

วงจรรองความถี่ชนิด FIR

FIR Digital Filter



โดย

นายการเอก เรืองชัยชาญกิจ

นายศุภชัย เกรียงธนสาร

ปริญญาานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหม.....

เลขทะเบียน..... 46466

วัน, เดือน, ปี..... 2 เม.ย. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรองความถี่ชนิด FIR

FIR Digital Filter

โดย

นายการเอก เรืองชัยชาญกิจ 42015003

นายศุภชัย เกรียงชนสาร 42015035

อาจารย์ที่ปรึกษา

ผศ. เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร. สุวิพล สิทธีชีวกาศ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

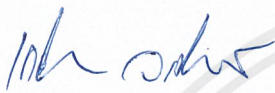
เรื่อง วจรกรองความถี่ชนิด FIR

FIR Digital Filter

ผู้จัดทำ

นายการเอก เรืองชัยชาญกิจ 42015003

นายศุภชัย เกรียงธนาสาร 42015035



(ผศ.เกรียงไกร วงศ์โรจนกรณ)

อาจารย์ที่ปรึกษา

(รศ.ดร สุวิพล สิริชีวะภาค)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรองความถี่ชนิด FIR

FIR Digital Filter

โดย นายการเอก เรืองชัยชาญกิจ 42015003

นายศุภชัย เกียรติธนสาร 42015035

อาจารย์ที่ปรึกษา ผศ.เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

โครงการนี้สามารถนำเสนอการศึกษา ออกแบบ และสร้างวงจรรองความถี่ แบบเชิงตัวเลข ชนิดที่สามารถโปรแกรมแถบกรองความถี่ได้ ควบคุมโดยใช้คอมพิวเตอร์ โดยใช้ TMS320C31 DIGITAL SIGNAL PROCESSING ในการประมวลสัญญาณ และเป็นวงจรรองความถี่ชนิด FIR

Abstract

This project can be propose, design and create FIR Digital Filter Programmable. It can be programable frequency by computer with TMS320C31 DIGITAL SIGNAL PROCESSING.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 การประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)	4
2.2 ข้อได้เปรียบและข้อด้อยของการประมวลผลสัญญาณเชิงเลข	5
2.3 ข้อดีของการประมวลผลสัญญาณเชิงเลข	5
2.4 ข้อด้อยของการประมวลผลสัญญาณเชิงเลข	6
2.5 บทนิยามของสัญญาณและระบบ	7
2.6 INTRODUCTION TO DIGITAL FILTER	8
2.7 Z-TRANSFORM	9
2.8 REALIZATION STRUCTURES FOR DIGITAL FILTER	10
2.9 TYPE OF DIGITAL FILTER	11
2.10 SUMMARY OF KEY CHARACTERISTIC FEATURES OF FIR FILTER	12
บทที่ 3 สถาปัตยกรรมของ TMS320C31	15
3.1 หน่วยประมวลผลกลาง (CPU)	15
3.2 หน่วยความจำ (Memory Organization)	18
3.3 Internal Bus Operation	21
3.4 External Bus Operation	21
3.5 การเชื่อมต่อระหว่าง TMS320C31 กับ TLC32040 AIC	24
บทที่ 4 ทฤษฎีและการออกแบบตัวกรองเชิงเลข	27
4.1 ตัวกรองผลตอบสนองอิมพัลส์จำกัด (Finite Impulse Response [FIR] Filter)	27
4.2 การออกแบบ	36
บทที่ 5 การทดลองและผลการทดลอง	39
5.1 วิธีการทดลอง	39
5.2 การทดลอง	39
บทที่ 6 สรุปแล่วิจารณ์ผล	61
สรุปผลการทดลอง	62
วิจารณ์ผลการทดลอง	64
เอกสารอ้างอิง	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

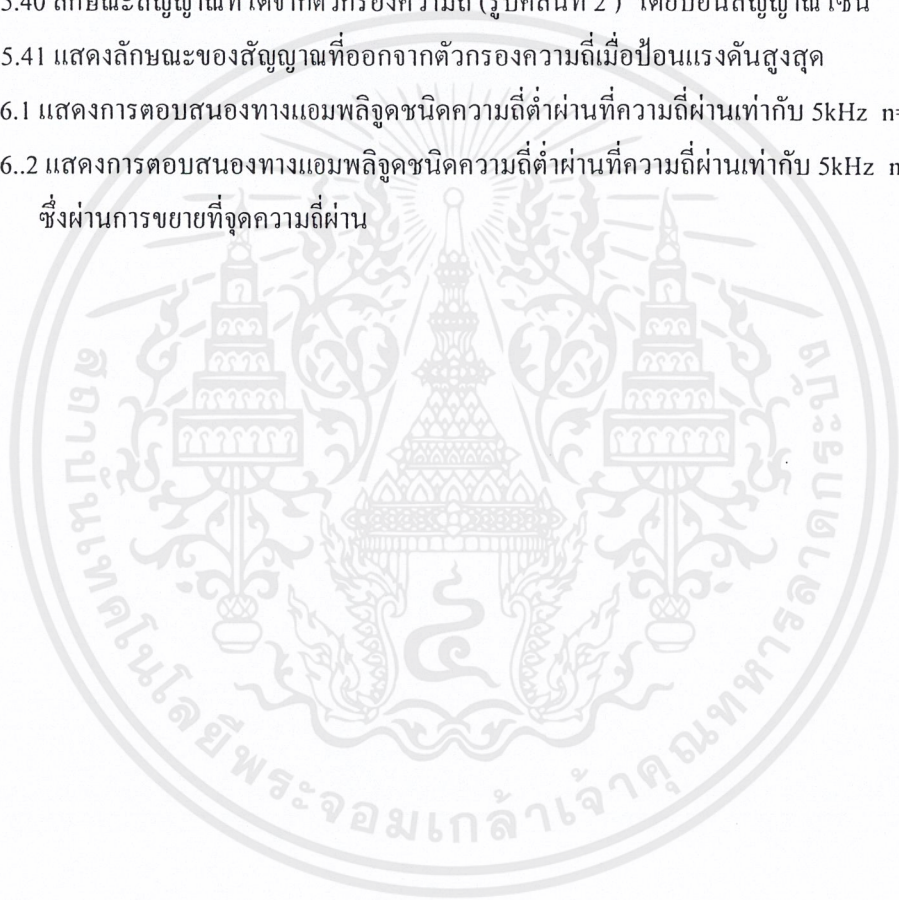
รูป		หน้า
รูปที่ 1.1	แสดงระบบที่ใช้ไมโครโปรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่	1
รูปที่ 1.2	แสดงส่วนประกอบของDSP	2
รูปที่ 2.1	Block Diagram ของ Transversal Filter โดยที่มี $h(k), (k=0,1,2,\dots,n-1)$	7
รูปที่ 2.2	การวิเคราะห์สมการ Difference Equation ในรูปแบบ Block Diagram	8
รูปที่ 2.3	การวิเคราะห์สมการ Difference Equation ในรูปแบบ Signal Flow Graph Diagram	11
รูปที่ 2.4	Block Diagram แสดงถึงหลักการของ Digital Filter	11
รูปที่ 2.5	โครงสร้างของ FIR Digital Filter	13
รูปที่ 2.6	โครงสร้าง IIR Digital Filter	14
รูปที่ 3.1	แสดงสถาปัตยกรรมของ TMS320C31	15
รูปที่ 3.2	แสดงหน่วยความจำของ TMS320C31	19
รูปที่ 3.3	แสดงการจัดหน่วยความจำของ TMS320C31	20
รูปที่ 3.4	memory-mapped external interface control register	22
รูปที่ 3.5	primary bus control register	22
รูปที่ 3.6	expansion bus control register	23
รูปที่ 3.7	แสดงการเชื่อมต่อระหว่าง TMS320C31 กับ TLC32040	25
รูปที่ 4.1	แสดงลักษณะสัญญาณของผลตอบสนองอิมพัลส์จำกัด	27
รูปที่ 4.2	ผลตอบสนองอุมคติ	29
รูปที่ 4.3	ผลตอบสนองแอมพลิจูดของวินโดว์สี่เหลี่ยม	31
รูปที่ 4.4	แสดงวิธีปฏิบัติการทำวินโดว์ในโดเมนความถี่	33
รูปที่ 4.5	สเปกตรัมของวินโดว์แบบสี่เหลี่ยม,แบบฮานและแบบแฮมมิง กรณี $N=11$	34
รูปที่ 4.6	แสดงบล็อกไดอะแกรมของโครงงาน	36
รูปที่ 4.7	แสดงการทำงานของโปรแกรม	37
รูปที่ 4.8	Flow chat แสดงการทำงานของ FIR Filter	38
รูปที่ 5.1	บล็อกไดอะแกรมแสดงวิธีการทดลอง	39
รูปที่ 5.2	ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน	40
	ที่ 0.3 ชนิด Blackman window ที่ได้จากการออกแบบ	
รูปที่ 5.3	ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน	40
	ที่ 0.3 ชนิด Blackman window ที่ได้จากการทดลอง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป	หน้า
รูปที่ 5.20 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n = 21 ที่ได้จากการออกแบบ	50
รูปที่ 5.21 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n = 21ที่ได้จากการทดลอง	50
รูปที่ 5.22 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45ที่ได้จากการออกแบบ	51
รูปที่ 5.23 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45ที่ได้จากการทดลอง	51
รูปที่ 5.24 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45 ที่ได้จากโปรแกรม MAT LAB	52
รูปที่ 5.25 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45 ที่ได้จากการทดลอง	52
รูปที่ 5.26 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45ที่ได้จากโปรแกรม MAT LAB	53
รูปที่ 5.27 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n = 45ที่ได้จากการทดลอง	53
รูปที่ 5.28 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n = 45ที่ได้จากโปรแกรม MAT LAB	54
รูปที่ 5.29 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n = 45ที่ได้จากการทดลอง	54
รูปที่ 5.30 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n=21 และ n = 45ที่ได้จากการทดลอง	55
รูปที่ 5.31 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n=21 และ n = 45ที่ได้จากการทดลอง	55
รูปที่ 5.32 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n=21 และ n = 45 ที่ได้จากการทดลอง	56
รูปที่ 5.33 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , n=21 และ n = 45ที่ได้จากการทดลอง	56
รูปที่ 5.34 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่ต่ำผ่าน ที่ความถี่ผ่านเท่ากับ 5 kHz จากการออกแบบ	57
รูปที่ 5.35 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่กลางผ่าน ที่ความถี่ผ่านเท่ากับ 2-3 kHz จากการออกแบบ	57

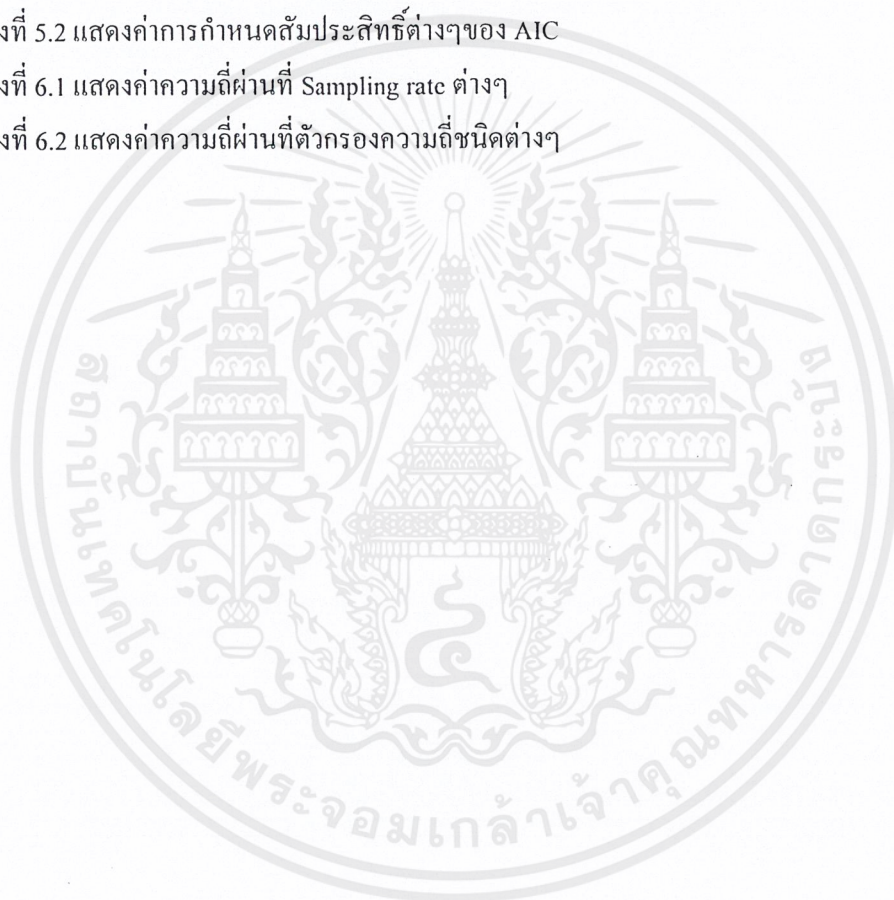
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป	หน้า
รูปที่ 5.36 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่ต่ำผ่าน ที่ความถี่ผ่านเท่ากับ 5 kHz จากการทดลอง	58
รูปที่ 5.37 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่กลางผ่าน ที่ความถี่ผ่านเท่ากับ 2-3 kHz จากการออกแบบ	58
รูปที่ 5.38 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณสี่เหลี่ยม	59
รูปที่ 5.39 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณสามเหลี่ยม	59
รูปที่ 5.40 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณไซน์	60
รูปที่ 5.41 แสดงลักษณะของสัญญาณที่ออกจากตัวกรองความถี่เมื่อป้อนแรงดันสูงสุด	60
รูปที่ 6.1 แสดงการตอบสนองทางแอมพลิจูดชนิดความถี่ต่ำผ่านที่ความถี่ผ่านเท่ากับ 5kHz n=21	63
รูปที่ 6.2 แสดงการตอบสนองทางแอมพลิจูดชนิดความถี่ต่ำผ่านที่ความถี่ผ่านเท่ากับ 5kHz n=21 ซึ่งผ่านการขยายที่จุดความถี่ผ่าน	63



สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 3.1 ชื่อและหน้าที่ต่างๆของรีจิสเตอร์ใน CPU	18
ตารางที่ 3.2 แสดงหน้าที่ต่างๆของ primary bus control register	23
ตารางที่ 4.1 ตารางแสดงคุณสมบัติของวินโดว์แบบต่างๆ	34
ตารางที่ 4.2 ตารางแสดงคุณสมบัติของวินโดว์แบบต่างๆเมื่อมีอันดับต่างกัน	35
ตารางที่ 5.1 แสดงฟังก์ชันต่างๆ ที่ใช้ในโปรแกรม MATLAB	39
ตารางที่ 5.2 แสดงค่าการกำหนดสัมประสิทธิ์ต่างๆของ AIC	45
ตารางที่ 6.1 แสดงค่าความถี่ผ่านที่ Sampling rate ต่างๆ	61
ตารางที่ 6.2 แสดงค่าความถี่ผ่านที่ตัวกรองความถี่ชนิดต่างๆ	61



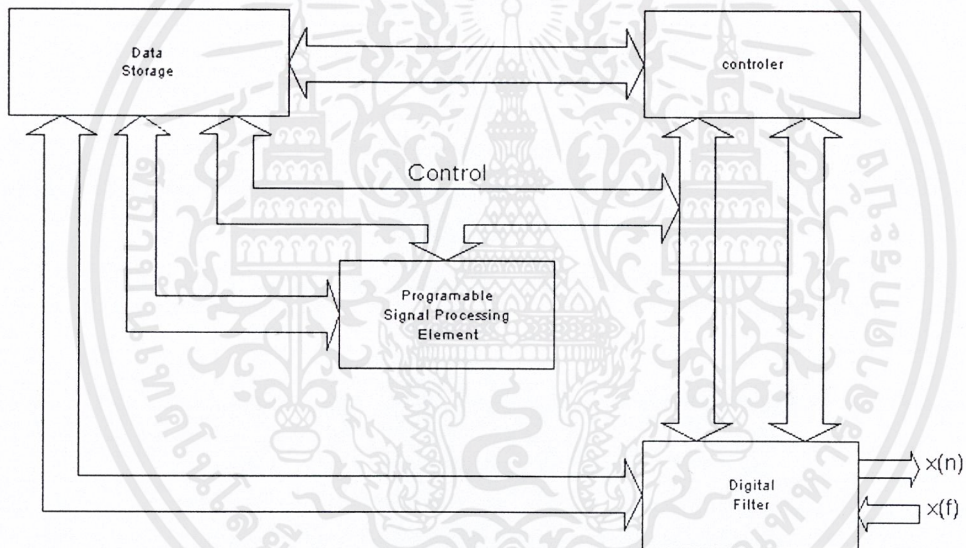
บทที่ 1

บทนำ

ดิจิตอลซิกแนลโพรเซสซิง (DSP) คือการเปลี่ยนรูปแบบของสัญญาณโดยใช้การคำนวณทางคณิตศาสตร์ด้วยดิจิตอลซิกแนลโพรเซสเซอร์ ซึ่งจะเห็นได้ว่าเราสามารถจะนำสัญญาณนั้นไปลงในสมการคณิตศาสตร์ แล้วได้ผลลัพธ์ออกมาโดยตรงเลยทีเดียว

ในช่วง 10 กว่าปีที่ผ่านมา ความก้าวหน้าทางด้านดิจิตอลซิกแนลโพรเซสซิงได้เกิดขึ้นเป็นอย่างมากมีการค้นพบอัลกอริทึม (Algorithms) ใหม่ ๆ เพิ่มขึ้นเรื่อยๆ และการพัฒนาเครื่องมือใหม่ๆ เพื่อช่วยในการออกแบบตามอัลกอริทึมนั้นอีกด้วย ระบบดิจิตอลซิกแนลโพรเซสซิงนั้นจะมีอยู่ 2 แบบใหญ่ๆ คือ

1. ระบบที่ออกมาเพื่อให้ทำงานอย่างใดอย่างหนึ่งโดยเฉพาะ
2. ระบบที่ใช้ไมโครโพรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่



รูปที่ 1.1 แสดงระบบที่ใช้ไมโครโพรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่

ความก้าวหน้าทางอิเล็กทรอนิกส์ครั้งใหญ่ เกิดขึ้นเมื่อมีการสร้างดิจิตอลซิกแนลโพรเซสเซอร์แบบความเร็วสูงขึ้น (High-Speed Signal Processor) ซึ่งปัจจุบันนี้ได้มีใช้กันอย่างแพร่หลายในราคาไม่แพงนัก

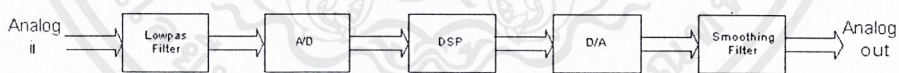
ดิจิตอลซิกแนลโพรเซสเซอร์จำเป็นจะต้องมีความเร็วสูงและถูกออกแบบมาโดยเฉพาะให้สามารถทำงานได้อย่างมีประสิทธิภาพ โดยใช้ข้อดีของการประมวลผลแบบขนาน และการใช้ชุดคำสั่งที่สร้างขึ้นมาโดยเฉพาะ ทั้งนี้ก็เพื่อการประมวลผลแบบเวลาจริง (Real time) นั่นเองการประมวลผลแบบเวลาจริงคือเมื่อเราใส่สัญญาณ อินพุตเข้าไปก็จะได้สัญญาณเอาต์พุตออกมาเลยโดยจะมีเวลาหน่วง (Delay) น้อยมาก ซึ่งจะมีประโยชน์ในการใช้งานมากกว่าการประมวลผลแบบไม่ใช้เวลาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยความสามารถของดิจิทัลซิกแนลโพรเซสซิง จึงได้นำไปใช้ในรูปแบบต่างๆ ซึ่งให้ผลลัพธ์ที่ดีกว่าใช้วงจรทางอนาล็อก เช่น ความน่าเชื่อถือ ความยืดหยุ่น การทดแทนได้ง่าย ขนาดที่เล็กกว่า และเสถียรภาพในระยะยาวและความสามารถของดิจิทัลซิกแนลโพรเซสเซอร์ที่สามารถเปลี่ยนโปรแกรมการทำงานได้ง่ายทำให้การใช้งานดิจิทัลซิกแนลโพรเซสเซอร์นั้นน่าสนใจ เช่น สามารถเปลี่ยนให้ดีขึ้นเมื่ออัลกอริทึมใหม่ ๆ ถูกค้นพบ หรือความสามารถทำงานได้หลายๆอย่างโดยเพียงแค่เปลี่ยนโปรแกรมให้กับมันเท่านั้น

ในทางอิเล็กทรอนิกส์ โดยส่วนใหญ่มักเกี่ยวข้องกับสัญญาณทางไฟฟ้า เช่นการกรองสิ่งที่ไม่ต้องการออกจากสัญญาณอินพุท การแยกเอาข้อมูลออกมาหรือไม่ ก็เป็นการสร้างรูปคลื่นที่ต้องการขึ้นมา หรือการเปลี่ยนลักษณะทางแอมพลิจูดของสัญญาณ ซึ่งในอดีตการกระทำเหล่านี้จะทำโดยใช้วงจรอนาล็อกที่ออกแบบมาเพื่อทำงานนั้นๆ นอกจากนี้วงจรอนาล็อกยังไม่เหมาะที่จะสร้างเป็นวงจรมิติใหญ่ (VLSI) อีกด้วย คุณสมบัติของวงจรมิติใหญ่นั้นจะขึ้นอยู่กับค่าของอุปกรณ์ที่ใช้ เช่น R,C ที่มีค่าไม่แน่นอน ที่สำคัญจะมีกระแสรั่วไหลที่แปรตามเวลา และยังขึ้นอยู่กับโวลต์เตจและอุณหภูมิอีกด้วย การออกแบบวงจรทางอนาล็อกนั้นก็ยังไม่ดีเพราะวงจรหนึ่งๆจะทำงานได้อย่างเดียวเท่านั้นแต่ถ้าต้องการให้ทำงานอื่นๆ ก็ต้องเปลี่ยนอุปกรณ์และก็ต้องเดินสายกันใหม่ นอกจากนี้แล้วการทดสอบวงจรมิติใหญ่นั้นก็เป็นปัญหาได้เนื่องจาก อิมพีแดนซ์ของเครื่องมือที่วัดต่อเข้าไปในวงจร ปัญหาพื้นฐานของวงจรมิติใหญ่นี้ สามารถแก้ไขได้โดยใช้ DSP ข้อได้เปรียบอีกประการหนึ่งของการใช้ DSP แทนวงจรมิติใหญ่คือการเปลี่ยนคุณสมบัติของการประมวลผลทำได้โดยง่าย ซึ่งจะทำให้เราสามารถทดลองในหลายๆทางที่เป็นไปได้เพื่อเลือกเอาผลที่ดีที่สุดที่เราพอใจ

โดยทั่วไประบบ DSP จะประกอบด้วยส่วนต่างๆดังรูป



รูปที่ 1.2 แสดงส่วนประกอบของDSP

สัญญาณอนาล็อกอินพุท จะถูกป้อนเข้าสู่ Low pass Filter เพื่อที่จะจำกัดช่วงความถี่ของสัญญาณอินพุท จากนั้นก็จะส่งต่อไปยัง อนาล็อก/ดิจิทัล คอนเวอร์เตอร์ (Analog to Digital Converter: A/D) แล้วส่งให้ระบบประมวลสัญญาณดิจิทัลต่อไป ระบบประมวลสัญญาณดิจิทัลก็จะนำเอาอัลกอริทึมที่มีอยู่มากระทำกับข้อมูล จากนั้นก็จะส่งสัญญาณข้อมูลที่ถูกแก้ไขแล้ว ไปให้ส่วน ดิจิทัล/อนาล็อก คอนเวอร์เตอร์ (Digital To Analog Converter : D/A) เพื่อเปลี่ยนสัญญาณข้อมูลกลับเป็นสัญญาณอนาล็อกตามต้องการ แต่จากเอาต์พุทของ D/A ซึ่งจะมีลักษณะเป็นขั้นๆแยกกัน ดังนั้นเราจำเป็นต้องใช้วงจรกรองสัญญาณให้เรียบ (Low Pass Filter) เพื่อที่จะขจัดสัญญาณความถี่สูงที่ไม่ต้องการออกไป

หากเราจะมองข้อเสียของ DSP ก็มี แต่ข้อเสียหรือปัญหาที่เกิดขึ้น จะแตกต่างไปจากที่พบในวงจรรอนาลอก เช่น การเกิดควอนไทซ์เซชัน (Quantization) , การเกิดน็อยส์ (Noise) , Idle channel noise , ความไม่ราบเรียบของสัญญาณ , ความละเอียดของสัญญาณ เป็นต้น ตัวแปรที่สำคัญอีกตัวแปรหนึ่งที่จะแยกระบบ DSP ต่างๆออกตามการใช้งานคือ ค่าอัตราการสุ่มข้อมูล (Sampling Rate) ยิ่งอัตราการสุ่มข้อมูลมากยิ่งต้องใช้ระบบประมวลผลที่มีความเร็วสูงยิ่งขึ้นด้วย อัตราการสุ่มข้อมูลนี้จะมีผลต่อการนำไปใช้งาน ถ้ามีอัตราการสุ่มข้อมูลที่สูง ก็จะใช้งานในย่านความถี่สูงขึ้นไปได้ โดยคุณภาพของสัญญาณข้อมูลยังคงยอมรับได้

บริษัทเทกซ์อินสตุเมนต์ ได้ผลิต ไอซีที่เป็นตัวประมวลผลออกมามีสำหรับใช้งานทางด้านของดิจิตอลซิกแนลโปรเซสซิงโดยเฉพาะ ซึ่งเป็นตระกูลที่มีชื่อว่า TMS320 ไอซีตระกูล TMS320 นี้จะเป็นลักษณะไมโครโปรเซสเซอร์แบบชิปเดี่ยว (Single Chip) สามารถทำงานได้กว้างขวาง และมีความเร็วในการประมวลผลสูง

เราจึงนำเอา TMS320C31 มาใช้ในงานวิเคราะห์สัญญาณแบบตัวกรองความถี่โดยเราจะออกแบบให้สามารถเปลี่ยนค่าความถี่ผ่านและอื่นๆโดยคอมพิวเตอร์ ทำให้สามารถเลือกใช้ และเปลี่ยนแปลงค่าความถี่ผ่านได้โดยง่าย ทางคอมพิวเตอร์ทั้งยังสามารถปรับปรุงและเปลี่ยนแปลงหรือพัฒนาโปรแกรมเพิ่มเติมได้อย่างง่ายดาย

บทที่ 2 ทฤษฎีและหลักการ

2.1 การประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)

สามารถแบ่งได้เป็น 2 หัวข้อใหญ่ๆคือ

2.1.1 การกรองเชิงเลข (Digital Filtering) การกรองเชิงเลข มีอยู่หลายประเภทด้วยกัน อย่างไรก็ตาม สำหรับในหัวข้องานวิจัยนี้ เราสนใจเฉพาะการกรองเชิงเลข แบบเป็นระบบเชิงเส้น ซึ่งมีตัวกรองอยู่ 2 ชนิดคือ ตัวกรองไม่ป้อนกลับเชิงเลข (non-recursive digital filter) และ ตัวกรองป้อนกลับเชิงเลข (recursive digital filter) ตัวกรองทั้งสอง ชนิดเป็นวงจรรพื้นฐานของวงจรกรองเชิงเลขแบบอื่นๆ รายละเอียดเกี่ยวกับคุณสมบัติ หลักการออกแบบหลักการสร้างจะได้กล่าวถึงในบทต่อไป

2.1.2 การแปลงเชิงเลข (Digital transform) การแปลงเชิงเลข โดยเฉพาะการแปลงฟูริเยอร์เต็มหน่วย (Discrete Fourier Transform หรือ DFT) ที่ทำการประมวลผลโดยใช้ขั้นตอนวิธีที่เรียกว่า การแปลงฟาสต์ฟูริเยอร์ (Fast Fourier Transform หรือ FFT) มีส่วนทำให้การประมวลผลสัญญาณเชิงเลขได้รับความนิยมมากขึ้น

ตัวอย่างการนำไปประยุกต์ใช้งานของการประมวลผลสัญญาณเชิงเลข มีให้เห็นอยู่มากในปัจจุบัน อาทิเช่น เทคนิคการสื่อสารเชิงเลข ซึ่งอาจเป็น การมอดูเลต (modulation) หรือการลงรหัส (coding) สัญญาณเชิงเลข นอกจากนี้ยังมีใช้ในระบบควบคุมเชิงเลข (digital control system) และอุปกรณ์เครื่องมือวัดเชิงเลข เราอาจหมายถึง

- ก. การประมวลผลเชิงเลขของสัญญาณใดๆที่ไม่ได้อยู่ในรูปแบบสัญญาณเชิงเลขหรือ
- ข. การประมวลผล (ด้วยวิธีใดก็ตาม) ของสัญญาณที่อยู่ในรูปแบบสัญญาณเชิงเลขแล้ว

ความหมายในข้อ (ก) นั้นหมายถึง การประมวลผลสัญญาณในธรรมชาติทั่วไป ซึ่งสัญญาณในธรรมชาตินั้นปกติอยู่ในรูปแบบ สัญญาณเชิงอุปมาน (analog signal) ก่อนการประมวลผลเราต้องทำการแปลงสัญญาณเชิงอุปมานให้เป็นสัญญาณเชิงเลขเสียก่อน ส่วนความหมายในข้อ (ข) เป็นการประมวลผลสำหรับสัญญาณที่มาจากอุปกรณ์เชิงเลข เช่น คอมพิวเตอร์ หรือ ระบบควบคุมเชิงเลข สัญญาณจากอุปกรณ์เหล่านี้เป็นแบบสัญญาณเชิงเลขอยู่แล้ว จึงไม่จำเป็นต้องมีการแปลงรูปสัญญาณอีก

เหตุผลสำคัญที่ทำให้การประมวลผลสัญญาณเชิงเลข เริ่มได้รับความสนใจมากขึ้นก็เนื่องมาจากเทคโนโลยีในการสร้าง วงจรรวมหรือ วงจร ไอซี (integrated circuit) สามารถสร้างเป็นวงจรรวมที่ผลิตบนสารกึ่งตัวนำขึ้นเดียวกันได้ และสามารถผลิตครั้งละจำนวนมากขึ้นได้ จึงทำให้ราคาต่อวงจรถูกลงเป็นอย่างมาก โดยเฉพาะอย่างยิ่งวงจรรวมแบบแอลเอสไอ (LSI = Large Scale Integrated Circuit) ที่สำคัญๆ เช่น ไมโครโปรเซสเซอร์ (microprocessor) ซึ่งมีราคาถูกลงมากและมีผู้สนใจนำมาทำเป็นตัวประมวลผลสัญญาณมากขึ้น นอกจากนี้การประมวลผลสัญญาณเชิงเลขยังให้ ความเชื่อถือได้และความแม่นยำในการคำนวณดีกว่าการประมวลผลสัญญาณเชิงอุปมานมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการประมวลผลสัญญาณเชิงอุปมานได้มีการศึกษา วิจัย และมีการพัฒนากันมานาน ทำให้อุปกรณ์ และระบบวงจรที่ใช้กันอยู่ในปัจจุบันในรูปแบบของ การประมวลผลสัญญาณเชิงอุปมานเกือบทั้งหมด การที่จะออกแบบสร้างระบบประมวลผลสัญญาณใหม่ ให้อยู่ในรูปแบบของการประมวลผลสัญญาณเชิงเลข อุปกรณ์ใหม่นั้นก็ต้องสามารถเชื่อมโยงกับอุปกรณ์เดิม ซึ่งเป็นระบบการประมวลผลสัญญาณเชิงอุปมานได้ ดังนั้น ความพยายามของนักวิจัยสมัยแรก ก็คือ การวิจัย และศึกษาเพื่อหาวิธีการนำเอาอุปกรณ์เชิงเลขและอุปกรณ์เชิงอนุมาณมาเชื่อมโยงกัน ดังสามารถเห็นได้จากสาขาที่แสดงถึงความสัมพันธ์กันนั้น ได้แก่ สาขาวิชา ทฤษฎีโครงข่าย (network theory) กับทฤษฎีกราฟ (graph theory) สาขาวิชา การทำนายข้อมูล (data Prediction) กับ การอัดแถบความถี่ปฏิบัติงาน (bandwidth compression) สาขาวิชา การแปลงฟูริเยอร์ (Fourier transform)) กับการแปลงฮาดามาร์ด (Hadamard transform) หรือสาขาวิชา การกรองเชิงเลข (digital filter) กับการกรองเชิงอุปมาน (analog filter) เป็นต้น ซึ่งในแต่ละสาขาวิชาที่กล่าวถึงนี้เป็นที่รู้จักกันเป็นอย่างดี และมีการประยุกต์ใช้มากในปัจจุบัน

2.2 ข้อได้เปรียบและข้อด้อยของการประมวลผลสัญญาณเชิงเลข

อาจกล่าวได้ว่า ข้อเด่นเป็นอย่างมากของการประมวลผลสัญญาณเชิงเลขก็คือ ความเที่ยงตรง ความแม่นยำ ในการประมวลผล อย่างไรก็ตามการนำเอาระบบประมวลผลสัญญาณเชิงเลขมาใช้ร่วมกับการประมวลผลสัญญาณเชิงอุปมานที่ยังมีการใช้กันมากในปัจจุบัน ก็ต้องมีข้อเสียเปรียบอยู่เหมือนกัน ดังนั้นการกล่าวถึงข้อได้เปรียบ และข้อด้อย บางประการของการประมวลผลสัญญาณเชิงเลขจึงอาจเป็นประโยชน์บ้าง และ นอกจากนี้ยังเป็นการให้ภาพกว้างๆ แก่ผู้ที่จะนำเอาระบบประมวลผลสัญญาณเชิงเลขไปประยุกต์ใช้ ได้อย่างเหมาะสมยิ่งขึ้น

2.3 ข้อดีของการประมวลผลสัญญาณเชิงเลข

ข้อได้เปรียบของการประมวลผลสัญญาณเชิงเลขอาจกล่าวได้เป็นข้อๆดังนี้

2.3.1 เหมาะสำหรับอุปกรณ์ที่ข้อมูลอยู่ในรูปแบบสัญญาณเชิงเลขอยู่แล้ว เช่น ผลลัพธ์จากคอมพิวเตอร์ ไมโครโปรเซสเซอร์ หรือ ข้อมูลจากระบบควบคุมเชิงเลข เป็นต้น ทั้งนี้เนื่องจากสัญญาณหรือข้อมูลจากอุปกรณ์เหล่านี้ ถ้าหากต้องการนำไปประมวลผลสัญญาณเชิงอุปมาน ก็จำเป็นต้องมี วงจรแปลงรูปแบบสัญญาณเชิงเลขเป็นเชิงอุปมาน (Digital to Analog converter หรือ D/A) เพื่อแปลงสัญญาณเชิงเลขให้เป็นสัญญาณเชิงอุปมานก่อน หลังจากประมวลผลด้วยระบบประมวลผลเชิงอุปมานแล้วก็ต้องแปลงผลลัพธ์ที่เป็นสัญญาณเชิงอุปมานเป็นเชิงเลข (Analog to Digital converter หรือ A/D) เพื่อส่งกลับไปยังอุปกรณ์เชิงเลขต่อไป ซึ่งจะเห็นได้ว่าระบบประมวลผลมีความซับซ้อนมากขึ้น นอกจากนี้ถ้าหากเป็นระบบที่ต้องการทำการประมวลผลสัญญาณพร้อมกันหลายสัญญาณ (ซึ่งในคอมพิวเตอร์มักทำในลักษณะนี้) ก็จะทำให้ระบบยุ่งยากมากขึ้นไปอีกทั้งในปัจจุบัน A/D หรือ D/A ที่มีประสิทธิภาพสูงและมีความเร็วในการแปลงสูงยังมีราคาแพงอยู่

2.3.2 อุปกรณ์ทางด้านเชิงเลขหรือดิจิทัลมีราคาถูก ขนาดเล็กมีประสิทธิภาพสูง ความแม่นยำ และความแน่นอนสูง นอกจากนี้การพัฒนาให้มีประสิทธิภาพสูงขึ้นก็เป็นไปอย่างรวดเร็ว ข้อนี้เป็นข้อได้เปรียบที่สำคัญ เพราะว่าการประมวลผลสัญญาณเชิงเลขสามารถคำนวณได้อย่างแม่นยำและต้องการให้มีความละเอียดเท่าใดก็ได้ แต่การประมวลผลเชิงอุปมาการคำนวณที่ให้ความละเอียดเกินกว่า หนึ่งในพันส่วน หรือ 0.1 นั้น ทำได้ยากมาก นอกจากนี้ชิ้นอุปกรณ์ที่ใช้ในระบบของการประมวลผลเชิงอุปมา เช่น ตัวความต้านทาน ตัวขยายสัญญาณ ยังมีคุณสมบัติแปรค่าได้ตามสภาพแวดล้อม เช่น อุณหภูมิ ความชื้น อายุการใช้งาน ทำให้ความแม่นยำและความน่าเชื่อถือ ของระบบการประมวลผลเชิงอุปมาต่ำ อย่างไรก็ตามในปัจจุบัน

การประมวลผลเชิงเลขที่ให้ความแม่นยำสูงต้องใช้อุปกรณ์ราคาแพง ถ้ามีการพัฒนาให้มีราคาถูกลงก็จะเป็นผลดียิ่ง

2.3.3 การรับ และส่งข้อมูลหรือสัญญาณเชิงเลข ทำได้แน่นอนกว่าทั้งนี้เนื่องจากสัญญาณเชิงเลขมีแค่สองระดับคือ 0 กับ 1 เท่านั้นถ้าหากอุปกรณ์สัญญาณผิดเพี้ยนไปก็สามารถแก้ไขและสร้างขึ้นมาให้เหมือนเดิมได้โดยง่าย

2.3.4 การประมวลผลสัญญาณเชิงเลข ทำได้ง่ายเนื่องจาก ขั้นตอนวิธี (algorithm) การประมวลผลสัญญาณมักจะประกอบด้วย การบวก การลบ การคูณ การหาร และการเลื่อนตัวเลขเท่านั้น ซึ่งข้อความนี้สามารถแสดงให้เห็นจริงได้ในบทต่อไป

2.3.5 ระบบประมวลผลสัญญาณเชิงเลข สามารถทำเป็นแบบ ระบบแบ่งกันใช้เวลา (time-shared system) ได้ดังนั้นจึงสามารถทำการประมวลผลพร้อมๆกันหลายช่องสัญญาณได้ และนอกจากนี้ในระบบเดียวกันยังสามารถโปรแกรมให้ทำงานได้หลายรูปแบบด้วย

2.3.6 ระบบประมวลผลสัญญาณเชิงเลข มีความคล่องตัวสูง ทั้งนี้เนื่องจากสามารถทำการมัลติเพล็กซ์ กับข้อมูลหรือสัญญาณเสียง หรือสัญญาณภาพได้ การมัลติเพล็กซ์ ยังสามารถทำเป็นแบบการมัลติเพล็กซ์แบบแบ่งเวลา (Time Division Multiplex หรือ TDM) การมัลติเพล็กซ์แบบแบ่งรหัส (Code Division Multiplex หรือ CDM) ได้ นอกจากนี้การส่งสัญญาณยังส่งได้ในอัตราบิต (bit rate) หรืออัตราบอด (baud rate) ต่างๆกันได้

2.4 ข้อดีของการประมวลผลสัญญาณเชิงเลข

นอกจากข้อดีที่กล่าวแล้ว การประมวลผลสัญญาณเชิงเลขทำให้เกิดข้อดีหลายประการคือ

2.4.1 ระบบประมวลผลสัญญาณเชิงเลขต้องมีสัญญาณสำหรับ การซิงโครไนซ์ (synchronize) การจับเวลา (timing) และการกำหนดกรอบ (framing) ของข้อมูลไว้ด้วย ถ้าหากสัญญาณเหล่านี้สูญหายหรือ ผิดพลาดไป การทำงานของระบบก็จะผิดพลาดไปด้วย

2.4.2 มีปัญหาการเชื่อมโยง (interfacing) กับระบบการประมวลผลสัญญาณเชิงอุปมา ทั้งนี้เนื่องมาจากระบบประมวลผลสัญญาณแบบเดิม ส่วนมากอยู่ในระบบเชิงอุปมา ดังนั้นระบบที่สร้างขึ้นใหม่

ก็ต้องสามารถเชื่อมโยงกับระบบเดิมให้ได้ ซึ่งอาจทำให้ระบบประมวลผลสัญญาณซับซ้อนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 สัญญาณหรือข้อมูลเชิงเลข ไม่ใช่สัญญาณตามธรรมชาติที่แท้จริงแต่ถูกสร้างขึ้นมา ดังนั้น การส่งสัญญาณชนิดนี้ไปในตัวกลางตามธรรมชาติทั่วไป ซึ่งมักมีแถบความถี่ปฏิบัติงาน (band width) จำกัดและยังมีผลตอบ สนองเฟสไม่เป็นเชิงเส้น (non-linear phase response) จึงอาจทำให้สัญญาณเกิดความผิดเพี้ยนได้ เช่น เมื่อเราส่งสัญญาณรูปเหลี่ยมไปบนสายส่ง สัญญาณที่ได้รับไม่เป็นรูปเหลี่ยมเหมือนเดิม

2.4.4 เนื่องจากเราต้องการออกแบบ ระบบประมวลผลสัญญาณเชิงเลข ให้มีความคล่องตัวสูง สามารถใช้งานได้หลายรูปแบบ ดังนั้นอาจทำให้การออกแบบระบบประมวลผลสัญญาณเชิงเลขมีความซับซ้อนมากซึ่งรวมทั้งด้านการซ่อมแซม บำรุงรักษา และจัดทำคู่มือใช้งานด้วย

2.4.5 แถบความถี่ปฏิบัติงานของระบบประมวลผลสัญญาณเชิงเลข ต่ำกว่าแถบความถี่ปฏิบัติงานของระบบประมวลผลเชิงอุปมานมาก ข้อจำกัดนี้เนื่องมาจากอุปกรณ์ที่ใช้หรือประกอบขึ้นเป็นระบบประมวลผลสัญญาณเชิงเลข เช่น วงจรเกต วงจรซีพรีจิสเตอร์ วงจรสุ่มและคงค่าสัญญาณ (Sampling and Hold circuit หรือ S/H) หรือ วงจร A/D และ D/A เป็นต้น วงจรเหล่านี้ต่างมีความเร็วสูงสุดในการทำงาน จำกัดอยู่ค่าหนึ่ง ซึ่งในปัจจุบันยังมีค่าต่ำมาก จึงเป็นผลทำให้ระบบประมวลผลสัญญาณเชิงเลขมีความเร็วต่ำ ตัวอย่างเช่น ถ้าไมโคร โพรเซสเซอร์ที่มีสัญญาณนาฬิกาขนาด 1 เมกกะเฮิร์ตซ์ เมื่อนำไปสร้างเป็นระบบประมวลผลสัญญาณเชิงเลขจะใช้ได้กับสัญญาณที่มีความถี่สูงสุดประมาณ 10 ถึง 20 กิโลเฮิร์ตซ์เท่านั้น

อย่างไรก็ตาม เมื่อพิจารณาข้อดีและข้อด้อยต่างๆ เหล่านี้แล้ว ถ้าเราต้องการระบบประมวลผลสัญญาณที่มีประสิทธิภาพดี ความแม่นยำสูง โดยไม่คำนึงว่าราคาสูงเท่าใดแล้ว ระบบประมวลผลสัญญาณเชิงเลขก็ให้ประโยชน์ได้มาก

บทนิยามของศัพท์เทคนิคในระบบประมวลผลสัญญาณเชิงเลข

2.5 บทนิยามของสัญญาณและระบบ

ก่อนที่จะกล่าวถึงส่วนสำคัญต่างๆ ของระบบประมวลผลสัญญาณเชิงเลข เรามาทำความรู้จักกับคำนิยามของศัพท์เทคนิคบางคำเสียก่อน เพื่อให้การสื่อความหมายและความเข้าใจเป็นแบบอย่างเดียวกัน โดยที่นิยามของคำศัพท์เหล่านี้มาจากบทความเรื่อง Terminology in digital signal processing ตีพิมพ์ในวารสาร IEEE Transaction on Audio and Electroacoustic , vol . AU-20 , pp. 323-337, Dec 1972.

ซึ่งนำเฉพาะคำที่สำคัญมากกล่าวถึงคือ

2.5.2 สัญญาณเชิงอุปมาน (Analog signal) ใช้กับกรกล่าวถึงสัญญาณที่มีรูปคลื่น (waveform) แปรไปอย่างต่อเนื่องกับพิสัยเวลา โดยที่แอมพลิจูด (amplitude) หรือค่าขนาดของสัญญาณก็มีการแปรไปอย่างต่อเนื่องด้วย เช่น สัญญาณไซน์ หรือ $\sin(\omega t)$ เป็นต้น คำว่า อุปมาน มีต้นกำเนิดมาจากการพัฒนาของ คอมพิวเตอร์เชิงอุปมาน (analog computer) ที่เลียนแบบปริมาณทางฟิสิกส์ต่างๆ (ซึ่งเป็นปริมาณต่อเนื่องกับเวลา) ด้วย ค่าขนาดศักย์ หรือ กระแสไฟฟ้า และใช้อุปกรณ์เชิงอุปมานเช่น วงจรขยายออป-แอมป์ (op-amp) มาช่วยในการเลียนแบบ

2.5.2 สัญญาณเชิงเวลาต่อเนื่อง (Continuous – time signal) สัญญาณแบบนี้รูปคลื่นของสัญญาณแปรค่าไปอย่างต่อเนื่องกับพิสัยเวลา แต่แอมพลิจูดไม่ได้เจาะจงว่าต้องแปรไปอย่างต่อเนื่อง ซึ่งก็หมายถึงว่าอาจแปรอย่างไม่ต่อเนื่องก็ได้ เพราะฉะนั้นอาจกล่าวได้ว่า สัญญาณเชิงอุปมานเป็นชนิดหนึ่งของสัญญาณเชิงเวลาต่อเนื่องได้ อย่างไรก็ตามในหนังสือทั่วไปมักใช้คำทั้งสองนี้ปนกันไป ดังนั้นโดยทั่วไปการพูดถึง สัญญาณเชิงอุปมาน จึงหมายถึงสัญญาณเชิงเวลาต่อเนื่องด้วย

2.5.3 สัญญาณเชิงเวลาเต็มหน่วย (Discrete-time signal) เป็นสัญญาณ $x(t)$ ที่ค่าของฟังก์ชันกำหนดเฉพาะเขตของเวลาที่แน่นอนอันหนึ่งเท่านั้น สัญญาณแบบนี้อาจแบ่งตามลักษณะของแอมพลิจูดได้เป็น 2 แบบคือ

2.5.3.1 สัญญาณเชิงข้อมูลเต็มหน่วย (Discrete-time signal) สัญญาณแบบนี้แอมพลิจูดจะต่อเนื่อง หรือกล่าวได้ว่า แอมพลิจูดมีค่าเท่ากันทุกประการกับสัญญาณเชิงอุปมาน ที่เป็นตัวต้นแบบในการสุ่มตัวอย่าง (sampling) ตัวอย่างเช่น สัญญาณออกของ วงจรสุ่มและค่าคงสัญญาณจะจัดเป็นสัญญาณเชิงเต็มหน่วย สัญญาณแบบนี้แบบนี้อาจเรียกว่า ข้อมูล หรือ สัญญาณเชิงเต็มหน่วย

2.5.3.2 สัญญาณเชิงเลข (Digital signal) สัญญาณแบบนี้แอมพลิจูดของสัญญาณมีค่าของมีค่าเฉพาะเขตของค่าที่แน่นอนเขตหนึ่งเท่านั้น เช่นสัญญาณที่ออกจากวงจร A/D เป็นต้น

2.5.4 ระบบเวลาจริง (Real-time system) โดยทั่วไปคำนี้ใช้กับ ระบบการประมวลผลสัญญาณ ที่คำนวณในการประมวลผลทำได้เสร็จสิ้น ก่อนจะมีการสุ่มตัวอย่างสัญญาณ ลำดับใหม่อย่างไรก็ตามคาบเวลาในการสุ่มตัวอย่างสัญญาณควรมีค่าน้อย ไม่ควรเป็นค่า วัน เดือน หรือ ปี

2.5.5 ลำดับ (Sequence) หนังสือบางเล่มอาจเรียกเป็นลำดับข้อมูล หรือลำดับสัญญาณ หรือ ลำดับสัญญาณ หรือ ลำดับก็ได้ ซึ่งคำว่า ลำดับ อาจใช้แทน สัญญาณเชิงเต็มหน่วย หรือสัญญาณเชิงเลขก็ได้

2.6 INTRODUCTION TO DIGITAL FILTER

สัญญาณเต็มหน่วย (discrete-time signal) ที่ได้จากการวัดอาจจะมีสัญญาณอย่างอื่นที่ต้องการซึ่งเรียกว่า สัญญาณรบกวน (noise) ปะปนผสมเข้ามาด้วย จึงจำเป็นต้องนำสัญญาณนี้ผ่านเข้าไปในอุปกรณ์อย่างหนึ่งซึ่งเรียกว่า ตัวกรองสัญญาณ (filter) ซึ่งทำหน้าที่กรองเอาสัญญาณรบกวนออก ที่ด้านออกของตัวกรองเราจะได้สัญญาณที่ต้องการ ในการกรองสัญญาณนั้นก่อนอื่นจะต้องรู้จักคุณลักษณะเฉพาะตัวของสัญญาณที่ต้องการจะวัดและของสัญญาณรบกวนที่เกิดขึ้น เพื่อให้สามารถสร้างตัวกรองสัญญาณที่มีคุณสมบัติยอมให้เฉพาะสัญญาณที่ต้องการผ่านออกมาด้านนอกได้เท่านั้น กระบวนการกรองสัญญาณเต็มหน่วยนี้หากใช้กระบวนการทางดิจิทัล เช่น ใช้คอมพิวเตอร์จะเรียกว่า ตัวกรองสัญญาณดิจิทัล (digital filter)

Digital Filter เป็นสิ่งที่สำคัญมากที่สุดอย่างหนึ่งในการนำไปใช้ในระบบDSP (Digital signal processing) เป็นการนำ Algorithm ทางคณิตศาสตร์มาใช้ในระบบ Hardware หรือ Software กล่าวคือ สัญญาณ Digital Input จะทำการสร้างสัญญาณ Digital Output โดยใช้ Filtering Algorithm เพื่อให้ได้รับ สัญญาณจากการ Filtering ตามความต้องการเพื่อนำไปใช้ใน Digital Filter สำหรับชนิดของ ตัวกรอง สัญญาณ ที่สำคัญจะถูกกำหนดโดย

$$y(n) = \sum_{k=0}^n h(k).x(n - k) \tag{1}$$

โดยที่ $h(k)$ คือ สัมประสิทธิ์ ของ Filter

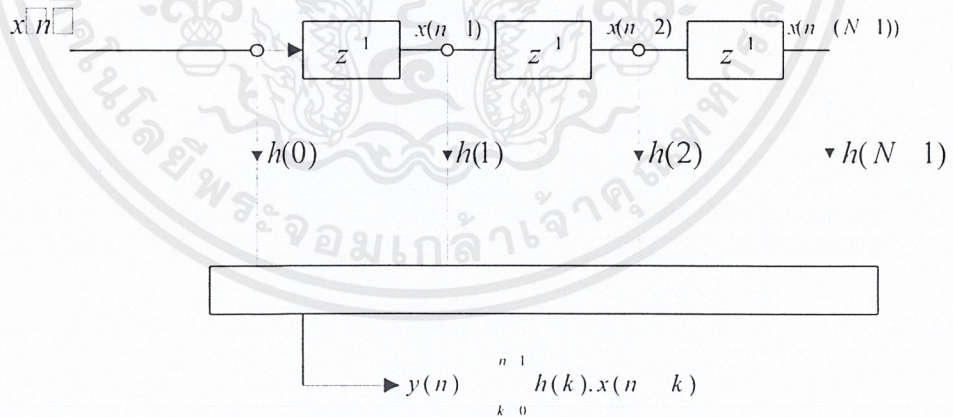
k คือ 0,1,2,...n-1

$x(n)$ คือ Input Filter

$y(n)$ คือ Output Filter

ซึ่งสำหรับตัวกรองสัญญาณ ที่ให้ค่าสัมประสิทธิ์จะกำหนด Characteristic ของ ตัวกรอง สัญญาณ Filtering นี้จะขึ้นอยู่กับ การ Convolution ที่แท้จริงของสัญญาณ และการตอบสนองต่อ Impulse ของตัวกรองสัญญาณใน Time Domain ซึ่งก็คือ $h(k)$

รูปที่จะกล่าวถึงแสดงถึง Block Diagram ซึ่งจะแสดงนิยามของ Digital Filter ที่กำหนดไว้ก่อนหน้า นี้ โดยที่รูปแบบ ตัวกรองสัญญาณ ชนิดนี้เป็นที่รู้จักอย่างแพร่หลาย เรียกว่า Transversal Filter จากรูป 2.1 z^{-1} จะแสดงถึง Delay ของช่วงเวลาในการสุ่มตัวอย่างหนึ่งครั้ง



รูปที่ 2.1 Block Diagram ของ Transversal Filter โดยมี $h(k), (k=0,1,2,\dots,n-1)$

วัตถุประสงค์ของตัวกรองสัญญาณ โดยรวมๆแล้วเป็นการกำจัดหรือลด Noise จากสัญญาณที่ต้องการดึงตัวอย่างคงรูปโดยการดึงข้อมูลของกลุ่มที่สนใจในช่วงหนึ่งออกมาซึ่ง Noise ในระบบอื่นๆอาจจะปะปนมาด้วยดังนั้นเราจึงทำการกรองสัญญาณที่แท้จริงออกมาเพื่อที่จะสามารถนำไปใช้ในการวิเคราะห์ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

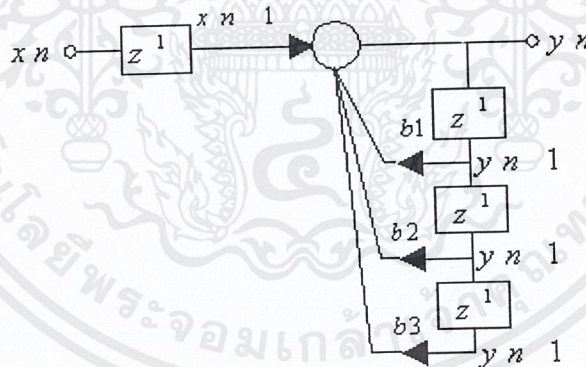
2.7 Z-TRANSFORM

Z-Transform เป็นการแปลงอีกชนิดหนึ่งซึ่งนักออกแบบตัวกรองสัญญาณทั้งหลายนิยมใช้อย่างถึงวิธีการนี้จะเป็นการแปลงลำดับซึ่งเป็นตัวเลขที่เขียนเรียงกันให้เป็นฟังก์ชันของตัวแปรเชิงซ้อน (Complex variable) หรือกล่าวให้ง่ายคือการเอาลำดับชุดหนึ่งไปสร้างฟังก์ชันตามกฎเกณฑ์ทางคณิตศาสตร์ที่กำหนดไว้ ดังนั้นคุณสมบัติต่างๆ ที่มีอยู่ในลำดับจะถูกส่งเข้าไปอยู่ในฟังก์ชันที่สร้างขึ้นด้วย และจะใช้อักษร Z ในภาษาอังกฤษเป็นสัญลักษณ์ของตัวแปรในฟังก์ชันที่สร้างขึ้น

Z-Transform เป็นสิ่งที่สำคัญที่สุดอย่างหนึ่งในระบบ DSP นั่นคือใช้ในการออกแบบเพื่อวิเคราะห์สัญญาณ Error ใน Digital Filter โดยเฉพาะในระบบ IIR Filter ซึ่งมักจะถูกใช้ในการคำนวณหาค่าสัมประสิทธิ์ของ Digital Filter และวิเคราะห์ผลของการ Quantization Errors คือใช้ในการแสดงโครงสร้างของ Digital Filter

2.8 REALIZATION STRUCTURES FOR DIGITAL FILTER

Discrete-Time Filter จะถูกนำมาแสดงบ่อยๆ ในรูปแบบของ Block Diagram หรือ Signal Flow Graph ซึ่ง Block Diagram จะเป็นการสะดวกกว่าที่ใช้ในการแสดง Difference Equations หรือ Equivalent ของ Transfer Function ซึ่งสามารถพิจารณาดังตัวอย่าง Discrete Filter ด้วย Difference Equations ดังรูปต่อไปนี้



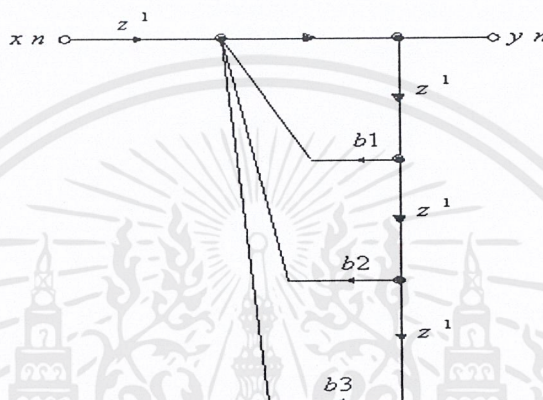
รูปที่ 2.2 การวิเคราะห์สมการ Difference Equation ในรูปแบบ Block Diagram

โดยการสมมุติสมการดังนี้

$$y(n) = x(n-1) - b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3)$$

Block Diagram แสดงถึงสมการดังกล่าวจากรูปสัญลักษณ์ Z จะแสดงถึง Delay ของช่วงเวลาหนึ่งหน่วย ซึ่งอาจพิจารณาใน Node ต่างๆ ลูกศรแสดงถึงการ Multiplies และค่าคงตัวที่ตัวถัดมา ทำการคูณกับตัวประกอบ ความสัมพันธ์ระหว่าง Difference Equations และ Block Diagram จะค่อยๆปรากฏขึ้น และ Signal Flow Graph Diagram ก็ใช้แสดงกับ Difference Equations เช่นเดียวกันดังรูป 2.3 โดยสมมติสมการดังนี้

$$y(n) = x(n-1) - b_1y(n-1) + b_2y(n-2) + b_3y(n-3)$$

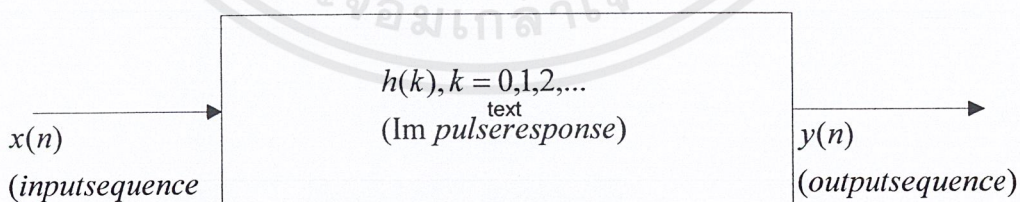


รูปที่ 2.3 การวิเคราะห์สมการ Difference Equation ในรูปแบบ Signal Flow Graph Diagram

2.9 TYPE OF DIGITAL FILTER

Digital Filter ที่ใช้อย่างกว้างขวางแบ่งออกเป็น 2 ชนิด

1. Finite Impulse Response (FIR)
2. Infinite Impulse Response (IIR)



รูปที่ 2.4 Block Diagram แสดงถึงหลักการของ Digital Filter

ตัวกรองสัญญาณแต่ละชนิด จะมีรูปแบบพื้นฐานแต่ละตัวซึ่งสามารถแสดงลำดับของการตอบสนองต่อ Impulse $h(k)$; ($k=0,1,2,\dots$) ดังรูปที่ 2.4 สัญญาณ Input และ Output ที่ต่อตัวกรองสัญญาณ จะถูกนำมาเกี่ยวพันโดยผลรวมของการ ซึ่งสมการแต่ละชนิดแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(n) = \sum_{k=0}^{n-1} h(k) \cdot x(n-k) \quad \text{:IIR digital filter (2)}$$

$$y(n) = \sum_{k=0}^{\infty} h(k) \cdot x(n-k) \quad \text{:FIR digital filter (3)}$$

จากสมการข้างต้น สำหรับ IIR Digital Filter นั้น ผลการตอบสนองต่อ Impulse จะอยู่ในช่วงที่ไม่จำกัด แต่สำหรับ FIR Digital Filter จะอยู่ในช่วงที่กำหนด โดยเริ่มที่ $h(k)$ สำหรับ FIR จะมีเพียง N ค่า ส่วนในการคำนวณหาค่า Output ของ IIR ไม่สามารถเป็นไปได้ เนื่องจากสมการมีความยาวของผลการตอบสนองต่อ Impulse นั้นยาวมาก (∞ : Infinite) ดังนั้นเราจึงแทนสมการของ IIR Digital Filter ในรูปของ Recursive Form

$$y(n) = \sum_{k=0}^{\infty} h(k) \cdot x(n-k) = \sum_{k=0}^{\infty} a_k \cdot x(n-k) - \sum_{k=1}^{\infty} a_k \cdot x(n-k) \quad (4)$$

โดยที่ a_k และ b_k เป็นสัมประสิทธิ์ของ ตัวกรองสัญญาณ คำนึงจากสมการข้างต้นคือสมการที่แตกต่างกันระหว่าง FIR และ IIR Digital Filter จากสมการเหล่านั้นจะมีค่า $h(k)$ สำหรับ FIR และ a_k และ b_k สำหรับ IIR ซึ่งสมการทั้งสองเป็นสิ่งที่สำคัญมากที่ใช้ในการออกแบบ ตัวกรองสัญญาณ เป็นส่วนใหญ่ จากสมการ กระแสการสุ่มตัวอย่าง Output $y(n)$ เป็น Function ของ Output ที่ผ่านมา ซึ่งค่าที่ถูกต้องจะเท่ากับตัวอย่าง Input ที่สุ่มมา ทั้งในอดีตและปัจจุบัน ซึ่งนั่นคือ IIR จะมีการ Feedback ซึ่งจะทำการเปรียบเทียบกับสมการ FIR แล้วกระแสการสุ่มตัวอย่าง Output $y(n)$ จะเป็นเพียง Function ของค่า Input ของอดีตและปัจจุบันเท่านั้นเอง แต่อย่างไรก็ตามเมื่อ b_k มีค่าเท่ากับ 0 สมการ IIR คือ สมการ FIR นั่นเอง

และจากสมการข้างล่าง นี้ก็คือ Transfer Function ของ FIR และ IIR Digital Filter ซึ่งค่า Transfer Function เหล่านี้จะมีประโยชน์มาก ในการคำนวณหาค่า Frequency Response

$$H(z) = \frac{\sum_{k=0}^n a_k \cdot z^{-k}}{1 + \sum_{k=1}^m b_k \cdot z^{-k}} \quad \text{:IIR digital filter (5)}$$

$$H(z) = \sum_{k=0}^{n-1} h(k) \cdot z^{-k} \quad \text{:FIR digital filter (6)}$$

2.10 SUMMARY OF KEY CHARACTERISTIC FEATURES OF FIR FILTER

2.10.1 Characteristic ของ FIR Digital Filter จะแสดงดังสมการ

$$y(n) = \sum_{k=0}^{n-1} h(k).x(n-k) \quad (7)$$

$$H(z) = \sum_{k=0}^{n-1} h(k).z^{-k} \quad (8)$$

โดยที่ $h(k)$; $k=0,1,2,\dots,n-1$ เป็นสัมประสิทธิ์ในการตอบสนองต่อ ของตัวกรองสัญญาณ

$H(z)$ เป็น Transfer Function ของ ตัวกรองสัญญาณ

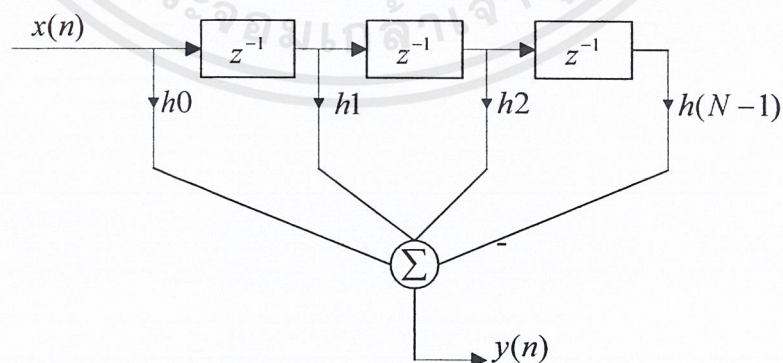
N เป็น Filter Length

นั่นก็คือจำนวนสัมประสิทธิ์ของตัวกรองสัญญาณของสมการแรกเป็น Fir Difference Equation ซึ่งมันจะเป็นสมการใน Time Domain และจะพิจารณา FIR Filter ในรูปของ Non Recursive Form ซึ่งจะทำให้การสุ่มตัวอย่าง Output $y(n)$ เป็น Function เพียงค่าของ Input ในอดีตและปัจจุบัน เท่านั้น $x(n)$ เมื่อตัวกรองสัญญาณ ถูกนำไปใช้ในรูปแบบนี้ นั่นคือ จะเป็นการหาค่าโดยตรง จะทำได้ค่าคงที่ (Stable) และ Transfer -Function ของตัวกรองสัญญาณ จะใช้ในการวิเคราะห์และหาค่าตัวกรองสัญญาณในทาง Frequency Response

2.10.2 FIR Filter มีความเป็น Linear Phase Response ที่แท้จริง

2.10.3 FIR Filter ง่ายที่จะนำไปใช้ในระบบ DSP Processor ทั้งหมดมีความเหมาะสมที่จะใช้กับ

FIR Filtering Non-Recursive FIR Filter จะมีข้อผิดพลาดน้อยกว่าผลของความยาวที่จำกัดกว่า



รูปที่ 2.5 โครงสร้างของ FIR Digital Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 SUMMARY OF THE BASIC FEATURES OF IIR FILTER

Characteristic ของ IIR Digital Filter ถูกกำหนดตาม Recursive ดังนี้

$$y(n) = \sum_{k=0}^{\infty} h(k) \cdot x(n-k) = \sum_{k=0}^{\infty} a_k \cdot x(n-k) - \sum_{k=1}^{\infty} a_k \cdot x(n-k) \quad (9)$$

โดยที่ $h(k)$ เป็นการตอบสนองต่อ Impulse ของตัวกรองสัญญาณ ซึ่งจะมีช่วงเวลาที่ไม่มีจำกัด

a_k และ b_k เป็นสัมประสิทธิ์ของ Filter

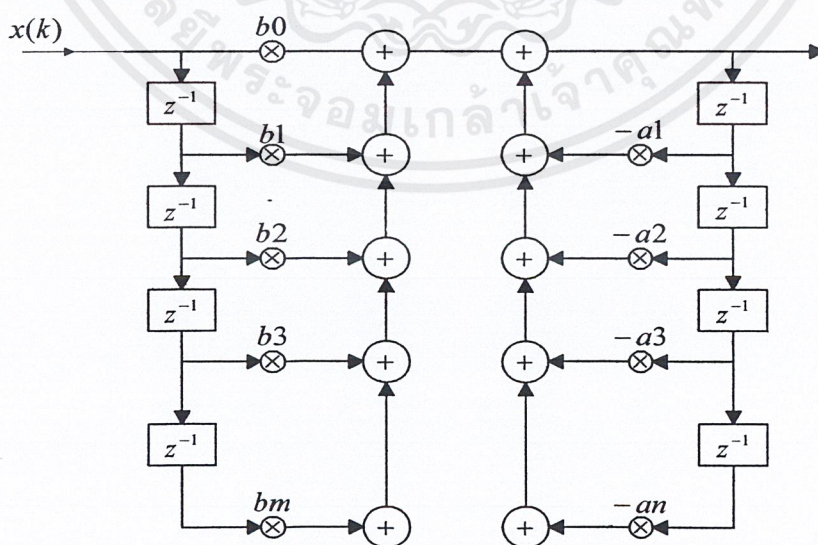
$x(n)$ และ $y(n)$ เป็น Input และ Output ของ Filter

Transfer Function สำหรับ IIR Filter ถูกกำหนดโดย

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}} = \frac{\sum_{k=0}^n a_k \cdot z^{-k}}{1 + \sum_{k=0}^m b_k \cdot z^{-k}} \quad (10)$$

ส่วนที่สำคัญในการออกแบบ IIR Filter คือการค้นหาค่าที่เหมาะสมสำหรับค่าสัมประสิทธิ์และให้สอดคล้องกับลักษณะของตัวกรองสัญญาณ ใน Frequency Response ดังนั้นสมการดังกล่าวทั้งสองจึงเป็น Characteristic Equation สำหรับ IIR Filter

จากสมการแรก กล่าวได้ว่ากระแสการสุ่มตัวอย่าง Output, $y(n)$ เป็น Function ของ Output ที่ผ่านมา $y(n)$ ซึ่งจะเท่ากับค่าการสุ่มตัวอย่าง Input ทั้งในปัจจุบันและอดีต นั่นคือ IIR Filter เป็นระบบ Feedback ซึ่งปกติแล้ว IIR Filter ต้องการเพียงสัมประสิทธิ์ 2-3 ตัวมากกว่า FIR Filter สำหรับข้อกำหนดเดียวกัน ซึ่ง IIR Filter จะถูกใช้เมื่อจุด Cut off ที่แคบและมี Throughput ที่สูง เป็นความต้องการที่สำคัญ เหตุผลที่จะทำให้ IIR Filter ไม่คงที่ (Unstable)



รูปที่ 2.6 โครงสร้าง IIR Digital Filter

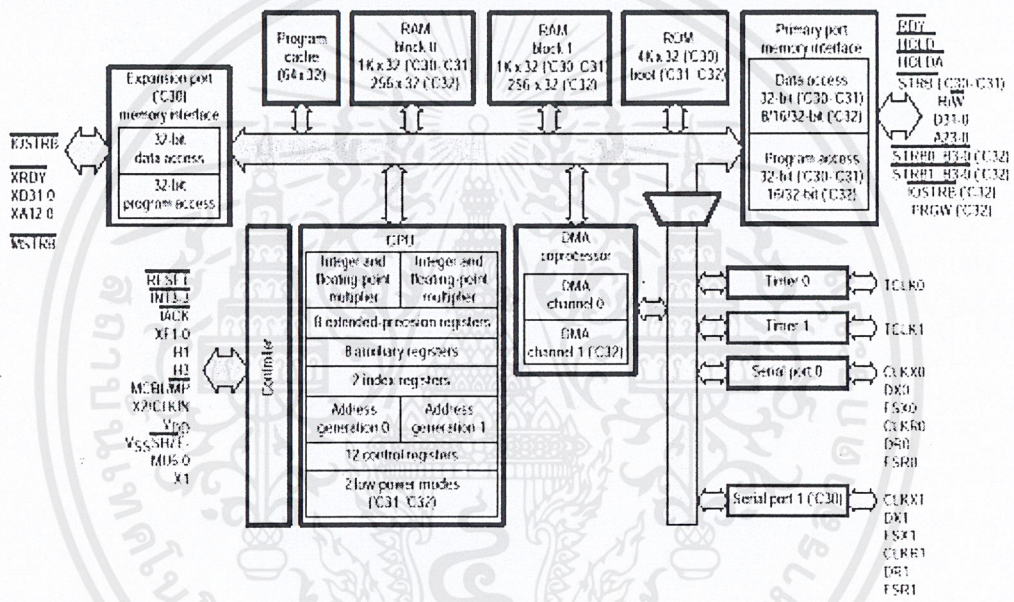
บทที่ 3

สถาปัตยกรรมของ TMS320C31

ไอซีเบอร์ TMS320C31 ถูกสร้างและออกแบบมาเพื่อตอบสนองความต้องการทางการประมวลผลบนพื้นฐานของคณิตศาสตร์ขั้นสูงและการแก้ปัญหาทาง Hard ware และ Soft ware

ประสิทธิภาพอันสูงของ TMS320C31 เกิดจากความแน่นอนของควมกว้างทางไดนามิกของส่วนคำนวณทางทศนิยม มีหน่วยความจำขนาดใหญ่ การทำงานแบบขนานและมี DMA Controller เพื่อการทำงานที่มีประสิทธิภาพมากขึ้น

Figure 1-1. TMS320C3x Devices Block Diagram



รูปที่ 3.1 แสดงสถาปัตยกรรมของ TMS320C31

สถาปัตยกรรม TMS320C31 ประกอบด้วยส่วนต่างๆที่สำคัญดังนี้

3.1 หน่วยประมวลผลกลาง (CPU)

หน่วยประมวลผลกลางจะประกอบไปด้วย

3.1.1 ตัวคูณ (Multiplier) มีหน้าที่การคูณ และสามารถคูณจำนวนเต็ม 24 บิต และ จำนวนทศนิยม 32 บิต โดยใช้เวลาเพียง 1 รอบ ซึ่งการคำนวณจำนวนทศนิยมจะใช้เวลา 50 ns ต่อรอบ และสามารถประมวลผลแบบขนานได้ เพื่อที่จะเพิ่มความสามารถในการประมวลผลข้อมูล โดยใช้คำสั่งแบบขนานให้ทำการคูณและการทำงานของหน่วยการคำนวณทางคณิตศาสตร์ (ALU) ภายใน 1 รอบ

เมื่อตัวคูณ ทำการคูณทศนิยม อินพุทที่ใช้มีขนาด 32 บิต และผลลัพธ์ที่ได้จะมีขนาด 40 บิต แต่ถ้าตัวคูณทำการคูณจำนวนเต็มอินพุทจะมีขนาดเพียง 24 บิต และจะได้ผลลัพธ์ขนาด 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 หน่วยการคำนวณทางคณิตศาสตร์ (Arithmetic Logic Unit [ALU]) ใน 1 รอบการทำงานของหน่วยการคำนวณทางคณิตศาสตร์ จะประมวลผลจำนวนเต็มได้ 32 บิต และทศนิยมได้ 40 บิต โดยผลลัพธ์จากหน่วยการคำนวณทางคณิตศาสตร์ ยังคงมีขนาด 32 บิต ถ้าเป็นจำนวนเต็ม และ 40 บิต ถ้าเป็นทศนิยม barrel shifter จะมีหน้าที่การเลื่อนข้อมูล 32 บิต ทั้งซ้ายและขวาใน 1 รอบการทำงาน

บัสภายใน (CPU1/CPU2 และ REG1/REG2) จะเป็นตัวพาข้อมูลที่จะนำมาประมวลผล 2 ตัวจากหน่วยความจำและอีก 2 ตัวจากรีจิสเตอร์ ดังนั้นจึงสามารถทำการคูณ บวกหรือลบแบบขนานได้ใน 1 รอบการทำงาน

3.1.3 หน่วยช่วยคำนวณทางคณิตศาสตร์ (Auxiliary Register Arithmetic Unit [ARAUs]) หน่วยช่วยคำนวณทางคณิตศาสตร์เป็นรีจิสเตอร์ที่ช่วยในการทำงานแบบขนานกับตัวคูณและหน่วยคำนวณทางคณิตศาสตร์ โดยหน่วยช่วยคำนวณทางคณิตศาสตร์ จะมี 2 ตัว คือ ARAU0 และ ARAU1 หน่วยช่วยคำนวณทางคณิตศาสตร์ จะสามารถเก็บแอดเดรสได้ 2 แอดเดรส ใน 1 รอบการทำงานเพื่อใช้ในการอ้างตำแหน่งแบบต่างๆ

3.1.4 CPU Register File ใน TMS320C31 จะมีรีจิสเตอร์อยู่ 28 ตัว ซึ่งตรงกับ CPU โดยรีจิสเตอร์ทั้งหลายเหล่านี้จะทำงานกับตัวคูณ และหน่วยคำนวณทางคณิตศาสตร์ และยังสามารถใช้เพื่อวัตถุประสงค์อื่นอีกด้วยอย่างไรก็ตามรีจิสเตอร์เหล่านี้ก็ยังมีหน้าที่พิเศษบางอย่างอีกด้วย เช่น ใช้สำหรับการทำงานที่เกี่ยวข้องกับจำนวนเลขทศนิยม และรีจิสเตอร์ช่วย (Auxiliary Register) ทั้ง 8 ตัว ยังใช้การอ้างแอดเดรสแบบทางอ้อม (indirect addressing register) ด้วยรีจิสเตอร์ที่เหลือจะทำหน้าที่เกี่ยวกับระบบต่างๆเช่นการอ้างแอดเดรส การจัดการเกี่ยวกับสแตค (stack) การบอกสถานะโปรเซสเซอร์ การอินเตอร์รัพต์ และการเคลื่อนย้ายข้อมูลเป็นบล็อก

-extend-precision register (R0-R7) สามารถใช้เก็บค่าจำนวนเต็ม 32 บิตและทศนิยม 40 บิต โดยจะเก็บค่าลงในบิตที่ 0-39 แต่ถ้าจำนวนเต็มแบบมีเครื่องหมายก็จะทำการเก็บบิตที่ 0-31 ส่วนบิตที่ 32-39 จะไม่ใช่

-Auxiliary Register (AR0-AR7) จะถูกใช้โดยหน่วยประมวลผลหลักและถูกควบคุมโดยหน่วยช่วยคำนวณทางคณิตศาสตร์ทั้ง 2 ตัว หน้าที่แรกของ AR ยังใช้ให้ส่วนที่เป็นตัวนับในการวนลูปหรือจะใช้เพื่องานอื่นๆที่เกี่ยวข้องกับตัวคูณและหน่วยคำนวณทางคณิตศาสตร์

-data page pointer (DP) ส่วนรีจิสเตอร์ขนาด 32 บิต โดยที่ 8 บิตแรก จะถูกใช้ในการอ้างแอดเดรสแบบโดยตรง (direct addressing mode) ซึ่งใช้เป็นตัวชี้ (pointer) ไปยังหน้า (page) ต่างๆของข้อมูล

-index register (IRO,IR1) จะถูกใช้โดยหน่วยช่วยคำนวณทางคณิตศาสตร์ ในการเก็บค่าดัชนีในการชี้แอดเดรส (indexed addressing mode)

-block size register (BK) มีขนาด 32 บิตซึ่งหน่วยช่วยคำนวณทางคณิตศาสตร์จะใช้ในการอ้างแอดเดรสแบบเก็บ circular addressing เพื่อเป็นการเก็บค่าของขนาดของบล็อกข้อมูล

-system stack pointer (SP) เป็นรีจิสเตอร์ขนาด 32 บิต ซึ่งเป็นค่าแอดเดรสของยอดของแอสแตก โดยแอสแตก (SP) จะชี้ไปยังค่าสุดท้ายซึ่งถูกเก็บอยู่บนแอสแตก การ PUSH จะกระทำก่อนที่จะเพิ่มค่าของแอสแตก และการ POP หลังการลดค่าของแอสแตก คำสั่งที่จะทำงานกับแอสแตกได้คือ interrupts, traps, PUSH และ POP

-status register (ST) เป็นรีจิสเตอร์ที่เก็บค่าข้อมูลต่างๆ ที่แสดงสถานะของ CPU เช่นค่าของ flag ซึ่งจะแสดงสถานะของผลลัพธ์ที่ได้จากการคำนวณทางลอจิกด้วย

-CPU/DMA interrupt enable register (IE) เป็นรีจิสเตอร์ขนาด 32 บิต ใช้สำหรับกำหนดขนาดเอนาเบิ้ลอินเตอร์รัพต์ (enable interrupt) ของ CPU โดยการเอนาเบิ้ลของ CPU จะใช้บิตที่ 0-10 และการเอนาเบิ้ลของ DMA จะใช้บิตที่ 16-26 ค่า "1" จะเป็นการเอนาเบิ้ลและ "0" จะเป็นการดีสเอนาเบิ้ล

-I/O Flag register (IOF) เป็นรีจิสเตอร์ในการควบคุมขา XFO และ XFI

-repeat counter (RC) เป็นรีจิสเตอร์ขนาด 32 บิต ใช้เก็บค่าของจำนวนเท่าหรือจำนวนครั้งของบล็อกข้อมูลซึ่งถูกกระทำซ้ำ เมื่อโปรเซสเซอร์ทำงานในโหมดของการกระทำข้อมูลซ้ำ (repeat mode)

-repeat start address register (RS) รีจิสเตอร์ขนาด 32 บิต จะถูกใช้ในการเก็บแอดเดรสเริ่มต้นของบล็อกข้อมูลที่จะถูกกระทำซ้ำ

-repeat end address register (RE) เป็นรีจิสเตอร์ขนาด 32 บิต จะถูกเก็บแอดเดรสสุดท้ายของบล็อกข้อมูลซึ่งถูกกระทำซ้ำ

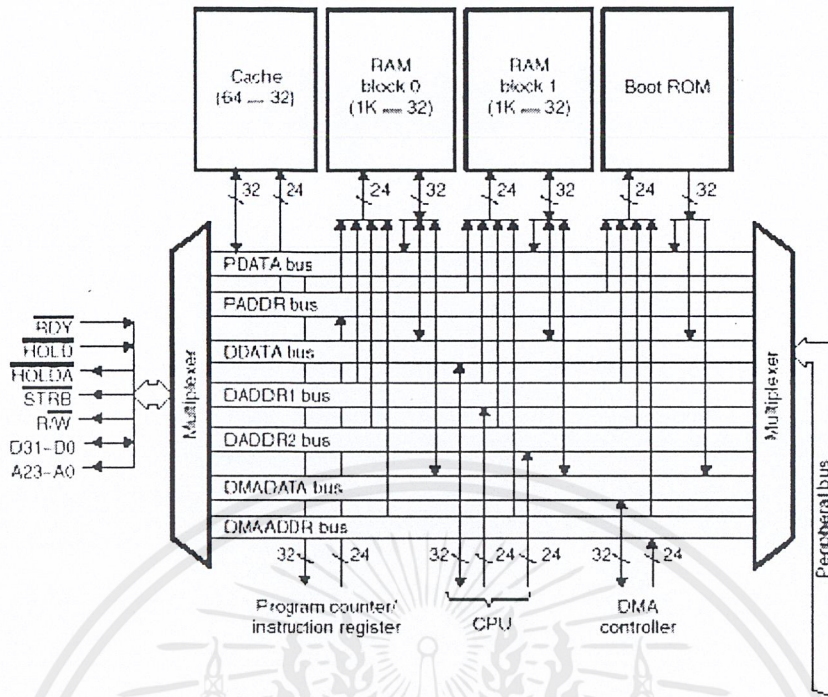
-program counter (PC) เป็นรีจิสเตอร์ขนาด 32 บิต ซึ่งเก็บแอดเดรสของคำสั่งที่จะถูกนำมาอ่านคำสั่งถัดไป

Register Symbol	Register Machine Value (hex)	Assigned Function Name
R0	00	Extended-precision register 0
R1	01	Extended-precision register 1
R2	02	Extended-precision register 2
R3	03	Extended-precision register 3
R4	04	Extended-precision register 4
R5	05	Extended-precision register 5
R6	06	Extended-precision register 6
R7	07	Extended-precision register 7
AR0	08	Auxiliary register 0
AR1	09	Auxiliary register 1
AR2	0A	Auxiliary register 2
AR3	0B	Auxiliary register 3
AR4	0C	Auxiliary register 4
AR5	0D	Auxiliary register 5
AR6	0E	Auxiliary register 6
AR7	0F	Auxiliary register 7
DP	10	Data-page pointer
IR0	11	Index register 0
IR1	12	Index register 1
BK	13	Block-size register
SP	14	System-stack pointer
ST	15	Status register
IE	16	CPU/DMA interrupt-enable
IF	17	CPU interrupt flags
IOF	18	I/O flags
RS	19	Repeat start address
RE	1A	Repeat end-address
RC	1B	Repeat counter

ตารางที่ 3.1 ชื่อและหน้าที่ต่างๆของรีจิสเตอร์ใน CPU

3.2 หน่วยความจำ (Memory Organization)

หน่วยความจำของ TMS320C31 จะมีขนาด 16M และมีขนาดของ 1word 32 บิต โดยในหน่วยความจำนี้จะใช้กับโปรแกรมข้อมูล และการอินพุต เอาท์พุต ดังนั้นค่าของสัมประสิทธิ์โปรแกรมหรือข้อมูลจะถูกเก็บได้ทั้ง RAM และ ROM



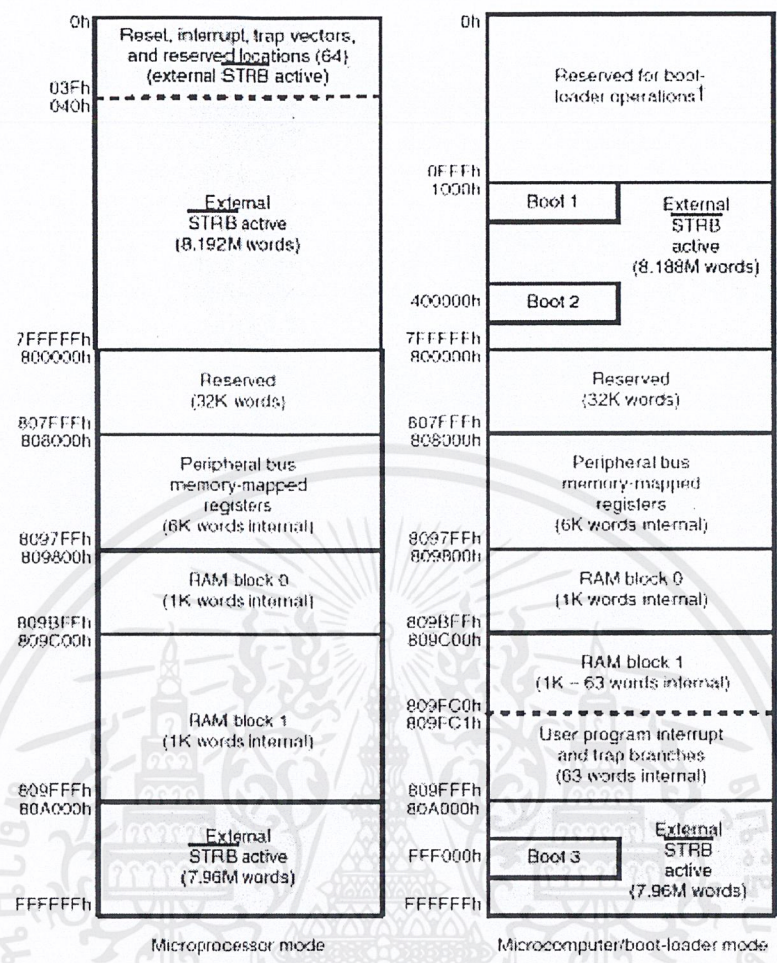
รูปที่ 3.2 แสดงหน่วยความจำของ TMS320C31

3.2.1 RAM , ROM

รูปที่ 3.2 จะแสดงการจัดหน่วยความจำภายใน TMS320C31 จะแบ่งออกเป็น บล็อก 0 และบล็อก 1 ซึ่งจะมีขนาดบล็อกละ 1Kx32 และ ROM จะมีขนาด 4Kx32 ทั้ง RAM และ ROM สามารถถูกเข้าใช้ได้โดย CPU 2 ครั้งใน 1 รอบของการทำงานได้ ดังนั้น การที่มีบัสของ โปรแกรมแยกกันและบัสของ DMA แยกกัน ทำให้การอ่านโปรแกรม การอ่านและการเขียนข้อมูลหรือการทำงานของ DMA ที่สามารถทำงานแบบขนานได้

3.2.1 ตารางหน่วยความจำ (Memory Maps)

ตารางหน่วยความจำจะขึ้นอยู่กับว่าจะให้โปรเซสเซอร์ประมวลผลในโหมดของโปรเซสเซอร์ หรือไมโครคอมพิวเตอร์ แต่ในโครงการนี้จะให้โปรเซสเซอร์ประมวลผลในโหมดของไมโครโปรเซสเซอร์ แสดงดังรูปที่ 3.3 แอดเดรสที่ 800000h จนถึง 801FFFh จะถูกใช้สำหรับ expansion bus ซึ่งจะถูกเข้าถึงได้โดยการให้สัญญาณ IOSTRB แอดเดรสที่ 806000h ถึง 807FFFh จะไม่ใช่ทริกเกอร์จะอยู่ที่แอดเดรส 808000h ถึง 8097FFFh RAM บล็อก 0 จะอยู่ที่ 809800h ถึง 809BFFFh และบล็อกที่ 1 จะอยู่ที่ 809C00h ถึง 809FFFh แอดเดรส 80A000h จนถึง 0FFFFFFFh จะถูกใช้โดยอุปกรณ์ภายนอก (เมื่อสัญญาณ /STRB แอดเดรสที่ f)



รูปที่ 3.3 แสดงการจัดหน่วยความจำของ TMS320C31

3.2.3 โหมดการอ้างอิงหน่วยความจำ (Memory Address Modes)

TMS320C31 จะแบ่งโหมดของการสร้างแอดเดรสออกเป็น 5 กลุ่ม โดยมีโหมดการอ้างอิงแอดเดรสทั้งหมด 6 แบบดังนี้

3.2.3.1 General addressing mode

- Register ค่าที่นำมาคำนวณจะเก็บที่ รีจิสเตอร์
- Short immediate ค่าที่นำมาคำนวณจะเป็นค่าที่ไหลโดยตรงขนาด 16 บิต จากคำสั่ง
- Direct ค่าที่นำมาคำนวณจะถูกเก็บอยู่ในตำแหน่ง ซึ่งมีแอดเดรสขนาด 24 บิตแสดงอยู่ในคำสั่ง
- Indirect รีจิสเตอร์จะช่วยเก็บแอดเดรสของข้อมูล

3.2.3.2 Three-operand addressing mode

- Register
- Indirect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.3 Parallel addressing mode

-Register ข้อมูลจะอยู่ใน

-Indirect

3.2.3.4 Long-immediate addressing mode

-Long-immediate ข้อมูลจะเป็นค่าที่โหลดโดยตรงขนาด 24 บิต จากคำสั่ง

3.2.3.5 Conditional branch addressing mode

-Register

-PC-relative ค่าที่มีเครื่องหมายขนาด 16 บิตจะถูกบวกค่าใน

3.3 Internal Bus Operation

ข้อดีที่สำคัญอีกประการหนึ่งของ TMS320C31 คือการที่มีบัสภายในและสามารถประมวลผลแบบขนานได้ บัสโปรแกรมที่แยกจากกัน (PADDR และ PDATA) บัสข้อมูลที่แยกจากกัน (DADDR1, DADDR2 และ DDATA) และบัส DMA ที่แยกกัน (DMAADDR และ DMADATA) ทำให้สามารถทำการประมวลผลแบบขนานได้ บัสเหล่านี้สามารถต่อได้กับหน่วยความจำภายใน หน่วยความจำภายนอก หรืออุปกรณ์ภายนอกอื่นๆ รูปที่ 3.4 จะแสดงการต่อบัสต่างๆเหล่านี้

PC จะถูกต่ออยู่กับ PADDR (ขนาด 24 บิต) IR จะต่ออยู่กับ PDATA (ขนาด 32 บิต) ซึ่งบัสเหล่านี้สามารถอ่านคำสั่ง ทุกๆรอบของการทำงาน

DADDR1 และ DADDR2 (ขนาด 24 บิต) และ DDATA (ขนาด 32 บิต) จะใช้ในการติดต่อกับหน่วยความจำภายในทุกๆรอบของการทำงาน DDATA จะใช้ในการส่งข้อมูลไปยัง CPU โดยผ่านบัส CPU1 และ CPU2 บัส CPU1 และ CPU2 สามารถส่งข้อมูลจากหน่วยความจำ 2 ข้อมูลไปยังตัวคูณ (Multiplier) หน่วยการคำนวณทางคณิตศาสตร์ (ALU) และรีจิสเตอร์ในทุกๆรอบของการทำงาน นอกจากนี้ภายใน CPU มีบัสรีจิสเตอร์ REG1 และ REG2 ซึ่งใช้ในการหาข้อมูล 2 ค่าจากรีจิสเตอร์ไปยังคูณ และหน่วยการคำนวณทางคณิตศาสตร์ในทุกๆรอบของการทำงาน รูปที่ 3.5 แสดงบัสภายในต่างๆ DMA Controller จะใช้ DMAADDR (ขนาด 24 บิต) และ DMADATA (ขนาด 32 บิต) โดยบัสเหล่านี้ทำให้ DMA สามารถติดต่อกับหน่วยความจำแบบขนานได้

3.4 External Bus Operation

หน่วยความจำและอุปกรณ์ภายนอกสามารถอินเตอร์เฟสกับ TMS320C31 ได้ 2 ทางคือทาง Primary Bus และ Expansion Bus การอินเตอร์เฟสกับหน่วยความจำหรืออุปกรณ์ภายนอกที่มีความเร็วต่ำกว่า TMS320C31 จะต้องเพิ่ม Wait State เข้าไป โดยอาศัยการควบคุมของ Memory Mapped Control Register และสัญญาณจากภายนอก

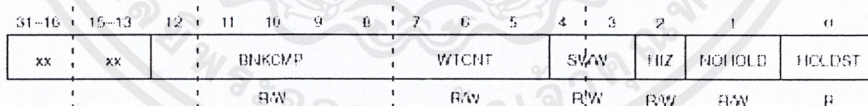
Primary Bus มีขนาดของบัสข้อมูล (data bus) 32 บิต, บัสข้อมูล (address bus) 34 บิต และสัญญาณควบคุม (control signal) อีก 1 ชุด โดยบัสทั้งสองสามารถถูกควบคุมได้โดยการใช้โปรแกรม (software) ตั้งงานและสัญญาณควบคุมจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Peripheral Address	
800060h	Expansion bus control (CS0 only)
800061h	Reserved
800062h	Reserved
800063h	Reserved
800064h	Primary bus control (CS0, CS1)
800065h	Reserved
800066h	Reserved
800067h	Reserved
800068h	Reserved
800069h	Reserved
80006Ah	Reserved
80006Bh	Reserved
80006Ch	Reserved
80006Dh	Reserved
80006Eh	Reserved
80006Fh	Reserved

รูปที่ 3.4 memory-mapped external interface control register

อุปกรณ์ภายนอกสามารถเข้าถึงได้โดยการใช้สัญญาณ /SPRB,/MSTRB และ /IOSTRB โดยสัญญาณ /STRB จะใช้เมื่อต้องการ primary bus ส่วน expansion จะสามารถใช้งานได้โดย 2 แบบคือ -การเข้าใช้ memory สามารถทำได้โดยให้ /MSTRB เป็น 0 โดย /MSTRB จะมีวงรอบการทำงานเหมือนกับ /STRB -อุปกรณ์ภายนอกสามารถเข้าใช้ได้โดยให้สัญญาณ /IOSTRB เป็น 0 primary bus และ expansion bus จะมี control register ดังรูปที่ 3.5



Notes: 1) xx - reserved bit, read as 0
 2) R - read, W - write

รูปที่ 3.5 primary bus control register

Abbreviation	Reset Value	Name	Description
HOLDST	0	Hold status bit	This bit signals whether the port is being held (HOLDST = 1) or is not being held (HOLDST = 0). This status bit is valid whether the port has been held through hardware or software.
NOHOLD	0	Port hold signal	NOHOLD allows or disallows the port to be held by an external HOLD signal. When NOHOLD = 1, the IC3x takes over the external bus and controls it, regardless of serviced or pending requests by external devices. No hold acknowledge (HOLDA) is asserted when a HOLD signal is received. It is asserted if an internal hold is generated (HIZ = 1).
HIZ	0	Internal hold	When set (HIZ = 1), the port is put in hold mode. This is equivalent to the external HOLD signal. By forcing a high-impedance condition, the IC3x can relinquish the external memory port through software. HOLDA goes low when the port is placed in the high-impedance state.
SWW	11	Software wait mode	In conjunction with WTCNT, this 2-bit field defines the mode of wait-state generation. (See Table 9-6.)
WTCNT	111	Software wait mode	This three-bit field specifies the number of cycles to use when in software wait mode for the generation of internal wait states. The range is 0 (WTCNT = 0 0 0) to 7 (WTCNT = 1 1 1) H1/H3 cycles. (See Section 9.4.)
BANKCMP	10000	Bank compare	This 5-bit field specifies the number of MSBs of the address to be used to define the bank size. (See Table 9-6.)

ตารางที่ 3.2 แสดงหน้าที่ต่างๆของ primary bus control register



รูปที่ 3.6 expansion bus control register

Abbreviation	Reset Value	Name	Description
SWW	11	Software wait mode	In conjunction with the WTCNT, 2-bit field defines the mode of wait-state generation. (See Table 9-5.)
WTCNT	111	Software wait mode	This 3-bit field specifies the number of cycles to use when in software wait mode for the generation of internal wait state. The range is 0 (WTCNT = 0 0 0) to 7 (WTCNT = 1) H1/H3 cycles. (See Section 9.4.)

ตารางที่ 3.2 แสดงหน้าที่ต่างๆของ expansion bus control register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น. อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address จะเปลี่ยนที่ขอบขาลงของ H1 ทุกๆวงรอบของสัญญาณบนสายบัสจะขึ้นอยู่กับสัญญาณนาฬิกา H1 โดย 1 รอบของ H1 เริ่มนับจากขอบขาลงจนถึงขอบขาลงของลูกถัดไป สำหรับการทำงานแบบความเร็วเต็มที (คือไม่ใช่ wait time) ในการเขียนข้อมูลจะใช้เวลา 2 รอบของ H1 และการอ่านข้อมูลจะใช้ 1 รอบของ H1 แต่ถ้าการอ่านข้อมูลกระทำต่อคำสั่งการเขียนข้อมูล การอ่านข้อมูลจะใช้เวลา 2 รอบของ H1 ซึ่งหลักการนี้จะใช้ได้ทั้งกับ primary bus และ expansion bus (เมื่อใช้ /MSTRB)

ถ้าการเขียนข้อมูลเป็นการกระทำภายในโดยไม่มีอินเทอร์เฟซกับอุปกรณ์ภายนอก (คือการใช้จาก CPU และ DMA) วงรอบของการเขียนข้อมูลจะใช้เพียง 1 รอบของ H1 เท่านั้น

สัญญาณ /MSTRB และ /STRB จะแอกทีฟที่สถานะ 0 สำหรับการอ่านและเขียนข้อมูลในขณะที่สภาวะก่อนหลังของการแอกทีฟของ /MSTRB และ /STRB ในการเขียนข้อมูลจะมี transition cycle ในระหว่าง transition cycle จะเกิด

1. /MSTRB และ /STRB จะเป็นสถานะ high
2. (X)R/W จะเปลี่ยนสถานะที่ขอบขาขึ้นของ H1
3. ถ้าเป็นสถานะของการเขียนข้อมูล address จะเปลี่ยนสถานะที่ขอบขาขึ้นของ H1 และถ้าเป็นสถานะของการอ่านข้อมูล

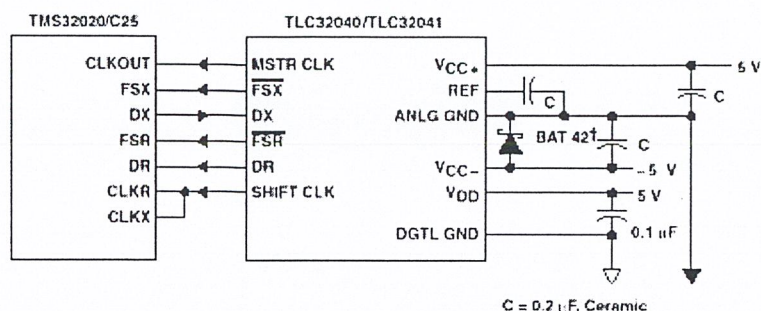
3.5 การเชื่อมต่อระหว่าง TMS320C31 กับ TLC32040 AIC

3.5.1 ลักษณะของ TLC32040 AIC

- ใช้ในการแปลงอนาลอกเป็นดิจิตอล และดิจิตอลเป็นอนาลอก ขนาด 14 บิต
- สามารถเปลี่ยนแปลงอัตราการแซมปลิง (Sampling) ทั้งของอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก ด้วยความแม่นยำที่ 20,000 ตัวอย่างต่อวินาที
- การติดต่อระหว่าง TLC32040 AIC กับ TMS320C31 จะมีการติดต่อผ่านทาง พอร์ตอนุกรม (Serial Port)
- ใช้เทคโนโลยีซีมอส (CMOS)
- สามารถเลือกใช้ช่องอนาลอกอินพุทช่วยเหลือ (auxiliary analog input channel) ได้

บนบอร์ดการเชื่อมต่อทาง Serial Port ระหว่าง TMS320C31 กับ TLC32040 AIC จะมีตัวหัวไว้เชื่อมตัว (เป็นลักษณะคล้ายกับ Jumper) เมื่อไม่มีการเชื่อมต่อก็คือการไม่ใช้ TLC32040 AIC แต่ถ้ามีการเชื่อมต่อก็จะเป็นการเชื่อมต่อและใช้งานระหว่าง TMS320C31 กับ TLC32040 AIC

จากตัว TLC32040 AIC จะเชื่อมต่อ analog input และ analog output กับหัวต่อแบบ RCA โดยสัญญาณที่เข้าและออกผ่านทางหัวต่อ RCA นี้จะมีแรงดัน +/-3V peak ที่หัวต่อเอาท์พุท สามารถเชื่อมต่อโดยตรงกับลำโพงก็ได้ แต่อาจไม่มีระดับสูงพอจึงอาจจะต้องมีวงจรขยายสัญญาณภายนอกหรือต่อกับลำโพงแบบ high impedance



รูปที่ 3.7 แสดงการเชื่อมต่อระหว่าง TMS320C31 กับ TLC32040

3.5.2 รายละเอียดของแต่ละขาของ TLC32040 AIC ที่ใช้เชื่อมต่อกับ TMS320C31

RESET ที่ขานี้จะเป็นอินพุทที่ใช้ในการรีเซ็ตตัว TMS320C31

SCLK (shift clock) เป็นคล็อกที่ใช้ในการส่งข้อมูลแบบอนุกรม (ผ่านทาง serial port) ความถี่นี้จะเกิดจาก Master Clock Signal Frequency หรือ MCLK ทหารด้วย 4

FSR (Frame Sync receive) ในหมวดการส่งข้อมูลแบบอนุกรมเมื่อขา FSR เป็น 0 จะทำให้พอร์ตอนุกรมของ TMS320C31 เริ่มรับค่าจากขา DR ของตัว TLC32040 AIC ค่าของบิต DR จะแสดงผลบนขา DR ก่อนที่ขา FSR จะเป็น 0

DR (Data Receive) ใช้สำหรับส่งข้อมูลจากอนาลอกเป็นดิจิตอล จาก TLC32040 AIC ไปที่พอร์ตอนุกรมของ TMS320C31 การส่งข้อมูลทีขานี้จะต้องทำการซิงโครไนซ์ (Synchronized) กับ (SCLK (Shift Clock Signal))

FSX (Frame Sync Transmit) เหมือน FAR แต่กลับกันเป็นการส่งค่าจากพอร์ตอนุกรมของ TMS320C31 ไปยังขา DX ของ TLC32040 AIC

DX (Data Transmit) เหมือน DR แต่กลับกันตรงที่เป็นตัวรับค่าดิจิตอลเป็นอนาลอกจากพอร์ตอนุกรมของ TMS320C31 โดยการส่งข้อมูลทีขานี้จะต้องทำการซิงโครไนซ์ (Synchronized) กับ SCLK (Shift Clock Signal)

MCLK (Master clock frequency) ขาที่ 6 ถูกใช้ป็นสัญญาณทางลอจิกที่สำคัญแก่ TLC32040 – AIC โดยสัญญาณคล็อกที่ป้อนเข้ามาทีขานี้ใช้สำหรับการแปลงเป็นสัญญาณต่างๆที่ใช้ใน TLC32040 AIC เช่น Shift Clock Signal, Swich Capacitor Filter Clock และสัญญาณการแปลงอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก

IN+ (Non –inverting Input) เป็นอินพุทของสัญญาณอนาลอกที่ไม่กลับเฟส

IN- (Inverting Input) เป็นอินพุทของสัญญาณอนาลอกที่กลับเฟส (ในที่นี้จะไม่ใช้อินพุทตัวนี้เลยทำการต่อกับกราวนด์)

Out + (Non-inverting Output) เป็นเอาต์พุทที่ไม่กลับเฟสของสัญญาณอนาลอก

Out- (Inverting Output) เป็นเอาต์พุทที่ไม่กลับเฟสของสัญญาณอนาลอก

VCC (Voltage Supply) เป็นสัญญาณไฟบวกที่ป้อนให้แก่ TLC32040 AIC มีขนาดแรงดัน

5. โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 รายละเอียดของขาแต่ละขาของ TMS320C31 ที่ใช้เชื่อมต่อกับ TLC32040

XFO เป็นขาที่ใช้ส่งสัญญาณในการจะทำการรีเซ็ต ตัว TLC32040 AIC

CLKRO เป็นขาที่ใช้ในการรับคล็อกที่ใช้ในการรับข้อมูลแบบอนุกรมโดยเชื่อมต่อกับ SCLK (Shift Clock) ของ TLC32040 เพื่อให้ซิงโครไนซ์ (Synchronized) กัน

CLKXO เป็นขาที่ใช้ในการรับคล็อกที่ใช้ในการส่งข้อมูลแบบอนุกรม โดยเชื่อมต่อกับ SCLK (Shift Clock) ของ TLC32040 เพื่อให้ซิงโครไนซ์ (Synchronized) กัน

FSRO เป็นการรับสัญญาณว่าจะมีการส่งข้อมูลมาจาก TLC32040 AIC ในการแปลงอนาลอกเป็นดิจิทัล

DRO เป็นขาที่ใช้สำหรับรับข้อมูลจาก TLC32040 AIC

FSXO เป็นขาที่ใช้ในการรับสัญญาณจาก TLC32040 เมื่อพร้อมที่จะรับค่าจาก TMS320C31 ในการแปลงดิจิทัลเป็นอนาลอก

DXO เป็นขาที่ใช้ในการส่งข้อมูลจาก TMS320C31 ไปยัง TLC32040 AIC

TCLKO ขาที่ใช้ในการส่งสัญญาณนาฬิกาจาก TMS320C31 เพื่อให้ TLC32040 AIC ไปใช้เป็น MCLK (Master Clock Frequency)

บทที่ 4

ทฤษฎีและการออกแบบตัวกรองเชิงเลข

ตัวกรองโดยทั่วไปนั้นสามารถแบ่งออกได้เป็นสองประเภทด้วยกันคือ

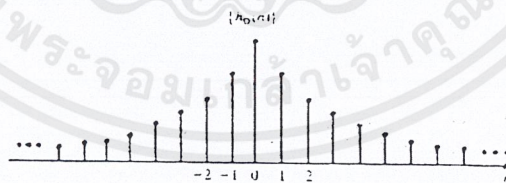
- ตัวกรองเชิงอุปมาน (Analog filter)
- ตัวกรองเชิงเลข (Digital filter)

โดยทั่วไปนั้น ตัวกรองเชิงเลขนี้ สามารถแบ่งย่อยได้อีกตามลักษณะของผลตอบสนองอิมพัลส์ของตัวกรอง ซึ่งจะเป็นตัวกำหนดคุณสมบัติของระบบเชิงเลข โดยจะแบ่งออกได้เป็นสองชนิดด้วยกันคือ

- 1 ตัวกรองผลตอบสนองอิมพัลส์จำกัด (Finite Impulse Response [FIR] Filter)
- 2 ตัวกรองผลตอบสนองอิมพัลส์อนันต์ (Infinite Impulse Response [IIR] Filter)

4.1 ตัวกรองผลตอบสนองอิมพัลส์จำกัด (Finite Impulse Response [FIR] Filter)

เป็นตัวกรองที่ผลตอบสนองอิมพัลส์ของมันมีลำดับที่มีค่าแตกต่างไปจากศูนย์ มีจำนวนพจน์คงที่ โดยทั่วไปมักจะเป็นตัวกรองแบบไม่มีการป้อนกลับของสัญญาณ ซึ่งตัวกรองที่ไม่มีการป้อนกลับของสัญญาณนี้จะมีข้อดีอยู่ที่ สามารถออกแบบให้มีผลตอบสนองเฟสเป็นเชิงเส้นได้ง่าย และจะเสถียรเสมอ จึงเหมาะที่จะนำไปใช้ในงานที่มีผลตอบสนองเฟสมีความสำคัญมากๆ เช่น ในงานที่ต้องการให้ผลตอบสนองเฟสเป็นสัญญาณเชิงเส้น เพื่อไม่ให้ลักษณะรูปคลื่นของสัญญาณ และเวลาในการเกิดสัญญาณผิดเพี้ยนไปได้ แต่ก็ไม่จำเป็นเสมอไปที่ตัวกรองที่มีผลตอบสนองอิมพัลส์แบบ FIR นี้จะต้องเป็นตัวกรองแบบไม่มีการป้อนกลับเชิงเลข เพราะได้มีนักวิจัยพิสูจน์มาแล้วว่าอาจสร้างโดยใช้ตัวกรองแบบป้อนกลับได้



รูปที่ 4.1 แสดงลักษณะสัญญาณของผลตอบสนองอิมพัลส์จำกัด

การออกแบบตัวกรองชนิดนี้มีหลายวิธีด้วยกัน เช่น การออกแบบโดยใช้อนุกรมฟูริเยอร์, การออกแบบโดยการสุ่มความถี่ (Frequency sampling method) และการออกแบบโดยใช้วิธีวินโดว์แบบต่างๆ เช่น วินโดว์สี่เหลี่ยม, วินโดว์แบบฮาน, แฮมมิง และ แบล็กแมน หรือ ใช้โคเชอร์วินโดว์ โดยจะขอกกล่าวถึงวิธีออกแบบแต่ละวิธีดังนี้

4.1.1 การออกแบบตัวกรองแบบไม่ป้อนกลับโดยใช้อนุกรมฟูริเยอร์

ตัวกรองแบบไม่ป้อนกลับนี้สามารถเขียนได้โดยสมการของอนุกรมฟูริเยอร์ (Fourier series) ซึ่งในการคำนวณจะเห็นว่าความยาวของตัวกรองนั้นมีจำกัด ถ้าให้ $x(n)$ เป็นลำดับสัญญาณเข้า และ $y(n)$ เป็นลำดับสัญญาณออก เราสามารถเขียนสมการผลต่างสืบเนื่องได้ดังนี้

$$\begin{aligned} y(n) &= \sum_{k=-M}^M C_k x(n-k) \\ &= C_0 x(n) + \sum_{k=-1}^{-M} C_k x(n-k) + \sum_{k=1}^M C_k x(n-k) \\ &= C_0 x(n) + \sum_{k=1}^M \{C_{-k} x(n+k) + C_k x(n-k)\} \end{aligned} \quad \dots(4.1)$$

สมมติให้ $T=1$ แล้วทำการแปลงเซททรานสฟอร์ม เพื่อเขียนฟังก์ชันถ่ายโอนได้เป็น

$$H(z) = C_0 + \sum_{k=1}^M \{C_{-k} z^k + C_k z^{-k}\} = \sum_{k=-M}^M C_k z^{-k}$$

หรือ

$$H(\omega) = C_0 + \sum_{k=1}^M \{C_{-k} e^{j\omega k} + C_k e^{-j\omega k}\} = \sum_{k=-M}^M C_k e^{-j\omega k}$$

ถ้าพิจารณากรณีตัวกรองสมมาตรคู่ หรือ $C_k = C_{-k}$

$$H(\omega) = C_0 + \sum_{k=1}^M C_k \{e^{j\omega k} + e^{-j\omega k}\} \quad \dots(4.2a)$$

$$= C_0 + \sum_{k=1}^M 2C_k \{e^{j\omega k} + e^{-j\omega k}\} / 2 \quad \dots(4.2b)$$

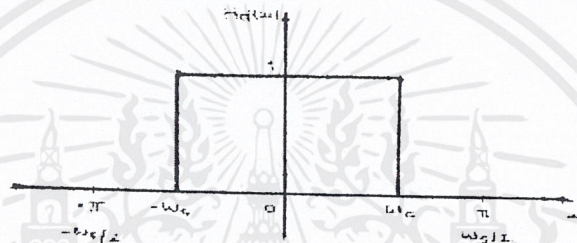
$$H(\omega) = a_0 + \sum_{k=1}^M a_k \cos \omega k$$

โดยกำหนดให้ $a_0 = c_0$ และ $a_k = c_k$ และ $k > 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จากสมการจะเห็นได้ว่า ตัวกรองไม่ป้อนกลับเชิงเลขคณิตสมมาตรคู่ นั้นสามารถเขียนอธิบายได้ โดยใช้ อนุกรมฟูรีเยอร์ที่ประกอบด้วยค่าสัมประสิทธิ์เฉพาะพจน์โคไซน์เท่านั้น และค่าสัมประสิทธิ์ของอนุกรมฟูรีเยอร์ α_k เป็นค่าสัมประสิทธิ์ของตัวกรองด้วย ในทำนองเดียวกัน เราอาจแสดงได้ว่าสำหรับตัวกรองไม่ป้อนกลับเชิงเลขแบบสมมาตรคู่ สามารถเขียนแทนด้วยอนุกรมฟูรีเยอร์ที่มีเฉพาะพจน์โคไซน์เท่านั้น และค่าสัมประสิทธิ์จะเป็นจำนวนจินตภาพเสมอ

โดยปกติแล้วการออกแบบตัวกรองสัญญาณนั้น กระทำโดยทำการประมาณค่าให้ตัวกรองที่ออกแบบมีผลตอบสนองความถี่ใกล้เคียงกับผลตอบสนองอุดมคติ ดังรูปที่ 4.1 ให้มากที่สุดเท่าที่จะทำได้



รูปที่ 4.2 ผลตอบสนองอุดมคติ

จากรูป ω_c เป็นค่าความถี่ตัด (Cut off frequency) จากผลตอบสนองอุดมคติ $H_d(\omega)$ สามารถกระจายให้อยู่ในพจน์ของ โดเมนเวลา (Time Domain) โดยประยุกต์การตัดแปลงฟูรีเยอร์ได้ดังนี้

$$H_d(\omega) = \sum_{-\infty}^{\infty} h_d(k) e^{-jk\omega} \quad \dots(4.3)$$

โดยในกรณีนี้ $h_d(k)$ หาได้จาก

$$h_d(k) = (1/2\pi) \int_{-\omega_c/2}^{\omega_c/2} H_d(\omega) e^{jk\omega} d\omega \quad \dots(4.4)$$

จากสมการด้านบนจึงได้ว่า ผลตอบสนองความถี่ของตัวกรองไม่ป้อนกลับเชิงเลขมีค่าเป็น

$$H_d(\omega) = \sum_{k=-M}^M C_k e^{-jk\omega} \quad \dots(4.5)$$

เมื่อเปรียบเทียบสมการกันแล้วจะเห็นว่า ผลตอบสนองความถี่ของตัวกรองจะใกล้เคียงกับผลตอบสนองอุดมคติก็ต่อเมื่อ M มีค่าเป็นอนันต์ แต่ในทางปฏิบัติเป็นไปได้ยาก จึงมีการตัดปลายอนุกรมฟูรีเยอร์ให้เหลือเท่าที่จำเป็นและจัดให้

$$C_k = h_d(k) \quad \dots(4.6)$$

โดยที่ $k = -M, \dots, 0, \dots, M$ ส่วนความยาวของตัวกรอง $N = (2M+1) = (n-1)/2$

จากผลตอบสนองแอมพลิจูด จะพบว่า ถ้า M หรือ N มีค่ามากจะมีผลทำให้

- 1.) ผลตอบสนองแอมพลิจูดใกล้เคียงผลตอบสนองอุดมคติมากขึ้น
- 2.) จำนวนลูกคลื่น (ripple) มากขึ้น โดยที่คาบของลูกคลื่นลดลง
- 3.) ผลตอบสนองแอมพลิจูดมีความคมมากขึ้น หรือบริเวณความถี่ที่ผลตอบสนองเปลี่ยนจากค่าหนึ่งเป็นค่าใกล้เคียงศูนย์ หรือที่เราเรียกว่า แถบเปลี่ยนสถานะ (Transition band) แคบลง

โดยทั่วไปในการออกแบบตัวกรองไม่ป้อนกลับเชิงเลข เราต้องการลดจำนวนลูกคลื่นให้น้อยที่สุด และให้มีช่วงแถบเปลี่ยนค่ากว้างขึ้น ดังนั้นในการออกแบบจึงจำเป็นต้องเลือกอย่างใดอย่างหนึ่งสำหรับการออกแบบตัวกรองไม่ป้อนกลับเชิงเลข โดยวิธีเขียนอนุกรมฟูรีเยอร์ อาจสรุปเป็นขั้นตอนการทำได้ดังนี้คือ

- 1.) กำหนดผลตอบสนองอุดมคติที่ต้องการ โดยกำหนดความถี่คัทออฟ
- 2.) ทำการอินทิเกรตผลตอบสนองอุดมคติ เมื่อหาลำดับผลตอบสนองอิมพัลส์ $h_d(k)$
- 3.) เลือกให้สัมประสิทธิ์ $C_k = h_d(k)$

4.1.2 การออกแบบตัวกรองแบบไม่ป้อนกลับโดยใช้วิธีหน้าต่าง (Window)

ในวิธีนี้จะขอเริ่มพิจารณาจากสมการพื้นฐานก่อน โดยเริ่มจาก $H_d(\omega)$ แทนผลตอบสนองความถี่อุดมคติ และผลตอบสนองนี้สามารถเขียนแทนด้วยอนุกรมฟูรีเยอร์ (Fourier series) ยาวอนันต์พจน์ ซึ่งมีสมการคือ

$$H_d(\omega) = \sum_{n=-\infty}^{\infty} h_d(n) e^{-jk\omega T} \quad \dots(4.7)$$

โดยที่ $h_d(n)$ หาได้จาก

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{jn\omega T} d\omega \quad \dots(4.8a)$$

โดยที่ค่า ω_s นี้เป็นค่าความถี่ในการสุ่มตัวอย่าง ในการออกแบบตัวกรองไม่ป้อนกลับเชิงเลขที่ผ่านมา เราพบว่าเกิดปัญหาขึ้นสองประการด้วยกันคือ

ประการแรก เราต้องทำการตัดปลายของอนุกรม ให้มีจำนวนพจน์จำกัด หรือ

$$H_d(\omega) = \sum_{n=-(N-1)/2}^{(N-1)/2} h_d(n) e^{-jn\omega T} \quad \dots(4.9)$$

ซึ่งผลลัพธ์ก็คือ ทำให้เกิดลูกคลื่นบนผลตอบสนองแอมพลิจูด

ประการที่สอง เมื่อตัดปลายของอนุกรมแล้ว ยังไม่สามารถนำไปใช้งานได้ เพราะผลตอบสนองอิมพัลส์ (Impulse Response) เริ่มจากลำดับที่ $n = -(N-1)/2$ ซึ่งในระบบเวลาจริง ไม่สามารถสร้างลำดับที่เวลาเป็นลบได้ เราจึงทำการแก้ไขโดยการหวนวงผลตอบสนองอิมพัลส์ออกไปโดยเริ่มจากค่า $n = 0$ หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H_d(\omega) = \sum_{n=0}^{(N-1)/2} h_d(n) e^{-jn\omega T} \quad \dots(4.10)$$

จะเห็นได้ว่าการตัดปลายอนุกรมฟูรีเยอร์นั้น ก็คือการที่เราเลือกให้

$$h(n) = h_d(n) \quad \dots(4.11)$$

ซึ่งสมการนี้ ถ้าหากเขียนให้อยู่ในรูปสมการทั่วไปก็คือ การนำเอาลำดับ มาคูณกับค่าลำดับจำกัด หรือ

$$h(n) = h_d(n)w(n) \quad \dots(4.12)$$

โดยที่ $w(n)$ แทนลำดับของวินโดว์แบบต่างๆ สำหรับสมการที่ผ่านมานั้น เป็นแบบวินโดว์สี่เหลี่ยม (Rectangular window) ที่ลำดับ มีคุณสมบัตินิยามได้เป็น

$$\begin{aligned} w_R(n) &= 1 && ; 0 \leq n \leq N-1 \\ &= 0 && ; n \text{ กรณีอื่น} \end{aligned} \quad \dots(4.13)$$

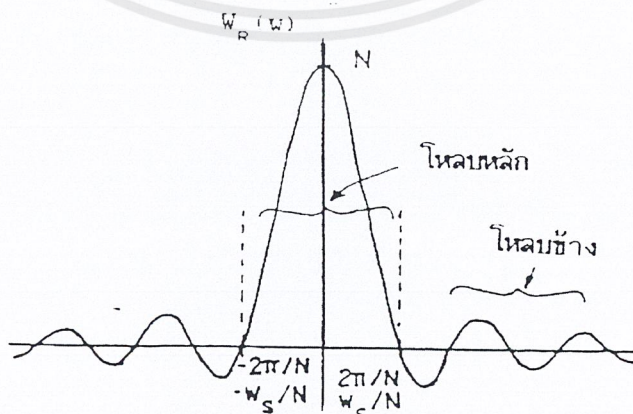
4.1.2.1 วินโดว์สี่เหลี่ยม (Rectangular window)

อาจนิยามโดยทั่วไปได้ดังนี้

$$\begin{aligned} w_R(n) &= 1 && ; |n| \leq (N-1)/2 \\ &= 0 && ; \text{otherwise} \end{aligned} \quad \dots(4.14)$$

ผลตอบสนองความถี่ของวินโดว์แบบนี้หาได้โดยการประยุกต์ใช้ผลการแปลงแซคเข้ากับสมการ แล้วทำการแทนค่าให้ $z = e^{j\omega T}$ เมื่อจัดรูปสมการใหม่จะได้

$$\begin{aligned} W_R(\omega) &= \sum_{n=-(N-1)/2}^{(N-1)/2} e^{-jn\omega T} \\ &= \frac{\{e^{j\omega(N-1)T/2} - e^{-j\omega(N+1)T/2}\}}{\{1 - e^{-j\omega T}\}} \\ &= \frac{\{e^{j\omega NT/2} - e^{-j\omega NT/2}\}}{\{e^{j\omega T/2} - e^{-j\omega T/2}\}} \\ &= \frac{\sin(\omega NT/2)}{\sin(\omega T/2)} \end{aligned}$$



รูปที่ 4.3 ผลตอบสนองแอมพลิจูดของวินโดว์สี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.3 แสดงผลตอบสนองแอมพลิจูดของวินโดว์สี่เหลี่ยมซึ่งจะพบว่ามีคุณสมบัติดังนี้

- 1.) ความกว้างของ Main Lobe มีค่า $4\pi / N$ และมีค่าขนาดเป็น N
- 2.) เมื่อ N มีค่ามากขึ้นความกว้างของ Main lobe ลดลง
- 3.) Side lobe มีการแกว่งโดยผ่านจุดศูนย์ทุกๆค่าความถี่ และมีขนาดลดลงไปเรื่อยๆและ $W_r(\omega) = 0$ ณ ค่าความถี่ $\omega = m \omega_c / N$ เมื่อ m เป็นค่าบวกและลบของ $1, 2, \dots$ ผลนี้เป็นที่มาว่าถ้า N มีค่ามาก โหลบข้างจะเข้ามารวมกันใกล้ความถี่ $\omega = 0$ มากขึ้น
4. ถ้านิยามให้พจน์ อัตราลู่คลื่น (Ripple Ratio) หรือ RR เป็น

$$RR = \frac{\text{ค่าขนาดสูงสุดของโหลบข้าง}}{\text{ค่าขนาดสูงสุดของโหลบหลัก}} \times 100 \quad \dots(4.15)$$

การที่เรา นำเอาลำดับผลตอบสนองอิมพัลส์ (Impulse Response) มาคูณกับลำดับของวินโดว์ (Window) ซึ่งเป็นการปฏิบัติกรใน Time Domain เมื่อพิจารณาใน Frequency domain ก็เท่ากับเป็นการนำเอาผลตอบสนองความถี่อุดมคติมาทำการคูณประสาน กับ ผลตอบสนองความถี่ของวินโดว์หรือ

$$H(\omega) = (1/2\pi) \int_{-\pi}^{\pi} H_d e^{j\theta} \cdot W_R e^{j(\omega-\theta)} d\theta \quad \dots(4.16)$$

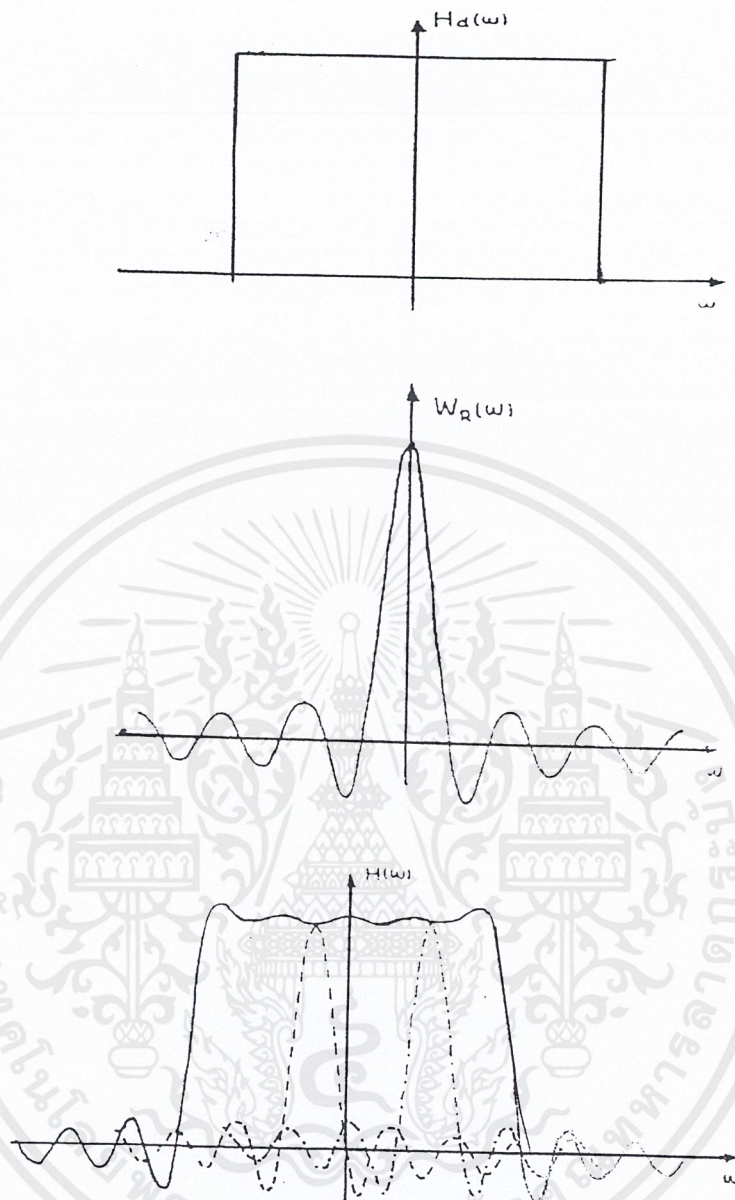
ผลการประสานนี้จะอธิบายได้ดังรูป ซึ่งจะเห็นได้ชัดว่า

1. ค่าความชันของโหลบหลักของเสาปัตยกรรมของวินโดว์ ทำให้เกิดแถบเปลี่ยนสถานะของผลตอบสนองความถี่ เพราะฉะนั้นแล้ว ถ้า N มากก็จะทำให้แถบเปลี่ยนสถานะแคบลง
2. ลู่คลื่นบนผลตอบสนองความถี่ เป็นผลมาจาก โหลบข้างของเสากรรมวินโดว์ ดังนั้น ถ้าเลือก N ค่ามาก โหลบข้างเลื่อนเข้าใกล้ความถี่ $\omega = 0$ มากขึ้น เป็นผลทำให้เกิดลู่คลื่นบนยอดของผลตอบสนองความถี่มากขึ้น

ด้วยผลทั้งสองประการนี้ เราจึงต้องหาวินโดว์ หรือดัดแปลงวินโดว์สี่เหลี่ยมเพื่อให้มีคุณสมบัติคือ งาน ของสัญญาณทั้งหมดไว้ให้ได้มากที่สุด

1. ให้โหลบหลักของผลตอบสนองความถี่แคบ แต่ภายในโหลบหลักสามารถครอบคลุมพลังงานของสัญญาณทั้งหมดไว้ให้ได้มากที่สุด

2. ให้ผลตอบสนองความถี่ของวินโดว์มีค่าขนาดของโหลบข้างน้อย และมีขนาดลดลงอย่างรวดเร็วเมื่อความถี่สูงขึ้น



รูปที่ 4.4 แสดงวิธีปฏิบัติการทำวินโดว์ในโดเมนความถี่

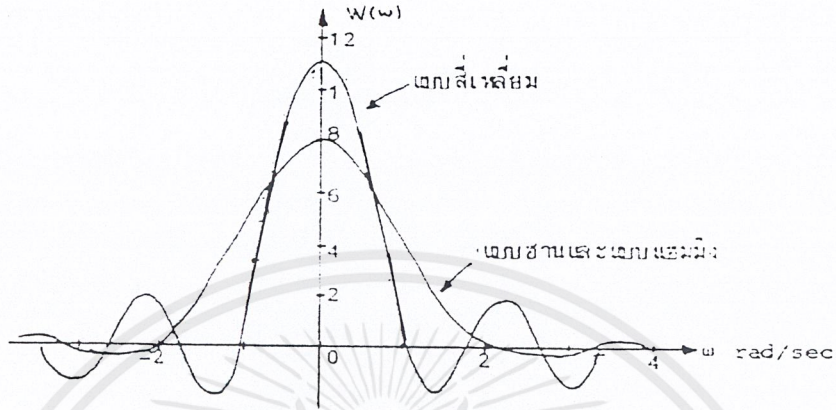
4.1.2.2 วินโดว์แบบ ฮาน และ แฮมมิง (Hann and Hamming window)

วินโดว์แบบฮาน (Hann window) และวินโดว์แบบแฮมมิง (Hamming window) สามารถเขียนเป็นสมการของลำดับได้ โดยให้ $T = 1$ วินาที คือ

$$\begin{aligned}
 W_H(n) &= \alpha + \{(1 - \alpha) \cos(2\pi n / N - 1)\} & ; |n| \leq (N - 1) / 2 \\
 &= 0 & ; \text{otherwise} \quad \dots(4.17)
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่แบบฮานวินโดว์ และแบบแฮมมิงวินโดว์ ดังในรูป ซึ่งแสดงการเปรียบเทียบกันระหว่างความถี่ของวินโดว์สามแบบด้วยกันคือ วินโดว์แบบสี่เหลี่ยม , วินโดว์แบบฮาน ,และวินโดว์แบบแฮมมิง ซึ่งจะเห็นได้ชัดเจนว่า วินโดว์แบบสี่เหลี่ยมนี้จะมีพลังงานนอกโหลบลึกอยู่มาก ส่วนวินโดว์แบบฮาน และ แฮมมิง พลังงานได้ถ่ายเข้าไปในโหลบลึกได้มากเกือบ 99 เปอร์เซ็นต์



รูปที่ 4.5 สเปกตรัมของวินโดว์แบบสี่เหลี่ยม,แบบฮานและแบบแฮมมิง กรณี N=11

4.1.2.3 วินโดว์แบบแบล็กแมน

$$W_m(n) = 0.42 + 0.5 \cos\left(\frac{2n\pi}{N-1}\right) + 0.08 \cos\left(\frac{4n\pi}{N-1}\right)$$

$$= 0 \quad ; \text{ otherwise} \quad \dots(4.18)$$

วินโดว์แบบนี้ให้คุณสมบัติที่ดีกว่าแบบฮานและ แฮมมิง คือ สามารถถ่ายทอดพลังงานเข้าไปในโหลบลึกได้มากกว่า ดังได้แสดงคุณสมบัติเพื่อเป็นการเปรียบเทียบไว้ในตารางที่ 4.1

ชนิดของวินโดว์	Main lobe width	Ripple Ratio % N=31	Peak-Amp of Side lobe	Min-stop band Att(dB)
Rectangular	$2\omega_c / N$	21.80	-13	-21
Hann	$4\omega_c / N$	2.67	-31	-44
Hamming	$4\omega_c / N$	0.82	-41	-53
Blackman	$6\omega_c / N$	0.12	-57	-74

ตารางที่ 4.1 ตารางแสดงคุณสมบัติของวินโดว์แบบต่างๆ

Windows	Ripple Ratio		
	N=11	N=21	N=31
Rectangular	22.34	21.89	21.8
Hann	2.62	2.67	2.67
Hamming	1.47	0.93	0.82
Blackman	0.08	0.12	0.12

ตารางที่ 4.2 ตารางแสดงคุณสมบัติของวินโดว์แบบต่างๆเมื่อมีอันดับต่างกัน

4.1.3 การออกแบบวงจรกรองผลการตอบสนองอิมพัลส์จำกัดที่ใช้ในการทดลอง

วิธีการออกแบบวงจรกรองความถี่ต่ำผ่าน โดยวงจรกรองความถี่ต่ำผ่านจะมีลักษณะดังสมการ

$$H(e^{j\omega}) = 1 \quad \begin{cases} |\omega| \leq \omega_c \\ \omega_c \leq |\omega| \leq \omega_s \end{cases} \quad \dots(4.19)$$

การหาความสัมพันธ์ของผลการตอบสนองความถี่ของวงจรกรองกับ ผลตอบสนองอิมพัลส์ของวงจรทำได้โดยการอาศัยการกระจายของอนุกรมฟูรีเยอร์ คือ

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(nT)e^{-j\omega nT} \quad \dots(4.20)$$

และ

$$h(nT) = (1/\omega_s) \int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega})e^{j\omega nT} d\omega \quad \dots(4.21)$$

เมื่อแทนค่าจากสมการผลตอบสนองความถี่อิดูคมคดลงในสมการ จะได้

$$\begin{aligned} h(nT) &= (1/\omega_s) \int_{-\omega_s/2}^{\omega_s/2} e^{j\omega nT} d\omega \quad \dots(4.22) \\ &= \left(\frac{1}{n\pi}\right) \{\sin(n\omega_c)\} \\ &= (\omega_c/\pi) \end{aligned}$$

สมการนี้จึงเป็นสมการผลตอบสนองอิมพัลส์ของวงจรกรองความถี่ต่ำผ่าน

วงจรกรองความถี่สูงผ่านนั้นก็คิดคำนวณมาจากสมการของวงจรกรองความถี่ต่ำผ่านนั่นเองโดยสมการผลตอบสนองอิมพัลส์ของวงจรกรองความถี่สูงผ่านคือ

$$h(nT) = (1/\omega_s) \left\{ \int_{-\pi}^{-\omega_c} e^{j\omega nT} d\omega + \int_{\omega_c}^{\pi} e^{j\omega nT} d\omega \right\} \quad \dots(4.23)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= -\left(\frac{1}{n\pi}\right)\{\sin(n\omega_c)\} \quad ; n \neq$$

$$= 1 - (\omega_c / \pi) \quad ; n =$$

สมการนี้จึงเป็นสมการผลตอบสนองอิมพัลส์ของวงจรกรองความถี่สูงผ่าน
 วงจรกรองแถบความถี่ผ่านนั้นคำนวณมาจากวงจรกรองความถี่ต่ำสองวงจรถือความถี่คัทออฟต่าง
 กันโดยจะมีผลตอบสนองอิมพัลส์ดังนี้

$$h(n) = \left\{ \left(1/\omega_s\right) \int_{\omega_{c1}}^{\omega_{c2}} e^{jn\omega} d\omega \right\} + \left\{ \left(1/\omega_s\right) \int_{-\omega_{c2}}^{-\omega_{c1}} e^{jn\omega} d\omega \right\} \quad \dots(4.24)$$

$$= \left(\frac{1}{n\pi}\right)\{\sin(n\omega_{c1})\} - \left(\frac{1}{n\pi}\right)\{\sin(n\omega_{c2})\} \quad ; n \neq$$

$$= (\omega_{c2} / \pi) - (\omega_{c1} / \pi) \quad ; n =$$

วงจรกรองลดทอนช่วงความถี่นั้นคำนวณมาจากวงจรกรองความถี่ช่วงผ่าน โดยจะมีผลการตอบ
 สอนอิมพัลส์ดังนี้

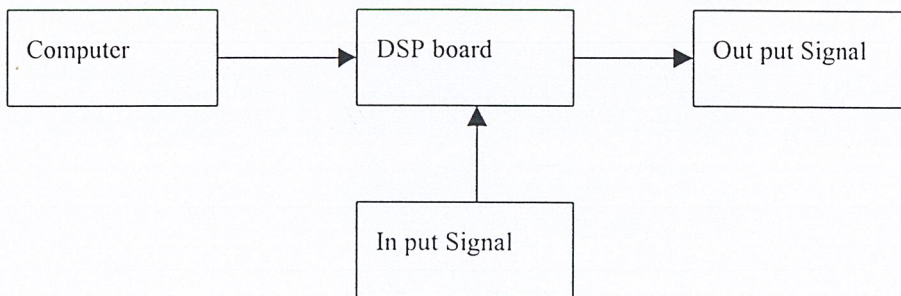
$$h(n) = \left(1/\omega_s\right) \left\{ \int_{-\pi}^{-\omega_{c2}} e^{jn\omega} d\omega + \int_{-\omega_{c1}}^{\omega_{c1}} e^{jn\omega} d\omega + \int_{\omega_{c2}}^{\pi} e^{jn\omega} d\omega \right\} \quad \dots(4.25)$$

$$= \left(\frac{1}{n\pi}\right)\{\sin(n\omega_{c1})\} - \left(\frac{1}{n\pi}\right)\{\sin(n\omega_{c2})\} \quad ; n \neq$$

$$= 1 - \{(\omega_{c2} / \pi) - (\omega_{c1} / \pi)\} \quad ; n =$$

4.2 การออกแบบ

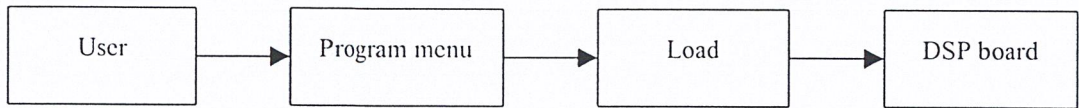
ในโครงการนี้เราทำ ตัวกรองเชิงเลข โดยใช้ Computer ในการควบคุม DSP board ผ่านทางพอร์
 ทาน โดยชนิดตัวกรองความถี่ที่เลือกใช้เราจะใช้ตัวกรองชนิด FIR (Finite Impulse Response) เพราะ
 สะดวกในการเปลี่ยนแปลงค่าสัมประสิทธิ์ ในโปรแกรมตัวกรองความถี่ทำได้ง่ายและสะดวกกว่า IIR
 (Ininite Impulse Response)



รูปที่ 4.6 แสดงบล็อกไดอะแกรมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

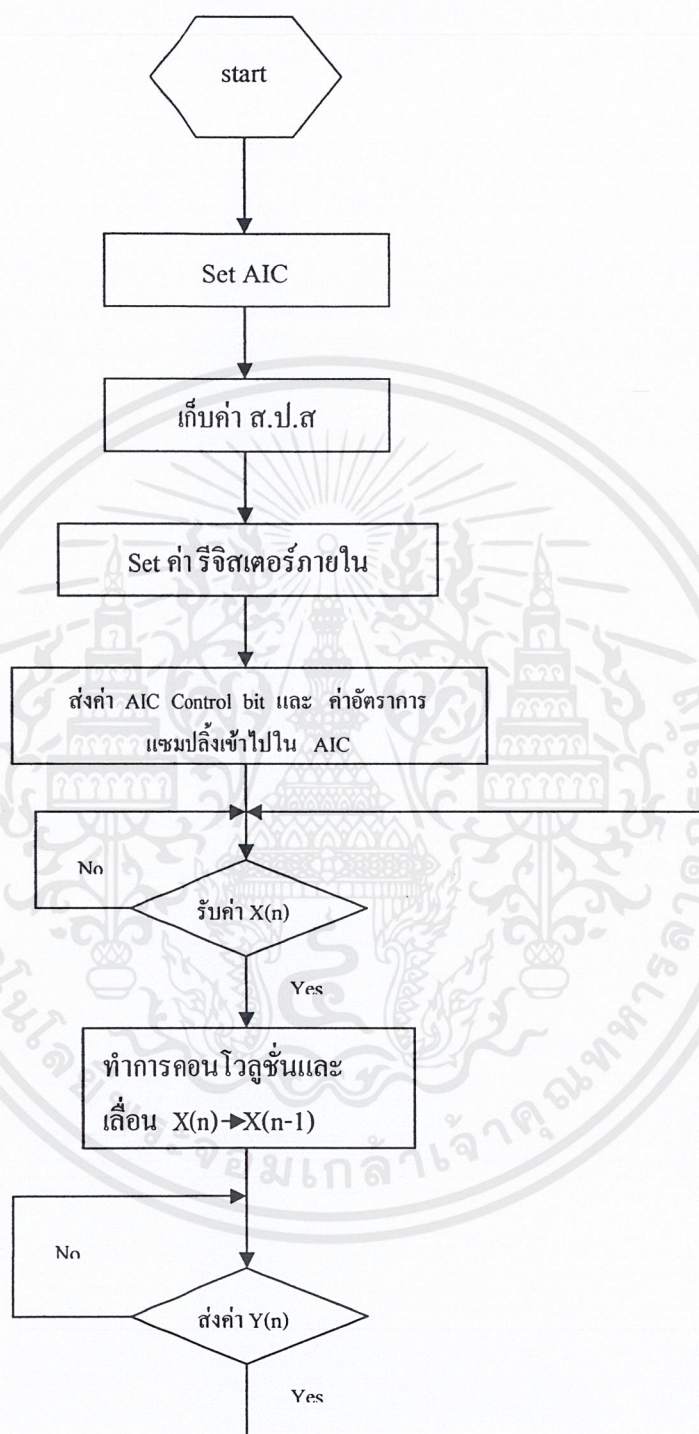
ส่วนโปรแกรมที่ใช้ในการควบคุมจากคอมพิวเตอร์นั้น จะทำงานโดยรับข้อมูลจากผู้ใช้ว่าจะเลือกชนิด,ค่าความถี่ผ่าน และจำนวนจุดข้อมูล แล้วโปรแกรมจะทำการเลือก ข้อมูล เพื่อ นำไปควบคุม DSP board



รูปที่ 4.7 แสดงการทำงานของโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 Flow chat แสดงการทำงานของ FIR Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

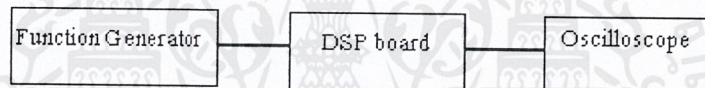
5.1 วิธีการทดลอง

ขั้นแรกทำการป้อนสัญญาณรูปคลื่นไซน์ขนาด $1 V_{p-p}$ ที่ความถี่ต่างๆที่ต้องการ วัดค่าเข้าที่ output ของ TMS320C31 ซึ่ง input ต่อเข้ากับ ออสซิโลสโคป เพื่อวัดค่าและรูปคลื่นของสัญญาณที่ออกมา โดยเปลี่ยนโปรแกรมที่ต้องการทดลอง บันทึกค่าที่วัดได้ นำค่าที่ผ่าน TMS320C31 มาทำการคำนวณโดยใช้สมการ

$$V_t = 20 \log (v/2) \tag{1}$$

โดย V_t คืออัตราการลดทอน (ค่าสูงสุดเท่ากับ 0 dB)

v คือค่าที่วัดได้จาก ออสซิโลสโคป



รูปที่ 5.1 บล็อกไดอะแกรมแสดงวิธีการทดลอง

5.2 การทดลอง

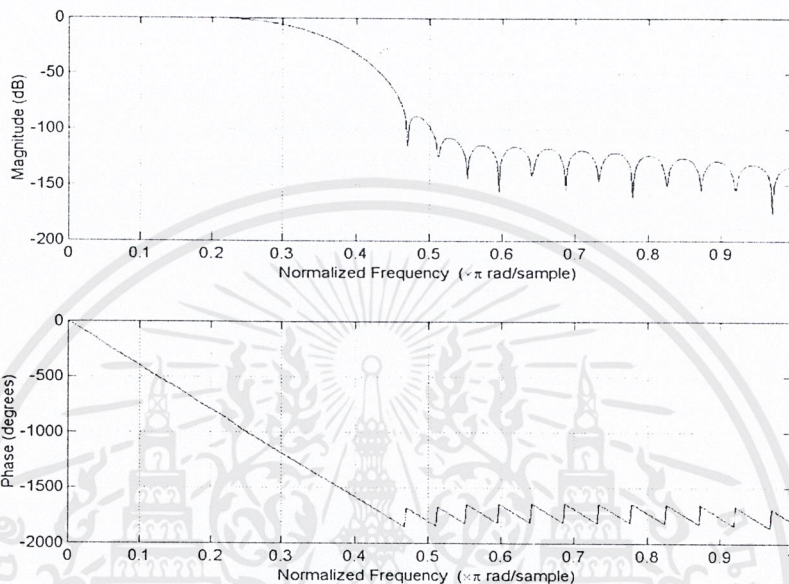
5.2.1 การทดลองที่ 1 ทำการทดลองวงจรองความถี่ผ่านเชิงเลข โดยใช้ค่าความถี่ผ่านเท่ากัน แต่เปลี่ยนชนิดของวงจรองความถี่ผ่านเชิงเลข ในการทดลองใช้ 5 ชนิด คือ วินโดว์แบบแฮมมิง,ฮานนิง,แบล็คแมน,สี่เหลี่ยม และสามเหลี่ยม โดยในการออกแบบค่าสัมประสิทธิ์ใช้โปรแกรม MATLAB คำนวณหาโดยใช้ฟังก์ชันต่างๆดังนี้

วินโดว์	ฟังก์ชัน	รายละเอียด
Blackman window	Blackman(n)	ค่าฟังก์ชันหน้าต่าง Blackman จำนวน n จุดข้อมูล
Rectangular window	Boxcar(n)	ค่าฟังก์ชันหน้าต่าง Boxcar จำนวน n จุดข้อมูล
Hamming window	Hamming(n)	ค่าฟังก์ชันหน้าต่าง Hamming จำนวน n จุดข้อมูล
Triangular window	Triang(n)	ค่าฟังก์ชันหน้าต่าง Triangular จำนวน n จุดข้อมูล
Hanning window	Hanning(n)	ค่าฟังก์ชันหน้าต่าง Hanning จำนวน n จุดข้อมูล

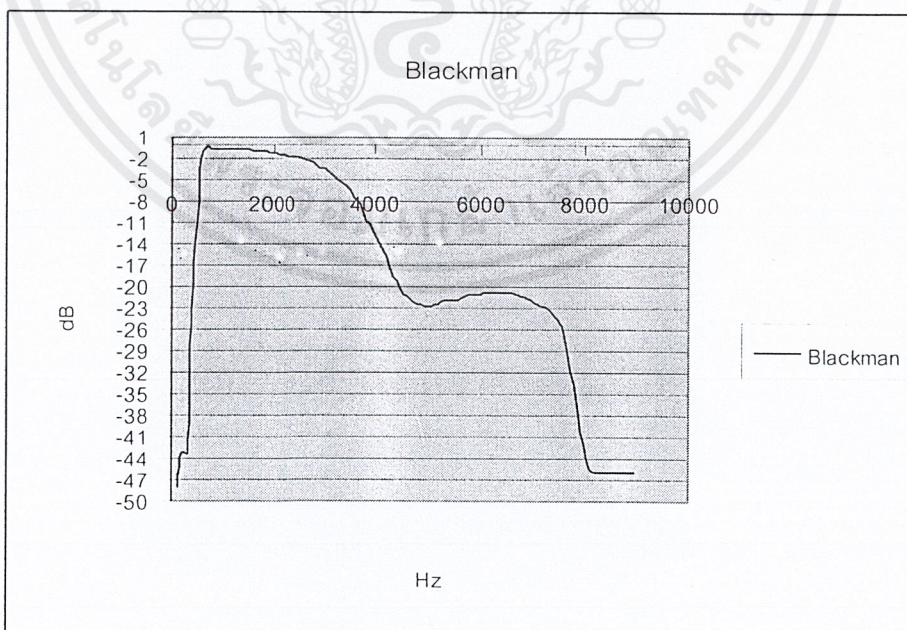
ตารางที่ 5.1 แสดงฟังก์ชันต่างๆ ที่ใช้ในโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการทดลองนี้เราจะใช้วงจรกรองความถี่เชิงเลขแบบความถี่ต่ำผ่าน (digital lowpass filter) ที่ค่าความถี่ผ่านเท่ากับ 0.3 โดยมีค่า Sampling Rate (F_s) เท่ากับ 20 kHz โดยเปลี่ยนชนิดของวินโดว์ และจะมีค่าความถี่สูงสุดประมาณ 8kHz



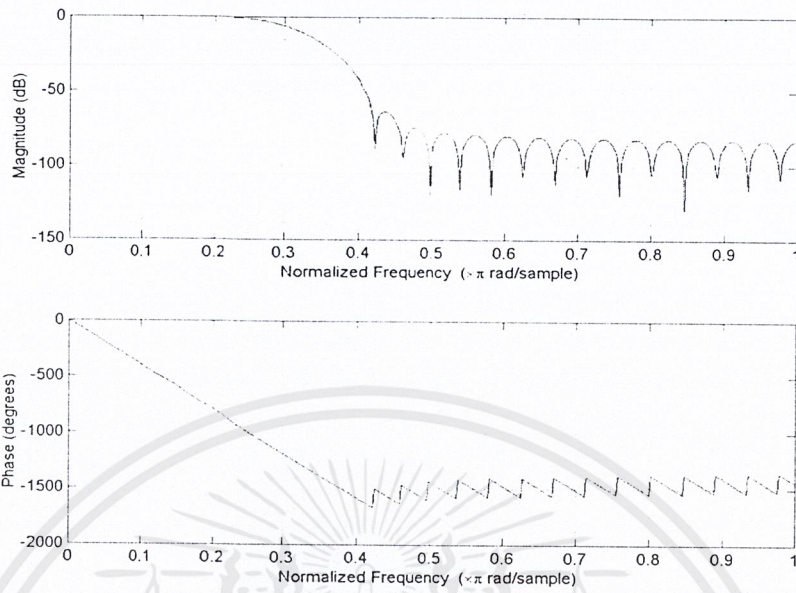
รูปที่ 5.2 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน ที่ 0.3 ชนิด Blackman window ที่ได้จากการออกแบบ



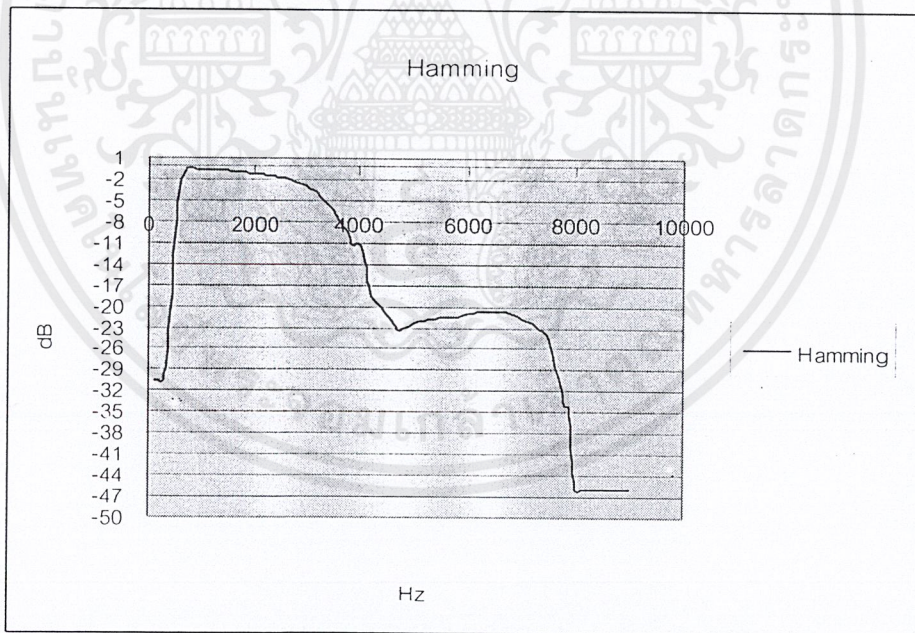
รูปที่ 5.3 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.3 ชนิด Blackman window ที่ได้จากการทดลอง

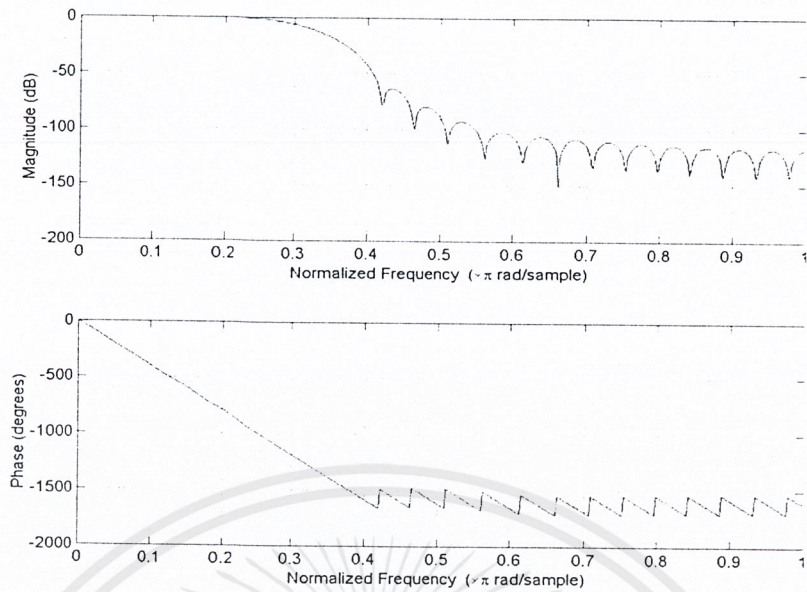


รูปที่ 5.4 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน
ที่ 0.3 ชนิดHamming window ที่ได้จากการออกแบบ

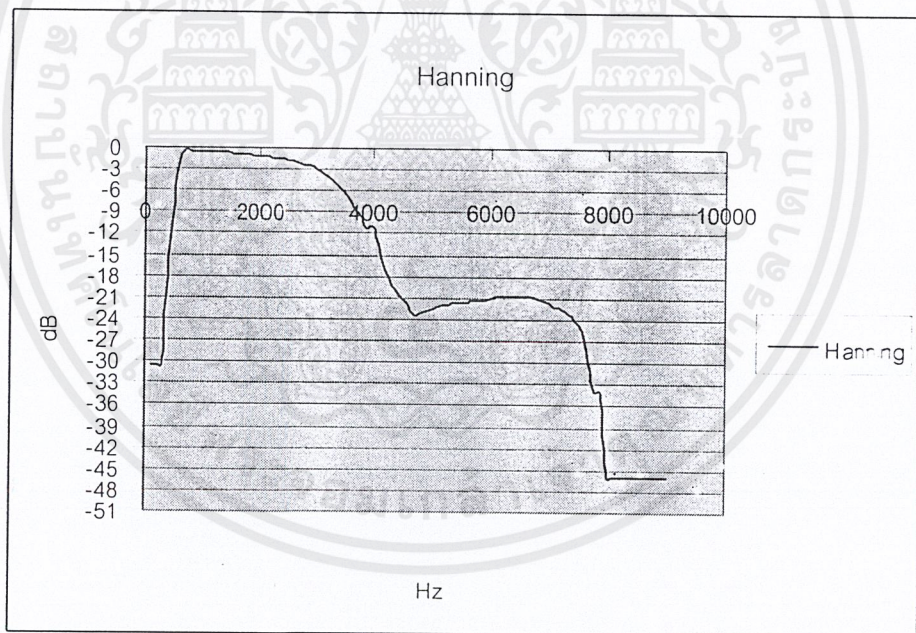


รูปที่ 5.5 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน
ที่ 0.3 ชนิดHamming window ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

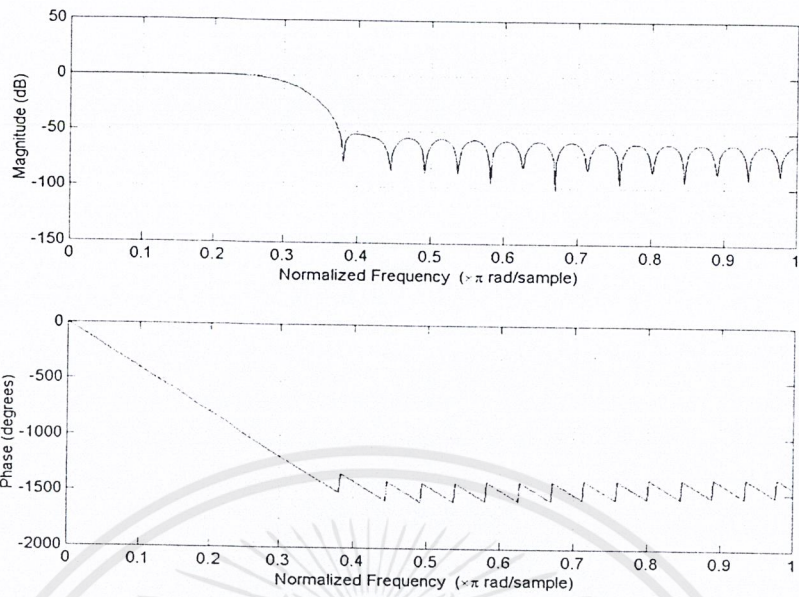


รูปที่ 5.6 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน
ที่ 0.3 ชนิด Hanning window ที่ได้จากการออกแบบ

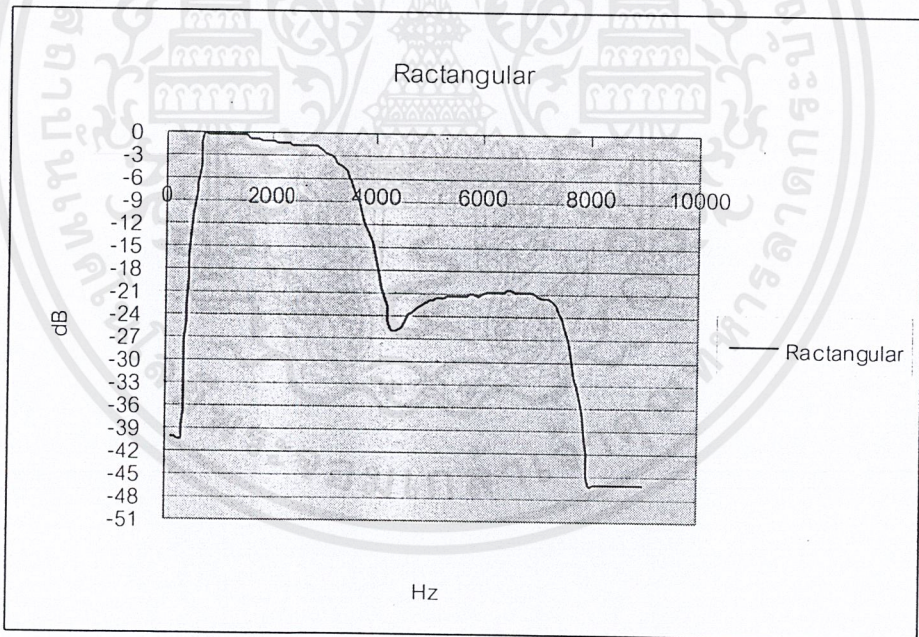


รูปที่ 5.7 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน
ที่ 0.3 ชนิด Hanning window ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

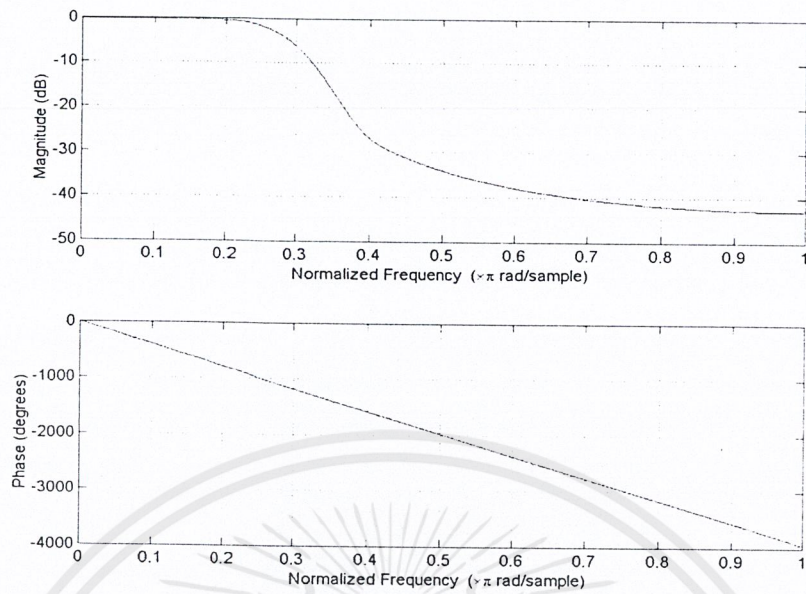


รูปที่ 5.8 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิดความถี่ต่ำผ่าน
ที่ 0.3 หน่วย Rectangular window ที่ได้จากการออกแบบ

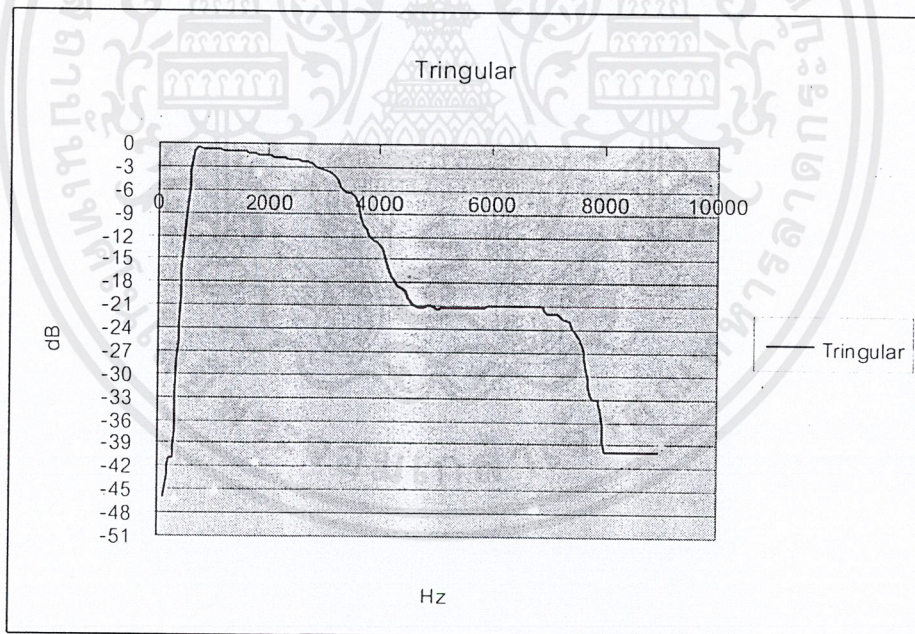


รูปที่ 5.9 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิดความถี่ต่ำผ่าน
ที่ 0.3 หน่วย Rectangular window ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

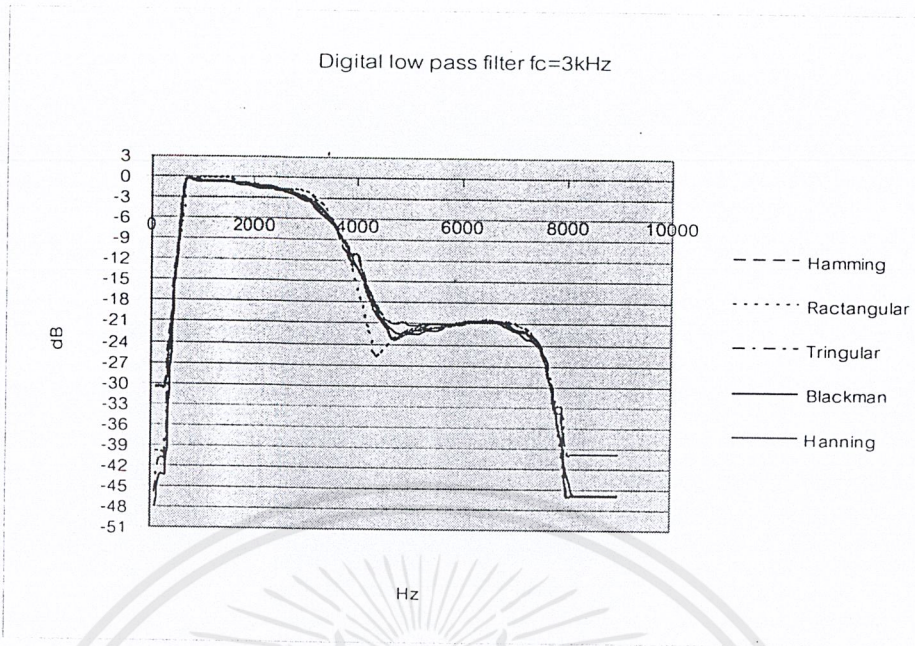


รูปที่ 5.10 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิดความถี่ต่ำผ่าน
ที่ 0.3 หน่วย Triangular window ที่ได้จากการออกแบบ



รูปที่ 5.11 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่าน
ที่ 0.3 หน่วย Triangular window ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 ผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่านที่ 0.3 ที่ วินโดว์แบบต่างๆเป็นค่าที่ได้จากการทดลอง โดยนำมาแสดงร่วมกันเพื่อเปรียบเทียบ

จากการทดลองนี้จะเห็นได้ว่าผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่านที่ 0.3 ที่ วินโดว์แบบต่างๆ ที่ได้จากการทดลอง เมื่อนำมาเปรียบเทียบกันจะมีค่าใกล้เคียงกันมาก แต่จะมีค่าน้อยกว่าที่ออกแบบไว้ในทุกๆวินโดว์ แล้วค่า ripple ซึ่งปรากฏในการออกแบบนั้น ไม่ปรากฏในการทดลองเนื่องจาก ค่าการลดทอนตัวกรองความถี่ที่ได้ในการทดลองมีค่าไม่ถึง -50 dB (จากการออกแบบ ค่า ripple จะปรากฏในช่วงการลดทอน ประมาณต่ำกว่า -50 dB) เนื่องจากที่ระดับของสัญญาณน้อยๆ การวัดขแรงดันทำได้ลำบาก

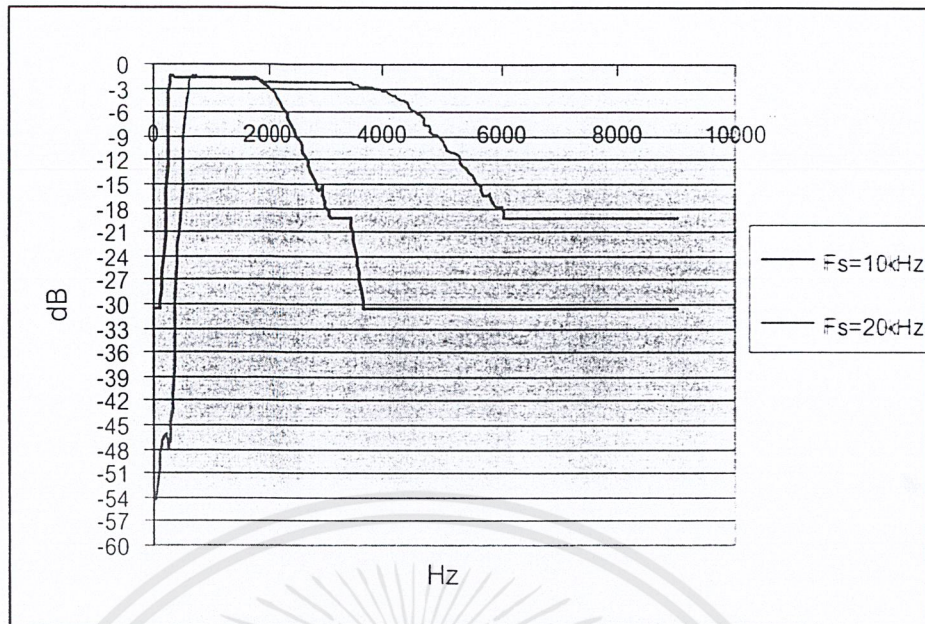
5.2.1 การทดลองที่ 2 เราจะให้ค่าความถี่ผ่านคงที่แต่จะเปลี่ยน ค่าความถี่ในการสุ่ม (Sampling frequency) เป็น 2 ค่าได้แก่ $F_s = 10\text{kHz}$ และ $F_s = 20\text{kHz}$ ที่ค่าความถี่ผ่านเท่ากับ $\omega =$ ถ้าค่าความถี่ในการสุ่มเท่ากับ 10 kHz จะได้ค่าความถี่ผ่านเท่ากับ 2kHz และหากใช้ค่าความถี่ในการสุ่มเท่ากับ 20kHz จะได้ค่าความถี่ผ่านเท่ากับ 4kHz โดยการกำหนดค่า AIC ในโปรแกรมดังนี้

F_s (Hz)	A	B
8,000	0x162C	0x4892
10,000	0x162C	0x3872
16,000	0x0E1C	0x3872
20,000	0x0A14	0x3E7E

ตารางที่ 5.2 แสดงค่าการกำหนดสัมประสิทธิ์ต่างๆของ AIC โดยที่ $F_s = 20,000\text{ kHz}$

จะมีแบนด์วิดท์เท่ากับ 8 kHz

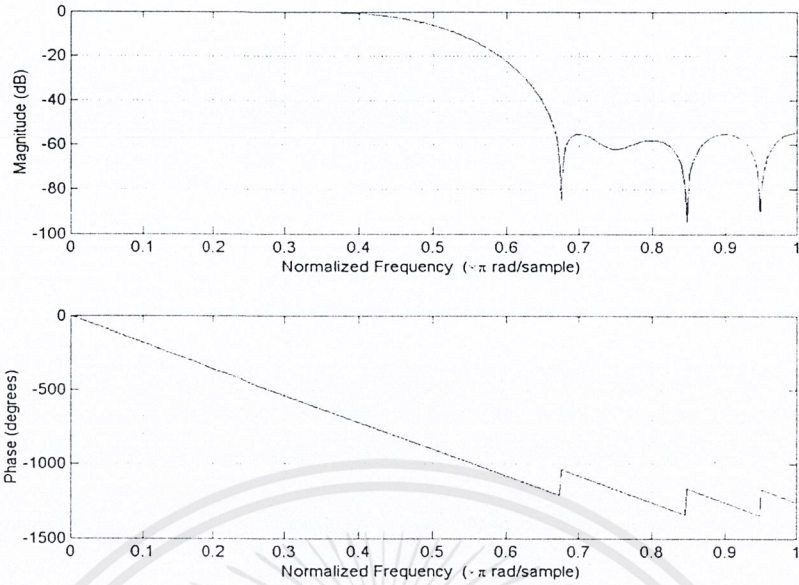
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



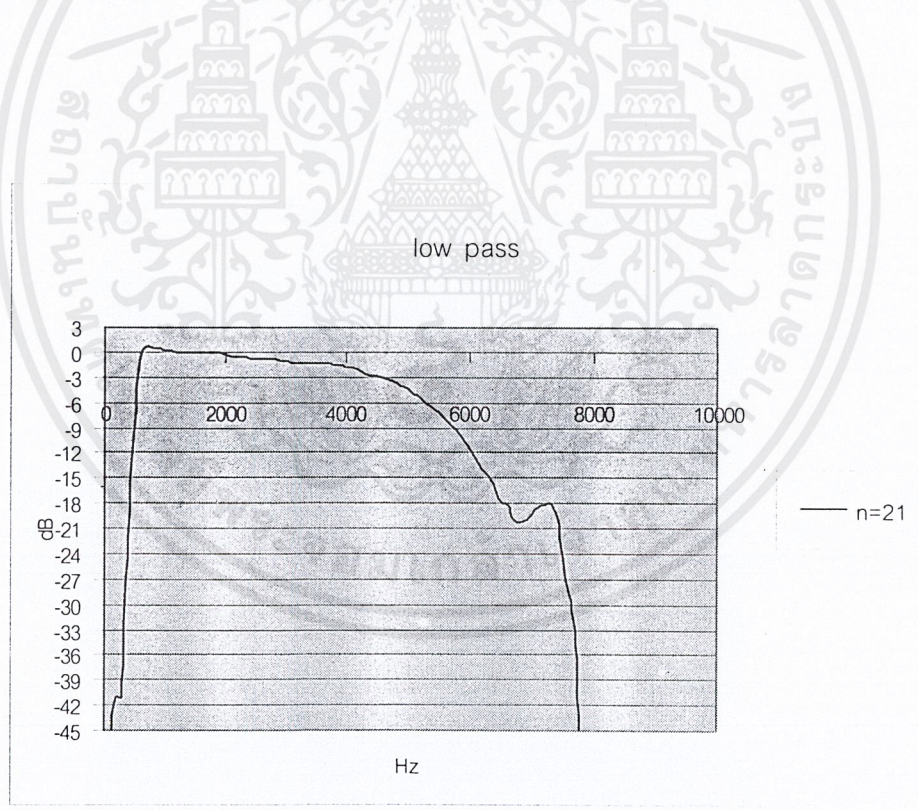
รูปที่ 5.13 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่
ที่ใช้ค่าความถี่ในการสุ่มเท่ากับ 10 kHz และ 20 kHz ตามลำดับ

5.2.3 การทดลองที่ 3 ทำการทดลองโดยเปลี่ยนค่าความถี่ผ่าน , จำนวนข้อมูล (๒) และชนิดของ
ตัวกรองความถี่เพื่อดูคุณสมบัติและนำคุณสมบัติเหล่านั้นมาเปรียบเทียบกับ โดยมีค่าดังนี้ (ซึ่งในการ
ทดลองนี้เราจะเลือกค่าที่นำมาทดสอบเพียงบางส่วน)

1. Low pass filter $n = 21$, $\omega_n = 0.5$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ 5 kHz
2. High pass filter $n = 21$, $\omega_n = 0.5$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ 5 kHz
3. Band pass filter $n = 21$, $\omega_n = 0.2 - 0.3$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ
2 kHz - 3 kHz
4. Band stop filter $n = 21$, $\omega_n = 0.2 - 0.3$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ
2 kHz - 3 kHz
5. Low pass filter $n = 45$, $\omega_n = 0.5$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ 5 kHz
6. High pass filter $n = 45$, $\omega_n = 0.5$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ 5 kHz
7. Band pass filter $n = 45$, $\omega_n = 0.2 - 0.3$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ
2 kHz - 3 kHz
8. Band stop filter $n = 45$, $\omega_n = 0.2 - 0.3$, $F_s = 20 \text{ kHz}$ ซึ่งจะมีค่าความถี่ผ่านเท่ากับ
2 kHz - 3 kHz

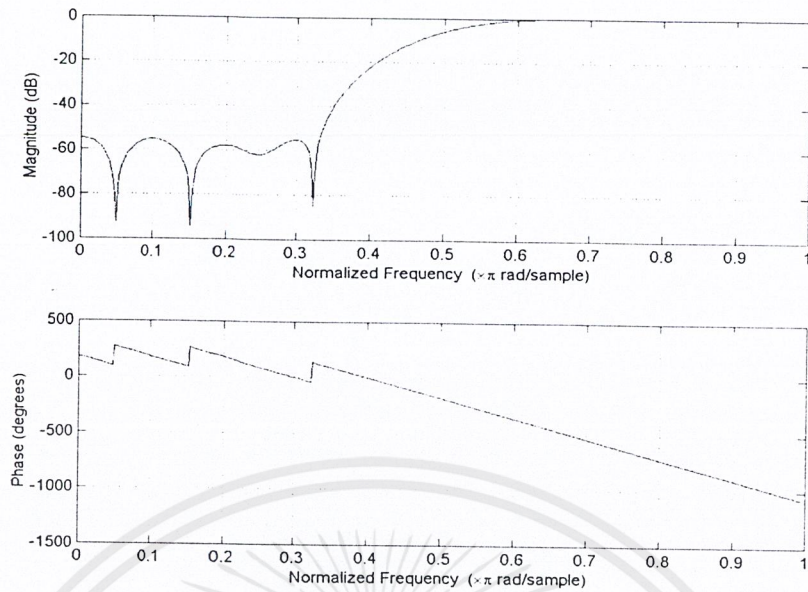


รูปที่ 5.14 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 21 ที่ได้จากโปรแกรม MAT LAB

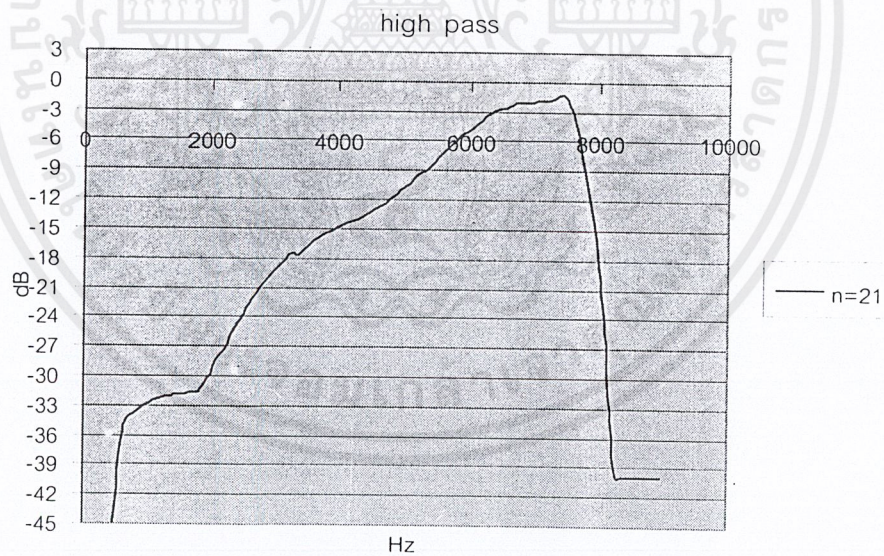


รูปที่ 5.15 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5kHz , n = 21ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

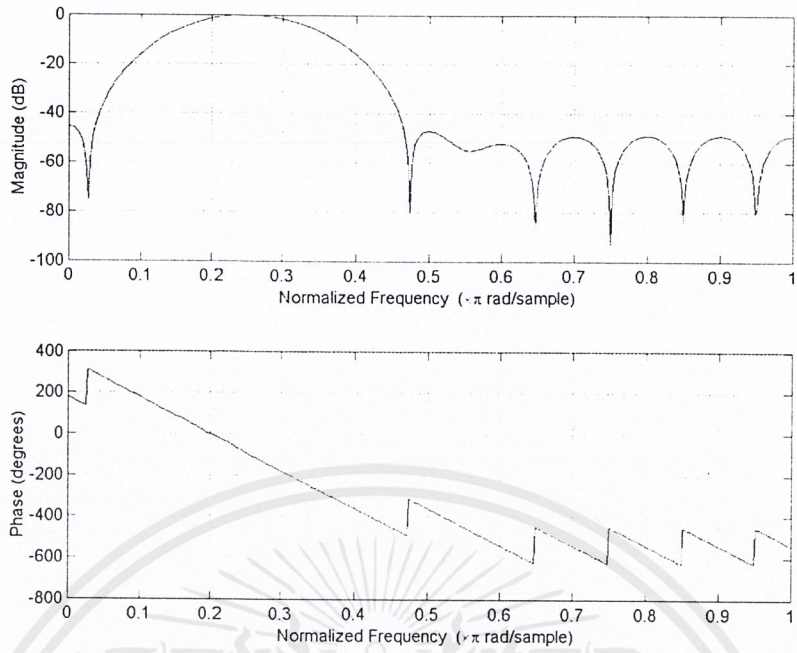


รูปที่ 5.16 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz, $n = 21$ ที่ได้จากโปรแกรม MATLAB

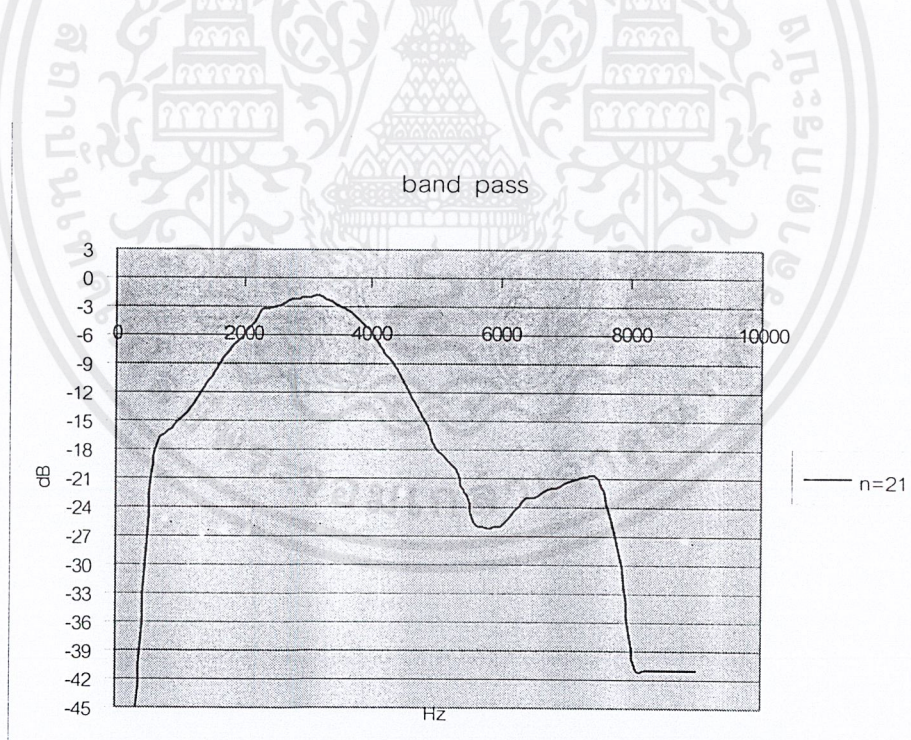


รูปที่ 5.17 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz, $n = 21$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

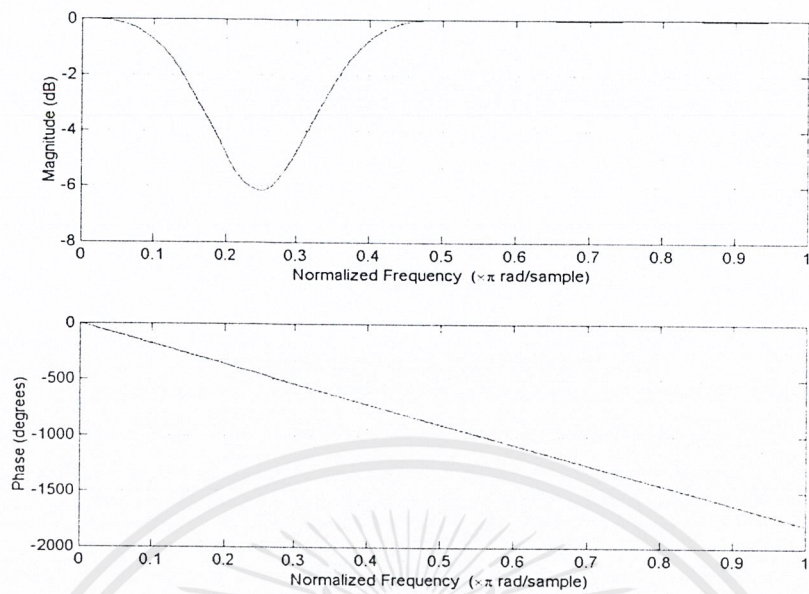


รูปที่ 5.18 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 21$ ที่ได้จากโปรแกรม MATLAB

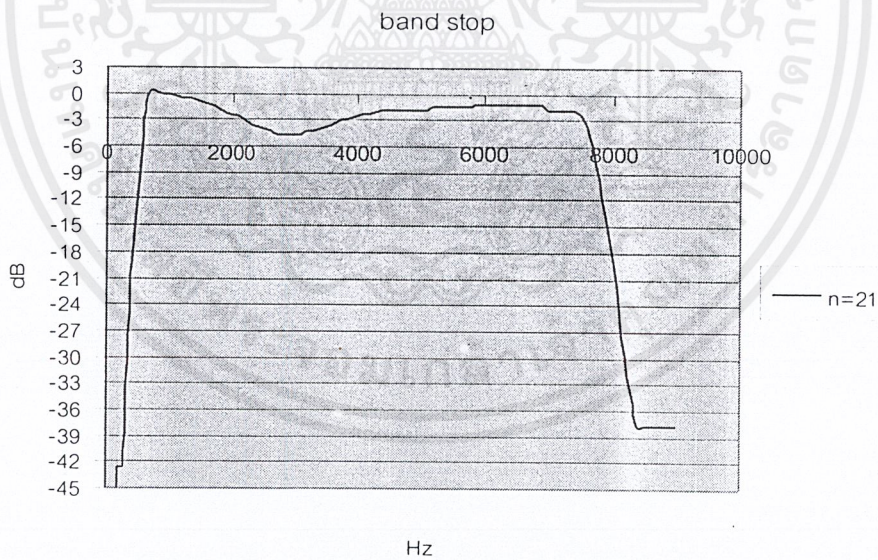


รูปที่ 5.19 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 21$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

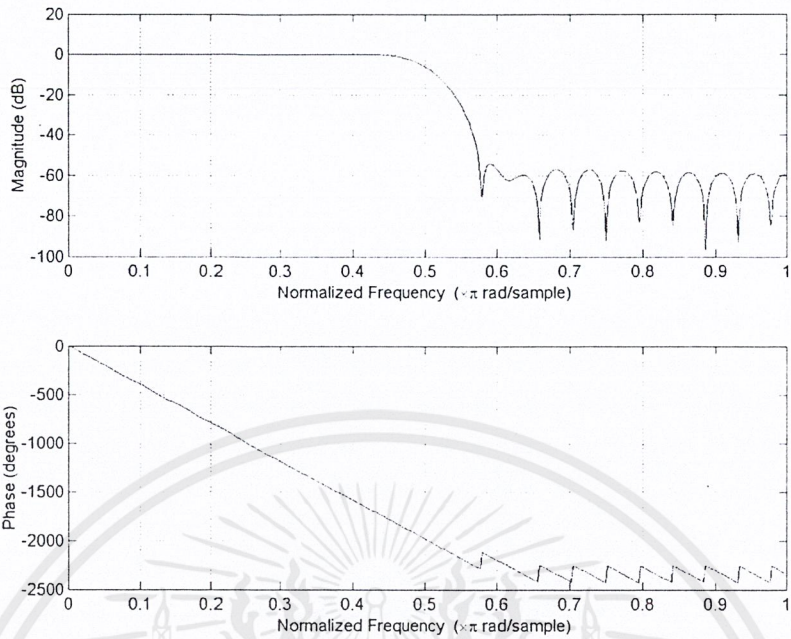


รูปที่ 5.20 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 21$ ที่ได้จากการออกแบบ

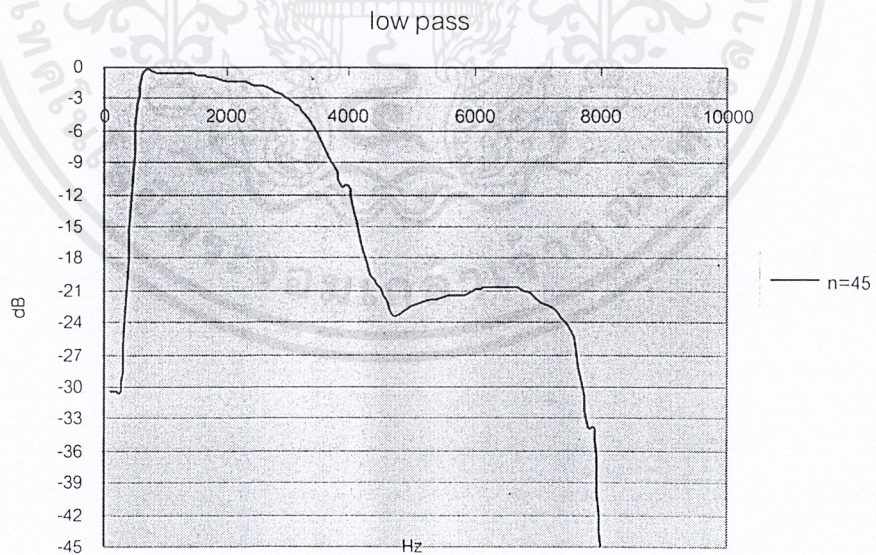


รูปที่ 5.21 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 21$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

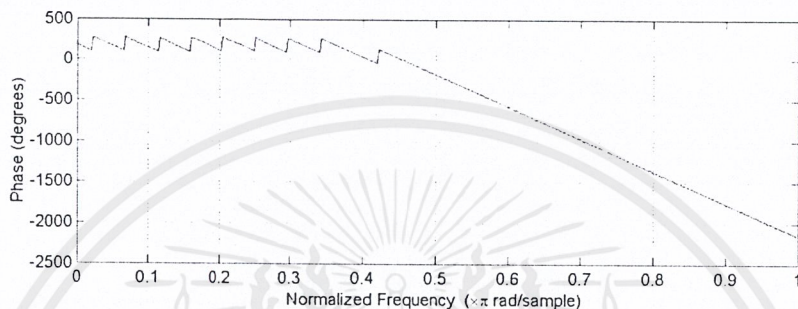
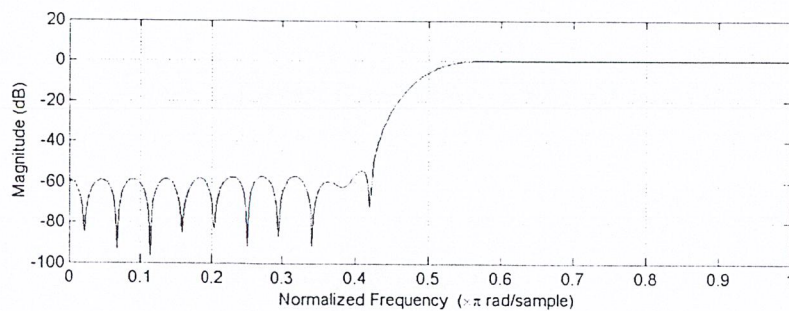


รูปที่ 5.22 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่านที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45ที่ได้จากการออกแบบ

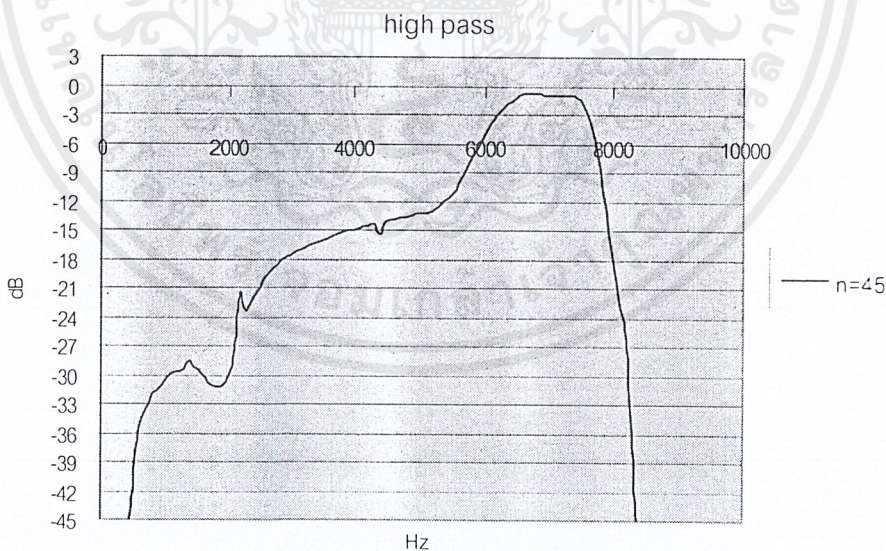


รูปที่ 5.23 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่านที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

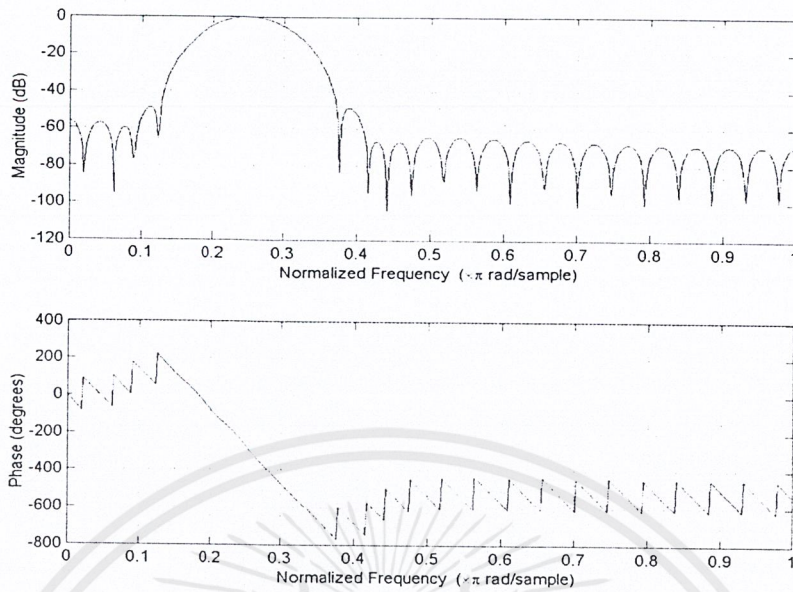


รูปที่ 5.24 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 45 ที่ได้จากโปรแกรม MAT LAB

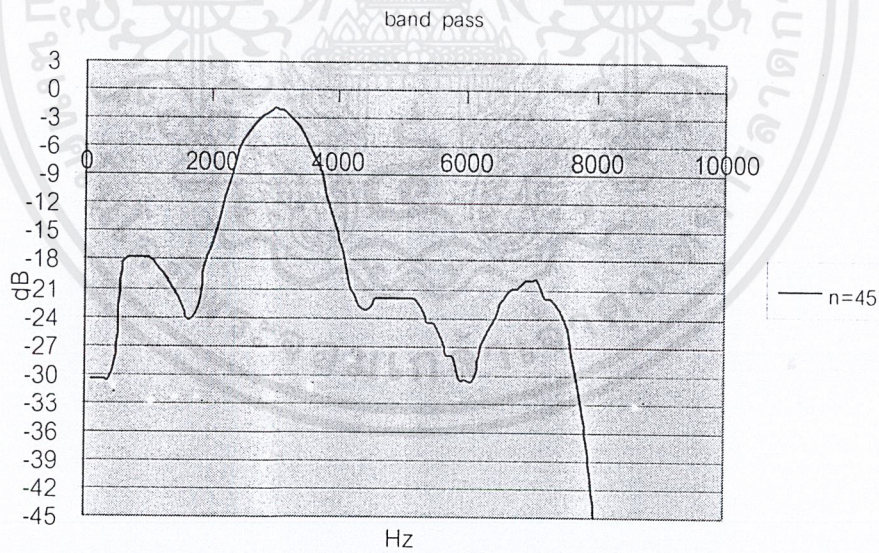


รูปที่ 5.25 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz , n = 4 ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

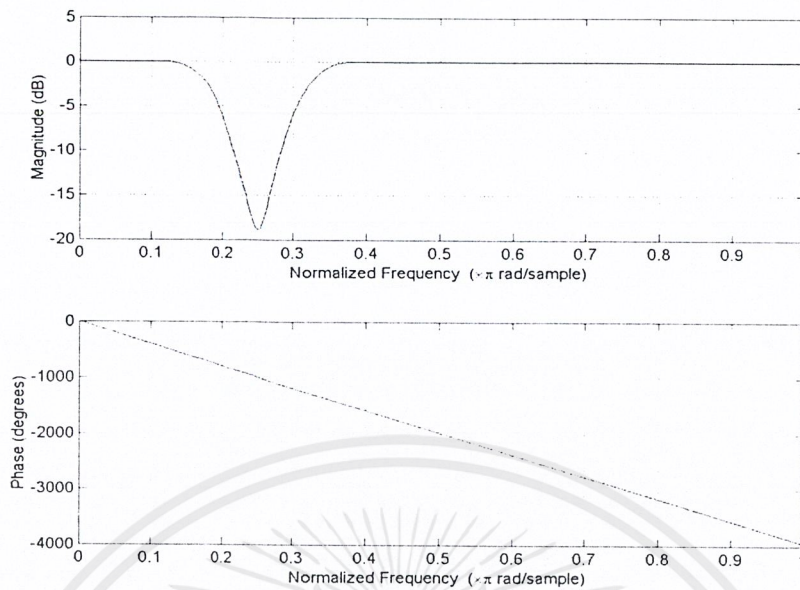


รูปที่ 5.26 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz, $n = 45$ ที่ได้จากโปรแกรม MATLAB

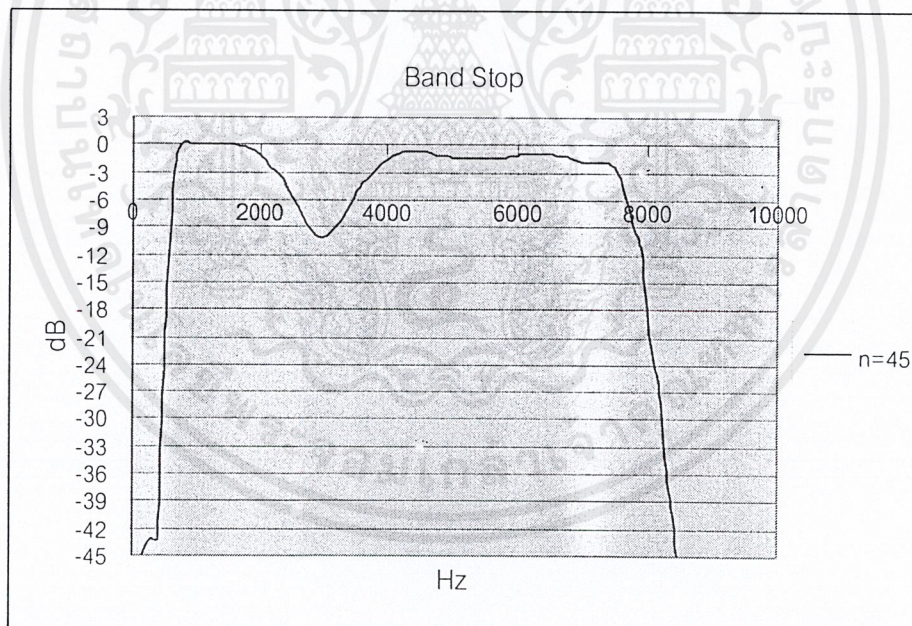


รูปที่ 5.27 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz, $n = 45$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

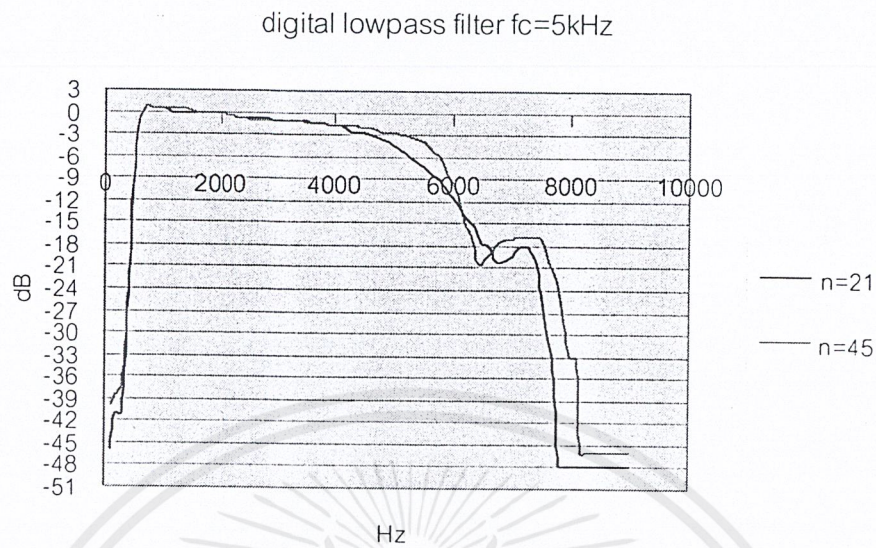


รูปที่ 5.28 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 45$ ที่ได้จากโปรแกรม MAT LAB

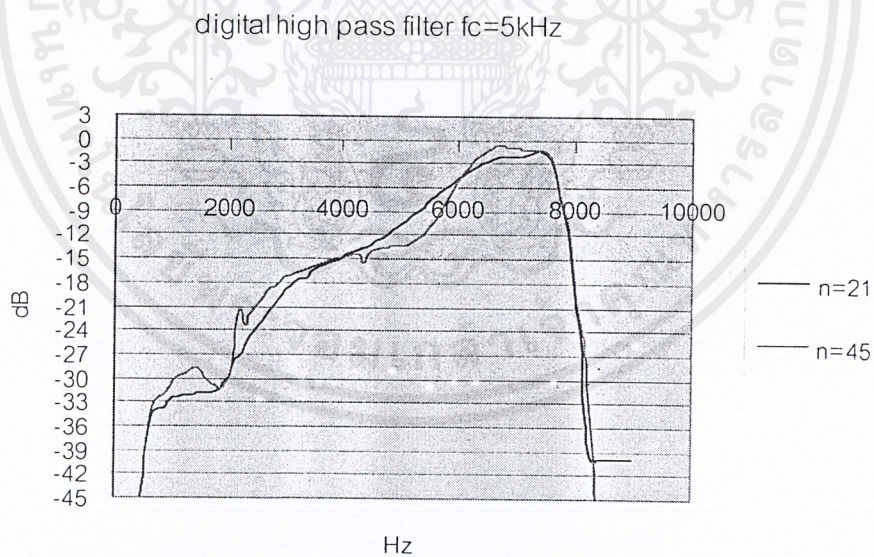


รูปที่ 5.29 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่กลางไม่ผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 2-3 kHz , $n = 45$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

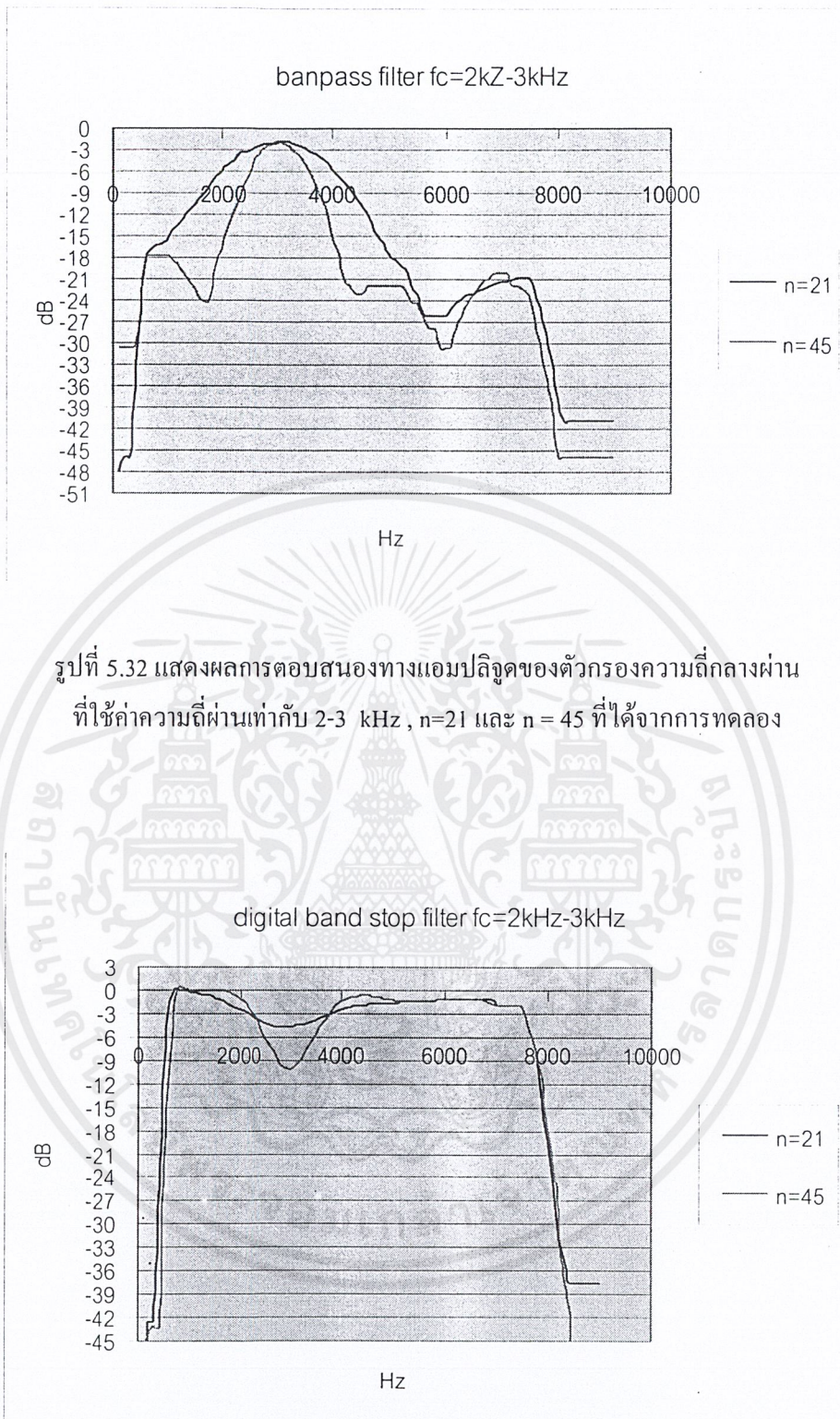


รูปที่ 5.30 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่ต่ำผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz, $n=21$ และ $n=45$ ที่ได้จากการทดลอง



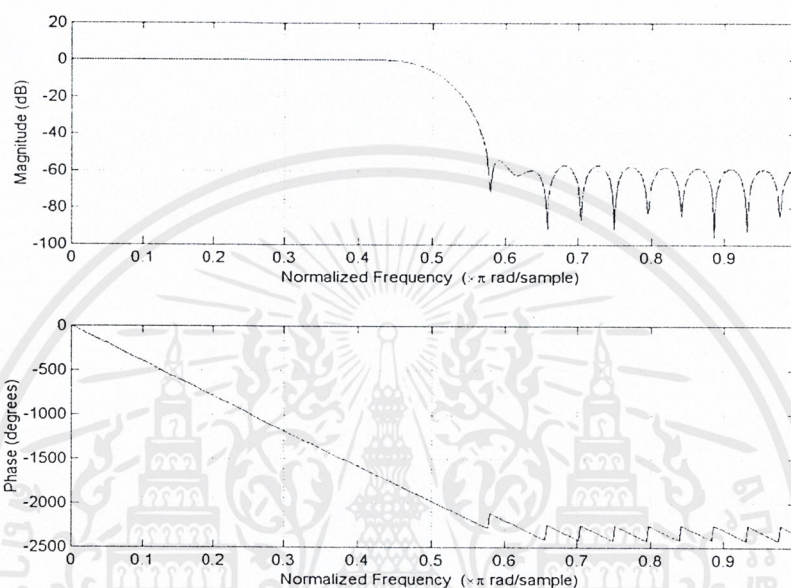
รูปที่ 5.31 แสดงผลการตอบสนองทางแอมพลิจูดของตัวกรองความถี่สูงผ่าน
ที่ใช้ค่าความถี่ผ่านเท่ากับ 5 kHz, $n=21$ และ $n=45$ ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

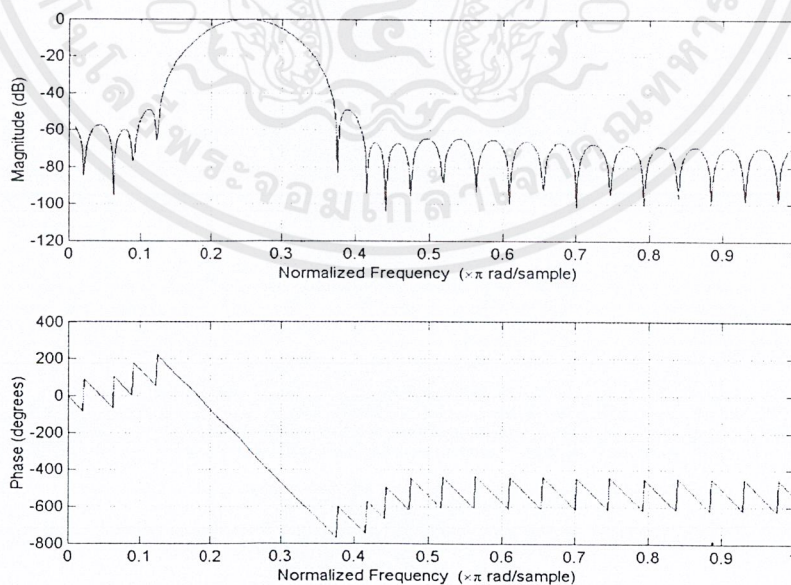


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 การทดลองที่ 4 เราจะดูผลการตอบสนองของเฟสว่าจะตรงกับที่ออกแบบไว้หรือไม่โดยตัวกรองความถี่ที่ใช้ทดสอบจะประกอบไปด้วย ตัวกรองความถี่ต่ำผ่านที่ 5 kHz และ ตัวกรองความถี่กลางผ่านที่ 2-3 kHz ที่ $n = 45$ โดยผลการตอบสนองของเฟสจากการ จะเป็นดังรูป 5.34 และ รูป 5.35 ตามลำดับ ซึ่งการตอบสนองของเฟสนั้นจะมีลักษณะเป็นเชิงเส้น (linear)

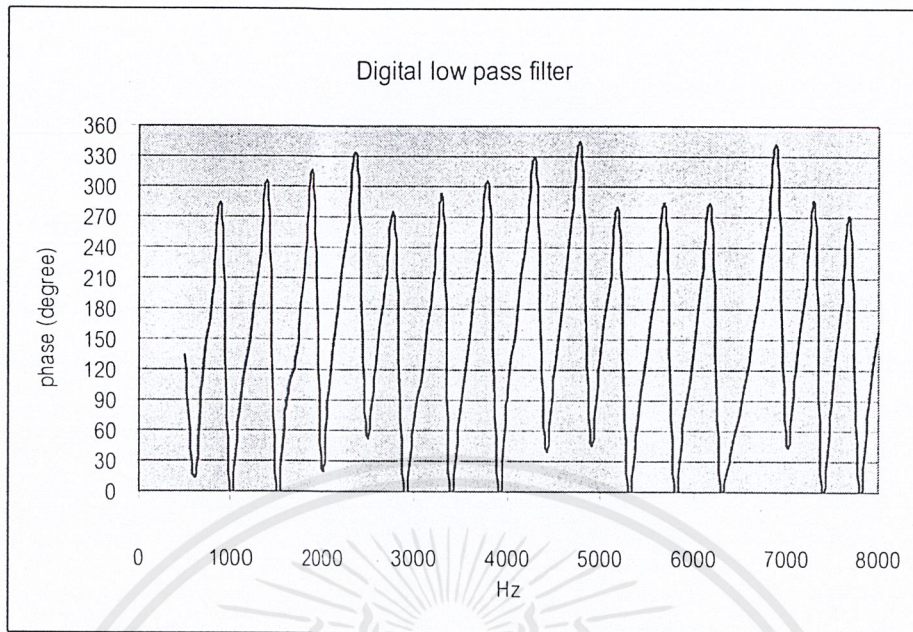


รูปที่ 5.34 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่ต่ำผ่าน ที่ความถี่ผ่านเท่ากับ 5 kHz จากการออกแบบ

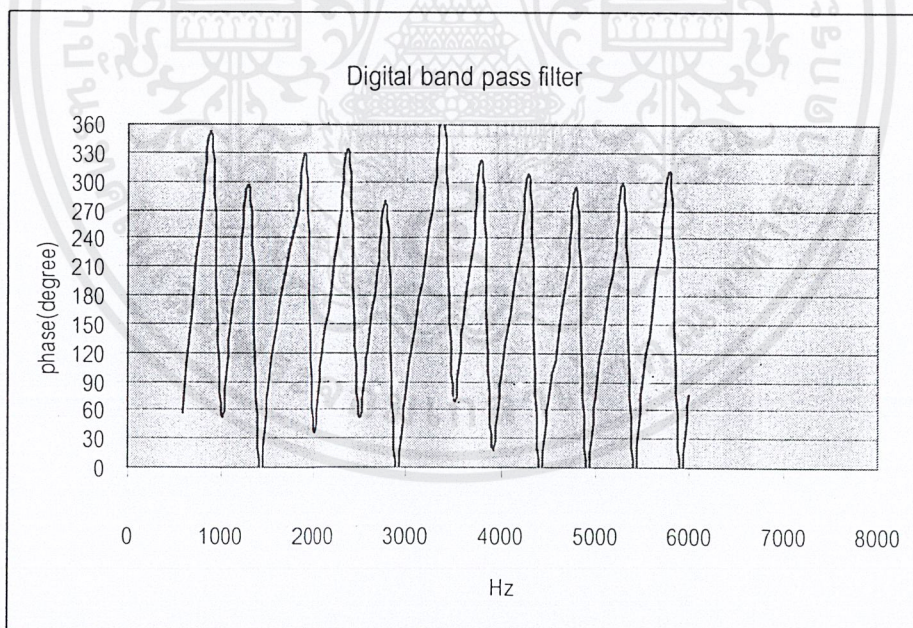


รูปที่ 5.35 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่กลางผ่าน ที่ความถี่ผ่านเท่ากับ 2-3 kHz จากการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.36 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่ต่ำผ่าน ที่ความถี่ผ่านเท่ากับ 5 kHz จากการทดลอง



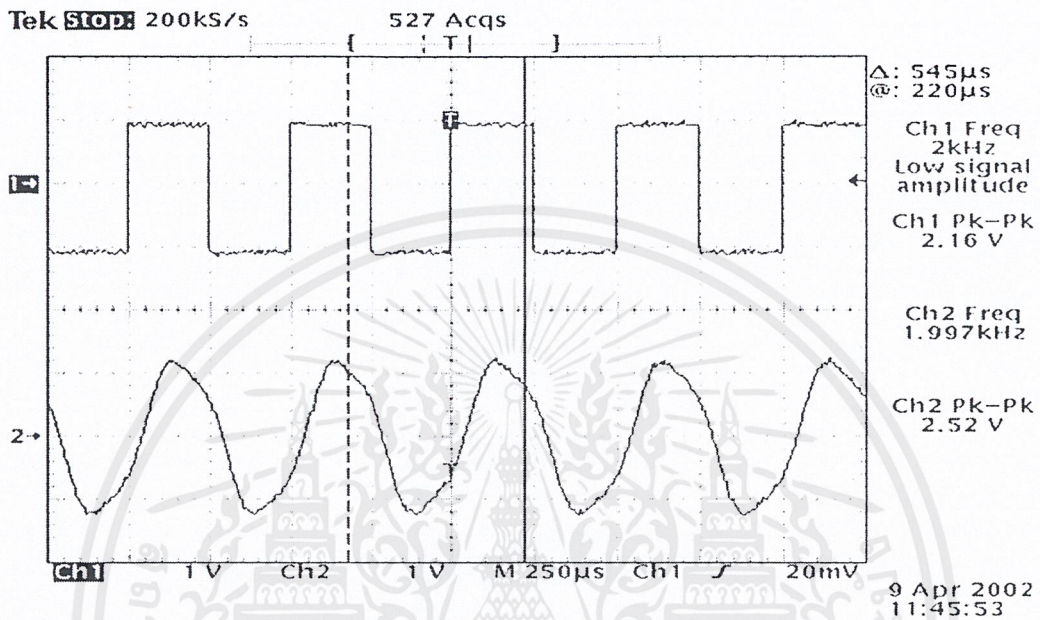
รูปที่ 5.37 ผลการตอบสนองทางแอมพลิจูดและเฟส ของตัวกรองความถี่กลางผ่าน ที่ความถี่ผ่านเท่ากับ 2-3 kHz จากการออกแบบ

จากผลการทดลองจะเห็นว่าผลการตอบสนองของเฟสจะมีลักษณะเป็นเชิงเส้น (linear)

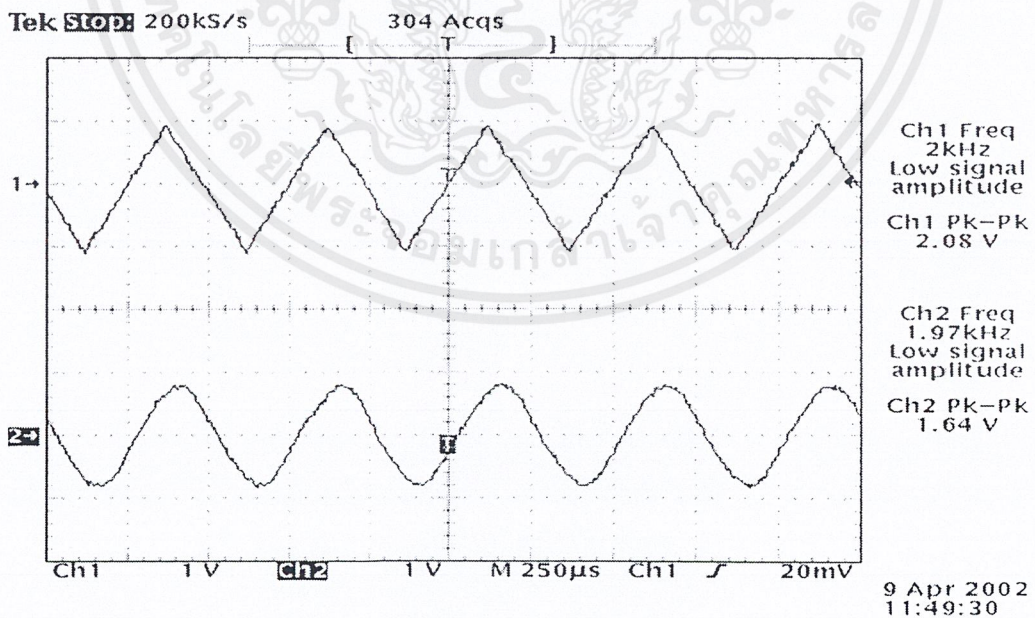
ซึ่งตรงตามการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 การทดลองที่ 5 เราจะดูลักษณะของสัญญาณที่ออกมาจาก ตัวกรองความถี่ ว่ามีลักษณะของสัญญาณถูกต้องตรงกับที่ป้อนเข้าไปหรือไม่ โดยการทดลองนี้เราจะทำการป้อนรูปคลื่นต่างๆเข้าสู่ตัวกรองความถี่ต่ำผ่านที่ความถี่ผ่านเท่ากับ 5 kHz โดยสัญญาณที่ป้อนจะมีความถี่เท่ากับ 2 kHz สัญญาณที่ป้อนนั้นจะประกอบไปด้วย สัญญาณคลื่นรูปสี่เหลี่ยม , สัญญาณคลื่นรูปสามเหลี่ยม และสัญญาณคลื่นไซน์

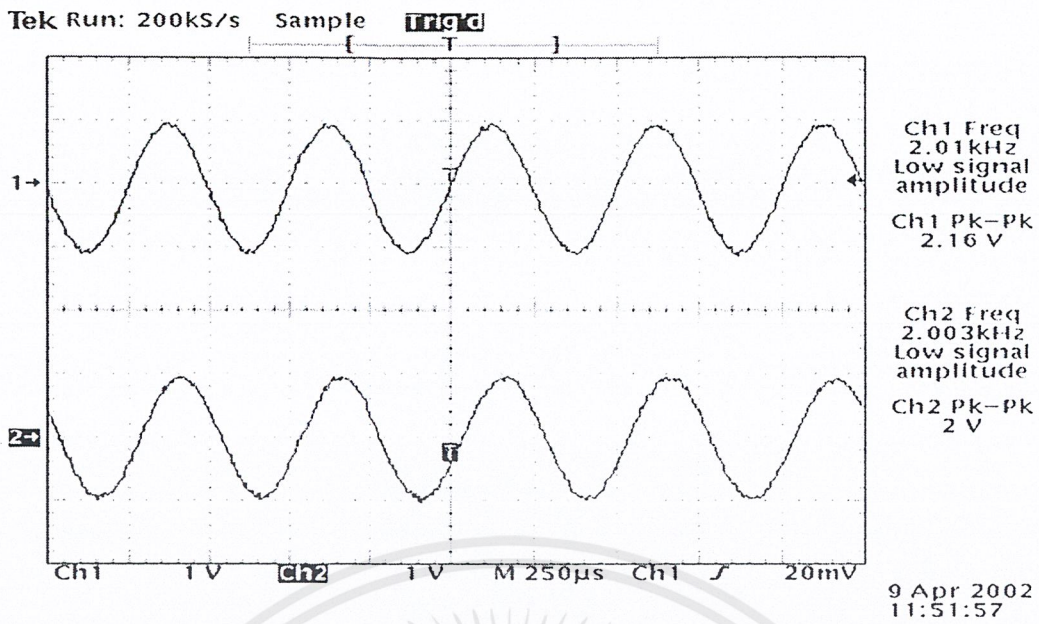


รูปที่ 5.38 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณสี่เหลี่ยม



รูปที่ 5.39 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณสามเหลี่ยม

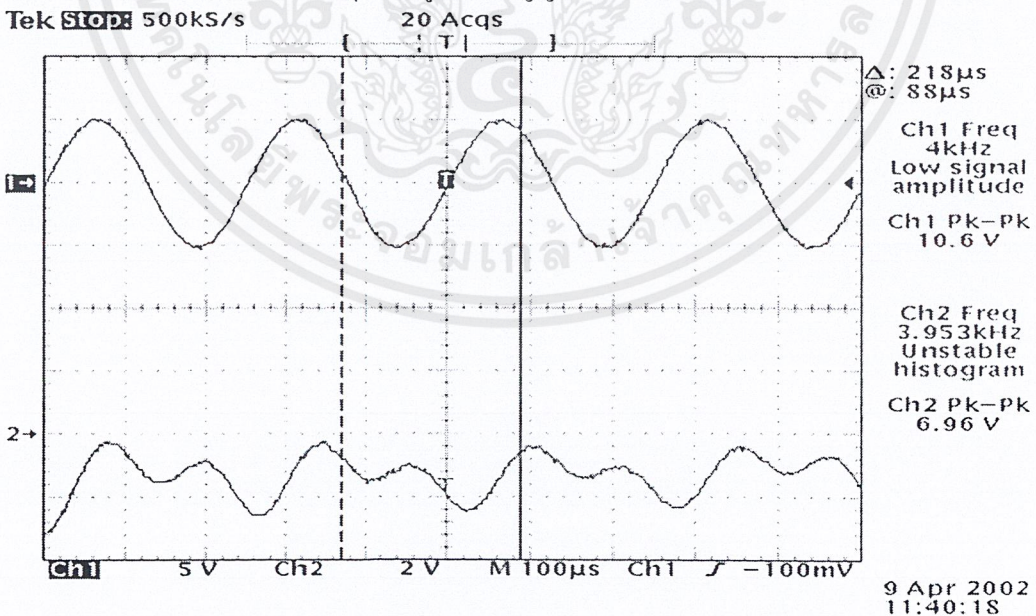
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.40 ลักษณะสัญญาณที่ได้จากตัวกรองความถี่ (รูปคลื่นที่ 2) โดยป้อนสัญญาณไซน์

จากการทดลองที่ 5 จะเห็นได้ว่าลักษณะของสัญญาณที่ได้จากตัวกรองความถี่นั้นจะมีค่าตรงกับสัญญาณที่ป้อนเข้า จะเห็นว่าเมื่อเราป้อนสัญญาณสี่เหลี่ยมและสัญญาณรูปสามเหลี่ยม รูปสัญญาณที่ได้จะไม่ตรงกับสัญญาณที่ออกมา เกิดจาก การผ่านตัวกรองความถี่ต่ำผ่าน ซึ่งจะกรองความถี่สูงทิ้ง ตรงตามทฤษฎี เนื่องจากองค์ประกอบของสัญญาณขาดหายไป

5.2.6 การทดลองที่ 6 เราต้องการหาขนาดของสัญญาณที่สูงสุดที่ป้อนเข้าสู่ตัวกรองความถี่ โดยทำการเพิ่มขนาดแรงดันทางด้านอินพุท จนรูปของสัญญาณที่ออกจากตัวกรองความถี่มีความผิดเพี้ยนไป



รูปที่ 5.41 แสดงลักษณะของสัญญาณที่ออกจากตัวกรองความถี่เมื่อป้อนแรงดันสูงสุด

จากรูปที่ 5.41 จะเห็นว่าค่าแรงดันสูงสุดที่ตัวกรองความถี่ตอบสนองต้องน้อยกว่า 10 Vp-p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปแลวิจาร์ณผล

จากการทดลองที่ 1 จะเห็นได้ว่าผลตอบสนองแอมพลิจูดของตัวกรองความถี่ชนิด ความถี่ต่ำผ่านที่ 0.3 ที่ วินโดว์แบบต่างๆ ที่ได้จากการทดลอง เมื่อนำมาเปรียบเทียบกันจะมีค่าใกล้เคียงกันมาก แต่จะมีค่าน้อยกว่าที่ออกแบบไว้ ในทุกๆวินโดว์ แล้วค่า ripple ซึ่งปรากฏในการออกแบบนั้น ไม่ปรากฏในการทดลองเนื่องจาก ค่าการลดทอนตัวกรองความถี่ที่ได้ในการทดลองมีค่าไม่ถึง -50 dB (จากรูป ค่า ripple จะปรากฏในช่วงการลดทอน ต่ำกว่า -50 dB) เนื่องจากที่ระดับของสัญญาณน้อยๆ การวัดขแรงดันทำได้ลำบาก และความกว้างของช่วงความถี่ที่เราวัดแต่ละจุดมีค่าห่างกัน (เราทำการวัดแรงดันที่ที่ความถี่ต่างๆ ที่ละ 100 Hz)

Sampling rate (Hz)	fc (Hz)
10,000	2,000
20,000	3,900

ตารางที่ 6.1 แสดงค่าความถี่ผ่านที่ Sampling rate ต่างๆ

จากการทดลองที่ 2 จะเห็นว่าเราสามารถเปลี่ยนแปลงค่าความถี่ในการสุ่มให้มีค่ามากขึ้นหรือลดลงได้ ถ้าค่าความถี่ในการสุ่มมีค่าสูงขึ้นจะทำให้ค่าความถี่ผ่านผิดพลาดมากขึ้นด้วย จากตาราง 6.1 ค่าความถี่ผ่านที่ใช้ค่าความถี่ในการสุ่มเท่ากับ 10 kHz เท่ากับ 2 kHz ($\omega_c = 0.4$) ซึ่งมีค่าตรงกับกรออกแบบ แต่ถ้าใช้ค่าความถี่ในการสุ่มเท่ากับ 20 kHz จะได้ค่าความถี่ผ่านเท่ากับ 3.9kHz ซึ่งมีค่าคลาดเคลื่อนแต่มีค่าไม่มาก

Type of digital filter	fc ที่ n = 21	fc ที่ n = 45
Low pass filter	4.6	5.2
High pass filter	6.4	6.2
Band pass filter	2.5-3.55	2.75-3.3
Band stop fliter	2.2-3.8	2.25-3.75

ตารางที่ 6.2 แสดงค่าความถี่ผ่านที่ตัวกรองความถี่ชนิดต่างๆ

จากผลการทดลองที่ 3 (ตารางที่ 6.3) จะเห็นว่า จำนวนจุดข้อมูลจะมีผลต่อค่าความถี่ผ่านมาก เนื่องจากการลดทอนของตัวกรองความถี่ในการออกแบบ กับการทดลองมีค่าต่างกันมาก กล่าวคือ ในการออกแบบค่าการลดทอนต่ำกว่าการทดลองมาก ทำให้ค่าความถี่ผ่านมีความคลาดเคลื่อนสูง (การลดทอนเท่ากับ 3 dB)

จากการทดลองที่ 4 จะเห็นว่าผลการตอบสนองของเฟสนั้นจะเป็นแบบเชิงเส้นซึ่งตรงกับ ทฤษฎีและการออกแบบ จะสังเกตได้ว่าจำนวนจุดทศนิยมและจำนวนจุดข้อมูล (N) ไม่มีผลเนื่องจาก โครงสร้างจะเป็นตัวกำหนดผลการตอบสนองของเฟส

จากการทดลองที่ 5 จะเห็นได้ว่าลักษณะของสัญญาณที่ได้จากตัวกรองความถี่นั้นจะมีค่าตรงกับสัญญาณที่ป้อนเข้า จะเห็นว่าเมื่อเราป้อนสัญญาณสี่เหลี่ยมและสัญญาณรูปสามเหลี่ยม รูป สัญญาณที่ได้จะไม่ตรงกับสัญญาณที่ออกมา เกิดจาก การผ่านตัวกรองความถี่ต่ำผ่าน ซึ่งจะกรอง ความถี่สูงทิ้ง ตรงตามทฤษฎี เนื่องจากองค์ประกอบของสัญญาณขาดหายไป

จากการทดลองที่ 6 เราจะได้ค่าแรงดันสูงสุดที่ตัวกรองความถี่สามารถทำงานได้อย่างถูกต้อง คือ ต้องน้อยกว่า 10 Vp-p และจะมีเกิดการขลิบของสัญญาณเกิดขึ้น แต่ว่ารูปร่างลักษณะของ สัญญาณจะผิดเพี้ยนไป

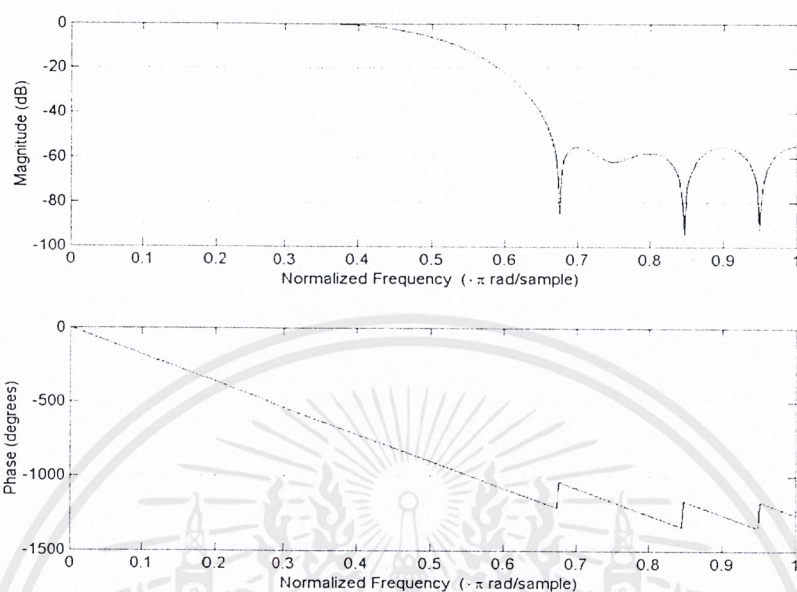
สรุปผลการทดลอง

จากการทดลองหาค่าความถี่ผ่านของตัวกรองเชิงเลข (cutoff frequency) ในแบบต่างๆ พบ ว่าค่าความถี่ผ่านที่ได้จากการทดลองนั้นมีค่าคลาดเคลื่อนไปจากที่ได้ทำการออกแบบไว้ ซึ่งอาจเกิด จากสาเหตุดังต่อไปนี้

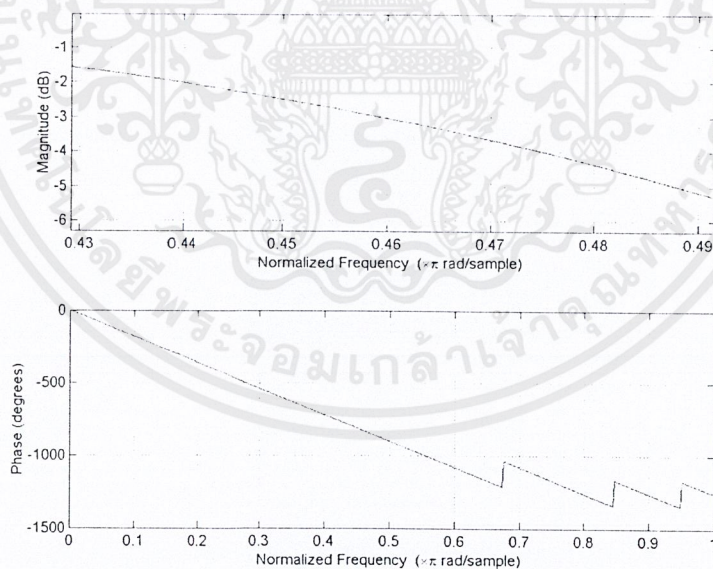
1. เกิดจากการป้อนค่าสัมประสิทธิ์ตัวกรองเชิงเลขที่มีจำนวนทศนิยมหลายตำแหน่ง ซึ่งใน การคำนวณค่าความถี่ผ่านนั้นทำให้เกิดผลในเชิงลบ กล่าวคือ ค่าที่ได้จะมีความคลาดเคลื่อน สูง เพื่อให้เข้าใจได้ง่ายขึ้นจะขอยกตัวอย่าง เช่น $1.2356243e-1$ ค่านี้ประกอบด้วยทศนิยม 7 ตำแหน่งแต่ในการทดลองจริงจะใช้ทศนิยมเพียง 3 ตำแหน่งเท่านั้น คือ $1.235e-1$ ซึ่งจะ ทำให้ได้ค่าความถี่ผ่านของตัวกรองเชิงเลขตรงตามที่คำนวณไว้ หมายความว่าค่าของทศ นิยมส่วนที่ไม่ใช่เท่ากับ 0.0006243 แม้ในทางคณิตศาสตร์จะถือว่ามีค่าน้อยมาก แต่ในการ คำนวณค่าสัมประสิทธิ์ของตัวกรองเชิงเลขที่มีค่าไม่เกิน 1 แล้วถือว่ามีค่าดังกล่าวส่งผลต่อการ คำนวณเป็นอย่างมาก เมื่อทำการเปรียบเทียบผลที่ได้จากการทดลอง

2. ในการออกแบบตัวกรองเชิงเลขโดยใช้โปรแกรม MATLAB เมื่อนำผลที่ได้ไปทำการ เปรียบเทียบกับผลการทดลองพบว่าค่าความถี่ผ่านของตัวกรองเชิงเลขไม่ตรงกับค่าที่ได้ทำ การออกแบบไว้ เช่น จากรูป 6.1 วงจรกรองความถี่ต่ำผ่านแบบเชิงเลข ค่าความถี่ผ่าน 5 kHz ซึ่งค่าที่อ่านได้จากกราฟรูปที่ 6.2 เท่ากับ 4.6 kHz ซึ่งจะเห็นได้ว่าคลาดเคลื่อนจากที่ได้ทำ การออกแบบไว้ ความผิดพลาดดังกล่าวอาจเกิดจากการนำค่าที่คำนวณจากโปรแกรม

MATLAB มาทำการแสดงผลเป็นกราฟดินแบบ ซึ่งมีจำนวนจริงจำกัด ทำให้ค่าที่ได้ไม่ครอบคลุมเท่าที่ควรจึงส่งผลต่อการเปรียบเทียบและสรุปผล



รูปที่ 6.1 แสดงการตอบสนองทางแอมพลิจูดชนิดความถี่ต่ำผ่านที่ความถี่ผ่านเท่ากับ 5kHz $n=21$



รูปที่ 6.2 แสดงการตอบสนองทางแอมพลิจูดชนิดความถี่ต่ำผ่านที่ความถี่ผ่านเท่ากับ 5kHz $n=21$
ซึ่งผ่านการขยายที่จุดความถี่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6.2 จะเห็นว่าค่า สัมประสิทธิ์ของตัวกรองความถี่ที่ได้มีค่าคาดเคลื่อนจากการออกแบบแต่จะเห็นว่าค่าที่เรา ออกแบบค่าความถี่ผ่านไว้ที่จริงที่ 4.6 kHz ตรงกันกับการทดลองและ การเปลี่ยนแปลงค่าจำนวนจุดข้อมูลและค่าแซมปลิงเรท มีผลต่อค่าความถี่ผ่านมาก

วิจารณ์ผลการทดลอง

จากการทดลองจะเห็นได้ว่า ลักษณะการตอบสนองทางแอมพลิจูดและเฟสมีค่าถูกต้องตาม ชนิดแต่ค่าความถี่ผ่านมีความคลาดเคลื่อนอยู่บ้าง ในการออกแบบถ้าเราสามารถออกแบบค่าความถี่ผ่านให้(ที่ออกแบบโดยโปรแกรม) ถูกต้องเราจะ ได้ผลการตอบสนองที่ถูกต้องขึ้นด้วย ตัวกรองความถี่เชิงเลขนี้สามารถนำไปใช้งานจริงได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ศ.ดร. วัลลภ สุระกำพลธร, “การประมวลผลสัญญาณเชิงเลข” ,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 384 หน้า , 2533
2. รศ.ดร. มนัส สัจจวิเศษ, “คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์” . กรุงเทพฯ อินโฟเพรส ,508 หน้า , 2543
3. RULPH CHASSAING, “Digital Signal Processing” , University of Massachusetts ,301 p,1998
4. TMS320C3x DSP Starter Kit User’s Guide



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIR FILTER

```

        .start    ".text",0x809900
        .start    ".data",0x809C00
        .include  "AICCOM31.ASM"
        .include  "45LPD3.COF"
        .data
XB_ADDR .word    XN+LENGTH-1
HN_ADDR .word    COEFF
AICSEC  .word    0A14h,1h,3E7Eh,67h
        .brstart "XN_BUFF",64
XN      .sect    "XN_BUFF"
        .loop    LENGTH
        .float   0
        .endloop
        .text
        .entry   BEGIN
BEGIN   LDP      AICSEC
        CALL    AICSET
        LDI    LENGTH,BK
        LDI    @XB_ADDR,AR1
FILT   LDI    LENGTH-1,AR4
LOOP   CALL    AICIO_P
        FLOAT  R6,R3
        STF   R3,*AR1++%
        LDI  @HN_ADDR,AR0
        LDF  0,R0      ;init R0
        LDF  0,R2      ;init R2
        RPTS LENGTH-1
        MPYF3 *AR0++,*AR1++%,R0
        ADDF3 R0,R2,R2
        DBNZD AR4,LOOP
        ADDF  R0,R2
        FIX  R2,R7
        NOP
        BR  FILT
        .end    ;end

```