สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาการเข้ารหัสตามมาตรฐาน MPEG-1/Audio Layer1 โดยใช้ DSP 56009 EVM

MPEG-1/Audio Layer 1 by using DSP 56009 EVM

T042289

จัดทำโดย

นาย จิรยุทธ์ ชดทิศ

นาย ชินเดช ดารานุวัฒน์

นาย ธโนบล เซ็นภักดี

เลขหมู่.....
เลขทะเบียน 42289
วัน, เดือน, ปี 16 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11201447

การพัฒนาการเข้ารหัสตามมาตรฐาน MPEG-1/AudioLayer1 โดยใช้ DSP 56009 EVM

MPEG-1/Audio Layer 1 by using DSP 56009 EVM

จัดทำโดย

นาย จิรยุทธ์	ชดทิศ	40010131
นาย ชินเดช	ดารานุวัฒน์	40010184
นาย ธโนบล	เซ็นภักดี	40010308

อาจารย์ที่ปรึกษา

ผศ.ดร. สมศักดิ์ ชุมช่วย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ภาคเรียนที่ 2 ปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาการเข้ารหัสตามมาตรฐาน MPEG-1/Audio Layer 1 โดยใช้ DSP 56009 EVM

ผู้จัดทำ

นาย จิรยุทธ์	ชดทิศ	40010131
นาย ชินเดช	ดารานุวัฒน์	40010184
นาย ธโนบ่ล	เซ็นภักดี	40010308

..... อาจารย์ที่ปรึกษา

(ผศ.ดร.สมศักดิ์ ชุมช่วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง การพัฒนาการเข้ารหัสตามมาตรฐาน MPEG-1/Audio Layer 1 โดยใช้ DSP 56009 EVM

MPEG-1/Audio Layer 1 by using DSP 56009 EVM

นาย จิรยุทธ์	ชดทิศ	40010131
นาย ชินเดช	ดารานุวัฒน์	40010184
นาย ธโนบล	เซ็นภักดี	40010308

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. สมศักดิ์ ชุมช่วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

งานชิ้นนี้สำเร็จลงได้ก็ด้วยความเอื้อเฟื้อและความอนุเคราะห์จากบุคคลหลายๆ ท่าน อาทิเช่น ผศ.ดร. สมศักดิ์ ชุมช่วย อาจารย์ที่ปรึกษาที่แสนใจดี ที่เอื้อเฟื้อหนังสือ และคำแนะนำต่างๆ เวลาที่พวกเรามีส่วนที่ไม่เข้าใจ ,อาจารย์สุรเดช ตรีไตรลักษณา ที่แนะนำหนังสือ ,พี่ต่อ พี่โอ๋ พี่บัญ พี่ปุ้ย พี่เทพ ที่เอื้อเฟื้อคอมพิวเตอร์เวลาที่เราไม่มีคอมใช้ และแนะนำ website ที่เกี่ยวข้อง , สันต์ เน ชาญ น้อง หนุ่ม โต้ง เพื่อนร่วมแล็ป ที่คอยให้กำลังใจ และยังมีบุคคลอื่นๆอีก ที่ไม่ได้เอ่ยนามในที่นี้ ขอขอบคุณทุกๆท่านครับและท้ายที่สุดที่ขาดไม่ได้คือ ขอขอบคุณ คุณพ่อ คุณแม่ ที่ทำให้พวกเราเกิดมาและให้การศึกษาเลี้ยงดูอบรมอย่างดี จนพวกเราได้มาทำโปรเจ็คนี้

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ในปัจจุบันนี้เทคโนโลยีทางด้านคอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันมาก ทั้งในด้านการใช้งานและความบันเทิง เช่น ด้านมัลติมีเดียทั้งทางภาพและเสียง ซึ่งเป็นการประมวลผลข้อมูลทางด้านดิจิตอลทั้งสิ้น ดังนั้นจึงได้มีผู้คิดค้นวิธีการต่างๆ เพื่อทำให้ข้อมูลภาพและเสียงเหล่านั้น ซึ่งเดิมทีมีขนาดใหญ่มากไม่สะดวกในการเก็บและส่งถ่ายข้อมูลนั้น ให้มีขนาดเล็กลงโดยบีบอัดข้อมูล ซึ่งหลังการบีบอัดข้อมูลนี้คุณภาพของภาพและเสียงก็เคียงใกล้เคียงเดิมก่อนการบีบอัดให้มากที่สุด แต่ในที่นี้เราจะสนใจแต่เสียงเพียงอย่างเดียว ซึ่งการบีบอัดสัญญาณเสียงทางดิจิตอลนี้จะสามารถทำให้เกิดการเก็บและการส่งถ่ายข้อมูลทางเสียงได้อย่างมีประสิทธิภาพ มีวิธีการในการบีบอัดข้อมูลเสียงนี้อยู่หลายวิธีแต่ละวิธีจะมีความซับซ้อนในการเข้ารหัสและถอดรหัสแตกต่างกันไป แตกต่างกันทางด้านคุณภาพของเสียงที่ได้ออกมาหลังการบีบอัด และแตกต่างกันทางด้านจำนวนของข้อมูลที่บีบอัดลงไป แต่ที่เราสนใจก็คือการบีบอัดแบบมาตรฐาน MPEG-1/Audio (ISO/IEC 11172-3) ซึ่งเป็นวิธีซึ่งมีความซับซ้อนมาก บีบอัดข้อมูลได้สูง และเป็นกระบวนการที่จะได้สัญญาณเสียงที่มีคุณภาพดี ซึ่งการบีบอัดข้อมูลนั้นจะมีทั้งการเข้ารหัสและการถอดรหัส แต่ที่เราจะสนใจคือการเข้ารหัสซึ่งจะเป็นกระบวนการเอาสัญญาณเสียงทางดิจิตอลมาบีบอัดให้เกิดข้อมูลที่มีขนาดเล็กลง(MPEG-1/Audio) หลังจากนั้นเราก็สามารถนำข้อมูลที่ถูกบีบอัดนั้นผ่านกระบวนการถอดรหัสทำให้เกิดเป็นเสียงขึ้นมาได้ แต่ขอบเขตของวิทยานิพนธ์เล่มนี้เป็นการศึกษากระบวนการเข้ารหัสตามมาตรฐาน MPEG-1/Audio(ISO/IEC 11172-3) ทั้งหมดอย่างคร่าวๆ แต่จะเจาะลึกเฉพาะส่วนของ Filterbank, FFT และส่วนของ Psychoacoustics โดยทำการเขียนโปรแกรมภาษาแอสเซมบลี้ โดยใช้ Motorola DSP56000 Software Simulator Version 6.3.0 เพื่อจำลองการทำงานในส่วนของ Filterbank และ FFT แทนตัวบอร์ด Motorola 56009 EVM และได้เขียนโปรแกรมใน Matlab เพื่อจำลองกระบวนการทำงานในส่วนของ Psychoacoustics เพิ่มเติมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

Up to now, computers are becoming more and more important in our daily lives in the area of working, entertainments, etc. The multimedia, which means sound and images used " Digital Signal Processing" to process any instructions we desired.

A large number of developers gradually try to create lots of methods in order to compress those sound and image data to make them become smaller and smaller in size and significantly, the quality of the compressed data should be the same as the original one. In this report, we are interested in digital audio compression, which enables more efficient storage and transmission of audio data. The many forms of audio compressions techniques offer a range of encoder and decoder complexity, and differing amounts of data compression.

The MPEG/Audio standard is a high complexity, high compression and high audio quality algorithms. We have been studying all the MPEG-1/Audio encoding algorithms thoroughly and attempting to understand DSP56009 Evaluation Board produced by Motorola. Not only have we implemented Filterbank,the first part of the algorithms ,in this semester, but also Fast Fourier Transform(FFT), one of the technics used to map time domain data into frequency domain data, and the last but not least part of our splendid project ,Psychoacoustics, which determine the number of bits for the audio data , are done in the same fashion depicted in the standard as well.

By the way, we use Motorola DSP 56000 Software Simulator Version 6.3.0 instead of Motorola 56009 EVM for the convenience to check the data in the memory,register,etc. In addition, we have also used Matlab to simulate every algorithm stated above to assure that the stuff we have done is absolutely correct

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตของโครงงาน

วัตถุประสงค์ของโครงงาน

ศึกษาและทำความเข้าใจมาตรฐาน MPEG-1/Audio (ISO/IEC 11172-3) เพื่อนำมาพัฒนาโปรแกรมเข้ารหัสสัญญาณเสียงแบบ PCM เพื่อให้ได้เอาท์พุทออกมาเป็นข้อมูลประเภท MPEG-1/Audio (ISO/IEC 11172-3) โดยการประยุกต์ใช้บอร์ด DSP56009EVM เพื่อนำอัลกอริทึ่มที่ศึกษาได้ไปพัฒนาชิป โดยใช้ภาษา VHDL ต่อไป

ขอบเขตของการปฏิบัติงาน

เป็นการศึกษากระบวนการเข้ารหัสตามมาตรฐาน MPEG-1/Audio(ISO/IEC 11172-3) ทั้งหมดอย่างคร่าวๆ แต่จะเจาะลึกเฉพาะส่วนของ Filterbank, FFT และส่วนของ Psychoacoustics โดยทำการเขียนโปรแกรมภาษาแอสแซมบลี้ โดยใช้ Motorola DSP56000 Software Simulator Version 6.3.0 เพื่อจำลองการทำงานในส่วนของ Filterbank และ FFT แทนตัวบอร์ด Motorola 56009 EVM และได้เขียนโปรแกรมใน Matlab เพื่อจำลองกระบวนการทำงานในส่วนของ Psychoacoustics เพิ่มเติมด้วย

ผลที่คาดว่าจะได้รับ

1. เข้าใจว่าทำไมต้องเกิดการบีบอัดข้อมูลตามมาตรฐาน MPEG-1/Audio (ISO/IEC 11172-3) และข้อดีของมัน

2.เข้าใจวิธีการในการเข้ารหัสตามมาตรฐาน MPEG-1/Audio (ISO/IEC 11172-3) ทั้งหมดอย่างคร่าวๆ แต่เข้าใจอย่างค่อนข้างลึกซึ้งในส่วนของฟิลเตอร์แบงก์,FFT และส่วนของ Psychoacoustics

3. ศึกษาในส่วนของ Filterbank, FFT และส่วนของ Psychoacoustics โดยทำการเขียนโปรแกรมภาษาแอสแซมบลี้ โดยใช้ Motorola DSP56000 Software Simulator Version 6.3.0 เพื่อจำลองการทำงานในส่วนของ Filterbank และ FFT แทนตัวบอร์ด Motorola 56009 EVM และได้เขียนโปรแกรมใน Matlab เพื่อจำลองกระบวนการทำงานใน ส่วนของ Psychoacoustics เพิ่มเติมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

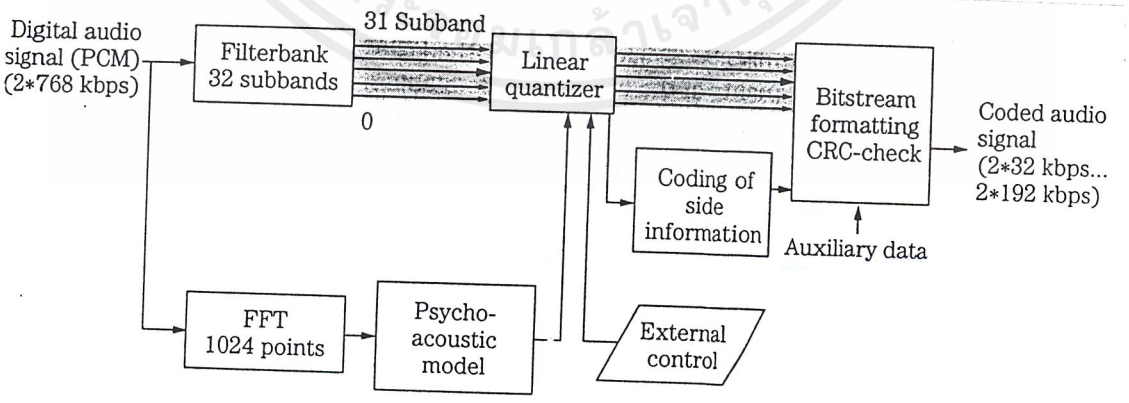
บทที่ 1
บทนำ

การติดต่อสื่อสารผ่านระบบดิจิตอลได้มีบทบาทมากขึ้นในปัจจุบันแต่เนื่องด้วยข้อจำกัดของระบบบางประการเมื่อระบบมีขนาดใหญ่ขึ้น เช่น ความกว้างแถบส่งสัญญาณ (Bandwidth) ของระบบเดิมมีขนาดจำกัด , ปริมาณหน่วยความจำที่จะต้องใช้มากขึ้นเมื่อมีผู้ใช้เพิ่มขึ้น ซึ่งถ้าเราสามารถหาวิธีการที่จะลดปริมาณข้อมูลภายในระบบลงได้นั่นหมายถึงต้นทุนของระบบจะลดลงตามไปด้วย

การบีบอัดข้อมูลเสียงมีความสำคัญมากสำหรับการติดต่อสื่อสารผ่านระบบดิจิตอลในปัจจุบัน ยกตัวอย่างเช่น โทรศัพท์มือถือ เครือข่ายอินเตอร์เน็ต เป็นต้น เนื่องด้วยในการถ่ายโอน (Transfer) ข้อมูลเสียงปริมาณมากๆจำเป็นจะต้องใช้หน่วยความจำและระยะเวลาจำนวนมาก

ปัจจุบันได้มีการนำเอาวิธีการบีบอัดข้อมูลเสียงแบบต่างๆมาใช้เพื่อลดปริมาณหน่วยความจำ โดยที่ความผิดพลาด (Error) ของข้อมูลเสียงนั้นจะต้องอยู่ในระดับที่สามารถยอมรับได้ วิธีการหนึ่งก็คือ การบีบอัดข้อมูลแบบ MPEG-Audio (Motion Picture Expert Group - Audio) ตามมาตรฐาน ISO/IEC 11172-3 โดยนำหลักการของความสามารถทางการได้ยินของมนุษย์(Psychoacoustic) มาเป็นตัวตัดสินใจว่าจะลดข้อมูลส่วนใดลงได้บ้าง

กระบวนการทำงานขอไซโคอะคูสติก (Psychoacoustic) รูปที่ 1.1 แสดงส่วนเริ่มแรกของกระบวนการทำงานทางไซโคอะคูสติก



รูปที่ 1.1 ส่วนเริ่มแรกของกระบวนการทำงานทางไซโคอะคูสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สี่ส่วนเริ่มแรกของตัวเข้ารหัสแบบไซโคอะคูสติกคือ

1. **ฟิลเตอร์แบงก์** (Filterbank)

ฟิลเตอร์แบงก์เป็นการเปลี่ยนแปลงสัญญาณจากโดเมนเวลาเป็นโดเมนความถี่ มีฟิลเตอร์แบงก์อยู่ 2 ชนิดที่ใช้ในกระบวนการทำงานของ ISO/IEC 11172-3 (MPEG – Audio) คือ โพลีเฟสฟิลเตอร์แบงก์ (Polyphase filterbank) และ ไฮบริดจ์โพลีเฟส/MDCT ฟิลเตอร์แบงก์ (Hybrid polyphase/MDCT filterbank) แต่ละชนิดจะมีการเปลี่ยนแปลงสัญญาณโดยเฉพาะของมันเอง ในโดเมนเวลาและโดเมนความถี่ ฟิลเตอร์แบงก์เหล่านี้เป็น critically sampled (ตัวอย่างเช่น มีจำนวนตัวอย่างที่สุ่มออกมาในโดเมนที่วิเคราะห์อยู่เท่ากับในโดเมนเวลา) ฟิลเตอร์แบงก์เหล่านี้จะเป็นการจัดแบ่งความถี่ในขั้นต้นเมื่อใช้ในตัวเข้ารหัส และจะเป็นตัวกรองสำหรับการสร้างสัญญาณขึ้นมาใหม่เมื่อใช้ในตัวถอดรหัส ตัวอย่างที่ออกมาจากฟิลเตอร์แบงก์ จะถูกนำมาควอนไตซ์(Quantize)

2. **แบบจำลองทางไซโคอะคูสติก** (The psychoacoustic model)

แบบจำลองทางไซโคอะคูสอะคูสติกเป็นการคำนวณหาระดับของเสียงรบกวนที่สามารถสังเกตได้สำหรับในแต่ละแถบความถี่ที่เกิดในตัวฟิลเตอร์แบงก์ ระดับของสัญญาณรบกวนเหล่านี้จะถูกใช้ในการจัดสรรทางบิต(Bit allocation) หรือ การจัดสรรทางเสียงรบกวน (noise allocation) เพื่อเป็นตัวกำหนดว่าจะใช้ตัวควอนไตซ์ที่แท้จริง(Actual Quantizers) และระดับของตัวควอนไตซ์(Quantizer levels) จะมีรูปแบบจำลองทางไซโคอะคูสอะคูสติกอยู่ 2 แบบ คือ แบบที่ 1 และแบบที่ 2 ซึ่งแบบที่ 1 นั้นจะใช้ในเลเยอร์ 1 และ เลเยอร์ 2 ส่วนแบบที่ 2 นั้นจะใช้ในเลเยอร์ 3 ไซโคอะคูสอะคูสติกทั้งสองแบบนั้นผลที่ได้ออกมาตอนท้ายสุดก็คืออัตราส่วนระหว่างสัญญาณจริงๆ กับสัญญาณที่ถูกบดบังไว้ (Signal-to-mask ratio--SMR) สำหรับแต่ละแถบความถี่(ในเลเยอร์ 1 และ 2) หรือ สำหรับในกลุ่มของแถบความถี่ (ในเลเยอร์ 3)

3. **การจัดสรรบิตหรือการจัดสรรเสียงรบกวน** (Bit or noise allocation)

ตัวที่ใช้จัดสรร(allocator) จะดูผลทั้งตัวอย่างที่ออกมาจากฟิวเตอร์แบงก์ และผลที่ได้ออกมาจากแบบจำลองทางไซโคอะคูสติก (SMR) แล้วนำผลทั้งคู่นี้มาปรับเปลี่ยนการจัดสรรทางบิต (เลเยอร์ 1 และ 2) หรือ ปรับเปลี่ยนการจัดสรรทางเสียงรบกวน (เลเยอร์ 3) เพื่อที่จะได้ทั้งอัตราการส่งข้อมูลแบบบิตและการบดบัง (masking) ตามความต้องการอย่างพร้อมกันทั้งสองอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวจัดรูปแบบของบิตสตรีม (The Bitstream formatter)

ตัวจัดรูปแบบของบิตสตรีมจะนำเอาเอาท์พุทที่ควอนไตซ์แล้วหลังออกมาจากฟิลเตอร์แบงก์ พร้อมกับค่าการจัดสรรทางบิต(เลเยอร์ 1 และ 2) หรือ การจัดสรรทางเสียงรบกวน(เลเยอร์ 3) และข้อมูลรอบข้างต่างๆ ที่จำเป็นมาใช้ จากข้อมูลทั้งหมดที่กล่าวมาจะนำไปเข้ารหัสและจัดรูปแบบข้อมูลเหล่านั้นในแนวทางที่มีประสิทธิภาพ ซึ่งในกรณีของเลเยอร์ 3 ที่จุดนี้จะนำเอาฮัฟแมนโค้ด (Huffman codes) มาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้พื้นฐานของสัญญาณเสียงดิจิตอล (Digital Audio)

2.1 การสุ่มตัวอย่างสัญญาณเชิงเวลาเต็มหน่วย (Discrete Time Sampling)

การสุ่มตัวอย่างสัญญาณเชิงเวลาเต็มหน่วย เป็นกลไกที่มีความจำเป็นซึ่งได้อธิบายระบบสัญญาณเสียงทางดิจิตอล , เปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล และ เป็นตัวแบ่งแยกสัญญาณดิจิตอลออกจากระบบอนาลอก

การสุ่มตัวอย่างสัญญาณเป็นกระบวนการที่ไม่เกิดการสูญเสีย ถ้าสัญญาณที่เข้ามามีสภาวะที่เหมาะสม เราจะสังเกตได้ว่าสัญญาณที่มีผลตอบสนองทางความถี่จำกัดจะสามารถถูกสุ่มตัวอย่างสัญญาณออกมาโดยปราศจากการสูญเสียของข้อมูลนั้น สัญญาณที่ถูกสุ่มตัวอย่างออกมาจะประกอบด้วยข้อมูลทั้งหมดที่บรรจุอยู่ในสัญญาณเริ่มแรกที่เข้ามา เราสามารถสร้างสัญญาณที่เข้ามาตอนแรกได้ใหม่จากสัญญาณที่ถูกสุ่มตัวอย่างขึ้นมา โดยทั่วๆ ไปแล้ว เราจะสังเกตได้ว่ามีวิธีสำหรับสร้างสัญญาณขึ้นมาใหม่จากขนาดค่าขนาด (Amplitude) ซึ่งเกิดที่จุดซึ่งห่างเท่าๆ กัน ในโดเมนเวลา

ทฤษฎีการสุ่มตัวอย่าง

ทฤษฎีการสุ่มตัวอย่างที่นิยมใช้กันอย่างแพร่หลาย ในวงการวิศวกรรมในปัจจุบันคือ ทฤษฎีของไนควิทซ์ (Nyquist Theorem) ซึ่งทฤษฎีนี้มีเนื้อหาว่า สัญญาณที่มีแถบความถี่จำกัด(Bandlimited) ซึ่งต่อเนื่อง สามารถถูกทดแทนได้ด้วยลำดับที่ถูกสุ่มตัวอย่างออกมาเป็นค่าๆ โดยปราศจากการสูญเสียข้อมูลใดๆ และยังอธิบายด้วยว่าสัญญาณเริ่มแรกที่เข้ามาสามารถสร้างขึ้นมาใหม่จากตัวอย่างที่สุ่มมาได้อย่างไร ยิ่งไปกว่านั้นทฤษฎียังเจาะจงลงไปด้วยว่าความถี่ของการสุ่มตัวอย่าง(Sampling Frequency) ต้องมีค่าอย่างน้อยที่สุด 2 เท่าของความถี่ที่สูงที่สุดของสัญญาณที่นำมาสุ่มตัวอย่าง(Signal Frequency) ซึ่งเขียนเป็นสมการได้คือ

$$Fs \geq 2Fa$$

Fs = ความถี่ของการสุ่มตัวอย่าง (Sampling Frequency)

Fa = ความถี่ที่สูงที่สุดของสัญญาณที่นำมาสุ่มตัวอย่าง(Signal Frequency)

ทฤษฎีการสุ่มตัวอย่างข้างต้นสามารถนำมาใช้กับสัญญาณเสียง สัญญาณเสียงที่เข้ามาจะถูกกรองความถี่ต่ำผ่าน(Lowpass filtered) ดังนั้นมันจะถูกจำกัดแถบความถี่ด้วยผลตอบสนองความถี่ทั้งหมดซึ่งต่ำกว่าความถี่ไนควิทซ์ ในทางอุดมคตินั้นตัวกรองจะถูกออกแบบโดยจะมีเพียงสัญญาณที่ความถี่สูงซึ่งมีค่าสูงกว่าความถี่สูงสุดที่มนุษย์จะสามารถได้ยินนั้นถูกกรองออกไป ตอนนี้สัญญาณก็สามารถที่จะถูกสุ่มซึ่งจะเป็นการกำหนดค่าขนาดของสัญญาณที่จุดนั้นๆ ตัวอย่างที่สุ่มมาได้จาก

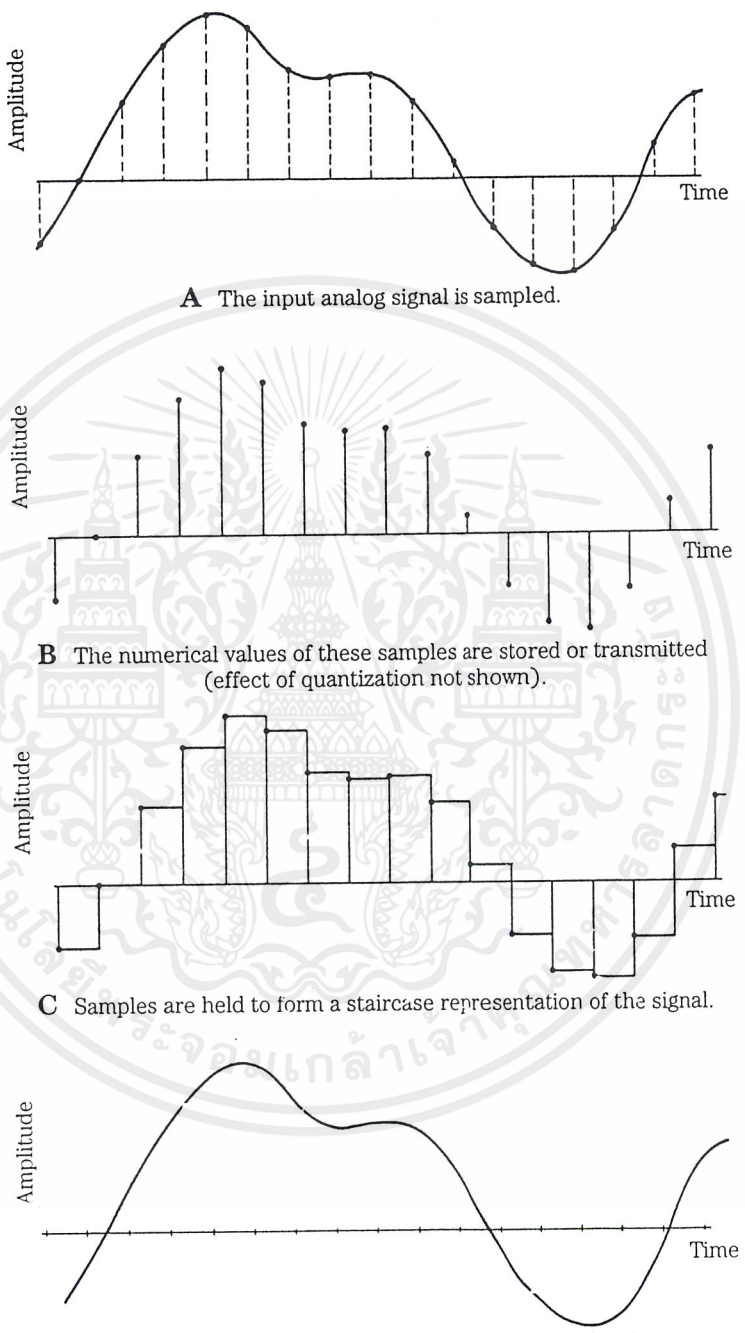
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่มีแถบความถี่จำกัดนี้จะประกอบด้วยข้อมูลเดียวกันกับสัญญาณที่มีแถบความถี่จำกัดซึ่งไม่ได้ถูกสุ่มตัวอย่าง ที่เอาท์พุทของระบบ สัญญาณจะสามารถถูกสร้างขึ้นมาใหม่และจะไม่มีการสูญเสียข้อมูล(จากการสุ่มตัวอย่าง) เมื่อเปรียบเทียบระหว่างสัญญาณเอาท์พุทและสัญญาณอินพุทที่ถูกกรองแล้ว จากคุณสมบัตินี้ของการสุ่มตัวอย่างสัญญาณเอาท์พุทที่ได้ออกมาไม่เป็นการประมาณค่า แต่จะเป็นค่าที่ถูกต้องแน่นอน สัญญาณที่มีแถบความถี่จำกัดจะถูกสร้างอีกครั้งหนึ่งดังแสดงได้ในรูปที่ 2.1

พิจารณาการเปลี่ยนแปลงอย่างต่อเนื่องของสัญญาณอนาลอกนั้นจะเห็นได้ว่าจะถูกสุ่มตัวอย่างเพื่อสร้างอนุกรมของพัลส์ ขนาดของแต่ละพัลส์เมื่อผ่านการ quantized แล้วจะได้ผลออกมาเป็นตัวเลขซึ่งแทนขนาดของสัญญาณในขณะนั้น เพื่อที่จะแก้ไขสถาณการณ์นี้เรากำหนดให้ความถี่ของการสุ่มตัวอย่างเป็นจำนวนของตัวอย่างที่สุ่มได้ต่อวินาที ยกตัวอย่างเช่น ถ้าความถี่ของการสุ่มตัวอย่างคือ 40,000 ตัวอย่างต่อวินาที ซึ่งเหมือนกับการบอกว่าสุ่ม 1 อย่างในเวลา 1/40,000 วินาที การเปลี่ยนรูปแบบสัญญาณอย่างรวดเร็วนั้นจำเป็นต้องการความถี่ในการสุ่มตัวอย่างสัญญาณที่สูงขึ้น ดังนั้นความถี่ในการสุ่มตัวอย่างของระบบทางดิจิตอลเป็นตัวกำหนดขอบเขตความถี่สูงของระบบ การเลือกความถี่ในการสุ่มตัวอย่างนั้นเป็นสิ่งสำคัญที่สุดในการออกแบบระบบทางดิจิตอล เพราะว่ามันจะเป็นตัวช่วยกำหนดแถบความกว้างของความถี่ (Bandwidth) ของระบบ

ถ้าเรากำหนดความถี่ในการสุ่มตัวอย่างเป็น S ตัวอย่างต่อวินาที มันจำเป็นต้องแทนรูปแบบสัญญาณได้อย่างสมบูรณ์ด้วยแถบความกว้างของความถี่ คือ S/2 Hz ในทางกลับกันความถี่ในการสุ่มตัวอย่างต้องมีค่าอย่างน้อยที่สุด คือ 2 เท่าของความถี่ที่สูงที่สุดของสัญญาณเสียงเพื่อที่จะได้เกิดการสุ่มตัวอย่างแบบไม่เกิดการสูญเสีย ตัวอย่างเช่น สัญญาณเสียงจะมีผลตอบสนองทางความถี่ตั้งแต่ 0 Hz ถึง 20 KHz ซึ่งจากทฤษฎีที่กล่าวมาแล้วต้องการความถี่ในการสุ่มตัวอย่างเริ่มต้นที่ 40 KHz จะสังเกตเห็นได้ว่าขอบเขตของทฤษฎีการสุ่มตัวอย่างนั้นจะทำการจำกัดความถี่ของสัญญาณที่เข้ามาไม่ให้มากกว่าครึ่งหนึ่งของความถี่ในการสุ่มตัวอย่าง (ความถี่ไนควิทซ์) ความถี่ของสัญญาณเสียงที่มีความถี่มากกว่าที่กล่าวไปนั้นจะเป็นสาเหตุที่ทำให้เกิดการเอเลียส(aliasing) ขึ้น ซึ่งจะอธิบายความหมายของคำว่าเอเลียสในบทนี้ ตัวกรองความถี่ต่ำผ่านจะถูกใช้ก่อนหน้าวงจรการสุ่มตัวอย่างเพื่อที่จะกำจัดความถี่ที่มีค่ามากกว่าครึ่งหนึ่งของความถี่ในการสุ่มตัวอย่าง ตัวกรองความถี่ต่ำผ่านก็จะถูกติดตั้งไว้ที่เอาท์พุทของระบบสัญญาณเสียงทางดิจิตอลเพื่อที่จะกำจัดความถี่สูงที่ถูกสร้างขึ้นมาจากภายในระบบ ตัวกรองนี้จะทำการสร้างสัญญาณเริ่มต้นที่เข้ามาขึ้นมาใหม่ ซึ่งจะได้กล่าวในรายละเอียดในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ด้วยการสุ่มตัวอย่างเชิงเวลาเต็มหน่วยจะทำให้สัญญาณที่มีแถบความถี่จำกัดสามารถสุ่มตัวอย่างและสร้างขึ้นมาใหม่โดยปราศจากการสูญหายของข้อมูล

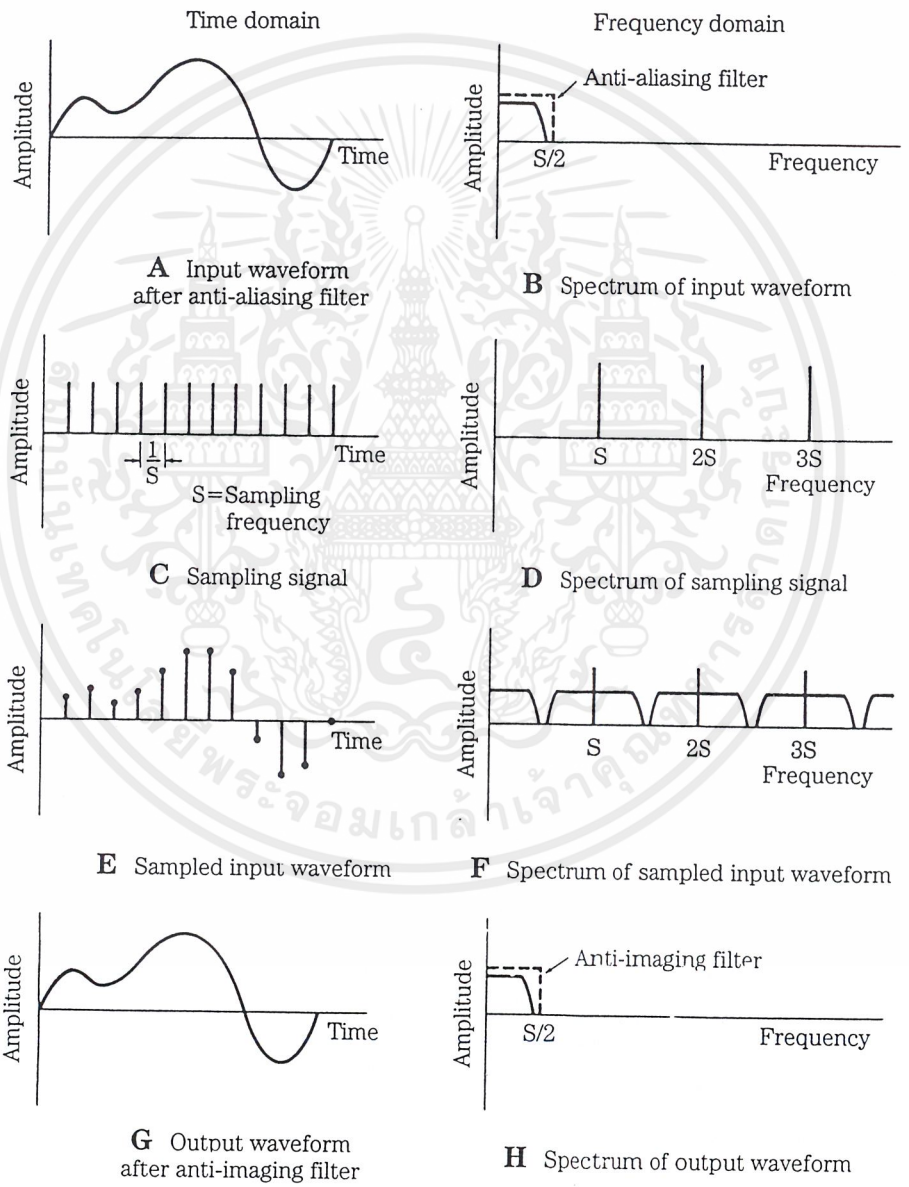
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะสังเกตได้ว่าที่ความถี่ต่ำๆ ของสัญญาณเสียงจะถูกสุ่มตัวอย่างได้อย่างง่ายดาย เพราะที่ความถี่ต่ำจะมีความยาวคลื่นยาวกว่าที่ความถี่สูง จะพบว่ามีตัวอย่างที่ถูกสุ่มออกมามากและพร้อมที่จะแสดงถึงค่าในแต่ละคาบ แต่เมื่อความถี่ของสัญญาณเสียงสูงขึ้นจะพบว่าคาบนั้นสั้นลงและมีจำนวนตัวอย่างที่ถูกสุ่มออกมาต่อคาบน้อยกว่า ท้ายที่สุดจะเห็นได้ว่ากรณีของขอบเขตทางทฤษฎีของ critical sampling ที่ความถี่ของสัญญาณเสียงซึ่งเป็นครึ่งหนึ่งของความถี่ในการสุ่มตัวอย่าง จะพบว่าจะมีเพียง 2 ตัวอย่างต่อคาบ อย่างไรก็ตามถึงแม้ว่าจะมีเพียงแค่ 2 ตัวอย่าง ก็สามารถที่จะแสดงค่าแทนรูปแบบสัญญาณได้ ตัวอย่างเช่น พิจารณากรณีที่ใช้ความถี่ในการสุ่มตัวอย่างคือ 40 KHz และสัญญาณที่เข้ามาเป็นคลื่นรูปซายน์ (sinewave) ที่มีความถี่ 20 KHz ตัวสุ่มตัวอย่างจะสามารถสุ่มตัวอย่างได้ 2 ค่า ซึ่งจะได้ผลคือ คลื่นรูปสี่เหลี่ยมที่มีความถี่ 20 KHz จะพบว่ารูปคลื่นที่ได้มาไม่เหมือนกับรูปคลื่นซายน์ที่เข้ามาแต่แรก อย่างไรก็ตามคลื่นรูปสี่เหลี่ยมที่มีความถี่ 20 KHz นี้จะประกอบด้วยฮาร์โมนิคคี่ (Odd Harmonics) คือ คลื่นรูปซายน์ที่ความถี่ 20 , 60 , 100 , 140 และ 180 KHz ฯลฯ ตัวกรองความถี่ต่ำที่เอาท์พุทของระบบทางสัญญาณเสียงดิจิตอลจะกำจัดความถี่ทั้งหมดที่สูงกว่าครึ่งหนึ่งของความถี่ในการสุ่มตัวอย่างออกไปทั้งหมด เมื่อฮาร์โมนิคสูงถูกกำจัดออกไปหมด เอาท์พุทของระบบจะสร้างคลื่นรูปซายน์ที่มีความถี่ 20 KHz ขึ้นมาใหม่ซึ่งจะมีลักษณะเหมือนกับรูปคลื่นที่เข้ามาตอนเริ่มแรก เรารู้ว่ารูปคลื่นที่ถูกสุ่มตัวอย่างเป็นคลื่นรูปซายน์เพราะว่าตัวกรองความถี่ต่ำผ่านทางด้านอินพุทจะไม่ยอมให้ความถี่สูงของรูปคลื่นผ่านไปถึงตัวสุ่มตัวอย่างได้ ถ้านำเรื่องความสามารถของหูของคนเรามาเกี่ยวข้องด้วย จะพบว่าคลื่นรูปซายน์จะเป็นคลื่นที่เหมาะสมมาก เพราะว่าความถี่ฮาร์โมนิกของคลื่นรูปซายน์ที่มีความถี่ 20 KHz จะสูงกว่าความถี่ที่หูของคนเราสามารถได้ยินได้

กระบวนการสุ่มตัวอย่างและการทำการสร้างสัญญาณขึ้นมาใหม่นั้นแสดงให้เห็นอยู่ในรูป 2.2 สัญญาณที่เกี่ยวข้องในการสุ่มตัวอย่างสัญญาณถูกแสดง ณ ที่จุดต่างๆ กันในกระบวนการ ยิ่งไปกว่านั้น รูปทางซ้ายมือจะแสดงถึงสัญญาณในโดเมนเวลาและส่วนรูปทางขวามือจะแสดงถึงสัญญาณที่เหมือนกันกับทางซ้ายมือแต่อยู่ในโดเมนความถี่ ในทางกลับกันเราสามารถสังเกตขนาดของสัญญาณเหนือแกนเวลา และสามารถสังเกตผลตอบสนองทางความถี่ของมันได้ เราจะสังเกตได้ว่าในรูป 2.2a และ 2.2b นั้นสัญญาณเสียงที่เข้ามาต้องถูกจำกัดแถบความถี่ที่ครึ่งหนึ่งของความถี่สุ่มตัวอย่าง (S/2) โดยใช้ตัวกรองที่ต่อต้านการเกิดเอเลียส ตัวกรองนี้จะกำจัดความถี่ทั้งหมดที่สูงกว่าความถี่ไนควิทซ์ทิ้งไป สัญญาณที่ถูกสุ่มตัวอย่างในรูปที่ 2.2c และ 2.2d จะเกิดขึ้นซ้ำๆ กันที่ความถี่ในการสุ่มตัวอย่าง S และสเปกตรัม (Spectrum) ของมันจะประกอบด้วยพัลส์ที่หลายๆ ความถี่ในการสุ่มตัวอย่าง เช่น S , 2S , 3S , ฯลฯ เมื่อสัญญาณเสียงถูกสุ่มตัวอย่างดังแสดงในรูปที่ 2.2e และ 2.2f ขนาดของสัญญาณ ณ ที่เวลาที่ถูกสุ่มตัวอย่างจะถูกสำรองไว้ อย่างไรก็ตามสัญญาณที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุ่มตัวอย่างมานี้จะประกอบด้วยลักษณะของรูปแบบของสเปกตรัมดั้งเดิมตั้งแต่ตอนแรก ซึ่งมีศูนย์กลางอยู่ที่ความถี่ในการสุ่มตัวอย่างหลายๆ ความถี่ เพื่อที่จะนำสัญญาณที่สุ่มตัวอย่างมากลับนำมาสร้างใหม่ ดังเช่นแสดงในรูปที่ 2.2g และ 2.2h ตัวอย่างถูกทำให้ผ่านตัวกรองความถี่ต่ำผ่านซึ่งมีคุณสมบัติในการต่อต้านรูปร่างที่ไม่ถูกต้อง(anti -imaging) เพื่อที่จะกำจัดรูปสัญญาณทั้งหมดที่อยู่เหนือความถี่ S/2



รูปที่ 2.2 โดเมนเวลา(ด้านซ้าย) และโดเมนความถี่(ด้านขวา) แสดงผลของการแซมปลิ้งสัญญาณที่มีแถบความถี่จำกัด และ การสร้างขึ้นมาใหม่(reconstruct)

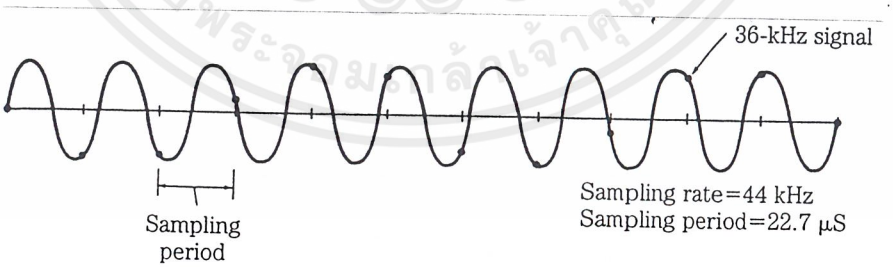
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Aliasing (**การซ้อนทับกันของสัญญาณ**)

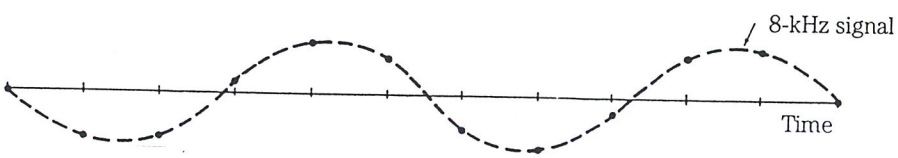
สิ่งที่เป็นเรื่องท้าทายสำหรับนักออกแบบระบบ Digital audio ก็คือ การซ้อนทับของสัญญาณ ซึ่งเป็นเรื่องที่สับสนเป็นอย่างมากในการสุ่มสัญญาณ การซ้อนทับกันของสัญญาณนี้จะเป็นผลทำให้องค์ประกอบของสัญญาณผิดเพี้ยนไปได้ สัญญาณที่ผิดพลาดนี้จะออกมาทางเอาท์พุท ทำให้เป็นการยากในการแยกแยะสัญญาณให้ถูกต้องตามแบบตัวต้นแบบ จึงควรคำนึงถึง สำหรับนักออกแบบในการป้องกันไม่ให้เกิดเหตุการณ์นี้ขึ้น

จากการสังเกต เราทราบว่าการสุ่มสัญญาณคือกระบวนการที่ไม่เกิดการสูญเสีย แต่สิ่งที่สำคัญที่สุด คือ อินพุทจะต้องมีช่วงความถี่ค่าหนึ่ง นั่นคือ จะต้องมีตัวกรองความถี่ต่ำไว้เป็นตัวกำหนดช่วงความถี่ การซ้อนทับกันของสัญญาณเป็นเรื่องที่ละเมิดกฎของการสุ่มสัญญาณ สัญญาณอนาลอกที่เราต้องการทำการสุ่มจะต้องเท่ากับ หรือ น้อยกว่าครึ่งหนึ่งของความถี่สุ่มสัญญาณ (sampling frequency) ถ้าเกิดสัญญาณอนาลอกตัวนั้นมีค่าความถี่มากกว่าครึ่งหนึ่งของความถี่สุ่มสัญญาณ จะเกิดการซ้อนทับของสัญญาณขึ้นได้ ถ้าความถี่ของสัญญาณอนาลอกเพิ่มขึ้น จำนวนจุดที่เราจะสุ่มออกมาได้ก็จะลดลง ถ้าถึงความถี่ Nyquist เมื่อไหร่ ก็จะทำให้มี 2 แซมเปิ้ลใน 1 คาบ ถ้าเรายังสุ่มสัญญาณที่ความถี่สูงกว่าความถี่ Nyquist ขึ้นไปอีก ก็จะทำให้ตัวอย่างที่ได้จากการสุ่มเกิดข้อผิดพลาดได้มากขึ้นๆไปอีก

พิจารณาในระบบดิจิตอล ที่ใช้ความถี่ในการสุ่มสัญญาณที่ 44 Khz และคิดว่าสัญญาณที่ความถี่ 36 Khz ได้ถูกส่งเข้าไป ดังรูป 2.3



A A 36-kHz waveform is sampled at 44 kHz.



B Upon reconstruction, the 36-kHz signal is filtered out, leaving an aliased 8-kHz signal.

รูปที่ 2.3 สัญญาณอินพุทซึ่งเกินครึ่งหนึ่งความถี่สุ่มจะเกิดการซ้อนทับของสัญญาณที่ความถี่ต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบมูลฐานของการซ้อนทับจะได้ว่า 44-36 = 8 Khz เห็นได้ว่าจะได้เอาท์พุทที่ไม่ถูกต้อง ทำให้เกิดการสับสนขึ้นได้ว่า สัญญาณที่เราป้อนเข้าไปเป็นความถี่ที่ 36 Kz หรือ 8 Khz ยิ่งไปกว่านั้นต้องคำนึงว่าที่ตัวกรองความถี่ต่ำ ที่ทางด้านเอาท์พุทของระบบดิจิตอล ซึ่งทำให้ฟังก์ชั่นขั้นบันไดเรียบขึ้น เพื่อจะให้สามารถสร้างออกมาให้เหมือนอินพุท และความถี่คัทออฟของตัวกรอง คือ ครึ่งหนึ่งของความถี่สุ่มสัญญาณ ดังนั้นความถี่อื่นๆที่เกินความถี่สุ่มก็จะถูกกำจัดออกไป ในกรณีนี้ ความถี่ที่ 36 Khz ก็จะถูกกำจัดออกไปด้วย แต่ที่ความถี่ 8 Khz หรือก็คือความถี่ซ้อนทับยังคงอยู่เหมือนเดิม นั่นคือ จะได้ความถี่ที่เราไม่ต้องการ ส่งผลให้สัญญาณอินพุทเกิดการลดทอน (distortion) ขึ้นอีก ดังรูป 2.3

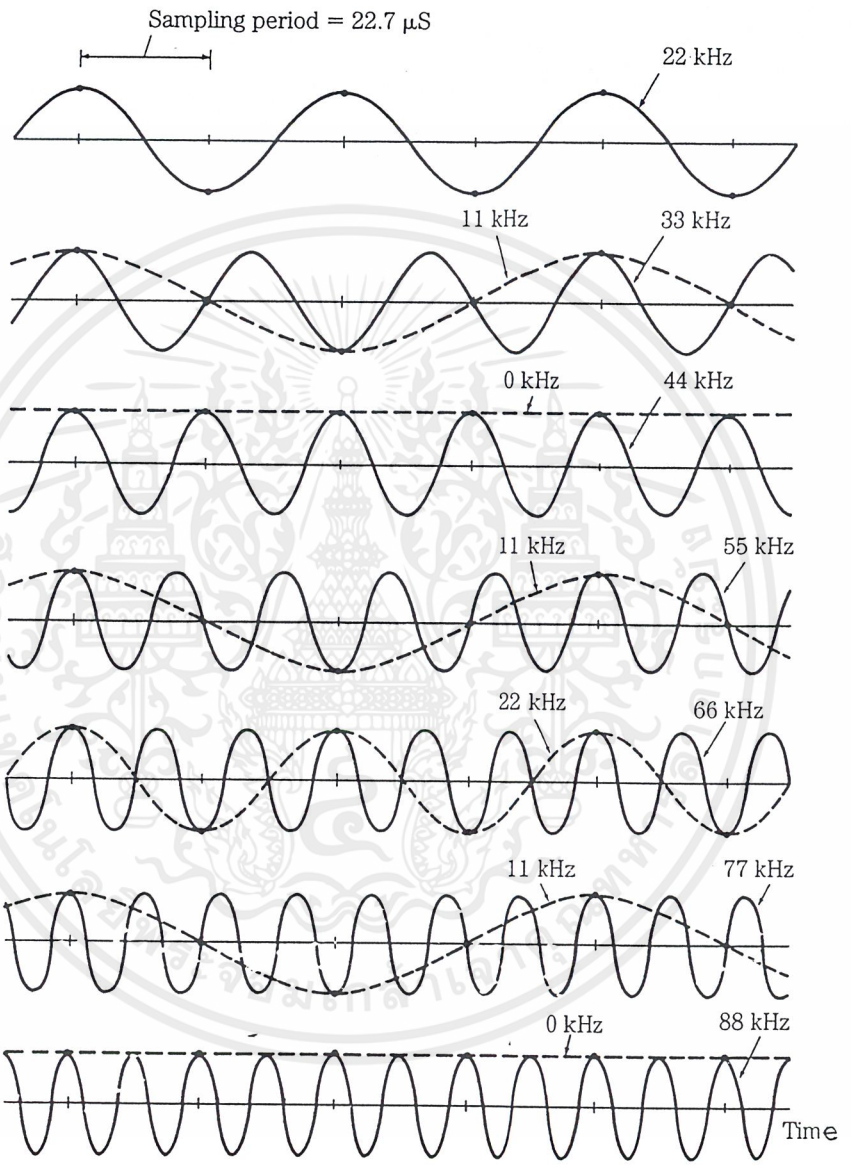
ยังมีวิธีแสดงการเกิดการซ้อนทับของสัญญาณอยู่หลายวิธี ถึงแม้ว่าจะเห็นแต่ความถี่ 8 Khz (ดังเช่นตัวอย่างที่ยกในตอนต้น) ปรากฏอยู่ในแถบความถี่ของสัญญาณอนาลอก ต้องพึงระวังเสมอว่าการซ้อนทับจะเกิดขึ้นไม่ว่าความถี่ของการสุ่มจะสูงเท่าไหร่ก็ตาม ลองพิจารณาให้ความถี่ในการสุ่มสัญญาณมีค่าเท่ากับ 44 Khz อินพุทที่ความถี่ตั้งแต่ 0-22 Khz จะให้ผลลัพธ์ที่ถูกต้อง แต่ถ้าความถี่อินพุทที่มีค่าตั้งแต่ 22-44 Khz ก็จะเกิดการซ้อนทับของสัญญาณจาก 22 Khz ถึง 0 Hz รูปที่ 2.4 แสดงผลของการเกิดการซ้อนทับจาก 0-22 Khz เมื่อเพิ่มความถี่อินพุท

การป้องกันการซ้อนทับของสัญญาณ(Aliasing Prevention)

ในทางปฏิบัติ ปัญหาของผลการซ้อนทับของสัญญาณจะสามารถถูกแก้ไขได้ ตามความเป็นจริงแล้ว ระบบการบันทึกทางด้านดิจิตอลที่ถูกออกแบบให้มีคุณภาพที่ดีแล้ว การซ้อนทับกันก็จะไม่เกิดขึ้น ผลลัพธ์ที่ได้ก็จะตรงไปตรงมา คือ สัญญาณอินพุทจะถูกจำกัดย่านความถี่ด้วยตัวกรองความถี่ต่ำที่คม ที่ให้ได้การลดทอนที่มีนัยสำคัญที่ความถี่ Nyquist เพื่อให้แน่ใจว่าสัญญาณที่จะทำการสุ่มไม่เกินความถี่ Nyquist ตัวกรองทางอุดมคติ จะต้องมีการลดทอนที่มีค่าอนันต์และมีความไวในการเปลี่ยนสถานะจากแถบผ่านเป็นแถบหยุดมากๆ อย่างไรก็ตามในทางปฏิบัติ ตัวกรองไม่สามารถออกแบบให้เป็นไปตามอุดมคติได้ ส่วนมากเราจะออกแบบให้แถบเปลี่ยนสถานะ (transition band) มีค่าการลดทอนที่ยอมรับได้

เป็นสิ่งที่สำคัญมาก ที่จะต้องตรวจสอบทฤษฎีการสุ่มสัญญาณ และตัวกรองความถี่ต่ำ ของสัญญาณอินพุทที่เข้ามาในระบบดิจิตอล ถ้าการซ้อนทับของสัญญาณเกิดขึ้น จะไม่มีเทคนิคใดที่จะเอาความถี่ซ้อนทับออกไปจากสัญญาณอนาลอกอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ความถี่ซ้อนทับ (เส้นประ) ในแถบสัญญาณออดิโอซึ่งแปรค่าตั้งแต่ 0 ถึง 22Khz เมื่อความถี่อินพุท(เส้นทึบ) เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Quantization

การวัดเหตุการณ์ใดๆก็ตามที่มีการเปลี่ยนของเรื่องเวลามาเกี่ยวข้อง จะมีความหมายถ้าทั้งค่าและเวลาของสิ่งที่เราวัดสามารถนำมาเก็บไว้ได้ การสุ่มสัญญาณจะแสดงเวลาของการวัด และการควอนไตซ์จะแสดงค่าของการวัด หรือในกรณีของสัญญาณอนาลอก ก็คือ ขนาดของแอมปลิจูดของรูปคลื่นนั่นเอง การสุ่มและการควอนไตซ์เป็นองค์ประกอบพื้นฐานของระบบดิจิตอล และทั้งการสุ่มและการควอนไตซ์จะเป็นตัวตัดสินความกว้างของสัญญาณและความละเอียด รูปคลื่นอนาลอกสามารถแสดงออกมาในแบบอนุกรมของพัลส์ แอมปลิจูดของแต่ละพัลส์จะได้ค่าที่แสดงค่าของสัญญาณอนาลอก ณ ช่วงเวลานั้น การควอนไตซ์นั้นความแม่นยำจะถูกจำกัดจากความละเอียดของระบบ เพราะว่าความยาวของค่านั้นๆมีอยู่จำกัด ความละเอียดของระบบสัญญาณดิจิตอลออดิโอก็จะถูกจำกัดด้วยทำให้เกิดความคลาดเคลื่อนในการวัดขึ้น โดยความคลาดเคลื่อนที่เกิดขึ้นนี้โดยปกติจะหมายถึงสัญญาณรบกวน (Noise) ในระบบสัญญาณอนาลอกนั่นเอง

ถ้าเราควอนไตซ์อย่างมีรูปแบบ (Uniform quantize) แอมปลิจูดของสัญญาณอนาลอกจะถูกแมปไปเป็นจำนวนของค่าที่ถูกควอนไตซ์ที่มีค่าเท่ากับความสูงของสัญญาณนั้นๆ การจะแสดงผลที่มีคุณภาพสูงจะต้องใช้ระดับของเลขที่มีค่ามาก อย่างเช่น สัญญาณเสียงที่มีคุณภาพสูง จะต้องการ 65536 ระดับแอมปลิจูด หรือมากกว่า อย่างไรก็ตาม มีเพียงแค่ 2-3 ระดับเท่านั้นที่จะมีข้อมูลอยู่ อย่างเช่น 2 ระดับแอมปลิจูดจะมีข้อมูลของเสียงพูดอยู่

พิจารณาโวลต์มิเตอร์ 2 อัน อันหนึ่งเป็นแบบอนาลอก อีกอันเป็นแบบดิจิตอล แต่ละอันจะวัดแรงดัน ซึ่งก็คือสัญญาณอินพุท ถ้าให้เป็นมิเตอร์ที่มีคุณภาพสูง และสังเกตค่าอย่างละเอียด เราจะอ่านค่าอนาลอกมิเตอร์ได้เป็น 1.27 โวลต์ แต่ดิจิตอลโวลต์มิเตอร์ที่มีแค่ 2 หลัก จะอ่านได้ 1.3 โวลต์ ถ้าเป็น 3 หลักได้ 1.27 โวลต์ ถ้าเป็น 4 หลักได้ 1.247 โวลต์ ทั้งอนาลอกและดิจิตอลมิเตอร์จะมีข้อผิดพลาดเกิดขึ้น ข้อผิดพลาดของมิเตอร์ทางอนาลอกก็คือ ผลของอุปกรณ์และความยากในการอ่าน ภายในสภาวะทางอุดมคติ ความละเอียดของการวัดเชิงอนาลอก จะถูกจำกัดด้วยสัญญาณรบกวนของมิเตอร์เอง

ถ้าทำการวัดด้วยดิจิตอลมิเตอร์ ผลของความคลาดเคลื่อนจะแตกต่างกัน ความแม่นยำจะถูกจำกัดโดยความละเอียดของตัวมิเตอร์ นั่นคือ ขึ้นอยู่กับจำนวนหลักที่แสดง ถ้ายิ่งหลักมาก ก็จะมีความแม่นยำมากด้วย ถ้าหลักสุดท้าย จะถูกปัดให้สัมพัทธ์กับค่าจริง เช่น 1.27 จะถูกปัดเป็น 1.3

การควอนไตซ์ก็คือ เทคนิคการวัดสัญญาณอนาลอก ให้เป็นค่าทางตัวเลข ระบบทางดิจิตอลจะใช้เป็นแบบเลขฐานสอง จำนวนของค่าที่เป็นไปได้ จะถูกตัดสินโดยความยาวของข้อมูลที่เป็นเลขฐานสองนั้น นั่นคือ จำนวนของบิตที่จะประกอบด้วยสัญญาณเดิม ในทางปฏิบัติ ความละเอียดจะเป็นอิทธิพลมาจากคุณภาพของ A/D (analog to digital converter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสุ่มสัญญาณช่วงจำกัดทางทฤษฎี จะเป็นกระบวนการที่ไม่เกิดการสูญเสีย แต่การเลือกค่าของแอมปลิจูดที่ความถี่สุ่มใดๆ ต้องคำนึงถึงเสมอว่า ค่าที่ได้มาจากการควอนไตซ์ จะเป็นเพียงแค่ค่าประมาณของสัญญาณจริงเท่านั้น

2.4 Pulse Code Modulation

ในทางทฤษฎี มีเทคนิคมากมายที่ใช้ในการถอดรหัสอนาลอกเป็นสัญญาณดิจิตอล จะเหมือนกันในกระบวนการที่จะเปลี่ยนสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอล แต่ทางปฏิบัติ แต่ละวิธีจะต่างกันในด้านคุณภาพทางด้านแบนด์วิธ และ S/N Ratio การมอดดูเลทก็จะไม่มีอะไรไปมากกว่าการถอดรหัสของข้อมูล สำหรับจุดประสงค์ในการส่งหรือเก็บข้อมูล เทคนิคต่างๆ เช่น AM,FM ได้ถูกใช้มาเป็นระยะเวลายาวนานในการมอดดูเลทความถี่พาหะ กับข้อมูลทางอนาลอกในการกระจายเสียงของวิทยุ ดังจะยกตัวอย่างในรูป 2.4

เมื่ออยู่ในระหว่างส่งผ่านข้อมูล มีพัลส์มอดดูเลชั่นอยู่หลายชนิดด้วยกัน เช่น pulse width หรือ positon in frequency ในช่วงเวลาใดๆ จะแสดงแอมปลิจูดของสัญญาณ ณ ช่วงเวลาใดๆ pulse width modulation ซึ่งเป็นตัวอย่างที่เกิดขึ้นมาก่อนส่วน pulse position modulation ได้เกิดขึ้นมาทีหลัง ซึ่งทั้งสองกรณี แอมปลิจูดของสัญญาณต้นแบบจะถูกโค้ดและถูกส่งผ่านในรูปของพัลส์ที่มีแอมปลิจูดคงที่ แอมปลิจูดของสัญญาณสามารถจะถูกส่งผ่านโดยตรงโดย modulation pulse,pulse-amplitude modulation(PAM) ก็จะเป็นอีกตัวอย่างหนึ่ง ซึ่งวิธีนี้แอมปลิจูดของพัลส์แต่ละลูก จะเท่ากับแอมปลิจูดของสัญญาณ ณ ที่ความถี่สุ่มใดๆ PWM,PPM และ PAM ได้ถูกแสดงดังรูป 2.5a ถึง d ถึงแม้จะใช้วิธีที่กล่าวมาข้าวต้นบ่อยครั้งก็ตามในการเปลี่ยนรูปแบบของสัญญาณ แต่ก็ยังไม่เหมาะสมที่จะส่ง หรือ บันทึกสัญญาณเพราะ ผลของข้อจำกัดทางด้านความคลาดเคลื่อน หรือความจำกัดของช่วงสัญญาณ

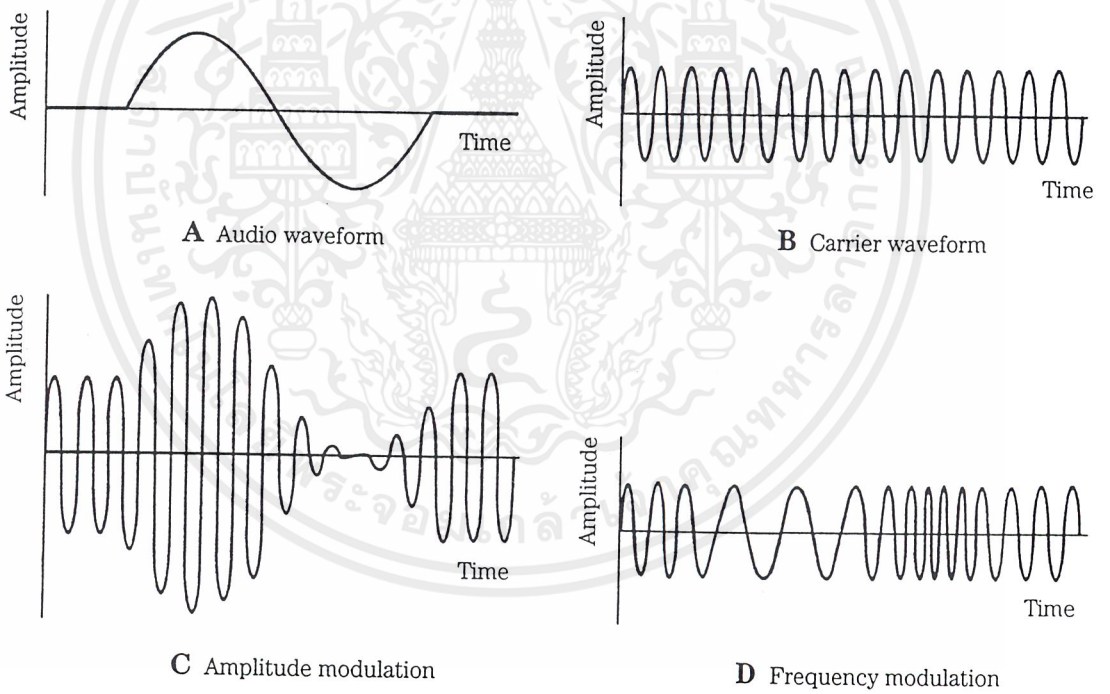
กรรมวิธีที่นิยมในการใช้มากที่สุดในการมอดดูเลทสัญญาณก็คือpulse-code modulation(PCM) เสนอโดย นาย Alec Reeves ในรูปของ PCM สัญญาณอินพุทจะต้องผ่านการสุ่มสัญญาณ ควอนไตซ์ และเข้ารหัส โดยการแสดงแอมปลิจูดของสัญญาณอนาลอกในรูปของความกว้างของพัลส์ เลขฐานสองสามารถที่จะแสดงค่าของแอมปลิจูด ที่ตัวรับ รหัสที่อยู่ในรูปของพัลส์จะถูกใช้ในการสร้างสัญญาณอนาลอกขึ้นมาใหม่ เลขไบนารี่ซึ่งแสดงแอมปลิจูดของสัญญาณที่เราสุ่มจะถูกโค้ดให้อยู่ในแบบ PCM ดังแสดงในรูป 2.5g

โดยวิธีการแบบ PWM, PPM และ PAM เราต้องการแค่พัลส์เดียวในการแสดงค่าของแอมปลิจูด แต่ PCM จะต้องการหลายพัลส์ ดังนั้น PCM จึงต้องการช่องสัญญาณที่มีช่องความถี่กว้างขึ้น นอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

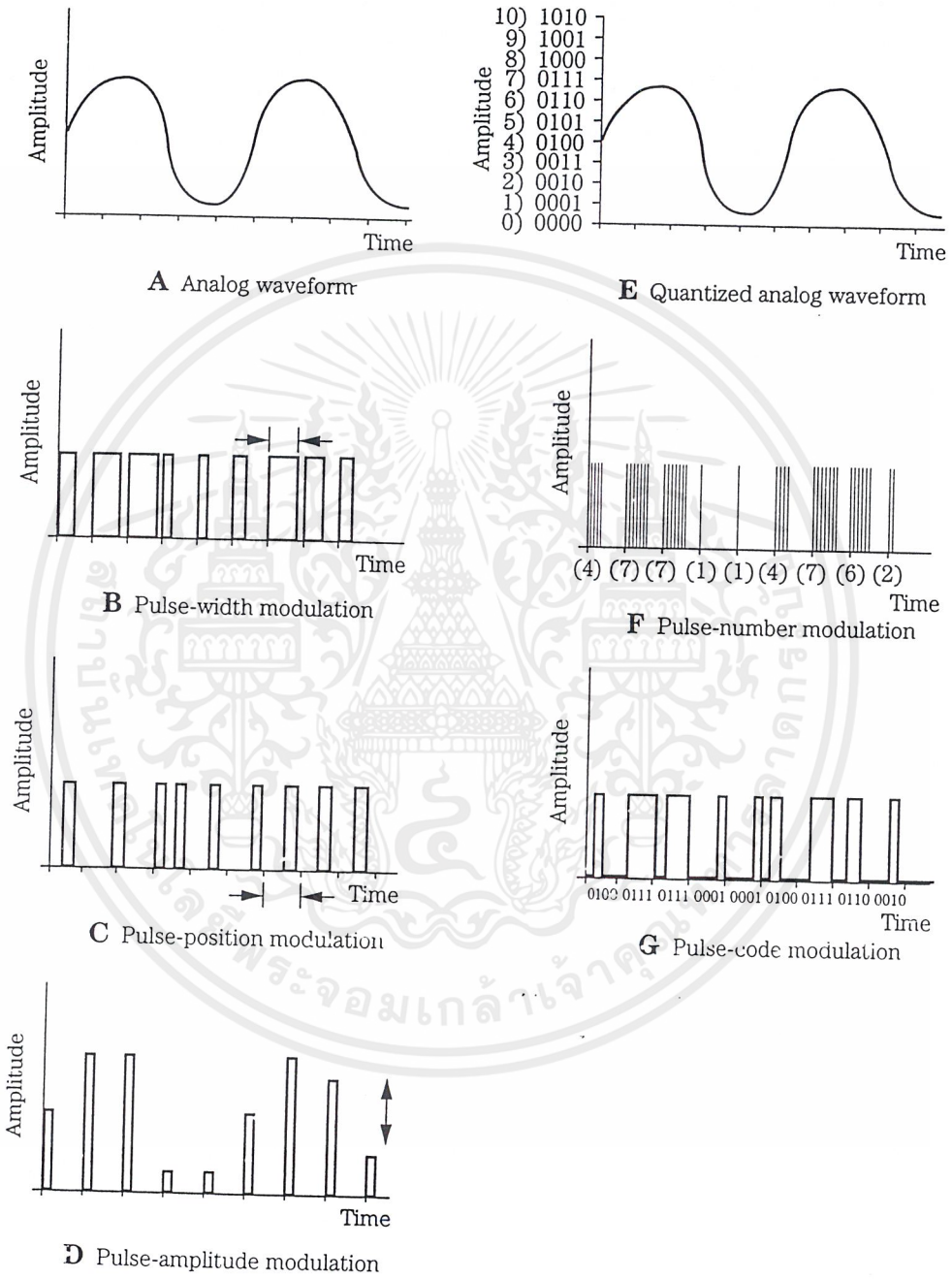
จากนั้นสัญญาณ PCM จะเป็นวิธีที่ไม่เกิดการสูญเสีย ดังนั้นคุณภาพของการส่งสัญญาณแบบ PCM จึงขึ้นกับคุณภาพของการสุ่มสัญญาณ และกระบวนการควอนไตซ์

รูปแบบของระบบแบบ PCM ก่อนอื่นจะมีคลื่นอนาลอกผ่านตัวกรอง และสุ่มสัญญาณ แล้วแอมปลิจูดที่ได้ ถูกควอนไตซ์โดย analog to digital converter เลขฐานสองจะเป็นตัวแสดงอนุกรมของพัลส์ที่ถูกมอดดูเลทโดยเป็นตัวแสดงแอมปลิจูดของรูปคลื่นที่เวลาการสุ่มใดๆ ในส่วนของเครื่องเล่น ข้อมูลจะถูกดีมอดดูเลท ถอดรหัสและแก้ไขข้อผิดพลาดเพื่อจะให้ได้ตรงตามสัญญาณเดิมมากที่สุด รูปคลื่นอนาลอกจะถูกทำให้กลับคืนมาเมื่อผ่าน digital to analog converter และตัวกรองความถี่ต่ำ



รูปที่ 2.5 amplitude modulation และ frequency modulation จะเป็นตัวอย่างของการมอดดูเลทที่ง่ายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 PWM,PPM,PAM เป็นตัวอย่างของ pulse-parameter modulation ส่วน PNM และ PCM เป็นตัวอย่างของ numerical pulse parameter modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การบีบอัดข้อมูลด้วยวิธีการแบบ MPEG-Audio

การบีบอัดสัญญาณอนาลอกเชิงดิจิตอลจะสามารถทำให้เกิดการเก็บข้อมูลและส่งข้อมูลได้อย่างมีประสิทธิภาพ เทคนิคการบีบอัดจะขึ้นอยู่กับระดับความซับซ้อน,คุณภาพของสัญญาณ และจำนวนของข้อมูล การแสดงผลเป็นสัญญาณดิจิตอลของข้อมูลที่เป็นอนาลอกนั้นมีประโยชน์ด้วยกันอยู่หลายประการ เช่น ลดผลของสัญญาณรบกวน มีเสถียรภาพมากขึ้น เป็นต้น สัญญาณอนาลอกที่อยู่ในรูปดิจิตอลสามารถทำให้เกิดการสะดวกในการผสมสัญญาณ,กรองสัญญาณ ผ่านดิจิตอลคอมพิวเตอร์ได้มาก

การเปลี่ยนจากสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอลเริ่มจากสุ่มสัญญาณอินพุทและและทำการเปลี่ยนระดับของสัญญาณที่ทำการสุ่มให้อยู่ในแบบของดิจิตอล โดยจะประกอบไปด้วยเลขฐานสองซึ่งเป็นตัวแสดงระดับของสัญญาณที่เราทำการสุ่ม วิธีการในการแสดงผลของสัญญาณแต่ละสัญญาณที่เราทำการสุ่มจะอยู่ในแบบ PCM (Pulse code modulation)

จากทฤษฎีของ Nyquist กล่าวไว้ว่า ช่วงเวลาของสัญญาณที่ถูกสุ่มอย่างน้อยต้องไม่น้อยไปกว่าครึ่งหนึ่งของอัตราที่เราใช้ในการสุ่มสัญญาณ (Sampling rate) อัตราการสุ่มสัญญาณโดยทั่วๆไปจะอยู่ในช่วง 8 KHz ถึง 48 Khz โดยในช่วง ความถี่สุ่มที่8 Khz จะคลุมความถี่อย่างมากสุดคือ 4 Khz ซึ่งก็จะเป็นช่วงความถี่เสียงที่มนุษย์เราสร้างขึ้น ส่วนช่วง 48 Khz จะคลุมความถี่เสียงทั้งหมดโดยอยู่ในช่วงประมาณ 20 Khz

3.1 การบีบอัดสัญญาณอนาลอกแบบ MPEG

คุณสมบัติเด่นของการบีบอัดแบบนี้คือ การนำวิธี Masking มาใช้ ซึ่งเป็นความสามารถที่น้อยสุดของหูคนเราโดยจะเกิดขึ้นเมื่อสัญญาณอนาลอกที่ดังกว่าไปข่มสัญญาณอนาลอกที่เบากว่าทำให้ไม่สามารถได้ยินเสียงที่เบากว่าอันนั้น ปรากฏการณ์เช่นนี้ได้ถูกสังเกตและนำมาใช้ควบคู่กับการทดลอง Pschoacoustic

จากผลการสังเกตทำให้เราทราบว่าหูคนเรามีย่านความถี่ที่ได้ยินจำกัดโดยจะเปลี่ยนไปตั้งแต่น้อยกว่า 100 Hz สำหรับความถี่น้อยที่สุดที่สามารถได้ยินได้ไปจนถึงมากกว่า 4 Hz ซึ่งเป็นความถี่มากสุด ดังนั้นสเปกตรัมของสัญญาณสามารถที่จะถูกแบ่งเป็นแถบวิกฤต (critical bands) ซึ่งเปรียบเสมือนผลสะท้อนของพลังงานโดยอยู่ในรูปของฟังก์ชั่นความถี่ สามารถดูได้จากตาราง Critical Band Boundaries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เนื่องจากความจำกัดของหูมนุษย์ในการได้ยินเสียง การเริ่มต้นของ masking ที่ความถี่ใดๆ จะขึ้นกับรูปแบบของสัญญาณภายในแถบวิกฤตของความถี่นั้นส่วนการบีบอัดสัญญาณอนาลอกจากคุณสมบัติดังกล่าวสามารถนำมาใช้ให้เป็นประโยชน์โดยการเปลี่ยนจากโดเมนเวลาเป็นโดเมนความถี่ จากนั้นก็ทำการแบ่งสเปกตรัมให้อยู่ในแถบย่อย (subband) โดยพิจารณาควบคู่กับแถบวิกฤตจากนั้นก็ quantize แต่ละแถบความถี่ด้วยวิธีการของ Psychoacoustic เพื่อการบีบอัดที่มีประสิทธิภาพมากที่สุดแต่ละแถบความถี่ควรจะถูก quantize ให้ไม่มากไปกว่าความจำเป็นเพื่อจะทำให้ไม่เกิดผลของ quantization noise

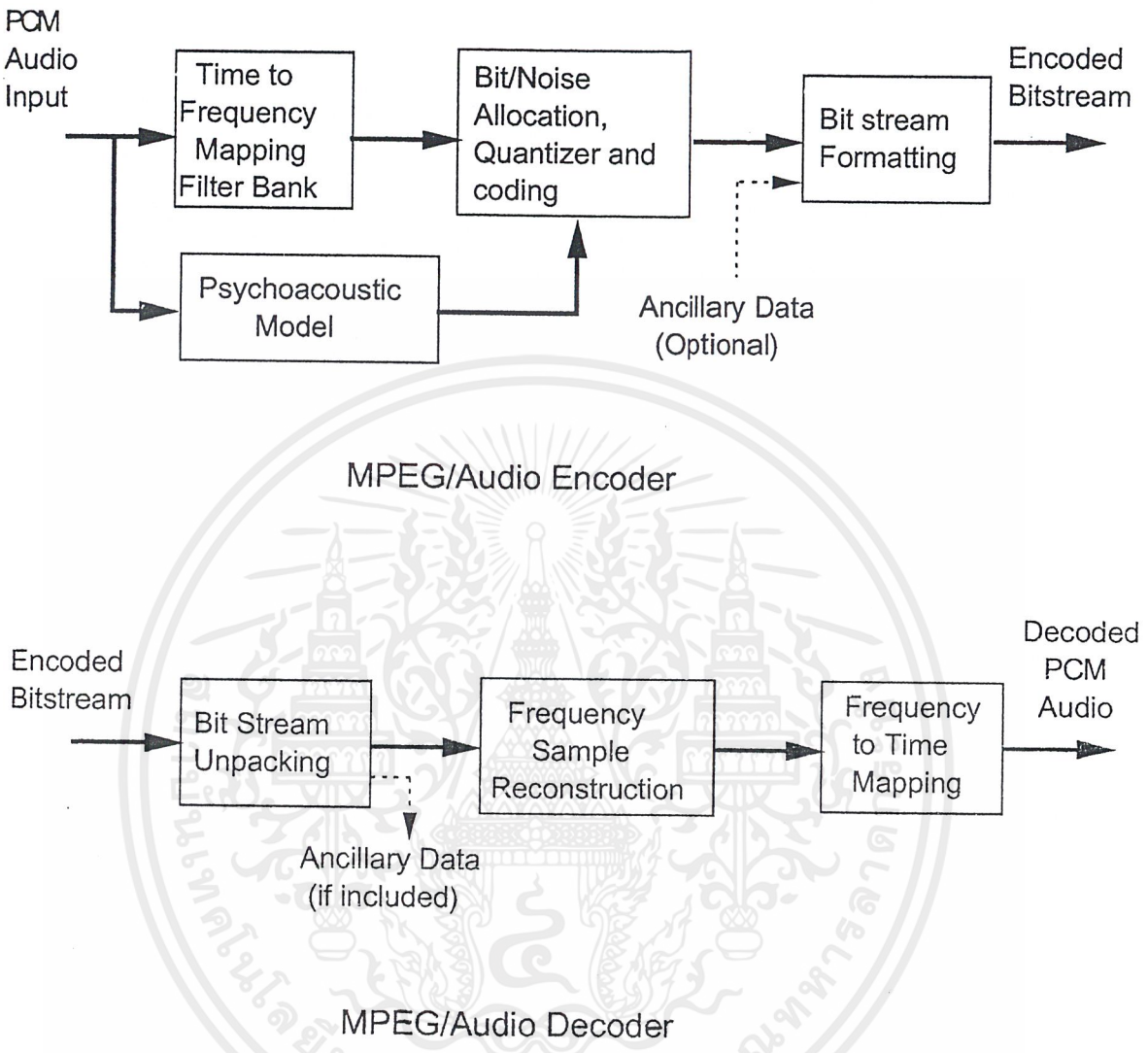
Mpeg/Audio Encoding and Decoding

รูป 3.1 แสดง block diagram ของ Mpeg/audio encoder and decoder หลักการก็คือเมื่อมีสัญญาณอนาลอกจะนำมาผ่าน filter bank โดยจะทำการแบ่งอินพุทให้เป็นแถบย่อย(subband) ในขณะเดียวกันอินพุทก็เข้ากระบวนการ Psychoacoustic โดยจะตัดสินใจ signal to mask ratio ของแต่ละแถบย่อย ส่วนการจัดเรียงบิต จะใช้ signal to mask ratio ตัดสินเพื่อแบ่งจำนวนบิตทั้งหมดในการ quantize ของแต่ละแถบย่อยเพื่อลดผลของ quantization noise บล็อกสุดท้ายจะแสดงสัญญาณอนาลอกที่ถูกquantize และจัดเรียงรูปแบบข้อมูลเพื่อจะนำไปใช้สำหรับการเข้ารหัส ซึ่งการเข้ารหัสก็จะเป็นการกลับกระบวนการทำงานกับแบบถอดรหัส

มาตรฐานของการบีบอัดในแบบ mpeg จำแนกออกได้เป็น 3 เลเยอร์ โดยเลเยอร์ที่1 จะเป็นรูปแบบที่ง่ายที่สุด ส่วนเลเยอร์ที่2 และ 3 จะซับซ้อนขึ้นมาอีกและบางวิธีการยังต้องใช้ผลของเลเยอร์ 1 ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42289



รูปที่ 3.1 mpeg/audio compression and decompression

Layer 1

รูปแบบวิธีการจะใช้ Filter bank เหมือนกันกับอีก 2 แบบ โดย filter bank นี้จะแบ่งสัญญาณอนาลอกออกเป็น 32 แถบความถี่ที่มีความกว้างเท่าๆกัน filter นี้จะสามารถสร้างได้ง่าย และให้ความละเอียดทางเวลาได้ดีและให้ความละเอียดทางด้านความถี่ที่สมเหตุสมผลโดยขึ้นกับสมบัติความสามารถในการได้ยินของหูมนุษย์ ก่อนอื่น 32 แถบความถี่ที่เท่าๆกันจะยังไม่ใช่แถบความถี่วิกฤตดังแสดงไว้ในรูปที่ 3.2 เนื่องจาก bandwidth กว้างเกินไปสำหรับความถี่ต่ำ ส่งผลให้จำนวนบิตที่เกิดจากการ quantizeไม่สามารถปรับให้ทันความไวของสัญญาณรบกวนที่อยู่ในแต่ละแถบวิกฤต โดยที่แถบวิกฤตที่มีความไวของสัญญาณรบกวนมากที่สุด จะเป็นตัวกำหนดจำนวนบิตที่จะ quantize สำหรับแถบความถี่ทั้งหมด ประการที่สอง filter bank และอินเวอร์สของมันจะทำให้เกิดการสูญเสียขึ้นได้ แต่ด้วยความโชคดีที่ว่าข้อผิดพลาดที่เกิดจาก filter bank เป็นปัญหาที่เล็กน้อย จนเราไม่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องคำนึงถึง ท้ายที่สุด filter bank ที่อยู่ใกล้กันจะมีความถี่ที่ซ้อนกันอยู่ สัญญาณที่ความถี่หนึ่งสามารถส่งผลต่อเอาท์พุทของ filter bank 2 แถบที่อยู่ใกล้กันได้ และ filter bank จะให้ 32 frequency sample มี 1 sample ต่อแถบความถี่ย่อย สำหรับ 32 audio sample โดยมีหลักการทำงาน คิอ จะจัดกลุ่ม 12 samples ให้อยู่ในแต่ละแถบความถี่ย่อยซึ่งมีอยู่ 32 แถบความถี่ย่อย แต่ละกลุ่มของ 12 sample นี้จะถูกผ่านกรรมวิธีจัดเรียงบิต (bit allocation) และถ้าการจัดเรียงบิตไม่เท่ากับ 0 จะต้องใช่สเกลแฟกเตอร์เข้าช่วยในการคำนวณอีกที่หนึ่ง การจัดเรียงบิตจะใช้ในการตัดสินใจว่า จะต้องใช้จำนวนบิตขนาดเท่าใดเพื่อจะสามารถแสดงค่าของ input sample และสเกลแฟกเตอร์ก็คือตัวคูณเพื่อกำหนดขนาดของsample เพื่อให้ได้ความละเอียดมากที่สุด layer1 นี้ประกอบไปด้วย 32 แถบความถี่ แถบละ 12 sample รวมกันเป็น 1 frame ทั้งหมดจะเป็น 384 sample นอกจากในแต่ละเฟรมจะมีข้อมูลทางอนาลอกแล้ว ยังมีส่วน header,CRC และข้อมูลเพิ่มเติม(ancillary data)

Layer 2

ขั้นตอนของวิธีนี้จะเป็นเพียงการเพิ่มเติมในบางส่วนของ layer1 โดยจะพัฒนาการบีบอัดข้อมูลโดยโค้ดข้อมูลให้มีขนาดใหญ่ขึ้น โดยที่ให้มี 1152 sample เป็น 1 ช่องอนาลอก(audio channel) ขณะที่ layer1 จะโค้ดข้อมูลเป็นกลุ่มเดียวที่ประกอบไปด้วย 12 sample ในแต่ละแถบความถี่ย่อย เพื่อที่จะลดข้อมูลที่ซ้ำซ้อน(redundancy coding) จะใช้ 1 bit allocation และ 3 scale factor สำหรับ 3*12 sample และตัวถอดรหัสจะให้ scale factor ที่เป็นเอกลักษณ์สำหรับแต่ละกลุ่มที่ประกอบด้วย 12 samples เมื่อจำเป็นเท่านั้นเพื่อหลีกเลี่ยงการลดทอนของสัญญาณ ด้วยวิธีดังกล่าวจะมีประสิทธิภาพมากกว่าของ layer1

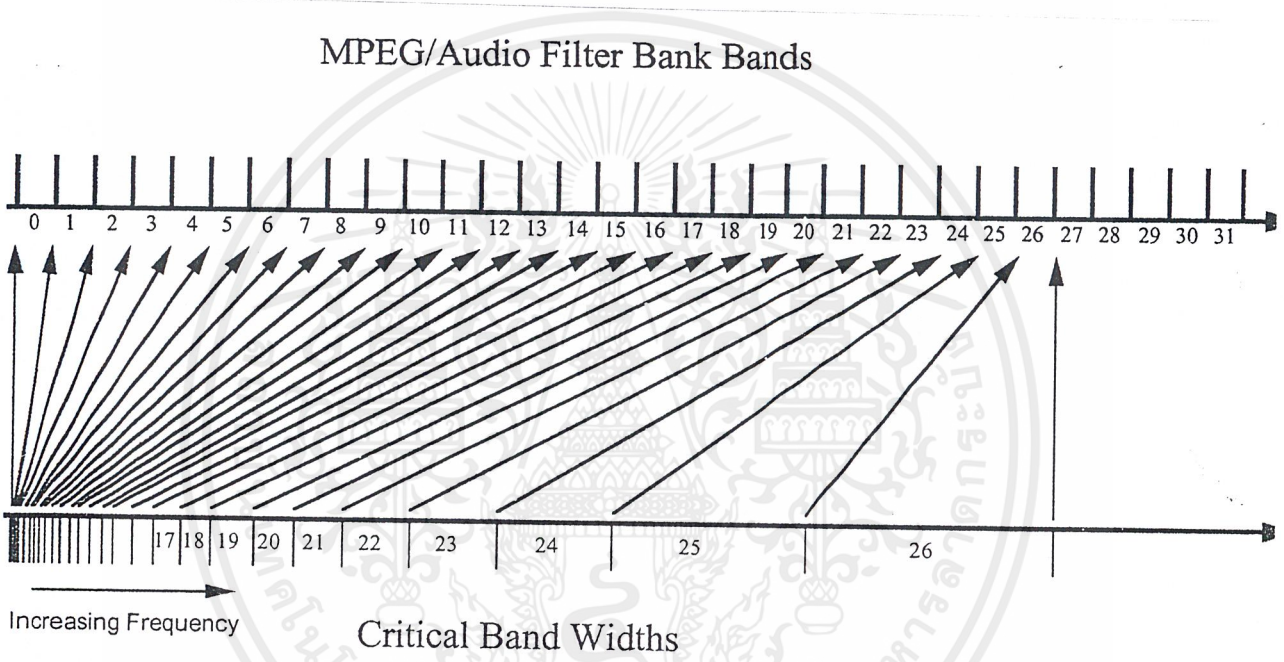
Layer 3

จะเป็นการปรับปรุงแก้ไขจากแบบก่อนๆ ถึงแม้จะใช้ filter bank เหมือนกับ 2 แบบแรก แต่แบบนี้จะชดเชยข้อบกพร่องของ filter bank โดยจัดการเอาท์พุทของ filter ด้วยวิธีการที่เรียกว่า MDCT(Modified discrete cosine transform)

MDCT จะแบ่งเอาท์พุทที่ออกมาจาก filter bank ในรูปความถี่เพื่อว่าจะได้ทำให้เกิดความละเอียดทางด้านความถี่(frequency resolution) ดีขึ้น แต่เนื่องจากเราไม่สามารถหลีกเลี่ยงในเรื่องของความละเอียดทั้งทางด้านเวลาและความถี่ จึงมีการคำนวณเป็นบล็อกๆ และกำหนดให้ความยาวของบล็อกทั้งสองแบบมีขนาดความยาวที่ไม่เท่ากัน โดยบล็อกยาวจะทำให้ความละเอียดทางด้านเวลาดีขึ้น ส่วนบล็อกสั้นจะมีความยาวเป็นหนึ่งในสามของบล็อกยาว นั่นคือ 3 บล็อกสั้นจะเป็น 1 บล็อกยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และการเปลี่ยนไปมาระหว่างบล็อกสั้นและบล็อกยาวจะไม่สามารถเปลี่ยนแปลงไปได้ทันที โดยขั้นตอนกรรมวิธีดังกล่าวจะกล่าวอย่างละเอียดอีกครั้งหนึ่ง



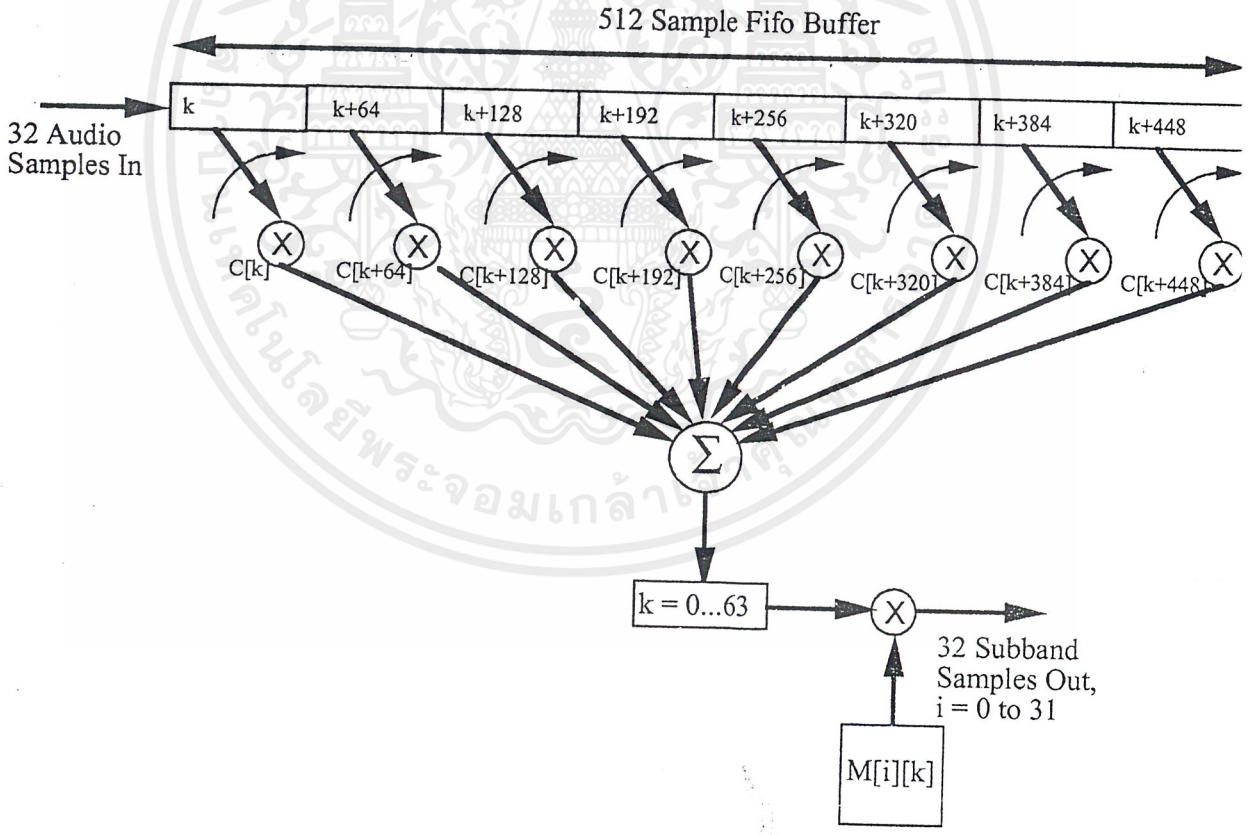
รูปที่ 3.2 แถบความถี่ที่ได้จาก filter bank เทียบกับแถบความถี่วิกฤต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

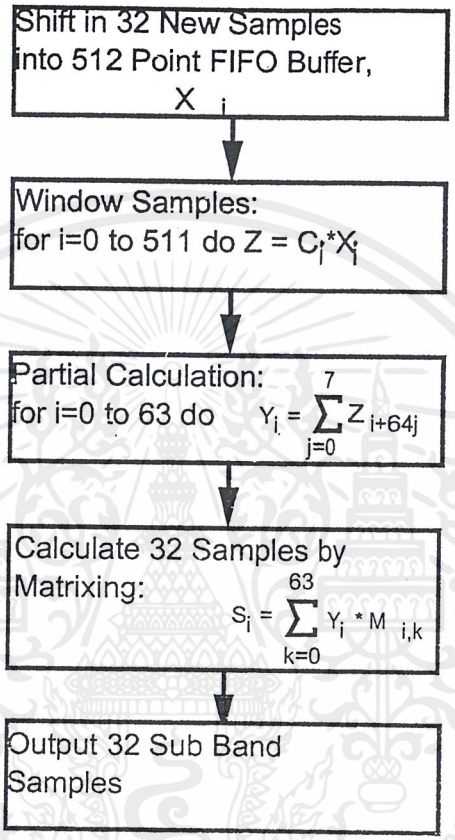
FILTER BANK

เพื่อจะให้มีความเข้าใจดีขึ้น ซึ่งเป็นประโยชน์อย่างมากในการเริ่มจะศึกษาตั้งแต่ต้นกำเนิดของมัน ISO/MPEG audio standard ได้อธิบายวิธีการในการคำนวณตัวกรองแบบโพลีเฟส รูปที่ 4.1 แสดงโครงสร้างสำหรับการคำนวณตัวกรองโพลีเฟสแบบถอดรหัสโดยเสนอโดย นาย Rothweiler เพื่อเป็นการเปรียบเทียบ รูปที่ 4.2 แสดงไดอะแกรมไหล(flow diagram) จากมาตรฐาน ISO/MPEG audio



รูปที่ 4.1 การสร้าง mpeg/audio encoder filter bank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ไดอะแกรมการไหลของ mpeg/audio encoder filter bank

$$S_t[i] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] * (C[k+64j] * x[k+64j]) \qquad \text{สมการ 1}$$

ที่ซึ่ง

I คือ ดัชนีของแถบความถี่ย่อย ซึ่งมีช่วงตั้งแต่ 0 ถึง 31

$S_t[I]$ คือ แซมเปิ้ลทางด้านเอาท์พุทของตัวกรองสำหรับแถบความถี่ย่อย ทีเวลา t โดย t เป็นเลขคูณจำนวนเต็ม ของ 32 ช่วงออดิโอแซมเปิ้ล

C[n] คือ หนึ่งในบรรดา 512 สัมประสิทธิ์ของวินโดว์ที่ถูกกำหนดในมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X[n] คือ อินพุทออดิโอโดยได้มาจาก 512 บัฟเฟอร์ และ

M[I][k] = cos [((2*I+10 * (k-16)π)/64)] คือ สัมประสิทธิ์ของแมทริกซ์

สมการข้างต้นเป็นการแบ่งทำทีละช่วงเพื่อลดการคำนวณให้น้อยลง เพราะว่าฟังก์ชั่นในวงเล็บไม่ขึ้นกับค่าของ i และ M[i][k] ก็เป็นอิสระกับค่าของ j ดังนั้น 32 เอาท์พุทที่ออกจากตัวกรองต้องการแค่ 512+32*64 = 2566 การคูณ และ 64*7 + 32*63 = 2464 การบวก และโดยคร่าว 80 การคูณและการบวกต่อ 1 เอาท์พุท ซึ่งการจะลดกระบวนการคูณหรือการบวก จะใช้วิธีการแบบ fast discrete cosine transform หรือ fast fourier transform

ยังมีอีกวิธีในการคำนวณสมการ 1 ให้เป็นแบบตัวกรองคูณประสาน (Filter convolution) ดังสมการ

$S_t[i] = \sum x(t-n) * H_i[n]$

ที่ซึ่ง

$x(\tau)$ เป็นออดิโอแซมเปิ้ล ที่เวลาτ และ

$H_i[n]$ = h(n) * cos [(2*i+1) * (n-16) * π)/64] โดย

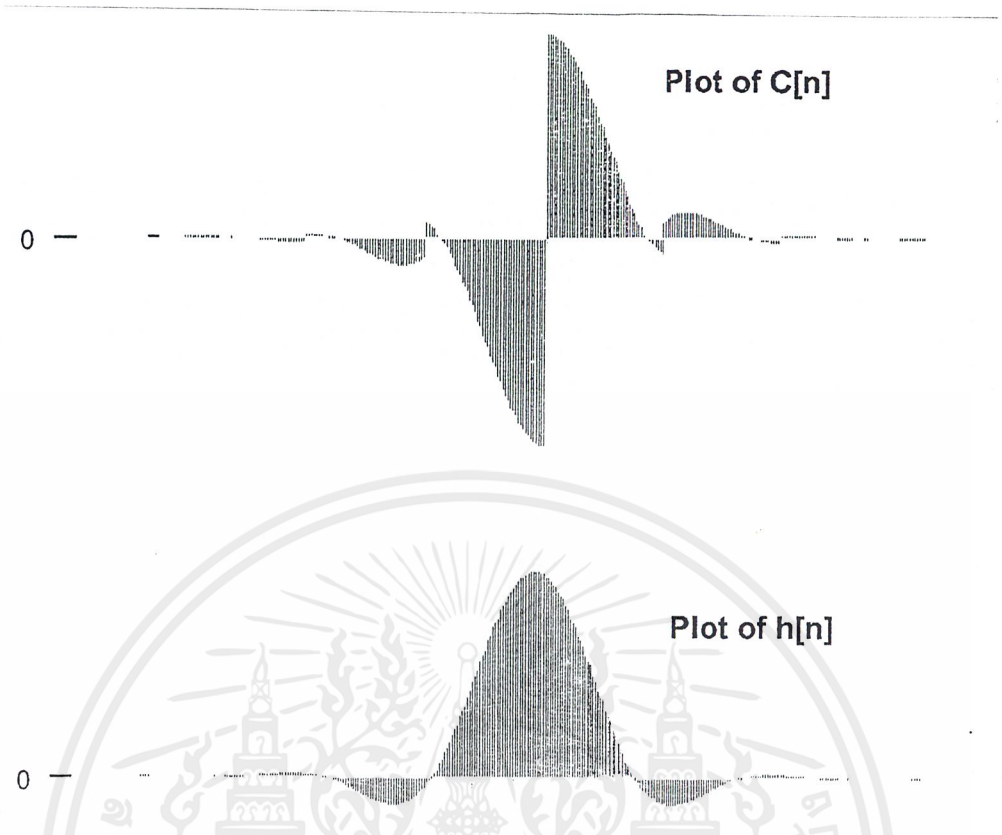
h[n] = -C[n] ถ้าส่วนที่เป็นเลขจำนวนเต็มของ (n/64) เป็นเลขคี่

= C[n] กรณีอื่นๆ สำหรับ n=0 ถึง 511

ในการคำนวณรูปแบบนี้ แต่ละแถบความถี่ย่อยของ filter bank จะมีผลตอบสนองแบบช่วงแถบความถี่ผ่านของตัวมันเอง(band-pass filter response) คือ $H_i[n]$ ถึงแม้ว่า แบบนี้จะสะดวกในการคำนวณแต่ผลลัพธ์ที่ได้จะยังไม่ค่อยมีประสิทธิภาพ การคำนวณตรงๆของสมการนี้ต้องการ 32*512 = 16384 การคูณ และ 32*511 = 61352 การบวก เพื่อจะคำนวณ 32 เอาท์พุทของตัวกรอง

สัมประสิทธิ์ h[n] ก็จะเป็นโปรโตไทป์ของผลตอบสนองตัวกรองความถี่ต่ำของตัวกรองแบบโพลีเฟส รูป 4.3 ได้เปรียบเทียบการพล็อท h[n] กับ c[n] โดย c[n] จะใช้ในสมการคำนวณทีละช่วง และจะมีกลุ่มของความถี่เลขคี่ 64 สัมประสิทธิ์ของ h[n] เพื่อที่จะชดเชยกับค่า M[i][k] พจน์โคไซน์ของ M[I][k] จะแปรในช่วง 0 ถึง 63 และครอบคลุมจำนวนคี่ของครึ่งวัฏจักร(cycle) ในขณะที่พจน์โคไซน์ของ $H_i[n]$ จะแปรค่าในช่วง 0 ถึง 511 และครอบคลุม 8 เท่าของครึ่งวัฏจักร

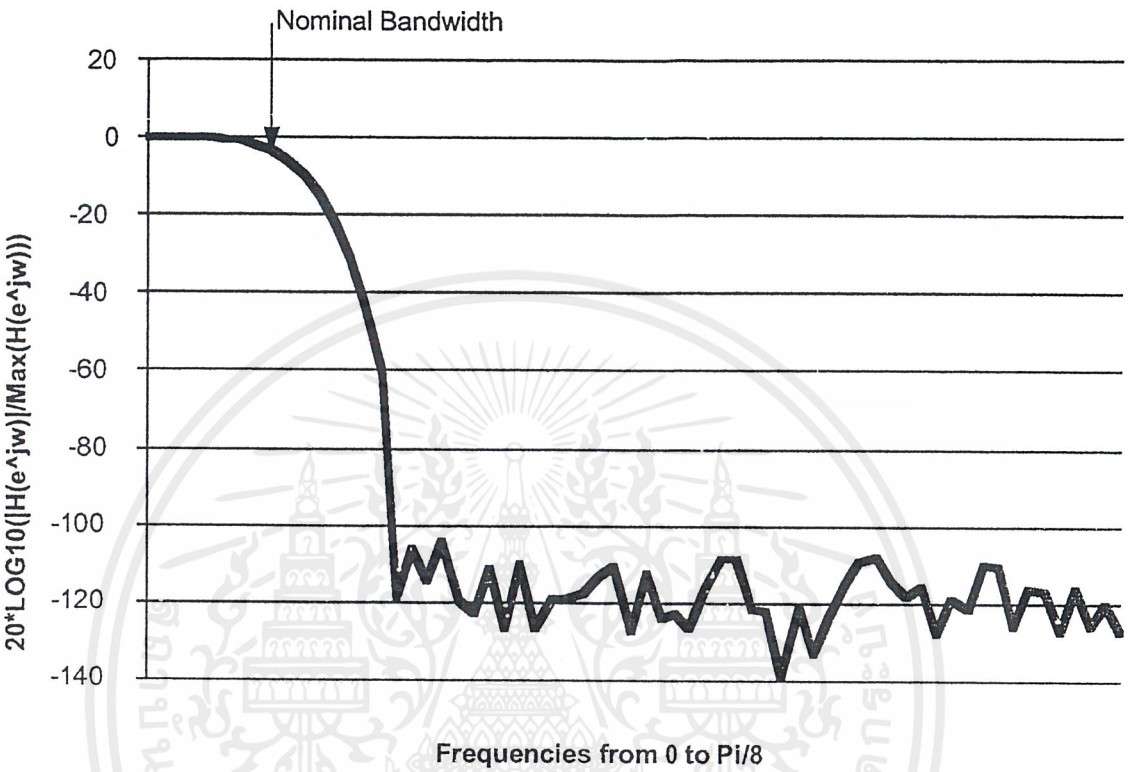
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 การเปรียบเทียบ ระหว่าง h[n] และ c[n]

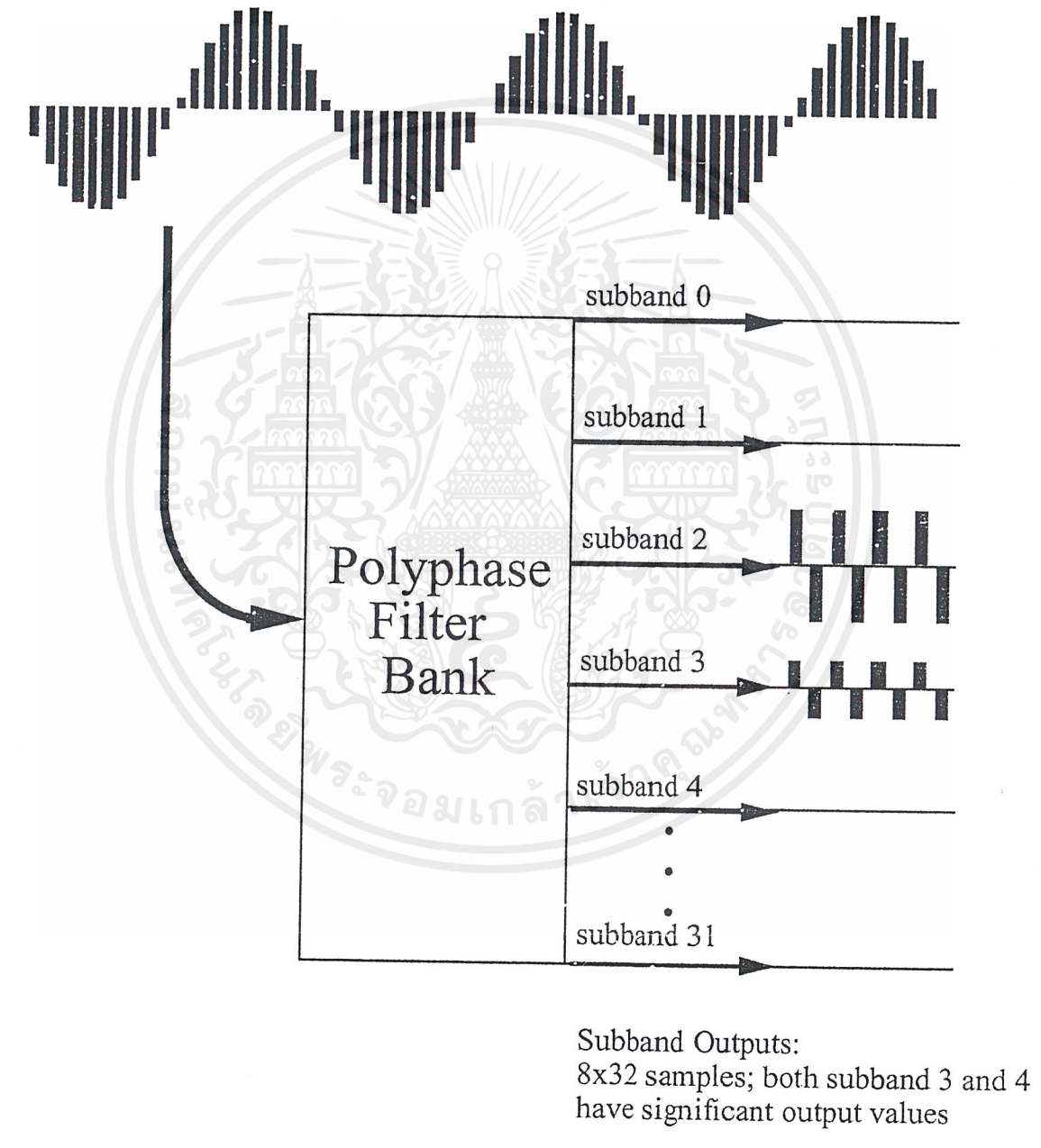
สมการ $H_i[n]$ แสดงว่า แซมเปิ้ลแต่ละตัวจะเป็นการมอดดูเลทของผลตอบสนองแบบโปรโตไทป์กับพจน์โคไซน์ เพื่อเลื่อนผลตอบสนองความถี่ต่ำให้ไปอยู่ในช่วงแถบความถี่ที่เหมาะสม จึงเป็นที่มาของคำว่า ตัวกรองโพลีเฟส ตัวกรองนี้จะมีความถี่ศูนย์กลางอยู่ที่จำนวนเท่าของเลขคี่ของ $\pi/(64T)$ โดยที่ T คือ คาบของการสุ่มออดิโอ และแต่ละตัวจะมีความกว้างของความถี่คือ $\pi/(32T)$ รูปที่ 4.4 แสดงผลตอบสนองของตัวกรองแบบโปรโตไทป์ แต่ยังไม่คมที่ช่วงคัทออฟ ดังนั้นเอาท์พุทที่ออกจากตัวกรอง จะสามารถเกิดการซ้อนทับของสัญญาณขึ้นได้ การออกแบบตัวกรองแบบโปรโตไทป์และการรวมกันของการเลื่อนเฟสให้เหมาะสมในพจน์ของโคไซน์ จะทำให้เกิดการหักล้างของการซ้อนทับของสัญญาณที่เอาท์พุทของ filter bank ของตัวเข้ารหัส (decoder) เหตุผลอีกอย่างหนึ่งของการใช้ตัวกรองที่มีช่วงความถี่กว้างกว่าปกติก็คือ การซ้อนทับของช่วงความถี่ของตัวกรองโพลีเฟสที่อยู่ใกล้กัน ผลกระทบอันนี้จะทำให้เกิดการเสียหายแก่การบีบอัดข้อมูลออดิโอ เพราะว่า พลังงานของสัญญาณที่ใกล้ขอบแถบความถี่ normalize จะปรากฏในรูปของ 2 เอาท์พุทที่ติดกันของตัวกรองโพลีเฟส รูป 4.5 แสดงรูปสัญญาณไซน์ ที่มีเพียงพลังงานเดียวที่ความถี่เดียว แต่ปรากฏที่เอาท์พุทของตัวกรองโพลีเฟสที่ 2 จำนวนแถบความถี่ย่อย(subband)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลตอบสนองทางความถี่ของตัวกรองโปรโตไทป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



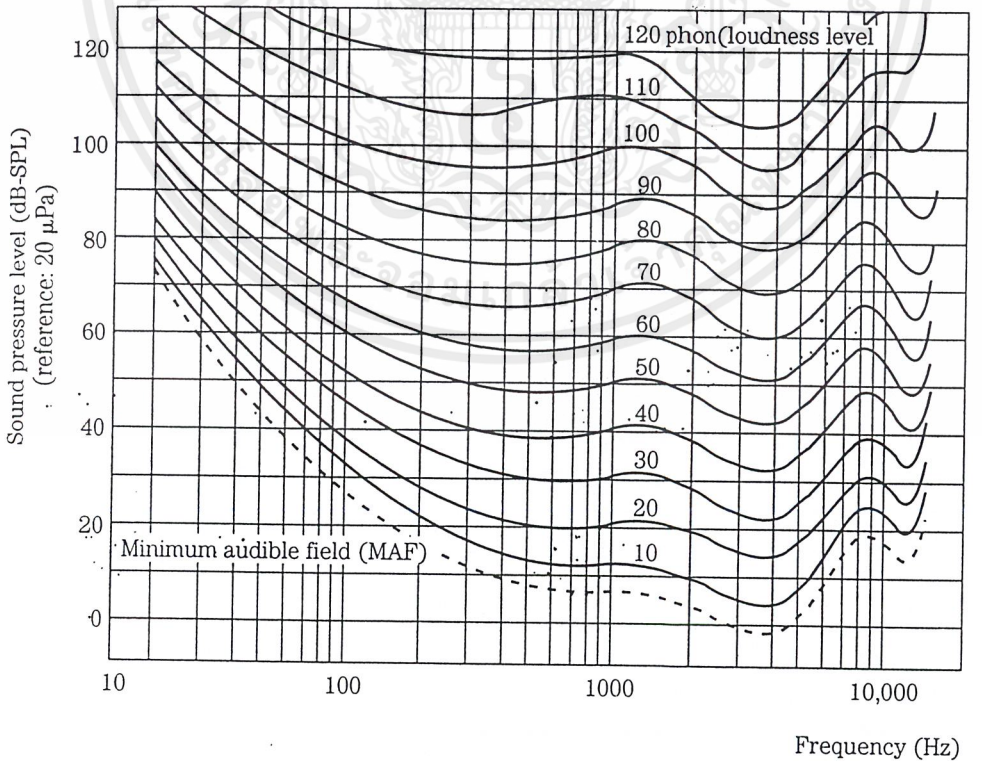
รูปที่ 4.5 การซ้อนทับของสัญญาณ

สัญญาณไซน์ที่ความถี่เดียวสามารถเกิดเอาท์พุทได้ 2 แถบความถี่ย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไซโคอะคูสติก (Psychoacoustics)

เมื่อเราได้ยินเสียงเพลง เราสามารถแยกเสียงฮาโมนิกส์ที่ 5 กับความถี่มูลฐานได้หรือไม่ หรือบอกข้อแตกต่างของเสียงที่ 1000 Hz กับ 1002 Hz ทั้งหมดที่กล่าวมาจะอยู่ในเรื่อง ไซโคอะคูสติก ซึ่งเป็นศาสตร์ที่เกี่ยวข้องกับการได้ยินของหูมนุษย์ โดยจะอธิบายผลตอบสนองของหูในทุกอย่างที่เราได้ยิน หูคนจะครอบคลุมความถี่ในช่วงที่กว้างมาก ขีดเริ่มของการได้ยินที่ 120 dB SPL จะมีความเข้มของเสียงเป็นล้านล้านเท่าของความถี่ขีดเริ่มในการได้ยินที่ 0 Db SPL เพื่อความสะดวกในการอธิบาย จึงเป็นการง่ายในการใช้หน่วยเดซิเบลเข้ามาช่วย เพื่อสะดวกในการพล็อตกราฟและอ่านค่า ความไวของหูคนจะมากที่สุดที่ความถี่ตั้งแต่ 1 KHz ถึง 5 KHz และความไวจะลดลงที่ความถี่สูงและความถี่ต่ำ จากการทดลองกล่าวว่า ทางเดินของความดังที่เท่ากัน(equal-loudness contour) เช่นแบบของ Robinson-Danson Curve บอกไว้ว่า ทางเดินที่ต่ำสุดจะบอกถึงสนามที่น้อยสุดที่จะได้ยิน ซึ่งก็หมายถึง ระดับพลังงานเสียงที่น้อยที่สุดภายใต้แถบความถี่ของการได้ยิน อย่างเช่น จุดเริ่มที่จะได้ยินของเสียงที่ ความถี่ 30 Hz จะต้องมาความดังเป็น 60 เท่าของเสียงที่ความถี่ 4 KHz ผลตอบสนองจะแปรเปลี่ยน โดยขึ้นกับความดัง ยิ่งเสียงดังมากเท่าไหร่ ผลตอบสนองก็จะยิ่งเรียบมากขึ้นเท่านั้น

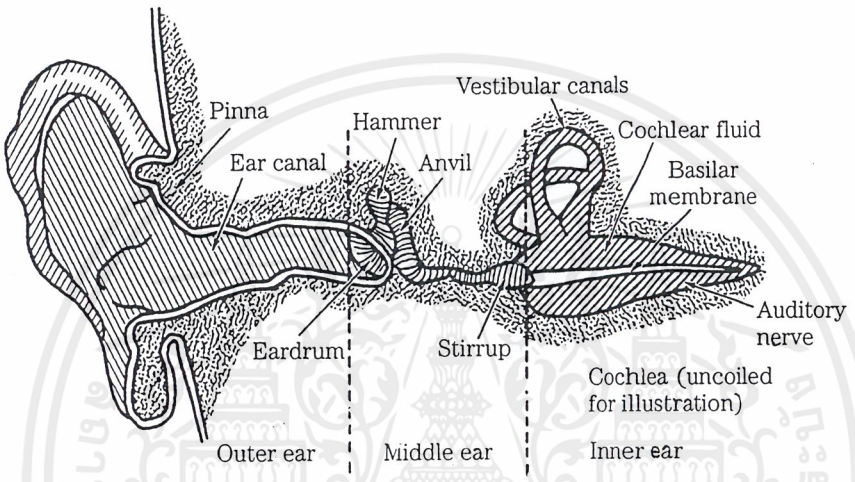


รูปที่ 4.6 แสดงทางเดินของเสียงที่ความดังเท่ากันบอกเป็นนัยว่าหูคนมีความสัมพันธ์ไม่เป็นเส้นตรงกับความถี่ และเส้นโค้งนี้ได้มาจาก การศึกษาไซโคอะคูสติก โดยใช้เสียงเป็นรูปซายน์

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรีระศาสตร์ของหูมนุษย์ และ แถบความถี่วิกฤต

หูคนจะเปลี่ยนพลังงานเสียงที่ได้ยินเป็นพลังงานกล และสุดท้ายจะเปลี่ยนเป็นพลังงานคลื่นไฟฟ้า และถูกส่งไปยังสมอง รูปที่ 4.7 แสดงส่วนต่างๆ ของหูคนเรา



รูปที่ 4.7 รูปแบบอย่างง่ายๆ ของหูมนุษย์

ได้มีการทดสอบ bisilar membrane ผลออกมาว่า หูประกอบด้วย hair cell ประมาณ 30000 เซลล์ ถูกจัดเรียงเป็นหลายๆ แถวตามแนวของ bisilar membrane ขนาดประมาณ 32 มิลลิเมตร ส่วนนี้จะเรียกว่า organ of corti โดยเซลล์เหล่านี้จะตรวจจับการสั่นสะเทือนของ bisilar membrane และส่งผ่านข้อมูลของเสียงไปยังสมองในรูปของคลื่นไฟฟ้า การแบ่งแยกความถี่ (Frequency Discrimination) บอกไว้ว่า ที่ความถี่ต่ำเสียงที่ต่างกัน 2-3 Hz หูก็จะสามารถแยกแยะได้ แต่เมื่อที่ความถี่สูง เสียงจะต้องต่างกันหลายร้อยเฮิร์ต ถึงจะได้ยิน และในแต่ละกรณี hair cell จะตอบสนองกับการกระตุ้นที่มากสุดในย่านความถี่ของมันเท่านั้น โดยจะเรียกย่านเหล่านี้ว่า แถบความถี่วิกฤต(Critical band) และสามารถจะประมาณได้ด้วยสมการ

Critcal Bandwidth in Hertz = 24.7 * (4.37F + 1)

เมื่อ F เป็นความถี่กึ่งกลางในหน่วยกิโลเฮิร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นาย Eberhard Zwicker ได้ประมาณหูคนว่ามี 24 แถบความถี่วิกฤตสำหรับความถี่ที่ต่ำกว่า 15 KHz และ แบนด์ที่ 25 จะคลอบคลุมในย่านตั้งแต่ 15 ถึง 20 KHz โดยตัวอย่างของแถบความถี่วิกฤต จะมีไว้ในรูป 4.8 สังเกตว่า แถบความถี่วิกฤตสำหรับความถี่ 1KHz ที่เป็นซายน์จะมีความกว้างประมาณ 100 Hz และมีความถี่กึ่งกลาง 1 KHz จะสามารถได้ยินเสียง ถ้าเกิดเสียงนั้นมีพลังงานมากกว่า เสียง 1 KHz ที่ระดับเดียวกัน

Critical band number (Bark)	Center frequency (Hz)	Critical band (Hz)	Lower cutoff frequency (Hz)	Upper cutoff frequency (Hz)
1	50	–	–	100
2	150	100	100	200
3	250	100	200	300
4	350	100	300	400
5	450	110	400	510
6	570	120	510	630
7	700	140	630	770
8	840	150	770	920
9	1000	160	920	1080
10	1170	190	1080	1270
11	1370	210	1270	1480
12	1600	240	1480	1720
13	1850	280	1720	2000
14	2150	320	2000	2320
15	2500	380	2320	2700
16	2900	450	2700	3150
17	3400	550	3150	3700
18	4000	700	3700	4400
19	4800	900	4400	5300
20	5800	1100	5300	6400
21	7000	1300	6400	7700
22	8500	1800	7700	9500
23	10,500	2500	9500	12,000
24	13,500	3500	12,000	15,500
25	18,775	6550	15,500	22,050

Tobias

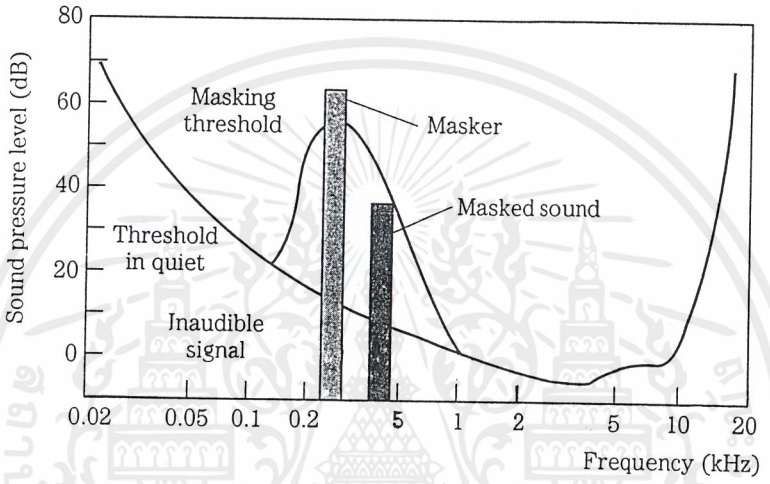
รูปที่ 4.8 เป็นตัวอย่างตารางของแถบความถี่วิกฤตในย่านการได้ยิน

ขีดเริ่มของการได้ยิน และการบดบัง(Threshold of Hearing and Masking)

ปรากฏการณ์ 2 อย่างที่เกี่ยวข้องกับการได้ยิน ของหูมนุษย์ คือ ขีดเริ่มน้อยสุดของการบดบัง(Minimum Masking Threshold) ดังรูปที่ 4.9 เส้นโค้งของขีดเริ่มการได้ยินจะอธิบายถึงระดับที่น้อยที่สุดซึ่งหูจะสามารถรับรู้ได้ ณ ความถี่ใดๆ ขีดเริ่มนี้จะเปรียบเทียบกับกับเสียงที่ 0 dB ความถี่ 1 KHz หูคนจะมีความไวมากสุดที่ความถี่ 1 ถึง 5 KHz ที่ซึ่งเราสามารถได้ยินเสียงที่ต่ำกว่า 0 dB ได้ โดยปกติสัญญาณเสียง 2 ชุดที่มีพลังงานเท่ากัน และคนละความถี่จะมีความดังไม่เท่ากัน และจะเหมือนกับการได้ยินเสียงสัญญาณรบกวน และการลดทอนสัญญาณจะเปลี่ยนตามความถี่ ความไว

เอกสารนี้สงวนลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะลดลงที่ความถี่สูง และความถี่ต่ำ อย่างเช่น เสียงที่ 20 Hz จะต้องมีพลังงานอย่างน้อยเท่ากับ 70 dB ของเสียง 1KHz เพื่อที่จะได้ยินที่ความดังเท่ากัน การเข้ารหัสจะเปรียบเทียบสัญญาณอินพุทกับค่าขีดเริ่มของการได้ยิน และทำการกำจัดสัญญาณที่อยู่ใต้ค่าขีดเริ่ม เพราะว่าหูเราจะไม่ได้ยินเสียงเหล่านี้



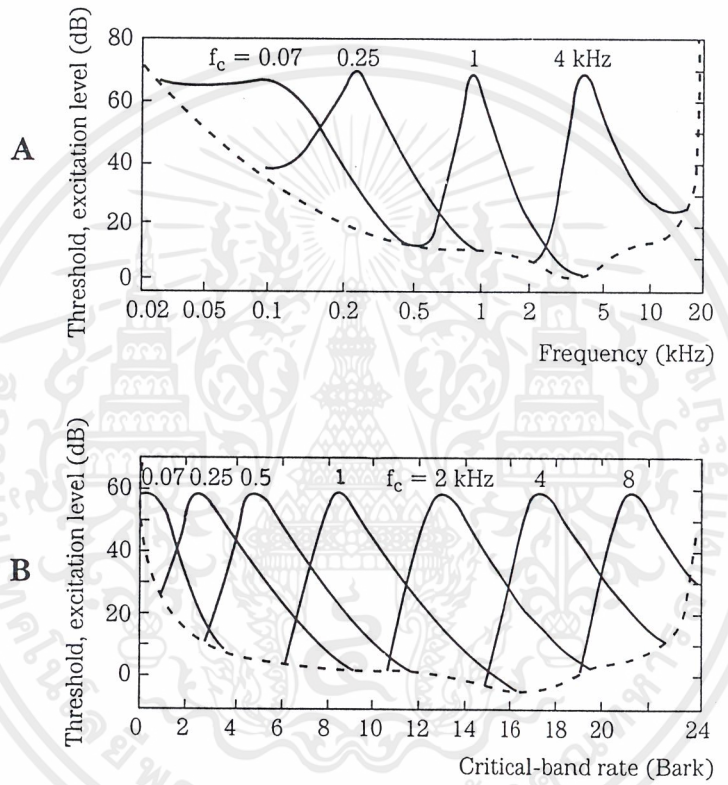
รูปที่ 4.9 ขีดเริ่มของการได้ยินอธิบายถึง เสียงที่อ่อนกับย่านการได้ยินของหู เสียงที่มีพลังงานมากจะยกระดับขีดเริ่มการได้ยินในย่านใกล้เคียง ทำให้ได้เส้นโค้งการบดบังชุดใหม่

การบดบังทางขนาดจะเกิดขึ้นเมื่อสัญญาณเสียงได้เลื่อนเส้นโค้งขีดเริ่มขึ้นในย่านของความถี่ที่อยู่ใกล้ๆ เสียงอันนั้น ขีดเริ่มการบดบังบอกระดับเสียงอันจะซึ่งเริ่มได้ยิน เมื่อมีเสียงหลายๆ ชุดถูกสร้างขึ้นมาพร้อมกัน การบดบังจะเกิดขึ้น เมื่อเสียงอันที่ดังกว่าจะไปบดบังเสียงที่อ่อนกว่า โดยที่เสียงอันที่ดังกว่าจะเรียกว่า ตัวบดบัง(Masker) และเสียงที่อ่อนกว่าจะเรียกว่า ตัวถูกบดบัง(Maskee) ทฤษฎีการบดบังได้บอกไว้ว่า เสียงที่ความถี่ต่ำสามารถจะบดบังเสียงที่ความถี่สูงกว่าได้ในแถบความถี่วิกฤตเดียวกัน เช่น เสียงที่ความถี่ 500 Hz สามารถจะบดบังเสียงที่อ่อนกว่าที่ความถี่ 600 Hz ได้ระบบการบันทึกเสียงจะไม่บันทึกเสียง 600 Hz ลงไปเพราะว่าหูเราจะไม่ได้ยินเสียงนี้ และจะบันทึกเสียงที่ความถี่ 500 Hz แทน โดยปกติมักจะขึ้นกับขนาด เสียงที่อ่อนกว่าจะถูกบดบังด้วยเสียงที่เข้มกว่า ที่ความถี่ใกล้เคียงกัน โดยส่วนมากจะอยู่ในช่วง 100 Hz ในช่วงความถี่ต่ำ หรือก็คือ แถบความถี่วิกฤตเดียวกัน

จากรูปที่ 4.10 เห็นได้ว่า ความถี่ต่ำสามารถรบกวนความถี่สูงได้ การบดบังจะมีการกินเข้าไปในแถบความถี่วิกฤตที่ใกล้เคียงกัน เมื่อเสียงอันดังและมีหลายฮาโมนิกส์ เช่น สัญญาณ 1 KHz

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

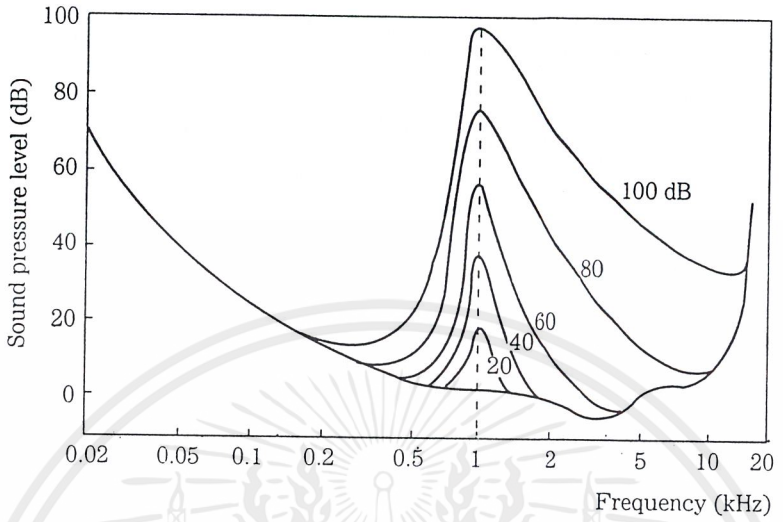
จะสามารถบดบังสัญญาณ 2 KHz ได้ เสียงที่มีแอมปลิจูดต่ำๆ จะบดบังเสียงอื่นได้น้อย เสียงที่มีความกว้างของแถบความถี่วิกฤตน้อยก็จะบดบังสัญญาณอื่นได้น้อยเช่นกัน ส่วนเสียงอันดังและมีหลายความถี่ จะมีการบดบังได้มาก โดยจะมีเส้นโค้งการบดบังขยายกว้าง



รูปที่ 4.10

เส้นโค้งการบดบัง ณ ที่ความถี่ใดๆ จะไม่สมมาตรกัน ความชันของเส้นโค้งที่ถูกเลื่อนขึ้นจะชันน้อยกว่าที่ด้านความถี่สูง ดังนั้นจึงเป็นเรื่องไม่ยากนักสำหรับเสียงที่ความถี่ต่ำจะบดบังเสียงที่ความถี่สูง และถ้าระดับพลังงานของตัวบดบังมากขึ้น ก็จะทำให้เส้นโค้งขีดเริ่มขยายตัวกว้างขึ้น และความชันของเส้นโค้งที่ความถี่สูงก็จะลดลงอีก แต่ความชันที่ความถี่ต่ำแทบจะไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ขีดเริ่มการบดบังจะเปลี่ยนแปลงตามระดับพลังงาน โดยทดลองที่ความถี่กลาง 1 KHz

เหตุผลของการโค้ดสัญญาณตามการได้ยินของหูมนุษย์

จุดประสงค์ของระบบการลดจำนวนข้อมูล คือ ลดอัตราข้อมูล(Data Rate) ซึ่งเป็นผลคูณของความถี่สุ่ม และ ความยาวของคำ (Wordlength) โดยสามารถลดบิตเรทลงได้ด้วยการลดความถี่สุ่ม แต่จากทฤษฎีของ Nyquist จะไปกระทบกับแถบความถี่ของสัญญาณ ในอีกทางหนึ่ง คือ ลดความยาวของคำลง อย่างไรก็ตามจะทำให้ไปลดทองสัญญาณเสียงนั้นๆ ลง 6 DB/bit หรือก็คือ เพิ่มระดับสัญญาณรบกวนขึ้นมา จึงมีการใช้หลักการทางไซโคอะคูสติก โดยจะคงค่าความถี่สุ่มเอาไว้ แต่จะลดขนาดของข้อมูลลง ตามเงื่อนไขของสัญญาณ โดยเฉพาะการบดบังและปัจจัยอื่นๆ จะถูกพิจรณาเพื่อทำให้ผลลัพธ์ที่เพิ่มระดับของสัญญาณรบกวนมีผลกระทบน้อยที่สุด

ระบบการเข้ารหัสสัญญาณจะพิจารณาความถี่และแอมปลิจูด ของสัญญาณอินพุท และเปรียบเทียบกับการรับรู้ของหูมนุษย์ ข้อมูลที่สอดคล้องกับรูปแบบก็หมายถึง ส่วนที่เราได้ยิน และจะนำมาเข้ารหัส ส่วนอื่นจะถูกกำจัดออกไป ผลลัพธ์ของกระบวนการคำนวณทางดิจิตอลทำให้ได้ข้อมูลส่วนที่สำคัญเท่านั้น เช่น สามารถจะลดอัตราข้อมูลลงไปได้ถึง ¼ จาก 1.41 Mbps เป็น 384 Kbps ปริมาณข้อมูลลดลงไปถึง 75% โดยวิธีการแบบ PCM ทุกๆสัญญาณจะถูกกำหนดให้มีความยาวเท่ากัน แต่การเข้ารหัสด้วยวิธีไซโคอะคูสติกจะกำหนดจำนวนบิตตามการได้ยิน ถ้าเสียวที่สำคัญจะกำหนดจำนวนบิตมากเพื่อจะแน่ใจว่าไม่สูญเสียข้อมูลในส่วนที่สำคัญไป ในทางตรงกันข้ามจำนวนบิตที่ไม่มากจะถูกกำหนดให้กับเสียงที่อ่อนกว่า ส่วนเสียงที่ไม่ได้ยินจะไม่มีการโค้ดเลย ด้วยวิธีนี้การลดบิตเรทก็จะทำได้สำเร็จ อัตราส่วนการลดข้อมูลเป็นอัตราส่วนของอินพุทบิตเรท ต่อเอาท์พุทบิทเรต อาจจะเป็น 4:1, 8:1 หรือ 16:1 ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติจะมีวิธีจัดเรียงบิตอยู่ 2 ชนิด แต่วิธีที่นิยมใช้คือ Forward Adaptive Allocation ซึ่งการจัดสรรจำนวนบิตจะทำให้เสร็จในส่วนของการเข้ารหัส และข้อมูลที่ถูกเข้ารหัสแล้วจะอยู่ในรูปของบิตสตรีม ประโยชน์ของวิธีนี้คือใช้วิธีทางไซโคอะคูสติก ในส่วนของตัวเข้ารหัส ส่วนตัวถอดรหัสไม่ต้องมีในส่วนนี้ เพราะว่าจะใช้ข้อมูลจากบิตสตรีม ในการสร้างสัญญาณขึ้นมาใหม่

Bits/sample	Comp ratio	58 kHz	44.1 kHz
16	1:1	768 kbps	705.6 kbps
8	2:1	384	352.8
4	4:1	192	176
2.67	6:1	128	117.7
2	8:1	96	88.2
1.45	11:1	69.6	64

รูปที่ 4.12-การลดจำนวนบิตเรท สำหรับสัญญาณสุ่มที่ความถี่ 44.1 และ 48 KHz

การโค้ดสัญญาณแบบการบีบอัดข้อมูล(Data Reduction Coding)

การโค้ดสัญญาณในโดเมนความถี่มี 2 ชนิดคือ การโค้ดแบบสับแบนด์ และการโค้ดแบบทรานฟอร์ม โดยทั่วๆไปการโค้ดแบบสับแบนด์จะทำให้ได้ความละเอียดทางเวลาสูงแต่จะแย่ในส่วนของความละเอียดทางความถี่ แต่การโค้ดแบบทรานฟอร์มจะให้ความละเอียดทางเวลาต่ำและความละเอียดทางความถี่สูง ทั้ง 2 แบบ แสดงในรูป 4.13 ไม่ว่าแซมเปิ้ลในโดเมนเวลา หรือ โดเมนความถี่อย่างใดอย่างหนึ่งจะถูกควอนไตซ์ตามรูปแบบของไซโคอะคูสติกที่อยู่ในส่วนของการเข้ารหัส การโค้ดแบบสับแบนด์จะแบ่งอินพุทออกเป็นหลายๆสับแบนด์ โดยการใช้ฟิลเตอร์แบงค์มาช่วย และพิจารณาพลังงานที่อยู่ในแต่ละสับแบนด์ ในส่วนของการโค้ดแบบทรานฟอร์ม สัญญาณอินพุทจะถูกเปลี่ยนเป็นบล็อกของสเปกตรัมในโดเมนความถี่ จากนั้นสัมประสิทธ์ของการเปลี่ยนจะถูกควอนไตส์และโค้ดตามแบบทางไซโคอะคูสติก

ทั้งในส่วนของการโค้ดแบบสับแบนด์ หรือ ทรานฟอร์มจะใช้วิธีทางไซโคอะคูสติกในการควอนไตส์เฉพาะสัญญาณในส่วนที่มีนัยสำคัญต่อการได้ยินเท่านั้น ในส่วนของสัญญาณที่ต่ำกว่าขีดเริ่ม หรือถูกบดบังโดยสัญญาณอื่นที่มีนัยสำคัญกว่าจะทำให้ไม่ได้ยิน และจะไม่มีการโค้ดสัญญาณนั้น นอกจากนั้นความละเอียดในการควอนไตส์จะต้องปรับเปลี่ยนให้ความผิดพลาดเกิดในส่วนที่มีนัยสำคัญ โดยหวังว่าเมื่อถอดรหัสสัญญาณออกมาแล้ว ข้อผิดพลาดจะถูกบดบังโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

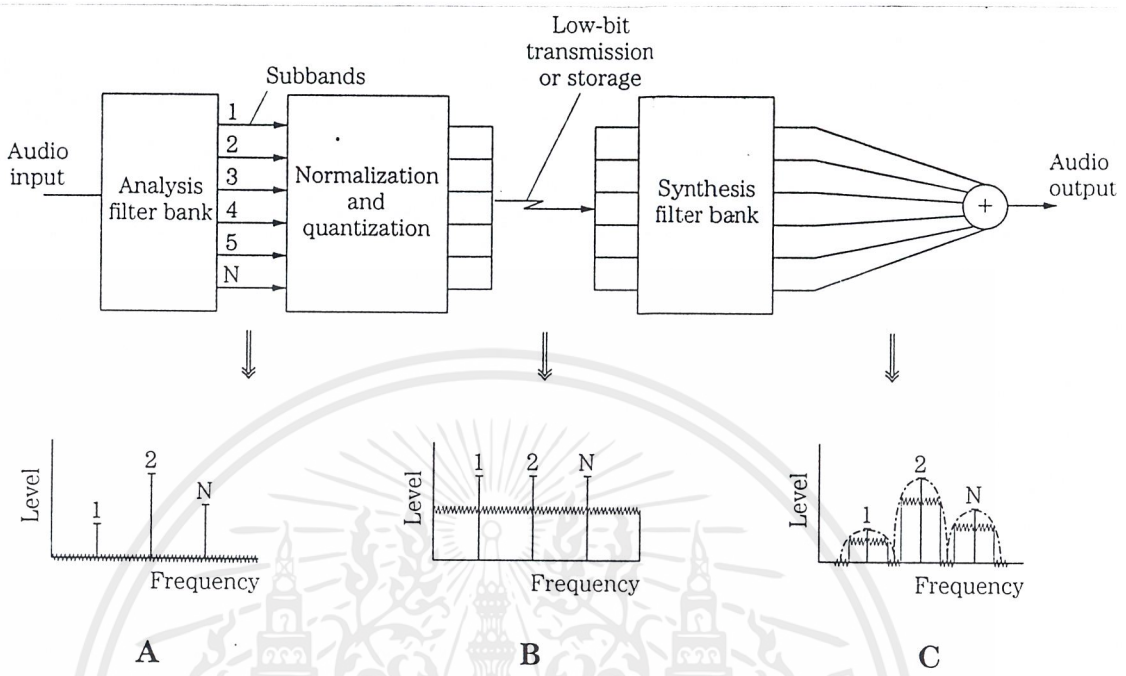
สัญญาณของมันเองอีกที วิธีการนี้จะทำให้ได้การบีบอัดที่ดี สามารถจะทำให้ได้อัตราส่วนการบีบอัด 4:1 หรือ 12:1 อย่างไม่ยากนัก

การโค้ดแบบสับแบนด์

การโค้ดแบบนี้ได้ถูกพัฒนาครั้งแรกในห้องทดลองเบลล์ในปี 1980 บล็อกของแซมเปิ้ลที่อยู่ในโดเมนเวลาซึ่งแสดงถึงสัญญาณแถบกว้างจะถูกนำมารวบรวมและป้อนเข้าไปในฟิลเตอร์แบงค์ โดยจะแบ่งสัญญาณออกเป็นช่องๆ เพื่อจะประมาณแถบความถี่วิกฤตของหูคนเรา โดยที่ฟิลเตอร์แบงค์นี้จะต้องมีความถี่คัทออฟที่คม เพื่อเลียนแบบผลตอบสนองของแถบความถี่วิกฤต และจำกัดสัญญาณรบกวนจากการควอนไตซ์ให้อยู่ในแบนด์นั้นๆ ซึ่งจะมีเพียงตัวกรองทางดิจิตอลเท่านั้นที่ทำได้ โดยแซมเปิ้ลในแต่ละสับแบนด์จะถูกนำมาพิจารณา และเปรียบเทียบกับรูปแบบทางไซโคอะคูสติก

แต่ละสับแบนด์ จะถูกโค้ดอย่างเป็นอิสระต่อกัน โดยอาจจะมีจำนวนบิตมากหรือน้อยต่างกันในแต่ละสับแบนด์ แต่ทุกๆกรณี จะเกิดสัญญาณรบกวนจากการควอนไตซ์เพิ่มขึ้นมา อย่างไรก็ตามเมื่อสัญญาณสร้างขึ้นมาใหม่ สัญญาณรบกวนจะถูกจำกัดให้อยู่เฉพาะในสับแบนด์นั้น และถูกบดบังโดยสัญญาณดังรูป 4.13 การจัดแจงบิต จะถูกพิจารณาโดยรูปแบบทางไซโคอะคูสติก โดยทำทุกๆสับแบนด์ในทุกๆบล็อกของข้อมูลใหม่ที่เข้ามา แซมเปิ้ลจะถูกควอนไตซ์ตามการได้ยินของสัญญาณและสัญญาณรบกวน

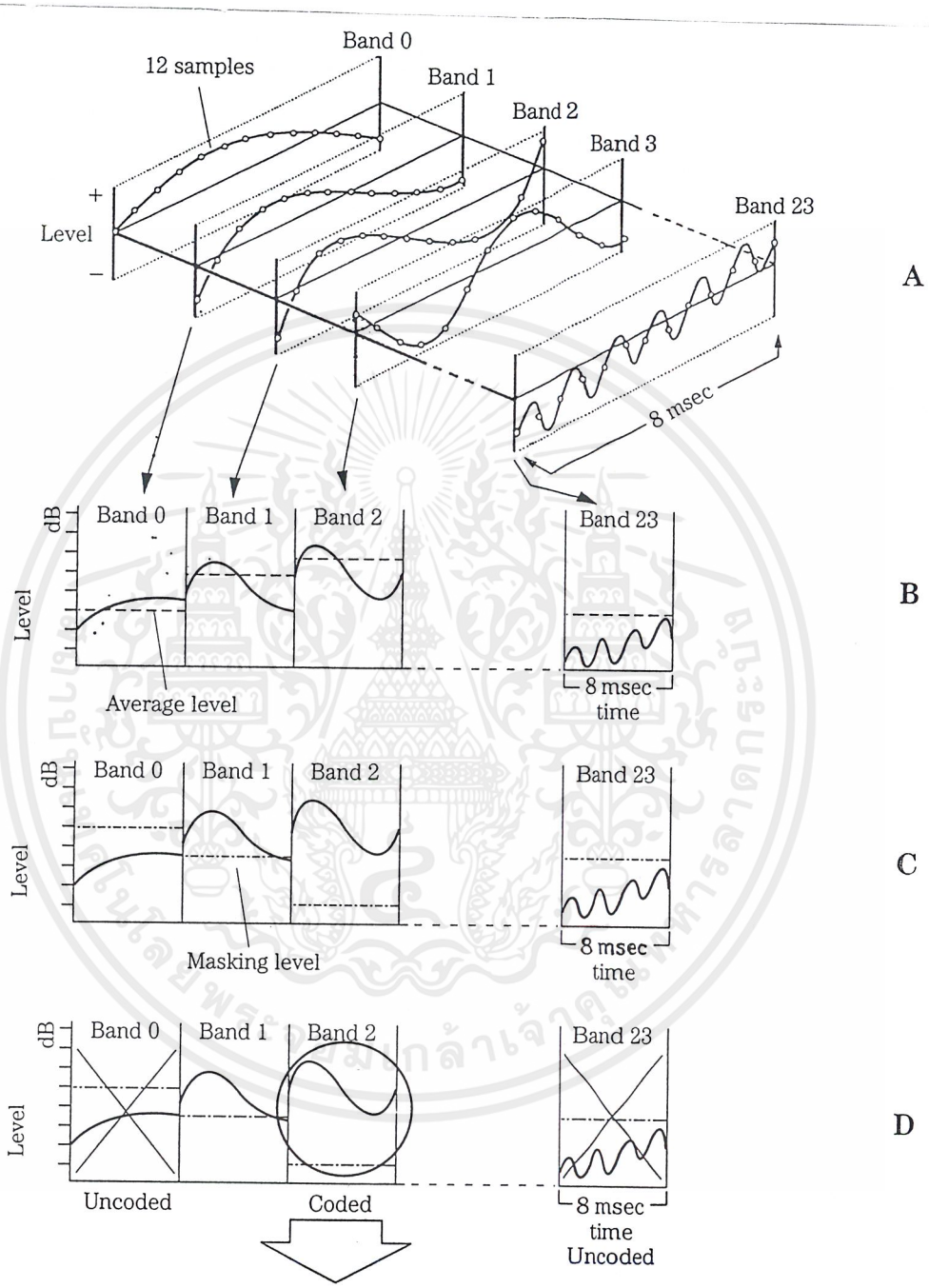
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 การโค้ดแบบสับแบนด์จะสร้างแซมเปิ้ลให้อยู่เป็นแบนด์แคบๆที่มีสัญญาณรบกวนระดับน้อยๆ แซมเปิ้ลในแต่ละสับแบนด์จะถูกควอนไตซ์ ทำให้ไปยกระดับสัญญาณรบกวนขึ้นมา เมื่อสัญญาณถูกสร้างขึ้นมาใหม่ ตัวกรองจะบังคับให้สัญญาณรบกวนอยู่ภายในสับแบนด์นั้นๆ ที่ซึ่งจะถูกบดบังโดยสัญญาณ A สับแบนด์แซมเปิ้ลที่มีความละเอียดทางความถี่สูง B ควอนไตซ์ให้ได้บิตเรทตามที่ต้องการ C เมื่อสร้างสัญญาณขึ้นมาใหม่ สัญญาณรบกวนจะถูกบดบังโดยสัญญาณนั้นๆ

ในการออกแบบบางวิธียังได้มีการป้อนสัญญาณเพื่อจะเปลี่ยนจากโดเมนเวลาเป็นโดเมนความถี่ เช่นการใช้ FFT เพื่อคำนวณพลังงานในแต่ละสับแบนด์ โดยค่านี้จะถูกป้อนไปให้รูปแบบของไซโคอะคูสติก เพื่อตัดสินเส้นโค้งผลรวมการบดบังที่จะให้กับสัญญาณในบล็อกนั้น โดยเฉพาะตัวเข้ารหัสจะพิจารณาพลังงานในแต่ละสับแบนด์ เพื่อดูว่าสับแบนด์ใดมีข้อมูลเสียงที่ได้ยินอยู่ การคำนวณเกิดขึ้นเพื่อดูพลังงานเฉลี่ยของแต่ละสับแบนด์ของข้อมูลบล็อกนั้น โดยค่าพลังงานเฉลี่ยนี้จะถูกใช้เพื่อคำนวณระดับการบดบัง(Masking level) และการบดบังของสัญญาณจากสับแบนด์ที่อยู่ใกล้เคียง ในที่สุด ค่าขีดเริ่มน้อยสุดของการได้ยิน(Minimum Masking Threshold) จะถูกคำนวณออกมาในแต่ละสับแบนด์เพื่อจะได้ระดับการบดบังขั้นสุดท้าย พลังงานสูงสุดที่อยู่ในแต่ละสับแบนด์จะถูกคำนวณ และ เปรียบเทียบกับระดับการบดบัง สับแบนด์ที่ไม่มีข้อมูลที่ได้ยินจะไม่ถูกโค้ด เสียงที่ถูกบดบังโดยเสียงที่ดังกว่าจะไม่ถูกโค้ด และในบางกรณีสับแบนด์สามารถจะบดบังสับแบนด์ข้างเคียงได้ ทำให้ไม่โค้ด ส่วนสับแบนด์ที่มีระดับพลังงานอยู่สูงกว่าระดับขีดเริ่มการบดบังน้อยสุด จะมีข้อมูลในส่วนที่เราได้ยิน และจะถูกโค้ด

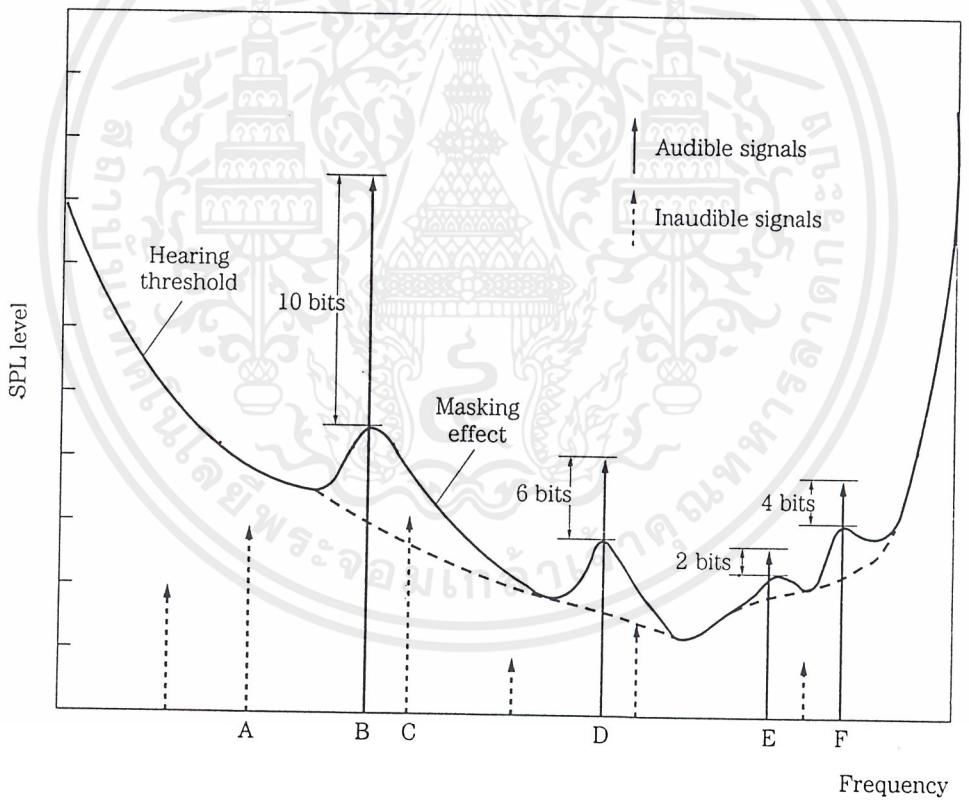
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 การโค้ดแบบสับแบนด์จะแบ่งสัญญาณออกเป็นสับแบนด์แคบๆ และคำนวณระดับพลังงานเฉลี่ย อีกทั้งระดับบดบัง แล้วก็จะควอนไตซ์แซมเปิ้ลในแต่ละสับแบนด์ A เอาท์พุทของตัวกรอง 24 สับแบนด์ B การคำนวณระดับเฉลี่ยในแต่ละสับแบนด์ C การคำนวณระดับการบดบังของแต่ละสับแบนด์ D สับแบนด์ที่ไม่ได้ยินเสียงจะไม่ถูกโค้ด E จำนวนบิตจะถูกจัดสรรให้ตามระดับของสัญญาณที่เหนือระดับบดบัง

เอกสารนี้เผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณจะเปรียบเทียบอัตราส่วนของพลังงานสูงสุดกับระดับบดบังในแต่ละสับแบนด์ ดังรูปที่ 4.15 แสดงเส้นแนวตั้งซึ่งเป็นพลังงานสูงสุด และระดับขีดเริ่มต่ำสุดของการได้ยิน ส่วนสัญญาณที่ต่ำกว่าขีดเริ่มการบดบังจะไม่ถูกโค้ด อย่างเช่น สัญญาณ A ต่ำกว่าเส้นโค้งจะไม่ถูกโค้ด ส่วนสัญญาณ B จะถูกโค้ด แต่การเกิดของมันได้ลดระดับแอมปลิจูดสัมพัทธ์ลง ส่วนของสัญญาณ B ระหว่างเส้นโค้งน้อยสุดและ เส้นโค้งบดบังแสดงถึงจำนวนแสดงถึงจำนวนบิตที่ถูกโค้ดเมื่อเกิดการบดบังขึ้น ในอีกแง่หนึ่ง แทนที่จะใช้ S/N Ratio เราจะใช้ S/M Ratio แทน โดยจะคำนวณมาจากทุกๆสับแบนด์ และผลต่างของค่าสัญญาณสูงสุด และขีดเริ่มต่ำสุดของการได้ยิน ถูกนำมาใช้ในการพิจารณาจำนวนบิตที่จะกำหนดให้ในแต่ละสับแบนด์



รูปที่ 4.15 การจักสรรบิตจะกำหนดจำนวนบิตตามการได้ยิน ส่วนที่ไม่ได้ยินจะไม่มีการโค้ด

จำนวนบิตที่ให้กับแต่ละสับแบนด์จะต้องเพียงพอที่จะให้ระดับของสัญญาณรบกวนต่ำกว่าระดับบดบัง ในตอนเราสร้างสัญญาณขึ้นมาใหม่ จำนวนของบิตนั้นได้มาจากค่า SMR Ratio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโค้ดแบบทรานฟอร์ม

ในการโค้ดแบบนี้ บล็อกของสัญญาฯในโดเมนเวลาจะเปลี่ยนเป็นโดเมนความถี่ โดยอาจใช้ Fast Fourier Transform มาช่วย สัมประสิทธิ์สเปกตรัมที่ได้จาก FFT จะถูกควอนไตซ์ตามรูปแบบของไซโคอะคูสติก ส่วนที่ถูกบดบังจะถูกกำจัดออกไป เนื่องจากแซมเปิ้ลที่อยู่ในโดเมนเวลาแล้วเปลี่ยนเป็นโดเมนความถี่จะให้เราได้สัมประสิทธ์สเปกตรัม จำนวนของสัมประสิทธิ์บางครั้งเราเรียกว่า จำนวนถังความถี่ เช่น 512 จุด จะทำให้ได้สัมประสิทธิ์ความถี่ 256 สเปกตรัม หรือ 256 ถังความถี่ สัมประสิทธิ์อาจจะเป็น 512 หรือ 1024 หรือมากกว่า จะถูกแบ่งให้เป็น 32 สับแบนด์ โดยสเปกตรัมเหล่านี้แสดงถึงบล็อกของสัญญาณในโดเมนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

DSP56009 Evaluation Module

DSP56009EVM เป็นบอร์ดที่ใช้สำหรับออกแบบและพัฒนาระบบเสียงเชิงดิจิตอล ซึ่งตัวประมวลผลหลักของโมดูลนี้เป็นตัวประมวลผลทางดิจิตอล (Digital Signal Processor) ของบริษัทโมโตโรล่า เบอร์ DSP56009

โดยมันมีคำสั่งรองรับการประมวลผลเชิงดิจิตอลโดยตรงนอกจากนี้ยังรองรับ Protocal ได้อีกหลายแบบด้วย

บนบอร์ดประกอบไปด้วยส่วนสำคัญต่างๆดังนี้

DSP56009 เป็น Digital Signal Processor ขนาด 24 บิต ทำงานที่ความถี่ 81 MHz

- สามารถทำงานได้ด้วยความเร็ว 40.5 ล้านคำสั่งต่อวินาที
- ประกอบด้วยบัสรับส่งข้อมูลภายในขนาด 24 บิต 4 ชุด , แอดเดรสบัสภายในขนาด 16 บิต 3 ชุด ที่สามารถอ้างถึงได้พร้อมกัน
- มีหน่วยความจำภายในทั้งแบบ ROM และ RAM ได้แก่

  a) X-data RAM ขนาด 4608 x 24 บิต และ X-data ROM ขนาด 3072 x 24 บิต

  b) Y-data RAM ขนาด 4352 x 24 บิต และ Y-data RAM ขนาด1792 x 24 บิต

  c) Program RAM ขนาด 512 x 24 บิต และ Program ROM ขนาด 10240 x 24 บิต

  d) สามารถแปลงเนื้อที่ X และ Y data RAM ขนาด 2304 x 24 บิตให้เสมือนเป็น Program RAMได้

  e) มี OnCE (On Chip Emulator) port สำหรับติดต่อกับ host ภายนอก

  f) มี Phase Lock Loop ที่สามารถโปรแกรมได้ใช้สำหรับสร้างฐานความถี่

หน่วยความจำภายนอก (External memory)

- มีหน่วยความจำแบบ SRAM และ non-volatile RAM ขนาด 8192 ไบต์ ที่สามารถเคลื่อนย้ายข้อมูลระหว่างกันแบบเป็นบล็อกได้
- มี slot มาตรฐานขนาด 30-pin แบบ SIMM ที่สามารถเพิ่มหน่วยความจำแบบ DRAM ได้สูงสุดถึง 4 M x 8

ส่วนติดต่อกับผู้ใช้ (User interface)

- ควบคุมโดยไมโครคอนโทรลเลอร์เบอร์ MC68HC711E9 ที่อนุญาตให้ผู้ใช้โปรแกรมได้ด้วยรหัสเฉพาะแบบ 68HC11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีหน่วยแสดงผลแบบ LCD (2 บรรทัด 16 ตัวอักษร / บรรทัด) สั่งงานได้ด้วยสวิทช์ 4 ตัว และผู้ใช้สามารถเพิ่มสวิทช์ควบคุมได้อีก 16 ตัว
- มีชิปเบอร์ MC68705k1 สำหรับแปลงการสื่อสารจากแบบ RS-232 ไปเป็น OnCE
- ควบคุมด้วยรีโมทคอนโทรลแบบอินฟาเรดได้

ส่วนรับส่งข้อมูล (Input / Output)

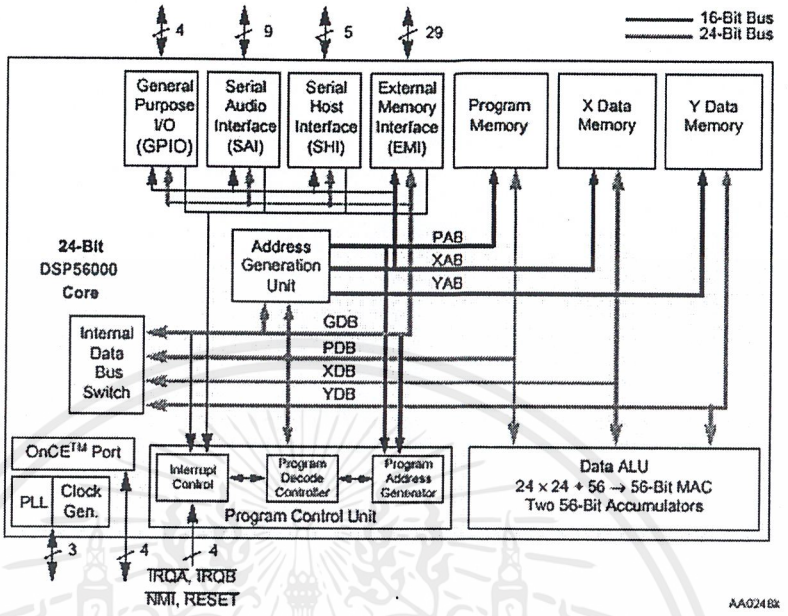
- A / D แบบ 2 ช่องสัญญาณขนาด 20 บิต
- D / A แบบ 6 ช่องสัญญาณขนาด 18 บิต
- สามารถเลือกความถี่สุ่ม (Sampling frequency) สำหรับ A / D และ D / A ได้ 2 แบบ คือ 44.1 และ 48 kHz
- มีตัวลดทอนสัญญาณเอาท์พุทแบบโปรแกรมได้
- สามารถรับ-ส่งสัญญาณผ่านทางแสงได้ โดยใช้มาตรฐานแบบ SPDIF / CP340
- มี connector ขนาด 50 pin ใช้กำหนดอินพุทและเอาท์พุทอื่นๆได้

5.1 สถาปัตยกรรมภายในชิป DSP56009 และโครงสร้างของบัส

(Architectural overview and bus structure)

สถาปัตยกรรมภายในของ DSP56009 ได้ถูกออกแบบให้รองรับการประมวลผลสัญญาณเชิงเลข (Digital Signal Processor) ได้อย่างมีประสิทธิภาพ โดยเฉพาะอย่างยิ่งการประยุกต์ใช้งานกับการประมวลผลสัญญาณเสียงเชิงเลข (Digital Audio Application) จากรูปที่ 5.1 จะเห็นว่า DSP56009 ได้ถูกออกแบบให้ส่วนคำนวณข้อมูล (Data ALU) , ส่วนอ้างอิงหน่วยความจำ (Address Generation Unit) และส่วนควบคุมโปรแกรม (Program Control Unit) แยกกันทำงานเป็นอิสระต่อกันทั้งหมด นั่นคือชิปจะทำการประมวลผลทั้ง 3 ส่วนไปพร้อมๆกัน ซึ่งทำให้การทำงานโดยรวมรวดเร็วขึ้นมาก รายละเอียดของส่วนต่างๆเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 สถาปัตยกรรมภายในของ DSP56009

5.1.1 Data Bus ประกอบด้วยบัสขนาด 24 บิต แบบ 2 ทิศทาง (Bidirectional)จำนวน 4 ชุด คือ X data bus (XDB) , Y data bus (YDB) , Program data bus (PDB) และ Global data bus (GDB)

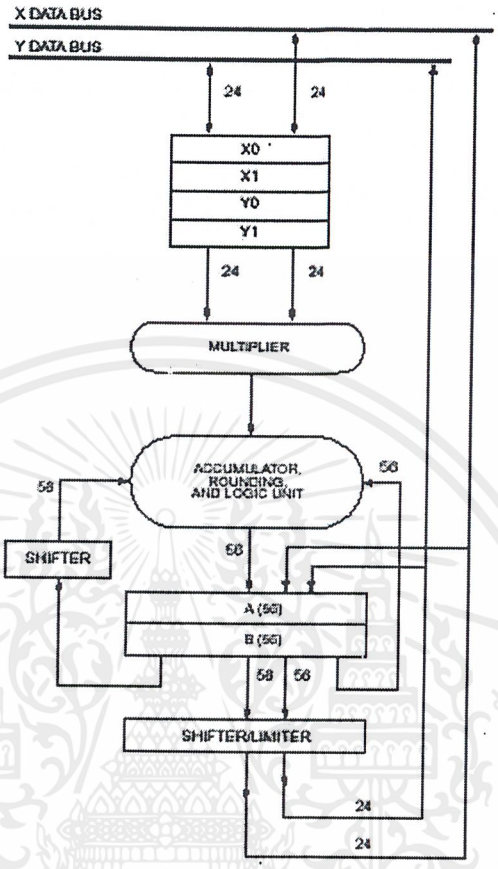
XDB และ YDB สามารถใช้ร่วมกันเพื่อประมวลผลทีละ 48 บิต , PDB ใช้สำหรับส่งผ่านคำสั่งไปประมวลผลโดยทำงานขนานกับบัสชุดอื่นๆ ส่วนการอ้างอิงข้อมูลรอบนอก (Peripheral) จะกระทำผ่าน GDB ข้อมูลในบัสทุกชุดสามารถถ่ายโอนระหว่างกันได้โดยผ่านทาง Internal Data Bus Switch

5.1.2 Address Bus ประกอบด้วยบัสขนาด 16 บิต แบบทิศทางเดียว (Unidirectional) จำนวน 3 ชุด ได้แก่ X address bus (XAB) , Y address bus (YAB) และ program address bus (PAB) โดยทุกชุดจะถูกควบคุมจาก Address Generation Unit

การอ้างอิงหน่วยความจำภายนอกสามารถทำได้สำหรับหน่วยความจำทุกแบบ แต่จะอ้างอิงได้เพียง 1 ตำแหน่งต่อ 1 รอบคำสั่ง (Instruction Cycle) เท่านั้น

5.1.3 Data ALU ประกอบด้วยรีจิสเตอร์และหน่วยคำนวณดังรูปที่ 5.2 รายละเอียดมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 Data ALU Block Diagram

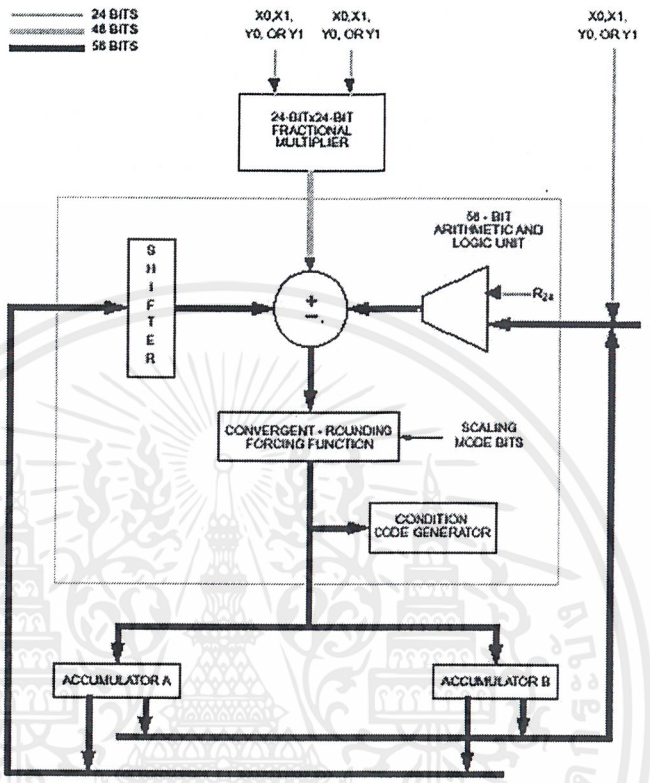
- Data ALU Input Register (X1 , X0 , Y1 , Y0) เป็นรีจิสเตอร์สำหรับข้อมูลทั่วไป (General-purpose register) ซึ่งสามารถใช้เป็นรีจิสเตอร์ขนาด 48 บิต X และ Y ได้ โดยมีการจัดเรียงลำดับข้อมูลเป็น X1:X0 และ Y1:Y0 นั่นคือ X1 และ Y1 เป็นข้อมูลที่มีความสำคัญมากที่สุด (Most Significant Word)

รีจิสเตอร์เหล่านี้ทำหน้าที่เป็น Input buffer ระหว่าง XDB , YDB กับ MAC Unit (Multiply -Accumulator / Logic Unit) โดยเขียนและอ่านผ่านทาง XDB และ YDB

- MAC and Logic Unit เป็นหน่วยประมวลผลหลักของ DSP56009 ในกรณีที่ทำการประมวลผลคำสั่งแบบการคำนวณเลข (Arithmetic) หน่วยนี้จะสามารถรับอินพุทได้มากที่สุด 3 แหล่ง และจะให้ผลลัพธ์ขนาด 56 บิตใน Accumulator A หรือ B (A2:A1:A0 , B2:B1:B0)โดยที่การคำนวณจะทำงานได้เป็นอิสระขนานกับการอ้างถึง XDB และ YDB

การประมวลผลทางตรรกะ (Logic) ได้แก่ AND , OR , EOR (Exclusive OR) และ NOT จะใช้เพียงรีจิสเตอร์ A1 หรือ B1 ขนาด 24 บิตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

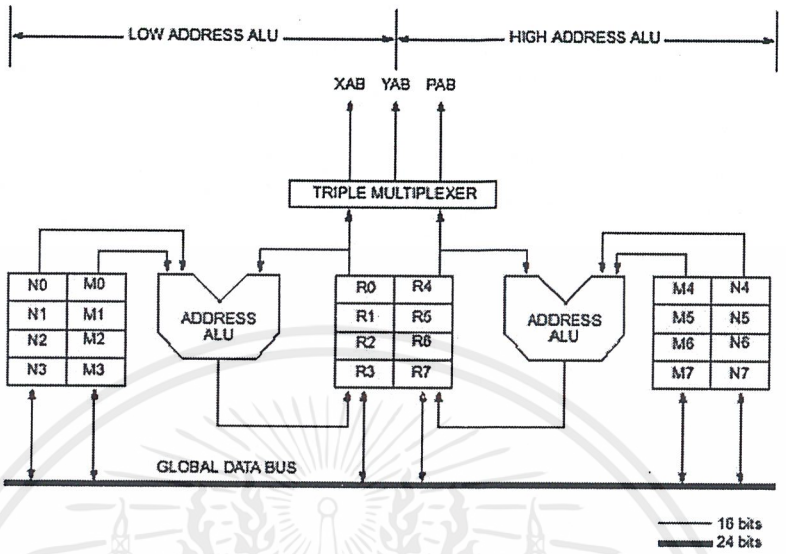


รูปที่ 5.3 MAC Unit

- Shifter and Limiter Shifter จะใช้เมื่อมีการเรียกใช้คำสั่งที่มีการเลื่อนบิตไปทางซ้ายหรือขวาเท่านั้น ส่วน Limiter จะใช้จำกัดค่าที่อ่านจาก Accumulator A หรือ B (ขนาด 56 บิต) ไปยัง XDB หรือ YDB (ขนาด 24 บิต) โดยจะทำการประมาณค่าให้ใกล้เคียงกับค่าใน Accumulator ให้มากที่สุด

5.1.4 Address Generation Unit เป็นหน่วยสำหรับคำนวณและสร้างสัญญาณที่ระบุตำแหน่งของหน่วยความจำที่จะนำไปใช้งานโดยส่วนอื่นๆของการประมวลผล ประกอบด้วยรีจิสเตอร์และหน่วยคำนวณดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 Address Generation Unit

- Address Register (Rn) ประกอบไปด้วยรีจิสเตอร์ขนาด 16 บิต 2 ชุด ชุดละ 4 ตัว (R0 – R3 และ R4 – R7) ซึ่งใช้ชี้ตำแหน่งของหน่วยความจำ โดยรีจิสเตอร์ทุกตัวจะเขียนและอ่านโดยผ่านทาง GDB

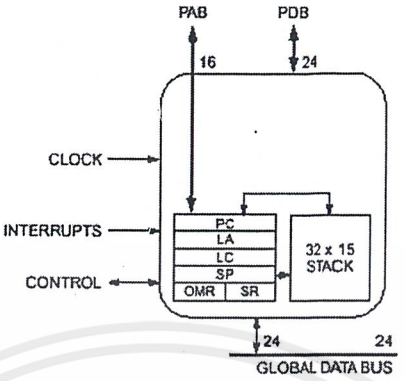
- Offset Register (Nn) ประกอบไปด้วยรีจิสเตอร์ขนาด 16 บิต 2 ชุด ชุดละ 4 ตัว (N0 – N3 และ N4 – N7) ใช้สำหรับเก็บค่า Offset ที่จะนำไปเพิ่มหรือลดค่าใน Rn โดยรีจิสเตอร์ทุกตัวจะเขียนและอ่านโดยผ่านทาง GDB

- Modifier Register (Mn) ประกอบไปด้วยรีจิสเตอร์ขนาด 16 บิต 2 ชุด ชุดละ 4 ตัว (M0 – M3 และ M4 – M7) ใช้สำหรับการอ้างอิงหน่วยความจำแบบ Modulo โดยรีจิสเตอร์ทุกตัวจะเขียนและอ่านโดยผ่านทาง GDB

- Address ALU เป็นหน่วยคำนวณตำแหน่งของหน่วยความจำโดยอ้างอิงข้อมูลจากรีจิสเตอร์ Rn , Nn และ Mn

5.1.5 Program Controller เป็นส่วนควบคุมการทำงานของชิป แสดงส่วนต่างๆได้ดังรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

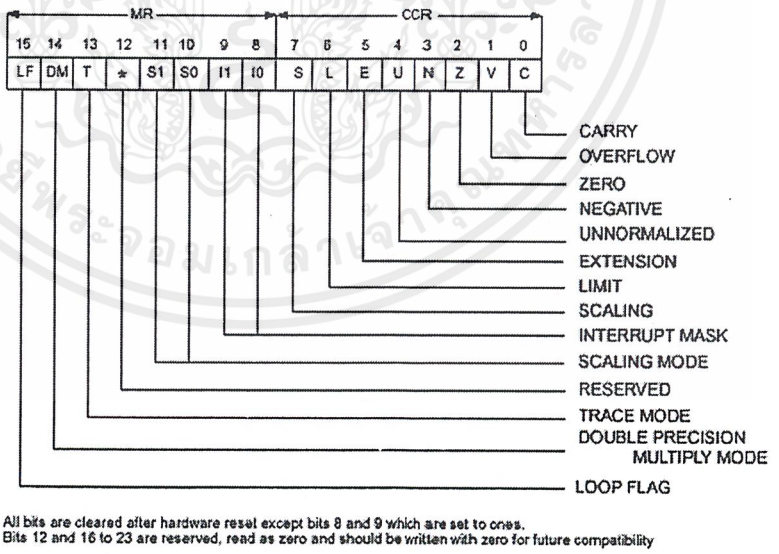


รูปที่ 5.5 Program Controller Block Diagram

- รีจิสเตอร์ ประกอบด้วยรีจิสเตอร์ควบคุมการทำงานดังนี้

* Program Counter (PC) รีจิสเตอร์ขนาด 16 บิต ซึ่งใช้เก็บค่าตำแหน่งของคำสั่งถัดไปที่จะทำการอ่าน (Fetch) เข้าสู่ส่วนถอดรหัสคำสั่ง (Decode)

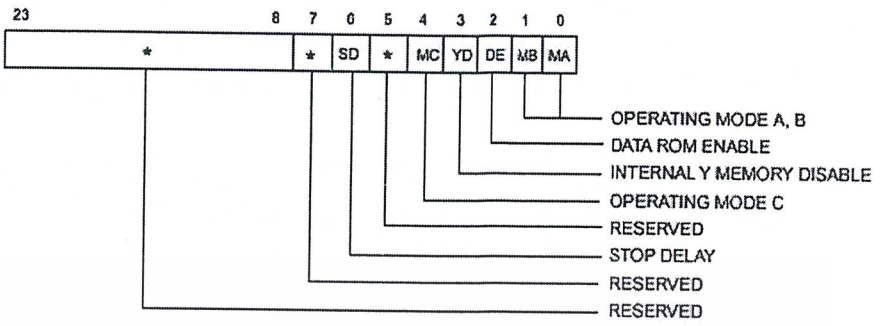
* Status Register (SR) รีจิสเตอร์ขนาด 16 บิต ประกอบด้วย Mode Register (MR) (ใน 8 บิตบน) Condition Code Register (CCR) (ใน 8 บิตล่าง) รายละเอียดของแต่ละบิตแสดงดังรูปที่ 5.6



รูปที่ 5.6 Status Register

* Operating Mode Register (OMR) รีจิสเตอร์ขนาด 24 บิต แต่ใช้เพียง 5 บิตเท่านั้น รายละเอียดแสดงในรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 Operating Mode Register

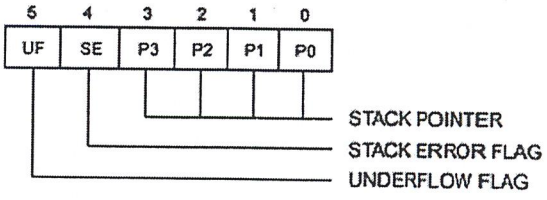
* Loop Address Register (LA) รีจิสเตอร์ขนาด 16 บิต ซึ่งใช้เก็บตำแหน่งของคำสั่งสุดท้ายของ Loop เมื่อมีการเรียกใช้ Hardware Loop (คำสั่ง DO) มีหน้าที่ตรวจสอบและควบคุมจำนวน Loop ที่จะต้องวน โดยจะทำงานร่วมกันกับ Loop Counter Register (LC)

* Loop Counter Register (LC) รีจิสเตอร์ขนาด 16 บิต ซึ่งใช้เก็บจำนวนครั้งที่ต้องทำซ้ำใน Hardware Loop

* System Stack (SS) เป็นหน่วยความจำแบบ 32 บิต แบ่งเป็น 2 ส่วนๆละ 16 บิต คือ SSH และ SSL โดย SSH จะใช้สำหรับเก็บค่าภายใน PC ส่วน SSL จะใช้เก็บข้อมูลภายใน SR เมื่อมีการเรียกใช้โปรแกรมย่อย (Subroutine) นอกจากนี้ยังใช้เก็บค่าใน LA และ LC ด้วยในกรณีที่มีการเรียกใช้ Hardware Loop

หน่วยความจำส่วนนี้มีด้วยกัน 15 ตำแหน่งนั่นหมายความว่า DSP56009 อนุญาตให้ผู้ใช้เรียกใช้โปรแกรมย่อยซ้อนกันได้มากที่สุด 15 ระดับ หรือเรียกใช้ Hardware Loop ซ้อนกันได้มากที่สุด 7 ระดับ

* Stack Pointer Register (SP) เป็นรีจิสเตอร์ขนาด 6 บิต ใช้สำหรับชี้ตำแหน่งของ System Stack 4 บิต ส่วนอีก 2 บิตที่เหลือจะใช้บอกสถานะ (Status Flag) ของ SS



รูปที่ 5.8 Stack Pointer Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถาปัตยกรรมภายใน Program Controller Unit

* Program Decode Control (PDC) เป็นส่วนถอดรหัสคำสั่งที่รับมาจากหน่วยความจำโปรแกรม (Program Memory) ซึ่งจะสร้างสัญญาณควบคุมที่จำเป็นสำหรับคำสั่งนั้นๆเพื่อให้ส่วนอื่นนำไปใช้ต่อไป

* Program Address Generator (PAG) ประกอบด้วยรีจิสเตอร์ PC , SP , SS , OMR , SR , LC และ LA ซึ่งจะใช้อ้างอิงตำแหน่งของหน่วยความจำเพื่อการควบคุมการทำงานต่อไป

* Program Interrupt Controller (PIC) เป็นส่วนรับการร้องขออินเทอรัพท์ (Interrupt Request) ที่เกิดขึ้นทั้งหมดทั้งภายในและภายนอกชิป โดยจะทำการสร้าง Interrupt Vector ส่งให้กับรีจิสเตอร์ PC เป็นการเรียกใช้โปรแกรมบริการอินเทอรัพท์ (Interrupt Service Routine) ซึ่งลำดับความสำคัญของอินเทอรัพท์แต่ละแบบนั้นถูกกำหนดไว้ตาม Interrupt Priority Level (IPL) ดังตารางที่ 1

Priority	Interrupt
Level 3 (Nonmaskable)	
Highest	Hardware $\overline{\text{RESET}}$
	Illegal Instruction
	$\overline{\text{NMI}}$
	Stack Error
	Trace
Lowest	SWI
Levels 0, 1, 2 (Maskable)	
Highest	$\overline{\text{IRQA}}$ (External Interrupt)
	$\overline{\text{IRQB}}$ (External Interrupt)
	SAI Receiver Exception
	SAI Transmitter Exception
	SAI Left Channel Receiver
	SAI Left Channel Transmitter
	SAI Right Channel Receiver
	SAI Right Channel Transmitter
	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error
	SHI Receive FIFO Full
	SHI Transmit Data
	SHI Receive FIFO Not Empty
	EMI EBAR0 Memory Wrap
	EMI EBAR1 Memory Wrap
	EMI Read Data
Lowest	EMI Write Data

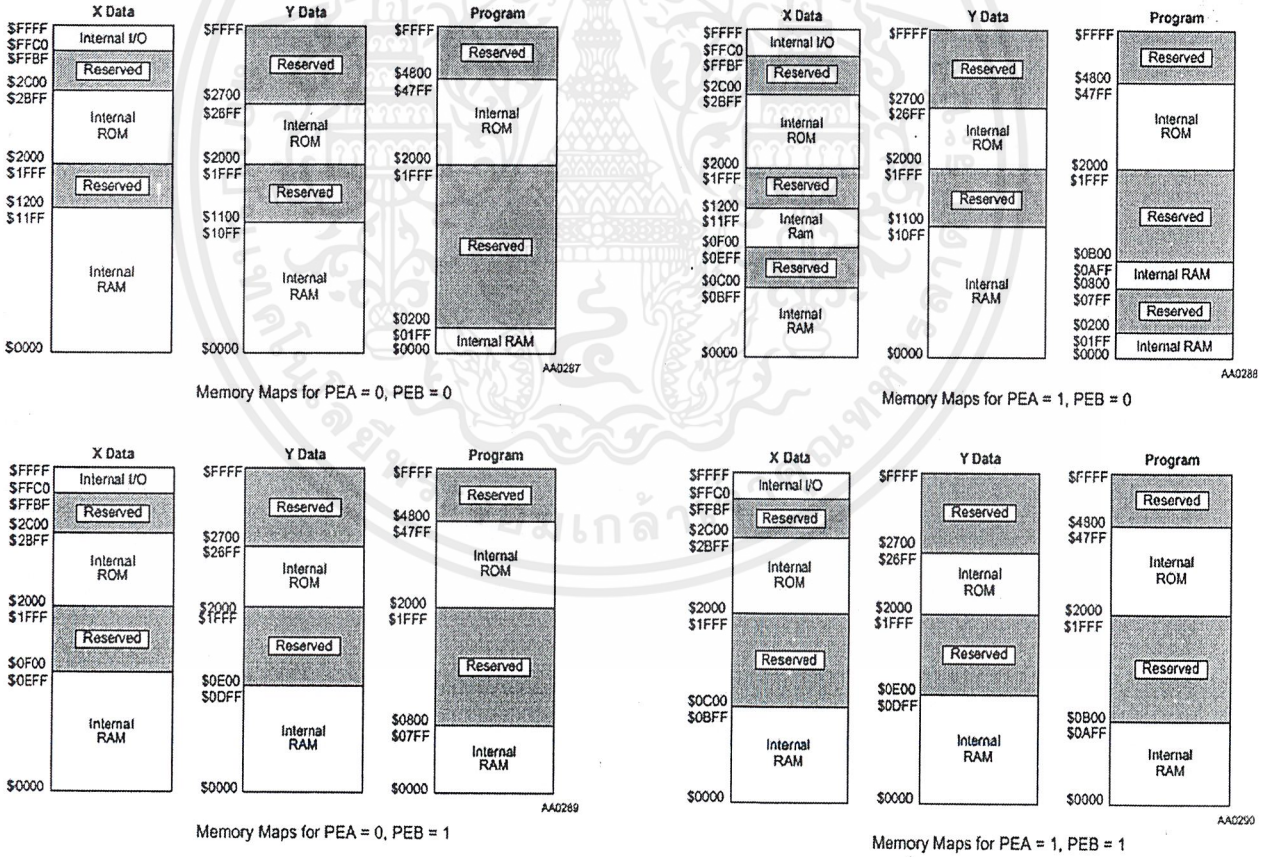
ตารางที่ 1 แสดง Interrupt Priority Level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การจัดสรรหน่วยความจำของ DSP56009 (Memory Map)

โปรแกรมของ DSP 56009 และข้อมูลที่อยู่ในหน่วยความจำจะเป็นอิสระต่อกัน และในชิป ข้อมูลในหน่วยความจำจะถูกแบ่งออกเป็นสองหน่วยความจำย่อย เรียกว่า X และ Y และก็จะมีสองหน่วยความจำถาวรในหน่วยความจำ X และ Y และหน่วยความจำถาวรบูทสแท็ป (Bootstrap ROM) สามารถครอบคลุมโปรแกรมหน่วยความจำชั่วคราว หน่วยความจำข้อมูลสามารถจะถูกแบ่งออกเป็นสองส่วนเพื่อจะทำงานร่วมกับสองแอดเดรสของ ALU เมื่อจะได้นำข้อมูลทีละสองข้อมูลป้อนไปเป็นข้อมูลเข้า ALU ได้อย่างทันทีทันใด

ด้วยการเปลี่ยนค่าบิต PEA,PEB (Program Ram Enable bits) ในโหมดปฏิบัติการรีจิสเตอร์ OMR จะได้โครงสร้างของ Memory Map ที่ต่างกันได้ 4 รูปแบบ ดังรูปที่ 5.9



รูปที่ 5.9 Memory Map แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 DSP 56009 Data and Program Memory

- **ข้อมูลหน่วยความจำถาวร X** ข้อมูลหน่วยความจำถาวร X (X Data Rom) จะครอบคลุมตำแหน่ง $2000-$2BFF ในตำแหน่งของข้อมูลหน่วยความจำถาวร
- **ข้อมูลหน่วยความจำถาวร** Y จะครอบคลุมตำแหน่ง $2000-$26FF ในตำแหน่งของข้อมูลหน่วยความจำถาวร Y
- **โปรแกรมรอม** จะกินพื้นที่ $2000-$47FF ในตำแหน่งของโปรมแกรมรอม
- **บูทสแท็ปรอม** จะกินพื้นที่ 0-31($0-$1F) และ 256-287($100-$11F) ในสองพื้นที่ในบูทสแท็ปแมป บูทสแท็ปรอมจะถูกโปรแกรมมาจากโรงงานเพื่อจะแสดงการกระทำทางบูทสแท็ปตามการรีเซ็ทของฮาร์ดแวร์ มันจะกระโดดไปที่ตำแหน่งเริ่มต้นของรอม(P:$2000) หรือ ดาวน์โหลดไปถึง 512 คำของโปรมแกรมผู้ใช้(user program) จาก EPROM จะถูกรวมกับพอร์ต EMI หรือจากพอร์ต SHI กิจกรรมของบูทสแท็ปรอมคือเข้าควบคุมโดยโหมดบิตใน OMR เมื่ออยู่ในโหมดบูทสแท็ป 512 คำแรกของโปรแกรมแรมจะไม่สามารถอ่านได้แต่เขียนได้

โปรแกรมจะถูกโหลดจาก EPROM ภายนอก ถ้า MC:MB:MA = 001 โปรแกรมแรมภายในด้วย 1536 ไบท์จาก EPROM ที่ต่ออยู่กับ EMI และ EPROM จะอยู่ในตำแหน่งที่ EMI แอดเดรส $0 เราจะสันนิษฐานว่า EPROM จะถูกเลือกผ่านขา GPIO3 และเอาท์พุทของขานี้จะถูกโปรแกรมให้เป็นแบบชนิด active high/active low

ไบต์จะถูกรวมให้เป็น 512 หรือ 24-bit word และจัดเก็บในโปรแกรมแรมที่อยู่ติดกัน เริ่มต้นที่ P:$0000

โปรแกรมจะถูกโหลดจาก SHI ในโหมด SPI ถ้า MC:MB:MA=101 หรือ โหมด IIC ถ้า MC:MB:MA=111 โปรแกรมแรมภายในถูกโหลดถึง 512 คำซึ่งมีขนาดความยาว24 บิต จะได้รับทาง SHI

- **ตำแหน่งหน่วยความจำที่สงวนไว้** (Reserve Memory Spaces) ตำแหน่งนี้ผู้ใช้ไม่สามารถใช้งานได้ จะถูกเก็บไว้ใช้ในการขยายในอนาคต การเขียนไปที่ตำแหน่งหน่วยความจำสงวนจะไม่มีผลอะไร ส่วนการอ่านจะส่งค่า $000005 ออกมา ถ้ามีการเข้าไปเอาข้อมูลของตำแหน่งเหล่านี้ ก็จะส่งค่า $000005 ออกมาเช่นกัน ซึ่งก็คือ Op-code ของคำสั่งที่ไม่ถูกต้องนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 โหมดการอ้างอิงหน่วยความจำแบบใช้รีจิสเตอร์ทางอ้อม (Address Register Indirect Modes)

วิธีการอ้างอิงข้อมูลในหน่วยความจำของ DSP56009 เราสามารถทำได้หลายแบบ ซึ่งวิธีหนึ่งที่สำคัญก็คือการใช้ Address Register (Rn) เป็นตัวชี้ (Pointer) ตำแหน่งหน่วยความจำ ซึ่งชุดคำสั่งของ DSP56009 มีวิธีในการ Update ค่าตัวชี้ได้อย่างรวดเร็วและหลากหลาย ได้แก่

- No Update ค่าภายใน Rn จะคงที่หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE A1,X:(R0) ย้ายข้อมูลจากรีจิสเตอร์ A1 ไปยัง X-Memory ที่ชี้โดยรีจิสเตอร์ R0 โดยไม่มีการเปลี่ยนค่าใน R0 หลังจาก ทำคำสั่งเสร็จสิ้น

- Post Increment by 1 ค่าภายใน Rn จะมีค่าเพิ่มขึ้น 1 หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE B0,Y:(R1)+ ย้ายข้อมูลจากรีจิสเตอร์ B0 ไปยัง Y-Memory ที่ชี้โดยรีจิสเตอร์ R1 โดยค่าใน R1 จะเพิ่มขึ้น 1 หลังจาก ทำคำสั่งเสร็จสิ้น

- Post Decrement by 1 ค่าภายใน Rn จะมีค่าลดลง 1 หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE Y0,Y:(R3)- ย้ายข้อมูลจากรีจิสเตอร์ Y0 ไปยัง Y-Memory ที่ชี้โดยรีจิสเตอร์ R3 โดยค่าใน R3 จะลดลง 1 หลังจาก ทำคำสั่งเสร็จสิ้น

- Post Increment by Offset Nn ค่าภายใน Rn จะมีค่าเพิ่มขึ้น Nn หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE X1,X:(R2)+N2 ย้ายข้อมูลจากรีจิสเตอร์ X1 ไปยัง X-Memory ที่ชี้โดยรีจิสเตอร์ R2 โดยค่าใน R2 จะถูกบวกเพิ่มด้วยค่าใน N2 หลังจาก ทำคำสั่งเสร็จสิ้น

- Post Decrement by Offset Nn ค่าภายใน Rn จะมีค่าลดลง Nn หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE X:(R4)-N4,A0 ย้ายข้อมูลจากหน่วยความจำ X-Memory ที่ชี้โดยรีจิสเตอร์ R4 ไปยังรีจิสเตอร์ A0 โดยค่าใน R4 จะถูกลดค่าลงด้วยค่าใน N4 หลังจาก ทำคำสั่งเสร็จสิ้น

- Index by Offset Nn อ้างถึงหน่วยความจำตำแหน่งที่ชี้โดยค่าผลบวกระหว่างค่าภายใน Rn และ Nn และรีจิสเตอร์ทั้งคู่จะมีค่าคงเดิม หลังจากที่ปฏิบัติคำสั่ง

  เช่น MOVE Y1,X:(R6+N6) ย้ายข้อมูลจากรีจิสเตอร์ Y1 ไปยัง X-Memory ที่ชี้โดยค่าผลรวมระหว่างรีจิสเตอร์ R6 กับ N6

- Pre Decrement by 1 ค่าภายใน Rn จะมีค่าลดลง 1 ก่อนที่จะปฏิบัติคำสั่ง

  เช่น MOVE X:-(R5),B1 ลดค่าในรีจิสเตอร์ R5 ลง 1 ก่อนที่จะย้ายข้อมูลจากหน่วยความจำ X-Memory ที่ชี้โดยรีจิสเตอร์ R5 ไปยังรีจิสเตอร์ B1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 รูปแบบการปรับค่าอ้างอิงหน่วยความจำ (Address Modifier Type)

เราสามารถกำหนดรูปแบบดังกล่าวได้ทั้งหมด 3 แบบ คือ Linear , Modulo และ Reverse-carry Modifier โดยจะใช้รีจิสเตอร์ Mn ในการกำหนดรูปแบบ

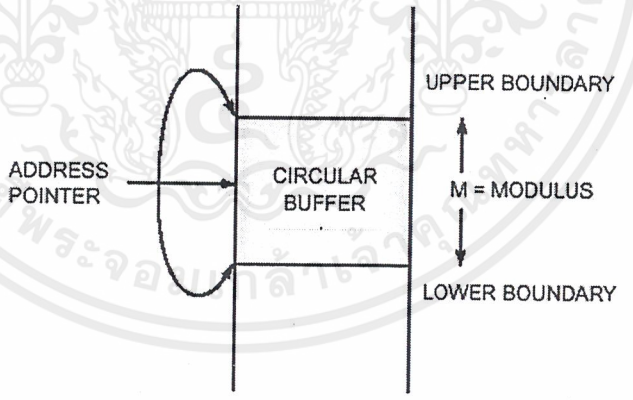
- **Linear Modifier** (Mn = $FFFF) เป็นการเปลี่ยนค่า Rn แบบเชิงเส้น
- **Modulo Modifier** (Mn = MODULUS – 1) การระบุตำแหน่งแอดเดรสจะถูกแสดงด้วย modulo M ซึ่ง M มีค่าความกว้างจาก 2 ถึง +32,768 (ดูตามตารางที่ 2) ค่าตัวเลขของ Modulo M ทำให้ค่าแอดเดรสรีจีสเตอร์ ยังคงอยู่ภายในช่วงความกว้างของขนาดของ M สามารถอธิบายได้โดยขอบเขตล่างและขอบเขตบนของแอดเดรส (ดูรูปที่ 5.10) ค่า m = M-1 จะถูกเก็บไว้ในโมดิไฟเออร์รีจีสเตอร์(Modifier Register) , Mn ค่าของขอบเขตล่าง(แอดเดรสฐาน) นั้นต้องมีค่าศูนย์ใน k LSBs ซึ่ง $2^k \geq M$ และดังนั้นต้องเป็นจำนวนเท่าของ 2^k ขอบเขตบนจะมีค่าเท่ากับค่าของขอบเขตล่างบวกขนาดของ modulo แล้วลบ 1 (แอดเดรสฐาน + M -1) ในกรณี $M \leq 2^k$ ดังนั้น M จะถูกเลือก ชุดของลำดับของบล็อกความจำ(memory block) (แต่ละบล็อกมีความยาว 2^k จะถูกสร้างขึ้นมา ที่ซึ่ง circular buffer นี้ สามารถถูกติดตั้งลงไปได้ ถ้า $M < 2^k$ ก็จะมีช่องว่างระหว่าง circular buffer ที่อยู่ติดกันคือ (2^k) - M ตัวอย่างเช่น ถ้าต้องการสร้าง circular buffer ที่มี 21 ชั้น คือ M = 21 และขอบเขตล่างของแอดเดรสจะต้องมีค่า LSBs 5 ค่าเท่ากับศูนย์ ($2^k \geq 21$ ดังนั้น $k \geq 5$) ส่วนรีจีสเตอร์ Mn จะถูกโหลดข้อมูลเข้าไปด้วยค่า 20 ขอบเขตล่างบางทีอาจจะเลือกค่าเป็น 0, 32 , 64 , 96 , 128 ,160, ... ขอบเขตบนของบัฟเฟอร์ค่าขอบเขตล่างบวกด้วย 21 และจะมีที่ตำแหน่งช่องว่างที่ไม่ได้ใช้ในหน่วยความจำซึ่งอยู่ระหว่างตำแหน่งแอดเดรสที่อยู่บนสุดและตำแน่งแอดเดรสล่างสุดของบัฟเฟอร์ที่ใช้งานได้ที่อยู่ถัดไป ซึ่งตัวชี้แอดเดรสไม่จำเป็นต้องอยู่ที่ตำแหน่งขอบเขตล่างสุดหรือที่ปลายสุดของขอบเขตบน ตัวชี้แอดเดรสสามารถที่จะเริ่มต้นชี้ที่ตำแหน่งใดๆ ก็ได้ภายในขอบเขตของตำแน่งของ modulo ที่ได้กำหนดไว้แล้ว ทั้งค่าแอดเดรสของขอบเขตล่างและขอบเขตบนของตำแหน่งของ modulo จะไม่ถูกเก็บค่าไว้ มีแต่เพียงขนาดของ modulo เท่านั้น ที่จะถูกเก็บค่าใน Mn ส่วนค่าของแอดเดรสทั้งขอบเขตล่างและบนจะถูกกำหนดโดยค่าที่อยู่ใน Rn พิจารณาที่ (Rn)+ ซึ่งเป็นการกำหนดค่าแอดเดรสทางอ้อม(Indirct addressing mode) จะพบว่าถ้าตัวชี้ของแอดเดรสรีจีสเตอร์เพิ่มมากขึ้นผ่านขอบเขตบนของบัฟเฟอร์ (ตำแหน่งแอดเดรสที่ฐานบวก M-1) มันก็จะม้วนหมุนย้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านตำแหน่งแอดเดรสที่ฐาน(ขอบเขตล่าง) อีกทางหนึ่งจากที่กล่าวไปแล้วคือ ถ้าเราพิจารณาที่ (Rn)- ซึ่งเป็นการกำหนดค่าแอดเดรสทางอ้อม จะพบว่าถ้าตำแหน่งแอดเดรสลดลงผ่านขอบเขตล่าง(ตำแหน่งแอดเดรสที่ฐาน) มันก็จะม้วนหมุนย้อนผ่านตำแหน่งแอดเดรสที่ฐานบวกด้วย M-1 (ขอบเขตบน)

MMMM	Addressing Mode Arithmetic
0000	Reverse Carry (Bit Reverse)
0001	Modulo 2
0002	Modulo 3
:	:
7FFE	Modulo 32767
7FFF	Modulo 32768
8000	Reserved

ตารางที่ 2 Address Modifier Type

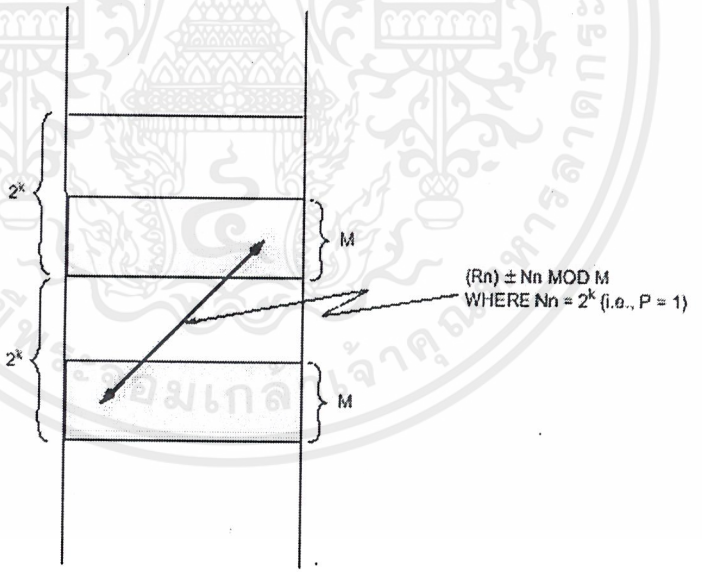


รูปที่ 5.10 Circular Buffer

ถ้าค่าออฟเซ็ต Nn ถูกใช้ในการคำนวนค่าตำแหน่งแอดเดรส ค่าสัมบรณ์ของ 16 บิตออฟเซ็ตคือ $|Nn|$ ต้องน้อยกว่าหรือเท่ากับ M สำหรับการระบุตำแหน่งแอดเดรสด้วย modulo ถ้า $Mn > M$ ผลที่ได้จะเป็นข้อมูลที่ไม่อิสระ(depentdent) และไม่สามารถคาดเดาได้ ยกเว้นแต่ในกรณีพิเศษที่ซึ่ง $Nn = P * 2^k$ คือจำนวนเท่าของขนาดของบล็อกซึ่งค่า P จะเป็นจำนวนเต็มบวกเสมอ สำหรับกรณีพิเศษนี้ เมื่อเราใช้ (Rn) + Nn เป็นวิธีในการระบุแอดเดรส(addressing mode) ตัวชี้ (Rn) จะกระโดดอย่างเป็นเส้นตรงถึง

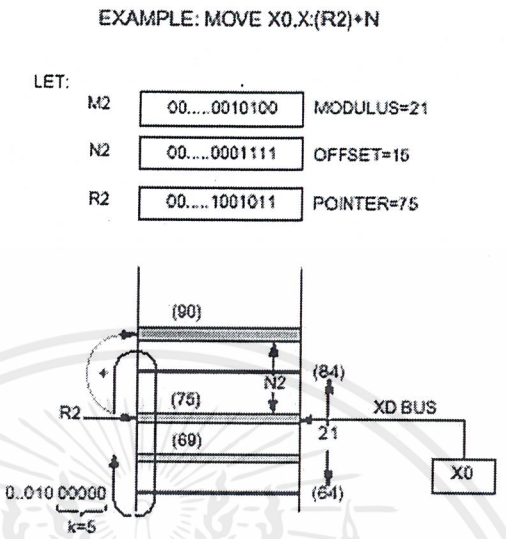
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งแอดเดรสที่มีค่าสัมพัทธ์เดียวกันในบัฟเฟอร์อันใหม่ ที่ซึ่งมี P บล็อกที่มุ่งหน้าไปในหน่วยความจำ ดูรูปที่ 5.11) ในทำนองเดียวกัน (Rn) -Nn ตัวชี้จะกระโดดจำนวน P บล็อกกลับเข้าไปในหน่วยความจำ การใช้เทคนิคในลักษณะนี้มีประโยชน์ในการคำนวนหลายๆ ตารางแบบเป็นการต่อเนื่อง หรือ เป็นอาเรย์ N มิติ (N-dimensional arrays) ขอบเขตของค่าสำหรับ Nn คือ -32,768 ถึง +32,767 ส่วนตัว modulo arithmetic unit จะม้วนหมุนตัวชี้ของตำแหน่งโดยค่าที่ต้องการอย่างอัตโนมัติ ชนิดของแอดเดรสโมดิฟิเคชั่นนี้มีประโยชน์ในการสร้าง circular buffer สำหรับ FIFO (การเข้าคิว), สายการหน่วงเวลา(delay lines) และบัฟเฟอร์ของตัวอย่างจะมีความยาวได้ถึง 32,768 สามารถนำไปใช้สำหรับการ decimation , interpolation และการสร้างรูปแบบสัญญาณขึ้นมาใหม่ ในกรณีพิเศษคือ (Rn) $\pm$ Nn mod M โดย Nn = P *2^k นี้จะมีประโยชน์คือจะใช้อัลกอริทึ่ม(algorithm) เดียวกันบนหลายๆ บล็อกของข้อมูลในหน่วยความจำ ยกตัวอย่างคือ กระบวนการกรองแบบขนานโดยใช้ infinite impulse response (IIR) filter



รูปที่ 5.11 Linear Addressing with a Modulo Modifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 Modulo Modifier Example

ตัวอย่างของ address register indirect modulo addressing นี้จะแสดงในรูปที่ 5.12 เริ่มต้นที่ตำแหน่ง 64 circular buffer ขนาด 21 ชั้น จะถูกสร้างขึ้นมา ตำแหน่งแอดเดรสที่ถูกสร้างขึ้นมาจะออฟเซ็ตไป 15 ตำแหน่ง ขอบเขตล่าง = $L * 2^k$ ซึ่ง $2^k \geq 21$ ดังนั้น k = 5 และค่าตำแหน่งแอดเดรสของขอบเขตล่างต้องเป็นจำนวนเท่าของ 32 ขอบเขตล่างอาจจะถูกเลือกเป็น 0 , 32 , 64 , 96 , 128 , 160 ,... สำหรับตัวอย่างนี้ L ถูกกำหนดให้เป็น 2 ทำให้ขอบเขตล่างเป็น 64 ขอบเขตบนของบัฟเฟอร์ก็จะเป็น 84 (ขอบเขตล่างบวก 20 (M-1)) รีจีสเตอร์ Mn จะถูกโหลดด้วยค่า 20 (M-1) เข้าไป ส่วนออฟเซ็ตรีจีสเตอร์จะถูกกำหนดให้เป็น 15 ($Nn \leq M$) ตัวชี้แอดเดรสไม่จำเป็นต้องชี้ที่แอดเดรสของขอบเขตล่างสุดและ จะสามารถเริ่มต้นชี้ที่ไหนก็ได้ภายในขอบเขตแอดเดรสของ modulo ที่กำหนดไว้ (ในตัวอย่างนี้จะอยู่ภายในขอบเขตล่าง + (2^k) ของขอบเขตแอดเดรส) ตัวชี้แอดเดรส Rn จะถูกเลือกให้มีค่าเป็น 75 ในตัวอย่างนี้ เมื่อ R2 ได้เพิ่มค่าไป 1 โดยออฟเซ็ตของคำสั่ง MOVE แทนที่จะชี้ไปยังตำแหน่ง 90 มันจะม้วนหมุนไปที่ตำแหน่ง 69 ถ้ารีจีสเตอร์ตัวชี้แอดเดรสผ่านขอบเขตบนของบัฟเฟอร์ก็จะเกิดการม้วนหมุนไปที่แอดเดรสฐาน ถ้าแอดเดรสลดลงผ่านขอบเขตล่างมันจะม้วนหมุนไปยังแอดเดรสฐานบวกกับ M-1

ถ้า Rn อยู่นอกขอบเขตของ modulo buffer และทำให้รีจีเตอร์ Rn มีการเปลี่ยนแปลงค่าใหม่เข้ามา ข้อมูลใน Rn จะถูกเปลี่ยนแปลงตามกฎของ modulo ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOVE B0,X:(R0) + N0 (เมื่อ R0 = 6 , M0 = 5 และ N0 = 0) จะทำให้ค่า R0 ไม่เปลี่ยนแปลงเพราะว่า N0 = 0 อย่างไรก็ตามเพราะ R0 อยู่เหนือขอบเขตบน ดังนั้น AGU จะคำนวนR0+N0-M0-1 สำหรับค่าใหม่ใน R0 และเซ็ต R0 = 0

คำสั่ง MOVE ในรูปที่ 5.12 จะนำข้อมูลของ X0 ไปเก็บในหน่วยความจำ X ซึ่งระบุตำแหน่งโดย(R2) แล้ว(R2) จะเปลี่ยนแปลงค่าใหม่ใน modulo ด้วยค่า 21 ค่าใหม่ของ R2 จะไม่ใช่ 90 แต่จะเป็น 69

- Reverse-carry Modifier (Mn = $0000) การกลับบิตตัวทดถูกเลือกโดยเซ็ทโมดิไฟล์รีจิสเตอร์(Modifier Register) ให้เป็นศูนย์(ดูจากตารางที่ 2) แอดเดรสโมดิฟิเคชั่นจะถูกทำในส่วนของฮาร์ดแวร์โดยการถอดแบบตัวทดในทิศทางที่ตรงกันข้ามการกลับบิตตัวทดจะเสมือนกับการรีเวอร์สบิตข้อมูลที่อยู่ใน Rn แล้วรวมกับค่าของออฟเซ็ทจากนั้นก็ทำการรีเวอร์สอีกที ถ้าการอ้างอิงแอดเดรสแบบ +Nn ถูกใช้กับแอดเดรสโมดิไฟเออร์นี้และ Nn มีค่าที่เป็น 2^{k-1}(จำนวนเท่าของสอง) วิธีแบบนี้จะเหมือนกับการกลับ k,LSB ของรีจิสเตอร์ Rn เพิ่มค่าของ Rn ไปหนึ่งแล้วทำการกลับ k,LSB ของ Rn อีกครั้ง วิธีการนี้จะเป็นประโยชน์มากกับการใช้งานในเรื่องของแฟกเตอร์ตัวบิด(twiddle factor) ของ FFT ช่วงค่าของ Nn จะแปรอยู่ในช่วง 0 ถึง +32K ซึ่งจะทำให้เกิดการกลับบิตขึ้นมากถึง 65536 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

Peripheral Section

เป็นส่วนที่ใช้สำหรับให้ผู้ใช้ชิป DSP สามารถติดต่อสื่อสารและรับส่งข้อมูลกับอุปกรณ์ภายนอกต่างๆ (I/O Device) หรือหน่วยประมวลผลอื่น (Host Interface) ได้ รวมถึงการอ้างถึงหน่วยความจำภายนอกและการควบคุมอื่นๆ ประกอบด้วย

1. External Memory Interface (EMI)
2. Serial Audio Interface (SAI)
3. Serial Host Interface (SHI)
4. General Purpose Input/Output (GPIO)

6.1 External Memory Interface (EMI)

เป็นส่วนควบคุมการอ้างถึง (Access) หน่วยความจำภายนอกทั้งแบบ Static และ Dynamic (SRAM, DRAM) ซึ่งโดยมากในงานประมวลผลสัญญาณเสียง (Digital Audio) จะใช้หน่วยความจำภายนอกในการเก็บข้อมูลที่ได้จากการสุ่ม (Sample) ซึ่งเหมาะจะใช้เป็นที่พักข้อมูลแบบหน่วงเวลา (Data-delay Buffer)

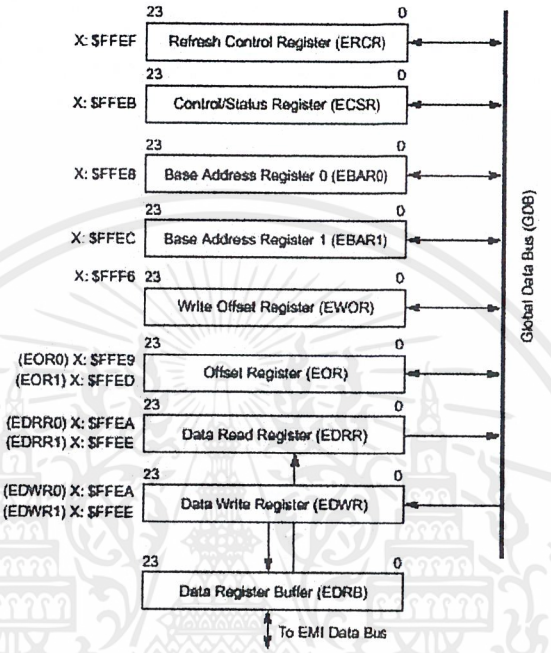
ความสามารถในการอ้างอิงหน่วยความจำแบบ SRAM ได้มากที่สุดขนาด 256x8 bits และหน่วยความจำแบบ DRAM ได้ถึง 4Mx4 bits บัสข้อมูลสามารถเลือกใช้แบบ 4 หรือ 8 บิตก็ได้ นอกจากนี้ยังสามารถเลือกขนาดของคำ (Data word length) เป็น 8, 16, 20 หรือ 24 บิตก็ได้ โดยการตัดต่อข้อมูลตามขนาดที่กำหนดจะเป็นไปอย่างอัตโนมัติ ส่วนโหมดการอ้างอิงหน่วยความจำเลือกได้ทั้งแบบ Absolute หรือ Relative ก็ได้

6.1.1 EMI Programming Model แสดงดังรูปที่ 6.1 รีจิสเตอร์แต่ละตัวสามารถเข้าถึงข้อมูลได้โดยคำสั่ง MOVEP (ยกเว้น EDRB ซึ่งเป็นบัฟเฟอร์รีจิสเตอร์) รายละเอียดของรีจิสเตอร์แต่ละตัวมีดังนี้

- **EMI Base Address Registers (EBAR0, EBAR1)** รีจิสเตอร์ขนาด 24 บิต ใช้สำหรับเก็บค่า Base address ที่จะใช้คำนวณ Address ที่จะอ้างถึงโดยการนำค่าในรีจิสเตอร์นี้ไปลบออกจากค่าในรีจิสเตอร์ EOR (กรณีอ่านค่าข้อมูล) หรือ EWOR (กรณีที่เขียนข้อมูล) ค่าใน EBARx นี้สามารถกำหนดให้เพิ่มขึ้น 1 หลังจากที่เข้าถึงหน่วยความจำแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดได้จากรีจิสเตอร์ ECSR ค่าภายในรีจิสเตอร์เก็บอยู่ในรูป (Unsigned integer)



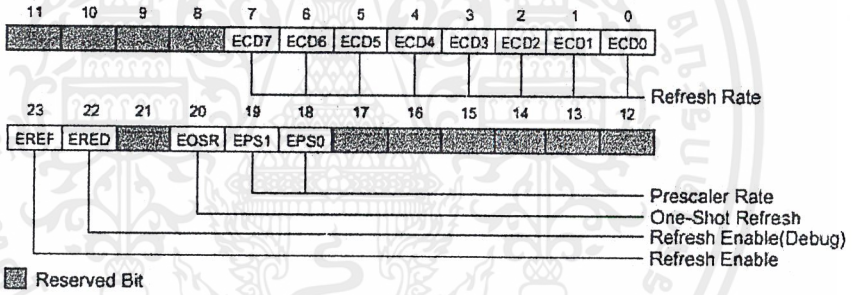
รูปที่ 6.1 EMI Programming Model

- **EMI Write Offset Register (EWOR)** รีจิสเตอร์ขนาด 24 บิต ใช้ในการคำนวณ address ที่จะทำการเขียนข้อมูลลงไป ร่วมกับค่าใน EBARx
- **EMI Offset Register (EOR)** รีจิสเตอร์ขนาด 24 บิต ใช้ในการคำนวณ address ที่จะทำการอ่านข้อมูลเข้าสู่ตัว DSP ร่วมกับค่าใน EBARx โดยทั่วไปในงานด้าน Digital audio จะใช้ในการอ้างถึงข้อมูลที่มีการหน่วงเวลา (Data-delay buffer)
- **EMI Data Write Register (EDWR0, EDWR1)** รีจิสเตอร์ที่เขียนได้เท่านั้นขนาด 24 บิต ใช้เก็บข้อมูลที่ต้องการจะเขียนลงไปที่หน่วยความจำภายนอก อยู่ที่ตำแหน่ง $FFEA(EDWR0) และ $FFEE (EDWR1) ตามลำดับ
- **EMI Data Read Register (EDRR)** รีจิสเตอร์ที่อ่านได้เท่านั้นขนาด 24 บิต ใช้เก็บข้อมูลที่ต้องการจะอ่านจากหน่วยความจำภายนอก อยู่ที่ตำแหน่ง $FFEA(EDRR0) และ $FFEE (EDRR1) ตามลำดับ
- **EMI Data Register Buffer (EDRB)** รีจิสเตอร์ขนาด 24 บิต ทำหน้าที่ Pack หรือ Unpack ข้อมูลที่ได้เขียนหรืออ่านจากหน่วยความจำภายนอก เนื่องจากบัสข้อมูลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

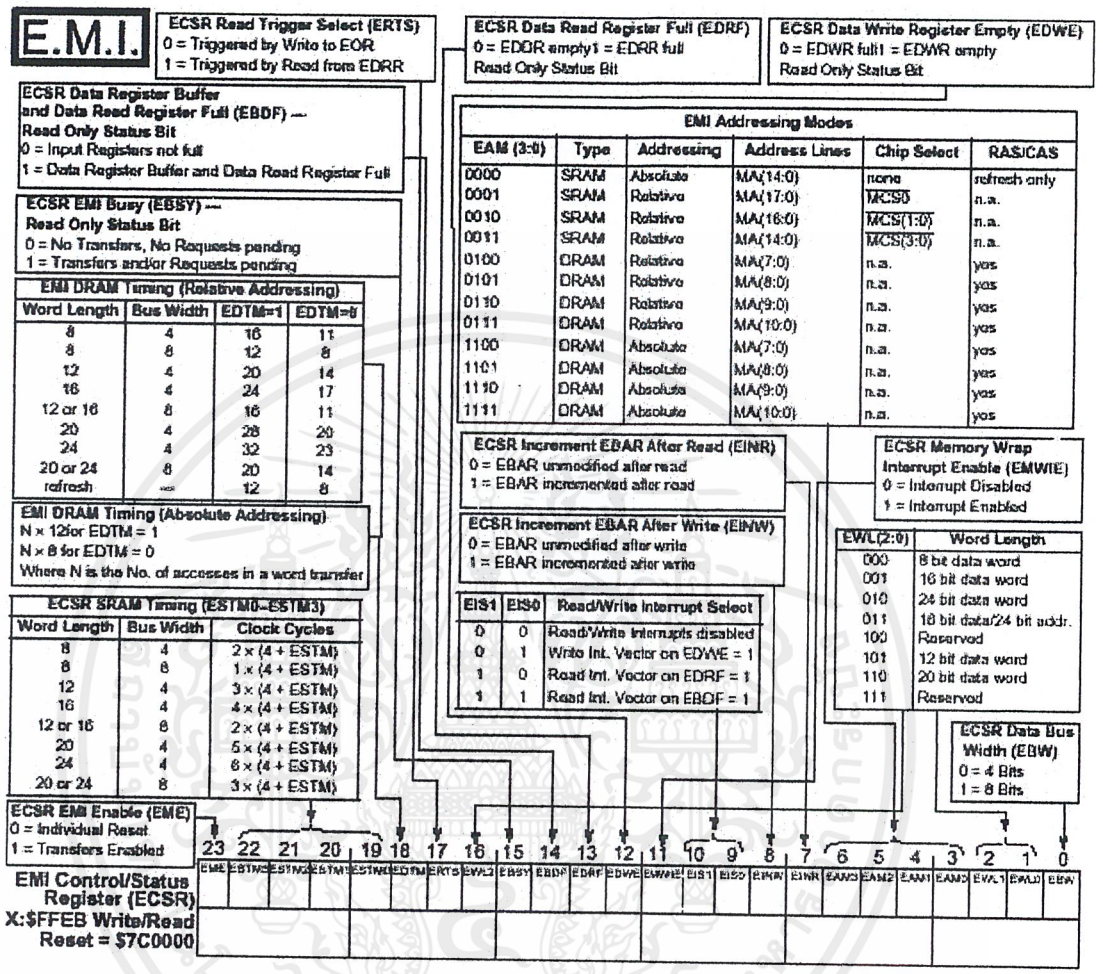
EMI มีขนาดได้เป็น 4 หรือ 8 บิตเท่านั้นส่วนขนาดของคำ (Word) อาจเป็นได้ทั้ง 8, 12, 16, 20 หรือ 24 บิต ดังนั้นข้อมูลจะต้องถูก Unpack ที่รีจิสเตอร์ตัวนี้ก่อน จึงค่อยส่งผ่านตามบัสข้อมูลไปยังหน่วยความจำภายนอก โดยเมื่อทำการเขียนข้อมูลที่มีความยาวของคำน้อยกว่า 24 บิต บิตบน (Most significant bit) เท่านั้นที่จะถูกส่งออกไป ทำนองเดียวกันกับการอ่านข้อมูล จะต้องมีการ Pack ก่อนจึงค่อยส่งให้รีจิสเตอร์ EDRR ต่อไป และมีการเติมศูนย์ลงไปที่ท้ายคำในกรณีข้อมูลที่อ่านได้มีความยาวของคำน้อยกว่า 24 บิต

- **EMI Refresh Control Register (ERCR)** รีจิสเตอร์ขนาด 24 บิตใช้ในการควบคุมการ Refresh หน่วยความจำชนิด DRAM แสดงรายละเอียดดังรูปที่ 6.2
- **EMI Control/Status Register (ECSR)** รีจิสเตอร์ขนาด 24 บิตใช้ในการควบคุมการทำงานของ EMI รายละเอียดของแต่ละบิตแสดงดังรูปที่ 6.3



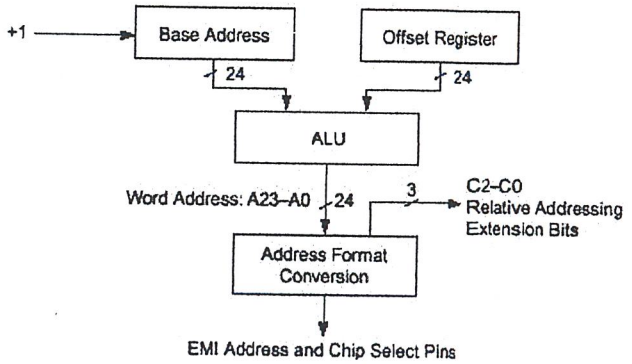
รูปที่ 6.2 EMI Refresh Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 EMI Control/Status Register

6.1.2 EMI Address Generation การคำนวณ Address จะกระทำโดย Address Generation Unit (AGU) ดังแสดงในรูปที่ 6.4



รูปที่ 6.4 EMI Address Generation Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AGU จะทำการคำนวณตำแหน่งของหน่วยความจำสำหรับการอ่านข้อมูลโดยการนำค่าในรีจิสเตอร์ EBARx ไปลบออกจากค่าในรีจิสเตอร์ EOR และทำนองเดียวกันกับการเขียนข้อมูลลงไปบนหน่วยความจำ ตำแหน่งของหน่วยความจำจะคำนวณได้จากการนำค่าในรีจิสเตอร์ EBARx ไปลบออกจากค่าในรีจิสเตอร์ EWOR

เราสามารถแบ่งการเข้าถึงตำแหน่งของข้อมูลได้ 4 แบบดังนี้

1) **SRAM Absolute Addressing** การเข้าถึงหน่วยความจำแบบนี้เหมาะสำหรับการอ่านข้อมูลแบบโปรแกรม(การอ่านโปรแกรมที่อยู่ภายนอก DSP)และสามารถใช้ร่วมกับการเข้าถึงแบบ Relative ได้ด้วย
2) **SRAM Relative Addressing** ใช้ในการสร้างที่พักข้อมูลบนหน่วยความจำแบบ SRAM(Data-delay buffer) โดยสามารถอ้างถึงหน่วยความจำได้มากที่สุด 256k ตำแหน่ง ขึ้นอยู่กับค่าในรีจิสเตอร์ควบคุม
3) **DRAM Absolute Addressing** การเข้าถึงข้อมูลจะช้ากว่าหน่วยความจำแบบ SRAM โดยอ้างอิงขนาดของหน่วยความจำได้มากที่สุดขนาด 4M
4) **DRAM Relative Addressing** ใช้ในการสร้างที่พักข้อมูลบนหน่วยความจำแบบ DRAM(Data-delay buffer)

6.1.3 DRAM Refresh หน่วยความจำแบบ DRAM ต้องการการ Refresh เพื่อให้ข้อมูลยังคงอยู่ในหน่วยความจำ ซึ่งวิธีการ Refresh หน่วยความจำนั้นมีด้วยกัน 2 แบบคือ

1) **Access Memory** การเข้าถึงหน่วยความจำทุกๆตำแหน่งด้วยเวลาที่เหมาะสม จะเสมือนกับการ Refresh หน่วยความจำ(จะต้องมีการเขียนโปรแกรมในการ Refresh หน่วยความจำเพิ่มเติม)
2) **Internal Refresh Timer** อีกวิธีหนึ่งคือการใช้ Internal Refresh Timer ที่มีอยู่ในตัว DSP โดยสามารถควบคุมการตั้งเวลา Refresh ได้ผ่านทางรีจิสเตอร์ควบคุม

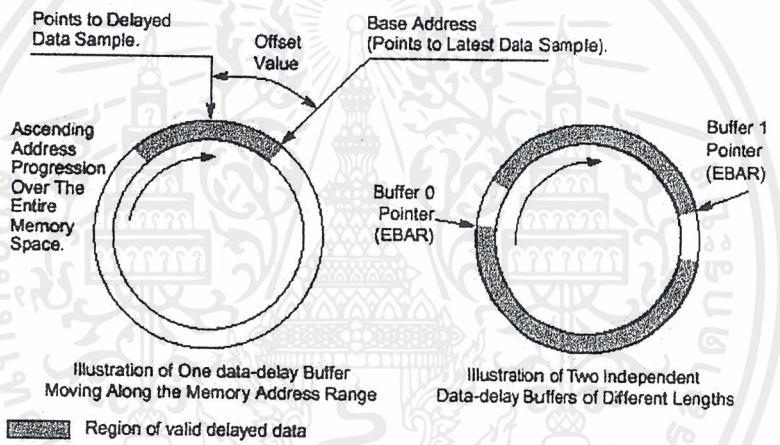
6.1.4 Read Data Transfer การอ่านค่าข้อมูลจากหน่วยความจำจะต้องมีการกระตุ้น(Trigger)เพื่อให้ DSP เริ่มต้นทำการอ่านข้อมูลเข้า โดยสามารถกระตุ้นได้ 2 รูปแบบคือ

1) กระตุ้นโดยการเขียนค่าลงในรีจิสเตอร์ EOR (ERTS=0)
2) กระตุ้นโดยการอ่านค่าจากรีจิสเตอร์ EDRR (ERTS=1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.5 Write Data Transfer การเขียนข้อมูลลงไปที่หน่วยความจำภายนอก จำเป็นจะต้องทำการกระตุ้นการเขียนเช่นเดียวกันกับการอ่าน โดยใช้วิธีการเขียนข้อมูลที่ต้องการจะนำไปเก็บในหน่วยความจำลงไปในรีจิสเตอร์ EDWR ได้เพียงวิธีเดียวเท่านั้น

6.1.6 Data-Delay Structure เป็นลักษณะการเก็บข้อมูลเป็นลำดับ (Sequential) ตามหน่วยเวลา โดยที่ตำแหน่งของหน่วยความจำจะสัมพันธ์กับเวลาที่หน่วง (Delay) ไป ซึ่งโครงสร้างของส่วน EMI มีรีจิสเตอร์รองรับการเก็บข้อมูลแบบนี้อยู่แล้วคือ EBARx และ EOR,EWOR (ในหัวข้อ 6.1.1) Data-Delay Structure แสดงโครงสร้างได้ดังรูปที่ 6.5



รูปที่ 6.5 Data-Delay Structure

6.2 Serial Audio Interface (SAI)

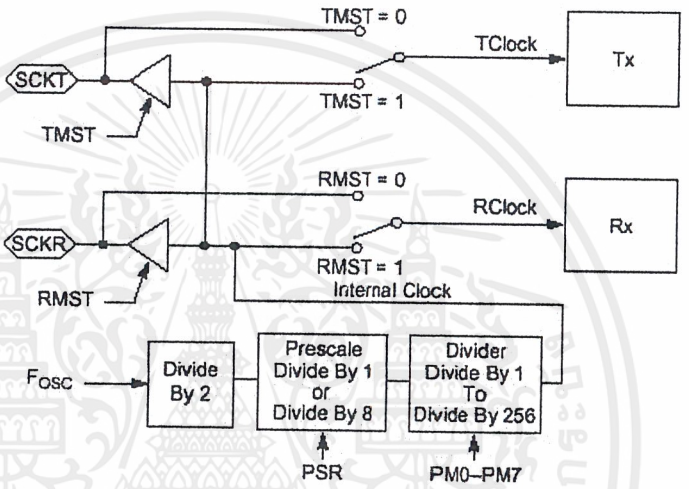
DSP สามารถติดต่อสื่อสารกับแหล่งข้อมูล (เช่น Audio Sample)ที่จะนำมาประมวลผลผ่านทาง Serial Audio Interface โดยเป็นพอร์ทแบบอนุกรมที่ทำงานแบบ Full-duplex ซึ่งสามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้หลากหลายรูปแบบ เช่น Analog-to-Digital (A/D) converter, Digital-to-Analog (D/A) converter เป็นต้น

ส่วนประกอบของ SAI ประกอบด้วยส่วนรับ(Receive)และส่ง(Transmit)แยกออกจากกัน ทั้งสองส่วนทำงานได้ทั้งโหมด Master และ Slave มีส่วนสร้าง Baud-rate ที่ผู้ใช้สามารถกำหนดความเร็วในการสื่อสารได้เอง มีรีจิสเตอร์รับข้อมูล 2 ชุดที่ทำงานได้พร้อมๆกัน(Fully synchronize)ภายใต้ชุดควบคุมเดียวกัน เช่นเดียวกันกับรีจิสเตอร์ชุดส่งที่มีอยู่ 3 ชุดทำงานได้พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

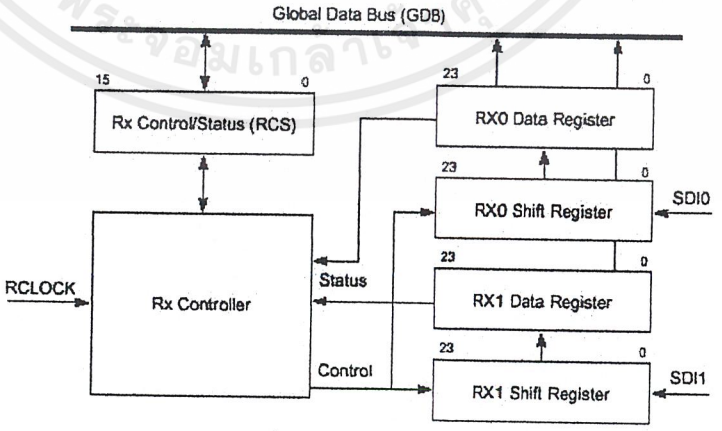
6.2.1 Serial Audio Interface Internal Architecture ประกอบด้วย 3 ส่วนย่อยๆได้แก่

1) Baud-Rate Generator สำหรับสร้าง clock ภายในเพื่อใช้ในระบบ SAI ทั้งชุดรับและส่ง แสดงได้ดังรูปที่ 6.6



รูปที่ 6.6 SAI Baud-Rate Generator Block Diagram

2) Receive Section ประกอบด้วยรีจิสเตอร์ควบคุมขนาด 16 บิต, รีจิสเตอร์เลื่อนขนาด 24 บิต 2 ชุด และรีจิสเตอร์ข้อมูลขนาด 24 บิต 2 ชุด(RX0 และ RX1) ดังรูปที่ 6.7



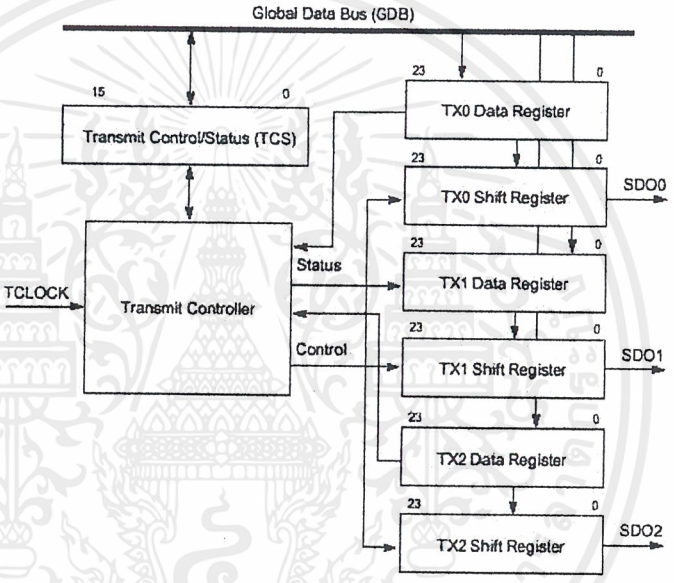
รูปที่ 6.7 SAI Receive Section Block Diagram

จากรูป ข้อมูลจะถูกเลื่อนเข้ามาที่รีจิสเตอร์เลื่อนจนกระทั่งครบตามจำนวนบิตที่ถูกกำหนดไว้ในรีจิสเตอร์ควบคุม ข้อมูลเหล่านั้นก็จะถูกเก็บเข้าสู่รีจิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์ข้อมูลเพื่อรอการอ่านจากหน่วยประมวลผลต่อไป โดยสามารถกำหนดขนาดข้อมูลได้เป็น 16, 24 หรือ 32 บิต โดยแบบ 32 บิตจะมีวิธีการควบคุมที่พิเศษกว่าแบบอื่น

3) SAI Transmit Section ประกอบด้วยรีจิสเตอร์ควบคุมขนาด 16 บิต, รีจิสเตอร์เลื่อนขนาด 24 บิต 3 ชุด และรีจิสเตอร์ข้อมูลขนาด 24 บิต 3 ชุด(TX0, TX1 และ TX2) ดังรูปที่ 6.8



รูปที่ 6.8 SAI Transmit Section Block Diagram

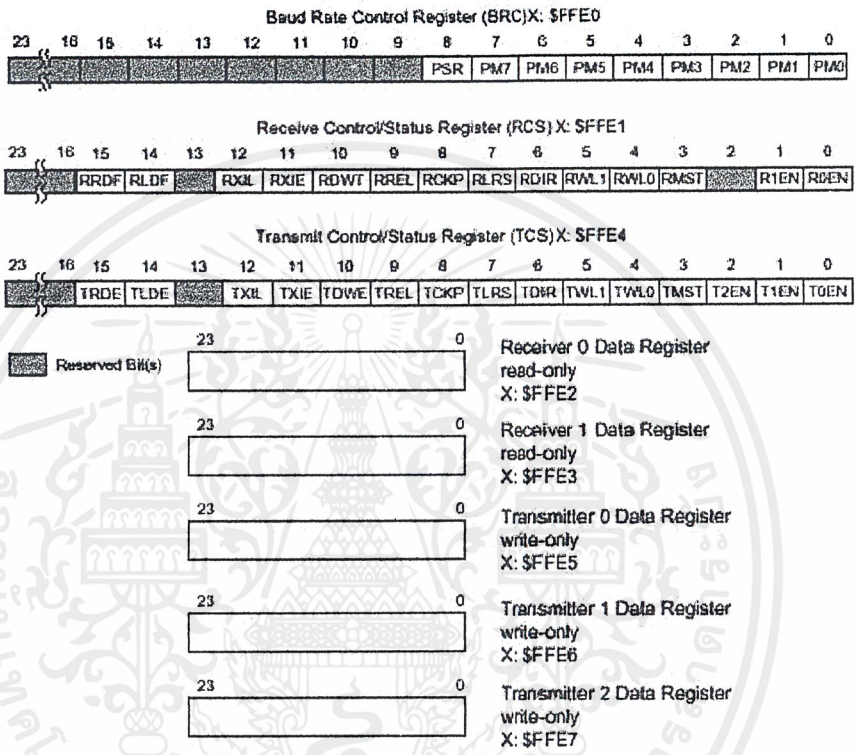
รีจิสเตอร์เลื่อนทั้ง 3 จะทำงานไปพร้อมๆกันในการส่งข้อมูลออกไปสู่ภายนอก ถ้ามีการอนุญาติให้ใช้งาน ซึ่งควบคุมจาก Transmit Control/Status Register(TCR) ซึ่งขนาดของข้อมูลที่จะส่งออกไปนั้น เป็นได้ทั้ง 16, 24 และ 32 บิต

6.2.2 Serial Audio Interface Programming Model รายละเอียดแสดงได้ดังรูปที่ 6.9 ประกอบด้วยรีจิสเตอร์ต่างๆดังนี้

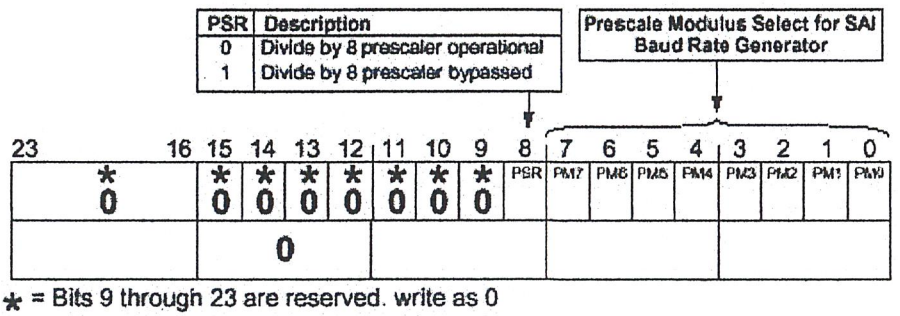
1) Baud Rate Control Register (BRC) เราสามารถกำหนดความถี่ของ clock ได้ผ่านทาง Baud Rate Control Register โดยความถี่ที่สามารถสร้างได้จะมีค่าไม่เกินหนึ่งในสี่ของความถี่ออสซิลเลเตอร์ (Fosc) และสามารถรับความถี่จากสัญญาณภายนอกได้มากที่สุดหนึ่งในสามของ ความถี่ออสซิลเลเตอร์ แสดงดังรูปที่ 6.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Receiver Control/Status Register (RCS) รีจิสเตอร์ควบคุมขนาด 16 บิต โดยจะทำหน้าที่ควบคุมการทำงานต่างๆของชุดรับข้อมูล รายละเอียดของแต่ละบิตแสดงในรูปที่ 6.11

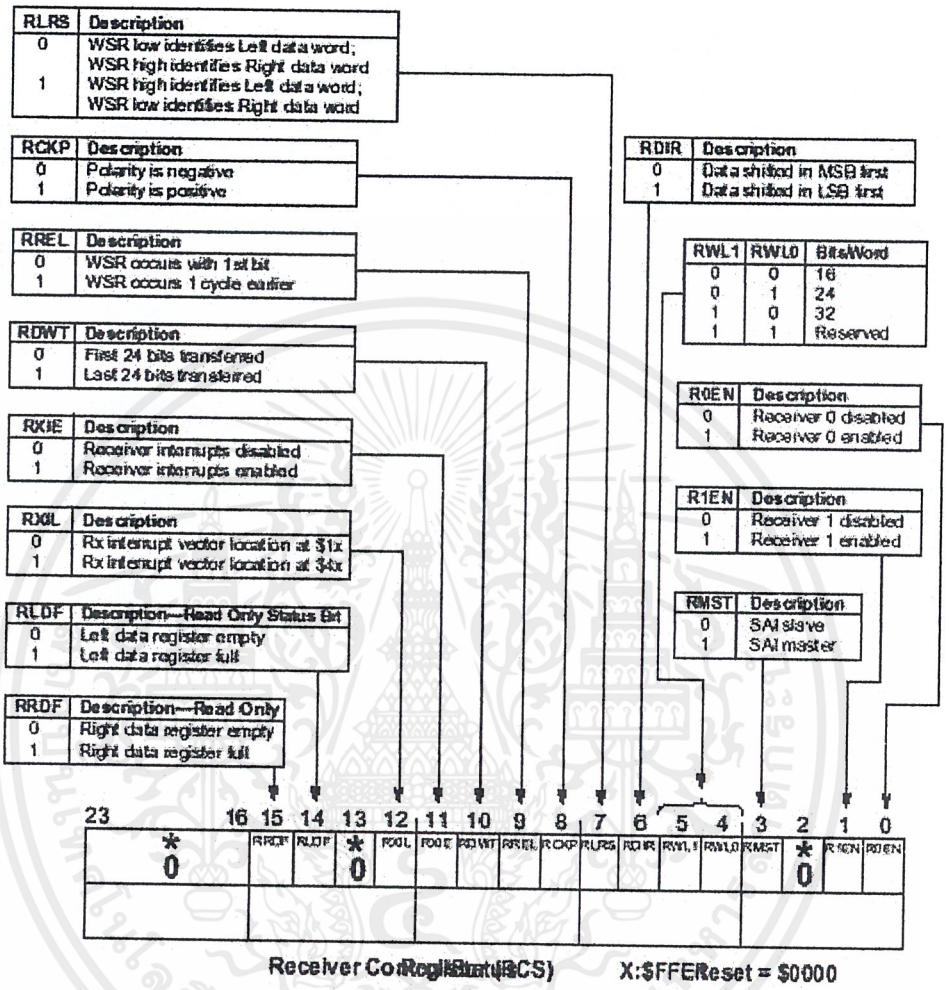


รูปที่ 6.9 SAI Programming Model



รูปที่ 6.10 Baud Rate Control Register (BRC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



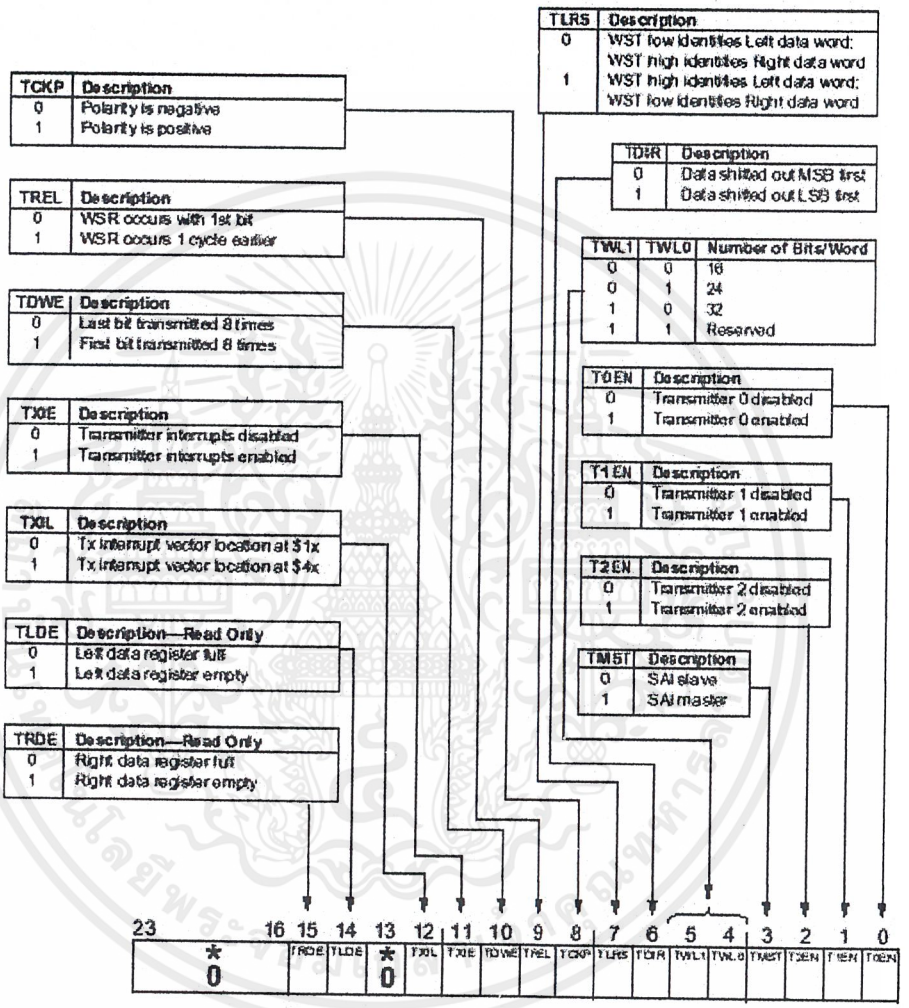
รูปที่ 6.11 Receiver Control/Status Register

3) Transmit Control/Status Register (TCS) รีจิสเตอร์ควบคุมขนาด 16 บิต โดยจะทำหน้าที่ควบคุมการทำงานต่างๆของชุดส่งข้อมูล รายละเอียดของแต่ละบิตแสดงในรูปที่ 6.12

4) SAI Receiver Data Register (RX0, RX1) รีจิสเตอร์ข้อมูลขนาด 24 บิต ใช้เก็บข้อมูลที่รับมาจากภายนอก ทำงานพร้อมกันทั้ง 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) SAI Transmit Data Register (TX0, TX1, TX2) รีจิสเตอร์ข้อมูลขนาด 24 บิต ใช้เก็บข้อมูลที่จะส่งออกไปสู่ภายนอก ทำงานพร้อมกันทั้ง 3 ตัว



รูปที่ 6.12 Transmit Control/Status Register

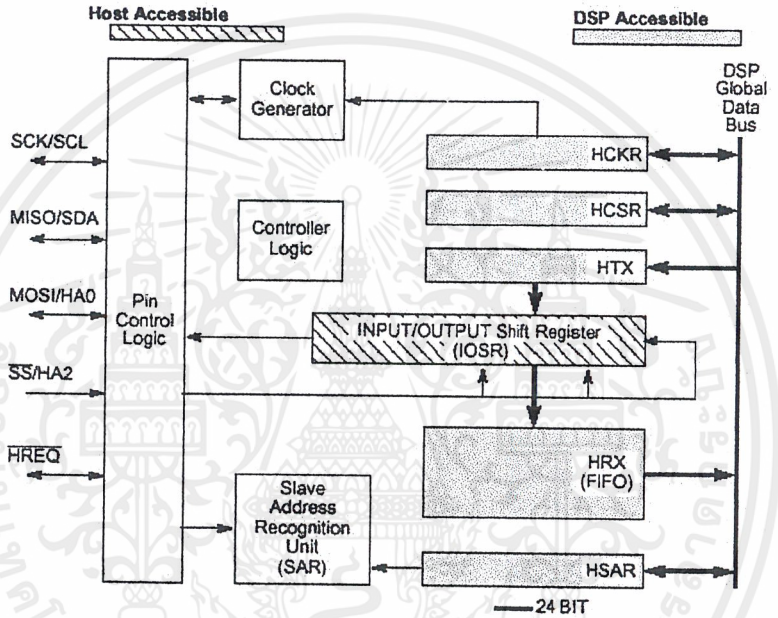
6.3 Serial Host Interface (SHI)

Serial Host Interface เป็นส่วนที่ใช้ติดต่อกับ Host processor ภายนอก โดยมีการอินเตอร์เฟสผ่านรูปแบบของบัสอนุกรมที่เป็นที่รู้จักกันอย่างกว้างขวาง (Synchronous Serial Bus) ได้แก่ Motorola Serial Peripheral Interface (SPI) และ Philips Inter-Integrated-circuit Control (I^2C) โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำงานได้ทั้งโหมด Master และ Slave รองรับความยาวของข้อมูลได้3แบบคือ 1, 2 และ 3 ไบต์ นอกจากนี้ยังมี Buffer พักข้อมูลที่รับมา 10 ระดับแบบ FIFO (First-In-First-Out)

6.3.1 Serial Host Interface Internal Architecture DSP จะมองเห็น SHI เป็นเสมือน Peripheral register ที่อยู่ในหน่วยความจำภายใน (X-Memory) ซึ่งสามารถแก้ไขและเคลื่อนย้ายข้อมูลได้โดยใช้คำสั่งพื้นฐานทั่วไป (MOVEP) Block diagram ของส่วนนี้แสดงในรูปที่ 6.13



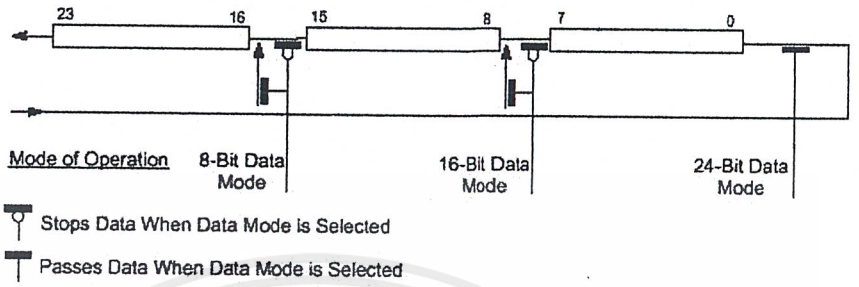
รูปที่ 6.13 Serial Host Interface Block Diagram

6.3.2 Serial Host Interface Programming Model

รายละเอียดของรีจิสเตอร์ต่างๆมีดังนี้

1) SHI Input/Output Shift Register (IOSR) เป็นรีจิสเตอร์รับ-ส่งข้อมูลระหว่าง Host ภายนอกกับ SHI โดยเสมือนเป็นตัวแปลงข้อมูลจากแบบขนาน-อนุกรม และอนุกรม-ขนานที่สามารถเลือกขนาดความยาวของข้อมูลได้เป็น 8, 16 หรือ 24 บิตดังแสดงในรูปที่ 6.14 โดยเมื่อเลือกใช้แบบ 8 บิต Most Significant Byte จะถูกใช้เป็น Shift register ทำนองเดียวกันกับแบบ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.14 SHI I/O Shift Register

2) SHI Host Transmit Data Register (HTX) ใช้ในการเคลื่อนย้ายข้อมูลจาก DSP ไปสู่ Host เป็นรีจิสเตอร์ขนาด 24 บิต ในการเคลื่อนย้ายข้อมูลไบต์เดี่ยว (8 บิต) ข้อมูลที่อยู่ใน Most Significant Byte ของ HTX เท่านั้นที่จะถูกส่งออกไปสู่ Host ทำนองเดียวกันกับข้อมูลขนาด 16 บิต

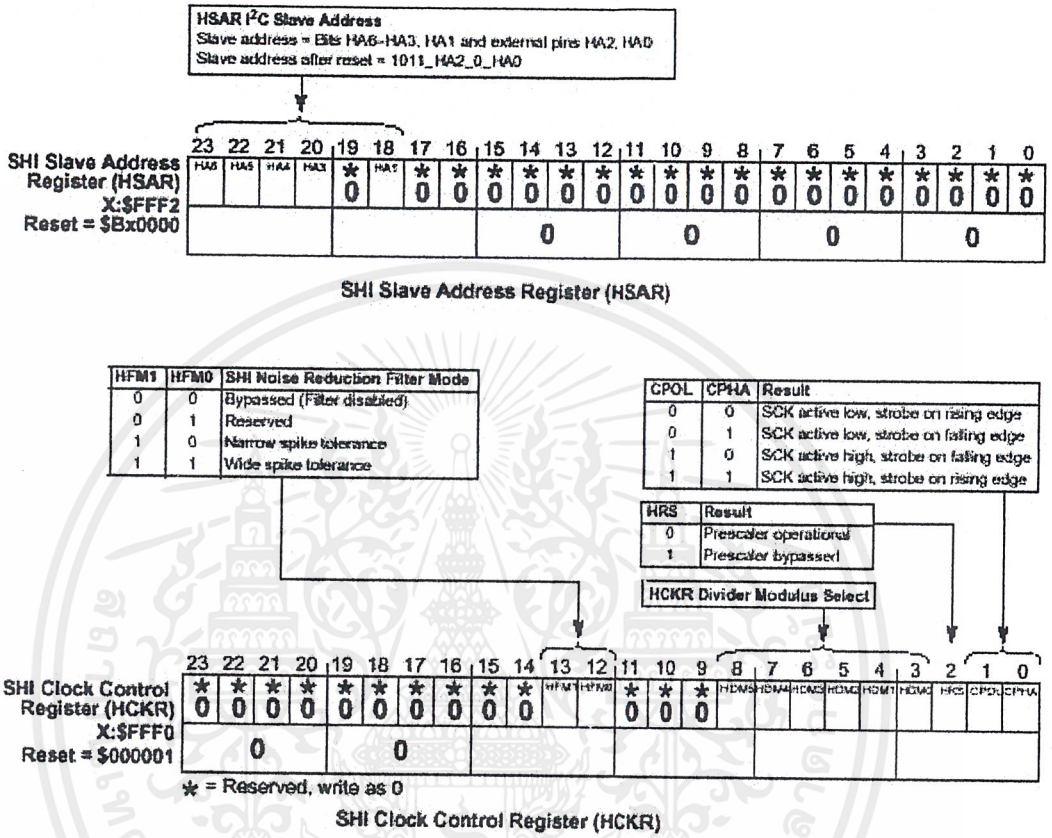
3) SHI Host Receive Data FIFO (HRX) เป็นรีจิสเตอร์แบบ First-In-First-Out ที่มีความลึก 10 ลำดับชั้น แต่ละชั้นมีขนาด 24 บิต ใช้สำหรับพักข้อมูลที่รับมาจาก Host ภายนอก โดยการรับข้อมูลที่มีความยาวน้อยกว่า 24 บิต Most Significant Byte ของ HRX ที่จะถูกใช้เก็บข้อมูลนั้น (ส่วนที่เหลือจะเติมศูนย์ให้อัตโนมัติ)

4) SHI Slave Address Register (HSAR) จะถูกใช้เมื่อ SHI ถูกใช้งานใน โหมด I^2C Slave โดยจะทำหน้าที่เก็บค่า Address ที่ใช้ร่วมกันกับโหมดนี้

5) SHI Clock Control Register (HCKR) เป็นรีจิสเตอร์ควบคุมการสร้าง Clock รวมทั้ง Phase และ Polarity ที่ใช้ในการติดต่อกับ Host โดยสามารถสร้างความถี่ได้สูงสุดหนึ่งในสี่ของความถี่ออสซิลเลเตอร์ (F_{osc})สำหรับโหมด SPI และหนึ่งในหกในกรณีโหมด I^2C (สามารถรับความถี่จากภายนอกได้มากที่สุดเป็นหนึ่งในสามและหนึ่งในห้าเมื่อใช้ในโหมด SPI และ I^2C ตามลำดับ)

6) SHI Control/Status Register (HCSR) รีจิสเตอร์ใช้ควบคุมการทำงานของ SHI เช่น โหมดการทำงาน เป็นต้น รายละเอียดแสดงในรูปที่ 6.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



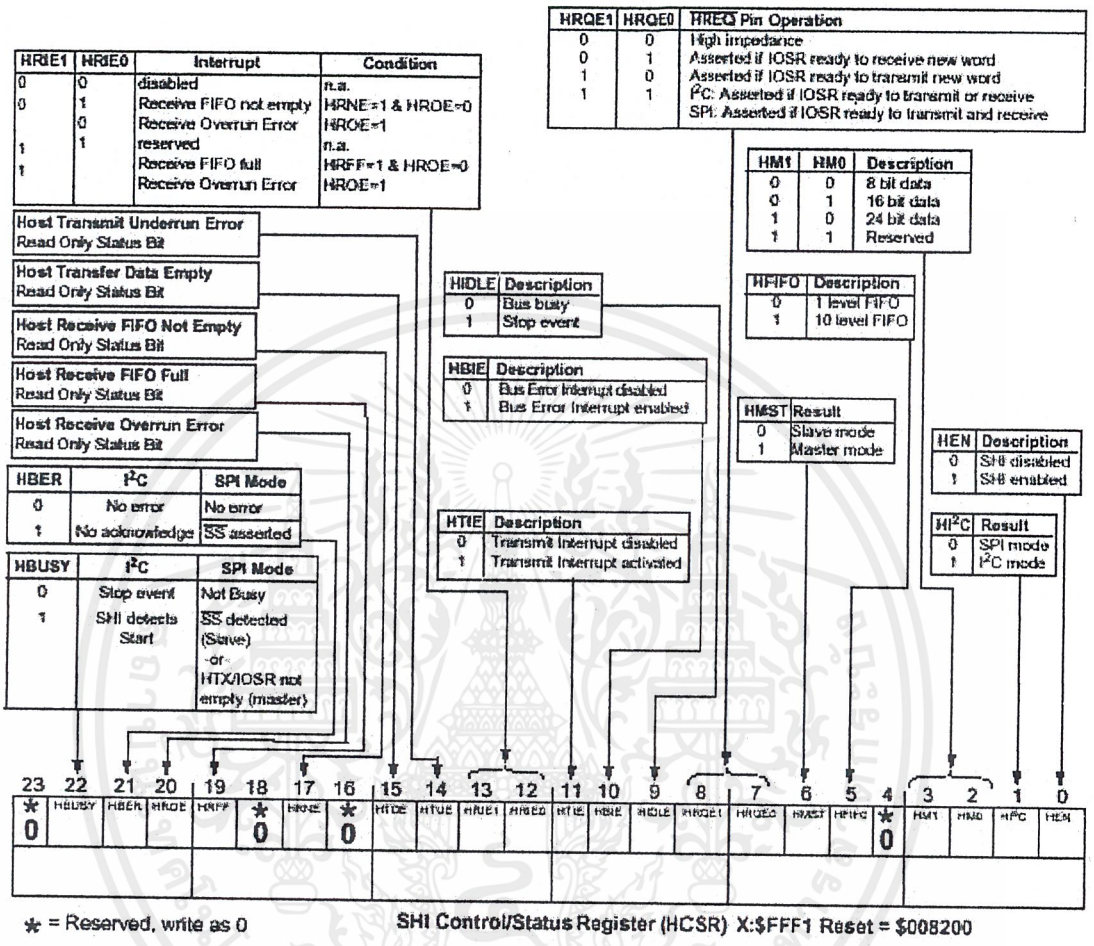
รูปที่ 6.15 SHI Slave Address Register And SHI Clock Control Register

6.3.3 Characteristics Of The SPI Bus บัสข้อมูลแบบ SPI ประกอบด้วยทางเดินข้อมูล 2 เส้น (ได้แก่ MISO และ MOSI), สัญญาณนาฬิกา (SCK) และ สัญญาณเลือก Slave Select(SS) สำหรับการรับและส่งข้อมูลสามารถทำไปได้พร้อมๆกันทั้ง 2 ทางโดยขาใดจะเป็น input หรือoutput นั้นก็ขึ้นอยู่กับการกำหนดโหมดการทำงานจาก Control Register

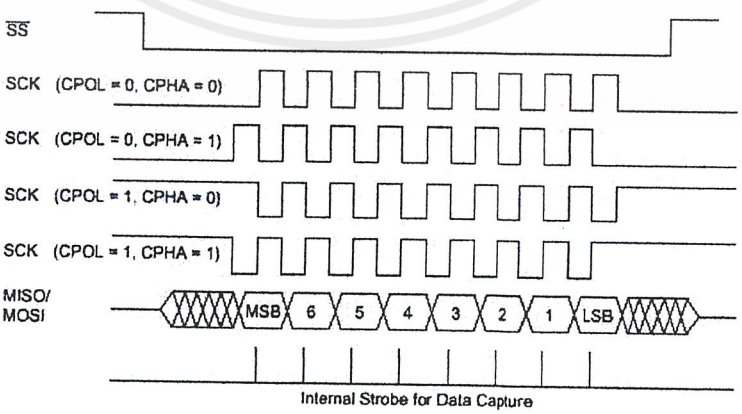
ส่วนของ Clock นั้น สามารถเลือก Polarity ได้ 2 แบบเพื่อให้สอดคล้องกับอุปกรณ์อื่นที่นำมาต่อใช้งานร่วมกัน โดยรูปแบบของสัญญาณต่างๆแสดงดังรูปที่ 6.17

สัญญาณ SS เป็น input ที่ให้อุปกรณ์อื่นส่งสัญญาณมาเลือกตัวมันให้ทำงานหรือไม่ก็ได้ (กรณีทำงานแบบ Slave) โดยถ้าไม่มีสัญญาณมาเลือก ขา output จะอยู่ในสภาวะ high-impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.16 SHI Control/Status Register



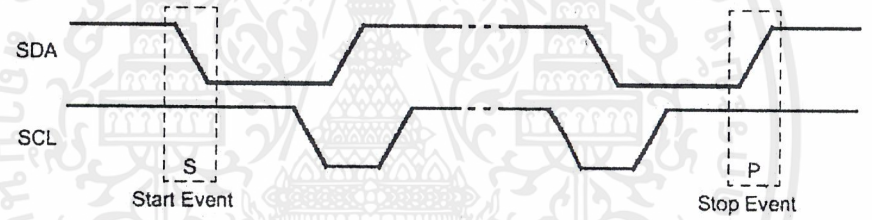
รูปที่ 6.17 Characteristics of SPI bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

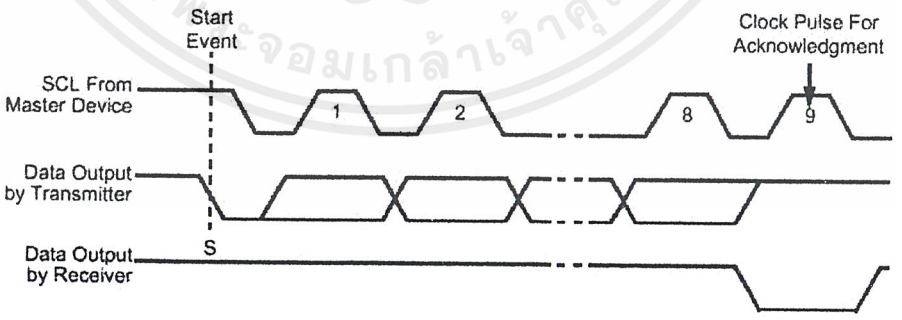
6.3.4 Characteristics Of The I²C Bus บัสข้อมูลแบบ I²C ประกอบด้วยขาข้อมูล 2 ทิศทาง (SDA) และสัญญาณนาฬิกา (SCL) โดยทั้ง 2 ขาต้องมีการต่อกับแหล่งจ่ายบวกผ่าน pull-up resistor อยู่ตลอดเวลาขณะใช้งาน

I²C Protocal ประกอบด้วยเหตุการณ์ต่างๆเพื่อใช้ในการรับส่งข้อมูลดังต่อไปนี้

- Bus not busy บัสข้อมูลและสัญญาณ clock มี logic high
- Start data transfer การเปลี่ยนสถานะของบัสข้อมูลจาก high ไปเป็น low ขณะที่สัญญาณ clock มีสถานะ high ดังรูปที่ 6.18
- Stop data transfer การเปลี่ยนสถานะของบัสข้อมูลจาก low ไปเป็น high ขณะที่สัญญาณ clock มีสถานะ high ดังรูปที่ 6.18



รูปที่ 6.18 I²C Bus Characteristics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

Software Algorithm

Polyphase Filterbank Algorithm

อธิบายได้เป็นขั้นตอนดังนี้

1) โปรแกรมจะทำการตรวจสอบว่าเป็นการผ่านข้อมูลเข้าสู่ Polyphase Filterbank ครั้งแรกหรือไม่(1st pass flag) ถ้าเป็นการผ่านครั้งแรก โปรแกรมจะทำการเคลียร์ค่าในเมโมรี่ที่เป็น Analysis Window ให้เป็นศูนย์ทั้งหมด แต่ถ้าไม่ใช่การผ่านครั้งแรกก็จะเข้าสู่ขั้นตอนถัดไป
2) ทำการ copy ข้อมูลของสัญญาณทดสอบจำนวน 32 ค่าที่อยู่ในเมโมรี่มาใส่ลงไป ณ.ตำแหน่งที่ 480 ถึง 511 (Newest input samples)
3) เริ่มเข้าสู่กระบวนการ Filter โดยทำการคำนวณ vector z จากข้อมูลใน Analysis Window ดังสมการ

$$Z(i)=C(i)*X(511-i)\ ;\ I=0,1,2,\ldots,511$$

โดย X(i)=input sample ใน Analysis Window
C(i)=Filter Coefficient
Z(i)=Vector Z

จะได้ผลลัพท์จำนวน 512 ค่า ใน vector z buffer

4) นำค่าใน vector z buffer ไปคำนวณ Partial Calculation ดังสมการ

$$Y(i) = \sum_{j=0}^{7} [\ Z(i+64j)\ ]\ ;\ i=0,1,2,\ldots,63$$

โดย Y(i)=Partial value
Z(i)=Vector Z

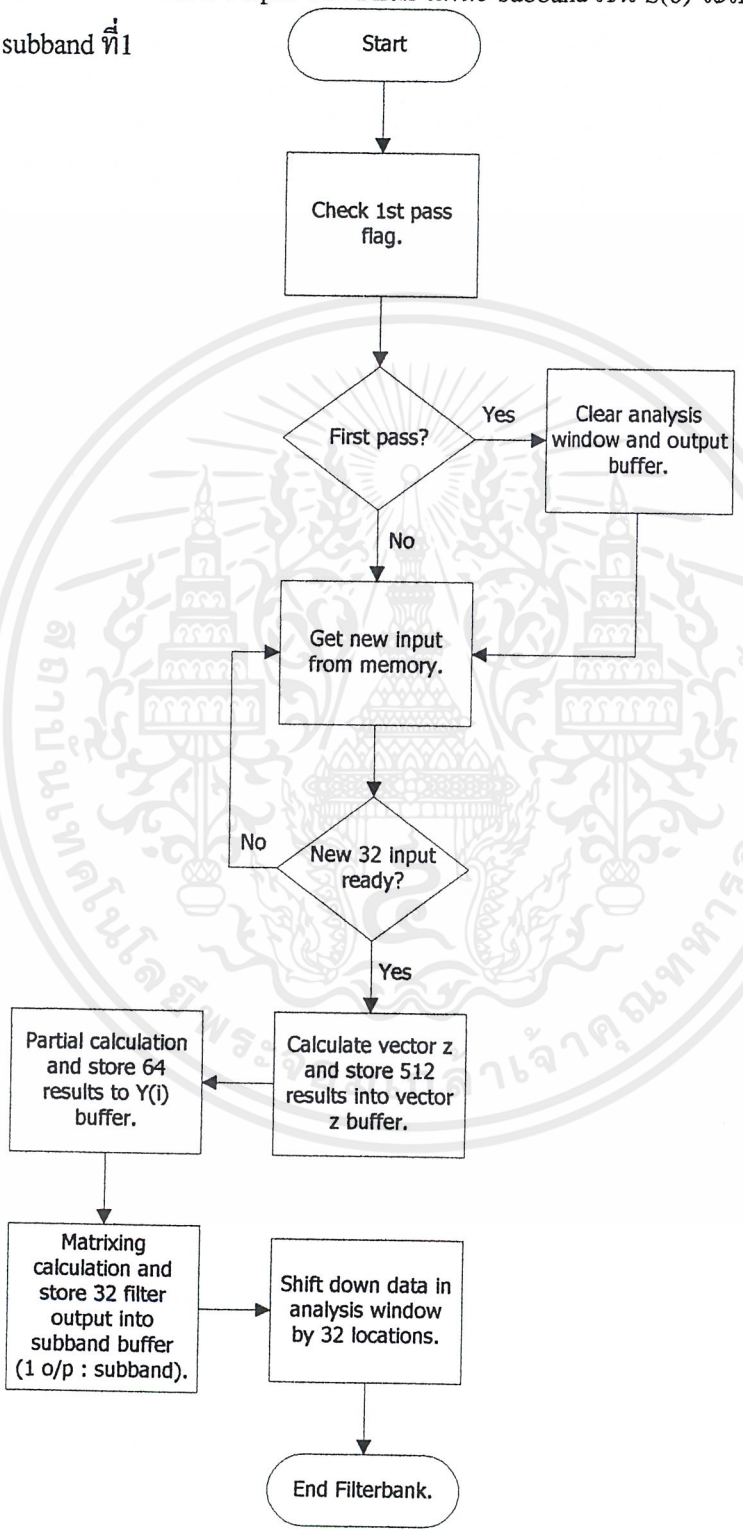
5) คำนวณ Matrixing ดังสมการ

$$S(i) = \sum_{k=0}^{63} [\ M(ik)*Y(k)\ ]\ ;\ i=0,1,2,\ldots,31$$

โดย $M(ik)=\cos[(2i+1)*(k+16)*\pi/64]$
Y(k)=Partial value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพท์ที่ได้จะเป็น output ของ Filter แต่ละ subband เช่น S(0) จะเป็น output ของ subband ที่1



รูปที่ 7.1 Flow Diagram ของโปรแกรม Polyphase Filterbank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FFT Algorithm

อธิบายขั้นตอนต่างๆได้ดังนี้

1) โปรแกรมจะทำการตรวจสอบว่าเป็นการผ่านข้อมูลเข้าสู่ FFT ครั้งแรกหรือไม่(1^{st} pass flag) ถ้าเป็นการผ่านครั้งแรก โปรแกรมจะทำการใส่ศูนย์ให้กับ 64 ตำแหน่งแรกของ FFT Analysis Buffer (Zero padding) แต่ถ้าไม่ใช่การผ่านครั้งแรกโปรแกรมจะดึงเอาข้อมูล 64 ตำแหน่งล่าสุด (Newest 64 samples) จาก Frame ก่อนหน้ามาใส่ที่ 64 ตำแหน่งแรกของ FFT Analysis Buffer
2) ทำการ copy ข้อมูลของสัญญาณทดสอบจำนวน 448 ค่าที่อยู่ในเมโมรี่มาใส่ลงไป ณ.ตำแหน่งที่เหลือจนครบเต็ม Frame ข้อมูล (Newest input samples)
3) นำ Frame ข้อมูลที่ได้ไปทำการ Windowing ด้วย Hann's Window โดยใช้ค่าจากตารางในเมโมรี่ที่ได้กำหนดไว้ก่อนแล้ว
4) ผ่านข้อมูลที่ได้เข้าสู่ FFT Calculation ขนาด 512 จุด จากนั้นทำการ Bit reversal เพื่อเรียงผลลัพท์ที่ได้ให้ถูกต้องตามลำดับ
5) นำผลลัพท์ 256 ค่าแรกที่ได้จาก FFT Calculation มาคำนวณ Power Spectrum จากสมการ

$$power\_spectrum(i)=x_i^2+y_i^2$$; เมื่อ x_i=real part of FFT

y_i=imaginary part of FFT

จากนั้นจึงแบ่ง Power Spectrum ออกเป็น 32 ส่วนๆละ 8 ค่าซึ่งเท่ากับจำนวน subband ของ Filterbank

6) คำนวณค่า Power Spectrum ให้อยู่ในหน่วย dB ดังสมการ

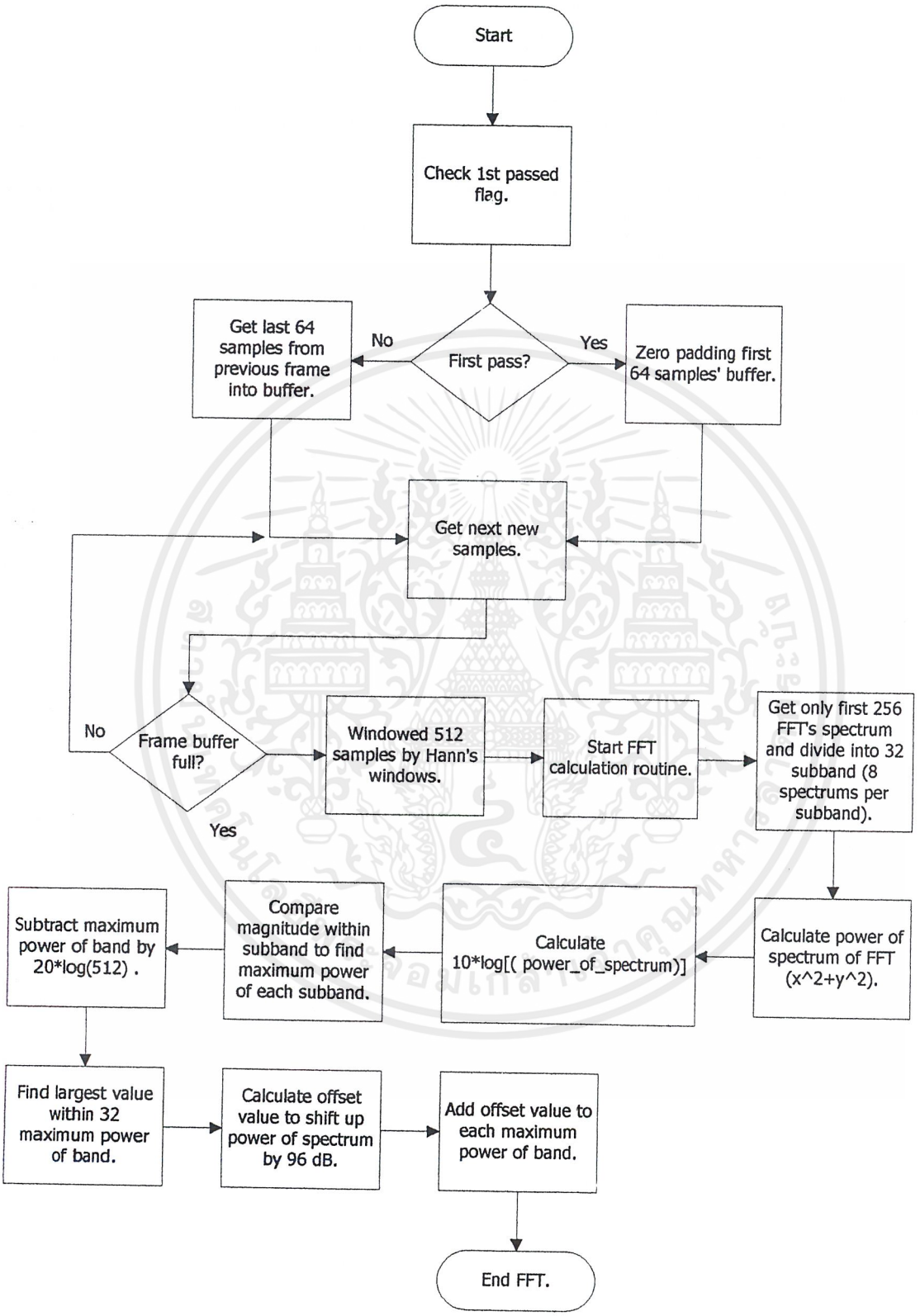
$$power\_spectrum(i)=10*Log10(x_i^2+y_i^2) \quad [dB]$$

7) นำค่าที่ได้มาเปรียบเทียบกันภายในส่วนที่ได้แบ่งไว้ (32 ส่วน) เพื่อหาค่าที่มากที่สุดสำหรับใช้เป็นตัวแทน 32 ค่า (P_i) ที่จะนำไปเปรียบเทียบกับ Scale Factor ซึ่งหาได้จาก Polyphase Filterbank
8) หาค่าที่มากที่สุดจากตัวแทน 32 ค่านั้น เพื่อใช้คิดค่า Offset ที่จะยกระดับพลังงานขึ้นไป 96 dB ดังนี้

$$Offset=96-MAX(P_i)$$

9) บวกค่า Offset เพิ่มให้กับ P_I ทุกๆค่า โดยค่าพลังงานที่มากที่สุดจะมีค่าเป็น 96dB พอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 FFT Flow Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Logarithm Base10 Algorithm [Log10(X)]

อธิบายได้ดังนี้

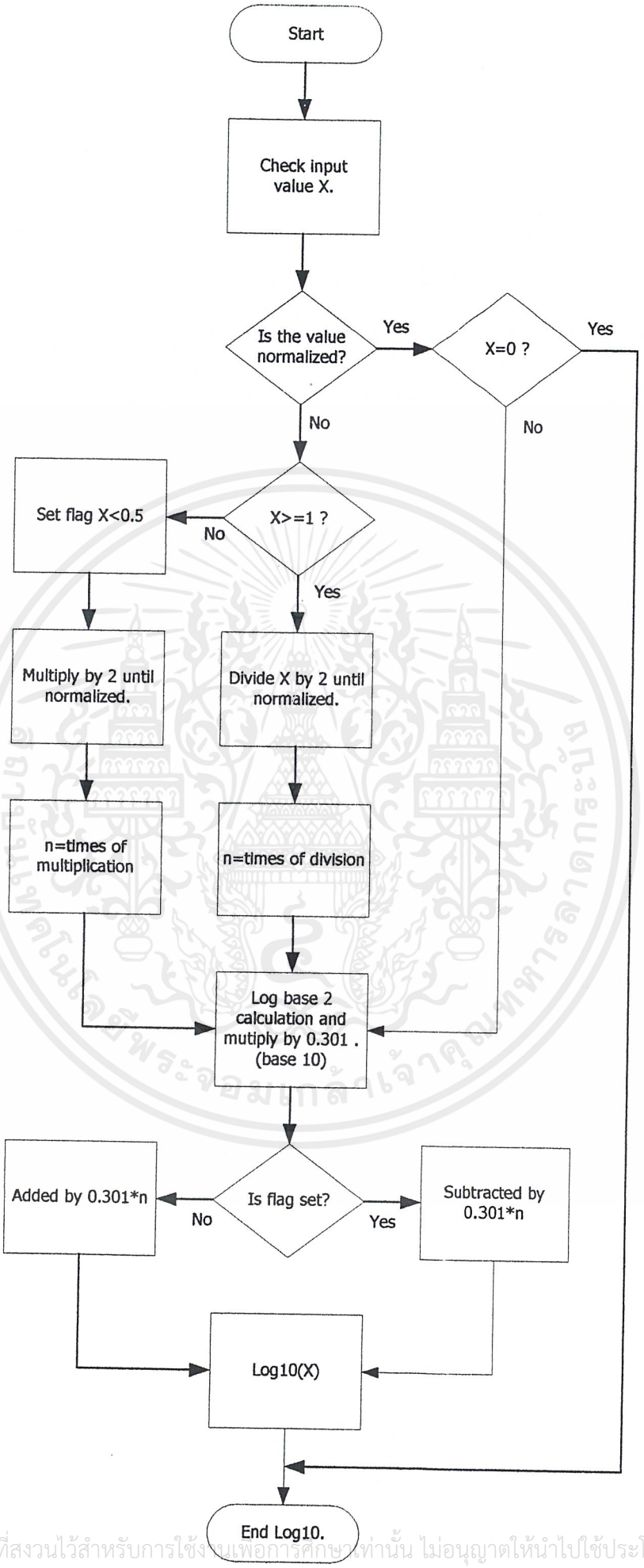
1) ตรวจสอบค่า X โดย
   - ถ้าค่า X=0 จะออกจากโปรแกรมทันที
   - ถ้า X>=1 หรือ X<0.5 (Unnormalized) โปรแกรมก็จะทำการ Normalize ค่า X โดย
     a) ถ้า X>=1 จะทำการหารค่า X ด้วย 2 จนกระทั่ง X<1 แล้วเก็บจำนวนครั้งที่หาร (n) ไว้
     b) ถ้า X<0.5 จะทำการคูณค่า X ด้วย 2 จนกระทั่ง X>=0.5 แล้วเก็บจำนวนครั้งที่คูณ (n) ไว้และ set flag X<0.5 ไว้ด้วย
   - ถ้า 0.5=< X<1 (Normalized) ก็จะทำการคำนวณ Function Logarithm ต่อไป
2) คำนวณ Logarithm ฐาน 2 [Log2(Y) ; Y=Normalized(X)] จากสมการ

   $Log2(Y)=4*(-.3372223\ Y^2 + .9981958\ Y - .6626105)$
3) แปลงให้อยู่ในรูป Logarithm ฐาน 10 โดยคูณผลลัพท์ที่ได้ด้วย Log10(2)=0.3010

   Log10(Y)=Log10(2)*Log2(Y)
4) ตรวจสอบสถานะของ Flag X<0.5
   - ถ้า Flag X<0.5 ถูก Set หมายถึงค่า input X มีค่าน้อยกว่า 0.5 นั่นคือเราจะต้องลบค่าที่ได้จากข้อ 3 ด้วย 0.301*n โดยที่ n คือจำนวนครั้งที่คูณค่า X ด้วย 2
   - ถ้า Flag X<0.5 ไม่ถูก Set หมายถึงค่า input X มีค่ามากกว่า 0.5 นั่นคือเราจะต้องบวกค่าที่ได้จากข้อ 3 ด้วย 0.301*n โดยที่ n คือจำนวนครั้งที่หารค่า X ด้วย 2 หรือถ้าหากว่าค่า X อยู่ในช่วง Normalize ค่า n ก็จะมีค่าเท่ากับศูนย์โดยอัตโนมัติ

ผลลัพท์ที่ได้หลังจากการบวกหรือลบนั้นจะเป็นค่าของ Log10(X) นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.3 Flow Diagram ของ Function Logarithm ฐาน 10

บทที่ 8

วิเคราะห์ผลการทดลองและสรุปผล

ผลการทดลองตอนที่ 1

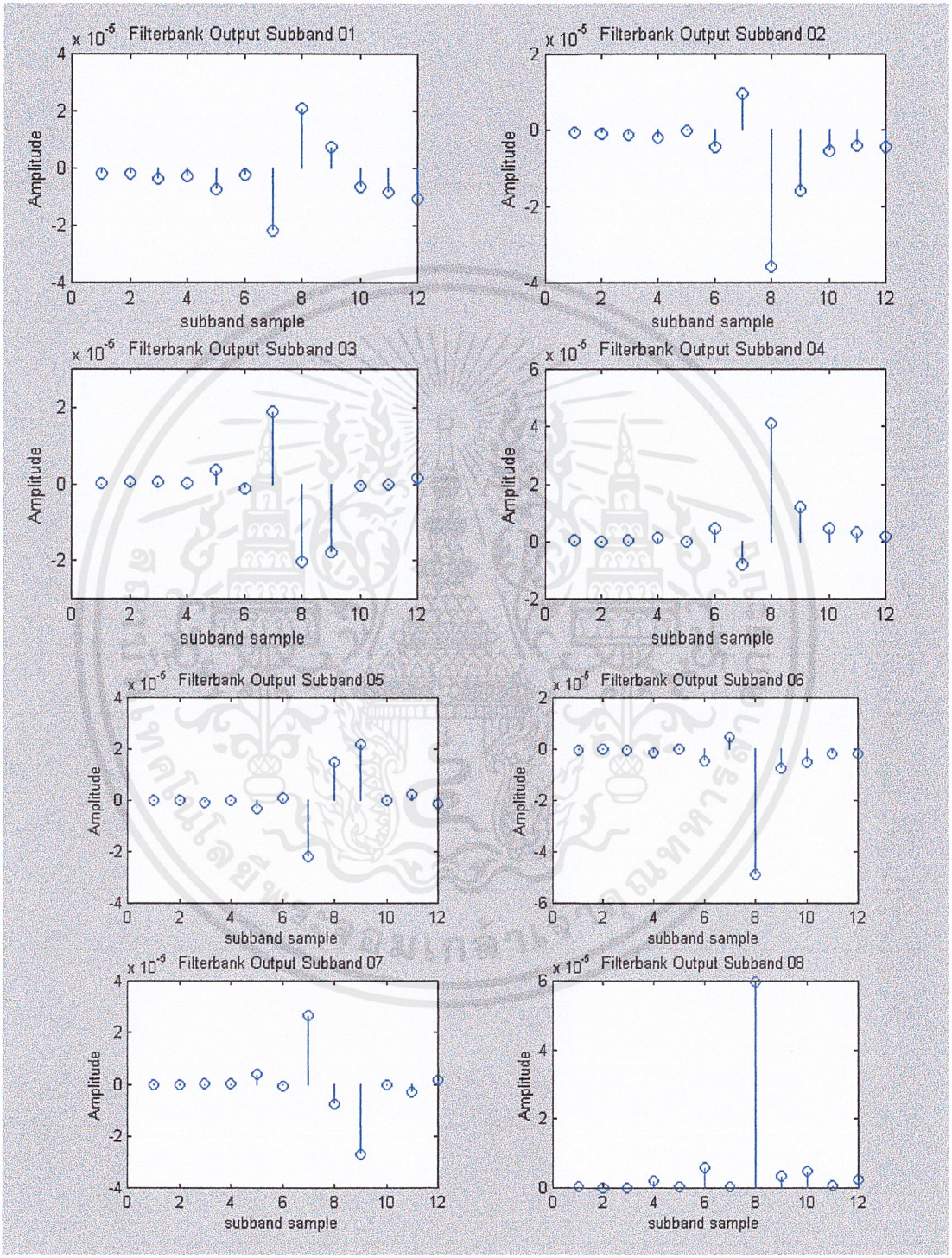
ผลการทดลองโปรแกรมภาษาแอสแซมบลี้ ในส่วนของ Filter bank และ FFT โดยการป้อนอินพุทรูปซายน์ 1 ความถี่ f = 10 kHz แอมปลิจูด A = 0.002 ที่อัตราการแซมปลิ้ง (Sampling rate) = 44.1 kHz และผลการทดลองโปรแกรมที่เขียนโดย MATLAB ในส่วนของ Psychoacoustics โดยการป้อนอินพุทรูปซายน์ 1 ความถี่ f = 10 kHz แอมปลิจูด A = 0.002 ที่อัตราการแซมปลิ้ง (Sampling rate) = 44.1 kHz

ผลการทดลองตอนที่ 2

ผลการทดลองโปรแกรมภาษาแอสแซมบลี้ ในส่วนของ Filter bank และ FFT โดยการป้อนอินพุทรูปซายน์ 3 ความถี่ 1) f = 1 kHz ,A = 0.003 2) f = 10 kHz , A=0.002 3) f = 11 kHz , A = 0.001 ที่อัตราการแซมปลิ้ง (Sampling rate) = 44.1 kHz และผลการทดลองโปรแกรมที่เขียนโดย MATLAB ในส่วนของ Psychoacoustics โดยการป้อนอินพุทรูปซายน์ 3 ความถี่ 1) f = 1 kHz ,A = 0.003 2) f = 10 kHz , A=0.002 3) f = 11 kHz , A = 0.001 ที่อัตราการแซมปลิ้ง (Sampling rate) = 44.1 kHz

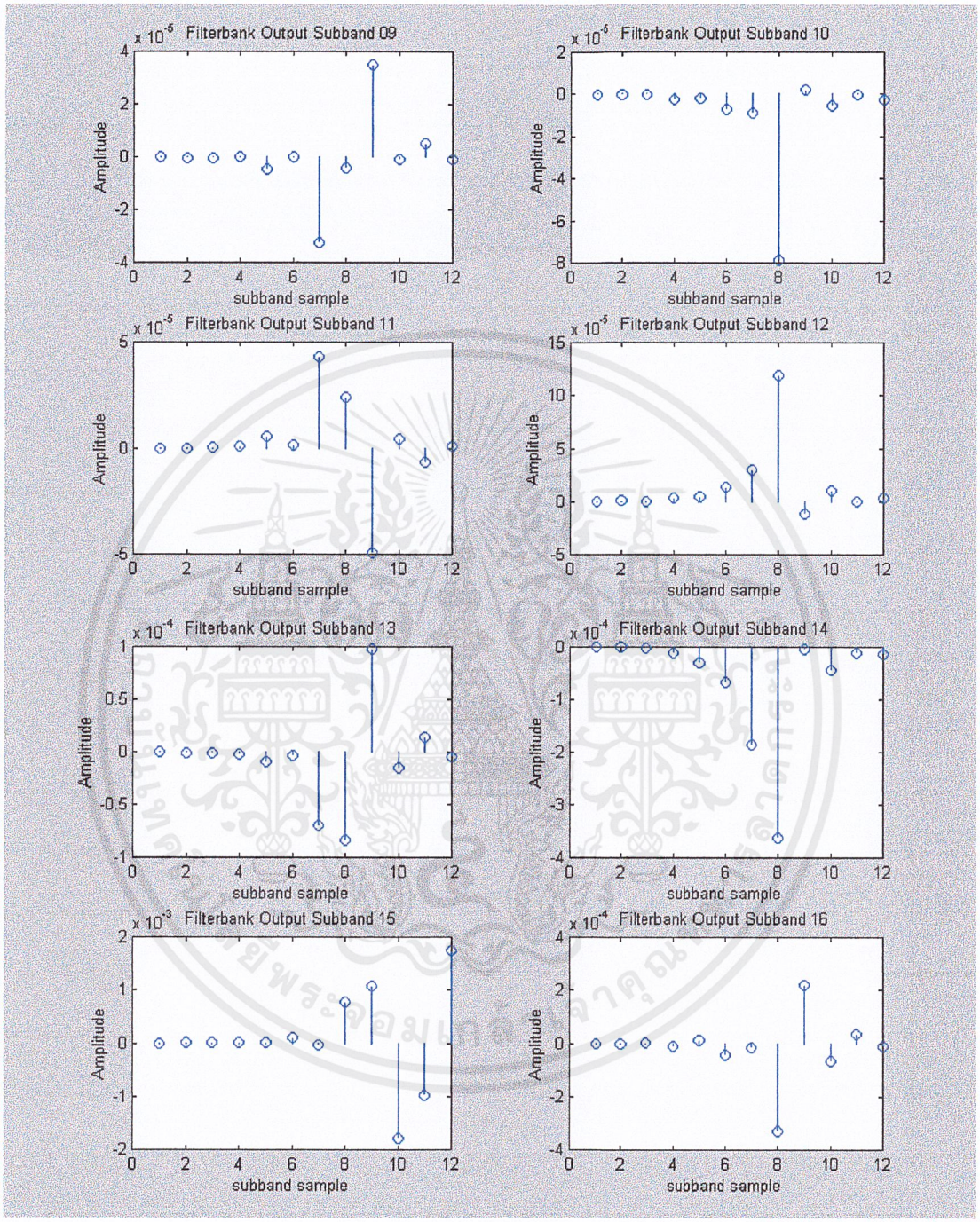
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองตอนที่ 1 ป้อนอินพุทรูปซายน์ 1 ความถี่ f = 10KHz , A= 0.003



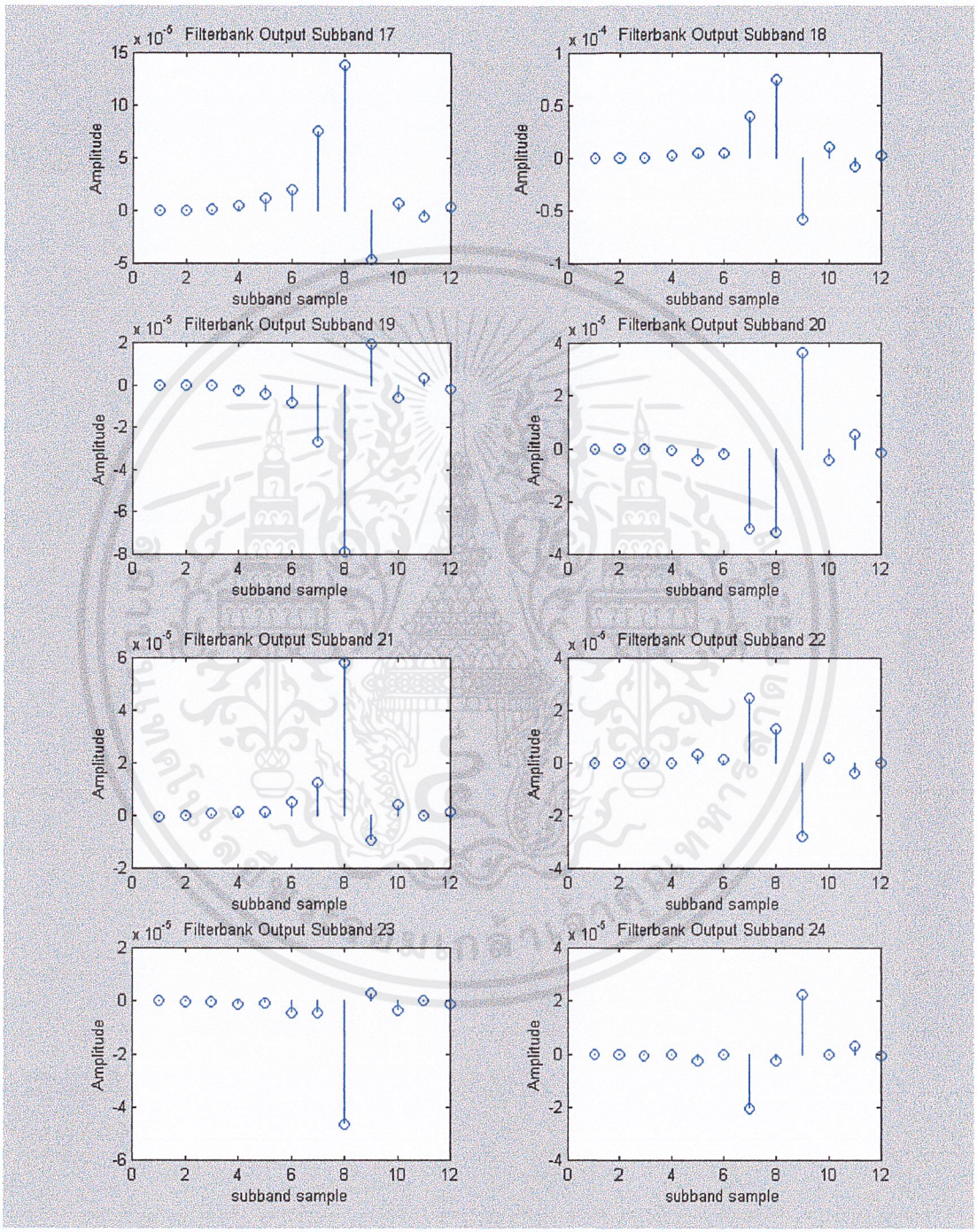
รูปที่ 8.1 ก แสดงกราฟ Filterbank Output Subband 01-08 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



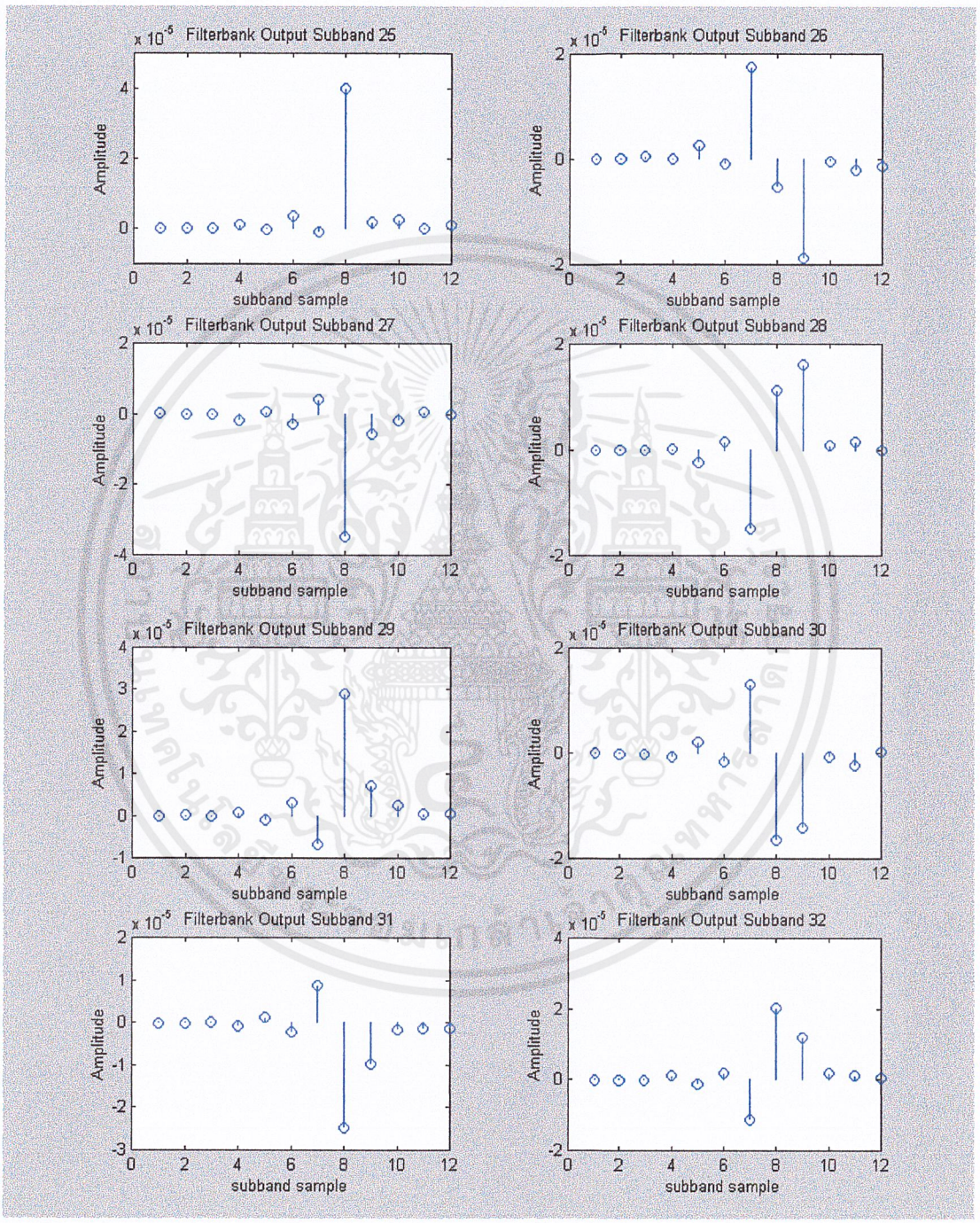
รูปที่ 8.1 ข แสดงกราฟ Filterbank Output Subband 09-16 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



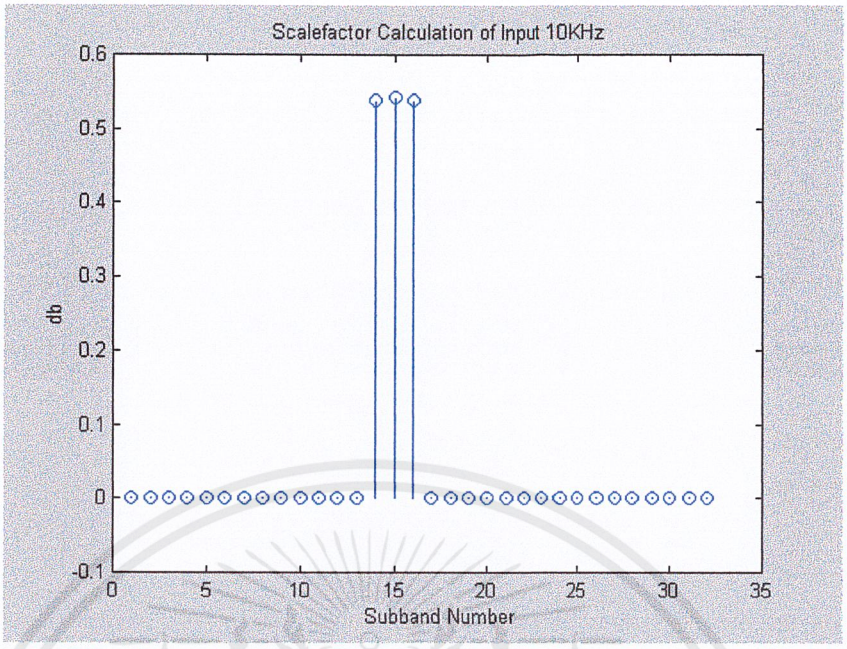
รูปที่ 8.1 ค แสดงกราฟ Filterbank Output Subband 17-24 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

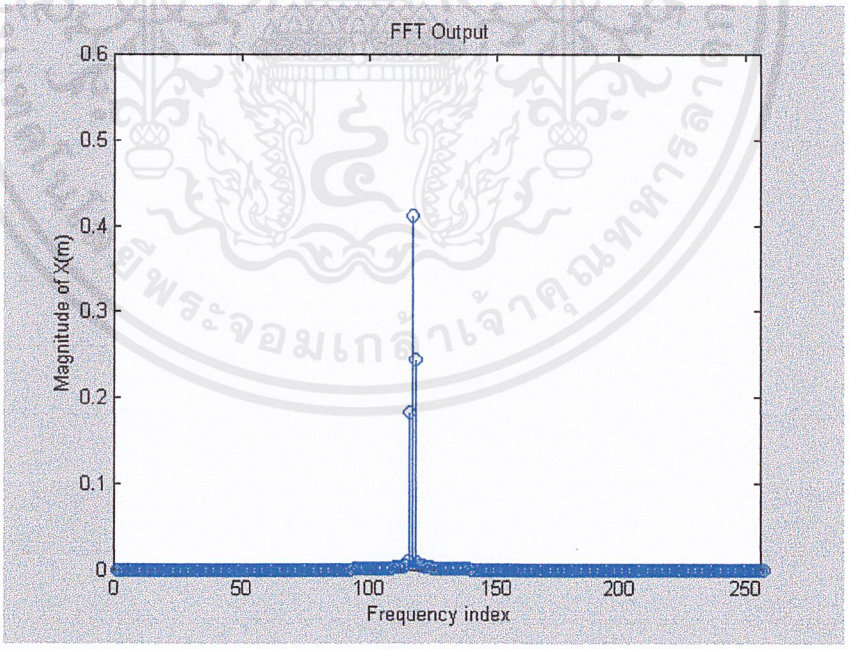


รูปที่ 8.1 ง แสดงกราฟ Filterbank Output Subband 25-32 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

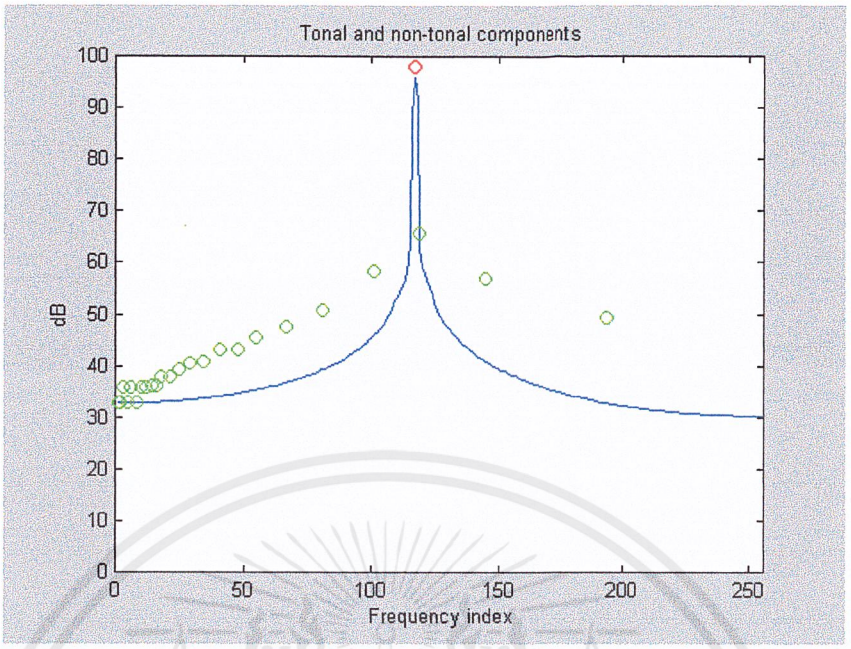


รูปที่ 8.2 แสดงกราฟ Scalefactor Calculation (In Unit) ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

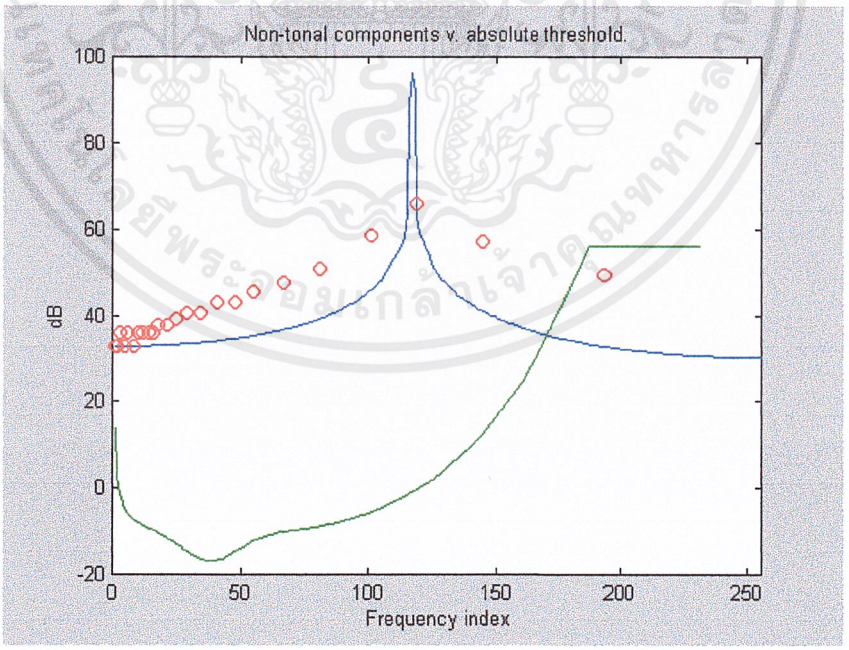


รูปที่ 8.3 แสดงกราฟ FFT – 256 In Maxnitute Of X(m) ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

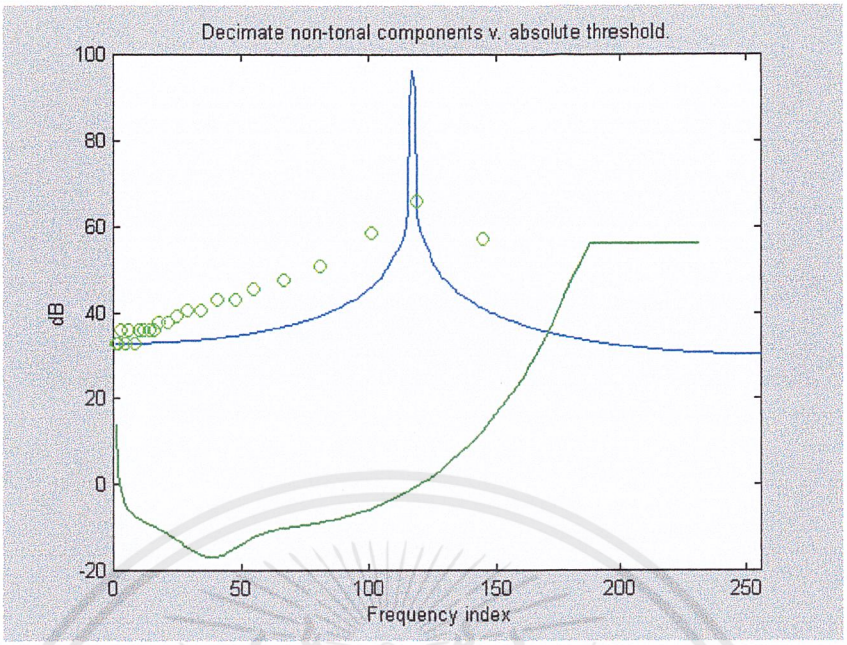


รูปที่ 8.6 แสดงกราฟ Tonal and non-tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

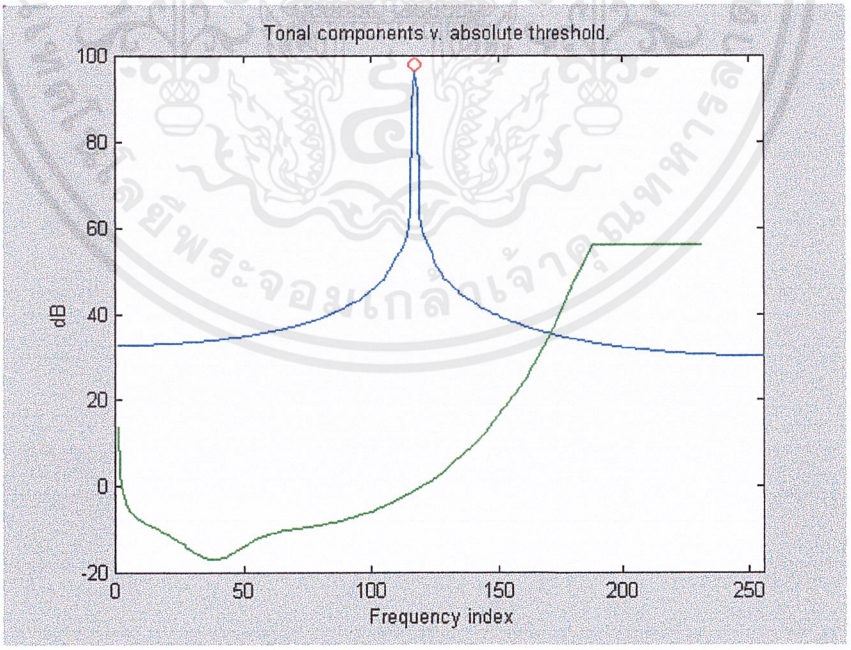


รูปที่ 8.7 แสดงกราฟ Non-tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

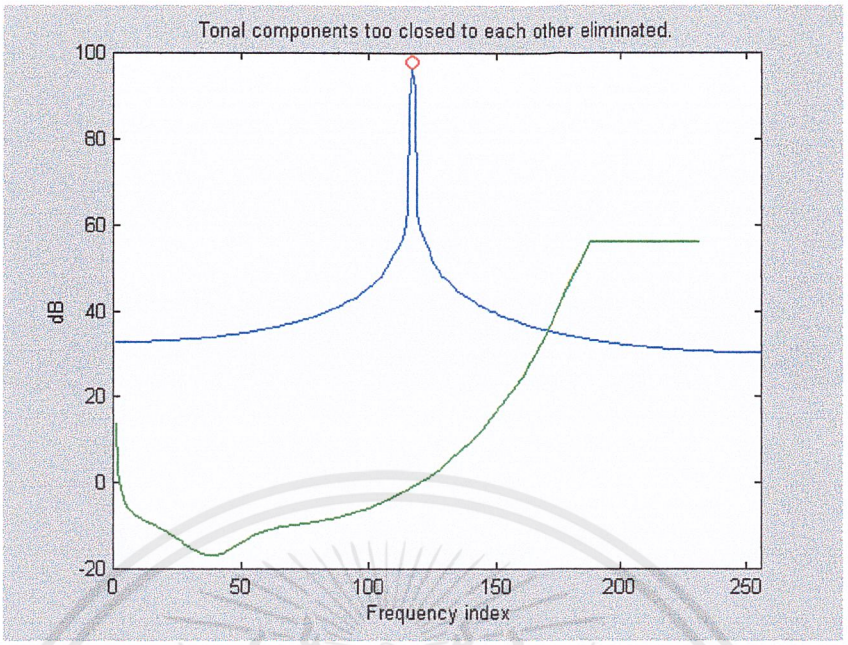


รูปที่ 8.8 แสดงกราฟ Decimate non-tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1KHz

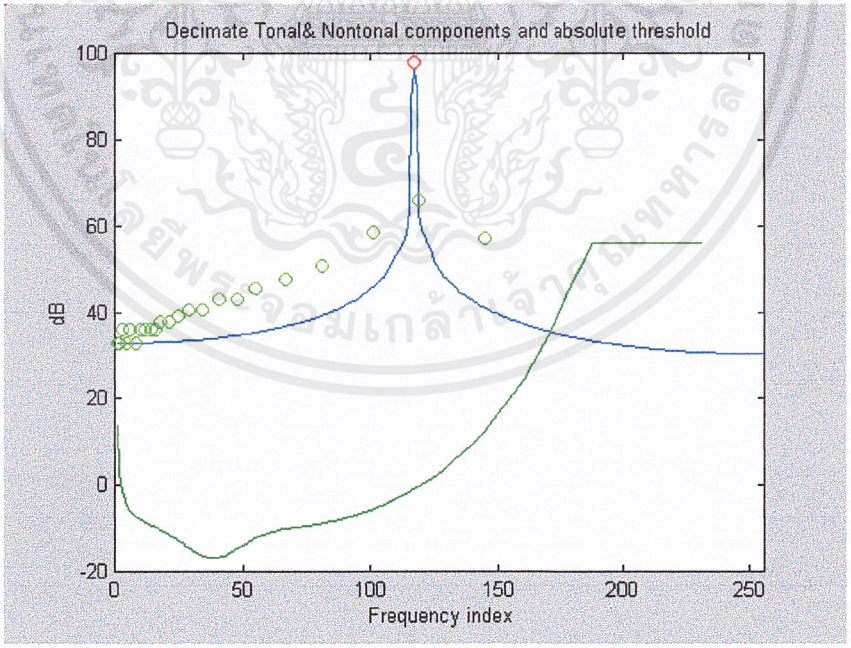


รูปที่ 8.9 แสดงกราฟ Tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

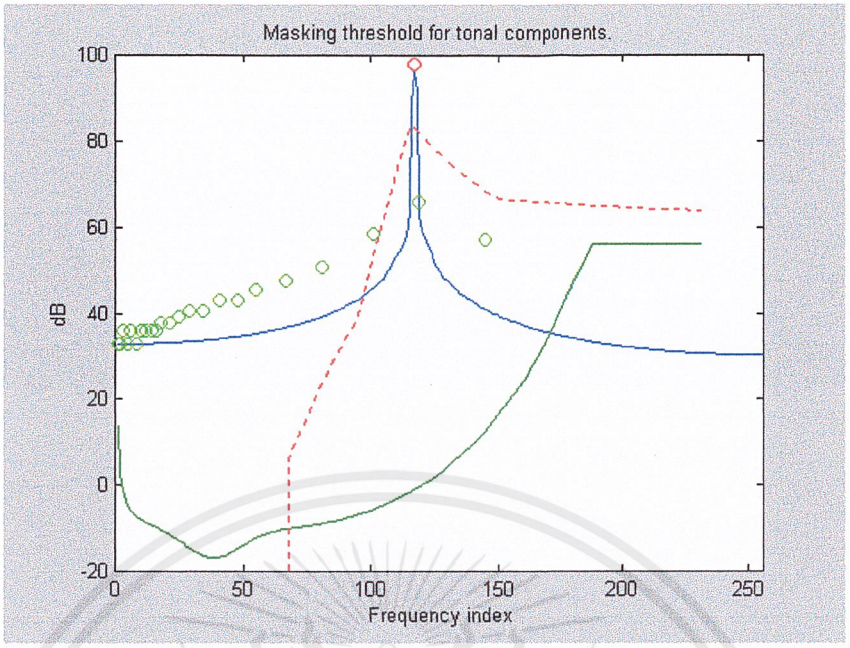


รูปที่ 8.10 แสดงกราฟ Tonal components too closed to each other eliminated จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1KHz

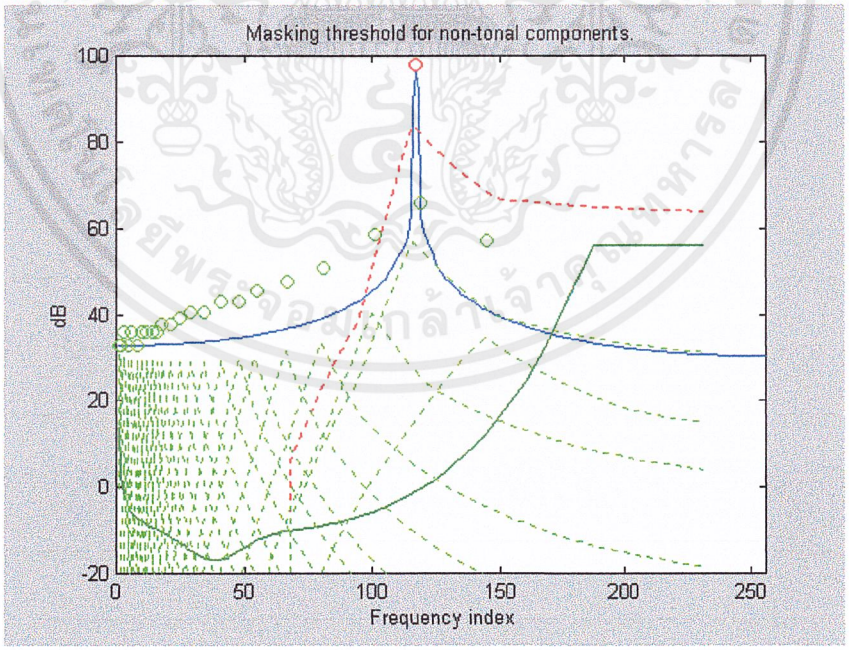


รูปที่ 8.11 แสดงกราฟ Decimate tonal &Non-tonal components and absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

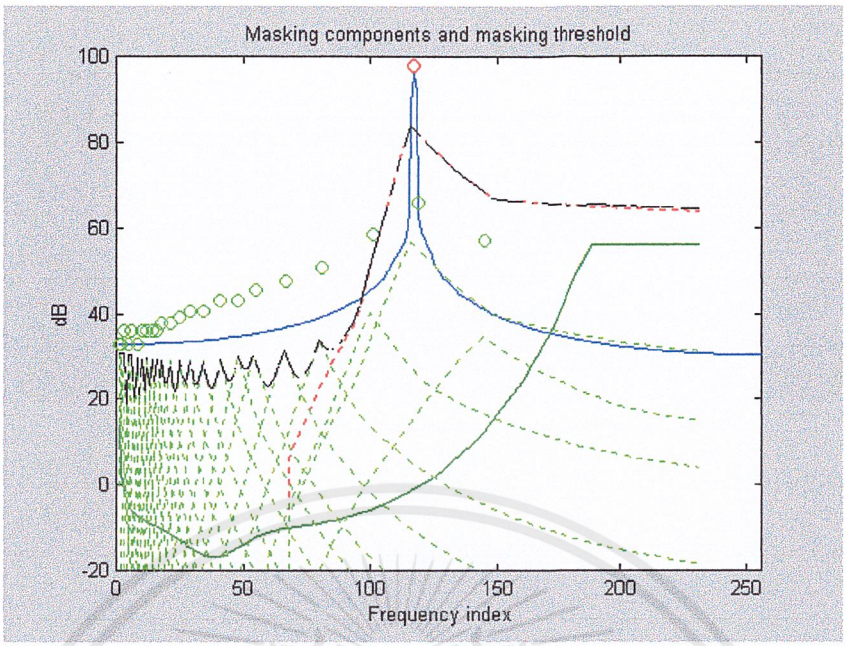


รูปที่ 8.12 แสดงกราฟ Masking threshold for tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

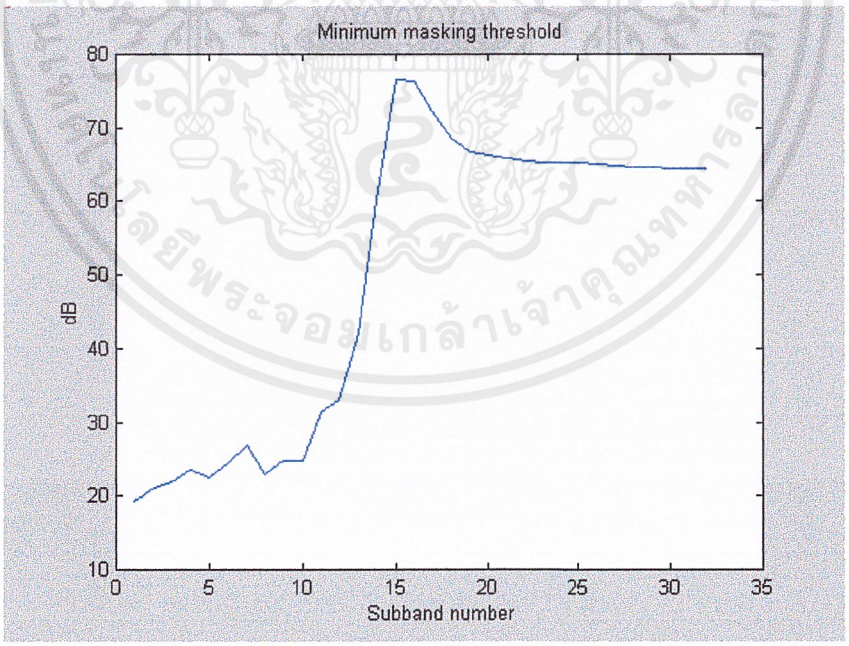


รูปที่ 8.13 แสดงกราฟ Masking threshold for non-tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

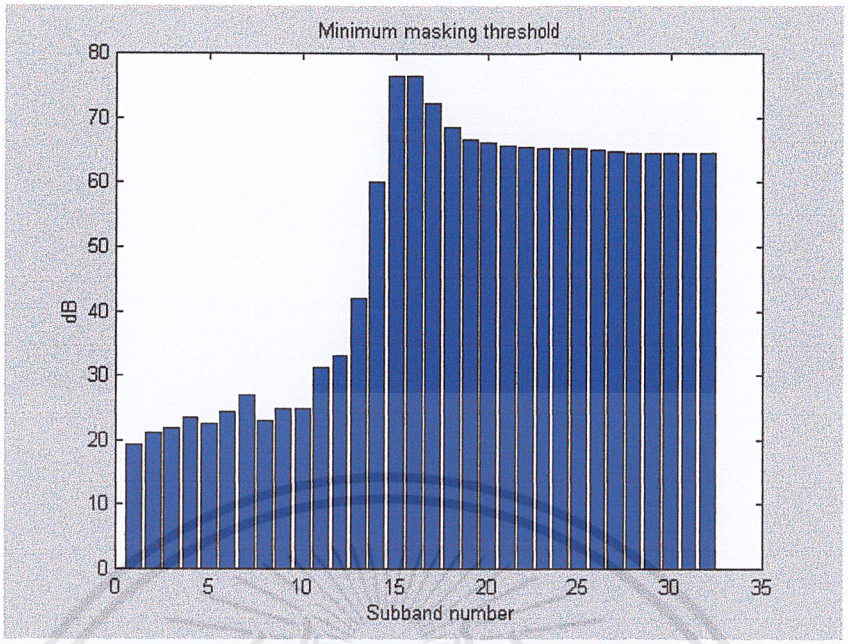


รูปที่ 8.14 แสดงกราฟ Masking components and masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

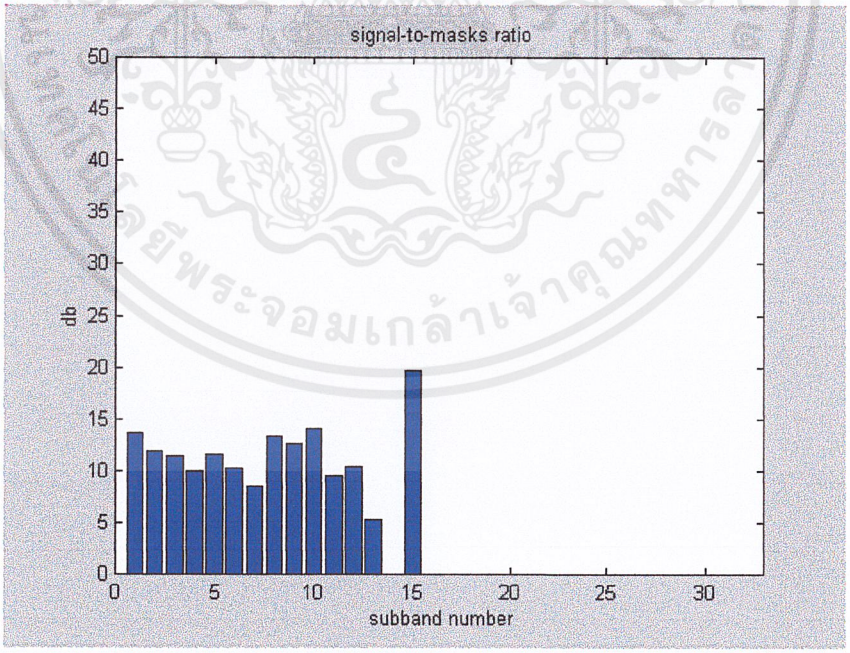


รูปที่ 8.15 แสดงกราฟ Minimum masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



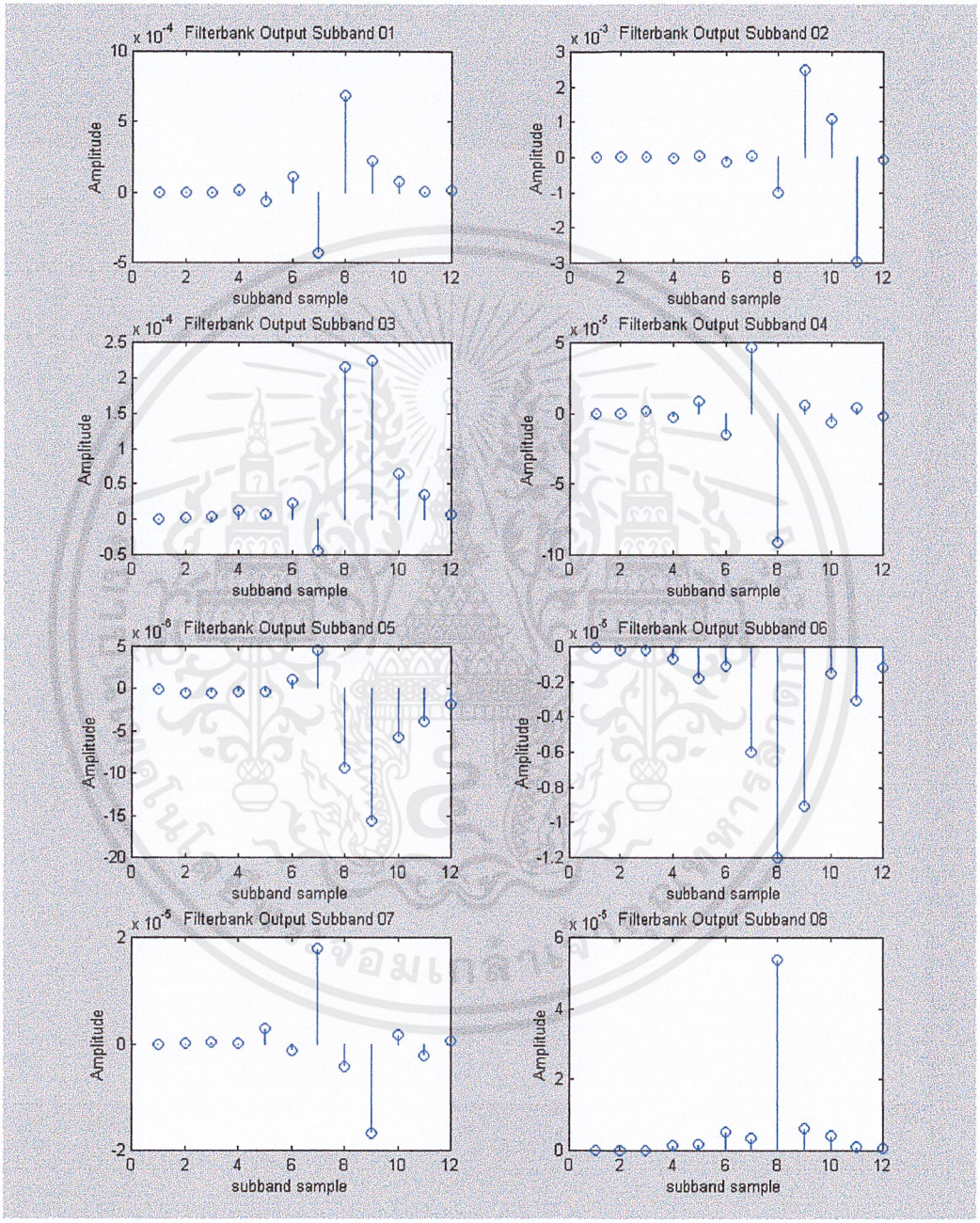
รูปที่ 8.16 แสดงกราฟ Minimum masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz



รูปที่ 8.17 แสดงกราฟ Signal-to-masks ratio จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไปหนึ่งความถี่ คือ f = 10 KHz , แอมปลิจูด A = 0.003 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

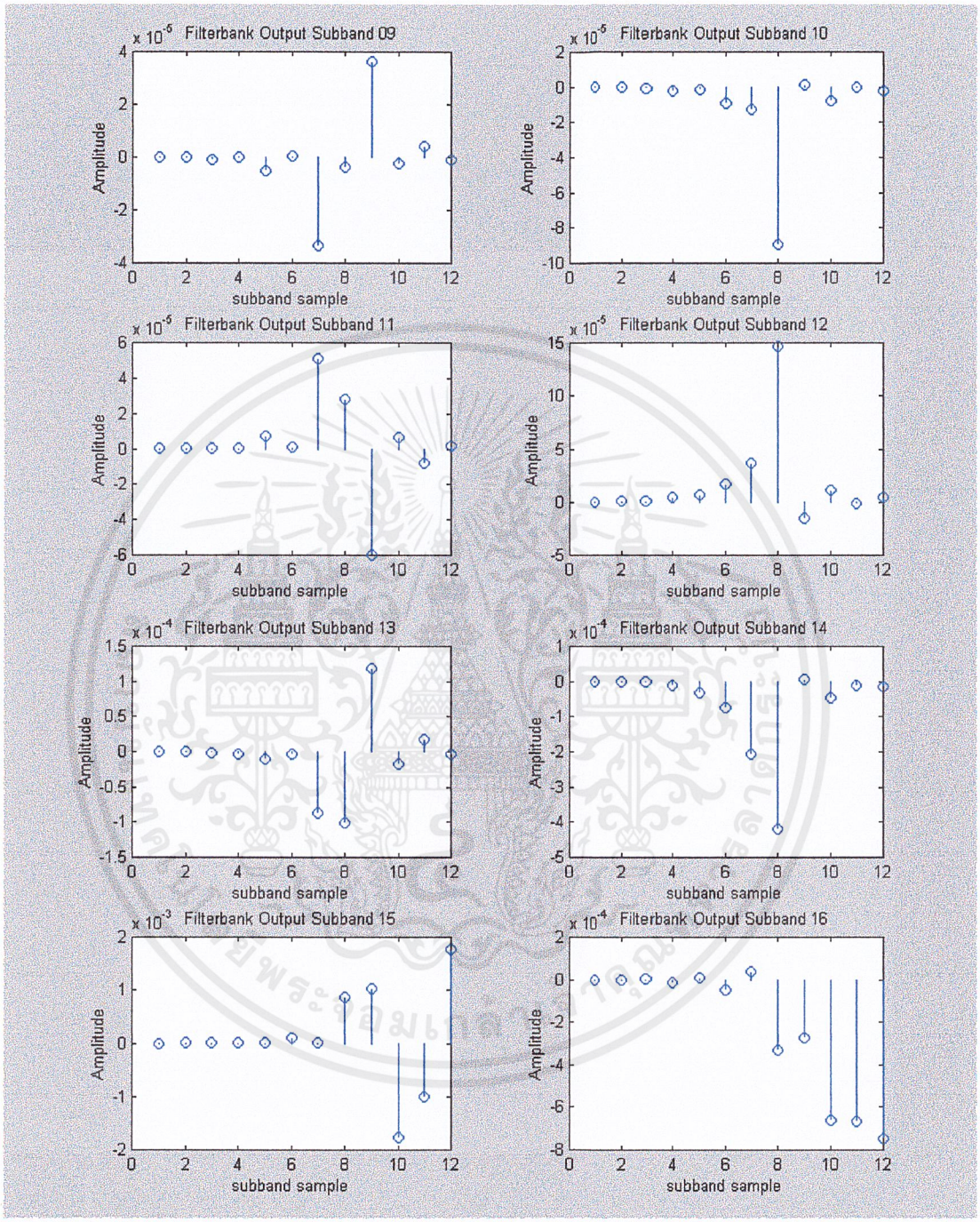
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองตอนที่ 2 ป้อนอินพุทรูปซายน์ 3 ความถี่ (1) f = 1KHz , A= 0.003
(2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001



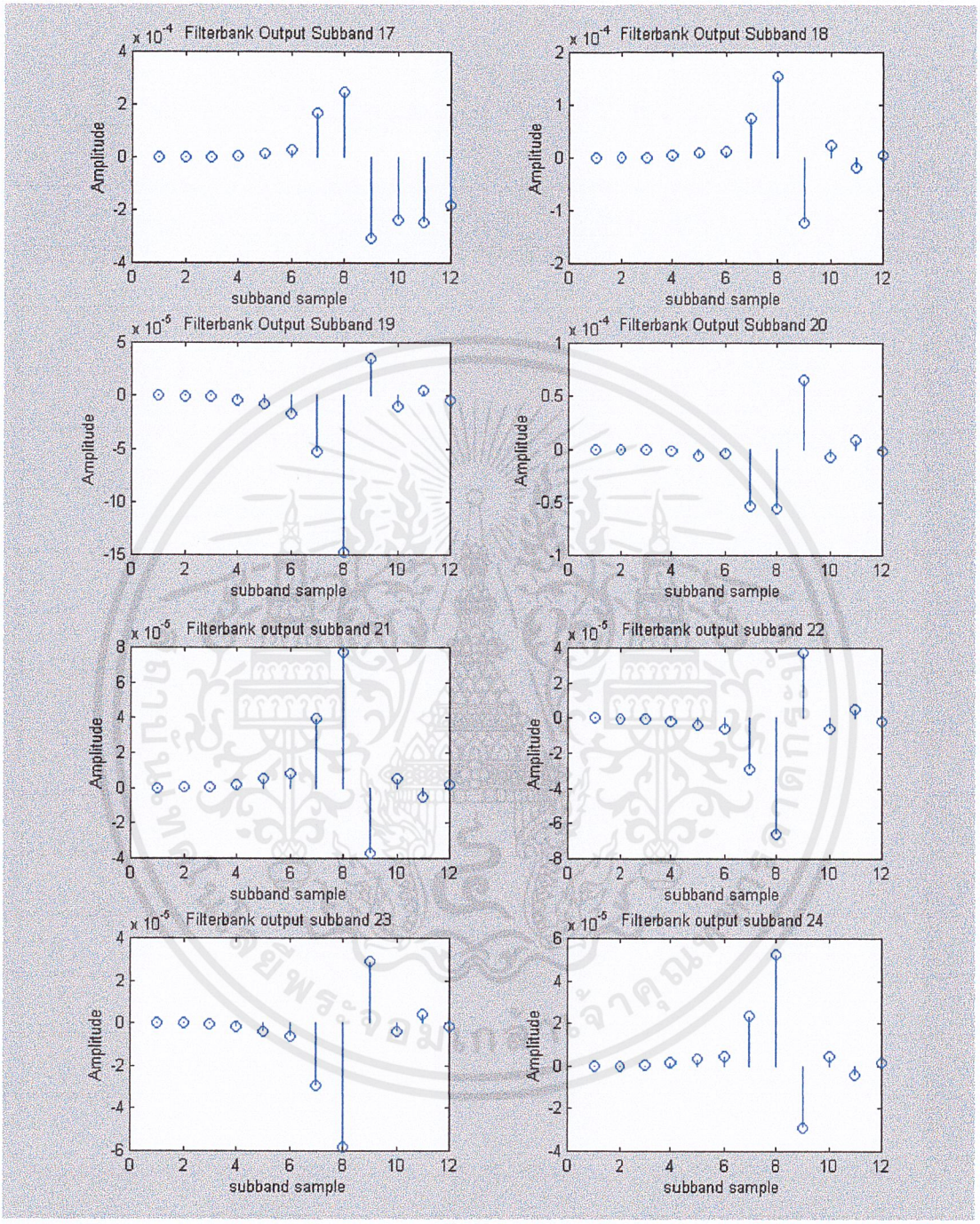
รูปที่ 8.18 ก แสดงกราฟ Filterbank Output Subband 01-08 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง (Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



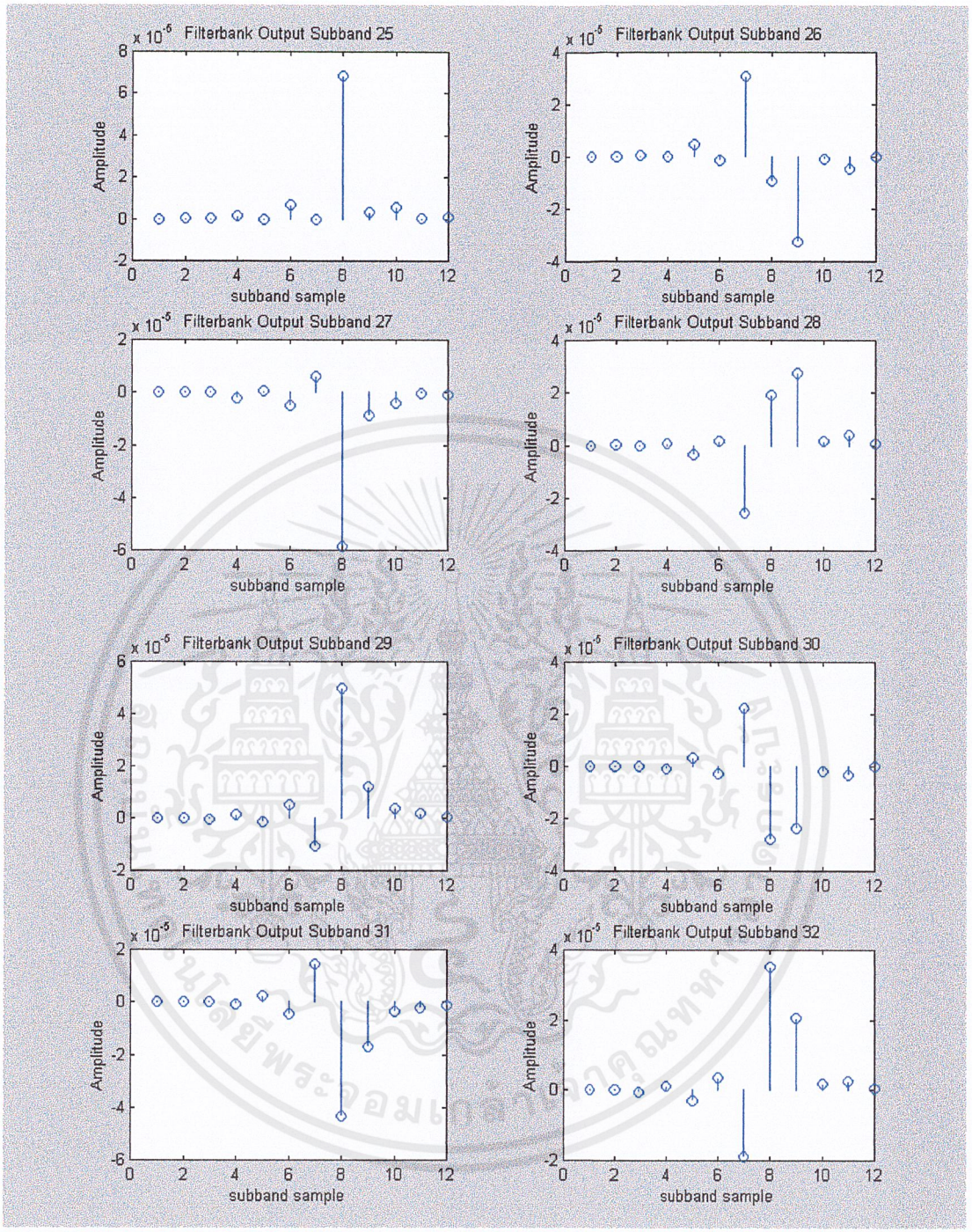
รูปที่ 8.18 ข แสดงกราฟ Filterbank Output Subband 09-16 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



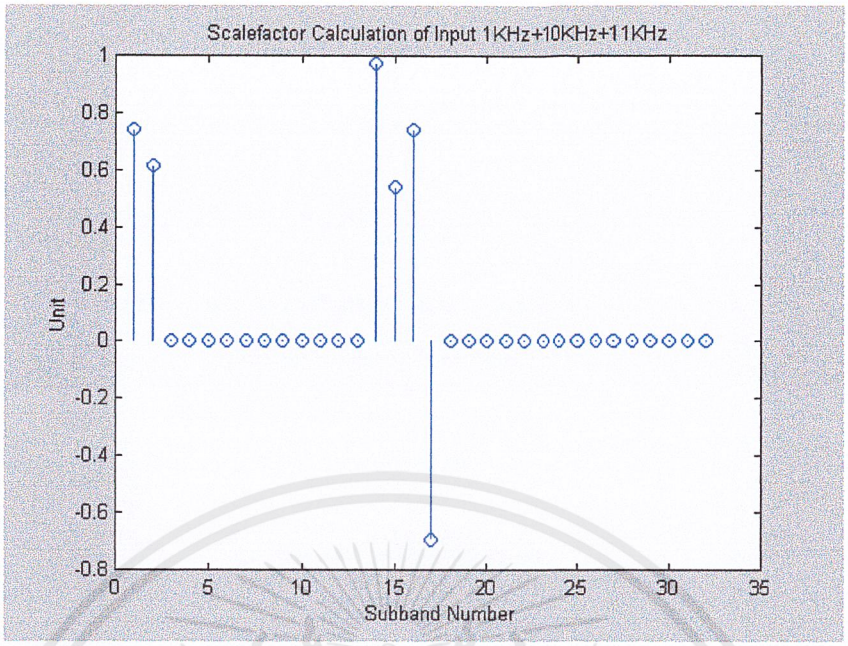
รูปที่ 8.18 ค แสดงกราฟ Filterbank Output Subband 17-24 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง (Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

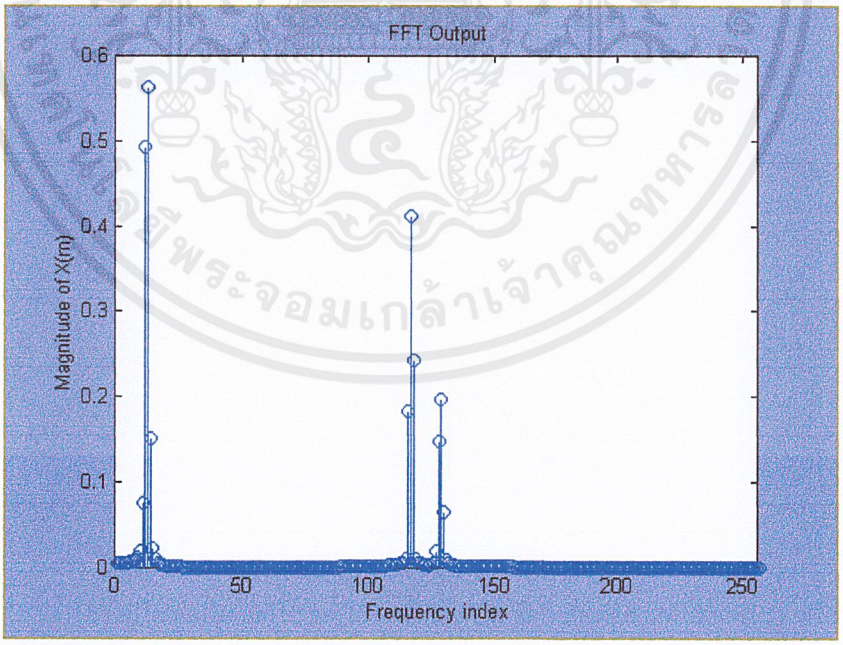


รูปที่ 8.18 ง แสดงกราฟ Filterbank Output Subband 25-32 ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง (Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

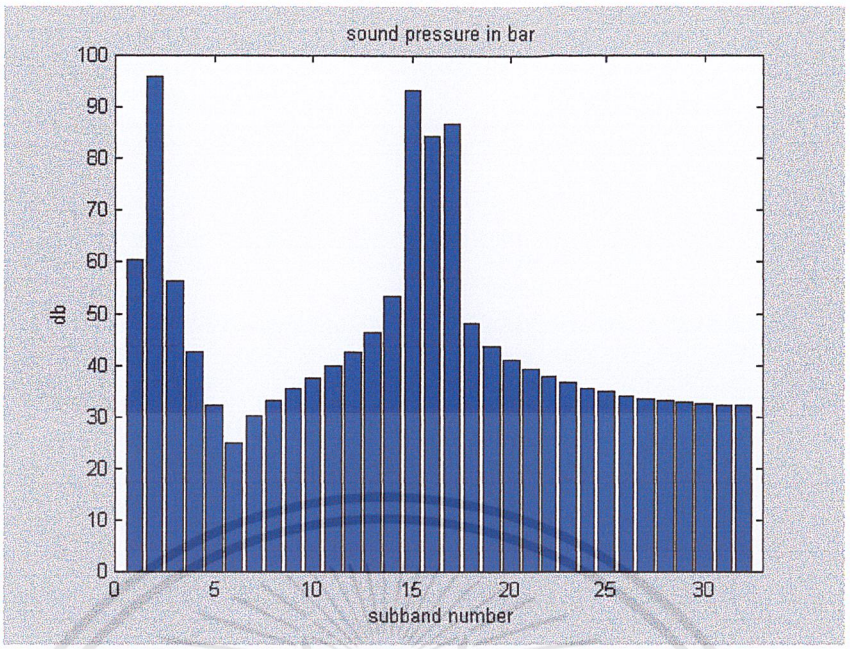


รูปที่ 8.19 แสดงกราฟ Scalefactor Calculation ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการแซมปลิ้ง(Sampling rate) = 44.1 KHz

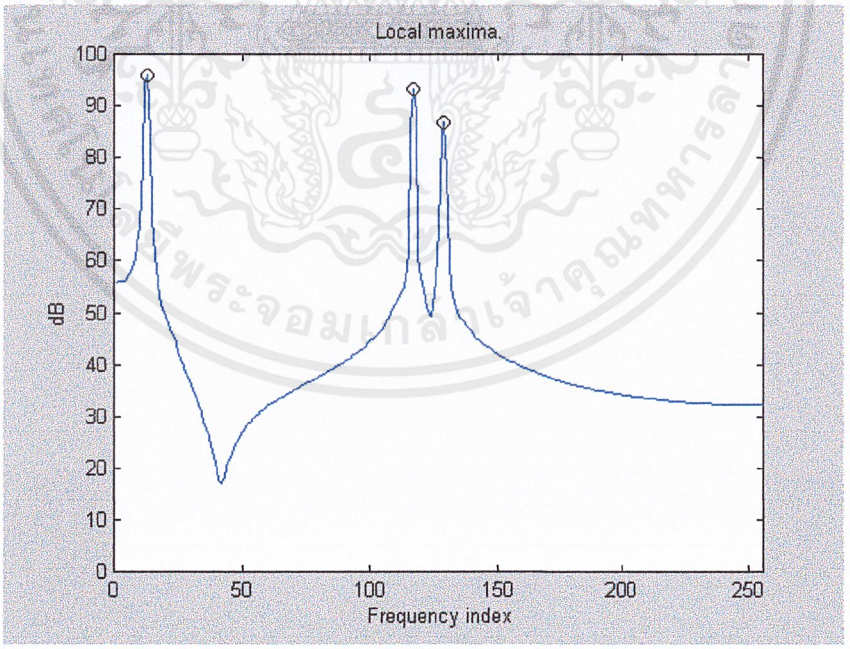


รูปที่ 8.20 แสดงกราฟ FFT – 256 In Maxnitute Of X(m) ซึ่งค่าที่นำมาพล็อตได้มาจาก Motorola DSP56000 Sofware Simulator โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1)f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการแซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้ใช้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

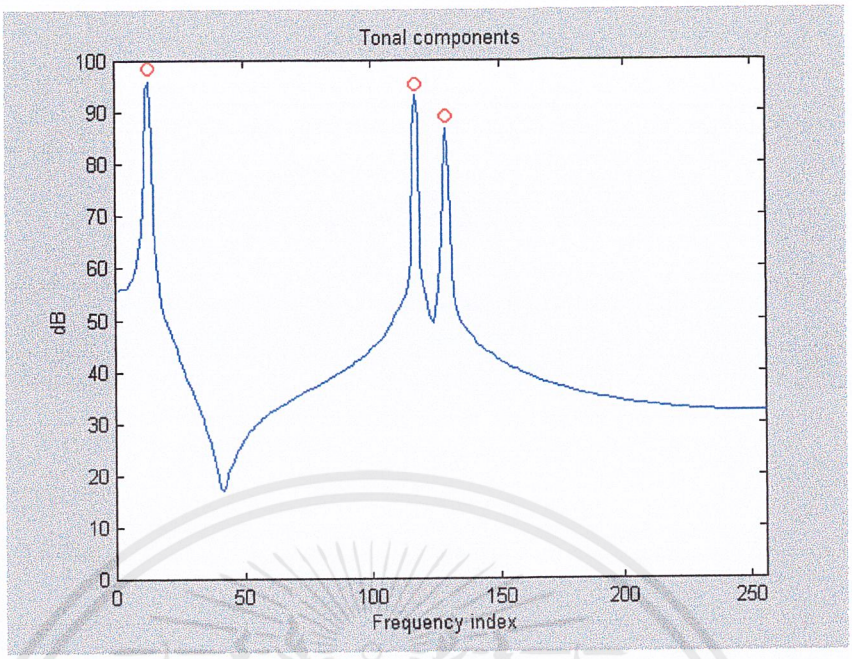


รูปที่ 8.21 แสดงกราฟ Sound pressure in bar จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1KHz

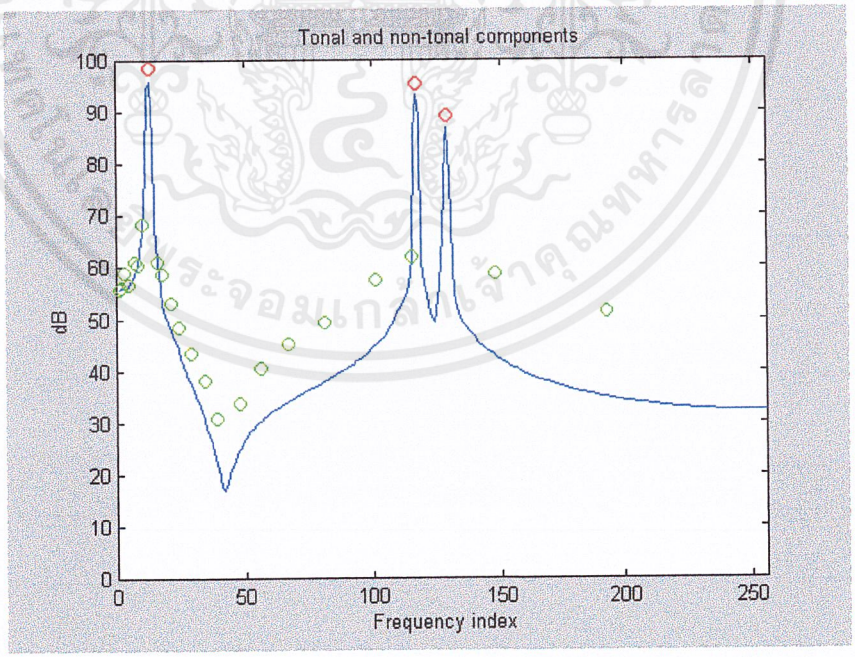


รูปที่ 8.22 แสดงกราฟ Local maxima จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

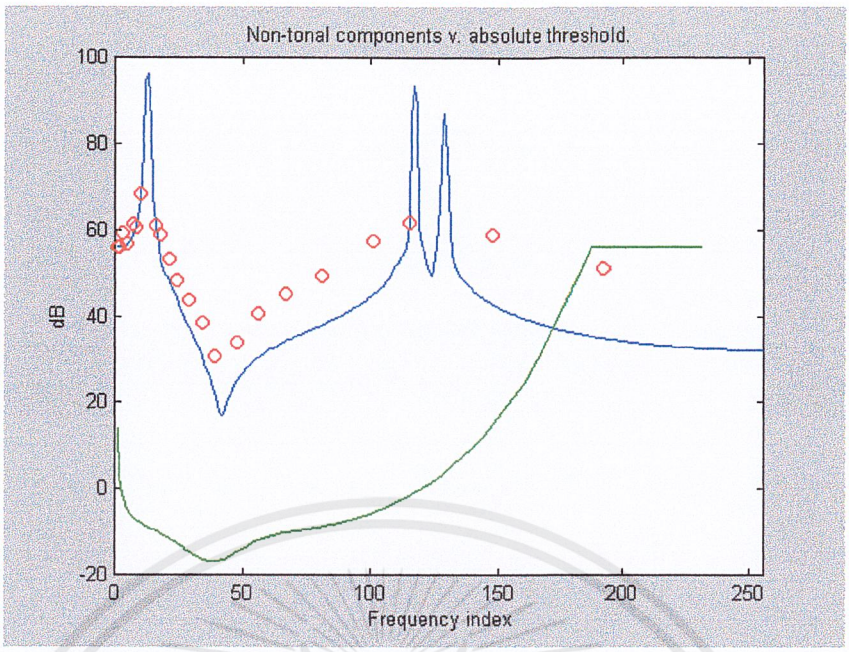


รูปที่ 8.23 แสดงกราฟ Tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

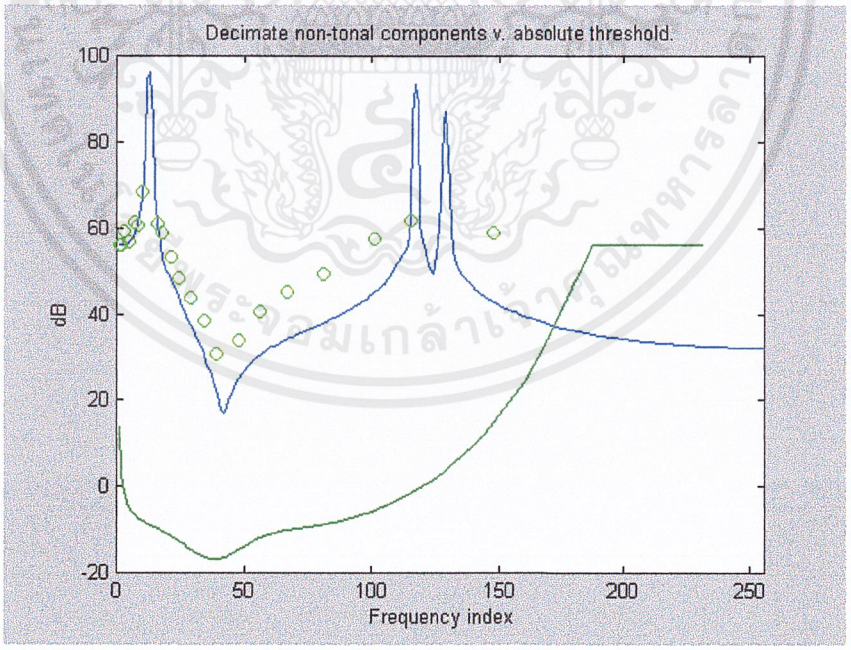


รูปที่ 8.24 แสดงกราฟ Tonal and non-tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

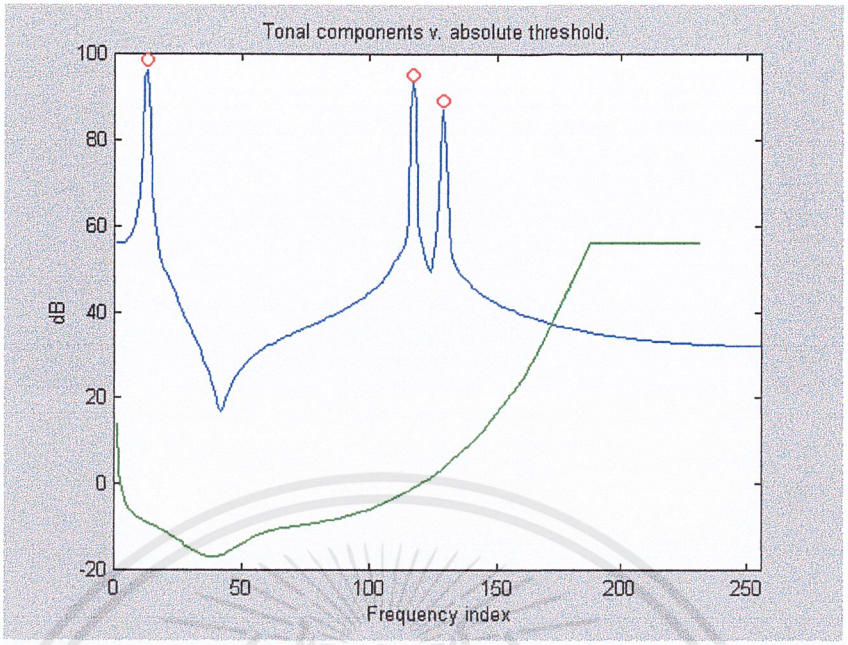


รูปที่ 8.25 แสดงกราฟ Non-tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1)f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการแซมปลิ้ง(Sampling rate) = 44.1 KHz

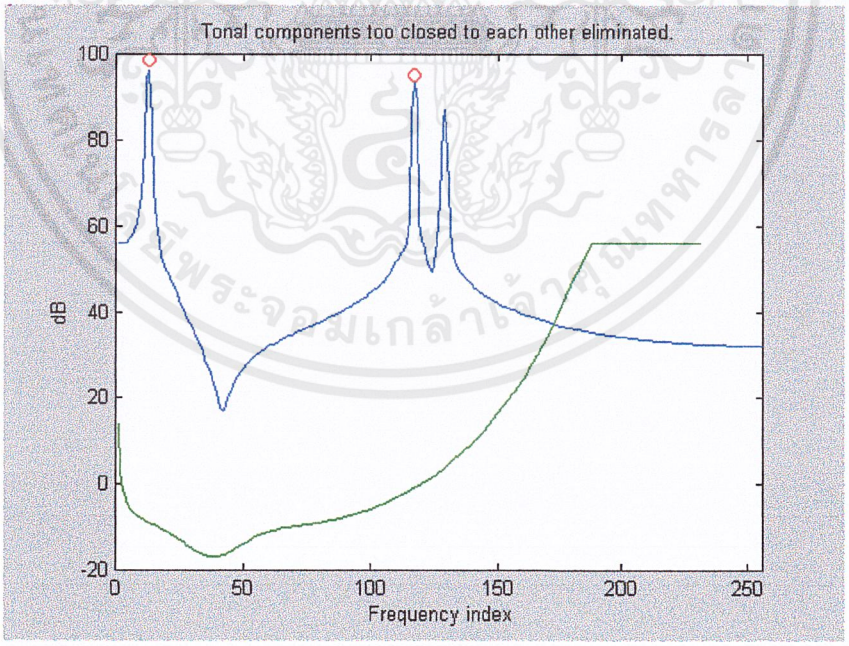


รูปที่ 8.26 แสดงกราฟ Decimate non-tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

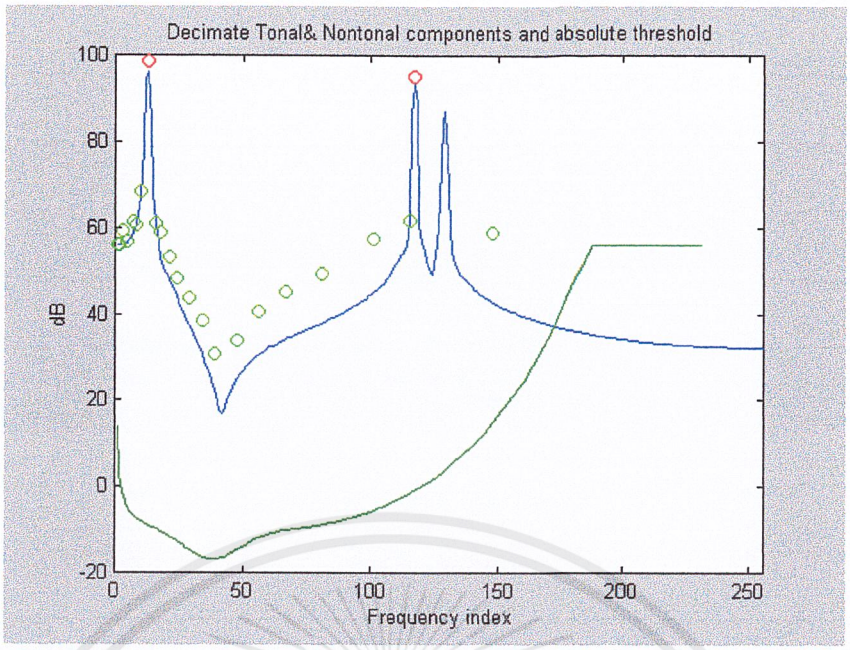


รูปที่ 8.27 แสดงกราฟ Tonal components v. absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1)f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการแซมปลิ้ง(Sampling rate) = 44.1 KHz

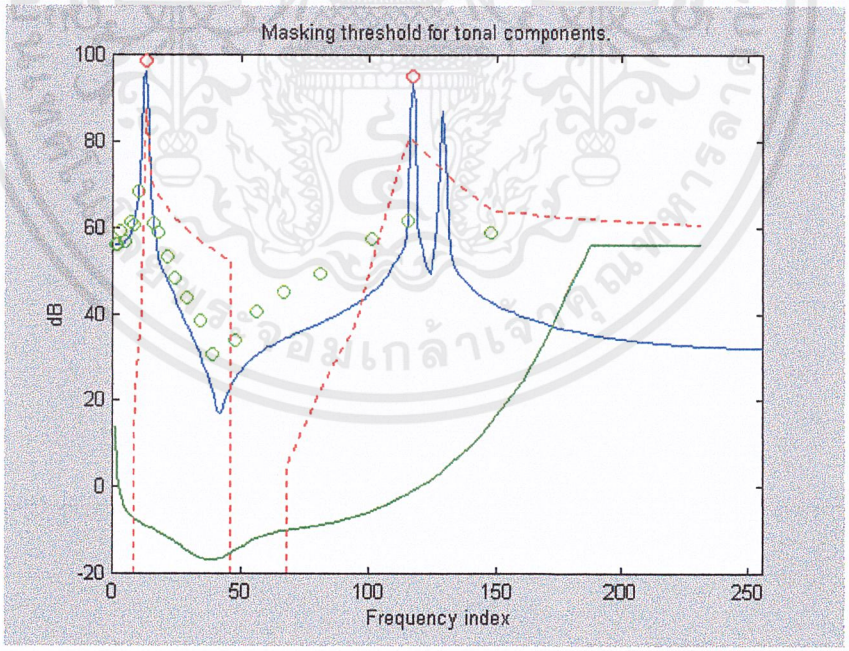


รูปที่ 8.28 แสดงกราฟ Tonal components too closed to each other eliminated จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

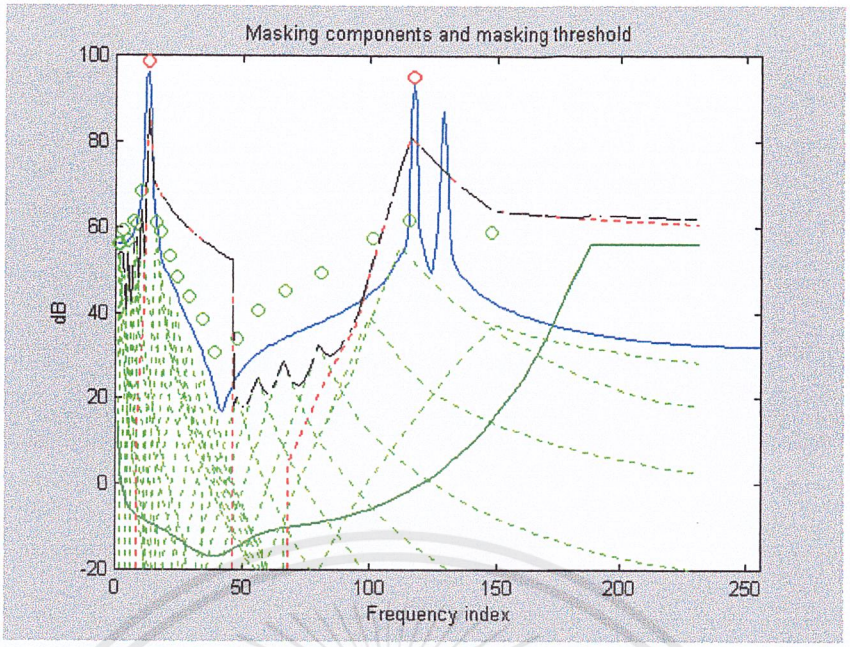


รูปที่ 8.29 แสดงกราฟ Decimate Tonal&Non-tonal components and absolute threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

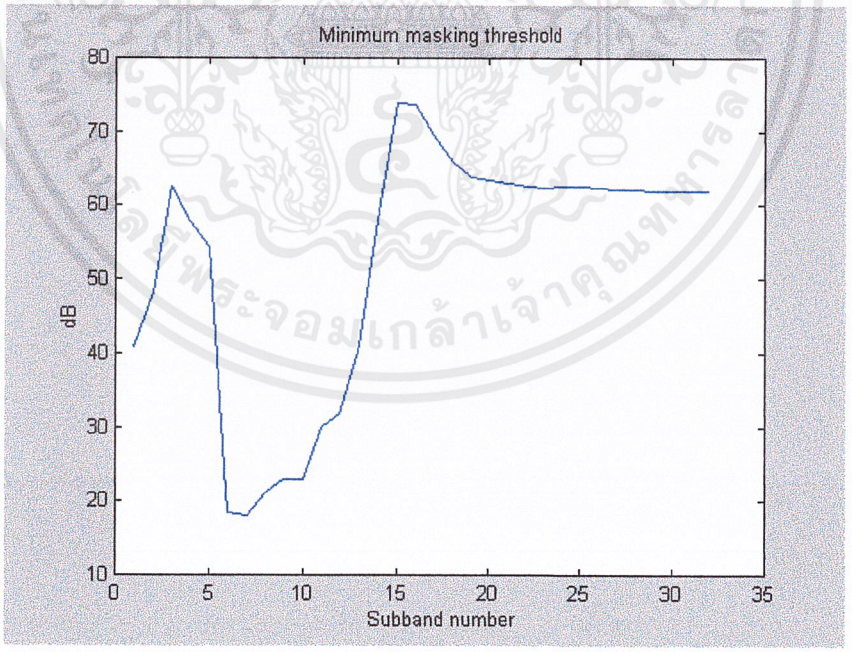


รูปที่ 8.30 แสดงกราฟ Masking threshold for tonal components จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1)f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการแซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

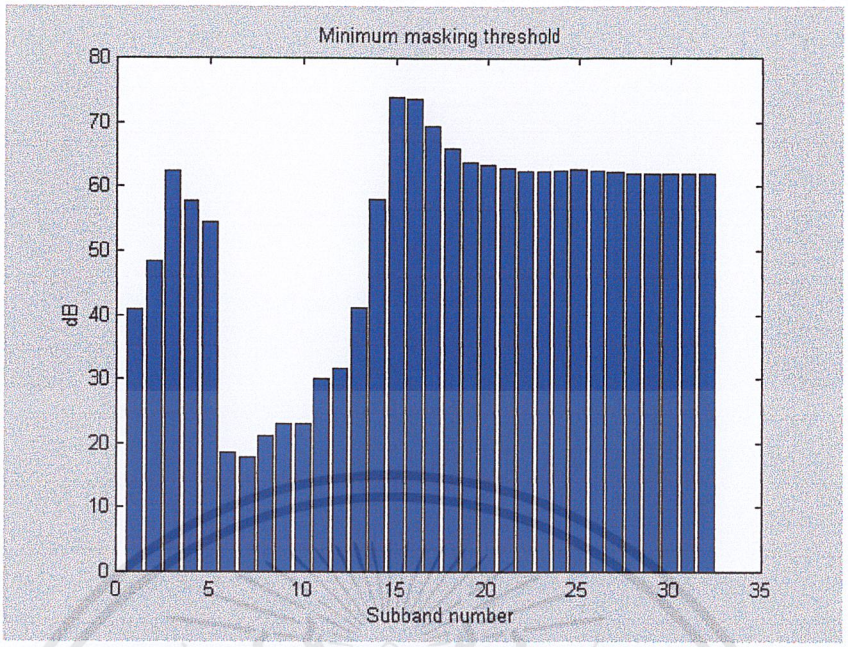


รูปที่ 8.31 แสดงกราฟ Masking components and masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

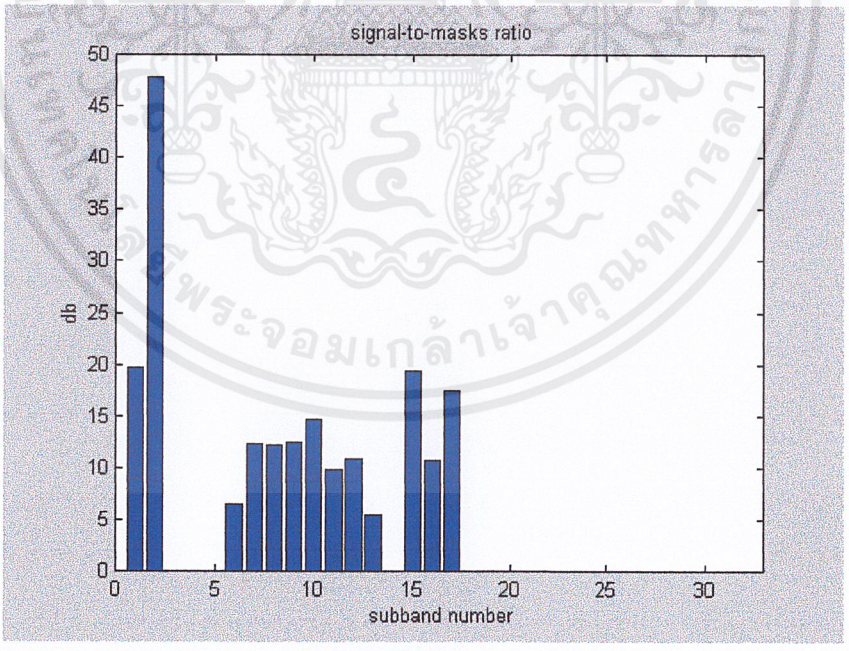


รูปที่ 8.32 แสดงกราฟ Minimum masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง (Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.33 แสดงกราฟ Minimum masking threshold จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz



รูปที่ 8.34 แสดงกราฟ Signal to masks ratio จากการเขียนโปรแกรมส่วน Psychoacoustics โดยใช้โปรแกรม MATLAB โดยที่ป้อนอินพุทคลื่นรูปซายน์เข้าไป 3 ความถี่ คือ (1) f = 1KHz , A= 0.003 (2) f = 10KHz , A= 0.002 (3) f = 11KHz , A= 0.001 ที่อัตราการ แซมปลิ้ง(Sampling rate) = 44.1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ผลการทดลอง

ส่วนที่ 1 เป็นคำอธิบายของอินพุท 10000 Hz ที่ 0.002

รูปที่ 8.1 ก- 8.1ง ที่อินพุท 10000 Hz ขนาด 0.002 จะเสมือนเป็นตัวกรองความถี่บางช่วงผ่าน (BandpassFilter) โดยจะแบ่งออกเป็น 32 แถบความถี่ย่อย ณ ที่ความถี่สุ่มที่ 41000 Hz เราจะสังเกตเห็นว่า แอมปลิจูดของความถี่อินพุท 10000 Hz จะออกไปปรากฏในตำแหน่งแถบความถี่ย่อยที่ 15 โดย Filter Bank แบบนี้จะเป็นการแบ่งสัญญาณอินพุทที่มีช่วงสัญญาณความถี่กว้างออกให้เป็นช่วงสัญญาณความถี่ที่เท่าๆกัน โดยมีจุดมุ่งหมายเพื่อจำลองรูปแบบของแถบความถี่วิกฤตของหูคนเรา ส่วนแอมปลิจูดจะมีขนาดใกล้เคียงกับในส่วนทางด้านอินพุท คือใกล้เคียง 0.002ด้วย แต่ที่ช่วงแถบความถี่ย่อยอื่นๆ จะเห็นได้อย่างชัดเจนว่ามีเอาพุทออกมาน้อยมากๆ ต่างกันเป็นร้อยเท่าขึ้นไป จึงสามารถบอกเราได้ว่าตัวกรอทางดิจิตอลงชนิดนี้มีประสิทธิภาพสูง(มีคัทออฟที่คม)ในส่วนของการประมาณว่า โดยอาจจะคำนวณโดยคร่าวๆว่าสัญญาณอินพุทที่เราป้อนเข้าไปนั้นไปปรากฏที่แถบความถี่ย่อยใด
หาได้จากสูตร

แถบความถี่ย่อยใดๆ = (64*input frequency) / Fs

รูปที่ 8.2 เป็นส่วนที่แสดงผลลัพธ์ของค่าที่ได้จากการเทียบค่าจากการโค้ดแบบสับแบนด์ หาได้จากทุกๆ 8 ค่าในสับแบนด์เดียวแล้วนำค่าที่มากที่สุดมาเปรียบเทียบกับค่าในตาราง โดยถือว่าจะเอาค่าที่มาเป็นค่าแรกมาเป็นตัวแทนของค่านั้นๆ เห็นได้ว่าจะมีเพียงสับแบนด์เดียวเท่านั้นที่มีค่าเด่นออกมา ซึ่งก็คือสับแบนด์ที่ 15 นั่นเอง ส่วนสับแบนด์อื่นๆ ก็ยังมีค่าอยู่เล็กน้อย ไม่มีผลอะไรมากมายนัก

รูปที่ 8.3 เป็นกราฟที่บอกค่าแมกนิจูดของสัญญาณในโดเมนเวลาที่เราป้อนเข้าไป หลังจากที่ผ่านการคำนวณแบบ FFTสังเกตว่า ณ ที่ความถี่ 10000 Hz จะมีบางสเปกตรัมเท่านั้นที่เด่นขึ้นมา และจะมีผลทำให้สปกตรัมที่อยู่ในย่านใกล้ๆ มีค่าขึ้นมาอีกเล็กน้อย อันเป็นผลมาจาก DFT Leakage ซึ่งหลีกเลี่ยงไม่ได้ แต่ต้องระวังว่าเหตุการณ์นี้จะไม่กระทบกับการคำนวณของเรา ถ้าสังเกตกราฟเราได้ว่าที่แอมปลิจูดสัญญาณอินพุทที่ 0.002 หลังจากผ่านการคำนวณ FFT แล้วจะให้แมกนิจูดที่ประมาณ 0.5โดยที่เราต้องใช้แอมปลิจูดขนาดนี้ก็เนื่องมาจากข้อจำกัดบางประการของ DSP

รูปที่ 8.4 เป็นส่วนที่สร้างขึ้นเพื่อการหาที่คาดว่าจะเป็นเสียงที่เราได้ยิน โดยมีหลักการว่าถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(X(k) > X(k-1) & X(k) >= X(k+1) & K > 2 & K< 250)จะกล่าวได้ว่าเป็นส่วนที่เราได้ยิน โดนก็คือการหาค่าที่มากสุดในย่านใกล้เคียงกัน อนึ่งไม่มีการระบุว่าจะต้องมาค่ามากเท่าไร เพียงแต่ให้ค่านั้นสอดคล้องกับเงื่อนไงข้างต้นก็เพียงพอแล้ว เมื่อได้ค่านี้จะนำไปคำนวณเพื่อหาเสียงที่ได้ยินจริงๆ ต่อไป สังเกตว่า ณ ที่ความถี่ 10000 Hz ก็จะมีเพียงแค่ความถี่เดียวที่สูงขึ้นมาเด่นกว่าคนอื่นในย่านเดียวกัน หรือในกรณีนี้ก็คือ ทั้งย่านสัญญาณเสียงเลย เป็นการสอดคล้องกับสมมติฐานของเราทุกประการ

รูปที่ 8.5 เป็นส่วนที่นำผลมาจาก local maxima แล้วพิจารณาอีกครั้ง เพื่อจะหาส่วนที่เราได้ยินจริงๆ โดยมีเงื่อนไขว่าอย่างน้อยพลังงานของสัญญาณเสียงนั้นๆ จะต้องมากกว่าพลังงานของเสียงในย่านเดียวกันไม่น้อยไปกว่า 7 dB ถ้าสอดคล้องกับเงื่อนไขนี้จะสามารถบอกได้ทันทีว่าเสียงนั้นๆ เราได้ยินแน่นอน แล้วจะถูกนำไปบันทึกเป็นรายการในส่วนของเสียงที่เราได้ยิน โดยจะมีการบันทึกทั้งดัชนี และ พลังงานของเสียงพลังงานนั้นๆ

รูปที่ 8.6 หลังจากที่ได้ในส่วนของเสียงที่เราได้ยินแล้ว เราจะต้องมาหาพลังงานของเสียงที่เราไม่ได้ยินด้วย โดยมีวิธีว่าจะต้องรวมพลังงานของเสียงในส่วนที่นอกเหนือจากค่าที่เราได้ยินแล้ว โดยจะรวมพลังงานตามช่วงความถี่ของแถบวิกฤตของพลังงานที่ไม่ใช่ส่วนที่เราได้ยิน แล้วรวมเป็น 1 พลังงานเสมือนหนึ่งว่าเราได้ยินเสียงนั้น ส่วนอินเด็กซ์ ซึ่งก็คือ ตัวบอกตำแหน่งของสัญญาณนั้นๆ จะสามารถหาได้จากค่าเฉลี่ยตามความกว้างของแถบความถี่วิกฤตตามมารตราฐาน ISO/IEC 11172-3

รูปที่ 8.7 เป็นกราฟที่แสดงเสียงที่เสมือนหนึ่งได้ยิน อันได้มาจากการรวมพลังงานเสียงในย่านความถี่วิกฤตนั้น โดยเปรียบเทียบค่าขีดเริ่มของการได้ยินของหูมนุษย์(Absolute Threshold) เห็นได้ว่าในทุกๆ แถบความถี่วิกฤต ก็จะทีส่วนที่เสมือนหนึ่งได้ยินเพียง 1 ค่าเท่านั้น

รูปที่ 8.8 เป็นกราฟที่จะมีการตัดบางดัชนีที่ไม่จำเป็นออก โดยจะสังเกตว่าพลังงานที่อยู่ต่ำกว่าขีดความถี่ขีดเริ่มของหูคนเราจะถูกตัดทิ้ง อันเนื่องมาจากว่า ค่าพลังงานนี้ยังไงเราก็ไม่ได้ยิน ส่วนเสียงที่มีพลังงานอยู่เหนือความถี่ขีดเริ่มก็จะยังคงค่านั้นไว้อยู่ ด้วยวิธีนี้ทำให้เราสามรถจัดข้อมูลที่ไม่จำเป็นบางค่าออกไป ทำให้ขนากของข้อมูลเล็กลง ส่งผลให้การลดขนาดของข้อมูลมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8.9เป็นกราฟที่เปรียบเทียบเสียงสัญญาณในส่วนที่เราได้ยินจากกราฟ tonal component กับความถี่ขีดเริ่มของกูคนเรา ณ ที่ความถี่เดียว ซึ่งก็คือ 10000 Hzก็ย่อมจะมีเพียงพลังงานเดียวเท่านั้นที่เด่นขึ้นมา

รูปที่ 8.10 เป็นกราฟที่จะมีการตัดบางดัชนีที่ไม่จำเป็นออกโดยมีเงื่อนไขใหญ่ๆ อยู่ 2 ประการ คือ อย่างที่หนึ่งจะสังเกตว่าพลังงานที่อยู่ต่ำกว่าขีดความถี่ขีดเริ่มของหูคนเราจะถูกตัดทิ้ง อันเนื่องมาจากว่า ค่าพลังงานนี้ยังไงเราก็ไม่ได้ยิน ส่วนเสียงที่มีพลังงานอยู่เหนือความถี่ขีดเริ่มก็จะยังคงค่านั้นไว้อยู่ ส่วนประการที่สองจะเป็นการมองเป็นแถบความถี่วิกฤตไป โดยถ้าพลังงานเสียงในแถบความถี่วิกฤตเดียวกันนั้นสามารถจะเกิดการบดบังกันได้ ดังนั้น ถ้าเสียงอันไหนที่ถูกบดบังก็จะสามารถตัดค่านั้นออกได้เช่นกัน เนื่องมาจากว่า ยังไงเราก็ไม่ได้ยินเสียงนั้นแน่นอน ด้วยวิธีนี้ทำให้เราสามรถจัดข้อมูลที่ไม่จำเป็นบางค่าออกไป ทำให้ขนาดของข้อมูลเล็กลง

รูปที่ 8.11เป็นกราฟที่บอกถึงค่าทั้งในส่วนของเสียงที่เราได้ยินจริงๆ และเสียงที่เราเสมือนหนึ่งได้ยิน โดยเปรียบเทียบกับความถี่ขีดเริ่มด้วยซึ่งในกรณีนี้เราได้ลดข้อมูลในส่วนของเสียงที่เสมือนจะได้ยินไปหนึ่งค่า ก็จะส่งผลให้ขนาดข้อมูลเราเล็กลงอีก ยิ่งขนาดเล็กเท่าไหร่ก็ยิ่งดี แต่ก็ต้องมีข้อแม้ว่าจะต้องไม่สูญเสียข้อมูลที่มีนัยสำคัญออกไป

รูปที่ 8.12 เป็นกราฟที่แสดงทางเดินของเส้นางการบดบัง โดยจะสามารถได้มาจากเงื่อนไขที่ว่า ระยะห่างในหน่วยของบาร์กจะต้องอยู่ในช่วงมากกว่า -3 แต่ต้องน้อยกว่า 8 ถ้าผ่านเงื่อนไขนี้ก็จะหาดัชนีการบดบัง และฟังก์ชั้นการบดบังตามลำดับ โดยจะเห็นว่า ณ ความถี่ในส่วนของเสียงที่ได้ยินก็จะมีทางเดินการบดบังเป็นของมันเอง ถ้ายิ่งสัญญาณนั้นมีพลังงานมากขึ้นเท่าไหร่ เส้นทางการบดบังก็จะยิ่งแผ่ขยายออกไปมากขึ้น และก็น่าจะสังเกตอีกว่าในทางด้านความถี่สูง ความชันของกราฟจะน้อยกว่าทางด้านความถี่ต่ำ อันเป็นผลทำให้เป็นการง่ายในการที่สัญญาณนั้นจะบดบังสัญญาณอื่นที่ความถี่สูง และก็ขอให้ดูอีกว่า ถ้ายิ่งพลังงานเสียงมากขึ้นเท่าไหร่ ความชันทางด้านความถี่สูงก็จะยิ่งลดลง ทำให้การบดบังเกิดได้ในย่านที่กว้างขึ้นกว่าเดิมเขึ้นไปอีก

รูปที่ 8.13 นอกจากเสียงที่เราได้ยินจะมีทางเดินการบดบังเป็นของมันแล้ว เสียงในส่วนที่เสมือนได้ยินก็จะมีเส้นทางการบดบังเป็นของมันเองเหมือนกันโดยจะสามารถได้มาจากเงื่อนไขที่ว่า ระยะห่างในหน่วยของบาร์กจะต้องอยู่ในช่วงมากกว่า -3 แต่ต้องน้อยกว่า 8 ถ้าผ่านเงื่อนไขนี้ก็จะหาดัชนีการบดบัง และฟังก์ชั้นการบดบังตามลำดับ โดยจะเห็นว่า ณ ความถี่ในส่วนของเสียงที่เสมือนได้ยินก็จะมีทางเดินการบดบังเป็นของมันเองเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8.14 เป็นกราฟที่แสดง Masking threshold ของเสียงที่อินพุท โดยจะสามารถหาได้มาจากเส้นทางการบดบัง โดยดูได้จากเส้นสีดำว่าขีดเริ่มของการบดบังจะได้มาจากการเชื่อมเส้นของเส้นทางการบดบังเข้าด้วยกัน เห็นได้อีกว่าพลังงานเสียงนั้นยังเลื่อนขีดเริ่มการได้ยินของหูคนเราขึ้นมา ทำให้ระดับขีดเริ่มบดบังก็จะสูงขึ้นมา อันเป็นการดีมาก ทำให้สามารถจะลดขนาดการโค้ดข้อมูลลงได้มากด้วยวิธีนี้ ซึ่งเป็นวิธีการที่ค่อนข้างจะยุ่งยากซับซ้อน แต่ทำให้ขนาดข้อมูลลดลงไปได้มาก ซึ่งทำให้มีการใช้กันอย่างแพร่หลาย ในระยะหลังต่อมา

รูปที่ 8.15 เป็นกราฟที่ขยายผลจากกราฟที่แล้วอีกครั้งminimum masking threshold เป็นการแสดงแถบความถี่บดบังโดยดูเป็นสับแบนด์ ถ้าอินพุทที่เราป้อนอยู่ในสับแบนด์ใดค่านี้ก็จะมากตามไปด้วย

รูปที่ 8.17 เป็นกราฟที่ได้มาจากการหาผลต่างของระดับพลังงานของเสียง (Sound Pressure Level) กับค่าขีดเริ่มการบดบัง โดยจะต้องหาค่าออกมาให้ได้ 32 ค่าตามจำนวนแถบความถี่วิกฤตผลลัพธ์อันนี้จะใช้เพื่อไปหา Siganl to Noise Ratio ต่อไป อันอยู่ในส่วนของการจัดเรียงบิตซึ่งมีวิธีการที่ยุ่งยากซับซ้อนมาก สังเกตว่าจะมีเพียงเส้นเดียวที่มีผลต่างของพลังงานเด่นขึ้นมามากสุด ส่วนเส้นอื่นก็อาจจะได้มาจากส่วนที่เสมือนได้ยิน ซึ่งจำนวนบิตที่จะโค้ดสัญญาณเหล่านี้ก็จะน้อยลงไปด้วยเป็นเงาตามตัว การทำแบบนี้จะไม่ยุ่งกับดัชนีความถี่ แต่จะหาพลังงานในแต่ละแถบความถี่ย่อยเป็นสำคัญ

ส่วนที่ 2 เป็นคำอธิบายอินพุทที่(0.003 ,1000 Hz), (0.002, 10000Hz) , (0.001 11000Hz)

รูปที่ 8.18ก – 8.18ง สังเกตว่าเมื่อป้อนอินพุทเข้าไปตัวกรองชนิดนี้ก็จะทำการแม็ปจากโดเมนเวลาเป็นโดเมนความถี่โดยสามารถใช้สูตรในคำอธิบายในกราฟชุดที่แล้วให้ผลออกมาว่า ทางด้านเอาพุทจะเด่นที่แบนด์ 2 ซึ่งแน่นอนจะต้องมีแอมปลิจูดและนอกจากนี้ก็ยังเด่นออกมาอีกทีในแบนด์ที่ 15 อันเป็นผลจากอินพุทเราอยู่ในช่วง 10000-11000Hz แทนที่น่าจะออกมาเป็น 2 แบนด์แต่เนื่องมันอยู่ในช่วงความถี่ใกล้เคียงกันจึงเด่นออกมาเพียงแบนด์เดียวเท่านั้น ส่วนขนาดสูงสุดก็แน่นอนที่ออกมาประมาณ 0.002(ตามค่ามากสุดของอินพุท) จึงทำให้เราทราบว่าไม่ว่าจะป้อนความถี่เป็นอย่างไรเท่าไร มันก็จะมีแบนด์ของมันเองเฉพาะตัว ไม่ไปยุ่งเกี่ยวกับแบนด์อื่น ซึ่งจะทำให้ง่ายขึ้น

เอกสารนี้อนุญาตให้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการพิจารณาสำหรับช่วงความถี่ที่เราสนใจ สำหรับส่วนอื่นที่ไม่สนใจ ก็จะสามารถละเลยไปได้บ้าง

รูปที่ 8.19 สังเกตได้ว่าจะมีบางช่วงสับแบนด์เท่านั้นที่ค่า scale factor มีค่าชัดเจน ซึ่งก็คือ แบนด์ที่ 2 และแบนด์ที่ 15,16 โดยจะนำค่าที่ได้จากการโค้ดแบบสับแบนด์ ส่วนสับแบนด์อื่นๆ ก็แน่นอนว่าจะมีค่าอยู่เล็กน้อยซึ่งก็ไม่สำคัญอะไร ค่าของขนาดก็จะเหมือนเดิม คือมีค่าใกล้เคียงกับขนาดแอมปลิจูดสัญญาณทางอินพุท

รูปที่ 8.20 เป็นกราฟที่เป็นการการโค้ดแบบทรานส์ฟอร์ม คือเป็นการแม็ปจากโดเมนเวลาเป็นโดเมนความถี่อีกแบบหนึ่ง ส่วนแบบตัวกรองความถี่ เราเรียกว่าการโค้ดแบบสับแบนด์ ด้วยวธีการนี้ก็อีกเช่นกันเราป้อนไป 3 อินพุท ก็จะต้องออกเส้นสเปกตรัมออกมาเด่น 3 ค่าเช่นกัน โดยเราจักเห็นว่านอกจากจะมีค่าที่เด่นออกมาแล้ว ยังมีบางค่าที่อยู่ในย่านใกล้ๆพอมีค่าออกมาบ้าง สำหรับขนาดของแมกนิจูดก็จะมีค่าประมาณ 0.06, 0.04 และ 0.02ตามลำดับ อันเป็นผลเนื่องาจากอินพุทที่เราป้อนเข้าไป ขนาด 0.003, 0.002 และ 0.001 ตามลำดับ ซึ่งผลออกมาก็เป็นการยืนยันว่าผลการทดลองน่าจะถูกต้อง

รูปที่ 8.21 เมื่อป้อนไป 3 อินพุท ก็ย่อมจะมีเพียงบางสับแบนด์เท่านั้นที่เด่นออกมา คือ สับแบนด์ที่ 2 , 15 และ 16

รูปที่ 8.22 แสดงพลังงานของเสียงที่ความถี่ 1000Hz, 10000Hz และ 11000Hz

รูปที่ 8.23 เป็นกราฟที่แสดงเสียงที่เราได้ยิน ซึ่งในรูปจะมีอยู่ 3 ความถี่ตรงตามที่เราป้อนเข้าไป

รูปที่ 8.24 เป็นกราฟที่บอกถึงเสียงส่วนที่เราได้ยิน และเสียงที่เสมือนได้ยิน

รูปที่ 8.25 เป็นการแสดงเสียงส่วนที่เสมือนจะได้ยินกับความถี่ขีดเริ่ม

รูปที่ 8.26 เป็นการตัดเสียงที่เสมือนหนึ่งได้ยินออกไปบางส่วนเพื่อลดขนาดข้อมูลลง

รูปที่ 8.27 เป็นการแสดงเสียงที่ได้ยิน กับความถี่ขีดเริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่8.28 เป็นการกำจัดเสียงส่วนที่ได้ยินออกไปบางค่า อันเป็นผลเนื่องมาจากเกิดการบดบังขึ้นภายในแถบความถี่วิกฤตนั้น

รูปที่ 8.29เป็นการแสดงเสียงทั้งส่วนได้ยิน และ เสียงในส่วนที่เสมือนได้ยินเทียบกับความถี่ขีดเริ่ม

รูปที่ 8.30เป็นการแสดงเส้นทางการบดบังของสัญญาณเสียงในส่วนที่เราได้ยิน

รูปที่ 8.31เป็นการแสดงเส้นทางการเดินทางการบดบังของเสียงส่วนที่เสมือนได้ยิน

รูปที่8.32เป็นกราฟที่แสดงถึงส่วนขีดเริ่มของความถี่ชุดใหม่อันเป็นผลมาจากอินพุทที่เราป้อนเข้าไป

รูปที่ 8.33 เป็นการแสดง minimum masking threshold โดยดูเป็นสับแบนด์

รูปที่8.34เป็นอัตราส่วนของพลังงานของสัญญาณเสียงเทียบกับขีดเริ่มการบดบัง โดยดูเป็นสับแบนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

รายละเอียดของชุดคำสั่งของชิป DSP 56009

(Instruction Set Detail)

Instruction Set Summary (Sheet 1 of 7)

Mnemonic	Syntax	Parallel Moves	Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
ABS	D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	—
ADC	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
ADD	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
ADDL	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	?	*
ADDR	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
AND	S,D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	—
AND(I)	#xx,D		1	2	?	?	?	?	?	?	?	?
ASL	D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	?	?
ASR	D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	0	?
BCHG	#n,X:<aa>		1 + ea	4 + mvb	?	?	?	?	?	?	?	?
	#n,X:<pp>											
	#n,X:<ea>											
	#n,Y:<aa>											
	#n,Y:<pp>											
	#n,Y:<ea>											
	#n,D											
BCLR	#n,X:<aa>		1 + ea	4 + mvb	?	?	?	?	?	?	?	?
	#n,X:<pp>											
	#n,X:<ea>											
	#n,Y:<aa>											
	#n,Y:<pp>											
	#n,Y:<ea>											
	#n,D											
BSET	#n,X:<aa>		1 + ea	4 + mvb	?	?	?	?	?	?	?	?
	#n,X:<pp>											
	#n,X:<ea>											
	#n,Y:<aa>											
	#n,Y:<pp>											
	#n,Y:<ea>											
	#n,D											

— Indicates that the bit is unaffected by the operation
* Indicates that the bit may be set according to the definition, depending on parallel move conditions
? Indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 Indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 2 of 7)

Mnemonic	Syntax	Parallel Moves	Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
BTST	#n,X:<aa>		1 + ea	4 + mvb	—	*	—	—	—	—	—	?
	#n,X:<pp>											
	#n,X:<ea>											
	#n,Y:<aa>											
	#n,Y:<pp>											
	#n,Y:<ea>											
	#n,D											
CLR	D	(parallel move)	1 + mv	2 + mv	*	*	?	?	?	?	?	—
CMP	S1,S2	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
CMPM	S1,S2	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
DEBUG			1	4	—	—	—	—	—	—	—	—
DEBUGcc			1	4	—	—	—	—	—	—	—	—
DEC	D		1	2	—	*	*	*	*	*	*	*
DIV	S,D		1	2	—	*	—	—	—	?	?	
DO	X:<ea>,expr		2	6 + mv	*	*	—	—	—	—	—	—
	X:<aa>,expr											
	Y:<ea>,expr											
	Y:<aa>,expr											
	#xxx,expr											
	S,expr											
ENDDO			1	2	—	—	—	—	—	—	—	—
EOR	S,D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	—
ILLEGAL			1	8	—	—	—	—	—	—	—	—
INC	D		1	2	—	*	*	*	*	*	*	*
Jcc	xxx		1 + ea	4 + jx	—	—	—	—	—	—	—	—
JCLR	#n,X:<ea>,xxxx		2	6 + jx	*	*	—	—	—	—	—	—
	#n,X:<aa>,xxxx											
	#n,X:<pp>,xxxx											
	#n,Y:<ea>,xxxx											
	#n,Y:<aa>,xxxx											
	#n,Y:<pp>,xxxx											
	#n,S,xxxx											

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 3 of 7)

Mnemonic	Syntax	Parallel Moves		Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
JMP	xxxx			1 + ea	4 + jx	—	—	—	—	—	—	—	—
	ea												
JScc	xxxx			1 + ea	4 + jx	—	—	—	—	—	—	—	—
	ea												
JSCLR	#n,X:<ea>,xxxx			2	6 + jx	*	*	—	—	—	—	—	—
	#n,X:<aa>,xxxx												
	#n,X:<pp>,xxxx												
	#n,Y:<ea>,xxxx												
	#n,Y:<aa>,xxxx												
	#n,Y:<pp>,xxxx												
	#n,S,xxxx												
JSET	#n,X:<ea>,xxxx			2	6 + jx	*	*	—	—	—	—	—	—
	#n,X:<aa>,xxxx												
	#n,X:<pp>,xxxx												
	#n,Y:<ea>,xxxx												
	#n,Y:<aa>,xxxx												
	#n,Y:<pp>,xxxx												
	#n,S,xxxx												
JSR	xxx			1 + ea	4 + jx	—	—	—	—	—	—	—	—
	ea												
JSSET	#n,X:<ea>,xxxx			2	6 + jx	*	*	—	—	—	—	—	—
	#n,X:<aa>,xxxx												
	#n,X:<pp>,xxxx												
	#n,Y:<ea>,xxxx												
	#n,Y:<aa>,xxxx												
	#n,Y:<pp>,xxxx												
	#n,S,xxxx												
LSL	D	(parallel move)		1 + mv	2 + mv	*	*	—	—	?	?	0	?
LSR	D	(parallel move)		1 + mv	2 + mv	*	*	—	—	?	?	0	?
LUA	<ea>,D			1	4	—	—	—	—	—	—	—	—
MAC	(±)S2,S1,D	(parallel move)		1 + mv	2 + mv	*	*	*	*	*	*	*	—
	(±)S1,S2,D	(parallel move)											

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 4 of 7)

Mnemonic	Syntax	Parallel Moves		Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
	(±)S,#n,D	(no parallel move)		1	2								
MACR	(±)S2,S1,D	(parallel move)		1 + mv	2 + mv	*	*	*	*	*	*	*	—
	(±)S1,S2,D	(parallel move)											
	(±)S,#n,D	(no parallel move)		1	2								
MOVE	S,D			1 + mv	2 + mv	*	*	—	—	—	—	—	—
No parallel data move		(.....)		mv	mv	—	—	—	—	—	—	—	—
Immediate short		(.....)#xx,D		mv	mv	—	—	—	—	—	—	—	—
data move													
Register to register		(.....)S,D		mv	mv	*	*	—	—	—	—	—	—
data move													
Address register update		(.....)ea		mv	mv	—	—	—	—	—	—	—	—
X memory data move		(.....)X:<ea>,D		mv	mv	*	*	—	—	—	—	—	—
		(.....)X:<aa>,D											
		(.....)S,X:<ea>											
		(.....)S,X:<aa>											
		(.....)#xxxxxx,D											
Register and X memory data move		(.....)X:<ea>,D1	S2,D2	mv	mv	*	*	—	—	—	—	—	—
		(.....)S1,X:<ea>	S2,D2										
		(.....)#xxxxxx,D1	S2,D2										
		(.....)A,X:<ea>	X0,A										
		(.....)B,X:<ea>	X0,B										
Y memory data move		(.....)Y:<ea>,D		mv	mv	*	*	—	—	—	—	—	—
		(.....)Y:<aa>,D											
		(.....)S,Y:<ea>											
		(.....)S,Y:<aa>											
		(.....)#xxxxxx,D											
Register and Y memory data move		(.....)S1,D1	Y:<ea>,D2	mv	mv	*	*	—	—	—	—	—	—
		(.....)S1,D1	S2,Y:<ea>										
		(.....)S1,D1	#xxxxxx,D2										
		(.....)Y0,A	A,Y:<ea>										
		(.....)Y0,B	B,Y:<ea>										

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 5 of 7)

Mnemonic	Syntax	Parallel Moves		Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
Long memory data move		{.....}L:<ea>,D		mv	mv	*	*	—	—	—	—	—	—
		{.....}L:<aa>,D											
		{.....}S,L:<ea>											
		{.....}S,L:<aa>											
XY memory data move		{.....}X:<eax>,D1	Y:<eay>,D2	mv	mv	*	*	—	—	—	—	—	—
		{.....}X:<eax>,D1	S2,Y:<eay>										
		{.....}S1,X:<eax>	Y:<eay>,D2										
		{.....}S1,X:<eax>	S2,Y:<eay>										
MOVE(C)	X:<ea>,D1			1 + ea	2 + mvc	?	?	?	?	?	?	?	?
	X:<aa>,D1												
	S1,X:<ea>												
	S1,X:<aa>												
	Y:<ea>,D1												
	Y:<aa>,D1												
	S1,Y:<ea>												
	S1,Y:<aa>												
	S1,D2												
	S2,D1												
	#xxxx,D1												
	#xx,D1												
MOVE(M)	P:<ea>,D			1 + ea	2 + mvm	?	?	?	?	?	?	?	?
	S,P:<ea>												
	S,P:<aa>												
	P:<aa>,D												
MOVE(P)	X:<pp>,D			1 + ea	2 + mvp	?	?	?	?	?	?	?	?
	X:<pp>,X:<ea>												
	X:<pp>,Y:<ea>												
	X:<pp>,P:<ea>												
	S,X:<pp>												
	#xxxxxx,X:<pp>												
	X:<ea>,X:<pp>												

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 6 of 7)

Mnemonic	Syntax	Parallel Moves	Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
MOVE(P) (continued)	Y:<ea>,X:<pp>											
	P:<ea>,X:<pp>											
	Y:<pp>,D											
	Y:<pp>,X:<ea>											
	Y:<pp>,Y:<ea>											
	Y:<pp>,P:<ea>											
	S,Y:<pp>											
	#xxxxxx,Y:<pp>											
	X:<ea>,Y:<pp>											
	Y:<ea>,Y:<pp>											
	P:<ea>,Y:<pp>											
MPY	(±)S2,S1,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	—
	(±)S1,S2,D	(parallel move)										
	(±)S,#n,D	(no parallel move)	1	2								
MPYR	(±)S2,S1,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	—
	(±)S1,S2,D	(parallel move)										
	(±)S,#n,D	(no parallel move)	1	2								
NEG	D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	—
NOP			1	2	—	—	—	—	—	—	—	—
NORM	Rn,D		1	2	—	*	*	*	*	*	?	—
NOT	D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	—
OR	S,D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	—
ORI	#xx,D		1	2	?	?	?	?	?	?	?	?
REP	X:<ea>		1	4 + mv	?	?	—	—	—	—	—	—
	X:<aa>											
	Y:<ea>											
	Y:<aa>											
	S											
	#xxx											
RESET			1	4	—	—	—	—	—	—	—	—
RND	D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	—

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Sheet 7 of 7)

Mnemonic	Syntax	Parallel Moves	Instruction Program Words	Osc. Clock Cycles	Status Request Bits: S	L	E	U	N	Z	V	C
ROL	D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	?
ROR	D	(parallel move)	1 + mv	2 + mv	*	*	—	—	?	?	0	?
RTI			1	4 + rx	?	?	?	?	?	?	?	?
RTS			1	4 + rx	—	—	—	—	—	—	—	—
SBC	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
STOP			1	n/a	—	—	—	—	—	—	—	—
SUB	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
SUBL	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	?	*
SUBR	S,D	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	*	*
SWI			1	8	—	—	—	—	—	—	—	—
Tcc	S1,D1		1	2	—	—	—	—	—	—	—	—
	S1,D1 S2,D2											
TFR	S,D	(parallel move)	1 + mv	2 + mv	*	*	—	—	—	—	—	—
TST	S	(parallel move)	1 + mv	2 + mv	*	*	*	*	*	*	0	—
WAIT			1	n/a	—	—	—	—	—	—	—	—

— indicates that the bit is unaffected by the operation
* indicates that the bit may be set according to the definition, depending on parallel move conditions
? indicates that the bit is set according to a special definition. See the instruction descriptions in Appendix A of the *DSP56000 Family Manual (DSP56KFAMUM/AD)*
0 indicates that the bit is cleared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมการเข้ารหัสในส่วนของ Filterbank, FFT

Filterbank test routine

```
;include files : flttest.asm, sign10k.asm, mik.asm, coef.asm
;See detail:each_band.txt
        include 'sign10k'
        include 'mik'
        include 'coef'
        include 'scf'
        include 'log10'
;=============================
;               Equate
;=============================
main                    equ     $40
flag                    equ     $1f
frame                   equ     1
subband                 equ     32
subband_sample          equ     12
input_buffer            equ     0
vectorz_buffer          equ     $200
yi_buffer               equ     $400
output_buffer           equ     $600
scale_buffer            equ     $800
data                    equ     $a00
scale_factor_table      equ     $a00
filter_coef             equ     $2000
mik                     equ     $2200
zero                    equ     0
yi_points               equ     64
points                  equ     512
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pcoef                   equ     $1
value_10                equ     0.625
value_20                equ     0.625
max_power_filter        equ     $830
times                   equ     25
minus                   equ     $ffffff

lower_05        equ     2
neg             equ     3
unnorm          equ     4
extent          equ     5
limit           equ     6

        org     y:pcoef
        dc      0.9981958
        dc      -0.3372223
        dc      -0.6626105
        dc      0.301029996

save    ds      21

        org     p:$0
        reset
        jmp     main

        org     p:main

;=============================================
;                Main Program
;=============================================
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move    #zero,x0
        move    x0,y0
        move    #zero,r0
        move    r0,r1

        rep     #1024
        move    x0,x:(r0)+

;Pointer initialization

        move    #subband_sample,r2      ;Loop counter for Si calculation
        move    #subband,r3             ;R3 as memory pointer
                                        ;(for store calculated data)
        move    #output_buffer,r7       ;Lower boundary of MUDULO(K=9)
        move    #480,r5
        move    #data,r0

start
        nop
        do      #subband_sample,_end_auto
        nop
        jsr     get_new_input

;Polyphase filterbank

        jsr     vectorz
        jsr     partial_calc
        jsr     filter_output
        jsr     shift
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jsset   #frame,y:flag,scale_factor
        nop
_end_auto
        jmp     start

;---------------------------
;   Get input sample
;---------------------------
get_new_input
        nop
        do      #subband,end_get
        move    x:(r0)+,x0
        move    x0,x:(r5)+
end_get
        move    #480,r5
        rts
;-------------------------------------------
;           Vector Z calculation
;-------------------------------------------
; Calculate vector Z from formular Z[i]=C[i]*X[511-i]
; Coefficient's start at p:$2000

vectorz
        move    r0,y:save
        move    r1,y:save+1
        move    r2,y:save+2
        move    #511,r0                 ;RO is raw data pointer
        move    #points,r1              ;R1 is vector Z stored
                                        ;Memory pointer
        move    #filter_coef,r2         ;R2 is cooeficient pointer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        do      #points,end_z
        move    x:(r0)-,x0
        movem   p:(r2)+,x1
        mpy     x0,x1,a

        move    a,x:(r1)+           ;Move to X memory by auto limiting
end_z
        move    y:save,r0
        move    Y:save+1,r1
        move    Y:save+2,r2
        rts
;---------------------------------
;     Partial calculation
;---------------------------------
;For i=0 to 63 DO Yi=SUM[Z(i+64j)] for J=0 to 7
;For example:Y0=Z(0+0)+Z(0+64)+Z(0+128)+...+Z(0+448)
;         :Y1=Z(1+0)+Z(1+64)+Z(1+128)+...+Z(1+448)
;                    :
;         Y63=Z(63+0)+Z(63+64)+Z(63+128)+...+Z(63+448)
;We use : r0 as i(point at vector Z memory)
;     : r1 as 1024(store Yi)
;     : r2 as Update n0(add 64 to n0)
;     : n0 as 64j
;     : n2 as Constant to update r2(64)
;---Left channel---

partial_calc
        move    r0,y:save
        move    r1,y:save+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move    r2,y:save+2
        move    n0,y:save+8
        move    n2,y:save+10

        move  #points,r0
        move  #zero,n0
        move  n0,r2
        move  #64,n2
        clr   a     #yi_buffer,r1

        do    #yi_points,end_partial
        nop
        do    #8,yil
        move  x:(r0+n0),b
        add   b,a
        lua   (r2)+n2,n0
        move    n0,r2
yil
        move  a,x:(r1)+   ;Save Yi into memory(start @1024)
        clr   a     (r0)+    ;Point at next vector Z
        move  #zero,r2        ;Clear update register
        move  r2,n0           ;Clear offset register
end_partial

        move    y:save,r0
        move    y:save+1,r1
        move    y:save+2,r2
        move    y:save+8,n0
        move    y:save+10,n2
        rts
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;----------------------------------------
;  Calculate 32 sample by matrixing
;----------------------------------------
;For i=0 to 31 do Si=SUM[Mik*Yk] for K=0 to 63
;Mik=cos[(2*i+1)(k-16)Pi/64]
;Example:S0=M(0,0)*Y(0)+M(0,1)*Y(1)+...+M(0,63)*Y(63)
;          S1=M(1,0)*Y(0)+M(1,1)*Y(1)+...+M(1,63)*Y(63)
;                                  :
;         S31=M(31,0)*Y(0)+M(31,1)*Y(1)+...+M(31,63)*Y(63)
;Each Si is a result of each subband(S0 for subband 0,...,S31 for subband 31)
;Use look up table(Mik)start at address $2200 in program memory.
;         We use : r0 as table pointer
;                : r1 as Yi pointer
;                : r2 as Countdown loop
;                : r7 as Result(Si)pointer
;                : n7 as Offset for store data
;                : m7 as Modulo size(384)
filter_output
        move    r0,y:save
        move    r1,y:save+1
        move    #mik,r0
        move    #yi_buffer,r1
        move    #subband_sample,n7
        move    #383,m7                 ;Modulo size=384

        do    #subband,end_keep_si
        clr     a
        do      #yi_points,end_get_si
        move    x:(r1)+,x1      ;Get Yi from memory
        movem p:(r0)+,x0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mac     x0,x1,a
end_get_si                      ;Get one of Si value

        move    a,x:(r7)+n7     ;Store data in buffer
        move    #yi_buffer,r1   ;Point at Y0 again
end_keep_si

        abs  a     (r7)+
        clr  a     (r2)-  ;Countdown loop
        move    r2,y0
        cmp     y0,a
;If R2=0 then jump to subroutine 'frame_full'
;else return to main program.
        jseq    frame_full
        move    y:save,r0
        move    y:save+1,r1
        rts
;---------------------------------------------------------
;Subroutine for moving data of subband to memory
;---------------------------------------- -- ---------------
frame_full
        move   #output_buffer,r4
        do     #subband_sample,end_collect
        move   x:(r4)+,x0
        move   x0,y:(r3)+
end_collect
        move    #subband_sample,r2
        move    #output_buffer,r7
        bset    #frame,y:flag
        rts
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------
; Start shift old data by X[i]=X[i-32]for i=32 to 511
;-------------------------------------------------
;For example:move X[511] to X[479]and then,put new data into X[511].
shift
        move    r0,y:save
        move    r1,y:save+1
        move    #zero,r0                ;Destination pointer
        move    #subband,r1             ;Source pointer

        do      #480,end1
        move    x:(r1)+,x0              ;Move X data
        move    x0,x:(r0)+
end1
        move    y:save,r0
        move    y:save+1,r1
        rts
;At the end of this subroutine,
;we're ready to receive new 32 input sample.
scale_factor
        bclr    #frame,y:flag
        move    r0,y:save
        move    r1,y:save+1
        move    r2,y:save+2
        move    r3,y:save+3
        move    r4,y:save+4
        move    r5,y:save+5

        move    #output_buffer,r5
        move    r5,r1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move    #scale_buffer,r2
        move    #scale_factor_table,r3
        move    #max_power_filter,r4

        do      #subband,end_max
        move    x:(r5)+,a

        do      #11,end_compare
        move    x:(r5)+,x0
        cmpm    x0,a

        jpl     no_change
        move    x0,a
no_change
        nop

end_compare

        move    a,y:(r1)+
        move    y:(r3)+,x1

        do      #59,loop
        move    x1,x0
        cmpm    x1,a    y:(r3)+,x1

        jpl     comp_scale
        move    x0,x:(r2)+
        enddo
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
comp_scale
        nop
loop

        move    x0,a            ;move scale factor to acc. a
        move    #value_10,b     ;value_10 = 0.625
        rep     #4
        asl     b               ;b=10

        rep     #15
        asl     a               ;multiply scale factor by 32,768
        sub     b,a
        jmi     mark_for_minus

        move    r1,y:save+9
        move    r2,y:save+10

        log10a  pcoef

        move    y:save+9,r1
        move    y:save+10,r2

        move    #value_20,y1    ;value_20 = 0.625
        move    #1,r0

        bclr    #limit,sr
        move    a,x0            ;log10(a)>=1?
        jlc     skipa
        rep     #times
        norm    r0,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
skipa
        move    a,x0
        mpy     x0,y1,a
        rep     #5
        asl     a               ;multiply a by 20

        bclr    #limit,sr
        move    a,x0            ;20*log10(a)>=1?
        jlc     skipb
        rep     #times
        norm    r0,a
skipb
        move    a,x:(r4)  ;store max power of filter
        move    r0,y:(r4)+
        jmp     continu

mark_for_minus
        move    #minus,y0
        move    y0,x:(r4)+

continu
        move    #scale_factor_table,r3
end_max
        move    y:save,r0
        movc    y:save+1,r1
        move    y:save+2,r2
        move    y:save+3,r3
        move    y:save+4,r4
        move    y:save+5,r5
        rts
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FFT test routine

```
        include 'sincos'
        include 'fft512'
        include 'exhann'
        include 'sign10k'
        include 'log10'

reset           equ     0
start           equ     $40
points          equ     512
coef            equ     $2000
pcoef           equ     $1
hann            equ     $2a00
input           equ     $a00
idata           equ     $200
odata           equ     $400
mean            equ     $600
power           equ     $800
max_value       equ     $20
offset_96db     equ     $40
max_power       equ     $45
max_value_96db          equ     $50
pi              equ     3.141592654
freq            equ     2.0*pi/@cvf(points)
fft_scale       equ     0.625
times           equ     25
value_150       equ     0.586661716
val_20log512    equ     0.846646864


flag            equ     $0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
newloop         equ     0
lower_05        equ     2
first_pass      equ     3
zero            equ     2
neg             equ     3
unnorm          equ     4
extent          equ     5
limit           equ     6

        sincos  points,coef     ;table start at $2000 in
                                ;X and Y memory
        org     y:pcoef
        dc      0.9981958
        dc      -0.3372223
        dc      -0.6626105
        dc      0.301029996

buffer  ds      21

        org     p:reset
        jmp     start

        org     p:start

        move    #0,x0
        move    #idata,r5       ;r0 updated by r5
        move    r5,r0

        rep     #64
        move    x0,x:(r0)+      ;zero padding for first calc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        bclr    #newloop,y:flag
        move    #$2a40,r1          ;hann's windows pointer(at 64th)
        move    #input,r2      ;input data pointer
        move    x0,r6
        move    #128,n2
        move    #64,n6
loop
        nop
        jclr    #newloop,y:flag,loop_1

        move    #0,x0
        do      #512,_clear
        move    x0,x:(r0)
        move    x0,y:(r0)+
_clear
        move    r5,r0
        do      #64,_last64
        movem   p:(r1)+,y0
        move    x:(r2)+,y1
        mpy     y0,y1,a
        move    a,x:(r0)+
_last64
        nop

loop_1
        nop

;hann windows
        do      #83,end_normal1
        movem   p:(r1)+,y0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move    x:(r2)+,y1
        mpy     y0,y1,b
        move    b,x:(r0)+
end_normal1
        nop
        do      #219,end_ex
        move    x:(r2)+,y0
        move    y0,a
        movem   p:(r1)+,y1
        mac     y0,y1,a
        clr     a       a,x:(r0)+
end_ex
        nop
        do      #146,end_normal2
        movem   p:(r1)+,y0
        move    x:(r2)+,y1
        mpy     y0,y1,a
        move    a,x:(r0)+
end_normal2
        nop
        move    r0,y:buffer     ;save pointer
        move    r1,y:buffer+1
        move    r2,y:buffer+2
        move    r4,y:buffer+4
        move    r5,y:buffer+5
        move    n6,y:buffer+14

        fft512  points,idata,odata,coef
;In this section,find the power of spectrum from FFT
;calculation : pwr=10*log10[(x^2 +y^2)]-20*log10[512]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
move    #odata,r3
move    #mean,r4
move    #max_value,r5
move    #$ffff,m4
move    m4,m5

do      #32,end_max
bclr    #first_pass,y:flag

do      #8,end_max_subband

move    x:(r3),x0
move    y:(r3)+,y0
mpy     x0,x0,a
mac     y0,y0,a
move    a,x:(r4)+       ;mean square calculation

move    b0,y:buffer+16  ;store value in acc b
move    b1,y:buffer+17
move    b2,y:buffer+18

log10a  pcoef

move    y:buffer+16,b0  ;recall value in acc b
move    y:buffer+17,b1
move    y:buffer+18,b2

move    #1,r0
bclr    #limit,sr
move    a,x0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jlc     skip4
        rep     #20
        norm    r0,a
skip4
        move    a,x0
        move    #fft_scale,x1     ;fft_scale = 0.625
        mpy     x0,x1,a

        rep     r0
        asl     a
        asr     a                 ;restore value in a by multiply by 2(r0 times)

        rep     #4                ;multiply by 16 (0.625*16=10)
        asl     a                 ;this value is a power

        jset    #first_pass,y:flag,skip5
        tfr     a,b
        bset    #first_pass,y:flag
skip5
        cmp     a,b
        tmi     a,b
end_max_subband

        move    #1,r0
        bclr    #limit,sr
        move    b,x0
        jlc     do_nothing
        rep     #times
        norm    r0,b
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
do_nothing
        move    r0,y:(r5)
        move    b,x:(r5)+
end_max

        move    #max_value,r5
        nop
        move    x:(r5),a
        move    y:(r5)+,r0

        rep     r0
        asl     a
        asr     a

        do      #31,end_max_power
        move    x:(r5),b
        move    y:(r5)+,r0
        rep     r0
        asl     b
        asr     b
        bclr    #neg,sr
        cmp     b,a
        tmi     b,a
        nop
end_max_power

        move    #offset_96db,r4
        move    #value_150,b    ;value_150=0.586661716
        rep     #8
        asl     b               ;0.586661716*2^8=150.1853992
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sub     a,b
        move    b0,x:(r4)+
        move    b1,x:(r4)+
        move    b2,x:(r4)       ;store 150.1853992-(max power of 32 subband)

        move    #max_power,r5
        move    #1,r0
        bclr    #limit,sr
        move    a,x0
        jlc     skip6
        rep     #times
        norm    r0,a
skip6
        move    a,x:(r5)
        move    r0,y:(r5)

        move    #val_20log512,a
        rep     #6
        asl     a               ; a = 20*log10(512) = 54.1853992

        sub     a,b             ; b = 150.1853992-(max power of 32 subband)-20*log10(512)

        move    #max_value,r5
        move    #max_value_96db,r4

        do      #32,end_offset_96db
        move    x:(r5),a
        move    y:(r5)+,r0
        rep     r0
        asl     a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        asr     a
        add     b,a

        move    #1,r1
        bclr    #limit,sr
        move    a,x0
        jlc     skip7
        rep     #times
        norm    r1,a
skip7
        move    a,x:(r4)
        move    r1,y:(r4)+
end_offset_96db

        move    y:buffer+14,n6  ;load pointer
        move    y:buffer+5,r5
        move    y:buffer+4,r4
        move    y:buffer+2,r2
        move    y:buffer+1,r1
        move    y:buffer,r0

        move    (r2)-n2         ;decrease value in register r2 by 128
        move    r5,r0           ;set pointer to get last 64
                                ;sample of n-1 th frame
        move    #$2a00,r1       ;and windowing
        bset    #newloop,y:flag

        jmp     loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Davis Yen Pan, "Digital Audio Compression", Digital Technical Journal Vol. 5 No. 2, Spring 1993
2. A. Oppenheim and Schafer, *Discrete Time Signal Processing* (Englewood Cliffs, NJ: Prentice Hall, 1989)
3. K. Pohlman, *Principles of Digital Audio* (Indianapolis, IN: Howard W. Sams and Co., 1989)
4. Motorola Inc., *DSP56000/DSP56001 Digital Signal Processor User's Manual*
5. Motorola Inc., *DSP56000 Digital Signal Processor Family Manual*
6. Motorola Inc., *DSP56009 Digital Signal Processor User's Manual*
7. Motorola Inc., *DSP56009EVM User's Manual*
8. John Lane and Garth Hillman, *Implementing IIR / FIR Filters with Motorola's DSP56000/DSP56001*, Motorola Inc., 1991
9. Keshab K. Parhi, Takao Nishitani, *Digital Signal Processing for Multimedia, Marcel Dekker*, Inc. USA 1999
10. ศ. ดร. วัลลภ สุระกำพลธร, *การประมวลผลสัญญาณเชิงเลข*, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ตุลาคม พ.ศ. 2533
11. ISO/IEC 11172-3 International Standard, 1993
12. Kent Salomonsen, Sten Sogaard, Eddie Proft Larsen, "Design and Implementation of an MPEG/Audio Layer III Bitstream Processor", Department of Communication Technology, Aalborg University, 1997
13. Emmanuel C. Ifeachor, Barnie W. Jervis, Digital Signal Processing A Practical Approach, Addison-Wesley Publishers Ltd., 1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบแก้คำผิด

- รูปที่ 8.1ก-8.17 หน้า 79-90 แก้จาก “A=0.003 เป็น A=0.002”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้