] := -80;
map_table[93] := -76;
map_table[94] := -72;
map_table[95] := -68;
map_table[96] := -64;
map_table[97] := -62;
map_table[98] := -60;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
map_table[99] := -58;
map_table[100] :=-56;
map_table[101] :=-54;
map_table[102] :=-52;
map_table[103] :=-50;
map_table[104] :=-48;
map_table[105] :=-46;
map_table[106] :=-44;
map_table[107] :=-42;
map_table[108] :=-40;
map_table[109] :=-38;
map_table[110] :=-36;
map_table[111] :=-34;
map_table[112] :=-32;
map_table[113] :=-30;
map_table[114] :=-28;
map_table[115] :=-26;
map_table[116] :=-24;
map_table[117] :=-22;
map_table[118] :=-20;
map_table[119] :=-18;
map_table[120] :=-16;
map_table[121] :=-14;
map_table[122] :=-12;
map_table[123] :=-10;
map_table[124] :=-8;
map_table[125] :=-6;
map_table[126] :=-4;
map_table[127] :=-2;
map_table[128] :=2;
map_table[129] :=4;
map_table[130] :=6;
map_table[131] :=8;
map_table[132] :=10;
map_table[133] :=12;
map_table[134] :=14;
map_table[135] :=16;
map_table[136] :=18;
map_table[137] :=20;
map_table[138] :=22;
map_table[139] :=24;
map_table[140] :=26;
map_table[141] :=28;
map_table[142] :=30;
map_table[143] :=32;
map_table[144] :=34;
map_table[145] :=36;
map_table[146] :=38;
map_table[147] :=40;
map_table[148] :=42;
map_table[149] :=44;
map_table[150] :=46;
map_table[151] :=48;
map_table[152] :=50;
map_table[153] :=52;
map_table[154] :=54;
map_table[155] :=56;
map_table[156] :=58;
map_table[157] :=60;
map_table[158] :=62;
map_table[159] :=64;
map_table[160] :=68;
map_table[161] :=72;
map_table[162] :=76;
map_table[163] :=80;
map_table[164] :=84;
map_table[165] :=88;
map_table[166] :=92;
map_table[167] :=96;
map_table[168] :=100;
map_table[169] :=104;
map_table[170] :=108;
map_table[171] :=112;
map_table[172] :=116;
map_table[173] :=120;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
map_table[174] :=124;
map_table[175] :=128;
map_table[176] :=136;
map_table[177] :=144;
map_table[178] :=152;
map_table[179] :=160;
map_table[180] :=168;
map_table[181] :=176;
map_table[182] :=184;
map_table[183] :=192;
map_table[184] :=200;
map_table[185] :=208;
map_table[186] :=216;
map_table[187] :=224;
map_table[188] :=232;
map_table[189] :=240;
map_table[190] :=248;
map_table[191] :=256;
map_table[192] :=272;
map_table[193] :=288;
map_table[194] :=304;
map_table[195] :=320;
map_table[196] :=336;
map_table[197] :=352;
map_table[198] :=368;
map_table[199] :=384;
map_table[200] :=400;
map_table[201] :=416;
map_table[202] :=432;
map_table[203] :=448;
map_table[204] :=464;
map_table[205] :=480;
map_table[206] :=496;
map_table[207] :=512;
map_table[208] :=544;
map_table[209] :=576;
map_table[210] :=608;
map_table[211] :=640;
map_table[212] :=672;
map_table[213] :=704;
map_table[214] :=736;
map_table[215] :=768;
map_table[216] :=800;
map_table[217] :=832;
map_table[218] :=864;
map_table[219] :=896;
map_table[220] :=928;
map_table[221] :=960;
map_table[222] :=992;
map_table[223] :=1024;
map_table[224] :=1088;
map_table[225] :=1152;
map_table[226] :=1216;
map_table[227] := 1280;
map_table[228] :=1344;
map_table[229] :=1408;
map_table[230] :=1472;
map_table[231] :=1536;
map_table[232] :=1600;
map_table[233] :=1664;
map_table[234] :=1728;
map_table[235] :=1792;
map_table[236] :=1856;
map_table[237] :=1920;
map_table[238] :=1984;
map_table[239] :=2048;
map_table[240] :=2176;
map_table[241] :=2304;
map_table[242] :=2432;
map_table[243] :=2560;
map_table[244] :=2688;
map_table[245] :=2816;
map_table[246] :=2944;
map_table[247] :=3072;
map_table[248] :=3200;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    map_table[249] :=3328;
    map_table[250] :=3456;
    map_table[251] :=3584;
    map_table[252] :=3712;
    map_table[253] :=3840;
    map_table[254] :=3968;
    map_table[255] :=4096;


    StepSizeTable[0] := 16;
    StepSizeTable[1] := 17;
    StepSizeTable[2] := 19;
    StepSizeTable[3] := 21;
    StepSizeTable[4] := 23;
    StepSizeTable[5] := 25;
    StepSizeTable[6] := 28;
    StepSizeTable[7] := 31;
    StepSizeTable[8] := 34;
    StepSizeTable[9] := 37;
    StepSizeTable[10] := 41;
    StepSizeTable[11] := 45;
    StepSizeTable[12] := 50;
    StepSizeTable[13] := 55;
    StepSizeTable[14] := 60;
    StepSizeTable[15] := 66;
    StepSizeTable[16] := 73;
    StepSizeTable[17] := 80;
    StepSizeTable[18] := 88;
    StepSizeTable[19] := 97;
    StepSizeTable[20] := 107;
    StepSizeTable[21] := 118;
    StepSizeTable[22] := 130;
    StepSizeTable[23] := 143;
    StepSizeTable[24] := 157;
    StepSizeTable[25] := 173;
    StepSizeTable[26] := 190;
    StepSizeTable[27] := 209;
    StepSizeTable[28] := 230;
    StepSizeTable[29] := 253;
    StepSizeTable[30] := 279;
    StepSizeTable[31] := 307;
    StepSizeTable[32] := 337;
    StepSizeTable[33] := 371;
    StepSizeTable[34] := 408;
    StepSizeTable[35] := 449;
    StepSizeTable[36] := 494;
    StepSizeTable[37] := 554;
    StepSizeTable[38] := 598;
    StepSizeTable[39] := 698;
    StepSizeTable[40] := 724;
    StepSizeTable[41] := 796;
    StepSizeTable[42] := 876;
    StepSizeTable[43] := 963;
    StepSizeTable[44] := 1060;
    StepSizeTable[45] := 1166;
    StepSizeTable[46] := 1282;
    StepSizeTable[47] := 1411;
    StepSizeTable[48] := 1552;

    valpred := 0;
    index := 0;

    ETotalWaveByte := 0;
    ETotalWaveByteData := 0;
    ETotalADPCMByte := 0;
    ETotalADPCMByteData := 0;
end;

function EncodeWave(wavefilename : string; var resultstr : string): boolean;
begin
    result := false;
    initialtable;

    assignfile(input_wavp,wavefilename);
    reset(input_wavp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if not check_wav_file then
    begin
      resultstr := 'File '+wavefilename+' is not .WAV file.'+#13+#10+
      'Can not compress file '+wavefilename;
    end {if not check_wav_file}
    else if not check_type_of_record then
    begin
      resultstr := 'File '+wavefilename+' is not PCM format.'+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else if check_channel = 2 then
    begin
      resultstr := 'Stereo channel.'+#13+#10+'Can not compress file '+
      wavefilename;
    end
    else if check_channel <> 1 then
    begin
      resultstr := 'Can not compress file '+wavefilename;
    end
    else if check_sampling_rate = 17 then
    begin
      resultstr := 'Sampling rate = 11.025 kHz.'+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else if check_sampling_rate = 34 then
    begin
      resultstr := 'Sampling rate = 22.05 kHz.'+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else if check_sampling_rate = 68 then
    begin
      resultstr := 'Sampling rate = 44.1 kHz.'+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else if not check_byte_per_sample  then
    begin
      resultstr := 'Not 1 byte per sample. '+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else if not check_bit_per_sample then
    begin
      resultstr := 'Not 8 bit per sample.'+#13+#10+
      'Can not compress file '+wavefilename;
    end
    else
    begin
      resultstr := 'File '+wavefilename+' is .WAV file.'+#13+#10+
      'File '+wavefilename+' is saved in PCM format.'+#13+#10+'Mono
channel.'+#13+#10+'Sampling rate = 8 kHz.'+#13+#10+'1 byte per
sample'+#13+#10+'8 bit per sample.';
      copy_all_to_DOC;
      open_output_wav_code_file;
      write_header_to_code_file;
      ADPCM_algorithm;
      open_output_DOC_code_file;
      closefile(input_wavp);
      closefile(output_wav_codep);
      result := true;
    end; {pass check}
  end;


  end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Source code ของโปรแกรมในส่วนของ unit NetvoiceDecodeLib**

```
unit NetVoiceDecodeLib;

interface

uses Windows, Sysutils;

const
  DecodeInDoc    = 'DecodeIn.doc';
  DecodeOutDoc   = 'DecodeOut.doc';
  DecodewaveOut  = 'Decode.wav';

var
  input_wav_codep,output_wavp:file of byte;
  input_docp,output_docp:textfile;
  SS,samp,amp:integer;
  index,index1,index2,valpred,valpred1,valpred2,data_byte:integer;
  StepSizeTable:array[0..48] of integer;
  adpcm_code:byte;
  code,code1,code2,output_level:integer;
  DTotalWaveByte,DTotalWaveByteData : integer;
  DTotalADPCMByte,DTotalADPCMByteData : integer;

  function DecodeWave(encodewavefilename : string; var resultstr : string):
boolean;

implementation

function check_wav_file() : boolean;
var ch:byte;
begin
  reset(input_wav_codep);
  seek(input_wav_codep,8);
  read(input_wav_codep,ch);
  if char(ch) = 'W' then
  begin
    read(input_wav_codep,ch);
    if char(ch) = 'A' then
    begin
      read(input_wav_codep,ch);
      if char(ch) = 'V' then
      begin
        read(input_wav_codep,ch);
        if char(ch) = 'E' then
        begin
          result := true;
        end
        else begin
          closefile(input_wav_codep);
          result := false;
        end;
      end
      else begin
        closefile(input_wav_codep);
        result := false;
      end;
    end
    else begin
      closefile(input_wav_codep);
      result := false;
    end;
  end {first if}
  else begin
    closefile(input_wav_codep);
    result := false;
  end;
end;

function check_type_of_record() : boolean;
var ch:byte;
begin
  reset(input_wav_codep);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  seek(input_wav_codep,20);
  read(input_wav_codep,ch);
  case ch of
    17 : result := true;
    else begin
      result := false;
      closefile(input_wav_codep);
    end;
  end;
end;

function check_channel() : integer;
var ch:byte;
begin
  reset(input_wav_codep);
  seek(input_wav_codep,22);
  read(input_wav_codep,ch);
  case ch of
    1 : result := 1;
    2 : begin
          result := 2;
          closefile(input_wav_codep);
        end;
    else begin
           result := 3;
           closefile(input_wav_codep);
    end;
  end;  {case}
end;

function check_sampling_rate() : integer;
var ch:byte;
begin
  reset(input_wav_codep);
  seek(input_wav_codep,24);
  read(input_wav_codep,ch);
  case ch of
    64 : result := 64;
    17 : begin
           result := 17;
           closefile(input_wav_codep);
         end;
    34 : begin
           result := 34;
           closefile(input_wav_codep);
         end;
    68 : begin
           result := 68;
           closefile(input_wav_codep);
         end;
  end;
end;

function check_bit_per_sample():boolean;
var ch:byte;
begin
  reset(input_wav_codep);
  seek(input_wav_codep,34);
  read(input_wav_codep,ch);
  if ch = 4 then
    result := true
  else begin
    result := false;
    closefile(input_wav_codep);
  end;
end;

procedure copy_all_to_DOC;
var ch:byte;
     i:string;
begin
  reset(input_wav_codep);
  assignfile(input_docp,DecodeInDoc);
  rewrite(input_docp);
  closefile(input_docp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  append(input_docp);
  while not eof(input_wav_codep) do
  begin
    read(input_wav_codep,ch);
    i := inttostr(ch)+' ';
    write(input_docp,i);
  end;
  closefile(input_docp);

  DTotalADPCMByteData := filesize(input_wav_codep);
  DTotalADPCMByte := filesize(input_wav_codep)-60;
  data_byte := filesize(input_wav_codep)-60;
end;

procedure open_output_wav_file;
begin
  assignfile(output_wavp,DecodewaveOut);
  rewrite(output_wavp);
  closefile(output_wavp);
  reset(output_wavp);
end;

procedure write_header_to_output_wav;
var i,size1,size2,size3,size4,data1,data2,data3,data4:integer;
    RIFF_size,data_size:integer;
    header,ch:byte;
begin
  i := 0;
  header := 0;
  reset(output_wavp);
  while i<=57 do
  begin
    write(output_wavp,header);
    i := i+1;
  end;

  seek(output_wavp,0);
  ch := ord('R');
  write(output_wavp,ch);
  ch := ord('I');
  write(output_wavp,ch);
  ch := ord('F');
  write(output_wavp,ch);
  ch := ord('F');
  write(output_wavp,ch);

  seek(output_wavp,filepos(output_wavp)+4);
  ch := ord('W');
  write(output_wavp,ch);
  ch := ord('A');
  write(output_wavp,ch);
  ch := ord('V');
  write(output_wavp,ch);
  ch := ord('E');
  write(output_wavp,ch);

  ch := ord('f');
  write(output_wavp,ch);
  ch := ord('m');
  write(output_wavp,ch);
  ch := ord('t');
  write(output_wavp,ch);
  ch := ord(' ');
  write(output_wavp,ch);
  ch := 18;
  write(output_wavp,ch);

  seek(output_wavp,filepos(output_wavp)+3);
  ch := 1;
  write(output_wavp,ch);

  seek(output_wavp,filepos(output_wavp)+1);
  ch := 1;
  write(output_wavp,ch);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
seek(output_wavp,filepos(output_wavp)+1);
ch := 64;
write(output_wavp,ch);
ch := 31;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+2);
ch := 64;
write(output_wavp,ch);
ch := 31;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+2);
ch := 1;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+1);
ch := 8;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+3);
ch := ord('f');
write(output_wavp,ch);
ch := ord('a');
write(output_wavp,ch);
ch := ord('c');
write(output_wavp,ch);
ch := ord('t');
write(output_wavp,ch);
ch := 4;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+7);
ch := ord('d');
write(output_wavp,ch);
ch := ord('a');
write(output_wavp,ch);
ch := ord('t');
write(output_wavp,ch);
ch := ord('a');
write(output_wavp,ch);

RIFF_size := (data_byte*2)+50;
size1 := RIFF_size shr 24;
size2 := (RIFF_size shr 16) and ($FF);
size3 := (RIFF_size shr 8) and ($FF);
size4 := (RIFF_size and $FF);
seek(output_wavp,4);
ch := size4;
write(output_wavp,ch);
ch := size3;
write(output_wavp,ch);
ch := size2;
write(output_wavp,ch);
ch := size1;
write(output_wavp,ch);

data_size := data_byte*2;
data1 := data_size shr 24;
data2 := (data_size shr 16) and ($FF);
data3 := (data_size shr 8) and ($FF);
data4 := data_size and $FF;
seek(output_wavp,46);
ch := data4;
write(output_wavp,ch);
ch := data3;
write(output_wavp,ch);
ch := data2;
write(output_wavp,ch);
ch := data1;
write(output_wavp,ch);

seek(output_wavp,filepos(output_wavp)+4);
ch := data4;
write(output_wavp,ch);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ch := data3;
  write(output_wavp,ch);
  ch := data2;
  write(output_wavp,ch);
  ch := data1;
  write(output_wavp,ch);
  closefile(output_wavp);
end;

function step_adjust(code:integer):integer;
begin
  case (code and $07) of
    $00 : result := -1;
    $01 : result := -1;
    $02 : result := -1;
    $03 : result := -1;
    $04 : result := 2;
    $05 : result := 4;
    $06 : result := 6;
    $07 : result := 8;
  end;
end;

procedure adpcm_decode(new_index,new_code:byte;new_valpred:integer);
var E,temp:integer;
    diff: integer;
begin
  SS := StepSizeTable[new_index],

  E := SS div 8;

  temp := new_code and $01;
  if temp <> 0 then
    E := E+(SS div 4);
  temp := new_code and $02;
  if temp <> 0 then
    E := E+(SS div 2);
  temp := new_code and $04;
  if temp <> 0 then
    E := E+SS;

  temp := new_code and $08;
  if temp <> 0 then
    diff := -E
  else diff := E;
  samp := new_valpred + diff;
  valpred := samp;

  //To clip valpred value

  if valpred < -4096 then valpred := -4096;
  if valpred > 4096 then valpred := 4096;

  index := index + step_adjust(new_code);

  //To clip index value

  if index < 0 then index := 0;
  if index >48 then index := 48;

  SS := StepSizeTable[index];
end;

function map_to_output_level(amp:integer):integer;
begin
  if ((amp>=-4096) and (amp<-4032))       then output_level := 0;
  if ((amp>=-4032) and (amp<-3904))       then output_level := 1;
  if ((amp>=-3904) and (amp<-3776))       then output_level := 2;
  if ((amp>=-3776) and (amp<-3648))       then output_level := 3;
  if ((amp>=-3648) and (amp<-3520))       then output_level := 4;
  if ((amp>=-3520) and (amp<-3392))       then output_level := 5;
  if ((amp>=-3392) and (amp<-3264))       then output_level := 6;
  if ((amp>=-3264) and (amp<-3136))       then output_level := 7;
  if ((amp>=-3136) and (amp<-3008))       then output_level := 8;
  if ((amp>=-3008) and (amp<-2880))       then output_level := 9;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((amp>=-2880) and (amp<-2752))       then output_level := 10;
if ((amp>=-2752) and (amp<-2624))       then output_level := 11;
if ((amp>=-2624) and (amp<-2496))       then output_level := 12;
if ((amp>=-2496) and (amp<-2368))       then output_level := 13;
if ((amp>=-2368) and (amp<-2240))       then output_level := 14;
if ((amp>=-2240) and (amp<-2112))       then output_level := 15;
if ((amp>=-2112) and (amp<-2016))       then output_level := 16;
if ((amp>=-2016) and (amp<-1952))       then output_level := 17;
if ((amp>=-1952) and (amp<-1888))       then output_level := 18;
if ((amp>=-1888) and (amp<-1824))       then output_level := 19;
if ((amp>=-1824) and (amp<-1760))       then output_level := 20;
if ((amp>=-1760) and (amp<-1696))       then output_level := 21;
if ((amp>=-1696) and (amp<-1632))       then output_level := 22;
if ((amp>=-1632) and (amp<-1568))       then output_level := 23;
if ((amp>=-1568) and (amp<-1504))       then output_level := 24;
if ((amp>=-1504) and (amp<-1440))       then output_level := 25;
if ((amp>=-1440) and (amp<-1376))       then output_level := 26;
if ((amp>=-1376) and (amp<-1312))       then output_level := 27;
if ((amp>=-1312) and (amp<-1248))       then output_level := 28;
if ((amp>=-1248) and (amp<-1184))       then output_level := 29;
if ((amp>=-1184) and (amp<-1120))       then output_level := 30;
if ((amp>=-1120) and (amp<-1056))       then output_level := 31;
if ((amp>=-1056) and (amp<-1008))       then output_level := 32;
if ((amp>=-1008) and (amp<-976)) then output_level := 33;
if ((amp>=-976) and (amp<-944))  then output_level := 34;
if ((amp>=-944) and (amp<-912))  then output_level := 35;
if ((amp>=-912) and (amp<-880))  then output_level := 36;
if ((amp>=-880) and (amp<-848))  then output_level := 37;
if ((amp>=-848) and (amp<-816))  then output_level := 38;
if ((amp>=-816) and (amp<-784))  then output_level := 39;
if ((amp>=-784) and (amp<-752))  then output_level := 40;
if ((amp>=-752) and (amp<-720))  then output_level := 41;
if ((amp>=-720) and (amp<-688))  then output_level := 42;
if ((amp>=-688) and (amp<-656))  then output_level := 43;
if ((amp>=-656) and (amp<-624))  then output_level := 44;
if ((amp>=-624) and (amp<-592))  then output_level := 45;
if ((amp>=-592) and (amp<-560))  then output_level := 46;
if ((amp>=-560) and (amp<-528))  then output_level := 47;
if ((amp>=-528) and (amp<-504))  then output_level := 48;
if ((amp>=-504) and (amp<-488))  then output_level := 49;
if ((amp>=-488) and (amp<-472))  then output_level := 50;
if ((amp>=-472) and (amp<-456))  then output_level := 51;
if ((amp>=-456) and (amp<-440))  then output_level := 52;
if ((amp>=-440) and (amp<-424))  then output_level := 53;
if ((amp>=-424) and (amp<-408))  then output_level := 54;
if ((amp>=-408) and (amp<-392))  then output_level := 55;
if ((amp>=-392) and (amp<-376))  then output_level := 56;
if ((amp>=-376) and (amp<-360))  then output_level := 57;
if ((amp>=-360) and (amp<-344))  then output_level := 58;
if ((amp>=-344) and (amp<-328))  then output_level := 59;
if ((amp>=-328) and (amp<-312))  then output_level := 60;
if ((amp>=-312) and (amp<-296))  then output_level := 61;
if ((amp>=-296) and (amp<-280))  then output_level := 62;
if ((amp>=-280) and (amp<-264))  then output_level := 63;
if ((amp>=-264) and (amp<-252))  then output_level := 64;
if ((amp>=-252) and (amp<-244))  then output_level := 65;
if ((amp>=-244) and (amp<-236))  then output_level := 66;
if ((amp>=-236) and (amp<-228))  then output_level := 67;
if ((amp>=-228) and (amp<-220))  then output_level := 68;
if ((amp>=-220) and (amp<-212))  then output_level := 69;
if ((amp>=-212) and (amp<-204))  then output_level := 70;
if ((amp>=-204) and (amp<-196))  then output_level := 71;
if ((amp>=-196) and (amp<-188))  then output_level := 72;
if ((amp>=-188) and (amp<-180))  then output_level := 73;
if ((amp>=-180) and (amp<-172))  then output_level := 74;
if ((amp>=-172) and (amp<-164))  then output_level := 75;
if ((amp>=-164) and (amp<-156))  then output_level := 76;
if ((amp>=-156) and (amp<-148))  then output_level := 77;
if ((amp>=-148) and (amp<-140))  then output_level := 78;
if ((amp>=-140) and (amp<-132))  then output_level := 79;
if ((amp>=-132) and (amp<-126))  then output_level := 80;
if ((amp>=-126) and (amp<-122))  then output_level := 81;
if ((amp>=-122) and (amp<-118))  then output_level := 82;
if ((amp>=-118) and (amp<-114))  then output_level := 83;
if ((amp>=-114) and (amp<-110))  then output_level := 84;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((amp>=-110) and (amp<-106))  then output_level := 85;
if ((amp>=-106) and (amp<-102))  then output_level := 86;
if ((amp>=-102) and (amp<-98))   then output_level := 87;
if ((amp>=-98) and (amp<-94))          then output_level := 88;
if ((amp>=-94) and (amp<-90))          then output_level := 89;
if ((amp>=-90) and (amp<-86))          then output_level := 90;
if ((amp>=-86) and (amp<-82))          then output_level := 91;
if ((amp>=-82) and (amp<-78))          then output_level := 92;
if ((amp>=-78) and (amp<-74))          then output_level := 93;
if ((amp>=-74) and (amp<-70))          then output_level := 94;
if ((amp>=-70) and (amp<-66))          then output_level := 95;
if ((amp>=-66) and (amp<-63))          then output_level := 96;
if ((amp>=-63) and (amp<-61))          then output_level := 97;
if ((amp>=-61) and (amp<-59))          then output_level := 98;
if ((amp>=-59) and (amp<-57))          then output_level := 99;
if ((amp>=-57) and (amp<-55))          then output_level := 100;
if ((amp>=-55) and (amp<-53))          then output_level := 101;
if ((amp>=-53) and (amp<-51))          then output_level := 102;
if ((amp>=-51) and (amp<-49))          then output_level := 103;
if ((amp>=-49) and (amp<-47))          then output_level := 104;
if ((amp>=-47) and (amp<-45))          then output_level := 105;
if ((amp>=-45) and (amp<-43))          then output_level := 106;
if ((amp>=-43) and (amp<-41))          then output_level := 107;
if ((amp>=-41) and (amp<-39))          then output_level := 108;
if ((amp>=-39) and (amp<-37))          then output_level := 109;
if ((amp>=-37) and (amp<-35))          then output_level := 110;
if ((amp>=-35) and (amp<-33))          then output_level := 111;
if ((amp>=-33) and (amp<-31))          then output_level := 112;
if ((amp>=-31) and (amp<-29))          then output_level := 113;
if ((amp>=-29) and (amp<-27))          then output_level := 114;
if ((amp>=-27) and (amp<-25))          then output_level := 115;
if ((amp>=-25) and (amp<-23))          then output_level := 116;
if ((amp>=-23) and (amp<-21))          then output_level := 117;
if ((amp>=-21) and (amp<-19))          then output_level := 118;
if ((amp>=-19) and (amp<-17))          then output_level := 119;
if ((amp>=-17) and (amp<-15))          then output_level := 120;
if ((amp>=-15) and (amp<-13))          then output_level := 121;
if ((amp>=-13) and (amp<-11))          then output_level := 122;
if ((amp>=-11) and (amp<-9))           then output_level := 123;
if ((amp>=-9) and (amp<-7))            then output_level := 124;
if ((amp>=-7) and (amp<-5))            then output_level := 125;
if ((amp>=-5) and (amp<-3))            then output_level := 126;
if ((amp>=-3) and (amp<0))             then output_level := 127;
if ((amp>=0) and (amp<3))        then output_level := 128;
if ((amp>=3) and (amp<5))        then output_level := 129;
if ((amp>=5) and (amp<7))        then output_level := 130;
if ((amp>=7) and (amp<9))        then output_level := 131;
if ((amp>=9) and (amp<11))             then output_level := 132;
if ((amp>=11) and (amp<13))            then output_level := 133;
if ((amp>=13) and (amp<15))            then output_level := 134;
if ((amp>=15) and (amp<17))            then output_level := 135;
if ((amp>=17) and (amp<19))            then output_level := 136;
if ((amp>=19) and (amp<21))            then output_level := 137;
if ((amp>=21) and (amp<23))            then output_level := 138;
if ((amp>=23) and (amp<25))            then output_level := 139;
if ((amp>=25) and (amp<27))            then output_level := 140;
if ((amp>=27) and (amp<29))            then output_level := 141;
if ((amp>=29) and (amp<31))            then output_level := 142;
if ((amp>=31) and (amp<33))            then output_level := 143;
if ((amp>=33) and (amp<35))            then output_level := 144;
if ((amp>=35) and (amp<37))            then output_level := 145;
if ((amp>=37) and (amp<39))            then output_level := 146;
if ((amp>=39) and (amp<41))            then output_level := 147;
if ((amp>=41) and (amp<43))            then output_level := 148;
if ((amp>=43) and (amp<45))            then output_level := 149;
if ((amp>=45) and (amp<47))            then output_level := 150;
if ((amp>=47) and (amp<49))            then output_level := 151;
if ((amp>=49) and (amp<51))            then output_level := 152;
if ((amp>=51) and (amp<53))            then output_level := 153;
if ((amp>=53) and (amp<55))            then output_level := 154;
if ((amp>=55) and (amp<57))            then output_level := 155;
if ((amp>=57) and (amp<59))            then output_level := 156;
if ((amp>=59) and (amp<61))            then output_level := 157;
if ((amp>=61) and (amp<63))            then output_level := 158;
if ((amp>=63) and (amp<66))            then output_level := 159;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((amp>=66) and (amp<70))           then output_level := 160;
if ((amp>=70) and (amp<74))           then output_level := 161;
if ((amp>=74) and (amp<78))           then output_level := 162;
if ((amp>=78) and (amp<82))           then output_level := 163;
if ((amp>=82) and (amp<86))           then output_level := 164;
if ((amp>=86) and (amp<90))           then output_level := 165;
if ((amp>=90) and (amp<94))           then output_level := 166;
if ((amp>=94) and (amp<98))           then output_level := 167;
if ((amp>=98) and (amp<102))          then output_level := 168;
if ((amp>=102) and (amp<106))         then output_level := 169;
if ((amp>=106) and (amp<110))         then output_level := 170;
if ((amp>=110) and (amp<114))         then output_level := 171;
if ((amp>=114) and (amp<118))         then output_level := 172;
if ((amp>=118) and (amp<122))         then output_level := 173;
if ((amp>=122) and (amp<126))         then output_level := 174;
if ((amp>=126) and (amp<132))         then output_level := 175;
if ((amp>=132) and (amp<140))         then output_level := 176;
if ((amp>=140) and (amp<148))         then output_level := 177;
if ((amp>=148) and (amp<156))         then output_level := 178;
if ((amp>=156) and (amp<164))         then output_level := 179;
if ((amp>=164) and (amp<172))         then output_level := 180;
if ((amp>=172) and (amp<180))         then output_level := 181;
if ((amp>=180) and (amp<188))         then output_level := 182;
if ((amp>=188) and (amp<196))         then output_level := 183;
if ((amp>=196) and (amp<204))         then output_level := 184;
if ((amp>=204) and (amp<212))         then output_level := 185;
if ((amp>=212) and (amp<220))         then output_level := 186;
if ((amp>=220) and (amp<228))         then output_level := 187;
if ((amp>=228) and (amp<236))         then output_level := 188;
if ((amp>=236) and (amp<244))         then output_level := 189;
if ((amp>=244) and (amp<252))         then output_level := 190;
if ((amp>=252) and (amp<264))         then output_level := 191;
if ((amp>=264) and (amp<280))         then output_level := 192;
if ((amp>=280) and (amp<296))         then output_level := 193;
if ((amp>=296) and (amp<312))         then output_level := 194;
if ((amp>=312) and (amp<328))         then output_level := 195;
if ((amp>=328) and (amp<344))         then output_level := 196;
if ((amp>=344) and (amp<360))         then output_level := 197;
if ((amp>=360) and (amp<376))         then output_level := 198;
if ((amp>=376) and (amp<392))         then output_level := 199;
if ((amp>=392) and (amp<408))         then output_level := 200;
if ((amp>=408) and (amp<424))         then output_level := 201;
if ((amp>=424) and (amp<440))         then output_level := 202;
if ((amp>=440) and (amp<456))         then output_level := 203;
if ((amp>=456) and (amp<472))         then output_level := 204;
if ((amp>=472) and (amp<488))         then output_level := 205;
if ((amp>=488) and (amp<504))         then output_level := 206;
if ((amp>=504) and (amp<528))         then output_level := 207;
if ((amp>=528) and (amp<560))         then output_level := 208;
if ((amp>=560) and (amp<592))         then output_level := 209;
if ((amp>=592) and (amp<624))         then output_level := 210;
if ((amp>=624) and (amp<656))         then output_level := 211;
if ((amp>=656) and (amp<688))         then output_level := 212;
if ((amp>=688) and (amp<720))         then output_level := 213;
if ((amp>=720) and (amp<752))         then output_level := 214;
if ((amp>=752) and (amp<784))         then output_level := 215;
if ((amp>=784) and (amp<816))         then output_level := 216;
if ((amp>=816) and (amp<848))         then output_level := 217;
if ((amp>=848) and (amp<880))         then output_level := 218;
if ((amp>=880) and (amp<912))         then output_level := 219;
if ((amp>=912) and (amp<944))         then output_level := 220;
if ((amp>=944) and (amp<976))         then output_level := 221;
if ((amp>=976) and (amp<1008))   then output_level := 222;
if ((amp>=1008) and (amp<1056))  then output_level := 223;
if ((amp>=1056) and (amp<1120))  then output_level := 224;
if ((amp>=1120) and (amp<1184))  then output_level := 225;
if ((amp>=1184) and (amp<1248))  then output_level := 226;
if ((amp>=1248) and (amp<1312))  then output_level := 227;
if ((amp>=1312) and (amp<1376))  then output_level := 228;
if ((amp>=1376) and (amp<1440))  then output_level := 229;
if ((amp>=1440) and (amp<1504))  then output_level := 230;
if ((amp>=1504) and (amp<1568))  then output_level := 231;
if ((amp>=1568) and (amp<1632))  then output_level := 232;
if ((amp>=1632) and (amp<1696))  then output_level := 233;
if ((amp>=1696) and (amp<1760))  then output_level := 234;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if ((amp>=1760) and (amp<1824))  then output_level := 235;
  if ((amp>=1824) and (amp<1888))  then output_level := 236;
  if ((amp>=1888) and (amp<1952))  then output_level := 237;
  if ((amp>=1952) and (amp<2016))  then output_level := 238;
  if ((amp>=2016) and (amp<2112))  then output_level := 239;
  if ((amp>=2112) and (amp<2240))  then output_level := 240;
  if ((amp>=2240) and (amp<2368))  then output_level := 241;
  if ((amp>=2368) and (amp<2496))  then output_level := 242;
  if ((amp>=2496) and (amp<2624))  then output_level := 243;
  if ((amp>=2624) and (amp<2752))  then output_level := 244;
  if ((amp>=2752) and (amp<2880))  then output_level := 245;
  if ((amp>=2880) and (amp<3008))  then output_level := 246;
  if ((amp>=3008) and (amp<3136))  then output_level := 247;
  if ((amp>=3136) and (amp<3264))  then output_level := 248;
  if ((amp>=3264) and (amp<3392))  then output_level := 249;
  if ((amp>=3392) and (amp<3520))  then output_level := 250;
  if ((amp>=3520) and (amp<3648))  then output_level := 251;
  if ((amp>=3648) and (amp<3776))  then output_level := 252;
  if ((amp>=3776) and (amp<3904))  then output_level := 253;
  if ((amp>=3904) and (amp<4032))  then output_level := 254;
  if ((amp>=4032) and (amp<=4096)) then output_level := 255;
  result := output_level;
end;

function map_to_level(amp:integer):integer;
begin
  //To clip amplitude of sample2 for 12 bit linear samples
  if amp < -4096 then amp := -4096;
  if amp > 4096 then amp := 4096;

  output_level := map_to_output_level(amp);
  result := output_level;
end;

procedure decoder_section;
begin
  read(input_wav_codep,adpcm_code);

  code1 := adpcm_code shr 4;
  code2 := adpcm_code and $0F;

  index1 := index;
  valpred1 := valpred;

  adpcm_decode(index1,code1,valpred1);
  output_level := map_to_level(valpred);

  write(output_wavp,output_level);
  index2 := index;
  valpred2 := valpred;

  adpcm_decode(index2,code2,valpred2);

  output_level := map_to_level(valpred);

  write(output_wavp,output_level);
end;

procedure ADPCM_algorithm;
begin
  seek(input_wav_codep,60);
  reset(output_wavp);
  seek(output_wavp,58);
  while not eof(input_wav_codep) do
    decoder_section;
end;

procedure open_output_DOC_file;
var ch : byte;
    i : string;
begin
  assignfile(output_docp,DecodeOutDoc);
  rewrite(output_docp);
  closefile(output_docp);
  append(output_docp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  reset(output_wavp);
  while not eof(output_wavp) do
  begin
    read(output_wavp,ch);
    i := inttostr(ch) + ' ';
    write(output_docp,i);
  end;
  closefile(output_docp);

  DTotalWaveByte := filesize(output_wavp);
  DTotalWaveByteData := filesize(output_wavp)-60;
end;

procedure initialtable;
begin
  StepSizeTable[0] := 16;
  StepSizeTable[1] := 17;
  StepSizeTable[2] := 19;
  StepSizeTable[3] := 21;
  StepSizeTable[4] := 23;
  StepSizeTable[5] := 25;
  StepSizeTable[6] := 28;
  StepSizeTable[7] := 31;
  StepSizeTable[8] := 34;
  StepSizeTable[9] := 37;
  StepSizeTable[10] := 41;
  StepSizeTable[11] := 45;
  StepSizeTable[12] := 50;
  StepSizeTable[13] := 55;
  StepSizeTable[14] := 60;
  StepSizeTable[15] := 66;
  StepSizeTable[16] := 73;
  StepSizeTable[17] := 80;
  StepSizeTable[18] := 88;
  StepSizeTable[19] := 97;
  StepSizeTable[20] := 107;
  StepSizeTable[21] := 118;
  StepSizeTable[22] := 130;
  StepSizeTable[23] := 143;
  StepSizeTable[24] := 157;
  StepSizeTable[25] := 173;
  StepSizeTable[26] := 190;
  StepSizeTable[27] := 209;
  StepSizeTable[28] := 230;
  StepSizeTable[29] := 253;
  StepSizeTable[30] := 279;
  StepSizeTable[31] := 307;
  StepSizeTable[32] := 337;
  StepSizeTable[33] := 371;
  StepSizeTable[34] := 408;
  StepSizeTable[35] := 449;
  StepSizeTable[36] := 494;
  StepSizeTable[37] := 544;
  StepSizeTable[38] := 598;
  StepSizeTable[39] := 658;
  StepSizeTable[40] := 724;
  StepSizeTable[41] := 796;
  StepSizeTable[42] := 876;
  StepSizeTable[43] := 963;
  StepSizeTable[44] := 1060;
  StepSizeTable[45] := 1166;
  StepSizeTable[46] := 1282;
  StepSizeTable[47] := 1411;
  StepSizeTable[48] := 1552;

  index := 0;
  valpred := 0;

  DTotalWaveByte := 0;
  DTotalWaveByteData := 0;
  DTotalADPCMByte := 0;
  DTotalADPCMByteData := 0;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function DecodeWave(encodewavefilename : string; var resultstr : string):
boolean;
begin
  result := false;
  initialtable;

  assignfile(input_wav_codep,encodewavefilename);
  reset(input_wav_codep);
  if not check_wav_file then
  begin
    resultstr := 'File '+encodewavefilename+' is not .WAV file.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else if not check_type_of_record then
  begin
    resultstr := 'File '+encodewavefilename+' is not PCM format.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else if check_channel = 2 then
  begin
    resultstr := 'Stereo channel.'+#13+#10+'Can not compress file '+
    encodewavefilename;
  end
  else if check_channel <> 1 then
  begin
    resultstr := 'Can not compress file '+encodewavefilename;
  end
  else if check_sampling_rate = 17 then
  begin
    resultstr := 'Sampling rate = 11.025 kHz.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else if check_sampling_rate = 34 then
  begin
    resultstr := 'Sampling rate = 22.05 kHz.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else if check_sampling_rate = 68 then
  begin
    resultstr := 'Sampling rate = 44.1 kHz.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else if not check_bit_per_sample then
  begin
    resultstr := 'Not 4 bit per sample.'+#13+#10+
    'Can not compress file '+encodewavefilename;
  end
  else begin
    resultstr := 'File '+encodewavefilename+' is .WAV file.'+#13+#10+
    'File '+encodewavefilename+' is saved in PCM format.'+#13+#10+'Mono
channel.'+#13+#10+'Sampling rate = 8 kHz.'+#13+#10+'4 bit per sample.';
    copy_all_to_DOC;
    open_output_wav_file;
    write_header_to_output_wav;
    ADPCM_algorithm;
    open_output_DOC_file;
    closefile(input_wav_codep);
    closefile(output_wavp);
    result := true;
  end;
end;


end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Source code ของโปรแกรมในส่วนของ unit NetVoicePlayP**

```
unit NetVoicePlayP;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, ComCtrls, MPlayer;

type
  TNetVoicePlayF = class(TForm)
    Bevel1: TBevel;
    MediaPlayer1: TMediaPlayer;
    ProgressBar1: TProgressBar;
    SaveDialog1: TSaveDialog;
    Timer1: TTimer;
    SaveAsBtn: TButton;
    CloseBtn: TButton;
    procedure Timer1Timer(Sender: TObject);
    procedure MediaPlayer1Notify(Sender: TObject);
    procedure MediaPlayer1Click(Sender: TObject; Button: TMPBtnType; var
DoDefault: Boolean);
    procedure SaveAsBtnClick(Sender: TObject);
    procedure CloseBtnClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure OpenWave;
  end;

var
  NetVoicePlayF: TNetVoicePlayF;

implementation

uses NetVoiceDecodeLib, NetVoiceP;

procedure TNetVoicePlayF.OpenWave;
begin
  MediaPlayer1.Close;
  MediaPlayer1.FileName := DecodewaveOut;
  MediaPlayer1.Open;
  ProgressBar1.Max := MediaPlayer1.Length;
  ProgressBar1.Position := 0;
end;

procedure TNetVoicePlayF.Timer1Timer(Sender: TObject);
begin
  ProgressBar1.Position := MediaPlayer1.Position
end;

procedure TNetVoicePlayF.MediaPlayer1Notify(Sender: TObject);
begin
  with Sender as TMediaPlayer do
  begin
    if Mode = mpStopped then
    begin
      Timer1.Enabled := False;
      ProgressBar1.Position := 0;
      Notify := True;
    end;
  end;
end;

procedure TNetVoicePlayF.MediaPlayer1Click(Sender: TObject; Button: TMPBtnType;
  var DoDefault: Boolean);
begin
  if Button = btPlay then Timer1.Enabled := True;
end;

procedure TNetVoicePlayF.SaveAsBtnClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  if SaveDialog1.Execute then
   if Not(CopyFile(DecodewaveOut,PChar(SaveDialog1.FileName),True)) then
     Showmessage('Cannot save file')
end;

procedure TNetVoicePlayF.CloseBtnClick(Sender: TObject);
begin
  MediaPlayer1.Close;
  Close;
  NetVoiceF.Show;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# The Microsoft Multimedia WAV Sound File Format

Written for the PC-GPE by Mark Feldman

e-mail address : u914097@student.canberra.edu.au

myndale@cairo.anu.edu.au

## Disclaimer

I assume no responsibility whatsoever for any effect that this file, the information contained therein or the use thereof has on you, your sanity, computer, spouse, children, pets or anything else related to you or your existance. No warranty is provided nor implied with this information.

## The RIFF File Format

WAV files use the RIFF file structure. The RIFF format was designed for multi-media purposes. A RIFF files consists of a number of "chunks":

| Byte Offset | Name | Length (in bytes) | Description |
|---|---|---|---|
| 00h | rID | 4h | "RIFF" |
| 04h | rLen | 4h | Длина последующей порции (chunk) |
| 08h | rData | rLen | Собственно "chunk" |

## The WAVE Form Definition

The rData chunk in a WAV file is split up into several further chunks:

| rData Byte Offset | Name | Length (in bytes) | Description |
|---|---|---|---|
| 00h | wID | 4h | "WAVE" |
| 04h | Format Chunk | 14h | Contains data which specifies the format of the Data in the Data Chunk |
| 18h | WAVE Data Chunk | ? | Contains the WAV audio data |

## The Format Chunk

The Format Chunk is split up into these fields:

| Format Chunk Offset | Name | Length (in bytes) | Description |
|---|---|---|---|
| 00h | fId | 4 | "fmt" |
| 04h | fLen | 4 | Length of data in the format chunk |
| 08h | wFormatTag | 2 | * |
| 0Ah | nChannels | 2 | Number of channels, 1=mono, 2=stereo |
| 0Ch | nSamplesPerSec | 2 | Playback frequency |
| 0Eh | nAvgBytesPerSec | 2 | ** |
| 10h | nBlockAlign | 2 | *** |
| 12h | FormatSpecific | 2 | Format specific data area |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- * The wFormatTag specifies the wave format, eg 1 = Pulse Code Modulation (or in plain english, regular 8 bit sampled uncompressed sound)
- ** Indicates the average number of bytes a second the data should be transferred at = nChannels * nSamplesPerSec * (nBitsPerSample / 8)
- *** Indicates the block alignment of the data in the data chunk. Software needs to process a multiplt of nBlockAlign at a time. nBlockAlign = nChannels * (nBitsPerSample / 8)

## The Data Chunk

The Data Chunk is split up into these fields:

| Data Chunk Offset | Name | Length (in bytes) | Description |
|---|---|---|---|
| 00h | dId | 4 | Contains the characters "data" |
| 02h | dLen | 4 | Length of data in the dData field |
| 00h | dData | dLen | The actual waveform data |

In mono 8-bit files each byte represents one sample. In stereo 8-bit files two bytes are stored for each sample, the first byte is the left channel value, the next is the right channel value.

*This page is located at* vt.miem.edu.ru

*Updated:* Tuesday, 22-Aug-2000 18:21:11 MSD

About bugs mail to WebMaster

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ผู้จัดทำ ต้องขอขอบพระคุณ ดร.สุทธิชัย นพนาคีพงษ์ อาจารย์ที่ปรึกษาที่ให้โอกาสในการทำโครงงานนี้ และอาจารย์ กฤษณ์ วงรุจิระ ที่ให้คำแนะนำในการทำโครงงาน อีกทั้งยังช่วยกระตุ้นให้ทำโครงงาน ขอขอบคุณ คุณดำรงศักดิ์ พิพิธวรรณ ที่ให้คำแนะนำต่างๆในส่วนของโปรแกรม และเว็บไซท์ต่างๆที่เป็นแหล่งข้อมูลในการเขียนโปรแกรม ทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หนังสืออ้างอิง


1. กมลมาศ กำธรกิจการ , "คู่มือพัฒนาโปรแกรมด้วย Delphi 4" , กรุงเทพมหานคร : บริษัท โปรวิชั่น จำกัด , พิมพ์ครั้งที่ 1 , มิถุนายน 2542

2. ถวิล กิ่งทอง , "เทคโนโลยีการส่งสัญญาณดิจิตอล" , กรุงเทพมหานคร : ตำราชุดวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุรุทหารลาดกระบัง

3. ภาสกร ไหลสกุล , "อินเตอร์เน็ตทำงานอย่างไร" , กรุงเทพมหานคร : บริษัทโปรวิชั่น จำกัด , 2539

4. สัจจะ จรัสรุ่งรวีวร , จักรพงษ์ สุขประเสริฐ , "เริ่มต้นอย่างมืออาชีพด้วย Delphi 5 ฉบับสมบูรณ์" , กรุงเทพมหานคร : อินโฟเพลส , พิมพ์ครั้งที่ 1 , กุมภาพันธ์ 2543

5. สุรศักดิ์ สงวนพงษ์ , "สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี" , กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2543

6. William Stallings , "Data&Computer Communication" , Sixth Edition , Prentice Hall International, Inc. , 2000



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้