



วงจรรองเชิงเลข
DIGITAL FILTER



โดย
นายณัฐจักร ตรีสัตยาเวช
นายภาสกร บาริสรี
นายยุทธการ ชุนสนิท

วัน เดือน ปี.....
เลขทะเบียน.....
เลขเรียกหนังสือ.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

๒๓
๓๖321๐
๒๕๓๙

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งในการนำไปใช้

วงกรองเชิงเลข
DIGITAL FILTER

โดย

นายณัฐจักร	ตรีสัตยาเวทย์	36014134
นายภาสกร	บาริสรี	36014318
นายยุทธการ	ขุนสนิท	36014335

อาจารย์ที่ปรึกษา
ศ.ดร. วัลลภ สุระกำพลธร

ปริญญานิพนธ์สำหรับปริญญาวិสากรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2539

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง วจรกรองเชิงเลข

ผู้จัดทำ

1. นายณัฐจักร ศรีสัตยาเวช
2. นายภาสกร บาริสรี
3. นายยุทธการ ชุนสนิท

ศ.ดร. วัลลภ สุระกำพลธร
(.....)

อาจารย์ที่ปรึกษา

วงจรรองเชิงเลข

DIGITAL FILTER

นายณัฐจักร์ ศรีสัตยาเวทย์

นายภาสกร บาริสรี

นายยุทธการ ชุนสนิท

โครงการได้รับการตรวจแล้ว พร้อมทั้งจะทำการสอบได้

ศ.ดร. วัลลภ สุระกำพลธร

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วจรกรองเชิงเลข

นายณัฐจักร์ ศรีสัตยาเวทย์
นายภาสกร บาริศรี
นายยุทธการ ชุนสนิพ
ศ.ดร. วัลลภ สุระกำพลธร
อาจารย์ที่ปรึกษา
ปีการศึกษา 2539

บทคัดย่อ

รายงานนี้จะนำเสนอและอธิบายเรื่องราวของการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing) โดยการเปลี่ยนแปลงรูปทางสัญญาณและทำการคำนวณทางคณิตศาสตร์ด้วย ไมโครโปรเซสเซอร์ การประมวลผลสัญญาณเชิงเลขนั้น สามารถนำไปประยุกต์ใช้งานได้มากมาย เช่น อิมเมจโปรเซสซิ่ง การแปลงฟาสต์ฟูริเยอร์ รวมทั้งการทำเป็นวงจรกรองสัญญาณเชิงเลข (Digital Filter) โดยเราจะใช้การ์ด DSK starter kit ซึ่งประกอบไปด้วยไมโครโปรเซสเซอร์ TMS320C50 ที่มีตัวแปลงทางอนาล็อก TLC32040 รวมอยู่ด้วย TMS 320C50 เป็น CPU ที่มีความเร็วและสถาปัตยกรรมที่ได้รับการปรับปรุงให้มีความสามารถดีขึ้น โดยการคำนวณแบบขนาน มีชุดคำสั่งที่สร้างขึ้นมาโดยเฉพาะ ทำให้คำนวณผลแบบเวลาจริงได้ จุดมุ่งหมายของเราคือการใช้ TMS320C50 ในการประยุกต์เกี่ยวกับการกรองสัญญาณเชิงเลข

DIGITAL FILTER

Nuttajuk Treesuttayawet

Passakorn Barisri

Yuttakarn Chunsanit

Prof. Wallop Surakhumpolthorn

1996

ABSTRACT

This thesis describes and presents digital signalProcessing that signal was transformed and calculated by method of mathematics on a Microprocessor. The digital Signal processing can apply in many ways such as Imageprocessing, fast fourier transform and digital filter. We use aDSK starter Kit towork for this project.The DSK starter Kit consistsof microprocessor TMS320C50 and a anolog interface coverter name as TMS32040. The TMS320C50 is a high speed and completed structure CPU to apply digital signal processing .TheTMS320C50 can proceed in parallel form and have sets of command to work in real time. We will use TMS320C50 to present digital filter.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1-2
บทที่ 2 ทฤษฎีพื้นฐานของวงจรกรองเชิงเลข	3-11
2.1 วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด (FIR FILTER)	4-6
2.2 วงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด (IIR FILTER)	7-8
2.3 เปรียบเทียบวงจรกรองแบบผลตอบสนองอิมพัลส์ จำกัดกับวงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด	9-10
2.4 การออกแบบวงจรกรองเชิงเลข	10-11
บทที่ 3 การออกแบบวงจรกรองเชิงเลข	12-35
3.1 ตัวกรองเชิงเลข	12-14
3.2 ตัวกรองเชิงเลขที่มีผลตอบสนองจำกัด	14-14
3.3 ความสัมพันธ์ของระบบการแปลงฟูริเยอร์	14-15
3.4 ผลตอบสนองความถี่ของระบบ	15-16
3.5 ความหมายของโครงข่าย	17-18
3.6 การสร้างตัวกรอง IIR แบบโดยตรง	18-19
3.7 การสร้างตัวกรอง IIR แบบอนุกรมและขนาน	19-21
3.8 การออกแบบวงจรกรองเชิงเลขแบบ FIR	22-35
บทที่ 5 ไมโครโปรเซสเซอร์ TMS320C50	39-55
5.1 จุดเด่นของ TMS320C50	39-40
5.2 การจัดหน่วยความจำ TMS320C50	43-47
5.2.1 หน่วยความจำข้อมูล (Data Memory)	43-45
5.2.2 หน่วยความจำโปรแกรม (Program Memory)	46-47
5.3 โหมดการอ้างอิงหน่วยความจำ (Memory Addressing Mode)	47-48
5.4 อุปกรณ์เสริม (Peripherals)	48-52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.1 อินเทอร์รัพท์ (Interrupt)	49-49
5.4.2 พอร์ตอนุกรม (Serail Port)	49-51
5.4.3 ไทม์เมอร์ (Timer)	51-52
5.5 ตัวสร้างสถานะรอที่สามารถโปรแกรมได้	52-55
บทที่ 6 วงจรอินเทอร์เฟซสัญญาณอนาล็อก TLC 32040	56-63
6.1 การทำงานของ TLC32040	57-61
บทที่ 7 การเชื่อมต่อแบบขนาน	64-68
7.1 โครงสร้าง หน้าที่ และการเชื่อมต่อขนาน	64-68
บทที่ 8 การออกแบบตัวกรองเชิงเลขแบบผลตอบสนองอิมพัลส์ไม่จำกัดและจำกัด	69-92
8.1 การเปรียบเทียบตัวกรองดิจิทัล	70-71
8.2 รูปแบบการออกแบบตัวกรองเชิงเลข	71-72
8.3 ตัวกรองความถี่ต่ำ	72--72
8.4 ตัวกรองความถี่ต่ำแบบบัทเทอร์เวิร์ธ	72-75
8.5 ช่วงความถี่การแปลง	75-77
8.6 การแปลงแบบเชิงเส้นคู่	77-80
8.7 สมการการออกแบบตัวกรองเชิงเลข	80-82
8.8 ฟังก์ชันส่งผ่านความถี่ต่ำสัมประสิทธิ์ของตัวตั้ง	82-83
8.9 ตัวกรองเชิงเลขความถี่ต่ำในส่วนอันดับที่สอง	83-85
8.10 สมการการออกแบบตัวกรองเชิงเลขความถี่ต่ำ	85-86
8.11 ฟังก์ชันส่งผ่านส่วนอันดับที่สองของสัมประสิทธิ์ตัวตั้ง	86-86
8.12 สมการการออกแบบตัวกรองเชิงเลขของแบนด์พาส	86-89
8.13 สมการการออกแบบตัวกรองเชิงเลขแบบแบนด์สตัด	90-91
8.14 การคำนวณสัมประสิทธิ์ตัวเลขของตัวกรองแบนด์สตัด	91-91
8.15 ตัวกรองเชิงเลขแบนด์สตัดอันดับที่สองสำหรับ N คือ	91-92
บทที่ 9 โครงสร้างของโครงการ	93-94
บทที่ 10 การอินเทอร์เฟซระหว่าง IBM ความพิวเตอร์กับบอร์ด DSK TMS320C50	95-98
บทที่ 11 โปรแกรมที่ใช้งานในการท่วงจรเชิงเลข	99-128

บทที่ 12 การทดลองและสรุปผล

129-179

สรุปผลการทดลองและวิจารณ์

177-179

ภาคผนวก

- กิตติกรรมประกาศ
- โปรแกรม fir.asm
- โปรแกรม iir-w.asm
- โปรแกรม ini.asm
- โปรแกรม load-dsk.pas
- โปรแกรม fir-iir.pas
- การอินเตอร์เฟสกับคอมพิวเตอร์
- วงจรบอร์ด TMS320C50
- วงจรบอร์ด อนุสอกรอินเตอร์เฟส



สารบัญรูป

	หน้า
รูปที่ 2.1 บล็อกไดอะแกรมของวงจรกรองเชิงเลข	3
รูปที่ 2.2 แสดงโครงสร้างของตัวกรองไม่ป้อนกลับเชิงเลขแบบโดยตรง	6
รูปที่ 2.3 แสดงโครงสร้างของตัวกรองป้อนกลับเชิงเลขแบบโดยตรง	8
รูปที่ 3.1 แสดงสมการค่าสัมประสิทธิ์ที่เชิงเส้นรูปแบบตรงของสมการ 3.2	12
รูปที่ 3.2 แสดงผลตอบสนองของตัวกรองหนึ่งของตัวกรอง FIR และ IIR	14
รูปที่ 3.3 แสดงความถี่สลับบนสเกลหนึ่งหน่วย	16
รูปที่ 3.4 แสดงรูปแบบของโครงข่ายเชิงเลข IIR(a) โครงสร้างแบบโดยตรง IIR อันดับสอง (b) ตัวกรอง IIR อันดับสองพร้อมกับส่วน Feed-Forward และ Feedback	17
รูปที่ 3.5 แสดงตัวอย่างการสลับ IIR(a) รูปแบบแบบสลับของรูปที่ 3.4b ตัว กรอง IIR อันดับสองกับหน่วย (b) รูปที่ 3.5 a เขียนใหม่โดยแสดง อินพุตอยู่ทางซ้ายมือรูปแบบมาตรฐาน	18
รูปที่ 3.6 แสดงการสร้างตัวกรอง IIR แบบโดยตรง	19
รูปที่ 3.7 แสดงโครง IIR อันดับหกแบบอนุกรมและขนาน (a) โครงสร้างแบบ อนุกรมโดยใช้ส่วนอันดับสอง	20
รูปที่ 3.8 แสดงตัวกรองคิซิงตัวเลข FIR(a) โครงแบบโดยตรง (b) โครงสร้าง แบบอนุกรม	21
รูปที่ 3.9 แสดงความสัมพันธ์ระหว่างเหตุภาพและไม่เป็นเหตุภาพของผล ตอบสนอง Impulse	28
รูปที่ 3.10 แสดงผลตอบสนองความถี่ในอุดมคติ(a) ตัวกรองความถี่ต่ำ(b) ตัวกรองความถี่สูง(c) ตัวกรองความถี่ผ่าน(d) ตัวกรองความถี่หยุด	31
รูปที่ 3.11 แสดงการออกแบบตัวกรองผ่านความถี่ต่ำ Kaiser Window	34
รูปที่ 4.1 บล็อกไดอะแกรมของ TMS320C50 DSK	36
รูปที่ 4.2 แสดงหน่วยความจำภายใน TMS320C50 DSK	37
รูปที่ 4.3 แสดงการต่อระหว่าง DSK และ PC โดยผ่านทาง RS232	38
รูปที่ 5.1 แสดงบล็อกไดอะแกรม TMS320C50	41
รูปที่ 5.2 บล็อกไดอะแกรมส่วนของหน่วยประมวลผลกลาง	42
รูปที่ 5.3 แสดงการเข้าถึงหน่วยความจำของ TMS320C50	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.4	แสดงการต่อพอร์ทอนุกรมกับพอร์ทภายนอก	50
รูปที่ 5.5	แสดงบล็อกไคอะแกรมการทำงานของพอร์ทอนุกรม	51
รูปที่ 5.6	แสดงบล็อกไคอะแกรมของไทม์เมอร์	52
รูปที่ 5.7	แสดงบล็อกไคอะแกรมของตัวสร้างสถานะรอ	55
รูปที่ 6.1	แสดงบล็อกไคอะแกรมการทำงานของ TLC 32040	56
รูปที่ 6.2	แสดงโครงสร้างของสัญญาณเวลาภายใน (Configuration Timing Internal)	59
รูปที่ 6.3	แสดงรูปแบบบิตใน AIC DR หรือ DX Word	59
รูปที่ 6.4	แสดงการเชื่อมต่อระหว่าง TMS320C50 และ TLC3204	60
รูปที่ 6.5	แสดงช่วงเวลาในการรับส่งข้อมูลระหว่าง TLC32040 กับ TMS320C50	62
รูปที่ 7.1	แสดงแผนภาพการอินเทอร์เฟซแบบขนาน	64
รูปที่ 8.1	แสดงการเปรียบเทียบระหว่างตัวกรองเชิงเลขความถี่ต่ำและตัวกรอง อนาล็อกความถี่ต่ำ	70
รูปที่ 8.2	แสดงรูปแบบลักษณะการออกแบบตัวกรองแบบต่างๆดังนี้(ก) ตัวกรอง ความถี่ต่ำ (ข) ตัวกรองความถี่สูง (ค)ตัวกรองแบบแบนด์พาส (ง)ตัวกรองแบนด์สต็อป	71
รูปที่ 8.3	แสดงผลตอบสนองขนาดกำลังสองของตัวกรองความถี่ต่ำ แบบบัทเทอร์เวิร์ธ	73
รูปที่ 8.4	แสดงการแปลงช่วงความถี่ (a) อนาล็อกเป็นอนาล็อก(b) ดิจิตอลไปสู่ ดิจิตอล	76
รูปที่ 8.5	ความถี่ในการส่งผ่านของการแปลงเชิงเส้นคู่	79
รูปที่ 8.6	แสดงการเชื่อมโยงแบบเชิงเส้นคู่ในระนาบ S และระนาบ Z	80
รูปที่ 9.1	บล็อกไคอะแกรมแสดงโครงสร้างของโครงการงาน	93
รูปที่ 10.1	แสดงหน้าที่ของแต่ละบิตในรีจิสเตอร์ ก) รีจิสเตอร์ข้อมูล ข) รีจิสเตอร์สถานะ ค) รีจิสเตอร์ควบคุม	97
รูปที่ 11.1	แสดงหน้าจอเมื่อเรียกโปรแกรม FIR-IIR	100
รูปที่ 11.2	แสดงโฟลชาร์ทการทำงานของโปรแกรม FIR-IIR	101
รูปที่ 11.3	แสดงโฟลชาร์ทการทำงานของโปรแกรมส่วนการคำนวณค่า สัมประสิทธิ์ของ FIR	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 11.4	แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบไม่ป้อนกลับ	102
รูปที่ 11.5	แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบความถี่ต่ำผ่าน	104
รูปที่ 11.6	แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบแถบความถี่ผ่าน	104
รูปที่ 11.7	แสดงหน้าจอให้เลือกแบบวินโดว์	106
รูปที่ 11.8	แสดงหน้าจอเมื่อผู้ใช้เลือกวินโดว์สี่เหลี่ยม	107
รูปที่ 11.9	แสดงหน้าจอเมื่อผู้ใช้เลือกวินโดว์แบบโคเชอร์	107
รูปที่ 11.10	แสดงวงจรกรองป้อนกลับเชิงเลขโครงสร้างโดยตรง 2 อันดับสอง	
	คาสเคดกัน	109
รูปที่ 11.11	แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบป้อนความถี่ต่ำผ่าน	110
รูปที่ 11.12	แสดงโพลซาร์ทการทำงานของโปรแกรมส่วนคำนวณค่า	
	สัมประสิทธิ์ ของ IIR	111
รูปที่ 11.13	แสดงตำแหน่งจุดทศนิยมของเลขฐานสองรูปแบบ Q15	117
รูปที่ 11.14	แสดงเส้นทางของข้อมูลใน TMS320C50	120
รูปที่ 11.15	โพลซาร์ทแสดงขั้นตอนการหาจำนวนชีพท์ของชีพท์เตอร์ใน	
	TMS320C50	122
รูปที่ 11.16	โพลซาร์ทแสดงการทำงานของโปรแกรมส่วนบอร์ค	124
รูปที่ 11.17	แสดงการแบ่งข้อมูลขนาด 16 บิต เป็นข้อมูล 8 บิต 3 ส่วน ใน	
	การส่งข้อมูล	126
รูปที่ 11.18	โพลซาร์ทแสดงการทำงานทั้งหมดของโครงงานวงจรกรองเชิงเลข	128
รูปที่ 12.1	แสดงสัญญาณในกรณีสัญญาณอินพุตเป็นรูปไซน์ในการทดลอง	
	ที่ 1 ที่ความถี่ 1.414 Khz	131
รูปที่ 12.2	กราฟแสดงค่า $A_v(\text{dB})$ ในกรณีที่สัญญาณอินพุตรูปต่างๆจากการ	
	ทดลองที่ 1	132
รูปที่ 12.3	กราฟแสดงความต่างเฟส ($^{\circ}$) ในกรณีที่สัญญาณอินพุตรูปต่างๆจาก	
	การทดลองที่ 1	132
รูปที่ 12.4	แสดงสัญญาณเมื่อใช้วงจรกรอง (FIR) วินโดว์ฮานที่ความถี่	
	2.237KHz จากการทดลองที่ 2 (50 mV/DIV, 0.2uS/DIV)	136
รูปที่ 12.5	แสดงสัญญาณเมื่อใช้วงจรกรอง (FIR) วินโดว์แฮมมิงที่ความถี่	
	2.237KHz จากการทดลองที่ 2 (50 mV/DIV, 0.2uS/DIV)	137
รูปที่ 12.6	แสดงสัญญาณเมื่อใช้วงจรกรอง (FIR) วินโดว์เบลทแมนที่ความถี่	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.237KHz จากการทดลองที่ 2 (50 mV/DIV,0.2uS/DIV)	137
รูปที่ 12.7 แสดงสัญญาณเมื่อใช้วงจรกรอง (FIR) วินโดว์โคเชอร์ที่ความถี่ 2.237KHz จากการทดลองที่ 2 (50 mV/DIV,0.2uS/DIV)	138
รูปที่ 12.8 กราฟแสดง Loss(dB) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 2	138
รูปที่ 12.9 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 2	139
รูปที่ 12.10 กราฟแสดงค่า Loss(dB) ของวงจรกรองแบบวินโดว์สี่เหลี่ยมที่อันดับต่างกัน	141
รูปที่ 12.11 กราฟแสดงค่า ความต่างเฟส($^{\circ}$) ของวงจรกรองแบบวินโดว์สี่เหลี่ยมที่อันดับต่างกัน	142
รูปที่ 12.12 กราฟแสดงค่า Loss(dB) ของวงจรกรองแบบวินโดว์ฮานที่อันดับต่างกัน	143
รูปที่ 12.13 กราฟแสดงค่า ความต่างเฟส($^{\circ}$) ของวงจรกรองแบบวินโดว์ฮานที่อันดับต่างกัน	144
รูปที่ 12.14 กราฟแสดงค่า Loss(dB) ของวงจรกรองแบบวินโดว์แฮมมิงที่อันดับต่างกัน	145
รูปที่ 12.15 กราฟแสดงค่า ความต่างเฟส($^{\circ}$) ของวงจรกรองแบบวินโดว์แฮมมิงที่อันดับต่างกัน	146
รูปที่ 12.16 กราฟแสดงค่า Loss(dB) ของวงจรกรองแบบวินโดว์เบลกแมนที่อันดับต่างกัน	147
รูปที่ 12.17 กราฟแสดงค่า ความต่างเฟส($^{\circ}$) ของวงจรกรองแบบวินโดว์เบลกแมนที่อันดับต่างกัน	148
รูปที่ 12.18 กราฟแสดง Loss(dB) ของวงจรกรองแบบความถี่สูงแบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 4	151
รูปที่ 12.19 กราฟแสดงความต่างเฟส($^{\circ}$) ของวงจรกรองแบบความถี่สูงแบบไม่ป้อนกลับ(FIR) จากการทดลองที่ 4	151
รูปที่ 12.20 กราฟแสดง Loss(dB) ของวงจรกรองแบบผ่านความถี่แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 5	155

รูปที่ 12.21 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองแบบผ่านความถี่แบบไม่ป้อนกลับ(FIR) จากการทดลองที่ 5	156
รูปที่ 12.22 กราฟแสดง Loss(dB) ของวงจรกรองกำจัดแถบความถี่แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 6	160
รูปที่ 12.23 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองกำจัดแถบความถี่แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 6	160
รูปที่ 12.24 แสดงสัญญาณของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR) ที่ความถี่ 2.237 K hz จากการทดลองที่ 7 (50 mV/DIV, 20 μ S/DIV)	162
รูปที่ 12.25 กราฟแสดง Loss(dB) ของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR)	162
รูปที่ 12.26 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR)	163
รูปที่ 12.27 กราฟแสดง Loss(dB) ของวงจรกรองผ่านความถี่สูงแบบป้อนกลับ (IIR)	164
รูปที่ 12.28 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองผ่านความถี่สูงแบบป้อนกลับ (IIR)	165
รูปที่ 12.29 กราฟแสดง Loss(dB) ของวงจรกรองแถบผ่านความถี่แบบป้อนกลับ (IIR)	166
รูปที่ 12.30 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองแถบผ่านความถี่แบบป้อนกลับ (IIR)	167
รูปที่ 12.31 กราฟแสดง Loss(dB) ของวงจรกรองกำจัดแถบความถี่แบบป้อนกลับ (IIR)	168
รูปที่ 12.32 กราฟแสดง ความต่างเฟส($^{\circ}$) ของวงจรกรองกำจัดแถบความถี่แบบป้อนกลับ (IIR)	169
รูปที่ 12.33 แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยมที่ความถี่ 490 Hz จากการทดลองที่ 11	171
รูปที่ 12.34 แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยมที่ความถี่ 1.727 Khz จากการทดลองที่ 11	171

รูปที่ 12.35	กราฟแสดง Loss(dB) ของวงจรกรอง ผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม	172
รูปที่ 12.36	กราฟแสดง ความต่างเฟส ($^{\circ}$) ของวงจรกรอง ผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม	172
รูปที่ 12.37	แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยมที่มีความถี่ 1.948 KHz จากการทดลองที่ 12	174
รูปที่ 12.38	แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยมที่มีความถี่ 2.459 KHz จากการทดลองที่ 12	175
รูปที่ 12.39	กราฟแสดง Loss(dB) ของวงจรกรอง ผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม	176
รูปที่ 12.40	กราฟแสดง ความต่างเฟส ($^{\circ}$) ของวงจรกรอง ผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม	176

สารบัญตาราง

	หน้า
ตารางที่ 5.1 การเซตบิตรีจิสเตอร์ควบคุมหน่วยความจำเพื่อใช้ หน่วยความจำข้อมูล	44
ตารางที่ 5.2 ตำแหน่งแอดเดรสข้อมูลเพจ 0	44
ตารางที่ 5.3 การกำหนดค่าสำหรับใช้หน่วยความจำโปรแกรม	46
ตารางที่ 5.4 ตำแหน่งของอินเทอร์รัพท์เวกเตอร์	47
ตารางที่ 5.5 ขาสัญญาณของพอร์ตอนุกรม	50
ตารางที่ 5.6 รีจิสเตอร์พอร์ตอนุกรม	50
ตารางที่ 5.7 รีจิสเตอร์ควบคุมไทม์เมอร์ (TCR)	52
ตารางที่ 5.8 แสดงว่าค่าในแต่ละบิตรีจิสเตอร์สถานะรอและแอดเดรสที่เกี่ยวข้อง	53
ตารางที่ 5.9 ค่าพีดส์สถานะรอ และ สถานะรอที่เกี่ยวข้องกับบิตใน CWSR	54
ตารางที่ 5.10 การควบคุมพื้นที่โดย CWSR บิต n	55
ตารางที่ 6.1 แสดงการกำหนดค่า D1 และ D0 ในการทำงาน ในโหมดต่าง ๆ ของ AIC	59
ตารางที่ 6.2 แสดง Secondary DXSerial Communication Protocol	60
ตารางที่ 7.1 รีจิสเตอร์ (Offset 00H)	66
ตารางที่ 7.2 รีจิสเตอร์สถานะ (Offset 01H)	67
ตารางที่ 7.3 รีจิสเตอร์ควบคุม (Offset 02H)	68
ตารางที่ 8.1 การแปลงช่วงความถี่อนาล็อก	76
ตารางที่ 8.2 แสดงการเชื่อมโยงการแปลงเชิงเส้นคู่	78
ตารางที่ 8.3 แสดงตัวแปรความถี่สเกล	81
ตารางที่ 8.4 แสดงค่าสมการอันดับของตัวกรองเชิงเลข	84
ตารางที่ 8.5 แสดง Digital Filter Numerator Coefficients	84
ตารางที่ 8.6 แสดงค่าความถี่พารามิเตอร์เฉพาะในแบนด์พาสและแบนด์สต็อป	90
ตารางที่ 10.1 แสดงหน้าที่ของแต่ละขาของพอร์ทพรีนเตอร์	95
ตารางที่ 12.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p	130
ตารางที่ 12.2 อินพุตสัญญาณรูปสี่เหลี่ยม ขนาด 0.1 Vp-p	131
ตารางที่ 12.3 อินพุตสัญญาณรูปสามเหลี่ยมขนาด 0.1 Vp-p	131
ตารางที่ 12.2.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองความถี่	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านแบบไคเซอร์วินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$	134
ตารางที่ 12.2.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองความถี่ ผ่านแบบวินโดว์สี่เหลี่ยม $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$	134
ตารางที่ 12.2.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองความถี่ ผ่านแบบฮานวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$	135
ตารางที่ 12.2.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองความถี่ ผ่านแบบแฮมมิงวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$	135
ตารางที่ 12.2.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองความถี่ ผ่านแบบแบล็กแมนวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$	136
ตารางที่ 12.3.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำวินโดว์สี่เหลี่ยม $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับของตัวกรอง เท่ากับ 100	140
ตารางที่ 12.3.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำความถี่แบบวินโดว์สี่เหลี่ยม $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับตัวกรอง เท่ากับ 2	141
ตารางที่ 12.3.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำฮานวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับของตัวกรอง เท่ากับ 100	142
ตารางที่ 12.3.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำความถี่แบบฮานวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับตัวกรอง เท่ากับ 2	143
ตารางที่ 12.3.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำแฮมมิงวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับของตัวกรอง เท่ากับ 100	144
ตารางที่ 12.3.6 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำความถี่แบบแฮมมิงวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับตัวกรอง เท่ากับ 2	145
ตารางที่ 12.3.7 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำแบล็กแมนวินโดว์ $F_p = 2 \text{ Khz}$ $F_s = 2.5 \text{ Khz}$ อันดับของตัวกรอง	

เท่ากับ 100	146
ตารางที่ 12.3.8 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำความถี่แบบแบล็กแมนวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$ อันดับ ตัวกรองเท่ากับ 2	147
ตารางที่ 12.4.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ สูงแบบไคเซอร์วินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ อันดับของตัวกรอง เท่ากับ 87	149
ตารางที่ 12.4.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ สูงแบบวินโดว์สี่เหลี่ยม $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ อันดับของตัวกรอง เท่ากับ 87	149
ตารางที่ 12.4.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ สูงแบบฮานวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ อันดับของตัวกรอง เท่ากับ 87	150
ตารางที่ 12.4.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ สูงแบบแฮมมิงวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ อันดับของตัวกรอง เท่ากับ 87	150
ตารางที่ 12.4.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ สูงแบบแบล็กแมนวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ อันดับของ ตัวกรอง เท่ากับ 87	150
ตารางที่ 12.5.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบไคเซอร์วินโดว์ $F_{p1} = 1 \text{ KHz}$ $F_{s1} = 0.5$ $F_{p2} = 2.0 \text{ KHz}$ $F_{s2} = 2.5 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	153
ตารางที่ 12.5.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบวินโดว์สี่เหลี่ยม $F_{p1} = 1 \text{ KHz}$ $F_{s1} = 0.5$ $F_{p2} = 2.0 \text{ KHz}$ $F_{s2} = 2.5 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	153
ตารางที่ 12.5.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบฮานวินโดว์ $F_{p1} = 1 \text{ KHz}$ $F_{s1} = 0.5$ $F_{p2} = 2.0 \text{ KHz}$ $F_{s2} = 2.5 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	154
ตารางที่ 12.5.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบแฮมมิงวินโดว์ $F_{p1} = 1 \text{ KHz}$ $F_{s1} = 0.5$ $F_{p2} = 2.0 \text{ KHz}$	

$F_{s2} = 2.5 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75 ตารางที่ 12.5.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบแบล็กแมนวินโดว์ $F_{p1} = 1 \text{ KHz}, F_{s1} = 0.5 \text{ KHz}$ $F_{p2} = 2.0 \text{ KHz}$ $F_{s2} = 2.5 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	155
ตารางที่ 12.6.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบไคเซอร์วินโดว์ $F_{p1} = 0.5 \text{ KHz}, F_{s1} = 1 \text{ KHz}, F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	157
ตารางที่ 12.6.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบสี่เหลี่ยมวินโดว์ $F_{p1} = 0.5 \text{ KHz}$ $F_{s1} = 1 \text{ KHz}, F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	158
ตารางที่ 12.6.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบฮานวินโดว์ $F_{p1} = 0.5 \text{ KHz}$ $F_{s1} = 1 \text{ KHz}, F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	158
ตารางที่ 12.6.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบแฮมมิงวินโดว์ $F_{p1} = 0.5 \text{ KHz}, F_{s1} = 1 \text{ KHz}, F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	159
ตารางที่ 12.6.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบแบล็กแมนวินโดว์ $F_{p1} = 0.5 \text{ KHz}$ $F_{s1} = 1 \text{ kHz}$ $F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$ อันดับของ ตัวกรองเท่ากับ 75	159
ตารางที่ 12.7 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ต่ำ แบบป้อนกลับ(IIR) $F_p = 2 \text{ KHz}$ $F_{s1} = 2.5 \text{ KHz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$	161
ตารางที่ 12.8 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่สูง แบบป้อนกลับ(IIR) $F_p = 2.5 \text{ KHz}$ $F_{s1} = 2 \text{ KHz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$	164
ตารางที่ 12.9 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองแถบผ่าน ความถี่แบบป้อนกลับ(IIR) $F_{p1} = 1 \text{ KHz}$ $F_{s1} = 0.5 \text{ KHz}$ $F_{p2} = 2 \text{ KHz}$ $F_{s2} = 2.5 \text{ KHz}$ $A_p = 1 \text{ dB}$ $A_s = 35 \text{ dB}$	166
ตารางที่ 12.10 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองก้ำกั้ดแถบ ความถี่แบบป้อนกลับ(IIR) $F_{p1} = 0.5 \text{ KHz}$ $F_{s1} = 1 \text{ KHz}$	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fp2 = 2.5 KHz Fs2 = 2 KHz Ap =1 dB As = 35 dB	168
ตารางที่ 12.11.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำแบบโคเชอร์วินโดว์ Fp = 2 KHz Fs = 2.5 KHz	
Ap =0.5 dB As = 40 dB	170
ตารางที่ 12.11.2 อินพุตสัญญาณรูปสี่เหลี่ยมขนาด 0.1 Vp-p วงจรกรองผ่าน ความถี่ต่ำแบบป้อนกลับ Fp = 2 KHz Fs = 2.5 KHz	
Ap =0.5 dB As = 40 dB	170
ตารางที่ 12.12.1 อินพุตสัญญาณสามเหลี่ยมขนาด 0.1 Vp-p วงจรกรองผ่านความถี่ ต่ำแบบโคเชอร์วินโดว์ Fp = 2 KHz Fs = 2.5 KHz	
Ap =0.5 dB As = 40 dB	173
ตารางที่ 12.12.2 อินพุตสัญญาณรูปสามเหลี่ยมขนาด 0.1 Vp-p วงจรกรองผ่าน ความถี่ต่ำแบบป้อนกลับ Fp = 2 KHz Fs = 2.5 KHz	
Ap =0.5 dB As = 40 dB	174

บทที่ 1

บทนำ

โดยปกติแล้วในการกรองสัญญาณ ซึ่งมักจะเป็นสัญญาณอนาล็อก หรือ สัญญาณเชิงอุปมาน เราจะให้ การกรองเชิงอุปมาน (Analog Filter) ในการประมวลผลสัญญาณ แต่จะพบว่า สัญญาณที่ผ่านกระบวนการกรองในเชิงอุปมานนี้ จะมีสัญญาณรบกวน (noise) และความผิดเพี้ยนของสัญญาณ (distortion) อยู่ ดังนั้น จึงมีการคิดประดิษฐ์วงจรกรองที่มีคุณสมบัติที่ดีกว่าขึ้นมา นั่นก็คือ การกรองเชิงเลข (Digital Filter)

การกรองเชิงเลข เป็นกระบวนการการประมวลผลสัญญาณอันหนึ่งที่มีความสำคัญ โดยที่ การกรองเชิงเลข มักจะเกี่ยวเนื่องกับ การกำจัดสัญญาณรบกวน และ ลดความผิดเพี้ยนของสัญญาณ โดยในปัจจุบันนี้ระบบการกรองเชิงเลขมีบทบาทสำคัญมากขึ้น สาเหตุก็อาจเนื่องมาจาก

1. การที่มีวิธีการออกแบบตัวกรองที่ง่ายแต่ใช้งานได้อย่างมีประสิทธิภาพ
2. พัฒนาการของการออกแบบวงจร วีแอลเอสไอ (VLSI) ที่ทำให้ การคูณตัวเลข การเก็บข้อมูล ของสัญญาณเชิงเลข ทำได้รวดเร็วขึ้นมาก
3. การที่มีคอมพิวเตอร์ฮาร์ดแวร์และซอฟต์แวร์ ที่มีประสิทธิภาพสูง รวมถึงการที่มีระบบไมโครโพรเซสเซอร์ และ ระบบวงจรเชิงเลข ที่มีประสิทธิภาพสูงด้วย

ด้วยเหตุผลเหล่านี้ ทำให้การกรองเชิงเลขถูกนำไปประยุกต์ใช้ในการประมวลผลสัญญาณที่สำคัญ ๆ เช่น การสื่อสารโทรคมนาคม ระบบเรดาร์โซนาร์ หรือ ระบบอากาศยาน เป็นต้น

การที่ตัวกรองเชิงเลขมีการนำประยุกต์ไปใช้งานกันอย่างกว้างขวางนั้น อาจมาจากข้อได้เปรียบหลายประการที่ดีกว่าการกรองในเชิงอุปมาน เช่น ผลตอบสนองความถี่ของตัวกรองเชิงเลข สามารถออกแบบให้มีความใกล้เคียงกับผลตอบสนองความถี่ที่กำหนดให้หรือ ผลตอบสนองความถี่ที่ต้องการได้ นอกจากนี้การออกแบบตัวกรองให้มี ผลตอบสนองเฟสเชิงเส้น ก็ทำได้ง่ายและถ้าพิจารณาในแง่ของ เสถียรภาพของตัวกรองความเชื่อถือได้ ราคา หรือ ขนาดของตัวกรองเชิงเลข สิ่งเหล่านี้กำลังได้รับการพัฒนาและปรับปรุง และมีแนวโน้มว่าจะให้ผลลัพธ์ที่ดีกว่าของ ตัวกรองเชิงอุปมานมาก

อย่างไรก็ตามถ้าพิจารณาในแง่ของการสร้างใช้งาน โดยเฉพาะการสร้างใช้งานโดยใช้เลขฐานสองที่มี ความยาวจำกัด (finite wordlength) นั้น การออกแบบและการสังเคราะห์ตัวกรองเชิงเลขแบบนี้มีความยุ่งยากมากขึ้น เนื่องจากต้องนำผลของปรากฏการณ์ที่ไม่เป็นเชิงเส้นที่อาจเกิดขึ้นได้มาพิจารณาด้วย ซึ่งการใช้ตัวเลข ความยาวของคำจำกัด ทำให้เกิดปรากฏการณ์ที่เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่เป็นเชิงเส้นได้ ตัวกรองเชิงเลขสามารถแบ่งได้ เป็นตัวกรองไม่ป้อนกลับเชิงเลข และ ตัวกรองป้อนกลับเชิงเลข ข้อดีของตัวกรองไม่ป้อนกลับเชิงเลข ก็คือสามารถออกแบบให้มีผล คอบสนองเฟสเป็นเชิงเส้นได้ และมีเสถียรภาพที่ดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

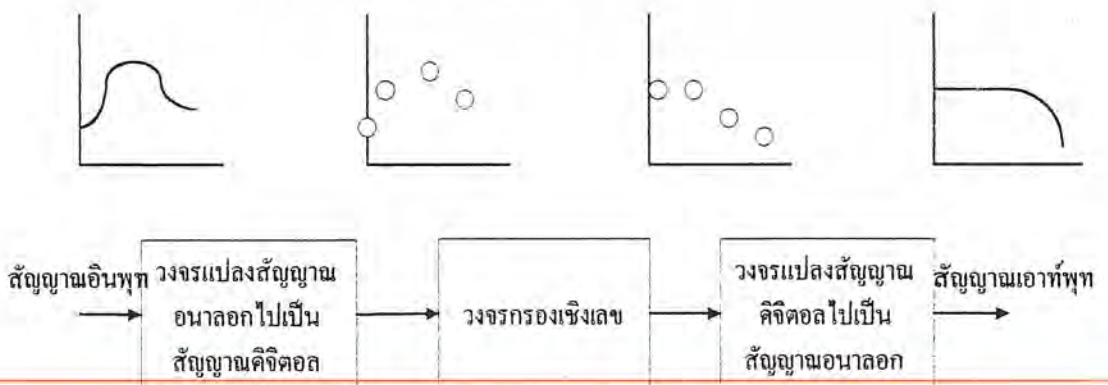
บทที่ 2

ทฤษฎีพื้นฐานของวงจรกรองเชิงเลข

การกรองเชิงเลขเป็นกระบวนการที่ไปดัดแปลงสเปกตรัมของสัญญาณให้มีสเปกตรัมให้เป็นไปตามข้อกำหนดที่ต้องการหรือกำหนดให้ซึ่งอาจเป็นการ เพิ่มค่า หรือ ลดทอน ค่าขนาดของสัญญาณในแถบความถี่ที่กำหนดให้ หรือ ทำการตัดสัญญาณในแถบความถี่ที่กำหนดให้ออกไป หรือทำการแยกสัญญาณในแถบความถี่ที่ต้องการออกมาใช้งานก็ได้ซึ่งกระบวนการเหล่านี้จุดประสงค์หลักก็เพื่อต้องการ

- 1) ลดความผิดเพี้ยนของสัญญาณ หรือ ลดสิ่งรบกวนที่อาจเป็นผลมาจากการส่งสัญญาณไปใน สายส่ง หรือ ตัวกลาง ที่มีคุณภาพไม่สมบูรณ์ หรือ มีสิ่งรบกวนมาก
- 2) ต้องการแยกสัญญาณออกจากสัญญาณผสมกันมาหลายสัญญาณที่ถูกส่งมาพร้อมกันบนสายส่งเส้นเดียวกัน
- 3) ใช้ในการแยกแยะสัญญาณใน โดเมนเวลาออกเป็นองค์ประกอบของสัญญาณใน โดเมนความถี่
- 4) ใช้ในการคีมอดูเลตสัญญาณ (demodulation)
- 5) ใช้ในการแปลงสัญญาณเชิงเต็มหน่วยออกเป็นองค์ประกอบทางความถี่

วงจรกรองเชิงเลขสามารถเขียนอธิบายในรูปของบล็อกไดอะแกรมได้ดังรูป 2.1 โดยสัญญาณอินพุตซึ่งเป็นสัญญาณอนาลอกจะถูกสุ่ม (sampled) ด้วยช่วงเวลาที่ยกที่กำหนด และสัญญาณที่ถูกสุ่มนี้จะถูกเปลี่ยนให้อยู่ในรูปเลขฐานสองโดยวงจรแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิทัลหรือสัญญาณเชิงเลข (analog to digital converter)



รูปที่ 2.1 บล็อกไดอะแกรมของวงจรกรองเชิงเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเลขฐานสองที่แทนสัญญาณอนาลอกที่เข้ามาทางอินพุตจะถูกกรองโดยวงจรกรองเชิงเลข (Digital Filter) การกรอง (filter) จะเป็นการคำนวณทางตัวเลข ซึ่งจะอาศัยวงจรที่ใช้เหมือนกับที่ใช้ในระบบคอมพิวเตอร์ ได้แก่ ตัวบวก (adder) ; ตัวคูณ (multiplier) ; รีจิสเตอร์เลื่อน (shift register) และ อุปกรณ์หน่วยความจำ (memory devices) ค่าเอาพุตที่ได้จากวงจรกรองเชิงเลขนี้จะถูกแปลงกลับเป็นสัญญาณอนาลอกอีกทีหนึ่งเป็นสัญญาณเอาพุตที่นำไปใช้งานได้ วงจรกรองเชิงเลขสามารถแบ่งได้เป็น 2 ประเภท ตามลักษณะผลตอบสนองอิมพัลส์ ได้แก่

1. วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด (Finite Impulse Response Filter :FIR)
2. วงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด(Infinite Impulse Response Filter : IIR)

วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด มักเป็นวงจรที่ไม่มีการป้อนกลับเป็นวงจรกรองที่มีโครงสร้างง่าย ๆ และมีเสถียรภาพที่ดี แต่มีข้อเสียที่จะต้องใช่วงจรกรองที่มีอันดับสูงถึงแม้จะต้องการให้มีลักษณะทางความถี่ที่ง่าย ๆ ก็ตาม

ส่วนวงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัดเป็นวงจรที่มีการป้อนกลับ เป็นวงจรกรองที่ใช้อันดับต่ำกว่าวงจรแบบผลตอบสนองอิมพัลส์จำกัดที่ความต้องการลักษณะทางความถี่เหมือนกัน แต่การออกแบบจะยุ่งยากกว่า และมีปัญหาในเรื่องความมีเสถียรภาพที่ไม่ดีนัก

2.1 วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด (FIR FILTER)

วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัดจะมีผลตอบสนองอิมพัลส์ที่มีจำนวนคงที่ค่าหนึ่ง โดยผลตอบสนองอิมพัลส์ของ วงจรกรองแบบผลตอบสนองอิมพัลส์จำกัดอันดับที่ N (N -th-order FIR Filter) สามารถแสดงได้เป็น

$$\{ h[n] \} = \{ h_0, h_1, \dots, h_{N-1} \} \quad (2.1)$$

จากสมการที่ (2.1) เราสามารถแสดงผลตอบสนองอิมพัลส์นี้ได้ในรูปอนุกรมของผลตอบสนอง ในโดเมนเวลา ของสัญญาณอินพุตหนึ่งหน่วย $u[n]$ ในลักษณะผลบวกเชิงประสาน

(convolution sum) ดังสมการที่ 2.2

$$y[n] = \sum_{i=0}^{N-1} h_i u[n-i] \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อประยุกต์ใช้การแปลงแซต (Z-transform) ซึ่งนิยามโดย

$$Y(z) = \sum_{n=-\infty}^{\infty} Y_n z^{-n} \quad (2.3)$$

จะได้ฟังก์ชันถ่ายโอน (transfer function) ตามสมการที่ 2.4

$$H(z) = \sum_{i=0}^{N-1} h_i z^{-i} \quad (2.4)$$

ถ้าเราให้สัญญาณเข้าในโดเมนของเวลาเป็น $x(n)$ และสัญญาณออกเป็น $y(n)$ สมการที่ 2.1-2.4 จะสามารถเขียนได้ใหม่ตามสมการที่ 2.5

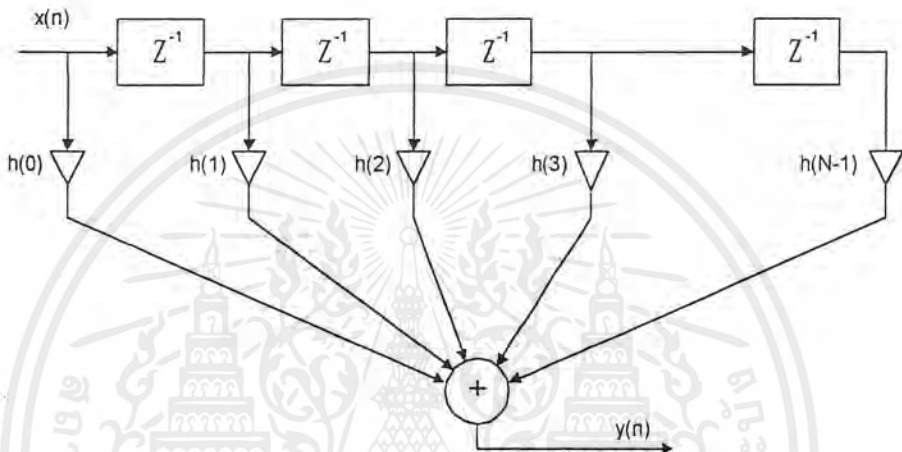
$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^{N-1} h_i z^{-i} \quad (2.5)$$

โดย $Y(z)$ เป็นสัญญาณออกในโดเมนของความถี่ และ $X(z)$ เป็นสัญญาณเข้าในโดเมนของความถี่เช่นกัน

จากสมการข้างต้นจะเห็นว่า ตัวกรองแบบนี้มีฟังก์ชันถ่ายโอนที่มีแต่เฉพาะอย่างเดียว เพราะเป็นตัวกรองที่ไม่มีการป้อนกลับของสัญญาณออก มาใช้ในการคำนวณสัญญาณออกลำดับถัดไป ดังนั้นลำดับสัญญาณออก $y(n)$ จึงขึ้นอยู่กับลำดับสัญญาณเข้า $x(n)$ เท่านั้น และทำให้การออกแบบวงจรกรองไม่ป้อนกลับเชิงเลข จำทำได้โดยไม่ต้องคำนึงถึงเสถียรภาพของตัวกรองเลย เพราะตัวกรองแบบนี้จะมีความเสถียรเสมอ

โครงสร้างของวงจรกรองแบบผลตอบสนองอิมพัลส์จำกัดนี้สามารถแสดงได้ดังรูปที่ 2.2 ซึ่งเป็นโครงสร้างแบบโดยตรง (Direct Form) นอกจากโครงสร้างที่แสดงในรูปที่ 2.2 แล้ว เรายังสามารถแสดงโครงสร้างของตัวกรองแบบนี้ได้อีกหลายรูปแบบ ลักษณะของรูปแบบขึ้นอยู่กับวิธีการเลือกใช้ ระเบียบวิธีการทำให้เป็นจริง (realization method) ซึ่งเป็น กระบวนการหรือ กรรมวิธี ในการแปลงฟังก์ชันถ่ายโอนให้เป็น โครงข่าย หรือ โครงสร้างวงจร ส่วนการจะเลือกนำโครงสร้างแบบใดไปใช้สร้างตัวกรองนั้น ไม่สามารถกำหนดลงไปอย่างแน่นอนได้ ทั้งนี้เนื่องจากมีองค์ประกอบหลายประการที่ต้องนำมาพิจารณาด้วย เช่น ความเร็วในการคำนวณ สภาพไวต่อการควอนไทซ์สัมประสิทธิ์ (sensitivity to coefficient quantization) ระดับของสัญญาณรบกวนที่จุดสัญญาณออกจากการควอนไทซ์ซึ่งอาจเป็นการ ปิดเศษ (rounded off) หรือ การตัดปลาย (chopped off) ของผลการคำนวณคณิตศาสตร์ หรือ ปรากฏการณ์ไม่เป็นเชิงเส้นที่อาจเกิดขึ้นได้ เป็นต้น

ในที่นี้เมื่อกล่าวถึงคำว่า โครงสร้าง (structure) เราจะเขียนแทนด้วยแผนภาพที่ประกอบขึ้นด้วย ตัวบวกสัญญาณ ตัวคูณสัญญาณ ตัวหน่วงสัญญาณ ส่วนทิศทางการไหลของสัญญาณจะแทนด้วยทิศทางของหัวลูกศร (ดังในรูปที่ 2.2 และ 2.3) ซึ่งแผนภาพที่ได้นี้แทน การคำนวณที่อาจสร้างเป็น วงจรเฉพาะ หรือ ฮาร์ดแวร์ หรือ เขียนโปรแกรม หรือ เป็นส่วนผสมระหว่างวงจรและโปรแกรมก็ได้



รูปที่ 2.2 แสดงโครงสร้างของตัวกรองไม่ป้อนกลับเชิงเลขแบบโดยตรง

วิธีการเขียนโครงสร้างของวงจรกรอง แบบโดยตรง นี้ทำได้โดยการสร้างจากสมการผลต่างสลับเนื่องโดยตรง คือ

$$\begin{aligned}
 y(n) &= h(0).x(n) + h(1).x(n-1) + \dots + h(N-1).x(n-N) \\
 &= \sum_{i=0}^{N-1} h_i x(n-i)
 \end{aligned}
 \quad 2-6$$

ในการออกแบบวงจรกรองแบบผลตอบสนองอิมพัลส์จำกัดนี้ไม่มีวิธีการออกแบบได้หลายวิธีวิธีหนึ่งที่นิยมใช้ในการออกแบบคือวิธีออกแบบโดยใช้วิน โคว์ ซึ่งจะกล่าวถึงรายละเอียดในภายหลัง

2.2 วงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด (IIR FILTER)

วงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด เป็นตัวกรองที่ผลตอบสนองอิมพัลส์ที่มีค่าไปอย่างไม่มีสิ้นสุด มักจะเป็นตัวกรองแบบที่มีการป้อนกลับ โดยสมการผลต่างสืบเนื่องอันดับที่ M (M th-order difference equation) ได้ดังนี้

$$y(nT) = \sum_{i=0}^N a_i x(nT-iT) - \sum_{i=1}^M b_i y(nT-iT) \quad (2.7)$$

โดยที่ $b_0 = 1$ สัญญาณเข้าในโดเมนของเวลาเป็น $x(nT)$ และสัญญาณออกเป็น $y(nT)$ และเมื่อทำการแปลงแซตทั้งสองข้างจะได้ฟังก์ชันถ่ายโอนเป็น

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^N a_i z^{-i} / \{ 1 + \sum_{i=1}^M b_i z^{-i} \} \quad (2.8)$$

โดย $Y(z)$ เป็นสัญญาณออกในโดเมนของความถี่ และ $X(z)$ เป็นสัญญาณเข้าในโดเมนของความถี่

จะเห็นว่าฟังก์ชันถ่ายโอนของวงจรกรองเชิงเลขแบบนี้เขียนอยู่ในพจน์ของ โพล และ ซีโร ได้ ดังนั้นเราควรจะพิจารณาเสถียรภาพของวงจรกรองแบบนี้ โดยการดูที่ตำแหน่งของโพล ซึ่งฟังก์ชันถ่ายโอนที่เขียนอยู่ในพจน์ของโพล และ ซีโร ได้ดังสมการที่ 2-7

$$H(z) = A \prod_{i=1}^N (1 - z_i Z^{-1}) / \prod_{i=1}^M (1 - p_i Z^{-1}) \quad (2.9)$$

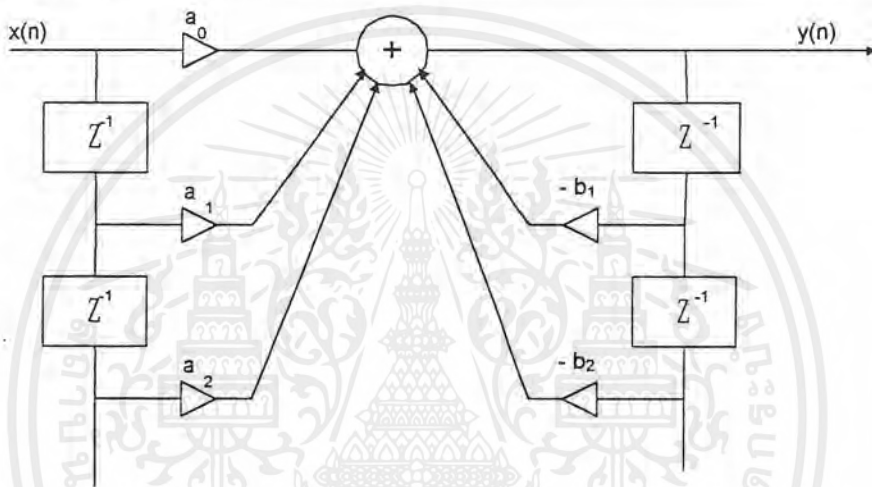
โดยที่ A เป็นค่าคงที่ แทนค่าขยายของระบบ

ซึ่งจะเห็นว่าวงจรกรองป้อนกลับเชิงเลขนี้จะเสถียรก็ต่อเมื่อโพลของฟังก์ชันถ่ายโอน วางตัวอยู่ภายในวงกลมหนึ่งหน่วยของระนาบเชิงซ้อน Z

ในการออกแบบตัวกรองป้อนกลับเชิงเลข ซึ่งมีแนวทางการออกแบบที่แตกต่างไปจากวิธีการออกแบบตัวกรองแบบไม่ป้อนกลับเชิงเลข และมีการออกแบบที่ค่อนข้างยุ่งยากกว่า ทั้งนี้เนื่องมาจากตัวกรองแบบนี้มีการนำเอาสัญญาณออกมาใช้ในการคำนวณหาสัญญาณออกลำดับถัดไป หรือ มีการป้อนกลับสัญญาณ ในการออกแบบจึงต้องคำนึงถึงเสถียรภาพของตัวกรองด้วย โดยทั่วไปมีวิธีการออกแบบตัวกรองป้อนกลับเชิงเลข 2 วิธี คือ วิธีการออกแบบโดยทำการแปลง จากฟังก์ชันถ่ายโอนของตัวกรองเชิงอุปมานที่มีเสถียรภาพดี วิธีการนี้ถ้าหากเรามีการแปลงที่ดี การออกแบบก็ไม่ต้องคำนึงถึงเสถียรภาพของตัวกรองอีก โดยที่ได้ผลตอบสนองแอมพลิจูด

ตามต้องการ วิธีนี้เหมาะสำหรับผู้ที่มีความรู้ หรือ เชี่ยวชาญการออกแบบตัวกรองเชิงอุปมานดีแล้ว ส่วนวิธีที่สองเป็นการออกแบบตัวกรองป้อนกลับเชิงเลขในโดเมนแซตโดยตรง ซึ่งวิธีนี้ยังรวมถึงการออกแบบโดยใช้คอมพิวเตอร์ช่วยด้วย

โครงสร้างของวงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัดนี้สามารถแสดงได้ดังรูปที่ 2.3 ซึ่งเป็นโครงสร้างแบบโดยตรง (Direct Form) เช่นเดียวกับวงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด นอกจากโครงสร้างที่แสดงในรูปที่ 2.3 แล้ว เรายังสามารถแสดงโครงสร้างของตัวกรองแบบนี้ได้อีกหลายแบบ



รูปที่ 2.3 แสดงโครงสร้างของตัวกรองป้อนกลับเชิงเลขแบบโดยตรง

วิธีการเขียนโครงสร้างของวงจรกรอง แบบโดยตรง นี้ทำได้โดยการสร้างจากสมการผลต่างสปีเนื่องโดยตรง จากสมการที่ 2.6 เช่นเดียวกับการเขียนโครงสร้างแบบโดยตรงของวงจรกรองที่มีผลตอบสนองอิมพัลส์จำกัด

2.3 เปรียบเทียบวงจรกรองแบบผลตอบสนองอิมพัลส์จำกัด

กับวงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัด

ในหัวข้อนี้เป็นการเปรียบเทียบข้อดี และ ข้อด้อยของวงจรทั้งสองแบบ โดยเป็นการเปรียบเทียบที่ตัวกรองอันดับเดียวกัน

ตัวกรองป้อนกลับเชิงเลข	ตัวกรองไม่ป้อนกลับเชิงเลข
1. อาจเกิดการไม่เสถียรได้	1. มีความเสถียรเสมอ
2. ให้ช่วงความถี่ที่ผลตอบสนองความถี่จากผ่านสัญญาณไปเป็นไม่ผ่านสัญญาณ หรือ ทรานซิชันแบนด์แคบกว่า	2. ให้ช่วงทรานซิชันแบนด์กว้างกว่า
3. ค่าการหน่วงสัญญาณ ที่จุดออกของสัญญาณ (overall delay) น้อยกว่า จึงเหมาะสำหรับใช้กับระบบเวลาจริง	3. ค่าการหน่วงออกไปของสัญญาณ ที่จุดสัญญาณออกมากกว่ามาก จึงไม่เหมาะกับระบบเวลาจริง
4. เกิดผลของ ความผิดเพี้ยนของผลตอบสนองเฟส (phase distortion) สูง	4. สามารถออกแบบให้มีผลตอบสนองเฟสเป็นแบบเชิงเส้นได้
5. ผลของสัญญาณรบกวน เนื่องมาจากการปัดเศษ (round-off error) มีค่ามาก	5. สัญญาณรบกวนจากการปัดเศษมีค่าน้อย
6. การออกแบบง่าย โดยเฉพาะการออกแบบโดยใช้วิธีการแปลงมาจากตัวกรองเชิงอุปมาน	6. การออกแบบให้มีผลตอบสนองแอมพลิจูดคมทำได้ยากกว่า

ในข้อ 1 นั้น เนื่องจาก ตัวกรองไม่ป้อนกลับ เป็นระบบที่มีเฉพาะ ซีโร หรืออาจเรียกว่าเป็น ตัวกรองมีเฉพาะซีโร (all zeroes filter) ดังนั้นจึงเสถียรเสมอ ผิดกับตัวกรองป้อนกลับที่เป็นระบบที่มีทั้ง โพล และ ซีโร เมื่อนำตัวกรองไปสร้างใช้งาน โดยการแทนค่าสัมประสิทธิ์ของตัวกรองด้วยเลขฐานสองที่มีจำนวนบิตจำกัด ทำให้ค่าสัมประสิทธิ์คลาดเคลื่อนไป ตำแหน่งของโพลอาจเคลื่อนไปในตำแหน่งที่ทำให้ตัวกรองไม่เสถียรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนในข้อ 2 นั้น เนื่องจากตัวกรองป้อนกลับทำการประมาณค่า ผลตอบสนองความถี่อุดมคติ โดยใช้อัตราส่วนของพหุนามสองพหุนาม หรือ แบบครกยะ ส่วนตัวกรองแบบไม่ป้อนกลับ ทำการประมาณค่า โดยใช้พหุนามธรรมดา และโดยทั่วไปการถือว่าตัวกรองเป็นตัวกรองอันดับเท่าใด เราจะดูจากกำลังสูงสุดของพหุนาม ตัวอย่างเช่น ตัวกรองอันดับ 2 ตัวกรองป้อนกลับประมาณค่าโดยใช้ พหุนามอันดับ 2 สองพหุนาม มาหารกัน ส่วนตัวกรองไม่ป้อนกลับ ใช้พหุนามอันดับ 2 อย่างเดียว ดังนั้นเมื่อเปรียบเทียบตัวกรองในอันดับเดียวกันตัวกรองแบบป้อนกลับจึงทำการประมาณค่าได้แม่นยำกว่า หรือได้ผลตอบสนองแอมพลิจูดได้คมกว่า นั่นก็คือ ช่วงความถี่ แถบเปลี่ยนสถานะ หรือ ทรานซิชันแบนด์ (Transition Band) หรือ ช่วงความถี่ที่ผลตอบสนองแอมพลิจูดเปลี่ยนจากผ่านสัญญาณเป็นไม่ผ่านสัญญาณ ของตัวกรองแบบป้อนกลับ จึงแคบกว่า

ส่วนข้อ 3 เป็นผลโดยตรงมาจากข้อ 2 สำหรับตัวกรองที่ให้ผลตอบสนองแอมพลิจูดอย่างเดียวกัน เราสามารถใช้ตัวกรองป้อนกลับที่มีอันดับต่ำกว่า เมื่อใช้อันดับน้อยกว่าโดยทั่วไปในการสร้างตัวกรอง ก็คือการอุปกรณ์หน่วยสัญญาณจำนวนน้อยกว่าด้วยข้อเปรียบเทียบข้อ 4 ตัวกรองป้อนกลับให้ผลตอบสนองเฟสที่ไม่เป็นเชิงเส้นการทำให้ตัวกรองแบบนี้มีผลตอบสนองเฟสเชิงเส้น อาจทำได้โดยในคอนประมวลผล เราทำการประมวลผล 2 ครั้ง ครั้งแรกทำการประมวลผลข้อมูลตามปกติ ส่วนครั้งที่สองทำการประมวลผลในทิศทางย้อนกลับ ซึ่งการทำอย่างนี้ไม่ใช่เป็นการประมวลผลในระบบเวลาจริง

2.4 การออกแบบวงจรกรองเชิงเลข

การที่จะออกแบบให้ตัวกรองมีคุณสมบัติตามต้องการนั้น ในขั้นแรกเราต้องกำหนดคุณสมบัติของตัวกรองขึ้นมา การกำหนดคุณสมบัติอาจอยู่ในรูป ผลตอบสนองความถี่ ซึ่งเป็นคุณสมบัติในโดเมนความถี่ หรือ อาจกำหนดในรูปผลตอบสนองอิมพัลส์ ซึ่งเป็นคุณสมบัติในโดเมนเวลา แล้วจากผลตอบสนองของระบบที่กำหนดขึ้นนี้เราใช้เป็นข้อกำหนดในการออกแบบ ซึ่งก็มีวิธีการออกแบบตัวกรองได้หลายวิธีการอย่างไรก็ตามทุกวิธีการออกแบบก็จะได้ผลลัพธ์เป็นฟังก์ชันถ่ายโอน และฟังก์ชันถ่ายโอนนี้จะเป็นฟังก์ชันที่ให้คุณสมบัติการกรองตามต้องการ

ดังนั้นสำหรับการออกแบบสร้างตัวกรองเชิงเลข โดยให้มีคุณสมบัติ ผลตอบสนองความถี่ตามอุดมคติ (ideal response) หรือ มีผลตอบสนองความถี่ตามต้องการ นั้น อาจกล่าวโดยทั่วไปได้ว่าผู้ออกแบบและสร้างต้องนำองค์ประกอบสำคัญต่าง ๆ มาร่วมพิจารณาดู ซึ่งเขียนสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดคุณสมบัติของตัวกรอง ในโดเมนความถี่ เช่น ค่าขนาด หรือ ค่าความหน่วงกลุ่ม หรือว่า เป็นการกำหนดคุณสมบัติของตัวกรองในโดเมนเวลา เช่น ผลตอบสนองอิมพัลส์
2. สำหรับตัวกรองป้อนกลับเชิงเลข ที่เราเขียนฟังก์ชันถ่ายโอนของมัน เป็นอัตราส่วนของ พหุนาม (polynomial) สองพหุนาม
3. เมื่อได้ฟังก์ชันถ่ายโอนแล้ว เราต้องทำการเลือกโครงสร้างตัวกรองให้เหมาะสม เพื่อที่ว่าเมื่อนำไปสร้างใช้งานโดยใช้ตัวเลขความยาวจำกัดแล้ว ไม่เกิดปรากฏการณ์ไม่เป็นเชิงเส้นขึ้น หรือ ถ้าหลีกเลี่ยงไม่ได้ก็ให้เกิดปรากฏการณ์ไม่เป็นเชิงเส้นขึ้นน้อยที่สุด



บทที่ 3

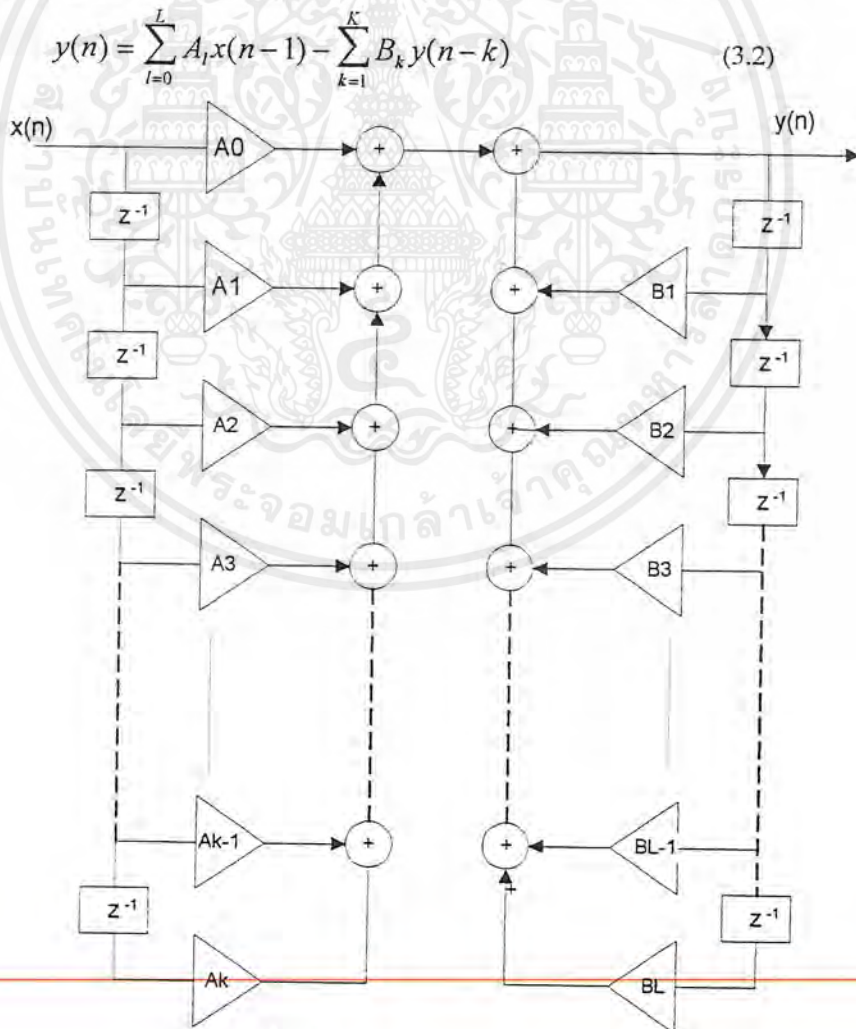
การออกแบบวงจรกรองเชิงเลข

3.1 ตัวกรองเชิงเลข

สำหรับ ตัวกรองดิจิทัล สามารถอธิบายในเชิงกว้างๆ ได้โดยอาศัย การเชิงเส้น, สัมประสิทธิ์ค่าคงที่, สมการผลต่างดังนี้

$$\sum_{k=0}^K D_k y(n-k) = \sum_{l=1}^L C_l x(n-l) \quad (3.1)$$

ผลรวมของค่า C_l และ D_k ถูกกำหนดไว้ที่ใดในในระบบ และค่า $y(n)$ และ $x(n)$ เป็นตัวใช้แสดง output และ input ตามลำดับ ถึงแม้จะเป็นนิพจน์โดยทั่วไปก็ตาม มีเพียงระบบ causal เท่านั้นที่จะใช้ทั้ง $y(n)$ และ $x(n)$ ให้เท่ากับ 0 เมื่อ $n < 0$ สมการ 3.1 ใช้เพื่อแสดงให้อยู่ในรูปของเทอม output และสัญญาณ input และ output ในอดีต



รูปที่ 3.1 แสดงสมการค่าสัมประสิทธิ์คงที่เชิงเส้นรูปแบบตรงของสมการที่ 3.2

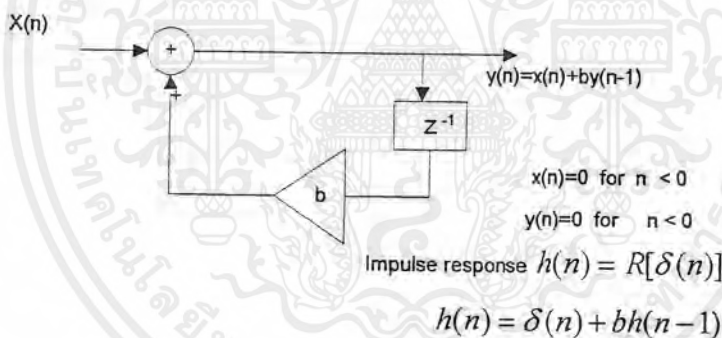
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิพจน์เหล่านี้จะถูกทำให้สมบูรณ์ประกอบไปด้วยการคูณ การบวก และการหน่วงเวลารูปที่ 3.1 แสดงให้เห็นรูปแบบตรง จำนวนของการออกแบบ ดิจิตอล ฟิวเตอร์ มุ่งหาสัมประสิทธิ์ของ สมการ 3.2 เพื่อค้นหาค่าวิกฤต สำหรับความสัมพันธ์ input output digital filter ที่สำคัญมีอยู่ 2 ชนิดคือ IIR และ FIR

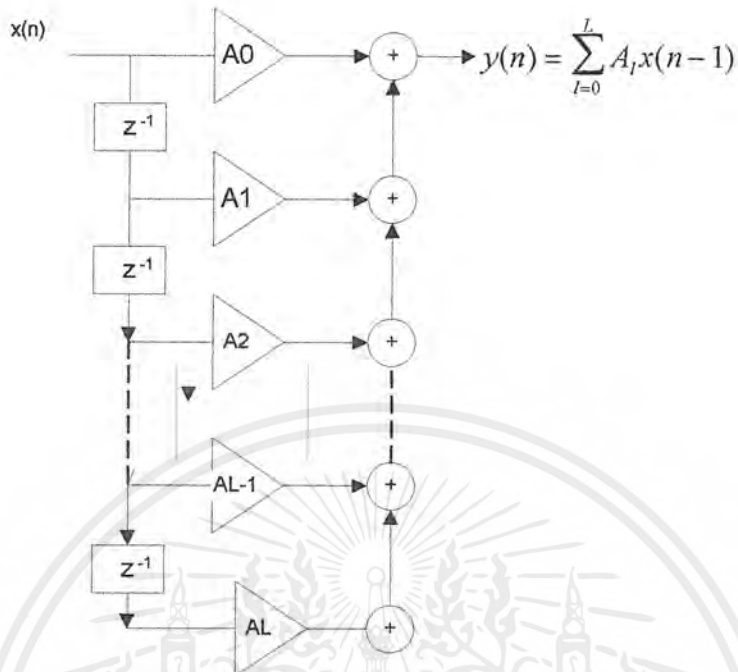
1. **IIR** (Infinite Impulse Respose) ผลตอบสนองของ IIR เป็นฟังก์ชันของค่าปัจจุบัน และอดีตของสัญญาณการสุ่มของ Input และ Output สมการทั่วไปแสดงไว้ในสมการที่ 3.2 การ ที่สัญญาณต่างขึ้นอยู่กับ Output ในอดีตจะทำให้การช่วงขึ้นไปสู่ช่วงอนันต์ของผลตอบสนอง Output ตัวกรองเมื่อค่า Input ได้หยุดลงเช่นสมการ $y(n) = x(n) + by(n-1)$ เป็นสมการผลต่าง อันดับที่หนึ่ง ทำให้เป็นค่าที่อยู่ในรูปของผลตอบอัมพลิตี้ได้ดังนี้

$$h(n) = R[\delta(n)] = b^n u(n)$$

การคำนวณแสดงได้ไว้ในรูปที่ 3.2 จะเห็นว่า เทอมย้อนกลับ (Feedback Term) ถูกคูณด้วยค่า b และรวมค่า Input ผลลัพธ์จะอยู่ในระบบ Output ระบบของการต้านทานไม่จำกัด (Infinite Duration)



(a) ผลตอบสนองอิมพัลส์หนึ่งหน่วยของตัวกรอง IIR อันดับที่หนึ่ง



(b) ผลตอบสนองตัวกรองหนึ่งหน่วยของตัวกรอง FIR อันดับ $L+1$

รูปที่ 3.2 แสดงผลตอบสนองตัวกรองหนึ่งหน่วยของตัวกรอง FIR และ IIR

3.2 ตัวกรองที่มีผลตอบสนองจำกัด (FIR : Finite Impulse response)

การสุ่ม Output ของระบบขึ้นอยู่กับ Input ปัจจุบันอย่างเดียวและเป็นค่าที่จำกัดในอดีต จะเป็นตัวกรองที่มีรูปแบบผลตอบสนองจำกัดสมการที่ 2.21 ค่า $B_k = 0$ แสดงตัวกรอง FIR ที่ค่า ด้านทาน L สมการ

$$y(n) = \sum_{l=0}^L A_l x(n-l)$$

สัมประสิทธิ์ของ FIR มีค่าเท่ากับผลตอบสนองอิมพัลส์ของตัวกรอง

3.3 ความสัมพันธ์ของระบบการแปลง ฟูรีเยอร์

สเปกตรัมความถี่ของสัญญาณขึ้นอยู่กับ โดเมนของเวลาที่แสดงถึงระบบเชิงเส้นที่มีค่าเวลาเป็น ช่วงผลตอบสนอง ฟูรีเยอร์ ของสัญญาณ Input ที่ใช้ใน ระบบการปรับเปลี่ยนการใช้งาน เพราะว่า Output ของระบบนั้นเป็นอีก โพรเนนเชียลเชิงซ้อน ซึ่งจะนำไปสู่ผลตอบสนองความถี่-Intermotion system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบในโดเมนเวลาแสดงไขผลบวก Convolution เป็นวิธีสำหรับระบบคำนวณสำหรับใช้หาผลตอบสนองอิมพัลส์ขณะใดขณะหนึ่ง เหมือนกับว่าในระบบในโดเมนความถี่ Output ของระบบผลิตผลตอบสนองความถี่ของระบบพร้อมกับผลตอบสนองความถี่ของสัญญาณ Input ด้วยวิธีนี้จะทำให้ได้ค่าที่เท่ากันในระหว่างการ Convolution ของสัญญาณใน โดเมนของเวลา และรูปแบบการย้อนกลับ พูรีเยร์ออกมา ถึงแม้ว่าผลลัพธ์ของสัญญาณสองสัญญาณในโดเมนของเวลาเท่ากับสองสัญญาณการแปลงของการแปลงฟูรีเยร์ของการconvolution ความสัมพันธ์ของการแปลงนี้ใช้ในการวิเคราะห์กับระบบที่มีการต้องการ

3.4 ผลตอบสนองความถี่ระบบ (System Frequency Response)

ผลตอบสนองความถี่ของระบบขึ้นอยู่กับความสัมพันธ์ของอินพุตเอาต์พุตสำหรับชุดของอินพุตค่าไซน์ที่ซับซ้อนทั้งหมดที่เป็นไปได้

$$y(n) = R[A_1 e^{j(2\pi f_1 nT)}] \quad (3.3)$$

สัญญาณนั้นขึ้นกับค่าความถี่ f_1 และ ขนาดแอมปริจูด A_1 ซึ่งจะให้เป็นตัวแสดงอินพุตค่าไซน์ที่ซับซ้อนทั้งหมดที่เป็นไปได้ ผลบวกการประสานสำหรับอินพุตของระบบสัญญาณไซน์

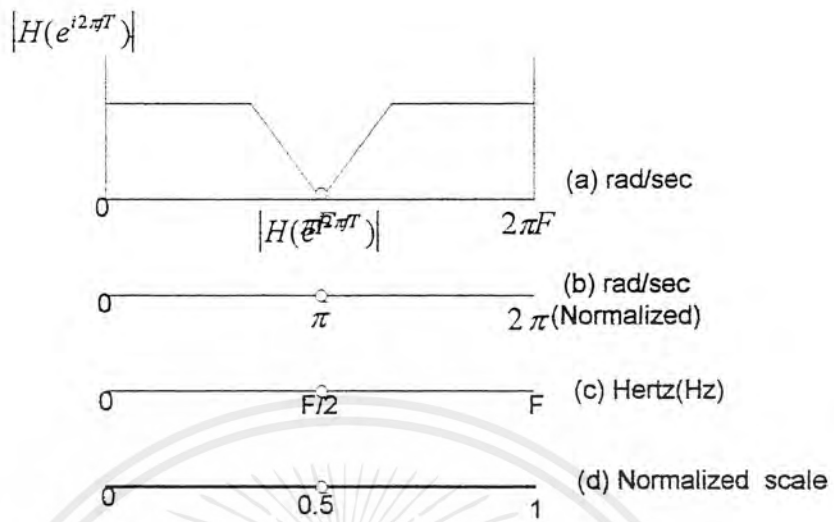
$$y(n) = \sum_{k=-\infty}^{\infty} A_1 h(k) e^{j(2\pi f_1 (n-k)T)} \quad (3.4)$$

สัญญาณนั้นขึ้นกับค่าความถี่ f_1 และ ขนาดแอมปริจูด A_1 ซึ่งจะให้เป็นตัวแสดงอินพุตค่าไซน์ที่ซับซ้อนทั้งหมดที่เป็นไปได้ ผลบวกการประสานสำหรับอินพุตของระบบสัญญาณไซน์

$$y(n) = A_1 e^{j(2\pi f_1 nT)} \sum_{k=-\infty}^{\infty} h(k) e^{-j(2\pi f_1 kT)} \quad (3.5)$$

ส่วนแรกเป็นการบ่งชี้สัญญาณอินพุต ในส่วนที่ 2 ซึ่งเป็นผลลัพธ์จำนวนครั้งในการคูณสัญญาณอินพุตผลบวกของตัวนี้เรียกว่าผลตอบสนองความถี่ระบบ(System Frequency Response) ดังนั้นสัญญาณไซน์ ที่ซับซ้อนนำไปสู่ระบบเวลาขึ้นยงเชิงเส้น เอาต์พุตเป็นรูปแบบไซน์ของความถี่เดียวกัน ได้ขยายโดยมีค่าเป็น

$$H(f_1) = \sum_{k=-\infty}^{\infty} h(k) e^{-j(2\pi f_1 kT)} \quad (3.6)$$



รูปที่ 3.3 แสดงความถี่สมมูลบนแกนสเกลหนึ่งหน่วย

3.5. ความหมายของโครงข่าย (network) และการทำงาน

พื้นฐานการทำงานอย่าง ได้แก่ การบวก การคูณ และการหน่วงเวลา จะได้นำมาใช้ในการสร้างระบบขึ้น พิจารณาคำตอบทั่วไปของผลตอบสนองของระบบ $y(nT)$ ในเทอมของสัญญาณเข้าและ network เราจะเขียนผลตอบสนองของระบบได้ดังนี้

$$Y(z) = X(z)H(z) \quad (3.7)$$

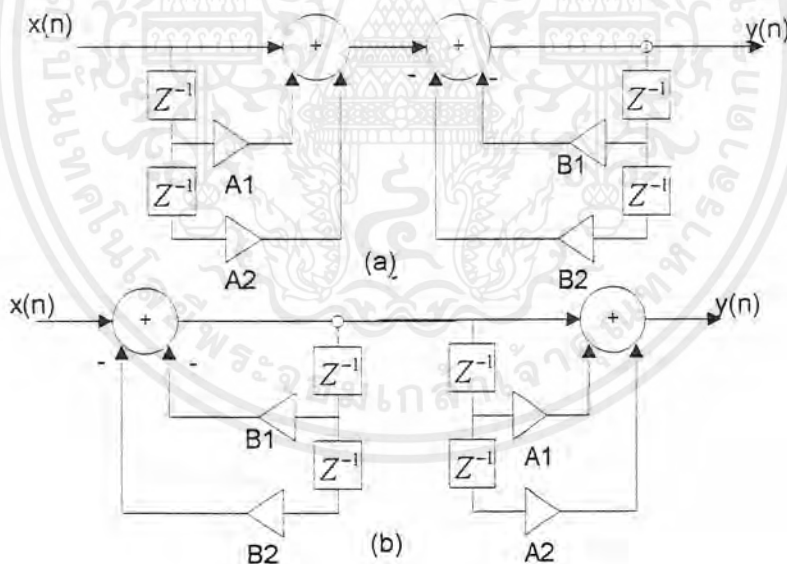
ดังนั้น ผลการแปลงแซดของ $Y(z)$ คือ ผลคูณของการแปลงแซดของฟังก์ชันตัวกระตุ้น $X(z)$ กับ ฟังก์ชันถ่ายโอน $H(z)$ ตัวอย่าง เช่น สมการผลต่างอันดับ 1 ถูกกำหนดโดย

$$y(n) = x(n) + A_1 x(n-1) + A_2 x(n-2) - B_1 y(n-1) - B_2 y(n-2)$$

จงเขียนวงจร โครงสร้างและฟังก์ชันตัวกรองถ่ายโอน

เริ่มแรกเขียนโครงข่ายเชิงเลขได้ดังรูปที่ 3.4a โดยใช้โครงสร้างแบบโดยตรง อีกรูปแบบดังรูป 3.4b เราใช้ผลการแปลงแซดของ $y(n)$

$$Y(z) = X(z) + A_1 X(z)z^{-1} + A_2 X(z)z^{-2} - B_1 Y(z)z^{-1} - B_2 Y(z)z^{-2}$$



รูปที่ 3.4 แสดงรูปแบบของโครงข่ายเชิงเลข IIR (a) โครงสร้างแบบโดยตรง IIR อันดับสอง

(b) ตัวกรอง IIR อันดับสองพร้อมทั้งส่วน Feed-Forward และ Feedback

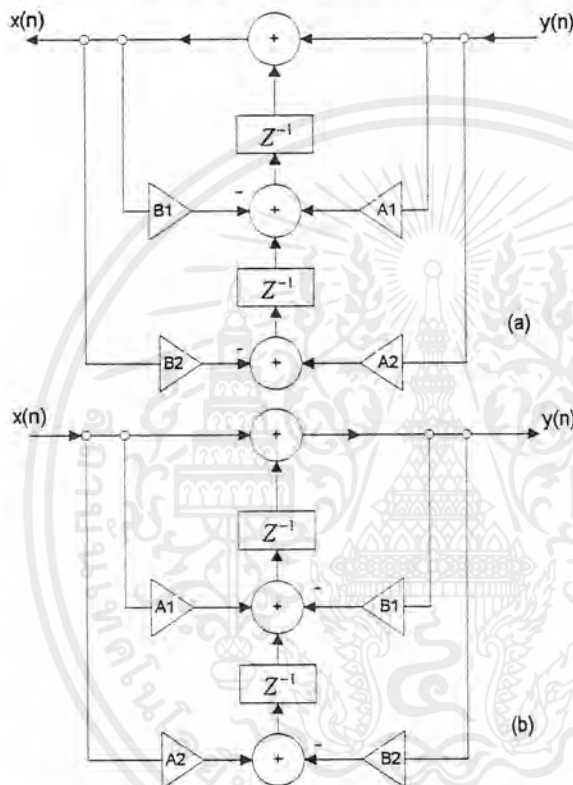
และฟังก์ชันถ่ายโอนของระบบกำหนดได้โดย

$$H(z) = \frac{1 + A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}}$$

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

transpose ของ network ถูกกำหนดโดยการกระทำดังนี้

- กลับทิศทางจุดแยกทั้งหมด
- สลับ input และ output
- เปลี่ยนจุดรวมเป็นจุดแยกและเปลี่ยนจุดแยกเป็นจุดรวม



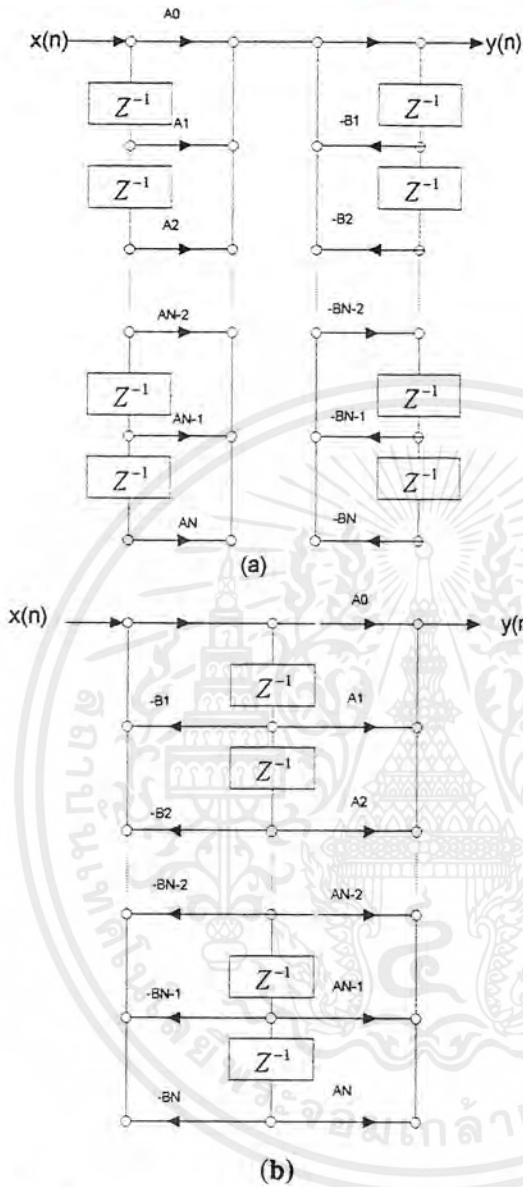
รูปที่ 3.5 แสดงตัวอย่างโครงข่ายแบบสลับ IIR (a) รูปแบบแบบสลับของรูปที่ 3.4b ตัวกรอง IIR อันดับสองกับหน่วยตัวหน่วง (b) รูป 3.5a เขียนใหม่โดยแสดงอินพุตอยู่ทางซ้ายมีรูมาตรฐาน

ในรูป 3.5a คือ transpose ของรูป 3.4b การทำ transpose ช่วยให้ประสิทธิภาพ multirate สูงขึ้น dual ของ network แสดง complementary function ของ network เดิม ตัวอย่างเช่น modulator กับ demodulator คือ dual operation commutation คือ การเปลี่ยนอันดับการกระทำของ network ฟังก์ชันถ่ายโอนไม่เปลี่ยนแต่ลดการคำนวณ

3.6 การสร้างตัวกรอง IIR แบบโดยตรง

โครงสร้างของรูป 3.4a เรียกว่าโครงสร้างแบบโดยตรง 1 ในการสร้างตัวกรอง IIR อันดับ 2 โดยส่วน feed-forward ถูกสร้างก่อน แล้วตามด้วยส่วนป้อนกลับ รูป 3.4b เรียกว่าโครงสร้างแบบโดยตรง 2 ส่วนป้อนกลับมาก่อนส่วน feed-forward อย่างไรก็ตามโครงสร้างแบบอนุกรมและแบบขนาน จะมีการคำนวณน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3.7 การสร้างตัวกรอง IIR แบบอนุกรมและแบบขนาน

จากรูป 3.6a โครงสร้างของ $H(z)$ อยู่ในรูปของฟังก์ชันถ่ายโอน 2 ส่วนดังนี้

$$H(z) = H_D(z) H_N(z) = \left(\frac{1}{\sum_{i=0}^N B_i z^{-i}} \right) \left(\sum_{i=0}^N A_i z^{-i} \right) \quad (3.8)$$

สมการ 3.8 อยู่ในรูปที่เรียกว่าอนุกรมของฟังก์ชันถ่ายโอน ในส่วนของ feed-forward เป็นผลมาจาก zero และส่วนป้อนกลับเป็นผลมาจาก pole เราสามารถสลับลำดับ(commutation)ของ cascade operation โดยไม่ทำให้ฟังก์ชันถ่ายโอนเปลี่ยนไป เมื่อตัวส่วนและตัวเศษถูก commute จากรูป 3.6a จะเห็นได้ว่า ตัวหน่วงเวลาทั้ง 2 ส่วนมีข้อมูลเหมือนกัน นั่นคือนำมารวมกันได้ ดังนั้นจะลดตัวหน่วง

เอกสารนี้ได้สร้างรูป 3.6b ที่ต้องการตัวหน่วงครึ่งหนึ่งของรูป 3.6a ซึ่งจะเรียกว่าโครงสร้างแบบโดยตรงสอง

หรือแบบบัญญัติ ที่มีจำนวนตัวหน่วงเท่ากับอันดับของตัวกรองจากรูป 3.6 จะเห็นได้ว่า ถ้า $B_i=0$ แล้วสมการ 3.9 ก็คือ FIR ซึ่ง $y(n)$ ขึ้นกับ input เท่านั้นรูป 3.7a และ 3.7b แสดงตัวกรองอันดับ 6 โดยการจัดตัวกรองอันดับ 2 3 ส่วนต่ออนุกรมและขนาน ค่าสัมประสิทธิ์ได้จาก partial fraction ของ $H(z)$

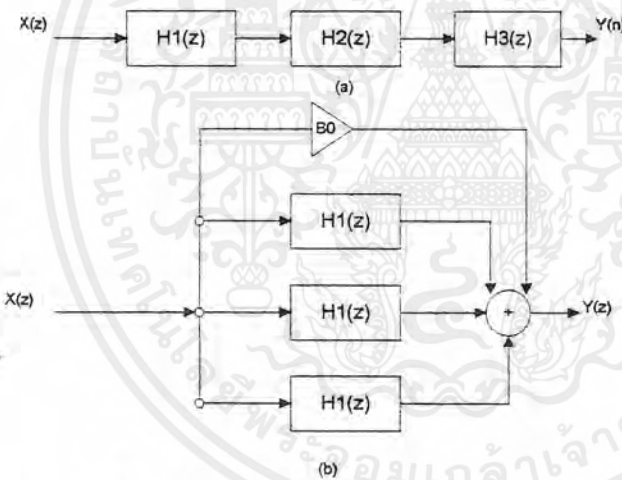
$$H(z) = \frac{1 + A_1 z^{-1} + A_2 z^{-2} + A_3 z^{-3} + A_4 z^{-4} + A_5 z^{-5} + A_6 z^{-6}}{1 + B_1 z^{-1} + B_2 z^{-2} + B_3 z^{-3} + B_4 z^{-4} + B_5 z^{-5} + B_6 z^{-6}}$$

นิพจน์อนุกรมจะอยู่ในรูปผลคูณของฟังก์ชันถ่ายโอน

$$H(z) = \prod_{i=1}^N H_i(z) \quad (3.9)$$

โดยที่ฟังก์ชันถ่ายโอน ของอันดับที่ i ในแบบอนุกรมคือ

$$H(z) = \frac{1 + A_{1i} z^{-1} + A_{2i} z^{-2}}{1 + B_{1i} z^{-1} + B_{2i} z^{-2}} \quad (3.10)$$



รูปที่ 3.7 แสดงโครง IIR อันดับหกแบบอนุกรมและขนาน (a) โครงสร้างแบบอนุกรมโดยใช้ส่วนอันดับสอง (b) โครงแบบขนานโดยใช้ส่วนอันดับสอง

สำหรับตัวกรองอันดับ 6 จะมีขั้นที่เป็นภาคอันดับ 1 ฟังก์ชันถ่ายโอนอนุกรมอันดับ 6 ก็คือผลคูณของภาคอันดับ 2 3 ส่วน

$$H(z) = H_1(z)H_2(z)H_3(z)$$

ในส่วนของโครงสร้างแบบขนานจะอยู่ในรูปผลรวมของฟังก์ชันถ่ายโอน

$$H(z) = B_0 + \sum_{i=1}^N H_i(z) \quad (3.11)$$

โดย B_0 เป็นค่าคงที่ และจากแบบโดยตรงแบ่งเป็น N ฟังก์ชันถ่ายโอน ในแต่ละส่วนก็คือ $H_i(z)$

ซึ่งได้จาก partial fraction ของฟังก์ชันถ่ายโอนทั้งหมด ในส่วนของอันดับ 2 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

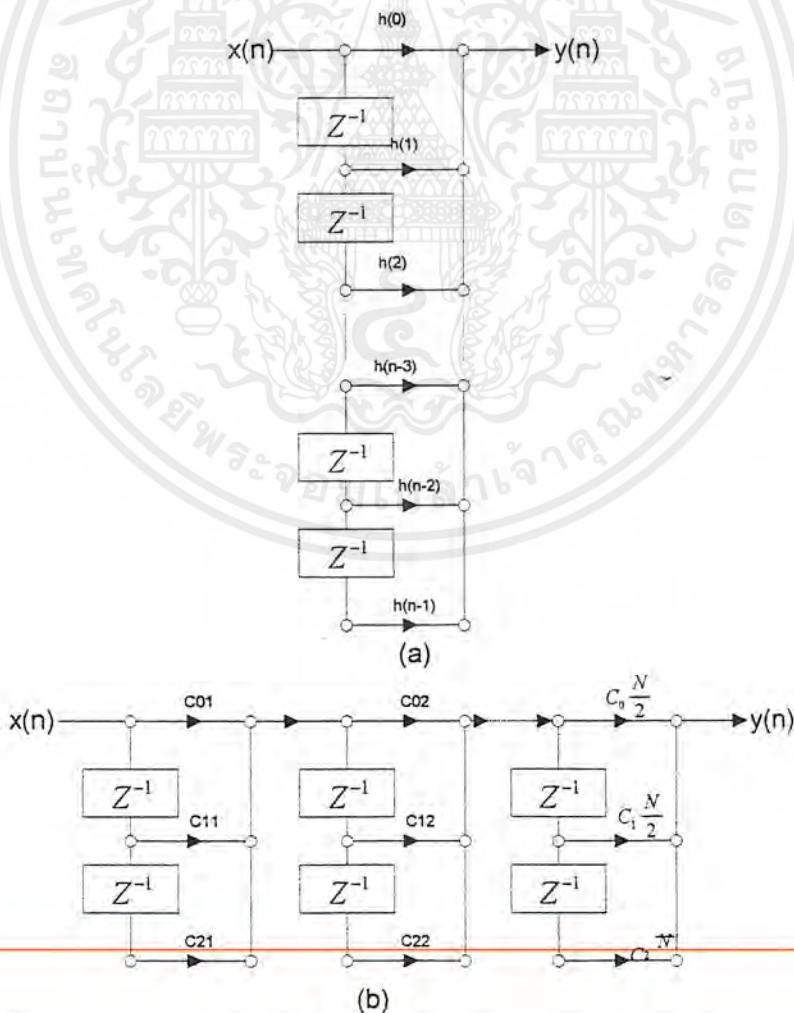
$$H_i(z) = \frac{A_{0i} + A_{1i}z^{-1}}{1 + B_{1i}z^{-1} + B_{2i}z^{-2}} \quad (3.12)$$

สำหรับภาคอันดับ 1 A_{1i} และ B_{2i} ถูกเซตให้เป็นศูนย์ ในสมการ 3.12 สำหรับฟังก์ชันถ่ายโอนอันดับ 6 ในรูปแบบขนานเขียนได้โดย

$$H(z) = H_1(z) + H_2(z) + H_3(z) + B_0$$

3.7 การสร้างตัวกรอง FIR

โครงสร้างแบบโดยตรงของ FIR แสดงดังรูป 3.8(a) จะเห็นได้ว่ามีความสมมูลกับส่วน feed-forward ของรูป 3.6a ของแบบโดยตรง IIR และเพื่อให้ความถูกต้องมากขึ้นที่อันดับสูง ก็อาจใช้การสร้างตัวกรองแบบอนุกรม ดังรูป 3.8(b)



รูปที่ 3.8 แสดงตัวกรองเชิงเลข FIR (a) โครงแบบโดยตรง (b) โครงสร้างแบบอนุกรม
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบวงจรกรองเชิงเลขแบบ FIR

สำหรับตัวกรองแบบ FIR ผลตอบสนองจะขึ้นกับ input เท่านั้น ขณะที่ IIR ผลตอบสนองขณะนั้น จะขึ้นกับ input และผลตอบสนองในอดีต

3.8.1 ลักษณะตัวกรองเชิงเลขแบบ FIR

ข้อได้เปรียบและข้อเสียเปรียบของตัวกรอง FIR เปรียบเทียบกับ IIR

1. ตัวกรองแบบ FIR สามารถออกแบบให้ได้ผลตอบสนองเชิงเส้นได้ โดยที่เฟสเชิงเส้นมีความสำคัญต่อการนำไปใช้งาน หากเกิดความผิดเพี้ยนทางเฟสจะลดประสิทธิภาพลงได้
2. ตัวกรองแบบ FIR จะเสถียรเสมอ
3. สัญญาณรบกวนจากการ quantize สามารถละทิ้งได้สำหรับ โครงสร้างแบบไม่ป้อนกลับ
4. ที่อันดับตัวกรองสูงๆ ต้องการตัวเก็บค่าสัมประสิทธิ์มากกว่าแบบ IIR

3.8.2 คุณสมบัติของตัวกรอง FIR

3.8.2.1 ตัวกรองเฟสเชิงเส้น

ฟังก์ชันถ่ายโอนของตัวกรอง FIR กำหนดโดย

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (3.13)$$

โดย $h(n)$ คือ ผลตอบสนองของ impulse ของตัวกรอง สมการผลต่างหาได้จาก

$$y(iT) = \sum_{n=0}^{N-1} h(n)x(iT - nT) \quad (3.14)$$

จากสมการ 5.1 การแปลง Fourier ของลำดับ $h(n)$ คือ

$$H(e^{j\omega T}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega nT} = |H(e^{j\omega T})| e^{j\theta(\omega)} \quad (3.15)$$

ค่าผลตอบสนองต่อขนาดและเฟส คือ

$$\begin{aligned} M(\omega) &= |H(e^{j\omega T})| \\ \theta(\omega) &= \tan^{-1} \frac{-\text{Im} H(e^{j\omega T})}{\text{Re} H(e^{j\omega T})} \end{aligned} \quad (3.16)$$

พิจารณาค่าหน่วยเฟสและค่าหน่วยกลุ่มของตัวกรอง

$$\tau_p = -\frac{\theta(\omega)}{\omega} \quad \text{และ} \quad \tau_g = -\frac{\partial \theta(\omega)}{\partial \omega} \quad (3.17)$$

สำหรับตัวกรองที่มีค่า τ_p และ τ_g คงที่ นั่นคือเป็นอิสระต่อความถี่ เรียกว่า ตัวกรองเฟสเชิงเส้น ดังนั้นสำหรับผลตอบสนองเฟสที่เป็นเชิงเส้น เราต้องการ

$$\theta(\omega) = -\tau\omega \quad -\pi < \omega < \pi \quad (3.18)$$

จากสมการ 3.15 , 3.16 และ 3.18 ผลตอบสนองเฟสสามารถเขียนได้เป็น

$$\theta(\omega) = -\tau\omega = \tan^{-1} \frac{-\sum_{n=0}^{N-1} h(n) \sin \omega n T}{\sum_{n=0}^{N-1} h(n) \cos \omega n T} \quad (3.19)$$

$$\tan \omega \tau = \frac{\sum_{n=0}^{N-1} h(n) \sin \omega n T}{\sum_{n=0}^{N-1} h(n) \cos \omega n T} \quad (3.20)$$

$$\text{สุดท้าย จะได้ว่า } \sum_{n=0}^{N-1} h(n) \sin(\omega \tau - \omega n T) = 0 \quad (3.21)$$

สามารถแสดงให้เห็นได้ว่าคำตอบของสมการ 3.21 คือ

$$\tau = \frac{(N-1)T}{2} \quad (3.22)$$

$$\text{และ } h(n) = h[(N-1-n)] \quad \text{สำหรับ } 0 < n < N-1 \quad (3.23)$$

ดังนั้นตัวกรอง FIR จะมีตัวหน่วงเฟสและตัวหน่วงกลุ่มคงที่ถ้าเงื่อนไขตามสมการ 3.22 และ 3.23 เป็นจริง จากคุณสมบัติความสมมาตรของสมการ 3.23 จะเห็นได้ว่าสมการ 3.13 จะมีผลลัพธ์เป็นพหุนาม mirror image ตัวอย่างแสดงดังนี้สำหรับตัวกรอง FIR ที่มีเฟสเชิงเส้น คุณสมบัติตามสมการ 3.22 และ 3.23 ต้องเป็นจริง กำหนดให้ $N = 11$ แสดงให้เห็นว่าสมการ 3.21 เป็นจริงคำตอบ

$$\begin{aligned} \sum_{n=0}^{10} h(n) \sin(5-n)\omega T &= h(0) \sin(5\omega T) + h(1) \sin(4\omega T) + h(2) \sin(3\omega T) \\ &\quad + h(3) \sin(2\omega T) + h(4) \sin(\omega T) + h(5) \sin(0) + \\ &\quad + h(6) \sin(-\omega T) + h(7) \sin(-2\omega T) + h(8) \sin(-3\omega T) + \\ &\quad + h(9) \sin(-4\omega T) + h(10) \sin(-5\omega T) \end{aligned}$$

เมื่อ $h(n) = h(N-1-n)$ จะได้ว่า

$$h(0) = h(10) \quad h(3) = h(7) \quad h(1) = h(9)$$

$$h(4) = h(6) \quad h(2) = h(8) \quad h(5)$$

สุดท้ายสมการที่ 3.21 เป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาเงื่อนไขความสมมาตรของผลตอบสนอง impulse สำหรับ N เป็นเลขคี่และเลขคู่ จากสมการ 3.22 τ เป็นเลขจำนวนเต็มคี่ และผลตอบสนอง impulse สมมาตรที่จุดศูนย์กลาง $(N-1)/2$ ถ้า N เป็นเลขคู่ τ จะไม่เป็นจำนวนเต็มตัวกรองโดยทั่วไปลำดับ impulse สำหรับ N เป็นเลขคู่และเลขคี่ ถ้า N เลขคู่จะอยู่ตรงกลางระหว่างค่าสัมประสิทธิ์ $(N-2)/2$ และ $N/2$ และถ้า N เป็นเลขคี่จะอยู่ที่ $(N-1)/2$ เป็นเลขคี่ ในการประยุกต์ต้องการค่าตัวหน่วงกลุ่มคงที่ ซึ่งจะได้ว่าผลตอบสนอง impulse จะอยู่ในรูปแบบ

$$h(n) = -h(N-1-n) \quad (3.24)$$

ผลตอบสนอง impulse ที่ได้จากสมการ 3.24 คือปฏิสมมาตรซึ่งมีการนำไปประยุกต์ใช้ อย่างเช่น wideband differentiator

ตัวอย่าง

ตัวกรองเชิงเลข FIR ที่มีคุณลักษณะตาม ฟังก์ชันถ่ายโอน ($N = 11$) จงหาผลตอบสนองต่อขนาด และแสดงว่าตัวหน่วงและกลุ่มเป็นค่าคงที่

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

คำตอบ

กระจายค่า $H(z)$ เราจะได้

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} \\ + h(6)z^{-6} + h(7)z^{-7} + h(8)z^{-8} + h(9)z^{-9} + h(10)z^{-10}$$

เมื่อ $\tau = 5$ ฟังก์ชันถ่ายโอน $H(z)$ สามารถแสดงในรูปของตัวหน่วงเฟสและตอบสนองไม่เป็นเหตุกภาพ นั่นคือ

$$H(z) = z^{-5} [h(0)z^5 + h(1)z^4 + h(2)z^3 + h(3)z^2 + h(4)z^1 + h(5) \\ + h(6)z^{-1} + h(7)z^{-2} + h(8)z^{-3} + h(9)z^{-4} + h(10)z^{-5}]$$

เมื่อ $h(n) = h(N-1-n)$ จะได้

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

โดย z^{-n} คือเวลาหน่วง nT วินาทีผลตอบสนองความถี่หาได้จากการแทนค่า $z =$ จะได้ว่า

$$H(e^{j\omega T}) = e^{-j5\omega T} [2h(0)\cos(5\omega T) + 2h(1)\cos(4\omega T) + 2h(2)\cos(3\omega T) \\ + 2h(3)\cos(2\omega T) + 2h(4)\cos(\omega T) + h(5)] \\ = e^{-j5\omega T} \left[h(5) + 2 \sum_{n=1}^5 h(n) \cos(n\omega T) \right] = e^{-j5\omega T} M(\omega)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์เป็นดังที่คาดไว้จากสมการ 3.15 และ 3.16 สุดท้ายจากสมการ 3.17 จะได้ $\tau_p = \tau_g = 5T$

3.8.2.2 ผลตอบสนองความถี่ของตัวกรอง FIR เชิงเส้น

ผลตอบสนองของตัวกรอง เหตุภาพเฟสเชิงเส้น สำหรับ N เป็นเลขคี่ และจากสมการ 3.15 คือ

$$H(e^{j\omega T}) = \sum_{n=0}^{(N-3)/2} h(n)e^{-j\omega nT} + h\left(\frac{N-1}{2}\right)e^{-j\omega(N-1)T/2} + \sum_{n=(N+1)/2}^{N-1} h(n)e^{-j\omega nT} \quad (3.25)$$

จาก 5.11 แทนค่า $m = N-1-n$ เปลี่ยนลิมิตของผลรวม สุดท้ายให้ $m = n$ แล้วจะได้

$$\sum_{n=(N+1)/2}^{N-1} h(n)e^{-j\omega nT} = \sum_{n=0}^{(N-3)/2} h(n)e^{-j\omega(N-1-n)T} \quad (3.26)$$

แทนค่าสมการ 3.26 ลงในสมการ 3.15 จะได้

$$H(e^{j\omega T}) = \sum_{n=0}^{(N-3)/2} h(n)[e^{-j\omega nT} + e^{-j\omega(N-1-n)T/2}] + h\left(\frac{N-1}{2}\right)e^{-j\omega(N-1)T/2} \quad (3.27)$$

ให้ $k = (N-1)/2 - n$ จะได้

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2} \times \left[\sum_{k=1}^{(N-1)/2} h\left(\frac{N-1}{2} - k\right)(e^{j\omega kT} + e^{-j\omega kT}) + h\left(\frac{N-1}{2}\right) \right] \quad (3.28)$$

จัดรูปให้ง่ายได้เป็น

$$\begin{aligned} H(e^{j\omega T}) &= e^{-j\omega(N-1)T/2} \left[\sum_{k=0}^{(N-1)/2} a(k) \cos \omega kT \right] \\ &= e^{-j\omega(N-1)T/2} M(\omega) \end{aligned} \quad (3.29)$$

$$a(0) = h\left(\frac{N-1}{2}\right) \quad a(k) = 2h\left(\frac{N-1}{2} - k\right)$$

ดังผลที่ได้จากตัวอย่าง สมการ 3.29 กำหนดผลตอบสนองความถี่ของตัวกรองเหตุภาพ FIR เชิงเส้น โดยทำนองเดียวกัน N เป็นเลขคู่ และกรณีปฏิสมมาตรจะได้ผลตอบสนองความถี่ ดังตาราง 3.1 สำหรับกรณีไม่เป็นเหตุภาพ $h(n) = h(-n)$ ก็สอดคล้องกับตัวกรองเฟสศูนย์ ผลตอบสนองความถี่ที่ได้

$$H(z) = \sum_{n=-M}^M h(n)z^{-n}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
&= \sum_{n=-M}^{-1} h(n)z^{-n} + h(0) + \sum_{n=1}^M h(n)z^{-n} \\
&= \sum_{n=1}^M h(-n)z^{-n} + h(0) + \sum_{n=1}^M h(n)z^{-n} \\
&= h(0) + \sum_{n=1}^M h(n)[z^{-n} + z^{-n}]
\end{aligned}$$

ถ้าให้ค่า $H(z)$ สำหรับ $z = e^{j\omega T}$ จะได้

$$H(e^{j\omega T}) = h(0) + \sum_{n=1}^M 2h(n) \cos \omega n T = \sum_{n=0}^M a(n) \cos \omega n T \quad 3.30$$

3.8.2.3 ตำแหน่งของ zeros ของตัวกรอง FIR เชิงเส้น

รากของพหุนาม FIR แสดงถึงเงื่อนไขความสมมาตรในระนาบ z คุณสมบัติของตัวกรองเชิงเส้นสามารถเขียนได้ดังสมการ 3.16 ในรูป

$$H(z) = z^{-(N-1)/2} \sum_{k=0}^{(N-1)/2} \frac{a(k)}{2} (z^k + z^{-k}) \quad (3.31)$$

แทนค่า a_0 และ a_k จากสมการ 3.29 จะได้

$$H(z^{-1}) = z^{(N-1)} H(z) \quad (3.32)$$

จากสมการ 3.32 จะเห็นว่า $H(z^{-1})$ และ $H(z)$ นั้นเหมือนกันเพียงแต่หน่วยไป $(N-1)$ ลำดับ นั่นคือค่าให้ zero $z_i = r_i e^{j\theta}$ แล้วตำแหน่ง zero จะมีดังต่อไปนี้

1. ถ้า $r_i \neq 1, \theta_i \neq 0, \pi$ แล้ว zeros จะเกิดใน quadruplets นั่นคือ คู่ conjugate จะเกิดนอก unit circle
2. ถ้า $r_i = 1, \theta_i \neq 0, \pi$ แล้ว zeros จะเกิดบน unit circle
3. ถ้า $r_i \neq 1, \theta_i = 0 \text{ or } \pi$ แล้ว zeros จะเป็นค่าจริง และเกิดบน unit circle
4. ถ้า $r_i = 1, \theta_i = 0, \pi$ แล้ว zeros วางอยู่บน +1 หรือ -1

3.8.3 การออกแบบโดยใช้อนุกรมฟูริเยร์

ผลตอบสนองความถี่อนุกรม Fourier ของตัวกรองเชิงเลข FIR สามารถเขียนได้โดยอนุกรม Fourier

$$H(e^{j2\pi f T}) = \sum_{n=-\infty}^{\infty} h_d(n) e^{-j2\pi f n T} \quad (3.33)$$

โดยที่สัมประสิทธิ์คือ ลำดับผลตอบสนอง impulse ของตัวกรองที่ต้องการสามารถเขียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$h_d(n) = \frac{1}{F} \int_{-F/2}^{F/2} H(e^{j2\pi fT}) e^{j2\pi f n T} df \quad (3.34)$$

จากสมการ 3.33 แทนค่า $e^{j\omega T} = z$ เราจะได้ ฟังก์ชันถ่ายโอนของตัวกรองเชิงเลข คือ

$$H(z) = \sum_{n=-\infty}^{\infty} h_d(n) z^{-n} \quad (3.35)$$

จากสมการ 3.35 เพื่อจะได้ตัวกรองเหตุการณ์จำกัด ก็ต้องตัดผลตอบสนอง impulse ที่ไม่จำกัดและคูณด้วย $z^{-(N-1)/2}$ จะได้

$$\begin{aligned} H(z) &= z^{-(N-1)/2} \sum_{n=(N-1)/2}^{(N-1)/2} h_d(n) z^{-n} \\ &= z^{-(N-1)/2} \left[h_d(0) + \sum_{n=1}^{(N-1)/2} h_d(n) (z^n + z^{-n}) \right] \end{aligned} \quad (3.36)$$

จากสมการ 3.36 จะเห็นได้ว่าผลการตัดปลายของ อนุกรม Fourier จะทำให้เกิดการ oscillate ในช่วงแถบผ่านและแถบหยุดซึ่งเรียกว่า Gibbs Phenomenon จะได้แสดงต่อไปว่า oscillate ที่ไม่ต้องการสามารถรู้ผลได้โดยการคูณผลตอบสนอง impulse ที่ต้องการด้วย window function ที่เหมาะสม ตัวอย่าง 3.3

a. ใช้วิธีอนุกรม Fourier ในการออกแบบตัวกรองความถี่ต่ำ

$$H(e^{j2\pi fT}) = \begin{cases} 1 & \text{for } |f| \leq f_p \\ 0 & \text{for } f_p < |f| \leq F/2 \end{cases}$$

โดย f_p คือความถี่ผ่าน และ F คือความถี่สูง

b. ถ้าให้ $N = 11$ จงแสดงว่า ฟังก์ชันถ่ายโอนที่ได้จากสมการ 3.36 เป็นเหตุการณ์จำกัด

a. จากสมการ 3.34

$$\begin{aligned} h_d(n) &= \frac{1}{F} \int_{-F/2}^{F/2} e^{j2\pi f n T} df = \frac{1}{n\pi} \sin 2\pi f_p n T \\ &= \left(\frac{2f_p}{F} \right) \frac{\sin 2\pi f_p n T / F}{2\pi f_p n T / F} \end{aligned} \quad (3.37)$$

b. จากสมการ 3.36 คูณตลอดให้ค่า $\tau_p = 5$ แล้วจะได้ ฟังก์ชันถ่ายโอนเหตุการณ์จำกัดนี้

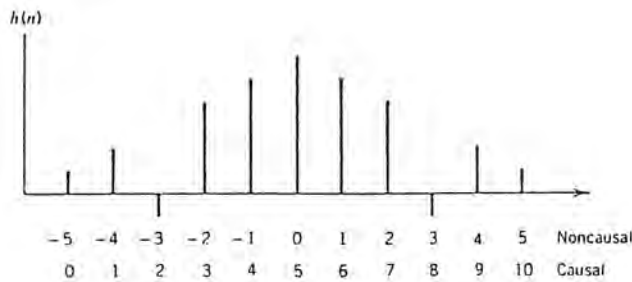
$$H(z) = h_d(0)z^{-5} + z^{-5} \sum_{n=1}^5 h_d(n) [z^n + z^{-n}] \quad (3.38)$$

กระจายสมการ 3.38 ได้

$$\begin{aligned} H(z) &= h_d(0)z^{-5} + h_d(1)(z^{-4} + z^{-6}) + h_d(2)(z^{-3} + z^{-7}) \\ &\quad + h_d(3)(z^{-2} + z^{-8}) + h_d(4)(z^{-1} + z^{-9}) + h_d(5)(z^0 + z^{-10}) \end{aligned} \quad (3.39)$$

จากตัวอย่าง 3.3 แสดงว่าค่าเหตุการณ์จำกัดทำได้โดยการคูณ $H(z)$ จากสมการ 3.35 ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงความสัมพันธ์ระหว่างเหตุภาพและไม่เป็นเหตุภาพของผลตอบสนอง impulse

สำหรับ $N = 11$ ผลตอบสนองต่อขนาดตัวกรอง FIR ได้มาจากสมการ 3.36 โดยแทนค่า $z = e^{j\omega T}$

$$M(\omega) = h_d(0) + 2 \sum_{n=1}^{(N-1)/2} h_d(n) \cos \omega n T \quad (3.40)$$

การคำนวณ ผลตอบสนองความถี่ของสมการ 3.40 ถ้ามีจำนวน N เพิ่มขึ้นจะส่งผลต่อการ oscillate ของการแถบผ่านและแถบหยุด นั่นคือ ความยาวของ เพิ่มขึ้น ขนาดของการ oscillate อย่างไรก็ ตามขนาดของลูกคลื่นใกล้จุดไม่ต่อเนื่อง ยังคงไม่แปรเปลี่ยน การนำ window function มาใช้ จะช่วยลดผลของการ oscillate

3.8.3.1 ฟังก์ชันวินโดว์ (Window Function) เพื่อที่จะลดผลของการ oscillate ก็จะต้องปรับปรุง สัมประสิทธิ์ของอนุกรม Fourier ฟังก์ชันที่นำมาถ่วงน้ำหนักนี้คือ window function เมื่อการตัด ปลาย อนุกรม Fourier ความยาวไม่จำกัดเทียบเท่าด้วยการคูณ window ที่เหลื่อม

$$a_R(n) = \begin{cases} 1 & \text{for } |n| \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3.41)$$

$$\text{จะได้ } h(n) = h_d(n) a_R(n) \quad (3.42)$$

เมื่อการคูณในโดเมนเวลาเทียบเท่ากับผลการประสานในโดเมนความถี่ ข้อกำหนดในการออกแบบ ตัวกรอง FIR เป็นการหา window function ที่จำกัด การแปลง Fourier ให้ไหลบ่าข้างต่ำเมื่อเทียบกับ โหลบหลัก

จาก ทฤษฎีผลการประสานเชิงซ้อน จะได้

$$H_A(e^{j\omega T}) = \frac{1}{2\pi F} \int_0^{2\pi F} H(e^{j\omega T}) A(e^{j(\omega-\Omega)T}) d\Omega \quad (3.43)$$

ดังนั้น ผลการประสานวัฏจักรจะเห็นได้ว่าความกว้างของแถบเปลี่ยนสถานะเป็นฟังก์ชันของความยาว N ของ window function

window สี่เหลี่ยม

window สี่เหลี่ยม ได้จากสมการ 3.41 spectrum ของ ได้จากสมการ 3.33 ดังเช่น

$$\begin{aligned} A_R(e^{j\omega T}) &= \sum_{n=-(N-1)/2}^{(N-1)/2} e^{-j\omega nT} = e^{j\omega(N-1)T/2} \sum_{n=0}^{N-1} e^{-j\omega nT} \\ &= e^{j\omega(N-1)T/2} \frac{1 - e^{j\omega NT}}{1 - e^{j\omega T}} = \frac{e^{j\omega NT/2} - e^{j\omega NT/2}}{e^{j\omega T/2} - e^{-j\omega T/2}} \\ &= \frac{\sin(\omega NT/2)}{\sin(\omega T/2)} \end{aligned} \quad (3.44)$$

ระดับโพลข้างสูงสุดมีค่าลดลง 13 เดซิเบล จากระดับยอดของโพลหลัก พิจารณา ผลตอบสนองความถี่ของ window ที่เป็นเหตุภาพคือ

$$A_R(e^{j\omega T}) = \sum_{n=0}^{N-1} e^{-j\omega nT} = e^{-j\omega(N-1)T/2} \frac{\sin(\omega NT/2)}{\sin(\omega T/2)} \quad (3.45)$$

ผลตอบสนองเฟสเชิงเส้นได้จาก $\theta(\omega) = -\omega(N-1)T/2$

Hamming Window

สมการของ Hamming function คือ

$$a_H(n) = \begin{cases} 0.54 + 0.46 \cos \frac{2\pi n}{N-1} & \text{for } |n| \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3.46)$$

spectrum ของ Hamming window ได้จาก

$$a_H(n) = a_R(n) \left[0.54 + 0.46 \cos \frac{2\pi n}{N-1} \right] \quad (3.47)$$

จากสมการ 5.36 จะได้

$$\begin{aligned} A_H(e^{j\omega T}) &= 0.54 \frac{\sin(\omega NT/2)}{\sin(\omega T/2)} + 0.46 \frac{\sin[\omega NT/2 - N\pi/(N-1)]}{\sin[\omega T/2 - \pi/(N-1)]} \\ &\quad + 0.46 \frac{\sin[\omega NT/2 + N\pi/(N-1)]}{\sin[\omega T/2 + \pi/(N-1)]} \end{aligned} \quad (3.48)$$

โดยที่ระดับโพลข้างสูงสุดมีค่าประมาณ -43 เดซิเบล

Blackman Window

Blackman function กำหนดโดย

$$a_B(n) = \begin{cases} 0.42 + 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{4\pi n}{N-1} & \text{for } |n| < \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \quad 5.38$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงให้เห็นได้ว่าระดับสูงสุดของโพลข้างลดลง -58 เดซิเบล จากระดับยอดโพลหลัก

Kaiser Window

สำหรับ window ก่อนหน้านี้ค่าความกว้างของโพลหลักเป็นส่วนกลับกับค่า N นั่นคือ เพิ่มค่าความยาวของ window จะลดค่าความกว้างของโพลหลัก ซึ่งผลก็คือลดแถบเปลี่ยนสถานะของตัวกรอง อย่างไรก็ตามค่าลดทอนในแถบหยุดน้อยสุดเป็นอิสระต่อความยาวของ window และขึ้นกับชนิดของ window ดังนั้น เพื่อที่จะได้ค่าลดทอนในแถบหยุดน้อยสุดตามต้องการ ผู้ออกแบบจะต้องหา window ที่เข้ากับ specification Kaiser Window มีตัวแปร α ซึ่งเป็นตัวควบคุมระดับของโพลข้างเทียบกับระดับยอดของโพลหลัก คล้ายกันกับ window ชนิดอื่น ความกว้างของโพลหลักสามารถปรับโดยความยาวของ window ดังนั้นวงจรกรองเชิงเลข FIR สามารถใช้ Kaiser Window ออกแบบได้อย่างมีประสิทธิภาพ ระบบ Kaiser ได้พัฒนาเบสเซลฟังก์ชันชนิดหนึ่งอันดับศูนย์ หรือ $I_0(x)$ มาใช้ออกแบบ window function คือ

$$a_K(n) = \begin{cases} \frac{I_0(\beta)}{I_0(\alpha)} & \text{for } |n| \leq \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3.49)$$

โดยที่ α เป็นตัวแปรอิสระ ค่า β หาได้จาก

$$\beta = \alpha \left[1 - \left(\frac{2n}{N-1} \right)^2 \right]^{0.5} \quad (3.50)$$

ค่า $I_0(x)$ ได้จากการกระจายอนุกรมกำลัง

$$\begin{aligned} I_0(x) &= 1 + \sum_{n=1}^{\infty} \left[\frac{1}{k!} \left(\frac{x}{2} \right)^k \right]^2 \\ &= 1 + \frac{0.25x^2}{(1!)^2} + \frac{(0.25x^2)^2}{(2!)^2} + \frac{(0.25x^2)^3}{(3!)^2} + \dots \end{aligned} \quad (3.51)$$

ในการใช้งานทั่วไปความละเอียด 25 พจนาก็เพียงพอ

ค่า spectrum ของ Kaiser Window จะได้จาก

$$\sum_{n=-(N-1)/2}^{(N-1)/2} a_K(n) e^{-j\omega n T} = a_K(0) + 2 \sum_{n=1}^{(N-1)/2} a_K(n) \cos \omega n T \quad (3.52)$$

พิจารณาข้อกำหนดในการออกแบบวงจรกรองความถี่ต่ำ โดยมีค่าขนาดของลูกคลื่นในแถบผ่าน A_p และค่าลดทอนแถบหยุดน้อยสุด A_s ในหน่วยเดซิเบล ดังนี้

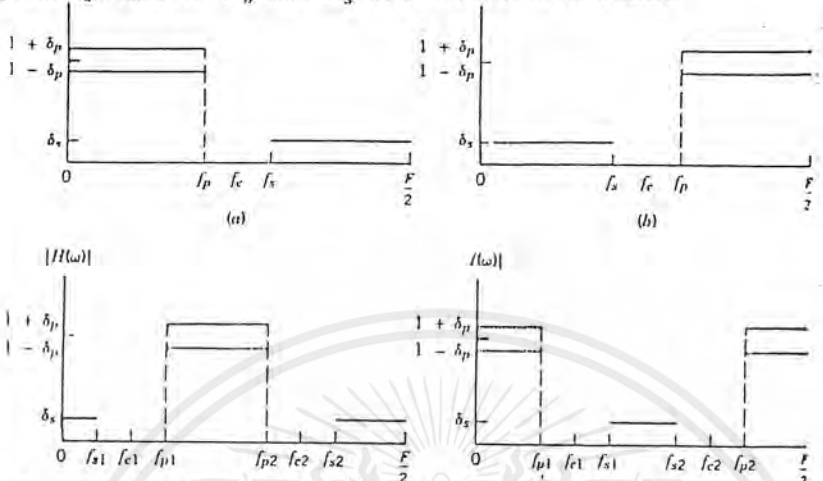
$$A_p = 20 \log_{10} \frac{1 + \delta_p}{1 - \delta_p} \quad (3.53)$$

$$A_s = -20 \log_{10} \delta_s \quad (3.54)$$

$$\text{ค่า bandwidth ในช่วงเปลี่ยนสถานะ คือ } \Delta F = f_s - f_p \quad (3.55)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อความสะดวกเรากำหนดว่า A'_p, A'_s เป็นค่าขนาดของลูกคลื่นในแถบผ่าน และ ค่าลดทอน แถบหยุดน้อยสุด และให้ A_p และ A_s เป็นค่าที่ได้จากการออกแบบ



รูปที่ 3.10 แสดงผลตอบสนองความถี่ในอุดมคติ (a) ตัวกรองความถี่ต่ำ (b) ตัวกรองความถี่สูง (c) ตัวกรองความถี่ผ่าน (d) ตัวกรองความถี่หยุด

$$\begin{aligned} A_p &\leq A'_p \\ A_s &\geq A'_s \end{aligned} \quad (3.56)$$

3.8.3.2 การออกแบบตัวกรอง FIR โดยใช้ Kaiser Window

ข้อกำหนดในการออกแบบ

1. ชนิดของตัวกรอง: วงจรกรองความถี่ต่ำ, วงจรกรองความถี่สูง, วงจรกรองความถี่ผ่าน, วงจรกรองความถี่หยุด
2. ค่าความถี่ผ่านและความถี่หยุดในหน่วย Hertz

LP/HP : f_p และ f_s

BP/BS : f_{p1}, f_{p2}, f_{s1} และ f_{s2}

3. ค่าขนาดของลูกคลื่นในแถบผ่าน และ ค่าลดทอนแถบหยุดน้อยสุด A'_p, A'_s
 4. ค่าความถี่สุ่มในหน่วย Hertz : F
 5. อันดับ (N) เป็นเลขคี่
- ขั้นตอนการออกแบบ

1. หาค่า δ ที่สอดคล้องกับสมการ 5.43 , 5.44 และ 5.46 และเลือก

$$\delta = \min(\delta_p, \delta_s) \quad (3.57)$$

โดยจากสมการ 5.43 และ 5.44 จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\delta_s = 10^{-0.05A_s} \quad \delta_p = \frac{10^{0.05A_p} - 1}{10^{0.05A_p} + 1} \quad (3.58)$$

2. คำนวณค่า A_s จากสมการ 5.44

3. หาค่า α จากสมการ

$$\alpha = \begin{cases} 0 & \text{for } A_s \leq 21 \\ 0.5842(A_s - 21)^{0.4} + 0.07886(A_s - 21) & \text{for } 21 \leq A_s \leq 50 \\ 0.1102(A_s - 8.7) & \text{for } A_s > 50 \end{cases} \quad (3.59)$$

4. หาค่า D จาก

$$D = \begin{cases} 0.9222 & \text{for } (A_s \leq 21) \\ \frac{A_s - 7.95}{14.36} & \text{for } A_s > 21 \end{cases} \quad (3.60)$$

5. คำนวณค่า N ที่เป็นเลขคี่ต่ำสุดตามเงื่อนไข

$$N \geq \frac{FD}{\Delta F} + 1 \quad (3.61)$$

6. คำนวณค่าผลตอบสนอง impulse ที่ปรับปรุงโดยใช้สมการ 3.33 ,3.49 และ 3.67

$$h(n) = a_K(n)h_d(n) \quad \text{for } |n| \leq \frac{N-1}{2} \quad (3.62)$$

7. ฟังก์ชันถ่ายโอนได้จาก

$$H(z) = z^{-(N-1)/2} \left[h(0) + 2 \sum_{n=1}^{(N-1)/2} h(n)(z^n + z^{-n}) \right] \quad (3.63)$$

โดยที่ $h(0) = a_K(0)h_d(0)$

$$h(n) = a_K(n)h_d(n) \quad (3.64)$$

ผลตอบสนองค่าขนาดได้จากสมการ 3.63 คือ

$$M(\omega) = h(0) + 2 \sum_{n=1}^{(N-1)/2} h(n) \cos 2\pi nT \quad (3.65)$$

3.8.3.3 สมการออกแบบทั่วไปของตัวกรอง FIR

ในส่วนนี้จะพิจารณาสมการการออกแบบสำหรับ วงจรกรองเชิงเลข FIR ชนิดต่าง ๆ

LOWPASS FIR FILTER

$$h_d(n) = \begin{cases} \left(\frac{2f_c}{F}\right) \frac{\sin 2\pi n f_c / F}{2\pi n f_c / F} & \text{for } n > 0 \\ \frac{2f_c}{F} & \text{for } n = 0 \end{cases} \quad (3.66)$$

$$\text{โดยที่ } f_c = 0.05(f_p + f_s) \quad \Delta F = f_s - f_p \quad (3.67)$$

HIGHPASS FIR FILTER

$$h_d(n) = \begin{cases} -\left(\frac{2f_c}{F}\right) \frac{\sin 2\pi n f_c / F}{2\pi n f_c / F} & \text{for } n > 0 \\ 1 - \frac{2f_c}{F} & \text{for } n = 0 \end{cases} \quad (3.68)$$

$$\text{โดยที่ } f_c = 0.5(f_p + f_s) \quad \Delta F = f_p - f_s \quad (3.69)$$

BANDPASS FIR FILTER

$$h_d(n) = \begin{cases} \frac{1}{n\pi} [\sin(2\pi n f_{c2} / F) - \sin(2\pi n f_{c1} / F)] & \text{for } n > 0 \\ \frac{2}{F}(f_{c2} - f_{c1}) & \text{for } n = 0 \end{cases} \quad (3.70)$$

$$\begin{aligned} \text{โดยที่ } f_{c1} &= f_{p1} - \frac{\Delta F}{2} & f_{c2} &= f_{p2} + \frac{\Delta F}{2} \\ \Delta F_1 &= f_{p1} - f_{s1} & \Delta F_h &= f_{s2} - f_{p2} \\ \Delta F &= \min[\Delta F_1, \Delta F_h] \end{aligned} \quad (3.71)$$

BANDSTOP FIR FILTER

$$h_d(n) = \begin{cases} \frac{1}{n\pi} [\sin(2\pi n f_{c1} / F) - \sin(2\pi n f_{c2} / F)] & \text{for } n > 0 \\ \frac{2}{F}(f_{c1} - f_{c2}) + 1 & \text{for } n = 0 \end{cases} \quad (3.72)$$

$$\text{โดยที่ } f_{c1} = f_{p1} + \frac{\Delta F}{2} \quad f_{c2} = f_{p2} - \frac{\Delta F}{2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}\Delta F_1 &= f_{s1} - f_{p1} & \Delta F_h &= f_{p2} - f_{s2} \\ \Delta F &= \min[\Delta F_1, \Delta F_h]\end{aligned}\quad (3.73)$$

ตัวอย่าง

จงใช้ Kaiser Window ออกแบบตัวกรองความถี่ต่ำ ตามเงื่อนไขดังนี้

$$A_p \leq 0.1 \text{ dB}$$

$$A_s \geq 44 \text{ dB}$$

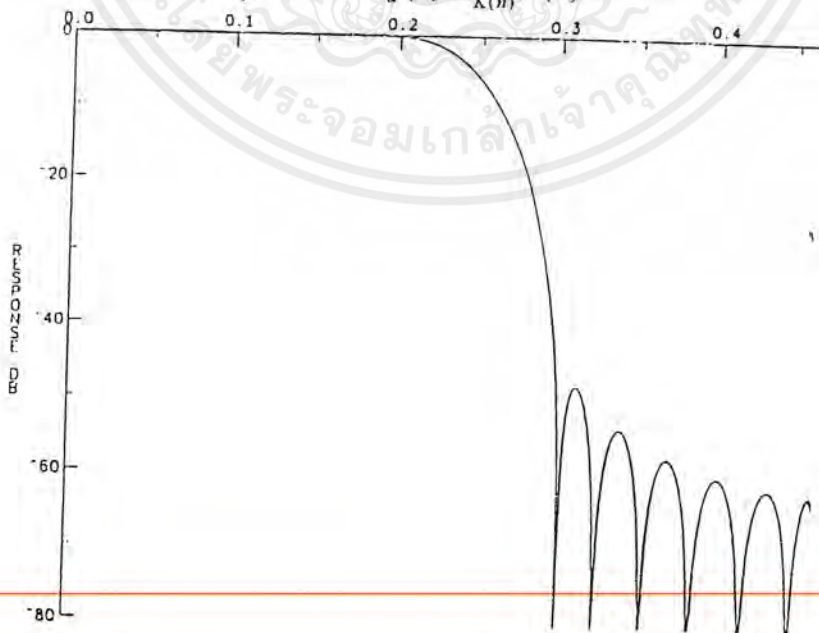
$$f_p = 500 \text{ Hz}$$

$$f_s = 750 \text{ Hz}$$

$$F = 2500 \text{ Hz}$$

คำตอบ

1. คำนวณค่าลูกลื่น จากสมการ 5.48 และ 5.49 $\delta = 0.005756$
2. คำนวณค่าลดทอนแถบหยุดน้อยสุด จากสมการ 5.44 $A_s = 44.796948$ เดซิเบล
3. หาค่า D จากสมการ 5.50 $D = 2.565943$
4. คำนวณอันดับตัวกรองจากสมการ 5.51 $N = 25.7$ (ปัดเป็น 27)
5. หาค่า α จากสมการ 5.49 $\alpha = 3.952354$
6. คำนวณผลตอบสนอง impulse ที่ปรับปรุงจากสมการ 5.56 , 5.39 และ 5.52 จะได้ผลตอบสนอง impulse ที่ไม่เป็นเหตุภาพของ $h_d(n), \alpha_{K(n)}, h(n)$



รูปที่ 3.11 แสดงการออกแบบตัวกรองผ่านความถี่ต่ำ Kaiser Window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\pm n$	$h_d(n)$	$a_{K(n)}$	$h(n)$
0	0.5	1.0	0.5
1	0.318310	0.989963	0.315115
2	0	0.960309	0
3	-0.106103	0.912385	-0.096807
4	0	0.848347	0
5	0.063662	0.771029	0.049085
6	0	0.683782	0
7	-0.045473	0.590272	-0.026841
8	0	0.494279	0
9	0.035368	0.399484	0.014129
10	0	0.309270	0
11	-0.028937	0.226556	-0.006556
12	0	0.153657	0
13	0.024485	0.092193	0.002257

ดังแสดงในรูป 3.10 ผลตอบสนองความถี่ต่อขนาด สำหรับวงจรกรองความถี่ต่ำสามารถหาได้จากสมการ 3.65

บทที่ 4

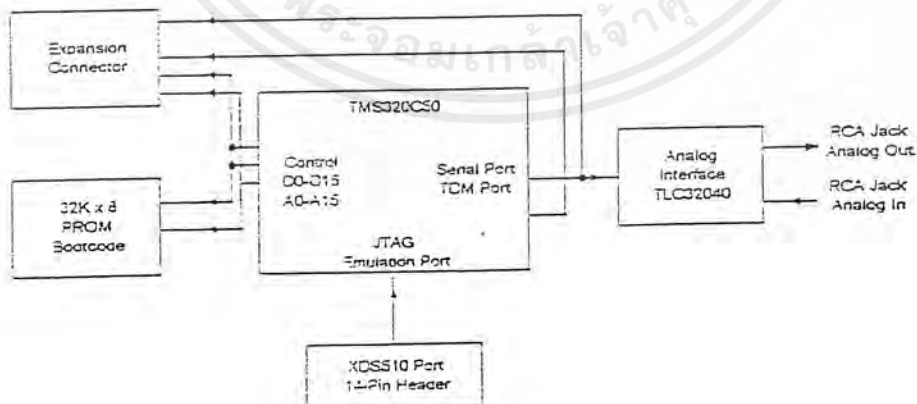
โครงสร้างของ TMS320C50 DSP Starter Kit

4.1 ลักษณะทั่วไปของบอร์ด

- ใช้ไมโครโพรเซสเซอร์เบอร์ TMS320C50 ซึ่งใช้งานโดยเฉพาะ
- แต่ละคำสั่งใช้เวลาประมาณ 50 ns (Instruction cycle)
- มี PROM 32 กิโลไบต์
- สามารถใช้ในช่วงความถี่เสียง โดยผ่านทาง TLC32040
- ใช้มาตรฐาน RCA คอนเนกเตอร์ สำหรับ อนาลอกอินพุต และ เอาท์พุต เพื่อให้สามารถต่อเข้ากับไมโครโฟน และลำโพงได้โดยตรง
- ในการติดต่อกับการควบคุม มีคอนเนกเตอร์ XDS510 ติดต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม
- สามารถขยาย I/O บัสได้ เพื่อสำหรับการออกแบบภายนอก

รูปที่ 4.1 แสดงบล็อกไดอะแกรมของบอร์ดซึ่งประกอบด้วย โซสอินเทอร์เฟซ ,อนาลอกอินเทอร์เฟซและอิมูเลชันพอร์ท ทำให้สามารถติดต่อกับพีซีได้โดยผ่านทาง RS232 นอกจากนี้ยังมี PROM ขนาด 32 Kbyte ที่ให้เก็บเคอร์เนลโปรแกรมไว้สำหรับบูทอีกด้วย

ในบอร์ดยังมี TLC 32040 วงจรอินเทอร์เฟซสัญญาณอนาลอก (Analog Interface Circuit :AIC) ที่มี RCA คอนเนกเตอร์ 2 ตัวสำหรับอินพุต และ เอาท์พุต



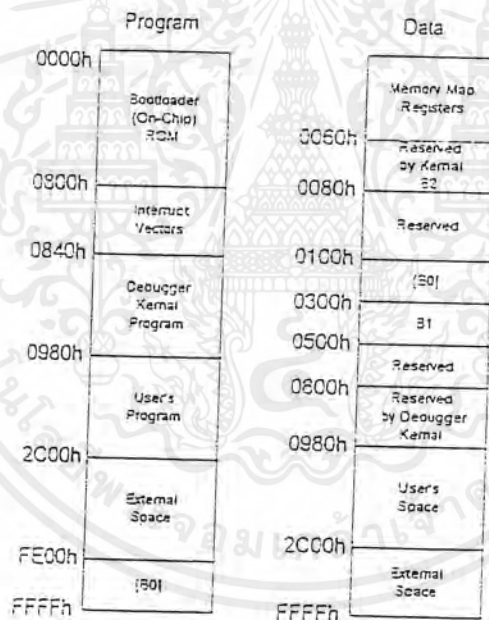
รูปที่ 4.1 บล็อกไดอะแกรมของ TMS320C50-DSK

หน่วยความจำใน TMS320C50 DSK นั้น จะแบ่งออกเป็นหน่วยความจำสำหรับโปรแกรม และหน่วยความจำสำหรับข้อมูลซึ่งในแต่ละแอดเดรสก็จะถูกใช้งานต่าง ๆ ตามแสดงในรูปที่ 4.2 และผู้ใช้สามารถใช้หน่วยความจำภายในได้ขนาด 10 k โดยไม่จำเป็นต้องต่อภายนอก

TLC32040 บนบอร์ดจะมีลักษณะดังนี้

- ใช้ในการแปลง A/D และ D/A ขนาด 14 บิต
- สามารถเปลี่ยนอัตราการแซมปลิงทั้งของ A/D ,D/A และความถี่ฟิลเตอร์ได้

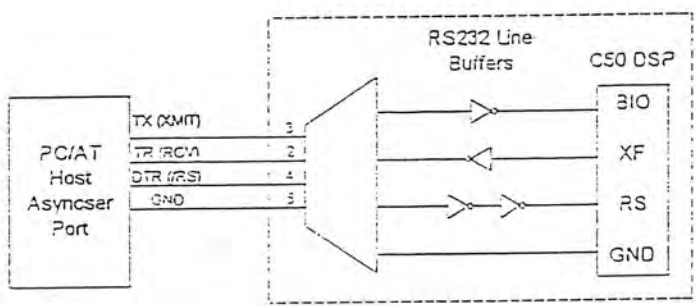
ต่อระหว่าง AIC และ TMS320C50 จะผ่านทางพอร์ตอนุกรมภายในบอร์ด



รูปที่ 4.2 แสดงหน่วยความจำภายใน TMS320C50 DSK

การต่อใช้งาน TMS320C50 DSK กับ PC จะใช้ขา XF และ BIO ที่ต่อผ่านพอร์ตอนุกรม RS232 ดังแสดงในรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการต่อระหว่าง DSK และ PC โดยผ่านทาง RS232



บทที่ 5

ไมโครโปรเซสเซอร์ TMS320C50

TMS320C50 เป็นโปรเซสเซอร์ที่ทำงานด้าน DSP (Digital Signal Processing) ซึ่งตระกูล TMS320C5X จะมีอยู่หลายเบอร์เช่น TMS320C50 TMS320C51 และ TMS320C530 ซึ่งใช้ชิพของ TMS320C25 ร่วมสถาปัตยกรรมและปรับปรุงให้ดีขึ้นทั้งด้านความเร็วและความสะดวกอื่น ๆ

5.1 จุดเด่นของ TMS320C50

1. มีแรมบนบอร์ด 10K เวิร์ด
2. ทำงานได้รวดเร็ว 30-50 ns ต่อหนึ่งคำสั่ง
3. ใช้งานแทน TMS320C1X และ TMS320C2X
4. มีแรมสำหรับโปรแกรม/ข้อมูล ขนาด 9Kx16 bit
5. มีรอมขนาด 2Kx16 bit สำหรับการบูต
6. สามารถต่อหน่วยความจำภายนอกได้ถึง 224Kx16 bit ซึ่งประกอบด้วย 64 K หน่วยความจำสำหรับเก็บโปรแกรม 64 K หน่วยความจำเก็บข้อมูล 64 K สำหรับ I/O และอื่น ๆ อีก 64 K
7. มี ALU (Arithmetic Logic Unit) ,ACC (Accumulator) และ ACCB (Accumulator Buffer) ขนาด 32 บิต
8. มี PLU (Parallel Logic Unit) ขนาด 16 บิต
9. มีคำสั่งในหารคูณ 16 บิต ที่ทำงานใน 1 ไชเคล
10. มีรีจิสเตอร์ถึง 8 ตัว ในการคำนวณและเก็บค่า
11. มีสแต็ก (Stack)
12. มีคำสั่งสำหรับการเลื่อนบิต (Shift) ตั้งแต่ 0-16 บิต
13. มีการอ้างแอดเดรสแบบเซอร์คิวลาร์ (Circular) โดยการใช้เซอร์คิวลาร์บัฟเฟอร์ (Circular Buffer)
14. มีคำสั่งสำหรับการทำซ้ำ ซึ่งเฉพาะสำหรับการเคลื่อนย้ายบล็อกในคำสั่งเดียว
15. มีคำสั่งสำหรับเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำข้อมูลและหน่วยความจำ

โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. มีพอร์ตอนุกรมที่รับส่งแบบฟูลดูเพล็กซ์ (full duplex) สำหรับ TMS320C50 กับอุปกรณ์อื่น ๆ

17. มี Time-division Multiple (TDM) ของพอร์ตอนุกรม

18. สามารถกำหนดสัญญาณนาฬิกา โดยใช้ไทม์เมอร์ (Timer), เคาน์เตอร์ (Counter)

19. มีซอฟต์แวร์ได้ทั้งหยุด (Stop) , เริ่ม (Start) และรีเซ็ต (Reset)

20. มีพอร์ต I/O ได้ถึง 64K และมี 16 ตำแหน่งสำหรับการเข้าถึงหน่วยความจำ

21. ทำงานแบบไปป์ไลน์ (Pipeline)

22. สามารถผลิตสัญญาณนาฬิกา โดยการหารสัญญาณนาฬิกาที่รับมาได้

23. ใช้เทคโนโลยีของซีมอส (CMOS) และใช้ไฟเลี้ยงเพียง 5V

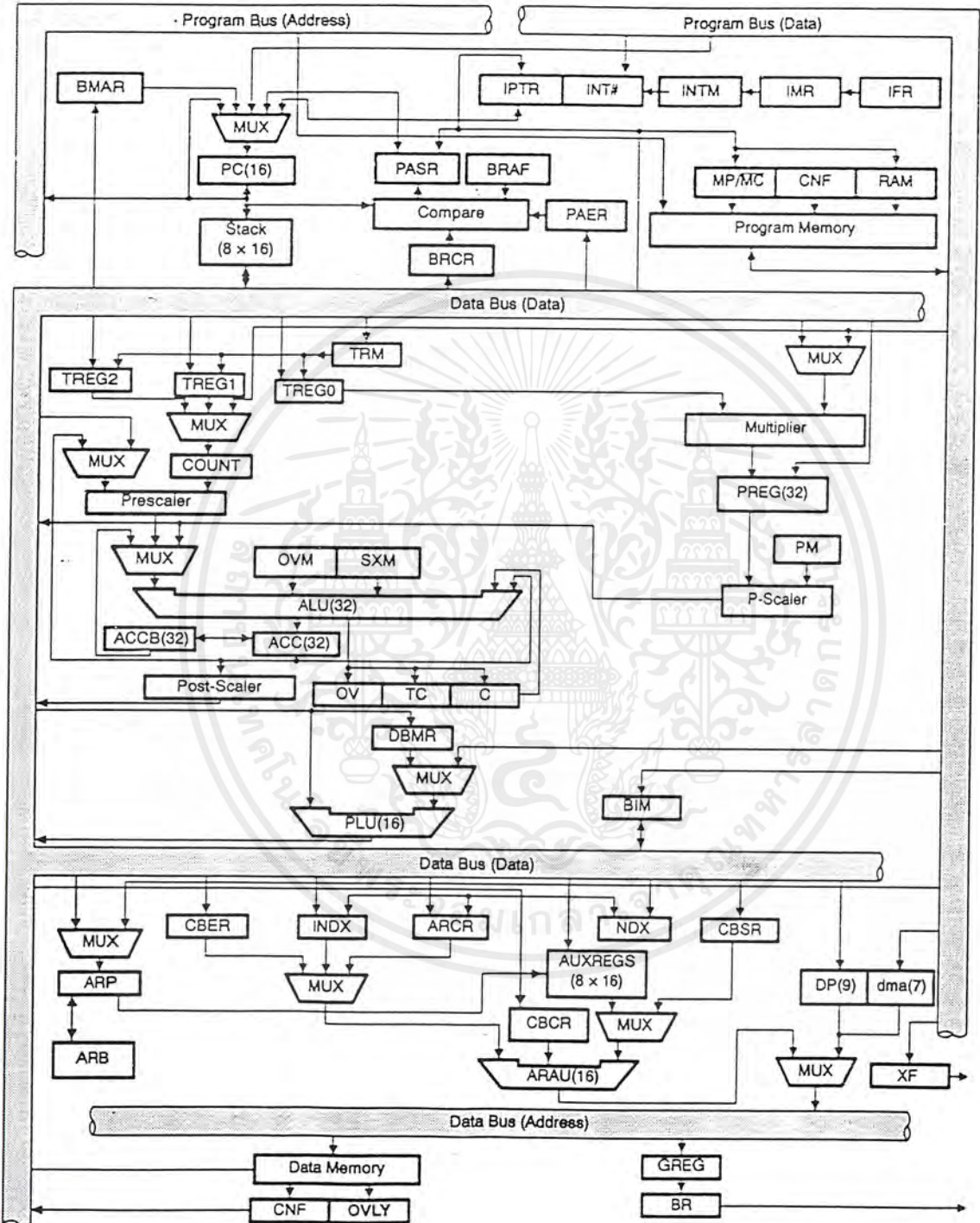
24. ไอซีมีขนาด 132 ขา

สถาปัตยกรรมของ TMS320C50 สร้างขึ้นมาเพื่อความเร็วในการทำงานด้าน DSP และเพื่อให้การทำงานของบัสไม่ขึ้นต่อกัน จึงแยกเป็นบัสโปรแกรม (Program Bus) และบัสข้อมูล (Data Bus) ออกจากกัน โดยบัสโปรแกรมจะเป็นทางเข้าของรหัสคำสั่งและโอเพอเรนด์ของคำสั่ง ส่วนบัสข้อมูลจะเชื่อมต่อโดยตรงกับวงจรการทำงานประมวลผล เช่น ALU และ ARO-AR7 ซึ่งโครงสร้างการคำนวณทางคณิตศาสตร์นี้ ยึดหลักการให้ทำงานด้วยประสิทธิภาพ เช่น การเลื่อนบิต การคูณ และคำสั่งทางลอจิก

นอกจากนี้ TMS320C50 ยังมีหน่วยประมวลผลกลาง (Central Arithmetic Logic Unit หรือ CALU) ไว้ทำงานด้านการคำนวณโดยเฉพาะ ซึ่งจะประกอบด้วยรีจิสเตอร์ต่าง ๆ คือ รีจิสเตอร์สำหรับการเลื่อน (Scaling Shifter) , วงจรคูณขนาด 16 x 16 บิต (16 x 16-bit Parallel Multiplier) , หน่วยประมวลผลทางคณิตศาสตร์ขนาด 32 บิตที่ช่วยให้การคำนวณทำได้อย่างรวดเร็ว รูปที่ 5.2 แสดงบล็อกไดอะแกรมส่วนประกอบของ CALU ซึ่งขั้นตอนโดยทั่วไปในการทำงานคำสั่งที่เกี่ยวกับการคำนวณ คือ

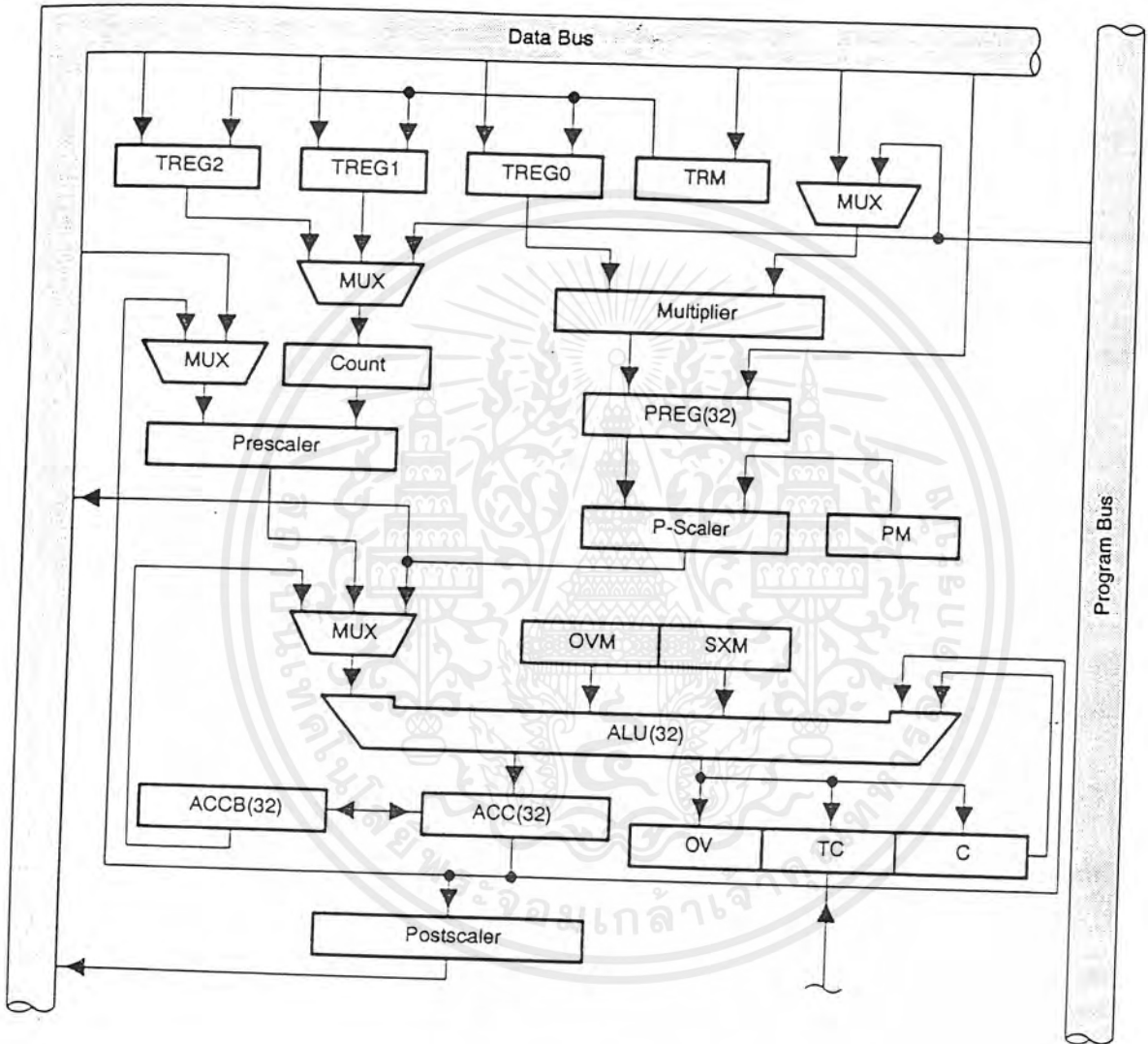
1. ข้อมูลถูกนำมาจากหน่วยความจำมาไว้บนบัสข้อมูล
2. ข้อมูลผ่านไปยังรีจิสเตอร์สำหรับการเลื่อน และ ALU เพื่อใช้คำนวณตามคำสั่ง
3. ผลการคำนวณที่ได้จะเก็บไว้ในแอคคิวมูเลเตอร์

ข้อมูลที้อนพุทที่ใช้ในการคำนวณในคำสั่งเกี่ยวกับการคำนวณจะเป็นข้อมูลในแอคคิวมูเลเตอร์ และอินพุตอีกตัวอาจจะมาจากรีจิสเตอร์ที่เก็บผลลัพธ์การคูณ (PREG) , รีจิสเตอร์สำหรับการเลื่อน , แอคคิวมูเลเตอร์บัพเฟอร์ หรือ อาจจะถูกโหลดจากหน่วยความจำข้อมูล หรือ อาจจะเป็นข้อมูลในแอคคิวมูเลเตอร์เองก็ได้



รูปที่ 5.1 แสดงบล็อกไดอะแกรม TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

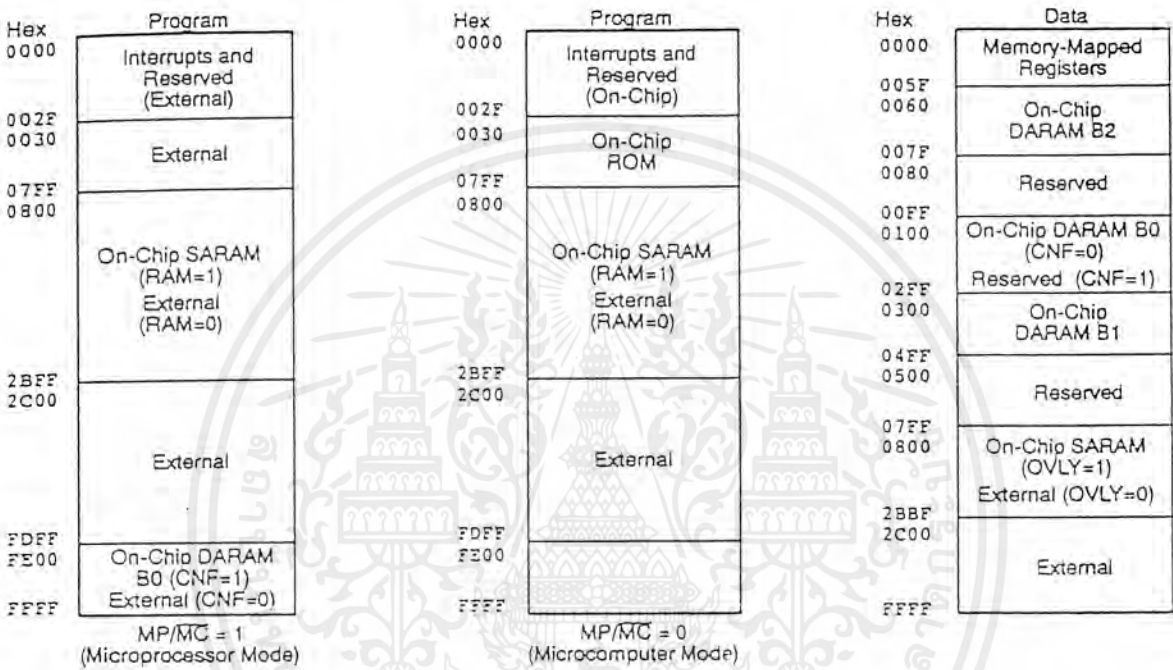


รูปที่ 5.2 แสดงบล็อกไดอะแกรมส่วนของหน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การจัดหน่วยความจำ TMS320C50

TMS320C50 จะมีการแบ่งหน่วยความจำเป็นของหน่วยความจำโปรแกรม และ หน่วยความจำข้อมูล ตามรูปที่ 5.3



รูปที่ 5.3 แสดงการเข้าถึงหน่วยความจำของ TMS320C50

5.2.1 หน่วยความจำข้อมูล (Data Memory)

ในการออกแบบของ TMS320C50 จะใช้สถาปัตยกรรมแบบฮาร์ดแวร์ (Hardware) กล่าวคือ มีการแยกหน่วยความจำข้อมูลและหน่วยความจำโปรแกรมออกจากกัน ซึ่งทำให้แยกส่วนอย่างชัดเจนซึ่งมีผลดีต่อความเร็ว

หน่วยความจำข้อมูลของ TMS320C50 สามารถขยายได้ถึง 64 Kx16 ซึ่ง TMS320C50 มีแรมบนชิปถึง 9 K ซึ่งแรมเป็นแบบ SARAM (Single Access RAM) และ DARAM (Dual Access RAM) ขนาด 1056 เวิร์ด บนชิปซึ่งมีข้อดีคือ

1. มีการปฏิบัติการได้เร็วไม่มีการรอ (Wait state)
2. ทำงานโดยวิธีการไปป์ไลน์ ทำให้ทำงานได้รวดเร็ว
3. ช่วยลดค่าใช้จ่ายเรื่องการต่อหน่วยความจำภายนอก
4. ช่วยประหยัดไฟ ซึ่งถ้าใช้การต่อหน่วยความจำภายนอกจะเปลืองกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการกำหนดการใช้หน่วยความจำข้อมูลจะต้องเช็คค่าของบิตรีจิสเตอร์ควบคุมหน่วยความจำ(Control Register) ดังนี้

ตารางที่ 5.1 การเซตบิตรีจิสเตอร์ควบคุมหน่วยความจำเพื่อใช้หน่วยความจำข้อมูล

CNF	OVLY	DARAM B0	DARAM B1	DARAM B2	SARAM	Off-Chip
0	0	100h–2FFh	300h–4FFh	60h–7Fh		800h–FFFFh
0	1	100h–2FFh	300h–4FFh	60h–7Fh	800h–2BFFh	2C00h–FFFFh
1	0	–	300h–4FFh	60h–7Fh		800h–FFFFh
1	1	–	300h–4FFh	60h–7Fh	800h–2BFFh	2C00h–FFFFh

เนื่องจากการติดต่อกับรีจิสเตอร์ต่างๆและการส่งค่าต่าง ๆ จะทำบนเพจ 0 ซึ่งจะเป็นการส่งแบบรีจิสเตอร์ (Register Memory Map) ดังตารางที่ 5.2

ตารางที่ 5.2 ตำแหน่งแอดเดรสข้อมูลเพจ 0

Name	Address		Description
	Dec	Hex	
Core Processor Memory-Mapped Registers			
—	0–3	0–3	Reserved
IMR	4	4	Interrupt Mask Register
GREG	5	5	Global Memory Allocation Register
IFR	6	6	Interrupt Flag Register
PMST	7	7	Processor Mode Status Register
RPTC	8	8	Repeat Counter Register
BRCR	9	9	Block Repeat Counter Register
PASR	10	A	Block Repeat Program Address Start Register
PAER	11	B	Block Repeat Program Address End Register
TREG0	12	C	Temporary Register Used for Multiplicand
TREG1	13	D	Temporary Register Used for Dynamic Shift Count (5 bits only)
TREG2	14	E	Temporary Register Used as Bit Pointer In Dynamic Bit Test (4 bits only)
DBMR	15	F	Dynamic Bit Manipulation Register
AR0	16	10	Auxiliary Register Zero
AR1	17	11	Auxiliary Register One
AR2	18	12	Auxiliary Register Two
AR3	19	13	Auxiliary Register Three
AR4	20	14	Auxiliary Register Four
AR5	21	15	Auxiliary Register Five
AR6	22	16	Auxiliary Register Six
AR7	23	17	Auxiliary Register Seven

ตารางที่ 5.2 ตำแหน่งแอดเดรสข้อมูลเพจ 0 (ต่อ)

INDX	24	18	Index Register
ARCR	25	19	Auxiliary Register Compare Register
CBSR1	26	1A	Circular Buffer 1 Start Register
CBER1	27	1B	Circular Buffer 1 End Register
CBSR2	28	1C	Circular Buffer 2 Start Register
CBER2	29	1D	Circular Buffer 2 End Register
CBCR	30	1E	Circular Buffer Control Register
BMAR	31	1F	Block Move Address Register
Peripheral Memory-Mapped Registers			
DRR	32	20	Data Receive Register
DXR	33	21	Data Transmit Register
SPC	34	22	Serial Port Control Register
—	35	23	Reserved

Peripheral Memory-Mapped Registers (Continued)

TIM	36	24	Timer Register
PRD	37	25	Period Register
TCR	38	26	Timer Control Register
—	39	27	Reserved
PDWSR	40	28	Program/Data S/W Wait-State Register
IOWSR	41	29	I/O Port S/W Wait-State Register
CWSR	42	2A	Control S/W Wait-State Register
—	43–47	2B–2F	Reserved for Test/Emulation
TRCV	48	30	TDM Data Receive Register
TDXR	49	31	TDM Data Transmit Register
TSPC	50	32	TDM Serial Port Control Register
TCSR	51	33	TDM Channel Select Register
TRTA	52	34	Receive/Transmit Address Register
TRAD	53	35	Received Address Register
—	54–79	36–4F	Reserved

Memory-Mapped I/O Ports

PA0	80	50	I/O Port 80
PA1	81	51	I/O Port 81
PA2	82	52	I/O Port 82
PA3	83	53	I/O Port 83
PA4	84	54	I/O Port 84
PA5	85	55	I/O Port 85
PA6	86	56	I/O Port 86
PA7	87	57	I/O Port 87
PA8	88	58	I/O Port 88
PA9	89	59	I/O Port 89
PA10	90	5A	I/O Port 90
PA11	91	5B	I/O Port 91
PA12	92	5C	I/O Port 92
PA13	93	5D	I/O Port 93
PA14	94	5E	I/O Port 94
PA15	95	5F	I/O Port 95
B2	96–127	60–7F	Scratch Pad RAM

เอกสารนี้เป็นเอกสารต้นฉบับที่ได้รับอนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 หน่วยความจำโปรแกรม (Program Memory)

ในการใช้งานของหน่วยความจำโปรแกรมซึ่งสามารถขยายได้ถึง 64K TMS320C50 มี รอม , SARAM และ DARAM ซึ่งมีความเร็วสูงโดยไม่มีสถานะการรอ

ในการทำงานของ TMS320C50 สามารถใช้ร่วมกับรอมภายในขนาด 4K สามารถ โปรแกรมได้จากโรงงานทำให้มีความเร็วในการทำงานเต็มที่ ในการเลือกใช้จะต้องกำหนดที่ขา ของ MP/MC หากขาดังกล่าวเป็นสถานะสูง ตำแหน่ง 4K แรกจะเป็นหน่วยความจำภายนอกชิพ แต่ถ้าเป็นสถานะต่ำ ตำแหน่ง 4K เวิร์ดแรกจะเป็นรอมภายในชิพ ดังรูปที่ 5.3

การกำหนดสถานะ (Status) ของค่าต่าง ๆ จะกำหนดตามตารางที่ 5.3

ตารางที่ 5.3 การกำหนดค่าสำหรับใช้หน่วยความจำโปรแกรม

CNF	RAM	MP/MC	ROM	SARAM	DARAM B0	Off-Chip
0	0	0	0000-07FF			0800-FFFF
0	0	1				0000-FFFF
0	1	0	0000-07FF	0800-2BFF		2C00-FFFF
0	1	1		0800-2BFF		0000-07FF 2C00-FFFF
1	0	0	0000-07FF		FE00-FFFF	0800-FDFF
1	0	1			FE00-FFFF	0000-FDFF
1	1	0	0000-07FF	0800-2BFF	FE00-FFFF	2C00-FDFF
1	1	1		0800-2BFF	FE00-FFFF	0000-07FF 2C00-FDFF

การใช้งานของการอินเตอร์รัพท์จะต้องกำหนดแอดเดรสของแอดเดรสในหน่วยความจำ โปรแกรมซึ่งอินเตอร์รัพท์แอดเดรสจะแสดงดังตารางที่ 5.4

ตารางที่ 5.4 ตำแหน่งของอินเทอร์รัพท์เวกเตอร์

Name	Location		Priority	Function
	Dec	Hex		
RS	0	0	1 (highest)	External reset signal
INT1	2	2	3	External user interrupt #1
INT2	4	4	4	External user interrupt #2
INT3	6	6	5	External user interrupt #3
TINT	8	8	6	Internal timer interrupt
RINT	10	A	7	Serial port receive interrupt
XINT	12	C	8	Serial port transmit interrupt
TRNT	14	E	9	TDM port receive interrupt
TXNT	16	10	10	TDM port transmit interrupt
INT4	18	12	11	External user interrupt #4
—	20–33	14–21	N/A	Reserved
TRAP	34	22	N/A	Software trap instruction
NMI	36	24	2	Nonmaskable interrupt
—	38–41	26–29	N/A	Reserved for emulation and test
—	42–47	2A–2F	N/A	Software interrupts

5.3 โหมดการอ้างถึงหน่วยความจำ (Memory Addressing Mode)

การอ้างแอดเดรสสามารถอ้างได้ 64K สำหรับหน่วยความจำโปรแกรม และ 96K สำหรับหน่วยความจำข้อมูล TMS320C50 มีวิธีการอ้างแอดเดรส ได้ถึง 8 วิธีดังนี้

- 1. วิธีเข้าถึงโดยตรง (Direct Address Bus) โดยการใช้แอดเดรสบัสโดยตรง ซึ่งต้องมี รีจิสเตอร์ชี้เพจของข้อมูล (Data Page Pointer (DP)) บอกด้วยว่าเป็นเพจใด ซึ่ง DP สามารถอ้างได้ 512 เพจ แต่ละเพจเท่ากับ 128 เวิร์ด ตัวอย่างเช่น ADD 01 0h หมายถึง เอาข้อมูลที่ชี้ที่ตำแหน่ง 010H (ซึ่งมี DP ชี้ออกว่าอยู่เพจใด) นำมาบวกกับ ACC
- 2. วิธีเข้าถึงหน่วยความจำ (Memory Map) จะเหมือนกับวิธีแรกแต่จะใช้เฉพาะเพจ 0 เท่านั้น ซึ่งเพจ 0 จะเป็นรีจิสเตอร์ที่ใช้เป็นรีจิสเตอร์ที่เป็นหน่วยความจำ (Memory Map Register) เช่น LAMM PMST โดยที่ PMST เป็นรีจิสเตอร์
- 3. วิธีเข้าถึงโดยใช้ออกซิลลารีรีจิสเตอร์ (Auxiliary register) โดยการอ้างแอดเดรสที่บรรจุใน AR0-AR7 เช่น ADD* ซึ่ง ARP จะเป็นตัวบอกว่าเป็น AR ตัวที่เท่าใด ตัวอย่างเช่น

ARP	4	4
AR4	0302h	0302h
หน่วยความจำ		
302h	2h	2h
ACC	2h	4h

จะเห็นว่าแอดเดรสที่ได้อยู่ใน AR4 (0302h)

4. วิธีใช้รีจิสเตอร์คำสั่ง (Instruction Register) ซึ่งเป็นการเอาแอดเดรสโดยตรงมาจากคำสั่งเลข เช่น

ADD # 0FFh เอาข้อมูลที่แอดเดรส FFh มาบวกกับ ACC

โดยวิธีนี้จะอ้างได้แบบสั้น (Short Immediate) คือ ค่าแอดเดรส จะเป็น 00-FF

5. โดยการใช้ PC (Program Counter) จะเหมือนการใช้รีจิสเตอร์คำสั่ง แต่สามารถอ้างแบบยาว (Long Immediate (0000-FFFF)) เช่น

ADD # 0FFFh

6. โดยเข้าถึงรีจิสเตอร์โดยตรง (Register Access) เป็นการใช้อรีจิสเตอร์ที่ทำงานโดยเฉพาะ เช่น TREG0,TREG1,TREG2,ARCR,DBMR ตัวอย่าง คำสั่ง

APL *,AR7

7. โดยใช้คำสั่งตัวที่ 2 เป็นตัวชี้แอดเดรส เช่น คำสั่ง BLDD # 02345h, 012h หมายถึง เอาข้อมูลที่ชี้แอดเดรส 02345h ไปเก็บที่แอดเดรส 012h (คิดป็นเพจ)

8. โดยอ้างถึงหน่วยความจำเป็นบล็อก (Block Memory Address Register) ซึ่ง BMAR จะเป็นตัวชี้ข้อมูล ตัวอย่าง เช่น

BLDD BMAR, 012h

5.4 อุปกรณ์เสริม (Peripherals)

TMS320C50 มีการอินเตอร์เฟสกับภายนอกได้ถึง 7 อุปกรณ์ ซึ่งประกอบด้วย

1. พอร์ตอนุกรม (Serial Port)
2. พอร์ตอนุกรมแบบแบ่งเวลา (TDM Serial Port)
3. ไทม์เมอร์ (Timer)
4. รีจิสเตอร์ใช้โปรแกรมสถานะการรอ (Software-programable Wait State)
5. พอร์ตติดต่อภายนอก (I/O Port)
6. รีจิสเตอร์หารสัญญาณนาฬิกา (Divide-by-one Clock)
7. XF และ BIO

ซึ่งทั้งหมดจะควบคุมผ่านรีจิสเตอร์ควบคุมโดยใช้วิธีการอ้างหน่วยความจำโดยการส่งลงที่แอดเดรสของหน่วยความจำโปรแกรมและข้อมูล

5.4.1 อินเทอร์รัพท์ (Interrupt)

TMS320C50 มีอินเทอร์รัพท์ทั้งหมด 16 อินเทอร์รัพท์ (INT16-INT1) แต่จะไม่ใช่พร้อมกันทั้งหมด (ปกติใช้ 9)

การรีเซต (Reset)

จะเป็นอินเทอร์รัพท์แบบนอน-มาร์กเอเบิลจากภายนอก (Nonmaskable External Interrupt)

ซึ่งจะเกิดขึ้นเมื่อมีการรีเซต ดังนี้

1. CNF = 0 หมายถึง การใช้หน่วยความจำข้อมูลบนชิพ
2. PC = 0000H
3. INTM = 1 และ IFR = 0 ทำให้อินเทอร์รัพท์ทุกตัวไม่ทำงาน
4. บิตสถานะ (Status Bit) จะเป็นดังนี้

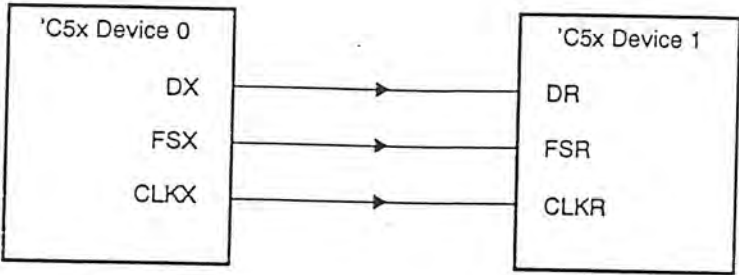
0 → OV	1 → XF	1 → SXM	0 → PM	1 → HM
0 → BRA1	0 → TRM	0 → NDX	0 → CENMI	0 → CENM2
0 → IPTR	0 → OVLY	0 → AVIS	0 → RAM	0 → BIG
0 → CNF	1 → INTM	MP/MC (pin) → PMST (MP/MC)	1 → C	
5. GREG = 000000
6. RPTC = 00
7. LACK จะแสดงสถานะการเก็บการรีเซต

ในการควบคุมการอินเทอร์รัพท์จะมี IFR (Interrupt Flag Register) แสดงสถานะอินเทอร์รัพท์ และ IMR (Interrupt Mask Register) แสดงการมาร์กการอินเทอร์รัพท์

ในการกำหนดให้ใช้อินเทอร์รัพท์จะต้องทำการเคลียร์ที่ INTM บิต ด้วย

5.4.2 พอร์ตอนุกรม (Serial Port)

พอร์ตอนุกรมของ TMS320C50 จะเป็นแบบฟูลดuple็กซ์ คือ สามารถรับและส่งภายในเวลาเดียวกัน ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกอื่น ๆ ได้ ไม่ว่าจะเป็น CODEC A/D หรือระบบอื่น ๆ ในการต่อขาต่าง ๆ จะเป็นตามรูปที่ 5.4



รูปที่ 5.4 แสดงการต่อพอร์ตอนุกรมกับพอร์ตภายนอก

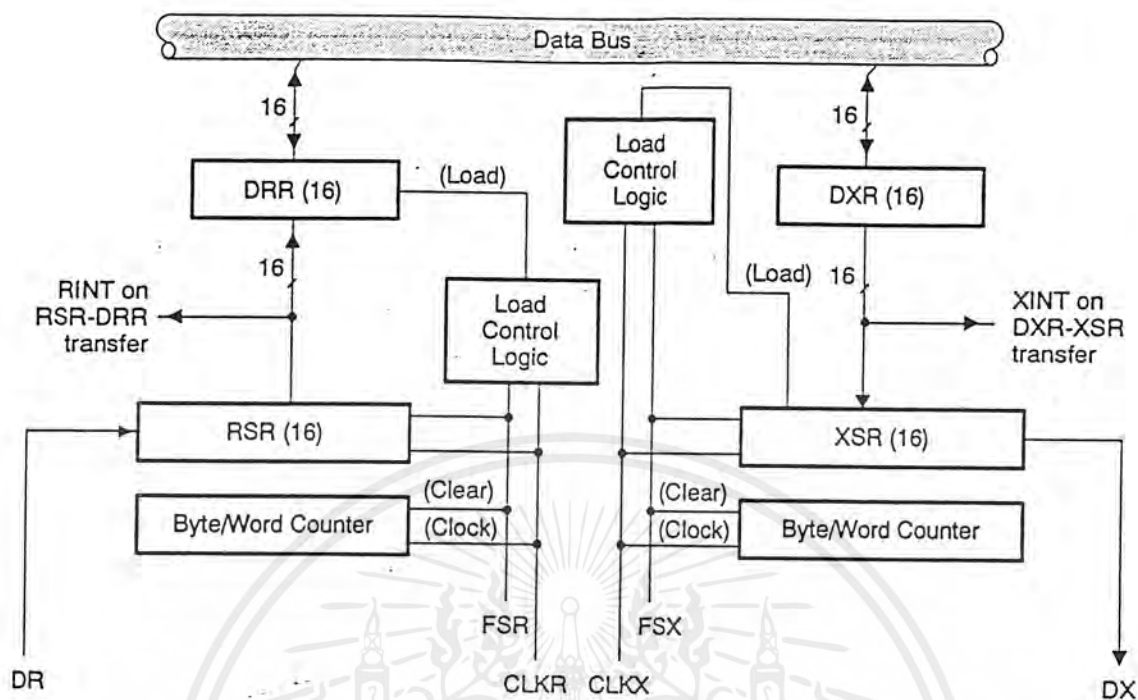
ตารางที่ 5.5 ขาสัญญาณของพอร์ตอนุกรม

ขาสัญญาณ	ความหมาย
CLKX	สัญญาณคล็อกที่ใช้ในการส่ง
CLKR	สัญญาณคล็อกที่ใช้ในการรับ
DX	สัญญาณข้อมูลอนุกรมในการส่ง
DR	สัญญาณข้อมูลอนุกรมในการรับ
FSX	สัญญาณเชิงโครโมโซมในการส่ง
FSR	สัญญาณเชิงโครโมโซมในการรับ

ในการควบคุมพอร์ตอนุกรมจะมีรีจิสเตอร์ที่จำเป็นตามตารางที่ 5.6

ตารางที่ 5.6 รีจิสเตอร์พอร์ตอนุกรม

รีจิสเตอร์	ความหมาย
SPC	รีจิสเตอร์ควบคุมพอร์ตอนุกรม
DXR	รีจิสเตอร์ส่งข้อมูล
DRR	รีจิสเตอร์รับข้อมูล
XSR	รีจิสเตอร์เลื่อนการส่ง
RSR	รีจิสเตอร์เลื่อนการรับ



ในการส่งข้อมูลทำได้โดยการเขียนข้อมูลลงใน DXR (Data Transmit Register) โดย XSR (Transmit Shift Register) จะเป็นตัวเลื่อนบิตตามสัญญาณนาฬิกาจนครบไปตามขา DX ส่วนการรับข้อมูลจะรับมาทางขา DR เข้ามาที่ RSR (Receive Shift Register) เมื่อครบแล้วจะทำการอินเทอร์รัพท์บอกให้ทราบ แล้วสามารถรับข้อมูลโดยการอ่านข้อมูลจาก DRR (Data Receive Register)

5.4.3 ไทม์เมอร์ (Timer)

ในการใช้ใหม่เมอร์ของ TMS320C50 ซึ่งจะสามารถตั้งคาบเวลาของสัญญาณนาฬิกาได้
โดยสัญญาณจะออกที่ขา Tout ของ TMS 320C50

ซึ่งการคำนวณคาบเวลาจะเป็นตามสูตร

$$T_{INT} = 1 / [t_c \times (TDDR + 1) \times (PRD + 1)]$$

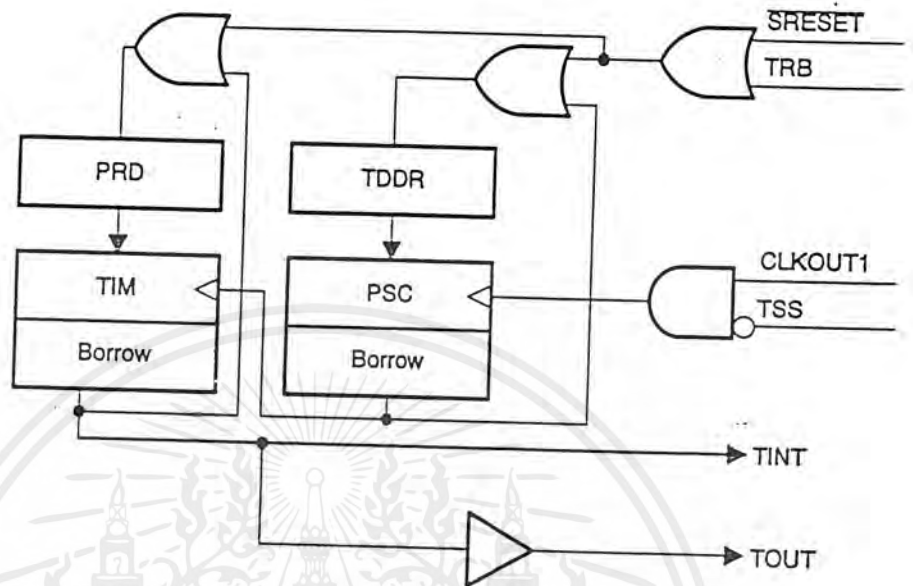
$$(5.1)$$

ซึ่ง t_c เป็นคาบเวลาของสัญญาณพิกที่จ่ายให้ที่ขา CLKOUT1

TDDR (Timer Divide Down Ratio)

PRD (Period Register) เป็นรีจิสเตอร์ที่สามารถใส่ค่าคาบเวลาได้

ในการทำงานจะต้องมีการใส่ค่าต่าง ๆ ลงบน TCR (Timer Control Register) ดังตารางที่ 5.7



รูปที่ 5.6 แสดง บล็อกไดอะแกรมของไทม์เมอร์

ตารางที่ 5.7 รีจิสเตอร์ควบคุมไทม์เมอร์ (TCR)

15-12	11	10	9-6	5	4	3-0
Reserved	SOFT	FREE	PSC	TRB	TSS	TDDR

5.5 ตัวสร้างสถานะรอที่สามารถโปรแกรมได้ (Software-Programmable Wait-State Generators)

ตัวสร้างสถานะรอที่สามารถโปรแกรมได้ ใช้สำหรับเพิ่มไซเคิลของสัญญาณนอกได้ถึง 7 แมกซีนไซเคิล ทำให้สะดวกในการอินเทอร์เฟสกับอุปกรณ์ภายนอกที่ไม่สามารถทำงานได้เร็วเท่ากับตัว TMS320C50 ในกรณีที่อุปกรณ์ภายนอกต้องการสถานะรอ (wait states) มากกว่า 7 ไซเคิลก็สามารถทำได้โดยอินเทอร์เฟส ผ่านทางขาสัญญาณ READY เมื่อทุกอุปกรณ์ภายนอกอยู่สามารถติดต่อกับ TMS320C50 ได้โดยไม่ต้องมีการสร้างสถานะรอ นาฬิกาภายใน (Internal Clock) ที่ส่งสัญญาณนาฬิกาให้แก่ตัวสร้างสถานะรอจะปิดลง (Shuff Off) ทำให้ TMS320C50 ทำงานอยู่ในสภาวะพลังงานต่ำ (Lower Power Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวสร้างสถานะรอที่สามารถที่โปรแกรมได้ จะถูกควบคุมโดย รีจิสเตอร์สถานะรอ 16 บิต 2 ตัว (PDWSR และ IOWSR) และ รีจิสเตอร์ควบคุม 5 บิต 1 ตัว (CWSR) แต่ละพื้นที่ภายนอก (โปรแกรม ,ข้อมูล ,อินพุต/เอาพุต) จะถูกกำหนดไว้ในฟิลด์ในรีจิสเตอร์สถานะ โดยสถานะรอสำหรับโปรแกรมและข้อมูลจะถูกกำหนดไว้ในส่วนครึ่งล่างและครึ่งบนของ IOWSR แต่ละบิตใน CWSR จะถูกกำหนดให้เป็นตัวบอกรีจิสเตอร์สถานะรอ และ จำนวนของสถานะรอ

แต่ละ พื้นที่โปรแกรม และ ข้อมูล จะอ้างอิงพื้นที่ได้ 64K แต่ละ 64K จะถูกแบ่งออกเป็น 16K WORD 4 บล็อก แต่ละ 16K ในพื้นที่โปรแกรม และ ข้อมูล จะเกี่ยวข้องกับ บิต 2 บิต ใน PDWSR ในแสดงในตารางที่ 5.8 ค่าที่ถูกกำหนดใน 2 บิต ที่กล่าวถึงข้างต้นจะเป็นตัวบอกว่าแอดเดรส และ พื้นที่ ไหนที่จะถูกเพิ่มสถานะรอ

ตารางที่ 5.8 แสดงค่าในแต่ละบิตรีจิสเตอร์สถานะรอและแอดเดรสที่เกี่ยวข้อง

รีจิสเตอร์	บิตที่	พื้นที่	แอดเดรสที่อ้างอิงถึง
PDWSR	0 - 1	โปรแกรม	0000h-3FFFh
	2 - 3		4000h-7FFFh
	4 - 5		8000h-0BFFFh
	6 - 7		0C000h-0FFFFh
	8 - 9	ข้อมูล	0000h-3FFFh
	10 - 11		4000h-7FFFh
	12 - 13		8000h-0BFFFh
	14 - 15		0C000h-0FFFFh
IOWSR	0 - 1	I/O	BIG = 0 Port 0/1,Port 10/11, etc.
			BIG = 1 0000h-1FFFh
	2 - 3		Port 2/3,Port 12/13, etc.
			2000h-3FFFh
	4 - 5		Port 4/5,Port 14/15, etc.
			4000h-5FFFh
	6 - 7		Port 6/7,Port 16/17, etc.
			6000h-7FFFh
	8 - 9		Port 8/9,Port 18/19, etc.
			8000h-9FFFh
	10 - 11		Port 0A/0B,Port 1A/1B, etc.
			0A000h-0BFFFh
	12 - 13		Port 0C/0D,Port 1C/1D, etc.
			0C000h-0DFFFh
	14 - 15		Port 0E/0F,Port 1E/1F, etc.
			0E000h-0FFFFh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์สถานะของพื้นที่ I/O (IOWSR) สามารถกำหนดการอ้างอิงได้ 2 แบบ ขึ้นอยู่กับ บิต BIG ใน CWSR รีจิสเตอร์ ถ้า บิต BIG = 0 I/O พอร์ตทั้ง 8 คู่จะมีตำแหน่งในแต่ละ 2 บิตฟิลด์ ใน IOWSR ดังแสดงในตารางที่ 5.8 สังเกตได้ว่าในกรณีที่ BIG = 0 พื้นที่ I/O ทั้งหมดจะถูกกำหนดสถานะรอในขอบเขต 2 เวอร์ด (ตัวอย่างเช่น Port 0/1 Port 10/11 และ Port 20/21 จะมีสถานะการรอเท่ากัน) การกำหนดในลักษณะนี้ทำให้มีความสะดวกอย่างมากเมื่อใช้เคิล I/O บัส เกี่ยวข้องกับอุปกรณ์เสริมภายนอก เช่น D/A และ A/D อย่างไรก็ตาม ถ้าการติดต่อกับอุปกรณ์ต้องมีการอ่าน หรือ เขียน ก็สามารถทำได้โดยการเซตให้บิต BIG = 1 (ตัวอย่างเช่น RAM ภายนอก) ในกรณีนี้ พื้นที่ I/O ทั้ง 64K สามารถแบ่งได้เป็น บล็อกขนาด 8K เวอร์ด จำนวน 8 บล็อก ซึ่งแต่ละบล็อกก็สามารถโปรแกรมจำนวนของสถานะรอโดยอิสระได้

สังเกตเห็นว่า ตัวสร้างสถานะรอ จะมีผลต่อการติดต่อภายนอกเท่านั้น การติดต่อภายในจะมีค่าสถานะรอเท่ากับ 0 เสมอ

บิตทั้ง 4 ใน CWSR ใช้สำหรับอนุญาตให้ผู้ใช้สามารถเลือกการกำหนดระหว่าง 2 บิตสถานะรอ และ จำนวนของสถานะรอ สำหรับพื้นที่ที่เกี่ยวข้อง ดังแสดงในตารางที่ 5.9 ถ้าบิตใดใน CWSR เท่ากับ 0 การกำหนดระหว่าง ค่าฟิลด์สถานะรอ และ จำนวนของสถานะรอ จะเกี่ยวข้องกันโดยตรง จำนวนของสถานะรอสำหรับการติดต่ออุปกรณ์ภายนอกในพื้นที่ที่เกี่ยวข้องกับคอนโทรลบิตนั้นจะเท่ากับค่าในฟิลด์ในสถานะรอ ถ้าคอนโทรลบิตใน CWSR เท่ากับ 1 จำนวนของสถานะรอจะถูกกำหนดดังแสดงในตารางที่ 5.9 ตารางที่ 5.10 แสดงการกำหนดค่าของ PDWSR และ IOWSR รีจิสเตอร์ใน CWSR รีจิสเตอร์ เราสามารถโปรแกรม CWSR รีจิสเตอร์ เพื่อกำหนดให้ PDWSR และ IOWSR รีจิสเตอร์ เพื่อป้องกันไม่ให้หน่วยความจำมีค่าสถานะรอที่น้อยเกินไปในขณะที่เริ่มตั้งค่าในรีจิสเตอร์สถานะรอ

ตารางที่ 5.9 ค่าฟิลด์สถานะรอ และ สถานะรอ ที่เกี่ยวข้องกับ บิต n ใน CWSR

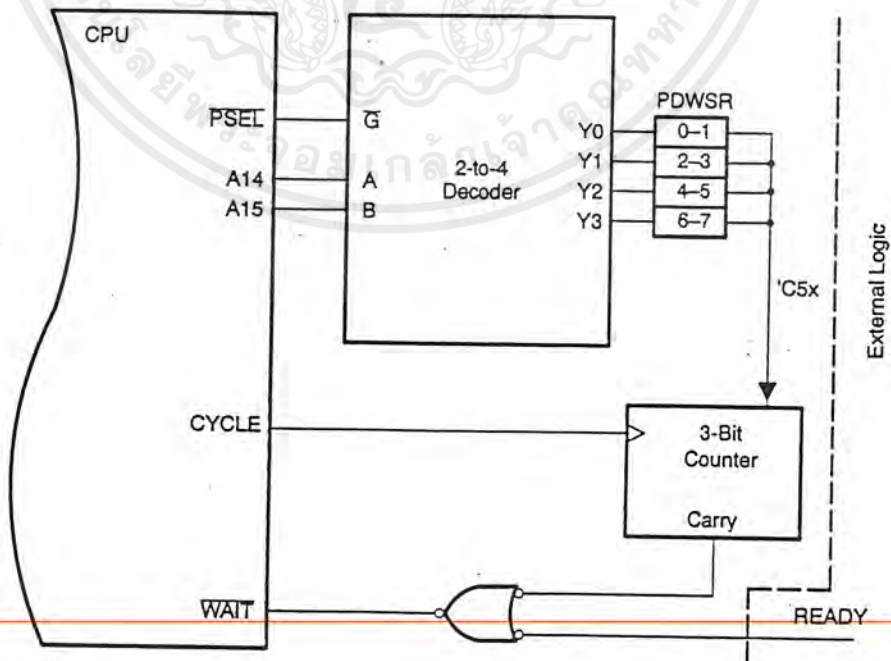
ค่าฟิลด์สถานะรอ ของ PDWSR หรือ IOWSR (ค่าเลขฐานสอง)"	จำนวนของสถานะรอ (CWSR บิต n = 0)	จำนวนของสถานะรอ (CWSR บิต n = 1)
00	0	0
01	1	1
10	2	3
11	3	7

" ค่าบิตฟิลด์จะเกี่ยวข้องกับบิตฟิลด์ในคอลัมน์ที่สองในตารางที่ 5.8

ตารางที่ 5.10 การควบคุมพื้นที่โดย CWSR บิต n

n (ตำแหน่งบิตใน CWSR)	พื้นที่
0	โปรแกรม
1	ข้อมูล
2	I/O (ครึ่งล่าง : Port 0 - Port 7 ถ้า BIG = 0 ,0000h - 7FFFh ถ้า BIG = 1)
3	I/O (ครึ่งบน : Port 8 - Port F ถ้า BIG = 0 ,8000h - 0FFFFh ถ้า BIG = 1)
4	BIG Mode Bit

รูปที่ 5.7 แสดงบล็อกไดอะแกรมของตัวสร้างสถานะรอสำหรับพื้นที่โปรแกรมภายนอก เมื่อมีการติดต่อกับโปรแกรมภายนอกถูกดีโคด ค่าในฟิลด์ที่เหมาะสมของ รีจิสเตอร์สถานะรอ PDWSR จะถูกโหลดเข้าไปไว้ในเคาน์เตอร์ ถ้าฟิลด์ไม่เป็น 000 สัญญาณบอกว่าไม่พร้อมจะถูกส่งไปยัง CPU สถานะไม่พร้อมจะคงอยู่จนกว่าเคาน์เตอร์จะถูกลดค่าลงเป็นศูนย์ และ สาย READY ภายนอกจะมีค่าสูง สาย READY ภายนอก และ รีจิสเตอร์สถานะรอจะถูก OR เข้าด้วยกันเพื่อสร้างสัญญาณ WAIT ให้แก่ CPU หลังจากทีรีเซ็ต รีจิสเตอร์ที่ใช้ควบคุมสถานะรอจะถูกเซตให้เป็น 7 CWSR จะถูกเซตให้เป็น 0Fh และ BIG บิตของ CWSR รีจิสเตอร์จะถูกเซตให้เป็น 0



รูปที่ 5.7 แสดงบล็อกไดอะแกรมของตัวสร้างสถานะรอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

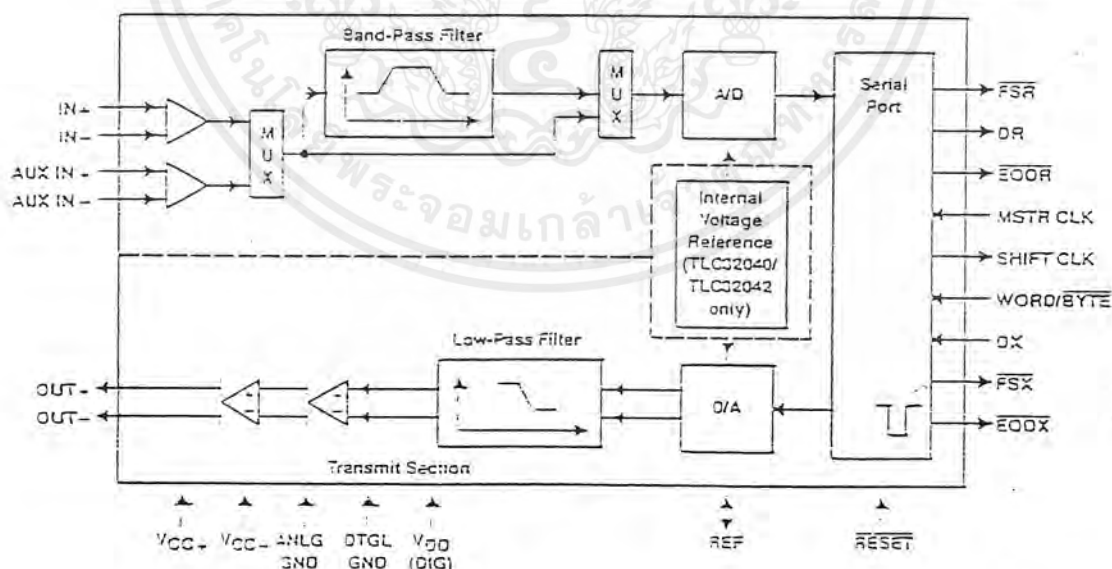
วงจรอินเทอร์เฟซสัญญาณอนาล็อก TLC 32040

ในการทำงานของ DSP จะใช้ชิพไอซี TLC32040 ซึ่งเป็นชิพซีมอสเพื่อแปลงสัญญาณอนาล็อกเป็นดิจิทัลและสามารถแปลงจากดิจิทัลเป็นอนาล็อกได้ด้วย

คุณสมบัติสำคัญของ TLC 32040 มีดังนี้

- ใช้เทคโนโลยีการผลิต Advanced LinCMOSTM Silicon - Gate Process
- ความละเอียดของ ADC และ DAC เป็น 14 บิต
- ความสามารถเปลี่ยนอัตราการ Sampling ของ ADC และ DAC ได้ถึง 19,200 ครั้ง
- มี Switched -Capacitor Antialiasing Input Filter และ Output- Reconstruction Filter
- มีพอร์ตสำหรับติดต่อโดยตรงกับ TMS3211, TMS320C17, TMS320C20 และ TMS320C25 DSP
- สามารถปรับอัตราการแปลงของ ADC และ DAC ทั้งแบบซิงโครนัส หรือ อซิงโครนัส โดยการใช้ Software

บล็อกไดอะแกรมของการทำงาน (Functional Block Diagram)



รูปที่ 6.1 แสดงบล็อกไดอะแกรมของการทำงานของ TLC 32040

6.1 การทำงานของ TLC32040

อินพุตทางอนาล็อก (Analog Input)

สำหรับอนาล็อกอินพุตจะมี 2 กลุ่ม คือ $IN+$, $IN-$ และ $AUX\ IN+$, $AUX\ IN-$ ซึ่งสามารถเลือกใช้กลุ่มใดกลุ่มหนึ่งก็ได้ โดยสามารถทำงานแยกกัน หรือ แบบ Single-ended mode ได้ เพราะสามารถทำได้ทั้งในแบบ common -mode และ Rejection mode Range และค่าเกนของอินพุต $IN+$, $IN-$ และ $AUX\ IN+$, $AUX\ IN-$ สามารถกำหนดค่าได้ มีทั้ง 3 ค่า คือ 1, 2 และ 4 การเลือกใช้กลุ่มอินพุตใดจะเลือกโดยใช้ Software ควบคุม

A/D Bandpass Filter , A/D Bandpass Filter Clocking , และ A/D conversion Timing

สำหรับ A/D แบนด์พาส ฟิลเตอร์ (A/D Bandpass Filter) สามารถที่จะเลือกใช้หรือไม่ใช้ก็ได้โดยฟังก์ชันถ่ายโอน (Transfer Function) ของฟิเตอร์โดยจะคิดอัตราส่วนจากความถี่สัญญาณพิกัดความควบคุมฟิเตอร์ 228 kHz ที่ความถี่ต่ำที่เริ่มมีลักษณะเป็นความถี่สูงผ่านจะมีความถี่เป็น 300 Hz อัตราการแปลง D/A ก็หาได้จากความถี่ที่หาร 228 kHz ด้วย RX เคนเตอร์ B

เอาต์พุตทางอนาล็อก (Analog Output)

อนาล็อกเอาต์พุตจะมีวงจรขยายกำลัง (Power Amplifier) มีเอาต์พุตทั้งแบบ Non-inverting และแบบ Inverting เนื่องจากมีการขยายกำลังสามารถขับ ทรานสเฟอร์เมอร์ชนิดไฮ-บริด (Transformer Hybrid) หรือโหลดอิมพีแดนซ์ต่ำได้โดยใช้ทั้งแบบ Differential และแบบ Single-ended ได้วงจรรองความถี่ต่ำของ D/A , วงจรรองความถี่แบบแบนด์พาสของ D/A . สัญญาณพิกัดควบคุมวงจรรองความถี่ต่ำ และอัตราการแปลงสัญญาณดิจิตอลเป็น อนาล็อก (D/A Lowpass Filter,A/D Bandpass Filter Clocking , และ A/D conversion Timing)

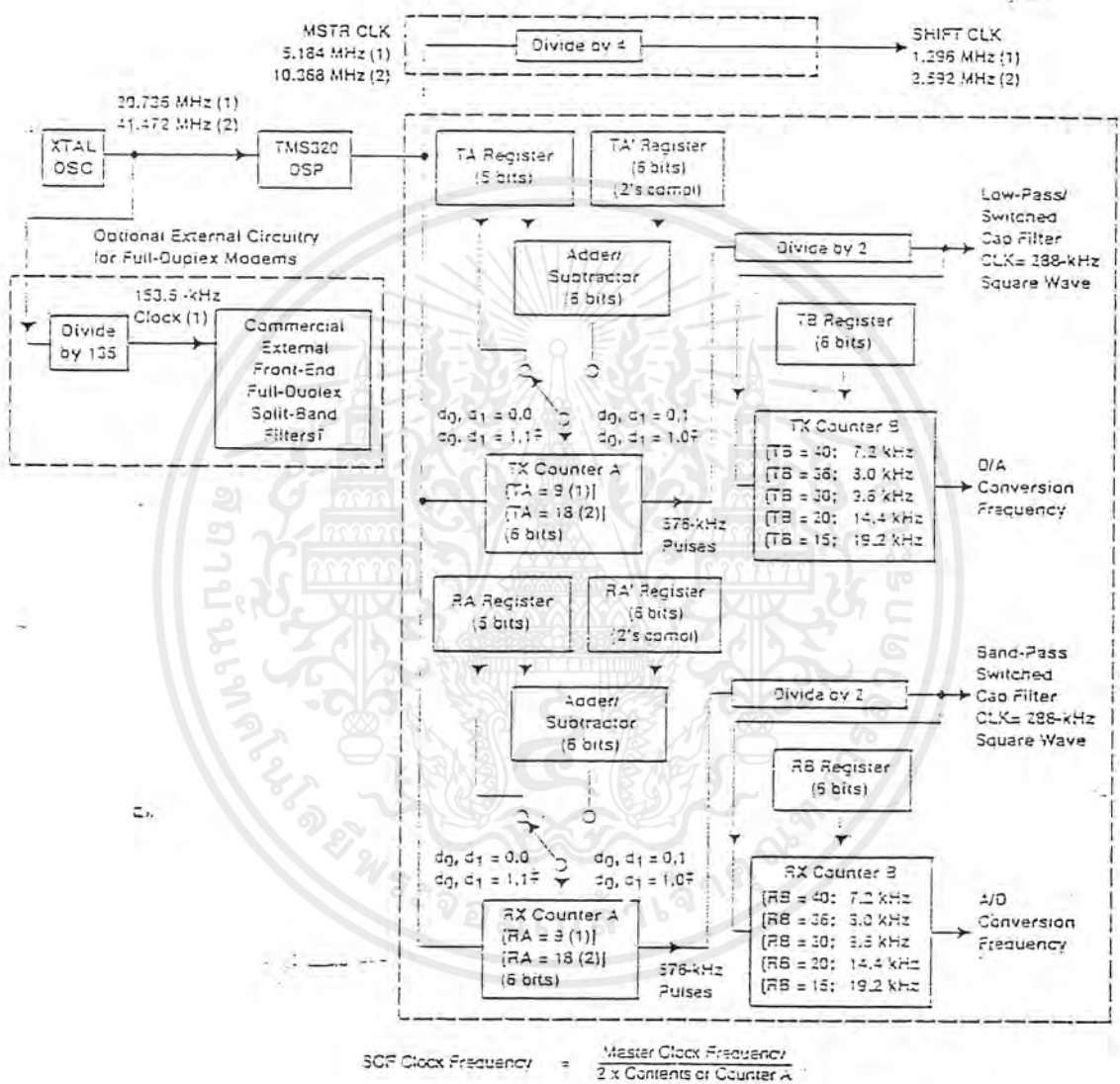
เช่นเดียวกับ A/D ฟิเตอร์โดยทรานเฟอร์เมอร์ฟังก์ชันของฟิเตอร์ถูกกำหนดจากอัตราส่วนกับความถี่ 228 kHz และอัตราการแปลง D/A ก็หาได้จากความถี่ 228 kHz หารด้วย TX เคนเตอร์ B

การต่อกลับ (Loopback)

การต่อกลับจะให้ผู้ที่ตรวจสอบวงจรโดย $OUT+$ และ $OUT-$ จะต่อเข้าภายในกับ $IN+$ และ $IN-$ ดังนั้นบิต DAC ($D15-D2$) จะถูกส่งไปยังขา DX และเปรียบเทียบกับบิต ADC ที่รับมาจากขา DR ซึ่งโดยปกติจะต้องมีค่าเท่ากับ (ในทางปฏิบัติอาจไม่เท่ากันก็ได้) ในการตรวจสอบ ถ้าใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา IN+ และ IN- สัญญาณภายนอกที่ต่อกับ IN+ และ IN- จะไม่มีผลแต่ถ้าใช้ AUX IN+ , AUX IN- สัญญาณภายนอกจะถูกรวมเข้ากับ OUT+ และ OUT- สำหรับการควบคุมการต่อกลับจำทำได้โดยการตั้งค่าในรีจิสเตอร์ควบคุม



รูปที่ 6.2 แสดงโครงสร้างของสัญญาณเวลาภายใน (Configuration Timing Internal)

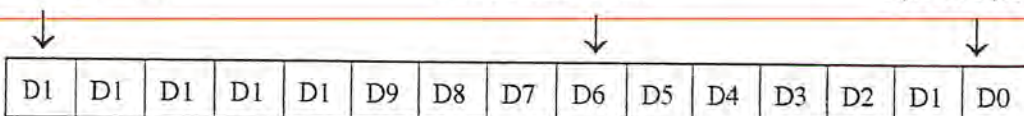
รูปแบบของบิตใน AIC DR หรือ DX Word

A/D or D/A MSB

1 st bit sent

1 st bit Sent of 2nd

A/D or D/A LSB



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5	3	2	1	0									
---	---	---	---	---	--	--	--	--	--	--	--	--	--

รูปที่ 6.3 แสดงรูปแบบบิตใน AIC DR หรือ DX Word

รูปแบบของบิตใน AIC DX (AIC DX Data Word Format Section)

D1	D1	D1	D1	D1	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
5	3	2	1	0										

ในรูปแบบการส่งครั้งแรก (Primary DX Serial Communication Protocol (D0-D15) นั้น เราจะใช้D1 และ D0 เป็นตัวเลือกส่วน D2 -D15 จะถูกส่งส่งไปยังรีจิสเตอร์ควบคุมการแปลง D/A (D/A converter Register) ซึ่งจะแบ่งการทำงานออกเป็นกรณีต่างๆได้ดังนี้คือ

ตารางที่ 6.1 แสดงการกำหนดค่า D1 และ D0 ในการทำงานในโหมดต่าง ๆ ของ AIC

D1	D0	โหมดการทำงาน
0	0	จะทำการโหลดค่าในรีจิสเตอร์ TA และRA ไปยัง TX และ RX เคาน์เตอร์ ตามลำดับและทำการโหลด TX และ RX เคาน์เตอร์ B ค่าในรีจิสเตอร์ TB และ RB
0	1	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่า TA+TA'และ RA+RA' แต่ จะทำการโหลด TX และ RX เคาน์เตอร์ B ด้วยค่าในรีจิสเตอร์ TB และ RB
1	0	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่า TA-TA'และRA-RTA+TA' แต่จะทำการ โหลด TX และ RX เคาน์เตอร์ B ด้วยค่าในรีจิสเตอร์ TB และ RB
1	1	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่า ในรีจิสเตอร์ TA และ RA ส่วน TX และ RX เคาน์เตอร์ B จะถูกโหลดด้วยค่าในรีจิสเตอร์ TB และ RB หลังจากนั้นก็จะรอเวลาประมาณ 4 สัญญาณสาฬิกาเลื่อนก็จะทำการส่งครั้งที่สอง (Secondary Communication) เพื่อไปโปรแกรม AIC ให้ทำงานตามต้องการ

รูปแบบของบิตในการส่งครั้งที่สอง

(Secondary DX Serial DX Serial Communication Protocol)

ตารางที่ 6.2 แสดง Secondary DXSerial Communication Protocol

XX 0 0	← toTA register→	X X	←to RA register→	D13 และ D6 เป็นMSR (เป็น บวก)
X 0 1	← to T A' register →	X	← to R A' register →	D14 และ D7 เป็นบิตแบบ 2' S
X 0	← to TB register→	X	← to RB register →	D14 และ D7 เป็น MSB (เป็น บวก)
X X X X X X X X	D7 D6 D5 D4 D3 D2 1 1	D2 = 0/1 ลด/เพิ่ม BandpassFilter D3 = 0/1 Disable / Enable ฟังก์ชัน D4 =0/1 Disable / Enable ขา AUX IN+ และ ขาAUX IN - D5 =0/1 บิตกำหนดการรับส่ง แบบ Asynchronous/ synchronous D6,D7 เป็นบิตควบคุม Gain		

ฟังก์ชันรีเซต (Reset)

ฟังก์ชันการรีเซตจะเป็นการตั้งค่าอัตราการเปลี่ยนแปลง A/D และ D/A ให้เป็น 8 kHz
ถ้า ใช้สัญญาณนาฬิกาหลัก 5.184 Mhz โดยค่าในรีจิสเตอร์จะถูกตั้งค่าเริ่มต้นเป็น TA = RA = 9
,TA' = RA' = 1 ,TB = RB = 24 และบิตในรีจิสเตอร์ควบคุมเป็น D7 = 1,D6=1,
D5=1,D4=0,D3=0,D2=1

ข้อบังคับในการกำหนดค่าในรีจิสเตอร์ของ AIC

1. รีจิสเตอร์ TA ในโหมดเวิร์คต้องมีค่า มากกว่าหรือเท่ากับ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รีจิสเตอร์ TA ในโหมดไบต์ ต้องมีค่า ≥ 5
3. รีจิสเตอร์ TA' สามารถเป็นได้ทั้ง ค่าบวก, ลบ หรือ ศูนย์
4. รีจิสเตอร์ RA ในโหมดเวิร์ด ต้องมีค่า ≥ 4
5. รีจิสเตอร์ RA ในโหมด ไบต์ ต้องมีค่า ≥ 5
6. รีจิสเตอร์สามารถมีค่าเป็นได้ทั้ง ค่าบวก, ลบ หรือ ศูนย์
7. $(TA \pm TA')$ จะต้องมีค่า > 1
8. $(RA \pm RA')$ จะต้องมีค่า > 1
9. TB จะต้องมีค่ามากกว่า 1

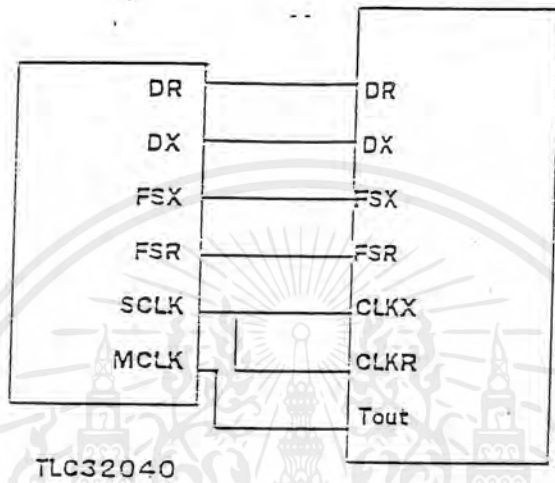
ถ้าไม่เป็นไปตามเงื่อนไขทั้ง 9 ข้อที่กล่าวมา AIC จะทำงานตามเงื่อนไขดังแสดงในตารางที่ 6.3 ตารางที่ 6.3

เงื่อนไข	การตอบสนองของ AIC
1. ถ้า $TA+TA' = 0$ หรือ 1 หรือ $TA-TA' = 0$ หรือ 1	จะทำการโหลดค่าในรีจิสเตอร์ TA ลงใน TX เคาเตอร์ A แทน
2. ถ้า $TA+TA' < 0$	MODULO 64 จะถูกใช้เพื่อให้แน่ใจว่า ค่าที่เป็นบวกถูกโหลดไปไว้ที่ TX เคาเตอร์ A แทน เช่น $TA+TA'+40h$ โหลดไปไว้ที่ TX ไว้ที่เคาเตอร์ A
3. ถ้า $RA+RA' = 0$ หรือ 1 หรือ $RA-RA' = 0$ หรือ 1	จะทำการโหลดค่าในรีจิสเตอร์ RA ลงใน RX เคาเตอร์ A
4. ถ้า $RA+RA' = 0$ หรือ 1	MODULO 64 ถูกใช้เหมือนข้อ 2 เพียงแต่โหลดไปที่ RX เคาเตอร์ A
5. ถ้า $TA = 0$ หรือ 1 หรือ $RA = 0$ หรือ 1	AIC หยุดทำงาน
6. ถ้า $TA < 4$ ในโหมดเวิร์ด $TA < 5$ ในโหมดไบต์ $RA < 4$ ในโหมดเวิร์ด $RA < 5$ ในโหมดไบต์	การติดต่อทางพอร์ตอนุกรมของ AIC ไม่ทำงาน
7. ถ้า $TB = 0$ หรือ 1	โหลดค่า TB ใหม่ ด้วยค่า 24H
8. ถ้า $RB = 0$ หรือ 1	โหลดค่า RB ใหม่ ด้วยค่า 24H
9. ถ้า AIC และ DSP ไม่สามารถติดต่อกันได้	จะเก็บค่า DAC Output ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การอินเตอร์เฟสระหว่าง TMS320C50 และ TLC32040

การใช้อินเตอร์รัพท์ของพอร์ตอนุกรม



รูปที่ 6.4 แสดงการเชื่อมต่อระหว่าง TMS320C50 และ TLC32040

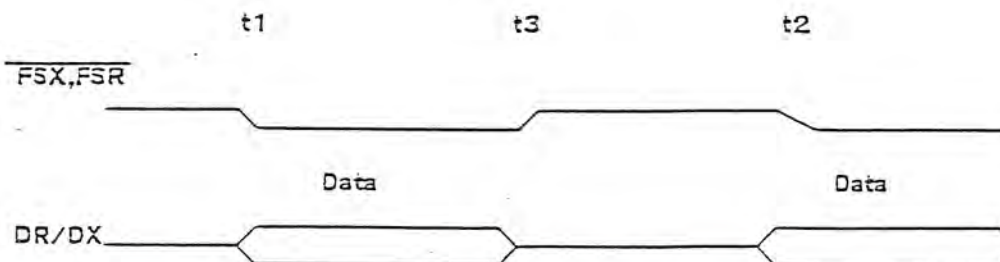
TMS320C50

การส่ง โดยการเขียน DXR จะเป็นการขออินเทอร์รัพท์ของ XINT (โดยไม่มีการมาร์ค)

การรับ เมื่อ RSR → DRR เต็มจะเกิดอินเทอร์รัพท์ของ RINT (โดยไม่มีการมาร์ค)

TLC32040

เนื่องจากการส่งระหว่าง TLC32040 กับ TMS320C50 เป็นแบบซิงโครนัส TLC32040 จะส่งและ รับทุก ๆ เฟรมซิงโครนัส (Frame Synchronous)



รูปที่ 6.5 แสดงช่วงเวลาในการรับส่งข้อมูลระหว่าง TLC32040 กับ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากความถี่ในการแซมปลิงสูงสุดเท่ากับ 19.2 kHz เพราะฉะนั้นข้อมูลต้องมาก่อน (ภายใน t_2-t_1) t_2-t_1 ต้องไม่น้อยกว่า 52.08 ไมโครวินาที ข้อมูลที่มาจาก TLC32040 แล้วทำการประมวลผลทางคณิตศาสตร์แล้วส่งกลับไปที่ TLC32040 อีกครั้ง ต้องใช้เวลาน้อยกว่า $1/F_s$ จึงจะสามารถได้สัญญาณที่มีความถี่เข้าเท่ากับความถี่ออก เนื่องจาก t_3-t_1 ช่องของข้อมูล (16 บิต) ใช้เวลาประมาณ $(1/\text{shift clock}) \times 16$ ประมาณเท่ากับ 6.17 ไมโครวินาที เพราะฉะนั้น 6.17 ไมโครวินาที เป็นเวลาที่ใช้ในการรับส่ง 1 ข้อมูล จะต้องใช้เวลาเหลือประมวลผล $(52.08 - 6.17)$ เท่ากับ 45.91 ไมโครวินาที หรือ ประมาณ $(45.91 \text{ ไมโครวินาที}) / (50 \text{ นาโนวินาที ต่อ 1 คำสั่ง})$ เท่ากับ 918 คำสั่ง

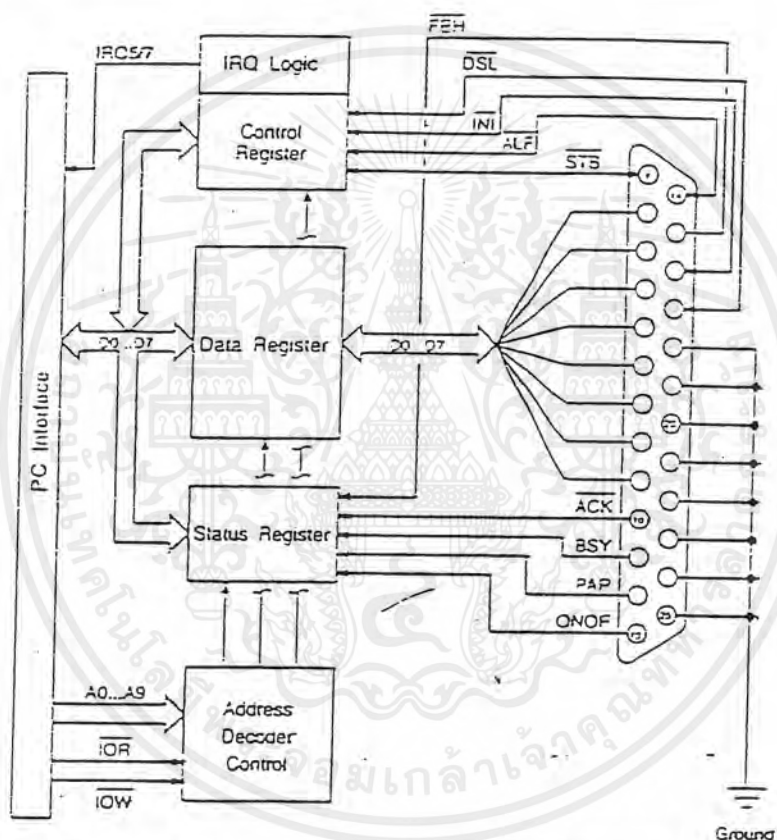
นั่นก็คือ สามารถเขียนคำสั่งโปรแกรมถึงกว่า 900 คำสั่ง เพื่อประมวลผลข้อมูล 1 ข้อมูล ก่อนที่จะเกิดการอินเทอร์รัพท์อีกครั้ง แต่เราใช้ไม่ถึงเพราะ F_s มักน้อยกว่า 19.2 kHz

บทที่ 7

การเชื่อมต่อแบบขนาน

7.1 โครงสร้าง หน้าที่ และการเชื่อมต่อขนาน

โดยปกติคอมพิวเตอร์ใช้การอินเทอร์เฟซแบบขนาน (Parallel) เพื่อส่งข้อมูล ออกไปยัง เครื่องพิมพ์ การอินเทอร์เฟซแบบขนาน มีโครงสร้างอย่างง่าย ๆ ดังต่อไปนี้



รูปที่ 7.1 แสดงแผนภาพการอินเทอร์เฟซแบบขนาน

การติดต่อกับระบบบัสของคอมพิวเตอร์นั้นเป็นการอินเทอร์เฟซของ PC ทั้งหมด ซึ่งการ์ดที่ใช้ในการใช้ในการเชื่อมต่อนั้นมี Databus ขนาด 8 บิตเท่านั้น แต่ก็สามารถใช้งานได้ ในคอมพิวเตอร์แบบ ISA หรือแบบ EISA ที่ขนาดของความกว้างเท่ากับ 8 บิต

การอินเทอร์เฟซเป็นการส่งข้อมูลที่มีขนาด 8 บิต ออกไปยังเครื่องพิมพ์และรีจิสเตอร์ภายในที่ใช้ในการอินเทอร์เฟซมีขนาด 8 บิต สัญญาณที่ใช้ในการอินเทอร์เฟซ คือ IOR และ IOW -โดยถ้าสัญญาณใดสัญญาณหนึ่งได้รับการติดต่อและมีการเลือกรีจิสเตอร์ ตัวถอดรหัสเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจะถอดรหัสตำแหน่งโดยใช้บิตที่ A0-A9 ซึ่งมีส่วนควบคุมการรับค่าเข้าและการส่งค่าออกของรีจิสเตอร์

การอินเทอร์เฟซแบบขนาน ประกอบด้วยส่วนต่างๆดังต่อไปนี้

1. รีจิสเตอร์ข้อมูล (Data Register)
2. รีจิสเตอร์สถานะ (Status Register)
3. รีจิสเตอร์ควบคุม (Control Register)
4. ส่วนควบคุมการถอดรหัสตำแหน่ง (Address Decoder Register)
5. ลอจิก IRQ (IRQ Logic)

รีจิสเตอร์ข้อมูล (Data Register) มีขนาด 8 บิต ซึ่ง CPU สามารถทำการอ่านหรือเขียนได้เรียกว่าเป็นการติดต่อแบบสองทิศทาง (Bi- directional) โดย CPU จะทำการส่งผ่านข้อมูลที่มีขนาดหลายไบต์จากรีจิสเตอร์ข้อมูลไปยังเครื่องพิมพ์ (Printer) ถ้า Microprocessor อ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลแล้วจะเกิดการ OR กันระหว่างค่าของไบต์ที่ถูกเขียนครั้งสุดท้ายกับค่าของสัญญาณที่เข้ามาอินเทอร์เฟซที่ขา 2-9 ซึ่งค่าที่ตอบกลับมาคือค่าที่ได้จากการ OR แล้ว โดยปกติเครื่องพิมพ์จะไม่ทำงานถ้าสายที่ขา 2-9 อยู่ในสถานะสูงจนกระทั่งได้รับคำสั่ง IN ก็จะได้ข้อมูลที่เขียนครั้งสุดท้ายกลับมา ในสถานะการณ์การเปลี่ยนแปลงที่กล่าวมานี้สายที่เชื่อมต่ออุปกรณ์ภายนอกจะอยู่ในสภาวะสูงการแลกเปลี่ยนข้อมูลที่เกิดขึ้นนั้นใช้การอินเทอร์เฟซแบบขนาน โดยถ้าเราสามารถเชื่อมต่อกับเครื่องพิมพ์ได้จะสามารถโหลดข้อมูลจากรีจิสเตอร์ข้อมูลได้

รีจิสเตอร์สถานะ (Status Register) จะแสดงค่าสถานะปัจจุบันขณะเชื่อมต่อกับเครื่องพิมพ์ ซึ่ง CPU สามารถอ่านค่าได้จากรีจิสเตอร์สถานะได้อย่างเดียวไม่สามารถที่จะเขียนลงในรีจิสเตอร์สถานะได้ เรียกว่าเป็นการต่อทิศทางเดียว (Uni-directional)

รีจิสเตอร์ควบคุม (Control Register) หน้าที่ของรีจิสเตอร์ควบคุม ควบคุมการเชื่อมต่อเครื่องพิมพ์, สัญญาณควบคุม และการอินเทอร์เฟซซึ่ง CPU สามารถที่จะอ่านค่าหรือเขียนค่าในรีจิสเตอร์ควบคุม ได้รีจิสเตอร์ควบคุม จะควบคุมการส่งข้อมูลไปยังเครื่องพิมพ์

ลอจิก IRQ (IRQ Logic) จะแสดงความพร้อมในการที่จะยอมรับในตัวอักษรตัวต่อไปเมื่อเกิดการอินเทอร์เฟซขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องพิมพ์ส่วนใหญ่จะต่อสัญญาณ DSL ลงกราวด์ ซึ่งเป็นการเลือกเครื่องพิมพ์ในระบบเก่าCPU สามารถที่จะใช้สัญญาณเพราะที่ จะเลือกหรือไม่เลือกเครื่องพิมพ์ เราสามารถส่งข้อมูลขนาด 8 บิต ที่ขา D0-D7 ไปยังเครื่องพิมพ์ได้ และสามารถที่จะเลือกข้อมูลขนาด 8 บิตที่ขา D0-D7 ได้ ด้วย สายสัญญาณ STR,ALE และ INI ของเครื่องพิมพ์ถูกควบคุมโดย CPU และการอินเทอร์เฟส การที่สัญญาณ STR เปลี่ยนสถานะจากสูงเป็นต่ำนั้น เครื่องพิมพ์จะรับข้อมูลจากขา D0-D7 ขณะที่มีการเปลี่ยนข้อมูล ส่วนของความหมายของสัญญาณอื่น ๆ สามารถดูได้จากที่กล่าวมาข้างต้น

การอินเทอร์เฟสแบบขนาน มีรีจิสเตอร์ 3 ตัวที่สามารถส่งข้อมูลควบคุมเครื่องพิมพ์ในการอินเทอร์เฟสได้ ตำแหน่งพื้นฐานของรีจิสเตอร์สำหรับการอินเทอร์เฟสทั้งหมดนั้น (LPT1-LPT4) ถูกเก็บไว้ในพื้นที่ของข้อมูลไบออส (BIOS)

รีจิสเตอร์ข้อมูลจะมีการออฟเซต (OFFSET) เท่ากับ 00H ส่วนรีจิสเตอร์สถานะมีค่าออฟเซตเท่ากับ 01H และรีจิสเตอร์ควบคุมมีค่าออฟเซตเท่ากับ 02H ค่าตำแหน่งพื้นฐานของ LPT1 เท่ากับ378H ส่วนค่าตำแหน่งพื้นฐานของ LPT 2 มีค่าเท่ากับ 278H ส่วนการอินเทอร์เฟสแบบขนานของการ์ด อแดปเตอร์ (Adapter) ของเซอร์คิวลัส (Hercules) หรือโมโนโครม (Monochrome) จะมีค่าตำแหน่งพื้นฐานเริ่มต้นที่ 3BCH

ในขณะที่เครื่องเริ่มทำการบูต (Booting) ไบออสทำการตรวจสอบตำแหน่งอินพุตและเอาต์พุตตามลำดับ ดังนี้ คือตรวจสอบที่ตำแหน่ง 3BCH ,378H, 278H, และ 2BCH แต่บางรุ่นของไบออสจะมีความแตกต่างออกไป คือ จะสนับสนุนการอินเทอร์เฟสแบบขนาน เพียงสองอย่างเท่านั้น ซึ่งคอนฟิกูเรชั่นจะทำการตรวจสอบโดยการอินเทอร์รัพท์ไบออส INT 11H ซึ่งจะคืนค่าติดตั้งการอินเทอร์เฟสเพียงสองเท่านั้น แม้จะมีการติดตั้งถึงสี่ตัว

ตารางที่ 7.1 รีจิสเตอร์ข้อมูล (Offset 00H)

บิตที่	7	6	5	4	3	2	1	0
ชื่อสัญญาณ	D7	D6	D5	D4	D3	D2	D1	D0
หมายเลขขาสัญญาณ	9	8	7	6	5	4	3	2

รีจิสเตอร์ข้อมูล มีความกว้างขนาด 8 บิต และมีการจัดส่งข้อมูลอย่างเหมาะสมเราสามารถทำการอ่านและเขียนข้อมูลรีจิสเตอร์นี้ อย่าง ๆ ง่าย โดยการใช้คำสั่ง อิน(IN) หรือ เอาท์ (OUT)

ตารางที่ 7.2 รีจิสเตอร์สถานะ (Offset 01H)

บิตที่	7	6	5	4	3	2	1	0
/FEH	/BSY	/ACK	PAP	OFON	/FEH	X	X	X
หมายเลขขาสัญญาณ	11	10	12	13	15	-	-	-

/BSY : Busy

1 = เครื่องพิมพ์ว่าง

0 = เครื่องพิมพ์ไม่ว่าง , ออฟไลน์

(OFF-LINE)

หรือเกิดข้อผิดพลาด

/ACK : Acknoeledge

1 = อยู่ในกระบวนการถ่ายโอนข้อมูล 0 = ข้อมูลส่งไปยังเครื่องพิมพ์เรียบร้อยแล้ว

PAP : Paper

$I =$ กระดาษหมด

0 = กระดาษมีอยู่

OFON : Off-line/ On-line

/FEH : เครื่องพิมพ์ดีดพลาสติก

1 = ไม่ผิดพลาด

0 = เครื่องพิมพ์ดีดพลาสติก

X : ไม่ใช่ (โดยทั่วไปเท่ากับ 1)

รีจิสเตอร์สถานะ สามารถอ่านได้อย่างเดียวและจะบอกค่าข่าวสารการควบคุมจากเครื่องพิมพ์ เมื่อทำการเคลียร์บิต /BSY เครื่องพิมพ์จะอยู่ในสถานะ /BSY ซึ่งแสดง Buffer ของเครื่องพิมพ์เต็ม ตัวอักษรกำลังถูกทำการเคลื่อนย้าย หรือเครื่องพิมพ์มีการเริ่มต้น การอินเทอร์เฟส จะไม่มีการส่งตัวอักษรใด ๆ จนกว่าบิต /BSY จะถูกเซต

บิต /ACK และ บิต STR ของรีจิสเตอร์ควบคุม มีไว้สำหรับการทำให้การส่งข้อมูลถูกต้อง
ต้องถ้าการอินเทอร์เฟสมีการส่งข้อมูลมายังเครื่องพิมพ์แล้ว ขาสัญญาณ /ACK ในรีจิสเตอร์
สถานะเป็นตัวบอกว่าได้รับข้อมูลเรียบร้อยแล้ว

ถ้าเครื่องพิมพ์ไม่มีกระดาษ บิต PAP ถูกเซตเป็น 1 และในเวลาเดียวกันบิต OFON ถูกเคลียร์ และบิต /FEH ถูกเซตเพื่อป้องกันเกิดผิดพลาดขึ้น

ตารางที่ 7.3 รีจิสเตอร์ควบคุม (Offset 02H)

บิตที่	7	6	5	4	3	2	1	0
ชื่อสัญญาณ	X	X	X	IRQ	DSL	/INI	ALE	STR
หมายเลขขาสัญญาณ	-	-	-	-	17	16	14	1

X : ไม่ใช่ (โดยทั่วไปเท่ากับ 1)

IRQ : Hardware Interrupt Request

1 = Enabled

0 = Disabled

DSL : การเลือกเครื่องพิมพ์

1 = เครื่องพิมพ์ถูกเลือก

0 = เครื่องพิมพ์ไม่ถูกเลือก

/INI : การให้ค่าเริ่มต้น (Initial)

1 = เครื่องพิมพ์ทำงานธรรมดา

0 = ทำงานกับค่าเริ่มต้น

ALE : ป้อนกระดาษอัตโนมัติ

1 = ทำการป้อนกระดาษอัตโนมัติโดย

0 = ทำการป้อนกระดาษโดยผู้
ให้เครื่องพิมพ์

STR : สโตรป (Srobe)

1 = ส่งข้อมูลไปยังเครื่องพิมพ์

0 = ไม่มีการส่งข้อมูล

รีจิสเตอร์ควบคุม จะทำการควบคุมเครื่องพิมพ์และควบคุมการเกิดฮาร์ดแวร์อินเทอร์รัพท์ โดยการอินเทอร์เฟสแบบขนานถ้าบิต IRQ ถูกเซตก็จะเกิดการอินเทอร์เฟสทางฮาร์ดแวร์เทอร์รัพท์ โดย IRQS (LP2) หรือ IRQ7 (LPT1) ติดต่อไปยัง INT 01H และ INT 0FH ตามลำดับ ถ้าสัญญาณ /ACK เกิดการเปลี่ยนแปลงระดับสูงเป็นระดับต่ำ ดังนั้นจะเกิดการอินเทอร์รัพท์ขึ้นจริงเมื่อเครื่องพิมพ์ได้รับตัวอักษร การจัดการควบคุมคู่กันไปเช่นนี้สามารถที่จะทำการส่งตัวอักษรไปยังเครื่องพิมพ์ที่รอรับข้อมูลอยู่วันแคว่บอของพีซี โดยทั่วไปไม่ได้ใช้อินเทอร์รัพท์นี้ไปจับข้อมูลให้ส่งออกไป แต่โดยทั่วไปจะใช้บิต /ACK ในการตรวจสอบการรับรู้โดยเครื่องพิมพ์

บิต DSL มีไว้เพื่อสำหรับเลือกจะใช้เครื่องพิมพ์หรือไม่ ส่วนบิต /INI ใช้สำหรับอินนิเชียลเครื่องพิมพ์ และถ้าทำการเซตบิต ALE เครื่องพิมพ์จะทำการป้อนกระดาษอัตโนมัติหลังจากมีการพิมพ์แต่ละบรรทัด

บทที่ 8

การออกแบบตัวกรองเชิงเลขแบบผลตอบสนองอิมพัลส์ไม่จำกัดและจำกัด

จะเน้นหนักในส่วนของเทคนิคที่ถูกใช้กว้างขวางในการออกแบบตัวกรองผลตอบสนองอิมพัลส์ไม่จำกัดใน ระบบเวลาขึ้นยง เชิงเส้น

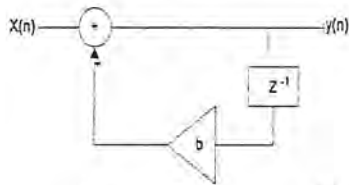
วิธีการออกแบบจะมีพื้นฐานมาจากการแปลงเชิงเส้นคู่ของการกรองอนุমানแบบดั้งเดิม (classic analog) การแปลงเชิงเส้นคู่ (Bilinear Transformation) จะเกิดขึ้นที่บริเวณด้านครึ่งซ้ายของระนาบ S ภายในวงกลมรัศมี หนึ่งหน่วยในระนาบ Z กระบวนการในการแปรเปลี่ยนตัวกรองแบบดั้งเดิมไปสู่ตัวกรองเชิงตัวเลขมีวิธีการดังนี้

1. ออกแบบสูตรในการสร้าง โพลและซีโรว์ของตัวกรองบัตเทอร์เวิร์ธ เชบปีเชฟ และ อิลิปติกความถี่ต่ำ
2. หาสูตรในการแปลงช่วงความถี่สำหรับการเปลี่ยนตัวกรองอนาล็อกความถี่ต่ำไปสู่อนาล็อกความถี่สูง แบนด์พาสและแบนด์สต็อป
3. การแปลงเชิงเส้นคู่ซึ่งเชื่อมโพลในระนาบ S ไปในระนาบ Z

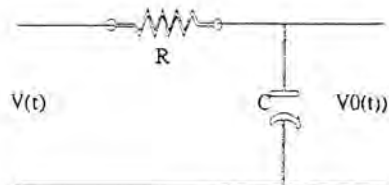
ในการออกแบบ IIR ของตัวกรองเชิงเลขขึ้นอยู่กับรูปแบบของตัวมันเองที่จะสร้างเป็นการเขียนโปรแกรมผ่านคอมพิวเตอร์ สมการรูปแบบปิดจะถูกออกแบบแบบเพื่อออกแบบตัวกรองเชิงเลขแบบช่วงความถี่ต่ำ ช่วงความถี่สูง แบนด์พาส และ แบนด์สต็อป โดยที่จะแสดงให้เห็นภาพโดยการใช้โคอะแกรมการไหลทางตรรกะ เพื่อจะหา โพลและซีโรว์ และสัมประสิทธิ์ของตัวกรองเชิงเลข อิลิปติก จะทำให้เห็นถึงบางอย่างของข้อได้เปรียบและเสียเปรียบของดิจิตอลฟิวเตอร์และอนาล็อกฟิวเตอร์ ตัวกรองเชิงเลขจะให้รูปแบบเดียวกันกับฟังก์ชันตัวกรอง ยกเว้นที่จะทำงานกับสัญญาณดิจิตอล ตัวอย่างพร้อมกับ ลักษณะตัวกรองดิจิตอลตัวอย่าง ตัวกรองอนาล็อกจะขึ้นกับทฤษฎีที่เป็นพื้นฐานจากสมการผลต่างเชิงเส้นและตัวกรองดิจิตอลจะขึ้นกับทฤษฎีที่เป็นพื้นฐานจากสมการผลต่างเชิงเส้น และตัวกรองดิจิตอลจะทำงานกับสัญญาณอนาล็อกและประกอบด้วย ตัวต้านทาน, ตัวเก็บประจุ และ ตัวเหนี่ยวนำ , ตัวกรองดิจิตอลจะทำงานในลักษณะเป็น การสุ่มเป็นช่วงๆ พร้อมด้วยตรรกะเชิงเลข การออกแบบตัวกรองดิจิตอลจะมุ่งที่จะหาสัมประสิทธิ์สมการผลต่างซึ่งจะทำให้เราได้ความถี่เฉพาะ และผลตอบสนองต่อเวลาเพราะว่าความต้องการนี้คล้ายกับการประมาณการของปัญหาสำหรับตัวกรองอนาล็อกสู่ดิจิตอล เทคนิคการเชื่อมโยงนี้ถูกพัฒนาให้เป็นที่ยอมรับได้ ในการออกแบบในรายละเอียด วิธีนี้เป็นวิธีที่นิยมใช้ในการออกแบบที่เป็นมาตรฐานของ ช่วงความถี่ต่ำ ช่วงความถี่สูง แบนด์พาส และแบนด์สต็อป การประมาณการอนุमानใช้การออกแบบตัวกรองดิจิตอลโดยใช้ตัวกรองแบบเชบปีเชฟ (ชนิดที่1,และ2) และ ตัวกรองอิลิปติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของดิจิทัล และตัวกรองอนาล็อกที่สัมพันธ์กัน คล้องกันในรูปแบบ ตัวกรองความถี่ต่ำแบบโพลเดี่ยวในรูปที่ 8.1 จะเห็นได้ว่า



ตัวกรองเชิงเลขความถี่ต่ำ



ตัวกรองอนาล็อกความถี่ต่ำ

- | | |
|-------------------------------------|--|
| 1. ช่วงเวลาไม่ต่อเนื่อง | 1. ช่วงเวลาต่อเนื่อง |
| 2. สมการผลต่าง | 2. สมการผลต่าง |
| $y(n) = by(n) + x(n)$ | $v(t) = RC(dv/dt) + v_0(t)$ |
| 3. วิเคราะห์จากระนาบ Z จากการแปลง Z | 3. วิเคราะห์จากการใช้ระนาบ S การแปลงลาปลาซ |

รูปที่ 8.1 แสดงการเปรียบเทียบระหว่างตัวกรองเชิงเลข ความถี่ต่ำและตัวกรองอนาล็อกความถี่ต่ำ

ตัวกรองขึ้นกับอุปกรณ์อย่าง adder , multiplier และ ความหน่วงเวลาในขณะที่ของตัวกรองอนาล็อกประกอบด้วย ความต้านทาน และ ตัวเก็บประจุ

8.1 การเปรียบเทียบตัวกรองดิจิทัล

ข้อได้เปรียบของตัวกรองดิจิทัล

1. เสถียรภาพด้านอุณหภูมิ ถ้าอุณหภูมิเปลี่ยนจะมีผลต่อ ตัวต้านทาน ตัวเก็บประจุ และ ตัวเหนี่ยวนำซึ่งเป็นข้อจำกัด
2. ความเที่ยงตรงรูปแบบตัวกรองจะดีขึ้นอยู่กับช่วงไดนามิค เสถียรภาพและความต้านทานของผลตอบสนองความถี่โดยการเพิ่มความยาว bit register processor
3. ความสามารถในการปรับเปลี่ยนผลตอบสนองความถี่ ตัวกรองเชิงเลขสามารถเปลี่ยนได้โดยการรับข้อมูลสัมประสิทธิ์ตัวกรองจากหน่วยความจำไปสู่อุปกรณ์ที่รับค่าสัมประสิทธิ์นั้น
4. ทำงานได้ในหลายรูปแบบ ถ้าอัตราบิตของอินพุตที่มีความสำคัญต่ำกว่าความสามารถของระบบดิจิทัลนั้นตัวกรองเชิงเลขสามารถที่จะถูกใช้ในการทำงานได้หลายอย่างเพื่อที่จะใช้ระบบนั้นได้อย่างมีประสิทธิภาพมากขึ้น

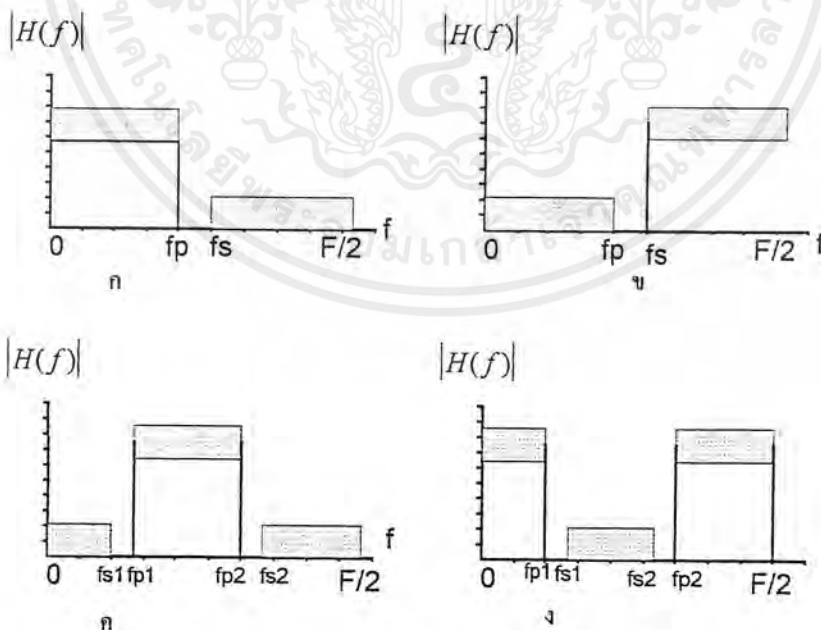
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของตัวกรองดิจิทัล

1. มีการจำกัดแบนด์วิธซึ่งเป็นผลจากขบวนการสุ่มของตัวแปลงจากอนาลอกเป็นดิจิทัล
2. ผลกระทบจากความยาวจำกัดของรีจิสเตอร์ซึ่งเป็นผลจากการใช้ข้อมูลและรีจิสเตอร์ที่เก็บค่าสัมประสิทธิ์ด้วยจำนวนบิตที่จำกัด

8.2 รูปแบบการออกแบบตัวกรองเชิงเลข

ตัวกรองเชิงเลขซึ่งถูกออกแบบให้ผ่านค่าสเปกตรัมที่เป็นส่วนประกอบของสัญญาณอินพุตที่เป็นช่วงเฉพาะความถี่พื้นฐานในการถ่ายโอนช่วงสเปกตรัม ให้สามารถผ่านเฉพาะช่วงที่เราต้องการได้ตัวกรองที่เป็นพื้นฐานที่สำคัญในการทำงานได้มี 4 ชนิด คือ ความถี่ต่ำ ความถี่สูง แบนด์พาส แบนด์สตอป ลักษณะผลความถี่ที่ต้องการแสดงในเส้นทึบในรูปที่ 8.2 ความถี่ $f \geq 0$ แถบพื้นที่ในรูปแสดงขอบที่ด้านทานของความถี่ที่ถูกประมาณการไว้ f_p แสดงช่วงผ่าน (bandpass) และ f_s แสดงช่วงหยุด (bandstop) ช่วงความถี่ $f_p \leq f \leq f_s$ เรียกว่าเป็น transition band ฟังก์ชันการประมาณการของตัวกรองจะมุ่งที่จะหาผลตอบสนองในอุดมคติภายในช่วงผ่านที่เจาะจงไว้ตามต้องการ



รูปที่ 8.2 แสดงรูปแบบลักษณะการออกแบบตัวกรองแบบต่างดังนี้ (ก) ตัวกรองความถี่ต่ำ

(ข) ตัวกรองความถี่สูง (ค) ตัวกรองแบนด์พาส (ง) ตัวกรองแบนด์สตอป

และที่ขอบเขตของช่วงหยุดที่ต้องการปัญหาของการออกแบบตัวกรองเชิงเลขไม่แตกต่างจากตัวกรองอนาล็อกนักเพราะขึ้นอยู่กับฟังก์ชันถ่ายโอน (Transfer function) ซึ่งต้องมีเสถียรภาพ ต้องเป็นโครงข่ายดิจิทัลคอลลอสที่สำคัญคือโพลของอนาล็อกต้องอยู่ด้านซ้ายของระนาบ S และที่สำคัญกำลังของตัวตั้งในฟังก์ชันถ่ายโอนต้องน้อยกว่าตัวหาร ตัวกรองเชิงเลขเราจะมุ่งสนใจลงไปใน การหาค่าสัมประสิทธิ์สมการผลต่างที่ทำให้เราได้ลักษณะตัวกรองที่ต้องการเช่น ผลตอบสนองเวลา หรือผลตอบสนองในโดเมนของความถี่ซึ่งเราได้ศึกษาการออกแบบตัวกรองต่อไป

8.3 ตัวกรองความถี่ต่ำ (Analog Lowpass Filter)

ปัญหาการประมาณการในการออกแบบ ของ ตัวกรองเชิงเลขแบบ IIR ปกติจะทำการแก้ปัญหามาโดยการ ใช้ การประมาณการตัวกรอง อนาล็อกตัวไปนี้

1. บัทเทอร์เวิร์ธ (Butterworth)
2. เชบชีเชฟ (Chebyshev)
3. เอลลิปติก (Elliptic)

คุณสมบัติของการประมาณเหล่านี้จัดแบบไว้โดยการจัดโครงสร้างตามนี้คือ

1. การกำหนดขนาดของผลตอบสนอง แอมปริจูดของ การประมาณการอนาล็อกความถี่ต่ำ
2. กำหนดสมการอันดับของตัวกรอง
3. กำหนดโพลของอนาล็อกความถี่ต่ำ และสมการที่จุดกำเนิดซีโร

ในการหาค่าและคุณสมบัติของตัวแปลงแบบต่ำในสามารถศึกษาได้จากหนังสือทั่วไปและในโครงการที่ศึกษานี้จะใช้การออกแบบโดยวิธีของ ตัวกรองแบบบัทเทอร์เวิร์ธเป็นหลัก โดยจะได้อธิบายและทำความเข้าใจต่อไปดังต่อไปนี้

8.4 ตัวกรองความถี่ต่ำแบบ บัทเทอร์เวิร์ธ (Analog Lowpass Butterworth Filters)

ตัวกรองแบบบัทเทอร์เวิร์ธสามารถถูกออกแบบโดยการกำหนดคุณสมบัติได้โดยผลตอบสนองขนาด โดยประมาณการแบบ แพลท ในส่วนของช่วงผ่าน (PassBand) พื้นฐานการออกแบบสูตรของ ตัวกรองความถี่ต่ำนี้จะได้แสดงดังต่อไปนี้

ฟังก์ชันขนาดกำลังสอง สำหรับลำดับที่ N ของ ตัวกรองอนาล็อกบัทเทอร์เวิร์ธ แสดงได้ดังนี้

$$|H(j\omega)|^2 = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p} \right)^{2N}} \quad (8.1)$$

ถูกแสดงได้ในรูปที่ 8.3 ดังนั้นฟังก์ชันมีลักษณะลดลงโดยที่ผลตอบสนองสูงสุดมีค่าที่ $\omega = 0$

จะสังเกตได้ว่าถ้า ขนาดของลำดับ N เพิ่มขึ้น ผลตอบสนองขนาดจะเข้าใกล้ ลักษณะตัวกรองในอุดมคติ เพราะว่า อนุพันธ์อันดับแรก $(2N-1)$ ของการกระจายอนุกรม แมกครอริน (Maclaurin Series) ของฟังก์ชันขนาด $|H(j\omega)|$ มีค่าที่ $\omega = 0$ หรือ Zero จากรูปที่ 8.3

โดยการออกแบบค่า Parameters ดังนี้

Passband Tolerance ที่ $|\omega| \leq \omega_p$,

$$|H(j\omega)|^2 \geq \frac{1}{1 + \varepsilon^2} \quad (8.2)$$

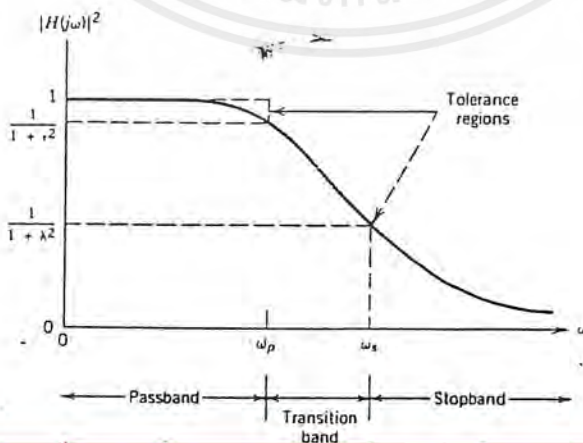
StopBand Tolerance ที่ $|\omega| \geq \omega_s$

$$|H(j\omega)|^2 \leq \frac{1}{1 + \lambda^2} \quad (8.3)$$

ในรูปที่ 8.3 นั้นจะได้ค่าของผลตอบสนองความถี่ต่ำโดยการให้ ค่า ε เข้าสู่ ศูนย์ ω_s เข้าใกล้ ω_p และเข้าใกล้ค่าอนันต์

ตอนนี้เราสามารถกำหนดการทำงานเพื่อมุ่งหาค่า สมการลำดับ ที่จะทำให้ได้ค่าที่เหมาะสม

โดยการ ให้ $\omega = \omega_s$ สมการที่ 8.2 จะกลายเป็นสมการที่ 8.3 ที่



รูปที่ 8.3 แสดงผลตอบสนองขนาดกำลังสองของตัวกรองความถี่ต่ำแบบ บัทเทอร์เวิร์ธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ค่าออกมาจากแก้สมการจะได้ค่าดังนี้

$$N \geq \frac{\log\left(\frac{\lambda}{\varepsilon}\right)}{\log\left(\frac{\omega_s}{\omega_p}\right)} \quad (8.4)$$

ดังนั้นในสมการที่ 8.2 และ 8.3

$$\varepsilon = \left(10^{0.1A_p} - 1\right)^{0.5} \quad (8.5a)$$

$$\lambda = \left(10^{0.1A_s} - 1\right)^{0.5} \quad (8.5b)$$

โดยที่ A_p เป็นค่าสูงสุดของการลดทอนในช่วงผ่านใน ค่าบวก dB และ A_s เป็นค่าต่ำสุดของการลดทอนในช่วงหยุด ในค่าบวกของ dB เพื่อให้เกิดความง่ายของการสมมูลค่าเราจะกำหนดให้ Parameters A และ K_0 ได้ดังต่อไปนี้

$$A = \frac{\lambda}{\varepsilon} = \left(\frac{10^{0.1A_s} - 1}{10^{0.1A_p} - 1}\right)^{0.5} \quad (8.6)$$

และ $K_0 = \omega_p/\omega_s$

ในที่สุดสมการอันดับ สำหรับตัวกรองอนาล็อกความถี่ต่ำได้ผลลัพธ์เป็นดังนี้

$$N > \frac{\log A}{\log\left(\frac{1}{K_0}\right)} \quad (8.7)$$

สมการอันดับ (The order equation) สำหรับ ตัวกรองเชิงเลขแบบ Highpass , BandPass และ Bandstop ให้ไว้ในตารางที่ 4.1

สามารถหา โพล นอร์มอลไลซ์ จาก s -plane หาได้จากการกำหนดค่าในตัวหารของสมการที่ 8.1 เท่ากับศูนย์ ผลลัพธ์นอร์มอลไลซ์ ได้จากการให้ $\omega_p = 1$ และ $\varepsilon = 1$ ดังนั้นผลลัพธ์ที่ได้คือ

$$1 + \omega^{2N} = 0$$

แทนค่าโดยให้ $s = j\omega$ จะได้ค่าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(-s^2)^N + 1 = 0$$

นิพจน์ -1 ในการสมมติค่าในรูปแบบเชิงขั้ว สำหรับ Multiple roots

$$(-1)^N s^{2N} = e^{j(2k-1)\pi} = -1 \quad k = 1, 2, 3, \dots, N$$

ในตอนนี้ รากอันดับที่ k สำหรับ โพล ในด้านครึ่งซ้ายของ ระนาบ s สามารถกำหนดเป็นค่าได้ดังนี้

$$s_k = \sigma_k + j\omega_k = e^{j(2k+N-1)\pi/2N} = je^{j(2k-1)\pi/2N}$$

ในที่สุด จะได้นอร์มอไลซ์อานาลอกโพลสำหรับ ตัวกรองความถี่ต่ำแบบบัทเทอร์เวิร์ธ ดังนี้

$$s_k = -\sin\left(\frac{2k-1}{2N}\right)\pi + j\cos\left(\frac{2k-1}{2N}\right)\pi \quad (8.8)$$

$$k = \begin{cases} 1, 2, \dots, (N+1)/2 & \text{for } N - \text{odd} \\ 1, 2, \dots, N/2 & \text{for } N - \text{even} \end{cases}$$

ในกรณีทั่วไป ที่ไม่โพลแบบนอร์มอไลซ์จะได้ค่าดังนี้

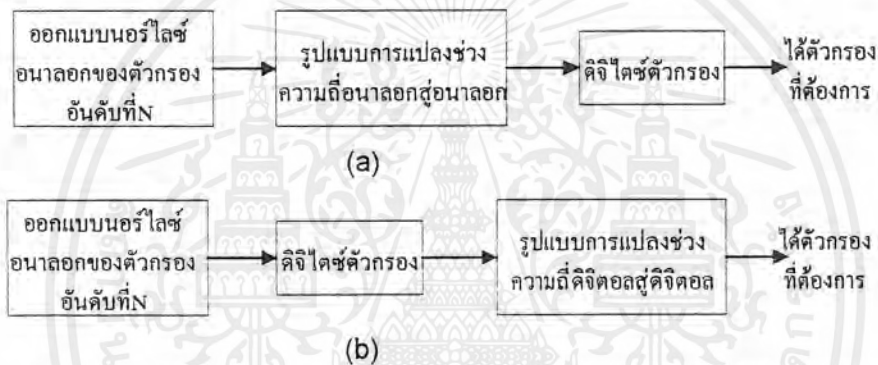
$$s'_k = s_k \omega_p^{-1/N} \quad (8.9)$$

ในสมการที่ 8.9 โพลถูกกระจายในส่วนต่างๆในโดยค่า π/N เรเดียน แบ่งเป็นส่วนๆใน s -plane บนวงกลมที่มีรัศมี $\omega_p e^{-1/N}$ นอร์มอไลซ์อานาลอกโพลหาได้จากสมการที่ 8.8 สำหรับกรณีที่ $\omega_p = 1$, $A_p \approx -3.01$ dB และโพลวางอยู่บนวงกลมหนึ่งหน่วย ที่แบ่งเป็นส่วนๆจากค่าของ π/N

8.5 ช่วงความถี่ของการแปลง (FREQUENCY BAND TRANSFORMATIONS)

ในส่วนนี้จะได้ศึกษาเทคนิควิธีการออกแบบของนอร์มอไลซ์อานาลอก (ความถี่ช่วงผ่าน 1 Hz) ตัวกรองความถี่ต่ำ การแปลงของนอร์มอไลซ์อานาลอกความถี่ต่ำไปเป็น ความถี่ต่ำที่ไม่เป็นนอร์มอไลซ์ หรือ ประเภท ความถี่สูง , ความถี่ช่วงผ่าน และ ความถี่ช่วงหยุด มีสองเทคนิคที่นิยมใช้ในการแปลงเพื่อจัดรูปแบบให้ได้ตัวกรองเชิงเลขตามที่เราต้องการ เทคนิคแรกแสดงให้เห็นในรูปที่ 8.4a เป็นการเปลี่ยนจาก นอร์มอไลซ์ ความถี่ต่ำเป็นตัวกรองอานาลอกที่ไม่เป็น นอร์มอไลซ์

โดยการดิเจ็ทซ์ค่าที่ต้องการเพื่อให้ได้ชนิดของตัวกรองเชิงเลขที่เหมาะสม เทคนิคที่สองแสดงในรูป 8.4b ในตอนแรกจะโยง (map) ตัวกรองอนาลอกความถี่ต่ำแบบนอร์มอลจาก ระบาย s ไปสู่ระบาย z ต่อจากนั้นใช้การแปลงช่วงความถี่ดิจิตอล ซึ่งแสดงให้เห็นนั้นจะให้การดิเจ็ทซ์ตัวกรองอนาลอกด้วยวิธีการแปลงเชิงเส้นคู่ (Bilinear transformation) ทำให้ได้ผลลัพธ์เหมือนกับการใช้การแปลงทางความถี่ (frequency transformation) เราจะใช้วิธีแรกในการออกแบบตัวกรองเชิงเลข สมการ การแปลงทางความถี่จะถูกใช้เพื่อพัฒนารูปแบบการออกแบบของ ตัวกรองเชิงเลขแบบ IIR ในตารางที่ 8.1



รูปที่ 8.4 แสดงการแปลงช่วงความถี่ (a) อนาลอกเป็นอนาลอก (b) ดิจิตอลไปสู่ดิจิตอล

ตารางที่ 8.1 การแปลงช่วงความถี่อนาลอก

ชนิดตัวกรอง	รูปแบบการแปลงจากตัวกรองความถี่ต่ำไปตัวกรองอื่นๆ
ความถี่ต่ำ→ความถี่ต่ำ	$s \rightarrow \frac{s}{\omega_u}$
ความถี่ต่ำ→ความถี่สูง	$s \rightarrow \frac{\omega_u}{s}$
ความถี่ต่ำ→ความถี่แบนด์พาส	$s \rightarrow \frac{s^2 + \omega_l \omega_u}{s(\omega_u - \omega_l)}$
ความถี่ต่ำ→ความถี่แบนด์สต็อป	$s \rightarrow \frac{s(\omega_u - \omega_l)}{s^2 + \omega_l \omega_u}$

ω_l = ความถี่ช่วงผ่านด้านช่วงล่าง

ω_u = ความถี่ช่วงผ่านด้านช่วงบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.6 การแปลงแบบเชิงเส้นคู่ (Bilinear Transformation)

การแปลงแบบเชิงเส้นคู่คือการเชื่อมโยงหรือการแปลงจากอนาล็อก S-Plane ไปสู่ดิจิทัล Z-Plane แปลงการเปลี่ยนโดยการเชื่อมโยงโพลและศูนย์อนาล็อกไปสู่โพลและศูนย์ดิจิทัล โดยที่จุดแต่ละจุดใน S-Plane ถูกเชื่อมโยงไปยังจุดเดียวของแต่ละตัวใน Z-plane การแปลงแบบเชิงเส้นคู่จะรักษาผลตอบสนองขนาดของตัวกรองอนาล็อก Piecewise Constant พร้อมกับค่าการลดทอนเฟส ถ้าทั้งเฟสเชิงเส้นมีค่าเท่ากับผลตอบสนองขนาด ผู้ออกแบบสามารถที่จะเก็บ เป็นตัวกรองเชิงเลขแบบ FIR (Finite Impulse) จะเห็นได้ว่าเฟสเชิงเส้นอาจจะใช้ไม่ได้ในรูปแบบทั้งหมดเช่นกระบวนการตรวจจับ (Detection Processing) ไม่ต้องการ เฟสเชิงเส้น ในขณะที่ Correlation processing ที่มีการเพิ่มส่วนของเวลาจะต้องการเฟสเชิงเส้น

การแปลงเชิงเส้นคู่เป็นการเชื่อมโยงที่มีความง่ายที่สุดของการขยาย S-Plane ไปสู่ Z-plane พื้นฐานก็คือเทคนิคจำนวนเต็มเชิงเลข ใช้เพื่อจำลองจำนวนเต็มของจำนวนอนาล็อกที่กำหนดโดย

$$s = \frac{z-1}{z+1} \quad (8.10)$$

สำหรับ $S=j\omega$ ในสมการ 8.11 จะเป็นดังนี้

$$j\omega' = \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} = \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{e^{j\omega T/2} + e^{-j\omega T/2}} \quad (8.11)$$

โดยที่ ω' คือตัวแปรอนาล็อกความถี่ และ ω คือตัวแปรความถี่ดิจิทัล โดยการให้ $\omega = \omega_i$ และความถี่ของการสุ่ม คือ $F = 1/T$ สมการ 4.46 ถูกใช้ในการหาความสัมพันธ์ tangent ของตัวแปรความถี่อนาล็อก และตัวแปรความถี่ดิจิทัลได้รับโดยสมการ

$$\omega' i = \tan(\omega_i T/2) = \tan(\pi f_i/F) \quad (8.12)$$

ความถี่ f_i คือ ค่าวิกฤตช่วงผ่านดิจิทัล และความถี่ช่วงหยุดที่มีความสัมพันธ์ในการกำหนดการออกแบบตัวกรอง จากสมการ 8.12 ความถี่ $\omega' i$ บนแกน ω -axis ที่เชื่อมโยงไปสู่ความถี่ f_i บนวงกลมรัศมีหนึ่งหน่วย ดังนั้น ตัวกรองอนาล็อกมี ฟังก์ชันการส่งผ่านที่มีความสัมพันธ์ล้อยตาม ฟังก์ชันการส่งผ่านตัวกรองดิจิทัล โดยความสัมพันธ์นั้นคือ

$$H(j\omega') = H(e^{j2\pi f_i T})$$

สังเกตได้ว่าความสัมพันธ์ในความถี่จะไม่เป็นเชิงเส้น ความไม่เป็นเชิงเส้นนี้เป็นผลกระทบต่อการแปลงแบบเชิงเส้น และเรียกว่าเป็นความถี่ warping การชดเชยความถี่ warping สามารถทำได้โดยการทำ prewarping ที่ค่าวิกฤตความถี่ดิจิทัลตาม

สมการที่ 8.12 โดยที่ความถี่วิกฤตนี้จะถูกเลื่อนกลับสู่ค่าที่ต้องการหลังจากการใช้การแปลงแบบเชิงเส้น

ตารางที่ 8.2 แสดงการเชื่อมโยงการแปลงเชิงเส้น

S-plane	Z-plane
$0+j0$	$1+j0$
$\infty+j0$	$-1+j0$
$0+j1$	$0+j1$
$-1+j0$	$0+j0$
$0+j\infty$	$-1+j0$
point on real axis	point on real axis
points on $j\omega$ -axis from $0 \rightarrow \infty$	points on unit circle from $0 \rightarrow \pi$
points on $j\omega$ -axis from $0 \rightarrow -\infty$	points on unit circle from $0 \rightarrow -\pi$
point on any line	point on circle passing through
	$-1+j0$
$a < 0$ (stability condition)	$ z < 1$
$a = 0$	$ z = 1$
$a > 0$	$ z > 1$

คุณสมบัติของการแปลงแบบเชิงเส้นแสดงให้ดูง่ายโดยการแทนค่า z ในเทอมของ s ในสมการที่ 8.10 และให้ $s = \sigma + j\omega'$ ผลลัพธ์ที่ได้คือ

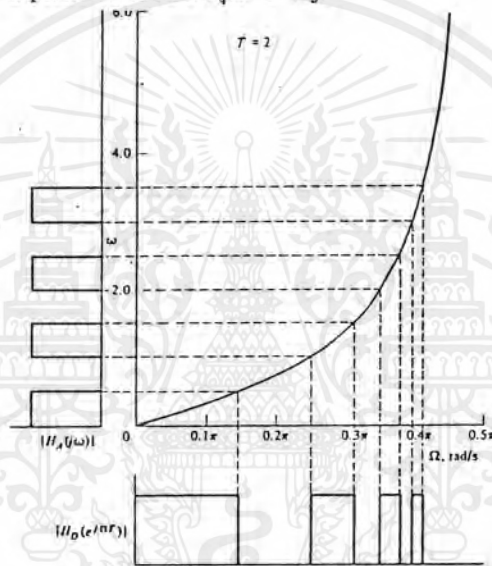
$$z = \frac{1+s}{1-s} = \frac{1+\sigma + j\omega'}{1-\sigma - j\omega'} \quad (8.13)$$

จากตารางที่ 8.2 แสดงให้เห็นชัดเจนว่า โพล ในส่วนครึ่งซ้ายของ s -plane ถูกโยงไปเป็นค่าภายในวงกลมรัศมีหนึ่งหน่วยใน z -plane ในที่สุด ค่า $j\omega$ ถูกโยงไปที่ขอบของวงกลมรัศมีหนึ่งหน่วย การแปลงแบบเชิงเส้นจะมีค่าที่กดไว้ไปเรื่อยในแกนของ $j\omega$ ในวงกลมรัศมีหนึ่งหน่วย การกดนี้ถูกเรียกว่าเป็นความถี่ warping ที่ได้อธิบายมาแล้ว ผลของความไม่เป็นเชิงเส้นถูก plot ในสมการ 8.12 โดยที่ $f = 0$ โยงเป็น $f' = 0$ และ $f = F/2$ ถูกโยงให้เป็น $f' = \infty$ ความถี่นี้สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูได้จากตารางที่ 8.2 เช่น $\infty + j0$ จะโยงไปที่ -1 ที่คล้อยตามค่า $F/2$ ตามสมการที่ 8.12 ถูก plot ในรูปที่ 8.5 ผลตอบสนองขนาดของตัวกรองเชิงเลขถูกขยายที่ความถี่ต่ำ และถูกกดที่ความถี่สูง โดยการเปรียบเทียบกับตัวกรองอนาล็อก

ระบบจะสามารถเสถียรได้โดยพล ทั้งหมคของ $H(z)$ จะต้องวางอยู่ภายในวงกลมรัศมีหนึ่งหน่วยของ z -plane เพื่อให้ได้ค่าตามนี้ดังนั้นการแปลงแบบเชิงเส้นคู่ ต้องเชื่อมโยงด้านซ้ายของ s -plane ไปบนส่วนของวงกลมรัศมีหนึ่งหน่วย ดังนั้นการมีเสถียรภาพของตัวกรองอนาล็อกจะทำให้ตัวกรองดิจิตอลมีเสถียรภาพด้วย จากรูปที่ 8.6 สังเกตได้ว่าจุด C และจุด D ใน s -plane ถูกโยงไปบนส่วนโค้งใน z -plane ผ่านไปถึงจุด $-1 + j0$

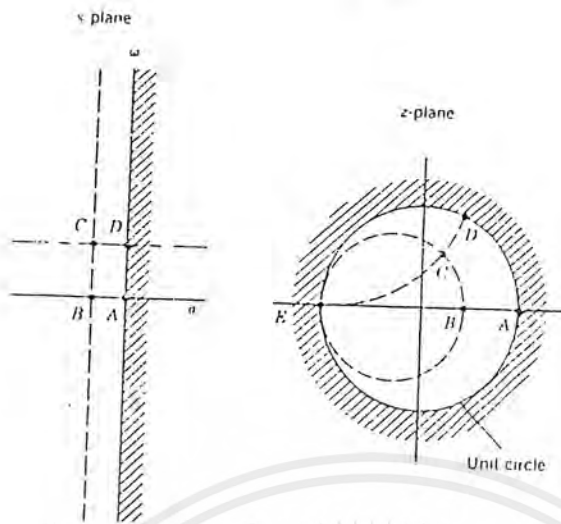


รูปที่ 8.5 ความถี่การส่งผ่านของการแปลงเชิงเส้นคู่

การออกแบบ ของ IIR จากรูปแบบแบบฉบับเดิมของตัวกรอง ใช้การแปลงแบบเชิงเส้นคู่เป็นไปตามกระบวนการออกแบบดังนี้

1. กำหนดจุด ค่าตัวแปรวิกฤตของตัวกรองดิจิตอล : ความถี่ของช่วงผ่าน และช่วงหยุด ในหน่วยของ Hz ช่วงผ่านและช่วงหยุดของการลดทอน ในหน่วย dB และความถี่ของการสุ่ม
2. Prewarp ความถี่อนาล็อกวิกฤต ในสมการที่ 8.12
3. ใช้ความถี่ที่ต้องการในตารางที่ 8.1
4. ใช้การแปลงแบบเชิงเส้นคู่ ของสมการที่ 8.10

รูปแบบของกระบวนการออกแบบได้ในมาใช้ และสมการรูปแบบปิดอย่างง่ายนำมาประยุกต์ใช้ในการออกแบบโดยจำลองผ่านคอมพิวเตอร์



รูปที่ 8.6 แสดงการเชื่อมโยงแบบเชิงเส้นคู่ในระนาบ s และระนาบ z

8.7 สมการการออกแบบตัวกรองเชิงเลข (Digital Filter Design Equation)

ปัจจัยในการออกแบบ ตัวกรองดิจิทัล IIR ในเทคนิควิธีการออกแบบตัวกรองอนาล็อกแบบดั้งเดิมแสดงมาตรฐานการเข้าสู่การออกแบบตัวกรอง IIR ในโครงการนี้จะได้ทำการออกแบบตัวกรองแบบบัทเทอร์เวิร์ธเท่านั้น วิธีการ Methodology ใช้เพื่อหาค่าสัมประสิทธิ์โปรแกรมทางคอมพิวเตอร์ สมการจะใช้ของ การแปลงแบบเชิงเส้นคู่ สำหรับสัมประสิทธิ์ตัวกรองอันดับที่สองของ pole และ zero และ scaling factors จะถูกขยายออกมาของการประมาณค่าตัวกรอง และชนิดทั้งหมด

8.7.1 สมการการออกแบบตัวกรองเชิงเลขความถี่ต่ำ (Lowpass Digital Filter Design Equations)

โดยการในการแปลงช่วงความถี่ สำหรับความถี่ต่ำ สู่ความถี่ต่ำ ที่ให้ไว้ในตารางที่ 8.1 คือ

$$S = s/\Omega$$

โดยที่เราแทนค่าตัวแปรสเกลความถี่ W_p โดยใช้ parameters สเกลความถี่ $\propto$ สำหรับ Different filter approximation ที่ให้ไว้ในตารางที่ 8.3

ตารางที่ 8.3 แสดงตัวแปรความถี่สเกล

FREQUENCY SCALING PARAMETER	α
Butterworth (lowpass)	$(10^{0.1A_p} - 1)^{-1/2N} \tan(\pi f_p / F)$
Butterworth (highpass)	$(10^{0.1A_p} - 1)^{1/2N} \tan(\pi f_p / F)$
Chebyshev	$\tan(\pi f_p / F)$
Elliptic	$(\tan(\pi f_p / F) \tan(\pi f_p / F))^{0.5}$

ตัวแปรความถี่ของการสเกล α สำหรับตัวกรองบัทเทอร์เวิร์ธ หาได้จากสมการ 8.9 ความถี่อนาล็อก W_p หาได้จากสมการ 8.12 การลดทอนของผลตอบสนองที่ความถี่ค่าวิกฤต W_p ถูกกระทบจากค่า $\epsilon^{\pm 1/N}$

พิจารณาที่ ฟังก์ชันการส่งผ่านใน pole เดี่ยวของตัวกรองอนาล็อก โดยการใช้ นอร์มอไลซ์ Pole ความถี่ต่ำ $S_{p,k}$ ของบัทเทอร์เวิร์ธดังนี้

$$H(s) = \frac{-s_{p,k}}{s - s_{p,k}} \quad k = 1, 2, \dots, N \quad (8.14)$$

แทนค่า ค่าสเกลความถี่ ในฟังก์ชันการส่งผ่าน

$$H(s) = \frac{-\alpha s_{p,k}}{s - \alpha s_{p,k}} \quad (8.15)$$

หาค่าของ $H(z)$ โดยคล้อยตามค่าของ $H(s)$ โดยการใช้การแปรแบบเชิงเส้นคู่จะได้ค่าดังนี้

$$H(z) = H(s) \Big|_{s=z-1/z+1} = \frac{\frac{-\alpha s_{p,k}}{1 - \alpha s_{p,k}} (z+1)}{z - \frac{1 + \alpha s_{p,k}}{1 - \alpha s_{p,k}}} \quad (8.16)$$

โดยที่จะให้ค่า $z_{p,k} = \frac{1 - \alpha s_{p,k}}{1 + \alpha s_{p,k}} \quad (8.17)$

รูปแบบทั่วไปของฟังก์ชันการส่งผ่านของตัวกรองดิจิทัลความถี่ต่ำของ IIR สำหรับ บัทเทอร์เวิร์ธและ เชบนิเชฟ คือ

$$H(z) = \frac{N(z)}{D(z)} = \frac{C_0(z+1)^N}{\prod_{K=1}^N (z - z_{p,k})(z - z_{p,k}^*)} \quad (8.18)$$

ค่า z_p เป็นค่าสังยุคจำนวนเชิงซ้อนของ z_p สังเกตได้ว่า Zero ทุกตัวจะอยู่ที่ค่า $z=-1$ บนวงกลมรัศมีหนึ่งหน่วยในระนาบ z โดยที่เราจะอ้างค่า z_p และ z_p^* ใน pole และ zero สำหรับ second-order ของ k ใน โดยทั่วไป ตัวกรองดิจิทัล แบบวงรี Zero เป็นจำนวนเชิงซ้อนและ $N(s)$ คือ

$$N(z) = C_0 \prod_{K=0}^N (z - z_{z,k})(z - z_{z,k}^*) \quad (8.19)$$

second -order ในที่สัมประสิทธิ์ เป็นจำนวนจริงคือ

$$\begin{aligned} A_{1,k} &= -2 \operatorname{Re}[z_{z,k}] \\ A_{2,k} &= \operatorname{Re}[z_{z,k}]^2 + \operatorname{Im}[z_{z,k}]^2 \\ B_{1,k} &= -2 \operatorname{Re}[z_{p,k}] \\ B_{2,k} &= \operatorname{Re}[z_{p,k}]^2 + \operatorname{Im}[z_{p,k}]^2 \end{aligned}$$

โดยที่ $\operatorname{Re}[z_{i,k}]$ แสดงส่วนจริงของของ $z_{i,k}$ และ $\operatorname{Im}[z_{i,k}]$ แสดงส่วนจินตภาพของ $z_{i,k}$ ตำแหน่งของ pole และ zero ระนาบ z หาได้จาก

$$z_{i,k} = \frac{1 - (a_{i,k}^2 + b_{i,k}^2)\alpha^2}{D_{i,k}} + j \frac{2b_{i,k}\alpha}{D_{i,k}} \quad (8.20)$$

โดยที่

$$D_{i,k} = 1 + 2|a_{i,k}|\alpha + (a_{i,k}^2 + b_{i,k}^2)\alpha^2 \quad (8.21)$$

และ

$$K = \begin{cases} 1, 2, \dots, N/2 & \text{for } N - \text{even} \\ 1, 2, \dots, (N+1)/2 & \text{for } N - \text{odd} \end{cases}$$

การประมาณการของบัทเทอร์เวิร์ธคือ $(a_{i,k}^2 + b_{i,k}^2 = 1)$

8.8 ฟังก์ชันส่งผ่านความถี่ต่ำสัมประสิทธิ์ของตัวตั้ง (Lowpass Transfer Function Numerator Coefficients)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 8.18 การประมาณความถี่ต่ำของ บัทเทอร์เวิร์ธและ เชบชีเชฟ มี zero 2 เท่าที่ $z = -1$ ดังนั้นสัมประสิทธิ์ตัวกรองดิจิทัลอันดับที่ 2 ของตัวตั้งค่า

$$(z+1)(z-1) = z^2 + 2z + 1 \quad (8.21)$$

α มีค่าเท่ากับ $1+2z^{-1}+z^{-2}$ สำหรับ บัทเทอร์และ เชบชีเชฟ $A_{1,k}=2$ และ $A_{2,k} = 1$ ดังนั้น zero ของสมการ 4.57 เป็นจำนวนจินตภาพแท้ ซึ่งลดรูปเป็น

$$z_{\pm,k} = \frac{1-b_{\pm,k}^2\alpha^2}{1+b_{\pm,k}^2\alpha^2} = j \frac{2b_{\pm,k}\alpha}{1+b_{\pm,k}^2\alpha^2} \quad (8.22)$$

8.9 ตัวกรองเชิงเลขความถี่ต่ำในส่วนอันดับที่สอง (Lowpass Digital Filter: First-order section)
สำหรับ อันดับคี่ สามารถจัดอยู่ในเทอมอันดับหนึ่งและอันดับสองได้ดังนี้

$$H(z) = H'_1(z) \prod_{K=2}^{(N+1)/2} H_K(z) \quad (8.23)$$

โดยที่

$$G'_1(z) = \frac{1+z^{-1}}{1+B_{1,k}z^{-1}} \quad (8.24)$$

ดังนั้น pole อนาคต สำหรับส่วนอันดับที่หนึ่งที่เป็นจำนวนจริงในสมการที่ 8.20 ลดรูปได้เป็น

$$z_{p,k} = \frac{1-(a_{p,k}\alpha)^2}{1+2|a_{p,k}\alpha|+(a_{p,k}\alpha)^2} = \frac{1-|a_{p,k}\alpha|}{1+|a_{p,k}\alpha|} \quad (8.25)$$

ค่าสัมประสิทธิ์หาได้จาก $B_{1,k}$ สามารถหาได้จาก

$$B_{1,k} = -z_{p,k}$$

สำหรับ บัทเทอร์เวิร์ธ ที่ส่วนอันดับที่หนึ่ง $a_{1,k} = 1$

ตารางที่ 8.4 แสดงค่าสมการอันดับของตัวกรองเชิงเลข

Approximation	Filter Order Equation
Butterworth	$N > \log A / (\log (1 / K))$
Chebyshev	$N > \cosh ^{-1} A / \cosh ^{-1} (1 / K)$
Elliptic	$N > \log 16 A^2 / \log (1 / q)$
LowPass	Highpass
$K=K_0=\frac{\tan (\pi f_p / F)}{\tan (\pi f_s / F)}$	$K=1 / K_0$
Bandpass	Bandstop
$K=K_1$, if $K_C \geq K_B$ K_2 , if $K_C \leq K_B$ $K_A=\tan (\pi f_{p 2} / F)-\tan (\pi f_{p 1} / F)$ $K_B=\tan (\pi f_{p 1} / F) \tan (\pi f_{p 2} / F)$ $K_C=\tan (\pi f_{s 1} / F) \tan (\pi f_{s 2} / F)$ $K_1=\frac{K_A \tan (\pi f_{s 1} / F)}{K_B-\tan (\pi f_{s 1} / F)}$	$K=1 / K_1$, if $K_C \geq K_B$ $1 / K_2$, if $K_C \leq K_B$ $K_2=\frac{K_A \tan (\pi f_{s 2} / F)}{\tan (\pi f_{s 2} / F)-K_B}$
Elliptic design parameter $q=q_0+2 q_0^5+15 q_0^9+150 q_0^{13}$	
$q_0=\frac{1-\left(1-K^2\right)^{0.25}}{2\left[1+\left(1-K^2\right)^{0.25}\right]}$	

ตารางที่ 8.5 แสดง Digital Filter Numerator Coefficients

coef.	Butterworth / Chebyshev					Elliptic							
	Filter Type					Filter Type							
	LP		HP		BP	BS	LP		HP		BP		BS
$A_{0,k}$	1	(1)	1	(1)	1	1	1	(1)	1	(1)	1	(1)	1
$A_{1,k}$	2	(1)	-2	(-1)	0	$A_{1,k}$	$A_{1,k}$	(1)	$A_{1,k}$	(-1)	$A_{1,k}$	(0)	$A_{1,k}$
$A_{2,k}$	1	(0)	1	(0)	-1	1	1	(0)	1	(0)	1	(-1)	1
Equation						4.99	4.60		4.74		4.91		4.100

The coefficients in parentheses correspond to the first and second-order odd section transfer functions (refer to Eqs. 4.110a-4.110d).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการการออกแบบโดยมุ่งหาอันดับของ N ของ บัทเทอร์เวิร์ธ โดยใช้ตาราง 4.4 โดยที่ตัวตั้งของสัมประสิทธิ์อันดับที่สองที่อยู่ในรูปพหุนามดังนี้

$$1 + A_{1,k}z^{-1} + A_{2,k}z^{-2}$$

8.10 สมการการออกแบบตัวกรองเชิงเลขความถี่สูง (Highpass Digital Filter Design Equations)

การแปลงช่วงการความถี่นาลอกสำหรับการออกแบบจากความถี่ต่ำเป็นความถี่สูงได้จาก

$$S = \alpha/S$$

โดยที่ α ได้จากตารางที่ 8.3 ฟังก์ชันการส่งผ่านเกลาความถี่ ของ pole เดี่ยว อนาลอกตัวกรองความถี่สูง หาได้ตามนี้

$$H(s) = \frac{-s_{i,k}}{s - s_{i,k}} \Big|_{s=\alpha/s} = \frac{-s_{i,k}s}{\alpha - s_{i,k}s} \quad (8.26)$$

แทนการแปลงเชิงเส้นคู่ ของ s เราจะได้ฟังก์ชันส่งผ่านของตัวกรองเชิงเลขในเทอมของ pole อนาลอกความถี่ต่ำ คือ

$$H(z) = H(s) \Big|_{s=z^{-1}/z+1} = \frac{\frac{-s_{i,k}s}{(\alpha - s_{i,k}s)}(z-1)}{z + \frac{\alpha + s_{i,k}}{\alpha - s_{i,k}}} \quad (8.27)$$

ดังนั้นสมการตำแหน่งของ pole หรือ zero ของดิจิทัลในระนาบ z คือ

$$z_{i,k} = \frac{-(\alpha + s_{i,k})}{\alpha - s_{i,k}} \quad I = p \text{ หรือ } z \quad (8.28)$$

โดยที่ $s_{i,k}$ คือ pole หรือ zero ของ บัทเทอร์เวิร์ธ และ ฟังก์ชันส่งผ่านของบัทเทอร์เวิร์ธ และเซบปีเซฟ ตัวกรองเชิงเลขคือ

$$N(z) = \frac{N(z)}{D(z)} = \frac{C_0(z-1)^N}{\prod_{k=1}^N (z - z_{p,k})(z - z_{i,k}^*)} \quad (8.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้เราให้ $s_{i,k} = a_{i,k} + jb_{i,k}$ แสดง pole และ zero ของอนาล็อกความถี่ต่ำ ของอันดับ k ของ second-order ระบุ z ของ pole และ zero มีค่าดังนี้

$$z_{i,k} = \frac{-[\alpha^2 - (a_{i,k}^2 + b_{i,k}^2)]}{D_{i,k}} \pm j \frac{2b_{i,k}\alpha}{D_{i,k}}$$

$$D_{i,k} = \alpha^2 + 2|a_{i,k}|\alpha + a_{i,k}^2 + b_{i,k}^2 \quad (8.30)$$

$$k = \begin{cases} 1, 2, \dots, N/2 & \text{for } N - \text{even} \\ 1, 2, \dots, (N+1)/2 & \text{for } N - \text{odd} \end{cases}$$

$i = p$ หรือ z

8.11 ฟังก์ชันส่งผ่านส่วนอันดับที่สองของสัมประสิทธิ์ตัวตั้ง (Second - order Section Transfer Function Numerator coefficients)

สำหรับการประมาณของบัทเทอร์เวิร์ธและเชบชีเฟฟที่ความถี่สูง มี zero สองตัวที่ $z = 1$ โดยมีสมการดังนี้

$$(z-1)(z-1) = z^2 - 2z + 1 \quad (8.31)$$

ดังนั้น $A_{1,k} = -2$ และ $A_{2,k} = 1$ (ตามตารางที่ 8.5)

8.12 สมการการออกแบบตัวกรองเชิงเลขของแบนด์พาส (Bandpass Digital Filter Design Equation)

การแปลงจากความถี่ต่ำเป็น ความถี่ช่วงผ่าน โดยใช้ค่าดังนี้

$$s_1 = \frac{s_b^2 + \omega_0'}{\beta \Delta \omega' s_b} \quad (8.32)$$

โดยที่ s_1 คือ pole ของอนาล็อกความถี่ต่ำ และ s_b คือ pole ของอนาล็อกแบนด์พาส ที่ได้จาก แก๊สมการ 8.32 s_b เราหาได้จากสมการนี้

$$s_b^2 - \beta \Delta \omega' s_1 s_b + \omega_0' = 0 \quad (8.33)$$

และ

$$\omega_u' = \tan(\pi f_{pu} / F)$$

$$\omega_l' = \tan(\pi f_{pl} / F)$$

$$\Delta \omega' = (\omega_u' - \omega_l')$$

$$\omega_0' = \omega_u' \omega_l'$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ f_{pu} และ f_{pl} เป็นความถี่ช่วงผ่านความถี่ต่ำและความถี่สูงตามลำดับ ค่าตัวแปล β ใช้เพื่อหาความถี่ที่ต้องการซึ่งได้จากตารางที่ 8.6 ที่เป็นฟังก์ชันของการประมาณการตัวกรองและชนิดตัวกรองความถี่ศูนย์กลางเลขาคณิตของช่วงผ่านของตัวกรอง bandpass ที่ได้จากค่ากลางเลขาคณิตการแก้สมการ 8.34 เพื่อหาโพล อนุลอก bandpass ได้ค่าดังนี้

$$s_b = \frac{\beta \Delta \omega'}{2} \left[s_l \pm \left(s_l^2 - \frac{4\omega_0'}{(\beta \Delta \omega')^2} \right)^{0.5} \right] \quad (8.35)$$

โดยที่เราจะให้โพลของอนุลอกความถี่ต่ำ มีค่าดังนี้

$$s_l = a_p + jb_p$$

ค่า a_p เป็นค่าบวกเราจะได้ค่าดังนี้

$$s_{bp} = \frac{\beta \Delta \omega'}{2} \left[-|a_p| + jb_p \pm \left(-c_1 - j_2 |a_p| b_p \right)^{0.5} \right] \quad (8.36)$$

เราจะได้ค่า

$$c_1 = b_p^2 + \left[\frac{4\omega_0'}{(\Delta \omega' \beta)^2} \right] - a_p^2 \quad (8.37)$$

ซึ่งมันแสดงให้เห็นค่าของรากที่สองของจำนวนเชิงซ้อนได้ดังนี้

$$(-c_1 + jy)^{0.5} = \left[\frac{(c_1^2 + y^2)^{0.5} - c_1}{2} \right]^{0.5} + j \left[\frac{(c_1^2 + y^2)^{0.5} + c_1}{2} \right]^{0.5} \quad (8.38)$$

จะให้ค่า $y = -2a_p b_p$ นำค่าในสมการ 4.83 แทนในสมการที่ 4.81 ได้ค่าดังนี้

$$s_{bp} = \frac{\beta \Delta \omega'}{2} \left\{ -|a_p| + jb_p \pm \left[\left(\frac{(c_1^2 + 4a_p^2 b_p^2)^{0.5} - c_1}{2} \right)^{0.5} + j \left(\frac{(c_1^2 + 4a_p^2 b_p^2)^{0.5} + c_1}{2} \right)^{0.5} \right] \right\} \quad (8.39)$$

ค่าโพลอนุลอก Bandpass prewarp หาได้ตามนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 s_{bp1} &= \frac{\beta\Delta\omega'}{2} \left[(-|a_p| - c_3) \pm j(b_p + c_4) \right] = a_{bp1} + jb_{bp2} \\
 s_{bp2} &= \frac{\beta\Delta\omega'}{2} \left[(-|a_p| + c_3) \pm j(b_p - c_4) \right] = a_{bp1} + jb_{bp2} \\
 c_2 &= \left[c_1^2 + (2a_p b_p)^2 \right]^{0.5} \\
 c_3 &= \left(\frac{c_2 - c_1}{2} \right)^{0.5} \\
 c_4 &= \left(\frac{c_2 + c_1}{2} \right)^{0.5}
 \end{aligned} \tag{8.39}$$

c_1 หาได้จากสมการที่ 8.39 และในกรณีในค่า c_5 ได้จากสมการที่ 8.38 ดังนี้

$$c_5 = b_{z,k}^2 + \frac{4\omega_0'}{(\beta\Delta\omega')^2} \tag{8.40}$$

และ $b_{z,k} = 1/\Omega_k$ เป็นค่า zero อนุลอกความถี่ต่ำ

สำหรับอันดับที่ N ของตัวกรอง บัทเทอร์เวิร์ธ หรือ เชบปีเซฟการแปลงความถี่bandpass เป็นผลลัพธ์ของการโยง zero ที่ N ที่จุดกำเนิดในระนาบ s การคิดโคสของตัวกรองอนุลอก Bandpass ใช้การแปลงเชิงเส้นคู่ที่มีค่า zero ที่ $\pm$ ในระนาบ z สำหรับช่วงอันดับที่สองดังนั้นสิ่งที่แสดงในตารางที่ 4.5 ค่าสัมประสิทธิ์ตัวเลขอันดับที่สองของ บัทเทอร์เวิร์ธ หรือ เชบปีเซฟจึงไม่จำเป็นต้องคำนวณ

ผลลัพธ์จากการวิเคราะห์นี้แสดงให้เห็นได้ว่าการแปลงจากความถี่ต่ำเป็นความถี่ช่วงผ่านสร้าง N bandpass ที่เป็นโพลคู่ที่คล้อยตามค่า ที่ N ของโพลอนุลอกความถี่ต่ำ ตัวกรองอันดับที่ N ของ Bandpass และ Bandstop ตัวกรองเชิงเลขแสดงจำนวนอันดับที่สอง สำหรับความถี่ต่ำและความถี่สูงการออกแบบนั้นจะสร้าง $N/2$ ขึ้นมา ดังนั้นโพลและซีโรช่วงผ่านอนุลอกได้จากโพลและซีโรอนุลอกความถี่ต่ำของตัวกรอง elliptic ได้โดยค่า

$$\begin{aligned}
 s_{bp1} &= a_{bp1} \pm jb_{bp1} \\
 s_{bp2} &= a_{bp2} \pm jb_{bp2} \\
 s_{bz1} &= \pm jb_{bz1} \\
 s_{bz2} &= \pm jb_{bz2}
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลของพาสแบนด์สามารถได้จากการแปลงเชิงเส้นคู่ตามค่าดังนี้

$$z_{p,k} = \frac{1 + s_{bp,k}}{1 - s_{bp,k}} \quad k = 1, 2, \dots, N \quad (8.41)$$

แทนค่าในสมการที่ 8.40 ได้ดังนี้

$$z_{p,k} = \frac{1 - (a_{bp,k}^2 + b_{bp,k}^2)}{D_{bp,k}} + j \frac{b_{bp,k}}{D_{bp,k}} \quad (8.42)$$

$$D_{bp,k} = 1 + 2|a_{bp,k}| + a_{bp,k}^2 + b_{bp,k}^2$$

8.12 ตัวกรองเชิงเลขแบบแบนด์พาส ในอันดับที่สองสำหรับ N ที่เป็นคี่ (Bandpass Digital Filter

:Second Order Section For N - odd)

สำหรับ N ที่เป็นคี่ ฟังก์ชันส่งผ่านผลลัพธ์ที่ได้คือ

$$H'_1(z) = \frac{1 - z^2}{1 + B_{1,k}z^{-1} + B_{2,k}z^{-2}} \quad (8.43)$$

ค่าสัมประสิทธิ์ $B_{1,k}$ และ $B_{2,k}$ หาได้จากการวิเคราะห์หว่าสำหรับ N คี่ โพลของอนาล็อกความถี่ต่ำเป็นจำนวนจริงแท้ สำหรับบัททอร์เวิร์ธ เชบชีเชฟ ค่าอนาล็อกที่แท้จริงหาได้จากสมการที่ 4.13 , 4.32 สำหรับนอร์มอลไลซ์โพลมีค่าเท่ากับ -1 ตัวอย่างเช่น พิจารณาที่ การประมาณการตัวกรองอิลิปติก โดยที่ $-a_p = \sigma_0$ จะได้ค่าจากสมการที่ 4.85a โพลอนาล็อกพาสแบนด์ คือ

$$s_{bp,k} = \frac{\beta \Delta \omega'}{2} [\sigma_0 \pm jc_1^{0.5}] = a_{bp,k} \pm jb_{bp,k} \quad (8.44)$$

ตารางที่ 8.6 แสดงค่า ความถี่พารามิเตอร์เฉพาะใน แบนด์พาสและแบนด์สต็อป

ชนิดและการประมาณการตัวกรอง	β
บัทเทอร์เวิร์ธ BPF	$(10^{0.1A_p} - 1)^{-1/2N}$
บัทเทอร์เวิร์ธ BSE	$(10^{0.1A_p} - 1)^{1/2N}$
เชบนิเชฟ (BPE/BSE)	1
Elliptic BPE	$K^{-0.5}$
Elliptic BSE	$K^{0.5}$

โดยที่ C_1 ได้จากสมการที่ 4.82 สังเกตได้ว่า $b_p = 0$

สำหรับนาลอกแบนด์พาสโพล ของตัวกรองelliptic ได้จากสมการ ที่ 8.44 ได้ค่าดังนี้

$$s_{bp,k} = 0.5\beta\sigma_0\Delta\omega' \pm j[\omega'_0 - (0.5\beta\sigma_0\Delta\omega')^2]^{0.5} \quad 8.45$$

โพลดิจิตอลได้ คือ

$$\tilde{z}_{bp,k} = \frac{1 - \omega'_0}{D_{bp,k}} + j \frac{2[\omega'_0 - (0.5\beta\sigma_0\Delta\omega')^2]^{0.5}}{D_{bp,k}}$$

$$\tilde{D}_{bp,k} = 1 - \beta\Delta\omega'\sigma_0 + \omega'_0$$

8.13 สมการการออกแบบตัวกรองเชิงเลขแบนด์สต็อป(Bandstop Digital Filter Design Equations)

จากตารางที่ 8.1 การแปลงจากความถี่ต่ำเป็นความถี่สูงได้จากค่านี้

$$s_1 = \frac{\beta\Delta\omega's_b}{s_b^2 + \omega'_0} \quad (8.46)$$

หาค่า s_b ได้จากแก้สมการซึ่งก็คือ

$$s_b = \frac{\beta\Delta\omega'}{2} \left[s_1^{-1} \pm (s_1^{-2} - \frac{4\omega'_0}{(\beta\Delta\omega')^2})^{0.5} \right] \quad (8.47)$$

ซึ่งเหมือนสมการที่ 8.33 ยกเว้นแต่ค่า s_1 ถ้าเราให้ค่าโพลนาลอกความถี่ต่ำเป็นดังนี้คือ

$s_1 = -a_p + jb_p$ ดังนั้นจะได้

$$-a'_p \pm jb'_p = \frac{1}{-a_p + jb_p} = \frac{-|a_p|}{a_p^2 + b_p^2} \pm j \frac{b_p}{a_p^2 + b_p^2} \quad (8.48)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ a_p มีค่าเป็นบวก เราสามารถออกแบบตัวกรองเชิงเลขแบบคัสต็อบได้จากใช้สมการที่ 8.40 ดังนั้นการออกแบบตัวกรองเชิงเลขแบบคัสต็อบเหมือนกับการออกแบบตัวกรองเชิงเลขแบบแบนด์พาสโดยการใช้วิธีง่ายจาก ค่าผ่านของโพลอนาลอกความถี่ต่ำได้

8.14 การคำนวณสัมประสิทธิ์ตัวเลขของตัวกรองแบบคัสต็อบ

ตัวกรองเชิงเลขแบบคัสต็อบมุ่งสนใจที่ค่าสัมประสิทธิ์ค่า $A_{1,k}$ ซึ่งเป็นฟังก์ชันของชนิดตัวกรองตัวกรองบัทเทอร์เวิร์ธและเชบชีเชฟ ซีโรนาลอกแบนด์พาสได้รับจากสมการที่ 8.47 เพราะวอร์มอไลซ์อนาลอกความถี่ต่ำมีซีโรที่อนันต์ในระนาบ s การแปลงความถี่แบบคัสต็อบจะโยงเข้าคู่ของซีโรที่

$$s_{bz} = \pm j(\omega'_0)^{0.5}$$

สำหรับตัวกรองelliptic ซีโรนาลอกความถี่ต่ำได้จากสมการสำหรับในกรณีนี้ซีโรของอนาลอกแบนด์คัสต็อบอยู่ที่

$$s_{bz,k} = \pm j \frac{\beta \Delta \omega'_0}{2} \left[\Omega_k \pm \left(\Omega_k^2 + \frac{4\omega'_0}{(\Delta \omega' \beta)^2} \right)^{0.5} \right] \quad (8.49)$$

8.15 ตัวกรองเชิงเลขแบบคัสต็อบอันดับที่สองสำหรับ N คี่ (Bandstop Digital Filter : Second - Order Section For N -odd)

โพลอนาลอกความถี่ต่ำลำดับคี่สำหรับชนิดตัวกรองทั้งหมดที่เป็นค่าจำนวนจริงแท้ สำหรับตัวกรองบัทเทอร์เวิร์ธและเชบชีเชฟ ได้จาก

$$s_{bp} = \frac{\beta \Delta \omega'}{2} \left[\frac{1}{a_p} + j(c_1)^{0.5} \right] \quad (8.50)$$

โดยที่ค่า c_1 ได้จากสมการที่ 8.37 ด้วยค่า $b_p = 0$ สำหรับบัทเทอร์เวิร์ธในกรณีเช่นนี้จะมี $a_p - 1$ สำหรับค่าทุกค่าของ N สำหรับตัวกรอง elliptic โพลที่เป็นจำนวนคี่ได้จากสมการนี้

$$s_{bz(1,2)} = \frac{\beta \Delta \omega'}{2} \left[\frac{1}{\sigma_0} \pm \left(\frac{1}{\sigma_0^2} - \frac{4\omega'_0}{(\Delta \omega' \beta)^2} \right)^{0.5} \right] \quad (8.51)$$

โพลของตัวกรองแบบคัสต็อบสำหรับอันดับที่สองได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} z_{p,1} &= \frac{1 - |s_{bp,1}|}{1 + |s_{bp,1}|} \\ z_{p,2} &= \frac{1 - |s_{bp,2}|}{1 + |s_{bp,2}|} \end{aligned} \quad (8.52)$$

ในที่สุดเราจะได้ตัวหารในสัมประสิทธิ์คือ

$$\begin{aligned} B_{1,k} &= -(z_{p,1} + z_{p,2}) \\ B_{2,k} &= z_{p,1} z_{p,2} \end{aligned}$$

ซีโรนาลอกแบนด์สตีปสำหรับตัวกรองทุกชนิดได้จากสมการที่ 8.52

และเราได้ซีโรดิคิตอลได้จากสมการ 8.42 ในรูปแบบดังนี้

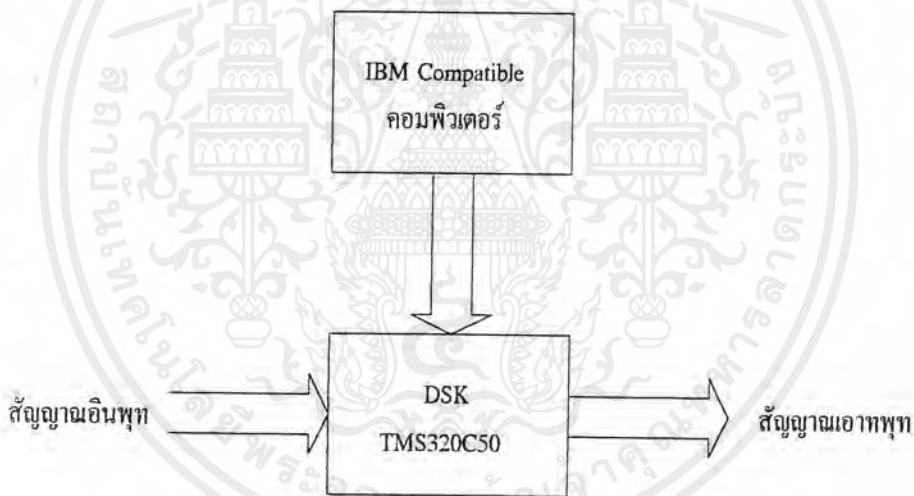
$$z_{pk} = \frac{1 - \omega'_0}{1 + \omega'_0} + j \frac{2\omega_0'^{0.5}}{1 + \omega'_0} \quad (8.53)$$

บทที่ 9

โครงสร้างของโครงการ

โครงสร้างของโครงการนี้สามารถแสดงได้ดังรูปที่ 9.1 โดยจากรูปจะเห็นว่าโครงการนี้จะประกอบด้วยส่วนสำคัญ 2 ส่วน ด้วยกัน คือ

1. IBM Compatible คอมพิวเตอร์ ที่ใช้ในการคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลข แล้วส่งค่าสัมประสิทธิ์เหล่านี้ไปให้ตัวบอร์ค DSK TMS320C50 ที่ทำตัวเป็นวงจรกรองเชิงเลข ซึ่งค่าสัมประสิทธิ์เหล่านี้จะรับค่าคุณสมบัติของวงจรกรองที่ต้องการจากผู้ใช้แล้วนำมาคำนวณค่าสัมประสิทธิ์ โดยใช้โปรแกรม TURBO PASCAL



รูปที่ 9.1 บล็อกไดอะแกรมแสดงโครงสร้างของโครงการ

2. บอร์ค DSK TMS320C50 ที่เขียนโปรแกรมให้ทำตัวเป็นวงจรกรองเชิงเลข โดยใช้ค่าสัมประสิทธิ์ที่ได้รับจากคอมพิวเตอร์ในการคำนวณ เป็นวงจรกรองเชิงเลข ทำการกรองสัญญาณอินพุตที่เข้ามาและได้สัญญาณเอาทพุตที่ออกจากบอร์ค

ซึ่งรายละเอียดของส่วนต่าง ๆ จะกล่าวถึงในบทต่อไป โดย บทที่ 10 จะกล่าวถึงการอินเทอร์เฟซระหว่าง IBM คอมพิวเตอร์ กับ บอร์ด DSK TMS 320C50 บทที่ 11 จะกล่าวถึงการเขียนโปรแกรมให้ตัวบอร์ด DSK TMS320C50 ทำตัวเป็นวงจรกรองเชิงเลข รวมถึงการเขียนโปรแกรม TURBO PASCAL ที่ใช้ในการคำนวณค่าสัมประสิทธิ์ และ ติดต่อกับตัวบอร์ด และ บทที่ 12 จะเป็นการทดลองวงจรกรองเชิงเลข เพื่อตรวจสอบสนองทางความถี่ว่าเป็นไปตามที่ออกแบบไว้หรือไม่ รวมไปถึงสรุปผลการทดลองที่ได้



บทที่ 10

การอินเทอร์เฟสระหว่าง IBM คอมพิวเตอร์ กับ บอร์ด DSK TMS 320C50

การติดต่อระหว่าง IBM คอมพิวเตอร์ กับ บอร์ด DSK TMS320C50 จะใช้การติดต่อผ่านพอร์ตนานซึ่งเป็นพอร์ทพริ้นเตอร์ของคอมพิวเตอร์ ซึ่งพอร์ตนานนี้จะมีทั้งหมด 25 ขา (25 pin) โดยในแต่ละขาของพอร์ทพริ้นเตอร์จะถูกกำหนดหน้าที่เอาไว้ดังแสดงในตารางที่ 10.1

ตารางที่ 10.1 แสดงหน้าที่ของแต่ละขาของพอร์ทพริ้นเตอร์

หน้าที่	รีจิสเตอร์ บิท	ตำแหน่งขา	ทิศทาง อินพุต/เอาพุต (I/O)
Strobe	C0	1	O
Databit0	D0	2	O
Databit1	D1	3	O
Databit2	D2	4	O
Databit3	D3	5	O
Databit4	D4	6	O
Databit5	D5	7	O
Databit6	D6	8	O
Databit7	D7	9	O
Acknowledge	S6	10	I
Busy	S7	11	I
Paper End	S5	12	I
Select End	S4	13	I
Auto Feed	C1	14	O
Error	S3	15	I

ตารางที่ 10.1 แสดงหน้าที่ของแต่ละขาของพอร์ตพรีนเตอร์ (ต่อ)

หน้าที่	รีจิสเตอร์ บิต	ตำแหน่งขา	ทิศทาง อินพุต/เอาพุต (I/O)
Initialize	C2	16	O
Select	C3	17	O
Ground	-	18 - 25	-

หมายเหตุ : Cn หมายถึง ตำแหน่งบิตที่ n ในรีจิสเตอร์ควบคุม

Dn หมายถึง ตำแหน่งบิตที่ n ในรีจิสเตอร์ข้อมูล

Sn หมายถึง ตำแหน่งบิตที่ n ในรีจิสเตอร์สถานะ

การติดต่อผ่านพอร์ตพรีนเตอร์นี้จะมีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 3 ตัวได้แก่

1. รีจิสเตอร์ข้อมูล (Data Register)
2. รีจิสเตอร์สถานะ (Status Register)
3. รีจิสเตอร์ควบคุม (Control Register)

ซึ่งรีจิสเตอร์ทั้งสามตัวนี้จะมีขนาด 8 บิต แต่ละบิตในรีจิสเตอร์จะกำหนดหน้าที่เอาไว้แน่นอน
สำหรับการใช้ในการติดต่อกับพรีนเตอร์ ดังแสดงในรูปที่ 10.1

โดยที่รีจิสเตอร์แต่ละตัวจะมีตำแหน่งแอดเดรสที่ใช้ในการติดต่อกับพรีนเตอร์ดังนี้

รีจิสเตอร์ข้อมูลมีตำแหน่งแอดเดรสที่ 0378h

รีจิสเตอร์สถานะมีตำแหน่งแอดเดรสที่ 0379h

รีจิสเตอร์ควบคุมมีตำแหน่งแอดเดรสที่ 037ah



รูปที่ 10.1 แสดงหน้าที่ของแต่ละบิตในรีจิสเตอร์ ก) รีจิสเตอร์ข้อมูล ข) รีจิสเตอร์สถานะ
ค) รีจิสเตอร์ควบคุม

ดูสรุปในรูปที่ 10.1 แสดงทิศทางของบิตตำแหน่งนั้น ๆ ในการติดต่อกันระหว่าง พอร์ทพรีนเตอร์กับคอมพิวเตอร์ โดยทิศทางออกหมายถึงคอมพิวเตอร์ส่งค่าบิตนั้นให้กับพรีนเตอร์ ส่วนทิศทางเข้าหมายถึงพรีนเตอร์ส่งค่าบิตนั้นให้แก่คอมพิวเตอร์

ในการติดต่อกันระหว่างคอมพิวเตอร์ กับ บอร์ด DSK TMS320C50 สามารถทำได้โดยการเขียนโปรแกรมให้คอมพิวเตอร์ "OUT" ค่าสัมประสิทธิ์ที่คำนวณได้ผ่านทางรีจิสเตอร์ข้อมูลที่มีขนาด 8 บิต ซึ่งบอร์ดก็จะรับค่าสัมประสิทธิ์ไปใช้ในการคำนวณวงจรกรองต่อไป แต่เนื่องจากขนาดของข้อมูลที่คอมพิวเตอร์สามารถส่งได้มีขนาดครั้งละ 8 บิต ในขณะที่ข้อมูลที่บอร์ดสามารถนำมาใช้ในการคำนวณมีขนาด 16 บิต หรือ 1 เวิร์ด ดังนั้นจึงต้องมีการจัดข้อมูลก่อนที่จะส่งให้แก่ตัวบอร์ด เพื่อให้สามารถใช้ประสิทธิภาพในการคำนวณของบอร์ดได้อย่างเต็มที่ ซึ่งรายละเอียดของการส่งข้อมูลจากคอมพิวเตอร์ไปให้แก่บอร์ดซึ่งเกี่ยวข้องกับการเขียนโปรแกรม จะขอกล่าวถึงในบทต่อไป

ทางด้านของบอร์ด DSK TMS320C50 จะรับข้อมูลจากคอมพิวเตอร์ผ่านทาง พอร์ทขนาน อินพุต/เอาพุต ซึ่งมีอยู่ทั้งหมด 16 พอร์ท โดยที่พอร์ทขนานเหล่านี้สามารถสั่งให้ทำการอ่านข้อมูลเข้ามา หรือ เขียนข้อมูลออกไป โดยใช้คำสั่ง IN หรือ OUT เมื่อพอร์ทมีการอ่านหรือเขียนข้อมูล ขาสัญญาณ A0 ถึง A15 จะมีสถานะลอจิกต่ำหรือสูงขึ้นอยู่กับค่าแอดเดรสของพอร์ทนั้น ๆ ในหน่วยความจำข้อมูล (50h ถึง 5fh) ของ TMS320C50 ขาสัญญาณ IS จะแอกทีฟหรือมีลอจิก

ค่า เพื่อแสดงว่ามีการติดต่อกับ I/O ส่วนขาสัญญาณ RD และ WE จะแอกทีฟขึ้นอยู่กับการอ่าน (RD) หรือ เขียน (WE) ข้อมูลโดยจะแอกทีฟ ที่ลอจิกค่า

วงจรที่ใช้ในการติดต่อระหว่างบอร์ด กับ คอมพิวเตอร์ที่ได้แสดงไว้ในภาคผนวก สามารถอธิบายการทำงานได้ดังนี้

1. เมื่อคอมพิวเตอร์ส่งข้อมูลออกมาข้อมูลนั้นจะถูกเก็บไว้ใน D Flip-Flop (74LS373)
2. เมื่อบอร์ด DSK TMS320C50 ทำการ "IN PORT" หรือทำการอ่านข้อมูล ขาสัญญาณ IS และ RD จะมีลอจิกค่า และ ขาสัญญาณ A0 ถึง A15 ซึ่งเป็นขาสัญญาณแอดเดรสของบอร์ดจะมีสถานะตามตำแหน่ง พอร์ตขนาน ของบอร์ด
3. ขาสัญญาณ A0 ถึง A3 และ ขาสัญญาณ IS ที่ต่ออยู่กับคิโคเคอร์ (74LS138) จะทำให้คิโคเคอร์ทำการคิโคค้อออกมาว่าติดต่อกับพอร์ตใด แล้วส่งสัญญาณเอาทพุทนี้ให้แก่ D Flip-Flop
4. ขาสัญญาณ RD และ สัญญาณเอาทพุทจากคิโคเคอร์ ซึ่งต่ออยู่กับขาสัญญาณ เอาทพุทคอนโทรล กับ ขาอินาเบิล ตามลำดับ จะทำให้ D Flip-Flop ส่งค่าข้อมูลที่ได้รับจากคอมพิวเตอร์ก่อนหน้านี้ส่งไปให้แก่ บอร์ด ทางบัสข้อมูล
5. หลังจากนั้นจะกลับไปเริ่มที่ข้อ 1 ใหม่ เพื่อทำการส่งข้อมูลตัวต่อไป โดยการทำงานทั้งหมดนี้จะอาศัยการเขียนโปรแกรมบนบอร์ด และ บนคอมพิวเตอร์ ให้ทำงานสัมพันธ์กันในการรับข้อมูล ซึ่งรายละเอียดจะกล่าวถึงในบทต่อไป

บทที่ 11

โปรแกรมที่ใช้ในการทำวงจรกรองเชิงเลข

โปรแกรมที่ใช้ในการทำวงจรกรองเชิงเลขจะประกอบด้วยโปรแกรม 2 ส่วนที่สำคัญด้วยกัน คือ

1. โปรแกรมที่เขียนด้วย TURBO PASCAL ที่ใช้ในการรับค่าคุณสมบัติของวงจรกรองที่ผู้ใช้งานต้องการ แล้วนำมาคำนวณค่าสัมประสิทธิ์ บนคอมพิวเตอร์ และส่งค่าสัมประสิทธิ์ที่ได้ให้แก่บอร์ด เพื่อใช้ในการคำนวณวงจรกรองเชิงเลขต่อไป จะประกอบด้วยโปรแกรม

1.1 FIR-IIR.PAS เป็นโปรแกรมที่ทำหน้าที่ในการรับค่าคุณสมบัติของวงจรกรองที่ต้องการจากผู้ใช้งาน แล้วนำมาคำนวณค่าสัมประสิทธิ์ของวงจรกรอง ส่งไปให้แก่บอร์ด

DSK

1.2 LOAD-DSK.PAS เป็นโปรแกรมที่ทำหน้าที่เรียก โหลดโปรแกรม .DSK ให้แก่บอร์ด โดยถ้าผู้ใช้งานต้องการวงจรกรองเชิงเลขแบบป้อนกลับ (FIR) โปรแกรมนี้จะโหลดโปรแกรม FIR.DSK ให้แก่บอร์ด DSK แต่ ถ้าผู้ใช้งานต้องการวงจรกรองเชิงเลขแบบ ไม่ป้อนกลับ (IIR) โปรแกรมนี้จะโหลด โปรแกรม IIR.DSK ให้แก่บอร์ด DSK

2. โปรแกรม .ASM ที่เขียนด้วยภาษาแอสเซมบลีที่เขียนให้บอร์ด DSK TMS320C50 ทำงานเป็นวงจรกรองเชิงเลข โดยใช้ค่าสัมประสิทธิ์ที่ได้รับจากคอมพิวเตอร์ จะประกอบโปรแกรม

2.1 FIR.ASM เป็นโปรแกรมที่ทำให้บอร์ด DSK ทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR)

2.2 IIR.ASM เป็นโปรแกรมที่ทำให้บอร์ด DSK ทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลขแบบป้อนกลับ (IIR)

โดยโปรแกรมทั้งสองนี้จะใช้ค่าสัมประสิทธิ์ที่คำนวณได้จากคอมพิวเตอร์มาใช้ในการคำนวณเป็นวงจรกรองเชิงเลข ซึ่งโปรแกรมทั้งสองนี้จะต้องทำการแปลงเป็นโปรแกรม .DSK ก่อนที่จะโหลดให้แก่บอร์ด DSK เพื่อทำให้บอร์ดทำหน้าที่เป็นวงจรกรองตามที่ต้องการได้

ทั้ง 4 โปรแกรมนี้สามารถดูได้ที่ภาคผนวก ส่วนรายละเอียดของแต่ละโปรแกรมจะได้อธิบายต่อไป

11.1 โปรแกรมที่ใช้ในการคำนวณค่าสัมประสิทธิ์

คือโปรแกรม FIR-IIR.PAS เป็นโปรแกรมที่ใช้ในการรับค่าคุณสมบัติของวงจรกรองที่ต้องการจากผู้ใช้ แล้วนำคุณสมบัติเหล่านั้นมาคำนวณค่าสัมประสิทธิ์ แล้วส่งไปให้แก่ บอร์ด DSK ในการคำนวณเป็นวงจรกรองเชิงเลขต่อไป เป็นโปรแกรมที่เขียนด้วย TURBO PASCAL สามารถแสดงเป็น โพลชาร์ท ได้ดังรูปที่ 11.2

จากรูปที่ 11.2 สามารถอธิบายการทำงานของโปรแกรมได้ดังนี้ เมื่อเริ่มทำงานโปรแกรม จะคอยรับค่าจากผู้ใช้ว่าจะให้เป็นวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) หรือ เป็นวงจรกรองเชิงเลขแบบป้อนกลับ (IIR) ที่หน้าจอจะแสดงดังรูปที่ 11.1 เพื่อให้ผู้ใช้เลือกว่าจะให้เป็นวงจรกรองเชิงเลขแบบใด โดยถ้าผู้ใช้เลือก 1. หมายถึงการเลือกวงจรกรองเชิงเลขแบบไม่ป้อนกลับ เลือก 2. หมายถึงการเลือกวงจรกรองเชิงเลขแบบป้อนกลับ

1. Use FIR Filter.

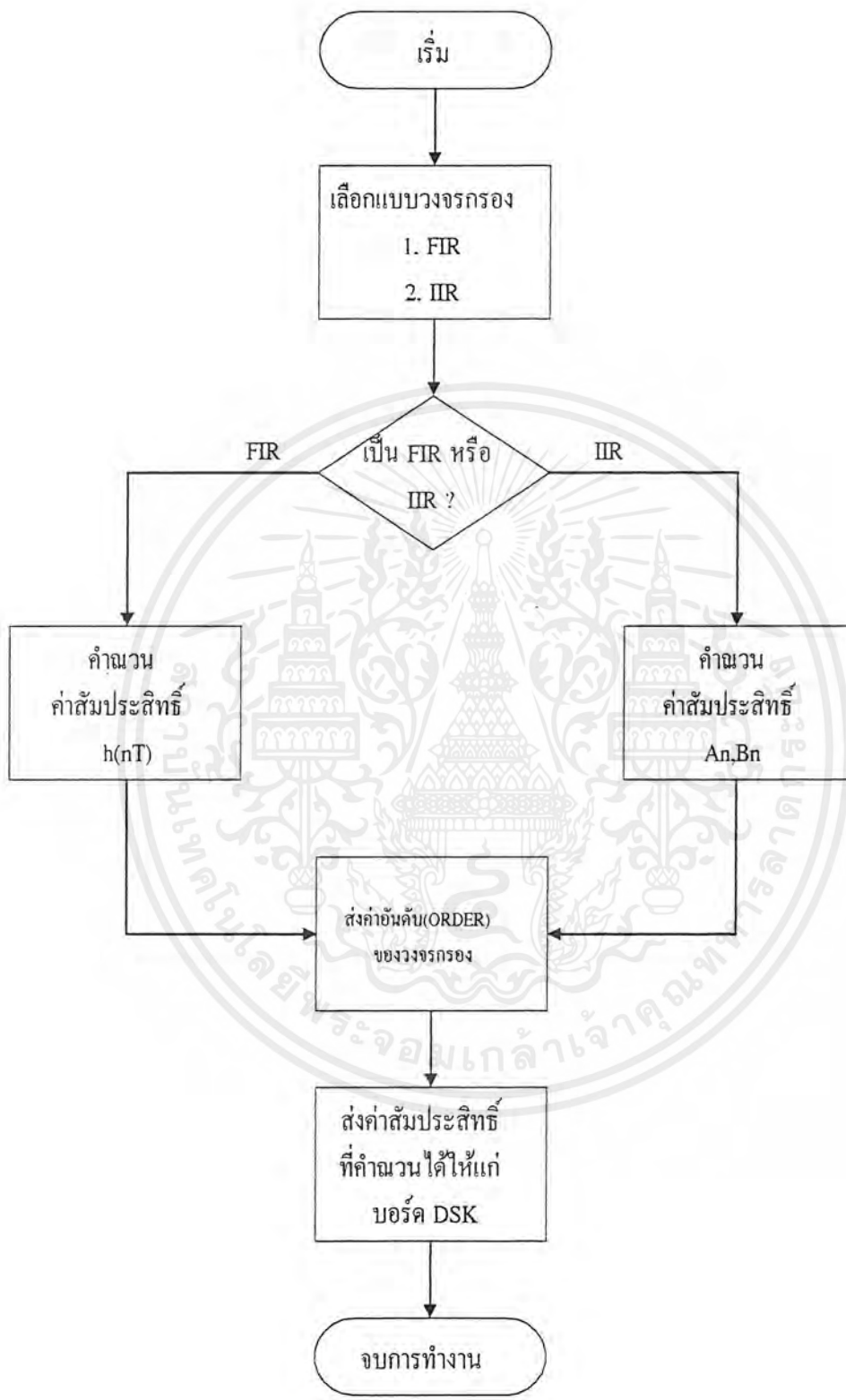
2. Use IIR Filter.

Please select type of filter : _

รูปที่ 11.1 แสดงหน้าจอเมื่อเรียกโปรแกรม FIR-IIR

หลังจากที่ผู้ใช้เลือกแบบของวงจรกรองแล้ว โปรแกรมจะเข้าสู่โปรแกรมการคำนวณค่าสัมประสิทธิ์ โดยถ้าเป็นวงจรกรองเชิงเลขแบบไม่ป้อนกลับ ค่าสัมประสิทธิ์ที่ได้ก็คือค่า $h(nT)$ ถ้าเป็นวงจรกรองเชิงเลขแบบป้อนกลับ ค่าสัมประสิทธิ์ที่ได้ก็คือค่า A_n , B_n (ซึ่งรายละเอียดของการคำนวณค่าสัมประสิทธิ์ของวงจรกรองแต่ละแบบจะได้กล่าวถึงในหัวข้อต่อไป)

ต่อจากนั้น โปรแกรมก็จะทำการส่งค่าสัมประสิทธิ์ที่คำนวณได้ให้แก่บอร์ด DSK โดยก่อนที่จะส่งค่าสัมประสิทธิ์ โปรแกรมจะทำการส่งค่าอันดับ (Order) ของวงจรกรองก่อน ซึ่งค่าอันดับของวงจรกรองนี้อาจจะเป็นค่าที่ได้รับมาจากผู้ใช้ หรือ เป็นค่าอันดับที่คำนวณได้ในขั้นตอนของการคำนวณค่าสัมประสิทธิ์ของวงจรกรองนั้น ขึ้นอยู่กับแบบของวงจรกรอง



รูปที่ 11.2 แสดงโฟลชาร์ทการทำงานของโปรแกรม FIR-IIR

11.1.1 โปรแกรมคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR)

หลังจากที่ผู้ใช้ได้เลือกแบบวงจรกรองเป็นแบบวงจรกรองแบบไม่ป้อนกลับแล้วโปรแกรมจะเข้าสู่โปรแกรมย่อยในการคำนวณค่าสัมประสิทธิ์ เนื่องจากวงจรกรองเชิงเลขแบบไม่ป้อนกลับที่ใช้ในโครงงานนี้ออกแบบโดยวิธีการวินโดว์ (Windows) และใช้โครงสร้างแบบโดยตรง (Direct Form) ค่าสัมประสิทธิ์ที่คำนวณได้คือค่า $h(nT)$ ตั้งแต่ 0 ถึง $N-1$ (N คือค่าอันดับของวงจรกรอง)

โพลซาร์ทการทำงานของโปรแกรมคำนวณค่าสัมประสิทธิ์ของวงจรกรองแบบไม่ป้อนกลับแสดงได้ดังรูปที่ 11.3 สามารถอธิบายการทำงานได้ดังนี้

1.เมื่อผู้ใช้ได้เลือกแบบของวงจรกรองเป็นแบบวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) แล้วโปรแกรมจะเข้าสู่ โปรแกรมการคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบไม่ป้อนกลับ ซึ่งจะทำการถามแบบของวงจรกรองว่าเป็นแบบ (1) ความถี่ต่ำผ่าน (2) ความถี่สูงผ่าน (3) แถบความถี่ผ่าน หรือ (4) กำจัดแถบความถี่ จะมีหน้าจอแสดงดังรูปที่ 11.4

What type of filter ?

1.Low pass

2.High pass

3.Band pass

4.Band stop

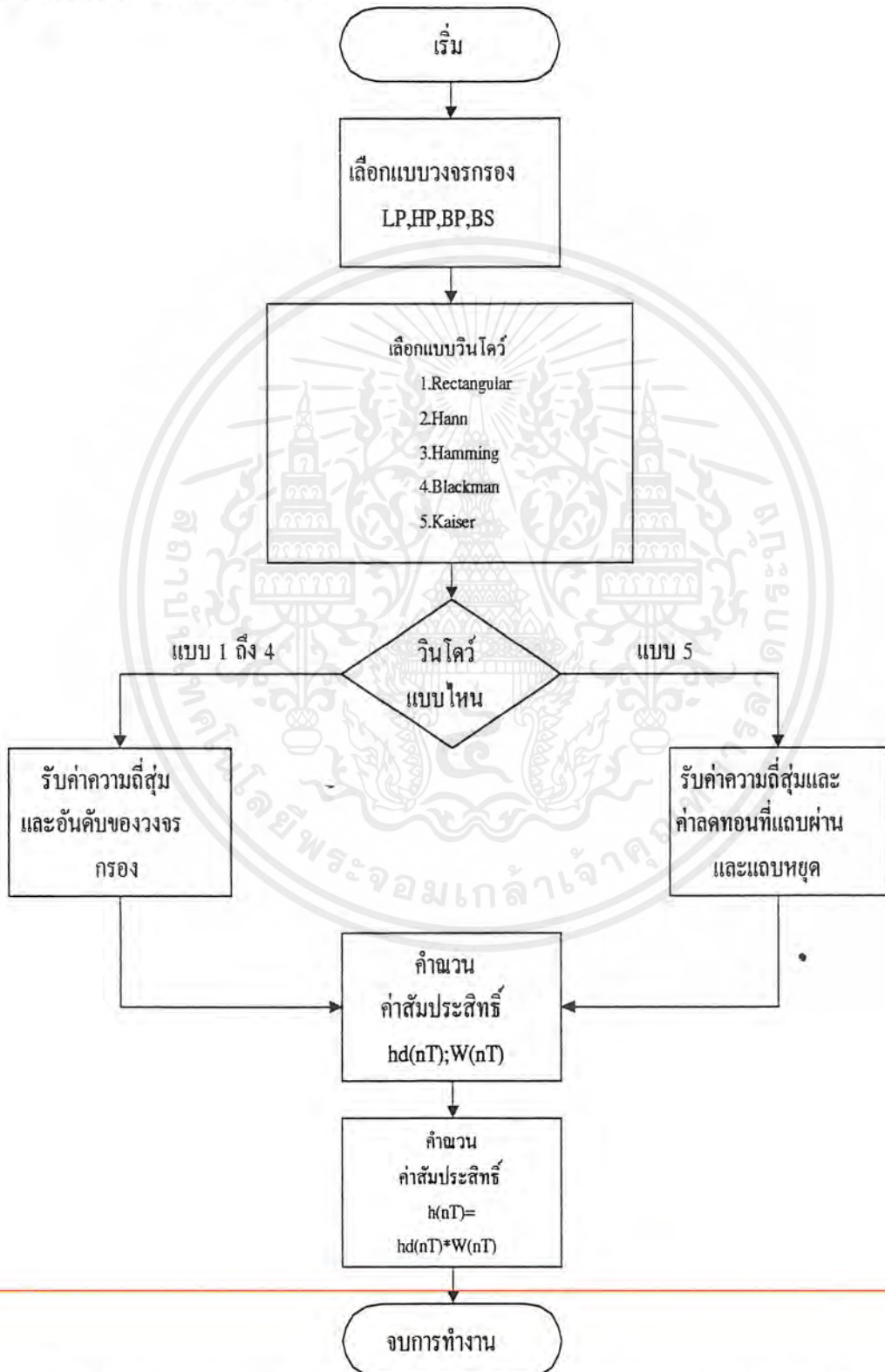
Enter type of filter : _

รูปที่ 11.4 แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบไม่ป้อนกลับ

2. เมื่อผู้ใช้เลือกแบบของวงจรกรองแล้ว (แบบ 1 ,2 ,3 หรือ 4) โปรแกรมจะทำการถามคุณสมบัติความถี่ของวงจรกรองที่ผู้ใช้ต้องการ โดยถ้าเป็นแบบ 1 หรือ 2 โปรแกรมจะให้ผู้ใช้งานป้อนความถี่แถบผ่าน (Pass band frequency : F_p) และความถี่แถบหยุด (Stop band frequency : F_s)

ถ้าเป็นแบบ 3 หรือ 4 โปรแกรมจะให้ผู้ใช้งานป้อนความถี่แถบผ่านต่ำ (Lower Pass band frequency : F_{p1}) ความถี่แถบผ่านสูง (Higher Pass band frequency : F_{p2}) ความถี่แถบหยุดต่ำ

(Lower Stop band frequency : F_{s1}) และความถี่แถบหยุดสูง (Higher Stop band frequency : F_{s2}) ดังแสดงในรูปที่ 11.5 และ 11.6 ตามลำดับ



เอกสารนี้รูปที่ 11.3 แสดงโฟลชาร์ทการทำงานของโปรแกรมส่วนการคำนวณค่าสัมประสิทธิ์ของ FIR การคำนวณค่าสัมประสิทธิ์ของ FIR นั้นมีขั้นตอนการคำนวณที่ซับซ้อนและต้องอาศัยความรู้ทางคณิตศาสตร์อย่างมากในการคำนวณค่าสัมประสิทธิ์ของ FIR ซึ่งถ้าหากเราไม่เข้าใจขั้นตอนการคำนวณค่าสัมประสิทธิ์ของ FIR ก็จะทำให้การคำนวณค่าสัมประสิทธิ์ของ FIR ไม่ถูกต้องและอาจทำให้เกิดข้อผิดพลาดในการใช้งานของ FIR ได้

What type of filter ?

- 1.Low pass
- 2.High pass
- 3.Band pass
- 4.Band stop

Enter type of filter : 1

Pass band frequency : KHz.

Stop band frequency : KHz.

รูปที่ 11.5 แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบความถี่ต่ำผ่าน

What type of filter ?

- 1.Low pass
- 2.High pass
- 3.Band pass
- 4.Band stop

Enter type of filter : 3

Lower Pass band frequency (Fp1) : KHz.

Lower Stop band frequency (Fs1) : KHz.

Higher Pass band frequency (Fp2) : KHz.

Higher Stop band frequency (Fs2) : KHz.

รูปที่ 11.6 แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบแถบความถี่ผ่าน

โดยค่าคุณสมบัติความถี่ต่าง ๆ ที่ผู้ใช้เลือกโปรแกรมจะนำมาคำนวณค่า $h_d(nT)$ ดังสมการต่อไปนี้

แบบที่ 1 วงจรกรองแบบความถี่ต่ำผ่าน

$$h_L(nT) = \frac{\sin(2\pi n f_c / F)}{n\pi}$$

$$h_L(0) = \frac{2fc}{F}$$

$$fc = \frac{1}{2}(Fp + Fs) \quad \Delta F = Fs - Fp \quad 11.1$$

แบบที่ 2 วงจรกรองแบบความถี่สูงผ่าน

$$h_H(nT) = - \frac{\sin(2\pi nfc / F)}{n\pi}$$

$$h_H(0) = 1 - \frac{2fc}{F}$$

$$fc = \frac{1}{2}(Fp + Fs) \quad \Delta F = Fp - Fs \quad 11.2$$

แบบที่ 3 วงจรกรองแบบแถบความถี่ผ่าน

$$h_{BP}(nT) = \frac{1}{n\pi} [\sin(2\pi nfc2 / F) - \sin(2\pi nfc1 / F)]$$

$$h_{BP}(0) = \frac{2}{F}(fc2 - fc1)$$

$$\Delta F_l = Fp1 - Fs1 \quad \Delta F_b = Fs2 - Fp2$$

$$\Delta F = \min[\Delta F_l, \Delta F_b]$$

$$fc1 = Fp1 - \frac{\Delta F}{2} \quad fc2 = Fp2 + \frac{\Delta F}{2} \quad 11.3$$

แบบที่ 4 วงจรกรองแบบก้ำจัดแถบความถี่

$$h_{BS}(nT) = \frac{1}{n\pi} [\sin(2\pi nfc1 / F) - \sin(2\pi nfc2 / F)]$$

$$h_{BS}(0) = \frac{2}{F}(fc1 - fc2) + 1$$

$$\Delta F_l = Fs1 - Fp1 \quad \Delta F_b = Fp2 - Fs2$$

$$\Delta F = \min[\Delta F_l, \Delta F_b]$$

$$fc1 = Fp1 + \frac{\Delta F}{2} \quad fc2 = Fp2 - \frac{\Delta F}{2} \quad 11.4$$

โดยที่ F คือค่าความถี่ศูนย์กลาง

-ในกรณีที่ป็นวงจรกรองผ่านความถี่ต่ำ

$$Fp < Fs, F > 2Fs$$

-ในกรณีที่ป็นวงจรกรองผ่านความถี่สูง

$$Fp > Fs, F > 2Fp$$

-ในกรณีที่ป็นวงจรกรองแถบผ่านความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$F_{s1} < F_{p1} < F_{p2} < F_{s2}, F > 2F_{s2}$$

-ในกรณีที่ป็นวงจรกรองกำจัดแถบความถี่

$$F_{p1} < F_{s1} < F_{s2} < F_{p2}, F > 2F_{p2}$$

3. หลังจากที่ใช้ผู้ทำการป้อนคุณสมบัติความถี่ของวงจรกรองที่ต้องการแล้วโปรแกรมจะให้ผู้เลือกใช้แบบของวินโดว์ที่ใช้ในการออกแบบวงจรกรอง ซึ่งมีอยู่ 5 แบบ ได้แก่

1. วินโดว์สี่เหลี่ยม (Rectangular Window)
2. วินโดว์แบบฮาน (Hann Window)
3. วินโดว์แบบแฮมมิง (Hamming Window)
4. วินโดว์แบบแบล็กแมน (Blackman Window)
5. วินโดว์แบบไคเซอร์ (Kaiser Window)

ซึ่งมีหน้าจอแสดงดังในรูปที่ 11.7

What type of Window ?

1. Rectangular Window
2. Hann Window
3. Hamming Window
4. Blackman Window
5. Kaiser Window

Enter type of Window :

รูปที่ 11.7 แสดงหน้าจอให้เลือกแบบวินโดว์

โดยการคำนวณค่าวินโดว์ฟังก์ชันแต่ละแบบสามารถหาได้ต่อไปนี้

- วินโดว์สี่เหลี่ยม (Rectangular Window)

$$W_R(nT) = 1 \quad ; |n| \leq \frac{N-1}{2} \quad 11.5$$

- วินโดว์แบบฮาน

$$W_H(nT) = 0.5 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) \quad ; |n| \leq \frac{N-1}{2} \quad 11.6$$

- วินโดว์แบบแฮมมิง

$$W_H(nT) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N-1}\right) \quad ; |n| \leq \frac{N-1}{2} \quad 11.7$$

- วินโดว์แบบแบล็กแมน

$$W_B(nT) = 0.42 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right); |n| \leq \frac{N-1}{2} \quad 11.8$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย N คือ อันดับของวงจรกรองเชิงเลข
 ซึ่งเมื่อผู้ใช้เลือกแบบวินโดว์ ทั้ง 4 แบบ นี้ โปรแกรมจะให้ผู้ใส่ป้อนค่าความถี่สุ่ม (Sampling Frequency : F) และอันดับของวงจรกรองที่ผู้ใช้ต้องการ โดยมีหน้าจอแสดงดังรูปที่ 11.8

You choose Rectangular Window.

Enter the variable

Sampling frequency : KHz.

Order of filter (2 - 100) :

รูปที่ 11.8 แสดงหน้าจอเมื่อผู้ใช้เลือกวินโดว์สี่เหลี่ยม

ในกรณีที่ผู้ใช้เลือกวินโดว์แบบไคเซอร์ โปรแกรมจะทำการถามค่าความถี่สุ่ม (Sampling Frequency) ค่าลดทอนที่แถบความถี่ผ่าน (Pass band attenuate : Ap) และ ค่าลดทอนที่แถบความถี่หยุด (Stop band attenuate : As) ดังแสดงในรูปที่ 11.9

You choose Kaiser Window.

Enter the variable

Sampling frequency : KHz.

Pass band attenuate : dB.

Stop band attenuate : dB.

รูปที่ 11.9 แสดงหน้าจอเมื่อผู้ใช้เลือกวินโดว์แบบไคเซอร์

ซึ่งค่าทั้งสามค่านี้จะนำมาใช้ในการคำนวณหาอันดับของวงจรกรองที่เหมาะสม และ ค่าวินโดว์ฟังก์ชัน โดยมีขั้นตอนดังนี้

- การคำนวณวินโดว์แบบไคเซอร์

$$3.1 \delta_s = 10^{-0.05A_s}$$

$$\delta_p = \frac{(10^{0.05A_p} - 1)}{(10^{0.05A_p} + 1)}$$

11.9

3.2 เลือก δ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\delta = \min (\delta_s, \delta_p) \quad 11.10$$

3.3 หา Aa โดย

$$Aa = -20\log\delta \quad 11.11$$

3.4 หา α โดยหาได้จาก

$$\alpha = \begin{cases} 0 & Aa \leq 21 \\ 0.5842(Aa - 21)^{0.4} + 0.07886(Aa - 21) & 21 < Aa \leq 50 \\ 0.1102(Aa - 8.7) & Aa > 50 \end{cases} \quad 11.12$$

3.5 หา D โดยหาได้จาก

$$D = \begin{cases} 0.922 & Aa \leq 21 \\ \frac{Aa - 7.95}{14.36} & Aa > 21 \end{cases} \quad 11.13$$

3.6 คำนวณหาค่าอันดับของวงจรกรอง (N) ที่เป็นเลขคี่ที่ต่ำที่สุดที่ทำให้ได้เงื่อนไข

$$N \geq \frac{FD}{\Delta F} + 1 \quad 11.14$$

3.7 คำนวณหาค่า $W_K(nT)$ จาก

$$\begin{aligned} W_K(nT) &= I_0(\beta) / I_0(\alpha) && \text{เมื่อ } |n| \leq (N-1)/2 \\ &= 0 && \text{กรณี } n \text{ ค่าอื่น} \end{aligned} \quad 11.15$$

โดยที่ α เป็นพารามิเตอร์อิสระที่สัมพันธ์กับพารามิเตอร์ β โดย

$$\beta = \alpha \cdot \sqrt{1 - (2n / (N-1))^2} \quad 11.16$$

และ $I_0(X)$ เป็น เบสเซลฟังก์ชันชนิดที่หนึ่งอันดับศูนย์ (ZERO th -ORDER Bessel Function of the first kind) ที่สามารถคำนวณได้โดยใช้ อนุกรมกำลัง คือ

$$I_0(X) = 1 + \sum_{k=1}^{\infty} \{(1/k!)(x/2)^k\}^2 \quad 11.17$$

ในการประยุกต์ใช้งานทั่วไป สำหรับความละเอียดในการคำนวณตามสมการข้างต้น อาจใช้จำนวนพจน์ระหว่าง 15 ถึง 25 พจน์ก็เพียงพอ

4. หลังจากนั้นโปรแกรมคำนวณค่าสัมประสิทธิ์ $h(nT)$ ที่ใช้ในวงจรกรอง โดยคำนวณได้

จาก

$$h(nT) = h_d(nT) * W(nT) \quad 11.18$$

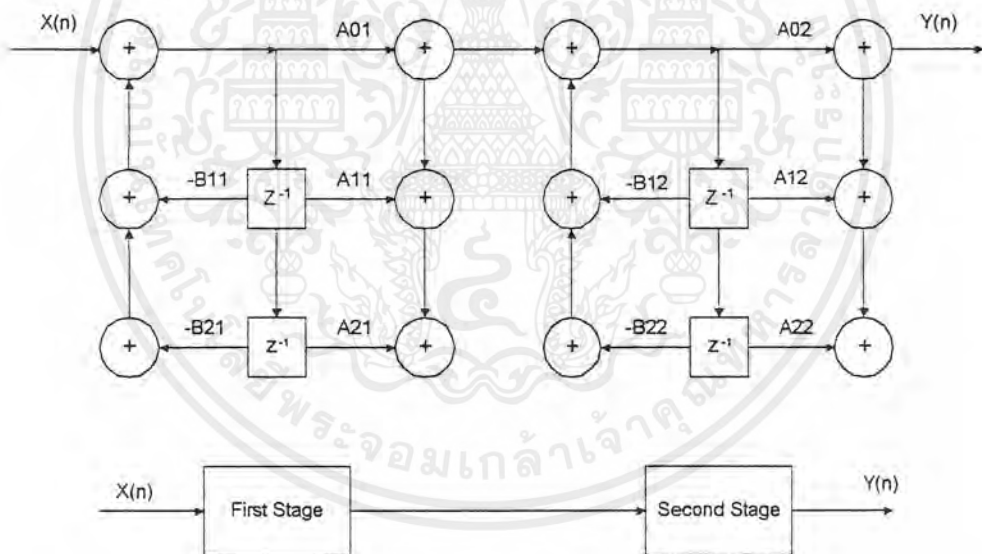
5. หลังจากที่ได้ค่าอันดับและค่าสัมประสิทธิ์ ($h(nT)$) ของวงจรกรองแล้วก็จะเข้าสู่

โปรแกรมส่วนส่งค่าสัมประสิทธิ์ และ อันดับของวงจรกรอง ให้แก่บอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.1.2 โปรแกรมคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับ (IIR)

เมื่อผู้ใช้ได้เลือกวงจรกรองเป็นแบบวงจรกรองแบบป้อนกลับ (IIR) แล้วโปรแกรมจะเข้าสู่โปรแกรมการคำนวณค่าสัมประสิทธิ์ ซึ่งวงจรกรองแบบป้อนกลับที่ใช้ในโครงงานนี้ใช้วงจรกรองแบบบัตเตอร์เวอร์ธในการออกแบบและใช้โครงสร้างแบบโดยตรง 2 (Direct Form II) อันดับสอง มาแสดงดังแสดงในรูปที่ 11.10 ดังนั้นค่าสัมประสิทธิ์ที่คำนวณได้คือ A_n และ B_n ของแต่ละสเตจ โดยหลังจากที่ผู้ใช้เลือกวงจรกรองแบบป้อนกลับและเลือกกว่าจะให้ป้อนวงจรกรองแบบ ความถี่ต่ำผ่าน ความถี่สูงผ่าน แถบความถี่ผ่าน หรือ กำจัดแถบความถี่ โปรแกรมจะถามคุณสมบัติทางความถี่ของวงจรกรองแต่ละแบบเช่นเดียวกับวงจรกรองแบบไม่ป้อนกลับ รวมถึงค่าความถี่สุ่ม ค่าลดทอนที่แถบความถี่ผ่าน และค่าลดทอนที่แถบกำจัดความถี่ มีหน้าจอแสดงในรูปที่ 11.11



รูปที่ 11.10 แสดงวงจรกรองป้อนกลับเชิงเลขโครงสร้างโดยตรง 2 อันดับสอง คาสเคดกัน

ในการคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับ จะใช้ค่าคุณสมบัติความถี่ของวงจรกรอง ค่าความถี่สุ่ม ค่าลดทอนที่แถบความถี่ผ่าน และ ค่าลดทอนที่แถบกำจัดความถี่ ในการคำนวณหาค่าสัมประสิทธิ์ A_n และ B_n รวมถึงค่าอันดับของวงจรกรองที่เหมาะสม โดยใช้วงจรกรองบัตเตอร์เวอร์ธเชิงเลข โปรแกรมส่วนคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับมีโฟลชาร์ตดังแสดงในรูปที่ 11.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

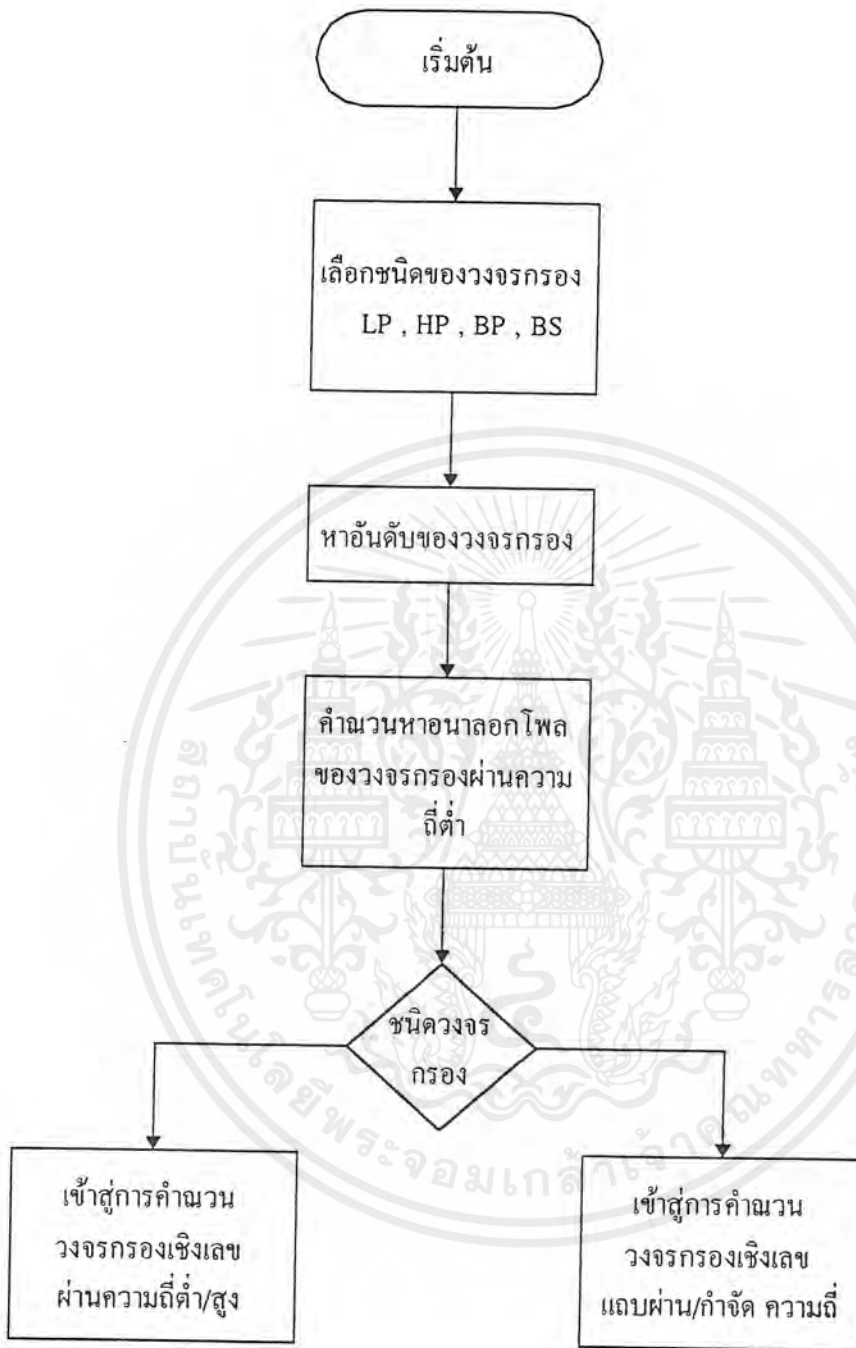
What type of filter ?

- 1.Low pass
- 2.High pass
- 3.Band pass
- 4.Band stop

Enter type of filter : 1

Pass band frequency	:	KHz.
Stop band frequency	:	KHz.
Sampling frequency	:	KHz.
Pass band attenuate	:	dB.
Stop band attenuate	:	dB.

รูปที่ 11.11 แสดงหน้าจอเมื่อผู้ใช้เลือกวงจรกรองแบบป้อนกลับความถี่ต่ำผ่าน



รูปที่ 11.12 แสดงโฟลชาร์ทการทำงานของโปรแกรมส่วนการคำนวณค่าสัมประสิทธิ์ของ IIR

จากรูปที่ 11.12 สามารถการทำงานของโปรแกรมส่วนการคำนวณค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หลังจากที่ได้รับค่าคุณสมบัติทางความถี่ของวงจรกรองที่ผู้ใช้งานต้องการแล้วโปรแกรมจะทำการคำนวณหาค่า $hd(nT)$ เช่นเดียวกับวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (สมการ 11.1 ถึง 11.4)

2. หาอันดับที่เหมาะสมของวงจรกรองได้จาก

$$N > \frac{\log A}{\log 1/K} \quad 11.19$$

และ K ของวงจรกรองแต่ละแบบหาได้ดังนี้

2.1 วงจรกรองผ่านความถี่ต่ำ

$$K = \frac{\tan(\Pi F_p / F)}{\tan(\Pi F_s / F)} \quad 11.20$$

2.2 วงจรกรองผ่านความถี่สูง

$$K = \frac{\tan(\Pi F_s / F)}{\tan(\Pi F_p / F)} \quad 11.21$$

2.3 วงจรกรองแถบผ่านความถี่

$$\begin{aligned} K &= K_1 && \text{ถ้า } K_C \geq K_B \\ &= K_2 && \text{ถ้า } K_C < K_B \end{aligned} \quad 11.22$$

2.4 วงจรกรองก้ำจืดแถบความถี่

$$\begin{aligned} K &= 1/K_1 && \text{ถ้า } K_C \geq K_B \\ &= 1/K_2 && \text{ถ้า } K_C < K_B \end{aligned} \quad 11.23$$

โดย ในข้อ 2.3 และ 2.4 ค่าต่าง ๆ จะหาได้จาก

$$K_A = \tan(\Pi F_{p2}/F) - \tan(\Pi F_{p1}/F) \quad 11.24$$

$$K_B = \tan(\Pi F_{p1}/F) - \tan(\Pi F_{p2}/F) \quad 11.25$$

$$K_C = \tan(\Pi F_{s1}/F) \tan(\Pi F_{s2}/F) \quad 11.26$$

$$K_1 = \frac{K_A \tan(\Pi F_{s1} / F)}{K_B - \tan^2(\Pi F_{s1} / F)} \quad 11.27$$

$$K_2 = \frac{K_A \tan(\Pi F_{s2} / F)}{\tan^2(\Pi F_{s2} / F) - K_B} \quad 11.28$$

3. คำนวณหาขนาดลอกโพลของวงจรกรองผ่านความถี่ต่ำ (S_p) ได้จาก

$$\begin{aligned} S_{pk} &= -\sin\left(\frac{2k-1}{2N}\right)\Pi + j\cos\left(\frac{2k-1}{2N}\right)\Pi \\ &= a_{pk} + jb_{pk} \end{aligned}$$

$$k = 1, 2, \dots, (N+1)/2$$

กรณีอันดับเลขคี่

$$k = 1, 2, \dots, N/2$$

กรณีอันดับเลขคู่

11.29

และอนาล็อกซีโร่ของวงจรกรองผ่านความถี่ต่ำ (S_z) จะมีค่า

$$S_z = 0$$

4. หลังจากที่ได้ค่าอนาล็อกโพลและซีโร่ของวงจรกรองผ่านความถี่ต่ำแล้วจะเข้าสู่การคำนวณของวงจรกรองเชิงเลขแต่ละแบบต่อไป

หลังจากที่ได้ค่าอนาล็อกโพลและซีโร่ของวงจรกรองผ่านความถี่ต่ำแล้วขั้นตอนต่อไปก็คือการแปลงค่าอนาล็อกโพลและซีโร่ไปเป็นโพลและซีโร่เชิงเลขเพื่อใช้ในการหาค่าสัมประสิทธิ์ของวงจรกรองแต่ละแบบต่อไป ในขั้นตอนนี้จะสามารถแยกการคำนวณหาค่าสัมประสิทธิ์ได้เป็น 2 กรณี คือ

5.1 การหาค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ / สูง

5.2 การหาค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแถบผ่านความถี่ / กำจัดแถบความถี่

5.1 การหาค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ / สูง จะมีขั้นตอนดังนี้

5.1.1 แปลงค่าอนาล็อกโพลและซีโร่ไปเป็น โพลและซีโร่เชิงเลขของวงจรกรองเชิงเลขผ่านความถี่ต่ำ / สูง สามารถหาได้จาก

$$Z_{zk} = \frac{1 - b_{zk}^2 \alpha^2}{1 + b_{zk}^2 \alpha^2} + j \frac{2b_{zk}\alpha}{1 + b_{zk}^2 \alpha^2} \quad 11.30$$

โดย ในกรณีที่ป็นวงจรกรองเชิงเลขผ่านความถี่ต่ำ

$$\alpha = (10^{0.1Ap} - 1)^{-1/2N} \tan(\prod F_p/F) \quad 11.31$$

ในกรณีที่ป็นวงจรกรองเชิงเลขผ่านความถี่สูง

$$\alpha = (10^{0.1Ap} - 1)^{1/2N} \tan(\prod F_p/F) \quad 11.32$$

5.1.2 หลังจากที่ได้ค่าโพลและซีโร่เชิงเลขแล้ว จะสามารถหาค่าสัมประสิทธิ์ของวงจรกรอง

(A_n และ B_n) ได้โดยจะแบ่งเป็น 2 กรณี คือ กรณีที่อันดับของวงจรกรองเป็นเลขคี่ และ กรณีที่อันดับของวงจรกรองเป็นเลขคู่

ในกรณีที่อันดับของวงจรกรองเป็นเลขคี่ โครงสร้างของวงจรกรองที่จะนำมาคาดเคลื่อนจะประกอบด้วยวงจรกรองอันดับแรก (First Order Section) 1 ชุด คาสเคดกับวงจรกรองอันดับสอง

(Second Order Section) อีก (N - 1) / 2 เซต โดย N คืออันดับของวงจรกรองที่เป็นเลขคี่ ดังนั้นในการหาค่าสัมประสิทธิ์ของวงจรกรองในกรณีที่อันดับของวงจรกรองเป็นเลขคี่ จะต้องทำการหาค่าสัมประสิทธิ์ของวงจรกรองอันดับแรก ที่จะนำมาคูณกับวงจรกรองอันดับสองจนได้อันดับครบตามที่ต้องการซะก่อน

ค่าสัมประสิทธิ์ของวงจรกรองอันดับแรกสามารถหาค่าได้ดังต่อไปนี้

- วงจรกรองเชิงเลขผ่านความถี่ต่ำ

$$A_0 = 1 ; A_1 = 1 ; A_2 = 0$$

$$B_1 = -Z_p$$

$$\text{โดย } Z_p = \frac{1 - |a_p| \alpha}{1 + |a_p| \alpha} \quad 11.33$$

- วงจรกรองเชิงเลขผ่านความถี่สูง

$$A_0 = 1 ; A_1 = -1 ; A_2 = 0$$

$$B_1 = \frac{\alpha - |a_p|}{\alpha + |a_p|} \quad 11.34$$

หลังจากที่ได้ค่าสัมประสิทธิ์ของวงจรกรองอันดับแรกแล้ว ขั้นตอนต่อไปก็คือการหาค่าสัมประสิทธิ์ของวงจรกรองอันดับสอง ซึ่งทั้งกรณีที่อันดับเป็นเลขคู่ และ เลขคี่ ค่าสัมประสิทธิ์ของวงจรกรองอันดับสองสามารถหาได้จากสมการเดียวกันดังนี้

$$A_{1k} = -2\text{Re}[Z_{zk}]$$

$$A_{2k} = \text{Re}[Z_{zk}]^2 + \text{Im}[Z_{zk}]^2$$

$$B_{1k} = -2\text{Re}[Z_{pk}]$$

$$B_{2k} = \text{Re}[Z_{pk}]^2 + \text{Im}[Z_{pk}]^2 \quad 11.35$$

5.2 การหาค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแถบผ่านความถี่ / กำจัดแถบความถี่ จะมีขั้นตอนดังนี้

5.2.1 หาค่าอานาล็อกโพลและซีโรว์ของวงจรกรองแถบผ่านความถี่ / กำจัดแถบความถี่จากอานาล็อกโพลและซีโรว์ของวงจรกรองผ่านความถี่ต่ำ โดยหาได้จาก

$$W_u = \tan(\prod F_{p2}/F) \quad 11.36$$

$$W_l = \tan(\prod F_{p1}/F) \quad 11.37$$

$$\Delta W = W_u - W_l \quad 11.38$$

$$W_o = W_u W_l \quad 11.39$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$C1 = b_p^2 + \left[\frac{4W_0}{(\Delta W \beta)^2} \right] - a_p^2 \quad 11.40$$

โดย ในกรณีวงจรกรองแถบผ่านความถี่

$$\beta = (10^{0.1Ap} - 1)^{-1/2N} \quad 11.41$$

ในกรณีวงจรกรองก้ำจัดแถบความถี่

$$\beta = (10^{0.1Ap} - 1)^{1/2N} \quad 11.42$$

$$C2 = [C1^2 + (2a_p b_p)^2]^{0.5} \quad 11.43$$

$$C3 = \left(\frac{C2 - C1}{2} \right)^{0.5} \quad 11.44$$

$$C4 = \left(\frac{C2 + C1}{2} \right)^{0.5} \quad 11.45$$

เมื่อได้ค่า C3 และ C4 แล้วเราสามารถหาค่าพอลิและซีโรของวงจรกรองแถบผ่านความถี่และวงจรกรองก้ำจัดแถบความถี่ได้ดังนี้

พอลิและซีโรของวงจรกรองเชิงเลขแถบผ่านความถี่หาได้จาก

$$S_{bp1} = \frac{\Delta W \beta}{2} [(-|a_p| - C3) \pm j(b_p + C4)] = a_{bp1} + jb_{bp1} \quad 11.46$$

$$S_{bp2} = \frac{\Delta W \beta}{2} [(-|a_p| + C3) \pm j(b_p - C4)] = a_{bp2} + jb_{bp2} \quad 11.47$$

พอลิและซีโรของวงจรกรองเชิงเลขก้ำจัดแถบความถี่หาได้จาก

$$S_b = \frac{-|a_p| \pm j b_p}{a_p^2 + b_p^2} = \frac{1}{-a_p + jb_p} \quad 11.48$$

$$S_{bz} = \pm j(W_0)^{0.5} \quad 11.49$$

5.2.2 หาค่าพอลิและซีโรเชิงเลขของวงจรกรองแถบผ่านความถี่โดย

$$Z_{pk} = \frac{1 - (a_{bpk}^2 + b_{bpk}^2)}{D_{bpk}} + j \frac{2b_{bpk}}{D_{bpk}} \quad 11.50$$

เป็นค่าพอลิและซีโรเชิงเลขที่แปลงมาจากพอลิและซีโร โดย

$$D_{bpk} = 1 + 2|a_{bpk}| + a_{bpk}^2 + b_{bpk}^2 \quad 11.51$$

5.2.3 หาค่าสัมประสิทธิ์ของวงจรกรองโดยสามารถแบ่งได้เป็น 2 กรณีเช่นเดียวกับการหาค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ / สูง คือ กรณีที่อันดับของวงจรกรองเป็นเลขคี่และกรณีที่อันดับของวงจรกรองเป็นเลขคู่ ในกรณีที่อันดับของวงจรกรองเป็นเลขคี่จะต้องหาค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคี่ (Odd Second-Order Section) แล้วจึง

หาค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคู่ที่จะนำมาคาดเคลื่อนจนครบ
อันดับของวงจรกรองที่ต้องการ

ค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคี่จะแบ่งได้เป็น 2 ส่วน คือ ค่า
สัมประสิทธิ์ของวงจรกรองแถบผ่านความถี่ และ วงจรกรองก้ำจัดแถบความถี่

5.2.3.1 ค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคี่ของวงจรกรองแถบ
ผ่านความถี่ ซึ่งจะมีค่าโพลและซีโรคือ

$$S_{bpk} = \frac{\Delta W \beta}{2} [\sigma_0 \pm j C_1^{0.5}] = a_{bpk} \pm j b_{bpk} \quad 11.52$$

$$\text{โดย } \sigma_0 = \frac{-2q^{0.25} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh[(2m+1)\Lambda]}{1 + 2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cosh(2m\Lambda)} \quad 11.53$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \quad 11.54$$

$$q_0 = \frac{1 - (1 - K^2)^{0.25}}{2[1 + (1 - K^2)^{0.25}]} \quad 11.55$$

หลังจากที่ได้ค่า S_{bpk} แล้วสามารถนำค่า S_{bpk} นี้มาหาค่าสัมประสิทธิ์ B1 , B2 ได้ตามสม
การที่ 11.35 โดยค่า $A_0 = 1$, $A_1 = 0$ และ $A_2 = -1$ ตามลำดับ

5.2.3.2 ค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคี่ของวงจรกรองก้ำจัด
แถบความถี่ ซึ่งจะมีค่าโพลและซีโรคือ

$$S_{bpk} = \frac{\Delta W \beta}{2} [-\frac{1}{a_p} + j C_1^{0.5}] \quad 11.56$$

หลังจากนั้นจะได้ค่าโพลเชิงเลขของวงจรกรองก้ำจัดแถบความถี่ของวงจรกรองอันดับสอง
ที่เป็นส่วนของเลขคี่ คือ

$$Z_{p1} = \frac{1 - |S_{bp1}|}{1 + |S_{bp1}|} \quad 11.57$$

$$Z_{p2} = \frac{1 - |S_{bp2}|}{1 + |S_{bp2}|} \quad 11.58$$

และค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขก้ำจัดแถบความถี่ส่วนของเลขคี่จะหาได้จาก

$$B_{1k} = -(Z_{p1} + Z_{p2}) \quad 11.59$$

$$B_{2k} = Z_{p1} Z_{p2} \quad 11.60$$

โดย $A_0 = 1$, $A_1 = A_{1k}$ และ $A_2 = 1$

ในส่วนค่าสัมประสิทธิ์ของวงจรกรองอันดับสองที่เป็นส่วนของเลขคู่สามารถหาได้จากสม

การที่ 11.35 เช่นเดียวกับวงจรกรองเชิงเลขผ่านความถี่ค่า / ดู

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในเนื้อหาและข้อมูล ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากที่ได้ค่าอันดับและค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับ แล้วก็จะเข้าสู่โปรแกรมส่วนส่งค่าสัมประสิทธิ์ และ อันดับของวงจรกรองให้แก่บอร์ด DSK

จากโฟลชาร์ทรูปที่ 11.2 หลังจากที่เราได้ค่าสัมประสิทธิ์ของวงจรกรองไม่ว่าจะเป็นแบบไม่ป้อนกลับหรือป้อนกลับ ขั้นตอนต่อมาคือการส่งค่าสัมประสิทธิ์ที่ได้ให้แก่บอร์ด DSK ซึ่งจะต้องมีการเปลี่ยนรูปแบบของค่าสัมประสิทธิ์ที่จะส่งให้แก่บอร์ด DSK เพราะค่าที่บอร์ด DSK สามารถนำมาค่า

คำนวณในการทำเป็นวงจรกรองเชิงเลขจะอยู่ในรูปเลขฐานสองแบบส่วนเต็มเต็มสอง ซึ่งจะกล่าวถึงในหัวข้อต่อไป

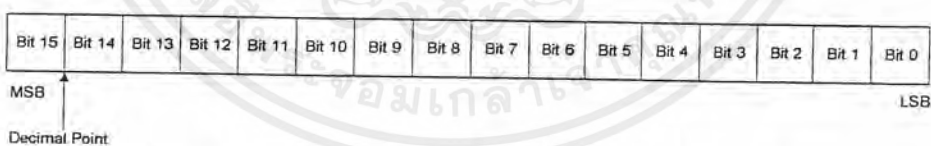
11.2 รูปแบบตัวเลขที่ใช้ในการติดต่อระหว่างบอร์ด กับ IBM PC

ก่อนที่จะเราจะส่งค่าสัมประสิทธิ์ที่เราได้จาก IBM PC ไปให้แก่ บอร์ด DSK ผ่านทางพอร์ตพริเตอร์จะต้องมีการเปลี่ยนค่ารูปแบบตัวเลขของค่าสัมประสิทธิ์ให้อยู่ในรูปเลขฐานสองแบบส่วนเต็มเต็มสอง ที่อยู่ในรูปแบบ Qxx ซึ่งเป็นรูปแบบที่บอร์ด DSK สามารถนำมาใช้ในการคำนวณเป็นวงจรกรองเชิงเลขได้

11.2.1 รูปแบบ Qxx หมายถึง เลขฐานสองแบบส่วนเต็มเต็มสองขนาด 16 บิต ที่มีจุดทศนิยม

(Decimal Point) อยู่ที่ตำแหน่ง XX จากบิตท้ายสำคัญน้อยที่สุด (Least Significant Bit : LSB)

ยกตัวอย่างเช่น รูปแบบ Q15 หมายถึงเลขฐานสองแบบส่วนเต็มเต็มสองขนาด 16 บิต ที่มีจุดทศนิยมอยู่ที่ตำแหน่ง 15 จากบิตท้ายสำคัญน้อยที่สุด ดังแสดงในรูปที่ 11.13



รูปที่ 11.13 แสดงตำแหน่งจุดทศนิยมของเลขฐานสองรูปแบบ Q15

รูปแบบ Qxx เป็นรูปแบบที่บอร์ด DSK มองว่าตัวเลขฐานสองที่อยู่ในรูปแบบนี้ เสมือนเป็นตัวเลขทศนิยม จากรูปที่ 11.13 เป็นรูปแบบ Q15 ที่บอร์ด DSK จะมองว่าข้อมูลขนาด 16 บิต นี้ มีบิตเครื่องหมาย 1 บิต ที่ตำแหน่ง MSB และมีบิตแสดงเลขเศษส่วน (Fractional Bits) 15 บิต ทำให้เราสามารถทำการคำนวณตัวเลขที่อยู่ในรูปทศนิยมได้

11.2.2 การแสดงข้อมูล และการหาจำนวนครั้งของการชีพ

เพื่อให้การติดต่อระหว่าง IBM PC และบอร์ด DSK สามารถรับส่งข้อมูลกันได้อย่างถูกต้องจำเป็นต้องมีการกำหนดรูปแบบข้อมูลที่จะใช้ติดต่อกัน รวมถึงจำนวนในการชีพท์ของรีจิสเตอร์แต่ละตัวของ TMS320C50 เพื่อที่จะทำให้การคำนวณเป็นไปได้อย่างถูกต้อง เพราะการชีพท์ของรีจิสเตอร์ในการคำนวณทำให้เกิดผลจากการ โอเวอร์โฟล (Overflow) ได้

ในการหาตำแหน่งของจุดทศนิยม หรือ ขนาดความยาวของเลขจำนวนเต็ม (Integer Word Lengths) เพื่อกำหนดว่าจะใช้รูปแบบไหน (Qxx) สามารถหาได้จาก ขนาด (Magnitude) หรือ ช่วง (Range) ของสัญญาณ เนื่องจากแต่ละสัญญาณสามารถมีค่าที่แตกต่างกันแต่อยู่ในช่วงของสัญญาณ ดังนั้นเราสามารถรู้ความยาวของขนาดของเลขจำนวนเต็มที่เหมาะสมเพียงขนาดเดียวในการแสดงแต่ละค่าของสัญญาณที่อยู่ในช่วงได้ ยกตัวอย่างเช่น เลขฐานสองขนาด 16 บิต ที่มีบิตเครื่องหมาย (Sign Bit) 1 บิต บิตแสดงเลขจำนวนเต็ม (Integer Bits) 2 บิต และ บิตแสดงเลขเศษส่วน 13 บิต จะสามารถแสดงข้อมูล หรือสัญญาณที่มีช่วงระหว่าง +4 ถึง -4 ได้ ซึ่งหมายความว่าจุดทศนิยมอยู่ที่ตำแหน่งบิตที่ 2 นับจากบิตเครื่องหมาย หรือ รูปแบบ Q12 นั่นเอง การใช้รูปแบบ Qxx นี้ไม่เพียงแต่ป้องกันการ โอเวอร์โฟลเท่านั้นแต่ยังมีการควอนไทซ์ (Quantization) ที่มีค่าค่าอีกด้วย

ขนาดของความยาวของเลขจำนวนเต็มของตัวแปร x [$W_I(x)$] สามารถหาได้จากช่วงของสัญญาณ $R(x)$ ตามสมการ

$$W_I(x) \geq \lceil \log_2 R(x) \rceil \quad 11.61$$

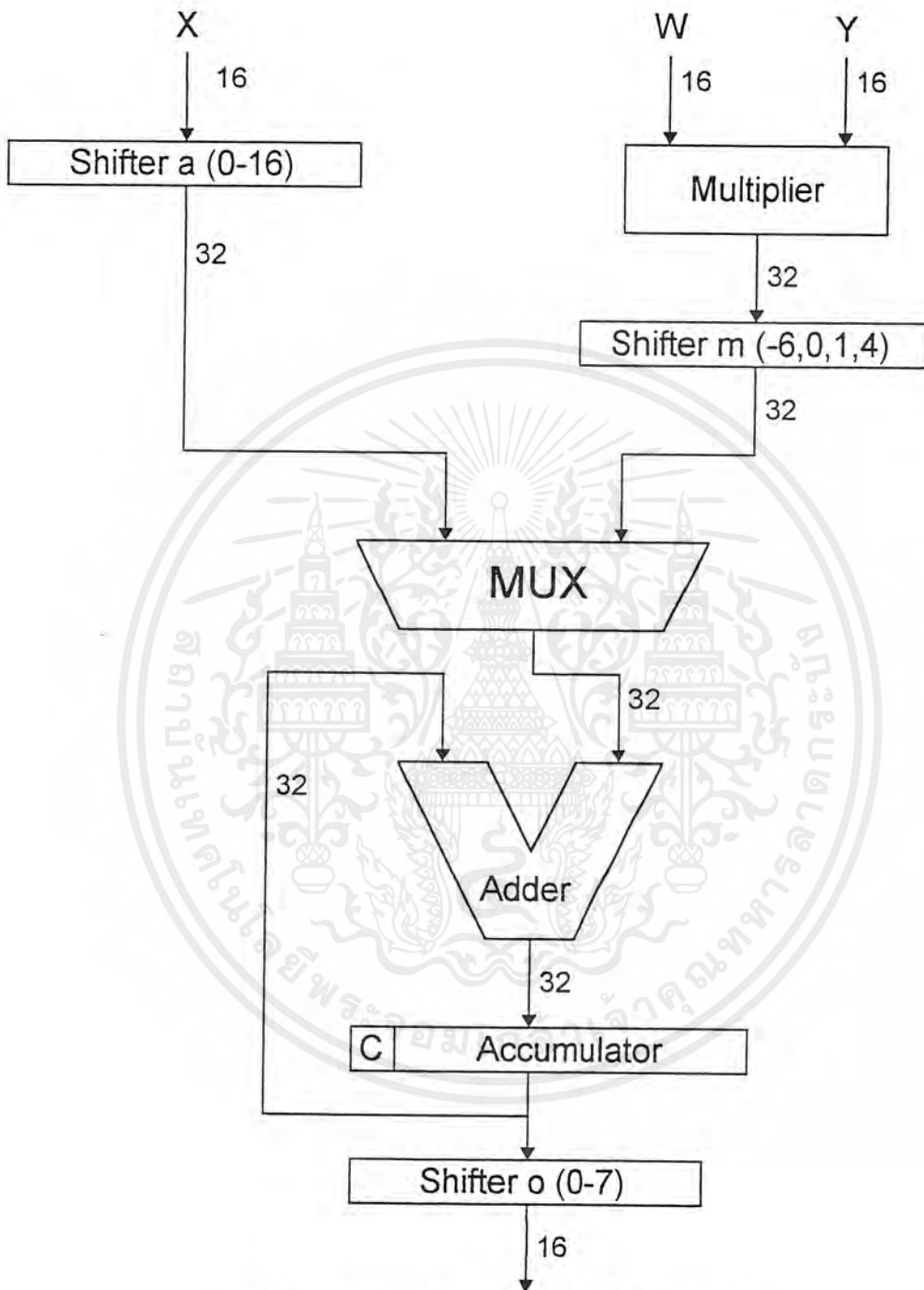
โดยที่ (x) เป็นค่าจำนวนเต็มทีน้อยที่สุดที่มีค่าเท่ากับหรือมากกว่า x สังเกตได้ว่าความยาวของส่วนเลขจำนวนเต็มหาได้จากช่วงของสัญญาณ ไม่ใช่ค่าสัญญาณขณะนั้น สมมุติว่าค่าข้อมูลในรูปแบบโดยตรง (Fixed Point Format) $x_{16}[100]$ คือ 01010.....0 หรือ 5000H ในรูปเลขฐานสิบหก ควรจะถูกมองเป็น 010.10.....0 หรือ 2.5 ในรูปแบบจำนวนอิงดัชนี (Floating Point Format) แต่ตัวเลขที่ ดิจิตอลซิกแนลโปรเซสเซอร์แบบโดยตรง (Fixed-Point Digital Signal Processors) เห็นจะเป็นแค่ตัวเลข 01010.....0 เท่านั้น ดังนั้นผู้ใช้งานจำเป็นต้องทราบขนาดความยาวของเลขจำนวนเต็มที่อยู่ในหน่วยความจำว่าแทนด้วยสัญญาณที่แท้จริงค่าใด

ถึงแม้ว่ารูปแบบ Qxx นี้ทำให้สามารถแทนค่าข้อมูลต่าง ๆ ได้ง่าย แต่ในการบวก หรือ ลบ จำนวนสองจำนวนที่มีขนาดความยาวของขนาดจำนวนเต็มแตกต่างกัน จำเป็นต้องมีการกำหนดจุดทศนิยมที่แน่นอนก่อน ตัวอย่างเช่น ตัวแปร $x(n)$ มีขนาดความยาวของขนาดของเลขจำนวนเต็มเท่ากับ 2 ไม่สามารถบวกกับตัวแปร $y(n)$ ที่มีขนาดความยาวของขนาดของเลขจำนวนเต็มเท่ากับ 1 ได้ อย่างไรก็ตาม ความยาวของขนาดของเลขจำนวนเต็มสามารถเปลี่ยนแปลงได้โดยใช้การเลื่อน (Shift) ทางคณิตศาสตร์ การชีพท์ไปทางขวาจำนวน n บิต จะมีค่าเท่ากับเพิ่มค่าความยาวของ

ขนาดของเลขจำนวนเต็มไปอีก n ในกรณีตัวอย่างข้างต้น ค่าตัวแปร $y(n)$ ควรจะมีการเลื่อนบิตไปทางขวา 1 บิต ซะก่อนที่จะทำการบวกกับ $x(n)$ ในกรณีที่เลข 2 จำนวนคูณกันผลการคูณที่ได้จะมีค่าความยาวของขนาดของเลขจำนวนเต็มที่มีค่าเท่ากับ ผลรวมของค่าความยาวของขนาดของเลขจำนวนเต็ม 2 จำนวนที่คูณกัน ทำให้มีจำนวนบิตเครื่องหมายเกินมาเนื่องจากการคูณกันแบบ ส่วนเต็มเต็มสอง ซึ่งการชิพที่ไปทางซ้าย 1 บิตสามารถกำจัดบิตที่เกินมานี้ได้

TMS320C50 มี ชิพรีจิสเตอร์อยู่ 3 ตัว อยู่ในเส้นทางการทำงานของข้อมูลดังแสดงในรูปที่ 11.14





รูปที่ 11.14 แสดงเส้นทางของข้อมูลใน TMS320C50

ชิพต์เตอร์ทั้ง 3 ตัวนี้ คือ ชิพต์เตอร์ A ,ชิพต์เตอร์ M , และ ชิพต์เตอร์ O โดย ชิพต์เตอร์ A ใช้สำหรับสเกลค่าอินพุตของเอกคิวมูเลเตอร์ ชิพต์เตอร์ M ใช้สำหรับสเกลค่าเอาพุตของมัลติไพเออร์ และ ชิพต์เตอร์ O ใช้สำหรับสเกลค่าเอาพุตของเอกคิวมูเลเตอร์ ข้อมูล 16 บิตที่เป็นอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ ชิพเตอร์ A จะอยู่ที่ตำแหน่ง LSB ของ 32 บิต เอาพุชิพเตอร์ เมื่อไม่มีการชิพที่ชิพเตอร์นี้ ดังนั้นเราสามารถแสดงค่าข้อมูลในรูปแบบใหม่ได้ดังนี้

$$W_I(x_{out}) = W_I(x_{in}) - (sa-16) \quad 11.62$$

โดย x_{in} คือ ชิพเตอร์อินพุต , x_{out} คือ ชิพเตอร์เอาพุต และ sa คือจำนวนของการชิพที่ชิพเตอร์ A สามารถชิพได้ทั้งทางคณิตศาสตร์และทางตรรกะ (Logic) ขึ้นอยู่กับ SXM บิต ซึ่งเป็นหนึ่งในแฟลกสถานะ (Flag Status) ของ TMS320C50 เนื่องจากชิพเตอร์ O สามารถชิพทางตรรกะได้เพียงอย่างเดียวเท่านั้น ดังนั้นชิพเตอร์ O สามารถให้เอาพุตที่มีการชิพบิตเครื่องหมายออกไปแล้ว ซึ่งจะต้องระมัดระวังในการชิพที่ชิพเตอร์นี้ เพราะจะทำให้การคำนวณผิดพลาดได้

จำนวนครั้งของการชิพในแต่ละชิพเตอร์สามารถหาได้โดยมีวิธีการหาดังโพลชาร์ทในรูปที่ 11.15 และสมการ 11.63

$$ACC = \sum_{i=1}^M Sft(sa[i])x[i] + \sum_{j=1}^M Sft(sm)y[j]$$

$$z = Sft(so)ACC \quad 11.63$$

ในรูปที่ 11.15 P คือค่าผลคูณสูงสุดของ $w[j]$ และ $y[j]$ ในสมการที่ 11.63 และ $Sft(n)$ หมายถึงการชิพซ้ายไป n บิต $W_I(t)$ คือ ค่าสูงสุดระหว่าง ความยาวของขนาดของเลขจำนวนเต็มของอินพุต เอกคิวมูเลเตอร์ และ ผลคูณ

การหาจำนวนของการชิพของชิพเตอร์แต่ละตัวจะเริ่มจากการหาค่าจำนวนของการชิพของชิพเตอร์ m ควรจะหาก่อน เนื่องจาก การชิพที่รีจิสเตอร์นี้เป็นการชิพที่ไม่ต่อเนื่อง นั่นคือ ชิพเตอร์ตัวนี้จะทำได้เพียง ชิพ 6,0,1 และ 4 บิต จำนวนของการชิพของชิพเตอร์ m (sm) สามารถหาได้โดยหาจำนวนการชิพที่ทำให้ค่าสมการที่ 11.64 มีค่ามากที่สุด

$$sm = -6,0,1, \text{ หรือ } 4$$

$$W_I(ACC) = W_I(w) + W_I(y) + 1 - sm \geq W_I(t) \quad 11.64$$

หลังจากที่เราได้ค่า $W_I(ACC)$ ที่หามาได้จากการเลือก sm ในขั้นตอนก่อนหน้านี้แล้ว จำนวนของการชิพของ ชิพเตอร์ A (sa[i]) และ ชิพเตอร์ M (so) สามารถหาได้จากสมการที่ 11.65

$$sa[i] = W_I(x[i]) + 16 - W_I(ACC)$$

$$so = W_I(ACC) - W_I(z) \quad 11.65$$

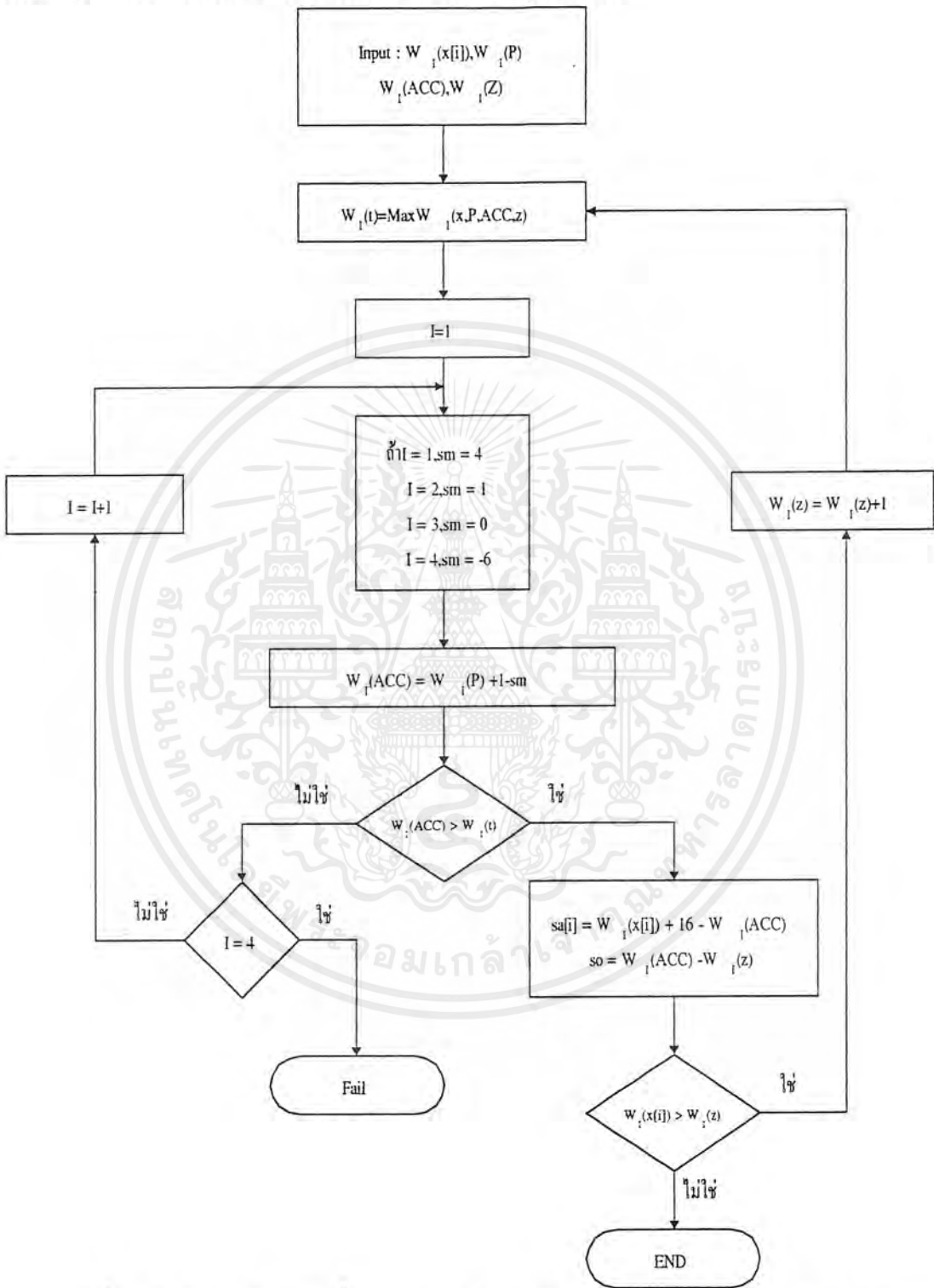
เมื่อจำนวนของการชิพซ้ายของชิพเตอร์ O มีค่ามากกว่า จำนวนของการชิพขวาของ

ชิพเตอร์ A ค่าความยาวของขนาดของเลขจำนวนเต็มของเอาพุตจะต้องบวกด้วย 1 เพื่อป้องกัน

บิตเครื่องหมายหายไป และ ขั้นตอนทั้งหมดจะทำซ้ำใหม่อีกครั้งไปเรื่อยๆ จนกระทั่งได้ค่าจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการชิพท์ของชิพท์แต่ละตัวที่เหมาะสม ดังแสดงในรูปที่ 11.15



รูปที่ 11.15 โพลซาร์ทแสดงขั้นตอนการหาจำนวนชิพท์ของชิพท์เตอร์ใน TMS320C50

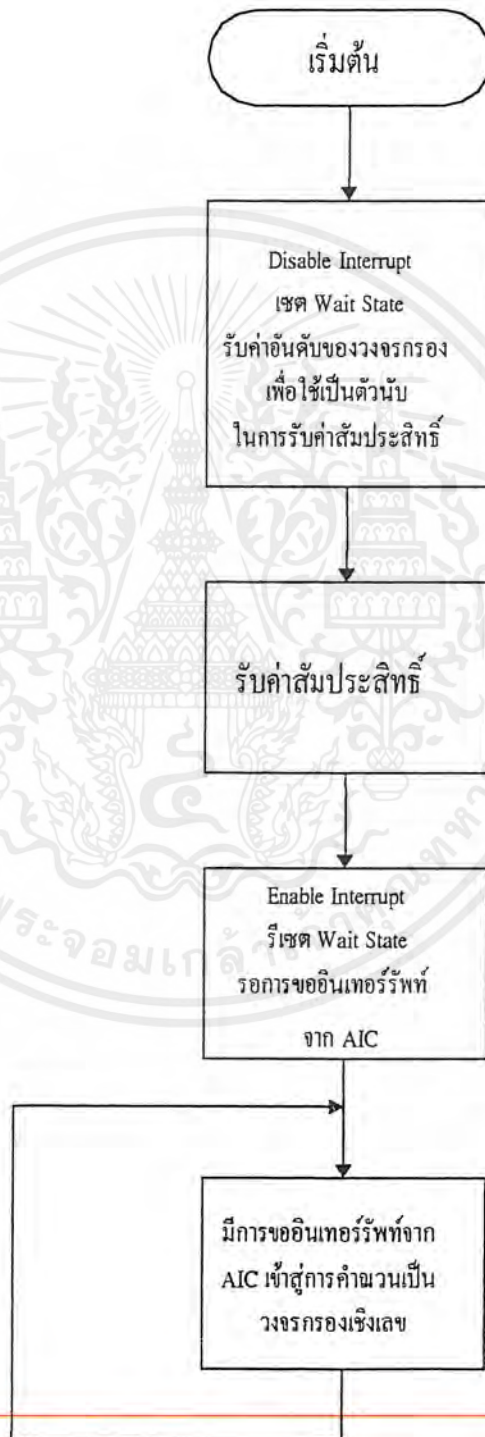
จากที่กล่าวมาข้างต้นเราสามารถหารูปแบบของ Qxx ที่เหมาะสมในการเขียนโปรแกรม DSK เพื่อให้การคำนวณเป็นวงจรต่อเนื่องเลขเป็นไปได้อย่างถูกต้อง และป้องกันการเกิดโอเวอร์โฟลจากการคำนวณโดยใช้โฟลชาร์ทในรูปที่ 11.15 ในการกำหนดการชิพท์ในชิพท์เตอร์แต่ละตัวได้ ซึ่งการกำหนดการชิพท์ในชิพท์เตอร์แต่ละตัวสามารถทำได้โดยใช้การเขียนโปรแกรม DSK ควบคุมอีกทีหนึ่ง

11.3 โปรแกรมส่วนบอร์ด DSK

โปรแกรมในส่วนบอร์ด DSK นี้คือ โปรแกรมที่ทำหน้าที่รับค่าสัมประสิทธิ์จาก IBM PC แล้วนำค่าสัมประสิทธิ์ที่ได้มาคำนวณทำเป็นวงจรต่อเนื่องเลข ประกอบไปด้วยโปรแกรม FIR.ASM และ IIR.ASM ซึ่งก่อนที่จะโหลดลงบอร์ด DSK จะต้องทำการแอสเซมบลีเป็นโปรแกรม .DSK ซะก่อน โดยใช้โปรแกรม DSK5A ในการแอสเซมบลีโปรแกรม.ASM ให้เป็นโปรแกรม .DSK และใช้โปรแกรม DSK5L ในการโหลดโปรแกรมลงบอร์ด ซึ่งโปรแกรมทั้งสองนี้ได้มากับบอร์ด เนื่องจากในขณะที่โปรแกรม DSK5L ทำการโหลดโปรแกรมลงบอร์ด โปรแกรม DSK5L จะมีการทำงานตลอดเวลา ดังนั้นในการใช้งานจริงจะต้องทำการเปิด MS-DOS Prompt 2 หน้าจอ โดยหน้าจอหนึ่งใช้สำหรับรันโปรแกรม FIR-IIR.EXE ที่ทำหน้าที่ในการรับค่าคุณสมบัติของวงจรกรองจากผู้ใ้ แล้วคำนวณค่าสัมประสิทธิ์ตามคุณสมบัติของวงจรกรองแล้วทำการส่งค่าสัมประสิทธิ์ที่คำนวณได้ให้แก่บอร์ด DSK ผ่านทางพอร์ทพริ้นเตอร์ โดยส่งในรูปแบบ Qxx ที่เหมาะสมกับการคำนวณ อีกหน้าจอหนึ่งใช้สำหรับรันโปรแกรม DSK5L ที่ทำการโหลดโปรแกรม .DSK ลงบอร์ดแล้วทำการรันโปรแกรม .DSK ที่โหลดลงไป เพื่อทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลข ซึ่งจำเป็นที่จะต้องเปิดหน้าจอ MS-DOS Prompt ทั้งสองตลอดเวลา เพราะจำเป็นที่จะต้องรันโปรแกรมทั้งสองไปพร้อม ๆ กัน

โปรแกรมในส่วนบอร์ดที่ทำหน้าที่ในการคำนวณเป็นวงจรกรองเชิงเลข ที่ประกอบด้วยโปรแกรม FIR.ASM ที่ทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลขแบบไม่ป้อนกลับ และ โปรแกรม IIR.ASM ที่ทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลขแบบป้อนกลับ โปรแกรมทั้งสองจะมีลักษณะหลัก ๆ คล้ายกัน คือ ทำหน้าที่รับค่าสัมประสิทธิ์ที่ส่งมาจาก IBM PC แล้วทำหน้าที่คำนวณเป็นวงจรกรองเชิงเลข ซึ่งในส่วนของการคำนวณเป็นวงจรกรองเชิงเลขจะเป็นโปรแกรมบริการอินเตอร์รัพท์ของ AIC (โปรแกรมย่อย RINT) เหมือนกัน แต่ในการคำนวณจะมีลักษณะแตกต่างกันตามแบบของวงจร

กรองเชิงเลขว่าเป็นแบบป้อนกลับ หรือ แบบไม่ป้อนกลับ รวมทั้งในส่วนของการรับค่าสัมประสิทธิ์จะมีลำดับในการรับค่าสัมประสิทธิ์ที่แตกต่างกัน เราสามารถแสดงโฟลชาร์ทของโปรแกรมทั้งสองได้ดังรูปที่ 11.16 ซึ่งรายละเอียดของแต่ละโปรแกรมจะกล่าวถึงต่อไป



รูปที่ 11.16 โฟลชาร์ทแสดงการทำงานของโปรแกรมส่วนบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโฟลชาร์ทรูปที่ 11.16 จะสามารถอธิบายการทำงานของแต่ละส่วนได้ โดยจะทำการเปรียบเทียบการทำงานที่แตกต่างกัน ระหว่าง โปรแกรม FIR และ IIR ในแต่ละส่วนเพื่อให้เห็นความแตกต่างในส่วนรายละเอียดได้ชัดเจนขึ้น

การทำงานของโปรแกรมทั้งสองมีลำดับขั้นตอนดังนี้

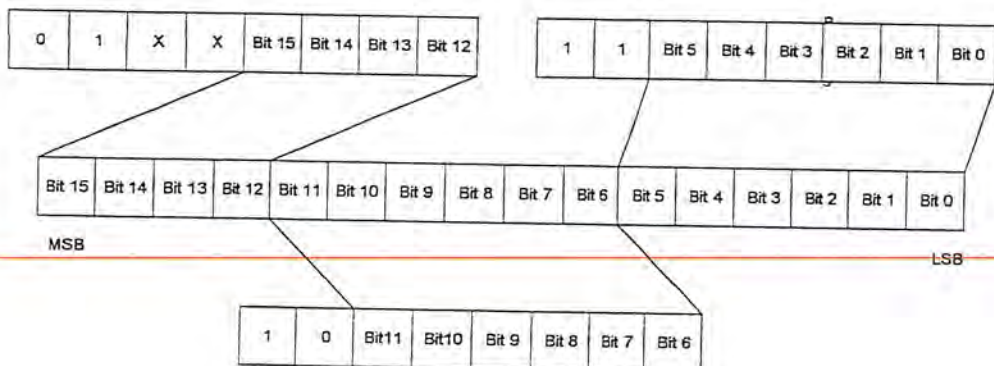
1. Disable Interrupt รับค่าอันดับของวงจรกรอง และ เซต Wait State

เมื่อเริ่มต้นทำงานโปรแกรมจะต้องทำการ Disable Interrupt เพื่อไม่ให้ AIC ทำการอินเทอร์รัพต์โปรแกรมในขณะที่ทำการรับค่าสัมประสิทธิ์จาก IBM PC ทั้งนี้เพราะในการคำนวณเป็นวงจรกรองเชิงเลขจะต้องรับค่าสัมประสิทธิ์ให้ครบซะก่อนที่จะทำการคำนวณเป็นวงจรกรองเชิงเลข ซึ่งการอินเทอร์รัพต์จาก AIC ก็คือการเรียกใช้โปรแกรมในส่วนที่ทำการคำนวณเป็นวงจรกรองเชิงเลขนั่นเอง หลังจากที่ได้ Disable Interrupt แล้วโปรแกรมจะรอรับค่าอันดับของวงจรกรองที่ถูกส่งมาจาก IBM PC เพื่อใช้เป็นตัวนับในการรับค่าสัมประสิทธิ์

การที่ต้องทำการเซตค่า Wait State เพื่อทำให้การติดต่อระหว่าง IBM PC และ บอร์ด DSK สามารถทำการติดต่อกันได้ในเวลาที่ตรงกัน (Synchronize) โดยสามารถทำได้โดยเขียนโปรแกรมควบคุมค่าของรีจิสเตอร์ IOWSR ของ TMS320C50 ดังที่ได้กล่าวมาแล้วก่อนหน้านี้

2. รับค่าสัมประสิทธิ์

หลังจากที่ได้รับค่าอันดับของวงจรกรองแล้ว โปรแกรมก็จะเริ่มรับค่าสัมประสิทธิ์ที่จะถูกส่งมาจาก IBM PC โดยจำนวนค่าสัมประสิทธิ์ที่จะรับเข้ามาจะมีค่าเท่ากับอันดับของวงจรกรองที่ได้รับมาก่อนหน้านี้ แต่เนื่องจากพอร์ทพรีนเตอร์ สามารถส่งข้อมูลได้ทีละ 8 บิต แต่ข้อมูลที่บอร์ด DSK จะนำมาใช้งานได้มีขนาด 16 บิต ดังนั้นในการส่งค่าสัมประสิทธิ์จาก IBM PC นอกจากที่จะต้องแปลงค่าสัมประสิทธิ์ให้อยู่ในรูปแบบ Qxx ที่ บอร์ดเข้าใจแล้ว ยังต้องทำการแบ่งข้อมูลขนาด 16 บิต เป็น 3 ส่วน ก่อนที่จะส่ง ดังแสดงในรูปที่ 11.17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 11.17 แสดงการแบ่งข้อมูลขนาด 16 บิต เป็นข้อมูล 8 บิต 3 ส่วน ในการส่งข้อมูล

จากรูปที่ 11.17 จะเห็นว่าข้อมูลขนาด 16 บิตจะถูกแบ่ง เป็นข้อมูลขนาด 8 บิต 3 ส่วน ก่อนที่ IBM PC จะส่งไปให้แอมบอร์ด โดย 2 บิตที่มีนัยสำคัญสูงสุด จะเป็นตัวบอกว่า ข้อมูลขนาด 8 บิตนั้นเป็นข้อมูลส่วนไหน เมื่อบอร์ดได้รับข้อมูลขนาด 8 บิต 3 ส่วนนี้แล้ว ก็จะมีการรวม ข้อมูลขนาด 8 บิต ทั้ง 3 ส่วนนี้ให้เป็นข้อมูลขนาด 16 บิตตามเดิม โดยอาศัย 2 บิตที่มีนัยสำคัญสูงสุดของข้อมูลแต่ละส่วนเป็นตัวบอกว่าข้อมูลส่วนนั้นเป็นข้อมูลส่วนไหน ข้อแตกต่างระหว่าง โปรแกรม FIR และ IIR ในขั้นตอนนี้คือรูปแบบของข้อมูล โดย ค่าสัมประสิทธิ์ที่โปรแกรม FIR จะนำไปใช้จะต้องอยู่ในรูป Q15 ส่วนค่าสัมประสิทธิ์ที่โปรแกรม IIR จะนำไปใช้จะต้องอยู่ในรูป Q13 (การหารูปแบบที่เหมาะสมของข้อมูลที่บอร์ดจะนำไปใช้งานได้สามารถหาได้จากโฟลชาร์ท ในรูปที่ 11.15)

ข้อแตกต่างระหว่างโปรแกรม FIR และ IIR อีกประการหนึ่ง คือ ลำดับของค่า สัมประสิทธิ์ที่จะถูกส่งมาโดย IBM PC เนื่องจากโครงสร้างของวงจรกรองที่แตกต่างกันดังที่ได้ กล่าวมาก่อนหน้านี้แล้ว โดย โปรแกรม FIR จะรับค่าสัมประสิทธิ์เป็นลำดับดังนี้ $h(0)$ $h(1)$ $h(2)$ $h(N-1)$ ตามลำดับ ส่วนโปรแกรม IIR จะรับค่าสัมประสิทธิ์เป็นลำดับดังนี้ -B2 -B1 A2 A1 A0 ของเสตจแรก หลังจากนั้นก็ส่งค่าสัมประสิทธิ์ของเสตจต่อมาจนถึงเสตจสุดท้าย โดยมี ลำดับการส่งค่าสัมประสิทธิ์เหมือนกัน (ลำดับในการส่งค่าสัมประสิทธิ์นี้ไม่ได้ลำดับรวมการส่ง ข้อมูล 8 บิต 3 ส่วน ซึ่งเป็นรูปแบบจริง ๆ ที่ IBM PC ส่งมา ทั้งนี้เพราะเราสามารถตรวจสอบได้ว่า เป็นข้อมูลส่วนไหน จาก 2 บิตนัยสำคัญสูงสุด) หลังจากที่ได้รับค่าสัมประสิทธิ์แล้ว โปรแกรม จะต้องทำการเก็บค่าสัมประสิทธิ์ไว้ในหน่วยความจำข้อมูล (Data Memory) ของบอร์ด ซึ่งจะต้อง มีการจองหน่วยความจำข้อมูลที่จะใช้เก็บค่าสัมประสิทธิ์นี้ไว้ก่อน โดยขนาดของหน่วยความจำ ข้อมูลที่จะใช้เก็บค่าสัมประสิทธิ์นี้จะมีขนาดเท่ากับจำนวนค่าสัมประสิทธิ์ที่เป็นไปได้สูงสุดของวง จรกรอง

3. Enable Interrupt รีเซต Wait State และ รอการอินเทอร์รัพท์จาก AIC

หลังจากที่บอร์ด DSK ได้รับค่าสัมประสิทธิ์ทั้งหมดจาก IBM PC แล้ว จะทำการรีเซต Wait State เพราะไม่ต้องทำการติดต่อกับ IBM PC แล้ว และการติดต่อยกเว้น TMS320C50 และ AIC ไม่จำเป็นต้องมีการรอเวลาในการติดต่อกัน การรีเซต Wait State สามารถทำได้โดยเขียน โปรแกรมเคลียร์ค่าในรีจิสเตอร์ IOWSR ให้เป็น 0 ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นก็ทำการ Enable Interrupt เพื่อรอการอินเทอร์รัพท์จาก AIC ซึ่งจะเข้าสู่โปรแกรมบริการอินเทอร์รัพท์ ซึ่งเป็นโปรแกรมที่คำนวณเป็นวงจรกรองเชิงเลขนั่นเอง

4. มีการอินเทอร์รัพท์จาก AIC

เมื่อมีการขออินเทอร์รัพท์จาก AIC โปรแกรมก็จะเข้าสู่โปรแกรมบริการอินเทอร์รัพท์ของ AIC (โปรแกรมย่อย RINT) ซึ่งโปรแกรมบริการอินเทอร์รัพท์ของ AIC นี้ จะอยู่แอดเดรสตำแหน่งที่ 080ah ในหน่วยความจำโปรแกรมของบอร์ด ที่ตำแหน่งนี้จะต้องทำการเขียนโปรแกรมบริการอินเทอร์รัพท์ให้เป็นโปรแกรมที่ทำการคำนวณเป็นวงจรกรองเชิงเลข และโปรแกรมก็จะทำการบริการการอินเทอร์รัพท์จาก AIC ไปเรื่อย ๆ จนกว่าจะมีการเปลี่ยนแบบวงจรกรองเชิงเลขเป็นแบบใหม่ โดยผ่านทาง IBM PC ซึ่งการทำงานจะเริ่มใหม่ตั้งแต่ต้น ดังแสดงในรูปที่ 11.18 ที่แสดงการทำงานโดยรวมทั้งหมดของโครงการนี้ โดยรายละเอียดของโปรแกรมทั้งหมดสามารถดูได้ที่ภาคผนวก





รูปที่ 11.18 โฟลชาร์ทแสดงการทำงานทั้งหมดของ โครงการวงจรกรองเชิงเลข

บทที่ 12

การทดลอง และ สรุปผล

ในบทนี้จะเป็นการทดลองวงจรกรองเชิงเลขที่ได้ทำไปแล้วเพื่อดูว่า วงจรกรองเชิงเลข ที่ได้ออกแบบไว้ สามารถทำงานได้ตามที่ออกแบบไว้หรือไม่ ซึ่งสามารถ แบ่งการทดลองได้ดังนี้

การทดลองที่ 1 การทดลองเพื่อหาความถี่ปฏิกิจงานของ AIC บนบอร์ด

การทดลองที่ 2 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR)

การทดลองที่ 3 การทดลองเพื่อทดสอบผลของอันดับของวงจรกรองต่อวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR)

การทดลองที่ 4 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR)

การทดลองที่ 5 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR)

การทดลองที่ 6 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขก้ำกั้แถบความถี่ แบบไม่ป้อนกลับ (FIR)

การทดลองที่ 7 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบป้อนกลับ (IIR)

การทดลองที่ 8 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบป้อนกลับ (IIR)

การทดลองที่ 9 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแถบผ่านความถี่ แบบป้อนกลับ (IIR)

การทดลองที่ 10 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขก้ำกั้แถบความถี่ แบบป้อนกลับ (IIR)

การทดลองที่ 11 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) กับสัญญาณรูปสี่เหลี่ยม

การทดลองที่ 12 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) กับสัญญาณรูปสามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : ในการทดลองทั้งหมดข้างต้นค่าความถี่สุ่มมีค่าคงที่เท่ากับ 19.2 KHz

อุปกรณ์ที่ใช้ในการทดลอง

1. ออสซิลอสโคป 20 MHz รุ่น COS 5020 เอาพุท 1 M Ω 25 pF
2. ฟังก์ชันเจนเนอเรเตอร์ รุ่น GFG-8050 เอาพุท 50 Ω

การทดลองที่ 1 การทดลองเพื่อหาความถี่ปฏิบัติงานของ AIC บนบอร์ด

จุดประสงค์การทดลอง

1. เพื่อหาค่าความถี่ปฏิบัติงานของ AIC บนบอร์ด เมื่อใช้ค่าความถี่สุ่ม 19.2 KHz

ขั้นตอนการทดลอง

- 1.1 เรียก โหลดโปรแกรม INI.DSK ลง บอร์ด โดย ใช้คำสั่ง
DSK5L C:\INI -C2
- 1.2 ป้อนสัญญาณรูปไซน์ขนาด 0.1 V_{p-p} เข้าทางอินพุทของ AIC บนบอร์ด
- 1.3 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟสระหว่างสัญญาณอินพุท และ เอาพุท
- 1.4 ทำการทดลองข้อ 1.2 ถึง 1.3 อีกครั้ง โดยเปลี่ยนรูปสัญญาณเป็นรูปสี่เหลี่ยม
- 1.5 ทำการทดลองข้อ 1.2 ถึง 1.3 อีกครั้ง โดยเปลี่ยนรูปสัญญาณเป็นรูปสามเหลี่ยม

หมายเหตุ : โปรแกรม INI.DSK เป็นโปรแกรมที่ทำการรับค่าสัญญาณที่เข้ามาทางอินพุทของ AIC แล้วนำค่าสัญญาณที่ได้ส่งออกจากเอาพุทของ AIC ทันที สามารถดูได้ที่ภาคผนวก

ผลการทดลอง

ตารางที่ 12.1.1 อินพุทสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

ความถี่ (KHz)	2.048	2.122	2.525	2.920	3.125	3.353	3.437	3.529	3.724	3.842
V _{OUT} (V _{p-p})	0.1	0.1	0.1	0.1	0.1	0.09	0.06	0.04	0.01	0
A _v (dB)	0	0	0	0	0	-0.915	-4.436	-7.958	-20	-
ความต่างเฟส(°)	75	143	90	0	0	-110	-80	-180	-90	-

ตารางที่ 12.1.2 อินพุตสัญญาณรูปสี่เหลี่ยมขนาด 0.1 V_{p-p}

ความถี่ (KHz)	1.987	2.134	2498	2.899	3.114	3.320	3.415	3.497	3.716	3.836
V _{OUT} (V _{p-p})	0.1	0.1	0.1	0.1	0.1	0.08	0.07	0.04	0.02	0
A _v (dB)	0	0	0	0	0	-1.938	-3.098	-7.958	-13.97	-
ความต่างเฟส(°)	90	180	60	45	-90	-180	-150	-110	-90	-

ตารางที่ 12.1.3 อินพุตสัญญาณรูปสามเหลี่ยมขนาด 0.1 V_{p-p}

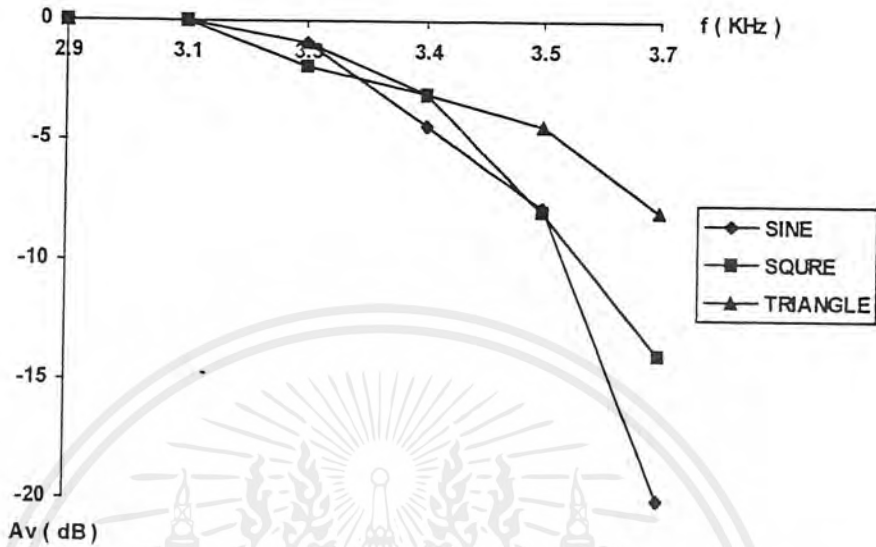
ความถี่ (KHz)	1.892	2.095	2.512	2.904	3.110	3.331	3.426	3.507	3.724	3.833
V _{OUT} (V _{p-p})	0.1	0.1*	0.1	0.1	0.1	0.09	0.07	0.06	0.04	0
A _v (dB)	0	0	0	0	0	-0.915	-3.098	-4.436	-7.958	-
ความต่างเฟส(°)	75	150	90	0	0	-90	-180	-110	-80	-

- สัญญาณเอาพุตที่ได้ในกรณีที่สัญญาณอินพุตเป็นรูปไซน์ จะเป็นรูปไซน์ในทุกความถี่
- สัญญาณเอาพุตที่ได้ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม ในช่วงที่สัญญาณเอาพุตยังมีขนาดเท่ากับสัญญาณอินพุตจะมีรูปสัญญาณเป็นรูปสี่เหลี่ยม ในช่วงที่สัญญาณเอาพุตมีขนาดลดลงน้อยกว่าขนาดสัญญาณอินพุตจะมีรูปสัญญาณเป็นรูปไซน์
- สัญญาณเอาพุตที่ได้ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม จะเป็นรูปไซน์ในทุกความถี่

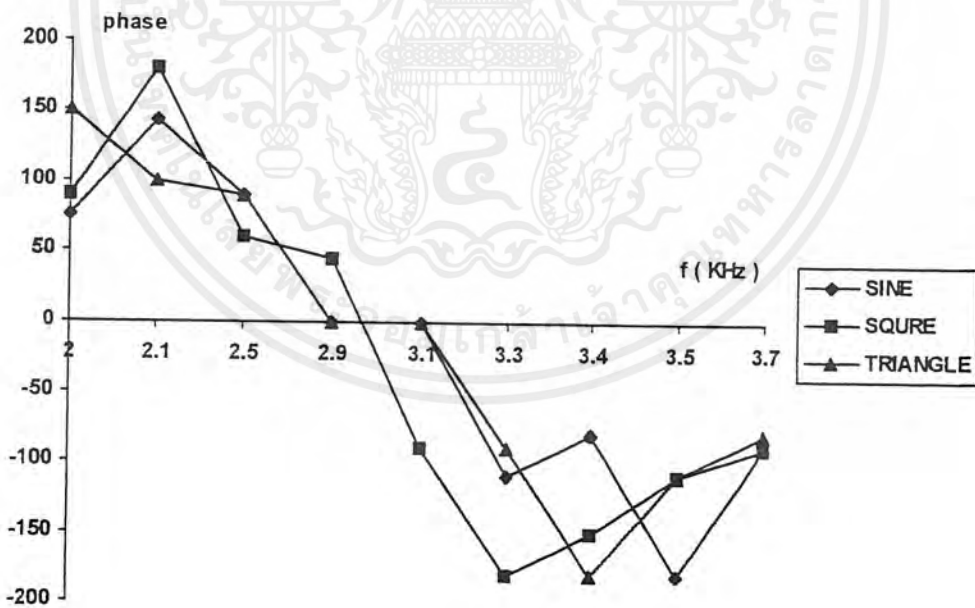


รูปที่ 12.1 แสดงสัญญาณในกรณีสัญญาณอินพุตเป็นรูปไซน์ในการทดลองที่ 1 ที่ความถี่ 1.414KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.2 กราฟแสดงค่า A_v (dB) ในกรณีที่สัญญาณอินพุตรูปต่าง ๆ จากการทดลองที่ 1



รูปที่ 12.3 กราฟแสดงค่าความต่างเฟส ($^{\circ}$) ในกรณีที่สัญญาณอินพุตรูปต่าง ๆ จากการทดลองที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่
2. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) ที่ใช้การออกแบบโดยวินโดว์ที่ต่างกันว่ามีคุณสมบัติที่แตกต่างกันอย่างไร

ขั้นตอนการทดลอง

- 2.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 2.2 เลือกใช้วงจรกรองผ่านความถี่ต่ำ โดยมีค่าความถี่ที่แถบผ่าน (Fp) 2 KHz ความถี่แถบหยุด (Fs) 2.5 KHz เลือกใช้ไคเซอร์วินโดว์ โดยมีค่าลดทอนที่แถบผ่านความถี่ (Ap) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (As) 40 dB
- 2.3 บันทึก ถิ่นดับของวงจรกรองที่ได้
- 2.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟสระหว่างสัญญาณอินพุท และ เอาพุท
- 2.5 ทำการทดลองข้อ 2.1 ถึง 2.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบสี่เหลี่ยม และ ใช้ถิ่นดับของวงจรกรองเท่ากับที่ได้จากข้อ 2.3
- 2.6 ทำการทดลองข้อ 2.1 ถึง 2.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบฮาน และ ใช้ถิ่นดับของวงจรกรองเท่ากับที่ได้จากข้อ 2.3
- 2.7 ทำการทดลองข้อ 2.1 ถึง 2.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแฮมมิง และ ใช้ถิ่นดับของวงจรกรองเท่ากับที่ได้จากข้อ 2.3
- 2.8 ทำการทดลองข้อ 2.1 ถึง 2.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแบล็กแมน และ ใช้ถิ่นดับของวงจรกรองเท่ากับที่ได้จากข้อ 2.3

ผลการทดลอง

ตารางที่ 12.2.1 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองผ่านความถี่ต่ำแบบโคเชอร์วินโดว์

$F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$

อันดับของวงจรกรองที่ได้ = 87

ความถี่ (KHz)	0.305	0.550	0.735	1.031	1.262	1.533	1.701	1.873
$V_{OUT} (V_{P-P})$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Loss (dB)	0	0	0	0	0	0	0	0
ความต่างเฟส($^{\circ}$)	0	90	60	0	-90	-120	0	138

ความถี่ (KHz)	1.985	1.996	2.098	2.109	2.291	2.413	2.499
$V_{OUT} (V_{P-P})$	0.1	0.08	0.07	0.055	0.03	0.012	0
Loss (dB)	0	1.938	3.098	5.193	10.45	18.41	-
ความต่างเฟส($^{\circ}$)	90	57	-150	-180	-234	-180	-

ตารางที่ 12.2.2 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองผ่านความถี่ต่ำแบบวินโดว์สี่เหลี่ยม $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	0.354	0.523	723	1.019	1.239	1.539	1.699	1.823
$V_{OUT} (V_{P-P})$	0.1	0.1	0.1	0.1	0.1	0.09	0.08	0.07
Loss (dB)	0	0	0	0	0	0.915	1.938	3.098
ความต่างเฟส($^{\circ}$)	0	180	0	0	-225	-180	0	50

ความถี่ (KHz)	1.900	2.028	2.102	2.224	2.311	2.410	2.511	2.593
$V_{OUT} (V_{P-P})$	0.07	0.07	0.07	0.05	0.03	0.008	0.006	0
Loss (dB)	3.098	3.098	3.098	6.021	10.45	21.94	24.44	-
ความต่างเฟส($^{\circ}$)	180	0	-14	-180	-100	180	0	-

ตารางที่ 12.2.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองผ่านความถี่ต่ำแบบฮานวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	0.296	0.520	0.701	0.998	1.223	1.537	1.704	1.817
$V_{OUT} (V_{P-P})$	0.1	0.1	0.1	0.1	0.09	0.09	0.09	0.09
Loss (dB)	0	0	0	0	0.915	0.915	0.915	0.915
ความต่างเฟส($^{\circ}$)	40	180	0	0	-180	-180	0	180

ความถี่ (KHz)	1.924	2.020	2.125	2.212	2.292	2.402	2.495	2.569
$V_{OUT} (V_{P-P})$	0.08	0.07	0.060	0.050	0.032	0.020	0.010	0
Loss (dB)	1.938	3.098	4.437	6.021	9.897	13.98	20.00	-
ความต่างเฟส($^{\circ}$)	180	0	0	90	-180	0	0	-

ตารางที่ 12.2.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองผ่านความถี่ต่ำแบบแฮมมิงวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	0.296	0.498	0.734	0.991	1.204	1.513	1.715	1.809
$V_{OUT} (V_{P-P})$	0.1	0.1	0.1	0.08	0.08	0.07	0.07	0.07
Loss (dB)	0	0	0	1.938	1.938	3.098	3.098	3.098
ความต่างเฟส($^{\circ}$)	40	150	0	-50	-180	-180	0	120

ความถี่ (KHz)	1.916	2.002	2.094	2.223	2.330	2.429	2.509
$V_{OUT} (V_{P-P})$	0.07	0.07	0.06	0.050	0.036	0.020	0
Loss (dB)	3.098	3.098	4.437	6.021	8.874	13.98	-
ความต่างเฟส($^{\circ}$)	180	90	0	-180	-120	0	-

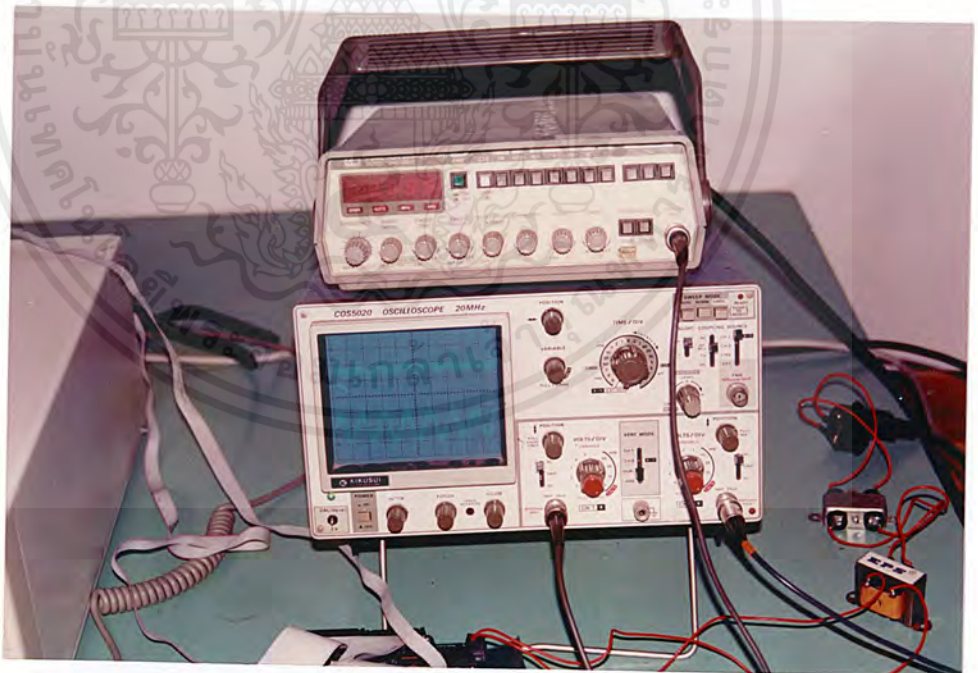
ตารางที่ 12.2.5 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองผ่านความถี่ต่ำแบบแบล็กแมนวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 87

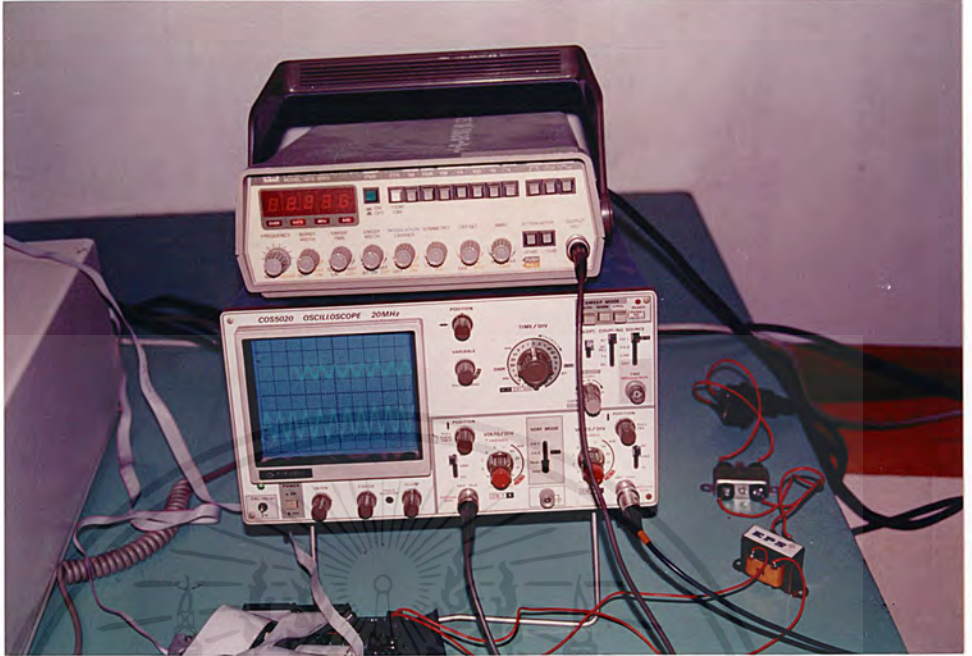
ความถี่ (KHz)	0.355	0.524	0.737	1.039	1.220	1.522	1.626	1.817
$V_{OUT} (V_{P-P})$	0.1	0.1	0.1	0.1	0.080	0.070	0.070	0.070
Loss (dB)	0	0	0	0	1.938	3.098	3.098	3.098
ความต่างเฟส($^{\circ}$)	0	180	0	0	-180	-90	120	180

ความถี่ (KHz)	1.955	2.056	2.157	2.212	2.310	2.406	2.513	2.689
$V_{OUT} (V_{P-P})$	0.070	0.070	0.050	0.050	0.040	0.020	0.010	0
Loss (dB)	3.098	3.098	6.021	6.021	7.598	13.98	20.00	-
ความต่างเฟส($^{\circ}$)	150	0	0	-180	-120	0	180	-

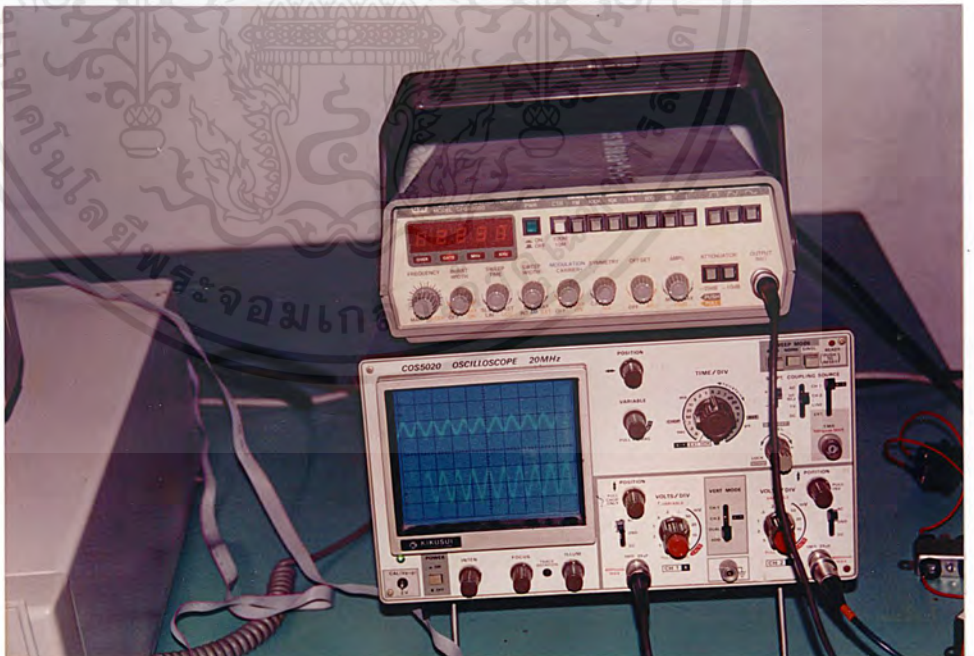


รูปที่ 12.4 รูปแสดงสัญญาณเมื่อใช้วงจรกรอง แบบไม่ป้อนกลับ (FIR) วินโดว์ฮาน ที่ความถี่ 2.237 KHz จากการทดลองที่ 2 (50 mV/DIV , 0.2 uS/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

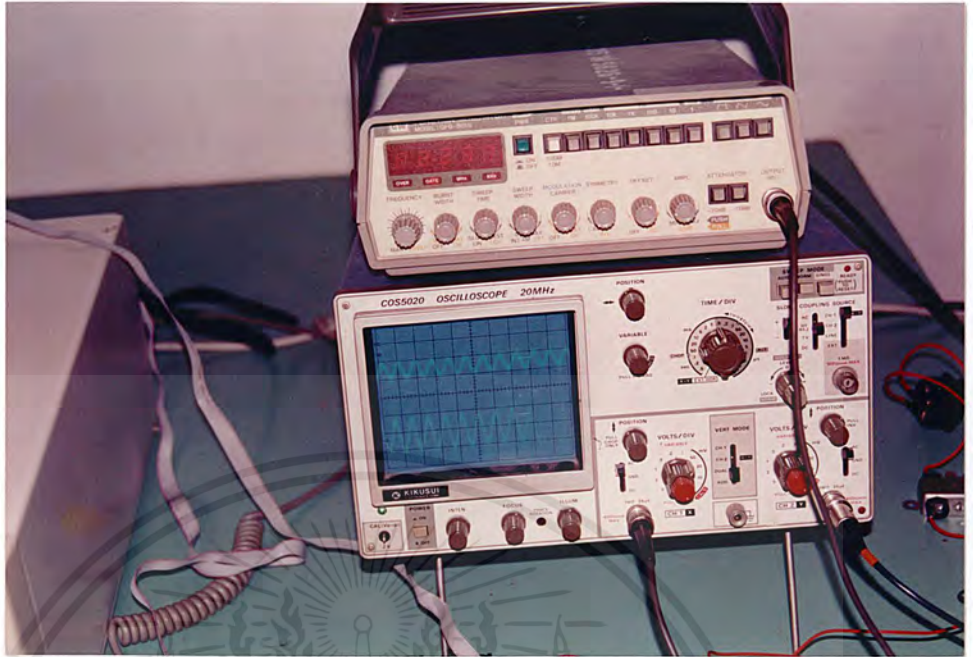


รูปที่ 12.5 รูปแสดงสัญญาณเมื่อใช้วงจรกรอง แบบไม่ป้อนกลับ (FIR) วินโดว์แฮมมิง ที่ความถี่ 2.237 KHz จากการทดลองที่ 2 (50 mV/DIV , 0.2 μ S/DIV)

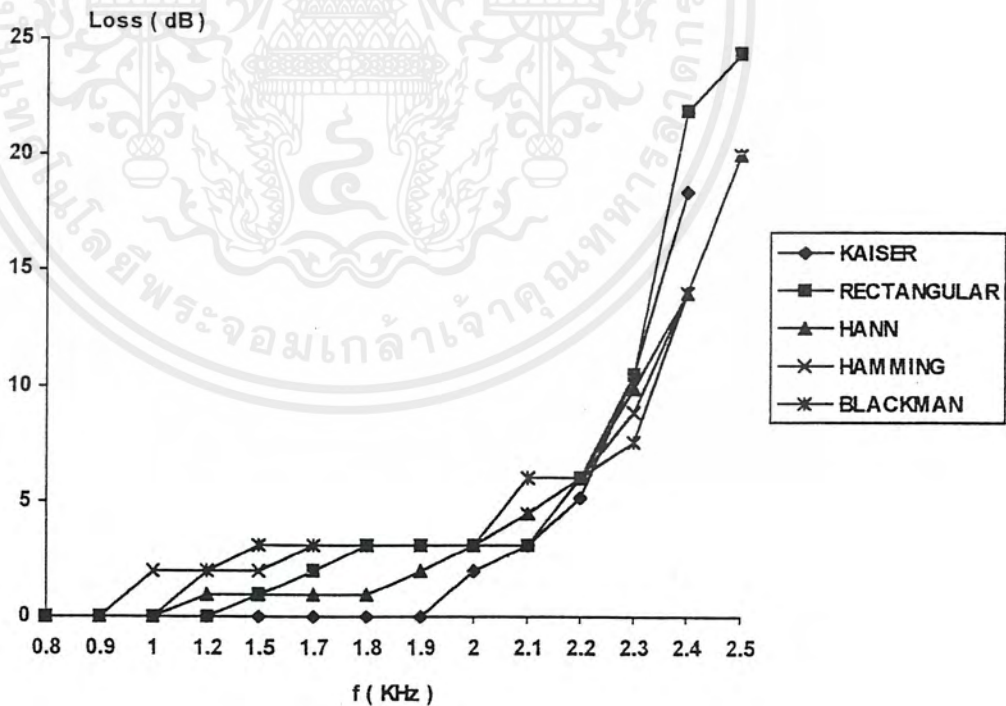


รูปที่ 12.6 รูปแสดงสัญญาณเมื่อใช้วงจรกรอง แบบไม่ป้อนกลับ (FIR) วินโดว์เบลคแมน ที่ความถี่ 2.237 KHz จากการทดลองที่ 2 (50 mV/DIV , 0.2 μ S/DIV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

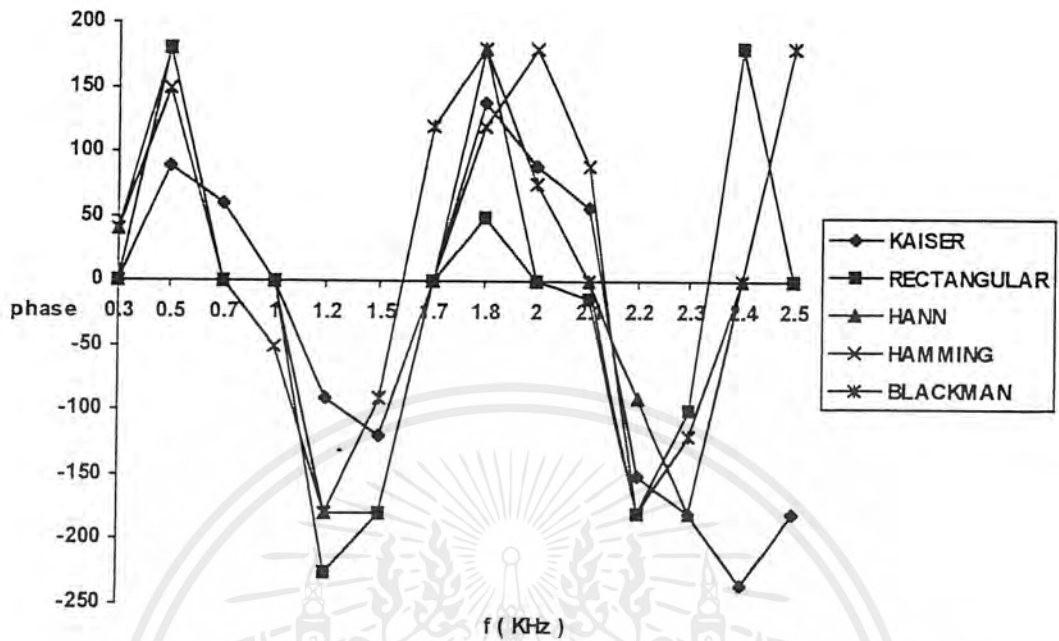


รูปที่ 12.7 รูปแสดงสัญญาณเมื่อใช้วงจรกรอง แบบไม่ป้อนกลับ (FIR) วินโดว์โคเชอร์ ที่ความถี่ 2.237 KHz จากการทดลองที่ 2 (50 mV/DIV , 0.2 uS/DIV)



รูปที่ 12.8 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.9 กราฟแสดง ความต่างเฟส ($^{\circ}$) ของวงจรกรองผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 2

การทดลองที่ 3 การทดลองเพื่อทดสอบผลของอันดับของวงจรกรองต่อวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR)

จุดประสงค์การทดลอง

1. เพื่อดูผลของอันดับของวงจรกรองต่อวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) ในแต่ละแบบวินโดว์

ขั้นตอนการทดลอง

- 3.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 3.2 เลือกใช้วงจรกรองผ่านความถี่ต่ำ โดยมีค่าความถี่ที่แถบผ่าน (Fp) 2 KHz ความถี่แถบหยุด(Fs) 2.5 KHz เลือกใช้วินโดว์สี่เหลี่ยม เลือกอันดับของวงจรกรองเท่ากับ 100
- 3.3 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟสระหว่างสัญญาณอินพุท และ เอาพุท
- 3.4 ทำการทดลองข้อ 3.2 อีกครั้ง โดยใช้อันดับวงจรกรองเท่ากับ 2
- 3.5 ทำการทดลองข้อ 3.2 ถึง 3.4 อีกครั้ง โดยเปลี่ยนเป็นแบบฮานวินโดว์
- 3.6 ทำการทดลองข้อ 3.2 ถึง 3.4 อีกครั้ง โดยเปลี่ยนเป็นแบบแฮมมิงวินโดว์
- 3.7 ทำการทดลองข้อ 3.2 ถึง 3.4 อีกครั้ง โดยเปลี่ยนเป็นแบบเบลทแมนวินโดว์

ผลการทดลอง

ตารางที่ 12.3.1 อินพุทสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่ต่ำแบบวินโดว์สี่เหลี่ยม Fp = 2 KHz Fs = 2.5 KHz

อันดับของวงจรกรอง = 100

ความถี่ (KHz)	0.316	0.531	0.701	1.083	1.209	1.503	1.729	1.830
$V_{OUT} (V_{p-p})$	0.08	0.06	0.055	0.055	0.055	0.055	0.060	0.060
Loss (dB)	1.938	4.437	5.193	5.193	5.193	5.193	4.437	4.437
ความต่างเฟส($^{\circ}$)	0	90	120	160	0	0	-180	0

ความถี่ (KHz)	1.891	2.024	2.114	2.197	2.328	2.410	2.543	2.564
$V_{OUT} (V_{p-p})$	0.060	0.070	0.090	0.080	0.050	0.020	0.010	0
Loss (dB)	4.437	3.098	0.915	1.938	6.021	13.98	20.00	-
ความต่างเฟส($^{\circ}$)	40	0	-60	-60	-180	0	90	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

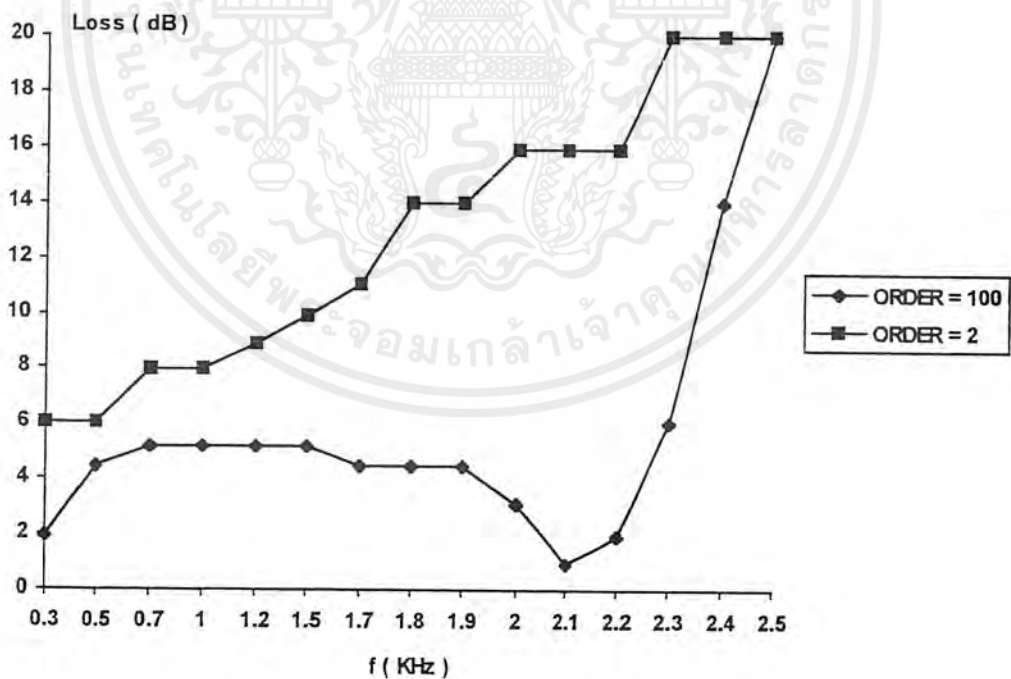
ตารางที่ 12.3.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองผ่านความถี่ต่ำแบบวินโดว์สี่เหลี่ยม $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 2

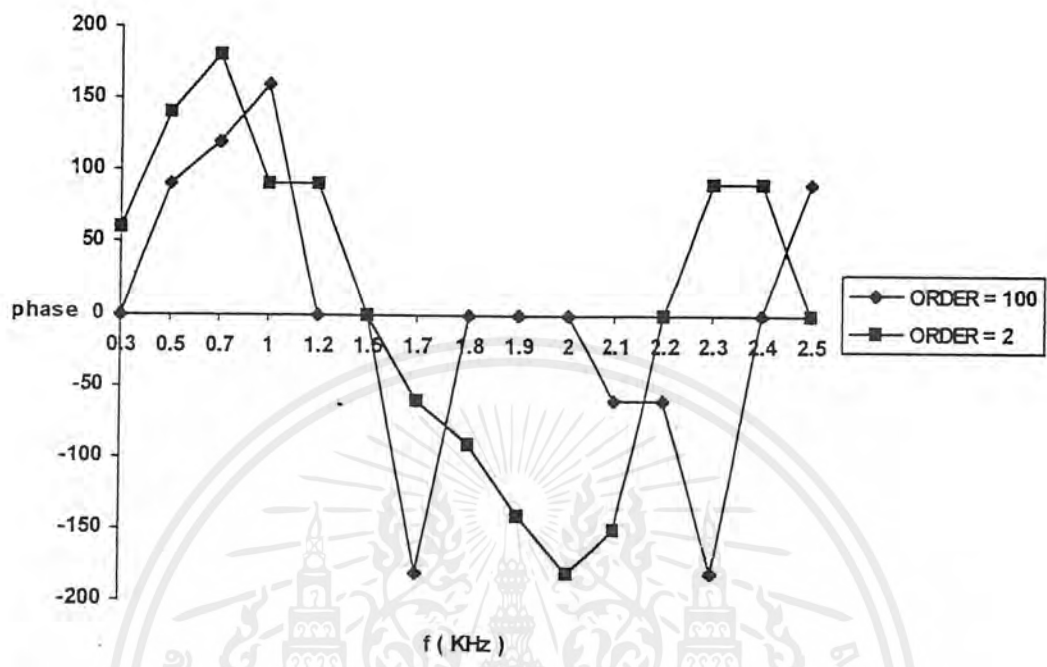
ความถี่ (KHz)	0.300	0.520	0.735	1.075	1.211	1.551	1.738	1.828
$V_{OUT} (V_{P-P})$	0.050	0.050	0.040	0.040	0.036	0.032	0.028	0.020
Loss (dB)	6.021	6.021	7.958	7.958	8.874	9.897	11.06	13.98
ความต่างเฟส($^{\circ}$)	60	140	180	90	90	0	-60	-90

ความถี่ (KHz)	1.930	2.069	2.118	2.212	2.313	2.417	2.536	2.715
$V_{OUT} (V_{P-P})$	0.020	0.016	0.016	0.016	0.010	0.010	0.010	0
Loss (dB)	13.98	15.92	15.92	15.92	20.00	20.00	20.00	-
ความต่างเฟส($^{\circ}$)	-140	-180	-150	0	90	90	0	-



รูปที่ 12.10 กราฟแสดงค่า Loss (dB) ของวงจรกรองแบบวินโดว์สี่เหลี่ยม ที่อันดับต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.11 กราฟแสดงค่า ความต่างเฟส (°) ของวงจรกรองแบบวินโดว์สี่เหลี่ยม ที่อันดับต่างกัน

ตารางที่ 12.3.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}
วงจรกรองผ่านความถี่ต่ำแบบฮานวินโดว์ F_p = 2 KHz F_s = 2.5 KHz
อันดับของวงจรกรอง = 100

ความถี่ (KHz)	0.301	0.506	0.734	1.030	1.223	1.514	1.732	1.830
V _{OUT} (V _{p-p})	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.07
Loss (dB)	3.098	3.098	3.098	4.437	4.437	4.437	4.437	3.098
ความต่างเฟส(°)	0	180	120	90	0	0	-180	0

ความถี่ (KHz)	1.896	2.048	2.122	2.221	2.303	2.438	2.532	2.609
V _{OUT} (V _{p-p})	0.07	0.07	0.06	0.06	0.05	0.032	0.020	0
Loss (dB)	3.098	3.098	4.437	4.437	6.021	9.897	13.98	-
ความต่างเฟส(°)	0	180	0	0	-180	0	0	-

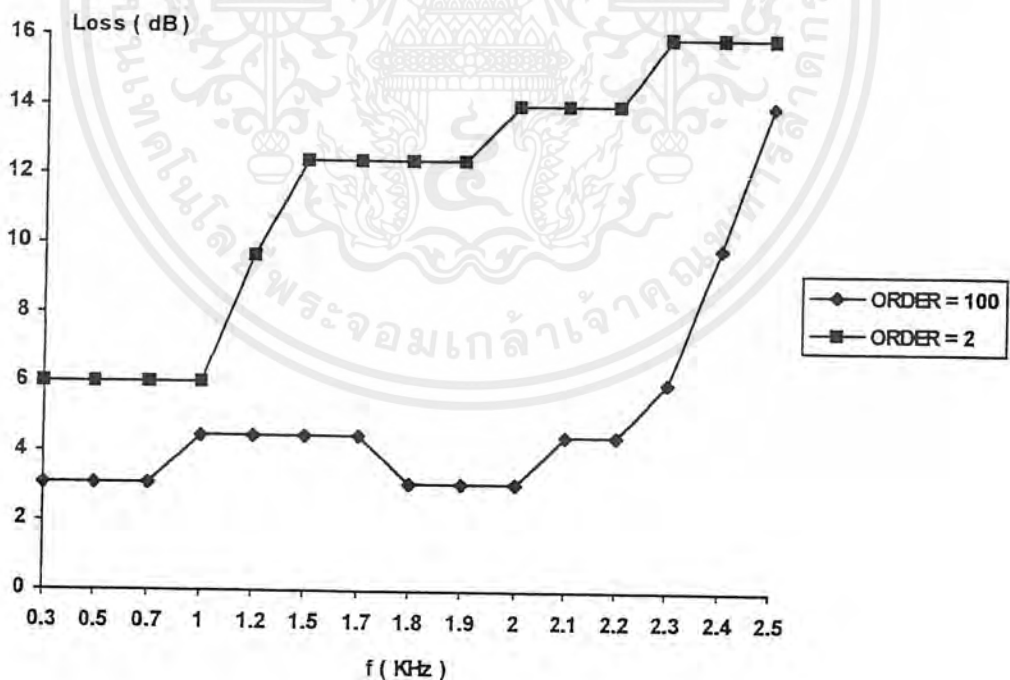
ตารางที่ 12.3.4 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองผ่านความถี่ต่ำแบบฮานวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 2

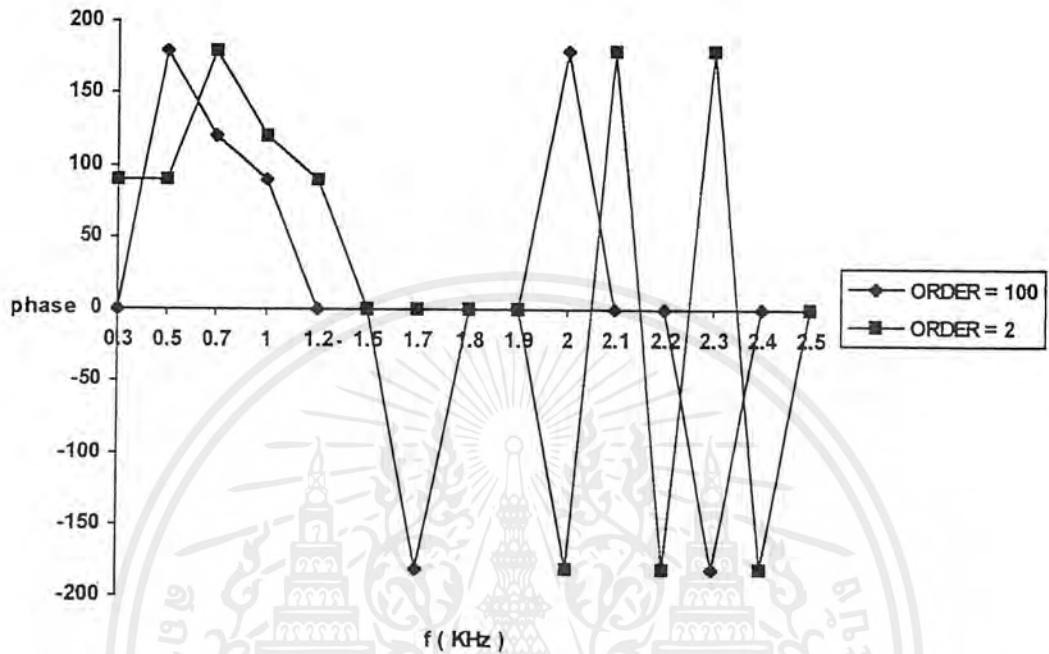
ความถี่ (KHz)	0.323	0.493	0.719	1.012	1.194	1.547	1.729	1.812
$V_{OUT} (V_{P-P})$	0.050	0.050	0.040	0.040	0.033	0.024	0.024	0.024
Loss (dB)	6.021	6.021	6.021	6.021	9.629	12.39	12.39	12.39
ความต่างเฟส($^{\circ}$)	90	90	180	120	90	0	0	0

ความถี่ (KHz)	1.910	2.000	2.118	2.219	2.300	2.438	2.523	2.614
$V_{OUT} (V_{P-P})$	0.024	0.020	0.020	0.020	0.016	0.016	0.016	0
Loss (dB)	12.39	13.98	13.98	13.98	15.92	15.92	15.92	-
ความต่างเฟส($^{\circ}$)	0	-180	180	-180	180	-180	0	-



รูปที่ 12.12 กราฟแสดงค่า Loss (dB) ของวงจรกรองแบบฮานวินโดว์ ที่อันดับต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.13 กราฟแสดงค่า ความต่างเฟส ($^{\circ}$) ของวงจรกรองแบบฮามมิงวินโดว์ ที่อันดับต่างกัน

ตารางที่ 12.3.5 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่ต่ำแบบแฮมมิงวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 100

ความถี่ (KHz)	0.325	0.492	0.690	1.012	1.224	1.527	1.701	1.817
$V_{OUT} (V_{p-p})$	0.070	0.070	0.070	0.060	0.050	0.050	0.060	0.065
Loss (dB)	3.098	3.098	3.098	4.437	6.021	6.021	4.437	3.742
ความต่างเฟส($^{\circ}$)	0	180	60	0	0	0	-180	0

ความถี่ (KHz)	1.918	2.014	2.126	2.246	2.310	2.440	2.511	2.728
$V_{OUT} (V_{p-p})$	0.070	0.070	0.065	0.050	0.040	0.024	0.020	0
Loss (dB)	3.098	3.098	3.742	6.021	7.958	12.39	13.98	-
ความต่างเฟส($^{\circ}$)	90	180	0	-60	-180	0	0	-

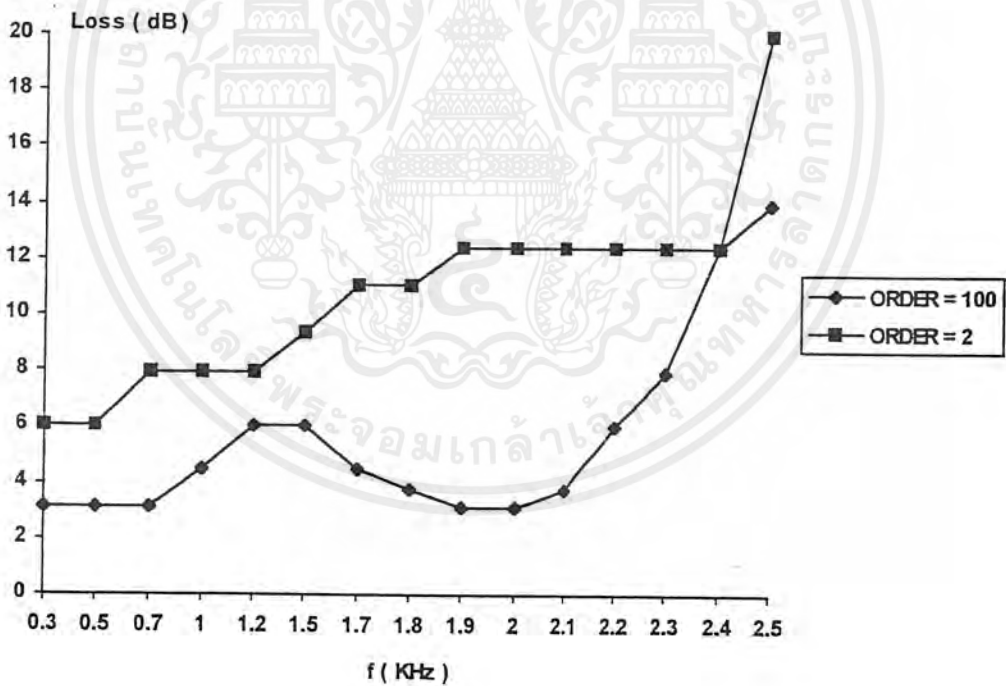
ตารางที่ 12.3.6 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรกรองผ่านความถี่ต่ำแบบแฮมมิงวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$
อันดับของวงจรกรอง = 2

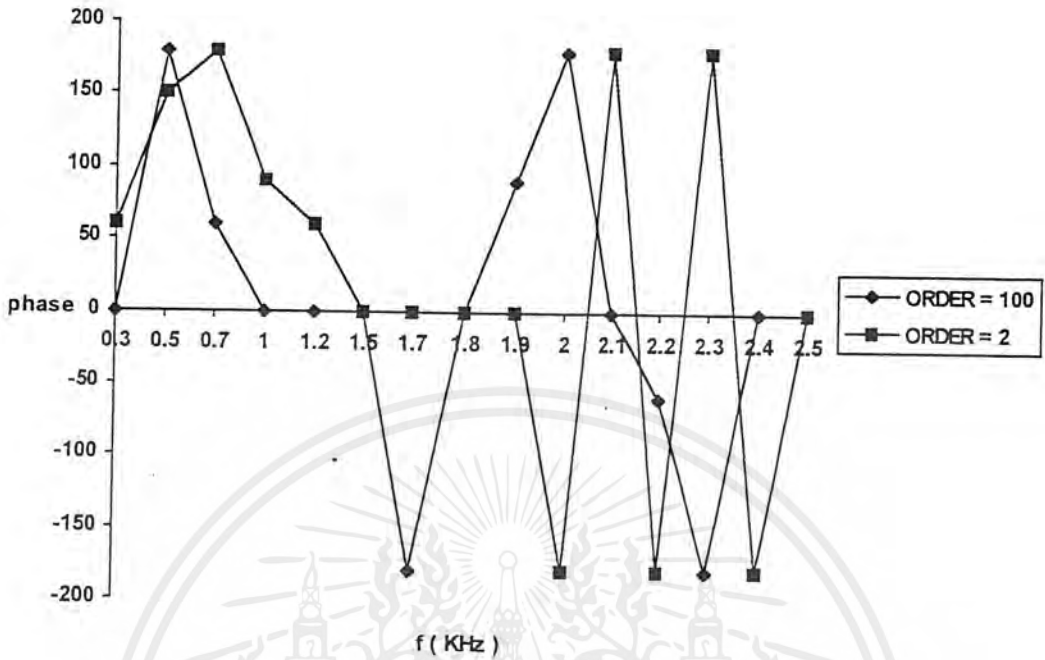
ความถี่ (KHz)	0.290	0.540	0.744	1.083	1.172	1.540	1.689	1.813	1.904
$V_{OUT} (V_{P-P})$	0.050	0.050	0.040	0.040	0.040	0.034	0.028	0.028	0.024
Loss (dB)	6.021	6.021	7.958	7.958	7.958	9.370	11.06	11.06	12.39
ความต่างเฟส($^{\circ}$)	60	150	180	90	60	0	0	0	0

ความถี่ (KHz)	2.002	2.097	2.216	2.294	2.594	2.417	2.523	2.785
$V_{OUT} (V_{P-P})$	0.024	0.024	0.024	0.024	0.024	0.024	0.010	0
Loss (dB)	12.39	12.39	12.39	12.39	12.39	12.39	20.00	-
ความต่างเฟส($^{\circ}$)	-180	180	-180	180	-180	-120	0	-



รูปที่ 12.14 กราฟแสดงค่า Loss (dB) ของวงจรกรองแบบแฮมมิงวินโดว์ ที่อันดับต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.15 กราฟแสดงค่า ความต่างเฟส ($^{\circ}$) ของวงจรกรองแบบแฮมมิงวินโดว์ ที่อันดับต่างกัน

ตารางที่ 12.3.7 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองผ่านความถี่ต่ำแบบเบตทแมนวินโดว์ $F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$

อันดับของวงจรกรอง = 100

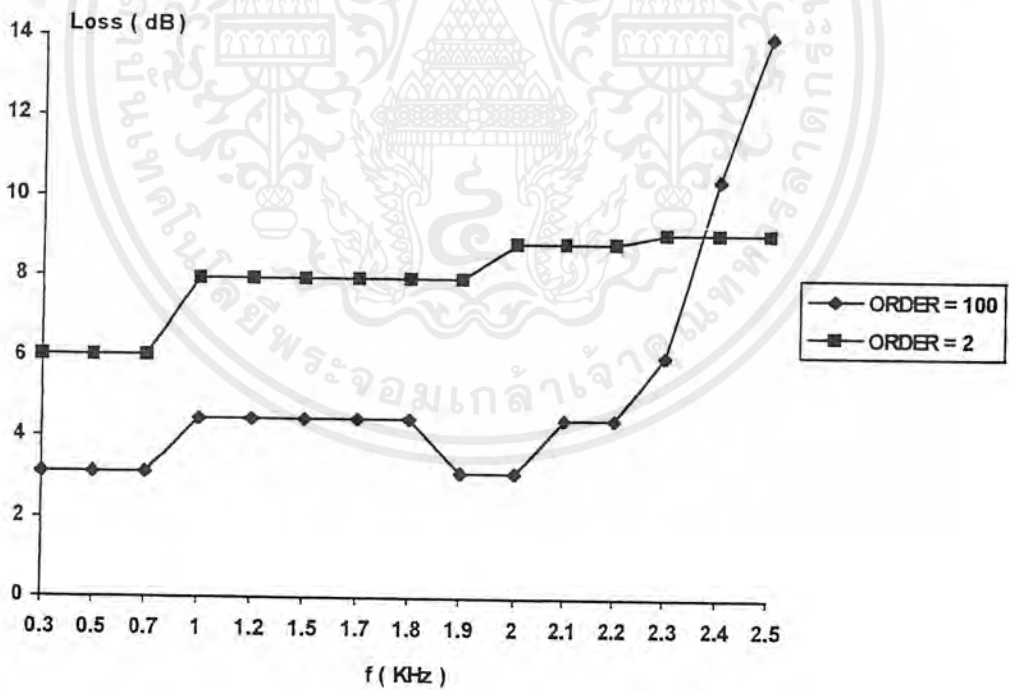
ความถี่ (KHz)	0.301	0.500	0.732	0.976	1.198	1.493	1.726	1.802
$V_{OUT} (V_{P-P})$	0.070	0.070	0.070	0.060	0.060	0.060	0.060	0.060
Loss (dB)	3.098	3.098	3.098	4.437	4.437	4.437	4.437	4.437
ความต่างเฟส($^{\circ}$)	0	180	120	0	0	0	-150	0

ความถี่ (KHz)	1.953	2.001	2.099	2.203	2.310	2.405	2.492	2.715
$V_{OUT} (V_{P-P})$	0.070	0.070	0.060	0.060	0.050	0.030	0.020	0
Loss (dB)	3.098	3.098	4.437	4.437	6.021	10.45	13.98	-
ความต่างเฟส($^{\circ}$)	90	180	60	40	-180	-100	0	-

ตารางที่ 12.3.8 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p,p}
วงจรกรองผ่านความถี่ต่ำแบบแบล็กแมนวินโดว์ F_p = 2 KHz F_s = 2.5 KHz
อันดับของวงจรกรอง = 2

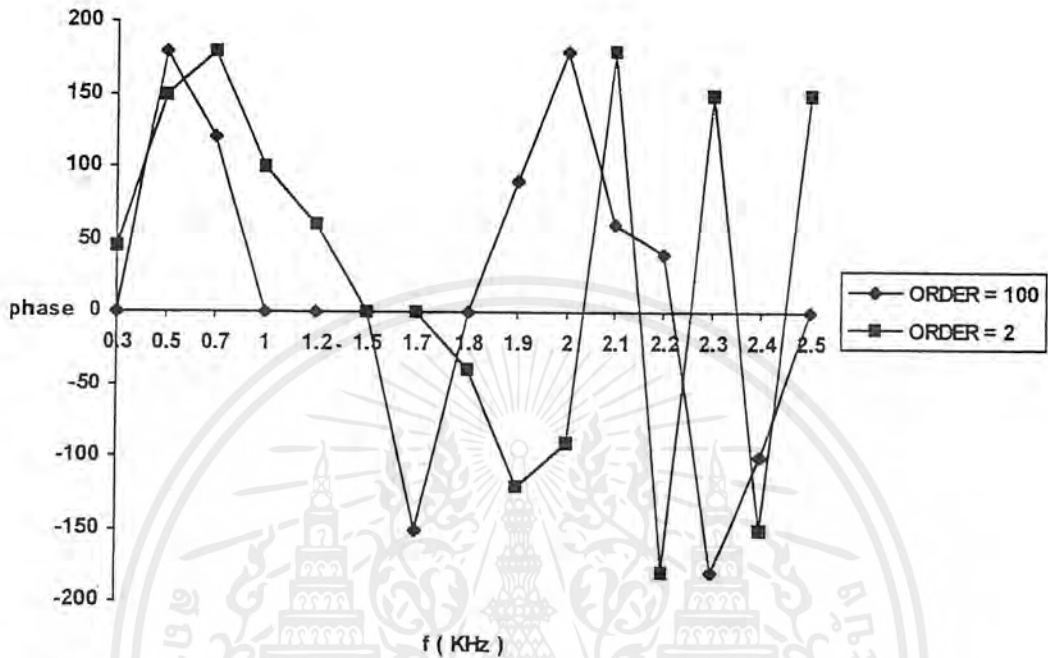
ความถี่ (KHz)	0.293	0.596	0.718	1.023	1.218	1.540	1.728	1.812
V _{OUT} (V _{p,p})	0.050	0.050	0.050	0.040	0.040	0.040	0.040	0.040
Loss (dB)	6.021	6.021	6.021	7.958	7.958	7.958	7.958	7.958
ความต่างเฟส(°)	45	150	180	100	60	0	0	-40

ความถี่ (KHz)	1.954	2.020	2.101	2.196	2.307	2.415	2.508	2.889
V _{OUT} (V _{p,p})	0.040	0.036	0.036	0.036	0.035	0.035	0.035	0
Loss (dB)	7.958	8.874	8.874	8.874	9.118	9.118	9.118	-
ความต่างเฟส(°)	-120	-90	180	-180	150	-150	150	-



รูปที่ 12.16 กราฟแสดงค่า Loss (dB) ของวงจรกรองแบบแบล็กแมนวินโดว์ ที่อันดับต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.17 กราฟแสดงค่า ความต่างเฟส ($^{\circ}$) ของวงจรกรองแบบเบลคแมนมิงวินโดว์ ที่ อันดันต่างกัน

การทดลองที่ 4 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่
2. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR) ที่ใช้การออกแบบโดยวินโดว์ที่ต่างกันว่ามีคุณสมบัติที่แตกต่างกันอย่างไร

ขั้นตอนการทดลอง

- 4.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p,p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 4.2 เลือกใช้วงจรกรองผ่านความถี่สูง โดยมีค่าความถี่ที่แถบผ่าน (F_p) 2.5 KHz ความถี่แถบหยุด (F_s) 2 KHz เลือกใช้โคเชอร์วินโดว์ โดยมีค่าลดทอนที่แถบผ่านความถี่ (A_p) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (A_s) 40 dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 บันทึกอันดับของวงจรกรองที่ได้

4.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟส ระหว่างสัญญาณอินพุท และ เอาพุท

4.5 ทำการทดลองข้อ 4.1 ถึง 4.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบสี่เหลี่ยม และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 4.3

4.6 ทำการทดลองข้อ 4.1 ถึง 4.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบฮาน และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 4.3

4.7 ทำการทดลองข้อ 4.1 ถึง 4.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแฮมมิง และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 4.3

4.8 ทำการทดลองข้อ 4.1 ถึง 4.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบเบลกแมน และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 4.3

ผลการทดลอง

ตารางที่ 12.4.1 อินพุทสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่สูงแบบโคเชอร์วินโดว์

$F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$

อันดับของวงจรกรองที่ได้ = 87

ความถี่ (KHz)	2.076	2.140	2.212	2.289	2.408	2.501	2.642	2.743
$V_{OUT} (V_{p-p})$	0.020	0.030	0.040	0.060	0.080	0.090	0.1	0.1
Loss (dB)	13.98	10.45	7.598	4.437	1.938	0.915	0	0
ความต่างเฟส($^{\circ}$)	0	90	180	50	0	-60	-45	180

ตารางที่ 12.4.2 อินพุทสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่สูงแบบวินโดว์สี่เหลี่ยม $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	2.029	2.170	2.210	2.308	2.445	2.594	2.639	2.721
$V_{OUT} (V_{p-p})$	0.020	0.020	0.040	0.070	0.08	0.09	0.1	0.1
Loss (dB)	13.98	13.98	7.958	3.098	1.938	0.915	0	0
ความต่างเฟส($^{\circ}$)	180	60	0	-60	-90	-40	0	40

ตารางที่ 12.4.3 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่สูงแบบฮานวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	2.052	2.100	2.209	2.348	2.434	2.528	2.605	2.785
$V_{OUT} (V_{p-p})$	0.010	0.020	0.040	0.070	0.08	0.09	0.1	0.1
Loss (dB)	20.00	13.98	7.958	3.098	1.938	0.915	0	0
ความต่างเฟส($^{\circ}$)	0	30	0	-30	-120	-150	-180	0

ตารางที่ 12.4.4 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่สูงแบบแฮมมิงวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$

อันดับของวงจรกรอง = 87

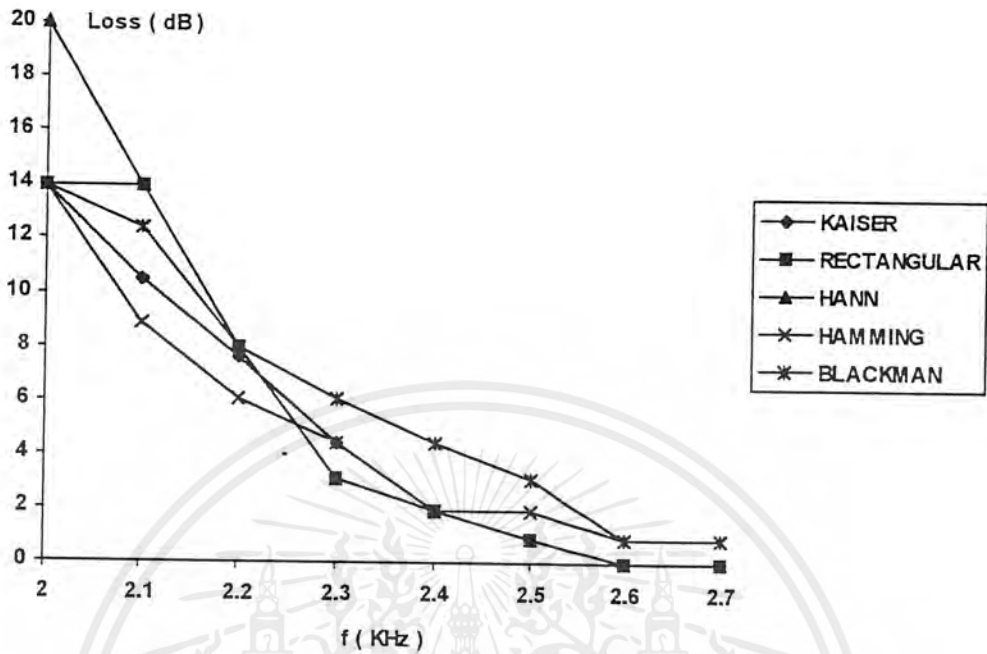
ความถี่ (KHz)	2.121	2.182	2.301	2.412	2.500	2.613	2.687	2.756
$V_{OUT} (V_{p-p})$	0.020	0.036	0.050	0.060	0.08	0.08	0.9	0.9
Loss (dB)	13.98	8.874	6.021	4.437	1.938	1.938	0.915	0.915
ความต่างเฟส($^{\circ}$)	40	180	90	0	-90	90	0	40

ตารางที่ 12.4.5 อินพุตสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$

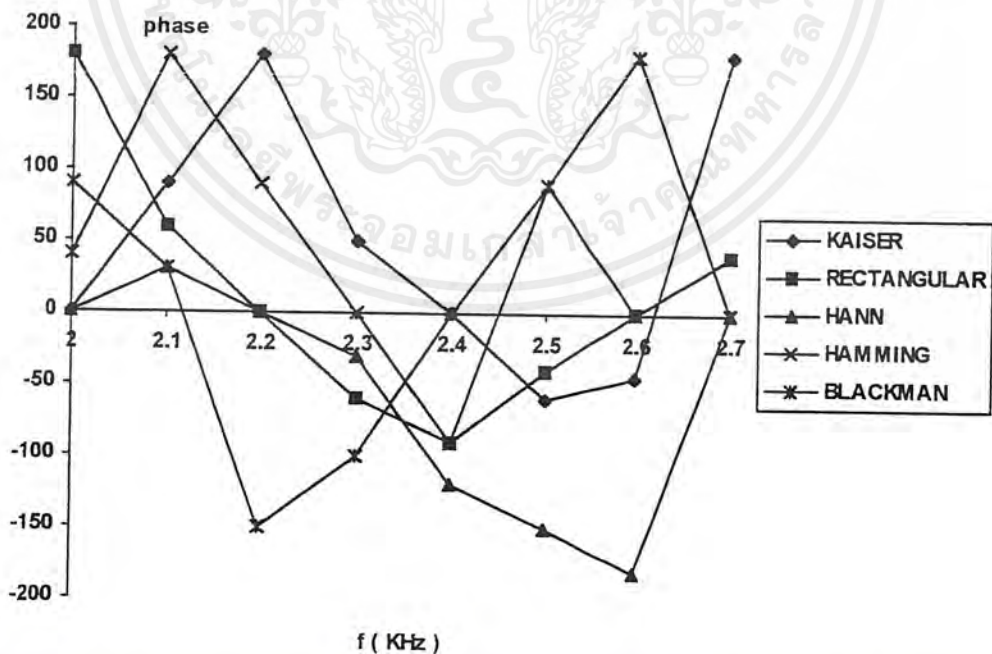
วงจรกรองผ่านความถี่สูงแบบเบลกแมนวินโดว์ $F_p = 2.5 \text{ KHz}$ $F_s = 2 \text{ KHz}$

อันดับของวงจรกรอง = 87

ความถี่ (KHz)	1.935	2.015	2.106	2.194	2.309	2.414	2.501	2.611	2.707
$V_{OUT} (V_{p-p})$	0.010	0.020	0.024	0.040	0.050	0.060	0.070	0.09	0.09
Loss (dB)	20.00	13.98	12.39	7.958	6.021	4.437	3.098	0.915	0.915
ความต่างเฟส($^{\circ}$)	150	90	30	-150	-100	0	90	180	0



รูปที่ 12.18 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 4



รูปที่ 12.19 กราฟแสดง ความต่างเฟส (°) ของวงจรกรองผ่านความถี่สูง แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่
2. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR) ที่ใช้การออกแบบโดยวินโดว์ที่ต่างกันว่ามีคุณสมบัติที่แตกต่างกันอย่างไร

ขั้นตอนการทดลอง

- 5.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 5.2 เลือกใช้วงจรกรองแถบผ่านความถี่ โดยมีค่าความถี่ที่แถบผ่านต่ำ (F_{p1}) 1 KHz ความถี่ที่แถบผ่านสูง (F_{p2}) 2 KHz ความถี่แถบหยุดต่ำ (F_{s1}) 0.5 KHz ความถี่แถบหยุดสูง (F_{s2}) 2.5 KHz เลือกใช้ไคเซอร์วินโดว์ โดยมีค่าลดทอนที่แถบผ่านความถี่ (A_p) 1 dB ค่าลดทอนที่แถบหยุดความถี่ (A_s) 35 dB
- 5.3 บันทึกอันดับของวงจรกรองที่ได้
- 5.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟสระหว่างสัญญาณอินพุท และ เอาพุท
- 5.5 ทำการทดลองข้อ 5.1 ถึง 5.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบสี่เหลี่ยม และ ใช้อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 5.3
- 5.6 ทำการทดลองข้อ 5.1 ถึง 5.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบฮาน และ ใช้อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 5.3
- 5.7 ทำการทดลองข้อ 5.1 ถึง 5.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแฮมมิง และ ใช้อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 5.3
- 5.8 ทำการทดลองข้อ 5.1 ถึง 5.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบเบลทแมน และ ใช้อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 5.3

ผลการทดลอง

ตารางที่ 12.5.1 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองแถบผ่านความถี่แบบโคเชอร์วินโดว์

Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

Ap = 1 dB As = 35 dB

อันดับของวงจรกรองที่ได้ = 75

ความถี่ (KHz)	0.595	0.810	1.055	1.117	1.497	1.713	1.903
V _{OUT} (V _{P-P})	0.020	0.060	0.090	0.1	0.1	0.1	0.090
Loss (dB)	13.98	4.437	0.915	0	0	0	0.915
ความต่างเฟส(°)	180	0	-60	0	0	180	0

ความถี่ (KHz)	2.000	2.099	2.190	2.314	2.475	2.563
V _{OUT} (V _{P-P})	0.080	0.070	0.060	0.030	0.010	0
Loss (dB)	1.938	3.098	4.437	10.45	20.00	-
ความต่างเฟส(°)	-60	-180	-60	0	180	-

ตารางที่ 12.5.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองแถบผ่านความถี่แบบวินโดว์สี่เหลี่ยม

Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.563	0.693	0.801	1.004	1.239	1.542	1.810
V _{OUT} (V _{P-P})	0.010	0.030	0.060	0.1	0.1	0.1	0.1
Loss (dB)	20.00	10.45	4.437	0	0	0	0
ความต่างเฟส(°)	0	-50	0	120	90	0	-140

ความถี่ (KHz)	2.006	2.101	2.209	2.321	2.403	2.546
V _{OUT} (V _{P-P})	0.09	0.09	0.060	0.020	0.010	0
Loss (dB)	0.915	0.915	4.437	13.98	20.00	-
ความต่างเฟส(°)	-60	180	90	0	0	-

ตารางที่ 12.5.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองแถบผ่านความถี่แบบฮานวินโดว์

Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.457	0.704	0.805	1.014	1.136	1.326	1.514	1.801
V _{OUT} (V _{p-p})	0.010	0.040	0.060	0.090	0.090	0.090	0.090	0.090
Loss (dB)	20.00	7.958	4.437	0.915	0.915	0.915	0.915	0.915
ความต่างเฟส(°)	0	40	0	-180	0	180	0	-90

ความถี่ (KHz)	1.894	2.017	2.113	2.199	2.312	2.415	2.517	2.618
V _{OUT} (V _{p-p})	0.080	0.070	0.060	0.050	0.040	0.020	0.010	0
Loss (dB)	1.938	3.098	4.437	6.021	7.958	13.98	20.00	-
ความต่างเฟส(°)	0	70	180	100	0	-120	-180	-

ตารางที่ 12.5.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองแถบผ่านความถี่แบบแฮมมิงวินโดว์

Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.433	0.506	0.720	0.873	1.025	1.117	1.515	1.777
V _{OUT} (V _{p-p})	0	0.010	0.040	0.070	0.090	0.090	0.090	0.090
Loss (dB)	-	20.00	13.98	3.098	0.915	0.915	0.915	0.915
ความต่างเฟส(°)	-	100	0	-80	-100	-50	0	180

ความถี่ (KHz)	1.995	2.085	2.197	2.301	2.404	2.498	2.618
V _{OUT} (V _{p-p})	0.090	0.060	0.050	0.040	0.020	0.010	0
Loss (dB)	0.915	4.437	6.021	7.958	13.98	20.00	-
ความต่างเฟส(°)	70	-180	-120	0	120	180	-

ตารางที่ 12.5.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

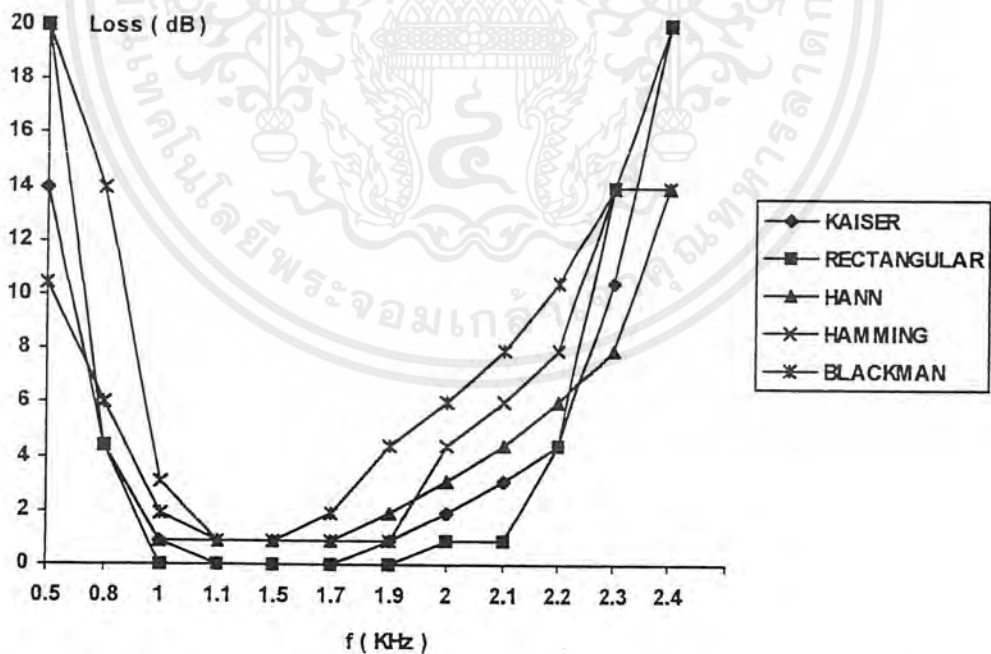
วงจรกรองแถบผ่านความถี่แบบเบลทแมนวินโดว์

Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

อันดับของวงจรกรอง = 75

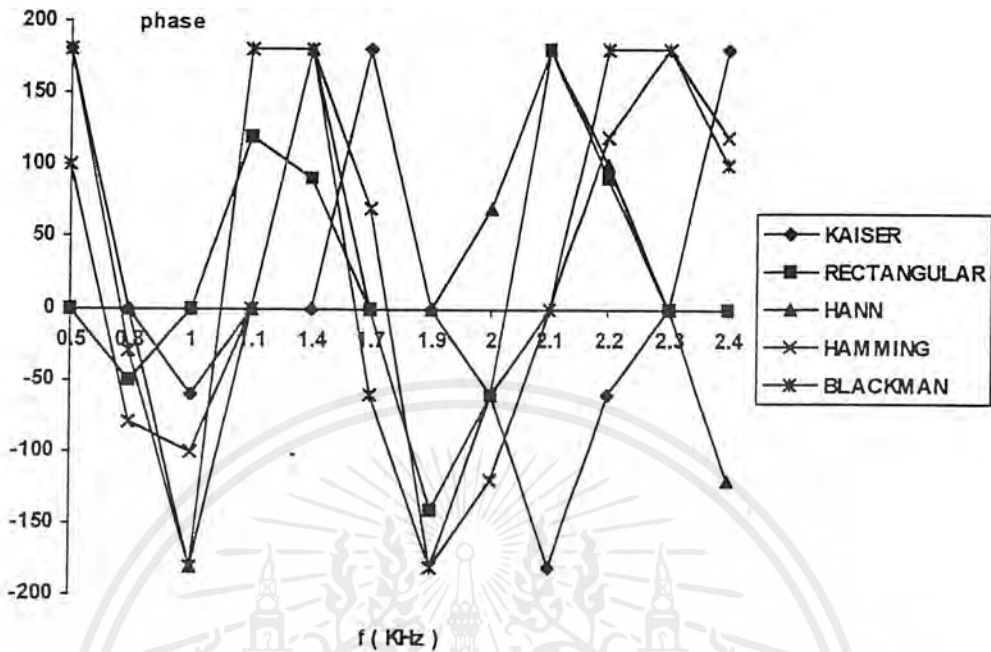
ความถี่ (KHz)	0.433	0.613	0.729	0.818	0.941	1.047	1.134	1.376	1.509
V _{OUT} (V _{p-p})	0.012	0.030	0.040	0.050	0.070	0.080	0.090	0.090	0.090
Loss (dB)	18.42	10.45	7.958	6.021	3.098	1.938	0.915	0.915	0.915
ความต่างเฟส(°)	0	180	0	-30	-180	-180	0	180	0

ความถี่ (KHz)	1.701	1.900	2.000	2.122	2.221	2.323	2.410	2.502	2.667
V _{OUT} (V _{p-p})	0.090	0.090	0.080	0.060	0.050	0.040	0.030	0.020	0
Loss (dB)	0.915	0.915	1.938	4.437	6.021	7.968	10.45	13.98	-
ความต่างเฟส(°)	180	0	-60	-180	-60	0	180	180	-



รูปที่ 12.20 กราฟแสดง Loss (dB) ของวงจรกรองแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.21 กราฟแสดง ความต่างเฟส ($^{\circ}$) ของวงจรกรองแถบผ่านความถี่ แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 5

การทดลองที่ 6 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขก่าจัดแถบความถี่ แบบไม่ป้อนกลับ (FIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขก่าจัดแถบความถี่ แบบไม่ป้อนกลับ (FIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่

2. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขก่าจัดแถบความถี่ แบบไม่ป้อนกลับ (FIR) ที่ใช้การออกแบบโดยวินโดว์ที่ต่างกันว่ามีคุณสมบัติที่แตกต่างกันอย่างไร

ขั้นตอนการทดลอง

6.1 ป้อนสัญญาณรูปไซน์ขนาด 0.1 V_{p-p} เข้าทางอินพุตของ AIC บนบอร์ด

6.2 เลือกใช้วงจรกรองก่าจัดแถบความถี่ โดยมีค่าความถี่ที่แถบผ่านต่ำ (Fp1) 0.5 KHz ความถี่ที่แถบผ่านสูง (Fp2) 2.5 KHz ความถี่แถบหยุดต่ำ (Fs1) 1 KHz ความถี่แถบหยุดสูง (Fs2) 2 KHz เลือกใช้โคเซอร์วินโดว์ โดยมีค่าลดทอนที่แถบผ่านความถี่ (Ap) 1 dB ค่าลดทอนที่แถบหยุดความถี่ (As) 35 dB

6.3 บันทึกอันดับของวงจรกรองที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟส ระหว่างสัญญาณอินพุท และ เอาพุท

6.5 ทำการทดลองข้อ 6.1 ถึง 6.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบสี่เหลี่ยม และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 6.3

6.6 ทำการทดลองข้อ 6.1 ถึง 6.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบฮาน และ ใช้อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 6.3

6.7 ทำการทดลองข้อ 6.1 ถึง 6.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแฮมมิง และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 6.3

6.8 ทำการทดลองข้อ 6.1 ถึง 6.4 อีกครั้ง โดยเปลี่ยนวินโดว์เป็นแบบแบล็กแมน และ ใช้ อันดับของวงจรกรองเท่ากับที่ได้จากข้อ 6.3

ผลการทดลอง

ตารางที่ 12.6.1 อินพุทสัญญาณรูปไซน์ขนาด $0.1 V_{P-P}$

วงจรกรองกำจัดแถบความถี่แบบโคเชอร์วินโดว์

$F_{p1} = 0.5 \text{ KHz}$ $F_{s1} = 1 \text{ KHz}$ $F_{p2} = 2.5 \text{ KHz}$ $F_{s2} = 2 \text{ KHz}$

$A_p = 1 \text{ dB}$ $A_s = 35 \text{ dB}$

อันดับของวงจรกรองที่ได้ = 75

ความถี่ (KHz)	0.631	0.676	0.798	0.928	1.234	1.423	1.769	1.945
$V_{OUT} (V_{P-P})$	0.070	0.050	0.030	0.010	0.010	0.010	0	0
Loss (dB)	3.098	6.021	10.45	20.00	20.00	20.00	-	-
ความต่างเฟส($^{\circ}$)	120	60	0	-180	-40	0	-	-

ความถี่ (KHz)	2.048	2.094	2.202	2.322	2.381	2.571	2.635	2.786
$V_{OUT} (V_{P-P})$	0.010	0.020	0.040	0.060	0.080	0.090	0.1	0.1
Loss (dB)	20.00	13.98	7.958	4.437	1.938	0.915	0	0
ความต่างเฟส($^{\circ}$)	180	-180	-40	0	50	180	80	0

ตารางที่ 12.6.2 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองกำจัดแถบความถี่แบบวินโดว์สี่เหลี่ยม

Fp1 = 0.5 KHz Fs1 = 1 KHz Fp2 = 2.5 KHz Fs2 = 2 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.405	0.502	0.602	0.720	0.852	1.012	1.884
V _{OUT} (V _{p-p})	0.1	0.1	0.080	0.060	0.020	0.016	0
Loss (dB)	0	0	1.938	4.437	13.98	15.92	-
ความต่างเฟส(°)	0	120	180	100	0	0	-

ความถี่ (KHz)	2.024	2.199	2.302	2.373	2.490	2.686
V _{OUT} (V _{p-p})	0.010	0.040	0.060	0.080	0.1	0.1
Loss (dB)	20.00	7.958	4.437	1.938	0	0
ความต่างเฟส(°)	90	60	0	-60	-180	0

ตารางที่ 12.6.3 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองกำจัดแถบความถี่แบบฮานวินโดว์

Fp1 = 0.5 KHz Fs1 = 1 KHz Fp2 = 2.5 KHz Fs2 = 2 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.405	0.536	0.695	0.808	0.906	1.056	1.128
V _{OUT} (V _{p-p})	0.1	0.090	0.060	0.040	0.020	0.010	0
Loss (dB)	0	0.915	4.437	7.958	13.98	20.00	-
ความต่างเฟส(°)	0	180	80	0	-180	0	-

ความถี่ (KHz)	1.936	2.140	2.273	2.426	2.593	2.754
V _{OUT} (V _{p-p})	0.010	0.030	0.045	0.070	0.090	0.090
Loss (dB)	20.00	10.45	6.936	3.098	0.915	0.915
ความต่างเฟส(°)	0	180	0	-150	-100	0

ตารางที่ 12.6.4 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองก่้าจัดแถบความถี่แบบแฮมมิงวินโดว์

Fp1 = 0.5 KHz Fs1 = 1 KHz Fp2 = 2.5 KHz Fs2 = 2 KHz

อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.362	0.371	0.485	0.597	0.731	0.782	0.973
V _{OUT} (V _{p-p})	0.1	0.090	0.090	0.070	0.040	0.040	0.020
Loss (dB)	0	0.915	0.915	3.098	4.437	4.437	13.98
ความต่างเฟส(°)	0	0	60	180	0	0	-180

ความถี่ (KHz)	1.035	1.937	2.009	2.209	2.322	2.519
V _{OUT} (V _{p-p})	0.010	0	0.010	0.040	0.060	0.090
Loss (dB)	20.00	-	20.00	7.958	4.437	0.915
ความต่างเฟส(°)	-90	-	100	80	0	-180

ตารางที่ 12.6.5 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

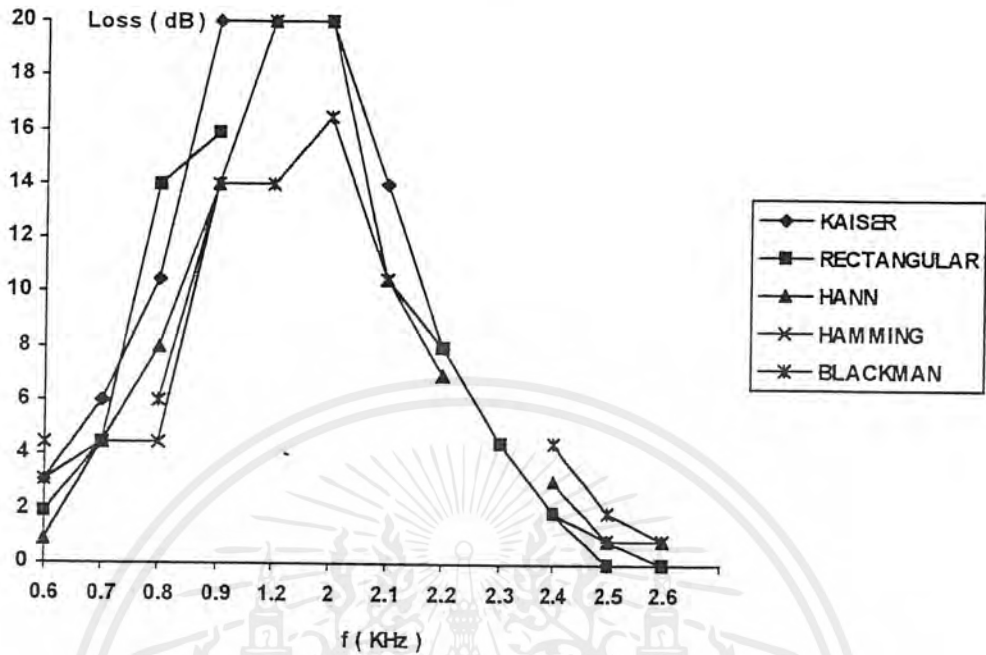
วงจรกรองก่้าจัดแถบความถี่แบบเบลทแมนวินโดว์

Fp1 = 0.5 KHz Fs1 = 1 KHz Fp2 = 2.5 KHz Fs2 = 2 KHz

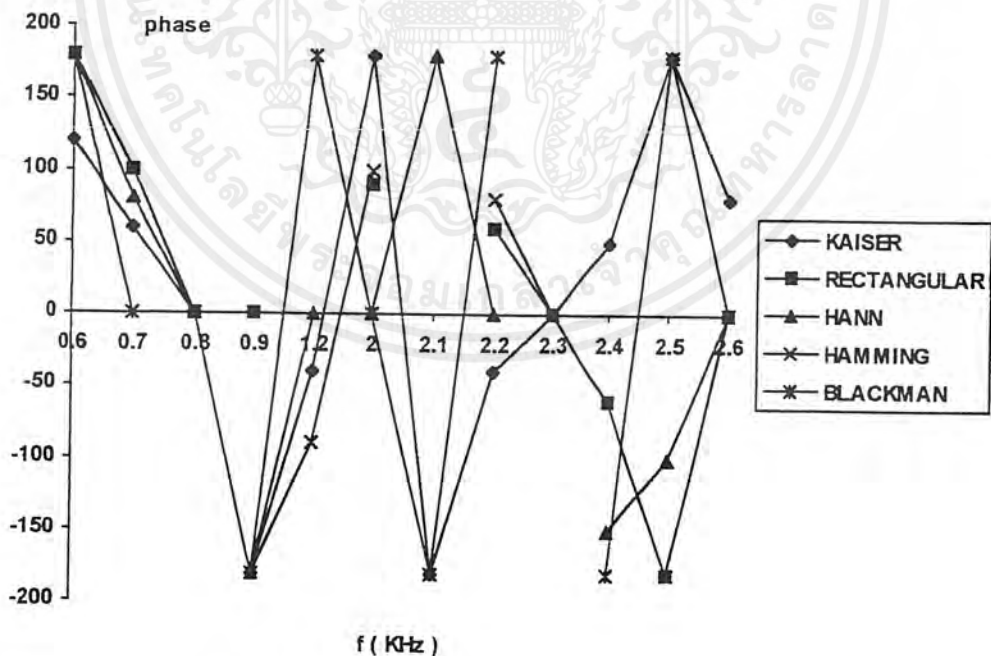
อันดับของวงจรกรอง = 75

ความถี่ (KHz)	0.361	0.486	0.634	0.765	0.950	1.118	1.065
V _{OUT} (V _{p-p})	0.1	0.090	0.060	0.050	0.020	0	0.020
Loss (dB)	0	0.915	4.437	6.021	13.98	-	13.98
ความต่างเฟส(°)	0	60	180	0	-180	-	180

ความถี่ (KHz)	1.970	2.117	2.199	2.431	2.528	2.661
V _{OUT} (V _{p-p})	0.015	0.030	0.040	0.060	0.080	0.09
Loss (dB)	16.48	10.45	7.958	4.437	1.938	0.915
ความต่างเฟส(°)	0	-180	180	-180	180	0



รูปที่ 12.22 กราฟแสดง Loss (dB) ของวงจรกรองกำเนิดแถบความถี่ แบบไม่ป้อนกลับ (FIR) จากการ ทดลองที่ 6



รูปที่ 12.23 กราฟแสดง ความต่างเฟส (°) ของวงจรกรองกำเนิดแถบความถี่ แบบไม่ป้อนกลับ (FIR) จากการทดลองที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบป้อนกลับ (IIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบป้อนกลับ (IIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่

ขั้นตอนการทดลอง

- 7.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 7.2 เลือกใช้วงจรกรองผ่านความถี่ต่ำ แบบป้อนกลับ (IIR) โดยมีค่าความถี่ที่แถบผ่าน (Fp) 2 KHz ความถี่แถบหยุด (Fs) 2.5 KHz โดยมีค่าลดทอนที่แถบผ่านความถี่ (Ap) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (As) 40 dB
- 7.3 บันทึกอันดับของวงจรกรองที่ได้
- 7.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาท์พุท และ ค่าความต่างเฟส ระหว่างสัญญาณอินพุท และ เอาท์พุท

ผลการทดลอง

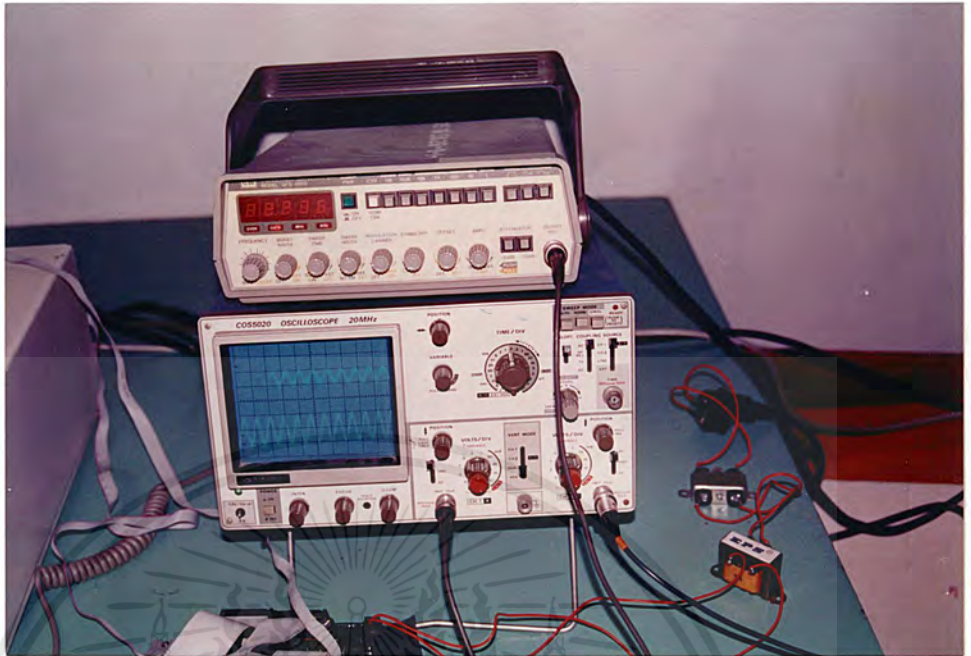
ตารางที่ 12.7 อินพุทสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$
วงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR)

Fp = 2 KHz Fs = 2.5 KHz Ap = 0.5 dB As = 40 dB

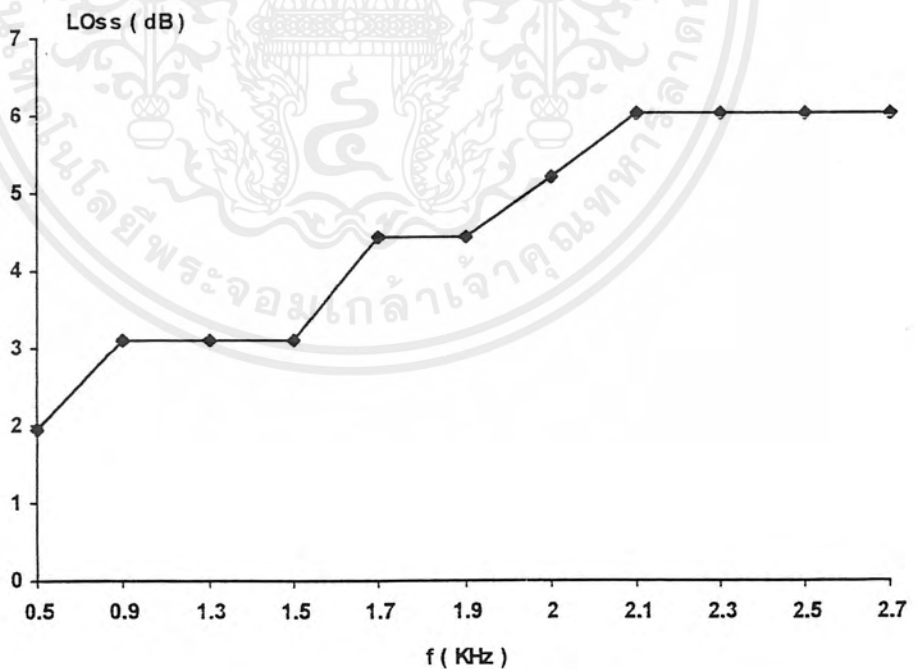
อันดับของวงจรกรองที่ได้ = 24

ความถี่ (KHz)	0.513	0.994	1.328	1.494	1.710	1.941
$V_{OUT} (V_{p-p})$	0.080	0.070	0.070	0.070	0.060	0.060
Loss (dB)	1.938	3.098	3.098	3.098	4.437	4.437
ความต่างเฟส($^{\circ}$)	180	-180	180	-180	0	90

ความถี่ (KHz)	2.042	2.153	2.331	2.510	2.667
$V_{OUT} (V_{p-p})$	0.055	0.050	0.050	0.050	0.050
Loss (dB)	5.193	6.021	6.021	6.021	6.021
ความต่างเฟส($^{\circ}$)	180	0	-90	0	0

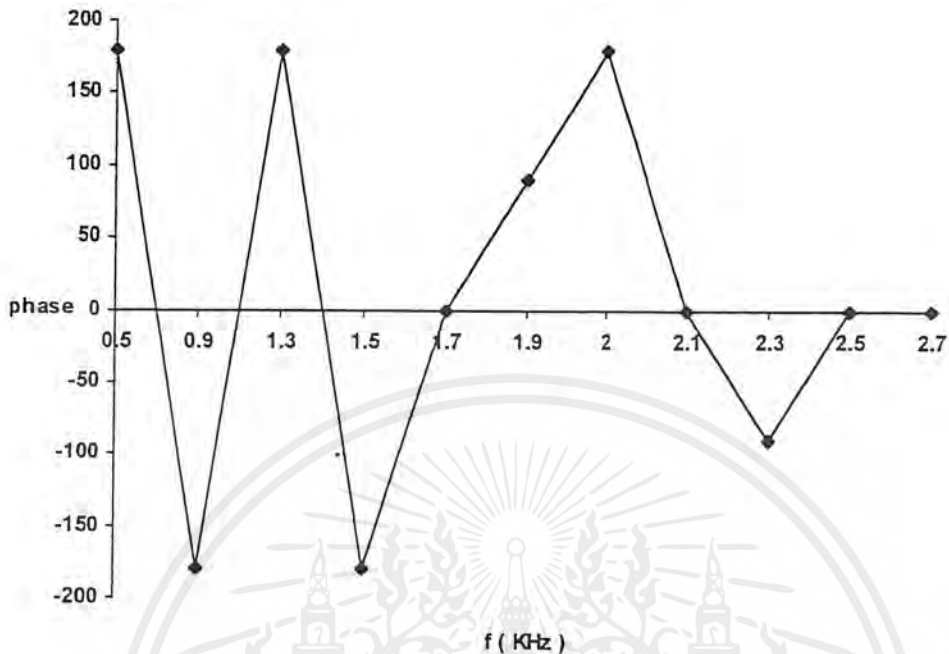


รูปที่ 12.24 แสดงสัญญาณของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR) ที่ความถี่ 2.237 KHz จากการทดลองที่ 7 (50 mV/DIV , 20 uS/DIV)



รูปที่ 12.25 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.26 กราฟแสดง ค่าความต่างเฟส ($^{\circ}$) ของวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR)

การทดลองที่ 8 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบป้อนกลับ (IIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่สูง แบบป้อนกลับ (IIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่

ขั้นตอนการทดลอง

- 8.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด
- 8.2 เลือกใช้วงจรกรองผ่านความถี่สูง แบบป้อนกลับ (IIR) โดยมีค่าความถี่ที่แถบผ่าน (F_p) 2.5 KHz ความถี่แถบหยุด (F_s) 2 KHz โดยมีค่าลดทอนที่แถบผ่านความถี่ (A_p) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (A_s) 40 dB
- 8.3 บันทึกอันดับของวงจรกรองที่ได้
- 8.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟส ระหว่างสัญญาณอินพุท และ เอาพุท

ผลการทดลอง

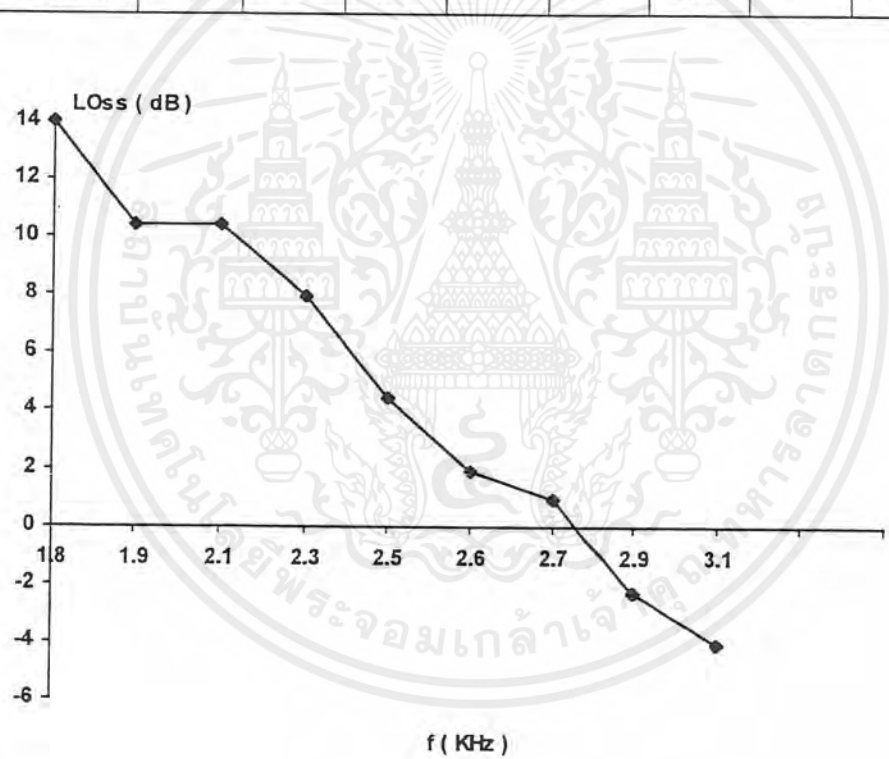
ตารางที่ 12.8 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{p-p}

วงจรกรองผ่านความถี่สูงแบบป้อนกลับ

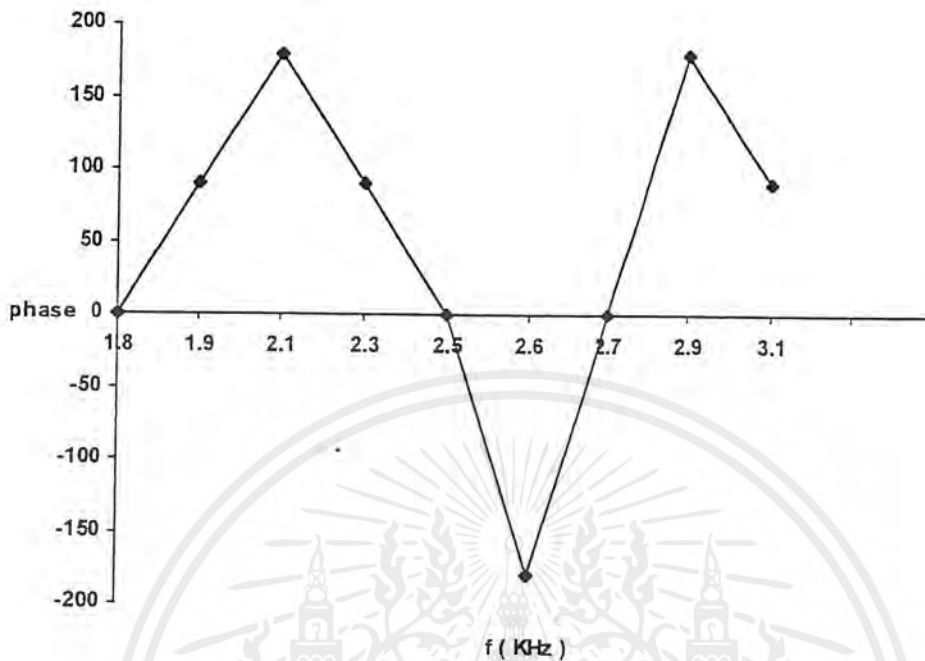
F_p = 2.5 KHz F_s = 2 KHz A_p = 0.5 dB A_s = 40 dB

อันดับของวงจรกรองที่ได้ = 24

ความถี่ (KHz)	1.842	1.915	2.162	2.329	2.490	2.649	2.744	2.972	3.161
V _{OUT} (V _{p-p})	0.020	0.030	0.030	0.040	0.060	0.080	0.090	0.13	0.16
Loss (dB)	13.98	10.45	10.45	7.958	4.437	1.938	0.915	-2.278	-4.082
ความต่างเฟส(°)	0	90	180	90	0	-180	0	180	90



รูปที่ 12.27 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่สูงแบบป้อนกลับ (IIR)



รูปที่ 12.28 กราฟแสดง ค่าความต่างเฟส ($^{\circ}$) ของวงจรกรองผ่านความถี่สูงแบบป้อนกลับ (IIR)

การทดลองที่ 9 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแบบผ่านความถี่ แบบป้อนกลับ (IIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขแบบผ่านความถี่ แบบป้อนกลับ (IIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่

ขั้นตอนการทดลอง

9.1 ป้อนสัญญาณรูปไซน์ขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด

9.2 เลือกใช้วงจรกรองแถบผ่านความถี่ แบบป้อนกลับ (IIR) โดยมีค่าความถี่ที่แถบผ่านต่ำ (Fp1) 1 KHz ความถี่ที่แถบผ่านสูง (Fp2) 2 KHz ความถี่แถบหยุดต่ำ (Fs1) 0.5 KHz ความถี่แถบหยุดสูง (Fs2) 2.5 KHz โดย มีค่าลดทอนที่แถบผ่านความถี่ (Ap) 1 dB ค่าลดทอนที่แถบหยุดความถี่ (As) 35 dB

9.3 บันทึกอันดับของวงจรกรองที่ได้

9.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟส ระหว่างสัญญาณอินพุท และ เอาพุท

ผลการทดลอง

ตารางที่ 12.9 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

วงจรกรองแถบผ่านความถี่แบบป้อนกลับ

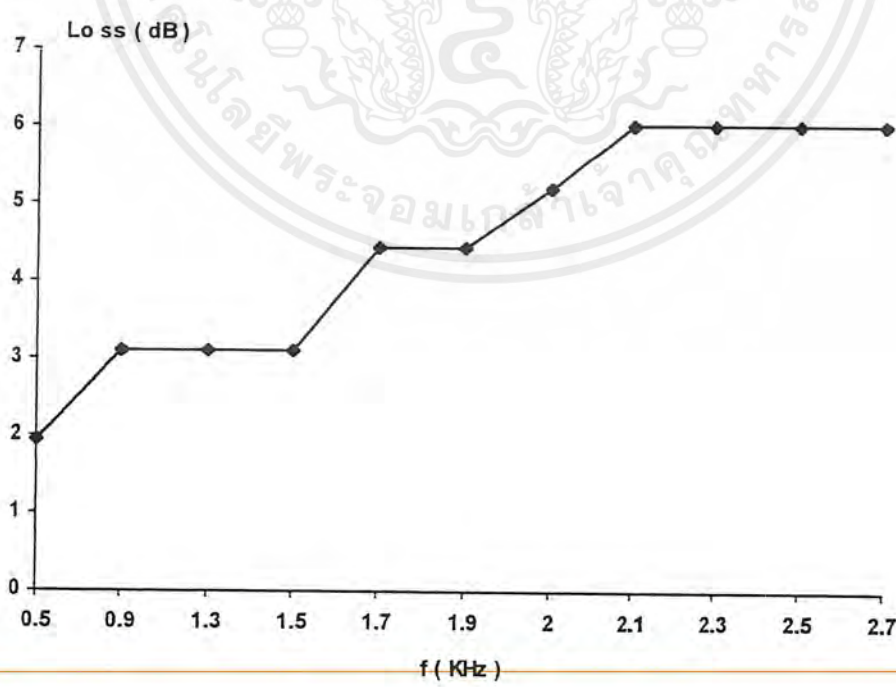
Fp1 = 1 KHz Fs1 = 0.5 KHz Fp2 = 2 KHz Fs2 = 2.5 KHz

Ap = 1 dB As = 35 dB

อันดับของวงจรกรองที่ได้ = 18

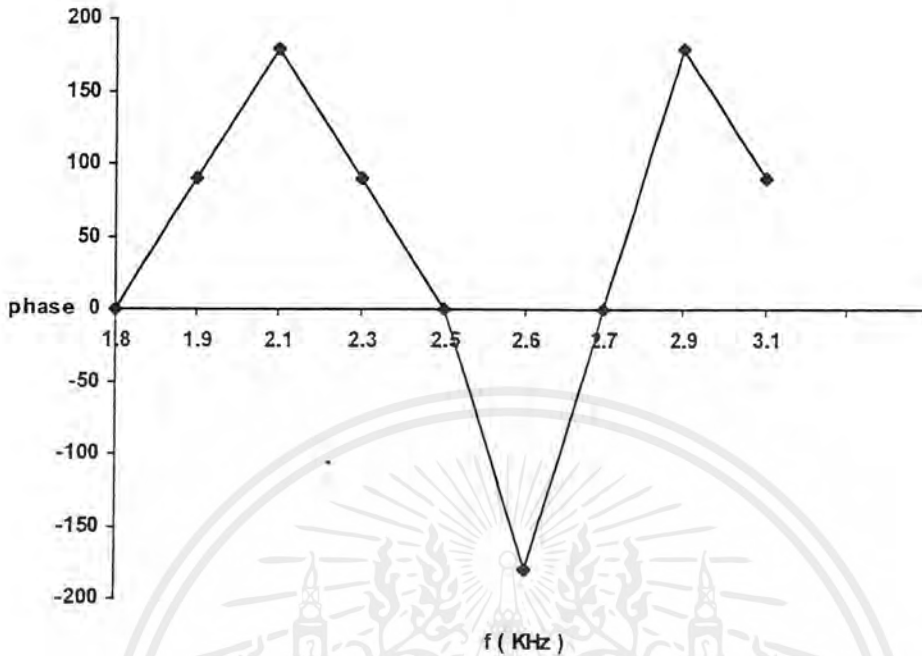
ความถี่ (KHz)	0.949	1.170	1.380	1.562	1.748	1.997	2.172
V _{OUT} (V _{P-P})	0.015	0.030	0.040	0.050	0.060	0.070	0.080
Loss (dB)	16.48	10.45	7.958	6.021	4.437	3.098	1.938
ความต่างเฟส(°)	180	0	0	-90	-180	0	180

ความถี่ (KHz)	2.326	2.415	2.524	2.711	3.119	3.214
V _{OUT} (V _{P-P})	0.1	0.13	0.15	0.20	0.20	0.25
Loss (dB)	0	-2.278	-3.522	-6.020	-6.020	-7.958
ความต่างเฟส(°)	90	0	-90	-100	0	150



รูปที่ 12.29 กราฟแสดง Loss (dB) ของวงจรกรองแถบผ่านความถี่แบบป้อนกลับ (IIR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.30 กราฟแสดง ค่าความต่างเฟส ($^{\circ}$) ของวงจรกรองแถบผ่านความถี่แบบป้อนกลับ (IIR)

การทดลองที่ 10 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขกำลัง
แถบความถี่ แบบป้อนกลับ (IIR)

จุดประสงค์การทดลอง

1. เพื่อทำการทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขกำลังแถบความถี่ แบบ
ป้อนกลับ (IIR) ว่ามีคุณสมบัติตามที่ได้ออกแบบไว้หรือไม่

ขั้นตอนการทดลอง

10.1 ป้อนสัญญาณรูปไซน์ขนาด 0.1 V_{p.p} เข้าทางอินพุตของ AIC บนบอร์ด

10.2 เลือกใช้วงจรกรองกำลังแถบความถี่ แบบป้อนกลับโดยมีค่าความถี่ที่แถบผ่านต่ำ
(Fp1) 0.5 KHz ความถี่ที่แถบผ่านสูง (Fp2) 2.5 KHz ความถี่แถบหยุดต่ำ (Fs1) 1 KHz ความถี่
แถบหยุดสูง (Fs2) 2 KHz โดย มีค่าลดทอนที่แถบผ่านความถี่ (Ap) 1 dB ค่า ลดทอนที่แถบหยุด
ความถี่ (As) 35 dB

10.3 บันทึกอันดับของวงจรกรองที่ได้

10.4 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟส
ระหว่างสัญญาณอินพุท และ เอาพุท

ผลการทดลอง

ตารางที่ 12.10 อินพุตสัญญาณรูปไซน์ขนาด 0.1 V_{P-P}

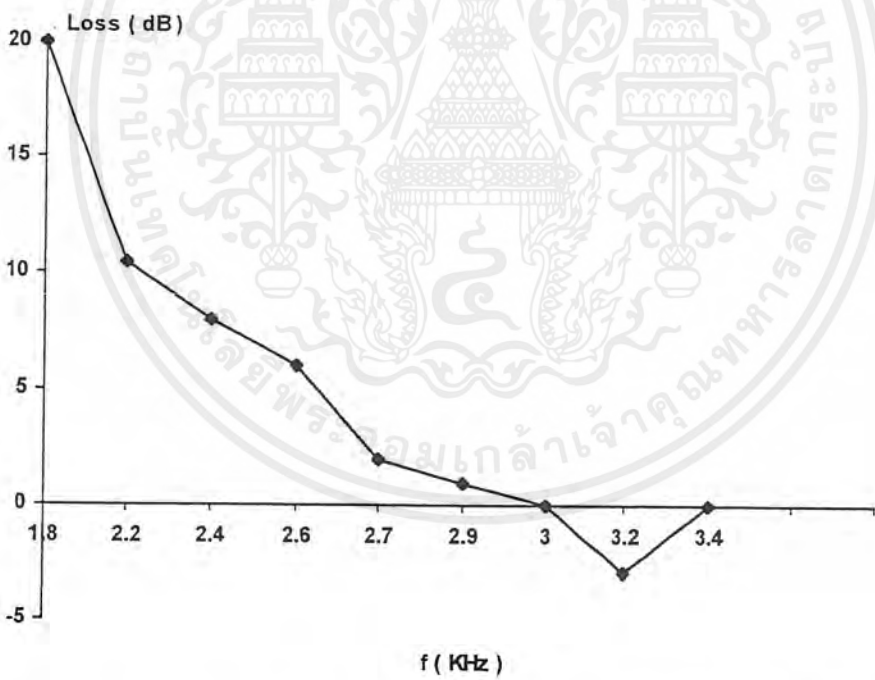
วงจรกรองก่าจัดแถบความถี่แบบป้อนกลับ

Fp1 = 0.5 KHz Fs1 = 1 KHz Fp2 = 2.5 KHz Fs2 = 2 KHz

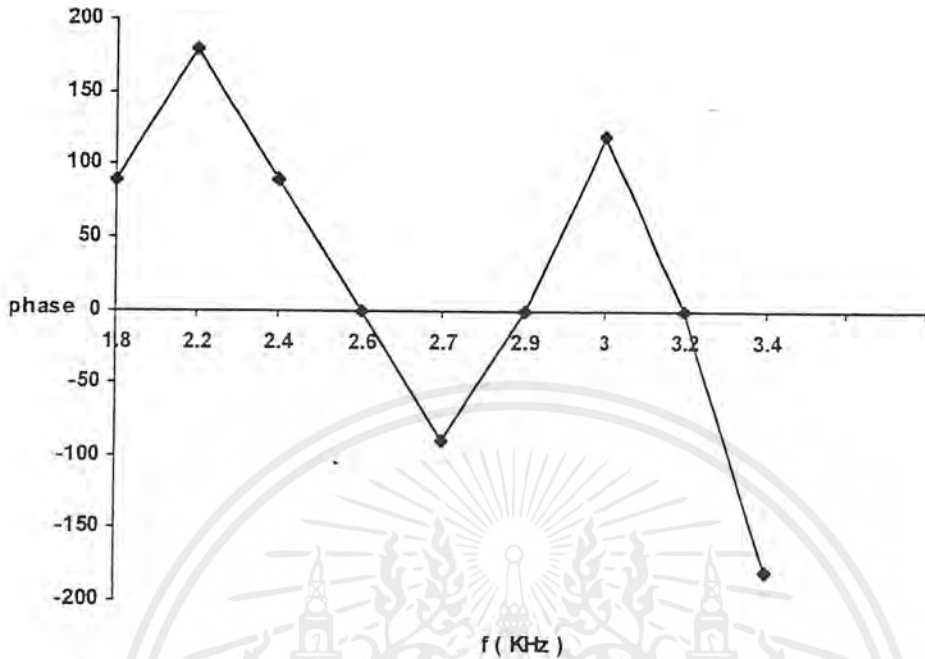
Ap = 1 dB As = 35 dB

อันดับของวงจรกรองที่ได้ = 24

ความถี่ (KHz)	1.856	2.262	2.458	2.580	2.754	2.917	3.037	3.233	3.406
V _{OUT} (V _{P-P})	0.010	0.030	0.040	0.050	0.080	0.09	0.1	0.14	0.1
Loss (dB)	20.00	10.45	7.958	6.021	1.938	0.915	0	-2.922	0
ความต่างเฟส(°)	90	180	90	0	-90	0	120	0	-180
ความต่างเฟส(°)	90	180	90	0	-90	0	120	0	-180



รูปที่ 12.31 กราฟแสดง Loss (dB) ของวงจรกรองก่าจัดแถบความถี่แบบป้อนกลับ (IIR)



รูปที่ 12.32 กราฟแสดงค่าความต่างเฟส ($^{\circ}$) ของวงจรกรองกำเนิดความถี่แบบป้อนกลับ (IIR)

การทดลองที่ 11 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) กับสัญญาณรูปสี่เหลี่ยม

จุดประสงค์การทดลอง

1. เพื่อดูผลของรูปสัญญาณสี่เหลี่ยม ต่อวงจรกรองเชิงเลข

ขั้นตอนการทดลอง

11.1 ป้อนสัญญาณรูปสี่เหลี่ยมขนาด $0.1 V_{p,p}$ เข้าทางอินพุตของ AIC บนบอร์ด

11.2 เลือกใช้วงจรกรองผ่านความถี่ต่ำ โดยมีค่าความถี่ที่แถบผ่าน (F_p) 2 KHz ความถี่แถบหยุด (F_s) 2.5 KHz เลือกใช้ไคเซอร์วินโดว์ โดยมีค่าลดทอนที่แถบผ่านความถี่ (A_p) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (A_s) 40 dB

11.3 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุต และ ค่าความต่างเฟสระหว่างสัญญาณอินพุต และ เอาพุต

11.4 ทำการทดลองตั้งแต่ข้อ 11.1 ถึง 11.3 อีกครั้ง โดยเปลี่ยนเป็นวงจรกรองแบบป้อนกลับ (IIR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

ตารางที่ 12.11.1 อินพุตสัญญาณรูปสี่เหลี่ยมขนาด 0.1 V_{P-P}

วงจรกรองผ่านความถี่ต่ำแบบโคเชอร์วินโดว์

F_p = 2 KHz F_s = 2.5 KHz A_p = 0.5 dB A_s = 40 dB

อันดับของวงจรกรองที่ได้ = 87

ความถี่ (KHz)	0.312	0.543	0.746	1.047	1.193	1.364	1.512
V _{OUT} (V _{P-P})	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Loss (dB)	0	0	0	0	0	0	0
ความต่างเฟส(°)	40	150	90	0	-90	-180	-60

ความถี่ (KHz)	1.723	1.987	2.112	2.342	2.412	2.510	2.645
V _{OUT} (V _{P-P})	0.1	0.08	0.07	0.06	0.04	0.01	0
Loss (dB)	0	1.938	3.098	4.437	7.958	20.00	-
ความต่างเฟส(°)	0	180	90	0	-150	0	-

ตารางที่ 12.11.2 อินพุตสัญญาณรูปสี่เหลี่ยมขนาด 0.1 V_{P-P}

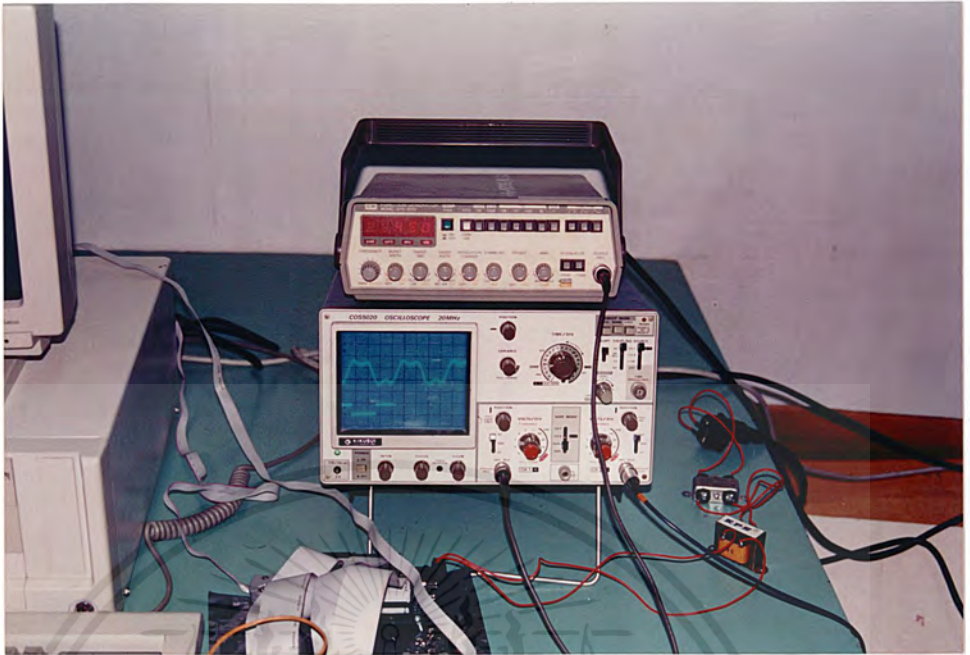
วงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ

F_p = 2 KHz F_s = 2.5 KHz A_p = 0.5 dB A_s = 40 dB

อันดับของวงจรกรองที่ได้ = 14

ความถี่ (KHz)	0.521	0.989	1.128	1.497	1.712	1.957	2.115	2.347	2.515	2.767
V _{OUT} (V _{P-P})	0.080	0.070	0.070	0.070	0.060	0.060	0.060	0.040	0.040	0.040
Loss (dB)	1.938	3.098	3.098	3.098	4.437	4.437	4.437	4.437	4.437	4.437
ความต่างเฟส(°)	180	-180	180	-180	0	150	-180	0	90	0

- สัญญาณเอาพุทที่ได้ในกรณีที่สัญญาณอินพุทเป็นรูปสี่เหลี่ยม ในช่วงที่สัญญาณเอาพุทยังมีขนาดเท่ากับสัญญาณอินพุทจะมีรูปสัญญาณเป็นรูปสี่เหลี่ยม ในช่วงที่สัญญาณเอาพุทมีขนาดลดลงน้อยกว่าขนาดสัญญาณอินพุทจะมีรูปสัญญาณเป็นรูปไซน์

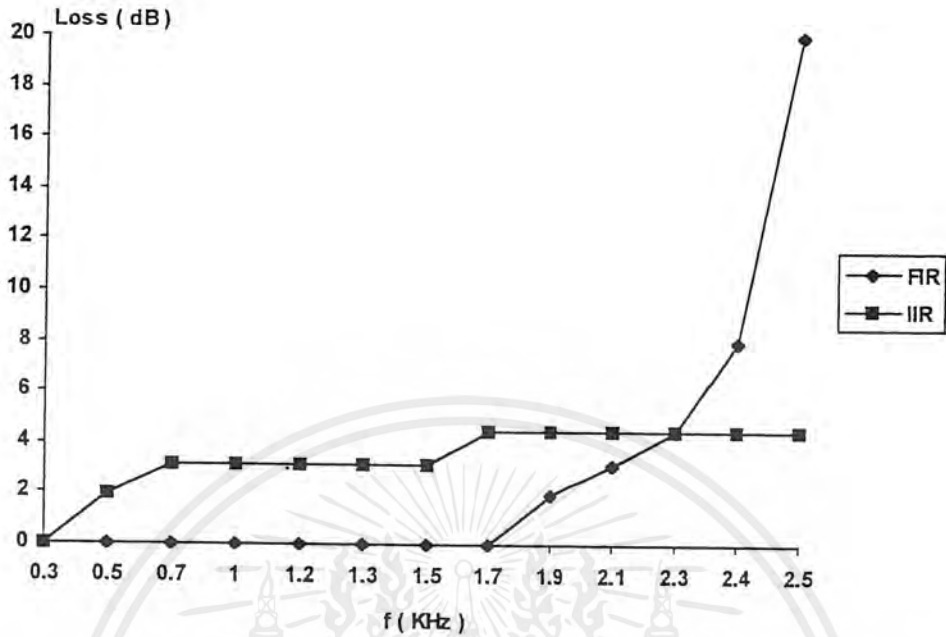


รูปที่ 12.33 แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม ที่ความถี่ 490 Hz จาก
การทดลองที่ 11

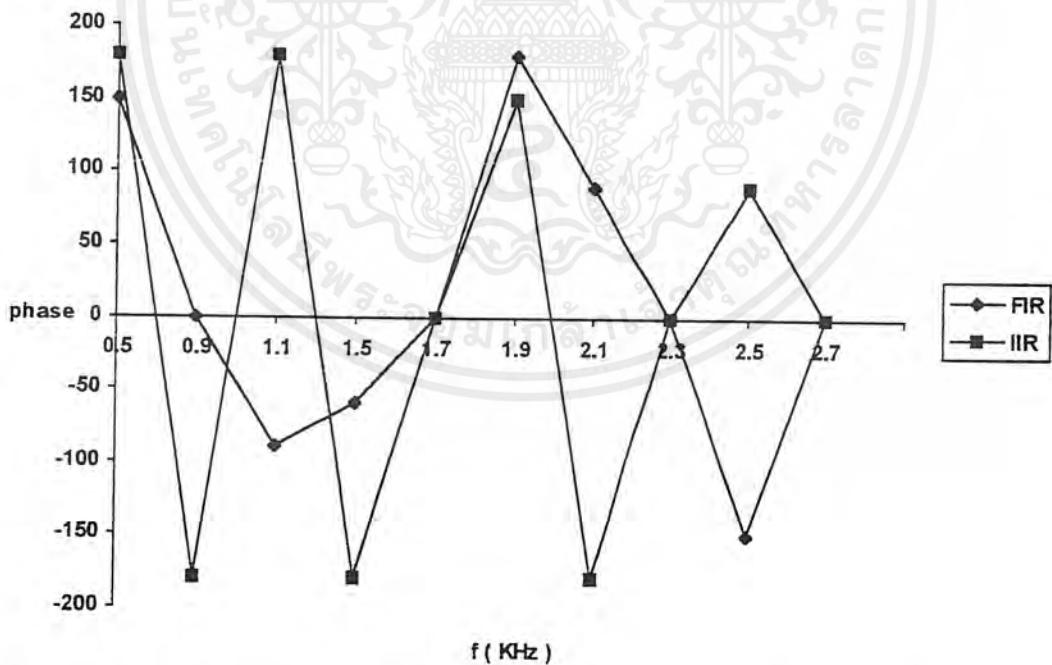


รูปที่ 12.34 แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม ที่ความถี่ 1.727 KHz จาก
การทดลองที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12.35 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม



รูปที่ 12.36 กราฟแสดง ค่าความต่างเฟส (°) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 12 การทดลองเพื่อทดสอบคุณสมบัติทางความถี่ของวงจรกรองเชิงเลขผ่านความถี่ต่ำ แบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) กับสัญญาณรูปสามเหลี่ยม

จุดประสงค์การทดลอง

1. เพื่อคุณผลของรูปสัญญาณสามเหลี่ยม ต่อวงจรกรองเชิงเลข

ขั้นตอนการทดลอง

12.1 ป้อนสัญญาณรูปสามขนาด $0.1 V_{p-p}$ เข้าทางอินพุตของ AIC บนบอร์ด

12.2 เลือกใช้วงจรกรองผ่านความถี่ต่ำ โดยมีค่าความถี่ที่แถบผ่าน (Fp) 2 KHz ความถี่แถบหยุด(Fs) 2.5 KHz เลือกใช้ไคเซอร์วินโดว์ โดย มีค่าลดทอนที่แถบผ่านความถี่ (Ap) 0.5 dB ค่าลดทอนที่แถบหยุดความถี่ (As) 40 dB

12.3 เพิ่มค่าความถี่ของสัญญาณ แล้วบันทึกผลของสัญญาณ เอาพุท และ ค่าความต่างเฟสระหว่างสัญญาณอินพุท และ เอาพุท

12.4 ทำการทดลองตั้งแต่ข้อ 12.1 ถึง 12.3 อีกครั้งโดยเปลี่ยนเป็นวงจรกรองแบบป้อนกลับ (IIR)

ผลการทดลอง

ตารางที่ 12.12.1 อินพุทสัญญาณรูปสามเหลี่ยมขนาด $0.1 V_{p-p}$

วงจรกรองผ่านความถี่ต่ำแบบไคเซอร์วินโดว์

Fp = 2 KHz Fs = 2.5 KHz Ap = 0.5 dB As = 40 dB

อันดับของวงจรกรองที่ได้ = 87

ความถี่ (KHz)	0.543	0.724	1.023	1.213	1.512	1.709	1.925
$V_{OUT} (V_{p-p})$	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Loss (dB)	0	0	0	0	0	0	0
ความต่างเฟส($^{\circ}$)	150	180	0	-90	-180	0	90

ความถี่ (KHz)	2.014	2.112	2.204	2.302	2.415	2.504	2.612
$V_{OUT} (V_{p-p})$	0.09	0.08	0.07	0.06	0.04	0.01	0
Loss (dB)	0.915	1.938	3.098	4.437	7.958	20.00	-
ความต่างเฟส($^{\circ}$)	40	-150	0	180	0	-150	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 12.12.2 อินพุตสัญญาณรูปสามเหลี่ยมขนาด $0.1 V_{P-P}$

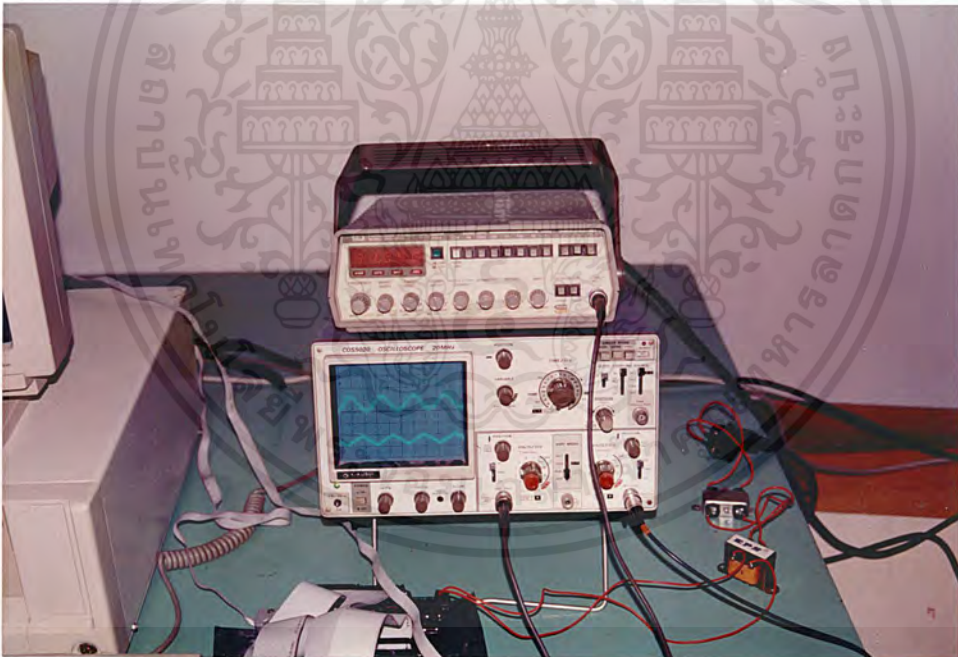
วงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ

$F_p = 2 \text{ KHz}$ $F_s = 2.5 \text{ KHz}$ $A_p = 0.5 \text{ dB}$ $A_s = 40 \text{ dB}$

อันดับของวงจรกรองที่ได้ = 14

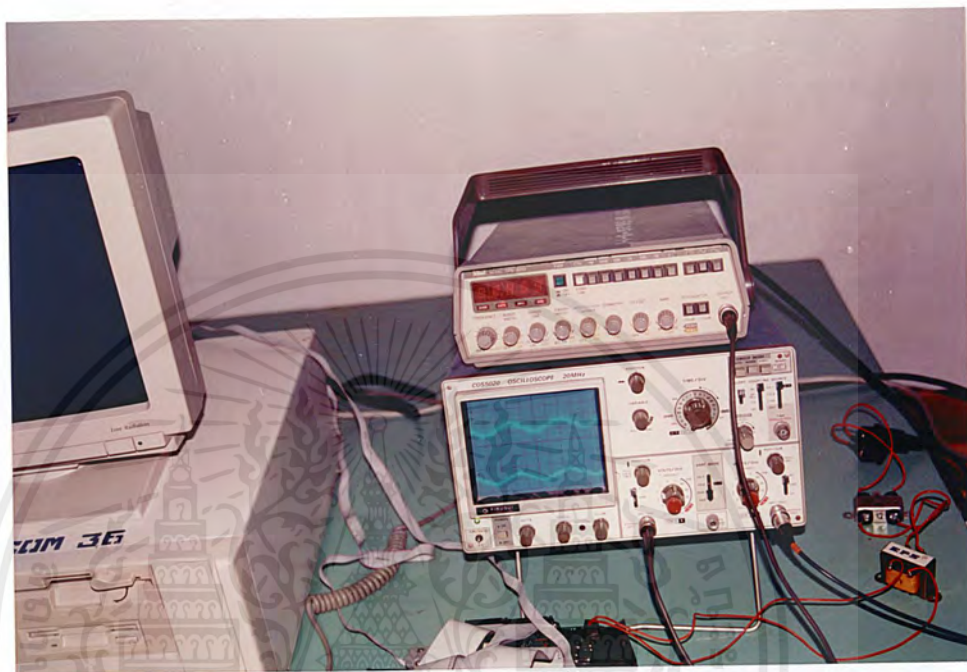
ความถี่ (KHz)	0.524	0.987	1.102	1.324	1.514	1.723	2.045	2.312	2.547	2.75
$V_{OUT} (V_{P-P})$	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.06	0.055	0.05
Loss (dB)	1.938	1.938	3.098	3.098	4.437	4.437	4.437	4.437	5.193	5.19
ความต่างเฟส($^{\circ}$)	180	-180	180	-180	0	180	-180	0	90	90

- สัญญาณเอาพุตที่ได้ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม จะเป็นรูปไซน์ในทุกความถี่

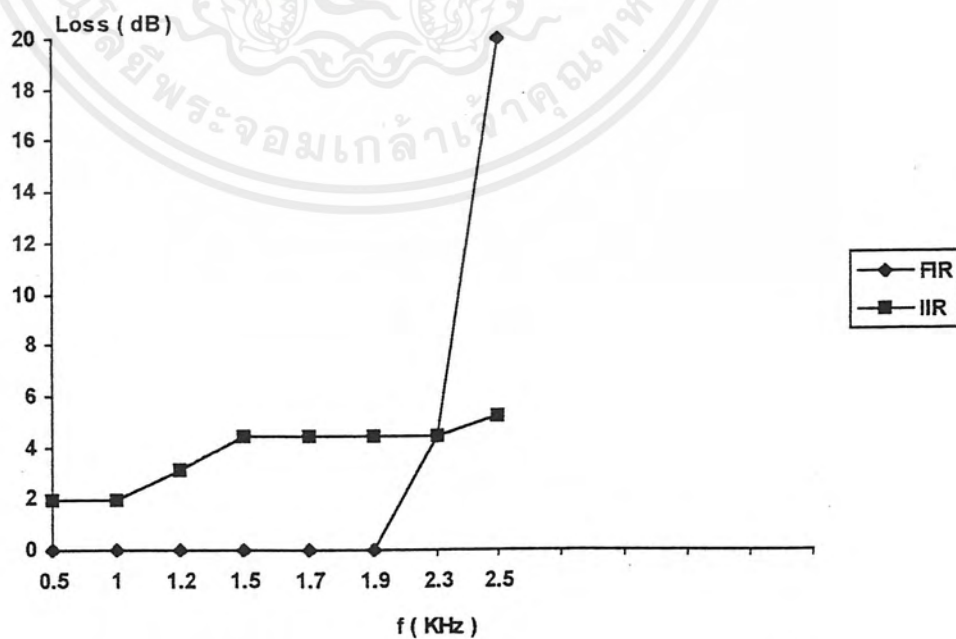


รูปที่ 12.37 แสดงสัญญาณในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม ที่ความถี่ 1.948 KHz
จากการทดลองที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

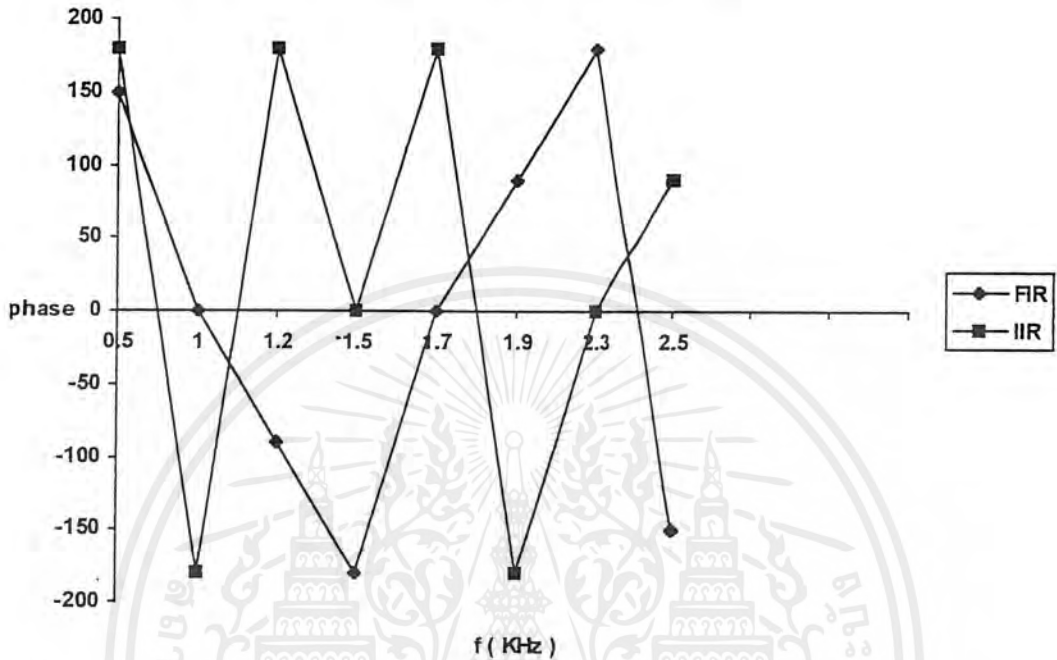


รูปที่ 12.38 แสดงสัญญาณ ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม ที่ความถี่ 2.459 KHz
จากการทดลองที่ 12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 12.39 กราฟแสดง Loss (dB) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และแบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม



รูปที่ 12.40 กราฟแสดง ค่าความต่างเฟส (°) ของวงจรกรองผ่านความถี่ต่ำแบบไม่ป้อนกลับ (FIR) และ แบบป้อนกลับ (IIR) ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม

สรุปผลการทดลอง และวิจารณ์

1. จากการทดลองที่ 1 เพื่อหาความถี่ปฏิบัติงานของ AIC บนบอร์ด จากกราฟ A_v (dB) ของสัญญาณในรูปที่ 12.2 จะเห็นว่าไม่ว่าสัญญาณอินพุตจะเป็นรูปไซน์ สี่เหลี่ยม และ สามเหลี่ยม กราฟ A_v ที่ได้จะเริ่มมีค่าลดลงที่ความถี่ประมาณ 3.1 KHz และ มีค่าต่ำสุดที่ 20 dB ที่ความถี่ประมาณ 3.7 KHz สามารถสรุปได้ว่า ความถี่ปฏิบัติงานของ AIC บนบอร์ด มีค่าประมาณ 3.4 KHz ในกรณีที่ตั้งค่าความถี่สุ่มไว้ 19.2 KHz ซึ่งมีค่าประมาณ 5.6 เท่า ของความถี่ปฏิบัติงาน ส่วนรูปสัญญาณเอาพุตที่ได้ในกรณีที่สัญญาณอินพุตเป็นรูปสี่เหลี่ยม ในช่วงความถี่ที่ค่าขนาดของสัญญาณยังไม่ลดลง (น้อยกว่า 3 KHz) จะมีรูปสัญญาณเป็นรูปสี่เหลี่ยม ในช่วงความถี่ที่ค่าขนาดของสัญญาณมีขนาดลดลง (มากกว่า 3 KHz) จะมีรูปสัญญาณเป็นรูปไซน์ ที่เป็นเช่นนี้เพราะสัญญาณสี่เหลี่ยมเกิดจากสัญญาณรูปไซน์หลายความถี่รวมกัน ในกรณีที่สัญญาณอินพุตเป็นรูปสามเหลี่ยม สัญญาณเอาพุตที่ได้จะเป็นรูปไซน์ทุกช่วงความถี่ เพราะสัญญาณรูปสามเหลี่ยมและสัญญาณรูปไซน์มีลักษณะคล้ายกัน

2. จากการทดลองที่ 2 , 4, 5 และ 6 จากกราฟรูปที่ 12.8, 12.18, 12.20 และ 12.22 แสดงให้เห็นว่าวงจรกรองแบบไม่ป้อนกลับ (FIR) ทุกชนิดไม่ว่าจะเป็น แบบผ่านความถี่ต่ำ แบบผ่านความถี่สูง แบบแถบผ่านความถี่ และ แบบกั้นแถบความถี่ สามารถทำงานได้ใกล้เคียงกับที่ออกแบบไว้ คือ ผ่านความถี่ในช่วงที่กำหนดไว้ และ ไม่ผ่านความถี่ในช่วงที่ไม่ได้กำหนดเอาไว้ได้ โดยแบบโคเชอร์วินโดว์ จะมีค่าใกล้เคียงอุดมคติมากที่สุดความแตกต่างระหว่างแต่ละวินโดว์อยู่ที่ความชันในช่วงแถบเปลี่ยนความถี่ (Transition Band) และ ค่าลดทอนที่แถบผ่าน กับ แถบหยุด โดยวินโดว์แบบสี่เหลี่ยมจะมีความชันน้อยที่สุด และ วินโดว์แบบโคเชอร์จะมีความชันมากที่สุด (ใกล้เคียงอุดมคติมากกว่า) แต่โดยรวมแล้ววินโดว์ทั้ง 5 แบบ สามารถทำงานได้ตามที่กำหนดไว้ได้

3. จากการทดลองที่ 3 และ จากกราฟรูปที่ 12.10, 12.12, 12.14 และ 12.16 ซึ่งเป็นการทดลองเปรียบเทียบผลของอันดับของวงจรกรองที่มีต่อวงจรกรองเชิงเลข แบบไม่ป้อนกลับ (FIR) สามารถสรุปได้ว่า ที่อันดับของวงจรกรองมาก วงจรกรองมีแถบเปลี่ยนความถี่ที่น้อยกว่า และ มีค่าลดทอนที่แถบความถี่ผ่าน กับ แถบความถี่หยุดได้ใกล้เคียงกับที่กำหนดไว้ ที่อันดับของวงจรกรองน้อย วงจรกรองมีแถบเปลี่ยนความถี่ที่มากกว่า แต่ค่าลดทอนที่แถบความถี่ผ่าน กับ แถบความถี่หยุดมีค่าแตกต่างกับค่าที่กำหนดไว้มากกว่ากรณีที่อันดับของวงจรกรองมากซึ่งผลของอันดับของวงจรกรองต่อวงจรกรองจะมีผลเหมือนกันในทุกแบบวินโดว์

4. จากการทดลองที่ 7 และ กราฟรูปที่ 12.25 เป็นการทดลองวงจรกรองผ่านความถี่ต่ำแบบป้อนกลับ (IIR) รวมถึงการทดลองที่ 8 กราฟรูปที่ 12.27 ซึ่งเป็นการทดลองวงจรกรองผ่านเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่สูงแบบป้อนกลับ (IIR) และ การทดลองที่ 9 กราฟรูปที่ 12.29 ผลตอบสนองที่ได้ถึงแม้ว่าจะสามารถลดทอนขนาดสัญญาณในช่วงแถบความถี่ได้แต่ก็ไม่ค่อยจะดีนัก เมื่อเปรียบเทียบกับวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR)

5. จากการทดลองที่ 10 กราฟรูปที่ 12.31 ซึ่งเป็นการทดลองวงจรกรองก้ำจัดแถบความถี่แบบป้อนกลับ (IIR) ผลตอบสนองที่ได้จะกลับมีลักษณะคล้ายกับวงจรกรองผ่านความถี่สูงมากกว่า รวมทั้งผลตอบสนองที่ได้ยังไม่ค่อยดีนัก เมื่อเปรียบเทียบกับวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR)

6. อันดับของวงจรกรองเชิงเลขแบบป้อนกลับ (IIR) (ประมาณ 15-20) จะมีค่าน้อยกว่า อันดับของวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) (ประมาณ 70-90) เมื่อเปรียบเทียบกับวงจรกรองชนิดเดียวกัน ที่มีการกำหนดค่าความถี่ที่จุดต่าง ๆ และ ค่าลดทอน ที่เหมือนกัน

7. จากข้อ 4 และ 5 ที่กล่าวมาข้างต้น พบว่าวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (IIR) ที่ออกแบบไว้มีคุณสมบัติทางความถี่ที่ไม่ดีนัก เมื่อเปรียบเทียบกับวงจรกรองแบบป้อนกลับ (FIR) ซึ่งตามทฤษฎีแล้ววงจรกรองเชิงเลขแบบป้อนกลับ (IIR) ควรจะมีคุณสมบัติทางความถี่ที่ดีกว่าวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) ซึ่งสาเหตุอาจเกิดมาจาก

7.1 TMS320C50 เป็นไมโครโปรเซสเซอร์ที่มีสถาปัตยกรรมที่ใช้การคำนวณแบบจุดทศนิยมคงที่ (Fixed Point Arithmetic) ซึ่งเหมาะกับการทำเป็นวงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) มากกว่า การทำเป็นวงจรกรองเชิงเลขแบบป้อนกลับ (IIR) ซึ่งใช้การคำนวณแบบอิงดรรชนี (Floating Point Arithmetic) ควรจะใช้ไมโครโปรเซสเซอร์ในตระกูล TMS320C3x และ TMS320C4x ที่เป็นไมโครโปรเซสเซอร์ที่มีสถาปัตยกรรมที่ใช้การคำนวณแบบอิงดรรชนีในการทำเป็นวงจรกรองเชิงเลขแบบป้อนกลับ (IIR)

7.2 การที่วงจรกรองเชิงเลขแบบไม่ป้อนกลับ (FIR) เหมาะสมกับการใช้กับไมโครโปรเซสเซอร์ที่มีสถาปัตยกรรมที่ใช้การคำนวณแบบจุดทศนิยมคงที่ เพราะ วงจรกรองเชิงเลขแบบไม่ป้อนกลับ มีค่าสัมประสิทธิ์ ที่มักจะอยู่ในช่วงระหว่าง 1 ถึง -1 ซึ่งสามารถใช้เลขฐานสองในรูปแบบ Q15 แทนค่าสัมประสิทธิ์ซึ่งเป็นรูปแบบที่ TMS320C50 สามารถนำไปใช้ในการคำนวณได้เลย โดยที่ไม่ต้องมีการชิพทรีนิจิตีส์ในการคำนวณ ทำให้ค่าสัมประสิทธิ์ที่อยู่ในรูปเลขฐานสองแบบ Q15 สามารถแทนค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบไม่ป้อนกลับได้ใกล้เคียงกับค่าที่แท้จริงได้

7.3 ในกรณีวงจรกรองเชิงเลขแบบป้อนกลับ (IIR) เหมาะสมกับการใช้กับไมโครโปรเซสเซอร์ที่มีสถาปัตยกรรมที่ใช้การคำนวณแบบอิงดรรชนี เพราะ วงจรกรองเชิงเลขแบบป้อนกลับ มีค่าสัมประสิทธิ์ ที่อยู่ในช่วงระหว่าง 3 ถึง -3 ซึ่งสามารถใช้เลขฐานสองในรูปแบบเอกสาร์นี้เป็นเอกสาร์ที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ Q13 แทนค่าสัมประสิทธิ์ได้ แต่ในการใช้ TMS320C50 ในการคำนวณได้ แต่ต้องมีการชีพท์ในชิพท์รีจิสเตอร์ทั้ง 3 ในการคำนวณ ดังที่ได้กล่าวไว้ในบทที่ 11 ทำให้การคำนวณอาจเกิดการคลาดเคลื่อนได้ เนื่องจากค่าสัมประสิทธิ์จะถูกผลของการตัดปลาย (truncate) และการปัดเศษ (rounding) ซึ่งเป็นผลจากการใช้เลขฐานสองที่มีความจำกัด (finite wordlength)

7.4 ในการทำวงจรกรองเชิงเลขแบบป้อนกลับ (IIR) โดยใช้ไมโครโปรเซสเซอร์ที่มีสถาปัตยกรรมการคำนวณแบบจุดทศนิยมคงที่ นอกจากใช้เลขฐานสองรูปแบบ Qxx ที่สามารถแทนค่าที่สามารถแทนค่าสัมประสิทธิ์ของวงจรกรองเชิงเลขแบบป้อนกลับแล้ว ยังสามารถทำได้โดยใช้วิธีการใช้เลขฐานสองรูปแบบ Q15 แทนค่าสัมประสิทธิ์ แต่ค่าสัมประสิทธิ์ก่อนที่จะแปลงเป็นรูปแบบ Q15 จะต้องมีการสเกลค่าสัมประสิทธิ์ซะก่อน ซึ่งการสเกลค่าสัมประสิทธิ์จะต้องใช้การคำนวณทางสถิติโดยการเปรียบเทียบคุณสมบัติทางความถี่ของวงจรกรองในทางทฤษฎีกับ คุณสมบัติทางความถี่ของวงจรกรองในทางปฏิบัติ ซึ่งมีขั้นตอนที่ยุ่งยากมาก (ผู้ที่สนใจสามารถหารายละเอียดเพิ่มเติมได้ในหนังสือเล่มที่ 6 ในภาคผนวก)

8. จากผลการทดลองทั้งหมดผลตอบสนองเฟสมีค่าไม่เป็นเชิงเส้น โดยจะมีค่าเปลี่ยนไปมาระหว่าง 180° ถึง -180°

9. ค่าลดทอนที่แถบความถี่หยุดจากการทดลองทั้งหมดจะมีวัดได้ค่าสูงสุดประมาณ 20 dB (ซึ่งค่าที่ตั้งเอาไว้อยู่ประมาณ 30 ถึง 40 dB) ทั้งนี้เพราะในการวัดผลไม่สามารถวัดผลต่อไปได้เนื่องจากรูปสัญญาณเอาพุทไม่ชัดเจน



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่โครงการนี้สามารถสำเร็จลุล่วงไปด้วย ขอขอบพระคุณ ศ.ดร. วัลลภ สุระกำพลธ อาจารย์ที่ปรึกษาที่ให้คำแนะนำที่ดีในการทำงานทำให้โครงการนี้เป็นรูปเป็นร่างออกมา ขอขอบพระคุณ พี่ๆ ที่ให้ความรู้เพิ่มเติมในการออกแบบโปรแกรม เพื่อนๆ ที่ช่วยกันคนละนิดคนละหน่อยในการเสนอความคิดในการออกแบบและเขียนโปรแกรม น้องๆ ที่ช่วยเป็นกำลังใจในการสู้กับงาน ขอขอบคุณ คุณพ่อ-แม่ รุ่งน้อมที่เอื้อเฟื้อสถานที่ในการทำงานและ เครื่องพิมพ์ในการพิมพ์งานให้ออกมาทันเวลาได้ ขอขอบคุณทุกคนที่มีส่วนในการทำให้โครงการที่สำคัญนี้สำเร็จและได้ผลออกมาอย่างดี คณะผู้จัดทำขอขอบคุณอย่างมาก




```

mmregs
.ds 0300h
temp .word 0 ; Temp for receive data from com

```

```

.ds 0301h
order_r .word 0 ; Order of filter receive from com
; ( Temp for receive data )

```

```

.ds 0302h
datap1 .word 0 ; Begin of data page1 from com

```

```

.ds 0372h
datap2 .word 0 ; Begin of data page2 from com

```

```

.ds 03e2h
datap3 .word 0 ; Begin of data page3 from com

```

```

.ds 0452h
coeffh .word 0 ; Begin of coefficient h(n)

```

```

.ds 0f00h
TA .word 18
RA .word 18 ; This set up of AIC registers give
; a sampling freq of 19.2 kHz

```

```

TB .word 15

```

```

RB .word 15

```

```

AIC_CTR .word 08h

```

```

OUTPUT .word 0

```

```

XN .word 0,0,0,0,0,0,0,0,0 ; 100 data locations for 100

```

```

XN1 .word 0,0,0,0,0,0,0,0,0 ; stage delay line

```

```

XN2 .word 0,0,0,0,0,0,0,0,0

```

```

XN3 .word 0,0,0,0,0,0,0,0,0

```

```

XN4 .word 0,0,0,0,0,0,0,0,0

```

```

XN5 .word 0,0,0,0,0,0,0,0,0

```

```

XN6 .word 0,0,0,0,0,0,0,0,0

```

```

XN7 .word 0,0,0,0,0,0,0,0,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XN8    .word  0,0,0,0,0,0,0,0,0
XN9    .word  0,0,0,0,0,0,0,0,0
XNLAST .word  0

ORDER  .word  0          ; real order for filt routine

        .ps    0080ah
rint:   B      RECEIVE    ; Recieve subroutine of AIC
xint:   B      TRANSMIT    ; Tranamit subroutine of AIC

        .ds    02b00h
h0      .word  0          ; coefficient h(n) in program memory

*****
*TMS320C50 INITIALIZATION *
*This rInine initializes the C50 registers,internal RAM *
*****

        .ps    0a00h
        .entry
INIT    SETC    INTM      ; Disable Interrupts

        LDP     #00h      ; Initialize Data Pointer
        OPL     #0834h,PMST ; Configure S/A RAM in data memory
        LACC     #0000h
        MAR     *,AR7
        LAR     AR7,#0300h
        RPT     #200h
        SACL     *+      ; Initialize B1 blocks = 0000h
        SAMM     PDWSR    ; for 0 waits in pgm & data memory

        ; Set Wait State Control Register
        lacc     #0ch
        SAMM     CWSR
        ; Increase wait state in I/O Port ( 7 waits state for interface com )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lacc #0aaaah
SMM IOWSR
```

*Start of main program *

```
ldp #temp
```

```
lar ar3,#0301h ; AR3 point to order_t
lar ar4,#0302h ; AR4 point to data page1
lar ar5,#0372h ; AR5 point to data page2
lar ar6,#03e2h ; AR6 point to data page3
lar ar7,#0452h ; AR7 point to coeffh
```

INORDER:

```
mar *,ar3
call INDATA ; Receive order of filter from com

lacc 01h
lar ar3,#0f6ah
sac1 * ; 'Order_t' --> 'Order'
```

```
lacc 01h
```

READ3P:

```
push
```

```
mar *,ar4
call INDATA ; Receive data page1 from com
```

```
mar *,ar5
call INDATA ; Receive data page2 from com
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

mar    *,ar6
call   INDATA      ; Receive data page.3 from com

pop

sub     #01h

bcnd    READ3P,NEQ    ; Receive until last data

```

```

*****
*This routine is sum data 3 pages from com to make coefficeint h (n) *
*****

```

```

lar     ar4,#0302h
lar     ar5,#0372h
lar     ar6,#03e2h
lar     ar7,#0452h

lacc    01h
SUMMAIN:
push

call    SUM

pop

sub     #01h

bcnd    SUMMAIN,NEQ    ; Sum 3 pages data until get last coefficient

```

```

*****
* This routine is move coefficient from data memory to program memory *
* data memory [ 0452h to 04b6 ] —> program memory B0 [ 2b00h to 2b64h ] *
*****

```

MOVE:

```

lacc    #02b00h

samm    BMAR      ; BMAR -> destination in program memory

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lar    AR7,#0452h    ; AR7 -> source in data memory
mar    *,AR7
rpt    #063h         ; Move 100 data to program memory space
bldp   *+

```

```

SETC   INTM
LDP    #0
OPL    #0834h,PMST
LACC   #0
SAMM   CWSR
SAMM   PDWSR        ; Set 0 wait state for AIC
SETC   SXM          ; SXM MUST BE SET
SPLK   #022h,IMR    ; This turns on receive interrupt only

CALL   AICINIT
SPLK   #12h,IMR
CLRC   OVM
SPM    0
CLRC   INTM

```

WAIT: ; a main program would go here

```
B      WAIT
```

```
ret
```

; END of main program

RECEIVE:

```
ldp    #temp
lacc   #0f06h        ; Load with 'address of XN'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

add    01h          ; add with 'order'
sac1   00h

lar    AR0,00h      ; load AR0 with address of last delay element

LDP    #XN

CLRC   INTM         ; ENABLE INTERRUPTS FOR DEBUGGING DSK PURPOSES
LAMM   DRR          ; load accumulator with word received from AIC
SACL   XN           ; store the value of received word to a variable

ZAP                      ; ZERO ACC AND PRODUCT REGISTERS
MAR    *.AR0         ; AR0 is current AR register

rpt    0f6ah
filt:
MACD   #h0,*-        ; repeat the next line 'order' times through
                        ; the complete coeff table.

APAC                      ; Accumulate last product

SACH   OUTPUT.1      ; ACC -> OUTPUT Get rid of xtra sign bit.
LACC   OUTPUT        ; OUTPUT -> ACC
SFL                      ; OUTPUT*2
AND     #0ffch       ; Last two must be zero for AIC
SAMM   DXR           ; write output word to transmit register
RETE

```

TRANSMIT:

RETE

```

* DESCRIPTION: This routine initializes the 'C50 serial port      *
*               and the TLC320C40's (AIC) TA,RA,TB,RB and       *
*               control registers                                *

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AICINIT:

```
SPLK #20h,TCR      ; To generate 10 MHz from Tout
SPLK #01h,PRD      ; for AIC master clock
MAR  *,AR0
LACC #0008h        ; Non continuous mode
SACL SPC           ; FSX as input
LACC #00c8h        ; 16 bit words
SACL SPC
LACC #080h         ; Pulse AIC reset by setting it low
SACH DXR
SACL GREG
LAR  AR0,#0FFFFh
RPT  #10000        ; and taking it high after 10000 cycles
LACC *,0,AR0       ; (.5ms at 50ns)
SACH GREG
;-----
LDP  #TA
SETC SXM
LACC TA.9          ; Initialize TA and RA register
ADD  RA.2
CALL AIC_2ND
;-----
LDP  #TB
LACC TB.9          ; Initialize TB and RB register
ADD  RB.2
ADD  #02h
CALL AIC_2ND
;-----
LDP  #AIC_CTR
LACC AIC_CTR.2     ; Initialize control register
ADD  #03h
CALL AIC_2ND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

AIC_2ND:

```
LDP    #0          ; Data page point is 0 (MM regs)
SACH   DXR          ; send ACChi 00
CLRC   INTM         ; enable interrupts
IDLE                    ; wait for interrupt
ADD    #6h,15       ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
SACH   DXR          ; send ACChi to initiate secondary protocol
IDLE                    ; wait for interrupt
SACL   DXR          ; send the T register data
IDLE                    ; wait for interrupt
LACL   #0           ; clear ACClo
SACL   DXR          ; send another to make sure 1st word got sent
IDLE                    ; wait for interrupt
SETC   INTM
RET
```

```
=====
; in data from com port
=====
```

INDATA:

GETZERO:

```
in     00h,PA0      ; 'temp' = data from com port
lacc   #0ffh
and    00h
bcnd   GETZERO,EQ   ; repeat until 'temp' <> 0

;store 'temp' to 'data'
bldd   #00h,*
mar    *+
```

WAITZERO:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

in    00h,PA0      ; 'temp' = data from com port
lacc  #0ffh
and   00h
bcnd  WAITZERO,NEQ ; repeat until 'temp' = 0

ret

```

```

;This routine for summain

```

SUM:

```

mar   *,ar4
lacc  *,5
sub   #20h
mar   *+,ar5
add   *
sub   #01h
rpt   #04h
sfl
mar   *+,ar6
add   *
sub   #01h
mar   *+,ar7
sac1  *+

```

```

ret

```

```

.end

```



```

        .mmregs
        .ds    0300h

temp    .word  0          ; Temp for receive data from com

        .ds    0301h
state_t .word  0          ; State of filter receive from com
        : ( Temp for receive data )

        .ds    0302h
datap1  .word  0          ; Begin of data page1 from com
        .ds    0372h
datap2  .word  0          ; Begin of data page2 from com
        .ds    03e2h
datap3  .word  0          ; Begin of data page3 from com

        .ds    0452h
coeffAB .word  0          ; Begin of coefficient a(n) , b(n)

        .ds    0f00h
TA      .word  18
RA      .word  18          ; This set up of AIC registers give
        : a sampling freq of 19.2 kHz

TB      .word  15
RB      .word  15
AIC_CTR .word  08h
OUTPUT  .word  0

XN      .word  0

state   .word  0          ; real state for filt routine

        .ds    02000h
d(n)    .word  0

```

```

        .ds    02100h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AB .word 0 ; coefficient AB in program memory

.ps 0080ah

rint: B RECEIVE ; Recieve subroutine of AIC

xint: B TRANSMIT ; Tranamit subroutine of AIC

*TMS320C50 INITIALIZATION *

*This rInine initializes the C50 registers,internal RAM *

.ps 0a00h

.entry

INIT SETC INTM ; Disable Interrupts

LDP #00h ; Initialize Data Pointer

OPL #0834h,PMST ; Configure S/A RAM in data memory

LACC #0000h

MAR *,AR7

LAR AR7,#0300h

RPT #200h

SACL *+ ; Initialize B1 blocks = 0000h

SAMM PDWSR ; for 0 waits in pgm & data memory

; Set Wait State Control Register

lacc #0ch

SAMM CWSR

; Increase wait state in I/O Port (7 waits state)

lacc #0aaaah

SAMM IOWSR

*Start of main program *

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ldp #temp

lar ar3,#0301h ; AR3 point to state_t
lar ar4,#0302h ; AR4 point to data page1
lar ar5,#0372h ; AR5 point to data page2
lar ar6,#03e2h ; AR6 point to data page3
lar ar7,#0452h ; AR7 point to coeffAB

INORDER:

mar *,ar3
call INDATA ; Receive state of filter from com

lacc 01h
lar ar3,#0f07h
sac1 * ; 'state_r' --> 'state'

clrc sxm
lacc 01h.2
add 01h

READ3P:

push

mar *,ar4
call INDATA ; Receive data page1 from com

mar *,ar5
call INDATA ; Receive data page2 from com

mar *,ar6
call INDATA ; Receive data page3 from com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

pop
sub    #01h
bcnd   READ3P,NEQ    ; Receive until last data

```

*This routine is sum data 3 pages from com to make coefficeint a(n) , b(n) *

```

lar    ar4,#0302h
lar    ar5,#0372h
lar    ar6,#03e2h
lar    ar7,#0452h

```

```

clrc   sxm
lacc   01h,2
add    01h

```

SUMMAIN:

```

push

call   SUM

pop
sub    #01h
bcnd   SUMMAIN,NEQ    ; Sum 3 pages data until get last coefficient

```

* This routine is move coefficient from data memory to program memory *

* data memory [0452h to 04b6] ----> program memory B0 [2100h to 2164h] *

MOVE:

```

lacc   #02100h
samm   BMAR    ; BMAR -> destination in program memory

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lar    AR7,#0452h    ; AR7 -> source in data memory
mar    *,AR7
rpt    #063h         ; Move 100 data to program memory space
bldp   *+

```

;

```

ldp    #AB
mar    *,ar0
lacc   #0000h
lar    ar0,#02000h
rpt    #0feh
sac1   *+           ; Initial d(n)

SETC   INTM
LDP    #0
OPL    #0834h,PMST
LACC   #0
SAMM   CWSR
SAMM   PDWSR
SETC   SXM          ; SXM MUST BE SET
SPLK   #022h,IMR    ; This turns on receive interrupt only

CALL   AICINIT
SPLK   #12h,IMR
CLRC   OVM

SPM    0

CLRC   INTM

```

WAIT: ; a main program would go here

B WAIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ret

; END of main program

RECEIVE:

ldp #XN

lacc 07h,1

add 07h

sub #01h

add #0200h,4

ldp #temp

sac1 00h

lar ar1,00h ; AR1 point to d(n-2) of 1st

lar ar0,#02100h ; AR0 point to -b2 of 1st

lacc 01h

sub #01h

ldp #00h

samm BRCR ; Setup count

LDP #XN

CLRC INTM ; ENABLE INTERRUPTS FOR DEBUGGING DSK PURPOSES

LAMM DRR ; load accumulator with word received from AIC

SACL XN ; store the value of received word to a variable

lacc XN,14

spm 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

SPLK  #2,INDX      ; Setup index register
ZPR                      ; Clear P register
MAR   *,AR1

RPTB  0a93h        ; repeat for N biquads
iii:  LT   *-,AR0      ; T = d(n-2)
      MPYA *,AR1        ; Acc = x(n), P = d(n-2)a2
      LTA  *-,AR0      ; Acc += d(n-2)a2, T = d(n-1)
      MPY  *+          ; P = d(n-1)a1
      LTA  *+,AR1      ; Acc += d(n-1)a1, T = b2
      SACH *0+,1       ; Save d(n)
      MPY  *-          ; P = d(n-2)b2
      LACL #0          ; Acc = 0
      LTD  *-,AR0      ; T = d(n-1), d(n-2) = d(n-1)
      MPY  *+,AR1      ; Acc += d(n-2)b2, P = d(n-1)b1
      LTD  *-,AR0      ; T = d(n), d(n-1) = d(n)
      MPY  *+,AR1      ; Acc += d(n-1)b1, P = d(n)b0
IIRLOOP:
      LTA  *          ; Final accumulation
      LDP  #XN
      SACH OUTPUT,1    ; Save output in Q15 format

                        ; ACC -> OUTPUT Get rid of xtra sign bit.
      LACC OUTPUT      ; OUTPUT -> ACC
      SFL                      ; OUTPUT*2
      AND  #0fffch      ; Last two must be zero for AIC
      SAMM DXR          ; write output word to transmit register
      RETE

```

TRANSMIT:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RETE

```
*****
* DESCRIPTION: This routine initializes the 'C50 serial port          *
*               and the TLC320C40's (AIC) TA,RA,TB,RB and            *
*               control registers                                     *
*****
```

AICINIT:

```
SPLK #20h,TCR      ; To generate 10 MHz from Tout
SPLK #01h,PRD      ; for AIC master clock
MAR  *,AR0
LACC #0008h        ; Non continuous mode
SACL SPC           ; FSX as input
LACC #00c8h        ; 16 bit words
SACL SPC
LACC #080h         ; Pulse AIC reset by setting it low
SACH DXR
SACL GREG
LAR  AR0,#0FFFFh
RPT  #10000        ; and taking it high after 10000 cycles
LACC *,0,AR0       ; (.5ms at 50ns)
SACH GREG
;_____
LDP  #TA
SETC SXM
LACC TA,9          ; Initialize TA and RA register
ADD  RA,2
CALL AIC_2ND
;_____
LDP  #TB
LACC TB,9          ; Initialize TB and RB register
ADD  RB,2
ADD  #02h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL AIC_2ND
;-----
LDP #AIC_CTR
LACC AIC_CTR,2 ; Initialize control register
ADD #03h
CALL AIC_2ND
RET

```

AIC_2ND:

```

LDP #0 ; Data page point is 0 (MM regs)
SACH DXR ; send ACChi 00
CLRC INTM ; enable interrupts
IDLE ; wait for interrupt
ADD #6h,15 ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
SACH DXR ; send ACChi to initiate secondary protocol
IDLE ; wait for interrupt
SACL DXR ; send the T register data
IDLE ; wait for interrupt
LACL #0 ; clear ACClo
SACL DXR ; send another to make sure 1st word got sent
IDLE ; wait for interrupt
SETC INTM
RET

```

```

;=====
; in data from com port
;=====

```

INDATA:

GETZERO:

```

in 00h,PA0 ; 'temp' = data from com port
lacc #0ffh
and 00h
bcnd GETZERO,EQ ; repeat until 'temp' <> 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
;store 'temp' to 'data'
```

```
bldd #00h,*
```

```
mar *+
```

WAITZERO:

```
in 00h,PA0 ; 'temp' = data from com port
```

```
lacc #0ffh
```

```
and 00h
```

```
bcnd WAITZERO,NEQ ; repeat until 'temp' = 0
```

```
ret
```

```
;This routine for summain
```

SUM:

```
mar *,ar4
```

```
lacc *,5
```

```
sub #20h
```

```
mar *+,ar5
```

```
add *
```

```
sub #01h
```

```
rpt #04h
```

```
sfl
```

```
mar *+,ar6
```

```
add *
```

```
sub #01h
```

```
mar *+,ar7
```

```
sac1 *+
```

```
ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.end



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.mmregs
.ps 080ah
B RINT ;0A;Serial port receive interrupt RINT
B XINT ;0C;Serial port transmit interrupt XINT

TA .set 18 ; Auxin —+ +— Loopback
RA .set 18 ; Synch —+ | | +— BP Filter
TAp .set 1 ; | | | |
RAp .set 1 ;+-----+-----+
TB .set 15 ;|00 00 G1 G0 | SY AX LB BP|
RB .set 15 ;+-----+-----+
AIC_CTW .set 018h ; GAIN
ORDER .set 1 ; Order of filter

.ds 0300h
TEMP .word 0h
xbuf .word 0b000h
rbuf .space 32*16

*****
*TMS320C50 INITIALIZATION *
*This routine initializes the C50 registers, internal RAM *
*****

.ps 0a00h
.entry
INIT SETC INTM ; Disable Interrupts
LDP #0 ; Initialize Data Pointer
OPL #0834h,PMST ; Configure S/A RAM in data memory
CLRC CNF ; Configure B0 block in datamemory
LACC #0000h
MAR *.AR7
LAR AR7,#0100h
RPT #1023
SACL *+ ; Initialize B0 and B1 blocks = 0000h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
SAMM CWSR      ; Set Wait State Control Register
SAMM PDWSR     ; for 0 waits in pgm & data memory
```

```
lacc #rbuf      ;
samm arl
samm cbsrl
add #1fh
samm cberl
```

```
splk #0a9h,cberl ; AR1 is assigned to circular buffer 1
                    ; AR2 is assigned to circular buffer 2
                    ; Enable circular buffer 1,2
lar ar0,#TEMP
```

 * This routine initializes the TLC320C40 and reset serial port *

```
opl #030h,IMR    ; Using RINT syn TX & RX
splk #20h,TCR
splk #1,PRD
mar *,AR0
laci #08h        ; set FSM bit for FSX/FSR per frame
samm spc         ; Configure for 16 bit mode with
laci #0C8h       ; external CLKX, reset tx and rx
samm spc
lacc #0080h
sach dxr         ; clear first int
saci GREG        ; Pulse AIC reset by setting it low
lar AR0,#0FFFFFh
rpt #10000       ; and taking it high after 1000 cycles
lacc *,0,AR0     ; (.5ms at 50ns)
sach GREG
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call  AIC_SET      ; DO NOT CHANGE DP WITHOUT RESTORING IT!
clrc  INTM         ; Enable Interrupt

```

```

; Start of main program

```

```

LOOP:

```

```

    B    LOOP

```

```

    ret

```

```

; End of main program

```

```

*****

```

```

* This Routine is transmit and recieve between TMS320c50 and AIC *

```

```

*****

```

```

;

```

```

; RECIEVER INTERRUPT SERVICE ROUTINE

```

```

;

```

```

RINT:

```

```

    MAR    *,AR3

```

```

    LAMM   DRR    ~    ; read new sample

```

```

    SACL   *+

```

```

    LACC   xbuf

```

```

    SAMM   dxr      ; send it out

```

```

    RETE

```

```

;_____

```

```

XINT:

```

```

    rete

```

```

;_____

```

```

*****

```

```

AIC_SET:      ;

```

```

    lacc    #AIC_CTW,2 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

add    #03h        ;
call   AIC_2nd      ; Set Control Register of AIC
;-----
lacc   #TB,9        ;
add    #RB,2        ;
add    #02h        ;
call   AIC_2nd      ; Set TB and RB
;-----
lacc   #TA,9        ;
add    #RA,2        ;
call   AIC_2nd      ; Set TA and RA
;-----
ret    ;
;-----
AIC_2nd:
ldp    #0           ;
sach   DXR          ;
clrc   INTM         ;
idle   ;
add    #6,15        ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
sach   DXR          ;
idle   ; ACCU_hi requests 2nd XMIT
sac1   DXR          ;
idle   ; ACCU_lo sets up registers
lac1   #0           ;
sac1   DXR          ; make sure the word got sent
idle   ;
setc   INTM
ret    ;

```

.END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

program load_dsk;
{$M $4000,0,0 } { 16K stack, no heap }
uses Dos;
var
  ProgramName, CmdLine: string;
  filter_type      : integer;
  fi               : text;
begin
  assign(fi,'fil-type');
  reset(fi);
  readln(fi,filter_type);
  close(fi);
  case filter_type of
    1:begin
      ProgramName:='d:\dsktools\dsk51.exe';
      SwapVectors;
      Cmdline:='d:\prg\fir -c2';
      Exec(ProgramName, CmdLine);
      SwapVectors;
      case DosError of
        2:begin writeln; writeln('File not found.');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Exec(ProgramName, CmdLine);  
SwapVectors;  
case DosError of  
  2:begin writeln; writeln('File not found.');  3:begin writeln; writeln('Path not found.');  5:begin writeln; writeln('Access denied.');  6:begin writeln; writeln('Invalid handle.');  8:begin writeln; writeln('Not enough memory.'); 10:begin writeln; writeln('Invalid environment.'); 11:begin writeln; writeln('Invalid format.'); 18:begin writeln; writeln('No more files.');end;  
end;  
end;  
end.  
end.
```

```

{$N+,E+}
{$M 60000 0 60000}

program filter;

uses crt;

type complex =object
    r,ima :real;
    procedure s(re,i:real);
end;

const max_order =100;
      max_series =30;
var type_of_filter :byte;
    type_of_Window :byte;
    name_of_Window :array[1..5]of string;
    cnt,cnt2,order :byte;
    Fc,Fs,Fp,delta_f :real;
    Fp1,Fp2,Fs1,Fs2 :real;
    Fc1,Fc2,delta_fl,delta_fh :real;
    sam_fre,T :real;
    cx,cy,n,choice :byte;
    hd :array[0..max_order-1]of real;
    tempo :array[0..((max_order+1) div 2)]of real;
    A0n,A1n,A2n,B1n,B2n :array[1..max_order]of real;
    W :array[0..max_order-1]of real;
    answer :array[0..max_order-1]of real;
    bef_poi :array[0..max_order-1]of string;
    aft_poi :array[0..max_order-1]of string;
    Ap,Aa,A :real;
    sig_p,sig_s,sig :real;
    Alp,D,Cmp :real;
    Beta :array[0..max_order-1]of real;
    series_term :double;
    Io_Beta :array[0..max_order-1]of real;
    Io_Alp :real;
    cnt_series :integer;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

ch                                     :char;
repeat_again.repeat_main             :boolean;
fi                                    :text;
procedure complex.s(re,i:real);
begin
    r:=re;
    ima :=i;
end;
procedure int2digi(ip,long:word;var op:string);
var    ans,sed,cnt :word;
        quo        :word;
begin
    op:="";
    for cnt:=long downto 1 do begin
        quo:=trunc(exp((cnt-1)*ln(2)));
        ans:=ip div quo;
        if ans=1 then op:=op+'1'
        else op:=op+'0';
        sed:=ip mod trunc(exp((cnt-1)*ln(2)));
        ip :=ip mod trunc(exp((cnt-1)*ln(2)));
    end;
end;
procedure partition(ip:word;var p1,p2,p3:byte);
begin
    p3:=ip mod 32;
    ip:=ip div 32;
    p2:=ip mod 32;
    ip:=ip div 32;
    p1:=ip;
end;
function power(base,pwr:byte):longint;
begin
    power:=trunc(exp(pwr*ln(base)));
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure q13(r:real;var w:word);
var    cnt,cnt1          :word;
        bw,aw            :word;
        ar                :real;
begin
    aw:=0;
    bw:=trunc(r)*8192;
    ar:=r-trunc(r);
    for cnt:=1 to 13 do begin
        ar:=ar*2;
        if ar>=1 then begin
            ar:=ar-1;
            aw:=(aw*2)+1;
        end
        else begin
            aw:=aw*2;
        end;
        end;
        w:=bw+aw;
    end;
    procedure q15(r:real;var w:word);
    var    cnt,cnt1          :word;
    begin
        w:=0;
        for cnt:=1 to 15 do begin
            if r<0 then begin
                r:=r*2;
            end
            if r>=1 then begin
                r:=r-1;
                w:=(w*2)+1;
            end
            else begin
                w:=w*2;
            end;
        end;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
else begin
  if cnt=1 then w:=1 else begin
    w:=(w*2)+1;
    for cnt1:=trunc(ln(w)/ln(2))+1 to 15 do w:=w*2;
  end;
  break;
end;
end
end;

procedure two_comple(var w:word);
var
  op      :word;
  cnt     :byte;
begin
  op:=0;
  for cnt:=1 to 16 do begin
    if (w div power(2,16-cnt)=0) then begin
      op:=(op*2)+1;
    end
    else begin
      op:=op*2;
    end;
    w:=w mod power(2,16-cnt);
  end;
  op:=op+1;
  w:=op;
end;

procedure cal_q13(r:real;var w:word);
begin
  if r>0 then begin
    q13(r,w);
  end
  else begin
    r:=r*-1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    q13(r,w);
    two_comple(w);
end;
end;
procedure cal_q15(r:real;var w:word);
begin
    if r>0 then begin
        q15(r,w);
    end
    else begin
        r:=r*-1;
        q15(r,w);
        two_comple(w);
    end;
end;
procedure hide;assembler;
asm
    mov ah,01
    mov cx,0100h
    int 10h
end;
procedure show;assembler;
asm
    mov ah,01
    mov cx,0D0Eh
    int 10h
end;
function fac(n:integer):real;
var    loop:integer;
        ans :real;
begin
    ans:=1;

    for loop:=n downto 2 do begin
        ans:=ans*loop;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

fac:=ans;

end;

procedure real2str(r:real;var b,a:string);
var    af,be :real;
        bef,aft :integer;
        loop :integer;
begin
    if r<0 then begin
        r:=r*-1;
        be:=trunc(r);
        af:=r-trunc(r);
        bef:=trunc(be)*-1;
        aft:=trunc(af*10000);
        str(bef,b);
        str(aft,a);
        if aft<10 then a:='000'+a
        else if aft<100 then a:='00'+a
            else if aft<1000 then a:='0'+a;
        r:=r*-1;
        if r<-0.0001 then b:='-'+b;
    end
    else begin
        be:=trunc(r);
        af:=r-trunc(r);
        bef:=trunc(be);
        aft:=trunc(af*10000);
        str(bef,b);
        str(aft,a);
        if aft<10 then a:='000'+a
        else if aft<100 then a:='00'+a
            else if aft<1000 then a:='0'+a;
    end;

    if length(a)<4 then begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    for loop:=1 to 4-length(a) do a:=a+'0';
end;
end;
procedure Window1;
begin
    W[0]:=1;
end;
procedure Window2;
begin
    { even order }
    if (order mod 2=0) then begin
        for n:=1 to (order div 2) do begin
            tempo[n]:=0.5+(0.5*cos((2*pi*n)/(order-1)));
        end;
        for n:=1 to (order div 2) do begin
            W[n-1]:=tempo[(order div 2)-n+1];
            W[order-n]:=W[n-1];
        end;
    end
    else begin
        { odd order }
        for n:=0 to ((order-1) div 2) do begin
            tempo[n]:=0.5+(0.5*cos((2*pi*n)/(order-1)));
        end;
        W[(order-1)div 2]:=tempo[0];
        for n:=1 to ((order-1) div 2) do begin
            W[n-1]:=tempo[(order div 2)-n+1];
            W[order-n]:=W[n-1];
        end;
    end;
end;
end;
procedure Window3;
begin
    { even order }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if (order mod 2=0) then begin
  for n:=1 to (order div 2) do begin
    tempo[n]:=0.54+(0.46*cos((2*pi*n)/(order-1)));
  end;
  for n:=1 to (order div 2) do begin
    W[n-1]:=tempo[(order div 2)-n+1];
    W[order-n]:=W[n-1];
  end;
end
else begin
  { odd order }
  for n:=0 to ((order-1) div 2) do begin
    tempo[n]:=0.54+(0.46*cos((2*pi*n)/(order-1)));;
  end;
  W[(order-1)div 2]:=tempo[0];
  for n:=1 to ((order-1) div 2) do begin
    W[n-1]:=tempo[(order div 2)-n+1];
    W[order-n]:=W[n-1];
  end;
end;
end;
procedure Window4;
begin
  { even order }
  if (order mod 2=0) then begin
    for n:=1 to (order div 2) do begin
      tempo[n]:=0.42+(0.5*cos((2*pi*n)/(order-1)))+0.08*cos((4*n*pi)/(order-1));
    end;
    for n:=1 to (order div 2) do begin
      W[n-1]:=tempo[(order div 2)-n+1];
      W[order-n]:=W[n-1];
    end;
  end
  end
  else begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

[odd order]
for n:=0 to ((order-1) div 2) do begin
    tempo[n]:=0.42+(0.5*cos((2*pi*n)/(order-1)))+0.08*cos((4*n*pi)/(order-1));
end;
W[(order-1)div 2]:=tempo[0];
for n:=1 to ((order-1) div 2) do begin
    W[n-1]:=tempo[(order div 2)-n+1];
    W[order-n]:=W[n-1];
end;
end;
end;
procedure Window5;
begin
    sig_p:=(exp(0.05*Ap*ln(10))-1)/(exp(0.05*Ap*ln(10))+1);
    sig_s:=exp(-0.05*Aa*ln(10));
    if sig_p<sig_s then sig:=sig_p else sig:=sig_s;
    A :=-20*(ln(sig)/ln(10));
    {case of A}
    if A<=21 then begin
        Alp:=0
    end
    else begin
        if ((A>21)and(A<=50)) then begin
            Alp:=(0.5842*exp(0.4*ln(A-21)))+(0.07886*(A-21));
        end
        else begin
            Alp:=(0.1102*(A-8.7));
        end;
    end;
end;
{Calculate D correspond with A}
if A<=21 then begin
    D:=0.9222;
end
else begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

D:=(A-7.95)/14.36;
end;
{ Find the best Order}
Cmp:=(Sam_fre*D/delta_F)+1;
if Cmp>Max_order-3 then begin
    Show;
    clrscr;
    TextColor(lightred+blink);
    writeln('Error !!! The order is more than ability of this program. ');
    NormVideo;
    halt;
end;
Order:=1;
while Order<Cmp do begin
    Order:=Order+2;
end;
writeln('_____');
writeln(' The best order is ',Order);
writeln('_____');
{ Calculate Beta }
for n:=0 to (Order-1)div 2 do begin
    case n of
        0:Beta[n]:=Alp;
        else begin
            Beta[n]:=Alp*sqrt(1-exp(2*ln(2*n/(Order-1))));
        end;
    end;
end;
{ Calculate Io_beta }
for n:=0 to (order-1)div 2 do begin
    series_term:=0;
    if Beta[n]=0 then series_term:=0
    else begin
        series_term:=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    for cnt_series:=1 to max_series do begin
        series_term:=series_term+exp(2*ln((1/fac(cnt_series))*exp(cnt_series*ln(Beta[n]/2))));
    end;
end;
Io_Beta[n]:=1+series_term;
end;
{Calculate Io_Alp}
series_term:=0;
if Alp=0 then begin
    series_term:=0;
end
else begin
    for cnt_series:=1 to max_series do begin
        series_term:=series_term+exp(2*ln((1/fac(cnt_series))*exp(cnt_series*ln(Alp/2))));
    end;
end;
Io_Alp:=1+series_term;
{even order}
if (order mod 2=0) then begin
    for n:=1 to (order div 2) do begin
        tempo[n]:=Io_Beta[n]/Io_Alp;
    end;
    for n:=1 to (order div 2) do begin
        W[n-1]:=tempo[(order div 2)-n+1];
        W[order-n]:=W[n-1];
    end;
end
else begin
    {odd order}
    for n:=0 to ((order-1) div 2) do begin
        tempo[n]:=Io_Beta[n]/Io_Alp;
    end;
    W[(order-1)div 2]:=tempo[0];
    for n:=1 to ((order-1) div 2) do begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        W[n-1]:=tempo[(order div 2)-n+1];
        W[order-n]:=W[n-1];
    end;
end;
end;
procedure user_input;
var    correct    :boolean;
        ans       :char;
        x,y       :integer;
begin
    repeat
        clrscr;
        cy:=whereY;
        writeln('You choose 'name_of_window[type_of_Window],'.');
        writeln('Enter the variable');
        write(' Sampling frequency          : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(sam_fre);
        sam_fre:=sam_fre*1000;
        T:=1/sam_fre;
        case type_of_Window of
            5:begin
                write(' Pass band attenuate          : ');
                write('          dB. ');
                cy:=whereY;
                cx:=42;
                gotoxy(cx,cy);
                readln(Ap);
                write(' Stop band attenuate          : ');
                write('          dB. ');
                cy:=whereY;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cx:=42;
        gotoxy(cx,cy);
        readln(Aa);
    end;
else begin
    cy:=whereY;
    order:=2;
    repeat
        gotoxy(1,cy);
        if ((order>100)or(order<2)) then begin
            write(' ');
            gotoxy(1,cy);
            write('Error!!! You must input order of filter between 2 to 100. ');
            delay(2000);
        end;
        gotoxy(1,cy);
        write(' ');
        gotoxy(1,cy);
        write(' Order of filter (2 - 100) : ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(order);
    until ( (order<=100) and (order>=2) );
end;
end;
write('Are you sure (Y/N) ? ');
x:=whereX;
y:=whereY;
repeat
    gotoxy(x,y);
    readln(ans);
    ans:=upcase(ans);
until ans in ['N','Y'];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

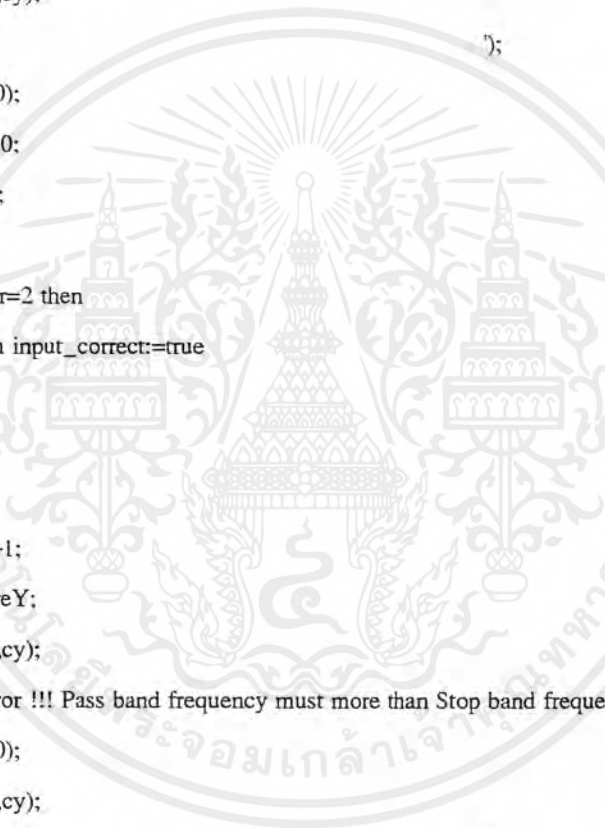
        if ans='Y' then correct:=true else correct:=false;
    until correct;
end;
procedure input_for_high_low;
var    input_correct:boolean;
    procedure user_input;
    begin
        cy:=7;
        gotoxy(1,cy);
        writeln('');
        writeln('');
        gotoxy(1,cy);
        write(' Pass band frequency      : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(Fp);
        cy:=whereY;
        gotoxy(1,cy);
        write(' Stop band frequency      : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(Fs);
    end;
begin
    input_correct:=false;
    user_input;
    repeat
        if type_of_filter=1 then
            if Fs>Fp then input_correct:=true
            else begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

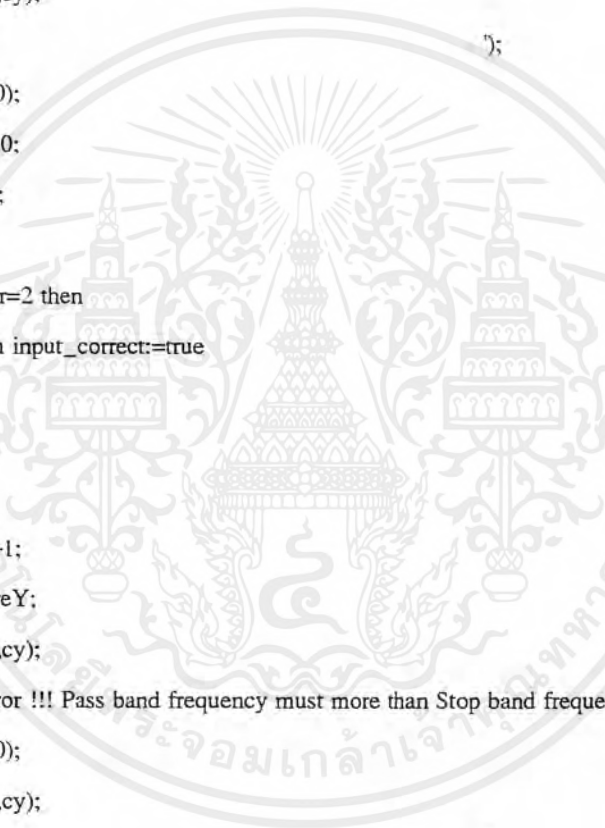
```

cnt:=0;
repeat
    cnt:=cnt+1;
    cy:=whereY;
    gotoxy(1,cy);
    write('Error !!! Stop band frequency must more than Pass band frequency.');
```



```

    delay(200);
    gotoxy(1,cy);
    write('
');
    delay(100);
until cnt=10;
user_input;
end;
if type_of_filter=2 then
    if Fp>Fs then input_correct:=true
    else begin
        cnt:=0;
        repeat
            cnt:=cnt+1;
            cy:=whereY;
            gotoxy(1,cy);
            write('Error !!! Pass band frequency must more than Stop band frequency.');
```



```

            delay(200);
            gotoxy(1,cy);
            write('
');
            delay(100);
        until cnt=10;
        user_input;
    end;
until input_correct;
Fp:=Fp*1000;
Fs:=Fs*1000;
(Calculate Fc)
Fc:=(Fp+Fs)/2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delta_f:=abs(Fs-Fp);
end;

procedure input_for_stop_pass;
var    input_correct:boolean;

    procedure user_input;
    begin
        cy:=7;
        gotoxy(1,cy);
        writeln('                ');
        writeln('                ');
        writeln('                ');
        writeln('                ');
        gotoxy(1,cy);
        {-----for lower-----}
        write(' Lower Pass band frequency (Fp1)  : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(Fp1);
        cy:=whereY;
        gotoxy(1,cy);
        write(' Lower Stop band frequency (Fs1)  : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);
        readln(Fs1);
        {-----for higher-----}
        write(' Higher Pass band frequency (Fp2)  : ');
        write('          KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

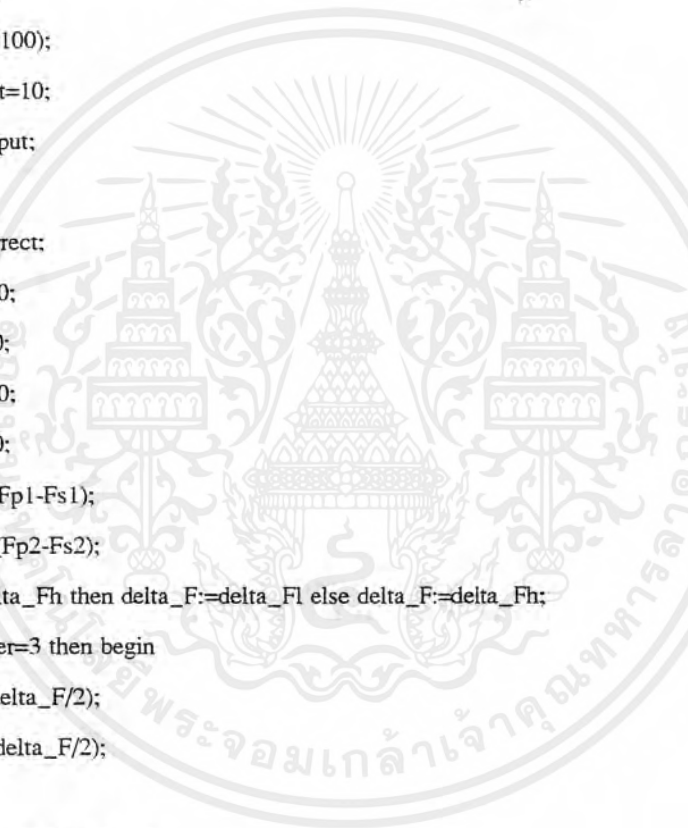
readln(Fp2);
cy:=whereY;
gotoxy(1,cy);
write(' Higher Stop band frequency (Fs2)  : ');
write('      KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fs2);
end;
begin
input_correct:=false;
user_input;
repeat
if type_of_filter=3 then
if((Fs2>Fp2)and(Fp2>Fp1)and(Fp1>Fs1))then input_correct:=true
else begin
cnt:=0;
repeat
cnt:=cnt+1;
cy:=whereY;
gotoxy(1,cy);
write('Error !!! Input should be Fs2>Fp2 and Fp2>Fp1 and Fp1>Fs1. ');
delay(200);
gotoxy(1,cy);
write('
delay(100);
until cnt=10;
user_input;
end;
if type_of_filter=4 then
if((Fp2>Fs2)and(Fs2>Fs1)and(Fs1>Fp1))then input_correct:=true
else begin
cnt:=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

repeat
    cnt:=cnt+1;
    cy:=whereY;
    gotoxy(1,cy);
    write('Error !!! Input should be Fp2>Fs2 and Fs2>Fs1 and Fs1>Fp1.');
```



```

    delay(200);
    gotoxy(1,cy);
    write('
    delay(100);
until cnt=10;
user_input;
end;
until input_correct;
Fp1:=Fp1*1000;
Fs1:=Fs1*1000;
Fp2:=Fp2*1000;
Fs2:=Fs2*1000;
delta_Fl:=abs(Fp1-Fs1);
delta_Fh:=abs(Fp2-Fs2);
if delta_Fl<delta_Fh then delta_F:=delta_Fl else delta_F:=delta_Fh;
if type_of_filter=3 then begin
    Fc1:=Fp1-(delta_F/2);
    Fc2:=Fp2+(delta_F/2);
end;
if type_of_filter=4 then begin
    Fc1:=Fp1+(delta_F/2);
    Fc2:=Fp2-(delta_F/2);
end;
end;
procedure Hd_low_cal;
begin
    if (order mod 2=0) then begin
        for n:=1 to (order div 2) do begin
            tempo[n]:=(1/(n*pi))*(sin(n*2*pi*Fc*T));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
for n:=1 to (order div 2) do begin
    hd[n-1]:=tempo[(order div 2)-n+1];
    hd[order-n]:=hd[n-1];
end;
end
else begin
    for n:=0 to ((order-1) div 2) do begin
        case n of
            0:begin
                tempo[0]:=2*Fc/sam_fre;
            end;
            else begin
                tempo[n]:=(1/(n*pi))*(sin(n*2*pi*Fc*T));
            end;
        end;
    end;
    hd[(order-1)div 2]:=tempo[0];
    for n:=1 to ((order-1) div 2) do begin
        hd[n-1]:=tempo[((order-1) div 2)-n+1];
        hd[order-n]:=hd[n-1];
    end;
end;
end;
end;
procedure Hd_hi_cal;
begin
    if (order mod 2=0) then begin
        for n:=1 to (order div 2) do begin
            tempo[n]:=-(1/(n*pi))*(sin(n*2*pi*Fc*T));
        end;
        for n:=1 to (order div 2) do begin
            hd[n-1]:=tempo[(order div 2)-n+1];
            hd[order-n]:=hd[n-1];
        end;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

end
else begin
  for n:=0 to ((order-1) div 2) do begin
    case n of
      0:begin
        tempo[0]:=1-2*Fc/sam_fre;
        end;
        else begin
          tempo[n]:=-(1/(n*pi))*(sin(n*2*pi*Fc*T));
          end;
        end;
      end;
    hd[(order-1)div 2]:=tempo[0];
    for n:=1 to ((order-1) div 2) do begin
      hd[n-1]:=tempo[(order-1) div 2-n+1];
      hd[order-n]:=hd[n-1];
    end;
  end;
end;
procedure Hd_pass_cal;
begin
  if (order mod 2=0) then begin
    for n:=1 to (order div 2) do begin
      tempo[n]:=(1/(n*pi))*(sin(n*2*pi*Fc2*T)-sin(n*2*pi*Fc1*T));
    end;
    for n:=1 to (order div 2) do begin
      hd[n-1]:=tempo[(order div 2)-n+1];
      hd[order-n]:=hd[n-1];
    end;
  end
else begin
  for n:=0 to ((order-1) div 2) do begin
    case n of
      0:begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        tempo[0]:=2*T*(Fc2-Fc1);
    end;
    else begin
        tempo[n]:=(1/(n*pi))*(sin(n*2*pi*Fc2*T)-sin(n*2*pi*Fc1*T));
    end;
end;
end;
hd[(order-1)div 2]:=tempo[0];
for n:=1 to ((order-1) div 2) do begin
    hd[n-1]:=tempo[((order-1) div 2)-n+1];
    hd[order-n]:=hd[n-1];
end;
end;
end;
procedure Hd_stop_cal;
begin
    if (order mod 2=0) then begin
        for n:=1 to (order div 2) do begin
            tempo[n]:=-(1/(n*pi))*(sin(n*2*pi*Fc2*T)-sin(n*2*pi*Fc1*T));
        end;
        for n:=1 to (order div 2) do begin
            hd[n-1]:=tempo[(order div 2)-n+1];
            hd[order-n]:=hd[n-1];
        end;
    end
    else begin
        for n:=0 to ((order-1) div 2) do begin
            case n of
                0:begin
                    tempo[0]:=(2*T*(Fc1-Fc2))+1;
                end;
            end;
        end;
        else begin
            tempo[n]:=-(1/(n*pi))*(sin(n*2*pi*Fc2*T)-sin(n*2*pi*Fc1*T));
        end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end;
    end;
    hd[(order-1)div 2]:=tempo[0];
    for n:=1 to ((order-1) div 2) do begin
        hd[n-1]:=tempo[((order-1) div 2)-n+1];
        hd[order-n]:=hd[n-1];
    end;
end;
end;
end;
procedure Hd_cal;
begin
    case type_of_filter of
        1:begin
            Hd_low_cal;
        end;
        2:begin
            Hd_hi_cal;
        end;
        3:begin
            Hd_pass_cal;
        end;
        else begin
            Hd_stop_cal;
        end;;
    end;
end;
end;
procedure win_cal;
begin
    case type_of_Window of
        1:Window1;
        2:Window2;
        3:Window3;
        4:Window4;
        5:Window5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

end;
end;
procedure send_data_iir;
var    fi      :text;
        st      :string;
        wd      :word;
        p1,p2,p3 :byte;
begin
    assign(fi,'test');
    rewrite(fi);
    port[$378]:=0;{ ——}
    clrscr;
    hide;
    write('Swap to DSK loader ,then swap to this program and press any key (  ).');
    cy:=whereY;
    for cnt:=20 downto 0 do begin
        gotoxy(67,cy);
        write(' ');
        gotoxy(67,cy);
        write(cnt:2);
        delay(500);
        if keypressed then break;
    end;
    writeln;
    if cnt=0 then begin
        writeln('Communication fail !,call this program again. ');
        halt;
    end;
    writeln('N = ',n);
    ch:=readkey;
    writeln('Start send data. ');
    writeln;
    show;

    port[$378]:=n;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay(50);
port[$378]:=0;
delay(50);
cnt:=0;
repeat
    cnt:=cnt+1;

    writeln('Send coefficient of state ',cnt:3,' ');

    write(fi,cnt:3);
    st:=' ';
    write(fi,st);

    [convert to q13 format]
    cal_q13(-B2n[cnt],wd);
    int2digi(wd,16,st);
    st:=st+' ';
    write(fi,st);

    partition(wd,p1,p2,p3);
    p1:=p1+1;
    p2:=p2+1;
    p3:=p3+1;

    int2digi(p1,7,st);
    st:=st+' ';
    port[$378]:=p1;
    write(fi,st);
    port[$378]:=0;

    int2digi(p2,6,st);
    st:=st+' ';
    port[$378]:=p2;
    write(fi,st);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
port[$378]:=0;
```

```
int2digi(p3,6,st);
```

```
st:=st+' ';
```

```
port[$378]:=p3;
```

```
writeln(fi,st);
```

```
port[$378]:=0;
```

```
{convert to q13 format}
```

```
cal_q13(-Bln[cnt],wd);
```

```
int2digi(wd,16,st);
```

```
st:=st+' ';
```

```
write(fi,st);
```

```
partition(wd,p1,p2,p3);
```

```
p1:=p1+1;
```

```
p2:=p2+1;
```

```
p3:=p3+1;
```

```
int2digi(p1,7,st);
```

```
st:=st+' ';
```

```
port[$378]:=p1;
```

```
write(fi,st);
```

```
port[$378]:=0;
```

```
int2digi(p2,6,st);
```

```
st:=st+' ';
```

```
port[$378]:=p2;
```

```
write(fi,st);
```

```
port[$378]:=0;
```

```
int2digi(p3,6,st);
```

```
st:=st+' ';
```

```
port[$378]:=p3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

writeln(fi,st);
port[$378]:=0;

[convert to q13 format]
cal_q13(A2n[cnt],wd);
int2digi(wd,16,st);
st:=st+' ';
write(fi,st);

partition(wd,p1,p2,p3);
p1:=p1+1;
p2:=p2+1;
p3:=p3+1;

int2digi(p1,7,st);
st:=st+' ';
port[$378]:=p1;
write(fi,st);
port[$378]:=0;

int2digi(p2,6,st);
st:=st+' ';
port[$378]:=p2;
write(fi,st);
port[$378]:=0;

int2digi(p3,6,st);
st:=st+' ';
port[$378]:=p3;
writeln(fi,st);
port[$378]:=0;

```

```
[convert to q13 format]
```

```
cal_q13(A1n[cnt],wd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int2digi(wd,16,st);  
st:=st+' '  
write(fi,st);  
  
partition(wd,p1,p2,p3);  
p1:=p1+1;  
p2:=p2+1;  
p3:=p3+1;
```

```
int2digi(p1,7,st);  
st:=st+' '  
port[$378]:=p1;  
write(fi,st);  
port[$378]:=0;
```

```
int2digi(p2,6,st);  
st:=st+' '  
port[$378]:=p2;  
write(fi,st);  
port[$378]:=0;
```

```
int2digi(p3,6,st);  
st:=st+' '  
port[$378]:=p3;  
writeln(fi,st);  
port[$378]:=0;
```

```
|convert to q13 format|  
cal_q13(A0n[cnt],wd);  
int2digi(wd,16,st);  
st:=st+' '  
write(fi,st);
```

```
partition(wd,p1,p2,p3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

p1:=p1+1;
p2:=p2+1;
p3:=p3+1;

int2digi(p1,7,st);
st:=st+' ';
port[$378]:=p1;
write(fi,st);
port[$378]:=0;

int2digi(p2,6,st);
st:=st+' ';
port[$378]:=p2;
write(fi,st);
port[$378]:=0;

int2digi(p3,6,st);
st:=st+' ';
port[$378]:=p3;
writeln(fi,st);
port[$378]:=0;

until cnt=n;
close(fi);
writeln;
writeln('Finish send data. ');
writeln('-----');

end;

procedure send_data_fir;
var   fi      :text;
      st      :string;
      wd      :word;
      p1,p2,p3 :byte;

begin
  assign(fi,'test');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

rewrite(fi);
port[$378]:=0:|——|
clrscr;
hide;
write('Swap to DSK loader ,then swap to this program and press any key (  ).');
cy:=whereY;
for cnt:=20 downto 0 do begin
  gotoxy(67,cy);
  write(' ');
  gotoxy(67,cy);
  write(cnt:2);
  delay(500);
  if keypressed then break;
end;
writeln;
if cnt=0 then begin
  writeln('Communication fail !,call this program again. ');
  halt;
end;
writeln('Start send data. ');
writeln;
show;
port[$378]:=order;
delay(50);
port[$378]:=0;
delay(50);
cnt:=0;
repeat
  writeln('Send ',cnt+1:3,' data. ');
  write(fi,cnt:3);
  st:='.';
  write(fi,st);

```

(convert to q15 format)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cal_q15(answer[cnt],wd);
```

```
int2digi(wd,16,st);
```

```
st:=st+' ';
```

```
write(fi,st);
```

```
partition(wd,p1,p2,p3);
```

```
p1:=p1+1;
```

```
p2:=p2+1;
```

```
p3:=p3+1;
```

```
int2digi(p1,7,st);
```

```
st:=st+' ';
```

```
port[$378]:=p1;
```

```
write(fi,st);
```

```
port[$378]:=0;
```

```
int2digi(p2,6,st);
```

```
st:=st+' ';
```

```
port[$378]:=p2;
```

```
write(fi,st);
```

```
port[$378]:=0;
```

```
int2digi(p3,6,st);
```

```
st:=st+' ';
```

```
port[$378]:=p3;
```

```
writeln(fi,st);
```

```
port[$378]:=0;
```

```
cnt:=cnt+1;
```

```
until cnt=order;
```

```
close(fi);
```

```
writeln;
```

```
writeln('Finish send data.');
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure fir:
begin
  repeat_again:=true;
  repeat
    clrscr;
    writeln('What type of filter ?');
    writeln('1.Low pass');
    writeln('2.High pass');
    writeln('3.Band pass');
    writeln('4.Band stop');
    cy:=whereY;
    repeat
      gotoxy(1,cy);
      write(' ');
      gotoxy(1,cy);
      write('Enter type of filter : ');
      readln(type_of_filter);
    until type_of_filter in[1,2,3,4];
    case type_of_filter of
      1,2:input_for_high_low;
      3,4:input_for_stop_pass;
    end;
    {initial name of Window}
    name_of_window[1]:=Rectangular Window';
    name_of_window[2]:=Hann Window';
    name_of_window[3]:=Hamming Window';
    name_of_window[4]:=Blackman Window';
    name_of_window[5]:=Kaiser Window';
    for cnt:=1 to 70 do write('-');
    writeln;
    writeln('What type of Window ? ');
    for cnt:=1 to 5 do writeln(cnt,'. ',name_of_window[cnt]);
    writeln;
    cy:=whereY;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

repeat
    gotoxy(1,cy);
    writeln(' ');
    gotoxy(1,cy);
    write('Enter type of Window : ');
    readln(type_of_Window);
until type_of_window in [1,2,3,4,5];
user_input;
win_cal;
hd_cal;
case type_of_Window of
    1:for n:=0 to order-1 do answer[n]:=hd[n];
    else begin
        for n:=0 to order-1 do begin
            answer[n]:=hd[n]*W[n];
        end;
    end;
end;
hide;
for n:=0 to order-1 do begin
    real2str(answer[n],bef_poi[n],aft_poi[n]);
    writeln('h[' ,n:2.'] = ',answer[n], ' ',bef_poi[n],',',aft_poi[n]);
    if (((n+1) mod 10)=0) then begin
        cy:=whereY;
        repeat
            ch:=chr(0);
            write('Press any key to continue. ');
            delay(500);
            gotoxy(1,cy);
            write(' ');
            delay(300);
            gotoxy(1,cy);

            if keypressed then ch:=readkey;
        until ch<>chr(0);
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
if (((n+1) mod 10)=0) then begin
    gotoxy(1,cy);
    writeln('-----');
end;
end;
if (((n+1) mod 10) <> 0) then begin
    writeln('-----');
    ch:=readkey;
end;
show;
{ Send Data }
Send_data_fir;
write('Do you want to select another filter (y/n)?');
repeat
    ch:=readkey;
    if ch in ['y','Y','n','N'] then begin
        ch:=upcase(ch);
    end;
until ch in ['Y','N'];
writeln(ch);
if ch='Y' then repeat_again:=true else repeat_again:=false;
until repeat_again=false;
end;
procedure iir;
var
    Alp,Beta           :real;
    Fc.Fs.Fp           :real;
    Fp1,Fp2.Fs1.Fs2    :real;
    Fc1,Fc2            :real;
    sam_fre,T           :real;
    cx,cy              :byte;
    type_of_filter      :byte;
    order.order_b       :real;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Ap,As          :real;
K,K1,K2,Ka,Kb,Kc      :real;
A              :real;
order_odd      :boolean;
Spn            :array[1..max_order]of complex;
Dpn,Db1,Db2,Dbz      :array[1..max_order]of real;
Zpn,Zzn        :array[1..max_order]of complex;
C1,C2,C3,C4        :array[1..max_order]of real;
ch             :char;
srib,stria        :string;
Wu,Wl,Wo,Del_w      :real;
Sbp1,Sbp2        :array[1..max_order]of complex;
Zbp1,Zbp2,Zb       :array[1..max_order]of complex;
Sig_o           :real;
Q,Qo            :real;
V              :real;
Sbz,Zbz         :array[1..max_order]of complex;

```

```

procedure real2str(r:real;var b,a:string);

```

```

var    af,be :real;
      bef,aft :integer;
      loop :integer;

```

```

begin

```

```

  if r<0 then begin

```

```

    r:=r*-1;

```

```

    be:=trunc(r);

```

```

    af:=r-trunc(r);

```

```

    bef:=trunc(be)*-1;

```

```

    aft:=trunc(af*10000);

```

```

    str(bef,b);

```

```

    str(aft,a);

```

```

    if aft<10 then a:='000'+a

```

```

    else if aft<100 then a:='00'+a

```

```

    else if aft<1000 then a:='0'+a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

r:=r*-1;
if ((r<-0.0001)and(b='0')) then b:='-'+b;
end
else begin
be:=trunc(r);
af:=r-trunc(r);
bef:=trunc(be);
aft:=trunc(af*10000);
str(bef,b);
str(aft,a);
if aft<10 then a:='000'+a
else if aft<100 then a:='00'+a
      else if aft<1000 then a:='0'+a;
end;
if length(a)<4 then begin
for loop:=1 to 4-length(a) do a:=a+'0';
end;
end;
function log(r:real):real;
begin
log:=ln(r)/ln(10);
end;
function tan(r:real):real;
begin
tan:=sin(r)/cos(r);
end;
function power(b,p:real):real;
var   ans           :real;
      sign.cnt      :integer;
      temp           :real;
begin
if b>0 then begin
temp:=ln(b);
power:=exp(p*temp)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
else begin
    b:=b*-1;
    temp:=ln(b);
    ans:=exp(p*temp);
    sign:=1;
    for cnt:=1 to trunc(p) do sign:=sign*-1;
    power:=sign*ans;
end;
end;
function cosh(r:real):real;
begin
    cosh:=(exp(r)+exp(-r))/2;
end;
function sinh(r:real):real;
begin
    sinh:=(exp(r)-exp(-r))/2;
end;
procedure hide;assembler;
asm
    mov ah,01
    mov cx,0100h
    int 10h
end;
procedure show;assembler;
asm
    mov ah,01
    mov cx,0D0Eh
    int 10h
end;
procedure size_com(ip:complex;var size:real);
begin
    size:=sqrt(sqrt(ip.r)+sqrt(ip.ima));
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure mul_com(ip1,ip2:complex;var op:complex);
```

```
begin
```

```
    op.r:=(ip1.r*ip2.r)-(ip1.ima*ip2.ima);
```

```
    op.ima :=(ip1.r*ip2.ima)+(ip2.r*ip1.ima);
```

```
end;
```

```
procedure quo_com(ip1,ip2:complex;var op:complex);
```

```
var    temp :complex;
```

```
    size :real;
```

```
begin
```

```
    ip2.ima:=ip2.ima*-1;
```

```
    mul_com(ip1,ip2,op);
```

```
    size_com(ip2,size);
```

```
    size:=sqr(size);
```

```
    op.r:=op.r/size;
```

```
    op.ima :=op.ima/size;
```

```
end;
```

```
procedure add_com(ip1,ip2:complex;var op:complex);
```

```
begin
```

```
    op.r:=ip1.r+ip2.r;
```

```
    op.ima :=ip1.ima +ip2.ima;
```

```
end;
```

```
procedure sub_com(ip1,ip2:complex;var op:complex);
```

```
var    temp,minus :complex;
```

```
begin
```

```
    minus.s(-1,0);
```

```
    mul_com(minus,ip2,temp);
```

```
    add_com(ip1,temp,op);
```

```
end;
```

```
procedure input_for_high_low;
```

```
var    input_correct:boolean;
```

```
    procedure user_input;
```

```
    begin
```

```
        cy:=7;
```

```
        gotoxy(1,cy);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

writeln('');
writeln('');
writeln('');
writeln('');
writeln('');
writeln('');
gotoxy(1,cy);
write(' Pass band frequency : ');
write(' KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fp);
cy:=whereY;
gotoxy(1,cy);
write(' Stop band frequency : ');
write(' KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fs);
end;
begin
input_correct:=false;
user_input;
repeat
if type_of_filter=1 then
if Fs>Fp then input_correct:=true
else begin
cnt:=0;
repeat
cnt:=cnt+1;
cy:=whereY;
gotoxy(1,cy);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write('Error !!! Stop band frequency must more than ');
write('Pass band frequency. ');
delay(200);
gotoxy(1,cy);
write(' ');
delay(100);
until cnt=10;
user_input;
end;
if type_of_filter=2 then
if Fp>Fs then input_correct:=true
else begin
cnt:=0;
repeat
cnt:=cnt+1;
cy:=whereY;
gotoxy(1,cy);
write('Error !!! Pass band frequency must more than ');
write('Stop band frequency. ');
delay(200);
gotoxy(1,cy);
write(' ');
delay(100);
until cnt=10;
user_input;
end;
until input_correct;
Fp:=Fp*1000;
Fs:=Fs*1000;
end;
procedure input_for_stop_pass;
var input_correct:boolean;
procedure user_input;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cy:=7;
gotoxy(1,cy);
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
writeln('                                     ');
gotoxy(1,cy);
{-----for lower-----}
write(' Lower Pass band frequency (Fp1)  : ');
write('      KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fp1);
cy:=whereY;
gotoxy(1,cy);
write(' Lower Stop band frequency (Fs1)  : ');
write('      KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fs1);
{-----for higher-----}
write(' Higher Pass band frequency (Fp2)  : ');
write('      KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fp2);
cy:=whereY;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

gotoxy(1,cy);
write(' Higher Stop band frequency (Fs2)  : ');
write('      KHz. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Fs2);

end;

begin
input_correct:=false;
user_input;
repeat
if type_of_filter=3 then
if((Fs2>Fp2)and(Fp2>Fp1)and(Fp1>Fs1))then input_correct:=true
else begin
cnt:=0;
repeat
cnt:=cnt+1;
cy:=whereY;
gotoxy(1,cy);
write('Error !!! Input should be Fs2>Fp2 and Fp2>Fp1 and ');
write('Fp1>Fs1. ');
delay(200);
gotoxy(1,cy);
write('
');
delay(100);
until cnt=10;
user_input;
end;
if type_of_filter=4 then
if((Fp2>Fs2)and(Fs2>Fs1)and(Fs1>Fp1))then input_correct:=true
else begin
cnt:=0;
repeat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cnt:=cnt+1;
        cy:=whereY;
        gotoxy(1,cy);
        write('Error !!! Input should be Fp2>Fs2 and Fs2>Fs1 and ');
        write('Fs1>Fp1. ');
        delay(200);
        gotoxy(1,cy);
        write(' ');
        delay(100);
    until cnt=10;
    user_input;
end;
until input_correct;
Fp1:=Fp1*1000;
Fs1:=Fs1*1000;
Fp2:=Fp2*1000;
Fs2:=Fs2*1000;
end;
procedure user_input;
var    correct    :boolean;
        ans       :char;
        x,y       :integer;
begin
    repeat
        case type_of_filter of
            1,2:input_for_high_low;
            3,4:input_for_stop_pass;
        end;
        cy:=whereY;
        write(' Sampling frequency      : ');
        write('      KHz. ');
        cy:=whereY;
        cx:=42;
        gotoxy(cx,cy);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

readln(sam_fre);
sam_fre:=sam_fre*1000;
T:=1/sam_fre;
write(' Pass band attenuate      : ');
write('      dB. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(Ap);
write(' Stop band attenuate      : ');
write('      dB. ');
cy:=whereY;
cx:=42;
gotoxy(cx,cy);
readln(As);
write('Are you sure (Y/N) ? ');
x:=whereX;
y:=whereY;
repeat
    gotoxy(x,y);
    readln(ans);
    ans:=uppercase(ans);
until ans in ['N','Y'];
if ans='Y' then correct:=true else correct:=false;
until correct;
end;
procedure cal_k;
begin
    case type_of_filter of
        1:K:=tan(pi*Fp/Sam_fre)/tan(pi*Fs/Sam_fre);
        2:K:=tan(pi*Fs/Sam_fre)/tan(pi*Fp/Sam_fre);
        3:begin
            Ka:=tan(pi*Fp2/Sam_fre)-tan(pi*Fp1/Sam_fre);
            Kb:=tan(pi*Fp1/Sam_fre)*tan(pi*Fp2/Sam_fre);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Kc:=tan(pi*Fs1/Sam_fre)*tan(pi*Fs2/Sam_fre);
K1:=(Ka*tan(pi*Fs1/Sam_fre))/(Kb-power(tan(pi*Fs1/Sam_fre),2));
K2:=(Ka*tan(pi*Fs2/Sam_fre))/(power(tan(pi*Fs2/Sam_fre),2)-Kb);
if Kc>=Kb then K:=K1 else K:=K2;
end;
4:begin
Ka:=tan(pi*Fp2/Sam_fre)-tan(pi*Fp1/Sam_fre);
Kb:=tan(pi*Fp1/Sam_fre)*tan(pi*Fp2/Sam_fre);
Kc:=tan(pi*Fs1/Sam_fre)*tan(pi*Fs2/Sam_fre);
K1:=(Ka*tan(pi*Fs1/Sam_fre))/(Kb-power(tan(pi*Fs1/Sam_fre),2));
K2:=(Ka*tan(pi*Fs2/Sam_fre))/(power(tan(pi*Fs2/Sam_fre),2)-Kb);
if Kc>=Kb then K:=1/K2 else K:=1/K1;
end;
end;
end;
procedure cal_lp_hp;
var cnt :byte;
up,down :complex;
begin
if type_of_filter=1 then begin
Alp:=power(power(10,0.1*Ap)-1,-1/(2*Order))*tan(pi*Fp/Sam_fre);
for cnt:=1 to n do begin
Zzn[cnt].s(-1,0);
Dpn[cnt]:=1+(2*Alp*abs(Spn[cnt].r))
+sqr(Alp)*(sqr(Spn[cnt].r)+sqr(Spn[cnt].ima));
up.s(1-(sqr(Spn[cnt].r)+sqr(Spn[cnt].ima))*sqr(Alp),
2*Alp*Spn[cnt].ima);
down.s(Dpn[cnt],0);
quo_com(up,down,Zpn[cnt]);
end;
end
else begin
Alp:=power(power(10,0.1*Ap)-1,1/(2*Order))*tan(pi*Fp/Sam_fre);
for cnt:=1 to n do begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Zzn[cnt].s(1,0);
Dpn[cnt]:=sqrt(Alp)+2*abs(Spn[cnt].r)*Alp+sqrt(Spn[cnt].r)
+sqrt(Spn[cnt].ima);
Zpn[cnt].s(-(sqrt(Alp)-(sqrt(Spn[cnt].r)+sqrt(Spn[cnt].ima)))/Dpn[cnt]
.2*Spn[cnt].ima*Alp/Dpn[cnt]));
end;
end;
end;
procedure cal_lh_A_B;
var cnt      :byte;
begin
{order is even}
if trunc(order)mod 2=0 then begin
for cnt:=1 to n do begin
A0n[cnt]:=1;
A1n[cnt]:=-2*Zzn[cnt].r;
A2n[cnt]:=sqrt(Zzn[cnt].r)+sqrt(Zzn[cnt].ima);
B1n[cnt]:=-2*Zpn[cnt].r;
B2n[cnt]:=sqrt(Zpn[cnt].r)+sqrt(Zpn[cnt].ima);
end;
end
{order is odd}
else begin
if type_of_filter=1 then begin
for cnt:=1 to n do begin
if cnt=1 then begin
Zpn[cnt].s((1-abs(Spn[cnt].r)*Alp)/(1+abs(Spn[cnt].r)*Alp),
0);
A0n[cnt]:=1;
A1n[cnt]:=1;
A2n[cnt]:=0;
B1n[cnt]:=-Zpn[cnt].r;
B2n[cnt]:=0;
end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else begin
    A0n[cnt]:=1;
    A1n[cnt]:=-2*Zzn[cnt].r;
    A2n[cnt]:=sqr(Zzn[cnt].r)+sqr(Zzn[cnt].ima);
    B1n[cnt]:=-2*Zpn[cnt].r;
    B2n[cnt]:=sqr(Zpn[cnt].r)+sqr(Zpn[cnt].ima);
end;
end;
end
else begin
    for cnt:=1 to n do begin
        if cnt=1 then begin
            A0n[cnt]:=1;
            A1n[cnt]:=-1;
            A2n[cnt]:=0;
            B1n[cnt]:=(Alp-abs(Spn[cnt].r))/(Alp+abs(Spn[cnt].r));
            B2n[cnt]:=0;
        end
        else begin
            A0n[cnt]:=1;
            A1n[cnt]:=-2*Zzn[cnt].r;
            A2n[cnt]:=sqr(Zzn[cnt].r)+sqr(Zzn[cnt].ima);
            B1n[cnt]:=-2*Zpn[cnt].r;
            B2n[cnt]:=sqr(Zpn[cnt].r)+sqr(Zpn[cnt].ima);
        end;
    end;
end;
end;
end;
end;
procedure cal_bs;
var    templ    :complex;
        size    :real;
begin
    for cnt:=1 to n do begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

size:=sqrt(Spn[cnt].r)+sqrt(Spn[cnt].ima);
temp1.s(-abs(Spn[cnt].r)/size,Spn[cnt].ima/size);
Spn[cnt]:=temp1;
end;
end;
procedure cal_bp_bs;
var    temp1_com :complex;
        temp2_com :complex;
        up,down  :real;
begin
if type_of_filter=4 then begin
    cal_bs;
    Beta:=power(power(10.0.1*Ap)-1,0.5/order);
end
else begin
    Beta:=power(power(10.0.1*Ap)-1,-0.5/order);
end;
Wu:=tan(pi*Fp2/Sam_fre);
Wl:=tan(pi*Fp1/Sam_fre);
Wo:=Wu*Wl;
Del_w:=Wu-Wl;
for cnt:=1 to n do begin
    C1[cnt]:=sqrt(Spn[cnt].ima)-sqrt(Spn[cnt].r)+4*Wo/sqrt(Del_w*Beta);
    C2[cnt]:=sqrt(sqrt(C1[cnt])+sqrt(2*Spn[cnt].r*Spn[cnt].ima));
    C3[cnt]:=sqrt((C2[cnt]-C1[cnt])/2);
    C4[cnt]:=sqrt((C2[cnt]+C1[cnt])/2);

    if (trunc(order)mod 2=0) then begin
        Sbp1[cnt].s(-abs(Spn[cnt].r)-C3[cnt],Spn[cnt].ima+C4[cnt]);
        Sbp2[cnt].s(-abs(Spn[cnt].r)+C3[cnt],Spn[cnt].ima-C4[cnt]);
    end
    else begin
        if cnt=1 then begin
            Qo:=(1-power((1-power(K.2)),0.25))/(2*(1+power((1-power(K.2)),0.25)));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Q :=Qo+(2*power(Qo,5))+(15*power(Qo,9))+(150*power(Qo,13));
V :=(1/(2*order))*ln((power(10,0.05*Ap)+1)/(power(10,0.05*Ap)-1));
up:=0;
for cnt2:=0 to 3 do begin
    up:=up+power(-1,cnt2)*power(Q,cnt2*(cnt2+1))*sinh((2*cnt2+1)*V);
end;
up:=up*(-2)*power(Q,0.25);
down:=0;
for cnt2:=1 to 4 do begin
    down:=down+power(-1,cnt2)*power(Q,cnt2*cnt2)*cosh(2*cnt2*V);
end;
down:=down*2+1;
Sig_o:=up/down;
if type_of_filter=3 then begin
    Sbp1[cnt].s(Sig_o,sqrt(C1[cnt]));
    Sbp2[cnt]:=Sbp1[cnt];
end
else begin
    Sbp1[cnt].s(-1/Spn[cnt].r,sqrt(C1[cnt]));
    Sbp2[cnt]:=Sbp1[cnt];
end;
{last}
end
else begin
    Sbp1[cnt].s(-abs(Spn[cnt].r)-C3[cnt],Spn[cnt].ima+C4[cnt]);
    Sbp2[cnt].s(-abs(Spn[cnt].r)+C3[cnt],Spn[cnt].ima-C4[cnt]);
end;
end;

temp1_com.s(Beta*Del_w*0.5,0);
mul_com(Sbp1[cnt],temp1_com,temp2_com);
Sbp1[cnt]:=temp2_com;

```

```
temp1_com.s(Beta*Del_w*0.5,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mul_com(Sbp2[cnt],temp1_com,temp2_com);
Sbp2[cnt]:=temp2_com;

if type_of_filter=4 then begin
  for cnt2:=1 to n do begin
    Sbz[cnt].s(0.sqr(Wo));
    Dbz[cnt]:=1+sqr(Sbz[cnt].ima);
    Zbz[cnt].s((1-sqr(Sbz[cnt].ima))/Dbz[cnt],2*Sbz[cnt].ima/Dbz[cnt]);
  end;
end;

Db1[cnt]:=1+2*abs(Sbp1[cnt].r)+sqr(Sbp1[cnt].r)+sqr(Sbp1[cnt].ima);
Db2[cnt]:=1+2*abs(Sbp2[cnt].r)+sqr(Sbp2[cnt].r)+sqr(Sbp2[cnt].ima);

Zbp1[cnt].s((1-(sqr(Sbp1[cnt].r)+sqr(Sbp1[cnt].ima)))/Db1[cnt],
  2*Sbp1[cnt].ima/Db1[cnt]);
Zbp2[cnt].s((1-(sqr(Sbp2[cnt].r)+sqr(Sbp2[cnt].ima)))/Db2[cnt],
  2*Sbp2[cnt].ima/Db2[cnt]);
end;

if (trunc(order) mod 2=0) then begin
  for cnt:=1 to n do Zb[cnt]:=Zbp1[cnt];
  for cnt:=1 to n do Zb[n+cnt]:=Zbp2[cnt];
end
else begin
  Zb[1]:=Zbp1[1];
  for cnt:=2 to n do Zb[(cnt-1)*2]:=Zbp1[cnt];
  for cnt:=2 to n do begin
    if cnt=2 then Zb[3]:=Zbp2[2];
    Zb[cnt+2]:=Zbp2[cnt];
  end;
end;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

procedure cal_ps_A_B;
begin
  case type_of_filter of
    3:begin
      if (trunc(order)mod 2=0) then n:=n*2 else n:=n*2-1;
      for cnt:=1 to n do begin
        A0n[cnt]:=1;
        A1n[cnt]:=0;
        A2n[cnt]:=-1;
      end;
    end;
    4:begin
      if (trunc(order)mod 2=0) then n:=n*2 else n:=n*2-1;
      for cnt:=1 to n do begin
        A0n[cnt]:=1;
        A1n[cnt]:=-2*Zbz[cnt].r;
        A2n[cnt]:=1;
      end;
    end;
  end;
  for cnt:=1 to trunc(order) do begin
    B1n[cnt]:=-2*Zb[cnt].r;
    B2n[cnt]:=sqrt(Zb[cnt].r)+sqrt(Zb[cnt].ima);
  end;
  [last]
end;

```

```

procedure cal_iir;
var  cnt  :byte;
begin
  cal_k;
  A :=sqrt((power(10,0.1*As)-1)/(power(10,0.1*Ap)-1));
  Order:=trunc(log(A)/log(1/K))+1;
  if (trunc(order)mod 2)=0 then begin
    n:=trunc(order) div 2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
else begin
    n:=(trunc(order)+1)div 2;
end;
for cnt:=1 to n do begin
    Spn[cnt].s(-sin((2*cnt-1)*pi/(2*order)),cos((2*cnt-1)*pi/(2*order)));
end;
case type_of_filter of
    1,2:begin
        cal_lp_hp;
        cal_lh_A_B;
        writeln('-----');
        writeln('    Order is ',trunc(order):3);
        writeln('-----');
    end;
    3 :begin
        cal_bp_bs;
        cal_ps_A_B;
        order:=order*2;
        writeln('-----');
        writeln('    Order is ',trunc(order):3);
        writeln('-----');
    end;
    4 :begin
        cal_bp_bs;
        cal_ps_A_B;
        order:=order*2;
        writeln('-----');
        writeln('    Order is ',trunc(order):3);
        writeln('-----');
    end;
end;

end;

end;

procedure show_A_B;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  for cnt:=1 to n do begin
    writeln('_____');
    real2str(A0n[cnt],srib,stria);
    writeln('A0n',cnt,' = ',srib,'.',stria,'          ',A0n[cnt]);
    real2str(A1n[cnt],srib,stria);
    writeln('A1n',cnt,' = ',srib,'.',stria,'          ',A1n[cnt]);
    real2str(A2n[cnt],srib,stria);
    writeln('A2n',cnt,' = ',srib,'.',stria,'          ',A2n[cnt]);

    real2str(B1n[cnt],srib,stria);
    writeln('B1n',cnt,' = ',srib,'.',stria,'          ',B1n[cnt]);
    real2str(B2n[cnt],srib,stria);
    writeln('B2n',cnt,' = ',srib,'.',stria,'          ',B2n[cnt]);
    writeln('_____');
    ch:=readkey;
  end;
end;
begin
  repeat_again:=true;
  repeat
    clrscr;
    writeln('What type of filter?');
    writeln('1.Low pass');
    writeln('2.High pass');
    writeln('3.Band pass');
    writeln('4.Band stop');
    cy:=whereY;
  repeat
    gotoxy(1,cy);
    write('_____');
    gotoxy(1,cy);
    write('Enter type of filter : ');
    readln(type_of_filter);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

until type_of_filter in [1,2,3,4];
user_input;
cal_iir;
show_A_B;
show;
[send data]
send_data_iir;
write('Do you want to select another filter (y/n)?');
repeat
    ch:=readkey;
    if ch in ['y','Y','n','N'] then begin
        ch:=upcase(ch);
    end;
until ch in ['Y','N'];
writeln(ch);
if ch='Y' then repeat_again:=true else repeat_again:=false;
until repeat_again=false;
end;
begin
    [clear port printer]
    port[$378]:=0;
    repeat
        clrscr;
        writeln('1. Use FIR Filter. ');
        writeln('2. Use IIR Filter. ');
        write('Please select type of filter : ');
        choice:=0;
        repeat
            ch:=readkey;
            choice:=ord(ch)-ord('0');
        until choice in [1,2];
        writeln(choice);
        assign(fi,'fil-type');
        rewrite(fi);

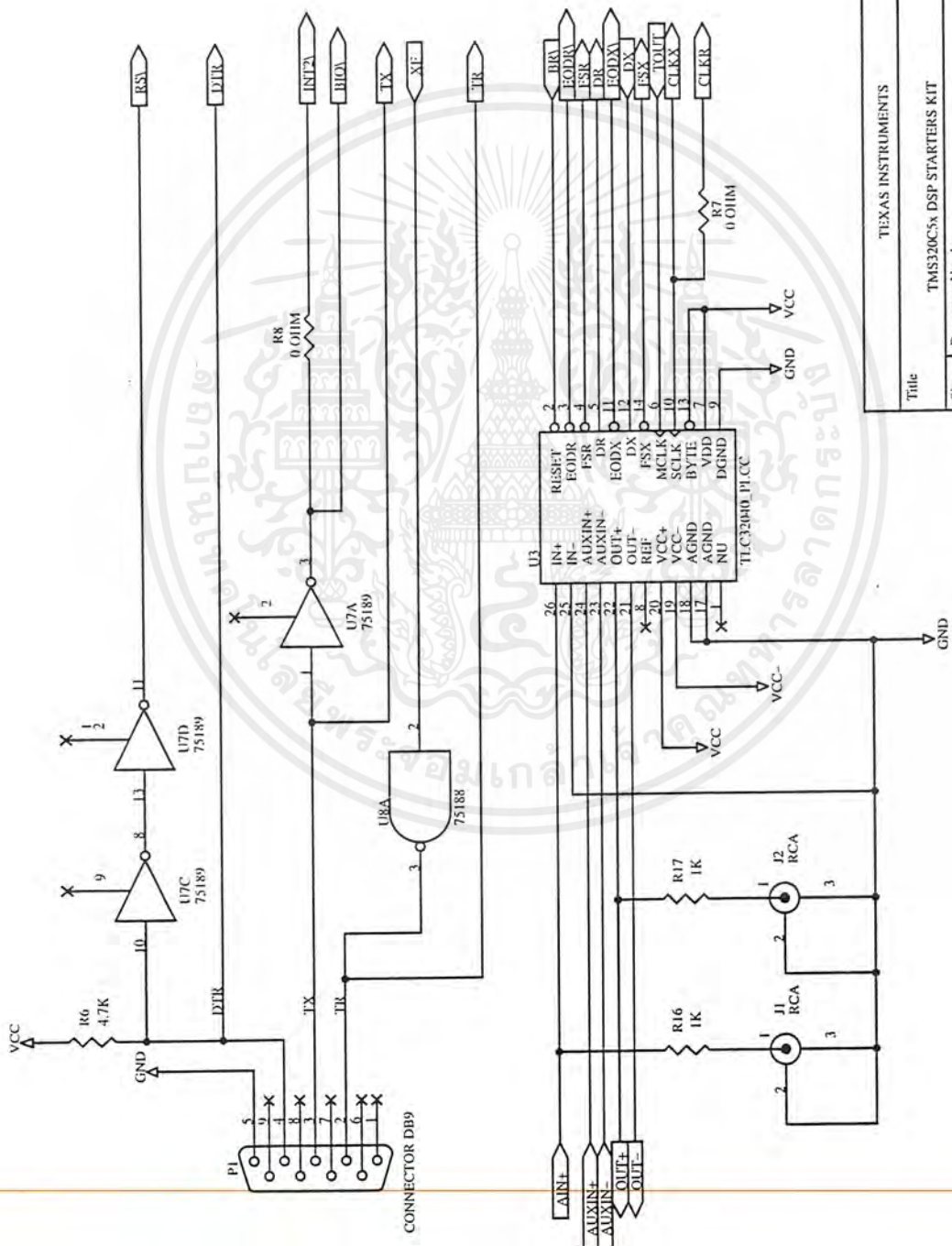
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
writeln(fi.choice);
close(fi);
delay(500);
if choice=1 then fir else iir;
clrscr;
gotoxy(20,13);
write('Do you want to calculate again (y/n)?');
repeat
  ch:=readkey;
  if ch in['y','Y','n','N'] then begin
    ch:=upcase(ch);
    end;
until ch in['Y','N'];
writeln(ch);
if ch='Y' then repeat_main:=true else repeat_main:=false;
until repeat_main=false;
end.
```


หนังสืออ้างอิง

1. ศ.ดร. วัลลภ สุระกำพลธร, “การประมวลผลสัญญาณเชิงเลข” ,คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ,348 หน้า ,2533
2. Gobind Daryanani ,”Principles of Active Network Synthesis and Design” ,John Wiley
& Sons,495 p. , 1976.
3. Fred J. Taylor ,”Principles of Signals and Systems” ,Mc Graw-Hill ,562p. ,1994.
4. Texas Instruments, “ TMS320C5X User’s Guide “ ,1993
5. Texas Instruments, “ TMS320C5X DSP Starter Kit User’s Guide “ ,1994
6. David J. Defatta,” Digital Signal Processing : A System Design Approach” ,John Wiley &
Sons ,661 p., 1988.
7. Seehyun Kim,” A Floating-Point to Fixed-Point Assembly Program Translator for the
TMS 320C25” , IEEE Trans.Analog and Digital Signal Processing,vol. 41,no. 11,pp. 730-739,
Nov. 1994.



TEXAS INSTRUMENTS

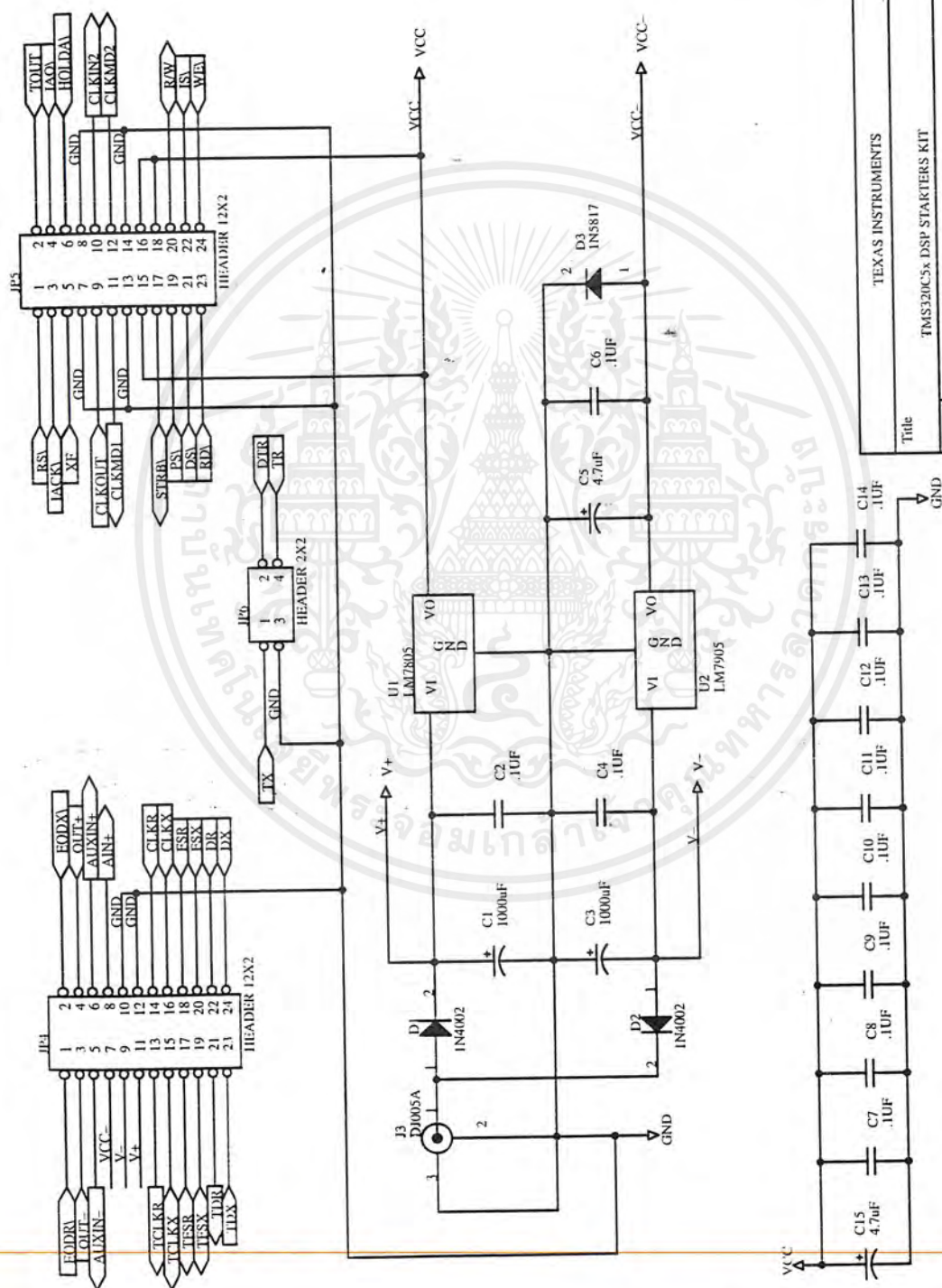
Title

TMS320C5x DSP STARTERS KIT

Size	Document Number
------	-----------------

2617687

Date: January 17, 1994



TEXAS INSTRUMENTS	
TMS320C5x DSP STARTERS KIT	
Title	REV
Size	A
Document Number	2617687
Date	January 21, 1991
Sheet	5 of 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

