

การบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ สำหรับการกระจายเสียง
ระบบเอฟเอ็ม โดยใช้ DSP TMS320C6713

นางสาวจุฑาทพร มากอยู่

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2549

**MULTIBAND DYNAMIC RANGE COMPRESSION FOR
FM BROADCASTING USING DSP TMS320C6713**

Jutaporn Markyou

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Telecommunication Engineering**

Suranaree University of Technology

Academic Year 2006

การบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ สำหรับการกระจายเสียงระบบเอฟเอ็ม
โดยใช้ DSP TMS320C6713

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยรับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(อาจารย์ ดร.ชาญชัย ทองโสภิต)

ประธานกรรมการ

(อาจารย์ ดร.รังสรรค์ ทองทา)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อาจารย์ ดร.ชุติมา พรหมมาก)

กรรมการ

(รศ. ดร.เสาวณี รัตนพานิช)

รองอธิการบดีฝ่ายวิชาการ

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

จุฑาทพร มากอยู่ : การบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ สำหรับการกระจายเสียงระบบเอฟเอ็ม โดยใช้ DSP TMS320C6713 (MULTIBAND DYNAMIC RANGE COMPRESSION FOR FM BROADCASTING USING DSP TMS320C6713)

อาจารย์ที่ปรึกษา : อาจารย์ ดร.รังสรรค์ ทองทา, 70 หน้า

ในงานวิจัยนี้ มีวัตถุประสงค์เพื่อทำการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ (multi-band dynamic range compression) สำหรับสัญญาณเสียง ก่อนทำการมอดูเลชันแบบเอฟเอ็ม เพื่อป้องกันการเกิดโอเวอร์มอดูเลชัน (overmodulation) ซึ่งจะส่งผลให้เกิดเสียงแตกพร่าไม่เป็นที่น่าฟังเป็นอย่างยิ่ง เทคนิคการแบ่งสัญญาณเสียงออกเป็นความถี่ 3 แบนด์นี้ มีข้อได้เปรียบกว่าการบีบอัดสัญญาณพัลส์พลวัตแบบแบนด์เดียว (single-band dynamic range compression) เนื่องจากโดยปกติแล้วสัญญาณเสียงที่มีความดังขึ้นมานั้น จะเป็นเสียงในย่านความถี่ใดความถี่หนึ่ง ดังนั้นการบีบอัดสัญญาณหรืออาจจะกล่าวอีกนัยหนึ่งคือ การลดอัตราขยายเฉพาะในย่านความถี่ที่มีปัญหานั้นย่อมไม่ทำให้อ่านความถี่อื่นถูกรบกวนไปด้วย เป็นผลให้เสียงที่ได้ในทุกๆ ย่านความถี่ มีระดับความดังที่ไม่เปลี่ยนแปลงมาก ซึ่งเป็นคุณลักษณะเสียงที่เป็นที่ต้องการเพื่อรักษาค่าดัชนีการเบี่ยงเบนให้คงที่ สำหรับการมอดูเลตในระบบเอฟเอ็ม

โดยปกติแล้วการทำการบีบอัดสัญญาณจะมีการหน่วงเวลาของเวลาเข้ากระทำ (attack time) และเวลาเลิกกระทำ (release time) ที่แตกต่างกัน และแตกต่างกันในแต่ละย่านความถี่ ทั้งนี้เพื่อไม่ให้ผู้ฟังรู้สึกถึงการกระทำที่เกิดขึ้น ที่เนื่องมาจากกระบวนการบีบอัดสัญญาณ และด้วยเหตุผลเดียวกันนี้ การแบ่งย่านความถี่มักจะทำให้มีการเหลื่อมกันด้วย เพื่อให้ระบบมีเสถียรภาพ ราคาถูกและใช้งานง่าย การประมวลสัญญาณทั้งหมดจะกระทำโดยใช้ DSP TMS320C6713 ของบริษัท Texas Instrument ซึ่งทำการประมวลสัญญาณดิจิทัลแบบจุดลอย (floating – point) และประมวลสัญญาณแบบเวลาจริง (real time) ที่ความเร็ว 255 MHz, ROM 512 Kbyte, RAM 256 Kbyte

JUTAPORN MAKYOU : MULTIBAND DYNAMIC RANGE

COMPRESSION FOR BROADCASTING USING DSP TMS320C6713.

THESIS ADVISOR : RANGSAN TONGTA, Ph.D. 70 PP.

MULTIBAND DYNAMIC RANGE COMPRESSION FOR BROADCASTING USING
DSP TMS320C6713

FM broadcasting has many advantages over AM broadcasting, e.g. audio quality, noise immunity. Therefore, there are a lot of new FM-radio stations built to service listeners. There are FM stations using carrier frequencies just right next to each others, 250 kHz in Thailand's regulation. Interference between close-carrier frequencies is more difficult to handle than those far-apart frequencies.

In order to prevent carrier frequency to deviate more than available bandwidth, audio signals must be kept within 15 kHz band and under a level that produces the maximum frequency modulation. Keeping the audio signals to be under 15 kHz can be accomplished by using some kind of lowpass filters (LPF). However, to maintain a proper level of audio signal requires more attention since the signal at different bands represent different modulation index.

In this thesis, the audio signal is divided into 3 subbands then process separately before recombine to produce an output signal. Each band has identical block diagrams, but different setting. The low frequency band has cutoff frequency at 800 Hz, 820 and 3000 Hz of mid frequency band and 3200 Hz for high frequency band.

To prevent overmodulation, each band has a level monitor follow by an adaptive dynamic gain adjusting. If the signal goes above a threshold level, its amplitude will be gradually compressed within amount of time called “attack time”. In the other hand, if the signal goes from above to under the threshold level, the signal will be gradually released from compression within amount of time called “release time”. The attack time and release time are required to allow the audient to hear the dynamic range of the sound, but fast enough to prevent the overmodulation.

The results show that the bandwidth of the FM spectrum is 186 kHz when using compressed audio signals, compare to 240 kHz when no compression, while the quality of the audio signals is acceptable. The results also show that the attack time and release time of each band are follow the setting. Since the high frequency band will produce wider bandwidth in the modulating FM spectrum, therefore the attack time and the release time in this band must be smaller

The entire audio processing are done in real time using the TMS320C6713 DSK, by Texas instrument, operate at 255 MHz clock speed with 512 kbytes of ROM and 256 kbytes of RAM.

School of Telecommunication Engineering Student's Signature _____

Academic Year 2007 Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างดียิ่ง ทั้งด้านวิชาการ และด้านการดำเนินงานวิจัย จากบุคคลและกลุ่มบุคคลต่าง ๆ ได้แก่

อาจารย์ ดร.รังสรรค์ ทองทา หัวหน้าสาขาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้โอกาสทางการศึกษา ให้คำแนะนำปรึกษา ช่วยแก้ปัญหา และให้กำลังใจแก่ผู้วิจัยมาโดยตลอด รวมทั้งช่วยตรวจทาน และแก้ไขวิทยานิพนธ์เล่มนี้จนเสร็จสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.รังสรรค์ วงศ์สรรค์ ผู้ช่วยอธิการบดี มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้โอกาสในการศึกษา คำปรึกษาด้านวิชาการ และให้กำลังใจแก่ผู้วิจัยมาโดยตลอด

อาจารย์ ดร.วิภาวี หัตถกรรม อาจารย์ ดร.ชุตินา พรหมมาก และ อาจารย์ ดร.ชาญชัย ทองโสภณ อาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี ที่กรุณาให้คำปรึกษาด้านวิชาการ และให้กำลังใจมาโดยตลอด

ขอขอบคุณ อาจารย์อุษา คงเมือง อาจารย์ประจำแผนกวิชาอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคล อีสาน และอาจารย์บุญเสริม มหัทธนกุล อาจารย์ประจำแผนกอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสุรนารี ที่ให้กำลังใจมาโดยตลอด

ขอขอบคุณ คุณประพล จาระตะคุ วิศวกรประจำศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ช่วยอำนวยความสะดวกทางด้านเครื่องมือและอุปกรณ์ ขอขอบคุณ พี่น้องบัณฑิตศึกษาทุกท่าน โดยเฉพาะอย่างยิ่ง คุณปิณฑงศ์ ชาญโลหะ คุณวันวิสาข์ ไทยวิโรจน์ และ คุณวิภาดา นฤพิพัฒน์ ที่ให้คำปรึกษาด้านวิชาการ และให้กำลังใจมาโดยตลอด

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์เล่มนี้ ผู้วิจัยขอมอบให้กับบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนคุณครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้วิจัยตลอดมา จนทำให้ประสบความสำเร็จในชีวิต

จุฑาทพร มากอยู่

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ญ
คำอธิบายสัญลักษณ์และคำย่อ.....	ฐ

บทที่

1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหาการวิจัย.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 สมมุติฐานของการวิจัย.....	2
1.4 ขอบเขตของการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 การจัดรูปเล่มวิทยานิพนธ์.....	3
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 กล่าวนำ.....	5
2.2 ทฤษฎีโมดูลเตแบบเอฟเอ็ม.....	5
2.2.1 เปอร์เซ็นต์ของการมอดูเลต.....	7
2.2.2 ครรชนีการมอดูเลต.....	8
2.2.3 อัตราส่วนการเบี่ยงเบน.....	8
2.2.4 ไซค์แบนด์เอฟเอ็ม.....	9
2.2.5 แบนด์วิดท์ของสัญญาณเอฟเอ็ม.....	10
2.3 ทฤษฎีระบบประสาทเกี่ยวกับการได้ยิน.....	11
2.3.1 ทฤษฎีเกี่ยวกับกายภาพของมนุษย์.....	11

สารบัญ (ต่อ)

หน้า

2.3.2	ความดังของเสียง.....	13
2.4	การประมวลผลสัญญาณ.....	14
2.4.1	โดเมน.....	14
2.4.2	โครงสร้างของระบบประมวลผลสัญญาณทางดิจิทัล.....	15
2.4.3	ตัวประมวลผลทางดิจิทัล.....	15
2.4.4	สัญญาณและระบบ.....	15
2.5	ทฤษฎีการออกแบบตัวกรอง.....	17
2.5.1	ตัวกรองดิจิทัลแบบเอฟไออาร์.....	18
2.5.2	คุณสมบัติลักษณะตัวกรองเอฟไออาร์.....	23
2.5.3	วิธีการใช้หน้าต่าง.....	24
2.5.4	หน้าต่างที่น่าสนใจ.....	29
2.5.5	ตัวกรองดิจิทัลแบบไอไออาร์.....	29
2.6	ทฤษฎีการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์.....	34
2.6.1	กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัต.....	34
2.6.2	คุณสมบัติพัลส์พลวัต.....	34
2.6.3	คุณสมบัติทางสถิติ.....	36
2.6.4	กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์.....	36
2.7	สรุป.....	38
3	การออกแบบและการใช้งาน.....	39
3.1	กล่าวนำ.....	39
3.2	โครงสร้าง TMS320C6713.....	39
3.2.1	TMS320C6713 DSK.....	39
3.2.2	สถาปัตยกรรมของ TMS320C6713.....	39
3.2.3	หน่วยความจำฟังก์ชัน.....	41
3.2.4	รีจิสเตอร์.....	41
3.2.5	หน่วยความจำโดยตรง DMA.....	42

สารบัญ (ต่อ)

หน้า

3.2.6	EMIF (External Memory Interface)	43
3.2.7	McBSP (Multichannel Buffer Serial Port)	43
3.2.8	วงจรถัดเวลา (Timer).....	44
3.2.9	อินเทอร์รัพท์พอร์ต โฮสต์ (Host Port Interface).....	44
3.2.10	อินเทอร์รัพท์ (Interrupts).....	44
3.2.11	รูปลักษณะบูต (Boot Configuration).....	46
3.2.12	บอร์ด DSK	46
3.3	การออกแบบประมวลผลแบบเวลาจริง.....	47
3.3.1	DSP/BIOS.....	47
3.3.2	มาตรฐานอัลกอริทึม.....	48
3.3.3	ออกแบบการทำงาน.....	50
3.3.4	หลักการงานเมื่อนำมาใช้กับDSP/BIOS.....	50
3.4	สรุป.....	51
4	ผลการทดลอง.....	52
4.1	กล่าวนำ.....	52
4.2	การออกแบบตัวกรอง.....	52
4.3	ผลการทดลองผลตอบสนองของตัวกรอง.....	53
4.4	ผลการทดลองช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ.....	58
4.5	ผลการทดลองบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์	63
4.6	ผลการทดลองวัดสเปกตรัมการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์	65
4.7	สรุป.....	67
5	บทสรุปและข้อเสนอแนะ.....	69
5.1	สรุปผลการวิจัย.....	69
5.2	ข้อเสนอแนะและแนวทางการพัฒนา.....	70

รายการอ้างอิง.....	71
--------------------	----

สารบัญ (ต่อ)

หน้า

ภาคผนวก	
ภาคผนวก ก โปรแกรม.....	73
ภาคผนวก ข บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา.....	85
ประวัติผู้เขียน.....	87

สารบัญตาราง

ตารางที่	หน้า
2.1 ความถี่เสียงที่นำมาทดสอบมีค่าต่าง ๆ กัน.....	10
2.2 หน้าต่างแบบต่าง ๆ กัน และค่าพารามิเตอร์ที่สำคัญ.....	29
3.1 แฟ้มรีจิสเตอร์คู่.....	42
3.2 บริการการอินเทอร์เน็ต.....	45
3.3 ฟังก์ชันการทำงาน	49

สารบัญรูป

รูปที่	หน้า
1.1 การเกิดโอเวอร์มอดูเลชัน.....	1
2.1 คลื่นพาหะ.....	6
2.2 สัญญาณมอดูเลต.....	6
2.3 มอดูเลตแบบเอฟเอ็ม.....	7
2.4 มอดูเลตทางเฟส.....	7
2.5 กราฟแอมพลิจูดของคลื่นพาหะและไซด์แบนด์ระบบเอฟเอ็ม.....	9
2.6 โครงสร้างของมนุษย์.....	11
2.7 ส่วนประกอบของคลอเคลีย.....	12
2.8 โดอะแกรมของเบสิคัลแมมเบรน ตำแหน่งผลตอบสนองคลื่นเสียง ความถี่ต่าง ๆ (Hz).....	13
2.9 ความดังของเสียงเท่ากับความดังที่มนุษย์ได้ยิน.....	14
2.10 ระบบการประมวลผลสัญญาณดิจิทัล.....	15
2.11 สัญญาณที่มีค่าต่อเนื่องทางเวลา.....	16
2.12 สัญญาณที่มีค่า дискретทางเวลา.....	16
2.13 แผนภาพบล็อกการทำงานของตัวกรองดิจิทัลพื้นฐาน.....	17
2.14 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่หนึ่ง.....	19
2.15 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่สอง.....	19
2.16 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบคาสเคด.....	20
2.17 ตัวกรองเอฟไออาร์ M เท่ากับ 6.....	22
2.18 ตัวกรองเอฟไออาร์ M เท่ากับ 7.....	22
2.19 การกำหนดคุณลักษณะของตัวกรองผ่านต่ำ.....	23
2.20 ตัวกรองในอุดมคติ.....	25
2.21 ฟังก์ชันถ่ายโอนของตัวกรองผ่านต่ำ.....	
26	
2.22 ผลตอบสนองอิมพัลส์ของตัวกรองผ่านต่ำในอุดมคติ.....	27
2.23 ก. ผลตอบสนองอิมพัลส์ของตัวกรองเอฟไออาร์ โดย M เท่ากับ 25.....	27

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23 ข. ฟังก์ชันถ่ายโอนของตัวกรองเอพไออาร์ที่ได้จากผลตอบสนองอิมพัลส์.....	27
2.24 ผลตอบสนองอิมพัลส์ของตัวกรองเอพไออาร์ มีคุณสมบัติคอซอล.....	28
2.25 โครงสร้างตัวกรองไอไออาร์แบบโดยตรงชนิดที่หนึ่ง.....	30
2.26 โครงสร้างตัวกรองไอไออาร์แบบโดยตรงชนิดที่สอง.....	31
2.27 โครงสร้างตัวกรองไอไออาร์แบบคาสเคด.....	32
2.28 โครงสร้างตัวกรองไอไออาร์แบบขนาน.....	33
2.29 แผนภาพบล็อกบีบอัดสัญญาณเสียงพิสัยพลวัต.....	34
2.30 ตัวกรองชนิดผลตอบสนองแบบไอไออาร์.....	35
2.31 แผนภาพบล็อกการบีบอัดสัญญาณพิสัยพลวัตแบบหลายแบนด์.....	37
3.1 แผนภาพบล็อกของ TMS320C6713.....	40
3.2 แผนภาพบล็อกภายใน McBSP.....	43
3.3 DSK บอร์ด TMS320C6713.....	46
3.4 ข้อมูลในท่อ.....	48
3.5 ลำดับการทำงานของวัตถุ.....	49
3.6 แผนภาพบล็อกการออกแบบการทำงานพื้นฐาน.....	50
3.7 แผนภาพบล็อกการทำงานเมื่อใช้ DSP/BIOS.....	51
4.1 ผลตอบสนองของตัวกรองผ่านต่ำ.....	54
4.2 ผลตอบสนองความถี่ตัดของตัวกรองผ่านต่ำ.....	55
4.3 ผลตอบสนองของตัวกรองผ่านแถบ.....	55
4.4 ผลตอบสนองความถี่ตัดทางด้านต่ำของตัวกรองผ่านแถบ.....	56
4.5 ผลตอบสนองความถี่ตัดทางด้านสูงของตัวกรองผ่านแถบ.....	56
4.6 ผลตอบสนองของตัวกรองผ่านสูง.....	57
4.7 ผลตอบสนองความถี่ตัดของตัวกรองผ่านสูง.....	57
4.8 ลักษณะการต่ออุปกรณ์เพื่อทดสอบวัดวัดช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ.....	58
4.9 แผนภาพบล็อกวิธีการวัดช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ.....	59
4.10 ผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที.....	60
4.11 ผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเลิกกระทำ 50 มิลลิวินาที.....	60

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 ผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเข้ากระทำ 20 มิลลิวินาที.....	61
4.13 ผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเล็กกระทำ 30 มิลลิวินาที.....	61
4.14 ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเข้ากระทำ 10 มิลลิวินาที.....	62
4.15 ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเล็กกระทำ 20 มิลลิวินาที.....	62
4.16 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่ผ่านต่ำ.....	63
4.17 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่แถบผ่าน.....	64
4.18 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่ผ่านสูง.....	64
4.19 ลักษณะการต่ออุปกรณ์เพื่อวัดสเปกตรัมบีบอัด สัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์.....	65
4.20 แผนภาพบล็อกวิธีวัดสเปกตรัมบีบอัดสัญญาณพิสัยพลวัตแบบหลายแบนด์.....	65
4.21 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด.....	66
4.22 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด.....	66
4.23 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด.....	67

คำอธิบายสัญลักษณ์และคำย่อ

$v(t)$	=	voltage carrier
V_c	=	amplitude
f	=	frequency
θ	=	anagle
AM	=	amplitude modulation
FM	=	frequency modulation
PM	=	phase modulation
Δf_c	=	frequency deviation
ΔF_c	=	maximum frequency deviation
$M.I.$	=	modulation Index
f_m	=	maximum frequency
F_m	=	maximum frequency modulation
ADC	=	analog to digital converter
DAC	=	digital to analog converter
FIR	=	finite impulse response
IIR	=	infinite impulse response
δ_p	=	passband ripple
δ_s	=	stopband attenuation
F_s	=	sampling frequency
ΔF	=	transition band
ω_c	=	cutoff frequency
$g(n)$	=	gain controller
$c(n)$	=	level detector
$x(n)$	=	input audio signal
a_{attack}	=	attack time
$a_{release}$	=	release time

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

VLIW	=	very-long-instruction-word
MFLOPS	=	million floating-point operations per second
MIPS	=	million instructions per second
McBSP	=	multichannel buffer serial port
HPI	=	host port interface
EMIF	=	enhance memory interface
DSP	=	digital signal processing
DMA	=	direct memory access
ALU	=	arithmetic logic units
.M	=	multiplier units
.L	=	arithmetic and logic units
.S	=	arithmetic operations
.D	=	data units
AMR	=	address mode register
EDMA	=	enhanced direct memory access channel
DX	=	data transmit
DR	=	data receive
CLKX	=	transmit clock
CLKR	=	receive clock
FSX	=	transmit frame synchronize
DRR	=	data receive register
XSR	=	transmit shift register
RSR	=	receive shift register
RBR	=	receive buffer register
SPCR	=	serial port control register
RCR/XCR	=	receive/transmit control register
RCER/XCER	=	receive/transmit channel enable register
PCR	=	pin control register

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

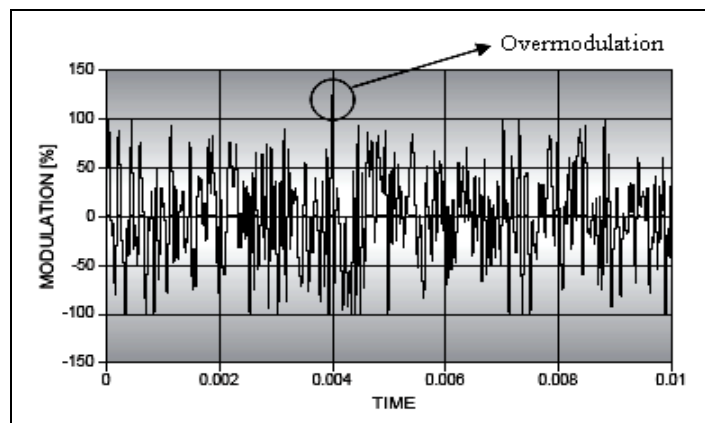
SRGR	=	sample rate generator register
ISR	=	interrupt service routine
CSR	=	control status register
GIE	=	global interrupt enable
IFR	=	interrupt flag register
ISR	=	interrupt set register
ICR	=	interrupt clear register
ISTP	=	interrupt service table pointer
IRP	=	interrupt return pointer
NRP	=	nonmaskable interrupt return pointer
IST	=	interrupt service table
FP	=	fetch packet
API	=	application programming interface
SWI	=	software interrupt
PRD	=	periodic interrupt
CCS	=	code composer studio
GUI	=	graphical user interface
F_s	=	set the sampling frequency
F_{pass}	=	set the beginning of the passband
F_{stop}	=	set the end of the passband
A_{stop}	=	set the stopband attenuation for both
A_{pass}	=	set the passband ripple
FCC	=	federal communication commission

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ในปัจจุบันการส่งกระจายเสียงระบบเอฟเอ็ม เป็นที่นิยมกันอย่างแพร่หลายเพราะสามารถกำจัดสัญญาณรบกวนได้ดี มีแบนด์วิดท์กว้าง (bandwidth) ซึ่งสามารถส่งแถบความถี่ตลอดย่านความถี่เสียง แต่เนื่องจากการส่งกระจายเสียงของเครื่องส่ง กระบวนการมอดูเลตสัญญาณให้ได้ค่าเปอร์เซ็นต์การมอดูเลตสูง ๆ ซึ่งจะส่งผลให้เครื่องรับสามารถแยกเอาสัญญาณเสียงออกมาจากสัญญาณมอดูเลชันได้อย่างสมบูรณ์ การมอดูเลชันโดยทั่วไปจะรักษาระดับให้ใกล้เคียง 100 เปอร์เซ็นต์ แต่ในบางครั้งระดับการมอดูเลชันสูงเกินกว่า 100 เปอร์เซ็นต์ ที่เรียกว่า โอเวอร์มอดูเลชัน (overmodulation) จะทำให้สัญญาณพาหะบางช่วงของสัญญาณที่โอเวอร์มอดูเลชันขาดหายไป (carrier off) ซึ่งเกิดขึ้นในช่วงสั้น ทำให้เกิดการบิดเบือนสัญญาณข้อมูลที่ส่งออกไปไม่ครบถ้วน และบางช่วงของสัญญาณพาหะหายไป ทำให้เกิดสัญญาณที่มีความถี่ใหม่ขึ้นและเกิดการรบกวนของสัญญาณข้างเคียง



รูปที่ 1.1 การเกิดโอเวอร์มอดูเลชัน

ในงานวิจัยนี้ ใช้เทคนิคกระบวนการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ (multiband dynamic range compression) เพื่อแก้ไขปัญหาการเกิดโอเวอร์มอดูเลชัน ในการส่งกระจายเสียงระบบ

เอฟเอ็ม โดยการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ ซึ่งแบ่งเป็น 2 แบบคือ การบีบอัดสัญญาณเสียงพิสัยพลวัตแบบแบนด์เดียว (single-band audio dynamic range compression) และการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ (multiband audio dynamic range compression)

เนื่องจากการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบแบนด์เดียวมีข้อดีคือ นำมาใช้งานได้ง่าย และมีความซับซ้อนในการออกแบบน้อย ส่วนข้อเสียคือ จำกัดความดังของสัญญาณเอาต์พุตสูง ส่วนการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์มีข้อดีคือ เนื่องจากในแต่ละแบนด์บีบอัดสัญญาณเสียงที่มีความดัง ส่วนสัญญาณเสียงที่มีความเบาไม่ถูกบีบอัด ทำให้สัญญาณเอาต์พุตจากการบีบอัดสัญญาณมีคุณภาพเสียงที่ดี

งานวิจัยนี้ทำการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ โดยใช้เทคนิคในการแบ่งสเปกตรัม (spectrum) ของสัญญาณเสียงออกเป็นหลายแบนด์ความถี่ และบีบอัดสัญญาณเสียง (compression) หรือจำกัดสัญญาณ โดยระหว่างแบนด์มีการเชื่อมต่อ (coupling) กันเพื่อป้องกันความแตกต่างในเรื่องอัตราขยายของแบนด์ที่ติดกัน (adjacent band) และประกอบด้วยเวลาเข้ากระทำ (attack time) เวลาเลิกกระทำ (release time) ที่เหมาะสม ซึ่งในกระบวนการหลายแบนด์ มีเวลาคงที่ในแต่ละความถี่ สำหรับแบนด์ที่มีความถี่ผ่านต่ำจะมีเวลาคงที่ช้ากว่าแบนด์ที่มีความถี่ผ่านสูง นอกจากนี้การบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ สามารถใช้แทนอีควอไลซ์เซอร์ (equalizer) ในการปรับอัตราขยายที่มีหลายแบนด์

งานวิจัยนี้นำ DSP chip หรือ digital signal processor chip ซึ่งเป็นไมโครโปรเซสเซอร์ (microprocessor) ประเภทหนึ่ง ที่ใช้ในงานด้านการประมวลผลสัญญาณดิจิทัลที่ต้องการทำงานแบบเวลาจริง (real-time) เนื่องจากเหตุผลทางด้านประสิทธิภาพการทำงานสูงของ DSP มีความยืดหยุ่นในการโปรแกรมการทำงาน โดยใช้ DSP TMS320C6713 ของบริษัท Texas Instrument เป็นกระบวนการสัญญาณดิจิทัลแบบจุดลอย (floating – point) ประมวลผลสัญญาณแบบเวลาจริง (real time) ความเร็ว 255 MHz, ROM 512 Kbyte, RAM 256 Kbyte

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 แก้ไขปัญหาโอเวอร์มอดูเลชัน (overmodulation) ในการส่งกระจายเสียงระบบเอฟเอ็ม
- 1.2.2 ทำการบีบอัดสัญญาณพิสัยพลวัตหลายแบนด์ (multiband dynamic range compression) แบบเวลาจริง (real time)

1.3 สมมติฐานของการวิจัย

- 1.3.1 สามารถเปลี่ยนแปลงการบีบอัดสัญญาณเสียงเป็นแบบอัตโนมัติ
- 1.3.2 ใช้การบีบอัดสัญญาณเสียงแบ่งเป็น 3 แบนด์

1.3.3 สามารถใช้บอร์ด TMS320C6713 ในการบีบอัดสัญญาณเสียงได้

1.4 ขอบเขตของการวิจัย

- 1.4.1 ทำการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ โดยใช้บอร์ด DSP TMS320C6713 แบบอัตราขยายอัตโนมัติ (automatic gain)
- 1.4.2 แบ่งการบีบอัดสัญญาณเป็น 3 ย่านความถี่
- 1.4.3 มีเวลาเข้ากระทำ (attack time) และเวลาเลิกกระทำ (release time)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 แก้ไขปัญหาโอเวอร์มอดูเลชัน ในการส่งกระจายเสียงระบบเอฟเอ็ม
- 1.5.2 มีประสิทธิภาพในการควบคุมสัญญาณเสียง
- 1.5.3 ใช้บอร์ด TMS320C6713 เพื่อทำการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์

1.6 การจัดรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์นี้ได้นำเสนอ การบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ ซึ่งมีเนื้อหาทั้งหมด 5 บท โดยแต่ละบทประกอบด้วยเนื้อหาสรุปได้ดังนี้

บทที่ 1 เป็นบทนำกล่าวถึงความสำคัญของปัญหาและสาเหตุปัญหา วัตถุประสงค์ ขอบเขตของงานวิจัย และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย รวมทั้งแนะนำเนื้อหาเบื้องต้นของวิทยานิพนธ์ฉบับ

บทที่ 2 กล่าวถึงทฤษฎีมอดูเลตแบบเอฟเอ็ม ทฤษฎีระบบประสาทเกี่ยวกับการได้ยิน ทฤษฎีการออกแบบตัวกรอง และทฤษฎีบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ โดยทฤษฎีทั้งหมดนี้นำมาใช้ในการแก้ไขปัญหาการเกิดโอเวอร์มอดูเลชัน

บทที่ 3 บทนี้เป็นการออกแบบและการใช้งาน ซึ่งจะกล่าวถึงโครงสร้าง TMS320C6713 การออกแบบประมวลผลแบบเวลาจริง ซึ่งนำ DSP/BIOS มาช่วยในการจัดลำดับการทำงาน โดยใช้มาตรฐานอัลกอริทึมของบริษัทเท็กซัสอินทรูเมนส์

บทที่ 4 กล่าวถึงผลการทดลองของการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ โดยแบ่งออกเป็น ผลการทดลองผลตอบสนองของตัวกรอง ผลตอบสนองเวลาเข้ากระทำและเลิกกระทำของแต่ละแบนด์ เปรียบเทียบระหว่างสัญญาณที่ถูกบีบอัดในอัตราส่วนต่าง ๆ และสัญญาณที่ไม่ถูกบีบอัด

บทที่ 5 เป็นส่วนของบทสรุป ผลการทดลองบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ ข้อเสนอแนะและแนวทางการพัฒนาในการศึกษาวิจัยเกี่ยวกับการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ในโอกาสต่อไป

ภาคผนวก ก โปรแกรม

ภาคผนวก ข บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 กล่าวนำ

กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ สำหรับการส่งกระจายเสียงของเครื่องส่งระบบเอฟเอ็ม เพื่อแก้ไขปัญหาการเกิดโอเวอร์มอดูเลตและการขาดหายของสัญญาณพาหะ ดังนั้นเนื้อหาในบทนี้จะกล่าวถึงทฤษฎีการมอดูเลตแบบเอฟเอ็ม ทฤษฎีระบบประสาทเกี่ยวกับการได้ยิน ทฤษฎีการออกแบบตัวกรอง และทฤษฎีบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ ซึ่งทฤษฎีทั้งหมดนี้เกี่ยวข้องกับทฤษฎีในบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์

2.2 ทฤษฎีมอดูเลตแบบเอฟเอ็ม

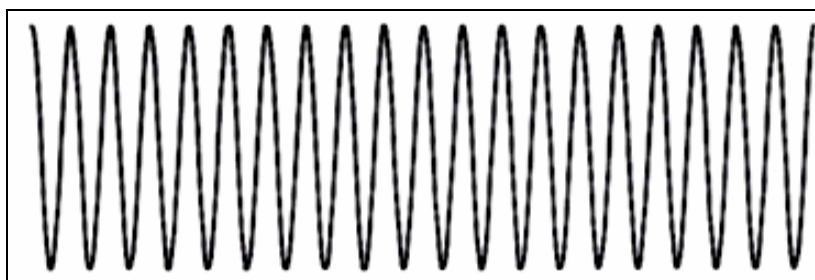
ในขบวนการมอดูเลต ใช้คลื่นรูปไซน์ที่มีความถี่ผ่านสูงเป็นพาหะ ซึ่งเปลี่ยนแปลงคุณสมบัติบางอย่างของพาหะด้วยสัญญาณข่าวสาร โดยทั่วไปสัญญาณข่าวสารได้แก่ สัญญาณเสียง (signal audio) สัญญาณภาพหรือข่าวสารอื่น ๆ การเปลี่ยนแปลงคุณสมบัติของคลื่นพาหะนี้ เรียกว่า การมอดูเลต ซึ่งคลื่นรูปไซน์ที่ใช้เป็นพาหะ สามารถเขียนเป็นสมการทางคณิตศาสตร์แทนได้ดังนี้

$$v(t) = V_c \sin(2\pi ft \pm \theta) \quad (2.1)$$

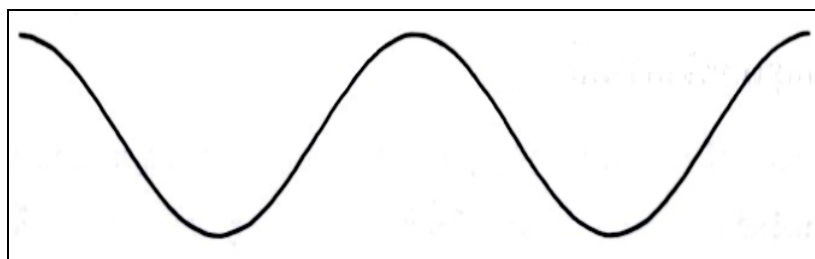
เมื่อ $v(t)$ คือค่าแรงดันของคลื่นพาหะ V_c คือแอมพลิจูดสูงสุด f คือความถี่ และ θ คือเฟสหรือมุมทางไฟฟ้า

จากสมการ (2.1) พบว่าคุณสมบัติของคลื่นรูปไซน์ที่สำคัญมี 3 ประการ สามารถเปลี่ยนแปลงหรือมอดูเลตได้ คือ แอมพลิจูดสูงสุด ความถี่ และเฟส ซึ่งการมอดูเลตกับคลื่นพาหะแบ่งออกเป็น 3 แบบคือ การมอดูเลตแบบเอเอ็ม (amplitude modulation หรือ AM) การมอดูเลตแบบเอฟเอ็ม (frequency modulation หรือ FM) และการมอดูเลตแบบเฟส (phase modulation หรือ PM) ซึ่งจากการมอดูเลตแบบเอฟเอ็มและการมอดูเลตแบบเฟสจะคล้ายกันมาก สามารถเรียกรวมกันได้ว่า การมอดูเลตแบบเชิงมุม (angle modulation) โดยการมอดูเลตแบบเอฟเอ็ม คือการนำความแรงของสัญญาณเสียงไปควบคุมความถี่ของคลื่นวิทยุ ซึ่งเมื่อความแรงของสัญญาณเสียงมากมีผลให้ความถี่ของคลื่นวิทยุจะสูงขึ้นแต่เมื่อความแรงของสัญญาณเสียงลดลงความถี่ของคลื่นวิทยุจะต่ำลง และถ้าไม่

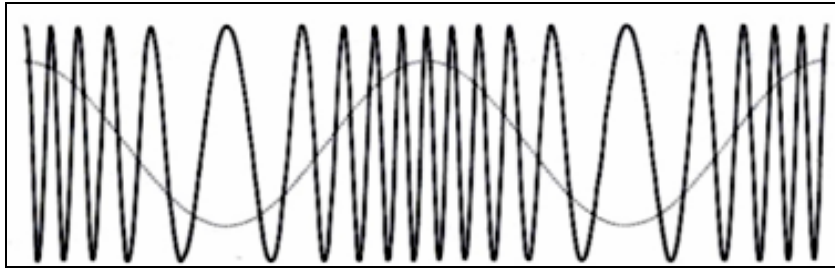
มีสัญญาณเสียงเข้ามาความถี่ของคลื่นวิทยุจะอยู่ในสภาพปกติคือ ความถี่ปานกลาง โดยจากรูปที่ 2.1 คือ คลื่นพาหะปกติเป็นคลื่นรูปไซน์ในที่นี้ให้เป็นคลื่นรูปสามเหลี่ยม ถ้าสัญญาณยังไม่ได้ทำการมอดูเลต ความถี่คลื่นพาหะเป็นรูปคลื่นที่มีแอมพลิจูดค่าคงที่ ส่วนในรูปที่ 2.2 คือ สัญญาณมอดูเลตคลื่นรูปไซน์ทางด้านบวกทำให้ความถี่ของคลื่นพาหะเพิ่มขึ้นตามสัดส่วน ส่วนที่ความถี่ผ่านสูงจะทำให้สัญญาณมอดูเลตมีแอมพลิจูดสูง และถ้าสัญญาณมอดูเลตมีแอมพลิจูดต่ำจะทำให้ความถี่ของคลื่นพาหะต่ำตามไปด้วย ถ้าสัญญาณมอดูเลตมีแอมพลิจูดเป็นศูนย์จะทำให้คลื่นพาหะอยู่ที่จุดกึ่งกลางความถี่ และเมื่อสัญญาณทางด้านลบเข้ามาเป็นผลให้ความถี่พาหะลดลง และจะลดลงอย่างต่อเนื่องจนกระทั่งถึงค่าสูงสุดของครึ่งไซเคิลเกิดด้านลบจากคลื่นไซน์การมอดูเลต ส่วนในรูปที่ 2.3 เป็นสัญญาณทางความถี่ที่ได้จากการมอดูเลต จะพบว่าพาหะรูปคลื่นไซน์เกิดการเปลี่ยนแปลงทางความถี่ ซึ่งความถี่มีระยะห่างบ้างถี่บ้างตามสัญญาณการมอดูเลต และในรูปที่ 2.4 เป็นกรณีของการมอดูเลตทางเฟส ซึ่งค่าเบี่ยงเบนความถี่ขึ้นอยู่กับค่าเบี่ยงเบนของเฟส ดังนั้นจะเห็นได้ว่าความถี่ชั่วขณะมอดูเลตแบบเฟส มีค่าสูงสุดในจังหวะที่สัญญาณมอดูเลตมีค่าความชันสูงสุด



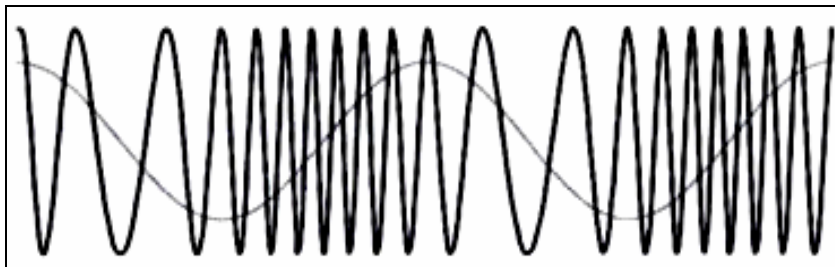
รูปที่ 2.1 คลื่นพาหะ



รูปที่ 2.2 สัญญาณมอดูเลต



รูปที่ 2.3 มอดูเลตแบบแอมพลิจูด



รูปที่ 2.4 มอดูเลตทางเฟส

ตัวอย่างเช่น ความถี่ของคลื่นพาหะที่ 150 เมกะเฮิร์ตซ์ แอมพลิจูดสูงสุดของสัญญาณมอดูเลต ทำให้ความถี่ผ่านสูงสุดเกิดการเบี่ยงเบน 30 กิโลเฮิร์ตซ์ ทำให้ความถี่ของคลื่นพาหะเบี่ยงเบนเพิ่มขึ้น 150.3 เมกะเฮิร์ตซ์ และลดลง 149.97 เมกะเฮิร์ตซ์ ผลรวมของความถี่เบี่ยงเบนเป็น $150.3 - 149.97 = 0.06$ เมกะเฮิร์ตซ์

2.2.1 เปอร์เซนต์ของการมอดูเลต

ในระบบแอมพลิจูด การเบี่ยงเบนของคลื่นที่มีค่าน้อยกว่าค่าสูงสุดที่กำหนดไว้ สามารถคิดเป็นเปอร์เซนต์ของการเบี่ยงเบนสูงสุดได้ ซึ่งปริมาณของการเบี่ยงเบนความถี่จะเป็นตัวกำหนด เปอร์เซนต์ของการมอดูเลต สำหรับคลื่นแอมพลิจูดนั้นเปอร์เซนต์ของการมอดูเลตคิดจากอัตราส่วนของการเบี่ยงเบนความถี่ (Δf_c) ต่อการเบี่ยงเบนความถี่ผ่านสูงสุด (ΔF_c) ที่กำหนดไว้ให้เป็นการมอดูเลตเต็ม (full modulate) ที่ 100% ซึ่งเขียนเป็นสมการได้ดังนี้

$$\% \text{การมอดูเลชัน} = \frac{\text{การเบี่ยงเบนความถี่}}{\text{การเบี่ยงเบนความถี่สูงสุด}} \times 100$$

$$= \frac{\Delta f_c}{\Delta F_c} \times 100 \quad (2.2)$$

ตัวอย่างเช่น เครื่องส่งวิทยุกระจายเสียงระบบเอฟเอ็ม ซึ่งกำหนดการเบี่ยงเบนความถี่ผ่านสูงสุด (Δf_c) ไว้เท่ากับ ± 75 กิโลเฮิร์ตซ์ สำหรับการมอดูเลตที่ 100% ดังนั้นการมอดูเลต 100% จะเกิดขึ้นก็ต่อเมื่อทำการมอดูเลตสัญญาณเข้าไปทำให้คลื่นพาหะของเครื่องส่ง มีการเบี่ยงเบนความถี่ไปเท่ากับ ± 75 กิโลเฮิร์ตซ์ และเมื่อทำการลดแอมพลิจูดของสัญญาณที่นำมามอดูเลตลง จนกระทั่งทำให้การเบี่ยงเบนความถี่ตกลงถึง ± 75 กิโลเฮิร์ตซ์ เครื่องส่งจะทำการมอดูเลตที่ 60%

2.2.2 ดรรชนีการมอดูเลต

ดรรชนีการมอดูเลต บางครั้งเรียกว่า มอดูเลชันแฟกเตอร์ (modulation factor) ซึ่งค่าของดรรชนีการมอดูเลตคิดเป็นอัตราส่วนระหว่างช่วงการเบี่ยงเบนความถี่ (Δf_c) ต่อความถี่ที่นำมามอดูเลตหรือความถี่เสียง (f_m) ซึ่งอัตราส่วนนี้สามารถคิดเป็นเศษส่วนหรือจำนวนเต็มได้ ดังแสดงโดยสมการ(2.3)

$$\begin{aligned} \text{ดรรชนีการมอดูเลต} &= \frac{\text{การเบี่ยงเบนความถี่}}{\text{ความถี่ที่นำมามอดูเลต}} \times 100 \\ M.I. &= \frac{\Delta f_c}{f_m} \end{aligned} \quad (2.3)$$

2.2.3 อัตราส่วนการเบี่ยงเบน

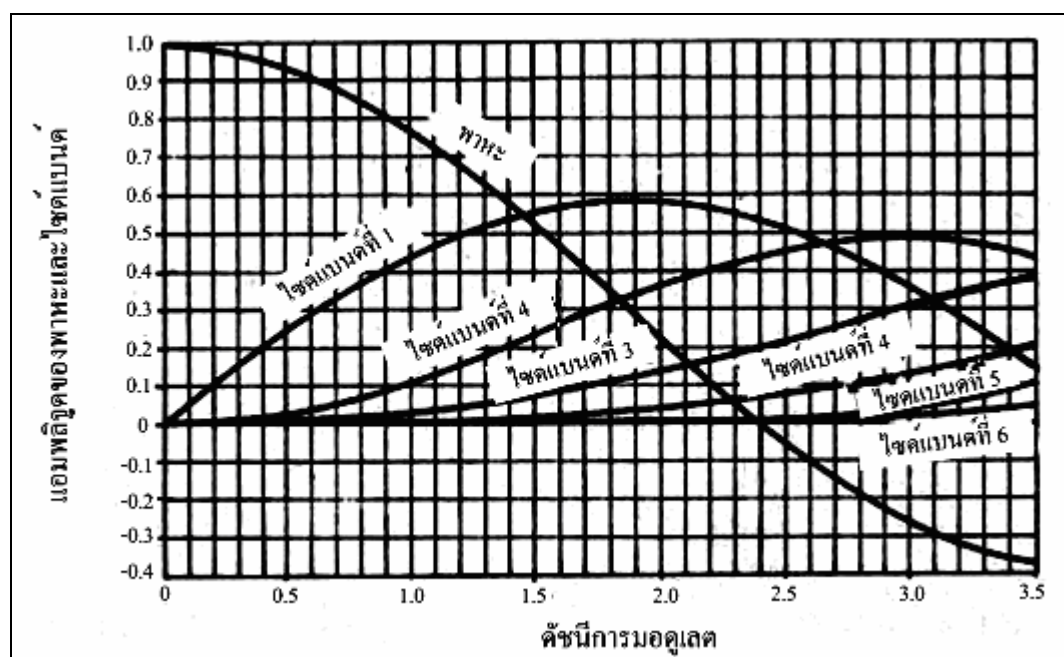
อัตราส่วนการเบี่ยงเบนเป็นอัตราส่วนการเบี่ยงเบนความถี่ผ่านสูงสุด (ΔF_c) ที่ 100% มอดูเลตต่อความถี่ที่นำมามอดูเลตสูงสุด (F_m) ซึ่งอัตราส่วนการเบี่ยงเบนนี้นำมาใช้ประโยชน์ในการหาค่าแบนด์วิดท์กว้างที่สุดที่ต้องการของระบบเอฟเอ็ม ซึ่งเขียนเป็นสมการได้ดังนี้

$$\begin{aligned} \text{อัตราส่วนการเบี่ยงเบน} &= \frac{\text{การเบี่ยงเบนความถี่สูงสุด}}{\text{ความถี่ที่นำมามอดูเลตสูงสุด}} \times 100 \\ &= \frac{\Delta F_c}{F_m} \end{aligned} \quad (2.4)$$

ในเครื่องส่งวิทยุกระจายเสียงระบบเอฟเอ็ม ซึ่งมีย่านความถี่อยู่ในช่วงตั้งแต่ 88 ถึง 108 เมกะเฮิร์ตซ์ โดยกำหนดการเบี่ยงเบนความถี่ผ่านสูงสุดไว้เท่ากับ ± 75 กิโลเฮิร์ตซ์ สำหรับการมอดูเลต 100% และจะสามารถทำงานได้ดีในย่านถึงความถี่ที่นำมามอดูเลตสูงสุด 15 กิโลเฮิร์ตซ์ จึงทำให้ได้อัตราส่วนการเบี่ยงเบนเท่ากับ 5

2.2.4 ไซด์แบนด์เอฟเอ็ม

ในระบบเอฟเอ็มเมื่อทำการมอดูเลตด้วยสัญญาณรูปคลื่นไซน์ จะทำให้เกิดความถี่ไซด์แบนด์จำนวนนับอนันต์ เนื่องจากการเบี่ยงเบนความถี่คลื่นพาหะทำให้เกิดความถี่เพิ่มขึ้นอีกมากมาย ความจริงแล้วไซด์แบนด์ที่อยู่ห่างจากความถี่แถบผ่านมาก ๆ จะมีแอมพลิจูดเล็กมากจนไม่ต้องคำนึงถึง ซึ่งในระบบเอฟเอ็มจะรักษาแอมพลิจูดไว้คงที่เสมอ คือกำลังของสัญญาณคลื่นพาหะยอมกระจายไปอยู่ในไซด์แบนด์ ความสัมพันธ์ของสัญญาณคลื่นพาหะกับไซด์แบนด์ในระบบเอฟเอ็มขึ้นอยู่กับครรชนิการมอดูเลต เนื่องจากครรชนิการมอดูเลตเป็นตัวกำหนดจำนวนของไซด์แบนด์ที่สำคัญ และแอมพลิจูดของสัญญาณคลื่นพาหะกับไซด์แบนด์ต่าง ๆ



รูปที่ 2.5 กราฟแอมพลิจูดของคลื่นพาหะและไซด์แบนด์ระบบเอฟเอ็ม
(สุชาติ กังวานจิตต์, 2538)

จากรูปที่ 2.5 แสดงกราฟแอมพลิจูดของสัญญาณคลื่นพาหะกับไซด์แบนด์ ที่เป็นดัชนีการมอดูเลตค่าต่าง ๆ พบว่าเมื่อครรชนิการมอดูเลตเป็นศูนย์จะมีแต่สัญญาณคลื่นพาหะอย่าง

เดียว เท่ากับ 1 หน่วย ซึ่งคลื่นไซค์แบนด์เป็นศูนย์ เมื่อดัชนีการมอดูเลตเพิ่มขึ้นจำนวนไซค์แบนด์จะเพิ่มขึ้น แอมพลิจูดของไซค์แบนด์ก็จะใหญ่ขึ้น แต่แอมพลิจูดของคลื่นพาหะกลับเล็กลงจนกระทั่งดัชนีการมอดูเลตเท่ากับ 2.4 คลื่นพาหะจะเป็นศูนย์ ในขณะที่กำลังของคลื่นเอฟเอ็มจะอยู่ในไซค์แบนด์ทั้งสิ้น เมื่อครรชนิการมอดูเลตเพิ่มขึ้นอีก คลื่นพาหะจะมีค่าเพิ่มขึ้นตาม

2.2.5 แบนด์วิดท์ของสัญญาณเอฟเอ็ม

ในระบบเอฟเอ็มจำนวนไซค์แบนด์และแอมพลิจูดของไซค์แบนด์ จะขึ้นอยู่กับครรชนิการมอดูเลต โดยที่ความถี่ของไซค์แบนด์มีค่าสัมพันธ์กับความถี่ของสัญญาณที่นำมามอดูเลต เช่น ไซค์แบนด์ของกลุ่มแรกมีความถี่เท่ากับ $f_c \pm f_m$ ไซค์แบนด์ของกลุ่มที่สองมีความถี่เท่ากับ $f_c \pm 2f_m$ และ ไซค์แบนด์ของกลุ่มที่สามมีความถี่เท่ากับเท่ากับ $f_c \pm 3f_m$ เป็นต้น จึงทำให้แบนด์วิดท์ของคลื่นเอฟเอ็มครอบคลุมจำนวนไซค์แบนด์ที่สำคัญทุกตัว นั่นคือ แบนด์วิดท์ขึ้นอยู่กับครรชนิการมอดูเลต และความถี่ของสัญญาณที่นำมามอดูเลต เช่น เครื่องส่งวิทยุเอฟเอ็มภาคเสียงของโทรทัศน์เครื่องหนึ่ง จะทำการมอดูเลตเต็มที 100% เพื่อให้เกิดการเบี่ยงเบนความถี่เท่ากับ ± 50 กิโลเฮิร์ตซ์ ด้วยสัญญาณเสียงความถี่ 15 กิโลเฮิร์ตซ์ มีค่าอัตราส่วนการเบี่ยงเบนเท่ากับ 3.3 ทำให้เกิดไซค์แบนด์จำนวน 6 คู่ และมีแบนด์วิดท์กว้าง 180 กิโลเฮิร์ตซ์ ซึ่งความถี่เสียงเมื่อนำมามอดูเลตจะมีค่าต่าง ๆ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ความถี่เสียงที่นำมามอดูเลตที่มีค่าต่าง ๆ กัน

f_m (kHz)	Δf_c (kHz)	$M.I = \frac{\Delta f_c}{f_m}$	จำนวนของคู่ ไซค์แบนด์	แบนด์วิดท์ (kHz)
15.0	50.0	3.3	6	180
10.0	50.0	5.0	8	160
5.0	50.0	10.0	14	140
3.3	50.0	15.0	19	127
2.5	50.0	20.0	25	125
2.0	50.0	25.0	30	120

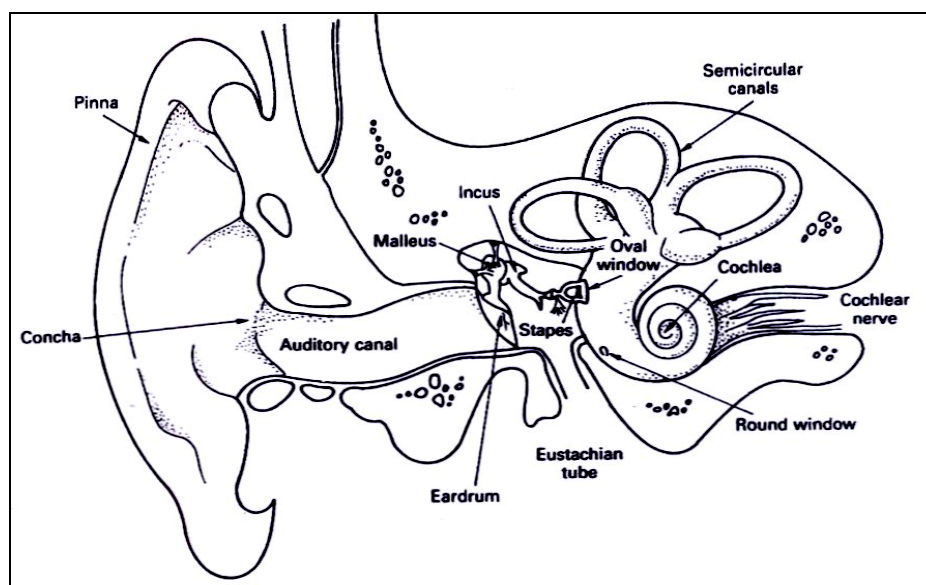
หมายเหตุ (วิโรจน์ แก้วจันทร์, 2543)

ในทางปฏิบัติแอมพลิจูดของสัญญาณเสียงที่มีความถี่ผ่านสูง ๆ ส่งผลกระทบกับระบบกระจายเสียง เนื่องจากถ้าแอมพลิจูดยิ่งสูงทำให้มีโอกาสเกิดโอเวอร์มอดูเลต ดังนั้นแบนด์วิดท์ของเครื่องส่งวิทยุกระจายเสียงระบบเอฟเอ็ม กำหนดให้แบนด์วิดท์แคบลงกว่าย่านสูงสุดของไซค์แบนด์ โดยไม่ทำให้เกิดความผิดเพี้ยนจนเป็นที่รู้สึกได้

2.3 ทฤษฎีระบบประสาทเกี่ยวกับการได้ยิน

2.3.1 ทฤษฎีเกี่ยวกับกายภาพของมนุษย์

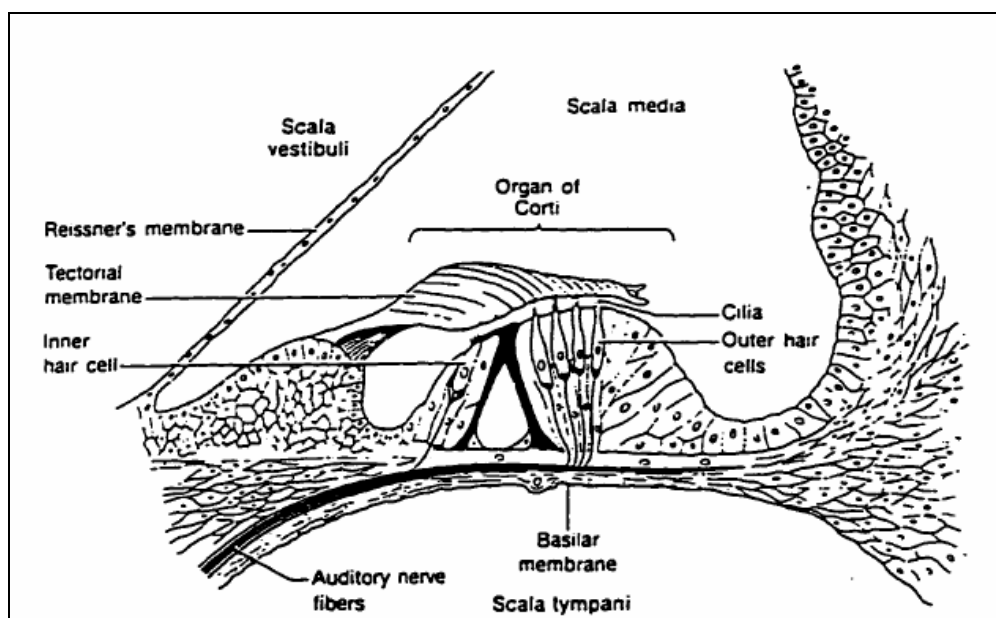
หูของมนุษย์แบ่งออกเป็น 3 ส่วน คือ หูชั้นนอก (outer ear) หูชั้นกลาง (middle ear) และหูชั้นใน (inner ear) หูชั้นนอก ประกอบด้วยใบหู (pinna) และช่องหู (ear canal) ซึ่งหูชั้นนอกมีหน้าที่ขยายการสั่นสะเทือนคลื่นเสียงของอากาศที่เข้ามา ส่วนหูชั้นกลาง ทำหน้าที่เป็นตัวแปลง (transduce) จากการสั่นของคลื่นเสียง ซึ่งประกอบด้วยเยื่อแก้วหู กระดูกค้อน (malleus) กระดูกทั่ง (incus) และกระดูกโกลน (stapes) โดยกระดูกเล็ก ๆ เหล่านี้มีกลไกในการทำงานที่ต่อเนื่องกัน ส่วนหูชั้นในทำหน้าที่เป็นตัวกรองและตัวแปลงคลื่นเสียง ซึ่งเกิดจากกลไกในการสั่นสะเทือนของไฮโดรไดนามิก (hydrodynamic) และจากการสั่นสะเทือนของอิเล็กโทรเคมีคอล (electrochemical) โดยผลที่ได้จากสัญญาณอิเล็กโทรเคมีคอลจะถูกส่งไปยังเส้นประสาทของสมอง ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 โครงสร้างของหูมนุษย์ (michael.,1999.)

โครงสร้างของหูชั้นนอกและหูชั้นในมีความไวต่อการได้ยิน ซึ่งจะขยายสัญญาณที่เข้ามาจากพลังงานคลื่นเสียงที่แพร่กระจายออกจากแหล่งกำเนิด โดยภายในของหูชั้นนอกมีใบหูทำหน้าที่คอยจับคลื่นเสียง และในส่วนของช่องหูทำหน้าที่ขยายคลื่นเสียง (resonator) ซึ่งอยู่ในย่านความถี่ตั้งแต่ 2 ถึง 5 กิโลเฮิร์ตซ์ ส่วนหูชั้นกลางหรือเยื่อแก้วหู (tympanic membrane) ทำหน้าที่รับการสั่นสะเทือนคลื่นเสียงที่เพิ่มขึ้นจากช่องหูและส่งผ่านกระดูกเล็ก ๆ (กระดูกค้อน กระดูกทั่ง กระดูกโกลน) ซึ่งสามารถขยายคลื่นเสียงเมื่อได้ยินเสียงเบา (soft sound) และสามารถปรับการเคลื่อนไหว

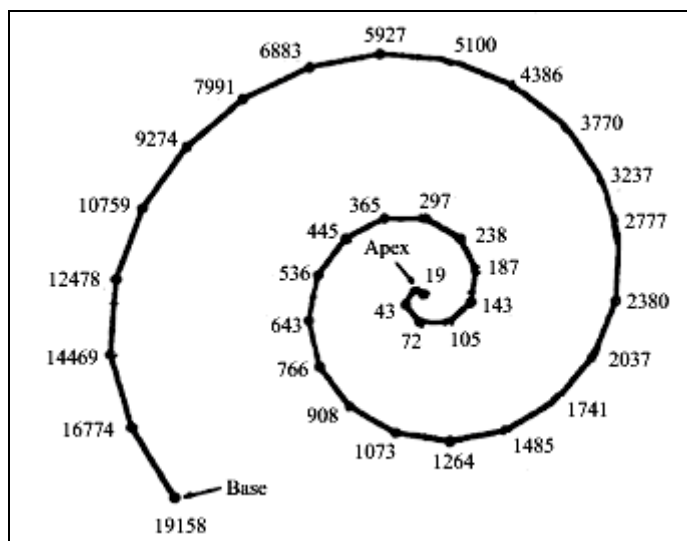
กล้ามเนื้อ (muscle action) ให้น้อยลงเพื่อป้องกันสัญญาณเสียงที่เริ่มมีความดัง ต่อจากนั้นคลื่นเสียงถูกส่งไปยังหน้าต่างวงรี (oval window)



รูปที่ 2.7 ส่วนประกอบในคลอเคลีย (Jon C., 1996.)

หูชั้นในประกอบด้วยอวัยวะหลอดกึ่งวง (semicircular canals) ที่ทำให้ออแกน (organ) มีความสมดุล นอกจากนี้ในคลอเคลียมีเบสิลาร์เมมเบรน (basilar membrane) และออแกนจากคอร์ตติ (organ of corti) ซึ่งมีกลไกการทำงานที่สัมพันธ์กันในการแปลงการสั่นสะเทือนของคลื่นเสียงเพื่อถอดรหัสสัญญาณเสียงสู่ระบบประสาท นอกจากนี้ออแกนจากคอร์ตติมีความไวต่อการสั่นสะเทือนคลื่นเสียงซึ่งมีความสำคัญต่อหูชั้นใน โดยเบสิลาร์เมมเบรนเป็นส่วนหนึ่งในคลอเคลียประกอบด้วยเซลล์ขน (hair cell) 4 แถว โดยจะถูกปกคลุมด้วยเยื่อหุ้มเซลล์ซึ่งสามารถเคลื่อนที่ตอบสนองตามการกระตุ้นจากการสั่นสะเทือนของเหลวที่บรรจุอยู่ในเยื่อแก้วหู (fluid-filled tympanic) และโพรงหูส่วนใน (vestibular canals) มีขนในรูปหู 16,000 ถึง 20,000 เส้น จะกระจายตามความยาวของเบสิลาร์เมมเบรนซึ่งเป็นวงก้นหอย (spiral) ของคลอเคลีย และสามารถวิเคราะห์ความแตกต่างของคลื่นเสียง 1,500 พิต (pitch) ดังแสดงในรูปที่ 2.7 ซึ่งเป็นที่ยอมรับกันว่าทฤษฎีตำแหน่งนั้น พิตเป็นส่วนกำหนดตำแหน่งความยาวของเซลล์ขนที่ถูกกระตุ้นด้วยเบสิลาร์เมมเบรน ส่วนทฤษฎีทางความถี่ของเบสิลาร์เมมเบรนสมมติว่า เซลล์ขนเคลื่อนที่ขึ้นและลงพร้อมกันกับการสั่นสะเทือนจากการบีบอัดคลื่นเสียง โดยจะเคลื่อนที่จากกระดูกโกลนไปยังหน้าต่างวงรี โดยแต่ละครั้งที่เซลล์ขนมีการเคลื่อนที่ขึ้นและลง จะมีผลต่อไฟริง 1 เส้นประสาท (one neural firing) ดังนั้นความถี่จะถูกเข้ารหัส

โดยตรงจากอัตราความเร็วขนในรูปหู ซึ่งจากรูปที่ 2.8 แสดงไคอะแกรมของเบซิลาร์เมมเบรน ตำแหน่งผลตอบสนองของคลื่นเสียงที่ความถี่ต่าง ๆ เมื่อพิจารณาจะเห็นว่าคลื่นเสียงที่มีความถี่ผ่านต่ำอยู่ด้านปลายของของเบซิลาร์เมมเบรน ส่วนของคลื่นเสียงที่มีความถี่ผ่านสูงนั้นจะอยู่ทางด้านฐานของเบซิลาร์เมมเบรน



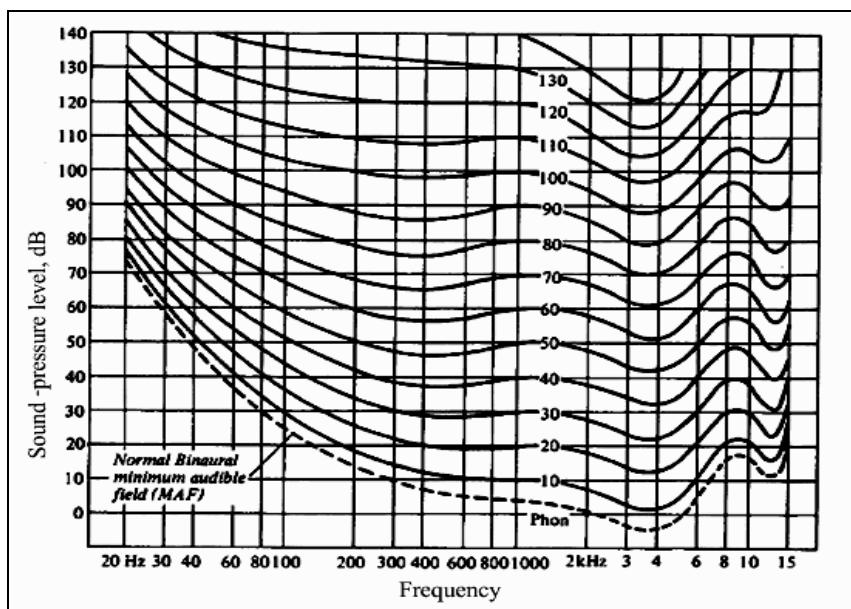
รูปที่ 2.8 ไคอะแกรมของเบซิลาร์เมมเบรน ตำแหน่งผลตอบสนองของคลื่นเสียงที่ความถี่ต่าง ๆ (Hz) (Loizou.,1998)

กระบวนการเกี่ยวกับการได้ยินมีความซับซ้อนต่อสมองมาก เนื่องจากจำกัดข้อมูลที่ส่งผ่านไปยังสมองโดยทางเส้นประสาท ซึ่งเส้นประสาทมาจากอิมพัลส์อิเล็กทริก (electrical impulses) ของคลอเคลียและอวัยวะหลอดกึ่งวง (semicircular canal) โดยทั้งสองถูกเชื่อมต่อกันด้วยพื้นที่เกี่ยวกับการได้ยินของสมอง

2.3.2 ความดังของเสียง

หูมนุษย์สามารถได้ยินเสียงขึ้นอยู่กับระดับความดังของเสียง ซึ่งอยู่ในช่วงความดังตั้งแต่ 0 ถึง 130 เดซิเบล และช่วงความถี่ 20 ถึง 20,000 เฮิรตซ์ ซึ่งสัมพันธ์กับการสั่นของเซลล์ขนภายในหูชั้นใน และมีความสัมพันธ์แบบไม่เป็นเส้นตรง กล่าวคือที่ระดับเสียงเท่ากัน แต่หูมนุษย์จะได้ยินเป็นความดังที่ไม่เท่ากันทุกความถี่ ซึ่งมีนักวิจัยหลายท่านได้พยายามอธิบาย ลักษณะของการได้ยินของหูมนุษย์นี้ ออกมาในรูปของกราฟความสัมพันธ์เพื่อให้เข้าใจถึงลักษณะการได้ยินของหูที่ระดับเสียงและความถี่ต่าง ๆ กัน ดังรูปที่ 2.9 แสดงระดับความดังของเสียงที่เท่ากันที่หูของมนุษย์

ได้ยิน พบว่าที่ความถี่ 20 เฮิรตซ์ ระดับเสียงต้องตั้ง 80 เฮิรตซ์ มนุษย์จะเริ่มได้ยินเสียง และที่ความถี่ 1000 เฮิรตซ์ มีความดังเท่ากับ 5 เดซิเบล ก็สามารถได้ยินเสียงนั้น



รูปที่ 2.9 ระดับความดังของเสียงที่เท่ากันที่หูของมนุษย์ได้ยิน (Lamancusa. J.S., 2000)

2.4 การประมวลผลสัญญาณ (Signal Processing)

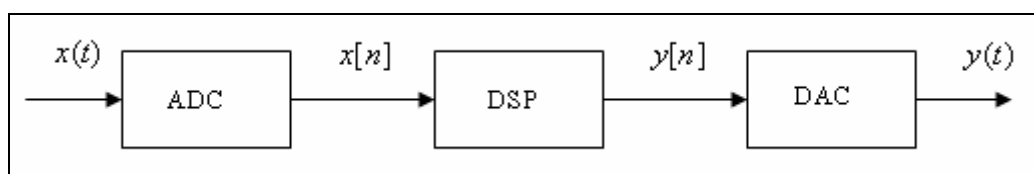
การประมวลผลสัญญาณคือ การนำสัญญาณมาปรับปรุงหรือเปลี่ยนรูป เพื่อประโยชน์ในการสื่อสารความหมาย การตีความ และการตัดสินใจ การควบคุมการประมวลผลสัญญาณนั้นอาจกระทำโดยตรงต่อสัญญาณ เช่น การขยายหรือลดทอน (expansion or attenuation) การกรอง ซึ่งอาจมีการเปลี่ยนรูปแบบของสัญญาณก่อน เช่น การประมวลผลในเชิงเลข และการประมวลผลในโดเมนที่แตกต่างออกไป ฟังก์ชันที่แม้จะมีการเปลี่ยนโดยการแทนสัญญาณ เช่น เปลี่ยนเป็นดิจิทัล หรือเปลี่ยนตามแวกซ์ในการประมวลผล เช่น กระทำในโดเมนอื่น สัญญาณจะยังคงเป็นสัญญาณเดิมอยู่อย่างครบถ้วน

2.4.1 โดเมน (Domain)

โดเมนในที่นี้หมายถึง สิ่งแวดล้อมในการอธิบายความหมายต่าง ๆ หรือก็คือตัวแปรนั่นเอง หากการอธิบายหรือการทำงานนั้นกระทำภายใต้ตัวแปรของเวลาคือ ฟังก์ชัน t จะเรียกว่า โดเมนทางเวลา (time domain) ถ้าหากการทำงานนั้นกระทำภายใต้ตัวแปรความถี่ คือฟังก์ชัน f หรือ ω จะเรียกว่า โดเมนทางความถี่หรือเฟรควเอนซีโดเมน (frequency domain) เป็นต้น โดยสัญญาณเดียวกันอาจมีคุณสมบัติบางอย่างที่สามารถมองเป็นเด่นชัดในอีกโดเมนหนึ่ง เช่น ในโดเมนทางเวลา

สามารถเห็นการเปลี่ยนแปลงของขนาดสัญญาณในขณะที่เวลาเปลี่ยนไปได้อย่างชัดเจน แต่จะบอกได้ยากกว่าสัญญาณนั้นมีความถี่อะไรบ้างเป็นองค์ประกอบ แต่ในโดเมนทางความถี่เราสามารถระบุได้ว่าสัญญาณนั้นประกอบด้วยสัญญาณความถี่อะไรบ้าง การเปลี่ยนรูปไปสู่โดเมนอื่นนั้นมีขั้นตอนสำคัญคือ การถ่ายโอน (transform)

2.4.2 โครงสร้างของระบบประมวลผลสัญญาณทางดิจิทัล (Structure of Digital Signal Processing)



รูปที่ 2.10 ระบบการประมวลผลสัญญาณดิจิทัล

จากรูปที่ 2.10 แสดงระบบการประมวลผลสัญญาณดิจิทัล โดยธรรมชาติของสัญญาณทั่วไป เป็นการเปลี่ยนแปลงของปริมาณกายภาพที่มีความต่อเนื่อง ทำให้การประมวลผลสัญญาณทางดิจิทัลจึงต้องอาศัยส่วนเปลี่ยนสัญญาณแอนะล็อกเป็นดิจิทัล (analog to digital converter หรือ ADC) ก่อน แล้วจึงทำการประมวลผลและเมื่อประมวลผลแล้ว จึงเปลี่ยนสัญญาณดิจิทัลเป็นแอนะล็อก (digital to analog converter หรือ DAC) อีกครั้งหนึ่ง

2.4.3 ตัวประมวลผลทางดิจิทัล

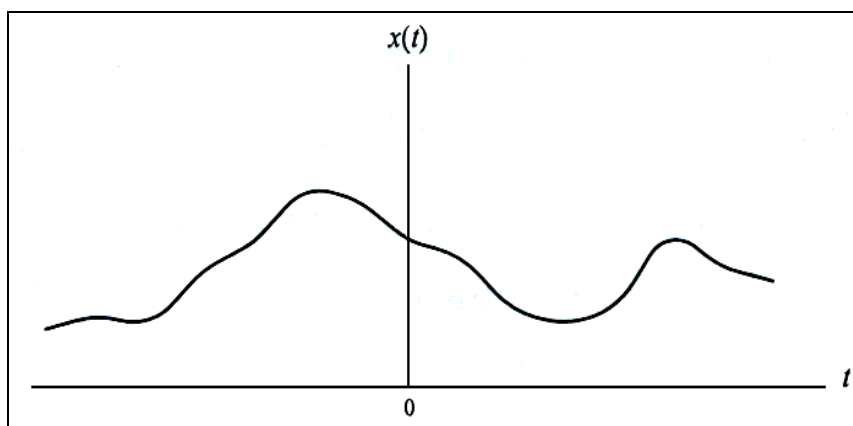
การประมวลผลสัญญาณในระบบดิจิทัล สามารถดำเนินการได้ทั้งทางฮาร์ดแวร์ (hardware) หรือซอฟต์แวร์ (software) หากต้องการประมวลผลโดยทางฮาร์ดแวร์ ตัวประมวลผลอาจมีหลายระดับขึ้นอยู่กับปัจจัยต่าง ๆ ดังนี้คือ วงจรพื้นฐาน เช่น ใช้เกทและสวิตช์ นอกจากนี้ยังมีส่วนประกอบของวงจรพื้นฐาน เช่น รีจิสเตอร์และวงจรวก เป็นต้น และอาจมีวงจรที่ซับซ้อนมากขึ้น เช่น วงจรคูณ วงจรการคำนวณทางคณิตศาสตร์ วงจรควบคุมอื่น ๆ เช่น สัญญาณนาฬิกา วงจรอินเตอร์เฟส และวงจรแปลงแอนะล็อกและดิจิทัล

2.4.4 สัญญาณและระบบ

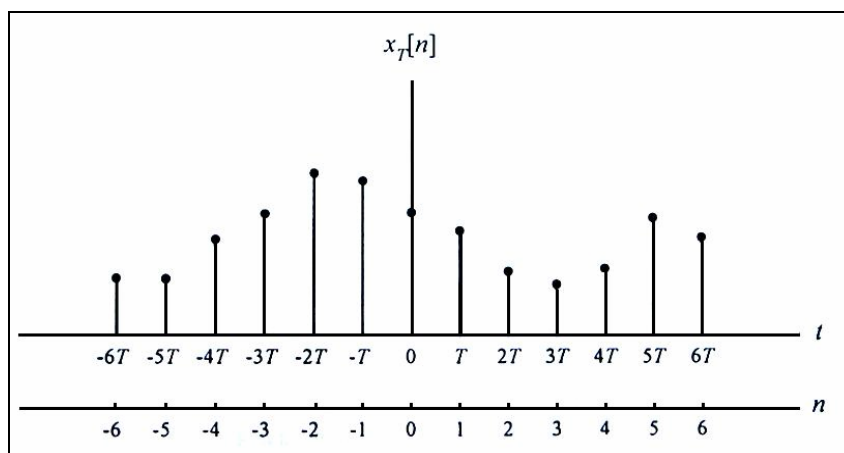
สัญญาณที่พบเห็นในทางปฏิบัติมีอยู่มากมายหลากหลายลักษณะ ดังนั้นการจัดแบ่งสัญญาณออกเป็นประเภทที่มีคุณลักษณะร่วมกันจึงเป็นประโยชน์ต่อการวิเคราะห์สัญญาณ และเข้าใจถึงคุณสมบัติเฉพาะตัวของมันได้อย่างเป็นระบบ โดยทั่วไปจะแบ่งสัญญาณออกเป็น 2 ประเภท

อย่างกว้าง ๆ คือ สัญญาณที่มีความต่อเนื่องทางเวลา (continuous-time signals) และสัญญาณที่มีค่า discrete ทางเวลา (discrete-time signals)

การแบ่งในลักษณะนี้พิจารณาจากค่าของสัญญาณที่แปรเปลี่ยนในเชิงเวลา ซึ่งจากรูปที่ 2.11 แสดงรูปแบบสัญญาณที่มีความต่อเนื่องทางเวลา เป็นสัญญาณที่มีค่าต่อเนื่องตลอดทุกช่วงเวลา แทนสัญลักษณ์ x และแทนเวลาด้วยสัญลักษณ์ t กล่าวได้ว่า x เป็นฟังก์ชันของ t หรือ x มีค่าที่เวลา t ใด ๆ เขียนแทนสัญญาณได้ว่า $x(t)$ ส่วนรูปที่ 2.12 เป็นสัญญาณที่มีค่า discrete ทางเวลา เป็นสัญญาณที่มีค่าเฉพาะบางจุดเวลา โดยทั่วไปเกิดจากการสุ่มสัญญาณต่อเนื่องด้วยคาบเวลาของการสุ่มคงที่ ใช้สัญลักษณ์ n แทนเวลาแบบ discrete



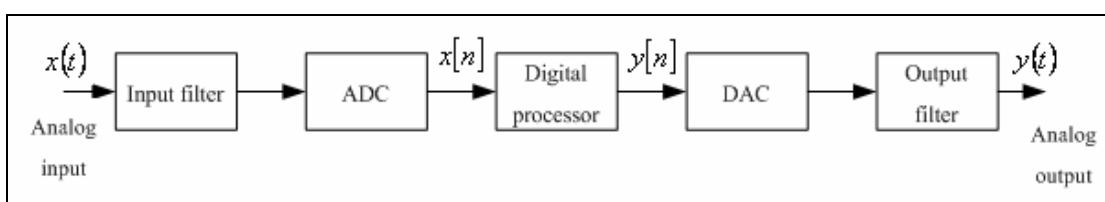
รูปที่ 2.11 สัญญาณที่มีค่าต่อเนื่องทางเวลา



รูปที่ 2.12 สัญญาณที่มีค่า discrete ทางเวลา

2.5 ทฤษฎีการออกแบบตัวกรอง

ตัวกรองดิจิทัล (digital filter) เป็นตัวกรองรูปแบบหนึ่งที่ถูกออกแบบและสร้างขึ้นโดย วงจรดิจิทัลมีบล็อกไดอะแกรมการทำงานดังรูปที่ 2.13 แสดงสัญญาณแอนะล็อก ให้ $x(t)$ เป็น สัญญาณอินพุตของตัวกรอง ซึ่งสัญญาณนี้จะถูกส่งผ่านวงจรแปลงสัญญาณแอนะล็อกให้เป็น สัญญาณดิจิทัล คือ $x[n]$ และสัญญาณดิจิทัลนี้ถูกส่งเข้าไปคำนวณในหน่วยประมวลผล และผลที่ ได้จากการคำนวณยังคงเป็นสัญญาณดิจิทัล คือ $y[n]$ จะถูกส่งผ่านมายังวงจรแปลงสัญญาณ เอาท์พุทของตัวกรองเป็นสัญญาณแอนะล็อก คือ $y(t)$



รูปที่ 2.13 แผนภาพบล็อกการทำงานของตัวกรองดิจิทัลพื้นฐาน

ชนิดของตัวกรองแบ่งออกเป็น 2 ชนิด คือ ตัวกรองดิจิทัลแบบเอฟไออาร์ (finite impulse response filter หรือ FIR) และตัวกรองดิจิทัลแบบไอไออาร์ (infinite impulse response filter หรือ IIR) ตัวกรองดิจิทัลทั้งสองชนิดมีพื้นฐานการทำงาน สามารถแสดงโดยลำดับผลตอบสนองต่อ สัญญาณอิมพัลส์ (impulse response) $h[k]$ ($k=0,1,\dots$) ซึ่งมีสัญญาณอินพุตและสัญญาณ เอาท์พุทของตัวกรองที่มีความสัมพันธ์กัน โดยผลรวมคอนโวลูชัน (convolution) ดังสมการที่ (2.5) เป็นสมการของตัวกรองดิจิทัลแบบเอฟไออาร์ ส่วนสมการที่ (2.6) เป็นสมการของตัวกรองดิจิทัล แบบไอไออาร์

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k] \quad (2.5)$$

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k] \quad (2.6)$$

ตัวกรองดิจิทัลแบบเอฟไออาร์ มีผลตอบสนองต่อสัญญาณอิมพัลส์ $h[k]$ มีความยาว M ค่าส่วนตัวกรองดิจิทัลแบบไอไออาร์มีผลตอบสนองต่อสัญญาณอิมพัลส์ไม่จำกัด จากสมการ(2.6) จัดสมการตัวกรองดิจิทัลแบบไอไออาร์เพื่อแสดงรีเคอร์ซีฟ (recursive) ได้สมการ (2.7)

$$\begin{aligned} y[n] &= \sum_{k=0}^{\infty} h[k]x[n-k] \\ &= \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k] \end{aligned} \quad (2.7)$$

โดย a_k และ b_k เป็นสัมประสิทธิ์ของตัวกรอง ดังนั้นจากสมการที่ (2.5) และ (2.6) เป็นสมการของตัวกรองดิจิทัลแบบเอฟไออาร์และตัวกรองดิจิทัลแบบไอไออาร์ตามลำดับ ซึ่งสมการที่ (2.7) ระบุว่าสัญญาณเอาต์พุต $y[n]$ ณ เวลาปัจจุบันมีค่าขึ้นอยู่กับสัญญาณอินพุต ณ ปัจจุบัน คือ $x(n)$ และสัญญาณอินพุตในอดีตอีก M ค่าได้แก่ $x[n-1], x[n-2], \dots, x[n-M]$ และสัญญาณเอาต์พุตในอดีตอีก N ค่าซึ่งประกอบด้วย $y[n-1], y[n-2], \dots, y[n-N]$ ซึ่งเรียกระบบนี้ว่าระบบเชิงเส้นชนิดที่ไม่แปรเปลี่ยนตามเวลา (linear time-invariant system) เนื่องจากสมการที่ (2.7) ถ้าให้ a_k เป็นศูนย์ จะได้สมการตัวกรองดิจิทัลแบบเอฟไออาร์ซึ่งเป็นแบบนอนรีเคอร์ซีฟ (non-recursive) เขียนเป็นสมการ (2.8) คือ

$$y[n] = \sum_{k=0}^M b_k x[n-k] \quad (2.8)$$

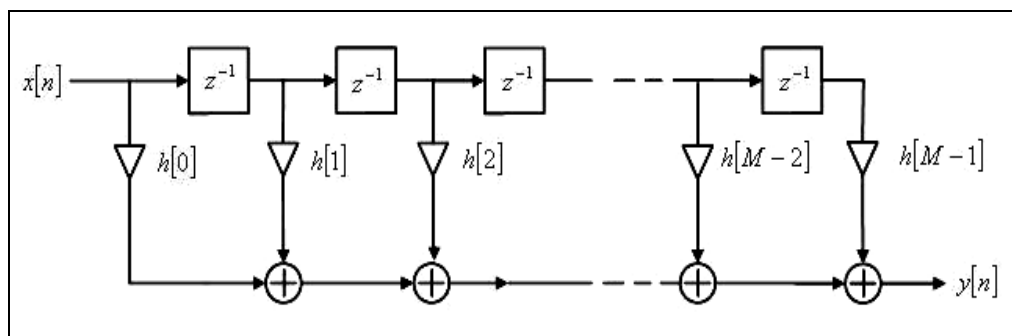
โดยที่สัญญาณเอาต์พุต $y[n]$ ขึ้นอยู่กับสัญญาณอินพุต ณ เวลาปัจจุบันและสัญญาณอินพุตในอดีตอีก N ค่าเท่านั้น

2.5.1 ตัวกรองดิจิทัลแบบเอฟไออาร์

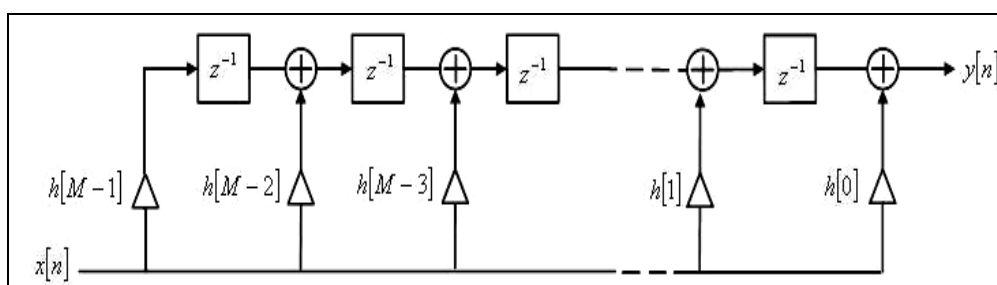
จากสมการที่ (2.5) สามารถอธิบายคุณลักษณะของผลตอบสนองอิมพัลส์ โดยมีความสัมพันธ์กับฟังก์ชันถ่ายโอนดังนี้

$$H(z) = \sum_{k=0}^{M-1} h[k]z^{-k} \quad (2.9)$$

สำหรับการสร้างวงจรให้มีคุณสมบัติตรงตามนิยามของตัวกรองดิจิทัลแบบเอฟไออาร์ สามารถทำได้หลายลักษณะ ได้แก่ โครงสร้างตัวกรองเอฟไออาร์รูปแบบโดยตรง คือการสร้างตัวกรองเอฟไออาร์ที่มีความยาวเท่ากับ M ซึ่งหมายถึง มีจำนวนสัมประสิทธิ์ทั้งสิ้น M ตัว หรือตัวกรองมีอันดับเท่ากับ $M - 1$ โดยอาศัยรูปแบบโดยตรงนั้น จะประกอบด้วยอุปกรณ์หน่วยเวลาจำนวน $M - 1$ ตัว วงจรคูณ M ชุดและวงจรวกอีก $M - 1$ ชุด ซึ่งเป็นคุณลักษณะเฉพาะของโครงสร้างรูปแบบโดยตรงนั้นอยู่ตรงที่สัมประสิทธิ์การคูณ จะเป็นค่าเดียวกับสัมประสิทธิ์ของฟังก์ชันถ่ายโอนของวงจรรอง ดังรูปที่ 2.14 แสดงโครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่หนึ่ง และรูปที่ 2.15 แสดงโครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่สอง โดยทั้งสองรูปให้ผลการทำงานเหมือนกัน แต่มีการจัดวางอุปกรณ์ที่แตกต่างกันตามลำดับ



รูปที่ 2.14 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่หนึ่ง



รูปที่ 2.15 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบโดยตรงแบบที่สอง

ส่วนโครงสร้างตัวกรองเอฟไออาร์รูปแบบคาตเคสคือ ในกรณีตัวกรองเอฟไออาร์มีอันดับสูง ๆ สามารถแยกตัวประกอบฟังก์ชันถ่ายโอนออกเป็นส่วนย่อย ๆ ที่มีอันดับต่ำลงได้ โดย

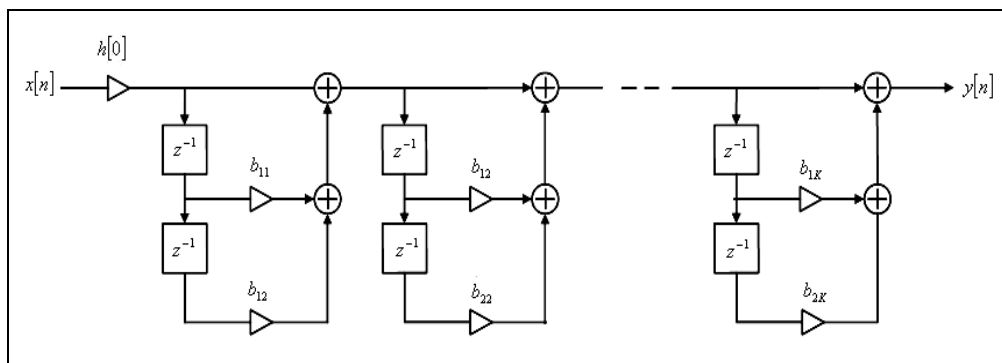
ใช้ฟังก์ชันถ่ายโอนแต่ละส่วนย่อยให้เหลือเป็นอันดับเท่ากับหนึ่งหรือสองเท่านั้น ดังนั้นจากสมการที่ (2.9) สามารถเขียนให้อยู่ในรูปของตัวประกอบได้สมการที่ (2.10)

$$H(z) = h[0] \prod_{k=1}^K (1 + b_{1k} z^{-1} + b_{2k} z^{-2}) \quad (2.10)$$

โดย $K = (M - 1)/2$ ถ้า M เป็นจำนวนคี่ และ $K = M/2$ ถ้า M เป็นจำนวนคู่ เมื่อให้สัมประสิทธิ์ $b_{2K} = 0$ เขียนสมการได้ดังนี้

$$H(z) = h[0] \prod_{k=1}^K (1 + b_{1k} z^{-1}) \quad (2.11)$$

จากสมการ (2.11) เมื่อทำการแยกตัวประกอบ สามารถนำมาเขียนเป็นโครงสร้างตัวกรองเอฟไออาร์แบบคาสเคดดังรูปที่ 2.16 พบว่าโครงสร้างย่อยแต่ละส่วนทำหน้าที่แทนฟังก์ชันถ่ายโอนย่อยที่แยกตัวประกอบออกมา ซึ่งในรูปนี้อาศัยโครงสร้างรูปแบบย่อยโดยตรงแบบที่หนึ่งเป็นพื้นฐาน อย่างไรก็ตามสามารถเลือกใช้โครงสร้างรูปแบบโดยตรงแบบที่สองด้วยเช่นกัน



รูปที่ 2.16 โครงสร้างตัวกรองเอฟไออาร์ รูปแบบคาสเคด

โครงสร้างตัวกรองเอฟไออาร์โครงสร้างสุดท้ายคือ โครงสร้างตัวกรองเอฟไออาร์แบบเฟสเชิงเส้นคือ ในการสร้างตัวกรองเอฟไออาร์เพื่อให้มีผลตอบสนองทางเฟสเป็นแบบเชิงเส้นจัดว่าเป็นเรื่องสำคัญสำหรับการประยุกต์ใช้งานบางประเภท เช่น เสียงดนตรี วิดีทัศน์ และการส่งผ่านข้อมูล เป็นต้น ดังนั้นการศึกษาและการเข้าใจในตัวกรองเอฟไออาร์ ซึ่งมีคุณสมบัติเฉพาะ

ประเภทนี้ ในรายละเอียดจึงเป็นเรื่องที่เป็นประโยชน์ต่อการใช้งานในทางปฏิบัติ ตัวกรองเอฟไออาร์ มีผลตอบสนองแบบเฟสเชิงเส้น จะก่อให้เกิดการหน่วงเวลากับองค์ประกอบแต่ละความถี่ของ สัญญาณอินพุท ในปริมาณที่เพิ่มขึ้นตามค่าความถี่เชิงเส้น สำหรับเงื่อนไขที่ทำให้ตัวกรองเอฟไออาร์ มีคุณสมบัติเชิงเส้นตามต้องการคือ ฟังก์ชันตอบสนองมีความสมมาตร (symmetric) ดังนี้

$$h[n] = h[M-1-n] \quad n = 0, 1, 2, \dots, M-1 \quad (2.12)$$

หรือฟังก์ชันตอบสนองอิมพัลส์มีความปฏิสมมาตร (antisymmetric)

$$h[n] = -h[M-1-n] \quad n = 0, 1, 2, \dots, M-1 \quad (2.13)$$

ประเด็นที่สนใจในที่นี้คือ การที่สามารถออกแบบโครงสร้างเฉพาะที่ช่วยลดหรือประหยัดจำนวน อุปกรณ์ที่ใช้ในการสร้างวงจรตัวกรองเอฟไออาร์แบบเฟสเชิงเส้น เมื่อเทียบกับการใช้โครงสร้าง รูปแบบโดยตรงได้ เนื่องจากวงจรตัวกรองเอฟไออาร์แบบเฟสเชิงเส้นนั้น มีคุณสมบัติที่พิเศษกว่า วงจรตัวกรองเอฟไออาร์ทั่วไป ทำให้สามารถออกแบบโครงสร้างสำหรับวงจรตัวกรองเอฟไออาร์ ชนิดนี้ ประหยัดตัวคูณลงได้ประมาณครึ่งหนึ่ง ซึ่งในการอธิบายโครงสร้างวงจรกรองแบบเชิงเส้นจะ อาศัยตัวอย่างของฟังก์ชันตอบสนองอิมพัลส์ ชนิดมีความสมมาตรที่มีความยาว M เท่ากับ 6 และ M เท่ากับ 7 พิจารณากรณีแรกเมื่อ M เท่ากับ 6 ฟังก์ชันถ่ายโอนของวงจรกรองสามารถเขียนได้เป็น

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} \quad (2.14)$$

สำหรับวงจรกรองเฟสเชิงเส้นสามารถใช้สมการที่ (2.10) มาช่วยลดรูปในสมการที่ (2.14) ได้เป็น สมการที่ (2.15)

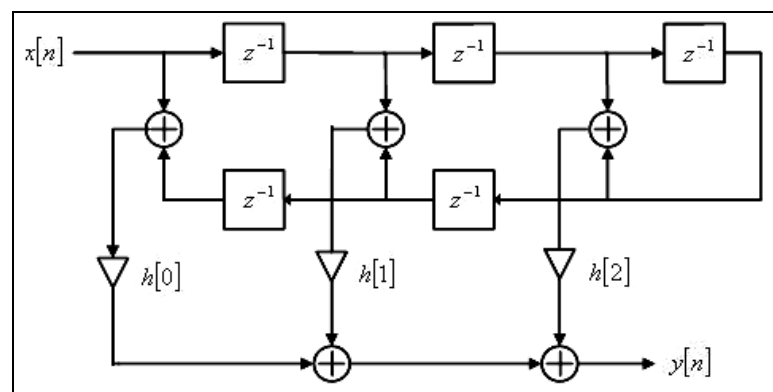
$$H(z) = h[0](1+z^{-5}) + h[1](z^{-1}+z^{-4}) + h[2](z^{-2}+z^{-3}) \quad (2.15)$$

ความสัมพันธ์ที่ได้มีโครงสร้างวงจรกรองเฟสเชิงเส้นได้ดังรูป 2.17 แสดงตัวกรองเอฟไออาร์ M เท่ากับ 6 ซึ่งโครงสร้างนี้ใช้อุปกรณ์การคูณเพียง 3 ชุด ในขณะที่ถ้าเป็นโครงสร้างรูปแบบโดยตรงต้อง ใช้อุปกรณ์การคูณถึง 5 ชุด สำหรับกรณี M เท่ากับ 7 ฟังก์ชันถ่ายโอนของวงจรกรองสามารถเขียน เป็นสมการได้ดังนี้

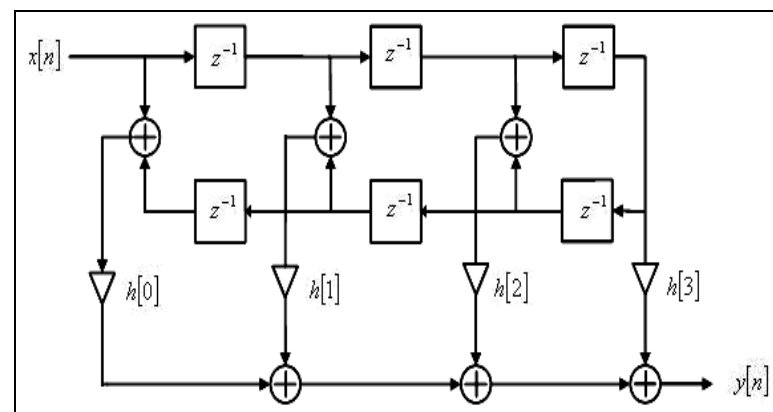
$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[6]z^{-6} \quad (2.16)$$

เมื่อลดรูปสมการโดยอาศัยคุณสมบัติของวงจรรองเชิงเส้นจะได้สมการที่ (2.17)

$$H(z) = h[0](1 + z^{-6}) + h[1](z^{-1} + z^{-5}) + h[2](z^{-2} + z^{-4}) + h[3]z^{-3} \quad (2.17)$$



รูปที่ 2.17 ตัวกรองเอฟไออาร์ M เท่ากับ 6



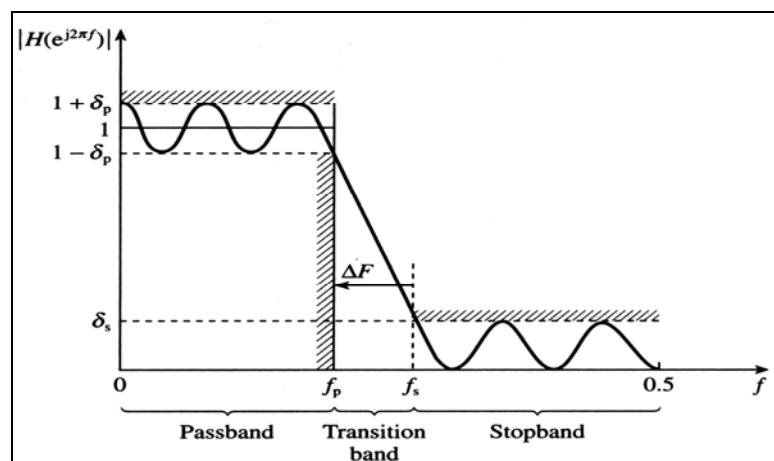
รูปที่ 2.18 ตัวกรองเอฟไออาร์ M เท่ากับ 7

ความสัมพันธ์ที่ได้มีโครงสร้างตัวกรองเฟสเชิงเส้นได้ดังในรูปที่ 2.18 แสดงตัวกรองเอฟไออาร์ M เท่ากับ 7 ซึ่งจะเห็นว่าโครงสร้างนี้ใช้อุปกรณ์การคูณเพียง 4 ชุด แต่ในขณะที่ถ้าเป็น

โครงสร้างแบบโดยตรงจะต้องใช้อุปกรณ์การคูณ 6 ชุด ตัวอย่างที่กล่าวมานี้แสดงให้เห็นว่า ถ้าตัวกรองเอฟไออาร์ มีคุณสมบัติทางเฟสเชิงเส้นแล้ว สามารถใช้งานโครงสร้างที่มีจำนวนอุปกรณ์การคูณลดลง

2.5.2 คุณสมบัติลักษณะตัวกรองเอฟไออาร์

ในทางปฏิบัติเป็นเรื่องปกติต้องมีการระบุคุณสมบัติของตัวกรองที่ต้องการ เพื่อใช้เป็นข้อกำหนดในการออกแบบ แน่แน่นอนว่าการกำหนดคุณสมบัติลักษณะในรายละเอียดมีปัจจัยที่ต้องพิจารณาอยู่หลายประการ อย่างไรก็ตามข้อกำหนดนี้จัดว่าเป็นหัวใจของตัวกรองที่ต้องให้ความสนใจเป็นอันดับแรก คือ ผลตอบสนองเชิงขนาด (magnitude response) และผลตอบสนองทางเฟส (phase response) โดยส่วนมากให้ความสำคัญผลตอบสนองเชิงขนาดมาก ซึ่งยอมให้ผลตอบสนองเฟสเป็นอย่างไรก็ได้ แต่งานบางลักษณะของผลตอบสนองทั้งสองมีความสำคัญมาก เช่น การประมวลผลคลื่นสัญญาณไฟฟ้าหัวใจ โดยลำดับต่อไปจะอธิบายถึงกรรมวิธีการเขียนข้อกำหนดของตัวกรองในส่วนของผลตอบสนองเชิงขนาด กำหนดคุณลักษณะของผลตอบสนองเชิงขนาดสำหรับตัวกรองผ่านต่ำ โดยแสดงในรูปที่ 2.19 ประกอบด้วยพารามิเตอร์ δ_p คือความพลิวของผ่านแถบ (passband ripple) δ_s คือการลดทอนของแถบหยุด (stopband attenuation) f_p คือความถี่ขอบผ่านแถบ (passband edge frequency) และ f_s คือความถี่ขอบแถบหยุด (stopband edge frequency) ส่วนพารามิเตอร์ F_s คือความถี่การสุ่ม (sampling frequency) และ ΔF คือแถบเปลี่ยน (transition band)



รูปที่ 2.19 การกำหนดคุณลักษณะของตัวกรองผ่านต่ำ (Emmanue and Barrie., 2002)

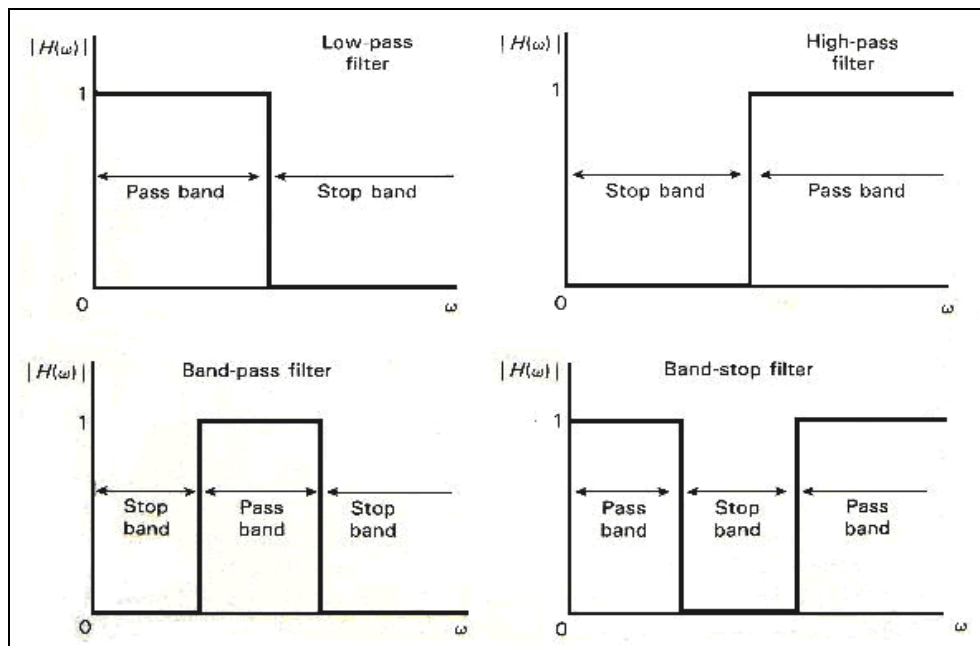
จากรูปสังเกตว่าได้มีการแบ่งคุณสมบัติทางความถี่ออกเป็นสามส่วน ได้แก่ ผ่านแถบ (passband) แถบเปลี่ยน (transition band) แถบหยุด (stopband) เมื่อพิจารณาช่วงของผ่านแถบก่อน จะ

เห็นว่าถูกกำหนดด้วยพารามิเตอร์สองตัวคือ δ_p และ f_p ซึ่งพารามิเตอร์ทั้งสองกำกับว่า เมื่อป้อนสัญญาณอินพุตสู่ตัวกรองผ่านต่ำแล้ว องค์ประกอบความถี่ตั้งแต่กระแสตรงไปจนถึงความถี่ f_p จะได้รับการส่งผ่านโดยมีค่าการผลตอบแทนเชิงขนาดอยู่ในช่วง $1 - \delta_p$ และ $1 + \delta_p$ เมื่อพิจารณาช่วงถัดมาซึ่งเป็นช่วงแถบเปลี่ยนนั้น จะมีพารามิเตอร์ที่กำกับอยู่สองตัวคือ f_p และ f_s ซึ่งความถี่ทั้งสองจะกำหนดช่วงความกว้างของความถี่สูงสุดที่ตัวกรองต้องเปลี่ยน และจากการทำงานในช่วงของผ่านแถบไปสู่ช่วงของแถบหยุด ตัวกรองที่ดีต้องกำหนดให้ช่วงเปลี่ยน ΔF มีขนาดเล็ก สำหรับช่วงสุดท้ายซึ่งเป็นช่วงของแถบหยุด ในช่วงนี้กำหนดด้วยพารามิเตอร์สองตัว ได้แก่ f_s และ δ_s โดยพารามิเตอร์ f_s เป็นตัวกำหนดค่าความถี่แรกที่ตัวกรองต้องทำงานในช่วงของแถบหยุด และส่วนของ δ_s เป็นค่าผลตอบแทนเชิงขนาด $|H(e^{j2\pi f})|$ สูงสุดที่ตัวกรองยอมรับได้สำหรับองค์ประกอบความถี่ของสัญญาณอินพุตที่สูงกว่า f_s

2.5.3 วิธีการใช้หน้าต่าง

วิธีการออกแบบโดยใช้หน้าต่างมีหลักการง่าย ๆ คือ ให้เริ่มต้นจากผลตอบแทนความถี่ที่สนใจ ได้แก่ ตัวกรองผ่านต่ำ (low-pass filter) ตัวกรองผ่านสูง (high-pass filter) ตัวกรองผ่านแถบ (band-pass filter) และตัวกรองหยุดแถบ (band-stop filter) ดังแสดงในรูปที่ 2.20 จากนั้นให้พิจารณาค่าผลตอบแทนอิมพัลส์ของตัวกรองดังกล่าว ซึ่งโดยทั่วไปแล้วตัวกรองอุดมคติเหล่านี้ล้วนแล้วแต่มีผลตอบแทนอิมพัลส์ที่ครอบคลุมช่วงเวลาที่เป็นอนันต์ จึงไม่สามารถสร้างได้จริงในทางปฏิบัติ ดังนั้นเพื่อให้สามารถใช้งานได้จริงจึงได้มีการเสนอนำหน้าต่างมาครอบ เพื่อให้ผลตอบแทนอิมพัลส์มีค่าจำกัดอยู่เฉพาะภายในหน้าต่างเท่านั้น การทำเช่นนี้ทำให้สัมประสิทธิ์ของตัวกรองที่สร้างขึ้นมีจำนวนที่จำกัด แต่จะส่งผลให้ผลตอบแทนความถี่ผิดเพี้ยนหรือเปลี่ยนไปจากค่าอุดมคติ ซึ่งการออกแบบ โดยวิธีการใช้หน้าต่างจึงเทียบได้กับการประมาณผลตอบแทนอิมพัลส์ของตัวกรองอุดมคติ

ในการอธิบายวิธีการออกแบบตัวกรองเอฟไออาร์โดยวิธีการใช้หน้าต่าง จะอาศัยตัวอย่างตัวกรองผ่านต่ำในการอธิบาย ซึ่งการเลือกอธิบายตัวกรองผ่านต่ำเป็นสื่อในการอธิบาย เพราะตัวกรองอื่น ๆ ในเชิงคณิตศาสตร์สามารถเขียนได้ในรูปของตัวกรองผ่านต่ำ



รูปที่ 2.20 คุณลักษณะของตัวกรองในอุดมคติ (Leslie Balmer., 1997)

เนื่องจากตัวกรองอื่น ๆ ในเชิงคณิตศาสตร์สามารถเขียนได้ในรูปของตัวกรองผ่านต่ำ ซึ่งในทางอุดมคติตัวกรองผ่านต่ำมีคุณสมบัติทางความถี่ในรูปที่ 2.21 โดย ω_c แทนความถี่ตัด (cutoff frequency) ส่วนผลตอบสนองอิมพัลส์หาได้จากการแปลงฟูริเยร์ผกผันของฟังก์ชันถ่ายโอน เขียนเป็นสมการได้ดังนี้

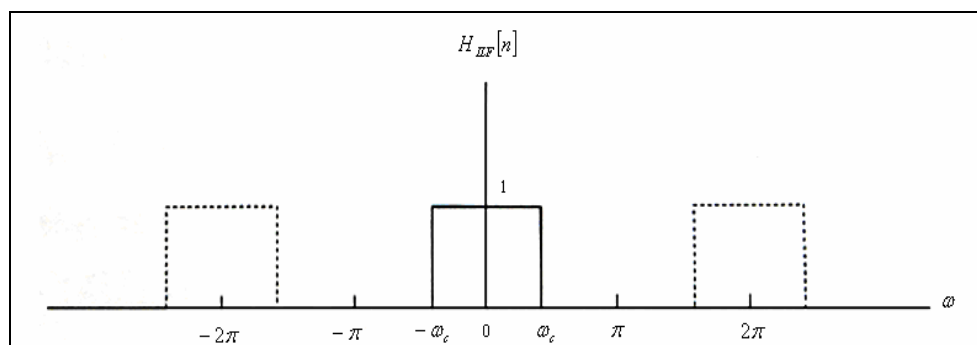
$$h_{ILF}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{ILF}(\omega) e^{j\omega n} d\omega \quad (2.18)$$

$$= \frac{2f_c \sin(n\omega_c)}{n\omega_c} \quad n \neq 0, -\infty \leq n \leq \infty \quad (2.19)$$

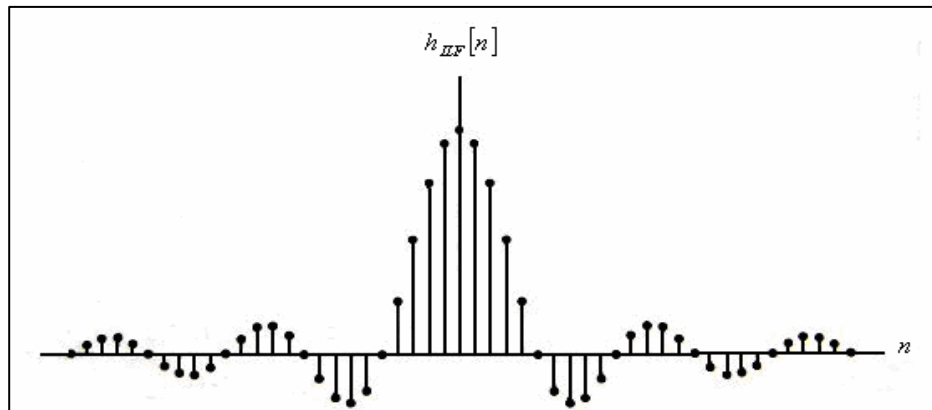
จากรูปที่ 2.22 แสดงผลตอบสนองอิมพัลส์ของตัวกรองผ่านต่ำในอุดมคติ ที่ได้จากสมการ (2.19) พบว่าผลตอบสนองอิมพัลส์ของตัวกรองผ่านต่ำในอุดมคติ $h_{ILF}[n]$ มีค่าตลอดทุกช่วงเวลา $-\infty$ ถึง ∞ ซึ่งไม่สอดคล้องกับโครงสร้างตัวกรองเอฟไออาร์ ที่จะมีสัมประสิทธิ์จากผลตอบสนองอิมพัลส์จำนวนจำกัด ฉะนั้นจึงเป็นไปได้ที่จะสร้างตัวกรองผ่านต่ำอุดมคติด้วยโครงสร้างตัวกรองเอฟไออาร์ อย่างไรก็ตามหากทำการปรับให้ผลตอบสนองอิมพัลส์ของตัวกรอง

ผ่านค่ามีจำนวนจำกัด เหลือเท่ากับความยาวของตัวกรองเอฟไออาร์ที่เลือกใช้ ในที่นี้กำหนดให้มีความยาวเท่ากับ M และผลตอบสนองอิมพัลส์ช่วงอื่น ๆ กำหนดให้มีค่าเป็นศูนย์ทั้งหมด ทำให้ผลลัพธ์ที่ได้มีค่าลักษณะดังรูปที่ 2.23 ก แสดงผลตอบสนองอิมพัลส์ของตัวกรองเอฟไออาร์ โดย M เท่ากับ 25 ซึ่งในรูปเป็นกรณีที่มีสัมประสิทธิ์เหลือเพียง $M = 25$ ส่วนในรูปที่ 2.23 ข. แสดงฟังก์ชันถ่ายโอนของตัวกรองเอฟไออาร์ที่ได้จากผลตอบสนองอิมพัลส์ ซึ่งเป็นการคำนวณหาฟังก์ชันถ่ายโอน $H(\omega)$ จะเห็นว่ามีคุณลักษณะที่ผิดเพี้ยนไปจากวงจรตัวกรองผ่านต่ำในอุดมคติ ทั้งนี้การเพิ่มจำนวนสัมประสิทธิ์ หรือความยาวของตัวกรองมากขึ้นจะช่วยให้ฟังก์ชันถ่ายโอนนั้น มีลักษณะใกล้เคียงกับตัวกรองผ่านต่ำในอุดมคติมากขึ้น

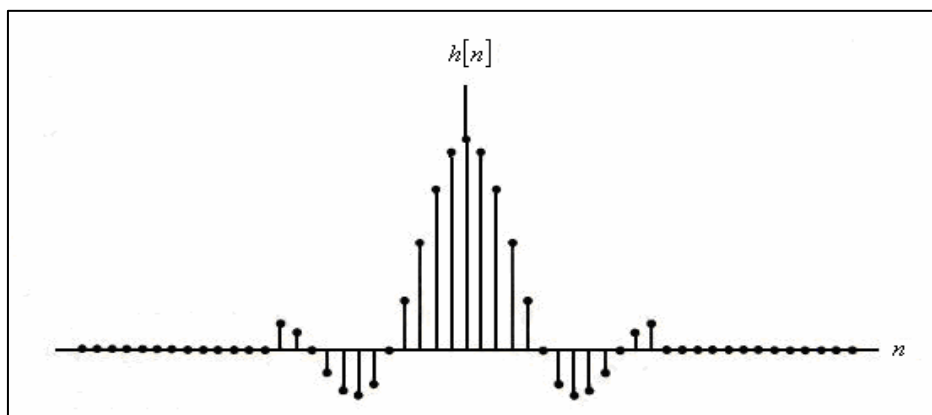
อย่างไรก็ตามจะสังเกตเห็นว่าตัวกรองที่ได้ไม่มีคุณสมบัติคอซอล เพราะผลตอบสนองอิมพัลส์ $h[n]$ มีค่าไม่เป็นศูนย์เมื่อ $n < 0$ หากต้องการตัวกรองที่มีคุณสมบัติคอซอลด้วย ให้ทำการเลื่อนผลตอบสนองอิมพัลส์ไปทางขวามือจำนวน $(M - 1)/2$ ค่า โดยรูปที่ 2.24 แสดงผลตอบสนองอิมพัลส์ของตัวกรองเอฟไออาร์ซึ่งมีคุณสมบัติคอซอล และเมื่อพิจารณาจะเห็นว่า การเลื่อนทางเวลาไม่ส่งผลกระทบต่อสเปกตรัมเชิงขนาดของตัวกรองแต่อย่างใด หากแต่คุณลักษณะทางเฟสของตัวกรองจะมีการเปลี่ยนแปลง แต่ตัวกรองในที่นี้มีคุณสมบัติของตัวกรองเฟสเชิงเส้น เพราะผลตอบสนองอิมพัลส์ตรงตามเงื่อนไขในสมการที่ (2.12)



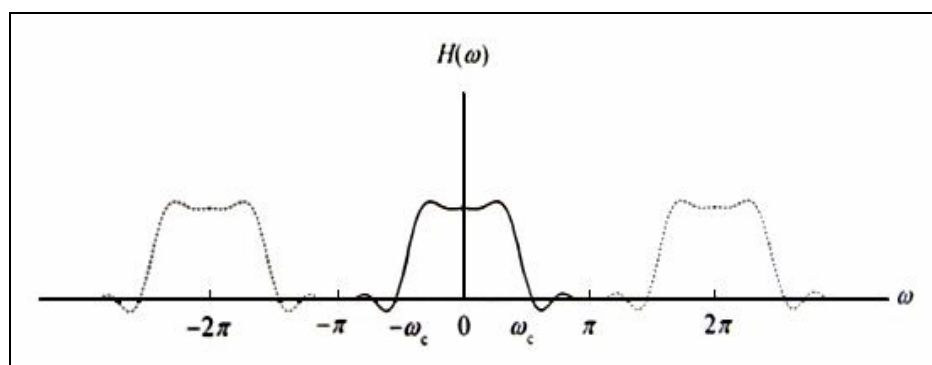
รูปที่ 2.21 ฟังก์ชันถ่ายโอนของตัวกรองผ่านต่ำ



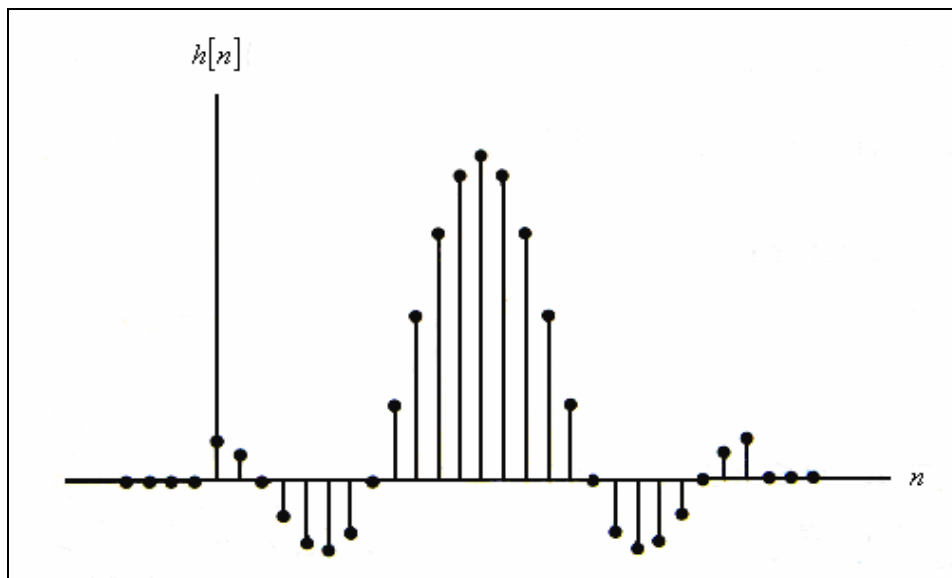
รูปที่ 2.22 ผลตอบสนองอิมพัลส์ของตัวกรองผ่านต่ำในอุดมคติ



รูปที่ 2.23 ก. ผลตอบสนองอิมพัลส์ของตัวกรองเอพ็ไออาร์ โดย M เท่ากับ 25



รูปที่ 2.23 ข. ฟังก์ชันถ่ายโอนของตัวกรองเอพ็ไออาร์ที่ได้จากผลตอบสนองอิมพัลส์



รูปที่ 2.24 ผลตอบสนองอิมพัลส์ของตัวกรองเฟอไออาร์ มีคุณสมบัติกอสเซล

ขั้นตอนการออกแบบให้ได้ตัวกรองผ่านต่ำ $h[n]$ ที่มีคุณสมบัติในข้างต้นเทียบได้กับการนำฟังก์ชัน $w[n]$ ที่ทำหน้าที่เหมือนหน้าต่างมาคูณกับผลตอบสนองอิมพัลส์ของตัวกรองผ่านต่ำ $h_{ILF}[n]$ ดังนี้คือ

$$h[n] = h_{ILF}[n]w[n] \quad (2.20)$$

โดยที่

$$w[n] = \begin{cases} 1, & |n| \leq (M-1)/2 \\ 0, & \text{other} \end{cases} \quad (2.21)$$

ดังนั้นให้ทำการเลื่อนทางเวลาฟังก์ชัน $h[n]$ ไปเท่ากับ $(M-1)/2$ จังหวะของเวลา เพื่อให้ได้เป็นตัวกรองผ่านต่ำที่มีคุณสมบัติกอสเซล การนำหน้าต่างสี่เหลี่ยมมาใช้ในการจำกัดช่วงของผลตอบสนองอิมพัลส์นั้น แม้ว่าจะกระทำได้ง่าย หากแต่หน้าต่างสี่เหลี่ยมส่งผลให้ฟังก์ชันถ่ายโอนที่ได้มีลักษณะของความพลัว (ripple) เกิดขึ้นดังรูปที่ 2.24 เป็นความพลัวนี้จะมีเกิดขึ้นแม้ว่าจะเพิ่มความยาวให้กับตัวกรอง และปัญหานี้รู้จักกันในชื่อปรากฏการณ์กิบบ์ (gibb's phenomenon)

ด้วยเหตุนี้จึงได้พัฒนาหน้าต่างที่มีคุณลักษณะแตกต่างไปจากหน้าต่างสี่เหลี่ยม เพื่อให้ได้ฟังก์ชันถ่ายโอนของตัวกรองที่มีคุณสมบัติดีขึ้น และตรงตามต้องการมากขึ้น

2.5.4 หน้าต่างที่น่าสนใจ

งานวิจัยที่ผ่านมาได้มีการพัฒนารูปแบบหน้าต่างที่มีคุณสมบัติ และมีประโยชน์สำหรับใช้งาน ในที่นี้จะกล่าวถึงหน้าต่างที่สำคัญ 4 รูปคือ หน้าต่างสี่เหลี่ยม (Rectangular) หน้าต่างแฮนนิ่ง (Hanning) หน้าต่างแฮมมิง (Hamming) และหน้าต่างแบล็กแมน (Blackman) สามารถสรุปเป็นตารางและค่าพารามิเตอร์ที่สำคัญในการออกแบบ

ตารางที่ 2.2 หน้าต่างแบบต่าง ๆ กัน และค่าพารามิเตอร์ที่สำคัญ

หน้าต่าง	ความกว้างแถบ เปลี่ยน(Hz)	ความพลั่วของผ่าน แถบ (dB)	การลดทอนของแถบ หยุด(dB)
สี่เหลี่ยม	$0.9/N$	0.7416	21
แฮนนิ่ง	$3.1/N$	0.0546	44
แฮมมิง	$3.3/N$	0.0194	53
แบล็กแมน	$3.5/N$	0.0017	75
โคเชอร์	$\frac{A-7.95}{14.36(N-1)}$	ปรับได้	ปรับได้

2.5.5 ตัวกรองดิจิทัลแบบไอโออาร์

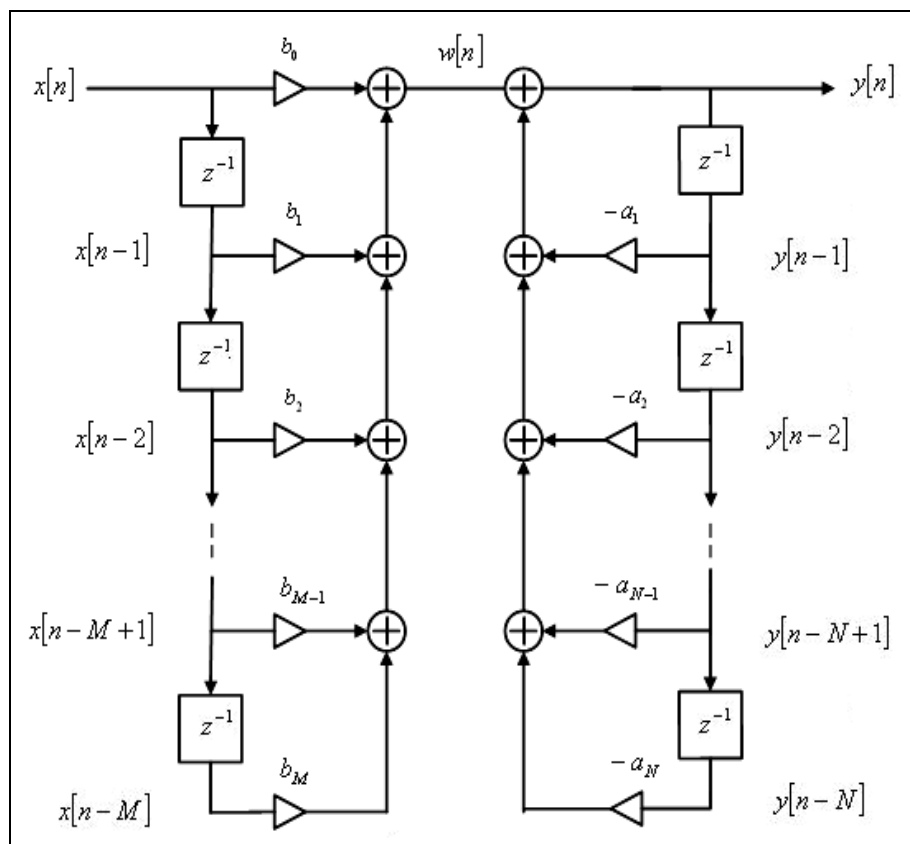
โครงสร้างพื้นฐานตัวกรองไอโออาร์ จากสมการ (2.7) มีฟังก์ชันถ่ายโอนของตัวกรองแบบไอโออาร์ มีลักษณะดังนี้

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

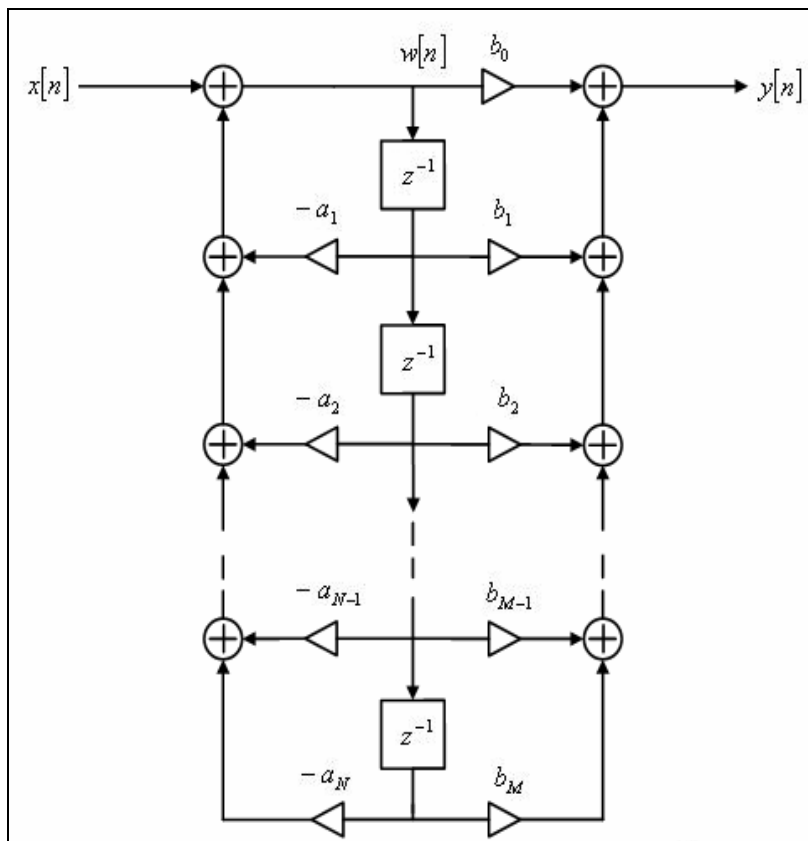
$$= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-N}} \quad (2.22)$$

สำหรับการสร้างวงจรให้มีคุณสมบัติตรงตามนิยามของตัวกรองดิจิทัลแบบไอโออาร์

สามารถทำได้หลายลักษณะคือ โครงสร้างตัวกรองไอโออาร์รูปแบบโดยตรง (direct form) จากสมการ (2.7) และสมการ (2.22) ตามลำดับ เขียนเป็นโครงสร้างแบบโดยตรงได้สองลักษณะคือ โครงสร้างในรูปที่ 2.25 มีชื่อเรียกว่าโครงสร้างรูปแบบโดยตรงชนิดที่หนึ่ง (direct form one structure) ส่วนโครงสร้างในรูปที่ 2.26 มีชื่อเรียกว่า โครงสร้างรูปแบบโดยตรงชนิดที่สอง (direct form two structure) ทั้งสองแบบแม้จะมีลักษณะการจัดวางตำแหน่งอุปกรณ์ที่แตกต่างไป แต่โดยหลักการแล้วให้ผลการทำงานที่เหมือนกัน



รูปที่ 2.25 โครงสร้างตัวกรองไอโออาร์แบบโดยตรงชนิดที่หนึ่ง



รูปที่ 2.26 โครงสร้างตัวกรองไอโออาร์แบบโดยตรงชนิดที่สอง

เนื่องจากตัวกรองมีอันดับสูง ๆ การสร้างตัวกรองไอโออาร์ที่มีโครงสร้างแบบโดยตรงทั้งสองชนิดสามารถทำได้ แต่อาจจะส่งผลกระทบต่อเรื่องความคลาดเคลื่อนของระบบ และความเสถียรภาพ โดยทั่วไปนิยมกระจายให้ H_z อยู่ในรูปฟังก์ชันที่อันดับต่ำ ๆ แล้วใช้รูปแบบการสร้างตัวกรองแบบคาสเคด หรือแบบขนานเข้ามาช่วย

โครงสร้างตัวกรองไอโออาร์รูปแบบคาสเคด (cascade form) สามารถแยกตัวประกอบฟังก์ชันถ่ายโอนออกเป็นส่วนย่อย ๆ ที่มีอันดับต่ำลง และโดยทั่วไปฟังก์ชันถ่ายโอนส่วนย่อยจะลดให้เหลือเป็นอันดับเท่ากับหนึ่งหรือสองเท่านั้น ดังนั้นสมการที่ (2.22) สามารถเขียนในรูปของตัวประกอบได้เป็น

$$H(z) = \prod_{k=1}^K \left[\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \right] \tag{2.23}$$

โดย $K = N/2$ เมื่อ N เป็นจำนวนคู่ และ $K = (N+1)/2$ เมื่อ N เป็นจำนวนคี่ ทั้งนี้ N คือ

อันดับของตัวกรอง สมการ (2.23) สามารถจัดให้กระชับขึ้นได้เป็น

$$\begin{aligned}
 H(z) &= \prod_{k=1}^K \frac{N_k(z)}{D_k(z)} \\
 &= \prod_{k=1}^K H_k(z)
 \end{aligned}
 \tag{2.24}$$

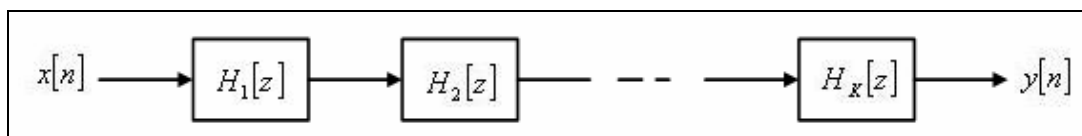
โดยที่

$$N_k(z) = b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2} \tag{2.25}$$

$$D_k(z) = 1 + a_{1k}z^{-1} + a_{2k}z^{-2} \tag{2.26}$$

$$H_k(z) = \frac{N_k(z)}{D_k(z)} \tag{2.27}$$

สังเกตว่าในกรณีที่ N เป็นจำนวนคี่ ทำให้ฟังก์ชันถ่ายโอนย่อย $H_k(z)$ ตัวใดตัวหนึ่งจะต้องมีอันดับเท่ากับหนึ่ง เมื่อฟังก์ชันถ่ายโอนที่ได้นำมาแยกตัวประกอบ สามารถเขียนเป็นโครงสร้างตัวกรองไอโออาร์แบบคาสเคด จะมีลักษณะดังรูปที่ 2.27 แสดงโครงสร้างแบบคาสเคด คือ การนำฟังก์ชันถ่ายโอนย่อยมาอนุกรมกัน



รูปที่ 2.27 โครงสร้างตัวกรองไอโออาร์แบบคาสเคด

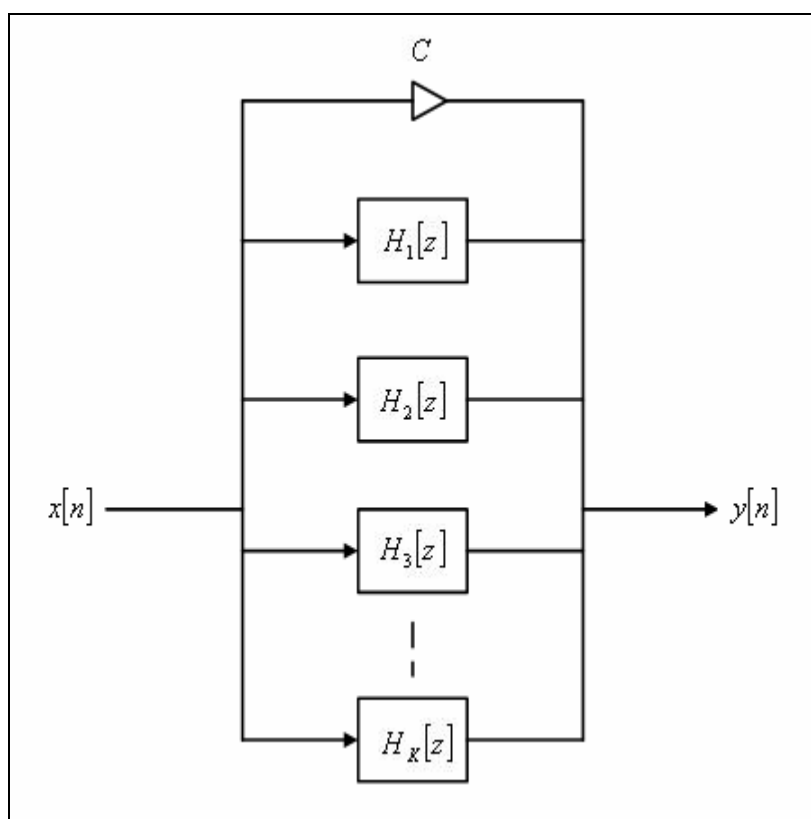
โครงสร้างตัวกรองไอโออาร์แบบสุดท้ายคือ โครงสร้างตัวกรองไอโออาร์แบบขนาน (parallel form) เป็นกรรมวิธีในการแยกโครงสร้างของตัวกรองไอโออาร์ ที่มีอันดับสูงๆ ออกเป็น

ส่วนประกอบย่อยอีกรูปแบบหนึ่ง นอกเหนือไปจากวิธีเศษส่วน ในการใช้รูปแบบขนานนั้นฟังก์ชันถ่ายโอนของตัวกรองถูกแยกเศษส่วนย่อย (partial fraction expansion) ดังแสดงในสมการต่อไป

$$H(z) = C + \sum_{k=1}^K H_k(z) \quad (2.28)$$

$$H(z) = \left[\frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \right] \quad (2.29)$$

โดย $K = \frac{N}{2}$ เมื่อ N คือ อันดับของตัวกรอง และ $C = \frac{b_N}{a_N}$



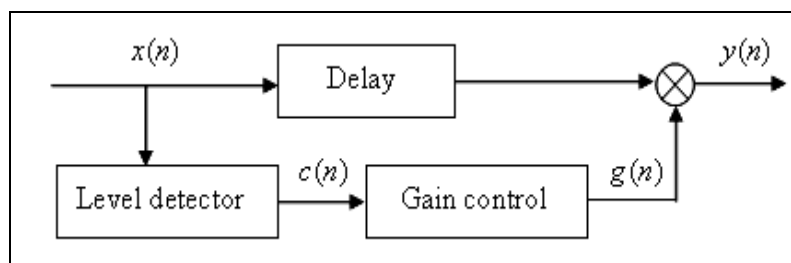
รูปที่ 2.28 โครงสร้างตัวกรองไอโออาร์แบบขนาน

จากรูปที่ 2.28 โครงสร้างตัวกรองไอโออาร์แบบขนาน พบว่าการแยกส่วนย่อยกับฟังก์ชันถ่ายโอนตามแบบขนานนี้มีข้อแตกต่างจากกรรมวิธีคาดตรงที่ลำดับการวางตำแหน่งของฟังก์ชันถ่ายโอนย่อย $H_k(z)$ ซึ่งไม่ส่งผลต่อสมรรถนะของตัวกรองแต่อย่างใด

2.6 ทฤษฎีการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์

2.6.1 กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัต

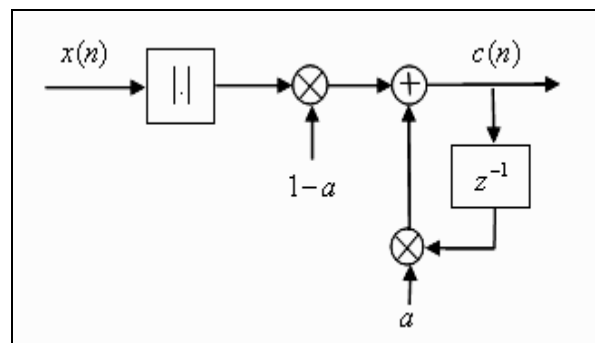
กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัต เป็นระบบที่ใช้ในการควบคุมแอมพลิจูดของสัญญาณเสียง โดยใช้เทคนิคบีบอัดสัญญาณเสียงพัลส์พลวัตแบบไปข้างหน้า (feedforward) ซึ่งมีพารามิเตอร์ช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ จากรูปที่ 2.29 แสดงแผนภาพบล็อกบีบอัดสัญญาณเสียงพัลส์พลวัต โดยสัญญาณเสียงจากอินพุตถูกแบ่งออกเป็น 2 ด้าน คือสัญญาณเสียงที่ผ่านด้านล่าง เรียกว่า ไซด์เชน (sidechain) ทำหน้าที่ควบคุมอัตราขยายของสัญญาณอินพุตที่มีความดังหรือแอมพลิจูดสูงมาก (overshoot) ประกอบด้วยตัวตรวจจับระดับสัญญาณ (level detector) โดยวัดแอมพลิจูดเอนVELOPE (envelop amplitude) ของสัญญาณอินพุตที่มีระดับความดังและควบคุมอัตราขยาย (gain controller) ทั้งนี้อัตราขยายสัญญาณเสียงจะเปลี่ยนแปลงตามควบคุมอัตราการขยาย ซึ่งแทนสัญลักษณ์ $g(n)$ และเป็นพื้นฐานของการคำนวณระดับตัวตรวจจับระดับ แทนสัญลักษณ์ $c(n)$ ของสัญญาณเสียงอินพุต แทนสัญลักษณ์ $x(n)$ ในส่วนของสัญญาณที่ผ่านทางด้านบน จะทำให้สัญญาณเสียงอินพุตถูกหน่วงด้วยอัตราการสุ่ม (sample) และต่อจากนั้นจะคูณด้วยอัตราขยาย ซึ่งในการหน่วงเวลาการสุ่ม (sample delay) เพื่อช่วยชดเชยในส่วนของสัญญาณที่ถูกหน่วงเวลาจากการคำนวณ $g(n)$ (Earl, 2001) (Brouckxon, Struyve, and Verhelst, 2006.) และ (Perttu, 2002)



รูปที่ 2.29 แผนภาพบล็อกบีบอัดสัญญาณเสียงพัลส์พลวัต

2.6.2 คุณสมบัติพัลส์พลวัต

จากแผนภาพกรอบตัวตรวจจับวัดระดับสัญญาณรูปที่ 2.29 ใช้ตัวตรวจจับวัดระดับสัญญาณเสียงในช่วงเวลาสั้น ๆ หรือสัญญาณแอนะล็อก $c(n)$ ของสัญญาณเสียงอินพุต $x(n)$ ซึ่งสามารถทำการกำหนดความเร็วของช่วงเวลาเข้ากระทำ (attack time) เพื่อเข้าสู่กระบวนการบีบอัดสัญญาณเสียงพิสัยพลวัต และกำหนดความเร็วของช่วงเวลาเลิกกระทำ (release time) เพื่อคืนสัญญาณเสียงกลับสู่สภาวะปกติ เมื่อสัญญาณเสียงมีระดับต่ำกว่าจุดเริ่มเปลี่ยน ซึ่งทั้งช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำกำหนดให้เป็นค่าคงที่ของแต่ละแบนด์



รูปที่ 2.30 ตัวกรองชนิดผลตอบสนองแบบไอโออาร์

จากรูปที่ 2.30 แสดงตัวกรองสัญญาณความถี่ผ่านต่ำชนิดผลตอบสนองแบบไอโออาร์ลำดับที่ 1 (infinite impulse response lowpass filter 1 order) โดยอ้างอิงจาก (Early, 2001) และ (Brouckxon, Struyve, and Verhelst, 2006) ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$c(n) = (1-a) \cdot |x(n)| + a \cdot c(n-1) \quad (2.30)$$

โดยสัมประสิทธิ์ a กำหนดความเร็วขาขึ้นและขาลง (rising and falling) ของระดับสัญญาณเสียงอินพุต ในที่นี้คือช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ เขียนเป็นสมการที่ (2.31) (Neeraj, Sanmatic, Frank and Maria, 2000)

$$a = \begin{cases} a = a_{attack} , & |x(n)| > c(n-1) \\ a = a_{release} , & |x(n)| < c(n-1) \end{cases} \quad (2.31)$$

เมื่อ $x(n)$ คือ สัญญาณอินพุต ส่วน $c(n-1)$ คือ สัญญาณผลตอบสนองไอโออาร์ที่ถูกหน่วงเวลา ซึ่ง กำหนดให้ a_{attack} คือช่วงเวลาเข้ากระทำ และ $a_{release}$ คือช่วงเวลาเลิกกระทำ โดยสัมพันธ์กับช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำของตัวตรวจจับวัฏระดับสัญญาณ เขียนเป็นสมการได้ดังนี้ (Brouckxon, H., Verhelst, W., Struyve, and Verhoeve P., 2000)

$$a_{attack} = e^{-T_s/t_{attack}} \quad (2.32)$$

$$a_{release} = e^{-T_s/t_{release}} \quad (2.33)$$

เมื่อ T_s คือ คาบอัตราการสุ่มของสัญญาณ ส่วน t_{attack} คือ ช่วงเวลาเข้ากระทำ และ $t_{release}$ คือ ช่วงเวลาเลิกกระทำ

2.6.3 คุณสมบัตินทางสถิติ

คุณสมบัตินทางสถิติของกระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัต จากแผนภาพกรอบรูปที่ 2.29 ซึ่ง $c(n)$ เป็นแอมพลิจูดของสัญญาณอินพุตถูกควบคุมโดยอัตราการขยาย จะมีความสัมพันธ์ระหว่างระดับสัญญาณเสียงอินพุตและระดับสัญญาณเสียงเอาต์พุต โดยมีพารามิเตอร์คือ สัญญาณเสียงอินพุตที่จุดเริ่มเปลี่ยน (threshold input audio signal) และอัตราส่วนบีบอัดสัญญาณ (compression ratio) เป็นผลให้ระดับสัญญาณเสียงเอาต์พุตเปลี่ยนแปลงเล็กน้อยซึ่งสอดคล้องกับการเปลี่ยนแปลงระดับสัญญาณเสียงอินพุต เขียนเป็นสมการได้ดังนี้ (Perttu, 2002)

$$g(n) = \begin{cases} c(n)/threshold^{r-1}, & c(n) > threshold \\ 1, & c(n) < threshold \end{cases} \quad (2.34)$$

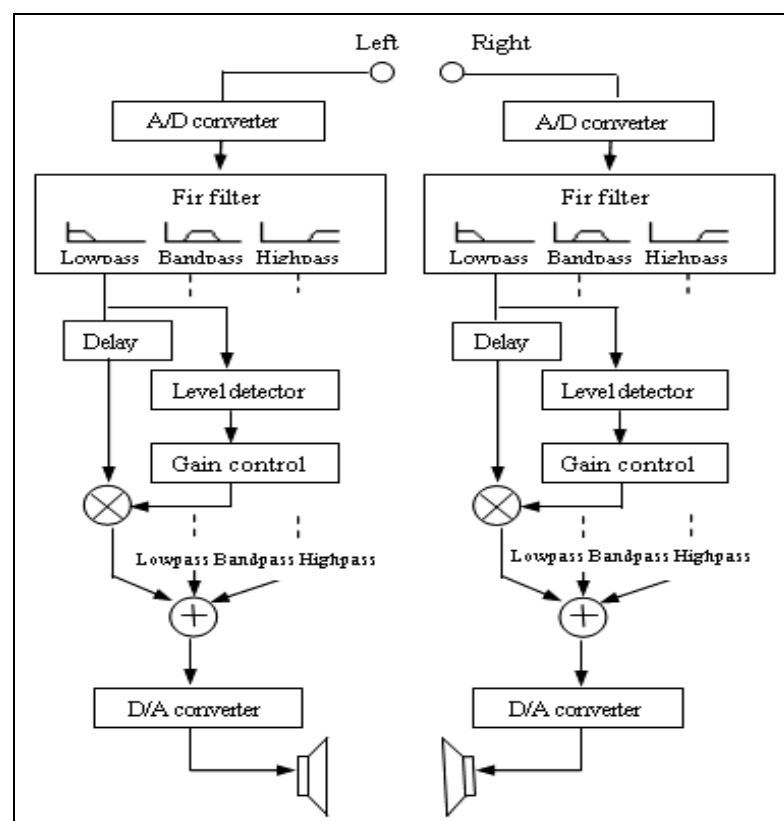
โดย r คือ อัตราส่วนของการบีบอัดสัญญาณเสียง ส่วน $threshold$ คือ จุดที่บีบอัดสัญญาณเสียง ถ้า $c(n) \geq threshold$ สัญญาณเสียงเข้าสู่กระบวนการบีบอัดสัญญาณ $c(n) \leq threshold$ สัญญาณเสียงไม่ถูกบีบอัดสัญญาณ

2.6.4 กระบวนการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์

การบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ โดยใช้เทคนิคในการแบ่งสเปกตรัม (spectrum) เสียงออกเป็นหลาย ๆ แบนด์ความถี่ ในที่นี้แบ่งออกเป็น 3 แบนด์ เนื่องจากเสียงดนตรีแบ่งออกเป็น 3 แบนด์คือ แบนด์เสียงต่ำ แบนด์เสียงกลาง และแบนด์เสียงสูงโดยอ้างอิงจากเว็บไซต์ (www.Orban.com) ต่อจากนั้นบีบอัดสัญญาณเสียงหรือจำกัดสัญญาณเสียง ซึ่งระหว่าง

แบนด์มีการเชื่อมต่อ (coupling) กันเพื่อป้องกันความแตกต่างในเรื่องอัตราขยายของแบนด์ที่ติดกัน (adjacent band) มีช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำที่เหมาะสม ในกระบวนการบีบอัดสัญญาณเสียงแบบหลายแบนด์มีเวลาคงที่ในแต่ละแบนด์ สำหรับแบนด์ที่มีความถี่ผ่านต่ำจะกำหนดเวลาเป็นค่าคงที่ซึ่งช้ากว่าแบนด์ที่มีความถี่ผ่านสูง และได้เปรียบกว่าการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบแบนด์เดียว (Dr.Luzheng, 1998) เนื่องจากสัญญาณที่มีความดังมักจะเป็นเสียงในแบนด์ความถี่ใดความถี่หนึ่ง

ดังนั้นการบีบอัดสัญญาณเสียงหรือการกล่าวอีกนัยหนึ่งคือ การลดอัตราขยายเฉพาะในย่านความถี่ที่มีความดัง ย่อมไม่ทำให้แบนด์ความถี่อื่นถูกระทบ เป็นผลให้เสียงที่ได้ในทุก ๆ แบนด์ความถี่มีระดับความดังไม่เปลี่ยนแปลงมาก ซึ่งเป็นคุณลักษณะเสียงที่ต้องการเพื่อรักษาค่าดัชนีการเบี่ยงเบนให้คงที่ สำหรับการมอดูเลตในระบบเอฟเอ็ม



รูปที่ 31 แผนภาพบล็อกการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์

จากรูปที่ 31 แสดงแผนภาพกรอบบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ โดยมีขั้นตอนการทำงานดังนี้ เมื่อมีสัญญาณอินพุตเข้ามาสัญญาณจะถูกแปลงจากแอนะล็อกเป็น

ดิจิทัล ผ่านกลุ่มตัวกรองสัญญาณความถี่ชนิดผลตอบสนองแบบจำกัด (infinite impulse response filter bank) เพื่อแบ่งสัญญาณอินพุตออกเป็นความถี่หลายแบนด์ แล้วผ่านเข้ากระบวนการบีบอัดสัญญาณพัลส์พลวัต ต่อจากนั้นจะทำการรวมสัญญาณเพื่อทำการแปลงรูปสัญญาณจากดิจิทัลเป็นแอนะล็อกออกลำโพง

2.7 สรุป

กระบวนการมอดูเลตแบ่งออกเป็น 3 แบบ คือ การมอดูเลตแบบแอมพลิจูด การมอดูเลตแบบเฟส และการมอดูเลตแบบเฟส ซึ่งการมอดูเลตแบบเฟส และการมอดูเลตแบบเฟสจะคล้ายกันมาก สามารถเรียกรวมกันได้ว่า การมอดูเลตแบบเชิงมุม การมอดูเลตแบบเฟสคือ ซึ่งเป็นการนำความแรงของสัญญาณเสียงไปควบคุมความถี่ของคลื่นวิทยุ เมื่อความแรงของสัญญาณเสียงมาก ส่งผลให้ความถี่ของคลื่นวิทยุจะสูงขึ้น แต่เมื่อความแรงของสัญญาณเสียงลดลงเป็นผลให้ความถี่ของคลื่นวิทยุต่ำลง ถ้าไม่มีสัญญาณเสียงเข้ามาความถี่ของคลื่นวิทยุจะอยู่ในสภาพปกติคือ ความถี่ปานกลาง นอกจากนี้การมอดูเลตยังขึ้นอยู่กับเปอร์เซ็นต์การมอดูเลต ดรรชนีการมอดูเลต และอัตราส่วนการเบี่ยงเบน เพื่อแก้ไขปัญหาการเกิดโอเวอร์มอดูเลชันและการขาดหายของสัญญาณพาหะ ซึ่งสัมพันธ์กับระบบประสาทที่เกี่ยวกับการได้ยินของมนุษย์ และการออกแบบตัวกรอง ซึ่งการออกแบบตัวกรองแบ่งเป็น 2 แบบคือ การออกแบบตัวกรองดิจิทัลแบบเอฟไออาร์ และการออกแบบตัวกรองดิจิทัลแบบไอไออาร์ โดยตัวกรองแบบเอฟไออาร์ ประกอบด้วยโครงสร้างตัวกรองไอไออาร์รูปแบบโดยตรง โครงสร้างตัวกรองไอไออาร์รูปแบบคาสเคด และโครงสร้างตัวกรองไอไออาร์แบบขนาน ซึ่งในการออกแบบจะต้องเลือกโครงสร้างที่เหมาะสมที่สุดในการใช้งาน และมีวิธีการออกแบบหน้าต่าง โดยอาศัยหลักการผลตอบสนองความถี่ที่สนใจ เช่น ตัวกรองผ่านต่ำ ตัวกรองผ่านสูง ตัวกรองผ่านแถบ และตัวกรองหยุดแถบ โดยพิจารณาจากค่าผลตอบสนองอิมพัลส์ของตัวกรองภายใต้หน้าต่างคือ หน้าต่างสี่เหลี่ยม หน้าต่างแฮนนิ่ง หน้าต่างแฮมมิ่ง และหน้าต่างแบล็กแมน ส่วนตัวกรองดิจิทัลแบบเอฟไออาร์มีโครงสร้างแบ่งออกเป็น โครงสร้างตัวกรองไอไออาร์รูปแบบโดยตรง โครงสร้างตัวกรองไอไออาร์รูปแบบคาสเคดและโครงสร้างตัวกรองไอไออาร์แบบขนาน สุดท้ายเป็นทฤษฎีการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ ซึ่งอาศัยคุณสมบัติทางสถิติและคุณสมบัติพัลส์พลวัต เพื่อทำการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์

บทที่ 3

การออกแบบและการใช้งาน

3.1 กล่าวนำ

ในปัจจุบันได้มีการนำชิพ (chip) ประมวลผลสัญญาณทางดิจิทัล ซึ่งเป็นไมโครโปรเซสเซอร์ (microprocessor) ประเภทหนึ่ง จะใช้ในงานด้านการประมวลผลสัญญาณดิจิทัลที่ต้องการทำงานแบบเวลาจริง เนื่องจากเหตุผลทั้งทางด้านประสิทธิภาพการทำงานของชิพสูง ใช้พลังงานที่ต่ำและมีความยืดหยุ่นในการโปรแกรมการทำงาน ทำให้เหมาะสมที่จะนำชิพมาประยุกต์ใช้ในการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ ซึ่งในบทนี้จะกล่าวถึงการประมวลผลสัญญาณดิจิทัลแบบเวลาจริง โดยใช้ชิพ DSP TMS320C6713 โดยตัวประมวลผลนี้สามารถประยุกต์ใช้งานทางด้านเสียงหลายช่องสัญญาณ เช่น การส่งกระจายเสียง การบันทึกเสียง เป็นต้น และกล่าวถึงสถาปัตยกรรม DSP TMS320C6713 รวมถึงนำมาประยุกต์ในการประมวลผลการบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ด้วย

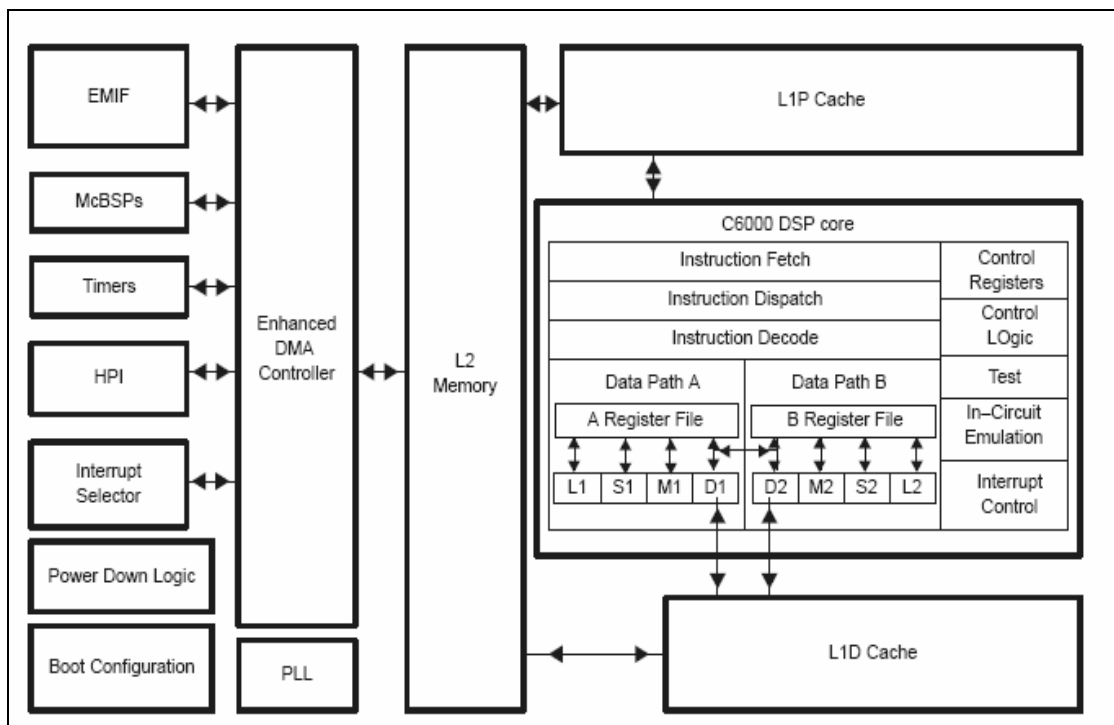
3.2 โครงสร้าง TMS320C6713

3.2.1 TMS320C6713 DSK

ชิพ TMS320C6713 อยู่ในตระกูลจุดลอย (float-point) ของบริษัทเท็กซัสอินสตรูเมนต์ (Texas instrument) ที่รวมฮาร์ดแวร์และซอฟต์แวร์ ใช้ในการประมวลผลแบบเวลาจริง

3.2.2 สถาปัตยกรรมของ TMS320C6713

สถาปัตยกรรมของ TMS320C6713 ซึ่งเป็นแบบ VLIW (very-long-instruction-word) และประกอบด้วยหน่วยความจำภายในแคช 2 ระดับ (cache 2 level) คือ แคชโปรแกรมระดับ 1 มีหน่วยความจำ 4 กิโลไบต์ของ (level 1 program cache : L1P) และแคชข้อมูลระดับ 1 มีหน่วยความจำ 4 กิโลไบต์ (level 1 data cache) และแรม (ram) 256 กิโลไบต์ หรือแคชระดับ 2 (level 2 cache) เป็นหน่วยความจำที่ใช้ร่วมกันระหว่างโปรแกรมและข้อมูล ซึ่งต้องการสัญญาณนาฬิกา (clock) ที่มีความเร็ว 225 เมกะเฮิรตซ์ สามารถคำนวณทางคณิตศาสตร์แบบจุดลอย 1350 ล้านต่อวินาที (million floating-point operations per second หรือ MFLOPS) หรือ 1800 ล้านคำสั่งต่อวินาที (million instructions per second หรือ MIPS) ดังรูปที่ 3.1 แสดงฟังก์ชันแผนภาพของ TMS320C6713 (function block diagram of TMS320C6713)



รูปที่ 3.1 แผนภาพบล็อกของ TMS320C6713

อุปกรณ์รอบนอกบนชิพ (on chip peripheral) ประกอบด้วย 2 พอร์ตอนุกรมบัฟเฟอร์หลายช่องสัญญาณ (multichannel buffer serial port หรือ McBSP) มี 2 ตัวจับเวลา (timer) แบบ 16 บิต นอกจากนี้มีการเชื่อมต่อพอร์ตโฮส (host port interface หรือ HPI) แบบ 32 บิต และเพิ่มอินเตอร์เฟซหน่วยความจำ (enhance memory interface หรือ EMIF) ซึ่งต้องการใช้ไฟเลี้ยงอินพุทและเอาต์พุท 3.3 โวลต์ และ 1.26 โวลต์ใช้เป็นไฟเลี้ยงหน่วยความจำหลักภายใน (internal core) ซึ่งโครงสร้างบัสภายในประกอบด้วย บัสตำแหน่งโปรแกรมแบบ 32 บิต (32-bit program address-bus) บัสข้อมูลโปรแกรมแบบ 256 บิต (256-bit program data bus) ซึ่งเหมาะสำหรับ 8 หน่วยฟังก์ชัน ส่วนบัสตำแหน่งข้อมูลเป็นแบบ 32 บิต บัสข้อมูลใช้ 2 บัสแบบ 64 บิต บัสเก็บข้อมูลใช้ 2 บัสแบบ 64 บิต และบัสตำแหน่งแบบ 32 บิต ซึ่งมีพื้นที่หน่วยความจำทั้งหมดเป็น $2^{32} = 4$ จิกะไบต์ (gigabyte) รวมกับหน่วยความจำภายนอก (external memory) ซึ่งบนชิพประมวลผลสัญญาณทางดิจิทัล (digital signal processing หรือ DSP) C6X แบ่งหน่วยความจำออกเป็นแบงก์ (bank) และแต่ละแบงก์เป็นหน่วยความจำที่เป็นอิสระต่อกัน เพื่อรองรับการเข้าถึง 2 หน่วยความจำใน 1 รอบชุดคำสั่งเท่านั้น โดยหน่วยความจำทั้ง 2 แบงก์เป็นอิสระต่อกัน จึงสามารถใช้บัสได้ 2 บัส และแต่ละบัสเป็นอิสระต่อกัน

ดังนั้นหน่วยความจำภายในคือ การจัดการภายในแเบงค์ของหน่วยความจำ ซึ่งใช้ 2 โหลด หรือเก็บ 2 คำสั่งและเป็นแบบขนาน ทั้งนี้เพื่อป้องกันการชนกันของการเข้าถึงข้อมูล ถ้าการเข้าถึงข้อมูลเข้ามาภายในแเบงค์ของหน่วยความจำที่แตกต่างกัน บัสถูกแบ่งออกเป็น บัสข้อมูล บัส โปรแกรม และการเข้าถึงหน่วยความจำโดยตรง (direct memory access หรือ DMA) เป็นผลให้ C6X ดำเนินการ โปรแกรมกระบวนการอ่านคำสั่ง (fetch) อ่านข้อมูล เขียนข้อมูล และกระบวนการทำงานของ DMA ซึ่งกระบวนการทั้งหมดนี้เกิดขึ้นพร้อมกัน

3.2.3 หน่วยความจำฟังก์ชัน

ซีพียู (CPU) ประกอบด้วย 8 หน่วยฟังก์ชัน (function unit) ซึ่งแบ่งข้อมูลออกเป็น 2 ส่วน คือ A และ B โดยแต่ละส่วนมีฟังก์ชันปฏิบัติการ (execute) คือ ตัวคูณ (multiplier) 2 หน่วยและการทำงานทางลอจิกและการคำนวณทางคณิตศาสตร์ (arithmetic logic units : ALU) 6 หน่วย ซึ่งประกอบด้วย หน่วยการคูณ (multiplier units หรือ .M) หน่วยการทำงานทางลอจิกและการคำนวณทางคณิตศาสตร์ (arithmetic and logic units หรือ .L) หน่วยการจัดการเกี่ยวกับบิตและการคำนวณทางคณิตศาสตร์ (arithmetic operations หรือ .S) และหน่วยโหลดและเก็บข้อมูล (data units หรือ .D) ซึ่งหน่วย .S และ .L เหมาะสำหรับการทำงานทางคณิตศาสตร์ ลอจิก และคำสั่งย่อย ส่วนการย้ายข้อมูลทั้งหมดใช้ .D หน่วย

3.2.4 รีจิสเตอร์

แฟมรีจิสเตอร์แบ่งออกเป็น 2 ส่วน ซึ่งแต่ละส่วนประกอบด้วยแฟมรีจิสเตอร์ 16 ตัว สำหรับใช้งานทั่วไปแบบ 32 บิต ดังตารางที่ 3.1 แสดงแฟมรีจิสเตอร์ A ซึ่งประกอบด้วยรีจิสเตอร์ ตั้งแต่ A0 ถึง A15 และแฟมรีจิสเตอร์ B ประกอบด้วยรีจิสเตอร์ตั้งแต่ B0 ถึง B15 ซึ่งรีจิสเตอร์ A0, A1, B0, B1 และ B2 ใช้เป็นรีจิสเตอร์ขึ้นอยู่กับเงื่อนไข (conditional register) ส่วนรีจิสเตอร์ A4 ถึง A7 และ B4 ถึง B7 เป็นตำแหน่งการหมุนเวียน (circular address) และรีจิสเตอร์ A0 ถึง A9 และ B0 ถึง B9 ยกเว้น B3 ซึ่งเป็นรีจิสเตอร์ชั่วคราว ส่วนรีจิสเตอร์ตั้งแต่ A10 ถึง A15 และ B10 ถึง B15 ใช้สำหรับการเก็บและดึงข้อมูลมาใช้ก่อนที่จะกลับมาที่รูทีนย่อย (subroutine) ถ้าค่าของข้อมูลเป็นแบบ 40 บิต สามารถใช้รีจิสเตอร์ข้าม โดยใช้บิตที่มีค่าต่ำสุด (least significant bits หรือ LSB) 32 บิต ซึ่งเป็นรีจิสเตอร์เก็บข้อมูลและเป็นรีจิสเตอร์คู่ เช่น A2 และเหลืออีก 8 บิตเป็นการเก็บข้อมูลถัดจาก LSB จะใช้เป็นรีจิสเตอร์ A3

นอกจากนี้มีรีจิสเตอร์ 32 ตัว เป็นรีจิสเตอร์กำหนดตามวัตถุประสงค์การใช้งาน ซึ่งสามารถควบคุมการประมวลผลและอินเตอร์รัพท์ เช่น รีจิสเตอร์โหมดตำแหน่ง (address mode register หรือ AMR)

ตารางที่ 3.1 แฟ้มรีจิสเตอร์คู่

แฟ้มรีจิสเตอร์ A	แฟ้มรีจิสเตอร์ B
A1:A0	B0:B1
A3:A2	B3:B2
A5:A4	B5:B4
A7:A6	B7:B6
A9:A8	B9:B8
A11:A10	B11:B10
A13:A12	B13:B12
A15:A14	B15:B14

3.2.5 หน่วยความจำโดยตรง DMA

หน่วยความจำโดยตรง ใช้ในการควบคุมการถ่ายโอนข้อมูลจากช่องว่างการประมวลผล DSP รวมถึงหน่วยความจำบนชิพ หน่วยความจำนอกชิพ และอุปกรณ์ภายนอก มีการทำงานเป็นอิสระต่อกัน โดย DMA เป็นการจัดการหลายช่องสัญญาณ ในที่นี้มี 16 ช่องสัญญาณเป็นช่องสัญญาณ ซึ่งเพิ่มช่องสัญญาณ DMA (enhanced DMA channel หรือ EDMA) ในการประมวลผล

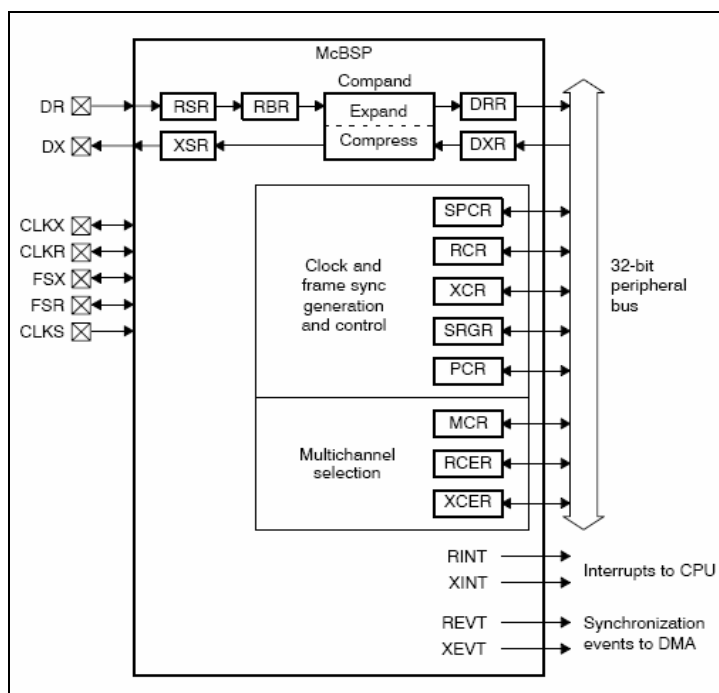
เนื่องจาก EDMA บนชิพนั้นมีลักษณะการทำงานแบบบล็อก โดยจะทำงานเป็นอิสระกับตัวประมวลผลหน่วยความจำหลัก (core) เพื่อสะดวกในการใช้งานเพราะมีประโยชน์ในช่วงเวลาการประมวลผล และในการทำงานการคำนวณทางคณิตศาสตร์ ซึ่งโครงสร้างของหน่วยความจำในระบบ DSP ให้ความสำคัญมาก ถ้า EDMA มีการดำเนินการที่สะดวกจากการใช้ฟังก์ชัน เป็นผลให้ EDMA สามารถเข้าถึงพื้นที่หน่วยความจำซึ่งสามารถเกิดขึ้นพร้อมกันกับการเข้าถึงข้อมูลโดยตัวประมวลผลหน่วยความจำหลัก ถ้า EDMA ประมวลผลและหน่วยความจำหลักประมวลผล ต้องการการเข้าถึงในพื้นที่หน่วยความจำในเวลาเดียวกัน ดังนั้นทำให้เกิดการชนกันของข้อมูลในหน่วยความจำ ซึ่งเหตุการณ์การชนกันของข้อมูลจะไม่เกิดขึ้นเนื่องจากแบ่งพื้นที่หน่วยความจำออกเป็นบล็อกหน่วยความจำ โดยแต่ละบล็อกเป็นอิสระต่อกันและการเข้าถึงเกิดขึ้นในเวลาเดียวกันหรือการประมวลผล EDMA ตามการเข้าถึงหน่วยความจำระหว่างส่วนว่าง (idle) ของรอบคำสั่ง DSP โดยใช้เทคนิคท่อส่ง (pipeline technique) เพื่อเพิ่มความเหมาะสมในการประมวลผล DSP

3.2.6 EMIF (External Memory Interface)

ในการประมวลผลหน่วยความจำบนชิพจะจำกัดในเรื่องการเก็บโปรแกรมและข้อมูล ซึ่งหน่วยความจำภายนอกเตรียมไว้เพื่อเก็บโปรแกรมและข้อมูล เป็นต้น ทำให้ต้องมีหน่วยความจำมากกว่าหน่วยความจำภายใน อย่างไรก็ตามในการเก็บโปรแกรมและข้อมูลนอกชิพทำให้เวลาเข้าถึงช้าลงและจำกัดบัสที่ใช้โอนถ่ายข้อมูล

3.2.7 McBSP (Multichannel Buffer Serial Port)

McBSP ใช้พอร์ตอนุกรมซิงโครไนซ์การสื่อสารข้อมูลแบบทั้งสองทิศทางได้พร้อมกัน (full duplex) ประกอบด้วย สัญญาณนาฬิกาเฟรม (frame) ซึ่งเป็นอิสระต่อกันทั้งทางด้านรับและด้านส่งข้อมูล ซึ่งสามารถเชื่อมต่อกับอุปกรณ์ภายนอก เช่น อุปกรณ์ตัวเข้ารหัส ถอดรหัส เมื่อมีการสื่อสารข้อมูลจากภายนอก จะเริ่มเคลื่อนย้ายข้อมูลภายใน



รูปที่ 3.2 แผนภาพบล็อกภายใน McBSP

จากรูปที่ 3.2 จะประกอบด้วยขาส่งข้อมูล (data transmit หรือ DX) และขารับข้อมูล (data receive หรือ DR) ซึ่งข้อมูลถูกควบคุมโดยสัญญาณนาฬิกาและซิงโครไนซ์เฟรม ประกอบด้วยขาสัญญาณ

นาฬิกาด้านส่ง (transmit clock หรือ CLKX) ขาสัญญาณนาฬิกาด้านรับ (receive clock หรือ CLKR) และขาซิงโครไนซ์เฟรมด้านส่ง (transmit frame synchronize หรือ FSX) โดยมีซีพียูหรือ EDMA ควบคุมการอ่านข้อมูลจากรีจิสเตอร์รับข้อมูล (data receive register หรือ DRR) ส่วนรีจิสเตอร์เลื่อนส่งข้อมูล (transmit shift register หรือ XSR) ทำการเลื่อนข้อมูลไปที่ขา DX และรีจิสเตอร์เลื่อนรับข้อมูล (receive shift register หรือ RSR) ทำการคัดลอกข้อมูลจากขา DR ไปยังรีจิสเตอร์บัฟเฟอร์ตัวรับ (receive buffer register หรือ RBR) และข้อมูลใน RBR ถูกคัดลอกข้อมูลส่งต่อไปยัง DRR เพื่อทำการอ่านข้อมูล ซึ่งถูกควบคุมโดย ซีพียู หรือ EDMA ส่วนรีจิสเตอร์อื่น ๆ คือ รีจิสเตอร์ควบคุมพอร์ตอนุกรม (serial port control register หรือ SPCR) รีจิสเตอร์ควบคุมการรับและส่ง (receive/transmit control register หรือ RCR/XCR) รีจิสเตอร์เปิดช่องสัญญาณด้านรับและส่ง (receive/transmit channel enable register หรือ RCER/XCER) รีจิสเตอร์ควบคุมขาส่ง (pin control register หรือ PCR) รีจิสเตอร์กำเนิดอัตราการสุ่ม (sample rate generator register หรือ SRGR) ซึ่งรีจิสเตอร์ทั้งหมดนี้รองรับการเพิ่มข้อมูลในการสื่อสาร

3.2.8 วงจรตั้งเวลา (Timer)

วงจรตั้งเวลาใช้ 32 บิต เป็นช่วงเวลานับเหตุการณ์หรือการอินเตอร์รัพท์ในซีพียู ซึ่งวงจรตั้งเวลาควบคุมช่วงเริ่มต้นแปลงสัญญาณจากแอนะล็อกเป็นสัญญาณดิจิทัลจากสัญญาณภายนอกหรือควบคุม โดย EDMA เพื่อเริ่มต้นถ่ายโอนข้อมูล ซึ่งวงจรตั้งเวลาประกอบด้วยรีจิสเตอร์คาบเวลา โดยเฉพาะวงจรตั้งเวลาความถี่และรีจิสเตอร์ตัวนับวงจรตั้งเวลา ซึ่งจะเพิ่มขึ้นตามตัวนับ ส่วนรีจิสเตอร์ควบคุมวงจรตั้งเวลา จะทำหน้าที่ควบคุมการทำงานของวงจรตั้งเวลา

3.2.9 อินเทอร์รัพท์พอร์ตโฮส (Host Port Interface)

อินเทอร์รัพท์พอร์ตโฮส ใช้เชื่อมต่อระหว่างตัวประมวลผลโฮส เช่น การประมวลผลไมโครโปรเซสเซอร์ ทั้งโฮสและตัวประมวลผล DSP เปลี่ยนข้อมูลโดยทางตัวประมวลผล DSP บนชิพสามารถโปรแกรมข้อมูลบนชิพหรือโปรแกรมหน่วยความจำ ข้อมูลจะถูกถ่ายโอนคำ (word) 8 บิต หรือ 16 บิต

3.2.10 อินเทอร์รัพท์ (Interrupts)

อินเทอร์รัพท์ มีทั้งอินเทอร์รัพท์ภายในและอินเทอร์รัพท์ภายนอก ทำหน้าที่หยุดการประมวลผลจากซีพียู และเปลี่ยนไปประมวลผลที่เกิดจากอินเทอร์รัพท์ (interrupt service routine หรือ ISR) ซึ่งแหล่งกำเนิดอินเทอร์รัพท์มาจากกระบวนการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลและสัญญาณนาฬิกา เป็นต้น ซึ่งซีพียูมีอินเทอร์รัพท์ 12 ชนิด ในช่วงตารางบริการอินเทอร์รัพท์ ตั้งแต่ 4 ถึง ตารางบริการอินเทอร์รัพท์ 11 (interrupt service table 4 to interrupt service table 11) มีรีจิสเตอร์ควบคุมอินเทอร์รัพท์ที่มีดังนี้ คือรีจิสเตอร์ควบคุมสถานะ (control status register หรือ CSR) ประกอบด้วยบิตเพื่อให้อินเทอร์รัพท์ทำงาน (global interrupt enable หรือ GIE) และบิตอื่น ๆ

ที่ใช้ในการควบคุม ซึ่งการทำงานรีจิสเตอร์ให้บริการอินเตอร์รัพท์ ทำหน้าที่โดยเฉพาะอินเตอร์รัพท์ ให้ทำงานและไม่ทำงาน (enable/disable) รีจิสเตอร์แฟล็กอินเตอร์รัพท์ (interrupt flag register หรือ IFR) แสดงสถานะอินเตอร์รัพท์รีจิสเตอร์เพื่อกำหนดค่าอินเตอร์รัพท์ (interrupt set register หรือ ISR) รีจิสเตอร์ให้อินเตอร์รัพท์ว่าง (interrupt clear register หรือ ICR) โดยการอินเตอร์รัพท์ ส่วนตัวชี้ตารางบริการอินเตอร์รัพท์ (interrupt service table pointer หรือ ISTP) ซึ่งกำหนดตำแหน่ง ISR นอกจากนี้ยังมีตัวชี้ส่งคืนอินเตอร์รัพท์ (interrupt return pointer หรือ IRP) และตัวชี้ส่งคืนอินเตอร์รัพท์แบบรอกไม่ได้ (nonmaskable interrupt return pointer หรือ NRP) ซึ่งในการคอมไพล์ (compile) สนับสนุนไลบรารี (library) โดยอัตโนมัติในการประมวลผลแบบเวลาจริง ซึ่งจะสร้างฟังก์ชัน `_c_int00` เพื่อเป็นเวกเตอร์ในการอินเตอร์รัพท์ จากตารางที่ 3.2 แสดงตารางบริการอินเตอร์รัพท์ (interrupt service table หรือ IST) ใช้เริ่มต้นอินเตอร์รัพท์แต่ละตำแหน่งภายในเป็นกลุ่มกระบวนการอ่านคำสั่ง (fetch packet หรือ FP) แต่ละอินเตอร์รัพท์ ส่วนทางด้านขวาของตารางแสดงตำแหน่งแต่ละการอินเตอร์รัพท์

ตารางที่ 3.2 ตารางบริการการอินเตอร์รัพท์

Interrupt	Offset
RESET	000h
NMI	020h
RESERVED	040h
RESERVED	060h
INT4	080h
INT5	0A0h
INT6	0C0h
INT7	0E0h
INT8	100h
INT9	120h
INT10	140h
INT11	160h
INT12	180h
INT13	1A0h
INT14	1C0h

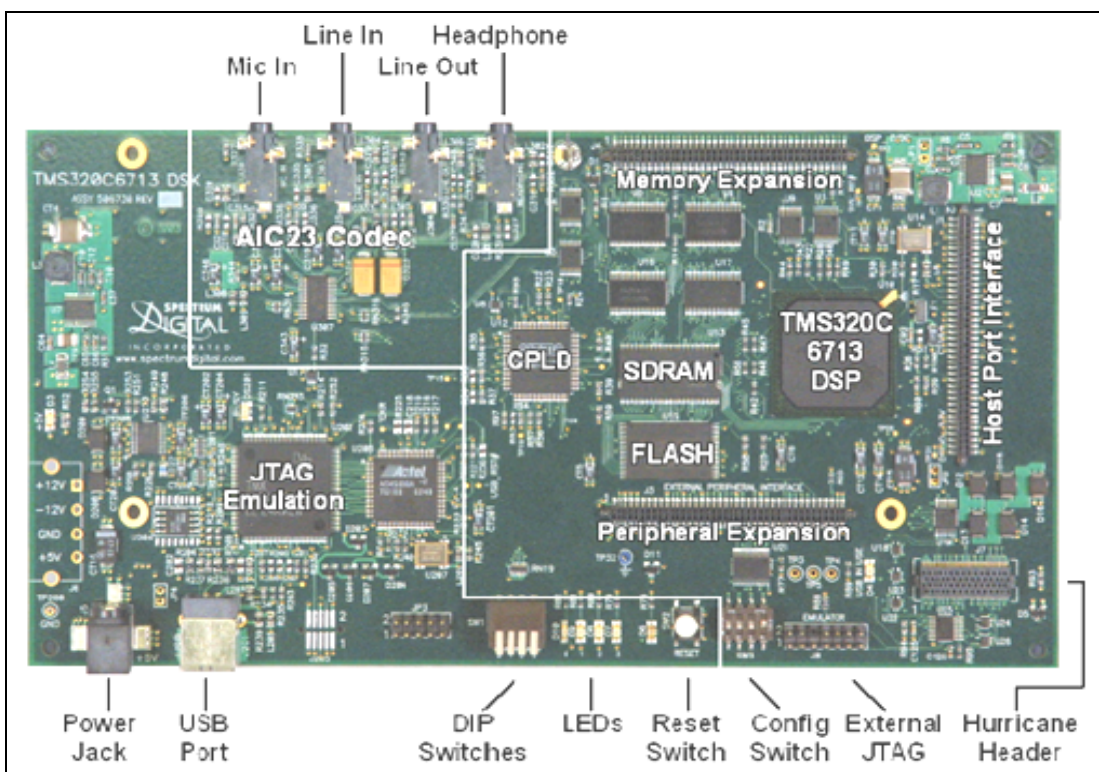
INT15	1E0h
-------	------

3.2.11 รูปลักษณะบูต (Boot Configuration)

ชิพ TMS320C6713 เปลี่ยนแปลงตามกำหนดรูปลักษณะบูต เมื่ออุปกรณ์ทำงานจะเริ่มต้นค่าใหม่ (reset) เพื่อความถูกต้องเริ่มต้นในการทำงาน อุปกรณ์บอร์ด DSK ทั้งหมดมีรูปลักษณะบูตคือ สามารถเลือกการเชื่อมต่อหน่วยความจำ โดยการกำหนดหน่วยความจำภายในหรือหน่วยความจำภายนอก โดยเริ่มต้นที่ตำแหน่ง 0 ถ้าเลือกหน่วยความจำภายนอกเพื่อเชื่อมต่ออุปกรณ์

3.2.12 บอร์ด DSK

บอร์ด DSK ใช้ชิพ TMS320C6713 ซึ่งมีประสิทธิภาพสูงและมีความเร็วประมวลผลสูง อีกทั้งรองรับมาตรฐานการเชื่อมต่อกับเครื่องคอมพิวเตอร์โดยใช้ซอฟต์แวร์ คือ code composer studio หรือ CCS มี SDRAM 16 เมกะไบต์ ใช้ TLV320AIC23 เป็นตัวแปลงสัญญาณเสียงอินพุตจากสัญญาณแอนะล็อกเป็นดิจิทัล และสัญญาณดิจิทัลเป็นแอนะล็อกเพื่อส่งสัญญาณเสียงออกเอาต์พุต จากรูปที่ 3.3 AIC23 มีช่องสัญญาณอินพุต 2 ช่องสัญญาณคือ ไมโครโฟน(microphone) และเส้นอินพุต (line input) ช่องสัญญาณเอาต์พุต 2 ช่องสัญญาณคือ หูฟัง (headphone) และเส้นเอาต์พุต (line output) ซึ่งรองรับอัตราการสุ่ม (sampling rate) ตั้งแต่ 8 กิโลเฮิร์ต ถึง 96 กิโลเฮิร์ต



รูปที่ 3.3 DSK บอร์ด TMS320C6713

3.3 การออกแบบประมวลผลแบบเวลาจริง

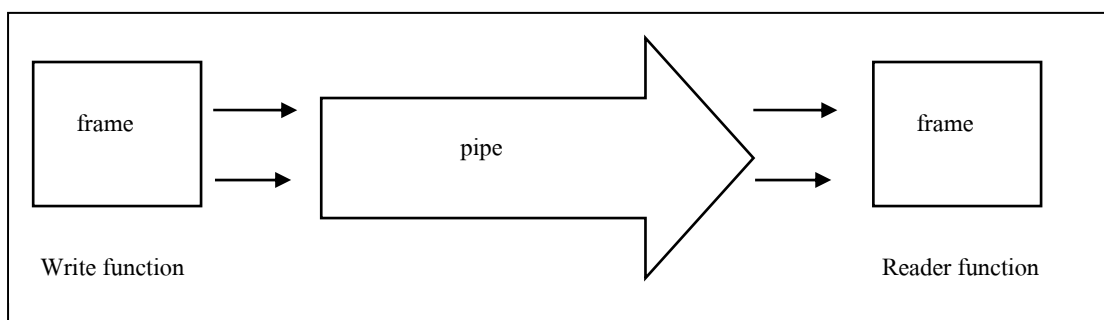
ซอฟต์แวร์ที่ใช้งานบอร์ดประมวลผลสัญญาณทางดิจิทัลคือ CCS ผู้เขียนโปรแกรมสามารถแปลโปรแกรมและส่งโปรแกรม (upload) ไปยัง TMS320C6713 นอกจากนี้ยังเหมาะสมกับการประมวลผลแบบเวลาจริง ที่เรียกว่า DSP/BIOS

3.3.1 DSP/BIOS

DSP/BIOS เป็นสเกลเบิ้ล (scalable) ของส่วนกลาง (kernel) แบบเวลาจริงพร้อมกับการจัดการเกี่ยวกับเวลาหลายสายโซ่ (thread) และใช้ในการจัดการอุปกรณ์รอบนอก ซึ่ง DSP/BIOS เป็นการวิเคราะห์แบบเวลาจริง โดยแสดงทางกราฟฟิกของลำดับปฏิบัติการสายโซ่ ในขณะที่ซีพียูจะทำการไหลซึ่งเป็นการดำเนินการแบบสถิติ ทั้งหมดนี้เป็นส่วนประกอบของ DSP/BIOS ซึ่งซีพียูจะสนับสนุนไลบรารี เพื่อเตรียมไว้ในการใช้ประโยชน์จากรูปลักษณะและจัดการอุปกรณ์รอบนอกบนชิพ DSP เช่น DMA พอร์ตอนุกรม ซึ่งเชื่อมต่อการประมวลผลของโฮสต์ตามมาตรฐานประยุกต์โปรแกรมตัวเชื่อมต่อ (application programming interface หรือ API) เพื่อให้สะดวกย้าย (migration) ข้อมูลโดยมีความแตกต่างระหว่างรูปแบบ DSP ซึ่ง DSP/BIOS รวมใน CCS

ในการที่จะสร้างโปรแกรม (build) หมายถึง ต้องการโมดูล DSP/BIOS เชื่อมโยงโมดูลทั้งหมด เช่น อินเทอร์รัพท์ฮาร์ดแวร์ (hardware interrupt หรือ HPI) ซึ่งในระบบส่วนกลาง DSP/BIOS มีโมดูล HPI ดำเนินการแบบเวลาจริงที่เตรียมไว้เมื่อเกิดเหตุการณ์อินเทอร์รัพท์จากฮาร์ดแวร์ โดยการปฏิบัติการต่อจากอินเทอร์รัพท์ฮาร์ดแวร์เป็นการจุดชนวน (trigger) ในรูปแบบลำดับการทำงานของวิกฤตทาส (critical task) ซึ่งอินเทอร์รัพท์ฮาร์ดแวร์เป็นสายโซ่ที่มีลำดับความสำคัญ (priority) สูงสุด เมื่อพิจารณาการประยุกต์ใช้งาน DSP/BIOS นอกจากนี้ยังมีอินเทอร์รัพท์ซอฟต์แวร์ (software interrupt หรือ SWI) เป็นอินเทอร์รัพท์ซอฟต์แวร์ ซึ่งเพิ่มระดับลำดับความสำคัญระหว่างอินเทอร์รัพท์ฮาร์ดแวร์ task และลำดับความสำคัญของอินเทอร์รัพท์ซอฟต์แวร์ สามารถตั้งค่าโดยใช้เครื่องมือรูปลักษณะ DSP/BIOS เป็นผลให้สายโซ่เริ่มปฏิบัติการ โดยเหตุการณ์ของ SWI สามารถประยุกต์ใช้ในการปฏิบัติของ task ในส่วนของอินเทอร์รัพท์คาบ (periodic interrupt หรือ PRD) เป็นโมดูลที่ใช้ในการจัดการเกี่ยวกับคาบของเหตุการณ์การกำหนดการ ซึ่งฟังก์ชัน PRD เป็นพื้นฐานบนการใช้อุปกรณ์อินพุตและอุปกรณ์เอาต์พุต หรือเหตุการณ์โปรแกรมอื่น ๆ ในการประยุกต์ใช้งานทั่วไปสามารถใช้ฟังก์ชันกำหนดบนสัญญาณนาฬิกาแบบเวลาจริง และท่อ (pipe หรือ PIP) เป็นการจัดการเกี่ยวกับท่อข้อมูล (data pipe) ท่อข้อมูลใช้อ่างอิงบัฟเฟอร์ (buffer) ในการไหลข้อมูลจากช่องสัญญาณอินพุตและช่องสัญญาณเอาต์พุต

เป็นไปได้อย่างต่อเนื่อง ซึ่งการจัดการจากท่อบัฟเฟอร์ DSP/BIOS หรือการจัดการโมดูล PIP บล็อก อินพุตและเอาต์พุตของข้อมูลใช้บัฟเฟอร์สตรีม (buffer stream) ของอินพุตและเอาต์พุตใน โปรแกรมโดยประยุกต์ใช้งานกับ DSP ฝังตัว (embedded DSP) แต่ละท่อมมีบัฟเฟอร์ซึ่งแบ่งความยาว เฟรมคงที่ โดยกำหนดชื่อ numframe และ framesize การจัดการปริมาณข้อมูลอินพุต และเอาต์พุต ของท่อขึ้นอยู่กับ 1 เฟรมของเวลา โดยแต่ละเฟรมมีความยาวคงที่ ส่วนท่อนั้นจะมี 2 ด้านคือ ด้านหนึ่ง เป็นผู้เขียนเมื่อมี โปรแกรมเขียนเฟรมข้อมูล ส่วนอีกด้านเป็นผู้อ่านเมื่อมี โปรแกรมอ่านเฟรมข้อมูล



รูปที่ 3.4 ข้อมูลในท่อ

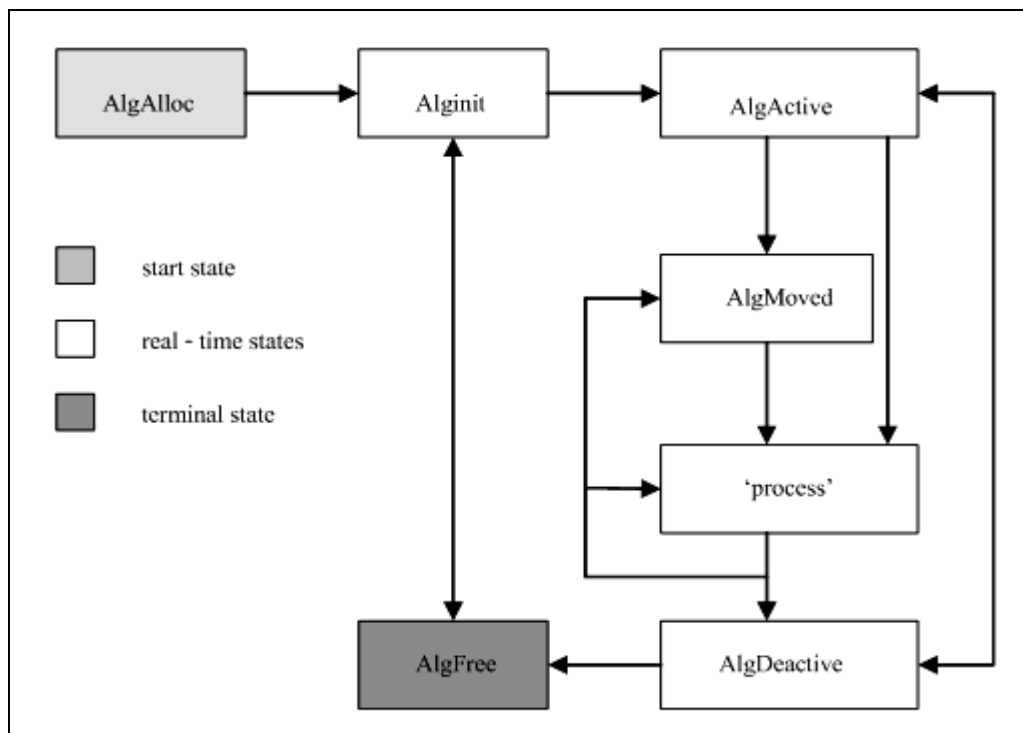
จากรูปที่ 3.4 ฟังก์ชันเขียนทำการเขียนเฟรม (write frame) ลงในท่อ เมื่อข้อมูลในท่อ เต็ม ฟังก์ชันอ่านทำการอ่านข้อมูล และสามารถกำหนดความยาวของเฟรมตามที่ผู้เขียนโปรแกรม กำหนด ท่อนี้เตรียมพร้อมไว้สำหรับ โครงสร้างข้อมูลในการส่งข้อมูลอินพุตและเอาต์พุตระหว่าง DSP และตัวเข้ารหัส ถอดรหัส (codec)

3.3.2 มาตรฐานอัลกอริทึม

ในปัจจุบันการพัฒนาโปรแกรมแต่ละ โปรแกรมอาจใช้อัลกอริทึมหลากหลาย ในการทำงานที่แตกต่างกัน เพราะฉะนั้นควรมีการวาง โครงร่างพื้นฐานที่ทำให้แต่ละอัลกอริทึมนั้นทำงาน ได้ในสภาพแวดล้อมเดียวกันและทำให้ผู้ใช้สามารถที่จะสลับสับเปลี่ยนแต่ละอัลกอริทึม ได้โดยไม่ต้องทำการเรียบเรียงโปรแกรมใหม่ ซึ่งประโยชน์ของมาตรฐานอัลกอริทึมคือ ผู้พัฒนาอัลกอริทึม จะต้องมีแบบแผนในการเขียน ไม่ต้องทำความเข้าใจในส่วนของจัดการหน่วยความจำที่ซับซ้อน และผู้พัฒนาโปรแกรม มีรูปแบบการติดต่อหรือเรียกใช้อัลกอริทึมต่าง ๆ ที่ถูกพัฒนาจากหลายผู้ผลิต และสามารถทำการเปรียบเทียบการทำงานของแต่ละอัลกอริทึมได้อย่างง่ายดาย

มาตรฐานอัลกอริทึม จะเน้นเฉพาะของส่วนการติดต่อระหว่างอัลกอริทึมและระบบ ทั้งหมดมากกว่าการพัฒนาประสิทธิภาพในการทำงานของอัลกอริทึมนั้น โดยจะเน้นไปที่การจัดการ ทรัพยากร เช่น การใช้หน่วยความจำ การจัดการรูปแบบส่งข้อมูลเข้าและออก โดยจะมีประโยชน์เมื่อ

มีหลายอัลกอริทึมทำงานร่วมกัน นอกจากนี้มาตรฐานอัลกอริทึมจะกำหนดให้แต่ละอัลกอริทึมไม่สามารถใช้ทรัพยากรของระบบโดยตรง แต่ให้จัดการผ่านส่วนที่มาตรฐานอัลกอริทึมกำหนดไว้ให้ API



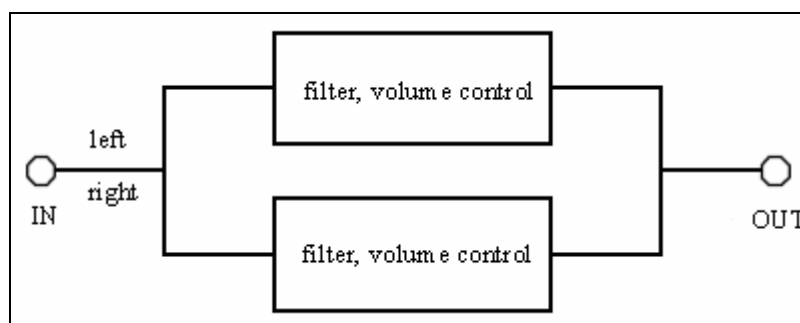
รูปที่ 3.5 ลำดับการทำงานของวัตถุ

ตารางที่ 3.3 แสดงฟังก์ชันการทำงาน

ฟังก์ชัน	คำอธิบาย
algActivate()	เป็นฟังก์ชันที่ทำหน้าที่เตรียมหน่วยความจำชั่วคราวที่จำเป็นในการคำนวณสำหรับอัลกอริทึมนั้น ๆ
algAlloc()	ทำหน้าที่จองหน่วยความจำสำหรับอัลกอริทึม
algDeactive()	ทำการบันทึกค่าของตัวแปรจากหน่วยความจำที่ใช้คำนวณ ไปไว้ในหน่วยความจำสำหรับเก็บค่า
algFree()	ทำหน้าที่คืนหน่วยความจำที่จองไว้สำหรับอัลกอริทึม
algInit ()	ทำการจัดเตรียมค่าที่จำเป็นสำหรับอัลกอริทึมในการประมวลผล
algMoved()	ทำการอัปเดต (update) ค่าหน่วยความจำในกรณีที่อัลกอริทึมถูกย้ายโดยผู้ใช้

3.3.3 ออกแบบการทำงาน

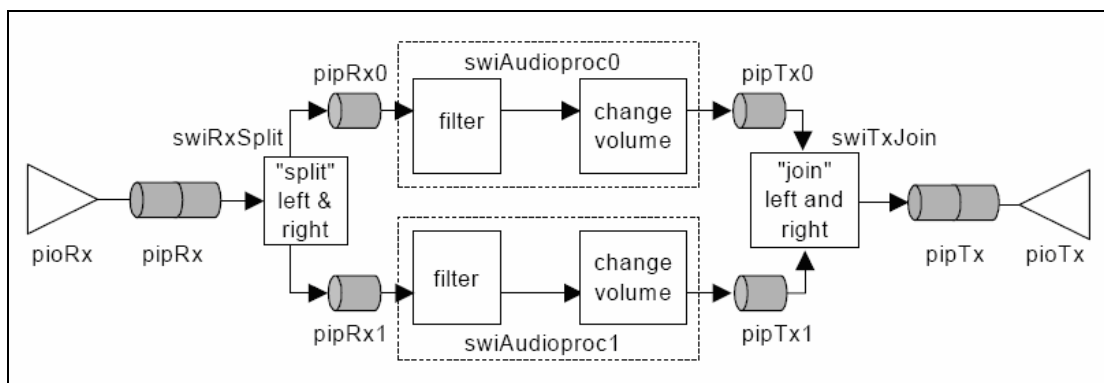
ทำการออกแบบสัญญาณเสียงที่มาจากอินพุต แปลงสัญญาณเป็นการสุ่มข้อมูลแบบดิจิทัล โดยแบ่งช่องสัญญาณออกเป็นช่องสัญญาณด้านซ้ายและช่องสัญญาณด้านขวา ซึ่งแต่ละบล็อกของสัญญาณเสียงใช้ 16 บิต ในการสุ่มสัญญาณ แต่ละช่องสัญญาณมีตัวกรองผ่านต่ำ ตัวกรองผ่านแถบ และตัวกรองผ่านสูง ส่วนของ volume control เป็นปัจจัยควบคุมแอมพลิจูดหรือควบคุมการบีบอัดสัญญาณเสียง ต่อจากนั้นสัญญาณเสียงแต่ละช่องสัญญาณรวมเข้าด้วยกันและทำการแปลงสัญญาณดิจิทัลเป็นแอนะล็อกเพื่อส่งสัญญาณออกเอาท์พุท



รูปที่ 3.6 แผนภาพบล็อกการออกแบบการทำงานพื้นฐาน

3.3.4 หลักการทำงานเมื่อนำมาใช้กับ DSP/BIOS

เมื่อทำการนำสัญญาณเสียงผ่าน โมดูล DSP/BIOS จะทำให้สามารถเขียนเป็นแผนภาพดังรูปที่ 3.7 ซึ่งใช้อัตราการสุ่มที่ 44.1 กิโลเฮิร์ตซ์ จำนวน 16 บิต ขนาดของเฟรม 80 คำ และดำเนินการป้อนสัญญาณเสียงอินพุตผ่านอุปกรณ์ pioRX เพื่อแปลงสัญญาณเสียงจากอุปกรณ์อินพุต 1 เฟรม (frame) จากสัญญาณเสียงแอนะล็อกเป็นสัญญาณดิจิทัล ต่อจากนั้นสัญญาณเสียงผ่านเข้าท่อข้อมูลอินพุต คือ pipRx ทำให้สายโซ่ swiRxSplit ทำงานเป็นโพส (post) เพื่อทำการคัดลอกสัญญาณเสียงจากอินพุต และเขียนข้อมูลสัญญาณเสียงให้กับท่อข้อมูลทางด้านซ้ายคือ pipRx0 และท่อข้อมูลทางด้านขวาคือ pipRx1 เป็นผลให้สายโซ่ swiAudioproc0 และ swiAudioproc1 มีการทำงานแบบโพสแทน ต่อจากนั้นสัญญาณเสียงแต่ละด้านผ่านตัวกรอง 3 ตัวกรองคือ ตัวกรองผ่านต่ำ ตัวกรองผ่านแถบและตัวกรองผ่านสูง เพื่อเข้าสู่กระบวนการบีบอัดสัญญาณเสียงที่ตำแหน่ง change volume แต่ละท่อข้อมูลจะเก็บข้อมูลไว้ที่ pipTx0 pipTx1 ซึ่งเป็นท่อส่งข้อมูลทางด้านซ้ายและด้านขวาตามลำดับ ต่อจากนั้นรวมสัญญาณเสียงทั้งสองช่องสัญญาณเข้าด้วยกันที่ pipTx ทำให้ swiTxJoin ทำงาน และดำเนินการส่งสัญญาณออกเอาท์พุตที่อุปกรณ์ pioTx เพื่อแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก



รูปที่ 3.7 แผนภาพบล็อกการทำงานเมื่อใช้ DSP/BIOS

3.4 สรุป

ในการบีบอัดสัญญาณเสียงพืสัยพลวัตแบบหลายแบนด์ เมื่อนำมาใช้กับบอร์ด DSP ควรจะเลือกชิพซึ่งเป็นตัวประมวลผลที่มีความเร็ว และมีประสิทธิภาพสูง ดังนั้นจึงเลือกชิพ TMS320C6713 ซึ่งมีคุณสมบัติตรงตามที่ต้องการ และเป็นชิพประมวลผลที่ผลิตโดยบริษัทเท็กซัสอินทรูเมนท์ ที่รวมฮาร์ดแวร์และซอฟต์แวร์เพื่อใช้ในการประมวลผลแบบเวลาจริง ซึ่งสามารถประมวลผลเป็นแบบ

จุดคงที่และจุดลอย เนื่องจากโครงสร้างภายนอกของบอร์ดมี AIC23 เป็นตัวเข้ารหัสและถอดรหัสหลายช่องสัญญาณ และรองรับอัตราการสุ่มสูงถึง 96 กิโลเฮิร์ตซ์ ซึ่งเหมาะสมนำมาใช้ในงานวิจัยนี้

ในส่วนทางด้านซอฟต์แวร์ทางบริษัทเท็กซัสอินสตรูเมนต์ มีมาตรฐานอัลกอริทึมเตรียมไว้สำหรับประยุกต์ใช้งาน ซึ่งในบทนี้ได้ทำการออกแบบโดยใช้ DSP/BIOS เป็นเครื่องมือช่วยจัดลำดับการทำงานต่าง ๆ เพื่อให้สามารถจัดการทรัพยากรและการทำงานของระบบได้อย่างมีประสิทธิภาพมากขึ้น

บทที่ 4

ผลการทดลอง

4.1 กล่าวนำ

ในบทนี้เป็นการนำทฤษฎีและหลักการทั้งหมดที่ได้กล่าวไว้แล้วในบทที่ผ่านมา เพื่อช่วยในการออกแบบตัวกรองและทดลองผลตอบสนองตัวกรองของแต่ละแบนด์ โดยทำการทดลองช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำของแต่ละแบนด์ ซึ่งกำหนดให้เป็นค่าคงที่และมีค่าแตกต่างกันในแต่ละแบนด์ ต่อจากนั้นทดลองบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ ซึ่งในแต่ละแบนด์ได้ทำการเปรียบเทียบสัญญาณที่ถูกบีบอัดในอัตราส่วนที่แตกต่าง โดยมีจุดเริ่มเปลี่ยนจุดเดียวกันและวัดสเปกตรัมของสัญญาณ เพื่อเปรียบเทียบระหว่างสัญญาณซึ่งผ่านกระบวนการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ในอัตราส่วนต่าง ๆ และสัญญาณที่ไม่ได้ผ่านกระบวนการบีบอัดสัญญาณ

4.2 การออกแบบตัวกรอง

เพื่อให้ง่ายในการออกแบบตัวกรอง จึงเลือกใช้ MATLAB นำมาวิเคราะห์และนำมาใช้ในการออกแบบตัวกรองเอฟไออาร์ โดยใช้เครื่องมือวิเคราะห์และการออกแบบตัวกรอง (filter design and analysis tool หรือ FDATool) ซึ่งใน MATLAB มีหน้าต่างคำสั่ง เมื่อทำการเปิด FDATool เพื่อทำการออกแบบ โดย FDATool ประกอบด้วยส่วนของเชื่อมโยงกราฟฟิก (graphical user interface หรือ GUI) ซึ่งมีแสดงค่าปริยาย (default) หน้าต่าง สามารถเลือกหน้าต่างออกแบบตัวกรอง โดยมีพารามิเตอร์ใช้ในการออกแบบชนิดของตัวกรอง เช่น ตัวกรองผ่านต่ำ ตัวกรองผ่านแถบ ตัวกรองผ่านสูง เป็นต้น สามารถเลือกลำดับสัมประสิทธิ์ของตัวกรอง โดยกำหนดอัตราการสุ่มความถี่ ซึ่งใช้พารามิเตอร์ F_s ส่วนพารามิเตอร์ F_{pass} คือเลือกขอบความถี่ผ่านแถบ และพารามิเตอร์ F_{stop} คือ ขอบความถี่แถบหยุด ซึ่งพารามิเตอร์ที่กล่าวมามีหน่วยความถี่ที่ตั้งไว้เป็นเฮิรตซ์ อย่างไรก็ตามสามารถเปลี่ยนหน่วยเป็นกิโลเฮิรตซ์ เมกะเฮิรตซ์ได้ทั้งนี้ขึ้นอยู่กับความถี่ที่ต้องการออกแบบ ส่วนพารามิเตอร์ A_{stop} เป็นผลตอบสนองเชิงขนาดของการลดทอนแถบหยุด และพารามิเตอร์ A_{pass} คือการลดทอนแถบผ่าน ซึ่งทั้งสองพารามิเตอร์มีหน่วยเป็นเดซิเบล

ในวิทยานิพนธ์นี้ได้ออกแบบตัวกรองแบบเอฟไออาร์ คือตัวกรองผ่านต่ำ ตัวกรองผ่านแถบและตัวกรองผ่านสูง ซึ่งตัวกรองทั้งหมดนี้ใช้หน้าต่างแฮมมิง และเมื่อพิจารณาโดเมนทางเวลา หน้าต่างแฮมมิงค่อย ๆ ลดระดับไซด์โลบ (side lobe) ลงสู่ศูนย์ และเมื่อพิจารณาโดเมนทางความถี่ จะทำให้แอมพลิจูด

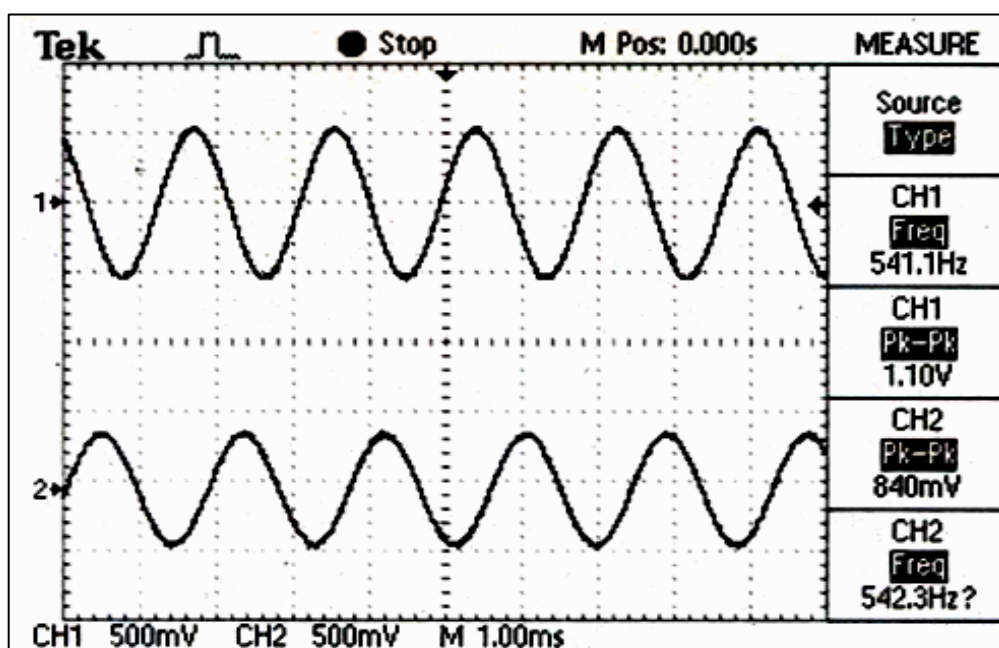
จุดของเมนโลบ (main lobe) กว้างมากเมื่อเทียบกับหน้าตาสีเหลี่ยม และไซด์โลบเล็กลงจากเมนโลบที่ระดับสูงสุด -41 เดซิเบล ซึ่งมีความพลั่วของผ่านแถบ 0.0194 เดซิเบล และการลดทอนของแถบหยุด -53 เดซิเบล ซึ่งเมื่อเทียบกับหน้าตาสีเหลี่ยม ที่มีไซด์โลบอยู่ในระดับสูงสุด -13 เดซิเบล ความพลั่วผ่านแถบอยู่ที่ 0.716 เดซิเบล และการลดทอนแถบหยุด -21 เดซิเบล (Emmanuel, and Barrie, 2002) ซึ่งในการออกแบบตัวกรองดิจิทัล เมื่อพิจารณาจากสัดส่วนของเมนโลบเปรียบเทียบกับไซด์โลบพบว่าถ้าหน้าตาสีเหลี่ยมให้สัดส่วนที่มีค่าสูงสุด ทำให้ช่วงแถบผ่านและช่วงแถบหยุดมีความพลั่วที่แกว่งขึ้นลงน้อย ซึ่งหน้าตาสีเหลี่ยมมีคุณสมบัติทางด้านความพลั่วดีกว่าหน้าตาสีเหลี่ยม

นอกจากนั้นกำหนดสัมประสิทธิ์ตัวกรอง 61 ลำดับ เพื่อให้ได้ตัวกรองแบบเอฟไออาร์ที่มีผลตอบสนองเฟสเชิงเส้น และกำหนดอัตราการสุ่มความถี่ 44.1 กิโลเฮิร์ตซ์ เนื่องจากการสุ่มด้วยอัตราที่สูงมาก ๆ ทำให้สัญญาณเอาท์พุทมีความคล้ายคลึงกับสัญญาณอินพุทมากขึ้น จากนั้นจะกำหนดให้ตัวกรองผ่านแถบมีความถี่ตัดที่ 800 เฮิร์ตซ์ ตัวกรองผ่านแถบ มีช่วงความถี่ผ่านได้ตั้งแต่ 820 เฮิร์ตซ์ จนถึง 3 กิโลเฮิร์ตซ์ ส่วนตัวกรองผ่านสูงมีความถี่ตัดที่ 3.2 กิโลเฮิร์ตซ์ ซึ่งช่วงความถี่ที่แบ่งตัวกรองเป็นช่วงที่หูของมนุษย์สามารถตอบสนองได้ดีตั้งแต่ความถี่ที่ 1 กิโลเฮิร์ตซ์ ถึง 5 กิโลเฮิร์ตซ์ (J. S. Lamancusa, 2000) ต่อจากนั้นนำไฟล์ออก (export file) เป็นไฟล์ส่วนหัวซี (C header file)

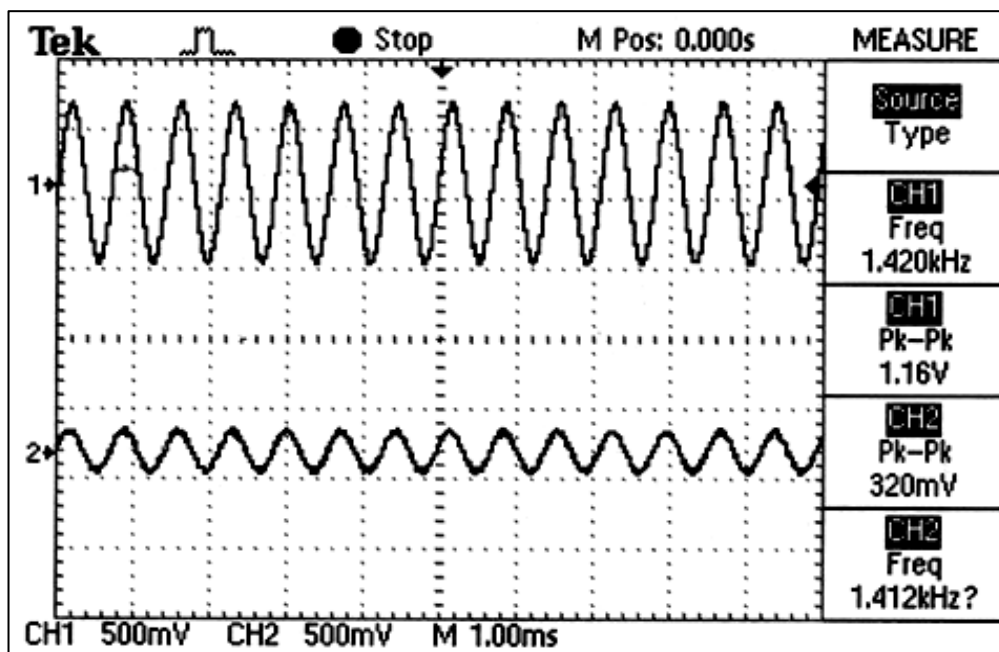
4.3 ผลการทดลองผลตอบสนองของตัวกรอง

เมื่อออกแบบทั้ง 3 ตัวกรองแล้ว จึงนำสัมประสิทธิ์ที่ได้จากการออกแบบทำการประมวลผลสัญญาณแบบเวลาจริง โดยใช้บอร์ด DSK TMS320C6713 ได้ผลตอบสนองของตัวกรองผ่านต่ำแบบเอฟไออาร์ ในรูปที่ 4.1 แสดงผลตอบสนองของตัวกรองผ่านต่ำ ทำการป้อนความถี่ 541.1 เฮิร์ตซ์ ของสัญญาณช่องที่ 1 ซึ่งเป็นช่องสัญญาณอินพุท ส่วนสัญญาณช่องที่ 2 ให้เป็นช่องสัญญาณเอาท์พุท พบว่าสัญญาณเอาท์พุทผ่านได้ เนื่องจากความถี่ที่ป้อนทางอินพุทอยู่ในช่วงความถี่ที่ต้องการออกแบบให้สัญญาณผ่านได้ ส่วนในรูปที่ 4.2 แสดงผลตอบสนองความถี่ตัดของตัวกรองผ่านต่ำ โดยกำหนดให้สัญญาณในช่องที่ 1 เป็นสัญญาณอินพุทและมีความถี่ 1.420 กิโลเฮิร์ตซ์ ส่วนสัญญาณในช่องที่ 2 จะเป็นช่องสัญญาณเอาท์พุท พบว่าสัญญาณเอาท์พุทมีแรงดันลดลง เนื่องจากตัวกรองผ่านต่ำได้ออกแบบความถี่ตัดไว้ที่ 800 เฮิร์ตซ์ ซึ่งความถี่ของอินพุทเป็นความถี่ที่อยู่ในช่วงความถี่ตัด จากรูปที่ 4.3 แสดงผลตอบสนองของตัวกรองผ่านแถบ โดยทำการป้อนสัญญาณอินพุทช่องที่ 1 มีความถี่ 2.849 กิโลเฮิร์ตซ์ และช่องสัญญาณเอาท์พุทช่องที่ 2 พบว่าสัญญาณผ่านได้และมีแรงดันที่ใกล้เคียงกับช่องสัญญาณอินพุท ซึ่งแรงดันที่ลดลงเนื่องจากเกิดการสูญเสียอัตราการขยายตัวเข้ารหัส ออครหัสของบอร์ด DSK ส่วนในรูปที่ 4.4 แสดงผลตอบสนองความถี่ตัดทางด้านต่ำของตัวกรองผ่านแถบ โดยป้อนสัญญาณอินพุททางช่องสัญญาณที่ 1 มีความถี่ 827.8 เฮิร์ตซ์ จะทำให้สัญญาณเอาท์พุทของ

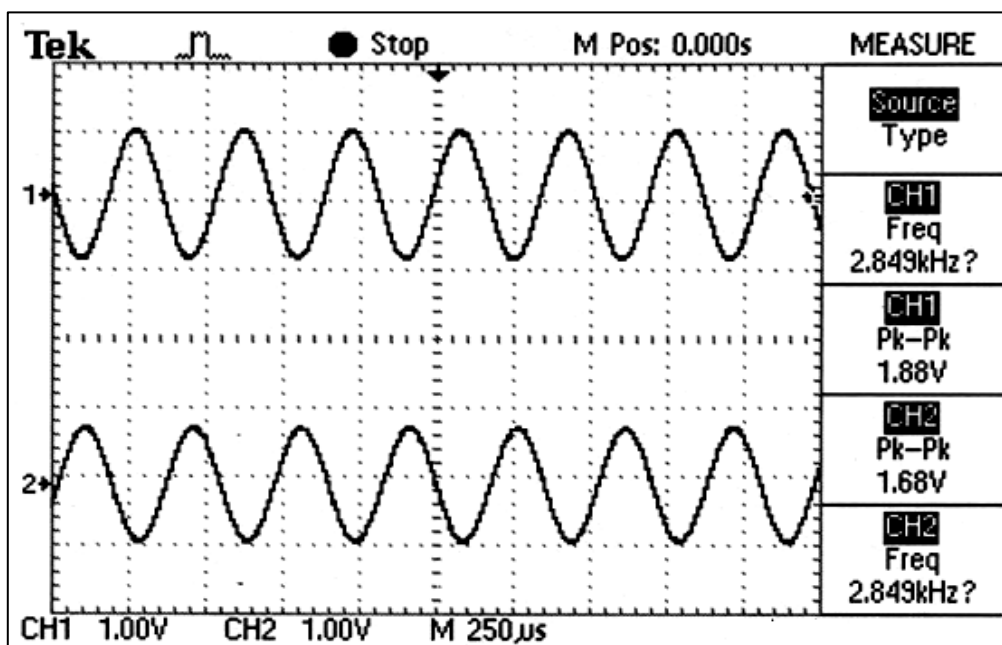
ช่องสัญญาณที่ 2 มีแอมพลิจูดลดลง เนื่องจากความถี่ของสัญญาณอินพุตอยู่ในช่วงความถี่ตัดของตัวกรองผ่านแถบ และในรูปที่ 4.5 แสดงผลตอบสนองความถี่ตัดทางด้านสูงของตัวกรองผ่านแถบ โดยป้อนสัญญาณอินพุตทางช่องสัญญาณที่ 1 มีความถี่ 4.587 กิโลเฮิร์ตซ์ ทำให้ช่องสัญญาณเอาต์พุตของช่องที่ 2 มีแอมพลิจูดลดลง เนื่องจากความถี่ของสัญญาณอินพุตอยู่ในช่วงความถี่ตัดของตัวกรองผ่านแถบ นั้นแสดงว่าตัวกรองแถบผ่านทำงานได้ตรงตามที่คาดหมายไว้ ส่วนรูปที่ 4.6 แสดงผลตอบสนองของตัวกรองผ่านสูง โดยป้อนสัญญาณอินพุตทางช่องสัญญาณที่ 1 มีความถี่ 4.545 กิโลเฮิร์ตซ์ ทำให้ได้สัญญาณเอาต์พุต ซึ่งให้เป็นช่องสัญญาณที่ 2 พบว่าสัญญาณผ่านได้ จากรูปที่ 4.7 แสดงผลตอบสนองความถี่ตัดของตัวกรองผ่านสูง โดยทำการป้อนสัญญาณอินพุตทางช่องสัญญาณที่ 1 มีความถี่ 2.849 กิโลเฮิร์ตซ์ จึงทำให้ช่องสัญญาณที่ 2 มีแรงดันเอาต์พุตลดลง เนื่องจากความถี่ของสัญญาณที่ป้อนให้อินพุตอยู่ในช่วงความถี่ตัดของแบนด์ความถี่สูง



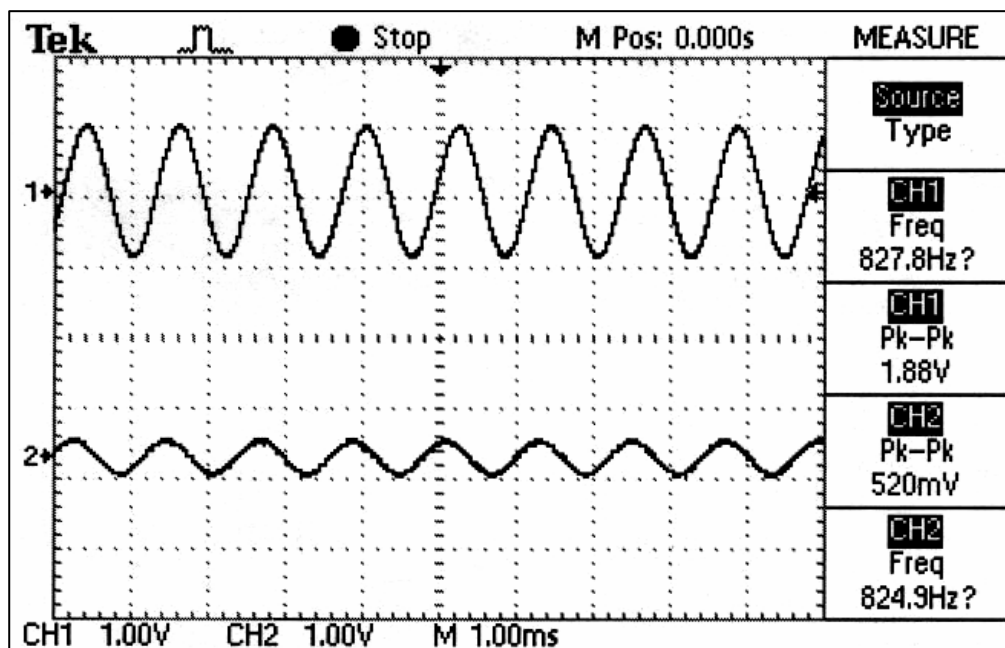
รูปที่ 4.1 ผลตอบสนองของตัวกรองผ่านต่ำ



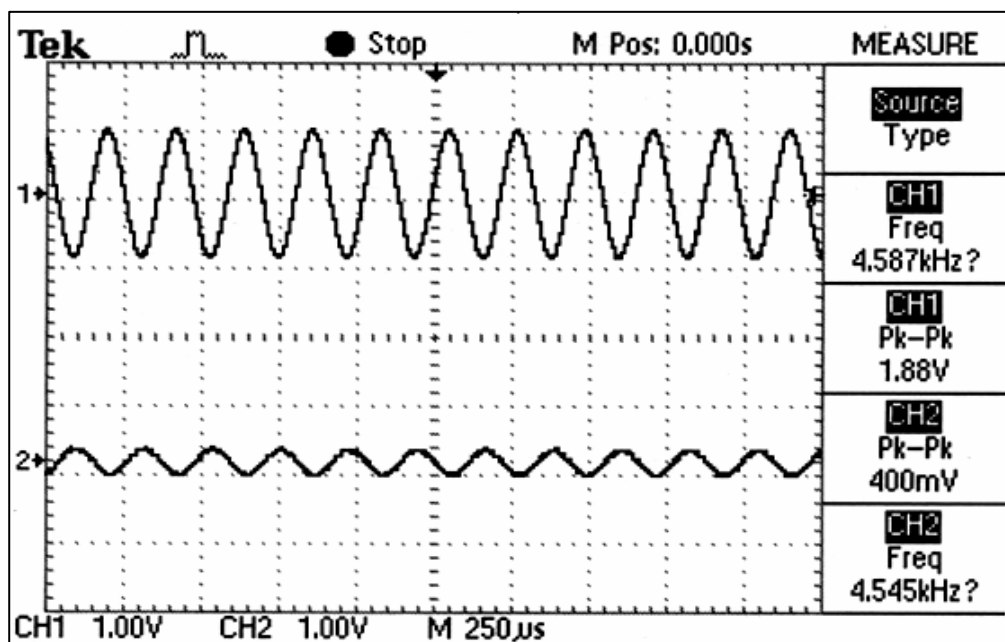
รูปที่ 4.2 ผลตอบสนองความถี่ต่ำของตัวกรองผ่านต่ำ



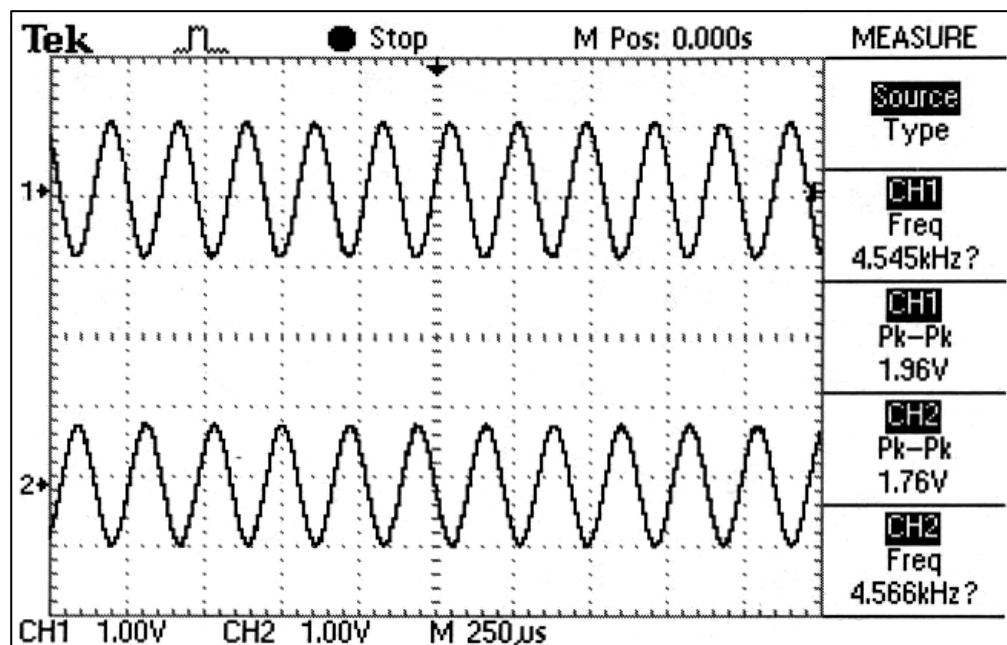
รูปที่ 4.3 ผลตอบสนองของตัวกรองผ่านแถบ



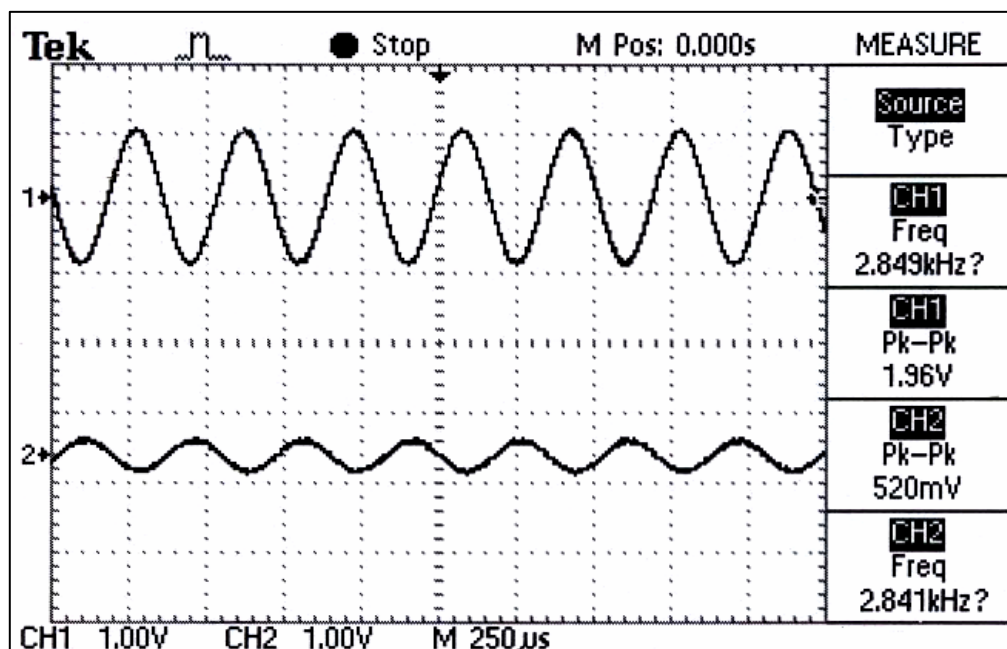
รูปที่ 4.4 ผลตอบสนองความถี่ตัดทางด้านต่ำของตัวกรองผ่านแถบ



รูปที่ 4.5 ผลตอบสนองความถี่ตัดทางด้านสูงของตัวกรองผ่านแถบ



รูปที่ 4.6 ผลตอบสนองของตัวกรองผ่านสูง



รูปที่ 4.7 ผลตอบสนองความถี่ตัดของตัวกรองผ่านสูง

4.4 ผลการทดลองช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำ

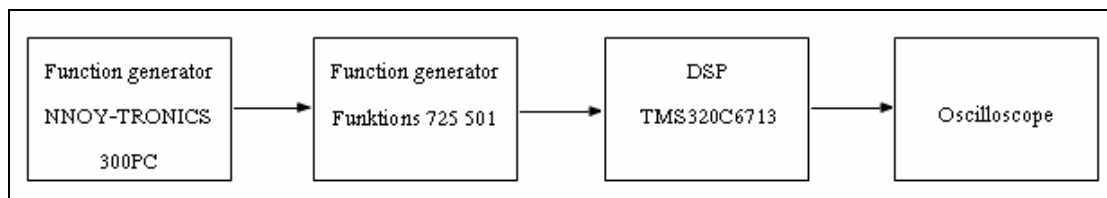
การทดลองช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำ แสดงในดั่งรูปที่ 4.8 แสดงลักษณะการต่ออุปกรณ์เพื่อทดลองวัดช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำของแต่ละแบนด์ ซึ่งแต่ละแบนด์มีช่วงเวลาย้ำกระทำและเลิกกระทำไม่เท่ากัน แบนด์ที่อยู่ในแบนด์ความถี่ผ่านสูงมีช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำเร็วกว่าแบนด์ที่อยู่ในแบนด์ความถี่ผ่านต่ำ เนื่องจากที่แบนด์ความถี่สูงสร้างปัญหาการเกิดโอเวอร์มอดูเลชัน



รูปที่ 4.8 ลักษณะการต่ออุปกรณ์เพื่อทดลองวัดช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำของแต่ละแบนด์

จากรูปที่ 4.9 แสดงแผนภาพลือวิธีการวัดช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำ โดยมีขั้นตอนการทดลองดังนี้ ใช้เครื่องกำเนิดสัญญาณ (Function generator) รุ่น NNOY-TRONICS, 300 PC เพื่อกำเนิดสัญญาณรูปคลื่นสี่เหลี่ยมที่มีความถี่ 0.01 เฮิร์ตซ์ แทนคลื่นพาหะ และใช้เครื่องกำเนิดสัญญาณ รุ่น Funktions generator, 725 501 เพื่อกำเนิดสัญญาณรูปคลื่นไซน์ที่มีความถี่ต่าง ๆ คือ ความถี่ผ่านต่ำ 511.9 เฮิร์ตซ์ ความถี่แถบผ่าน 1.705 กิโลเฮิร์ตซ์ และความถี่ผ่านสูง 5.405 กิโลเฮิร์ตซ์ ต่อจากนั้นนำสัญญาณทั้งสองมอดูเลตแบบแอมพลิจูด เพื่อสังเกตผลตอบสนองของช่วงเวลาย้ำกระทำและช่วงเวลาเลิกกระทำของแต่ละแบนด์ ต่อจากนั้นนำสัญญาณที่ได้จากการมอดูเลตแบบแอม

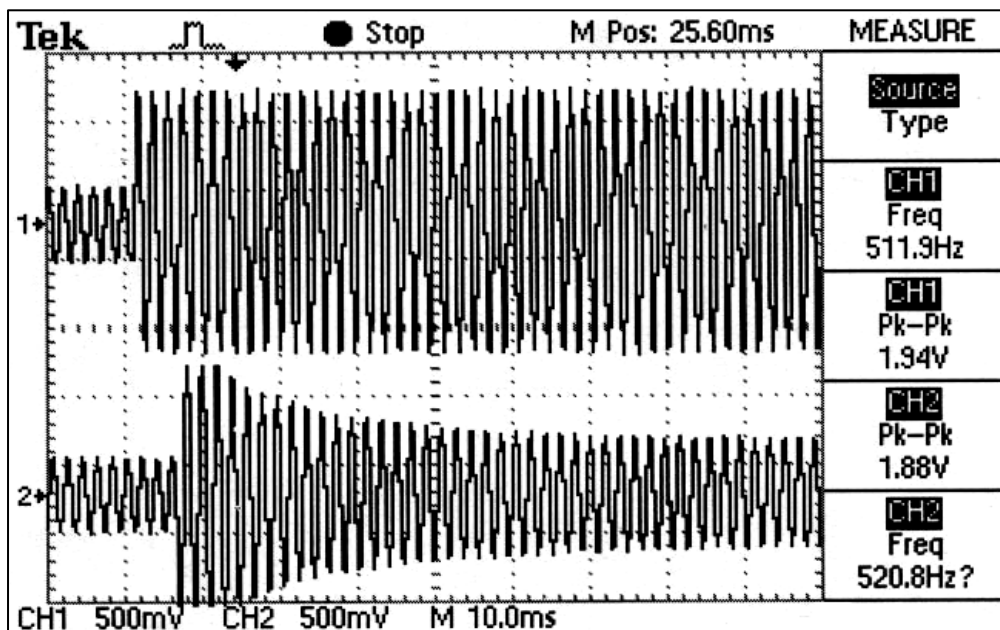
พลิจูดป้อนเข้าสู่เส้นอินพุทของบอร์ด DSK TMS320C6713 เพื่อทำการประมวลผลสัญญาณ และส่งสัญญาณออกทางเส้นเอาต์พุท เพื่อป้อนเข้าสู่ออสซิลโลสโคป



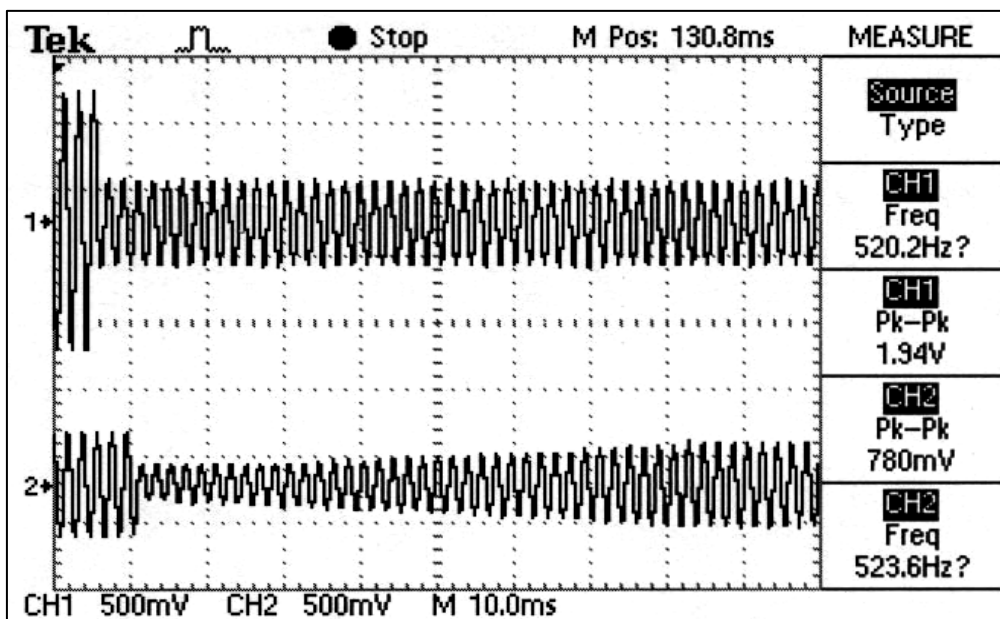
รูปที่ 4.9 แผนภาพบล็อกวิธีการวัดช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ

จากผลการทดลอง ได้รูปสัญญาณช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำที่แบนด์ต่าง ๆ ดังรูปที่ 4.10 แสดงผลตอบสนองของความถี่ผ่านต่ำที่มีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที เมื่อพิจารณาสัญญาณช่องที่ 1 ซึ่งเป็นสัญญาณอินพุทที่มีความถี่ 511.9 เฮิรตซ์ จะสังเกตเห็นว่า สัญญาณช่วงแรกมีความเบาทำให้สัญญาณเอาต์พุท ซึ่งกำหนดให้เป็นสัญญาณช่องที่ 2 พบว่าสัญญาณไม่ถูกบีบอัด แต่เมื่อสัญญาณช่องที่ 1 มีความแรงมากกว่าระดับจุดเปลี่ยนที่กำหนดไว้ ทำให้สัญญาณช่องที่ 2 เข้าสู่ช่วงเวลาเข้ากระทำ 30 มิลลิวินาที เพื่อปรับสัญญาณเข้าสู่กระบวนการบีบอัดสัญญาณ ส่วนรูปที่ 4.11 แสดงผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเลิกกระทำ 50 มิลลิวินาที เมื่อสัญญาณช่องที่ 1 ซึ่งเป็นสัญญาณอินพุทมีความแรงของสัญญาณลดลงภายใต้ระดับจุดเปลี่ยน ทำให้สัญญาณเอาต์พุทจากสัญญาณช่องที่ 2 ใช้ช่วงเวลาเลิกกระทำ 50 มิลลิวินาทีเพื่อปรับสัญญาณเข้าสู่สภาวะปกติ ส่วนในรูปที่ 4.12 แสดงผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเข้ากระทำ 20 มิลลิวินาที โดยให้สัญญาณช่องที่ 1 ป้อนสัญญาณอินพุท และสัญญาณช่องเอาต์พุทเป็นสัญญาณช่องที่ 2 พบว่าเมื่อสัญญาณช่องที่ 1 มีความแรงมาก จะทำให้สัญญาณช่องที่ 2 ใช้เวลาเข้ากระทำ 20 มิลลิวินาที ซึ่งสังเกตว่าสัญญาณเกิดทรานเซียนต์ (transient) เพื่อไม่ให้เกิดการสะดุดในช่วงรอยต่อระหว่างระดับสัญญาณปกติกับระดับสัญญาณที่ถูกบีบอัด ต่อจากนั้นสัญญาณจะเข้าสู่สถานะอยู่ตัว (steady state) ส่วนในรูปที่ 4.13 แสดงผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเลิกกระทำ 40 มิลลิวินาที เมื่อสัญญาณอินพุทจากสัญญาณช่องที่ 1 มีระดับความแรงของสัญญาณลดลง สัญญาณเอาต์พุทที่สัญญาณช่องที่ 2 ปรากฏสถานะทรานเซียนต์ ซึ่งเป็นช่วงเวลาเลิกกระทำและใช้เวลา 40 มิลลิวินาที ต่อจากนั้นสัญญาณเข้าสู่สถานะอยู่ตัว ในรูปที่ 4.14 แสดงผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเข้ากระทำ 10 มิลลิวินาที และรูปที่ 4.15 แสดงผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเลิกกระทำ 30 มิลลิวินาที ซึ่งทั้งในรูปที่ 4.14 และรูปที่ 4.15 จะให้ผลตอบสนองช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำเหมือนกับที่แบนด์ความถี่อื่น แต่จะแตกต่างในการกำหนดช่วงเวลาเข้ากระทำและช่วงเวลาเลิกกระทำ โดยกำหนดให้เร็วขึ้นเพราะที่แบนด์ความถี่ผ่านสูง สัญญาณจะสร้างปัญหาโอ

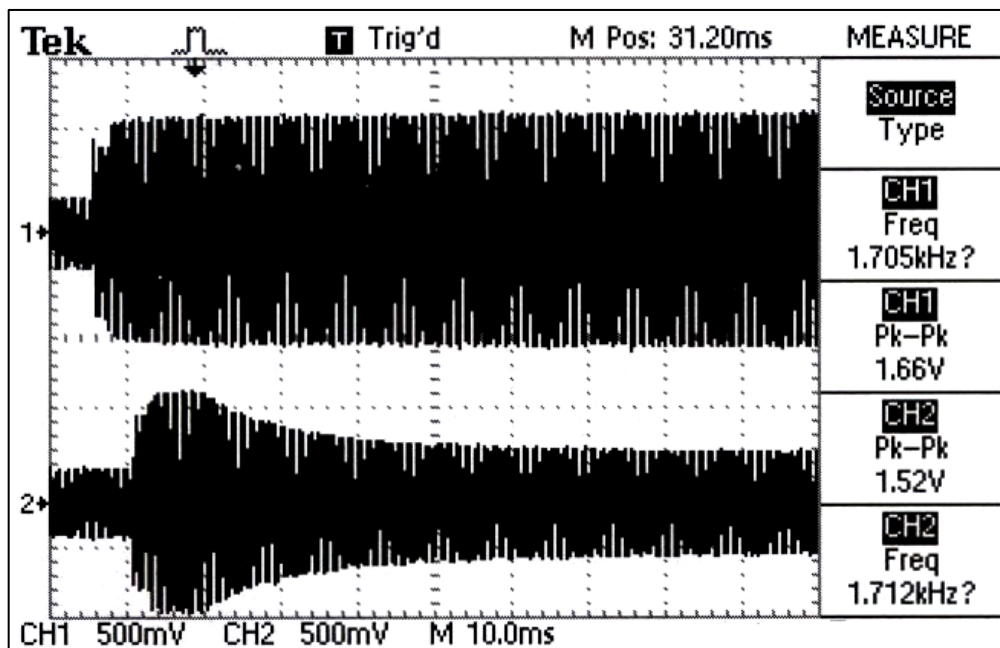
เวอร์มอดูเลชันมากกว่าแบนด์ความถี่ผ่านต่ำ เนื่องจากที่แบนด์ความถี่สูง เมื่อนำสัญญาณเสียงมามอดูเลตแบบเอฟเอ็ม ทำให้ได้แบนด์วิดท์กว้างกว่ามาตรฐานแบนด์วิดท์ของการส่งกระจายเสียงระบบเอฟเอ็ม



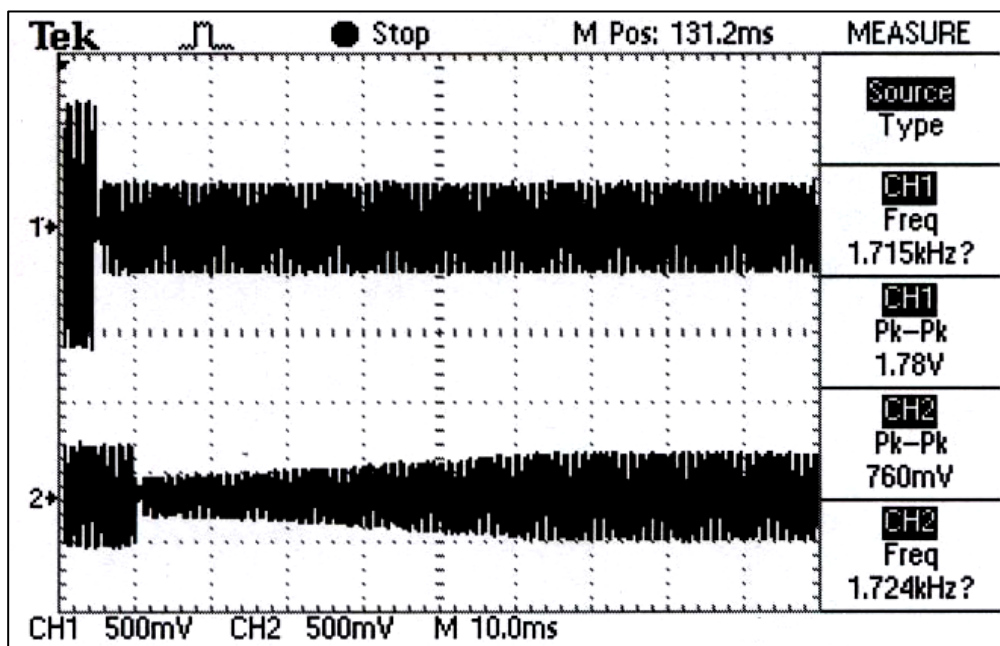
รูปที่ 4.10 ผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที



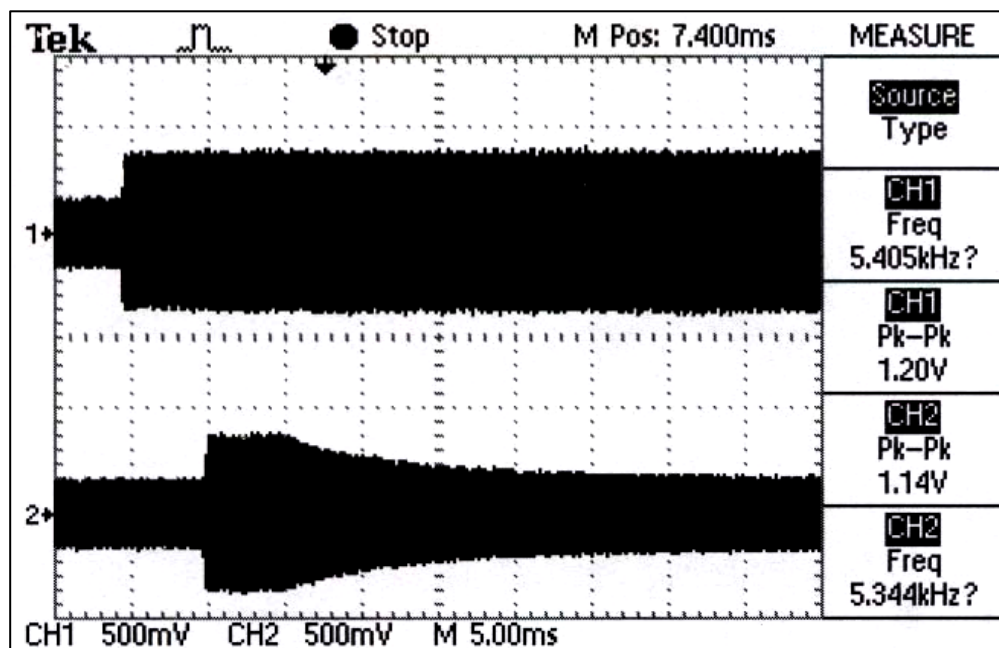
รูปที่ 4.11 ผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเล็กกระทำ 50 มิลลิวินาที



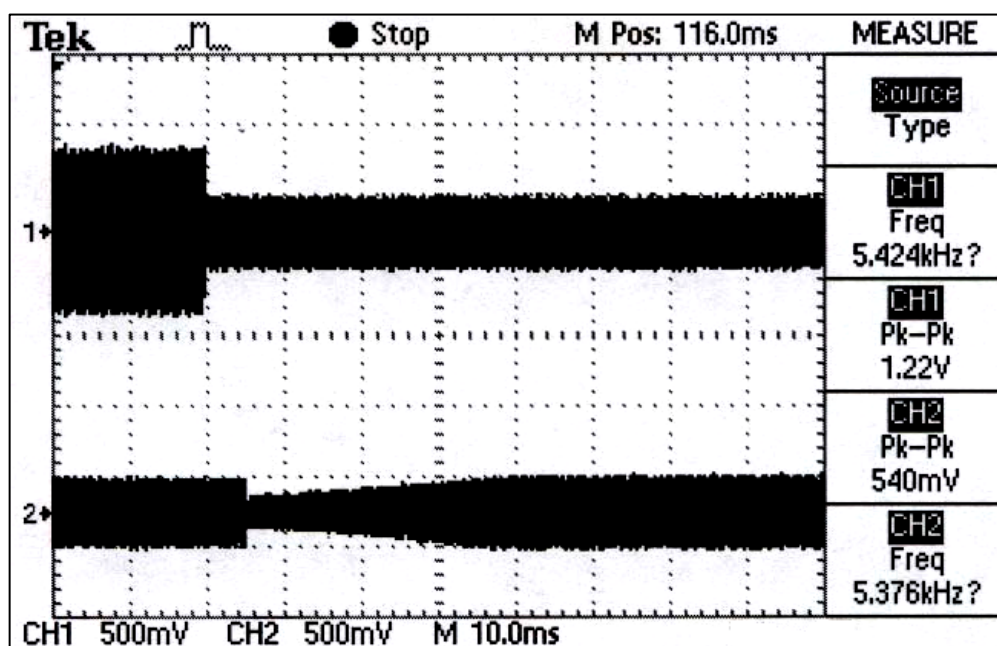
รูปที่ 4.12 ผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเข้ากระทำ 20 มิลลิวินาที



รูปที่ 4.13 ผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาดึกกระทำ 40 มิลลิวินาที



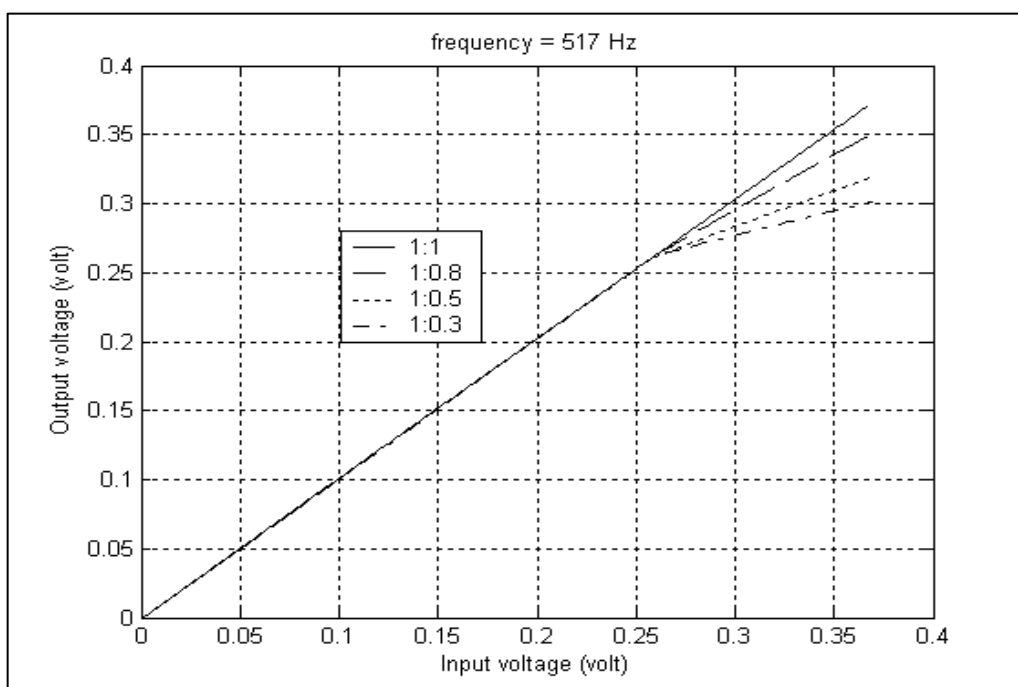
รูปที่ 4.14 ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเข้ากระทำ 10 มิลลิวินาที



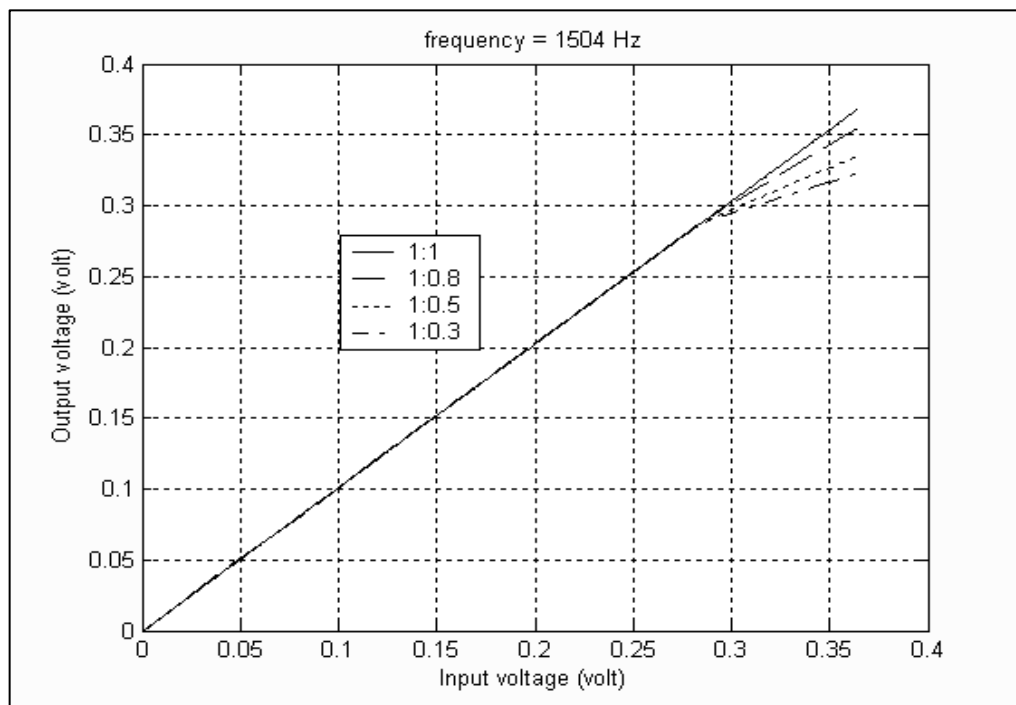
รูปที่ 4.15 ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเล็กระทำ 30 มิลลิวินาที

4.5 ผลการทดลองบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์

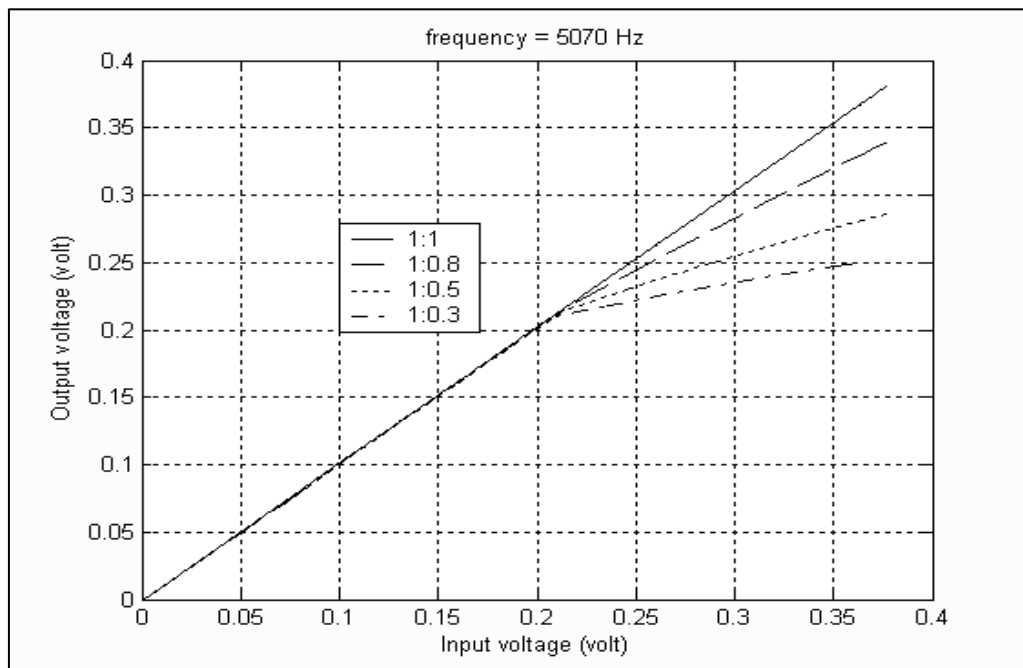
ผลการทดลองบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ โดยใช้ MATLAB พล็อตกราฟ (plot graph) เพื่อเปรียบเทียบสัญญาณที่ถูกบีบอัดในอัตราส่วนแตกต่างกัน ดังรูปที่ 4.16 แสดงผลการบีบอัดสัญญาณในอัตราส่วนที่ต่างกันของแบนด์ความถี่ผ่านต่ำ ซึ่งในแบนด์นี้ได้ทำการเปรียบเทียบสัญญาณที่ถูกบีบอัดในอัตราส่วนต่าง ๆ โดยกราฟเส้นตรงของสัญญาณมีอัตราการขยายเป็นหนึ่งในหนึ่ง (unity gain) ซึ่งสัญญาณไม่ถูกบีบอัด ในส่วนของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 คือสัญญาณอินพุตมีอัตราการขยายเป็นหนึ่งในหนึ่ง และเมื่อสัญญาณถูกบีบอัด เป็นผลให้สัญญาณเอาต์พุตที่ได้มีอัตราการขยาย 0.3 เท่า และสัญญาณที่อัตราส่วนอื่นมีการเปรียบเทียบเหมือนกันกับสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 แต่อัตราการขยายสัญญาณเอาต์พุตจะแตกต่างกันตามอัตราส่วนของสัญญาณที่ถูกบีบอัด และจากรูปที่ 4.17 แสดงผลการบีบอัดสัญญาณในอัตราส่วนที่ต่างกันของแบนด์ความถี่แถบผ่าน ส่วนในรูปที่ 4.18 แสดงผลการบีบอัดสัญญาณในอัตราส่วนที่ต่างกันของแบนด์ความถี่ผ่านสูง เนื่องจากสัญญาณที่แบนด์ความถี่นี้มีระดับความถี่สูงกว่าระดับแบนด์ความถี่อื่น ซึ่งบอร์ดสามารถตรวจจับสัญญาณได้ดีกว่า ทำให้สัญญาณมีระดับจุดเปลี่ยนต่ำ เป็นผลให้สัญญาณถูกบีบอัดที่ระดับจุดเริ่มเปลี่ยนต่ำกว่าแบนด์อื่น ๆ



รูปที่ 4.16 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่ผ่านต่ำ



รูปที่ 4.17 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่แถบผ่าน



รูปที่ 4.18 ผลการบีบอัดสัญญาณในอัตราส่วนที่แตกต่างกันของแบนด์ความถี่ผ่านสูง

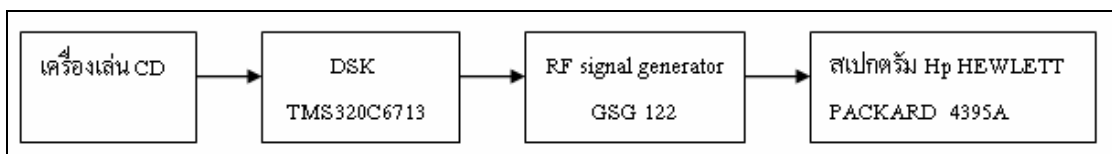
4.6 ผลการทดลองวัดสเปกตรัมการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์

ผลการทดลองวัดสเปกตรัมการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ จากรูปที่ 4.19 แสดงลักษณะการต่ออุปกรณ์เพื่อวัดสเปกตรัมบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์

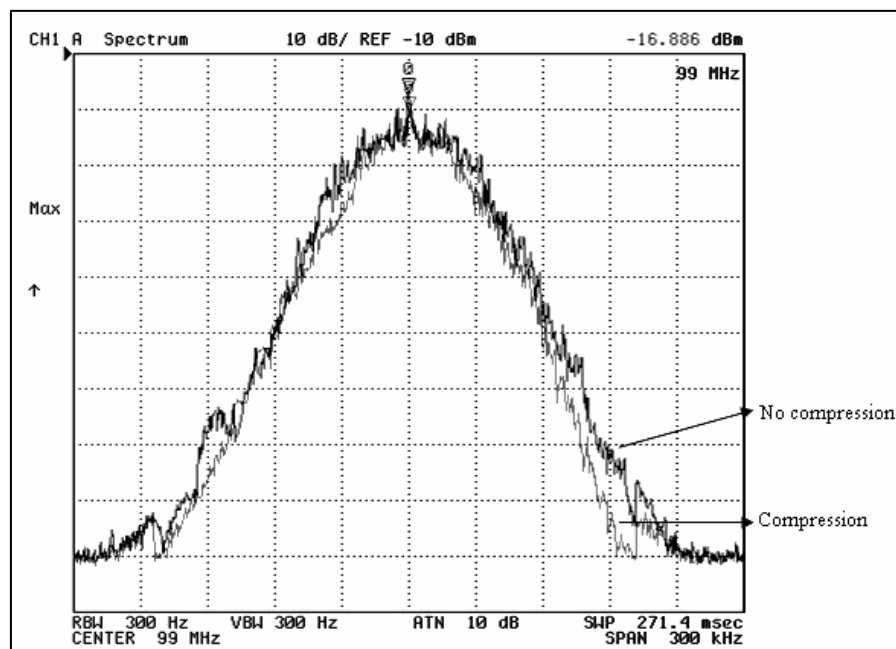


รูปที่ 4.19 ลักษณะการต่ออุปกรณ์เพื่อวัดสเปกตรัมบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์

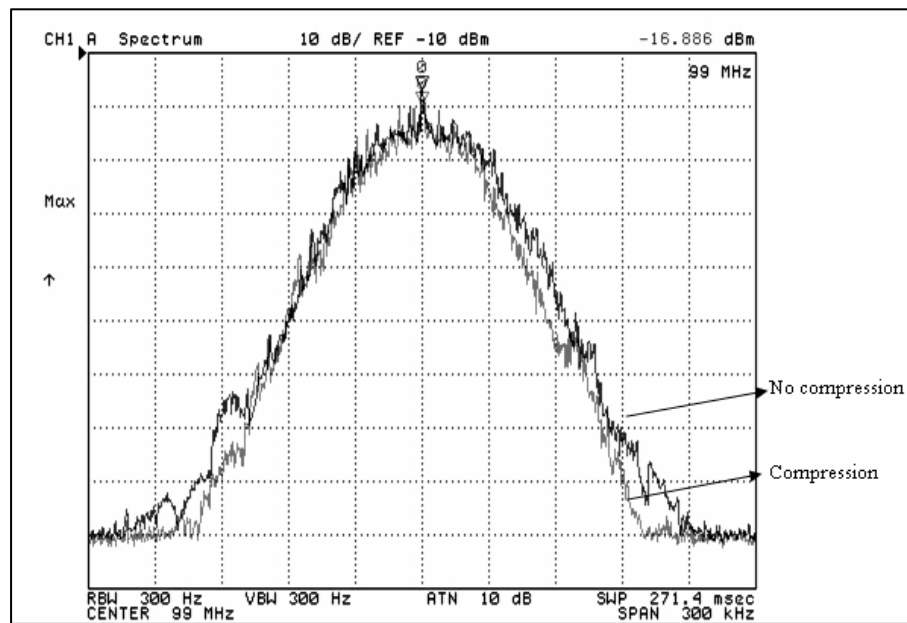
จากการทดลองรูปที่ 4.20 แสดงแผนภาพวิธีวัดสเปกตรัมบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ โดยป้อนสัญญาณเสียงจากเครื่องเล่น CD ซึ่งมีระดับแอมพลิจูดของสัญญาณเสียง 60 เเปอร์เซ็นต์ ซึ่งเป็นระดับแอมพลิจูดที่ตัวเข้ารหัส ถอดรหัสของบอร์ด DSK สามารถรองรับแอมพลิจูดได้สูงสุด ต่อจากนั้นทำการป้อนสัญญาณเสียงเข้าสู่บอร์ด DSK TMS320C6713 เพื่อทำการประมวลผลบีบอัดสัญญาณ จากนั้นนำสัญญาณที่ถูกบีบอัดป้อนเข้าสู่เครื่องกำเนิดสัญญาณความถี่ผ่านสูง (RF signal generator) รุ่น GSG-122 เพื่อทำการมอดูเลตแบบเอฟเอ็ม และวัดสเปกตรัมของสัญญาณเพื่อเปรียบเทียบสเปกตรัมของสัญญาณที่ถูกบีบอัด โดยใช้สเปกตรัมรุ่น Hp HEWLETT PACKARD 4395A



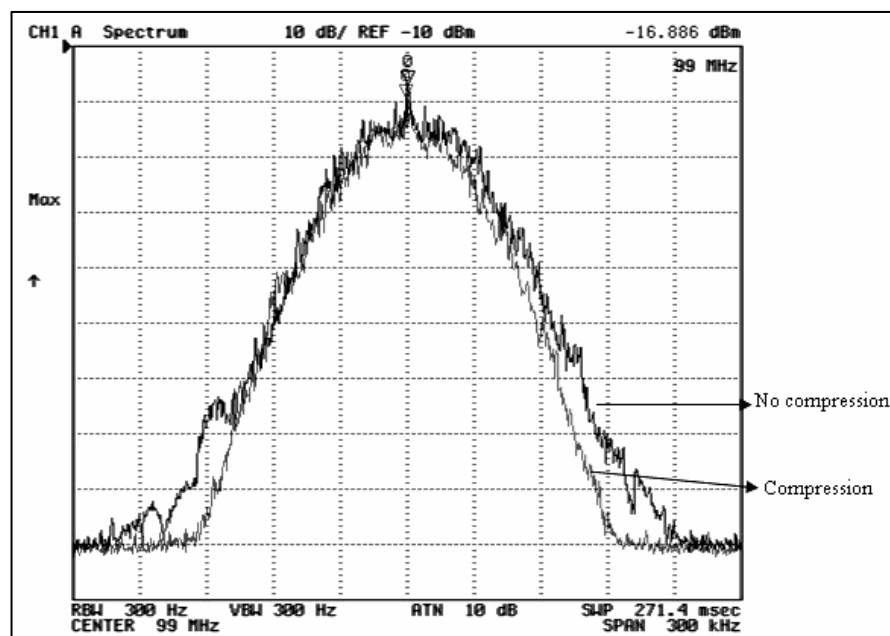
รูปที่ 4.20 แผนภาพวิธีวัดสเปกตรัมบีบอัดสัญญาณพิสัยพลวัตแบบหลายแบนด์



รูปที่ 4.21 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8
เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด



รูปที่ 4.22 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5
เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด



รูปที่ 4.23 สเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3
เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด

จากผลการทดลองในรูปที่ 4.21 แสดงสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด พบว่าสเปกตรัมของสัญญาณที่ได้จากการถูกบีบอัดมีแบนด์วิดท์เท่ากับ 210 กิโลเฮิร์ตซ์ ทำให้แบนด์วิดท์ที่ถูกบีบอัดแคบลงกว่าเมื่อเทียบกับสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด ซึ่งมีแบนด์วิดท์เท่ากับ 240 กิโลเฮิร์ตซ์ และเมื่อพิจารณาผลการทดลองจากรูปที่ 4.22 แสดงสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด ซึ่งมีแบนด์วิดท์แคบลงมากกว่าสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 และในรูปที่ 4.23 แสดงสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 เปรียบเทียบสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด โดยสัญญาณที่ถูกบีบอัดมีแบนด์วิดท์เท่ากับ 186 กิโลเฮิร์ตซ์ จะทำให้แบนด์วิดท์แคบลงมากกว่าสัญญาณที่ถูกบีบอัดในอัตราส่วนอื่น ๆ ที่ได้กล่าวมา และส่วนของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 จะเป็นสัญญาณที่ถูกบีบอัดมากที่สุด

4.7 สรุป

ในบทนี้เป็นการทดลองผลตอบสนองของตัวกรอง ผลตอบสนองของช่วงเวลาเข้ากระทำ และช่วงเวลาเล็กระกระทำ เพื่อเปรียบเทียบการบีบอัดสัญญาณพัลส์พลวัตแต่ละแบนด์ในอัตราส่วนต่าง ๆ และวัดสเปกตรัมการบีบอัดสัญญาณพัลส์พลวัต เพื่อแสดงว่าสามารถนำไปใช้งานได้จริงในการบีบอัดสัญญาณเพื่อส่งกระจายเสียงในระบบเอฟเอ็ม

จากผลการทดลองที่ได้จากการออกแบบตัวกรองผ่านต่ำ ตัวกรองผ่านแถบ และตัวกรองผ่านสูง ซึ่งใช้โปรแกรม FDATool ช่วยในการออกแบบตัวกรองทั้งหมด ต่อจากนั้นนำมาประยุกต์ใช้กับบอร์ด DSK TMS320C6713 เพื่อสังเกตการประมวลผลตอบสนองของตัวกรอง พบว่าตรงตามคุณสมบัติของตัวกรองที่ได้ออกแบบ

ผลการทดลองจากการวัดช่วงเวลาเข้ากระทำและช่วงเวลาเล็กระกระทำของแต่ละแบนด์ โดยกำหนดช่วงเวลาเข้ากระทำเร็วกว่าช่วงเวลาเล็กระกระทำ เพื่อให้สัญญาณเข้าสู่กระบวนการบีบอัดสัญญาณได้ทัน เมื่อสัญญาณมีระดับแอมพลิจูดสูงกว่าจุดเริ่มเปลี่ยนที่กำหนดไว้ โดยแบนด์ความถี่ผ่านต่ำมีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที และช่วงเวลาเล็กระกระทำ 50 มิลลิวินาที แบนด์ความถี่แถบผ่านมีช่วงเวลาเข้ากระทำ 20 มิลลิวินาทีและช่วงเวลาเล็กระกระทำ 40 มิลลิวินาที ส่วนแบนด์ความถี่ผ่านสูงมีช่วงเวลาเข้ากระทำ 10 มิลลิวินาทีและช่วงเวลาเล็กระกระทำ 30 มิลลิวินาที พบว่าเมื่อนำมาประยุกต์ใช้กับบอร์ดในการประมวลผล ได้ผลตอบสนองของช่วงเวลาเข้ากระทำและช่วงเวลาเล็กระกระทำของแต่ละแบนด์ตรงตามที่คาดหมายไว้

ผลการทดลองบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ ซึ่งเป็นการเปรียบเทียบสัญญาณที่ถูกบีบอัดในอัตราส่วนต่าง ๆ จากผลการทดลองในแบนด์เดียวกัน โดยกำหนดสัญญาณที่ถูกบีบอัดในอัตราส่วนต่างกัน พบว่าสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 ซึ่งเป็นสัญญาณที่ถูกบีบอัดมากกว่า

สัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5 และ 1 : 0.8 ซึ่งตรงตามที่คาดหมายในการบีบอัดพัลส์พลวัตแบบหลายแบนด์

ผลการทดลองวัดสเปกตรัมการบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ พบว่าสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 ทำให้ได้แบนด์วิดท์แคบลงเหลือเพียง 186 กิโลเฮิร์ตซ์ ซึ่งเป็นอัตราส่วนที่สัญญาณถูกบีบอัดมากที่สุดในการทดลอง ส่งผลให้แบนด์วิดท์แคบลงมากเมื่อเปรียบเทียบกับสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 มีแบนด์วิดท์เท่ากับ 210 กิโลเฮิร์ตซ์ และสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5 มีแบนด์วิดท์เท่ากับ 198 กิโลเฮิร์ตซ์ พบว่าสัญญาณถูกบีบอัดน้อยกว่าสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.3 จึงทำให้แบนด์วิดท์ที่ได้แคบลงเล็กน้อย ซึ่งจากผลการทดลองนี้แสดงว่า การบีบอัดสัญญาณพัลส์พลวัตแบบหลายแบนด์ สามารถแก้ไขปัญหาการเกิดโอเวอร์มอดูเลชันได้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 กล่าวนำ

วิทยานิพนธ์ฉบับนี้ได้นำเสนอ กระบวนการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ เพื่อป้องกันการเกิดโอเวอร์มอดูเลชันในการส่งกระจายเสียงระบบเอฟเอ็ม โดยใช้ชิพ DSP TMS320C6713 ประมวลผลจากการบีบอัดสัญญาณ ทำการแบ่งสเปกตรัมของสัญญาณเสียงออกเป็น 3 แบนด์ความถี่ โดยออกแบบตัวกรองเอฟไออาร์ 3 ตัวกรองคือ ตัวกรองผ่านต่ำ กำหนดความถี่ตัด 800 เฮิรตซ์ ตัวกรองแถบผ่าน กำหนดความถี่ตัด 820 เฮิรตซ์ และ 3 กิโลเฮิรตซ์ ส่วนตัวกรองผ่านสูง กำหนดความถี่ตัด 3.2 กิโลเฮิรตซ์ ซึ่งตัวกรองเอฟไออาร์ทั้งหมดนี้มีอัตราการสุ่มความถี่ 44.1 กิโลเฮิรตซ์ มีสัมประสิทธิ์ของตัวกรองยาว 61 ลำดับและใช้หน้าต่างแฮมมิง โดยแต่ละแบนด์ความถี่ ประกอบด้วยกระบวนการบีบอัดสัญญาณพิสัยพลวัต มีพารามิเตอร์ช่วงเวลาเข้ากระทำและช่วงเวลา เลิกกระทำ โดยกำหนดให้ความถี่ผ่านต่ำมีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที และช่วงเวลาเลิกกระทำ 50 มิลลิวินาที ส่วนความถี่แถบผ่านกำหนดช่วงเวลาเข้ากระทำ 20 มิลลิวินาที ช่วงเวลาเลิกกระทำ 40 มิลลิวินาที และที่ความถี่ผ่านสูงกำหนดช่วงเวลาเข้ากระทำ 10 มิลลิวินาที ช่วงเวลาเลิกกระทำ 30 มิลลิวินาที โดยอ้างอิงจากเว็บไซต์ (www.alesis.com) ซึ่งจากการทดลองพบว่า ผลตอบสนองของ ความถี่ผ่านต่ำมีช่วงเวลาเข้ากระทำ 30 มิลลิวินาที จะช้ากว่าผลตอบสนองช่วงเวลาเข้ากระทำของ แบนด์ความถี่อื่น เนื่องจากแบนด์ความถี่นี้มีแอมพลิจูดต่ำ ดังนั้นโอกาสที่จะเกิดแอมพลิจูดสูงมีเพียง เล็กน้อย เมื่อเทียบกับผลตอบสนองของความถี่แถบผ่านที่มีช่วงเวลาเข้ากระทำ 20 มิลลิวินาที และ ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเข้ากระทำ 10 มิลลิวินาที จะเห็นว่าแบนด์ความถี่สูงมี ช่วงเวลาเข้ากระทำเร็ว เนื่องจากแบนด์ความถี่สูงมีผลให้แอมพลิจูดสูง ทำให้เกิดปัญหาโอเวอร์มอดูเล ชัน ดังนั้นจึงต้องกำหนดช่วงเวลาเข้ากระทำที่แบนด์ความถี่สูงเร็วขึ้น เมื่อพิจารณาผลการทดลอง ผลตอบสนองของความถี่ผ่านต่ำมีช่วงเวลาเลิกกระทำ 50 มิลลิวินาทีที่มีผลตอบสนองช่วงเวลาเลิก กระทำช้ากว่า ผลตอบสนองของความถี่แถบผ่านมีช่วงเวลาเลิกกระทำ 40 มิลลิวินาที และ ผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเลิกกระทำ 30 มิลลิวินาที เพื่อป้องกันไม่ให้เกิดการสะดุด ระหว่างรอยต่อระดับสัญญาณปกติกับระดับสัญญาณที่ถูกบีบอัด และเมื่อพิจารณาผลตอบสนองของ ความถี่แถบผ่านมีช่วงเวลาเลิกกระทำ กับผลตอบสนองของความถี่ผ่านสูงมีช่วงเวลาเลิกกระทำเร็วขึ้น ตามลำดับ ถ้าที่แบนด์ความถี่สูงกำหนดช่วงเวลาเลิกกระทำช้าลง จะทำให้สัญญาณปกติถูกบีบอัด

จากผลการทดลอง เมื่อทำการวัดสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 ซึ่ง ทำให้ได้ผลการทดลองมีแบนด์วิดท์เท่ากับ 210 กิโลเฮิรตซ์ โดยเทียบกับสเปกตรัมของสัญญาณที่ไม่

ถูกบีบอัด ซึ่งมีแบนด์วิดท์เท่ากับ 240 กิโลเฮิร์ตซ์ พบว่าสเปกตรัมของสัญญาณที่ถูกบีบอัดมีแบนด์วิดท์แคบลงเพียงเล็กน้อย เนื่องจากสัญญาณที่ถูกบีบอัดในอัตราส่วนนี้ สัญญาณจะถูกบีบอัดน้อยเมื่อเปรียบเทียบกับสัญญาณที่ถูกบีบอัดในอัตราส่วนอื่น ทำการปรับอัตราส่วนสัญญาณที่ถูกบีบอัดเป็น 1 : 0.5 ซึ่งผลการทดลองมีแบนด์วิดท์เท่ากับ 198 กิโลเฮิร์ตซ์ ทำให้แบนด์วิดท์แคบลงมากกว่าเมื่อเปรียบเทียบกับสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.8 และทำการปรับอัตราส่วนของสัญญาณที่ถูกบีบอัดเป็น 1 : 0.3 ได้แบนด์วิดท์เหลือเพียง 186 กิโลเฮิร์ตซ์ ซึ่งสัญญาณที่ถูกบีบอัดที่อัตราส่วนนี้ สัญญาณจะถูกบีบอัดมากที่สุดเมื่อเทียบกับสัญญาณที่ถูกบีบอัดในอัตราส่วนที่กล่าวมาข้างต้น และจากผลการทดลองทั้งหมดนี้ แสดงให้เห็นว่ากระบวนการบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ สามารถแก้ไขปัญหาคาการเกิดโอเวอร์มอดูเลชันได้ เนื่องจากแบนด์วิดท์ที่ได้แคบลงเมื่อเปรียบเทียบกับสเปกตรัมของสัญญาณที่ไม่ถูกบีบอัด ซึ่งสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วนที่เหมาะสม คือ 1 : 0.5 มีแบนด์วิดท์เท่ากับ 198 กิโลเฮิร์ตซ์ ตามมาตรฐานเอฟซีซี (federal communication commission หรือ FCC) กำหนดให้แบนด์วิดท์การส่งกระจายเสียงในระบบเอฟเอ็มมีความกว้างไม่เกิน 200 กิโลเฮิร์ตซ์ ซึ่งจากการทดลองวัดสเปกตรัมของสัญญาณที่ถูกบีบอัดในอัตราส่วน 1 : 0.5 ได้แบนด์วิดท์กว้างไม่เกินมาตรฐานที่กำหนดไว้

5.2 ข้อเสนอแนะและแนวทางการพัฒนา

จากการทดลองบีบอัดสัญญาณเสียงพิสัยพลวัตแบบหลายแบนด์ เมื่อพิจารณาการออกแบบตัวกรองเอฟไออาร์ ซึ่งมีสัมประสิทธิ์เพียง 61 ลำดับ แต่ถ้าต้องการออกแบบตัวกรองเอฟไออาร์ให้มีสัมประสิทธิ์มากขึ้น เพื่อความคมของจุดตัดความถี่และสามารถประยุกต์เป็นตัวปรับเท่า (equalizer) ซึ่งต้องเพิ่มแบนด์มากขึ้น เป็นผลให้ต้องเพิ่มตัวกรองเพื่อแบ่งความถี่ออกเป็นหลายแบนด์ โดยบอร์ด DSK TMS320C6713 มีข้อจำกัดทางด้านบัฟเฟอร์และข้อจำกัดในการประมวลผล ดังนั้นควรเพิ่มหน่วยความจำ เพื่อให้สามารถประมวลผลได้เร็ว

นอกจากนี้ในการเขียนโปรแกรมใช้มาตรฐานอัลกอริทึมของบริษัท TI ทำให้สามารถนำไปพัฒนาต่อได้สะดวก แต่มาตรฐานอัลกอริทึมจะต้องใช้เวลาในการศึกษานานและบางอัลกอริทึมไม่สามารถศึกษาได้อย่างลึกซึ้ง อีกประการหนึ่งต้องระมัดระวังเส้นตาย (deadline) ซึ่งเป็นการทำงานของสายโซ่ จากปฏิบัติการของกราฟข้อมูล DSP/BIOS จากวัตถุ (object) ของโปรแกรม

รายการอ้างอิง

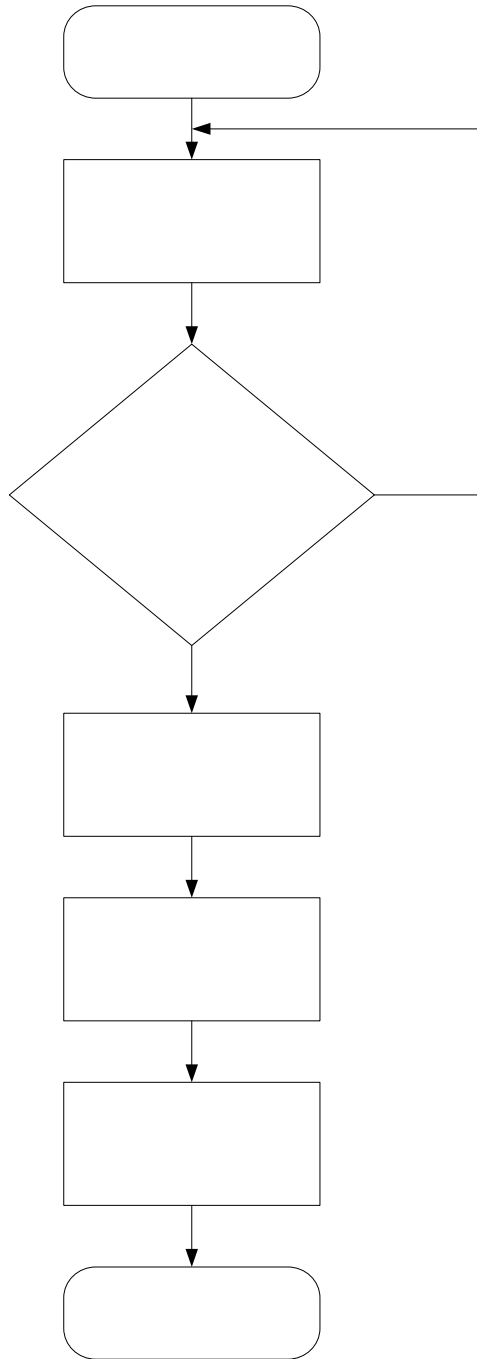
- ดิษฐ์กร วุฒิสัทธาภิบาลกิจ. (2547). **พื้นฐานกรรมวิธีสัญญาณดิจิทัล**. กรุงเทพฯ : จุฬาลงกรณ์มหาวิทยาลัย.
- วิโรจน์ แก้วจันทร์. (2543). **ทฤษฎีเครื่องส่งวิทยุ**. กรุงเทพฯ : สยามสปอร์ต ซินดิเคท
- สุชาติ กังวานจิตต์. (2538). **หลักการทํางานเครื่องรับส่งวิทยุ และระบบวิทยุสื่อสาร**. (หน้า 32)
กรุงเทพฯ: บริษัท ซีเอ็ดดูเคชั่น จำกัด
- Alesis 3630 Compressor. (2006). Alesis 3630 Compressor [Online]. Available: <http://www.alesis.com>
- Brouckxon, H., Verhelst, W., Struyve, and Verhoeve P. (2006). Design and Evaluation of a Microphone Signal Conditional System [Online]. Available: <http://www.etro.vub.ac.be>.
- Brouckxon, H., Struyve, K., and Verhelst, W. (2006). Microphone Signal Conditioning with the TMS320C6713 [Online]. Available: <http://www.etro.vub.ac.be>.
- De Gennaro, S. Krieg, K. Braida, L. and Durlach N. (1981). Third-octave Analysis of Multichannel Amplitude Compressed Speech. **IEEE International on Acoustics, Speech, and signal processing** . AP-6: 125-128.
- Earl, Vickers. (2001). **Automatic Long-Term Loudness and Dynamics**. New York. USA
- Emmanue, and Barrie., (2002). **Digital signal Processing**. (pp. 350-370)., Prentice-Hall
- Jon, C., Schmidt, and Janet, C., Rutledge. (1996). Multichannel Dynamic Range Compression For Music Signals. **IEEE Internatonal conference on Acoustic, Speech and Signal processing** . AP-2: 1013-1016
- Lamancusa, J. S. (2000). **Human Response to Sound** [Online]. Available: <http://www.fdt.org>.(p.3.3).
- Leslie, Balmer, (1997). **Signal and System**. (pp. 374-375)., Prentice-Hall.
- Loizou.(1998). Mimicking the Human Ear. **IEEE Signal Processing magazine**. AP-15(5) : 101-130.
- Luzheng, LU. (1998). A Digital Realization of Audio Dynamic Range Control. **IEEE 4th International Conference on Signal Processing**. AP-2: 1424 -1427.
- Michael. (1999). **Audio Engineer's reference book**. (pp.1-38). Prentice-Hall.

- Neeraj, Sanmati, Frank, and Maria. (2000) Development and Fixed-point Implementation of A Multiband Dynamic Range Compression (MDRC) Algorithm. **Conference Record of the 4th Asilomar on signals, systems and Computers**. AP-1:428 -432
- Orban. (2004). Optimod 8400 [Online]. Available: [http:// www.Orban.com](http://www.Orban.com).
- Perttu (2002). Smoothing of The Control Signal Without Clipped Output in Digital Peak Limiters. **5th International Conference on Digital Audio Effect**. (pp. 1-4).
- Ryan, J. Cassidy. (2004). Dynamic Range Compression of Audio Signals Consistent with Recent with Recent Time-Varying Loudness Models. **IEEE International Conference on Acoustic, Speech and Signal processing** . AP-4: 213-216.
- Schmidt, J.C. and Rutledge, J.C. (1997). Multiband Compression Hearing Aids:Developing A Performance Metric. **International Symposium on Circuits and Systems** AP-1: 573 - 577).
- Schneider, T. and Brennan, R. (1997). A Multichannel Compression Strategy for A Digital Hearing Aid. **IEEE International on Acoustic, Speech and Signal processing** . AP-1: 411-414.
- Shugang, and Kensuke., (2005). Dynamic Range Compression Characteristics Using an Interpolating Polynomial for Digital Audio Systems. **IEICE Trans. Fundamentals**. AP-E88: 586-589.
- Texas Instrument. TMS320C6713 [Online]. Available: [http:// www.TI.com](http://www.TI.com)
- Ventura, J. C. (1989). Digital Audio Gain Control for Hearing Aids. **IEEE Internatonal conference on Acoustic, Speech and Signal processing** . AP-3: 2049-2052.

ภาคผนวก ก

โปรแกรม

แผนผังโปรแกรม



St

Input PIP

ตรวจสอบ
เต็มและ 0
ว่างห

```

main

#include <std.h>
#include <pip.h>
#include <string.h>
#include <utl.h>          /* debug/diagnostics utility functions */
#include <math.h>
#include "appResources.h" /* application-wide common info */
#include "appThreads.h"  /* thread-wide common info */
#include "fir.h"          /* interface for FIR algorithm */
#include "vol.h"          /* interface for VOL algorithm */
#include "thrAudioproc.h" /* definition of thrAudioproc objects */

#define NumCoeff 61
#define SamRate 44100
const float vth = 5461.16;

/* attack time and release time at lowpass filter*/
const float tl= 0.03 ;
const float rl= 0.05;
#define atl ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) tl))));
#define arl ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) rl))));
/* attack time and release time at bandpass filter*/
const float tb= 0.02;
const float rb= 0.04;
#define atb ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) tb))));
#define arb ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) rb))));
/* attack time and release time at highpass filter*/
const float th = 0.01;
const float rh = 0.03;
#define ath ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) th))));
#define arh ((float) exp((float)((float) -1.0 / ((float) SamRate * (float) rh))));
short buffer[NUMCHANNELS][3][FRAMELEN];
short oldInput[NUMCHANNELS][FRAMELEN+NumCoeff];

```

```

short oldOut[NUMCHANNELS][3];
/* Functions */
extern void load(unsigned int loadValue);
/* Static part of thread initialization */
/*
* Initialization of the thread resources structure:
* NULL for algorithm handles, addresses of appropriate
* pipe objects for input and output pipes,
* NULL for globally visible temporary pointers to pipe frames,
* addresses of intermediate buffers for this thread,
* and everything else thread-specific.
*/
ThrAudioproc thrAudioproc[ NUMCHANNELS ] = {
    { /* channel #0 */
        /* algorithm handle(s) (to be initialized in runtime) */
        NULL,          /* algFIR */
        NULL,          /* algVOL */
        /* input pipe(s) */
        &pipRx0,        /* pipIn */
        /* output pipe(s) */
        &pipTx0,        /* pipOut */
        /* buffer(s) */
        bufAudioproc,  /* bufInterm */
        /* everything else private for the thread */
    },
    /* end channel # 0 */
    /* channel #1 */
    /* algorithm handle(s) (to be initialized in runtime) */
    {
        NULL,          /* algFIR */
        NULL,          /* algVOL */

```



```

/* input pipe(s) */
    &pipRx1,      /* pipIn */
/* output pipes */
    &pipTx1,      /* pipOut */
/* buffer(s) */
    bufAudioproc, /* buffer */
/* everything else private for the thread */
}, /* end channel # 1 */
};

static Sample filterCoefficientsNew[3][61] = {
/*short hl[NumCoeff] = low800Hz */
{ -28, -29, -32, -35, -39, -42, -43, -41, -34,
  -21,  0,  30,  71, 124, 188, 266, 354, 454,
  564, 681, 803, 928, 1051, 1171, 1283, 1384, 1472,
  1543, 1595, 1628, 1638, 1628, 1595, 1543, 1472, 1384,
  1283, 1171, 1051, 928, 803, 681, 564, 454, 354,
  266, 188, 124, 71, 30, 0, -21, -34, -41,
  -43, -42, -39, -35, -32, -29, -28
},
/*short hb[NumCoeff] = band820-3k*/
{  2,  1,  9,  29,  60,  95, 121, 123,  85,
  6, -102, -206, -269, -259, -172, -41,  65,  58,
 -135, -541, -1110, -1713, -2167, -2279, -1911, -1029, 267,
 1755, 3136, 4112, 4465, 4112, 3136, 1755, 267, -1029,
 -1911, -2279, -2167, -1713, -1110, -541, -135,  58,  65,
 -41, -172, -259, -269, -206, -102,  6,  85, 123,
 121,  95,  60,  29,  9,  1,  2
},
/*short hh[NumCoeff] = high3.2k*/
{ -25, -18, -7, 10, 31, 54, 73, 78, 63,

```

```

    20, -49, -134, -216, -271, -272, -198, -44, 178,
    430, 658, 794, 770, 536, 65, -626, -1486, -2424,
-3329, -4082, -4581, 28013, -4581, -4082, -3329, -2424, -1486,
-626, 65, 536, 770, 794, 658, 430, 178, -44,
-198, -272, -271, -216, -134, -49, 20, 63, 78,
    73, 54, 31, 10, -7, -18, -25
},
};

void compress( short x[],short *oldOut, int a)
{
int i=0;
float g[FRAMELEN];
float c[FRAMELEN];
float y[FRAMELEN];
double at,an,ar;
    if (a==0)
        { at=atl; ar=arl;}
    else if(a==1)
        { at=atb; ar=arb;}
    else
        { at=ath; ar=arh;}
    if (abs(x[0]) > *(oldOut))
        {an = at;}
    else
        {an = ar; }
    c[0] = (float)((float)(1-an)* (abs((float)x[0]))) +(an*(*(oldOut)));
    for( i=0;i<FRAMELEN;i++)
        {
            if (i > 0)
                {
                    if (abs(x[i]) > c[i-1])

```

```

        {an = at;}
    else
        {an = ar;}
    c[i] = (float)((float)(1-an)*(abs((float)x[i]))) +(an*(float)(c[i-1]));
    }
    if (abs(c[i]) >= vth)
    {
        g[i] = ((float)(1.24))*((float)powsp((float)(c[i] / (float)vth),(float)-0.5));
    }
    else
        { g[i] = (float)1.24;}
    x[i] =(float) x[i]*g[i];
}
*oldOut = c[FRAMELEN-1];
};

/* Dynamic part of thread initialization*/
/*
* ===== thrAudioprocInit =====
* Initialization of data structures for the thread(s), called from
* appThreads.c:thrInit() at init time.
* Here we create one instance of FIR algorithm per channel and
* one instance of VOL algorithm per channel. In a loop, we create
* parameters for algorithm instance for each channel by using the
* default parameters and modifying fields that are different.
* (If the parameters are the same across channels, they can be
* created outside of the loop.)
*/

Void thrAudioprocInit( Void )
{
    /* declaration of filter, volume parameter structures */
    FIR_Params firParams;

```

```

VOL_Params volParams;

Int i,loop;

    for (i = 0; i < NUMCHANNELS; i++) {
        for(loop = 0; loop <140 ;loop++)
            oldInput[i][loop]=0;
        for(loop =0;loop <3 ;loop++)
            oldOut[i][loop] = 0;
        /*
        * Set the parameters structure to the default, i.e.
        * the one used in i<alg>.c, and modify fields that are different.
        */
        firParams = FIR_PARAMS;                                /* default parameters */
        firParams.coeffPtr = (Short *)filterCoefficientsNew[i]; /* filter coefficients */
        firParams.filterLen = sizeof( filterCoefficientsNew[i]) / sizeof( Sample );
        firParams.frameLen = FRAMELEN; /* frame size in samples */
        /* create algorithm instance for channel #i */
        thrAudioproc[i].algFIR = FIR_create( &FIR_IFIR, &firParams );
        /* Confirm that the instantiation was successful. If it failed,
        * most likely the heap is not big enough. To find out the needed
        * value (rather than to guess), in appThreads.c you can do
        * ALGRF_setup( EXTERNALHEAP, EXTERNALHEAP ); i.e. force all
        * allocation in external memory, run the initialization functions,
        * and examine the reports from UTL_showAlgMem() below.
        */
        TL_assert( thrAudioproc[i].algFIR != NULL );
        /* and show algorithm memory usage */
        UTL_showAlgMem( thrAudioproc[i].algFIR );
        /* do the same for the VOLUME algorithm: create parameters structure */
        volParams = VOL_PARAMS;                                /* default parameters */
        volParams.frameSize = FRAMELEN; /* size in samples */
        volParams.gainPercentage = 100; /* default gain */
    }

```

```

/* create instance, confirm creation success, show memory usage */
thrAudioproc[i].algVOL = VOL_create( &VOL_IVOL, &volParams );
UTL_assert( thrAudioproc[i].algVOL != NULL );
UTL_showAlgMem( thrAudioproc[i].algVOL );
    }
}
/*
 * Runtime thread code, invoked by the appropriate SWI object
 * every time the object is posted
 */
/*
 * ===== thrAudioprocRun =====
 * The "body" of the swiAudioproc0, swiAudioproc1,... threads.
 * The single argument of this function is the channel number:
 * 0, 1, 2 etc. up to NUM_CHANNELS - 1. All the SWI objects
 * that invoke this function pass the channel number as the
 * argument.
 * Based on the channel number, the thread -- the procedure --
 * decides which thread resource object to access.
 */
Void thrAudioprocRun( Arg aChan )
{
Sample *src, *dst;
Int  size;    /* in samples */
Int  chan;
/* VOL algorithm control structure */
IFIR_Status status;
Int  numFilter;
Sample tmpBuffer[FRAMELEN+NumCoeff-1];
Int  loop;
Int  cases;

```

```

/* cast 'Arg' types to 'Int'. This is required on 55x large data model
 * since Arg is not the same size as Int and Ptr in that model.
 * On all other devices (54x, 55x small, 6x) ArgToInt is a simple cast
 */
chan = ArgToInt(aChan);

/* Check that the preconditions are met, that is the in-pipe
 * has a ready-to-consume buffer of data and the out-pipe
 * has a free buffer, in other words that this thread has
 * not been posted in error.
 */
UTL_assert( PIP_getReaderNumFrames( thrAudioproc[chan].pipIn ) > 0 );
UTL_assert( PIP_getWriterNumFrames( thrAudioproc[chan].pipOut ) > 0 );

/* get the full buffer from the input pipe */
PIP_get( thrAudioproc[chan].pipIn );
src = (Sample *)PIP_getReaderAddr( thrAudioproc[chan].pipIn );
/* get the size in samples (the function below returns it in words) */
size = sizeInSamples( PIP_getReaderSize( thrAudioproc[chan].pipIn ) );
/* get the empty buffer from the out-pipe */
PIP_alloc( thrAudioproc[chan].pipOut );
dst = (Sample *)PIP_getWriterAddr( thrAudioproc[chan].pipOut );
    {
        for (numFilter =0; numFilter<3 ;numFilter++) // chang number for individual band
            {
                status.coeffPtr = filterCoefficientsNew[numFilter];
                FIR_control(thrAudioproc[chan].algFIR,IFIR_SETSTATUS,&status);
/* apply filter and store result in intermediate buffer */
                FIR_apply( thrAudioproc[chan].algFIR, src, thrAudioproc[chan].bufInterm
);
/*Copy Filtered data into buffer*/

```

```

        memcpy((void *)buffer[chan][numFilter] ,(void
*)thrAudioproc[chan].bufInterm,(FRAMELEN* sizeof(XDAS_Int16)));
/** *****/
/** ***** COMPRESSION HERE *****/
/** *****/

        compress(buffer[chan][numFilter],&(oldOut[chan][numFilter]),numFilter);

        if (numFilter <2)
        {
                /* return shift because we process old data*/
                FIR_apply( thrAudioproc[chan].algFIR, oldInput[chan],
thrAudioproc[chan].bufInterm );
        }
}

for (loop=0;loop<FRAMELEN;loop++)
        buffer[chan][0][loop]+=buffer[chan][1][loop]+buffer[chan][2][loop];

/* Copy processed data to bufInterm for represent output */
        memcpy((void *)thrAudioproc[chan].bufInterm,(void
*)(buffer[chan][0]),(FRAMELEN* sizeof(XDAS_Int16)));
}

        memcpy((void *)oldInput[chan] ,(Void *)src, sizeof(XDAS_Int16)*140);

/* amplify the signal in the interm. buffer and store result in dst */
VOL_apply( thrAudioproc[chan].algVOL, thrAudioproc[chan].bufInterm, dst );

/* Record the amount of actual data being sent */
PIP_setWriterSize( thrAudioproc[chan].pipOut, sizeInWords( size ) );

/* Free the receive buffer, put the transmit buffer */
PIP_free( thrAudioproc[chan].pipIn );
PIP_put ( thrAudioproc[chan].pipOut );
}

/* ===== thrAudioprocSetOutputVolume =====
* Procedure that changes output volume for an Audioproc thread
* This procedure is called from the likes of the Control thread.

```

```
* Instead of having the control thread directly write to
* thread Audioproc's variables, it calls this function which
* changes the state of the VOL algorithm instance for the
* given channel.
*/
Void thrAudioprocSetOutputVolume( Int chan, Int volume )
{
/* VOL algorithm control structure */
IVOL_Status status;
    UTL_assert( chan < NUMCHANNELS ); /* sanity checking */
/* Apply volume gain information to the appropriate alg. instance.
* Retrieve the current parameters, change the fields that need
* to be changed, and apply the new parameters. Volume gain is
* a percentage, a number from 0 to 200, 100 being the normal
* volume (100%).
*/
VOL_control( thrAudioproc[ chan ].algVOL, IVOL_GETSTATUS, &status );
status.gainPercentage = volume;
VOL_control( thrAudioproc[ chan ].algVOL, IVOL_SETSTATUS, &status );
}
```


ภาคผนวก ข

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

จุฑาทพร มากอยู่ (2549). การบีบอัดสัญญาณเสียงพัลส์พลวัตแบบหลายแบนด์ สำหรับการส่งกระจายเสียงระบบเอฟเอ็ม โดยใช้ DSP TMS320C6713. การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 29 (EECON-29),, หน้า 1049-1052

