สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดการทดลอง DSP

DSP LABORATORY

โดย

นายชัยธรณ์ จิระวงศ์อร่าม

นายณัฐชัย วัชราภินชัย

เลขหมู่.....
เลขทะเบียน 50344
วัน,เดือน,ปี 13 พ.ค. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดการทดลอง DSP

DSP LABORATORY

โดย

นายชัยธรณ์ จิระวงศ์อร่าม 42010078

นายณัฐชัย วัชราภินชัย 42010104

อาจารย์ที่ปรึกษา

อาจารย์เทอดศักดิ์ ลิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดการทดลอง DSP

จัดทำโดย

1. นายชัยธรณ์ จิระวงศ์อร่าม
2. นายณัฐชัย วัชราภินชัย

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบ

.....

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฟอร์มรับรองความพร้อมการสอบ

ปริญญานิพนธ์เรื่อง ชุดทดลอง DSP

DSP laboratory

จัดทำโดย

1. นายชัยธรณ์ จิระวงศ์อร่าม
2. นายณัฐชัย วัชราภินชัย

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบ

.....

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดการทดลอง DSP

ชัยธรณ์ จิระวงศ์อร่าม (42010078)
ณัฐชัย วัชราภินชัย (42010104)

อาจารย์ เทอดศักดิ์ ลิ่วหาทอง
(อาจารย์ที่ปรึกษา)
ปีการศึกษา 2545

บทคัดย่อ

โครงงานชิ้นนี้เป็นโครงงานที่เกี่ยวข้องกับการออกแบบการและพัฒนาการชุดทดลองที่ใช้ในการประมวลผลเชิงเลข(Digital Signal Processing :DSP) ซึ่งนำเอาทฤษฎีต่างๆที่เกี่ยวข้องกับการประมวลสัญญาณเชิงเลขมาจำลองเป็นการทดลองแบบต่างๆ เช่น การทดลองผลของการ Aliasing และ Fold-back,การทดลองสร้างสัญญาณรูปแบบต่างๆ และการทดลองการสร้างตัวกรองความถี่แบบ FIR และ IIR โดยใช้บอร์ดทดลอง DSP starter KIT ซึ่งใช้ TMS320c26 เป็นตัวประมวลผล โดยในส่วนของการทดลอง FIR และ IIR จะแบ่งออกเป็น 2 ส่วนที่สำคัญคือ การเขียนโปรแกรมการประมวลผลของ DSP starter KIT และ การออกแบบผลการตอบสนองอิมพัลส์โดยใช้โปรแกรม MAthlab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DSP LABORATORY

Chaiyatorn Chirawongaram (42010078)

Nattachai Watcharapinchai (42010104)

Thurdsak Leauhatong (Advisor)

Education Year 2002

ABSTRACT

In this project ,we present digital signal processing (DSP) laboratory by using DSP Starter Kit board and TMS320C26 is the central processing unit .The DSP laboratories use digital signal processing theories about Aliasing and Fold-back, generation some periodic, FIR and IIR filter design. In FIR and IIR filter design section consist 2 part: part one implement FIR/IIR designing for TMS320C26 and part two impulse response FIR/IIR designing by mathlab.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	หน้าที่
บทคัดย่อ	**I**
ABSTRACT	**II**
สารบัญ	**III**
สารบัญรูปภาพ	**VI**
สารบัญตาราง	**IX**
บทที่1 บทนำ	1
บทที่2 TMS320C26	2
2.1 สถาปัตยกรรมของ TMS320C26	2
2.2 โครงสร้างทั่วไป	3
2.3 สถาปัตยกรรมโดยทั่วไป	4
2.4 ฟังก์ชั่นบล็อกไดอะแกรม (Function Block Diagram)	8
2.5 Internal Hardware Summary	10
2.6 การจัดหน่วยความจำ	12
2.6.1 หน่วยความจำข้อมูล	13
2.6.2 หน่วยความจำโปรแกรม	14
2.6.3 การจัดผังหน่วยความจำ	14
2.6.4 การจัดรีจิสเตอร์ของผังหน่วยความจำ	16
2.6.5 อ๊อกซิลารี่รีจิสเตอร์ (Auxiliary Register)	16
2.6.6 การอ้างแอดเดรสของหน่วยความจำ	19
2.6.7 การเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำ	20
2.6.8 หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิก	20
2.6.9 สเกลลิ่งชิฟเตอร์	22
2.6.10 แอคคิวมูเลเตอร์ และ ALU	22
2.6.11 ตัวคูณ , รีจิสเตอร์ T และ P	23
2.6.12 โปรแกรมเคาน์เตอร์และสแตค	23
2.6.13 ปฏิบัติการไปป์ไลน์ (Pipeline Operation)	24
2.6.14 รีจิสเตอร์สถานะ	25
2.6.15ปฏิบัติการเวลา	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.16 การนับที่ทำให้เกิดการทำงานซ้ำ (Repeate Counter) 29
2.6.17 โหมดเพาว์เวอร์ดาวน์ 30
2.7 อินเตอร์รัพท์ (Interrupt) 30
2.7.1 การทำงานของอินเตอร์รัพท์ (Interrupt Operation) 30
2.8 พอร์ตอนุกรม (Serial Port) 32
2.9 การส่งและรับข้อมูล (Transmit and Receive Operation) 34
2.9.1 การส่งสัญญาณในแบบ เบิร์สโหมด (Burst- Mode Operation) 36
2.9.2 การรับส่งข้อมูลในโหมดต่อเนื่องแบบใช้สัญญาณเฟรมซิงค์ 38
(Continuous Operation Using Frame Syn Pulse)
2.9.3การรับส่งข้อมูลในโหมดต่อเนื่องแบบไม่ใช้สัญญาณเฟรมซิงค์ 39
(Continuous Operation without Frame Syn Pulse)
2.10 Analog Interface Circuit (AIC) 41
บทที่ 3 ทฤษฎีเบื้องต้น 46
3.1 การประมวลสัญญาณ (Signal processing) 46
3.1.1 โครงสร้างของระบบประมวลผลสัญญาณเชิงเลข 47
(Structure of Digital Signal Processing)
3.1.2 สัญญาณและระบบ (Signal and System) 48
3.1.3 ระบบและการแบ่งระบบ (System and System Classification) 52
3.1.4 การวิเคราะห์ระบบเวลาเต็มหน่วย (Analysis of Discrete System) 64
3.1.5 ฟังก์ชันของระบบ 69
3.1.6 การหาฟังก์ชันของระบบ 70
3.2 การออกแบบวงจรกรองความถี่แบบ FIR (FIR Filter Design) 71
3.2.1 คุณสมบัติความเป็นเชิงเส้นทางเฟสในวงจรกรองความถี่แบบFIR 72
(Properties of Linear-Phase FIR Filter)
3.2.2 ผลการตอบสนองทางความถี่ (Frequency response) 73
3.2.3 การออกแบบ FIR แบบความถี่ต่ำผ่านโดยใช้เทคนิคหน้าต่าง 75
(Window Design Techniques)
3.2.4 การออกแบบวงจรกรองความถี่ชนิดอื่น 80
3.3 การออกแบบวงจรกรองความถี่แบบ IIR (IIR Filter Design) 83
3.3.1 หลักพื้นฐานของวงจรกรองความถี่เชิงอุปมาน 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 คุณสมบัติของต้นแบบของวงจรกรองความเชิงอุปมาน 85
(Characteristics of prototype Analog filters)
3.3.3 Analog-to-digital filter transformations 89
3.3.4 Bilinear transformation 89
3.3.5 Frequency-Band Transformations 90
บทที่ 4 การสร้างวงจรกรองความความถี่เชิงเลขโดยใช้ TMS320C26 92
4.1 ขั้นตอนการออกแบบวงจรกรองความถี่เชิงเลข 92
4.2 การสร้างโปรแกรมสำหรับวงจรกรองความถี่เชิงเลข 92
4.3 การสร้างวงจรกรองความถี่เชิงเลขแบบระบบตัวอย่างเดี่ยวเชิงจำกัด 93
4.4 การเขียนโปรแกรมของตัวกรองความถี่แบบ FIR ด้วย TMS320C26 94
4.5 การสร้างโปรแกรมของตัวกรองความถี่แบบ IIR ด้วย TMS320C26 97
4.6 การสร้างโปรแกรมแบบโครงสร้างแบบตรงแบบที่ 2 ของวงจรกรองความถี่แบบ IIR 98
4.7 การออกแบบวงจรกรองความถี่แบบ IIR แบบเทียมเดียว 101
บทที่ 5 การทดลอง 103
5.1ผลการทดลองเรื่องการสุ่มสัญญาณ 104
5.2 ผลการทดลองเรื่อง FIR และ IIR 108
5.2.1 ผลการทดลองของ FIR ที่ได้จากโปรแกรม MATLAB 109
5.2.2 ผลการทดลองของ IIR ที่ได้จากโปรแกรม MATLAB 115
บทที่ 6สรุปและวิจารณ์ 118
ภาคผนวก 121
กิตติกรรมประกาศ 131
เอกสารอ้างอิง 132

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้าที่
รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์เพื่อทำการทดลอง	1
รูปที่ 2-1 ลักษณะภายนอกของ TMS320C26	4
รูปที่ 2-2 บล็อกไดอะแกรมอย่างง่ายของ TMS320C26	5
รูปที่ 2-3 บล็อกไดอะแกรมภายในของ TMS320C26	9
รูปที่ 2-4 หน่วยความจำข้อมูลภายในชิพของ TMS320C26	13
รูปที่ 2-5 การจัดผังหน่วยความจำหลังจากใช้คำสั่ง CNFP ของ TMS320C26	15
รูปที่ 2-6 ตัวอย่างของอ๊อกซิลารี่รีจิสเตอร์ที่ใช้ในการอ้างแอดเดรสแบบทางอ้อม	17
รูปที่ 2-7 Auxiliary Register File	18
รูปที่ 2-8 ส่วนประกอบในคำสั่งซึ่งใช้ในการอ้างแอดเดรสแบบต่างๆ	19
รูปที่ 2-9 หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิก	21
รูปที่ 2-10 โครงสร้างของสแตค และ โปรแกรมเคาน์เตอร์	24
รูปที่ 2-11 การจัดรูปแบบของรีจิสเตอร์สถานะ (Status Register)	26
รูปที่ 2-12 บล็อกไดอะแกรมของวงจรตั้งเวลา	29
รูปที่ 2-13 แสดง Interrupt mask register	31
รูปที่ 2-14 แสดง DXR และ DRR รีจิสเตอร์	33
รูปที่ 2-15 แสดง Serial Port Block Diagram	34
รูปที่ 2-16 แสดงผังเวลาในการส่งของ Serial Port	35
รูปที่ 2-17 แสดงผังเวลาในการรับของ Serial Port	35
รูปที่ 2-18 แสดงผังเวลาของการส่งในแบบ Burst- Mode	37
รูปที่ 2-19 แสดงผังเวลาของการรับในแบบ Burst- Mode	37
รูปที่ 2-20 แสดงผังเวลาของการส่ง Continuous แบบใช้ Frame Syn Pulse	38
รูปที่ 2-21 แสดงผังเวลาของการรับ Continuous แบบใช้ Frame Syn Pulse	39
รูปที่ 2-22 แสดงผังเวลาของการส่ง Continuous แบบไม่ใช้ Frame Syn Pulse	40
รูปที่ 2-23 แสดงผังเวลาของการรับ Continuous แบบไม่ใช้ Frame Syn Pulse	40
รูปที่ 2-24 แสดง Analog Interface Circuit Chip	41
รูปที่ 2-25 ฟังก์ชันบล็อกไดอะแกรมของ AIC	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-1 การประมวลสัญญาณเชิงอุปมาน และการประมวลสัญญาณเชิงเลข 46
รูปที่ 3-2 ระบบการประมวลผลสัญญาณเชิงเลข 47
รูปที่ 3-3 ส่วนเปลี่ยนสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข 48
รูปที่ 3-4 แสดงสัญญาณที่ต่อเนื่องและสัญญาณไม่ต่อเนื่อง 49
รูปที่ 3-5 แสดงสัญญาณเชิงอุปมานและสัญญาณเชิงเลข 49
รูปที่ 3-6 แสดงสัญญาณที่ไม่มีคาบเวลาและสัญญาณที่มีคาบเวลา 50
รูปที่ 3-7 แสดงการสุ่มสัญญาณ $\delta(t) \leftrightarrow \delta(\omega); x(t) \leftrightarrow X(\omega)$ 55
รูปที่ 3-8 แสดงการสุ่มสัญญาณอธิบายในโดเมนความถี่ 56
รูปที่ 3-9 แสดงการเกิด Aliasing Effect ในโดเมนความถี่ 56
รูปที่ 3-10 แสดงการสร้างคืนสัญญาณที่ถูกสุ่มอธิบายในโดเมนความถี่ 57
รูปที่ 3-11 ชุดแถวที่ได้จากการสุ่มสัญญาณต่อเนื่อง 59
รูปที่ 3-12 ชุดแถวพื้นฐาน 60
รูปที่ 3-13 ชุดแถวของ $x(n) = A\alpha^n$ เมื่อ α มีค่าต่างๆ 61
รูปที่ 3-14 ชุดแถวรูปซายน์ 62
รูปที่ 3-15 แสดงการบวกและการคูณกันระหว่างชุดแถว 62
รูปที่ 3-16 แสดงอินพุตและเอ้าท์พุตของระบบ 63
รูปที่ 3-17 แสดงการบวกสัญญาณ 2 สัญญาเข้าด้วยกัน 63
รูปที่ 3-18 แสดงการคูณสัญญาณด้วยค่าคงที่ 63
รูปที่ 3-19 แสดงการคูณสัญญาณ 2 สัญญาเข้าด้วยกัน 63
รูปที่ 3-20 แสดงการหน่วงเวลา 64
รูปที่ 3-21 FIR filter specification:(a)Absolute (b)Relative 72
รูปที่ 3-22 Bartlett (triangular) window of frequency response 77
รูปที่ 3-23 Hanning window of frequency response 77
รูปที่ 3-24 Hamming window of frequency response 78
รูปที่ 3-25 Blackman window of frequency response 78
รูปที่ 3-26แสดงการหาวงจรกรองความถี่แถบความถี่ผ่าน 80
รูปที่ 3-27 แสดงการหาวงจรกรองความถี่สูงผ่าน 81
รูปที่ 3-28 แสดงการหาวงจรกรองความถี่แบบหยุดความถี่ 82
รูปที่ 3-29 block diagram แสดงถึงการออกแบบ IIR filter 83
รูปที่ 3-30 Analog lowpass filter specifications 84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-31 typical pole-zero pattern of $H_s(s)H_s(-s)$ 85
รูปที่ 3-32 Magnitude-squared response of butter-worth lowpass filter 85
รูปที่ 3-33 magnitude-squared response of Chebyshev lowpass filter 87
รูปที่ 3-34 Magnitude-squared response of elliptic filter 88
รูปที่ 3-35 Complex-plane mapping in bilinear transformation 89
รูปที่ 3-36 ชนิดของfilterแบบต่างๆ 91
รูปที่ 4-1 โครงสร้างของแบบตรงแบบที่ 1 ของ FIR filter 93
รูปที่ 4-2 แสดงถึงการเก็บข้อมูลอินพุทในหน่วยความจำข้อมูล 94
รูปที่ 4-3 แสดงความสัมพันธ์ของค่าในหน่วยความจำข้อมูล 95
รูปที่ 4-4 แสดงถึงการจัดเก็บผลการตอบสนองอิมพัลส์ในหน่วยความจำข้อมูล 96
รูปที่ 4-5 โครงสร้างแบบตรงแบบที่ 1 98
รูปที่ 4-6 โครงสร้างแบบตรงแบบที่ 2 99
รูปที่ 4-7 วงจรกรองความถี่แบบ IIR ลำดับที่ 2 โครงสร้างแบบตรงแบบที่ 2 99
รูปที่ 4-8 การเก็บของค่าของการหน่วงเวลาในหน่วยความจำข้อมูล 100
รูปที่ 4-9 การเก็บของค่าการหน่วงเวลาของหน่วยความจำข้อมูล 100
รูปที่ 4-10 วงจรกรองความถี่ IIR อันดับ 4 แบบ Cascade 101
รูปที่ 4-11 การจัดหน่วยความจำของค่าการหน่วงเวลา แบบ Cascade 102
รูปที่ 5-1 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 100Hz 104
รูปที่ 5-2 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 500Hz 105
รูปที่ 5-3 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 1kHz 105
รูปที่ 5-4 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 4 kHz 106
รูปที่ 5-5 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 5 kHz 106
รูปที่ 5-6 รูปของ output ที่เกิดจากสัญญาณ Sine wave 10kHz 107
รูปที่ 5-7 Magnitude response ของตัวกรองความถี่แบบ retangecular window 109
รูปที่ 5-8 Magnitude response ของตัวกรองความถี่แบบ triangle window 110
รูปที่ 5-9 Magnitude response ของตัวกรองความถี่แบบ hanning window 111
รูปที่ 5-10 Magnitude response ของตัวกรองความถี่แบบ hamming window 112
รูปที่ 5-11 Magnitude response ของตัวกรองความถี่แบบ blackman window 113
รูปที่ 5-12 Magnitude response ของตัวกรองความถี่แบบ Kaiser window 114
รูปที่ 5-13 Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ butterworth 115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5-14 Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ chebyshev 116
รูปที่ 5-15Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ elliptic 116
รูปที่ 5-16 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ butterworth 116
รูปที่ 5-17 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ chebyshev 117
รูปที่ 5-18 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ elliptic 117

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้าที่

ตารางที่ 2-1 TMS320C26 Internal Hardware 10

ตารางที่ 2-2 การจัดรีจิสเตอร์ของผังหน่วยความจำ 16

ตารางที่ 2-3 ผลลัพธ์ที่เกิดในแอคคิวมูเลเตอร์ 22

ตารางที่ 2-4 การตรวจสอบสถานะของแอคคิวมูเลเตอร์ 23

ตารางที่ 2-5 แสดง Interrupt Location และ Priority 30

ตารางที่ 2-6 แสดง Serial Port Bits , Pins และ Register 32

ตารางที่ 2-7 แสดงการเซ็ตบิตที่ AIC 44

ตารางที่ 2-8 แสดงการเซ็ตบิตที่ secondary DX serial communication protocol 45

ตารางที่ 3-1 การแปลงแซดของชุดแถวที่สำคัญๆ 66

ตารางที่ 3-2 ตารางการแปลงกลับ 67

ตารางที่ 3-3 Summary of commonly used window function characteristic 79

ตารางที่ 4-1 ตารางการควบคุมการเลือกชนิดของหน่วยความจำในTMS320C26 96

ตารางที่ 6-1 ตารางเปรียบเทียบจุดคัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบจำกัด 118

ตารางที่ 6-2 เปรียบเทียบความถี่คัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบอนันต์ 119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

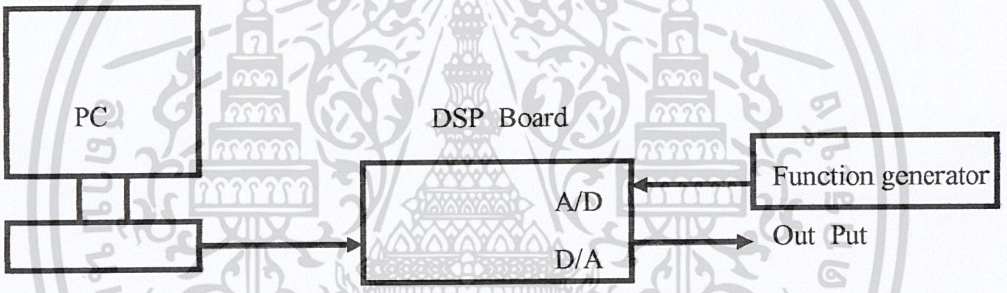
สารบัญตาราง

ตารางที่	หน้าที่
ตารางที่ 2-1 TMS320C26 Internal Hardware	10
ตารางที่ 2-2 การจัดรีจิสเตอร์ของผังหน่วยความจำ	16
ตารางที่ 2-3 ผลลัพธ์ที่เกิดในแอคคิวมูเลเตอร์	22
ตารางที่ 2-4 การตรวจสอบสถานะของแอคคิวมูเลเตอร์	23
ตารางที่ 2-5 แสดง Interrupt Location และ Priority	30
ตารางที่ 2-6 แสดง Serial Port Bits , Pins และ Register	32
ตารางที่ 2-7 แสดงการเซ็ตบิตที่ AIC	44
ตารางที่ 2-8 แสดงการเซ็ตบิตที่ secondary DX serial communication protocol	45
ตารางที่ 3-1 การแปลงแซดของชุดแถวที่สำคัญๆ	66
ตารางที่ 3-2 ตารางการแปลงกลับ	67
ตารางที่ 3-3 Summary of commonly used window function characteristic	79
ตารางที่ 4-1 ตารางการควบคุมการเลือกชนิดของหน่วยความจำในTMS320C26	96
ตารางที่ 6-1 ตารางเปรียบเทียบจุดคัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบจำกัด	119
ตารางที่ 6-2 เปรียบเทียบความถี่คัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบอนันต์	120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ชุดทดลองการประมวลผลสัญญาณเชิงเลข หรือ ดิจิตอลซิกแนลโปรเซสซิง (Digital Signal Processing : DSP) นี้เป็นการนำเอาความรู้ทางด้านการประมวลผลเชิงเลขมาทำการสร้างเป็นการทดลองต่างๆที่เกี่ยวข้องกับการประมวลผลเชิงเลขทั้งหมด 4 การทดลองคือ การทดลองเกี่ยวกับผลของความถี่การสุ่มสัญญาณ ,การทดลองเกี่ยวกับการสร้างสัญญาณในรูปแบบต่างๆ การทดลองการสร้างตัวกรองเชิงเลขแบบ FIR (Finite Impulse Response Digital Filter) หรืออาจจะเรียกได้ว่าเป็นตัวกรองชนิดไม่มีการป้อนกลับ (Nonrecursive Filter) และ การทดลองการสร้างตัวกรองแบบ IIR หรืออาจจะเรียกได้ว่าเป็นตัวกรองแบบมีการป้อนกลับ (Recursive Filter)



รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์เพื่อทำการทดลอง

ชุดทดลองการประมวลผลสัญญาณเชิงเลขนี้ใช้บอร์ด DSK (DSP Starter Kit) ที่มีซีพียูเบอร์ TMS320C26 เป็น ตัวประมวลผลทำการอินเตอร์เฟสกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม และ ใช้โปรแกรมบน PC โดยที่ผู้ใช้ไม่ต้องเขียนโปรแกรมภาษาแอสเซมบลีบน DSK ให้ยุ่งยากเพื่อควบคุมการทำงานของ DSK การทดลองแบบต่างๆ

ชุดทดลองการประมวลผลสัญญาณเชิงเลขนี้ สามารถทำเป็นตัวกรองความถี่ได้ในช่วงของความถี่เสียง (Voice) เพราะมีข้อจำกัดของการแปลงดิจิตอลเป็นอนาลอก (D/A) และอนาลอกเป็นดิจิตอล (A/D) ของไอซีเบอร์ TLC32040 ที่เป็นวงจรอินเตอร์เฟสสัญญาณอนาลอก ที่มีความถี่สุ่ม (Sampling Frequency) ในช่วงความถี่ของเสียง

ในการทำการทดลองจะต้องอาศัยสัญญาณที่ได้รับจาก Signal generator และในบางการทดลองต้องอาศัยการวัดโดยใช้ Oscilloscope เพื่อทำการวัดผลที่ได้จากการทดลอง และ ในบางการทดลองสามารถดูผลจากการทดลองได้จากจอคอมพิวเตอร์ในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

2. โครงสร้างของ DSP Board

2.1 สถาปัตยกรรมของ TMS320C26

สถาปัตยกรรมและโครงสร้างภายในโดยทั่วๆ ไปของTMS320C26 จะเห็นได้ว่าโครงสร้างภายในต่างๆ นั้นได้มีการพัฒนา และเพิ่มขีดความสามารถของการทำงานในด้านต่างๆ ขึ้นจากรุ่นก่อนคือ TMS320C1X มาก ได้แก่ การคูณเลขขนาด 16x16 บิต ใช้เวลาทำงานเพียงไซเกิลเดียว คือ 100 นาโนวินาทีเท่านั้น และในการต่อกับหน่วยความจำภายนอกสามารถอ้างแอดเดรส ในส่วนของหน่วยความจำข้อมูล และหน่วยความจำโปรแกรมอย่างละ 64 กิโลเวิร์ด และมีจำนวนฮาร์ดแวร์สแตคเพิ่มขึ้นถึง 8 ระดับ เป็นต้น นอกจากนี้ยังประกอบด้วยส่วนต่างๆ ที่เกี่ยวข้อง ซึ่งจะเป็นการเอื้ออำนวยประโยชน์ในการนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง ส่วนต่างๆ สรุปได้ดังนี้

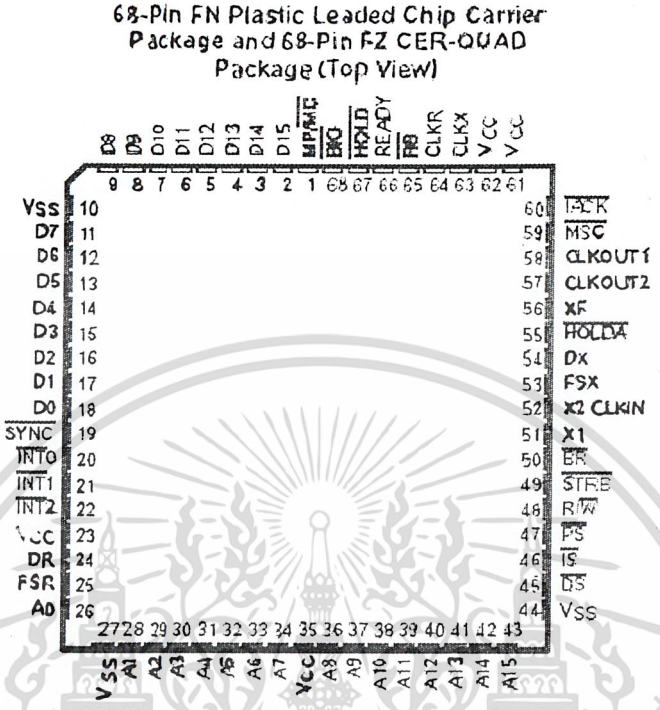
- โครงสร้างโดยทั่วไป
- สถาปัตยกรรมโดยทั่วไป
- ฟังก์ชั่นบล็อกไดอะแกรม (Function Block Diagram)
- การจัดหน่วยความจำ
- หน่วยความจำข้อมูล
- หน่วยความจำโปรแกรม
- การจัดผังหน่วยความจำ
- การจัดรีจิสเตอร์ของผังหน่วยความจำ
- อ๊อกซิลารี่รีจิสเตอร์ (Auxiliary Register)
- โหมดการอ้างแอดเดรสของหน่วยความจำ
- การเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำ
- หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิก
- สเกลลิ่งชิฟเตอร์ (Scaling Shifter)
- แอคคิวมูเลเตอร์ และ ALU
- ตัวคูณ รีจิสเตอร์ T และ P
- ส่วนควบคุมระบบ
- ปฏิบัติการไปป์ไลน์ (Pipeline)
- รีเซท (Reset)
- รีจิสเตอร์สถานะ (Status Register)
- ปฏิบัติการเกี่ยวกับเวลา (Timer Operation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างทั่วไป

- วัฏจักรในการทำงานของคำสั่งใช้เวลา 100 นาโนวินาที
- มีหน่วยความจำข้อมูลภายในตัวชิพ (On –Chip) ขนาด 544 กิโลเวิร์ด
- มีหน่วยความจำโปรแกรมภายในตัวชิพ (On –Chip) ขนาด 4 กิโลเวิร์ด
- สามารถต่อหน่วยความจำเพิ่มในส่วนของ ข้อมูล/ โปรแกรม ได้รวมทั้งหมด 28 กิโลเวิร์ด
- ALU และแอคคิวมูเลเตอร์ (Accumulator) มีขนาด 32 บิต
- การคูณจะเป็นแบบขนานขนาด 16x16 บิต ซึ่งจะได้ผลลัพธ์เป็นเลข 32 บิต
- สามารถทำการคูณเลขโดยใช้เวลาเพียงไซเกิลเดียว (100 นาโนวินาที)
- มีคำสั่งในการเคลื่อนย้ายข้อมูล เป็นบล็อกในการจัดการกับข้อมูลหรือโปรแกรม
- มีวงจรนาฬิกา (Timer) ภายในชิพ เพื่อใช้ควบคุมการทำงานต่างๆ
- มี อ๊อกซิลารี่รีจิสเตอร์ (Auxiliary Register) จำนวน 6 ตัว
- เพิ่มจำนวนฮาร์ดแวร์สแตคให้มีเพิ่มขึ้นถึง 8ระดับ
- มีอินพุท และเอาท์พุตขนาด 16 แชนแนล
- มีตัวเลื่อนบิต (Shifter) แบบขนานขนาด 16 บิต
- มีสัญญาณ Wait Status เพื่อสำหรับในการติดต่อกับอุปกรณ์ หรือหน่วยความจำภายนอกที่ทำงานช้ากว่า
- มีพอร์ตอนุกรม เพื่อใช้สำหรับในการเชื่อมต่อ
- มีตัวกำหนดสัญญาณนาฬิกา (Clock) ภายในตัวชิพ
- ต้องการแหล่งจ่ายไฟฟ้าจากภายนอกเพียงชุดเดียว (5V.)
- เป็นอุปกรณ์ที่มีการบรรจุแบบ 68-Lead PLCC
- ใช้เทคโนโลยีในการผลิตแบบซีมอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 ลักษณะภายนอกของ TMS320C26

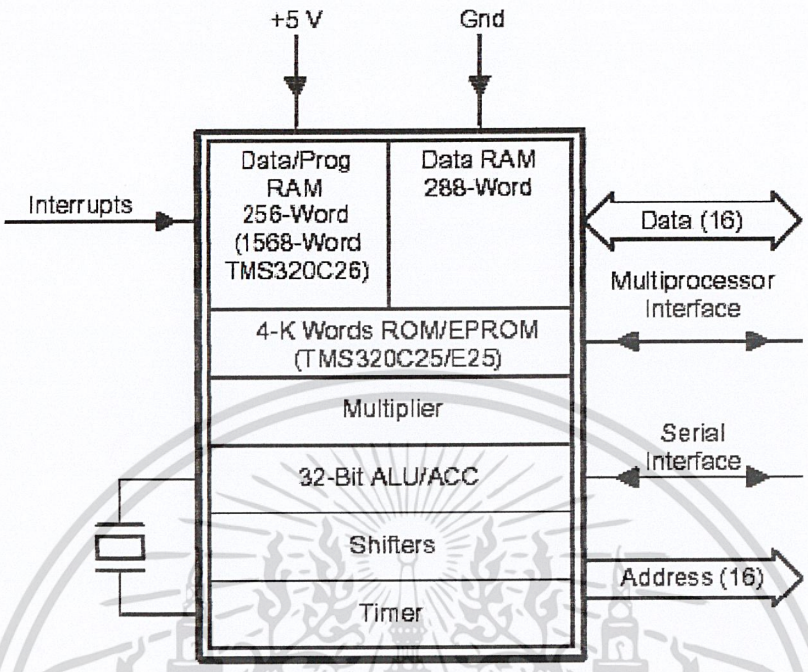
2.3 สถาปัตยกรรมโดยทั่วไป

Hardware Architecture TMS320C2x เป็น digital signal processor ที่มี Hardware Architecture ชนิดที่ทำงานที่กำลังสูงสุดโดยใช้ memory busแยกกันระหว่าง Program memory และ Data memory สำหรับการ execute เต็ม speed คำสั่งนั้นถูกใช้เพื่อโยกย้ายข้อมูลระหว่าง 2 space

Program memory และData memory สามารถที่จะ multiplex ไปใน BUS เดียวกันได้เพื่อว่าจะได้ address range มากที่สุดสำหรับทั้ง 2 space ขณะที่ใช้ขาอุปกรณ์น้อยที่สุด

On-chip Memory TMS320C25 ได้เพิ่มการพลิกแพลงได้ในการออกแบบระบบ โดยการใช้ on-chip RAM 2 block ใหญ่ๆ (รวมเป็น 544 16-bit word) โดยที่ blockใด block หนึ่งเป็นได้ทั้ง program หรือ data memory ตามรูปข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 บล็อกไดอะแกรมอย่างง่ายของ TMS320C26

TMS320C26 มี on-chip RAM 3 block ใหญ่ๆ ซึ่งจะแยก Data และ Program memory ออกจากกัน หรือ ทั้งสามบล็อกเป็น Data memory ที่ต่อเนื่องกันเพื่อเพิ่มการพลิกแพลงได้ในการออกแบบระบบ Data memory ที่สามารถอ้างอิง address โดยตรงภายนอก chip ขนาด 64 K-word ถูกรวมเข้ามาเพื่อให้ความสะดวกสบายของเครื่องมือ DSP algorithms

On-chip masked ROM ขนาดใหญ่บน TMS320C25 สามารถลดราคาของระบบ และจัดเตรียมไว้สำหรับการเป็นsingle-chip DSP ที่แท้จริง programs ที่มีขนาดใหญ่กว่า 4K wordsสามารถเก็บไว้ใน internal program ROM ส่วนที่เหลือจะเก็บไว้ที่ 64K word program memory ภายนอก chipโปรแกรมใหญ่ๆ สามารถ execute ที่ full speed จาก memoryส่วนนี้ programจะถูก Download จาก external ที่ช้ามายัง on-chip RAM สำหรับ full-speed operation

On-chip EPROM ขนาด 4 Kword บนTMS320C25จะเป็น real time code development และ ถูกเปลี่ยนแปลงสำหรับการประเมินค่าอย่างทันทีทันใด ของการปฏิบัติการของระบบคำสั่งนั้นสามารถ execute จาก EPROM ที่ full speed

Memory interface TMS320c2x local memory interface ประกอบด้วย 16-bit parallel data bus (D15-D0 , 16-bit address bus (A15-A0) , 3 pin สำหรับเลือกระหว่าง data/program memory หรือ I/O ($\overline{DS}$, $\overline{PS}$ และ $\overline{IS}$) และสัญญาณควบคุมระบบอื่นๆ R / $\overline{W}$ เป็นสัญญาณควบคุมทิศทางการเคลื่อนย้ายข้อมูล และสัญญาณ $\overline{STRB}$ เป็นสัญญาณ timing เพื่อควบคุมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการใช้

เคลื่อนย้าย เมื่อใช้ on-chip program RAM , ROM / EPROM หรือ external program memory ที่มีความเร็วสูง แล้วนั้น TMS320C2x จะทำงานที่ full speed โดยไม่ต้องรอสถานะการใช้สัญญาณ READY เพื่อรอสถานะสำหรับการสื่อสารกับ off-chip memory ที่ช้ากว่า

8 ระดับของ hardware stack จัดเตรียมไว้สำหรับเก็บค่า content ของ program counter ระหว่าง interrupt และ การเรียกของ subroutine คำสั่งจะสามารถเก็บ context อย่างสมบูรณ์ของอุปกรณ์คำสั่ง PUSH และPOP จะอนุญาตให้ระดับของกลุ่มที่ถูกควบคุมโดย available RAM การ interrupt ที่ใช้คำสั่งนี้จะสามารถ mask ได้

การปฏิบัติการควบคุมทั้งหมดบน TMS320C2x นั้นจะใช้ memory-mapped 16-bit timer, repeat counter , 3 external mask able user interrupts และ internal interrupt ที่สร้างโดย serial port หรือ โดย timer การสร้าง mechanism ป้องกันจากคำสั่งที่ซ้ำหรือกลายเป็น multicycle เนื่องจากสัญญาณ READY หรือจาก hold และinterrupts

Arithmetic Logic Unit TMS320C26 เป็นโปรเซสเซอร์ที่ใช้สำหรับในการประมวลผลสัญญาณเชิงเลข ซึ่งจะมีลักษณะการทำงานคล้ายคลึงกับ TMS320C1x (เป็น CPU เบอร์แรกในตระกูล TMS 320) และใช้สถาปัตยกรรมแบบเดียวกัน คือสถาปัตยกรรมแบบ “ ฮาวาร์ด ”โดยจะแบ่งโครงสร้างของหน่วยความจำออกเป็น 2 ส่วนแยกจากกัน คือ หน่วยความจำโปรแกรม และหน่วยความจำข้อมูล ซึ่งจะทำให้การทำงานต่างๆ เป็นไปด้วยความรวดเร็วสูง นอกจากนี้ยังสามารถทำการโอนย้ายข้อมูลระหว่างกันและกันได้

ในการคำนวณทางคณิตศาสตร์นั้นจะทำการคำนวณโดยใช้เลขแบบทูคอมพลีเมนท์ โดยใช้ ALU และ แอคคิวมูเลเตอร์ ซึ่งมีขนาด 32 บิต ALU ซึ่งเป็นหน่วยกระทำการทางคณิตศาสตร์ และลอจิกจะใช้ตัวกระทำขนาด 16 บิตเวิร์ด ซึ่งมาจากหน่วยความจำข้อมูล หรือ ได้มาจากคำสั่งในการอ้างถึงแบบทันที ในการใช้คำสั่งในการบวกจะแบ่งออกได้เป็น 2 ส่วน คือทางด้านบิตสูง (ACCH) และ ทางด้านบิตต่ำ (ACCL) ซึ่งก็คือบิต 31-16 และบิต 15-0 ตามลำดับ

Multiplier สำหรับการคูณเลขขนาด 16x16 บิต สามารถจะทำการคูณได้ภายในไซเกิลเดียว (ใช้เวลา 100 นาโนวินาที) ในการคูณเลขจะประกอบด้วย 3 ส่วนใหญ่ๆ ด้วยกัน คือ T , P รีจิสเตอร์ และมัลติ-พลายเออร์แอร์แรย์ โดย T จิสเตอร์จะใช้สำหรับเก็บตัวตั้งขนาด 16 บิต ส่วน P รีจิสเตอร์จะใช้เก็บค่าของผลคูณขนาด 32 บิต ค่าของตัวคูณอาจจะมาจากหน่วยความจำข้อมูล หรือ มาจากหน่วยความจำโปรแกรม โดยการใช้คำสั่ง MAC / MAC หรือ ได้มาจากการใช้คำสั่ง MPYK (Multiply Immediate) ภายในตัวชิพจะมีการคำนวณต่างๆ ด้วยความเร็วโดยการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติการพื้นฐานของ DSP ซึ่งได้แก่ การคอนโวลูชั่น (Comvolution), คอร์รีเลชั่น (Correlation) การกรอง (Filtering)

ในส่วนของการเลื่อนบิตของ TMS320C26 นั้นจัดให้มีสเกลลิ่งชิฟเตอร์ (Scaling Shifter) ซึ่งมีอินพุทขนาด 16 บิต ต่ออยู่กับบัสข้อมูลและเอาท์พุทขนาด 32 บิต ซึ่งต่ออยู่กับ ALU โดยสามารถทำการเลื่อนบิตของข้อมูลอินพุทไปทางซ้ายได้ 0-16 บิต โดยบิตซึ่งมีนัยสำคัญต่ำ (LSBs) จะถูกเติมด้วย 0 และบิตที่มีนัยสำคัญสูง (MSBs) อาจจะถูกเติมด้วย 0 หรือใช้เป็นส่วนขยายเครื่องหมาย

นอกจากนี้ยังมีฮาร์ดแวร์สแตคเพิ่มขึ้นถึง 8 ระดับ หน้าที่ของสแตคก็คือ ใช้ในการเก็บค่าของโปรแกรมเคาน์เตอร์ (PC) ระหว่างที่มีการขออินเตอร์รัพท์หรือมีการเรียกใช้โปรแกรมย่อย (Subroutine)

Serial Port on-chip full-diplex serial port ถูกจัดเตรียมไว้เพื่อการสื่อสารโดยตรงกับ serial devices เช่น codecs , serial A/D converters และระบบ serial อื่นๆ สัญญาณ interface จะเข้ากันได้กับ codecs และ serial devices อื่นๆ ด้วยการใช้ external hardware น้อยที่สุด serial port memory maps register ทั้งสอง register นั้นจะสามารถทำงานได้ทั้ง 8-bit byte หรือ 16- bit word mode แต่ละ register จะมี external clock input , frame synchronization input และshift register ที่สัมพันธ์กัน

Multiprocessing Application TMS320C2x จะมีการแบ่งสรร global memory space และ การสื่อสารกับspace นั้นโดยใช้สัญญาณ $\overline{BR}$ (BUS request) และสัญญาณ READY memory mapped register ขนาด 8 bit เกิดจากการแบ่ง global memory (GREG) จำนวน 32K word ของ TMS320C2x Data memory จะทำตัวเหมือน global external memory contents ของ register จะกำหนดขนาดของ global memory space ถ้าคำสั่งกำหนด operand ที่ address ของ GREG นี้ BR จะถูก assert เพื่อควบคุมการร้องขอ BUS ความยาวของ memory cycle จะถูกควบคุมโดยสายที่ต่อกับ READY

Direct Memory Access TMS320C2x มีการเข้าถึงข้อมูลโดยตรง (direct memory access) ไปยัง external program/data memory โดยใช้สัญญาณ $\overline{HOLD}$ และ $\overline{HOLDA}$ ตัวปฏิบัติการอื่นๆ สามารถควบคุม external memory ของ TMS320C2x โดยการทำให้ $\overline{HOLD}$ เป็น LOW ซึ่งเหตุผลนี้เองคือสาเหตุที่ทำไม TMS320C2x ให้ address , data และ central line อยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะ high-impedance Signaling ระหว่าง external Processor และ TMS320C2x สามารถปฏิบัติการโดยใช้การ interrupt บน TMS320C2x นั้นจะมีการทำงาน 2 mode โดยโหมดแรกจะเป็นการ execute ที่ถูกยกเลิกระหว่างที่ $\overline{HOLD}$ เป็น LOW และอีกโหมดก็คือ DMA modeในการที่ TMS320C2X execute program ของมันอย่างต่อเนื่อง ในขณะที่กำลังทำงานจาก internal RAM หรือ ROM ซึ่งทำให้ throughput ใน Application ที่เน้นหนักในด้านข้อมูลเพิ่มขึ้น

2.4 ฟังก์ชั่นบล็อกไดอะแกรม (Function Block Diagram)

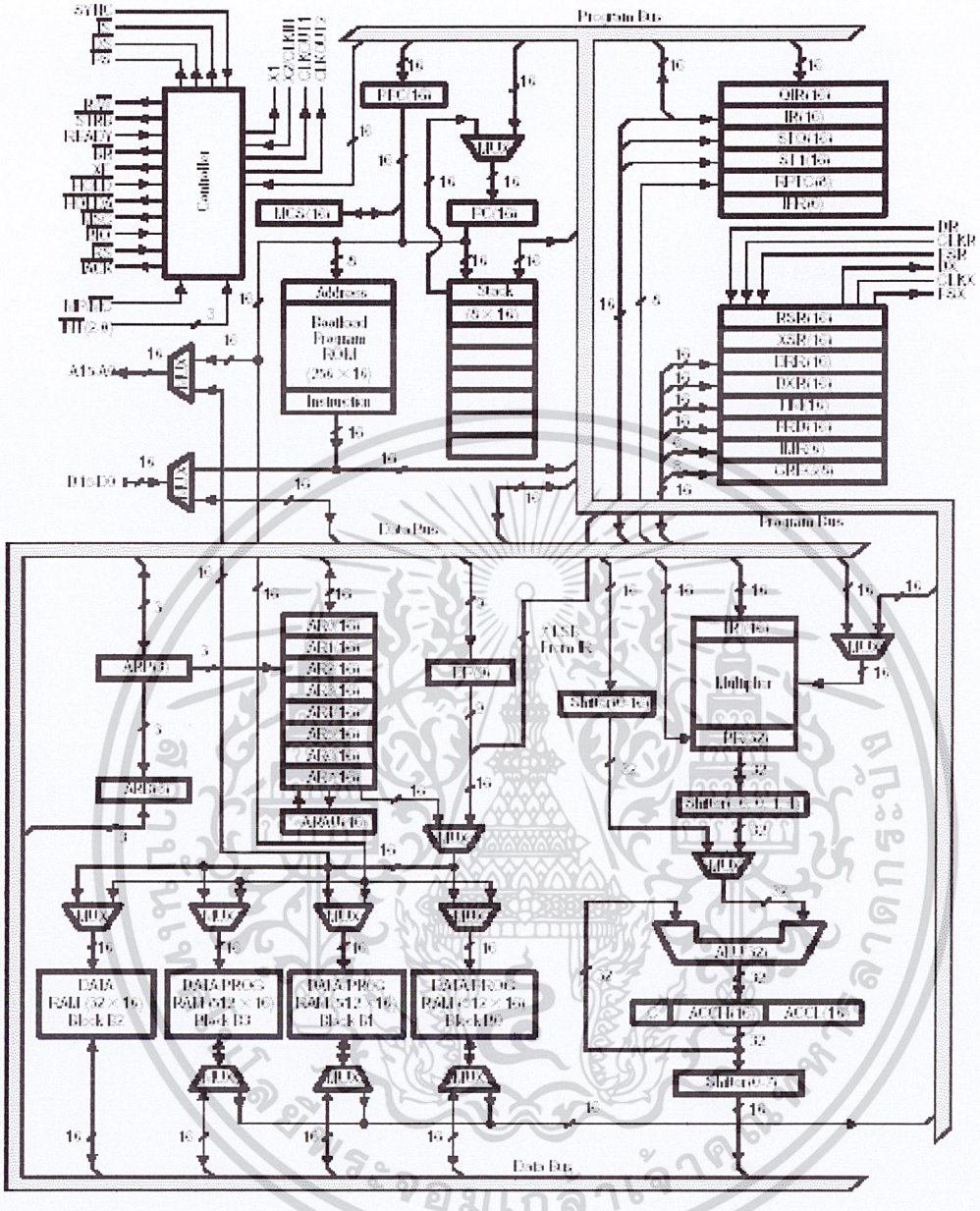
Function Block Diagram จะแสดง Block ที่สำคัญ และเส้นทางเดินของข้อมูลใน TMS320C26 Processor ซึ่งมีความหมายของสัญลักษณ์ที่ใช้ในรูป Block Diagram นั้นแสดงใน Internal Hardware Summaryในรูป Block diagram แสดง interface pin ทั้งหมดของ TMS320C26

สถาปัตยกรรมของ TMS320C26 นั้นจะมี BUS ที่สำคัญ 2 BUS ด้วยกันคือ program bus และ data bus โดยที่ program bus จะนำ instruction code และ immediate operand จาก program memory , data bus จะเชื่อมต่ออุปกรณ์หลายๆอย่างเข้าด้วยกัน เช่น เชื่อม Central arithmetic logic unit (CALU) , register ที่ใช้สนับสนุน (Auxiliary register) เข้ากับ Data RAM program และ data bus สามารถที่จะนำข้อมูลจาก on-chip data RAM และ internal หรือ external program memory ไปยัง multiplier ใน single cycle สำหรับ multiply/accumulate operation

คำอธิบาย

ACCH = Accumulator high
ACCL = Accumulator LOW
ALU = Arithmetic Logic Unit
ARAU = Auxiliary register arithmetic Unit
ARB = Auxiliary register pointer buffer
ARP = Auxiliary register pointer
DP = Data memory page pointer
DRR = Serial port data receive register
DXR = Serial port data transmit register
IFR = Interrupt flag register
IMR = Interrupt mask register
MCS = Micro call stack
QIR = Queue instruction register
PR =Product register
PRD =Period register for timer
TIM =Timer
TR =Temporary register
PC =Program counter
PFC = Prefetch counter
RPTC =Repeat instruction counter
GREG = Global memory allocation register
RSR =Serial port receive shift register
XSR =Serial port receive shift register
ARO-AR=Auxiliary registers
STO.ST=Status registers
C =Carry bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTE: Shaded areas indicate a bus.

รูปที่ 2-3 บล็อกไดอะแกรมภายในของ TMS320C26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Internal Hardware Summary

ตารางข้างล่างนี้จะแสดง Internal Hardwareทั้งหมดของ TMS320C26 ซึ่งจะรวมอุปกรณ์ภายในที่ใช้ Process, register และ buses การจัดกลุ่มของฟังชั่นจะเรียงตามตัวอักษรสัญลักษณ์ที่ใช้ในตารางนี้ จะสอดคล้องกับสัญลักษณ์ที่ใช้ใน block diagram ของ TMS320C26

ตารางที่ 2-1 TMS320C26 Internal Hardware

Unit	Symbol	Function
Accumulator	ACC (31–0) ACCH (31–16) ACCL (15–0)	A 32-bit accumulator split in two halves: ACCH (accumulator high) and ACCL (accumulator low). Used for storage of ALU output.
Arithmetic Logic Unit	ALU	A 32-bit twos-complement arithmetic logic unit having two 32-bit input ports and one 32-bit output port feeding the accumulator.
Auxiliary Register Arithmetic Unit	ARAU	A 16-bit unsigned arithmetic unit used to perform operations on auxiliary register data.
Auxiliary Register File	AR0–AR7 (15–0)	A register file containing eight 16-bit auxiliary registers (AR0–AR7), used for addressing data memory, temporary storage, or integer arithmetic processing through the ARAU.
Auxiliary Register File Bus	AFB(15–0)	A 16-bit bus that carries data from the AR pointed to by the ARP.
Auxiliary Register Pointer	ARP(2–0)	A 3-bit register used to select one of five or eight auxiliary registers.
Auxiliary Register Pointer Buffer	ARB(2–0)	A 3-bit register used to buffer the ARP. Each time the ARP is loaded, the old value is written to the ARB, except during an LST (load status register) instruction. When the ARB is loaded with an LST1, the same value is also copied into ARP.
Central Arithmetic Logic Unit	CALU	The grouping of the ALU, multiplier, accumulator, and scaling shifter.
Data Bus	D(15–0)	A 16-bit bus used to route data.
Data Memory Address Bus	DAB(15–0)	A 16-bit bus that carries the data memory address.
Data Memory Page Pointer	DP(8–0)	A 9-bit register pointing to the address of the current page. Data pages are 128 words each, resulting in 512 pages of addressable data memory space (some locations are reserved).
Direct Data Memory Address Bus	DRB(15–0)	A 16-bit bus that carries the direct address for the data memory, which is the concatenation of the DP register with the seven LSBs of the instruction.
Global Memory Allocation Register	GREG(7–0)	An 8-bit memory-mapped register for allocating the size of the global memory space.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit	Symbol	Function
Instruction Register	IR(15–0)	A 16-bit register used to store the currently executing instruction.
Interrupt Flag Register	IFR(5–0)	A 6-bit flag register used to latch the active-low external user interrupts $\overline{\text{INT}}$(2–0), the internal interrupts XINT/RINT (serial port transmit/receive), and TINT (timer) interrupts. The IFR is not accessible through software.
Interrupt Mask Register	IMR(5–0)	A 6-bit memory-mapped register used to mask interrupts.
Microcall Stack	MCS (15–0)	A single-word stack that temporarily stores the contents of the PFC while the PFC is being used to address data memory with the block move (BLKD/BLKP), multiply-accumulate (MAC/MACD), and table read/write (TBLR/TBLW)and table read/write (TBLR/TBLW) instruction
Multiplier	MULT	A 16 × 16-bit parallel multiplier.
Period Register	PRD (15–0)	A 16-bit memory-mapped register used to reload the timer.
Prefetch Counter	PFC (15–0)	A 16-bit counter used to prefetch program instructions. The PFC contains the address of the instruction currently being prefetched. It is updated when a new prefetch is initiated. The PFC is also used to address program memory when using the block move (BLKP), multiply-accumulate (MAC/MACD), and table read/write (TBLR/TBLW) instructions and to address data memory when using the block move (BLKD) instruction.
Product Register	PR(31–0)	A 32-bit product register used to hold the multiplier product. The PR can also be accessed as the most or least significant words by using the SPH/SPL (store P register high/low) instructions.
Program Bus	P(15–0)	A 16-bit bus used to route instructions (and data for the MAC and MACD instructions).
Program Counter	PC (15–0)	A 16-bit program counter used to address program memory. The PC always contains the address of the next instruction to be executed. The PC contents are updated following each instruction decode operation.
Program Memory Address Bus	PAB(15–0)	A 16-bit bus that carries the program memory address.
Queue Instruction Register	QIR(15–0)	A 16-bit register used to store prefetched instructions.
Random Access Memory (data or program)	RAM (B0)	A RAM block with 256 × 16 locations configured as either data or program memory. (512 × 16 for TMS320C26)
Random Access Memory (data only)	RAM (B1)	A data RAM block, organized as 256 × 16 locations. (512 × 16 can be configured as program or data for TMS320C26)
Random Access Memory (data only)	RAM (B2)	A data RAM block, organized as 32 × 16 locations.
Random Access Memory (data or program)	RAM (B3) (TMS320C26 only)	A RAM block with 512 × 16 locations configured as either data or program memory (TMS320C26 only).
Read Only Memory	ROM	A ROM block, 4096 × 16 (256 × 16 for TMS320C26; 8192 × 16 for TMS320C28).
Repeat Counter	RPTC (7–0)	An 8-bit counter to control the repeated execution of a single instruction.
Serial Port Data Receive Register	DRR(15–0)	A 16-bit memory-mapped serial port data receive register. Only the eight LSBs are used in the byte mode.
Serial Port Data Transmit Register	DXR(15–0)	A 16-bit memory-mapped serial port data transmit register. Only the eight LSBs are used in the byte mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit	Symbol	Function
Serial Port Receive Shift Register	RSR(15–0)	A 16-bit register used to shift in serial port data from the RX pin. RSR contents are sent to the DRR after a serial transfer is completed. RSR is not directly accessible through software.
Serial Port Transmit Shift Register	XSR(15–0)	A 16-bit register used to shift out serial port data onto the DX pin. XSR contents are loaded from DXR at the beginning of a serial port transmit operation. XSR is not directly accessible through software.
Shifters	—	Shifters are located at the ALU input, the accumulator output, and the product register output. Also, an in-place shifter is located within the accumulator.
Stack	Stack(15–0)	A 4 × 16 or 8 × 16 hardware stack used to store the PC during interrupts or calls. The ACCL and data memory values may also be pushed onto and popped from the stack.
Status Registers Temporary Register	ST0,ST1 (15–0)	Two 16-bit status registers that contain status and control bits. A 16-bit register that holds either an operand for the multiplier or a shift code for the scaling shifter.
Temporary Register	TR(15–0)	A 16-bit register that holds either an operand for the multiplier or a shift code for the scaling shifter.
Timer	TIM (15–0)	A 16-bit memory-mapped timer (counter) for timing control.

2.6 การจัดหน่วยความจำ

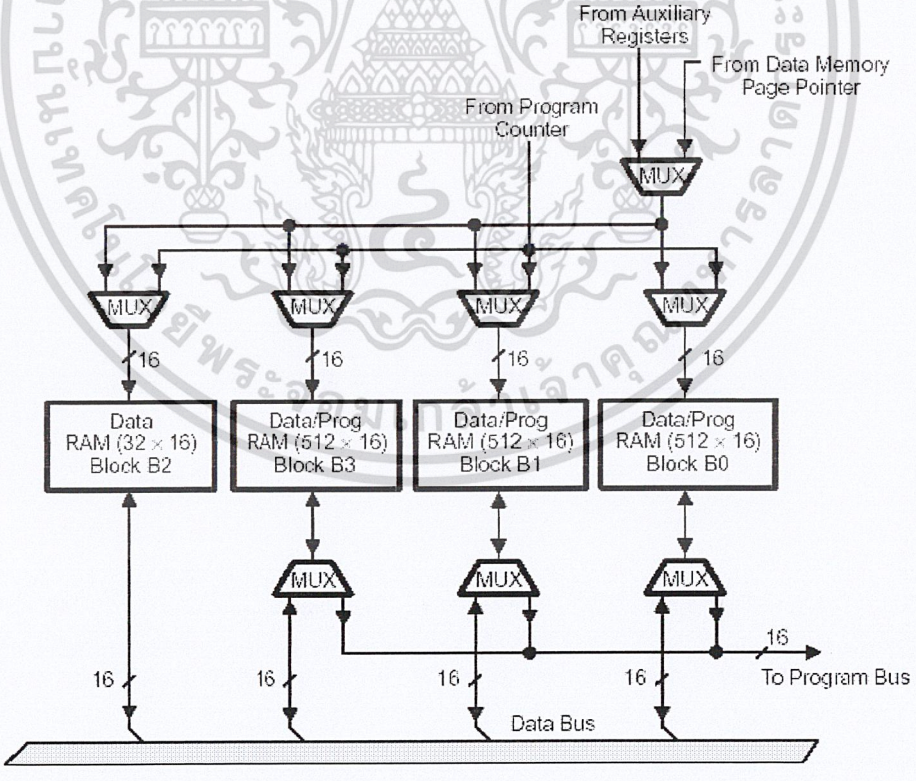
ลักษณะภายในของ TMS320C26 ได้กำหนดให้มีแรมข้อมูล (Data RAM) ภายในตัวชิพ (On chip) ซึ่งมีขนาด 544 16 บิตเวิร์ด ซึ่งในส่วนของ 288 เวิร์ดแรกนั้นจะถูกใช้เป็นหน่วยความจำข้อมูล หรือหน่วยความจำโปรแกรมก็ได้ นอกจากนี้ยังมี มาสเคเบิลโปรแกรมรอม (Mask able Program ROM) ขนาด 4 กิโลเวิร์ดอีกด้วยในส่วนต่อไปนี้จะเป็นการอธิบายถึงการจัดการเกี่ยวกับข้อมูลภายในชิพหน่วยความจำข้อมูลหน่วยความจำโปรแกรมการจัดผังหน่วยความจำ การจัดรีจิสเตอร์ของผังหน่วยความจำอ๊อกซิลารี่รีจิสเตอร์ โหมดในการอ้างแอดเดรสของหน่วยความจำและการเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 หน่วยความจำข้อมูล

จากที่ทราบมาแล้วว่าแรมข้อมูลภายในตัวชิพนั้นจะมีขนาดทั้งหมด 544 เวิร์ด จากรูป 2-3 จะเห็นว่าสามารถแบ่งออกได้เป็น 3 บล็อก คือ B0, B1, B2 ในส่วนของบล็อก B0 ซึ่งมีขนาด 256 เวิร์ด จะถูกใช้งานร่วมกันระหว่างเป็นหน่วยความจำข้อมูล หรือหน่วยความจำโปรแกรมซึ่งก็แล้วแต่ความต้องการ(โดยการใช้คำสั่ง CNFP) และในส่วนของบล็อก B1 และ B2 ซึ่งมีขนาด 258 เวิร์ด จะถูกใช้งานเป็นหน่วยความจำข้อมูลเท่านั้น นอกจากนี้ยังสามารถต่อเพิ่มหน่วยความจำภายนอกได้อีก TMS320C26ยอมให้มีการต่อเพิ่มหน่วยความจำภายนอกได้อีกถึง 64 กิโลเวิร์ด (หน่วยความจำข้อมูลภายในชิพและพื้นที่ภายในบางส่วนซึ่งถูกสงวนไว้รวมแล้วจะกินเนื้อที่ประมาณ 1 กิโลเวิร์ด ซึ่งจะดูได้จากรูปที่ 2-4) สำหรับสัญญาณ READY นั้นจะใช้สำหรับในการเชื่อมต่อกับอุปกรณ์ที่มีความเร็วต่ำและหน่วยความจำน้อย เช่น DRAMs

ตัวอย่างของคำสั่งต่างๆ ที่ใช้งาน ได้แก่ คำสั่ง TBLW เป็นการส่งข้อมูลจากหน่วยความจำภายนอกเข้ามายังหน่วยความจำข้อมูลและคำสั่ง IN เป็นการอ่านข้อมูลจากพอร์ตภายนอกเข้ามายังหน่วยความจำข้อมูลภายใน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-4 หน่วยความจำข้อมูลภายในชิพของ TMS320C26

2.6.2 หน่วยความจำโปรแกรม

ในส่วนของ TMS320C26 จะมีหน่วยความจำรอมภายในตัวชิพขนาด 4 กิโลเวิร์ด ซึ่งจะเรียกว่า " มาสโปรแกรม (Mask-Program) " โดยผู้ใช้สามารถจะสั่งให้โรงงานผู้ผลิตเป็นผู้ผลิตโปรแกรมในส่วนนี้ได้ สำหรับรอมภายในตัวชิพจะยอมให้มี การexecute โปรแกรมจากหน่วยความจำภายนอกหน่วยความจำภายนอกได้ การใช้งานหน่วยความจำนี้จะยอมให้บัสข้อมูลภายนอกเป็นอิสระต่อการเข้าถึงหน่วยความจำข้อมูลภายนอก ในการจัดผังหน่วยความจำของหน่วยความจำโปรแกรม ว่าต้องการจะใช้หน่วยความจำภายใน หรือต่อจากภายนอก จะเป็นส่วนที่ผู้ใช้จะเป็นผู้กำหนดโดยหมายถึงขา MP/MC บน TMS320C26 คือ ถ้ากำหนดให้ขา MP/MC เป็นLOW จะหมายถึงเป็นการใช้หน่วยความจำภายในชิพ สำหรับขา XF (External Flag) จะใช้ในการสลับขา MP / MC เพื่อให้รอมภายในชิพนั้นอีนาเบิล (Enable) หรือดิสเอเบิล (Disable)

2.6.3 การจัดผังหน่วยความจำ

การจัดผังหน่วยความจำของ TMS320C26 นั้น จะมีการแบ่งส่วนของแอดเดรส ออกเป็น 3 ส่วนด้วยกัน คือ หน่วยความจำโปรแกรม หน่วยความจำข้อมูล และ I / O ซึ่งแสดง ดังรูปที่ 2-5 ถ้ามองจากภายนอก หมายถึง สัญญาณ $\overline{PS}$,$\overline{DS}$ และ $\overline{IS}$ (เป็นสัญญาณที่ใช้ในการเลือกโปรแกรม ข้อมูล หรือI/O) ซึ่งสัญญาณ $\overline{PS}$, $\overline{DS}$, $\overline{IS}$ และ $\overline{STRB}$ จะแอคทีฟ เมื่อมีแอดเดรสจากหน่วยความจำภายนอกเข้ามา

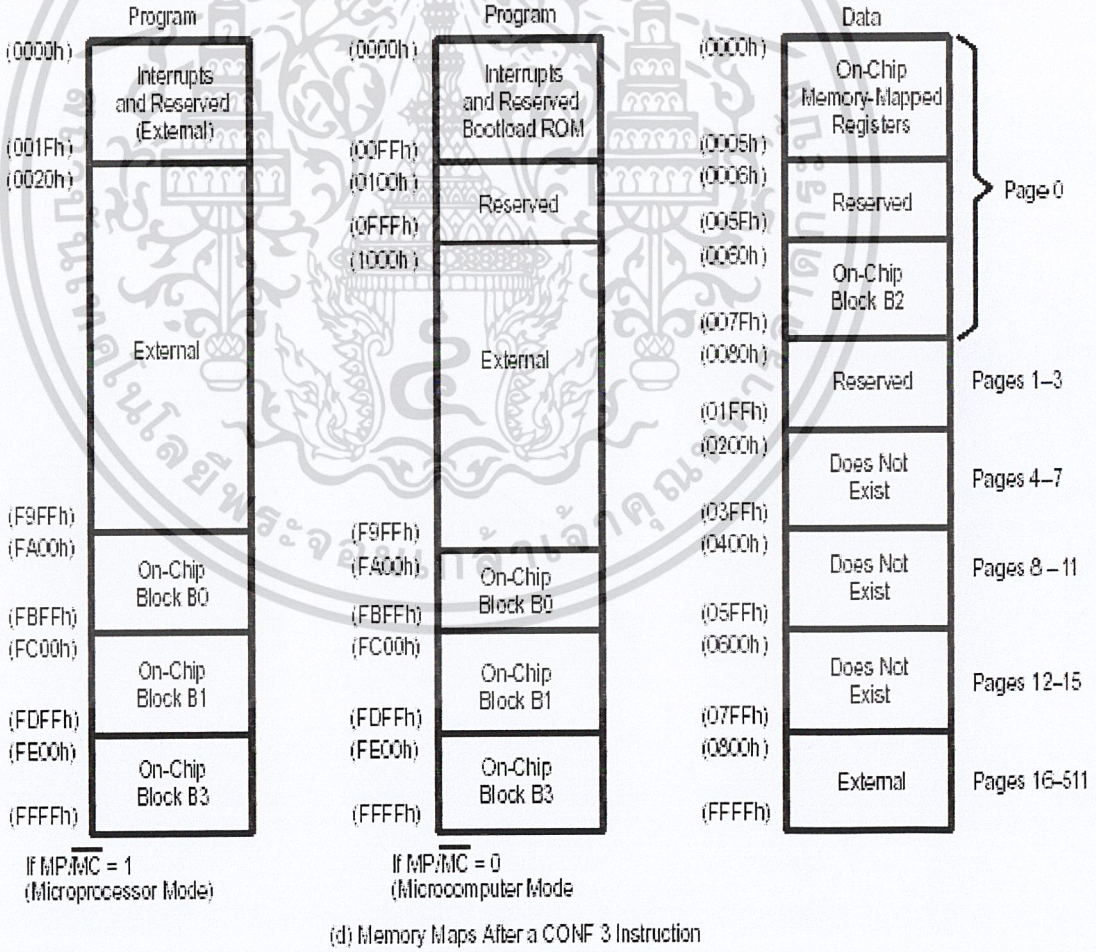
ในส่วนของแรมในตัวชิพนั้นจะประกอบด้วย 3 ส่วน คือ บล็อก B0 ,B1 ,B2 ซึ่งรวมกันแล้วจะมีขนาด 544 เวิร์ด ในส่วนที่ใช้เป็นโปรแกรม / ข้อมูล (แรมบล็อก B0 ขนาด 256 เวิร์ด) จะอยู่ในหน้าที่ 4 และ 5 ของผังหน่วยความจำข้อมูล (เมื่อใช้งานเป็นแรมข้อมูล) และจะอยู่ในแอดเดรส > FF00 ถึง > FFFF (เมื่อใช้งานเป็นโปรแกรมแรม) ส่วนบล็อก B1 จะอยู่ในหน้าที่ 6 และ 7 ของผังหน่วยความจำข้อมูลในขณะที่บล็อก B2 จะอยู่ในส่วนของ 32 เวิร์ดด้านบนของหน้า 0 และส่วนที่เหลืออยู่ในหน้า 0 นั้นจะประกอบด้วย การจัดรีจิสเตอร์ของผังหน่วยความจำทั้ง 6 ตัว และพื้นที่ภายในซึ่งจองหรือสงวนไว้ ในพื้นที่ที่จองหรือ สงวนไว้นี้จะไม่สามารถนำมาใช้เก็บข้อมูลหรือใช้งานใดๆ และตั้งแต่แอดแดรสที่ 1024 -65535 ซึ่งอยู่ในหน้าที่ 8-511 นั้นจะใช้สำหรับในการต่อกับหน่วยความจำภายนอก

ส่วนต่อไปจะกล่าวถึงการจัดผังหน่วยความจำโปรแกรม จากที่ทราบมาแล้วว่าTMS320C26 มีโหมดการทำงานอยู่ 2โหมด ด้วยกันคือโหมดไมโครคอมพิวเตอร์ และโหมดไมโครโปรเซสเซอร์ ซึ่งทั้ง 2 โหมดนี้จะมีการจัดผังของหน่วยความจำโปรแกรมไม่เหมือนกันดังจะกล่าวต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้ขา MP/MC = 1 (กำหนดให้อยู่ในโหมดไมโครคอมพิวเตอร์) ในโหมดนี้จะไม่มีรอบภายในตัวชิพ แต่จะเป็นการต่อหน่วยความจำจากภายนอกทั้งหมด 64 เวิร์ด

ถ้ากำหนดให้ขา MP/MC = 0 (กำหนดให้อยู่ในโหมดไมโครคอมพิวเตอร์) ในโหมดนี้ TMS320C26 จะกำหนดให้มีรอบภายในตัวชิพขนาด 4 กิโลเวิร์ด และ ยังสามารถทำการต่อขยายหน่วยความจำภายนอกได้อีก คือตั้งแต่แอดเดรสที่ 4096 - 65535 นอกจากนี้ยังมีคำสั่งซึ่งใช้ในการจัดผังหน่วยความจำใหม่ คำสั่งนั้นคือ CNFP จากรูปที่ 2-5 เป็นการจัดผังหน่วยโปรแกรมใหม่ภายหลังจากการใช้คำสั่ง CNFP ซึ่งจากรูปจะเห็นว่า ในส่วนของหน่วยความจำโปรแกรมตำแหน่งที่ 65280 - 65535 จะถูกใช้เป็นโปรแกรมแรม ซึ่งก็คือ บล็อก B0 ของหน่วยความจำข้อมูลนั่นเอง โดยโปรแกรมแรมในส่วนนี้ สามารถใช้ในการเขียนโปรแกรมลงไป หรืออ่านออกมาภายนอกได้ และมีขนาด 256 เวิร์ด



(d) Memory Maps After a CONF 3 Instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-5 การจัดผังหน่วยความจำหลังจากใช้คำสั่ง CNFP ของ TMS320C26

2.6.4 การจัดรีจิสเตอร์ของผังหน่วยความจำ

จากตารางที่ 2-2 แสดงถึงแอดเดรสและหน้าที่ของรีจิสเตอร์ภายในทั้ง 6 ตัวของผังหน่วยควบจำข้อมูล และแสดงบล็อกไดอะแกรมของรีจิสเตอร์ต่างๆ ดังในรูปที่ 2-3

การจัดผังรีจิสเตอร์หน่วยความจำนี้ อาจมีลักษณะการจัดเหมือนกับในหน่วยความจำข้อมูลอื่นๆ แต่จะมีข้อยกเว้น คือ ไม่สามารถจะใช้คำสั่งการเคลื่อนย้ายข้อมูลเป็นบล็อก (BLKD) จากรีจิสเตอร์หน่วยความจำเหล่านี้ได้

Memory – Mapped Registers

Register name	Address location	Definition
DRR (15-0)	0	Serial port data receive register
DXR (15-0)	1	Serial port data transmit register
TIM (15-0)	2	Timer register
PRD (15-0)	3	Period register
IMR (5-0)	4	Interrupt mask register
GREG (7-0)	5	Global memory allocation register

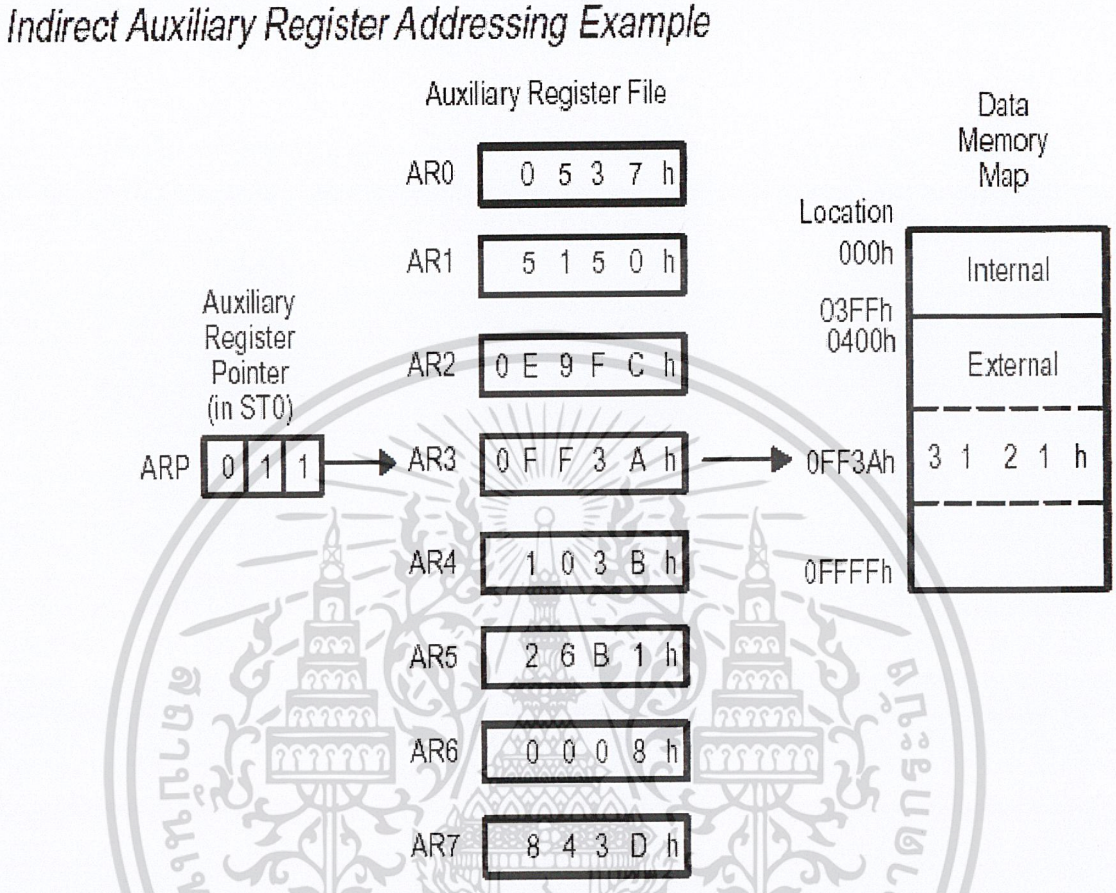
ตารางที่ 2-2 การจัดรีจิสเตอร์ของผังหน่วยความจำ

2.6.5 อ๊อกซิลารี่รีจิสเตอร์ (Auxiliary Register)

สำหรับอ๊อกซิลารี่รีจิสเตอร์ภายในของ TMS320C26 นั้นได้กำหนดให้มีจำนวนถึง 8 ตัว คือ AR0 - AR7 โดยอ๊อกซิลารี่รีจิสเตอร์ขนาด 16 บิต ซึ่งใช้ในการอ้างแอดเดรสแบบทางอ้อมของหน่วยความจำข้อมูล หรือใช้สำหรับเป็นที่เก็บข้อมูลชั่วคราว รีจิสเตอร์นี้จะถูกชี้ค่าโดยค่าที่อยู่ใน ARP (เป็นรีจิสเตอร์ขนาด 3 บิต) ซึ่งมีค่าระหว่าง 0 – 7 จะหมายถึง AR0 - AR7 และอ๊อกซิลารี่รีจิสเตอร์นี้อาจจะทำการโหลดค่ามาจากหน่วยความจำข้อมูล หรืออาจจะทำการโหลดค่ามาจากโอเปอร์แรนด์ของคำสั่งแบบทันทีก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2-6 ตัวอย่างของอ๊อกซิลารี่รีจิสเตอร์ที่ใช้ในการอ้างแอดเดรสแบบทางอ้อม

อ๊อกซิลารี่รีจิสเตอร์ (AR0-AR7) นี้จะต่ออยู่กับ Auxiliary Register Arithmetic Unit (ARAU) แสดงดังรูปที่ 2-7 จากรูปจะเห็นว่า AR0 จะถูกต่อเป็นอินพุทหนึ่งของ ARAU ส่วนอินพุตอื่นจะมาจาก AR ที่ใช้อยู่ในปัจจุบันโดยที่ ARAU จะมีการปฏิบัติการตามฟังก์ชั่นดังต่อไปนี้

AR(ARP) + AR0 ⟶ AR (ARP) เป็นการชี้ตำแหน่งปัจจุบันของ AR โดยการบวกด้วยเลขจำนวนเต็มขนาด16บิต ซึ่งเก็บอยู่ใน AR0

AR(ARP) - AR0 ⟶ AR (ARP) เป็นการชี้ตำแหน่งปัจจุบันของ AR โดยการลบด้วยเลขจำนวนเต็มขนาด16บิต ซึ่งเก็บอยู่ใน AR0

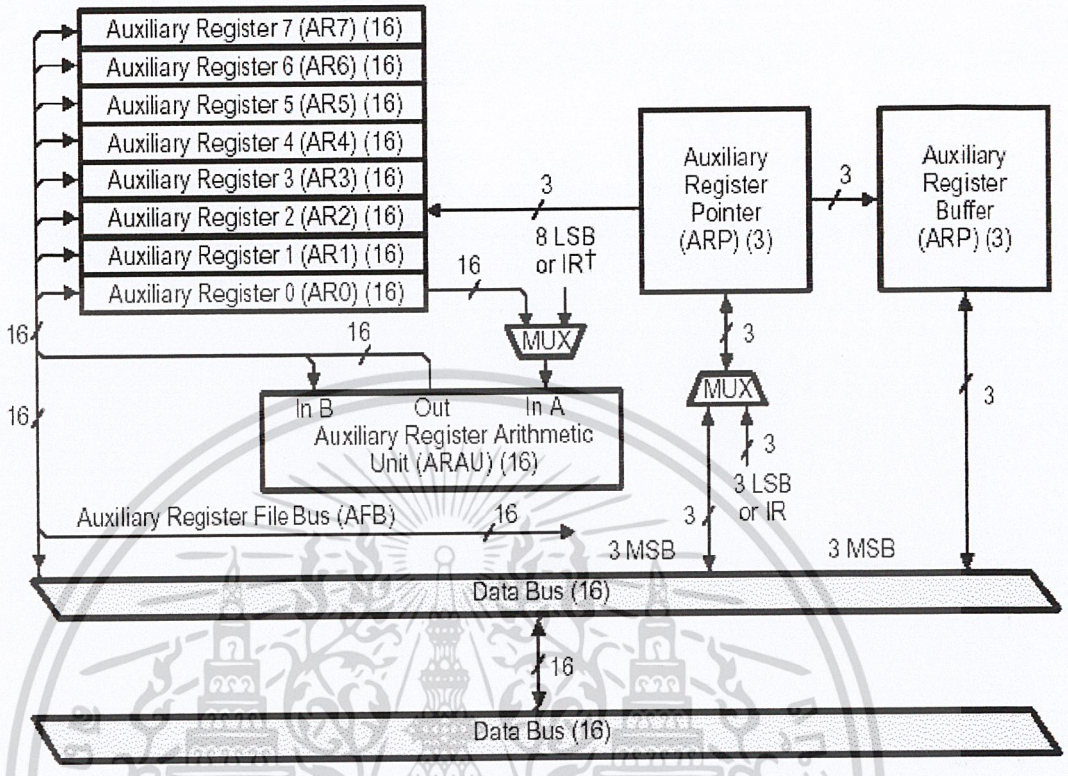
AR(ARP) + 1 ⟶ AR(ARP) เพิ่มค่าของ AR ขึ้นอีก

AR(ARP) – 1 ⟶ AR(ARP) ลดค่าของ AR ลง 1

AR(ARP) ⟶ AR(ARP) ไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50344



รูปที่ 2-7 Auxiliary Register File

ในส่วนของฟังก์ชั่นที่นอกเหนือไปจากนี้ ARAU ของ TMS320C26 จะมีการปฏิบัติตามฟังก์ชั่นดังต่อไปนี้

AR(ARP) + IR (7-0) ⟶ AR(ARP) บวกตำแหน่งของ AR ปัจจุบันด้วยค่าขนาด 8 บิต

AR(ARP) – IR (7-0) ⟶ AR(ARP) ลบตำแหน่งของ AR ปัจจุบันด้วยค่าขนาด 8 บิต

AR(ARP) +rcAR0 ⟶ AR(ARP) กลับบิตตัวชี้แล้วทำการบวก AR0 ด้วย reverse-carry (rc)

AR(ARP) -rcAR0 ⟶ AR(ARP) กลับบิตตัวชี้แล้วทำการลบ AR0 ด้วย reverse-carry (rc)

ส่วน Auxiliary Register Point Buffer (ARB) ดังแสดงในรูปที่ 3-6 ซึ่งเป็นรีจิสเตอร์ขนาด 3 บิตนั้นจะใช้สำหรับเก็บค่าของ ARP ในขณะที่เกิดการอินเตอร์รัพท์ หรือมีการเรียกใช้โปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

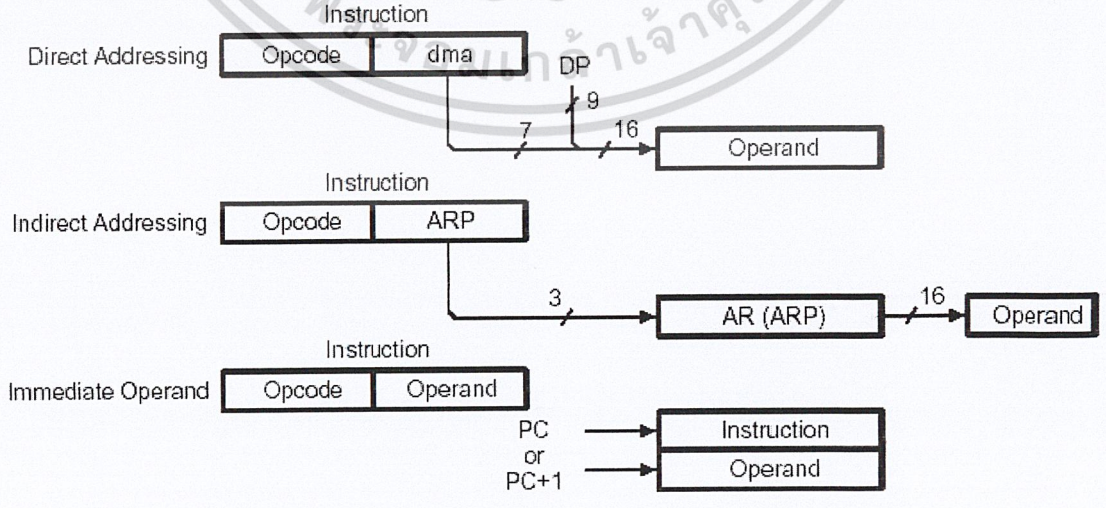
2.6.6 การอ้างแอดเดรสของหน่วยความจำ

TMS320C26 สามารถอ้างแอดเดรสของหน่วยความจำโปรแกรม ได้ทั้งหมด 64 เวิร์ด โดยส่วนหนึ่งจะถูกใช้เป็นรอมภายในตัวชิพ เมื่ออยู่ในโหมดไมโครคอมพิวเตอร์ และนอกจากนี้ยังสามารถอ้างแอดเดรสของหน่วยความจำข้อมูลได้อีก 64 กิโลเวิร์ด โดยแอดเดรสในส่วนนี้จะถูกใช้ในแรมข้อมูลภายในตัวชิพส่วนหนึ่ง

ในการใช้ 16-bit Data Address Bus (DAB) อ้างแอดเดรสของหน่วยความจำข้อมูลนั้นสามารถทำได้โดยใช้วิธีหนึ่งวิธีใด ดังต่อไปนี้

1. โดยการใช้ direct address bus (DRB) ซึ่งจะใช้ใน โหมดการอ้างแอดเดรสแบบโดยตรง

2. โดยการใช้ Auxiliary Register file bus (AFB) ซึ่งจะใช้ในโหมดการอ้างแอดเดรสแบบทางอ้อมสำหรับในโหมดการอ้างแอดเดรสแบบทันทีนั้น โอเปอร์แรนด์ ที่อยู่ใน PC จะถูกใช้เป็นแอดเดรสในโหมดการอ้างแอดเดรสแบบโดยตรงนั้นจะใช้ 9 บิตของ Data Memory Page Pointer (DP) เป็นตัวชี้ในหน้าหนึ่งหน้าใดจากทั้งหมด 512 หน้า ซึ่งในแต่ละหน้าจะมีจำนวน 128 Words สำหรับ Data Memory address (dma) จะใช้ 7 บิตทางด้านต่ำของคำสั่งเป็นตัวชี้คำในแต่ละหน้านั้น ดังนั้นแอดเดรสที่ปรากฏบน DRB จึงประกอบด้วย DP จำนวน 9 บิตกับ dma จำนวน 7 บิต ในโหมดการอ้างแอดเดรสแบบทางอ้อมนั้น การเลือกแอดเดรสของ AR (ARP) นั้น หน่วยความจำข้อมูลจะทำการผ่านค่าไปยัง AFB ขณะที่ทำการเลือก AR แล้วข้อมูล และแอดเดรสของหน่วยความจำข้อมูลจะถูกเคลื่อนย้ายโดย CALU และค่าที่อยู่ใน AR จะถูกเคลื่อนย้ายผ่าน ARAU



รูปที่2-8 ส่วนประกอบในคำสั่งซึ่งใช้ในการอ้างแอดเดรสแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.7 การเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำ

TMS320C26 มีคำสั่งซึ่งใช้ในการเคลื่อนย้ายข้อมูลเป็นบล็อกเพื่อใช้สำหรับจัดการกับข้อมูลหรือโปรแกรม ซึ่งฟังก์ชั่นการเคลื่อนย้ายข้อมูลนี้ จะเป็นประโยชน์อย่างมาก สำหรับแรมภายในชิพ

สำหรับคำสั่ง BLKD (block move from Data Memory to Data Memory) จะใช้ในการเคลื่อนย้ายข้อมูลภายในหน่วยความจำข้อมูล ส่วนคำสั่ง BLKP (block move from Program Memory to Data Memory) จะใช้สำหรับในการเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำโปรแกรมไปยังหน่วยความจำข้อมูล เมื่อใช้งานร่วมกับคำสั่ง RPT/RPTX คำสั่ง BLKD/BLKP สามารถจะทำการเคลื่อนย้ายข้อมูลจากหน่วยความจำภายในหรือภายนอกชิพได้

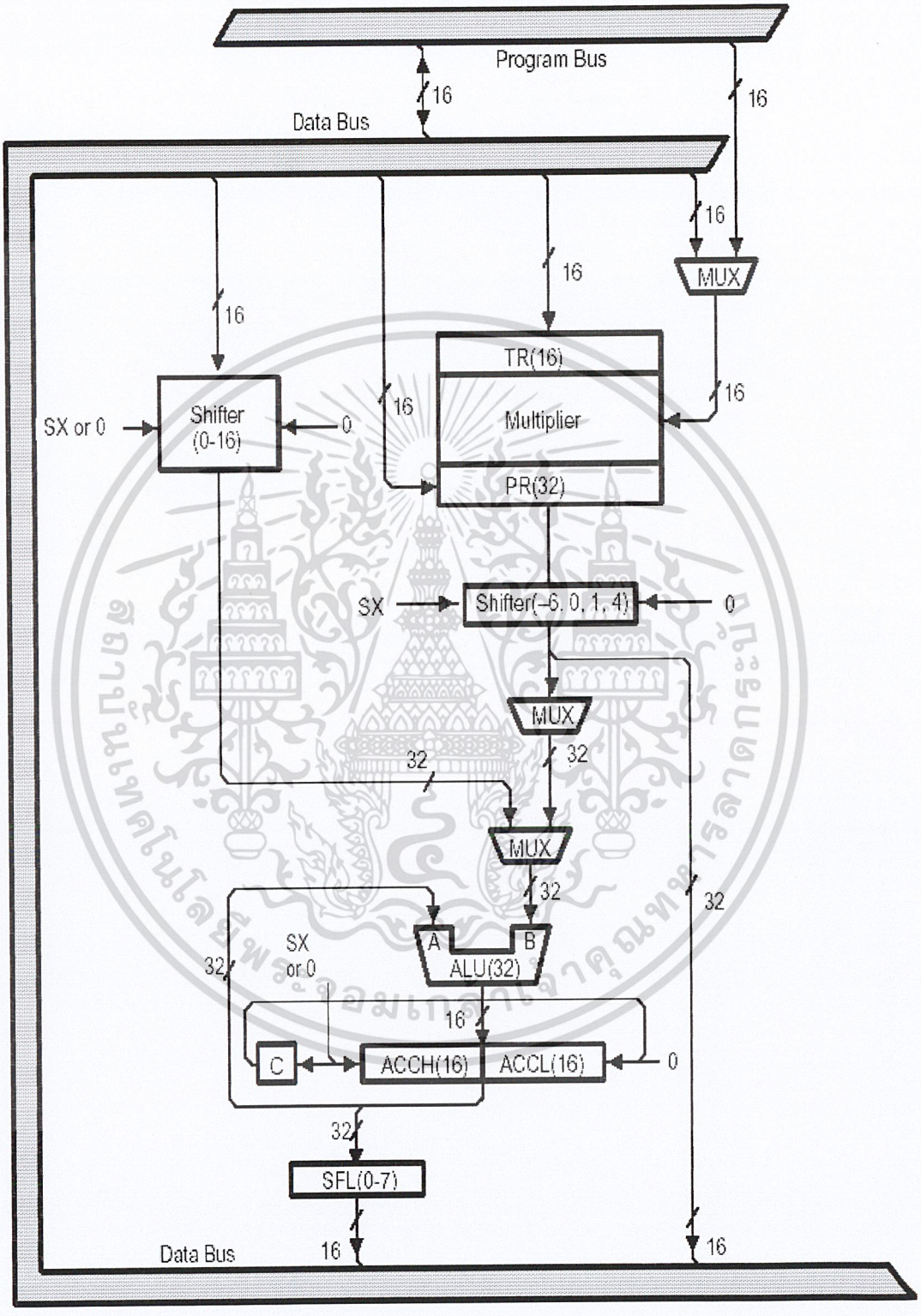
2.6.8 หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิก

หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิกภายใน TMS320C26 นั้น จะประกอบไปด้วยสเกลลิ่งชิฟเตอร์ขนาด16 บิต ตัวคูณแบบขนานขนาด 16x16 บิตALU ขนาด 32 บิต ACC ขนาด 32 บิต และตัวเลื่อนบิตที่ต่ออยู่กับ ACC และมัลติพลายเออร์(Multiplier) จากรูปที่ 2-9 จะแสดงถึงบล็อกไดอะแกรมของส่วนประกอบภายในต่างๆ ของ CALU

ลำดับขั้นต่างๆ ที่เกิดขึ้นภายในของ CALU หลังจากที่มีการใช้คำสั่งที่เกี่ยวกับ ALU

1. ข้อมูลต่างๆ ที่ต้องการประมวลผลจะถูกเฟ็ทซ์จากแรมมาบนบัสข้อมูล
2. ข้อมูลต่างๆ จะผ่านมายังสเกลลิ่ง ชิฟเตอร์และ ALU เพื่อมาทำการประมวลผล
3. ผลลัพธ์ที่ได้จากการประมวลผลจะถูกนำไปเก็บไว้ในแอคคิวมูเลเตอร์ โดยอินพุตส่วนหนึ่งของ ALU จะได้มาจากแอคคิวมูลเลเตอร์ ส่วนอินพุทที่เหลืออาจจะโอนย้ายมาจากรีจิสเตอร์ที่ใช้เก็บผลคูณ (PR) ของมัลติพลายเออร์ หรือมาจากสเกลลิ่งชิฟเตอร์ซึ่งโหลดค่าจากหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 หน่วยศูนย์กลางในการคำนวณทางคณิตศาสตร์และลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.9 สเกลลิ่งชิฟเตอร์

ในการเลื่อนบิทของ TMS320C26 นั้น จัดให้มีสเกลลิ่งชิฟเตอร์ มีอินพุทขนาด 16 บิท ต่ออยู่กับบัสข้อมูล และเอาท์พุทขนาด 32 บิท ซึ่งต่ออยู่กับ ALU โดยสามารถทำการเลื่อนบิทของข้อมูลอินพุทไปทางซ้ายได้ 0-16 บิท โดยบิทซึ่งมีนัยสำคัญต่ำ (LSBs) จะถูกเติมด้วย 0 และบิทที่มีนัย สำคัญสูง (MSBs) อาจจะถูกเติมด้วย 0 หรือใช้เป็นส่วนขยายเครื่องหมาย โดยจะขึ้นอยู่กับสถานะที่บิท SXM (Sign Extension Mode) ของรีจิสเตอร์สถานะ ST1

นอกจากนี้ภายใน TMS320C26 ประกอบไปด้วยตัวเลื่อนบิตอื่นๆ อีกซึ่งจะมีการใช้งานในด้านต่างๆ เช่น Bit Extraction การคำนวณแบบที่ต้องการความแม่นยำสูง และป้องกันการเกิดค่าเกิน (Overflow) โดยตัวเลื่อนบิตเหล่านี้จะต่ออยู่ที่เอาท์พุทของมัลติพลายเออร์ และที่แอคคิวมูเลเตอร์

2.6.10 แอคคิวมูเลเตอร์ และ ALU

ALU จะเป็นหน่วยทำงานทางคณิตศาสตร์และลอจิกทั่วไปที่มีขนาด 32 บิต ซึ่งสามารถทำการบวกลบและทำงานทางลอจิกได้โดยแอคคิวมูเลเตอร์จะเป็นตัวทำงานทั้งก่อนและหลังเสมอ การทำงานของลอจิก จะแสดงดังตารางที่ 2-3 ค่าในหน่วยความจำข้อมูลจะใช้กระทำกับบิตทางด้านต่ำของแอคคิวมูเลเตอร์ (บิต = 15 ถึง บิต = 0) ส่วนค่า Zero จะใช้กระทำกับค่าทางด้านบิตสูงของแอคคิวมูเลเตอร์

FUCTION	ACCUMULATOR RESULT	
	ACC BITS 31 THROUGH 16	ACC BITS 15 THROUGH 0
XOR	(zero) ÷ (ACC bits 31-16)	(data memory value)÷(ACC bits15-0)
AND	(zero) . (ACC bits 31-16)	(data memory value) . (ACC bits15-0)
OR	(zero) + (ACC bits 31-16)	(data memory value)+(ACC bits15-0)

ตารางที่ 2-3 ผลลัพธ์ที่เกิดในแอคคิวมูเลเตอร์

แอคคิวมูเลเตอร์จะเป็นตัวเก็บผลลัพธ์ของ ALU หรืออาจจะเก็บตัวตั้งของ ALU แอคคิวมูเลเตอร์นี้จะมีขนาด 32 บิต และถูกแบ่งออกเป็น 2 ส่วน คือ บิตบน (บิต 31-16) และบิตล่าง (บิต 15-0) การเก็บค่าในบิตบนหรือบิตล่างทำได้โดยคำสั่ง SACH และ SACL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะของการเกิดค่าเกินที่แอคคิวมูเลเตอร์นั้น สามารถดูได้จากแฟลกรีจิสเตอร์ที่แสดงค่าเกิน (OV) ซึ่งรีจิสเตอร์นี้จะเซ็ทเมื่อเกิดค่าเกินขึ้นในแอคคิวมูเลเตอร์ ซึ่งประโยชน์ของ OV จะใช้ในการทำคำสั่งที่มีการตัดสินใจของ DSP ซึ่งคำสั่งต่างๆเหล่านี้แสดงดัง ตารางที่ 2-4

Instruction	ACCUMULATOR COMDITION TESTED
BLZ	<0
BLEZ	<0
BGZ	>0
BGEZ	>0
BNZ	<>0
BZ	=0

ตารางที่ 2-4 การตรวจสอบสถานะของแอคคิวมูเลเตอร์

2.6.11 ตัวคูณ , รีจิสเตอร์ T และ P

สำหรับการคูณเลขขนาด 16 x 16 บิต นั้น สามารถจะทำการคูณได้ภายในไซเกิลเดียว (ใช้เวลา 100 นาโนวินาที)ในการคูณนั้นจะประกอบไปด้วย 3 ส่วนใหญ่ๆ ด้วยกัน คือ T, P รีจิสเตอร์และมัลติพลายเออร์แอร์แรย์ โดยรีจิสเตอร์ T จะใช้สำหรับเก็บตัวตั้งขนาด 16 บิต ส่วนรีจิสเตอร์ P จะใช้เก็บผลคูณขนาด 32 บิต

2.6.12 โปรแกรมเคาน์เตอร์และสแตค

โปรแกรมเคาน์เตอร์ (PC) และสแตค นั้นจะมีประโยชน์ในการทำงานดังนี้ การกระโดดข้ามตำแหน่งในการทำงาน การเรียนโปรแกรมย่อย การอินเตอร์รัพท์และการทำคำสั่ง TBLR หรือ TBLW (อ่านและเขียนตาราง)

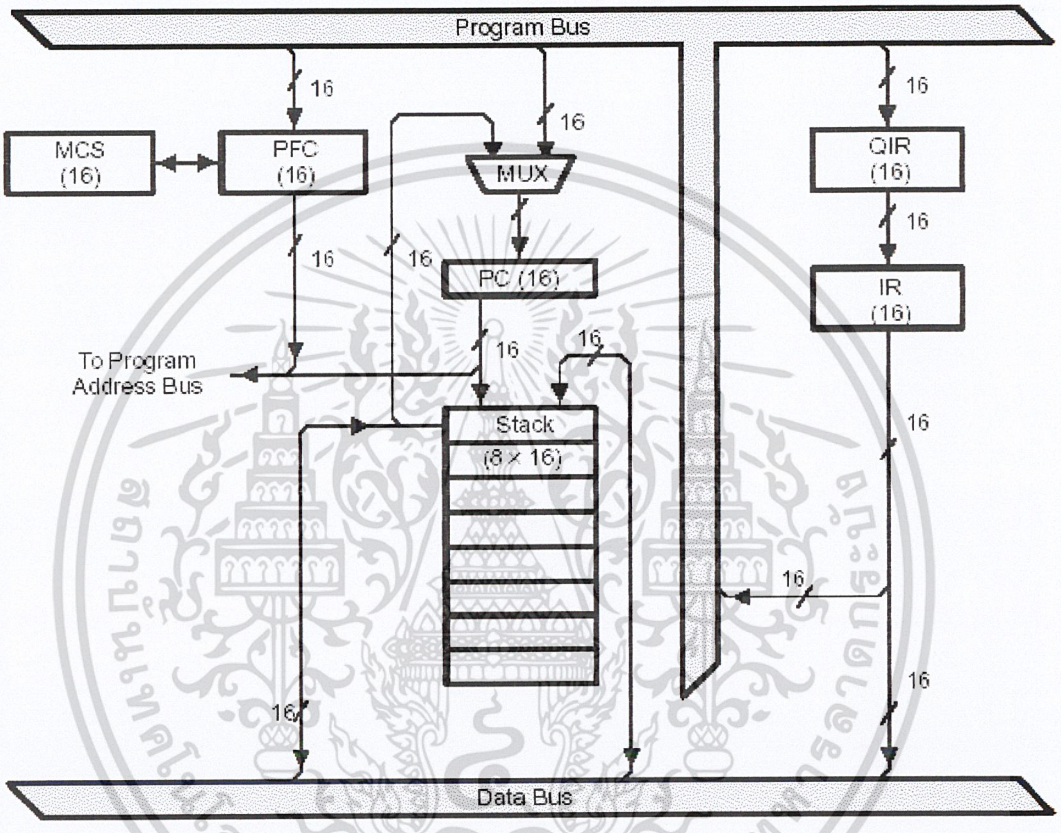
โปรแกรมเคาน์เตอร์ (PC) เป็นรีจิสเตอร์ขนาด 16 บิต ที่จะเก็บตำแหน่งของหน่วยความจำโปรแกรมของคำสั่งต่อไปที่จะถูกกระทำซึ่งค่าใน PC นี้จะเป็น “0” ทุกครั้งที่มีการรีเซทที่ตัวDSPชิพ

นอกจากนี้ค่าของ PC สามารถเปลี่ยนแปลงค่าได้โดย การใช้คำสั่งกระโดด (Branch) ซึ่งจะทำให้ค่าของ PC เพิ่มหรือลดตามค่าที่กำหนดในโปรแกรม

สแตคมีขนาด16บิตและแบ่งออกเป็น 8 ระดับ เมื่อใช้คำสั่ง PUSH จะมีผลให้ DSP เก็บค่า 16 บิต ด้านต่ำของแอคคิวมูเลเตอร์ลงชั้นบนสุดของสแตค (TOS) และถ้าใช้คำสั่ง POP จะเป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่า 16 บิตของ TOS มาไว้ในแอคคิวมูเลเตอร์ ถ้าหากว่าใช้คำสั่ง SACL แล้วจะเป็นการนำค่าจาก TOS ไปเก็บไว้ในหน่วยความจำข้อมูล ซึ่งเป็นประโยชน์ในการเพิ่มความจุของสแตค



รูปที่ 2-10 โครงสร้างของสแตค และ โปรแกรมเคาน์เตอร์

2.6.13 ปฏิบัติการไปป์ไลน์ (Pipeline Operation)

จากรูปที่ 2-11 แสดงให้เห็นถึงการซ้อนทับของการทำงานในการเฟ็ทซ์ และexecute ที่ของขาล่างของ CLKOUT , ค่าของ PC จะถูกโหลด (PC2)เพื่อที่จะตามคำสั่งเมื่อกระบวนการถอดรหัสคำสั่งแรก (Execute 1) เริ่มขึ้น เช่นเดียวกัน คำสั่งที่ 3 (Fetch 2) เมื่อกระบวนการทำงานตาม คำสั่งที่ 1 (Execute 1) เริ่มขึ้น เช่นเดียวกัน คำสั่งที่ 3 (Fetch 3) จะเริ่มเมื่อมีการกระทำตามคำสั่งที่ 2 (Execute 2) ได้เริ่มขึ้น ซึ่งจะเรียกลักษณะการทำงานเช่นนี้ว่า “Pipeline-Operation”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีเซ็ท (Reset)

รีเซ็ท (RS) เป็นสัญญาณอินเทอร์รัพท์จากภายนอกโดยเป็นสัญญาณอินเทอร์รัพท์ชนิด " นอนมาสเคเบิล " ซึ่งจะเกิดขึ้นได้ทุกๆ ขณะ และTMS320C26 สามารถที่จะทำการตรวจสอบได้ การนำสัญญาณรีเซ็ทไปประยุกต์ใช้งานตามตัวอย่างนี้มีการใช้งานเมื่อเริ่มเปิดเครื่องใหม่

เมื่อมีสัญญาณเข้ามาที่ขาRS จะมีผลให้โปรแกรมเคาน์เตอร์มีค่าเป็น "0" และมีการเปลี่ยนแปลงเกิดขึ้นที่รีจิสเตอร์และบิตสถานะ สัญญาณรีเซ็ทนี้จะต้องเกิดขึ้นอย่างน้อย 3 สัญญาณนาฬิกา เพื่อเป็นการยืนยันการรีเซ็ทของอุปกรณ์

เมื่อ TMS320C26 ได้รับสัญญาณรีเซ็ทจะมีการปฏิบัติตามกระบวนการ ดังต่อไปนี้

1. ที่บิต CNF ของรีจิสเตอร์สถานะ ST1 จะถูกโหลดด้วย"0" ซึ่งจะมีผลให้แรมทั้งหมดมีลักษณะเป็นหน่วยความจำข้อมูล
2. โปรแกรมเคาน์เตอร์จะถูกเซ็ทให้เป็น 0 แอดเดรสบัส A15-A0 ก็จะมีค่าเป็น "0"ด้วย
3. บัสข้อมูล D15-D0 จะอยู่ในสถานะเป็นไฮอิมพีแอนซ์
4. สัญญาณที่ใช้ในการควบคุมต่างๆ (PS, DS, IS, R/w, STRB และ BR) จะมีสถานะเป็น "1"
5. สัญญาณอินเตอร์รัพท์อื่นๆ จะถูกดิสเอเบิลโดยการเซ็ทบิต INTM ให้เป็น "1" และอินเตอร์รัพท์แฟลกรีจิสเตอร์(IFR) จะถูกเซ็ทให้เป็น"0 "ทั้งหมด
6. ที่บิตสถานะต่างๆ จะมีผลดังนี้

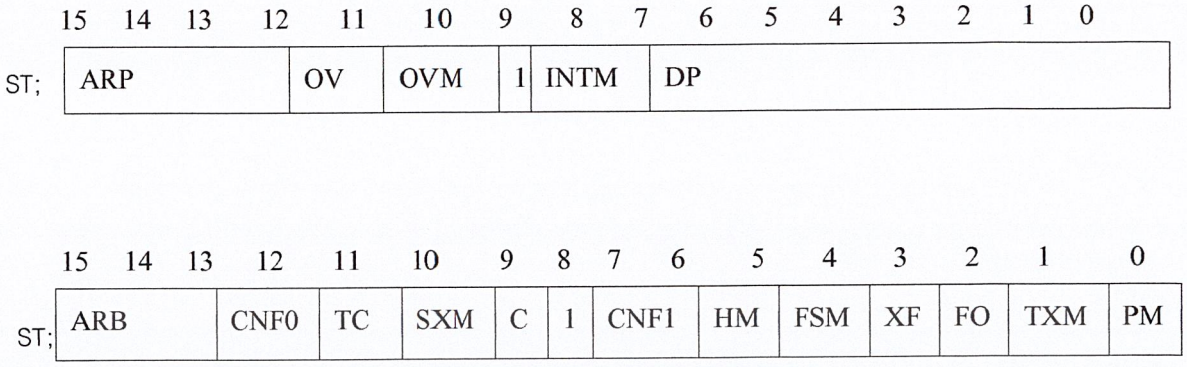
1 ⟶ SXM , 0 ⟶ HM, 0 ⟶ FO , 1 ⟶ C และ 1 ⟶ FSM

7. ข้อมูลต่างๆ ใน RPTC จะถูกเคลียร์ให้เป็น "0"
8. ขา DX (Data Transmit) จะอยู่ในสถานะเป็นไฮอิมพีแอนซ์
9. รีจิสเตอร์ TIM จะถูกเซ็ทให้เป็นค่าสูงสุด (>FFFF) และเมื่อสัญญาณ RS ถูกเปลี่ยนกลับ ไปอยู่ในสถานะ "1" ก็จะมีการเริ่มการทำงานใหม่ โดยเริ่มจากพื้นที่"0"ของโปรแกรม

2.6.14 รีจิสเตอร์สถานะ

ในรีจิสเตอร์สถานะทั้งสองคือ ST0 และ ST1 จะประกอบไปด้วยเงื่อนไขและโหมดต่างๆ ในการทำงานสามารถที่จะเปลี่ยนแปลงได้โดยรีจิสเตอร์สถานะนี้สามารถเก็บค่าต่างๆไว้ในหน่วยความจำข้อมูลและสามารถโหลดค่าออกมาได้ ในการเขียนค่ากับบิตสถานะต่างๆ เหล่านี้สามารถทำได้โดยการใช้คำสั่ง LST/LST1 และ SST/SST1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-11 การจัดรูปแบบของรีจิสเตอร์สถานะ (Status Register)

จากรูปที่ 2-11 แสดงถึงการจัดบิตต่างๆ ภายในรีจิสเตอร์สถานะทั้งสอง และจากรูปที่ 2-11 จะเห็นว่ามีการแยกส่วนต่างๆ ของรีจิสเตอร์ DP, ARP, ARB ออกมาให้เห็นอย่างชัดเจนเพราะว่ารีจิสเตอร์เหล่านี้ ไม่มีคำสั่งที่ใช้ในการแยกเก็บค่าเหล่านี้ลงในแรมในส่วนอื่นๆ ของคำสั่งหรือฟังก์ชั่นที่อาจจะมีผลต่อบิตสถานะต่างๆ เหล่านี้ สามารถจะอธิบายได้ดังนี้

ARB (Auxiliary Register Pointer Buffer) เมื่อใดก็ตามที่มีการโหลดค่าเข้ามายัง ARP ค่าเก่าของ ARP จะถูกคัดลอกมายัง ARB ยกเว้นที่มีการใช้คำสั่ง LST เมื่อ ARB ถูกโหลดผ่านคำสั่ง LST1 ค่าที่เหมือนกันจะถูกคัดลอกไปยัง ARP

ARP (Auxiliary Register Pointer) ซึ่งมีขนาด 3บิต ใช้ในการเลือก AR โดยใช้ในการอ้างแอเดรสแบบทางอ้อม ARP อาจจะมีการเปลี่ยนแปลงโดยการใช้คำสั่งอ้างอิงหน่วยความจำ เมื่อใช้การอ้างแอดเดรสแบบทางอ้อมด้วยคำสั่ง LARP, MAR, และLSP โดย ARP จะถูกโหลดด้วยค่าที่เหมือนกันกับใน ARB เมื่อใช้คำสั่ง LST1 ในการเอ็กซีคิ้วซ์(Execute)

C บิตตัวทดนี้จะถูกเซ็ทเป็น “1” ถ้าผลลัพธ์จากการบวกเกิดมีตัวทด และจะคำสั่งเป็น “0” เมื่อผลลัพธ์ที่เกิดจากการลบมีการขอยืม นอกจากกรณีที่กล่าวมาแล้ว มันจะเกิดการรีเซ็ทหลังจากที่ใช้คำสั่งการบวก หรือรีเซ็ทหลังจากการที่ใช้คำสั่งการลบ ยกเว้น ถ้าใช้คำสั่ง ADDH หรือ SUBH คำสั่ง ADDH สามารถทำให้เกิดการเซ็ทและคำสั่ง SUBH จะทำให้เกิดการรีเซ็ทกับบิตตัวทด นอกจากนี้คำสั่งที่ใช้ในการเลื่อนบิตหรือหมุนบิตก็จะมีผลต่อบิตตัวทดนี้ เช่น คำสั่ง SC, RC และ LST1 และเมื่อมีสัญญาณรีเซ็ทบิตนี้จะถูกเซ็ทเป็น “1”

CNF บิตนี้จะใช้ในการควบคุมการใช้แรมบนตัวชิพ ถ้าบิตนี้ถูกเซ็ท “0” บล็อก BO จะถูกกำหนด ให้เป็นหน่วยความจำข้อมูล นอกเหนือจากนี้บล็อก BO จะถูกกำหนดให้เป็นหน่วยความจำโปรแกรมบิต CNF นี้อาจจะมีการเปลี่ยนแปลงได้โดยการใช้คำสั่ง CNFD, CNFP และ LST1 และเมื่อมีสัญญาณรีเซ็ทบิตนี้จะถูกเซ็ทเป็น “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DP (Data Memory Page Pointer) 9 บิตของ DP นี้จะมีความสัมพันธ์กับ 7 บิตทางด้านต่ำของเวิร์ดสั่ง ซึ่งเมื่อรวมกันแล้วจะเป็นรูปแบบของการอ้างแอดเดรสแบบโดยตรงขนาด 16 บิต DP นี้อาจจะมีการเปลี่ยนแปลงได้โดยการใช้คำสั่ง LST ,LDP และ LDPK

FO บิตนี้แสดงการจัดรูปแบบเมื่อบิตนี้ถูกเซ็ทเป็น "0" รีจิสเตอร์ของพอร์ตอนุกรมมี เป็นลักษณะรีจิสเตอร์ขนาด 16 บิต และเมื่อถูกเซ็ทเป็น "1" รีจิสเตอร์ที่พอร์ตจะมีหน้าที่รับและส่งข้อมูล 8 บิตไบต์ บิตFO นี้อาจจะมีการเปลี่ยนแปลงได้โดยการใช้คำสั่ง FORT และ LST1 เมื่อมีสัญญาณรีเซ็ทบิตนี้จะถูกเซ็ทเป็น "0"

FSM (Frame Synchronization Mode Bit) บิตนี้ใช้ในการแสดงปฏิบัติการของพอร์ตอนุกรมด้วยสัญญาณเฟรมซิงค์โดยใช้พัลส์จากภายนอก เมื่อ FSM = "1" จะมีการปฏิบัติการตามสัญญาณพัลส์ของเฟรมซิงค์ ซึ่งเป็นอินพุทของ FSX / FSR และถ้า FSM = 0 จะไม่สนใจอินพุทของ FSX / FSR และการปฏิบัติการบนพอร์ตอนุกรมจะทำต่อไปโดยไม่ต้องการเฟรมซิงค์พัลส์ และเมื่อมีสัญญาณรีเซ็ทบิตนี้จะถูกเซ็ทเป็น "1"

HM บิตแสดงสภาวะ โฮลด์เมื่อ HM = "1" ขบวนการต่างๆ ภายในก็จะหยุดเอ็กซ์ซีคิ้ว และถ้า HM = "0" ขบวนการต่างๆ ก็จะปฏิบัติการต่อจากที่หยุดไว้บิตซึ่งจะถูกเซ็ทเป็น " 1" เมื่อมีสัญญาณรีเซ็ท

INTM บิตนี้ใช้แสดงสถานะของการอินเตอร์รัพท์ เมื่อถูกเซ็ทเป็น "0" สัญญาณก็จะได้รับอีนาเบิล แต่ถ้าบิตนี้ถูกเซ็ทเป็น "1" สัญญาณอินเทอร์รัพท์แบบมาสเคเบิลทั้งหมดก็จะถูกดิสเอเบิล INTM นี้จะถูกเซ็ท และรีเซ็ทโดยใช้คำสั่ง DINT และ EINT นอกจากนี้ในการใช้งานคำสั่ง LST จะไม่เกิดผลใดๆ ต่อบิต INTM นี้

OV บิตแสดงการเกิดค่าเกินบิต OV นี้จะเซ็ทเป็น "1" เมื่อเกิดค่าเกินขึ้นใน ALU นอกจากนี้ถ้าต้องการเคลียร์ค่าต่างๆ ใน OV เราก็สามารถทำได้โดยการใช้คำสั่ง BV , BNV และ LST

OVM บิตแสดงสถานะเมื่ออยู่ในโหมดค่าเกินบิตนี้จะมีสถานะเป็น "0" ในกรณีที่เกิดการยกเลิกโหมดนี้ ในการเซ็ทและรีเซ็ทค่าที่บิตนี้เราจะใช้คำสั่ง SOVM และ ROVM ตามลำดับ นอกจากนี้คำสั่ง LST ก็จะมีผลให้เกิดการเปลี่ยนแปลงที่บิตนี้

PM โหมดการเลื่อนของผลคูณ ถ้าสองบิตนี้มีค่าเป็น 00 จะเป็นการโหลดค่าตัวคูณขนาด 32 บิต เข้าไปใน ALU โดยไม่มีการเลื่อน ถ้า PM = 01 เอาท์พุทของ PR จะถูกเลื่อนไปทางซ้าย 1 ครั้งก่อนโหลดเข้าไปใน ALU โดยบิตที่มีนัยสำคัญต่ำ (LSBs) จะถูกเติมด้วย 0 ถ้า PM = 10 เอาท์พุทของ PR จะถูกเลื่อนไปทางซ้าย 4 บิตก่อนจะถูกโหลดเข้าไปใน ALU โดยบิตที่มีนัยสำคัญต่ำ (LSBs)จะถูกเติมด้วย 0 และเมื่อ PM = 11 ผลจากการคูณจะถูกเลื่อนไปทางขวา 6 บิต บิต PM นี้สามารถทำการโหลดค่าเข้าไปได้โดยใช้คำสั่ง SPMและLST1เมื่อเกิดสัญญาณRSที่บิตนี้จะถูกเคลียร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SXM บิตแสดงเครื่องหมาย ถ้า SXM =1 ผลที่เกิดจากการคูณจะมีการคิดเครื่องหมาย แต่ถ้า SXM = 0 จะไม่มีการคำนึงถึงเครื่องหมาย ตัวอย่างเช่น คำสั่ง ADDS เป็นคำสั่งการบวกโดยไม่คิดเครื่องหมาย และไม่คำนึงถึงผลของ SXM การจะเซ็ทหรือรีเซ็ทบิตนี้สามารถทำได้โดยการใช้คำสั่ง SSXM และ RSXM นอกจากนี้ยังสามารถใช้คำสั่ง LST1 ในการโหลดค่าได้ด้วยบิต SXM จะเซ็ทเป็น "1" เมื่อเกิดสัญญาณรีเซ็ท

TC บิตทดสอบและควบคุมแฟลก บิต TC นี้จะมีผลเมื่อมีการเรียกใช้คำสั่ง BIT, BITT, CMPR, LST1, และ NORM บิตนี้จะเซ็ทเป็น "1" ถ้าการทดสอบบิต (Test Bit) ด้วยคำสั่ง BIT หรือ BITT เป็น "1" หรือเมื่อมีการทดสอบด้วยคำสั่ง NORM แล้ว Exclusive-OR ฟังก์ชั่น ของสองบิตที่มีนัยสำคัญสูงของแอคคิวมูเลเตอร์จากการเปรียบเทียบกันแล้วเป็นจริง

TXMบิตนี้ใช้แสดงสถานการณ์ส่งเมื่อ TXM=1 ขา FSX ของพอร์ตอนุกรมจะถูกกำหนดให้เป็นเอาท์พุท ในโหมดนี้เมื่อ DXR ถูกโหลดค่าจะมีการสร้างพัลส์ให้กับ FSX การส่งจะเริ่มจากที่ขา DX ถ้า TXM = 0 ขา FSX จะมีลักษณะเป็นอินพุท บิต TXM นี้จะสามารถเซ็ทหรือรีเซ็ทได้โดยการใช้คำสั่ง STXM และ RTXM เมื่อมีสัญญาณรีเซ็ทบิต TXM นี้จะเป็น "0"

XF ที่บิตนี้จะเป็นการแสดงสถานะของขา XF ซึ่งโดยทั่วไปแล้วจะทำหน้าที่เป็นขาเอาท์พุท บิต XF นี้จะทำการเซ็ทหรือรีเซ็ทได้โดยการใช้ คำสั่ง SXF และ RXF หรืออาจจะโหลดค่าโดยการใช้คำสั่ง LST1 บิต XF นี้เมื่อเกิดสัญญาณรีเซ็ทจะถูกเซ็ทให้เป็น "1"

2.6.15 ปฏิบัติการเวลา

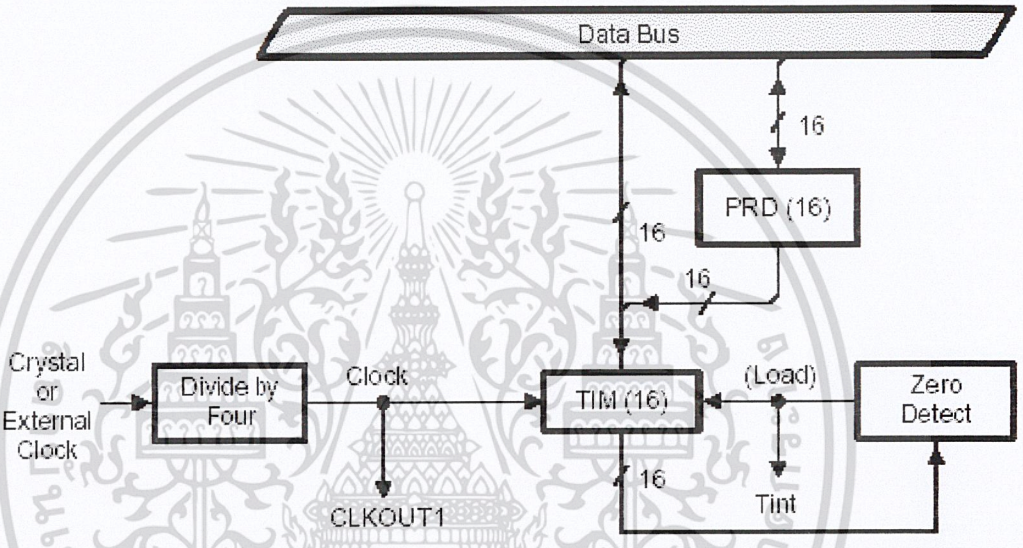
การจัดผังหน่วยความจำของ TMS320C26 กำหนดให้รีจิสเตอร์ไทม์เมอร์ (TIM) และพีเรียด (PRD) มีขนาด 16 บิต ซึ่งแสดงดังรูป 2-12 ส่วนวงจรนาฬิกาภายในตัวชิพจะใช้สัญญาณคล็อก (Clock) จาก CLKOUT1

ในขณะที่เกิดสัญญาณรีเซ็ทนั้นรีจิสเตอร์ TIM และรีจิสเตอร์ PR นี้ จะถูกเซ็ทให้มีค่าสูงสุด (>FFFF) และจะลดค่าลงหลังจากมีการถอนสัญญาณรีเซ็ท จากที่กล่าวมานี้รีจิสเตอร์ TIM และ PRD ยังสามารถจะโหลดค่าใหม่ได้โดยใช้การโปรแกรมคอนโทรลสำหรับ TIM รีจิสเตอร์หน่วยความจำข้อมูลพื้นที่ 2 ไทม์เมอร์จะเริ่มนับที่ทุกๆ $N \times CLKOUT1$ เมื่อ N = 1 รีจิสเตอร์TIM นี้จะลดค่าลงหนึ่งส่วนรีจิสเตอร์PRD แลหน่วยความจำข้อมูลพื้นที่ 3 จะเริ่มนับโดยวงจรนาฬิกาในส่วนไทม์เมอร์อินเทอร์รัพท์ (TINT) จะสร้างสัญญาณออกมาตลอดเวลาที่ไทม์เมอร์เริ่มลดค่าลงไปถึง 0 ไทม์เมอร์นี้เราสามารถที่จะโหลดค่าเข้าไปใหม่ได้โดยใช้ค่าที่อยู่ในรีจิสเตอร์ PRD ภายในไซเกิลถัดไปหลังจากที่ลดค่าลงจนถึง 0 ไทม์เมอร์ และพีเรียด รีจิสเตอร์สามารถที่จะอ่านหรือเขียนได้ภายในไซเกิลหนึ่งไซเกิลใดก็ได้ ซึ่งสามารถตรวจค่าข้อมูลได้โดยการอ่านค่าจากรีจิสเตอร์ TIM นี้ และถ้า

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่มีการใช้งานวงจรนาฬิกาเราสามารถที่จะทำให้ TINT นี้ดิสเคเบิลต่อสัญญาณอินเตอร์รัพท์ชนิดมาสเคเบิลได้ โดยการใช้คำสั่ง DINT จุดประสงค์ทั่วไปของรีจิสเตอร์ PRD คือใช้เป็นพื้นที่ของหน่วยความจำข้อมูล และถ้ามีการใช้งาน TINT รีจิสเตอร์ TIM และ PRD จะถูกใช้เป็นโปรแกรมก่อนที่จะถอนสัญญาณเจาะ TINT

Timer Block Diagram



รูปที่ 2-12 บล็อกไดอะแกรมของวงจรตั้งเวลา

2.6.16 การนับที่ทำให้เกิดการทำงานซ้ำ (Repeate Counter)

วงจรนับที่ทำให้เกิดการทำงานซ้ำ (RPTC) จะเป็นวงจรนับขนาด 8 บิต เมื่อมีการโหลดด้วยค่าตัวเลข N จะมีผลให้เกิดการเอ็กซ์ซีคิวคำสั่งพัดไปที่เวลา N÷1 โดยค่าที่สามารถโหลดเข้าไปยัง RPTK (Repeat Immediate) และผลลัพธ์ที่สามารถจะเอ็กซ์ซีคิวได้คือ 256 ข้อมูลต่างๆใน RPTC นี้จะถูกเคลียร์โดยสัญญาณรีเซ็ท

ลักษณะของการทำงานแบบซ้ำๆ กันนี้ จะมีใช้ในคำสั่งที่เกี่ยวกับการคูณ (MAC / MACD) การเคลื่อนย้ายข้อมูลเป็นบล็อก (BLKD/BLKP) การโอนย้ายข้อมูลระหว่าง I/O (IN/OUT) และการอ่านเขียนตาราง (TBLR/TBLW) คำสั่งเหล่านี้ในการทำงานโดยปกติแล้วจะใช้หลายไซเกิล แต่ถ้าใช้การทำงานในลักษณะซ้ำๆ (REPEAT) จะทำให้สามารถทำงานได้โดยใช้เพียงไซเกิลเดียว ตัวอย่าง เช่น คำสั่งในการอ่านตารางในการเอ็กซ์ซีคิวจะใช้ 3 ไซเกิลหรือมากกว่า แต่ถ้าใช้การทำงานลักษณะซ้ำๆ กันนี้ พื้นที่ในตารางสามารถที่จะทำการอ่านได้ในทุกๆ ไซเกิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.17 โหมดเพาว์เวอร์ดาวน์

เมื่อมีการทำงานในโหมดเพาว์เวอร์ดาวน์ TMS320C26 จะอยู่ในสภาวะที่ไม่มีการเปลี่ยนแปลง โดยต้องการกำลังงานเพียงครึ่งหนึ่งของกำลังงานปกติเท่านั้น เพื่อจ่ายให้กับอุปกรณ์โหมดเพาว์เวอร์ดาวน์นี้สามารถจะทำการกระตุ้นได้โดยการใช้คำสั่ง IDLE หรือทำให้สัญญาณโฮลด์มีสถานะเป็นโลว์ โดยที่บิตสถานะ HM ถูกเซ็ทเป็น "1" การยกเลิกโหมดเพาว์เวอร์ดาวน์นี้ ทำได้โดยการใช้คำสั่ง IDLE หรือ โดยการยกเลิกสัญญาณโฮลด์

2.7 อินเตอร์รัพท์ (Interrupt)

TMSS20C2X มี 3 mask able อินเตอร์รัพ ภายนอก ($\overline{INT2}$ - $\overline{INT0}$) สำหรับอุปกรณ์ภายนอกที่จะมาต่อส่วน Interrupt ภายในก็จะมี XINT/RINT ซึ่งถูกสร้างโดยพอร์ตอนุกรม (Serial Port) , TINT ซึ่งถูกสร้างโดยไทม์เมอร์และ TRAP ซึ่งถูกสร้างซอร์ฟแวร์ โดยรีเซ็ต (Reset) จะมีความสำคัญ (Priority)สูงสุดและ XINT จะมีความสำคัญต่ำที่สุด

2.7.1 การทำงานของอินเตอร์รัพท์ (Interrupt Operation)

ในส่วนนี้จะอธิบายถึงรายละเอียดการทำงานและการจัดการของอินเตอร์รัพท์ ตาราง 2-5 แสดงตำแหน่งของอินเตอร์รัพท์ (Vector allocation) และความสำคัญของทั้งอินเตอร์รัพท์ภายในและภายนอกคำสั่ง TRAP ซึ่งถูกใช้โดยซอร์ฟแวร์ นั้นไม่ถูกจัดลำดับความสำคัญไว้ แต่จะถูกรวมอยู่ในที่นี้ด้วยเพราะว่ามันมีตำแหน่งของมันเอง

ตารางที่ 2-5 แสดง Interrupt Location และ Priority

Interrupt Name	Memory Location	Priority	Function
RS	0h	1 (Highest)	External reset signal
INT0	1h	2	External user interrupt # 0
INT1	2h	3	External user interrupt # 1
INT2	3h	4	External user interrupt # 2
	8-17h		Reserved locations
TINT	18h	5	Internal timer Interrupt
RINT	1Ah	6	Serial Port receive Interrupt
XINT	1Ch	7 (Lowest)	Serial Port receive Interrupt
TRAP	1Eh	N/A	Trap instruction address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีอินเตอร์รัพท์เกิดขึ้น มันจะถูกเก็บไว้ใน IFR (Interrupt Flag Register) รีจิสเตอร์ 6 BIT FIR นี้จะถูกตั้งค่าโดยอุปกรณ์ภายนอก ($\overline{INT2}$ - $\overline{INT0}$) และอินเตอร์รัพท์ภายใน RINT, XINT และTINT แต่ละการอินเตอร์รัพท์จะถูกเก็บไว้ใน IFR จนกว่ามันจะถูกใช้ และหลังจากนั้นมันก็จะถูกลบทิ้งโดยอัตโนมัติ โดย $\overline{IACK}$ (Interrupt Acknowledge) หรือ $\overline{RS}$ (reset) สัญญาณ RS จะไม่ถูกเก็บใน IFR และไม่มีคำสั่งในการอ่านหรือเขียนใน IFR ได้

TMS320C26 จะมีรีจิสเตอร์ IMR (Interrupt mask register) สำหรับมาสกิ้ง (Masking) ทั้งอินเตอร์รัพท์ภายนอกและภายใน รายละเอียดของ IMR จะเป็นดังรูป 2-13 โดยจะใช้เพียง 6 bit คือ บิต 0-5 ของ IMRเท่านั้น ดังรูป และเราสามารถที่จะอ่านและเขียน IMR ได้ แต่ไม่สามารถอ่านโดยคำสั่ง BLKD ได้ เมื่อ IMR ถูกอ่านบิตที่ไม่ใช้ (15 ถึง 6) ก็จะถูกอ่านเท่ากับ 1 สังเกตว่า $\overline{RS}$ จะไม่ถูกรวมอยู่ใน IMR เพราะฉะนั้น Reset จะไม่มีผลกับ IMR

15 ... 6	5	4	3	2	1	0
RESERVED	XINT	RINT	TINT	$\overline{INT2}$	$\overline{INT1}$	$\overline{INT0}$

รูป 2-13 แสดง Interrupt mask registerv

INTM (Interrupt Model) bit ซึ่งเป็นบิตที่ 9 ของรีจิสเตอร์ ST0 จะเป็นตัว enable/disable มาสกิ้งอินเตอร์รัพท์ (Mask able Interrupt) ทุกตัวโดย

INTM=0 จะ enable ทุกอันมาสกิ้งอินเตอร์รัพท์ (Unmask able Interrupt)

INTM=1 จะ disable ทุกอันมาสกิ้งอินเตอร์รัพท์

INTM จะถูก set ให้เป็น “1” โดย $\overline{IACK}$ โดยคำสั่ง DINTหรือ Reset บิต INTM จะถูก reset เป็น “0” โดยคำสั่ง EINT สังเกตว่า INTM ไม่ได้ทำการเปลี่ยน IMR หรือ IFR โดยตรง

IMS320C26 มีวงจรป้องกันการอินเตอร์รัพท์ ในระหว่างการทำคำสั่งที่มีหลายๆ Cycle รวมอยู่ด้วยในตัว ถ้าเกิดการอินเตอร์รัพท์ขึ้นในระหว่างคำสั่ง ใช้หลายไซเคิล (Cycle) เครื่องจะรอจนกว่าคำสั่งนั้นจะเสร็จสิ้นแล้วถึงทำการอินเตอร์รัพท์ซึ่งจะรวมไปถึงคำสั่งที่ต้องรอสัญญาณ Ready จนต้องใช้หลายไซเคิลด้วย

นอกจากนี้อุปกรณ์ยังไม่อนุญาตให้ทำการอินเตอร์รัพท์ ในระหว่างการทำคำสั่ง RPT หรือ RPTK (Repeat) อินเตอร์รัพท์จะถูกเก็บไว้ใน IFR จนกว่ารีพีทเคาน์เตอร์ (Repeat Counter) จะลดลงเหลือ 0 (ทำคำสั่ง repeat เสร็จ) จากนั้นอินเตอร์รัพท์จึงจะทำงาน แม้ว่าอินเตอร์รัพท์จะไม่ถูก

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ระหว่างการทำคำสั่ง RPT หรือ RPTK อินเตอร์รัพท์ยังคงอ่านและเก็บไว้ใน IFR และรอจนกว่ารีพีทเคาน์เตอร์จะลดลงเหลือ 0

ถ้า $\overline{HOLD}$ และอินเตอร์รัพท์ เกิดขึ้นในระหว่างคำสั่งที่ใช้หลายไซเคิล หรือคำสั่งทำซ้ำ จะทำ HOLD ก่อนในทันทีหลังจากเสร็จสิ้นคำสั่งหลายไซเคิล หรือคำสั่งทำซ้ำ เมื่อ $\overline{HOLD}$ หมดไปถึงจะทำ อินเตอร์รัพท์ต่อได้

อินเตอร์รัพท์จะไม่สามารถทำงานในระหว่างคำสั่ง EINT และคำสั่งต่อไปในโปรแกรมที่ตามมา เช่น ถ้าเกิดมีการร้องขออินเตอร์รัพท์ ในระหว่างการทำคำสั่งEINT เครื่องจะทำคำสั่ง EINT และคำสั่งถัดมาให้เสร็จก่อนจึงทำการอินเตอร์รัพท์ ซึ่งมักจะใช้คำสั่ง RET เป็นคำสั่งที่ต่อจาก EINT

2.8 พอร์ตอนุกรม (Serial Port)

พอร์ตอนุกรมบน TMS320C26 นี้เป็นชนิด Full Duplex ซึ่งจะสนับสนุนการทำงานของอุปกรณ์อนุกรมต่างๆ เช่น Codes , A/C Converter และยังสามารถใช้ในการติดต่อกันในระหว่างโปรเซสเซอร์กรณีใช้งานหลายโปรเซสเซอร์

ทั้งการส่งและรับนั้นจะใช้บัฟเฟอร์สองตัว (Double Buffer) บน TMS320C26 ดังนั้นจะอนุญาตให้ส่งต่อเนื่อง (Continuous Bit Stream) แม้ว่า FSX จะถูกใช้เป็นเอาท์พุท การใช้บิตเฟรมซิงค์โหมด (Frame Syn Mode : FSM) จะทำให้เกิดการส่งในแบบต่อเนื่องซึ่งไม่ต้องการสัญญาณเฟรมซิงค์ (Frame Syn Pulse) และไม่ต้องการความถี่ CLKR หรือ CLKX (fmin = 0Hz)

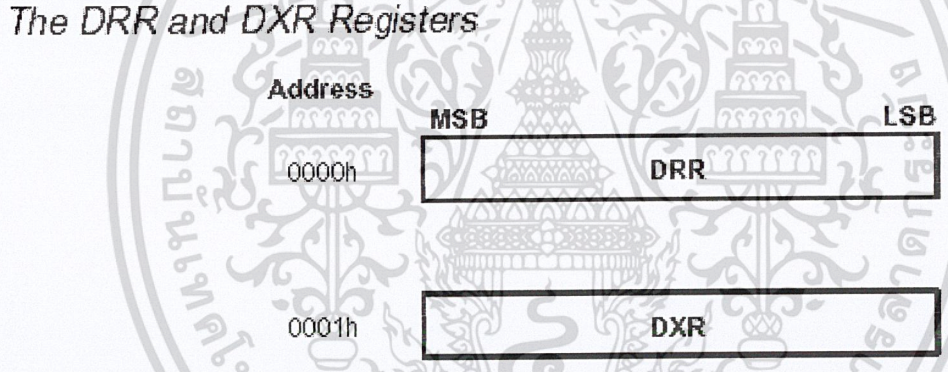
Serial Port Bits, Pins, and Registers

Serial Port Bits/Pins/Registers		TMS320C25
FO	Format bit	Yes
TXM	Transmit mode bit	Yes
FSM	Frame synchronization mode bit	Yes
CLKX	Transmit clock signal	Yes
CLKR	Receive clock signal	Yes
DX	Transmitted serial data signal	Yes
DR	Received serial data signal	Yes
FSX	Transmit framing synchronization signal	Yes
FSR	Receive framing synchronization signal	Yes
DXR	Data transmit register	Yes
DRR	Data receive register	Yes
XSR	Transmit shift register	Yes
RSR	Receive shift register	Yes

ตาราง 2-6 แสดง Serial Port Bits,Pins และ Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2-6 แสดงบิต ขา register ที่ควบคุมการทำงานของพอร์ตอนุกรม โดยจะมีรีจิสเตอร์ 2 ตัว คือ DXR (Data Transmit Register) ซึ่งเก็บค่าข้อมูลที่จะถูกส่งโดยพอร์ตอนุกรมและ DRR (Data Receive Register) ซึ่งเก็บค่าที่รับได้ ดังรูป 2-14 ทั้งคู่จะทำงานได้ทั้งแบบ 8 bit byte mode หรือ bit word mode และใช้งานได้เหมือนกับหน่วยความจำข้อมูล (Data memory) ตำแหน่งอื่นๆ ใน register แต่ละตัวจะมีสัญญาณนาฬิกาจากภายนอก (External CLK) , สัญญาณ เฟรมซิงค์ และชิฟท์รีจิสเตอร์ (Shift Register) คำสั่งทุกคำสั่งที่ใช้ในการอ่าน หรือเขียนหน่วยความจำข้อมูล สามารถใช้กับรีจิสเตอร์ 2 ตัวนี้ (DXR และ DRR) ได้แก่ BLKD (Block Move from Data Memory to Data Memory) ไม่สามารถนำมาใช้อ่านรีจิสเตอร์เหล่านี้ได้ DXR และ DRR จะถูกแมพ (Map) ไว้ที่ตำแหน่ง 0และ 1 ในหน่วยความจำข้อมูล และ XSR กับ RSR ไม่สามารถใช้งานได้โดยตรงผ่านซอฟแวร์



รูปที่ 2-14 แสดง DXR และ DRR รีจิสเตอร์

ถ้าพอร์ตอนุกรมไม่ถูกใช้ DXR และ DRR จะใช้เป็นเพียงรีจิสเตอร์ธรรมดาๆ ได้ ในกรณีนี้ CLKR หรือ FSR ควรต่อกับลอจิก LOW

ใน ST1 (Status Register) บิตที่ใช้เพื่อควบคุมการทำงานของพอร์ตอนุกรม คือ FO, TXM และ FSM โดยบิต FO (format) จะเป็นตัวกำหนดว่าจะส่งและรับข้อมูลในแบบ 8บิต หรือ 16 บิต ถ้า F0 = 0 ก็จะใช้แบบ 16 บิต ถ้า F0 =1 ข้อมูลก็จะรับส่งแบบ 8 บิต ใน 8 บิต Mode นั้น 8 บิตล่างเท่านั้นที่จะใช้ในการส่งและรับข้อมูล F0 จะถูกกำหนดโดยคำสั่ง FORT (Format Serial Port Register) ในกรณีรีเซ็ท F0 จะเป็น

บิต TXM (Transmit Mode) จะใช้ในการกำหนดว่า Frame Syn Pulse ของการส่งข้อมูลจะถูกสร้างจากภายในหรือภายนอก ถ้าTXM = 1 ขา FSX จะเป็นขาเอาท์พุทและสัญญาณเฟรมซิงค์

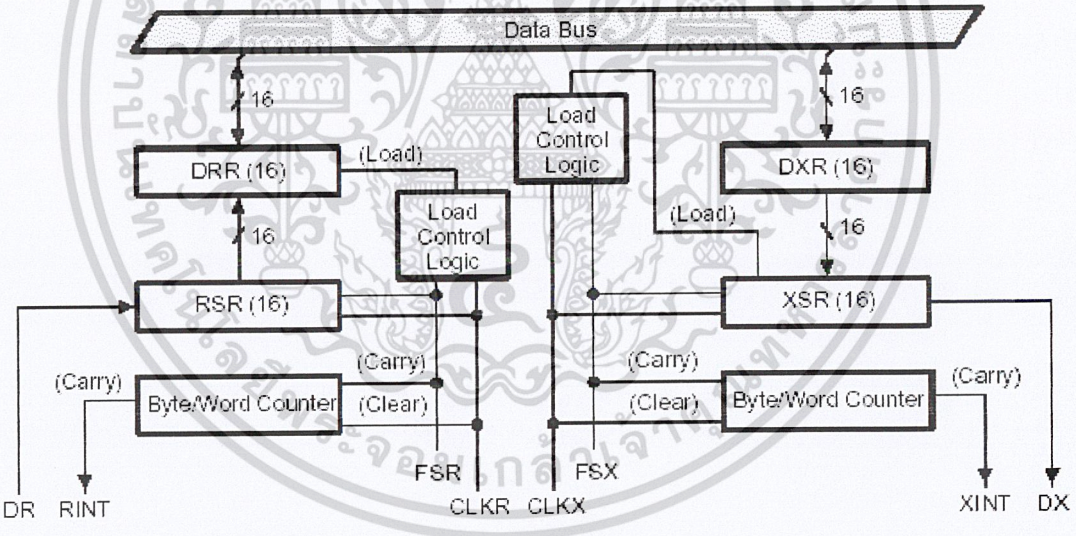
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะถูกสร้างขึ้นบนขา FSX ทุกๆ ครั้งที่ DXR ได้รับข้อมูลที่จะส่ง และสัญญาณเฟรมซิงค์นี้จะถูกกำหนดให้ ซิงโครไนส์ (Synchronize) กับ CLKX ที่ขอบขาขึ้น ถ้า TXM = 0 ขา FSXจะเป็นขาอินพุท TMS320C26 จะรอสัญญาณเฟรมซิงค์จากภายนอกก่อนแล้วค่อยเริ่มต้นส่ง ในกรณีรีเซ็ท TXM จะถูกรีเซ็ทเป็น 0, FSX เป็นอินพุท และบิต TXM สามารถใช้คำสั่ง LST1, STXM หรือ RTXM ในการกำหนดค่าได้

บิตFSM (Frame Syn Mode) ใน ST1 กำหนดว่าจะใช้สัญญาณเฟรมซิงค์หรือไม่ ถ้าFSM= 1 จะใช้สัญญาณเฟรมซิงค์และ FSM = 0 จะไม่ใช้สัญญาณเฟรมซิงค์ SFSM จะทำให้ FSM = 1 และ RFSM จะทำให้ FSM = 0

2.9 การส่งและรับข้อมูล (Transmit and Receive Operation)

ในส่วนของการส่งและรับข้อมูลของพอร์ตอนุกรมจะแยกกันไม่ขึ้นแก่กัน รูป2-15 แสดงรีจิสเตอร์และขาที่ใช้ในการส่งและรับข้อมูล



รูป 2-15 แสดง Serial Port Block Diagram

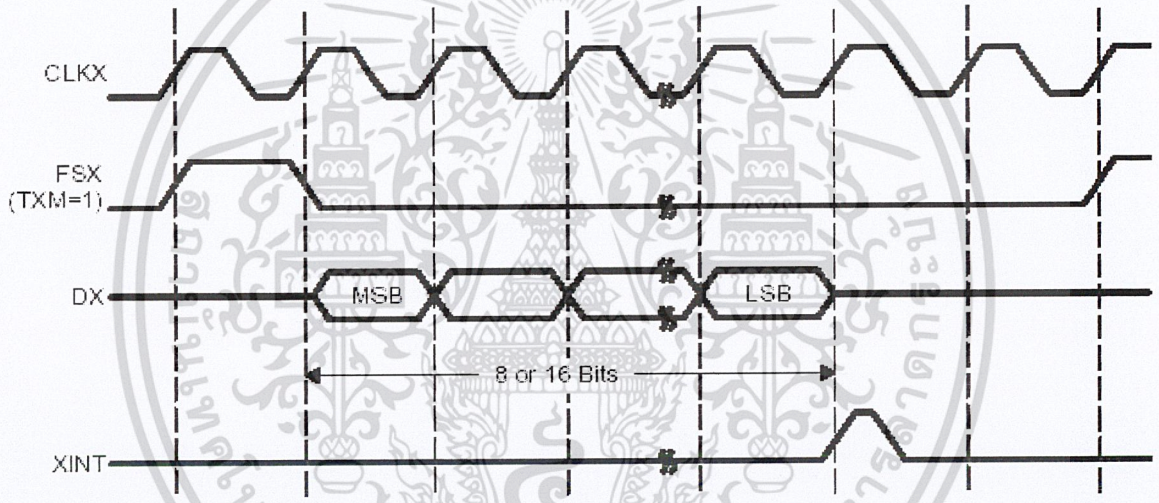
ข้อมูลขา XSR จะถูกส่งออกขา DX โดยมี CLKX เป็นตัวกำหนดจังหวะและข้อมูลจากขา DR จะเข้ามายัง RSR โดยมี CLKR เป็นตัวกำหนดจังหวะ โดย CLKX และ CLKR นี้สามารถที่จะหยุดไม่ทำงานเวลาที่ไม่มีการส่งหรือรับได้ คือ ถ้ามีการส่งหรือรับ CLKX หรือ CLKR จะทำงานกำหนดจังหวะ แต่ถ้าไม่มีข้อมูลส่งรับ CLKX หรือ CLKR ก็จะมีค่าคงที่ ที่ระดับหนึ่ง (หยุดการทำงาน) ข้อมูลที่ขา DX จะส่งที่ขอบขาขึ้นของ CLKX ในขณะที่ข้อมูลจากขา DR จะถูกรับที่ขอบขาลงของ CLKR และจะส่งบิตสูงก่อนแล้วค่อยส่งบิตต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XSR และ RSR จะต่อเข้ากับ DXR และ DRR ตามลำดับ ในการส่งข้อมูลจาก DXR จะถูกส่งไปยัง XSR เมื่อมีการเริ่มส่งข้อมูลใหม่ แต่ในการรับข้อมูล ข้อมูลในRSR จะถูกส่งไปยัง DRR เมื่อ RSR รับข้อมูลครบทุกบิต (8 หรือ 16 บิต) แล้ว ดังนั้นข้อมูลใน DXR และ DRR สามารถที่จะเป็นคนละตัวกับข้อมูลที่ส่งหรือรับอยู่ได้ เพราะว่ามีบัฟเฟอร์ 2 ตัวนั่นเอง

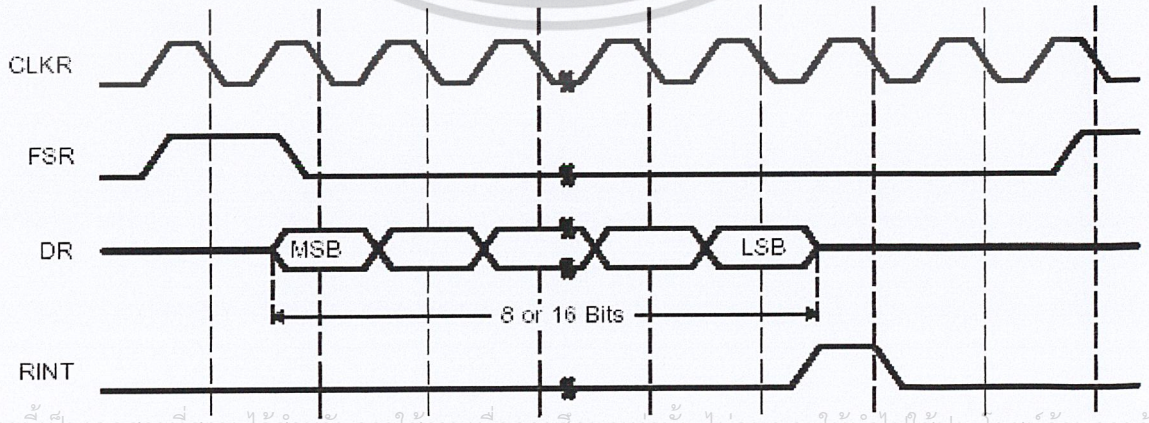
กระบวนการโอนถ่ายข้อมูลของพอร์ตอนุกรมบน TMS320C26 นั้นปกติจะเริ่มต้นด้วยสัญญาณเฟรมซิงค์ ยกเว้นกรณีที่ใช้โหมดการรับส่งแบบต่อเนื่อง (FSM = 0) และสัญญาณเฟรมซิงค์ของการส่งและการรับจะเป็นอินพุท

Serial Port Transmit Timing Diagram



รูป2-16 แสดงผังเวลาในการส่งของ Serial Port

Serial Port Receive Timing Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2-17 แสดงผังเวลาในการรับของ Serial Port

การส่งข้อมูลจะมีผังเวลา ดังรูป 2-16 การส่งจะเริ่มเมื่อ ข้อมูลถูกเขียนลงใน DXR เครื่องก็จะเริ่มส่งข้อมูล เมื่อสัญญาณเฟรมซิงค์เป็น 0 และ CLKX เริ่มเป็น 1 หรือเป็น 1 การส่งจะเริ่มจาก MSB และเลื่อนการส่งไปบิตต่อไปทุกๆ ขอบขาขึ้นของ CLKX ออกทางขา DX เมื่อส่งเสร็จทุกบิตแล้ว XINT ก็จะถูกสร้างขึ้นที่ขอบขาขึ้นของ CLKX เมื่อไม่มีการส่งข้อมูล DX จะอยู่ในสภาวะอิมพีแดนซ์สูง

$\overline{HOLD}$ จะไม่มีผลกับ DX และ FSX เลย คือถ้าหากมีสัญญาณ $\overline{HOLD}$ เข้ามา เครื่องก็จะทำการส่งข้อมูลที่มีอยู่ผ่านทางขา DX ให้เสร็จก่อน แล้วจึงเข้าสู่สภาวะอิมพีแดนซ์สูง และ FSX ก็ยังคงเป็นเอาท์พุทหรืออินพุท ตามที่เคยตั้งไว้

ในส่วนขบวนการรับข้อมูลก็คล้ายกัน ดังแสดงในรูป 2-17 เพียงแต่ว่าการรับจะเริ่มต้นด้วยสัญญาณเฟรมซิงค์ที่ขา FSR หลังจากที่ FSR เริ่มเป็น 0 ข้อมูลในขา DR จะถูกส่งเข้าสู่รีจิสเตอร์ RSR ที่ทุกๆ ขอบขาลงของ CLKR ข้อมูลที่รับได้บิตแรกจะเป็น MSB อีกเช่นกัน หลังจากเสร็จสิ้นการรับข้อมูลทุกบิตแล้ว RINT ก็จะถูกสร้างขึ้นที่ขอบขาขึ้นของ CLXR และค่าใน RSR จะถูกนำไปเก็บไว้ใน DRR

หลังจากที่เสร็จการโอนถ่ายข้อมูลแล้ว อินเตอร์รัพท์ภายใน (XINT / RINT) โดย XINT จะใช้สำหรับการส่ง RINTจะใช้สำหรับการรับ สังเกตว่าถ้า DRR ถูกอ่านก่อนที่จะมี RINT เข้ามา ข้อมูลที่อ่านได้จะเป็นข้อมูลตัวเดิมไม่ใช่ข้อมูลตัวปัจจุบัน เช่นเดียวกัน ถ้า DXR ใส่ข้อมูลมากกว่า 1 ครั้ง ต่อ 1 XINT แล้ว ค่าที่ส่งจะมีแค่ค่าสุดท้ายที่ใส่เข้าไปใน DXR เท่านั้นส่วนค่าอื่นๆ จะไม่ถูกส่งออกไป

เมื่อ TMS320C26 ถูกรีเซ็ท TXM = 0 และ DX จะอยู่ในสภาวะอิมพีแดนซ์สูง การรับส่งใดๆ ในระหว่างนั้นจะถูกหยุดหมด

สัญญาณเฟรมซิงค์ของการส่งจะสามารถเลือกได้ว่าจะถูกสร้างจากภายในหรือภายนอก ความเร็วสูงสุดในการรับส่งข้อมูลของพอร์ตอนุกรม คือ 5 MHz และยังสามารถใช้ได้กับ CODEC ตระกูล TI / Intel 23C1x โดยการกลับค่าสัญญาณเฟรมซิงค์

2.9.1 การส่งสัญญาณในแบบ เบิร์สโหมด (Burst- Mode Operation)

ในการทำงานแบบเบิร์สโหมดของพอร์ตอนุกรมนั้น การโอนถ่ายข้อมูลจะไม่ต่อเนื่อง คือจะทำการรับส่งข้อมูลเป็นช่วงๆ การส่งแบบนี้ บิต FSM=1 และผังเวลาของการส่งแสดงใน รูป 2-18

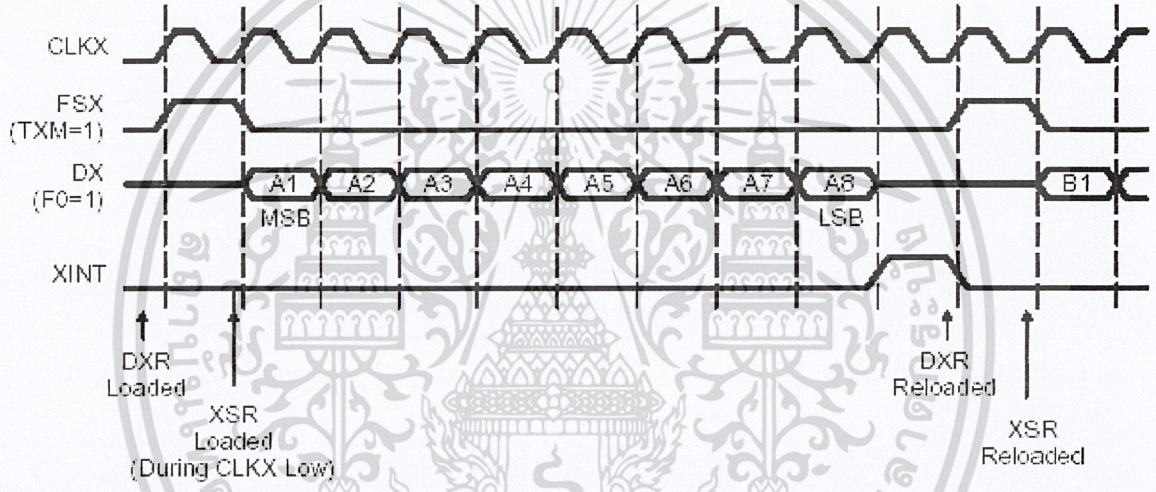
เมื่อ TXM=1 (FSX เป็นเอาท์พุท) และ DXR ถูกใส่ข้อมูล สัญญาณเฟรมซิงค์จะถูกส่งออกมาที่ขาขึ้นของ CLKX ลูกถัดไป และข้อมูลใน DXR ก็จะนำมาใส่ใน XSR เมื่อ FSX เป็นลอจิกสูง และCLKX เป็นสถานะต่ำ การส่งข้อมูลจะเริ่มเมื่อ FSX เป็นลอจิกต่ำ และ CLKX เป็นลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูง รูป 2-18 แสดงผังเวลาในกรณีที่ส่งแบบ 8บิต (FO = 1) เมื่อส่งเสร็จ DX จะอยู่ในสภาวะอิมพีแดนซ์สูง และ XINT ก็จะถูกสร้าง

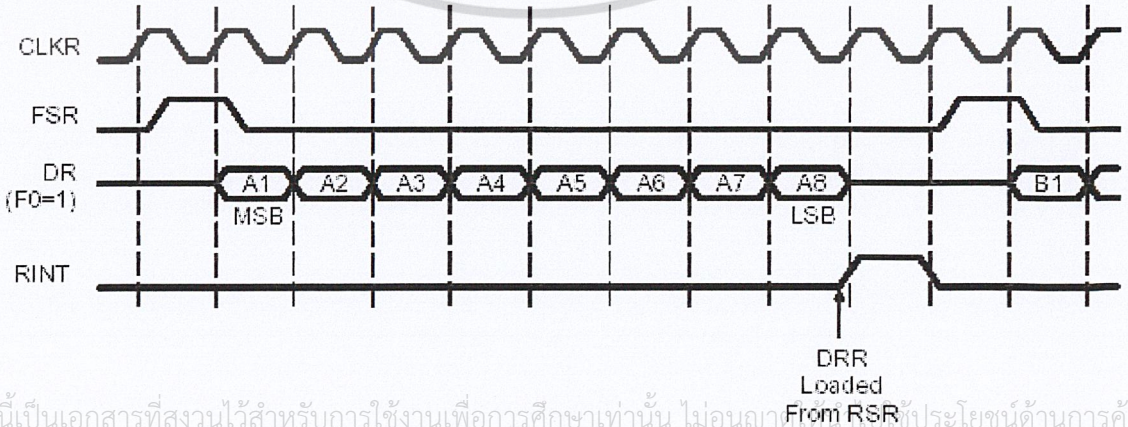
ในส่วนการรับก็จะคล้ายกับการส่ง คือข้อมูลใน RSR จะถูกใส่ลงใน DRRในขณะที่ CLKR มีลอจิกต่ำ ทันทีที่รับบิตสุดท้ายเสร็จ ดังในรูป 2-19 RINT จะถูกสร้างขึ้นที่ขอบขาขึ้นของ CLKR ลูกถัดไป เราสามารถอ่านค่าใน DRR เมื่อไรก็ได้ก่อนที่จะมีการรับค่าบิตสุดท้ายของข้อมูลตัวต่อไป

Burst-Mode Serial Port Transmit Operation



รูปที่ 2-18 แสดงผังเวลาของการส่งในแบบ Burst- Mode

Burst-Mode Serial Port Receive Operation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

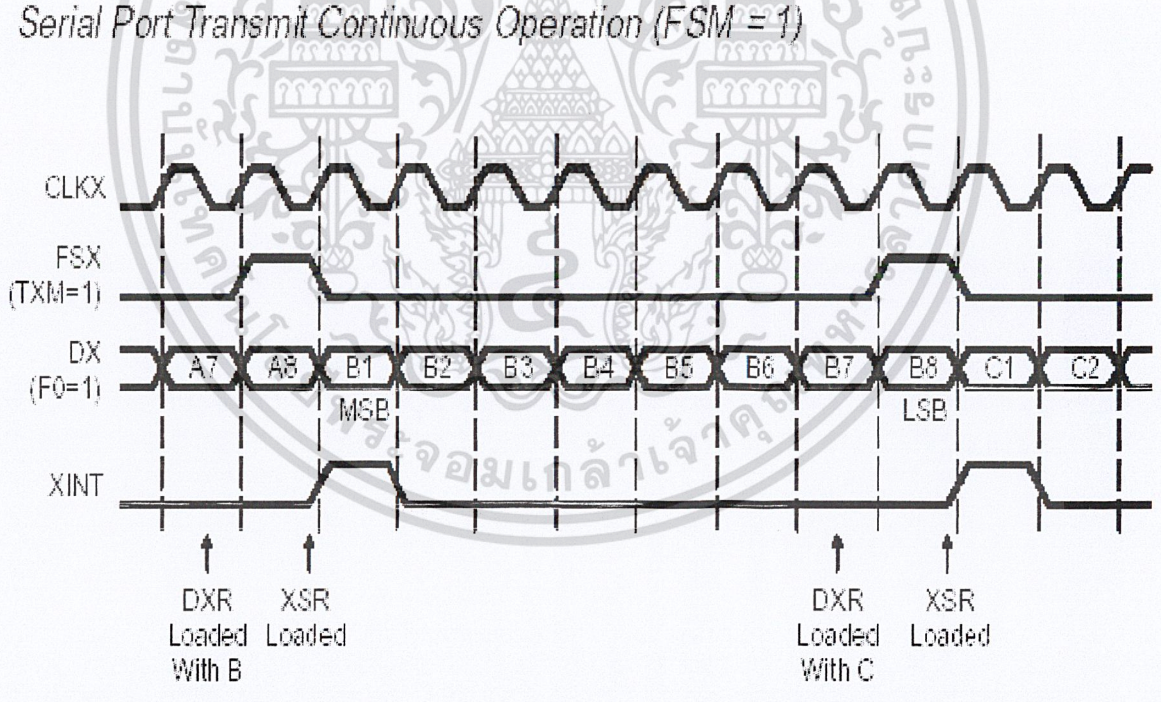
รูปที่ 2-19 แสดงผังเวลาของการรับในแบบ Burst- Mode

2.9.2 การรับส่งข้อมูลในโหมดต่อเนื่องแบบใช้สัญญาณเฟรมซิงค์

(Continuous Operation Using Frame Syn Pulse)

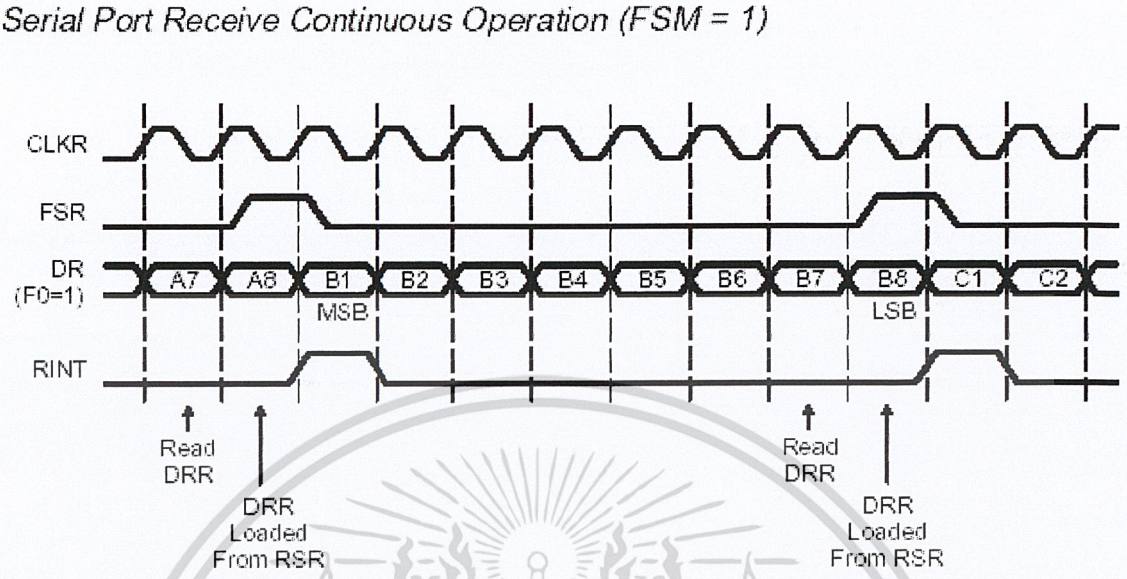
จะเป็นการส่งที่ FSM = 1 และค่าใหม่ถูกใส่ลงใน DXR ก่อนที่จะส่งบิตสุดท้ายของค่าเก่าเสร็จ เพราะว่าพอร์ตอนุกรมที่มีอยู่นั้น มี BUFFER 2 ตัว การเขียนข้อมูลตัวใหม่ลงใน DXR จะไม่ทำให้การส่งข้อมูลที่มีอยู่นั้นหยุดลง แต่ DXR จะเก็บค่าใหม่ไว้จนกว่าจะมีการส่งค่าเก่าเสร็จสิ้น จึงสามารถที่จะส่งในแบบต่อเนื่องได้ แม้ว่า TXM=1 ก็ตาม

สัญญาณเฟรมซิงค์ของการส่งจะมีลอจิกสูงที่ขอบขาขึ้นของ CLKX ในขณะส่งบิตสุดท้าย และเป็นลอจิกต่ำที่ขอบขาลงของ CLKX ลูกถัดมา การส่งจะเป็นแบบต่อเนื่องตราบเท่าที่อยู่ภายใต้เงื่อนไขดังกล่าวข้างต้น หากว่าข้อมูลใหม่ไม่ถูกใส่ลงใน DXR ก่อนที่จะส่งค่าเก่าเสร็จ DX ก็จะกลับเป็นสภาวะอิมพีแดนซ์สูง คือ กลับไปทำงานแบบ เบิร์สโหมดนั่นเอง รูป 2-20 และ 2-21 แสดงผังเวลาของการรับส่งในแบบนี้



รูป 2-20 แสดงผังเวลาของการส่ง Continuous แบบใช้ Frame Syn Pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2-21 แสดงผังเวลาของการรับ Continuous แบบใช้ Frame Syn Pulse

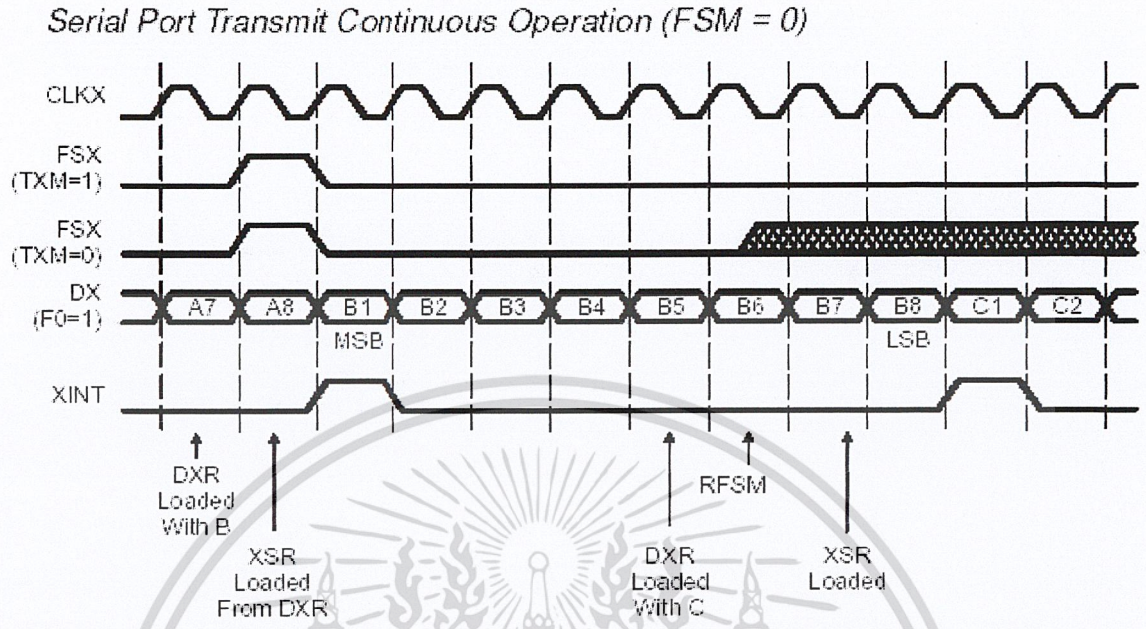
2.9.3 การรับส่งข้อมูลในโหมดต่อเนื่องแบบไม่ใช้สัญญาณเฟรมซิงค์ (Continuous Operation without Frame Syn Pulse)

ในการส่งแบบนี้จะต้องให้ FSM=0 สามารถเลือกใช้ได้ทั้งในแบบการส่ง8บิตและ16 บิตรูป 2-22 และ รูป 2-23 แสดงการทำงานของพอร์ตอนุกรม เมื่อ TXM = 0 และ TXM = 1 เครื่องจะเริ่มต้นด้วย FSM = 1 ซึ่งเราจะต้องเปลี่ยนให้ FSM = 0 ก่อนที่การส่งข้อมูลชุดแรกจะเสร็จสิ้น (ก่อนที่จะมีการอินเตอร์รัพท์) โดยการใช้คำสั่ง RFSM (Reset FSM) ในกรณีนี้เราจะไม่ใช้ FSX และ FSR ในการควบคุมการรับการส่งอีกต่อไป แต่จะรับส่งข้อมูลไปเรื่อยๆ จนกว่า CLKX และ CLKR จะหมดไป

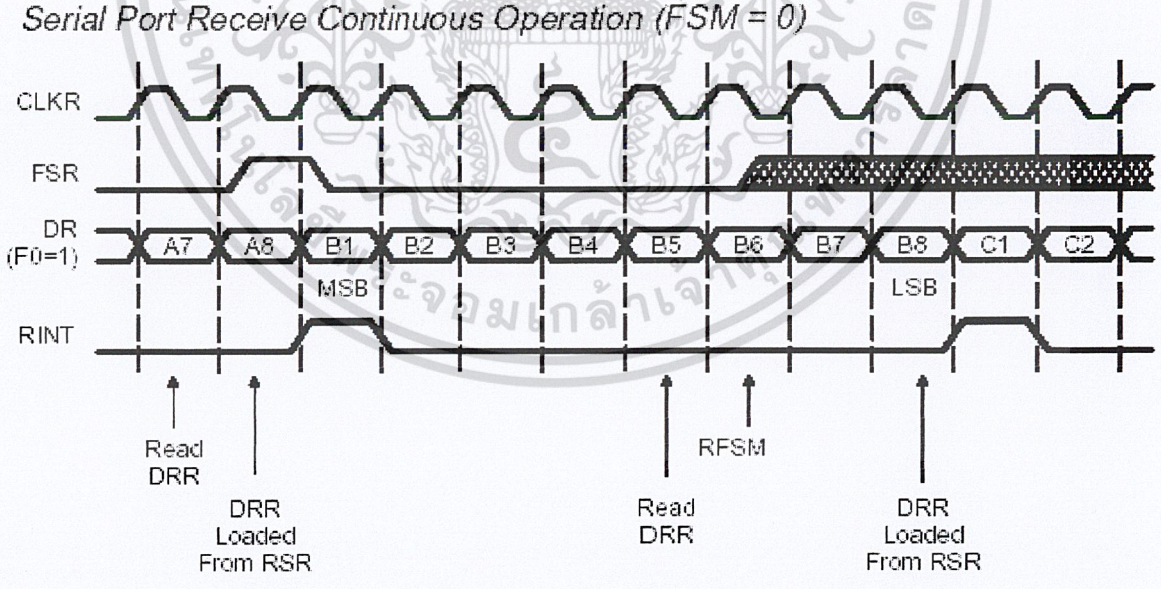
สัญญาณเฟรมซิงค์จะถูกใช้เพียงครั้งแรกของการรับส่งเท่านั้น คือจะเป็นตัวให้สัญญาณเริ่มการรับส่ง ถ้า FSX เป็นเอาท์พุท (TXM = 1) มันจะคงค่า 0 ไปจนกว่า FSM จะถูกเปลี่ยนกลับเป็น 1 หรือ DXR จะถูกใส่ค่าใหม่ ถ้า XINT เกิดแล้วแต่ยังไม่มีค่าใหม่ใส่ลงใน DXR ค่าที่อยู่ใน DXR ตัวเดิมจะถูกส่งออกไปเรื่อยๆ จนกว่าจะมีค่าใหม่ใส่ลงใน DXR สังเกตว่าจะต่างจาก 2 กรณีแรกที่หากว่าไม่มีค่าใหม่ใส่ลงไปใน DXR แล้ว DX จะมีสถานะเป็นอิมพีแดนซ์สูงไปเรื่อยๆ จนกว่าจะมีค่าใหม่มา เช่น ถ้า C ไม่ถูกใส่ลงใน DXR แล้ว B ก็ยังคงถูกส่งต่อไปเรื่อยๆจนกว่า C จะเข้ามา

สำหรับการรับค่าใน RSR จะถูกนำมาใส่ใน DRR ทุกๆ 8 บิต หรือ 16 ไซเคิล (ขึ้นอยู่กับ FO ว่าใช้การส่งแบบไหน) โดยไม่สนใจว่า DRR จะถูกอ่านหรือไม่ และถูกอ่านไปกี่ครั้งวิธีที่จะหยุดการรับส่งแบบนี้ก็คือ เปลี่ยนค่า FSM ให้เป็น 1 หรือหยุด CLKX และ CLKR เสีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป2-22 แสดงผังเวลาของการส่ง Continuous แบบไม่ใช้ Frame Syn Pulse



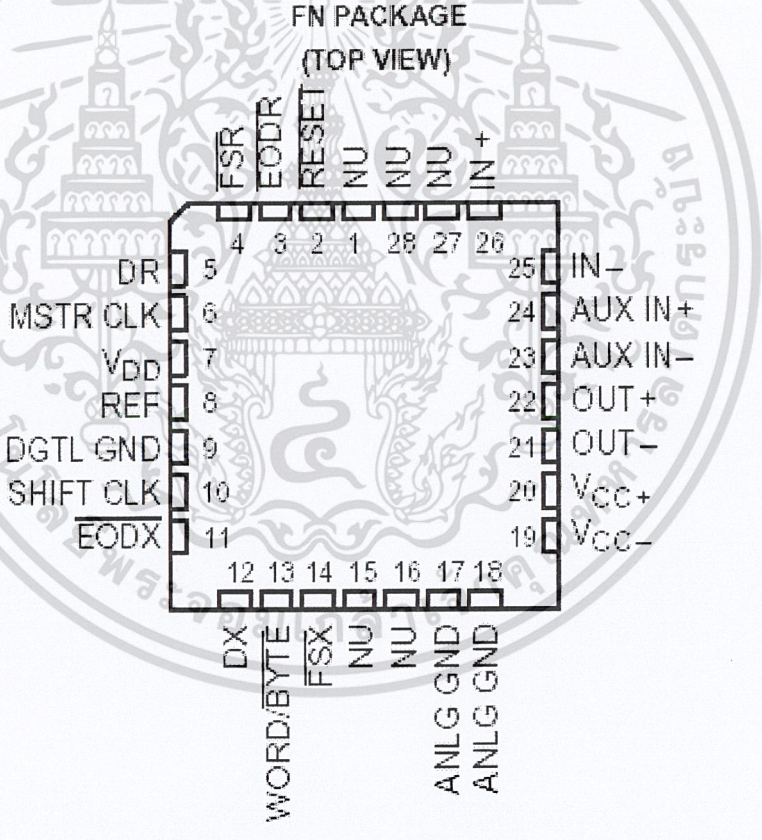
รูป2-23 แสดงผังเวลาของการรับ Continuous แบบไม่ใช้ Frame Syn Pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Analog Interface Circuit (AIC)

คุณสมบัติ

- 14-บิต Dynamic Range ADC และ DAC
- สามารถเปลี่ยนค่า Sampling Rate ของ ADC และ DAC ได้มากถึง 19,200 Sample/sec
- มี Switched-Capacitor Antiliasing input filter และ output reconstruction filter
- ติดต่อกับ TMS320C25 โดยพอร์ตอนุกรม
- สามารถควบคุมโดยอาศัยสัญญาณนาฬิกาและไม่อาศัยสัญญาณนาฬิกา
- เป็น 2's complement format
- CMOS technology

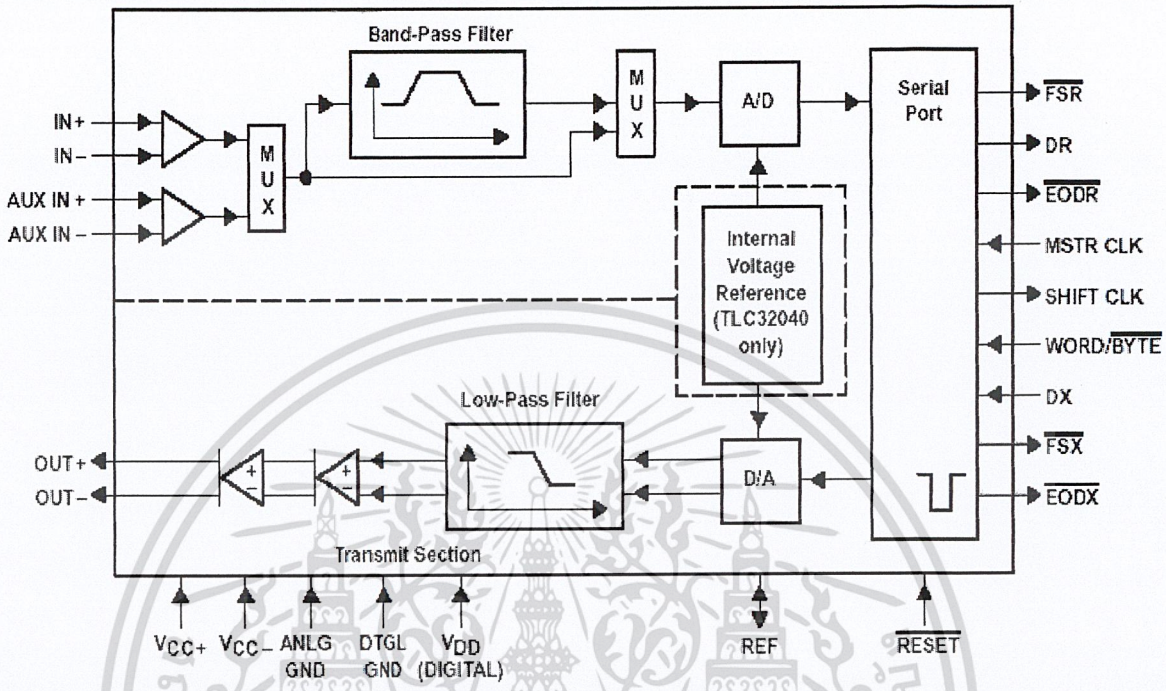


รูปที่ 2-24 แสดง Analog Interface Circuit Chip

TLC 32040 เป็นการรวมของระบบA / D และ D / A ทั้ง input และoutput แบบชิพเดี่ยว อุปกรณ์นี้รวมตัวกรองสัญญาณขาเข้าแบบช่วงความถี่ผ่าน , 14 – บิต ของค่าการแปลง A / D , ตัวประมวลผล 4 ตัวที่เข้ากันได้กับพอร์ตอนุกรม , 14 - บิตของการแปลงค่า D / A , และ ตัวกรองสัญญาณขาออกแบบความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

functional block diagram



รูปที่ 2-25 ฟังก์ชันบล็อกไดอะแกรมของ AIC

Analog Input

การเตรียมพร้อมสำหรับสัญญาณขาเข้าทั้งสองคือ โดยปกติ IN+ และ IN- จะถูกทำให้อยู่ในการใช้งาน อย่างไรก็ตาม AUX IN + และ AUX IN - สามารถใช้ได้ถ้าต้องการ แต่ละ การเซ็ตอินพุตสามารถทำงานในโหมด Differential หรือ Single-ended

A/D band pass filter, A/D band pass filter clocking, and A/D conversion timing

ตัวกรองA / D แบบช่วงความถี่ผ่านสามารถเลือกใช้หรือเลือกไม่ใช้ได้ โดยโปรแกรมปกติจะให้ความถี่สัญญาณนาฬิกาที่ 288kHz ถ้าค่ามันไม่เท่ากับ288 kHz ฟังก์ชันการส่งผ่านของตัวกรองเป็นความถี่ที่ถูกสเกลด้วยอัตราส่วนสัญญาณนาฬิกาหลัก อัตราการเปลี่ยนของ A/D ถือให้ได้ความถี่ 288 kHz Rx counter B เพื่อไม่ให้เกิดการ aliasing ที่เกิดจาก A/D

A/D converter performance specification

พื้นฐานการการเปลี่ยนคุณสมบัติวงจรเปลี่ยน A/D ใช้เทคนิคของ inherent sample-and-hold

Analog output

วงจรต่อเนื่องเอาต์พุตเป็นตัวขยายกำลังต่อเนื่องเอาต์พุตทั้งตัวขยายไม่กลับเฟสและกลับเฟสรวมอยู่ในวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D/A low – pass filter, D/A low – pass filter clocking, and D/A conversion timing

ปกติจะให้ความถี่นาฬิกาที่ 288 kHz เหมือน A/D ฟังก์ชันการส่งผ่านของตัวกรองนี้ ความถี่จะถูกสเกลเมื่อความถี่ไม่เท่ากับ 288 kHz ตัวกรองเวลาต่อเนื่องจะถูกแสดงในเอาต์พุตของตัวกรองความถี่ต่ำผ่านใน D/A

อัตราการแปลง D/A จะได้ 288 kHzด้วย TX counter B เมื่อไม่ต้องการให้เกิด aliasing เนื่องจากอัตราการเปลี่ยน D/A

D/A converter performance specification

พื้นฐานการการเปลี่ยนคุณสมบัติวงจรเปลี่ยน D/A ใช้ sample-and-hold

Explanation of internal timing configuration

$$\text{SCF Clock Frequency} = \frac{\text{Master clock frequency}}{2 * \text{Contents of Counter A}}$$

$$\text{Conversion Frequency} = \frac{\text{SCF Clock Frequency}}{\text{Contents of Counter B}}$$

$$\text{Shift Clock Frequency} = \frac{\text{Master clock frequency}}{4}$$

TX counter A และ TX counter B จะถูกโหลดอีกครั้งทุกจุดการแปลง D/A ขณะที่

RX counter A และ RX counter B จะถูกโหลดอีกครั้งทุกจุดการแปลง A/D

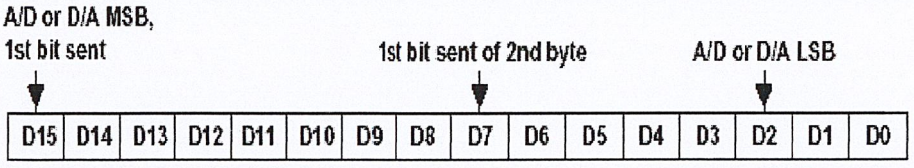
TX counter B และ RX counter B จะถูกโหลดค่าจาก TB และ RB register

TA counter A สามารถโหลดด้วย TA register, TA register – TA' register , TA register + TA' register

ส่วน RX counter A สามารถโปรแกรมควบคุมด้วย RA register , RA register – RA' register, RA register + RA' register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AIC DR or DX word bit pattern



AIC DX data word format section

d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0	COMMENTS
primary DX serial communication protocol																
←d15 (MSB) through d2 go to the D/A converter register →														0	0	The TX and RX counter As are loaded with the TA and RA register values. The TX and RX counter Bs are loaded with TB and RB register values.
←d15 (MSB) through d2 go to the D/A converter register →														0	1	The TX and RX counter As are loaded with the TA + TA' and RA + RA' register values. The TX and RX counter Bs are loaded with TB and RB register values. Bits d1 = 0 and d0 = 1 cause the next D/A and A/D conversion periods to be changed by the addition of TA' and RA' master clock cycles, in which TA' and R/A can be positive or negative or zero (refer to Table 1).
←d15 (MSB) through d2 go to the D/A converter register →														1	0	The TX and RX counter As are loaded with the TA – TA' and RA – RA' register values. The TX and RX counter Bs are loaded with TB and RB register values. Bits d1 = 1 and d0 = 0 cause the next D/A and A/D conversion periods to be changed by the subtraction of TA' and RA' master clock cycles, in which TA and R/A can be positive or negative or zero (refer to Table 1).
←d15 (MSB) through d2 go to the D/A converter register →														1	1	The TX and RX counter As are loaded with the TA and RA register values. The TX and RX counter Bs are loaded with the TB and RB register values. After a delay of four shift clock cycles, a secondary transmission immediately follows to program the AIC to operate in the desired configuration.

NOTE: Setting the two least significant bits to 1 in the normal transmission of DAC information (primary communications) to the AIC initiates secondary communications upon completion of the primary communications.

Upon completion of the primary communication. $\overline{\text{FSX}}$ remains high for four SHIFT CLK cycles and then goes low and initiates the secondary communication. The timing specifications for the primary and secondary communications are identical. In this manner, the secondary communication, if initiated, is interleaved between successive primary communications. This interleaving prevents the secondary communication from interfering with the primary communications and DAC timing, thus preventing the AIC from skipping a DAC output. In the synchronous mode, $\overline{\text{FSR}}$ is not asserted during secondary communications.

ตารางที่ 2-7 แสดงการเซ็ตบิตที่ AIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

secondary DX serial communication protocol

x x | ← to TA register → | x x | ← to RA register → |	0 0	d13 and d6 are MSBs (unsigned binary)
x | ← to TA' register → | x | ← to RA' register → |	0 1	d14 and d7 are 2's complement sign bits
x | ← to TB register → | x | ← to RB register → |	1 0	d14 and d7 are MSBs (unsigned binary)
x x x x x x x x d7 d6 d5 d4 d3 d2 ← Control register →	1 1	d2 = 0/1 deletes/inserts the bandpass filter d3 = 0/1 disables/enables the loopback function d4 = 0/1 disables/enables the AUX IN+ and AUX IN– terminals d5 = 0/1 asynchronous/synchronous transmit receive sections d6 = 0/1 gain control bits (see gain control section) d7 = 0/1 gain control bits (see gain control section)

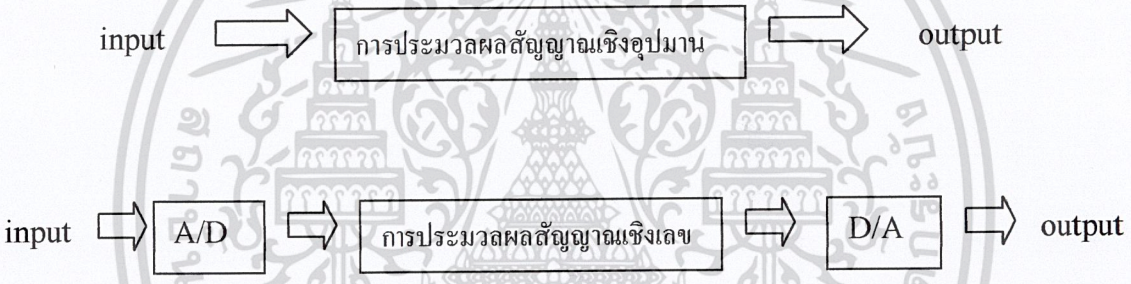
ตารางที่ 2-8 แสดงการเซ็ตบิตที่ secondary DX serial communication protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีเบื้องต้น

3.1 การประมวลสัญญาณ (Signal processing)

การประมวลผลสัญญาณคือการนำสัญญาณมาปรับปรุง หรือเปลี่ยนรูป เพื่อประโยชน์ในการสื่อสารความหมาย การตีความ และการตัดสินใจ การควบคุมการประมวลสัญญาณนั้นอาจกระทำโดยตรงต่อสัญญาณ เช่น การขยาย (หรือลดทอน) การกรอง อาจมีการเปลี่ยนรูปแบบของสัญญาณไปก่อน เช่นการประมวลในแบบเชิงเลข และการประมวลในโดเมนที่แตกต่างออกไป พึงสังเกตว่าแม้จะมีการเปลี่ยนการแทนสัญญาณไป (เช่นเปลี่ยนเชิงเลข) หรือเปลี่ยนแวดล้อมในการประมวลผลไป (เช่นกระทำในโดเมนอื่น) สัญญาณจะยังคงเป็นสัญญาณเดิมอยู่อย่างครบถ้วน



รูปที่3-1 เปรียบเทียบBlock diagramของการประมวลสัญญาณเชิงอุปมาน และการประมวลสัญญาณเชิงเลข

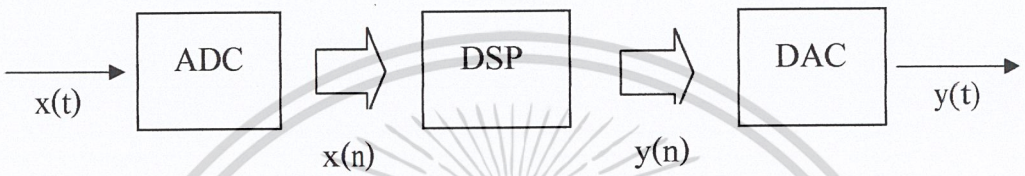
โดเมน (Domain)

โดเมนในที่นี้หมายถึง แวดล้อมในการอธิบายความหมายต่างๆ หรือก็คือตัวแปรนั้นเองหากการอธิบายหรือการทำงานนั้นกระทำภายใต้ตัวแปรของเวลา (คือฟังก์ชัน t) ก็จะเรียกว่าโดเมนเวลา หรือไทม์โดเมน (Time domain) หรือถ้าหากการทำงานนั้นกระทำภายใต้ตัวแปรของความถี่ (คือฟังก์ชัน f หรือ ω) ก็จะเรียกว่า โดเมนความถี่ หรือ ฟรีเควนซีโดเมน (Frequency domain) เป็นต้น สัญญาณเดียวกันคุณสมบัติบางอย่าอาจมองเห็นได้เด่นชัดในอีกโดเมนหนึ่งเช่น ในโดเมนเวลาเราสามารถจะเห็นการเปลี่ยนแปลงของขนาดของสัญญาณในขณะที่เวลาเปลี่ยนไปได้อย่างชัดเจน แต่จะจะบอกได้ยากว่าสัญญาณนั้นมีความถี่อะไรบ้างเป็นองค์ประกอบ แต่ในโดเมนความถี่เราสมารถที่ระบุได้ว่าสัญญาณนั้นประกอบด้วยสัญญาณความถี่อะไรบ้างแต่ละความถี่นั้นมีพลังงานเท่าไร การเปลี่ยนรูปไปสู่โดเมนอื่นนั้นจะมีขั้นตอนสำคัญคือการทรานสฟอร์ม (Transform) ซึ่งมีหลายรูปแบบ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1โครงสร้างของระบบประมวลผลสัญญาณเชิงเลข (Structure of Digital Signal Processing)

เนื่องจากธรรมชาติของสัญญาณโดยทั่วไป เป็นการเปลี่ยนแปลงของปริมาณกายภาพที่มีความต่อเนื่อง การประมวลสัญญาณแบบเชิงเลขจึงต้องอาศัยส่วนเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter , ADC)ก่อนแล้วจึงทำการประมวลเมื่อประมวลผลแล้วอาศัยส่วนเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital to Analog Converter , DAC) อีกทีหนึ่ง



รูปที่ 3-2 ระบบการประมวลผลสัญญาณเชิงเลข

1. ตัวประมวลผลแบบเชิงเลข

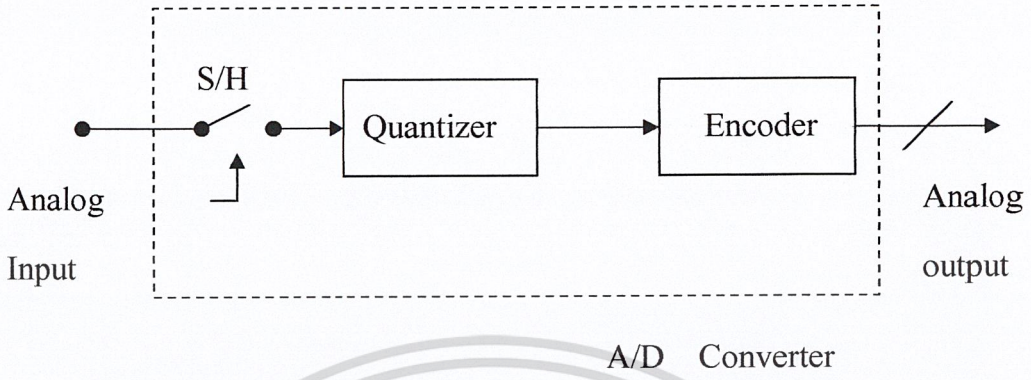
การประมวลผลสัญญาณในระบบเชิงเลข สามารถที่จะดำเนินการได้ทั้งทางฮาร์ดแวร์ และซอฟ์ฟแวร์ หากประมวลโดยฮาร์ดแวร์ ตัวประมวลก็อาจมีได้หลายระดับ ขึ้นอยู่กับ

1. วงจรพื้นฐาน เช่น เกท และสวิทช์
2. ส่วนประกอบของวงจรพื้นฐานเช่นรีจิสเตอร์ และวงจรบวกเป็นต้น
3. วงจรที่ซับซ้อนมากขึ้นเช่นหน่วยความจำ
4. วงจรที่ซับซ้อนมากเช่นวงจรคูณ และ ส่วนทำงานคณิตศาสตร์ (ALU)
5. วงจรควบคุมอื่นๆเช่นสัญญาณนาฬิกา วงจรอินเตอร์เฟส วงจรแปลงอนาลอกและดิจิตอล
6. ตัวประมวลผลเฉพาะสำหรับการประมวลผลเชิงเลข(DSP Processor)

2. ส่วนเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณเชิงเลข

สัญญาณโดยทั่วไปเป็นสัญญาณที่ต่อเนื่องหรือก็คือสัญญาณอนาลอก หรือสัญญาณเชิงอุปมานนั้นเองการประมวลผลเชิงเลขจึงต้องอาศัยหลักการเปลี่ยนจากสัญญาณอนาลอกเป็นสัญญาณเชิงเลข(Analog to Digital converter, ADC)วงจรจะรับสัญญาณอนาลอกมาทางอินพุทและเปลี่ยนให้อยู่ในรูปของบิตรวมกัน(คำ : word)จำนวนบิตในหนึ่งคำจะบอกถึงความละเอียดที่ระบุถึงขนาดของสัญญาณอนาลอก ส่วนการเปลี่ยนจากสัญญาณอนาลอกเป็นสัญญาณเชิงเลขจะประกอบด้วยส่วนหลัก 3 ส่วน คือส่วนสุ่มตัวอย่างสัญญาณ (Sample & Hold, S/H) ส่วนจัดลำดับสัญญาณ (Quantiser, Q) และส่วนเข้ารหัสสัญญาณ (Encoder) ดังรูปที่ 1-3

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 ส่วนเปลี่ยนสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข

3.1.2 สัญญาณและระบบ

สัญญาณและประเภทของสัญญาณ (Signal & Classification of Signal)

สัญญาณก็คือฟังก์ชั่นที่ใช้แทนปริมานทางกายภาพ ซึ่งโดยทางคณิตศาสตร์แล้วสัญญาณมักแทนด้วยฟังก์ชั่นที่มีตัวแปรอิสระคือ t ซึ่งในความหมายของ t คือเวลาดังนั้นสัญญาณจึงทักเขียนแทนโดย x(t) ในทางวิศวกรรมไฟฟ้ามักแทนสัญญาณด้วยแรงดันหรือกระแส

1. สัญญาณที่มีความต่อเนื่องและสัญญาณเต็มหน่วย

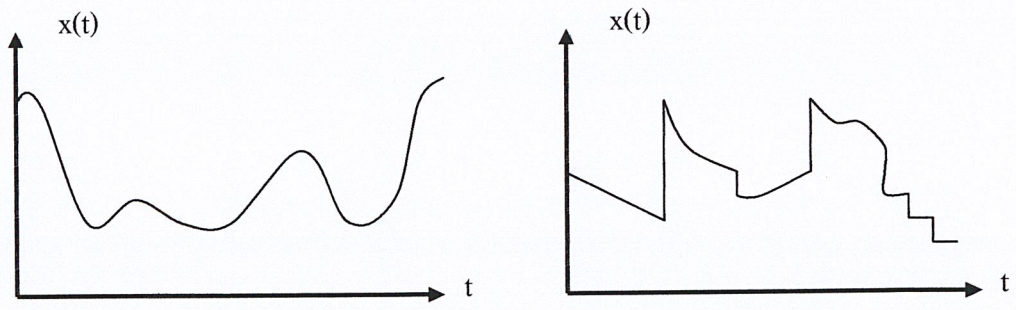
สัญญาณ x(t) จะถือว่าเป็นสัญญาณต่อเนื่อง (Continuous-time Signal) เมื่อ t เป็นตัวแปรที่มีความต่อเนื่อง ถ้า t เป็นตัวแปรที่ไม่ต่อเนื่องหรือตัวแปรขาดช่วง x(t) ก็จะเป็นสัญญาณขาดช่วง (Discontinuous-timer Signal) เมื่อสัญญาณขาดช่วงและนิยามขนาดในเวลาที่ขาดช่วงนั้นเป็นขนาดเต็มหน่วย ก็จะเรียกว่า สัญญาณเต็มหน่วย (Discrete-timer Signal)ในบางครั้งสัญญาณดังกล่าวจึงมักนิยามในลักษณะชุดแถว(Sequence) ของตัวเลขซึ่งเขียนโดย {x(n)} หรือ x[n] เมื่อ n คือเลขจำนวนเต็ม

หากสัญญาณเป็นสัญญาณต่อเนื่องจะได้ว่า

$$x(t)\Big|_{t=t_0^-} \qquad x(t)\Big|_{t=t_0} \qquad x(t)\Big|_{t=t_0^+}$$

สัญญาณต่อเนื่องจะมีอนุพันธ์ที่เป็นสัญญาณต่อเนื่องเช่นกัน สัญญาณที่ไม่เข้าเงื่อนไขดังกล่าวนี้ถือว่าไม่เป็นสัญญาณต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



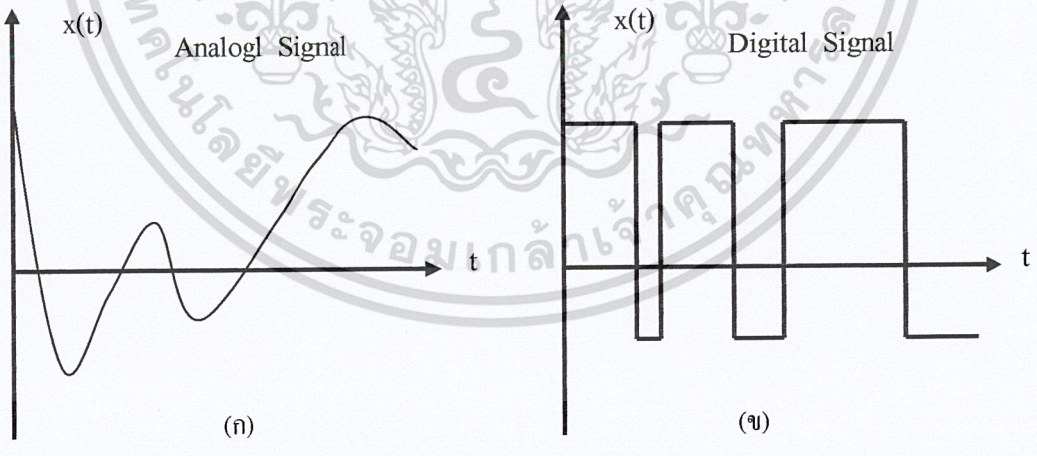
(ก) (ข)

รูปที่ 3-4 แสดง(ก) สัญญาณที่ต่อเนื่อง (ข) สัญญาณไม่ต่อเนื่อง

2. สัญญาณเชิงอุปมานและสัญญาณเชิงเลข

หากสัญญาณต่อเนื่อง x(t) มีค่าอยู่ในช่วงที่มีความยาวต่อเนื่อง (a,b) ซึ่งaอาจมีค่าลบอนันต์ และbอาจมีค่าบวกอนันต์ สัญญาณ x(t) นี้ก็จะเรียกว่าสัญญาณเชิงอุปมาน(Analog Signal) ถ้าหากสัญญาณขาดช่วง x[n] มีค่าได้เพียงค่าจำกัดที่แตกต่างกัน สัญญาณนั้นก็จะเรียกว่าสัญญาณเชิงเลข (Digital Signal)

เนื่องจากสัญญาณขาดช่วงx[n]มักจะได้จากการสุ่มสัญญาณต่อเนื่องx(t)จึงได้ $x[n] = x[nT_s]$ เมื่อ T_sคือช่วงเวลาการสุ่ม(Sampling Interval)



(ก) (ข)

รูปที่ 3-5 แสดง(ก)สัญญาณเชิงอุปมาน (ข)สัญญาณเชิงเลข

3. สัญญาณจริงและสัญญาณเชิงซ้อน

สัญญาณ x(t) เป็นสัญญาณจริง เมื่อค่าของสัญญาณนั้นเป็นเลขจำนวนจริง และจะเป็นสัญญาณเชิงซ้อน ความจริงแล้วสัญญาณโดยทั่วไปเป็นสัญญาณจริงเสมอแต่เพื่อสะดวกในการวิเคราะห์จึงมักพิจารณาเป็นสัญญาณเชิงซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัญญาณที่ระบุได้และสัญญาณที่ระบุไม่ได้

สัญญาณที่ระบุได้ (Deterministic Signal) ก็คือสัญญาณที่สามารถที่จะระบุค่าได้อย่างแน่นอนไม่ว่าจะเป็นเวลาใดๆส่วนสัญญาณที่ระบุไม่ได้(Random Signal)ก็คือสัญญาณที่ไม่สามารถระบุค่าได้อย่างชัดเจน แต่พอที่จะแยกแยะได้โดยวิธีทางสถิติ

5. สัญญาณที่มีคาบเวลา และสัญญาณที่ไม่มีคาบเวลา

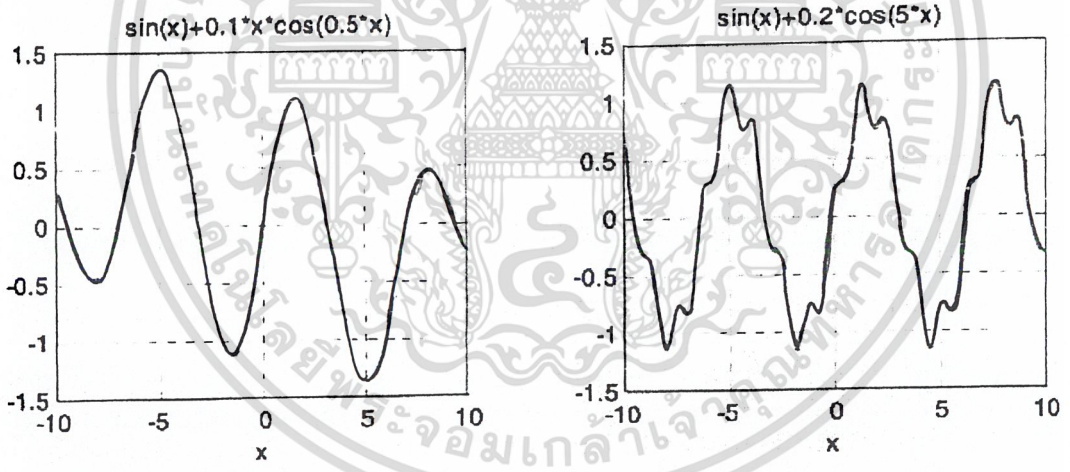
สัญญาณ x(t) จะเป็นสัญญาณที่มีคาบเวลาถ้าหากมีเลขจำนวนบวก T_0ที่ทำให้

$$x(t + nT_0) = x(t)$$

โดยที่ n เป็นเลขจำนวนเต็มบวก ค่าบวกที่น้อยที่สุดของ T_0 เรียกว่าคาบเวลา (Period) และ $1/T_0$ เรียกว่าความถี่มูลฐาน(Fundamental Frequency , f_0)

$$f_0 = \frac{1}{T_0} \quad \text{เฮิร์ท}$$

สัญญาณอื่นใดที่ไม่มีค่าT_0 ที่สอดคล้องกับสมการข้างบน x(t) ก็ถือว่าสัญญาณนั้นเป็นสัญญาณที่ไม่มีคาบเวลา (Non Periodic or A periodic Signal)



(ก) (ข)

รูปที่ 3-6 แสดง(ก)สัญญาณที่ไม่มีคาบเวลา (ข) สัญญาณที่มีคาบเวลา

6.สัญญาณพลังงานและสัญญาณกำลังงาน

ค่าสาระพลังงานนอมัลไลซ์(Normalized Energy Content : E) ของสัญญาณจะหาได้จาก

$$E = \int_{-\infty}^{+\infty} |x(t)|^2 \, dt$$

ค่ากำลังงานเฉลี่ยนอร์มัลไลซ์(Normalized Average Power : P) ของสัญญาณหาได้จาก

$$P = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |x(t)|^2 \, dt$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่หาค่าพลังงานได้หรือ0 < P < ∞ เรียกสัญญาณนั้นว่าสัญญาณพลังงาน(Energy Signal) และที่หาค่ากำลังงานได้หรือ 0 < P < ∞ เรียกสัญญาณนั้นว่าสัญญาณกำลังงาน(Power Signal)

ข้อสังเกต 1

สัญญาณเชิงซ้อนอาจแสดงได้ด้วย $x(t) = Ae^{j(\omega t+\phi)}$ ซึ่ง

$$x(t) = Ae^{j(\omega t+\phi)} = Ae^{j\phi} \cdot e^{j\omega t}$$

เพราะว่า $e^x = \sum_{n=0}^{n=\infty} \frac{x^n}{n!}$

$$Ae^{(j\omega t+\phi)} = A\left(1 + \frac{j(\omega t+\phi)}{1} + \frac{j^2(\omega t+\phi)^2}{2} + \frac{j^3(\omega t+\phi)^3}{3} + \ldots\right)$$

$$= A\cos(\omega t+\phi) + jA\sin(\omega t+\phi)$$

และ $$|x(t)| = A\sqrt{\cos^2(\omega t+\phi) + \sin^2(\omega t+\phi)} = A$$

ข้อสังเกต 2

สัญญาณที่มีคาบเวลาสามารถที่จะหาค่าเฉลี่ย และค่าอาร์เอ็มเอสได้จาก

$$x_{avg} = \frac{1}{T}\int_0^T x(t)dt$$

และ

$$x_{rms} = \sqrt{\frac{1}{T}\int_0^T |x(t)|^2 dt}$$

ข้อสังเกต 3

สัญญาณที่มีคาบเวลาใดสามารถที่ที่จะเขียนอนุกรมฟูริเยร์ได้

ข้อสังเกต 4

สัญญาณใดๆจะถือเป็นสัญญาณมีเหตุหรือถูกสร้างขึ้นมาเมื่อสัญญาณปรากฏที่เวลาเท่ากับหรือมากกว่าศูนย์หาก y(t) เป็น causal signal

$$y(t) = \begin{cases} x(t) & for \quad t \geq 0 \\ 0 & for \quad t < 0 \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

สัญญาณคลื่นรูปซายน์ $x(t) = \sin(\omega t)$ ถือว่ามีปรากฏอยู่ตลอด ถือว่าเป็น non-causal signal

ฟังก์ชันเดลต้าหรือดิเร็กพัลส์

ฟังก์ชันเดลต้า หรือดิเร็กพัลส์นี้เป็นสิ่งที่มีอยู่ในเชิงทฤษฎีไม่สามารถที่จะปรากฏเป็นสัญญาณจริงได้ แต่จะมีประโยชน์มากในการอธิบายสิ่งอื่นๆมีความจำกัดความที่เข้าใจได้ง่ายดังนี้ พิจารณา x(t) โดย

$$x(t)=f(t\varepsilon)=\begin{cases}\dfrac{1}{2\varepsilon} & for \quad |t|<\varepsilon \\ 0 & for \quad |t|>\varepsilon\end{cases}$$ ซึ่งจะได้ $F_\varepsilon(\omega)=\dfrac{\sin(\omega\varepsilon)}{\omega\varepsilon}$

โดยที่ขนาดพื้นที่เท่ากับ 1 เมื่อขนาดของ ε เล็กลงความสูงจะสูงขึ้นเมื่อ $\varepsilon \geq 0$ จะได้ว่าฟังก์ชันเดลต้าก็คือ $\delta(t)=\lim_{\varepsilon\to 0} f_\varepsilon(t)$ ซึ่งก็เป็นการนำมาถึงการอธิบายฟังก์ชันเดลต้าด้วยคำอธิบายอื่น เช่น

$$\delta(t)=\begin{cases}1; & t=0 \\ 0; & t \geq\neq 0\end{cases} \quad \text{หรือ} \quad \int_{-\infty}^{+\infty}\delta(t)dt=1$$

หรือเมื่อ x(t) คือฟังก์ชันใดๆ $\int_{-\infty}^{+\infty}x(t)\delta(t)dt=x(0)$

หากทำการแปลงฟูริเยร์ของ $\delta(t)$ จะพบว่า

$$\lim_{\varepsilon\to 0}F_\varepsilon(\omega)=\lim_{\varepsilon\to 0}\frac{\sin(\omega\varepsilon)}{\omega\varepsilon}=1 \quad \text{หรือ} \quad \delta(t)\Leftrightarrow 1$$

หรือ

$$\Im\{\delta(t)\}=\int_{-\infty}^{+\infty}\delta(t)e^{-j\omega t}dt=1$$

คุณสมบัติที่สำคัญฟังก์ชันเดลต้าสรุปได้ดังนี้

1. $\delta(t-t_0)$ หรือ $\delta(t_0-t)$ เป็นการระบุการปรากฏของเดลต้าที่ $t=t_0$
2. $\int_{-\infty}^{+\infty}x(t)\delta(t-t_0)=\int_{-\infty}^{+\infty}x(t)\delta(t_0-t)=x(t_0)$ เป็นการแทนฟังก์ชันที่ $t=t_0$
3. $x(t)\delta(t-t_0)=x(t_0)\delta(t-t_0)$
4. $x(t)*\delta(t-t_0)=x(t-t_0)$ เครื่องหมาย * คือคอนโวลูชันหรือการคูณประสาน

3.1.3 ระบบและการแบ่งระบบ (System and System Classification)

ระบบ คือแบบจำลองของกระบวนการทางกายภาพที่เชื่อมโยงสัญญาณเข้า(แหล่งจ่ายหรือสัญญาณเข้า) ไปยังสัญญาณออก(หรือสัญญาณที่ตอบสนอง) เมื่อให้x(t) และ y(t) เป็นสัญญาณเข้าและสัญญาณออกจากระบบดังนั้นระบบก็สามารถที่จะมองได้เป็นการจับคู่จาก x(t) ไปยัง y(t) โดยอาจเขียนได้คือ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(t)=\Re[f(t)]$$

1. ระบบที่ต่อเนื่องทางเวลาและระบบที่ไม่ต่อเนื่องทางเวลา

หากสัญญาณเข้าและสัญญาณออก x(t) และ y(t) เป็นสัญญาณที่มีความต่อเนื่องทางเวลาแล้ว ระบบนั้นถือว่าเป็นระบบที่ต่อเนื่องทางเวลา (Continuous – time system) แต่ถ้าหาก x(t) และ y(t) เป็นสัญญาณที่ไม่ต่อเนื่องทางเวลา(Discrete – time system)

1.1 ระบบเชิงเส้น

โดยมีคุณสมบัติที่สำคัญคือ

ก. การบวกกันของสัญญาณ $x_1(t)$ และสัญญาณ $x_2(t)$

$$[x_1(t) + x_2(t)] = [x_1(t)] + [x_2(t)] = y_1(t) + y_2(t)$$

ข. ความเป็นเอกพันธ์

$$[ax(t)] = a[x(t)] = ay(t)$$

สำหรับสัญญาณเข้า x(t) ทุกรูปแบบและสเกล a

1.2 ระบบเวลาไม่เปลี่ยนแปร

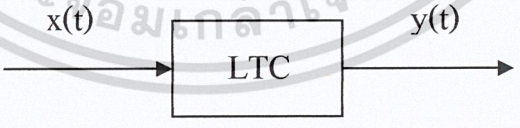
ระบบที่สอดคล้องตามเงื่อนไขต่อไปนี้จะเรียกว่าเป็นระบบที่ไม่เปลี่ยนแปรตามเวลาหรือระบบคงที่ (time-invariant , fixed system)

$$[x(t-t_0) = y(t-t_0)]$$

เมื่อ t_0 เป็นตัวคงที่ที่มีค่าจริงใดๆ

1.3 ระบบเชิงเส้นเวลาไม่เปลี่ยนแปร

พิจารณาระบบเวลาต่อเนื่อง ซึ่งมีอินพุตเป็น x(t) และให้เอ้าท์พุตเป็น y(t) ระบบดังกล่าวจะเป็นระบบเชิงเส้นเวลาไม่เปลี่ยนแปร (เวลายืนยง) เมื่อมีคุณสมบัติดังนี้



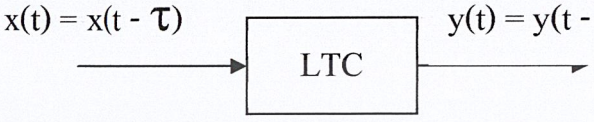
- คุณสมบัติเชิงเส้น (Linearity)

ระบบจะมีคุณสมบัติเชิงเส้นเมื่อสัญญาณอินพุต $x(t) = ax_1(t) + bx_2(t)$ สำหรับค่าคงที่ a และ b ใดๆ ทำให้ได้เอ้าท์พุต $y(t) = ay_1(t) + by_2(t)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คุณสมบัติเวลาไม่เปลี่ยนแปร (Time invariance)

ระบบจะมีคุณสมบัติเวลาไม่เปลี่ยนแปร (เวลายืนยง) เมื่อสัญญาณอินพุต x(t-τ) สำหรับเวลาหน่วงτ ใดๆทำให้ได้เอ้าท์พุต y(t-τ)



- คุณสมบัติการเสถียร (Stability)

ระบบจะมีเสถียรภาพเมื่อสัญญาณอินพุต x(t) ที่ระบุขนาดได้ $|x(t)|_{max} \leq A$ ทำให้ได้เอ้าท์พุตที่สอดคล้องกับที่ระบุขนาดได้ $|y(t)|_{max} \leq B$

- คุณสมบัติการมีเหตุ (Causality)

ระบบจะถือว่าเป็นระบบมีเหตุเมื่อสัญญาณอินพุต x(t) ที่ $t = t_0$ ทำให้ได้เอ้าท์พุตที่สอดคล้องกันที่ t หลังจาก t_0

1.4 การตอบสนองความถี่ (Frequency Response)

เมื่อ h(t) คือการตอบสนองต่ออิมพัลส์ของระบบผลการแปลงฟูริเยร์ของ h(t) จะทำให้ได้ $H(\omega)$ กล่าวคือ

$$H(\omega) = \int_{-\infty}^{+\infty} h(t) e^{-j\omega t} dt$$

$H(\omega)$ จะอธิบายถึงพฤติกรรมของระบบซึ่งก็คือการตอบสนองความถี่นั่นเองในโดเมนเวลาเราได้

$$y(t) = x(t) * h(t) = h(t) * x(t)$$

หาก $y(t) \leftrightarrow Y(\omega), x(t) \leftrightarrow X(\omega)$ และ $h(t) \leftrightarrow H(\omega)$ แล้วเราสามารถพิสูจน์ได้ว่าในโดเมนความถี่

$$Y(\omega) = X(\omega) \cdot H(\omega) = H(\omega) \cdot X(\omega)$$

หรือการคูณประสานในโดเมนเวลามีค่าสอดคล้องกับการคูณในโดเมนความถี่ หรือในทางกลับกันการคอนโวลูชั่นในโดเมนความถี่มีค่าสอดคล้องกับการคูณในโดเมนเวลา

$$z(t) = x(t) * h(t) \longleftrightarrow Z(\omega) = \frac{1}{2\pi} X(\omega) * H(\omega)$$

1.5 การสุ่มสัญญาณต่อเนื่อง (Sampling of Continuous Signal)

ในทางปฏิบัติเรามักมีสัญญาณต่อเนื่องทางเวลาเสมอแต่เมื่อต้องการประมวลสัญญาณในเชิงเวลาเต็มหน่วย สัญญาณต่อเนื่องดังกล่าวจะถูกเปลี่ยนรูปให้เป็นสัญญาณเวลาเต็มหน่วยที่สอดคล้องกัน

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น

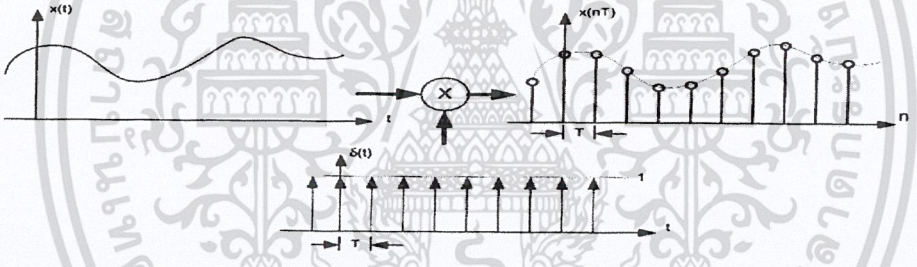
วิธีการก็คือการแทนสัญญาณต่อเนื่องเป็นช่วงๆห่างกันเวลา T_s คงที่(คือการสุ่มตัวอย่าง) ถ้าหาก T_s มีค่าที่เหมาะสมแล้วการแทนค่าดังกล่าวจะยังคงความถูกต้องเอาไว้ดังนั้นผลการสุ่มก็คือการคูณสัญญาณต่อเนื่องด้วยสัญญาณอิมพัลส์ที่เลื่อนไปเรื่อยๆแต่ละครั้งก็ห่างกันเป็นเวลา $T = T_s = \frac{1}{f_s}$

$$y(t) = x(nT) = x(t) \cdot \sum_{\tau=-\infty}^{+\infty} \delta(t - nT)$$

$$= \sum_{\tau=-\infty}^{+\infty} x(t)\delta(t - nT)$$

$$= \sum_{\tau=-\infty}^{+\infty} x(nT)\delta(t - nT)$$

สังเกตว่า x(t) ซึ่งเป็นสัญญาณต่อเนื่องจะแทนด้วยชุดแถวของตัวแทนที่นับด้วยค่าของเลขจำนวนเต็ม n คือ x(nT) โดยแต่ละตัวอย่างนั้นห่างกันด้วยเวลา T



รูปที่ 3-7 แสดงการสุ่มสัญญาณ $\delta(t) \leftrightarrow \delta(\omega); x(t) \leftrightarrow X(\omega)$

กระบวนการนี้อาจแสดงได้ในโดเมนความถี่ หาก $\delta(t) \leftrightarrow \delta(\omega); x(t) \leftrightarrow X(\omega)$ การคูณในโดเมนเวลาก็คือการคูณประสานในโดเมนความถี่

$$\sum_{n=-\infty}^{+\infty} \delta(t - nT) \quad \Longleftrightarrow \quad \frac{2\pi}{T} \sum_{n=-\infty}^{+\infty} \delta\left(\omega - \frac{2n\pi}{T}\right)$$ ดังนั้น

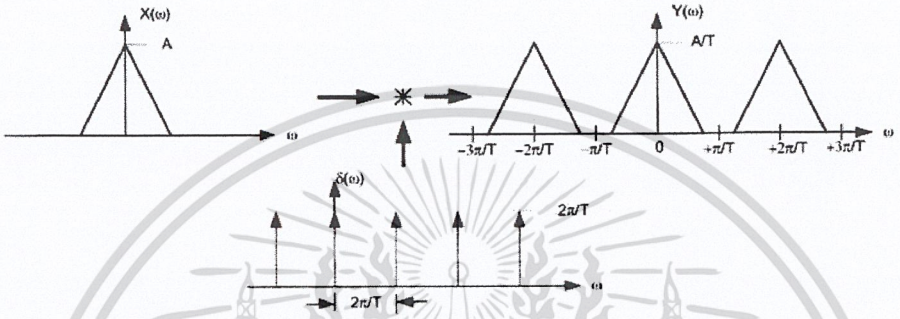
$$Y(\omega) = \frac{1}{2\pi} X(\omega) * \frac{2\pi}{T} \sum_{n=-\infty}^{+\infty} \delta\left(\omega - \frac{2n\pi}{T}\right)$$

$$= \frac{1}{T} \sum_{n=-\infty}^{+\infty} X(\omega) * \delta\left(\omega - \frac{2n\pi}{T}\right)$$

$$= \frac{1}{T} \sum_{n=-\infty}^{+\infty} X\left(\omega - \frac{2n\pi}{T}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงว่าขนาดของ $X(\omega)$ จะถูกคูณด้วย $\frac{1}{T}$ และจะกระจายหรือเลื่อนไปห่างกันออกไปด้วยความห่างช่วงละ $\frac{2\pi}{T}$ ดังนั้นการป้องกันไม่ให้เกิดการซ้อนทับกันของ $X(\omega)$ หรือเกิด Aliasing Effect ขึ้น $X(\omega)$ จะต้องมีช่วงความถี่อยู่ระหว่าง $-\pi/T$ ถึง $+\pi/T$ หรือ $X(\omega) = 0$ เมื่อ $|\omega| > \pi / T$ หรือ $2\pi f_a < \pi f_s$ เมื่อ ω หรือ f_a เป็นความถี่ของสัญญาณต่อเนื่อง



รูปที่ 3-8 แสดงการสุ่มสัญญาณอธิบายในโดเมนความถี่

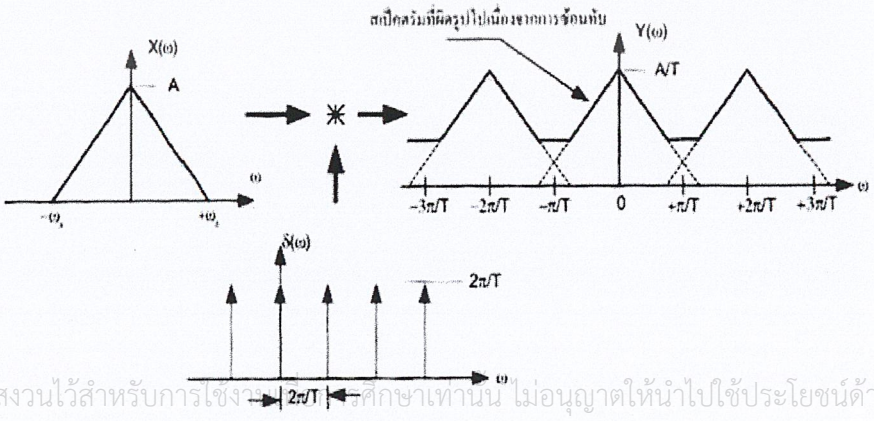
1.6 ทฤษฎีการสุ่มสัญญาณ ถ้าหากสัญญาณต่อเนื่อง $x_a(t)$ ที่มีความถี่ไม่เกิน $\omega_{\max} = 2\pi f_{amxa}$ ข้อมูลของสัญญาณต่อเนื่องนั้นสามารถที่จะอธิบายได้ด้วย x(nT) เมื่อ 1/T = f_s ถ้าหาก $f_s > 2f_{a\max}$

1.7 การเกิด Alias

พิจารณาเมื่อความถี่ในการสุ่มสัญญาณน้อยกว่า 2 เท่าของความถี่ของสัญญาณที่ถูกสุ่มหรือ $2f_a > f_s$ หรือ $|\omega_a| > \pi / T$ ผลจากการสุ่มสัญญาณในโดเมนความถี่จะได้

$$Y(\omega) = \frac{1}{T}\sum_{n=-\infty}^{+\infty} X\left(\omega_a - \frac{2\pi}{T}\right)$$

หาก $|\omega_a| > \pi / T$ จะเห็นได้ว่ามีการซ้อนทับกันของสเป็กตรัมเกิดขึ้น การบวกกันในบริเวณที่ซ้อนทับกันทำให้ค่าสเป็กตรัมผิดไปจากค่าที่ควรเป็นเมื่อสร้างคืนสัญญาณก็จะได้ค่าที่ไม่ถูกต้อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-9 แสดงการเกิด Aliasing Effect ในโดเมนความถี่

1.8 การสร้างคืนสัญญาณ (Signal Reconstruction)

การสร้างคืน x(t) จาก y(t) = x(nT) นั้นสามารถที่จะกระทำได้อย่างถูกต้องหากการสุ่มเป็นไปตามทฤษฎีการสุ่มสัญญาณ เมื่อผ่านสัญญาณ x(nT) เข้าไปยังวงจรกรองความถี่ต่ำก็จะได้ x(t) การกระทำดังกล่าวอธิบายได้ทั้งในโดเมนความถี่และโดเมนเวลา ผลการสุ่มทำให้สเปกตรัมในขอบเขต $\pm\omega_a$ กระจายออกด้วยความห่างช่วงความถี่ $2\pi/T$ ในการนำกลับคืนเราต้องการเฉพาะ $X(\omega)$ ช่วง $\pm\omega_a$ เท่านั้น ดังนั้นนอกช่วงดังกล่าวจะต้องคูณด้วยศูนย์และคูณด้วย T ในช่วง $\pm\omega_a$ หรือ

$$X(\omega) = Y(\omega) \times H(\omega)$$

วงจรกรองความถี่ต่ำมีคุณสมบัติการตอบสนองความถี่คือ

$$H(\omega) = \begin{cases} T & |\omega| < \omega_a \\ 0 & elsewhere \end{cases}$$

ดังนั้นเมื่อแปลงกลับฟูริเยร์ก็จะได้

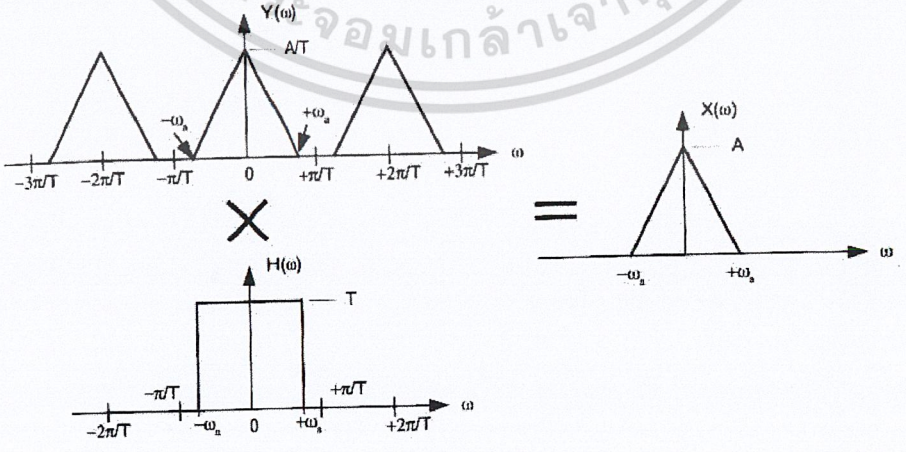
$$h(t) = \Im\{H(\omega)\} = \frac{\sin(\pi t/T)}{\pi t/T}$$

การคูณในโดเมนเวลาก็คือการคูณประสานในโดเมนเวลา

$$x(t) = y(t) * h(t) = h(t) * y(t)$$

$$= \frac{\sin(\pi t/T)}{\pi t/T} * \sum_{n=-\infty}^{+\infty} x(nT)\delta(t-nT)$$

$$= \sum_{n=-\infty}^{+\infty} x(nT)\left\{\delta(t-nT) * \frac{\sin(\pi t/T)}{\pi t/T}\right\}$$



รูปที่ 3-10 แสดงการสร้างคืนสัญญาณที่ถูกสุ่มอธิบายในโดเมนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

โดยอาศัยคุณสมบัติที่ว่า $x(t) * \delta(t - t_0) = x(t - t_0)$ จึงจะได้

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT) \frac{\sin\{\pi(t - nT)/T\}}{\pi(t - nT)/T}$$

$$= \sum_{n=-\infty}^{+\infty} x(nT) \sin c\left(\frac{\pi(t - nT)}{T}\right)$$

ก็คือการรวมกันของสัญญาณ $\frac{\sin x}{x}$ ที่ให้น้ำหนักโดย x(nT) และการเลื่อนไปทีละเวลา T ซึ่งวิธีการนี้จะเรียกว่าการอินเตอร์โปเลชัน (Interpolation)

1.9 การคูณประสาน (Convolution)

การคูณประสานสองสัญญาณคือสัญญาณ $x_1(t)$ กับสัญญาณ $x_2(t)$ แสดงได้โดย $x_1(t) * x_2(t)$ จะทำให้ได้สัญญาณใหม่ x(t) ที่นิยามโดย

$$x(t) = x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(\tau) x_2(t - \tau) d\tau$$

คุณสมบัติของการคูณประสาน

$$x_1(t) * x_2(t) = x_2(t) * x_1(t)$$
$$[x_1(t) * x_2(t)] * x_3(t) = x_1(t) * [x_2(t) * x_3(t)]$$
$$x_1(t) * [x_2(t) + x_3(t)] = x_1(t) * x_2(t) + x_1(t) * x_3(t)$$

การคูณประสานกับฟังก์ชันอิมพัลส์

$$x(t) * \delta(t) = x(t)$$

ทฤษฎีการคูณประสาน $x_1(t) \leftrightarrow X_1(\omega)$ และ $x_2(t) \leftrightarrow X_2(\omega)$
จะได้

$$x_1(t) * x_2(t) \leftrightarrow X_1(\omega) * X_2(\omega)$$
$$x_1(t) x_2(t) \leftrightarrow \frac{1}{2\pi} X_1(\omega) * X_2(t)$$

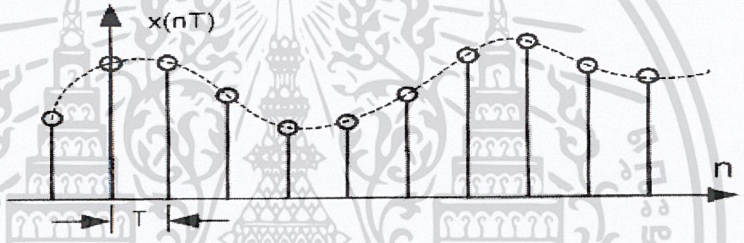
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบเวลาเต็มหน่วย (Discrete Time System)

สัญญาณเวลาเต็มหน่วยเป็นสัญญาณที่ได้จากการสุ่มสัญญาณต่อเนื่องด้วยความถี่ f_s ซึ่ง $f_s > 2f_a$ ดังนั้นสัญญาณที่สุ่มนี่จึงมีตัวอย่างที่เรียงกันเรียกว่าชุดแถว การสุ่มสัญญาณที่มีคาบเวลาก็จะได้ชุดแถวที่มีคาบเวลาด้วย

2.1 ชุดแถว (Sequence)

ชุดแถวก็คือชุดของสัญญาณเต็มหน่วยที่ได้จากการสุ่มสัญญาณต่อเนื่อง ดังนั้นแต่ละสัญญาณจะเว้นระยะกัน T_s เมื่อ $T_s = \frac{1}{f_s}$ โดยที่ f_s คือความถี่ที่ใช้สุ่ม ค่าของสัญญาณ ณ เวลาต่างๆจะแสดงโดย $x(nT_s) = x(t)\,|_{t=nT_s}$ อย่างไรก็ตามในระบบสัญญาณเต็มหน่วยระยะเวลา T_s จะไม่มาพิจารณาแต่จะพิจารณาลำดับ n แทน โดยเรียกชุดแถวว่า x[n]



รูปที่ 3-11 ชุดแถวที่ได้จากการสุ่มสัญญาณต่อเนื่อง

2.2 ชุดแถวพื้นฐาน

1. ตัวอย่างเดี่ยวหรือชุดแถวอิมพัลส์

$$x(n) = \delta(n) = \begin{cases} 1 & , n = 0 \\ 0 & , n \neq 0 \end{cases}$$

หรือ

$$\{x(n)\} = \{\ldots\ldots\underset{\uparrow}{1},1,1,1,1,1,1,1,\ldots\ldots\}$$

2. ชุดแถวค่าคงที่

$$x(n) = A \qquad ;-\infty \leq n \leq +\infty$$

หรือ

$$\{x(n)\} = \{\ldots\ldots\underset{\uparrow}{A},A,A,A,A,A,A,A,\ldots\ldots\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ชุดแถวขั้นบันไดหนึ่งหน่วย

$$x(n)=U(n)=\begin{cases}1 & ,0\le n\le+\infty\\ 0 & ,n<0\end{cases}$$

หรือ

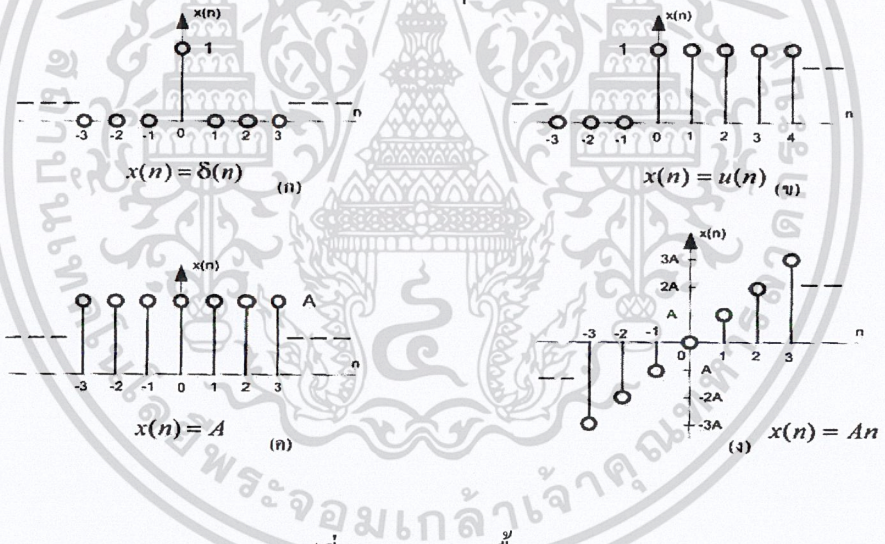
{x(n)} = {……0,0,0,0,1,1,1,1,……}

4. ชุดแถวค่าแปรเชิงเส้น

$$x(n)=An;\qquad -\infty\le n\le+\infty$$

หรือ

{x(n)} = {……-4A,-3A,-2A,-A,0,A,2A,3A,4A,……}



รูปที่ 3-12 ชุดแถวพื้นฐาน

2.3 การเลื่อนของชุดแถวและชุดแถวที่ซับซ้อนขึ้น

1. ชุดแถวอิมพัลล์เลื่อน และชุดแถวขั้นบันไดหนึ่งหน่วยเลื่อน

ชุดแถวอิมพัลล์เลื่อนนิยามโดย

$$\delta(n-m)=\begin{cases}1, n=m\\ 0, n\ne m\end{cases}$$

และชุดแถวขั้นบันไดหนึ่งหน่วยเลื่อน

$$u(n-m)=\begin{cases}1 & ,m\le n\le+\infty\\ 0 & ,n<m\end{cases}$$

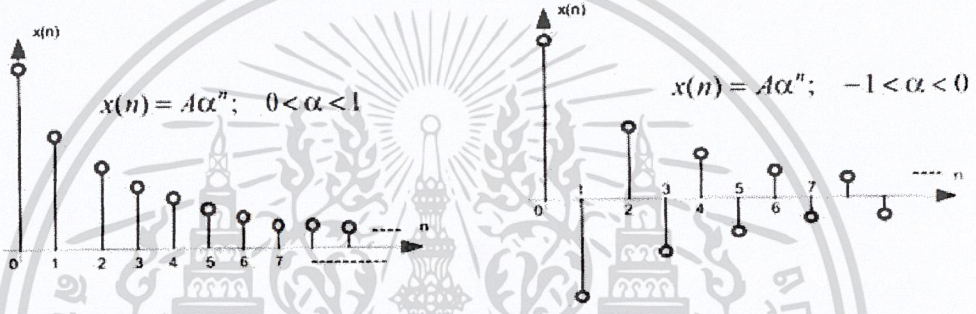
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ชุดแถวเอ็กโพเนนเชียล และชุดแถวรูปซายน์

รูปทั่วไปของชุดแถวเอ็กโพเนนเชียลสามารถเขียนได้โดย

$$x(n) = A\alpha^n$$

ถ้าหาก A และ α เป็นค่าจริง x(n) ก็จะเป็นค่าจริงและถ้า $0 < \alpha < 1$ และ A เป็นบวกชุดแถวมีค่าเป็นบวกและค่อยๆลดลง ถ้า $-1 < \alpha < 0$ ชุดแถวจะค่อยๆลดลงแต่จะสลับขนาดกัน และหาก $|\alpha| > 1$ ค่าของชุดแถวจะเพิ่มขึ้นเมื่อ n มากขึ้น



รูปที่ 3-13 ชุดแถวของ $x(n) = A\alpha^n$ เมื่อ α มีค่าต่างๆ

ส่วนชุดแถวรูปซายน์นั้นมีรูปทั่วไปคือ

$x(n) = A\cos(n\omega_0 + \phi)$ เมื่อ A และ ϕ เป็นค่าจริง

สมมติ ว่าทั้ง A และ α เป็นจำนวนเชิงซ้อน คือ $A = |A| e^{j\phi}$ จะได้ว่า

$$\begin{aligned} x(n) = A\alpha^n &= |A|e^{j\phi} \cdot |\alpha|^n e^{jn\omega_0} \\ &= |A||\alpha|^n e^{j(n\omega_0+\phi)} \\ &= |A||\alpha|^n \{\cos(n\omega_0 + \phi) + j\sin(n\omega_0 + \phi)\} \end{aligned}$$

เป็นกรณีเฉพาะหากให้ $\omega_0 = \pi$ และ $\phi = 0$ เมื่อ $|\alpha| > 1$ ส่วนที่เป็นจำนวนจริงของ x(n) จะมีค่า

$$\begin{aligned} \Re\{x(n)\} &= |A||\alpha|^n \cos(n\pi) \\ &= |A||\alpha|^n (-1)^n \end{aligned}$$

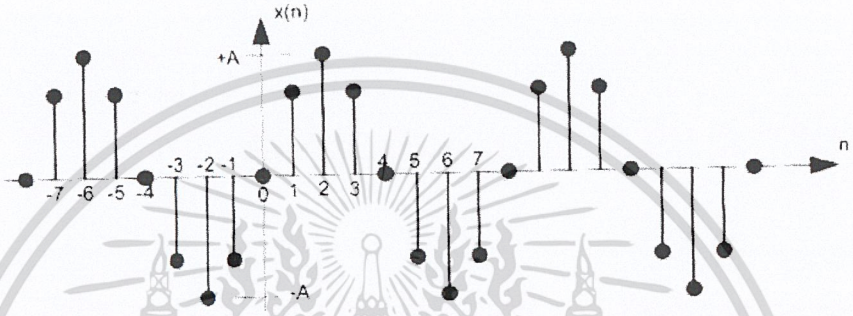
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นชุดแถวเอ็กโพเนนเชียลที่มีค่าเพิ่มขึ้น แต่ขนาดจะสลับกันตามค่า n ส่วนจินตภาพนั้นก็จะมีค่าศูนย์ทั้งหมด (เพราะ $\sin(n\pi) = 0$; ทุกค่าของ n)

เมื่อ $|\alpha| = 1$; x(n) จะเรียกชุดแถวเอ็กโพเนนเชียลเชิงซ้อน คือ

$$x(n) = |A|e^{j(n\omega_0+\phi)} = A\{\cos(n\omega_0 + \phi) + j\sin(n\omega_0 + \phi)\}$$

โดยที่ค่าจริงและค่าจินตภาพจะเปลี่ยนแปลงอย่างรูปซายน์

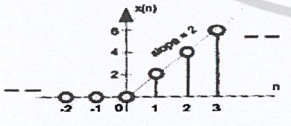


รูปที่ 3-14 ชุดแถวรูปซายน์

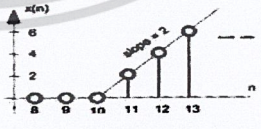
3. การบวกและการคูณกันระหว่างชุดแถว

ในระบบเดียวกันจะถือว่าคาบเวลาในการสุ่มสัญญาณมีค่าเดียว ดังนั้นการบวกหรือการการคูณกันระหว่างชุดแถวจะเป็นการกระทำแบบจุดต่อจุด หรือตัวอย่างต่อตัวอย่าง และเพื่อให้เข้าใจง่ายก็ควรจัด x(0) ของทุกชุดแถวให้ตรงกันจากข้อสังเกตจะได้ว่า

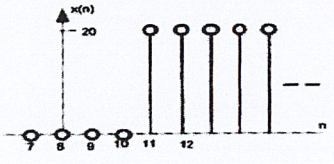
$$\delta(n) = u(n) - u(n-1)$$



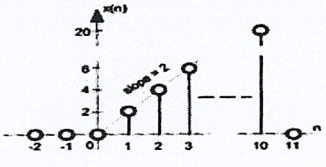
(ก) $x_1(n) = 2nu(n)$



(ข) $x_2(n) = 2(n-10)u(n-11)$



(ค) $x_3(n) = 20u(n-11)$



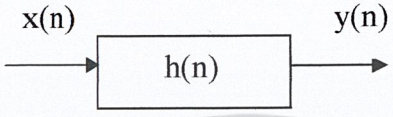
(ง) $x(n) = x_1(n) - x_2(n) - x_3(n)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-15 แสดงการบวกและการคูณกันระหว่างชุดแถว

2.4 อินพุตและเอ้าท์พุตของระบบ

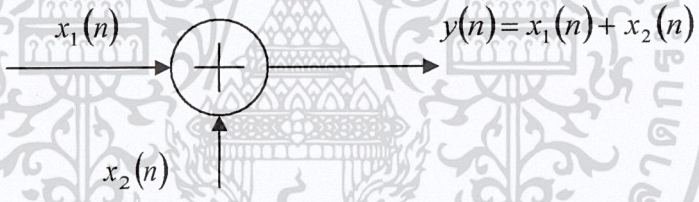
ทำนองเดียวกันกับระบบต่อเนื่อง ระบบเวลาเต็มหน่วยจะมีฟังก์ชันของระบบ ซึ่งบอกความสัมพันธ์ระหว่างอินพุตและเอ้าท์พุต



รูปที่ 3-16 แสดงอินพุตและเอ้าท์พุตของระบบ

2.5 บล็อกไดอะแกรมที่ใช้แทนระบบเวลาเต็มหน่วย

ตัวบวก ที่จะบวกสองสัญญาณเข้าด้วยกัน สังเกตว่าในกรณีนี้ชุดแถวทั้งสองสามารถที่จะบวกเข้าด้วยกันได้เลย ไม่จำเป็นจะต้องเก็บชุดแถวหนึ่งเอาไว้ก่อน

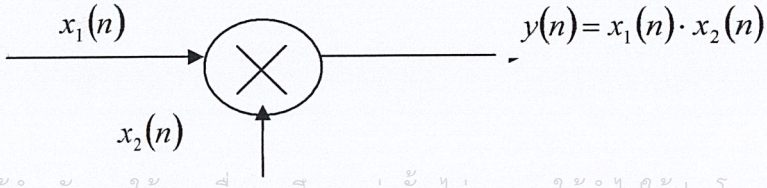


รูปที่ 3-17 แสดงการบวกสัญญาณ 2 สัญญาเข้าด้วยกัน

ตัวคูณด้วยค่าคงที่ เป็นการเพิ่มขนาดของชุดแถวด้วยค่าคงที่ค่าหนึ่งการทำงานนี้ไม่จำเป็นต้องมีการเก็บสัญญาณไว้ก่อน

รูปที่ 3-18 แสดงการคูณสัญญาณด้วยค่าคงที่

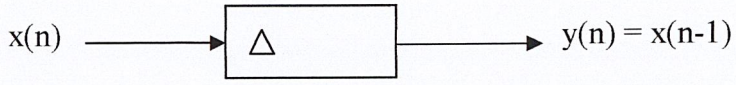
ตัวคูณสัญญาณ ตัวคูณสัญญาณจะคูณสองสัญญาณเข้าด้วยกันโดยไม่จำเป็นต้องเก็บชุดแถวใดชุดแถวหนึ่งไว้ก่อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

รูปที่ 3-19 แสดงการคูณสัญญาณ 2 สัญญาเข้าด้วยกัน

หน่วยหน่วงเวลา เป็นหน่วยที่หน่วงสัญญาณที่ผ่านไป 1 ตัวอย่าง เช่นถ้าหากอินพุตเป็นสัญญาณ x(n) เอาท์พุตก็จะเป็น x(n-1) หรือก็คือสัญญาณ x(n-1) จะต้องเก็บไว้ในหน่วยความจำและเรียกคืนมาเมื่อเวลา n



หรือ หน่วงเวลา

หน่วงเวลา

รูปที่ 3-20 แสดงการหน่วงเวลา

หน่วยก้าวหน้าเวลาจะตรงกันข้ามกับหน่วยหน่วงเวลา หน่วยก้าวหน้าจะทำการเลื่อนชุดแถวล่วงหน้า 1 หน่วย เช่นหากสัญญาณอินพุตเป็น x(n) ก็จะได้เอาท์พุตเป็น x (n + 1) สังเกตว่าในระยะเวลาจริง การก้าวหน้าไม่อาจจะเกิดขึ้นได้ แต่ถ้าหากชุดแถวถูกเก็บไว้ในหน่วยความจำแล้ว การก้าวหน้าของหน่วยสัญญาณจึงเป็นไปได้

3.1.4 การวิเคราะห์ระบบเวลาเต็มหน่วย (Analysis of Discrete System)

1. การตอบสนองของระบบต่ออิมพัลส์

การอธิบายระบบเวลาเต็มหน่วยในรูปทั่วไป สามารถใช้สมการความแตกต่าง (Difference Equation) คือ

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + \ldots + a_N y(n-N)$$

$$= b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M)$$

เมื่อ a และ b เป็นค่าจริง และระบบดังกล่าวก็เรียกว่าระบบอันดับที่ N เพราะเอ้าท์พุต y(n) จะขึ้นอยู่กับเอ้าท์พุตก่อนๆจาก(n - 1) จนถึง (n - N) จะได้

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{M=0}^{M} b_m x(n-m)$$

หรือ

$$y(n) = -\sum_{k=0}^{N} a_k y(n-k) + \sum_{M=0}^{M} b_m x(n-m)$$

จะใช้กล่าวแทนระบบที่วนซ้ำ (Recursive System) ในขณะเดียวกัน หากระบบที่ไม่วนซ้ำ (Non-recursive System) สมการจะเหลือเพียง

$$y(n) = \sum_{M=0}^{M} b_m x(n-m)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบสนองต่ออินพุต x(n) = δ(n) จะเรียกว่าการตอบสนองต่ออิมพัลส์ ระบบที่มีการสิ้นสุดของ h(n) จะเรียกว่าระบบตอบสนองต่ออิมพัลส์สิ้นสุด (Finite Impluse Response; IIR)

2. การแปลงแซด (Z Transform)

การแปลงแซด (Z Transform , ZT) เป็นวิธีการหนึ่งที่สำคัญในการวิเคราะห์ การออกแบบ และการสร้างแบบในการกระมวลสัญญาณเต็มหน่วย มีการวิเคราะห์ และการประมวลผลสัญญาณจำนวนมากที่กระทำภายใต้แวดล้อม Z หรือโดเมน Z (Z Domain) การแปลงเป็นการแปลงชุดแถวให้กลายเป็นฟังชันที่จะสามารถนำไปวิเคราะห์ได้กว้างขวางกว่าการแปลงฟูริเยร์

3. นิยามการแปลงแซด (Define of ZT)

พิจารณาชุดแถว x[n] ซึ่งนิยามโดย $x(n)=\sum_{m=-\alpha}^{m=\alpha} x(m)\delta(n-m)$ สำหรับ m ที่เป็นจำนวนเต็มบวกใดๆการแปลงแซดของ x(n) สามารถที่จะเขียนแทนได้ดังนี้

$$z[x(n)] = X(z) = \sum_{n=-\alpha}^{n=\alpha} x(n) z^{-n}$$

เมื่อ z คือตัวแปรเชิงซ้อน $z = \mathrm{Re}\, z + \mathrm{Im}\, z$

4. การแปลงแซดของชุดแถวที่สำคัญ

1. **ชุดแถวของอิมพัลส์**

$$f(n) = A\,\delta(n)$$

$$F(Z) = \sum_{-\alpha}^{\alpha} A\,\delta(n) z^{-n}$$

$$= AZ^{0} = A$$

2. **ชุดแถวขั้นบันได**

$$f(n) = A\,\delta(n), f(n) = \begin{cases} 0; n < 0 \\ A; n > 0 \end{cases}$$

$$F(Z) = \sum_{-\alpha}^{\alpha} Au(n) z^{-n}$$

$$= A\sum_{n=0}^{n=\infty} z^{-n} = A\sum_{n=0}^{n=\infty}\left(\frac{1}{z}\right)^{n} = \frac{A}{1-\frac{1}{z}}, \text{เพราะว่า} \left[\sum_{n=0}^{n=\infty}\left[\frac{1}{z}\right]^{n} = \frac{1}{1-\frac{1}{z}}\right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \frac{Az}{z-1}, ROC\ |z| > 1$$

3. ชุดแถวเอ็กโพเนนเชียล (ชุดแถวจริง)

$$f(n) = Aa^n \begin{cases} n \geq 0 \\ 0; n < 0 \end{cases}$$

$$F(z) = \sum Aa^n z^{-n} = A\sum \left(\frac{a}{z}\right)^n = \frac{A}{1-\frac{a}{z}}$$

4. ชุดแถวเชิงซ้อน

$$f(n) = Aa^n e^{jn\theta} \quad \text{เมื่อ} \quad n \geq 0$$

$$F(Z) = \sum_{-\infty}^{\infty} Aa^n e^{jn\theta} z^{-n}$$

$$= A\sum_{n=0}^{\infty} a^n e^{jn\theta} z^{-n}$$

$$= A\sum \left(ae^{j\theta z^{-1}}\right)^n$$

$$= \frac{A}{1-\frac{ae^{j\theta}}{z}}$$

ROC $\left|ae^{j\theta} z^{-1}\right| < 1$ หรือ $|z| > \left|ae^{j\theta}\right|$ หรือก็คือ $|z| > |a|$ เพราะว่า $\left|e^{j\theta}\right| = 1$

ตารางที่ 3-1 การแปลงแซดของชุดแถวที่สำคัญๆ

ชุดแถว	X(n)	X(z)	ROC
	x(n – m)	$z^{-m}X(z)$	
	$a^n x(n)$	X(z/a)	
	nx(n)	-zd{X(z)}/dz	
ตัวอย่างเดี่ยว	$\delta(n)$	1	ทุกค่าของ z
	$\delta(n-m)$ m>0	z^{-m}	$\|z\| > 0$
	$\delta(n-m)$ m>0	z^{-m}	$\|z\| < \infty$
ชุดแถวขั้นบันได	u(n)	z /(z - 1)	$\|z\| > 1$
	- u(-n-1)	z /(z - 1)	$\|z\| < 1$
เอ็กโพเนนเชียล	$a^n u(n)$	z /(z - a)	$\|z\| > \|a\|$
	$-b^n u(-n-1)$	z /(z - b)	$\|z\| < \|b\|$
	$na^n u(n)$	$az^{-1}/(1-az^{-1})^2$	$\|z\| > \|a\|$
	$-nb^n u(-n-1)$	$az^{-1}/(1-az^{-1})^2$	$\|z\| < \|b\|$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การแปลงกลับแซด (Inverse Z Transform : IZT)

x(n) สามารถที่จะหาจาก X(z) โดยการแปลงกลับแซด โดยใช้กระบวนการการประเมินค่าที่ $z = e^{j\theta}$ ซึ่งเขียนได้โดย

$$X(z)\Big|_{z=e^{j\theta}} = \sum_{n=-\infty}^{n=+\infty} x(n)e^{-jn\theta}$$

ดังนั้นบนวงกลมรัศมีหนึ่งหน่วย ($z = e^{j\theta}$), X(z) จะเป็นฟังก์ชันมีคาบ 2π ของ θ , x(n) จึงเขียนได้เป็น

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(e^{j\theta}\right)e^{j\theta}\,d\theta$$

ที่ $z = e^{j\theta}$, $dz = e^{j\theta}\,d\theta$ หรือ $dz = jz\,d\theta$ แทนกลับไปในสมการจะได้

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(z)z^{n}\left(dz / jz\right)$$

หรือ

$$x(n) = \frac{1}{2\pi j}\oint_{c} X(z)z^{n-1}\,dz$$

การอินทิเกรดบนเส้นระดับในระนาบ z ทวนเข็มนาฬิกาตามเส้นรอบใดๆรอบๆ z นั้น เป็นสูตรตามทฤษฎีที่มักจะไม่ได้ใช้จริงมากนักเพราะมักใช้วิธีหนึ่งใน 3 วิธี คือการหารยาว การแยกเศษส่วนย่อย และการใช้ตาราง

ตารางที่ 3-2 ตารางการแปลงกลับ

k	X(z)	x(n); ROC
1	$X(z) = z / (z - a)$	$= a^{n} u(n)$
2	$X(z) = z / (z - a)^{2}$	$= na^{n-1} u(n)$
3	$X(z) = z / (z - a)^{3}$	$= \frac{n(n-1)a^{n-2}}{2!} u(n)$
4	$X(z) = z / (z - a)^{4}$	$= \frac{n(n-1)(n-2)a^{n}}{3!} u(n)$
k	$X(z) = z / (z - a)^{k}$	$= \frac{n(n-1)(n-2)...(n-k-2)a^{n-k+1}}{(k-1)!} u(n)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คุณสมบัติของการแปลงแซด (Property of ZT)

เมื่อ x(n) เป็นชุดแถวอันดับ N โดยที่คู่การแปลงแซดคือ f(n) $\longleftrightarrow$ F(z)

1. ความเป็นเชิงเส้น

ถ้าหาก $z(f_1(n)) = F_1(z)$; และ $z(f_2(n)) = F_2(z)$ จะได้

$$Z[(af_1(n)) + (bf_2(n))] = aF_1(z) + bF_2(z)$$

สำหรับ $f_1(n)$, $f_2(n)$, a และ b ใดๆ

ดังนั้น

$$F(z) = F_1(z) + F_2(z) + F_3(z) + \ldots F_k(z)$$

หรือ

$$\sum_{K=1}^{P} C_K f_K(n) \Leftrightarrow \sum_{K=1}^{P} C_K F_K(z)$$

2. การเลื่อนชุดแถว

หาก x(n - m) เป็นชุดแถวที่เลื่อนของชุดแถว x(n) จะสามารถแสดงได้ว่า

$$z[f(n-m)] = \sum_{n=-\infty}^{\infty} f(n-m)z^{-n}$$

กำหนดให้ l = n – m ; - n = -l – m

$$z(f(l)) = \sum_{l=-\infty}^{l=\infty} f(l)z^{-l-m}$$

เนื่องจาก m เป็นอิสระจาก l

$$= z^{-m} \sum_{l=-\infty}^{\infty} f(l)z^{-l}$$

$$= z^{-m} F(z)$$

3. การคูณประสาน และ การแปลงแซด

เราทราบมาแล้วว่าเอ้าท์พุตของระบบ LTI สามารถที่จะเขียนในรูปแบบของการคูณประสาน โดย

$$y(n) = h(n) * x(n)$$

$$= \sum_{m=-\infty}^{\infty} h(m)x(n-m)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดำเนินการแปลงแซดต่อ y(n) จะได้

$$z(y(n)) = z\left[\sum_{n=-\infty}^{\infty} h(m)x(n-m)\right]$$

$$Y(z) = \sum_{n=-\infty}^{\infty}\left[\sum_{m=-\infty}^{\infty} h(m)x(n-m)\right]z^{-n}$$

$$= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(m)X(n-m)z^{-n}$$

$$= \sum_{m=-\infty}^{\infty} h(m)\sum_{n=-\infty}^{\infty} z(n-m)z^{-n}$$

กำหนดให้ 1 = n- m หรือ ก็คือ – n = -1- m

$$Y(z) = \sum_{m=-\alpha}^{\alpha} h(m)\sum_{l=-\alpha}^{\alpha} n(l)z^{-l}z^{-m}$$

$$= \sum_{m=-\infty}^{\infty} h(m)z^{-m}\sum_{l=-\infty}^{\infty} x(l)z^{-l}$$

สรุปก็คือ

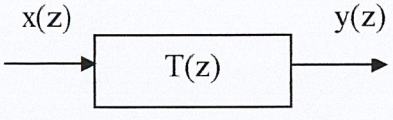
$$Y(z) = H(z)X(z)$$

3.1.5 ฟังก์ชันของระบบ

ฟังก์ชันถ่ายโอนสามารถเขียนได้โดย $H(z) = \frac{z-transform_of_o/p}{z-transform_of_i/p} = T(z)$

นิยามเมื่อ z(x(n)) = X(z) และ z(y(n)) = Y(z) จะได้ว่า

$$H(z) = \frac{Y(z)}{X(z)}$$



1. ฟังก์ชันถ่ายโอนของตัวอย่างเดี่ยว

กรณีนี้ $x[n] = \delta(n)$ และทำให้ได้เอ้าท์พุต y [n] = h(n) ดังนั้น

$$\frac{Y(z)}{X(z)} = H(z) = z(h(n))$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ฟังก์ชันถ่ายโอนและการแปลงฟูริเยร์

ในการหาการตอบสนองทางความถี่ของระบบ หาได้โดยการแปลงฟูริเยร์ของการตอบสนองตัวอย่างเดี่ยว h(n)

$$H\left(e^{jn\theta}\right)=\sum_{n=-\infty}^{\infty}h(n)e^{-jn\theta}$$

แต่ถ้าเรามีฟังก์ชันของตัวอย่างเดี่ยวคือ

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

ดังนั้นการตอบสนองทางความถี่ของระบบ ก็คือฟังก์ชันถ่ายโอนเมื่อ $z=e^{jn\theta}$ หรือ

$$H\left(e^{jn\theta}\right)=H(z)\Big|_{z=e^{j\theta}}$$

3.1.6 การหาฟังก์ชันของระบบ

เราได้เคยศึกษามาแล้วว่า h(n) คือการตอบสนองของระบบเมื่อป้อนอินพุต เป็น $\delta(n)$ และ H ($e^{j\omega}$) ซึ่งได้จากการแปลงฟูริเยร์ ของ h(n) จะเรียกว่าการตอบสนองความถี่ ตอนนี้การแปลงแซดของ h(n) จะเรียกว่าฟังก์ชันของระบบ (System Function , H(z))

การหา H(z) นั้นเราทำได้ 3 วิธีคือ

1. เมื่อทราบค่า h(n) ก็หาค่า H(z) ได้

$$h(n)\underset{ZT}{\longrightarrow}H(z)\ \text{คือ}\ H(z)=\sum_{n=-\infty}^{\infty}h(n)z^{-n}$$

2. ป้อนอินพุตให้กับระบบด้วย $x(n)=z^n$ ซึ่งเอ้าท์พุต Y(n) ของระบบหาได้โดย

$$y(n)=\sum_{i=-\alpha}^{\alpha}h(i)x(n-i)$$

สำหรับ $x(n)=z^n$ จะได้ $z[x(n-i)]=X(n-i)=z^{n-i}$

$$\therefore\quad y(n)=\sum_{j=-\alpha}^{\alpha}h(i)z^{n-i}$$

$$=\sum_{i=-\infty}^{\infty}h(i)z^{-i}=z^nH(z)$$

เราจึงสามารถที่จะหา H(z) ได้จากการหาร y(n) ด้วย z^n

3. จากระบบสมการใช้การเลื่อนของหน่วยเวลา x(n-i) $\longleftrightarrow$ $z^i X(z)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบวงจรกรองความถี่แบบ FIR (FIR Filter Design)

ในการประมวลผลเชิงเลขจะมีระบบที่สำคัญ 2 ระบบ คือระบบที่อยู่ในมิติของความเวลา (Time domain) เรียกว่า Digital filter และ ระบบที่อยู่ในมิติของความถี่ (Frequency domain) เรียกว่า Spectrum analyzers แต่เราจะศึกษากันเรื่องระบบที่อยู่ในมิติของเวลา ในการออกแบบจะเป็นการออกแบบเพื่อใช้ในการเลือกความถี่ที่ต้องการ อันได้แก่ วงจรกรองความถี่ต่ำผ่าน (Low pass), วงจรกรองความถี่สูง (High pass), วงจรกรองความถี่แถบกลาง (Band pass), และวงจรลดทอนความถี่แถบกลาง (Bandstop pass) ซึ่งในบทนี้จะออกแบบวงจรกรองความถี่แบบFIR

การออกแบบวงจรกรองความถี่ประกอบด้วยกัน 3 ขั้นตอน ในการใช้งานการประมวลผลสัญญาณวงจรกรองความถี่เชิงเลขถูกออกแบบมาเพื่อใช้ในการเลือกความถี่ที่จะส่งผ่าน ดังนั้นเราจึงพิจารณามิติของความถี่เพื่อหาผลตอบสนองของขนาดและเฟสของวงจรกรองความถี่เชิงเลข

ผลตอบสนองของขนาด(Magnitude response) จะมีเราพิจารณาด้วยกัน 2 ลักษณะคือในลักษณะของ Absolute specification, $\left|H(e^{j\omega})\right|$ และในลักษณะของ relative specification คืออยู่ในรูปแบบของdecibels(dB) โดยที่

$$dB\ \ scale = -20\log_{10}\frac{\left|H(e^{j\omega})\right|}{\left|H(e^{j\omega})\right|_{\max}} \geq 0$$

จากสมการนิยมใช้กันทั้งใน FIR และ IIR จากลักษณะทั้ง 2 เราจะพิจารณาวงจรกรองความถี่ต่ำเป็นตัวอย่าง

จากรูปที่ 3-21 เป็นลักษณะของวงจรกรองความถี่ต่ำ จากรูปที่ 3-21 (a) จะเป็นลักษณะ Absolute Specification โดยที่

- แถบความกว้าง $[0, \omega_P]$ เรียกว่า passBand และ δ_1 เป็น ripple ที่เราสามารถเราสามารถยอมรับได้
- แถบความกว้าง $[\omega_S, \pi]$ เรียกว่า Stopband และ δ_2 เป็น ripple เช่นกัน
- แถบความกว้าง $[\omega_P, \omega_S]$ เรียกว่า transition band

และรูปที่ 3-21(a) จะเป็น Relative (DB) Specification โดยที่

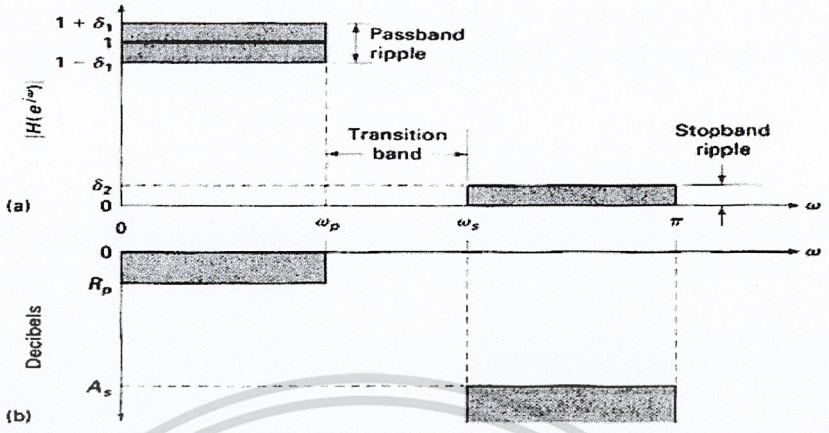
- R_P เป็น passband ripple ในหน่วย DB
- A_S เป็น stopband attenuation ในหน่วย DB

โดยที่ พารามิเตอร์ทั้งสองของทั้งสองลักษณะจะมีความสัมพันธ์กันดังนี้

$$R_P = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} > 0(\approx 0)$$

$$A_S = -20\log_{10}\frac{\delta_2}{1+\delta_1} > 0(>> 1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-21 FIR filter specification:(a)Absolute (b)Relative

3.2.1 คุณสมบัติความเป็นเชิงเส้นทางเฟสในวงจรกรองความถี่แบบFIR

(Properties of Linear-Phase FIR Filter)

พิจารณาลักษณะของผลตอบสนองอิมพัลส์ และผลตอบสนองความถี่ และตำแหน่งของฟังก์ชั่นZero ของวงจรความเป็นเชิงเส้นทางเฟสของ FIR โดยให้ $h(n), 0 \le n \le M-1$ เป็นผลตอบสนองอิมพัลส์ความยาว M เมื่อระบบมีฟังก์ชั่นเป็น

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

เมื่อมี $M-1$ โพลที่ออริจิน $z=0$ และ $M-1$ Zeros ตำแหน่งต่างๆในระนาบ Z ผลตอบสนองความถี่มีฟังก์ชั่นเป็น

$$H(e^{jw}) = \sum_{n=0}^{M-1} h(n) e^{-j\omega n}, -\pi < \omega < \pi$$

จากทั้งเราจะออกแบบให้วงจรกรองความถี่ให้มีลักษณะตามที่เราต้องการจาก $h(n)$ และ $H(e^{j\omega})$ ถ้าเรากำหนดให้เป็น Linear-phase constraint จะได้ว่า

$$\angle H(e^{j\omega}) = -\alpha\omega, \qquad -\pi < \omega \le -\pi$$

เมื่อ α เป็น Constant phase delay จากลักษณะความสมมาตรของ $h(n)$ จะได้ว่า

$h(n) = h(M-1-n), \ 0 \le n \le M-1$ เมื่อ $\alpha = \dfrac{M-1}{2}$

เมื่อ $h(n)$ สมมาตรจะทำให้ α เป็นลำดับของการสมมาตร จะได้เกิดการสมมาตรได้ 2 ลักษณะ คือ

- เมื่อ M เป็นจำนวนคี่ ในกรณีนี้จะทำให้ $\alpha = \dfrac{M-1}{2}$ จะเป็นจำนวนเต็ม
- เมื่อ M เป็นจำนวนคู่ในกรณีนี้จะทำให้ $\alpha = \dfrac{M-1}{2}$ จะไม่เป็นจำนวนเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราอาจจะกำหนดผลการตอบสนองทางเฟสใหม่เป็น

$$\angle H(e^{j\omega}) = \beta - \alpha\omega$$

จากสมการพบว่าจะไม่ใช่เส้นตรงที่ผ่านจุดออริจิน ในกรณี α จะไม่ใช่ Constant phase delay แต่

$$\frac{d\angle H(e^{j\omega})}{d\omega} = -\alpha$$

จะเป็นค่าคงที่ซึ่งเรียกว่า Group delay ดังนั้น α เรียกว่า constant group delay โดยในกรณีนี้เฟสและความถี่จะถูกหน่วงด้วยอัตราที่คงที่ แต่จะมีบางความถี่อาจจะถูกหน่วงมากกว่าหรืออาจจะน้อยกว่า สำหรับ Linear phase สามารถแสดงได้ดังนี้

$$h(n) = -h(M-1-n), 0 \le n \le (M-1); \alpha = \frac{M-1}{2}, \beta = \pm\frac{\pi}{2}$$

นั้นหมายความว่าผลตอบสนองอิมพัลส์ $h(n)$ จะเป็นแบบ Antisymmetric เมื่อลำดับของดัชนีเป็น $\alpha = (M-1)/2$ และมีรูปแบบที่เป็นไปได้ 2 รูปแบบเช่นกันคือ

- M เป็นจำนวนคี่ในกรณีนี้ $\alpha = (M-1)/2$ จะเป็นจำนวนเต็ม และที่ตำแหน่ง $\alpha = (M-1)/2$ จะมีค่าเท่ากับ 0
- M เป็นจำนวนคู่ในกรณีนี้ $\alpha = (M-1)/2$ จะเป็นไม่จำนวนเต็ม และที่ตำแหน่ง

3.2.2 ผลการตอบสนองทางความถี่ (Frequency response)

จากที่ได้กล่าวมาจะมีกรณีที่เกิดขึ้นเป็นเมื่อรวมกันของ Symmetry และ Antisymmetry และกรณีจำนวนคู่และจำนวนคี่ได้ 4 กรณีที่เป็น linear-phase FIR filter สามารถเขียนสมการเพื่ออธิบายรูปร่างโดยแทนด้วย $H(e^{j\omega})$ เป็น

$$H(e^{j\omega}) = H_r(\omega)e^{j(\beta-\alpha\omega)}; \ \beta = \pm\frac{\pi}{2}, \ \alpha = \frac{M-1}{2}$$

เมื่อ $H_r(\omega)$ เป็นฟังก์ชั่นผลการตอบสนองของขนาด (Amplitude response) ซึ่งเป็นฟังก์ชั่นค่าจริงมีได้ทั้งบวกและลบ โดยที่ phase response มีส่วนร่วมกับ magnitude response ในลักษณะฟังก์ชั่นไม่ต่อเนื่อง (discontinuous function) ในขณะที่ phase response จะมีส่วนร่วมกับ Amplitude response ในลักษณะฟังก์ชั่นต่อเนื่องเชิงเส้น (continuous linear function)

<u>รูปแบบที่ 1</u> Linear-phase FIR filterกรณีที่เป็น Symmetrical impulse responseโดย M เป็นจำนวนคี่ ในกรณีนี้ $\beta = 0, \alpha = (M-1)/2$ เป็นจำนวนเต็ม และ $h(n) = h(M-1-n), 0 \le n \le M-1$ เรา สามารถเขียนได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(e^{j\omega}) = \left[\sum_{n=0}^{(M-1)/2} a(n)\cos\omega n \right] e^{-j\omega(M-1)/2}$$

เมื่อ $a(n)$ เป็นลำดับที่ได้มาจาก $h(n)$ ดังนี้

$$a(0) = h\left(\frac{M-1}{2}\right) \qquad \text{: the middle sample}$$

$$a(n) = 2h\left(\frac{M-1}{2} - n\right),\ 1 \le n \le \frac{M-3}{2}$$

จากสมการจะได้ว่า

$$H_r(\omega) = \sum_{n=0}^{(M-1)/2} a(n)\cos\omega n$$

<u>รูปแบบที่ 2</u> Linear-phase FIR filter กรณีที่เป็น Symmetrical impulse response โดย M เป็นจำนวนคู่ ในกรณีนี้ $\beta = 0, h(n) = h(M-1-n), 0 \le n \le M-1$ และ $\alpha = (M-1)/2$ ไม่เป็นจำนวนเต็ม เราสามารถเขียนอยู่รูปของ

$$H(e^{j\omega}) = \left[\sum_{n=1}^{M/2} b(n)\cos\left\{\omega\left(n - \frac{1}{2}\right)\right\} \right] e^{-j\omega(M-1)/2}$$

เมื่อ

$$b(n) = 2h\left(\frac{M}{2} - n\right),\ n = 1,2,\ldots,\frac{M}{2}$$

ดังนั้น $H_r(\omega) = \sum_{n=1}^{M/2} b(n)\cos\left\{\omega\left(n - \frac{1}{2}\right)\right\}$

เมื่อ $\omega = \pi$ จะได้ว่า $H_r(\pi) = 0$

<u>รูปแบบที่ 3</u> Linear-phase FIR Filter กรณีเป็น Antisymmetric impulse response เมื่อ M เป็นจำนวนคี่ ในกรณีนี้ $\beta = \pi/2, \alpha = (M-1)/2$ เป็นจำนวนเต็มและ $h(n) = -h(M-1-n)$, $0 \le n \le M-1$ และ $h((M-1)/2) = 0$ จะได้ว่า

$$H(e^{j\omega}) = \left[\sum_{n=1}^{(M-1)/2} c(n)\sin\omega n \right] e^{j\left[\frac{\pi}{2} - \left(\frac{M-1}{2}\right)\omega\right]}$$

เมื่อ

$$c(n) = 2h\left(\frac{M-1}{2} - n\right),\ n = 1,2,3,\ldots,\frac{M-1}{2}$$

และ $H_r(\omega) = \sum_{n-1}^{(M-1)/2} c(n)\sin\omega n$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>รูปแบบที่ 4</u> Linear-phase FIR Filter ในกรณีที่เป็นAntisymmetric impulse response เมื่อ M เป็นจำนวนคู่ ในกรณีนี้จะเป็นเช่นเดียวกันกับแบบที่ 2 จะได้ว่า

$$H(e^{j\omega}) = \left[\sum_{n=1}^{M/2} d(n)\sin\left\{\omega\left(n-\frac{1}{2}\right)\right\}\right] e^{j\left[\frac{\pi}{2}-\omega(M-1)/2\right]}$$

$$d(n) = 2h\left(\frac{M}{2}-n\right), n = 1,2,\ldots,\frac{M}{2}$$

และ $$H_r = \sum_{n=1}^{M/2} d(n)\sin\left\{\omega\left(n-\frac{1}{2}\right)\right\}$$

3.2.3 การออกแบบ FIR แบบความถี่ต่ำผ่านโดยใช้เทคนิคหน้าต่าง (Window Design Techniques)

ถ้าเราแทนวงจรกรองความถี่ต่ำผ่านในกรณีอุดมคติเป็น $H_d(e^{j\omega})$ ที่แถบความถี่เป็น $\omega_c < \pi$ จะได้ว่า

$$H_d(e^{j\omega}) = \begin{cases} 1\cdot e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

ในขณะที่ ω_c เรียกว่าความถี่คัทออฟ(Cutoff frequency) และ α คือการหน่วงเวลา (delay) ดังนั้นจากผลตอบสนองทางความถี่จากสมการ เราแปลงให้เป็นผลตอบสนองอิมพัลส์ได้เป็นได้เป็น

$$h_d(n) = \Im^{-1}\left[H_d(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1\cdot e^{-j\alpha\omega}e^{j\omega n}d\omega$$

$$= \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

ผลของการตอบสนองอิมพัลส์จะได้มาจากฟังก์ชั่นของ $h_d(n)$ และฟังก์ชั่นของหน้าต่าง ซึ่งจะได้ผลตอบสนองอิมพัลส์ของวงจรกรองความถี่ FIR เป็น $h(n)$ ที่มีความยาวเป็น M จะได้ว่า

$$h(n) = \begin{cases} h_d(n), 0 \le n \le M-1 \\ 0, elsewhere \end{cases} \quad \text{และ } \alpha = \frac{M-1}{2}$$

จากสมการเรียกว่า Windowing โดยที่ $h(n)$ เกิดจากผลคูณของ $h_d(n)$ กับฟังก์ชั่นหน้าต่าง (window function, $w(n)$) ตามสมการด้านล่าง

$$h(n) = h_d(n)w(n)$$

โดยที่ $w(n)$ เป็นฟังก์ชั่นที่อาจจะสมมาตรขึ้นอยู่กับว่าเราจะใช้ฟังก์ชั่นใดและในทางมิติของความถี่จะเป็นการconvolution กันระหว่าง $H_d(e^{j\omega})$ และ $W(e^{j\omega})$ จะได้ว่า

$$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\lambda})H_d(e^{j(\omega-\lambda)})d\omega$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นหน้าต่างในลักษณะต่างๆ(Basic Window Design Ideal)

1.Rectangular Window

เป็นฟังก์ชั่นหน้าต่างอย่างง่าย โดยที่เราจะนิยามฟังก์ชั่นหน้าต่างเป็น

$$w(n)=\begin{cases}1, 0\le n\le M-1\\ 0, otherwise\end{cases}$$

เราจะได้ผลตอบสนองความถี่ของฟังก์ชั่นหน้าต่างนี้เป็น

$$W\left(e^{j\omega}\right)=\left[\frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}\right]e^{-j\omega\frac{M-1}{2}}\Rightarrow W_r(\omega)=\frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

ถ้าเราพิจารณาเฉพาะ Amplitude response จะได้ผลตอบสนอง $H_r(\omega)$ เป็น

$$H_r(\omega)\approx\frac{1}{2\pi}\int_{-\pi}^{\omega+\omega_c}W_r(\lambda)d\lambda=\frac{1}{2\pi}\int_{-\pi}^{\omega+\omega_c}\frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}d\lambda, M>>1$$

ปัญหาที่เกิดขึ้นที่สำคัญ 2 ประการในการออกแบบเป็น rectangular window นั้นคือค่าต่ำสุดของ attenuation ใน stopband เป็น 21 dB ไม่เพียงพอสำหรับบางลักษณะงาน และอาจจะเกิด Gibbs phenomenon โดยที่ถ้าเราเพิ่ม M ให้มากขึ้นความกว้างของแต่ละ lobe จะมีค่าน้อยลงแต่พื้นที่ในแต่ละlobe จะเหมือนค่าคงที่ ดังนั้น relative amplitude ของแต่ละ lobe จะเหมือนกับค่าคงที่และมี ค่า attenuation เป็น 21dB แต่จะมี ripple ที่บริเวณใกล้กับขอบของแถบความถี่

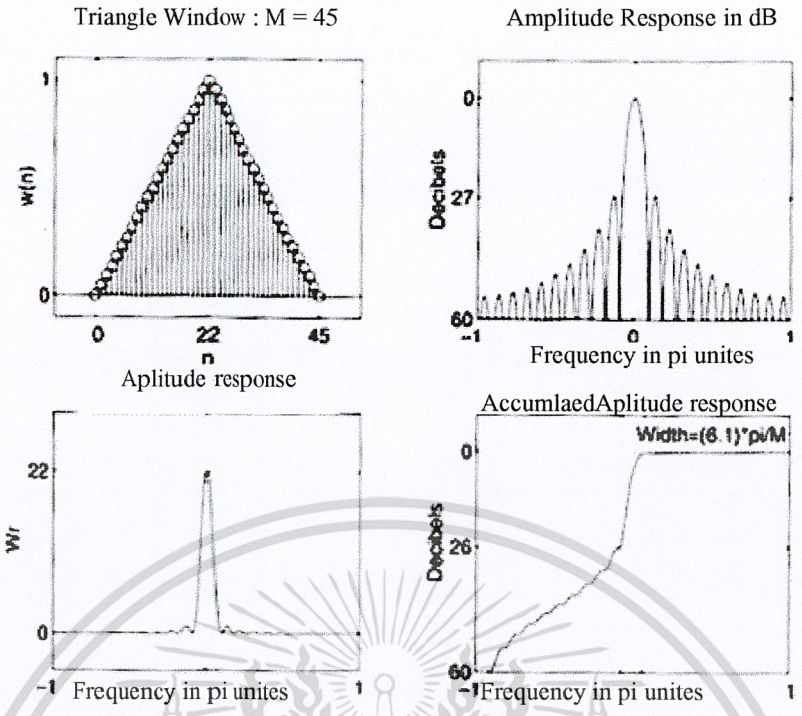
2. Bartlett window

จากปรากฏการณ์ Gibbs phenomenon ใน rectangular window ที่เกิดจากการเปลี่ยนจาก 0 ไปเป็น 1 หรือ จาก 1 เป็น 0 ดังนั้นเราจะเขียนอยู่ในรูปแบบของฟังก์ชั่นสามเหลี่ยม (triangular window) โดยให้

$$w(n)=\begin{cases}\frac{2n}{M-1}, & 0\le n\le\frac{M-1}{2}\\ 2-\frac{2n}{M-1}, & \frac{M-1}{2}\le n\le M-1\\ 0, & otherwise\end{cases}$$

ผลการตอบสนองทางความถี่เป็นดังรูปที่ 3-22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



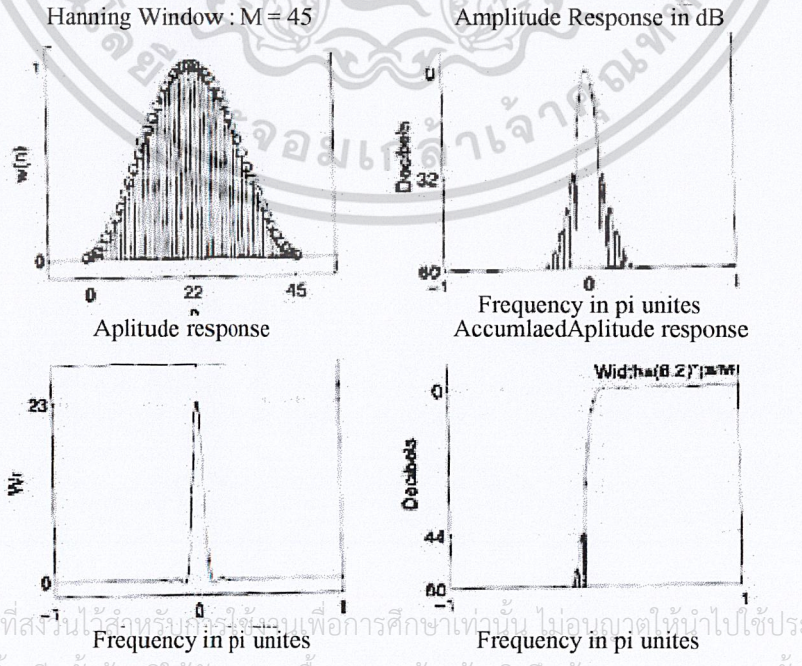
รูปที่ 3-22 Bartlett (triangular) window of frequency response

3. Hanning window

โดยการเพิ่ม cos window เข้าไป จะได้เป็น

$$w(n) = \begin{cases} 0.5\left[1-\cos\left(\dfrac{2\pi n}{M-1}\right)\right], & 0 \le n \le M-1 \\ 0, & otherwise \end{cases}$$

ผลการตอบสนองได้ดังรูปที่ 3-18

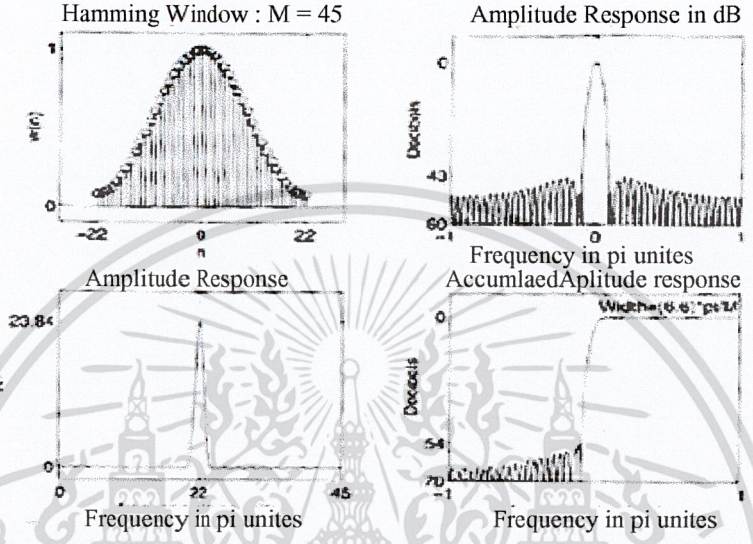


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-23 Hanning window of frequency response

4. Hamming window จะมีลักษณะเหมือนกับ Hanning window จะได้ว่า

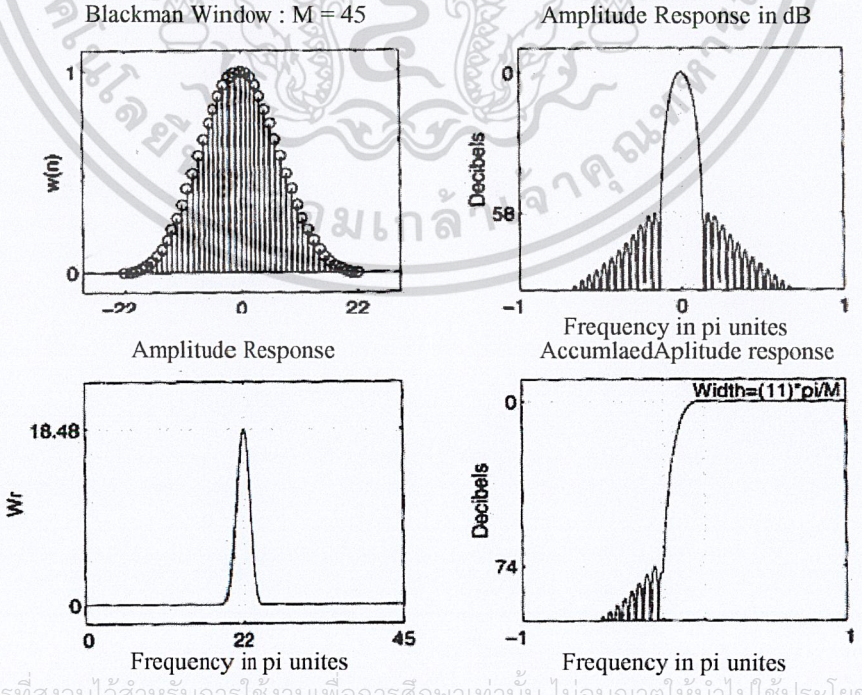
$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{M-1}\right), & 0 \le n \le M-1 \\ 0, \quad otherwise \end{cases}$$



รูปที่ 3-24 Hamming window of frequency response

5. Blackman window

$$w(n) = \begin{cases} 0.42 - 0.5\cos\left(\dfrac{2\pi n}{M-1}\right) + 0.08\cos\left(\dfrac{4\pi n}{M-1}\right), & 0 \le n \le M-1 \\ 0, \quad otherwise \end{cases}$$



รูปที่ 3-25 Blackman window of frequency response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหน้าต่างทั้ง 5 แบบ สามารถสรุปได้ดังตารางที่ 3-3 ในลักษณะของ Transition width และ ค่า Minimum stopband attenuation in DB ซึ่งเป็นประโยชน์ในการออกแบบ

ตารางที่ 3-3 Summary of commonly used window function characteristic

Window name	Transition Width $\Delta\omega_d$		Min. Stopband attenuation
	Approximate	Exact Values	
Rectangular	$\frac{4\pi}{M}$	$\frac{1.8\pi}{M}$	21 dB
Barrtlett	$\frac{8\pi}{M}$	$\frac{6.1\pi}{M}$	25 dB
Hanning	$\frac{8\pi}{M}$	$\frac{6.2\pi}{M}$	44 dB
Hamming	$\frac{8\pi}{M}$	$\frac{6.6\pi}{M}$	53 dB
Blackman	$\frac{12\pi}{M}$	$\frac{11\pi}{M}$	74 dB

6. Kaiser window

$$w(n)=\frac{I_0\left[\beta\sqrt{1-\left(1-\frac{2n}{M-1}\right)}\right]}{I_0[\beta]},\quad 0\le n\le M-1$$

เมื่อ $I_0[\cdot]$ เป็น modified zero-order Bessel function และ β เป็นพารามิเตอร์ที่ขึ้นอยู่กับ M โดยที่ในหน้าต่างแบบนี้สามารถเลือกค่า transition widths และค่า stop attenuation สามารถทำให้ได้ดีที่สุดได้ โดยสมการในการออกแบบได้ดังนี้

$$Norm.\ transition\ \ width=\Delta f\cong\frac{\omega_s-\omega_p}{2\pi}$$

$$Filter\ \ order\quad M\approx\frac{A_s-7.95}{14.39\Delta f}+1$$

$$Parameter\ \ \beta=\begin{cases}0.1102(A_s-8.7), & A_s\ge 50\\ 0.5842(A_s-21)^{0.4} & \\ +0.07886(A_s-21), & 21\le A_s\le 50\end{cases}$$

ตัวอย่างการออกแบบวงจรกรองความถี่แบบ FIR

ตัวอย่างที่ 1 จงออกแบบวงจรกรองความถี่ต่ำผ่านที่มีเงื่อนไขดังนี้

$$\omega_p=0.2\pi,\quad R_p=0.25dB$$

$$\omega_s=0.3\pi,\quad A_s=50dB$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

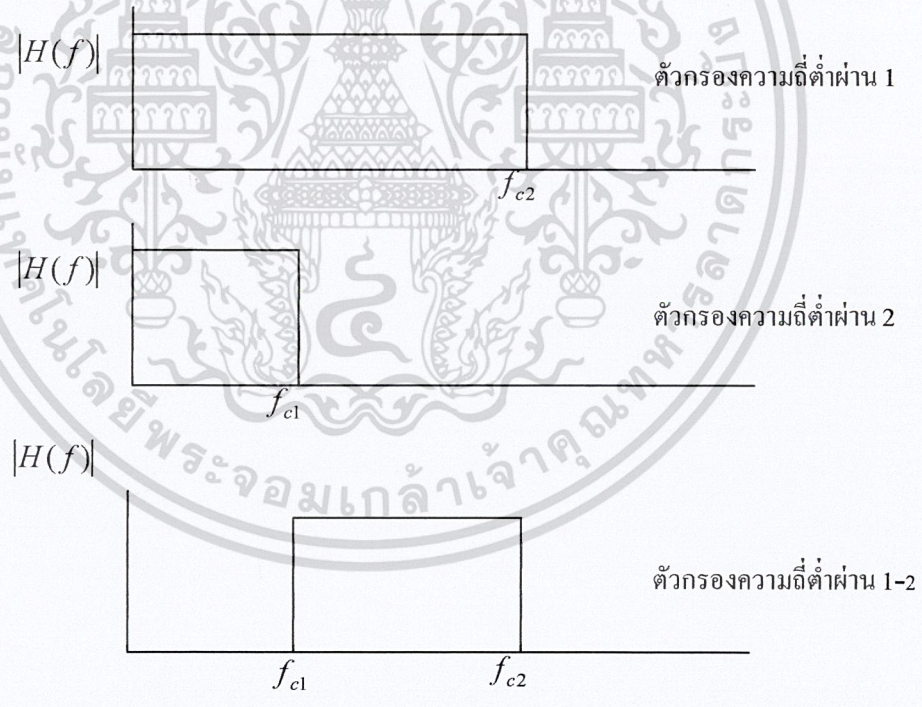
จากตารางที่ 3-3 ระหว่าง Hamming window และ Blackman window ทั้งสองจะมี attenuation มากกว่า 50 dB เราจะเลือกใช้ Hamming window เพราะมีแถบความถี่ที่ผ่านแคบกว่าซึ่งจะทำให้ความยาวของวงจรกรองน้อยกว่า

3.2.4 การออกแบบวงจรกรองความถี่ชนิดอื่น

ที่ได้ออกแบบวงจรกรองความซึ่งเป็นการออกแบบวงจรกรองความถี่แบบต่ำผ่าน จะนำมาประยุกต์สร้างวงจรกรองความถี่แบบอื่นเช่น วงจรกรองความถี่แบบสูงผ่าน (High pass) วงจรกรองความถี่แบบแถบความถี่ผ่าน (Band pass) และวงจรกรองความถี่แบบหยุดแถบความถี่ (Band stop)

1. การสร้างวงจรกรองความถี่แบบแถบความถี่ผ่าน(Band pass filter design)

เราสามารถสร้างวงจรกรองความถี่แบบแถบความถี่ผ่านได้จากวงจรกรองความถี่แบบต่ำผ่านได้โดยการการสร้างวงจรกรองความถี่แบบต่ำผ่านให้ Upper cutoff frequency ครอบคลุม Lower cutoff frequency เราจะได้วงจรกรองความถี่แบบแถบความผ่าน ซึ่งดูได้จากรูป 3-26



รูปที่ 3-26แสดงการหาวงจรกรองความถี่แถบความถี่ผ่าน

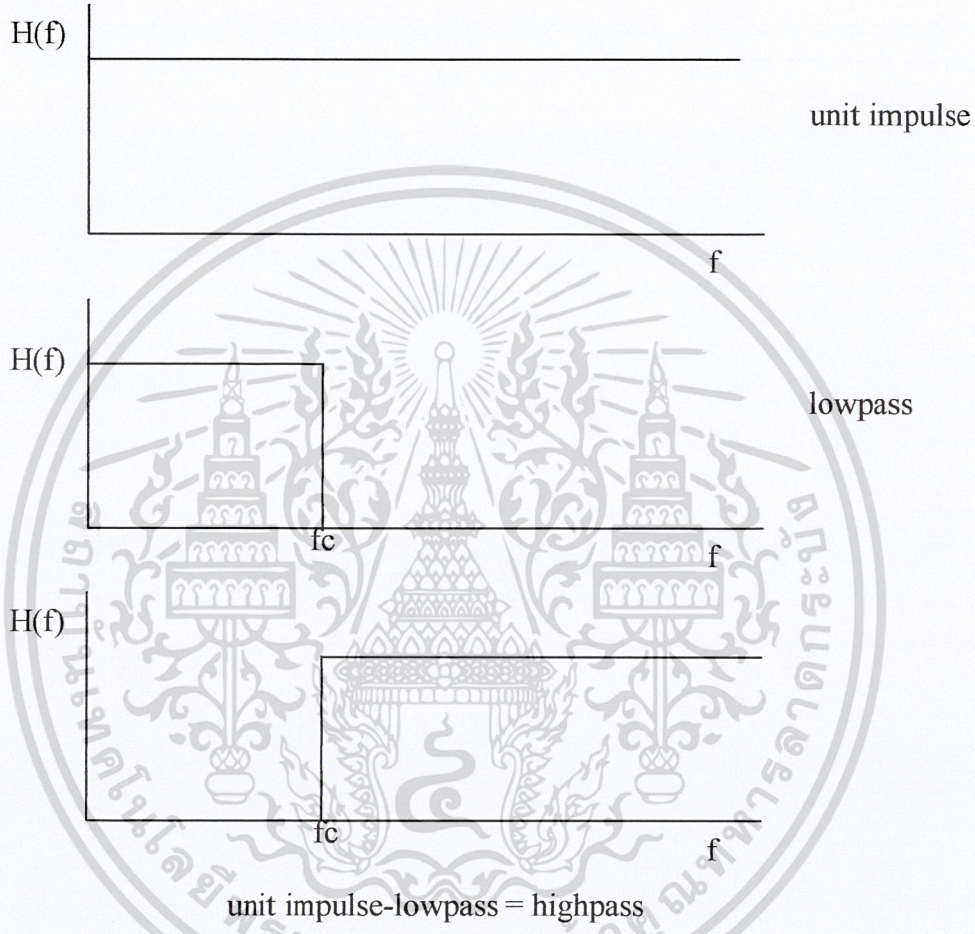
จึงจากรูปที่ 3-26 จะได้ว่าจะต้องนำสัมประสิทธิ์ของวงจรกรองความถี่ต่ำมาลบกันจะได้ว่า

$$h_d(n) = \sin\left(\frac{\omega_{c2}(n-\alpha)}{\pi(n-\alpha)}\right) - \sin\left(\frac{\omega_{c1}(n-\alpha)}{\pi(n-\alpha)}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การออกแบบวงจรกรองความถี่แบบสูงผ่าน (High pass design)

เช่นเดียวกับ Band pass แต่เป็นค่าผลต่างระหว่าง unit impulse มาลบกับ วงจรกรองความถี่แบบต่ำผ่านดังรูปที่ 3-27



รูปที่ 3-27 แสดงการหาวงจรกรองความถี่สูงผ่าน

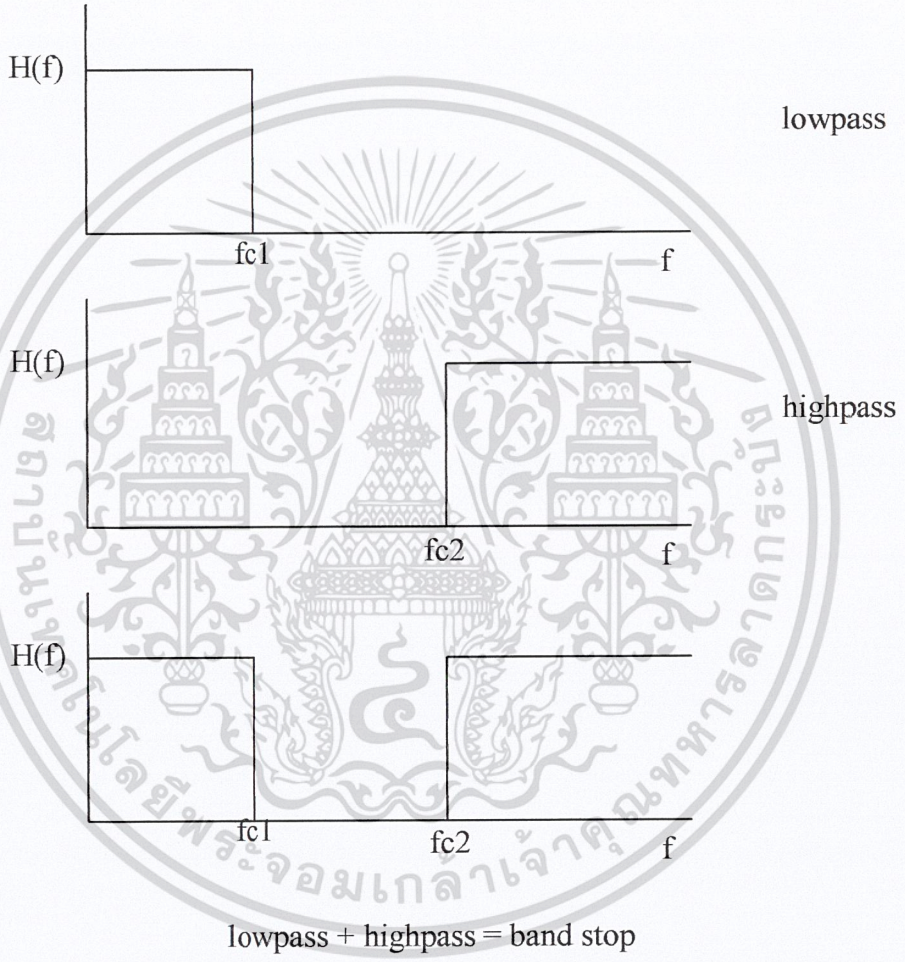
ซึ่งจากรูปที่ 3-27 เราจะได้ว่า

$$h_d(n) = \delta(n) - \sin\left(\frac{\omega_c(n-\alpha)}{\pi(n-\alpha)}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การออกแบบวงจรกรองความถี่แบบหยุดแถบบางความถี่ (Band stop design)

ลักษณะจะเป็นเช่นเดียวกับวงจรแถบความถี่แต่แตกต่างกันที่เราจะออกแบบวงจกรองความถี่แบบต่ำผ่านเพื่อกำหนด Lower cutoff frequency และออกแบบวงจรกรองความถี่สูงผ่านเพื่อกำหนด Upper cutoff frequency แล้วนำมาบวกกัน ดังรูปที่ 3-28



รูปที่ 3-28 แสดงการหาวงจรกรองความถี่แบบหยุดความถี่

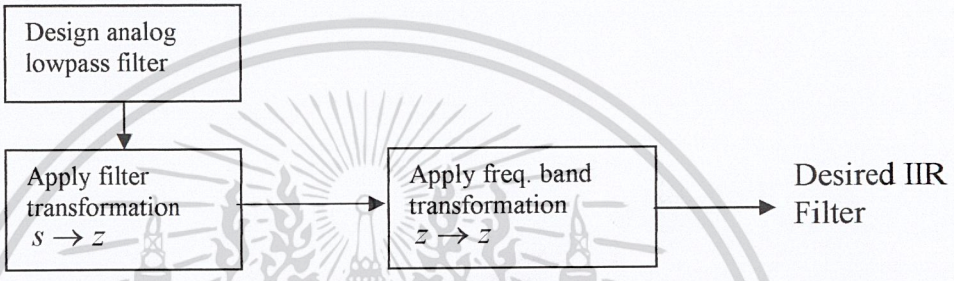
จะได้ว่า

$$h_d(n) = \sin\left(\frac{\omega_{c1}(n-\alpha)}{\pi(n-\alpha)}\right) + \delta(n) - \sin\left(\frac{\omega_{c2}(n-\alpha)}{\pi(n-\alpha)}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบวงจรกรองความถี่แบบ IIR (IIR Filter Design)

วงจรกรองความถี่แบบ IIR มีผลตอบสนองอิมพัลส์เป็นอนันต์ จึงสามารถใช้คุณสมบัติจากวงจรกรองความถี่เชิงอุปมาน (Analog filter) มาร่วมในการออกแบบ ซึ่งเป็นเทคนิคอย่างง่าย จากวงจรกรองความถี่เชิงอุปมานเปลี่ยนเป็นวงจรกรองความถี่เชิงเลข (Digital Filter) โดยใช้วิธีการ complex-valued mapping วิธีนี้เราสามารถได้ใกล้เคียงกับความเป็นจริงมากที่สุด วิธีการนี้เรียกว่า A/D filter Transform โดยมีวิธีการอย่างคร่าวดังรูปที่ 3-29



รูปที่ 3-29 block diagram แสดงถึงการออกแบบ IIR filter

จากรูปที่ 3-29 วงจรกรองความถี่แบบ IIR มีเทคนิคในการออกแบบโดยมีขั้นตอนดังนี้

- ออกแบบวงจรกรองความถี่เชิงอุปมานแบบความถี่ต่ำผ่าน
- Transform จากวงจรกรองความถี่เชิงอุปมานให้เป็น digital lowpass filter
- ออกแบบวงจรกรองความถี่ชนิดอื่นจาก digital lowpass filter

ปัญหาสำคัญคือไม่สามารถควบคุมคุณสมบัติทางเฟสของวงจรกรองความถี่แบบ IIR ได้ ดังนั้นพิจารณาเฉพาะ Magnitude response ในการออกแบบเราจะพิจารณาจากวงจรกรองความถี่เชิงอุปมานจากคุณสมบัติจากวงจรกรองความถี่ 3 แบบ คือ Butterworth, Chebyshev และ Elliptic และจากนั้นเราจะแปลงให้อยู่ในวงจรกรองความถี่แบบต่างๆในรูปแบบวงจรกรองความถี่เชิงเลข

3.3.1 หลักพื้นฐานของวงจรกรองความถี่เชิงอุปมาน

สิ่งที่จะพิจารณามีสิ่งที่สำคัญ 2 ประการคือพิจารณาลักษณะ Magnitude-squared response แต่ละชนิดมีความแตกต่างกันอย่างไร ซึ่งถูกแทนด้วยแบบ relative linear scale และ คุณสมบัติของ Magnitude-squared response

1. Relative Linear Scale

ให้ $H_a(j\Omega)$ แทนด้วยผลตอบสนองความถี่ของวงจรกรองความถี่เชิงอุปมาน แล้ววงจรกรองความถี่แบบต่ำผ่านจะมีลักษณะในรูปแบบของ magnitude-squared response ดังนี้

$$\frac{1}{1+\varepsilon^2} \le |H_a(j\Omega)|^2 \le 1, \qquad |\Omega| \le \Omega_p$$

$$0 \le |H_a(j\Omega)|^2 \le \frac{1}{A^2}, \qquad \Omega_s \le |\Omega|$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย ε เป็นripple parameterในส่วนของ passband และ Ω_p เป็น passband cutoff frequency ในหน่วย $rad/\sec$ และ A เป็น attenuation parameter ในช่วง stopband และ Ω_s เป็น stopband cutoff frequency โดยมีลักษณะดังรูปที่ 3-30 และโดยที่ $|H_a(j\Omega)|^2$ ถูกกำหนดให้เป็นลักษณะดังนี้

$$|H_a(j\Omega_p)|^2 = \frac{1}{1+\varepsilon^2} \quad at \quad \Omega = \Omega_p$$

$$|H_a(j\Omega_s)|^2 = \frac{1}{A^2} \quad at \quad \Omega = \Omega_s$$

โดยที่ Parameters ε และ A มีความสัมพันธ์กับ R_p และ A_s ในหน่วยของ dB ดังนี้

$$R_p = -10\log\frac{1}{1+\varepsilon^2} \Rightarrow \varepsilon = \sqrt{10^{R_p/10}-1}$$

และ

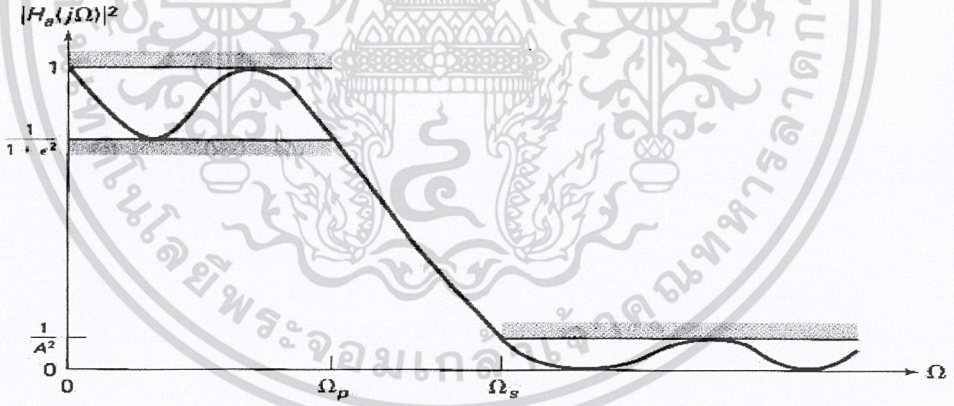
$$A_s = -10\log\frac{1}{A^2} \Rightarrow A = 10^{A_s/20}$$

และ ขนาดของ ripple δ_1 และ δ_2 ในเทอมของ ε และ A ดังนี้

$$\frac{1-\delta_1}{1+\delta_2} = \sqrt{\frac{1}{1+\varepsilon^2}} \Rightarrow \varepsilon = \frac{2\sqrt{\delta_1}}{1-\delta_1}$$

และ

$$\frac{\delta_2}{1+\delta_1} = \frac{1}{A} \Rightarrow A = \frac{1+\delta_1}{\delta_2}$$



รูปที่ 3-30 Analog low pass filter specifications

2. Properties of $|H_a(j\Omega)|^2$

สารถหาค่าของ s-domain ของระบบได้มีฟังก์ชั่นเป็น $H_a(s)$ ดังนี้

$$H_a(j\Omega) = H_s(s)\big|_{s=j\Omega}$$

จะได้ว่า

$$|H_a(j\Omega)|^2 = H_a(j\Omega)H_a^*(j\Omega) = H_a(j\Omega)H_a(-j\Omega) = H_a(s)H_a(-s)\big|_{s=j\Omega}$$

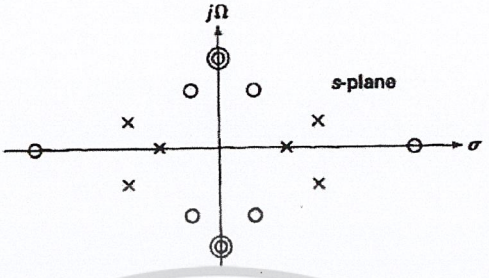
ดังนั้น

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2\big|_{\Omega=s/j}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาโพลและซีโลของ Magnitude-squared function สามารถกระจายในลักษณะของ mirror-image symmetry ในแกน $j\Omega$ หรือ conjugate pair ซึ่งแสดงให้เห็นดังรูปที่ 3-31 และถ้า

ต้องการให้ วงจรกรองความถี่มีลักษณะเป็น causal และ stable จะต้องสร้างให้ โพลของ $H_a(s)$ อยู่ของทางด้านครึ่งซ้ายของระนาบ



รูปที่ 3-31 typical pole-zero pattern of $H_s(s)H_s(-s)$

3.3.2 คุณสมบัติของต้นแบบของวงจรกรองความเชิงอุปมาน

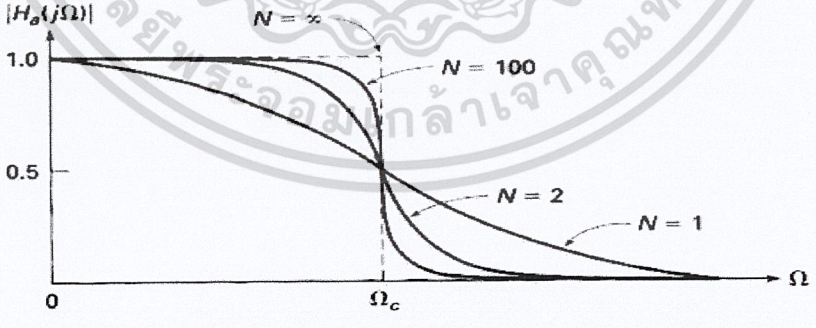
(Characteristics of prototype Analog filters)

1. Butter-worth lowpass filter

วงจรกรองความถี่แบบนี้จะมีผลตอบสนองที่เป็นเรียบทั้ง lowpass และ stoppass โดย magnitude-squared response ของวงจรกรองความถี่แบบต่ำผ่านอันดับที่ N กำหนดให้เป็น

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

โดยที่ N เป็นอันดับของวงจรกรองความถี่ และ Ω_c เป็น cutoff frequency ในหน่วย *rad* / sec เราสามารถพล็อตกราฟได้ดังรูปที่ 3-32



รูปที่ 3-32 Magnitude-squared response of butter-worth lowpass filter

จะได้ฟังก์ชั่นของระบบ $H_a(s)$ เป็น

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2\big|_{\Omega=s/j} = \frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{(j\Omega)^{2N}}{s^{2N}+(j\Omega_c)^{2N}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รากของ denominator ของ polynomial (โพลของ $H_a(s)H_a(-s)$ จะได้ว่า

$$p_k = (-1)^{\frac{1}{2N}}(j\Omega) = \Omega_c e^{j\frac{\pi}{2N}(2k+N+1)}, \quad k = 0,1,...,2N-1$$

สมการออกแบบวงจรกรองความถี่แบบ Butter-worth

จากวงจรกรองความถี่ต่ำผ่านต้องมีการกำหนดพารามิเตอร์ต่าง คือ Ω_p, R_p, Ω_s และ A_s เพื่อเป็นตัวกำหนดค่าของ N ซึ่งเป็น อันดับของวงจรกรองความถี่และ cutoff frequency Ω_c จะได้ว่า

ที่ $\Omega = \Omega_p, -10\log_{10}\left|H_a(j\Omega)\right|^2 = R_p$ หรือ $-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}}\right) = R_p$

ที่ $\Omega = \Omega_s, -10\log_{10}\left|H_a(j\Omega)\right|^2 = A_s$ หรือ $-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_s}{\Omega_c}\right)^{2N}}\right) = A_s$

จากทั้ง 2 สมการเราสามารถหา N และ Ω_c จะได้ว่า

$$N = \frac{\log_{10}\left[\left(10^{R_p/10}-1\right)/\left(10^{A_S/10}-1\right)\right]}{2\log_{10}\left(\Omega_p/\Omega_s\right)}$$

โดยทั่วไปแล้วค่า N จะไม่ใช่จำนวนเต็ม แต่ต้องการค่าที่เป็นจำนวนเต็มโดยการประมาณค่า โดยเลือกจากที่น้อยที่สุดจากมากที่สุดของ N และ หา Ω_c จะได้ว่า

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{\left(10^{R_P/10}-1\right)}}$$

หรือ

$$\Omega_c = \frac{\Omega_s}{\sqrt[2N]{10^{A_S/10}-1}}$$

2. Chebyshev Lowpass Filter

วงจรกรองความถี่แบบนี้ เราจะพิจารณา ripple ใน band มาก ซึ่งจะได้ จำนวนอันดับของวงจรกรองความถี่ที่ต่ำกว่าแบบ Butterworth เมื่อมีคุณสมบัติที่คล้ายกัน

และกำหนดให้ Magnitude-squared response ของ chebyshev filter เป็น

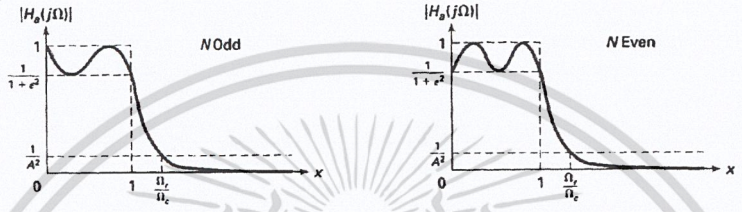
$$\left|H_a(j\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 T_N^2\left(\frac{\Omega}{\Omega_c}\right)}$$

โดยที่ N เป็นอันดับของวงจรกรองความถี่, ε เป็น passband ripple factor ในขณะที่ความสัมพันธ์ระหว่าง R_p และ $T_N(x)$ เป็น N th-order Chebyshev polynomial ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_N(x)=\begin{cases}\cos\left(N\cos^{-1}(x)\right), & 0\le x\le 1\\ \cosh(\cosh^{-1}(x)), & 1<x<\infty\end{cases}\qquad where\quad x=\frac{\Omega}{\Omega_c}$$

rippleที่ได้จะเป็นไปตาม $T_N(x)$ โดยที่ ถ้า $0<x<1$ แล้ว $T_N(x)$ จะแกว่งอยู่ระหว่าง -1 ถึง 1 และถ้า $1<x<\infty$ จะเพิ่มแบบ monotonically ถึง ∞ โดยที่ $|H_a(j\Omega)|^2$ มีรูปร่าง 2 รูปที่เป็นไปได้ คือในกรณีที่ N เป็นคี่และเป็นคู่ดังที่แสดงรูปที่ 3-33 และ $x=\Omega/\Omega_c$ เป็นnormalized frequency



รูปที่ 3-33 magnitude-squared response of Chebyshev lowpass filter

ในการหา causal และ stable ของ $H_a(s)$ จะต้องหาจากตำแหน่งของโพลที่อยู่ทางซ้ายของระนาบ S จาก $H_a(s)H_a(-s)$ โดยการหาจากรากของสมการ

$$1+\varepsilon^2 T_N^2\left(\frac{s}{j\Omega_c}\right)$$

ผลที่ได้จากสมการนี้เราเขียนอยู่ในรูปของ $p_k=\sigma_k+j\Omega_k$ โดยที่ $k=0,1,\ldots,N-1$ แล้ว

$$\sigma_k=(a\Omega_c)\cos\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]$$
$$\Omega_k=(b\Omega_c)\sin\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]\qquad k=0,\ldots,N-1$$

เมื่อ $a=\frac{1}{2}\left(\sqrt[N]{\alpha}-\sqrt[N]{1/\alpha}\right)$, $b=\frac{1}{2}\left(\sqrt[N]{\alpha}+\sqrt[N]{1/\alpha}\right)$ $\alpha=\frac{1}{\varepsilon}+\sqrt{1+\frac{1}{\varepsilon^2}}$

จะได้ว่า $H_a(s)=\dfrac{K}{\prod_k(s-p_k)}$

เมื่อ K ค่าที่เกิดจากการ normalizing ที่เลือกจาก

$$H_a(j0)=\begin{cases}1, & N\ odd\\ \dfrac{1}{\sqrt{1+\varepsilon^2}} & N\ even\end{cases}$$

สมการการออกแบบวงจรกรองความถี่แบบ chebyshev

ในการออกแบบจะมีการกำหนด Ω_p,Ω_s,R_p และ A_s ใน chebyshev-I จะต้องหาพารามิเตอร์ 3 ตัว คือ ε,Ω_c และ N จะได้ว่า

$$\varepsilon=\sqrt{10^{0.1R_p}-1}\qquad \text{และ}\qquad A=10^{A_s/20}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคุณสมบัติของ chebyshev จะได้ว่า

$$\Omega_c = \Omega_p \qquad \text{และ } \Omega_r = \frac{\Omega_s}{\Omega_p}$$

และหา อันดับของวงจรกรองความถี่จะได้ว่า

$$g = \sqrt{\left(A^2 - 1\right)/\varepsilon^2}$$

$$N = \left\lceil \frac{\log_{10}\left(g + \sqrt{g^2 - 1}\right)}{\log_{10}\left(\Omega_r + \sqrt{\Omega_r^2 - 1}\right)} \right\rceil$$

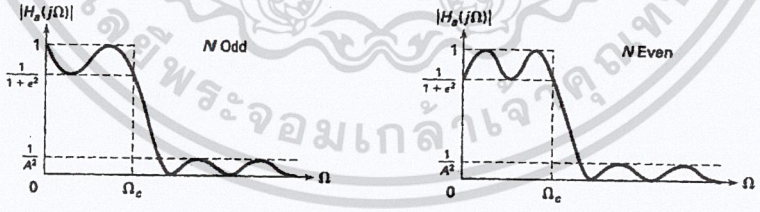
จากนั้นจะหา $H_a(s)$ ได้

3. Elliptic lowpass filter

ในวิธีนี้จะมี ripple ที่เกิดขึ้นทั้ง passband และ stopband ซึ่งจะมีลักษณะคล้ายกับแบบ FIR โดยที่ magnitude-squared response ของ elliptic filter ให้เป็น

$$\left|H_a(j\Omega)\right|^2 = \frac{1}{1 + \varepsilon^2 U_N^2\left(\frac{\Omega}{\Omega_c}\right)}$$

เมื่อ N เป็นอันดับของวงจรกรองความถี่ ε เป็น passband ripple และ $U_N(\cdot)$ เป็น N th order Jacobian elliptic function ซึ่งผลที่ได้จากสมการเป็นดังรูปที่ 3-34



รูปที่ 3-34 Magnitude-squared response of elliptic filter

ในการวิเคราะห์สมการ เป็นเรื่องที่ยากดังนั้นจึงไม่กล่าววิธีการพิสูจน์ไว้ผลที่ได้ดังนี้

$$N = \frac{K(k)K\left(\sqrt{1-k_1^2}\right)}{K(k_1)K\left(\sqrt{1-k^2}\right)}$$

เมื่อ $k = \dfrac{\Omega_p}{\Omega_s}$, $k_1 = \dfrac{\varepsilon}{\sqrt{A^2-1}}$ และ $K(x) = \displaystyle\int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - x^2\sin^2\theta}}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

3.3.3 Analog-to-digital filter transformations

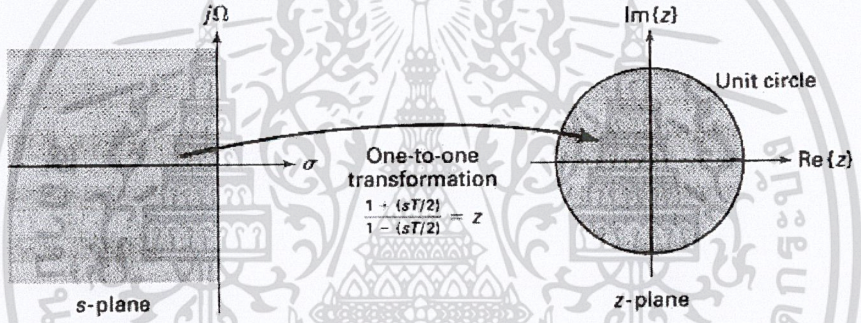
การที่ได้ออกแบบวงจรกรองความถี่เชิงอุปมานจากนั้นต้องมีการแปลงให้อยู่ในรูปของวงจรกรองความถี่เชิงเลข โดยใช้วิธีการ Complex-valued mapping การแปลงรูปแบบนี้ยังรักษาลักษณะของรูปร่างที่ได้ออกแบบไว้ในการออกแบบเชิงอุปมาน โดยในหัวข้อนี้จะมีการพิจารณาการแปลง 2 รูปแบบคือ impulse invariance transformation และ bilinear transformation

3.3.4 Bilinear transformation

การแปลงรูปแบบนี้เป็นวิธีที่ให้มากที่สุด โดยสมการการแปลงเป็น

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \Rightarrow z = \frac{1+sT/2}{1-sT/2}$$

ซึ่งการ mapping แสดงได้ดังรูปที่ 3-35



รูปที่ 3-35 Complex-plane mapping in bilinear transformation

จากรูปที่ 3-35 จะสังเกตพบว่า

1. ให้ $s = \sigma + j\Omega$ จะได้ว่า

$$z = \left(1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}\right) \Big/ \left(1+\frac{\sigma T}{2}-j\frac{\Omega T}{2}\right)$$

ดังนั้น

$$\sigma < 0 \Rightarrow |z| = \left|\frac{1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}}{1-\frac{\sigma T}{1}-j\frac{\Omega T}{2}}\right| < 1$$

$$\sigma = 0 \Rightarrow |z| = \left|\frac{1+j\frac{\Omega T}{2}}{1-j\frac{\Omega T}{2}}\right| = 1$$

$$\sigma > 0 \Rightarrow |z| = \left|\frac{1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}}{1-\frac{\sigma T}{2}-j\frac{\Omega T}{2}}\right| > 1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เป็นการ mapจากฝั่งซ้ายในระนาบs เข้าไปใน unit circle ดังนั้นเป็นการแปลงแบบ stable

3. เป็นการแปลงแบบ One-to-one mapping

แทน $\sigma = 0$ จะได้ว่า

$$z = \frac{1 + j\frac{\Omega T}{2}}{1 - j\frac{\Omega T}{2}} = e^{j\omega}$$

ถ้า magnitude เป็น 1 หา ω ที่เป็นฟังก์ชั่นของ Ω จะได้ว่า

$$\omega = 2\tan^{-1}\left(\frac{\Omega T}{2}\right) \qquad \text{หรือ} \qquad \Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right)$$

ขั้นตอนการออกแบบ

การออกแบบจะต้องกำหนด ให้ ω_p, ω_s, R_p และ A_s ต้องการหา $H(z)$ โดยมีขั้นตอนดังนี้

1. เลือกค่า T อาจจะสมมติให้เป็น 1 ก็ได้
2. หา Ω_p, Ω_s จาก

$$\Omega_p = \frac{2}{T}\tan\left(\frac{\omega_p}{2}\right), \qquad \Omega_s = \frac{2}{T}\tan\left(\frac{\omega_s}{2}\right)$$

3. ออกแบบวงจรกรองความถี่เชิงอุปมาน $H_a(s)$ จากคุณสมบัติ Ω_p, Ω_s, R_p และ A_s
4. สุดท้ายแทนค่าด้วย

$$H(z) = H_a\left(\frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

3.3.5 Frequency-Band Transformations

ก่อนหน้านี้เราออกแบบตัวกรองความถี่ต่ำผ่านจากวงจรกรองความถี่เชิงอุปมานถ้าต้องการออกแบบวงจรกรองความถี่แบบอื่นๆ เช่น ความถี่สูงผ่าน,ช่วงความถี่ผ่าน, และช่วงหยุดความถี่ผ่านจะทำได้ด้วยการแปลงแกนความถี่ของกรองความถี่ต่ำดังนั้นมันปฏิบัติตัวเหมือนตัวกรองแบบเลือกความถี่ การแปลงบน complex variable z เหมือนbilinear transform และสมการการออกแบบคือ algebraic การปฏิบัติที่จะออกแบบตัวกรองเลือกความถี่ผ่านเป็นอย่างแรกที่จะออกแบบ digital prototype low pass filter และจากนั้นประยุกต์การแปลงalgebraic จะอธิบายตรรกะพื้นฐานหลังการ mapping และ การรวมการแปลงช่วงความถี่ผ่านใน s-plane การแสดงการใช้z-plane mapping แต่ละชนิดพื้นฐานที่สำคัญของตัวกรองสัญญาณเลือกความถี่ผ่านแสดงดังรูป 3-36ให้ $H_{LP}(Z)$ เป็นข้อมูลพื้นฐานที่ออกมาจากตัวกรองความถี่ต่ำผ่านแบบดิจิตอลและให้ H(z) เป็นตัวกรองความถี่เลือกที่ต้องการ สังเกตว่าเราใช้ค่าที่ต่างกันระหว่าง Z และ z ด้วย $H_{LP}(Z)$ และ H ตามดับจำกัดความ mapping โดย

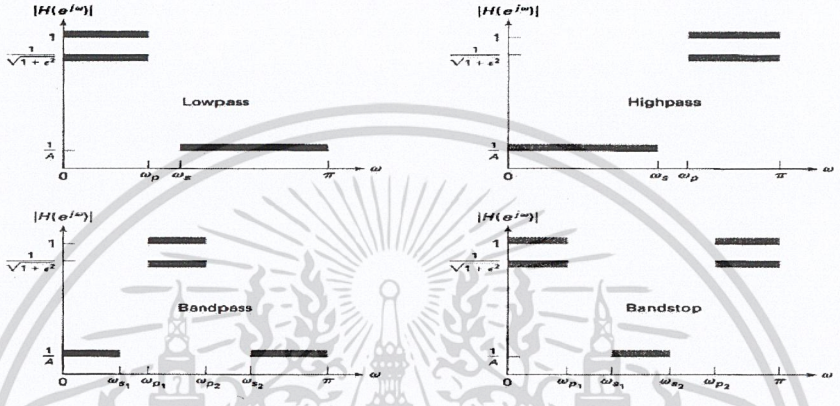
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Z^{-1} = G\left(z^{-1}\right)$$

เช่น

$$H(z) = H_{LP}(Z)\Big|_{z^{-1}=G(z-1)}$$

เพื่อที่จะทำแบบนี้เราสมมติแทน Z^{-1} ในทุกๆ H_{LP} โดยฟังก์ชั่น $G\left(z^{-1}\right)$ ดังนั้น $H_{LP}(Z)$เป็นค่าคงที่และเป็นcausal filter เรามักจะต้องการ $H(z)$ คงที่และcausal มันจะถูกกำหนดดังนี้



รูปที่ 3-36 ชนิดของfilterแบบต่างๆ

1. $G(\cdot)$ จะเป็นสัมประสิทธิ์การขยายของฟังก์ชั่น z^{-1} ดังนั้น $H(z)$เป็นเครื่องมือ
2. วงกลมหนึ่งหน่วยของ Z-plane จะmapบนวงกลมหนึ่งหน่วยของ z-plane
3. สำหรับตัวกรองที่คงที่ ภายในวงกลมหนึ่งหน่วยของ Z-plane จะต้อง map ภายในวงกลมหนึ่งหน่วยของ z-plane

ให้ ω' และ ω เป็นตัวแปรความถี่ของ Z และ z โดยลำดับ จะได้ $Z = e^{j\omega'}$ และ $z = e^{j\omega}$ ตามลำดับของวงกลมหนึ่งหน่วยดังข้อกำหนดทั้งสองจะได้

$$\left|Z^{-1}\right| = \left|G\left(z^{-1}\right)\right| = \left|G\left(e^{-j\omega}\right)\right| = 1$$

และ

$$e^{-j\omega'} = \left|G\left(z^{-1}\right)\right| e^{j\angle G\left(e^{-j\omega}\right)}$$

หรือ

$$-\omega' = \angle G\left(e^{-j\omega}\right)$$

รูปแบบโดยทั่วไปของฟังก์ชั่น $G(\cdot)$ ที่เป็นที่พึงพอใจตามความต้องการ คือ อัตราส่วนของชนิด all-pass โดย

$$Z^{-1} = G\left(z^{-1}\right) = \pm\prod_{k=1}^{n} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}}$$

ถ้า $|\alpha_k| < 1$ สำหรับการคงที่และเพื่อความต้องการในข้อ 3

การเลือกลำดับ n ที่เหมาะสมและ $\{\alpha_k\}$, เราสามารถเพิ่มค่าของการmapping อย่างกว้างสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การสร้างวงจรกรองความความถี่เชิงเลขโดยใช้ TMS320C26

(Implementation of FIR/IIR Filter with TMS320C26)

การสร้างวงจรกรองความถี่เชิงเลข (digital filter) ทั้งแบบระบบตอบสนองตัวอย่างเดี่ยวเชิงจำกัด (Finite Impulse Response, FIR) และ แบบระบบตัวตอบสนองตัวอย่างเดียวไม่จำกัดจำนวน (Infinite Impulse Response, IIR) โดยใช้ TMS320C26 เป็นตัวประมวลผล

4.1 ขั้นตอนการออกแบบวงจรกรองความถี่เชิงเลข (Design of Digital filter)

1. การประมาณค่า(Approximate) คือขบวนการสร้างฟังก์ชั่นถ่ายโอน (Transfer functions) ให้ได้ตามที่ออกแบบวงจรไว้ที่ประกอบด้วย ผลตอบสนองทางเวลา (Time response), ผลตอบสนองทางความถี่ (Frequency response) ของวงจรกรองความถี่เชิงเลข

2. การสร้างให้วงจรให้เป็นจริง(Realization) ประกอบด้วยการเปลี่ยนจากฟังก์ชั่นถ่ายโอน (Transfer functions) มาเป็นวงจรกรองความถี่ ในบทนี้จะกล่าวถึงการออกแบบ โครงสร้างดังนี้

2.1 โครงสร้างแบบตรงแบบที่ 1 (Direct-form I)

2.2 โครงสร้างแบบตรงแบบที่ 2 (Direct-form II)

2.3 โครงสร้างแบบเทียมเดี่ยว (Cascade form)

3. การศึกษาผลของความคาดเคลื่อนจากการเขียนโปรแกรม(Study of arithmetic errors) และการสร้างโปรแกรมสำหรับวงจรกรองความถี่เชิงเลข (Implementation)

4.2 การสร้างโปรแกรมสำหรับวงจรกรองความถี่เชิงเลข (Digital Filter Implementation)

จากการประยุกต์การใช้งานที่หลากหลายของวงจรกรองความถี่ทำให้เราสามารถเขียนสมการเพื่อแสดงความสัมพันธ์ระหว่างลำดับของอินพุท (Input Sequence, $x(n)$) กับลำดับของเอาท์พุท (Output sequence, $y(n)$) ได้ดังนี้

$$y(n) = \sum_{k=0}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (4.1)$$

จากสมการที่ 4.1 เป็นสมการผลต่างที่สัมประสิทธิ์มีค่าคงที่แบบเชิงเส้น (Linear constant-coefficient difference equation) โดยที่ตัวกรองความถี่เชิงเลขทั้ง 2 ชนิด มีคุณสมบัติเป็นสมการผลต่างที่สัมประสิทธิ์ที่ค่าที่แบบเชิงเส้น คือ

1. วงจรกรองความถี่เชิงเลขแบบระบบตอบสนองตัวอย่างเดี่ยวเชิงจำกัด (Finite Impulse Response, FIR)

2. วงจรกรองความถี่เชิงเลขแบบระบบตัวตอบสนองตัวอย่างเดียวไม่จำกัดจำนวน (Infinite Impulse Response, IIR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การสร้างวงจรกรองความถี่เชิงเลขแบบระบบตัวอย่างเดี่ยวเชิงจำกัด (FIR Filter Design)

สำหรับตัวกรองความถี่แบบFIR ค่าสัมประสิทธิ์ของ a_k ในสมการที่ 4.1 สามารถเขียนใหม่ได้เป็น

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{4.2}$$

โดยที่ $(M+1)$ เป็นความยาวของวงจรกรองความถี่ (length of the filter)

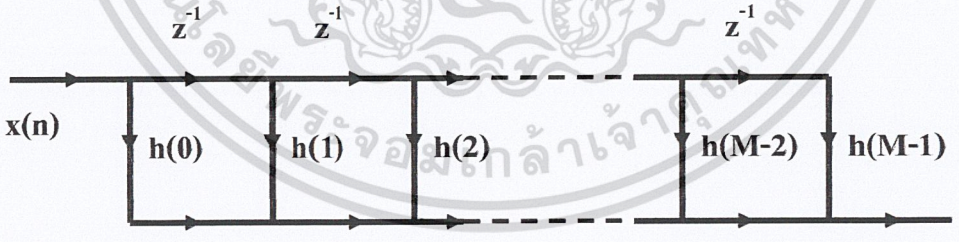
และเอาท์พุทที่ได้จากวงจรกรองความถี่ชนิดนี้ เกิดจากผลบวกของผลคูณของค่าสัมประสิทธิ์ b_k กับลำดับอินพุทปัจจุบัน และลำดับอินพุทที่เกิดก่อนอินพุทปัจจุบัน และ ถ้าผลตอบสนองอิมพัลส์ (Impulse response) ของตัวกรองความถี่เขียนได้เป็น $h(n)$ ดังนั้นจากสมการที่ 4.2 เขียนใหม่ได้เป็น

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k) \tag{4.3}$$

จากสมการที่ 4.3 จะเห็นได้ว่าวงจรกรองความถี่แบบFIR มีผลตอบสนองอิมพัลส์ ที่มีความยาวที่จำกัด และถ้าเราแปลงให้อยู่ในรูปของ Z โดยที่ $x(n), y(n)$ และ $h(n)$ แปลงแล้วอยู่ในรูป $X(Z), Y(Z)$ และ $H(Z)$ ตามลำดับ แล้วจะได้สมการฟังก์ชั่นถ่ายโอนในรูป Z เป็น

$$H(Z) = \frac{Y(Z)}{X(Z)} = \sum_{k=0}^{M} b_k Z^{-k} = \sum_{k=0}^{M} h(k)Z^{-k} \tag{4.4}$$

จากสมการที่ 4.3 และ 4.4 สามารถเขียนโครงสร้างได้ดังรูปที่ 4-1



รูปที่ 4-1 โครงสร้างของแบบตรงแบบที่ 1 ของ FIR filter

จากโครงสร้างดังกล่าวเป็นโครงสร้างแบบตรงแบบที่ 1 และ เนื่องจากสัมประสิทธิ์ของตัวกรองความถี่สามารถต่อได้โดยตรงกับผมการผลต่าง(สมการที่ 4.3) และเส้นที่กำกับด้วย Z^{-1} ในรูปที่ 4-1 เป็นการหน่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเขียนโปรแกรมของตัวกรองความถี่แบบ FIR ด้วย TMS320C26

(TMS320C26 Implementation of FIR filter)

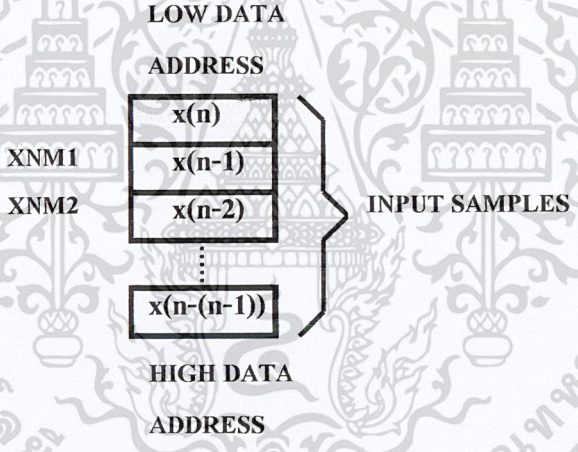
ในการเขียนโปรแกรมให้กับวงจรกรองความถี่แบบ FIR จะมีหลักพื้นฐานที่สำคัญ 3 ประการของ TMS320C26 ที่ควรจะเข้าใจคือ

1. ความสัมพันธ์ระหว่างผลตอบสนองอิมพัลส์ของวงจรกรองความถี่กับโครงสร้างวงจรกรองความถี่แบบ FIR

2. ประโยชน์ของการให้คำสั่ง LTD และ MPY ควบคู่กัน

3. ลำดับของสัญญาณสุ่มสัญญาณของอินพุทในหน่วยความจำข้อมูลในTMS320C26 เพื่อใช้ในการประมวลผลแบบคาบเวลาจริง

ลำดับของอินพุทจะถูกเก็บในหน่วยความจำข้อมูลดังรูปที่ 4-2 โดยที่ $x(n)$ แทนด้วย XN และ $x(n-1)$ แทนด้วย XNM1



รูปที่ 4-2 แสดงถึงการเก็บข้อมูลอินพุทในหน่วยความจำข้อมูล

จากสมการที่ 4.3 ในแต่ละอินพุทจะถูกการคูณแล้วเลื่อนค่าไปทีละหนึ่ง เสมอ ดังนั้นสามารถเขียนโปรแกรมบางส่วนได้เป็น

```
LTD    XNM1
MPY    H1
```

โดยที่คำสั่ง LTD XNM1 จะเป็นคำสั่งในการโหลดค่า T- resister ด้วยค่าในตำแหน่งที่ (Address) XNM1 และ บวกค่าผลที่ได้หลังจากการคูณกับ Accumulator จากนั้นจึงเลื่อนค่าของข้อมูลในตำแหน่งที่มากกว่าXNM1 ในหน่วยความจำข้อมูล ถ้าการกำหนดตำแหน่งของข้อมูลอินพุทเป็นดังรูปที่ 4-2 ผลที่ได้จากคำสั่งที่แสดงไว้ด้านบนคือ การเลื่อนข้อมูลจากตำแหน่งที่XNM1 ไปที่ XNM2 หลังจากการโหลดข้อมูลจากXNM1 ไปที่ T-register แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในคำสั่ง MPY H1 จะเป็นคำสั่งในการคูณเลขระหว่างค่าใน T-register กับ ค่าในตำแหน่งที่ H1

ในการใช้รูปแบบนี้ในการประมวลผลแบบเวลาจริงนั้นต้องใช้อย่างระมัดระวังในเรื่องตำแหน่งที่ถูกต้องของอินพุทในลำดับถัดมาของตัวกรองความถี่

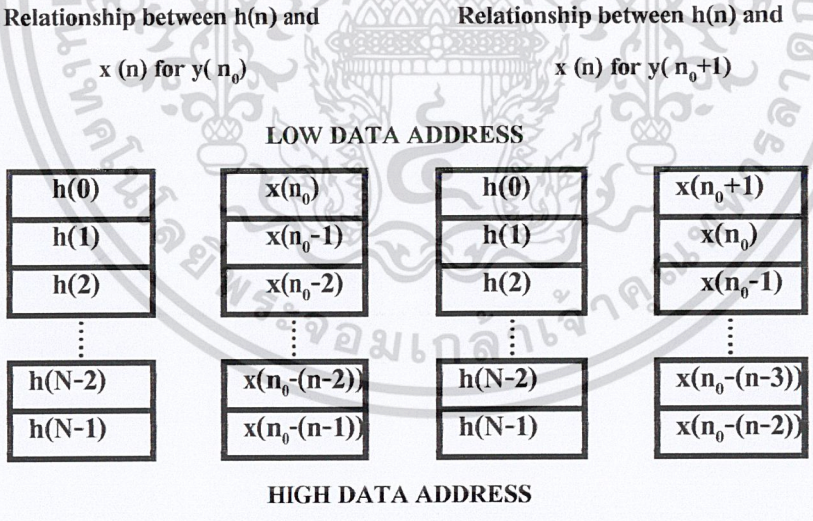
เพื่อเพิ่มความเข้าใจรูปแบบในการเขียนโปรแกรมมากยิ่งขึ้น เราจะแสดงความสัมพันธ์ระหว่างลำดับอินพุทกับลำดับเอาท์พุทจากสมการที่ 4.3 โดยที่เราจะพิจารณาใน n แต่ละค่า ตัวอย่างเช่น n_0 จะได้ว่า

$$y(n_0) = \sum_{k=0}^{N-1} h(k)x(n_0 - k) \tag{4.5}$$

และถ้าเอาท์พุทลำดับถัดไป มีค่าเป็น $y(n_0 + 1)$ จากสมการที่ 5.3 จะได้ว่า

$$y(n_0 + 1) = \sum_{k=0}^{N-1} h(k)x(n_0 + 1 - k) \tag{4.6}$$

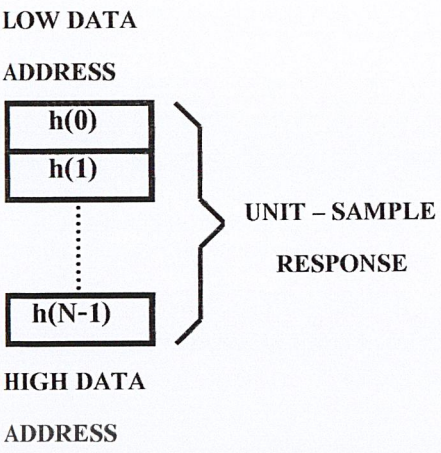
จากสมการที่4.5 และ 4.6 แสดง $x(n)$ ว่ามีความเกี่ยวข้องกับ $h(k)$ ในสมการที่ 4.5 ด้วยการเลื่อนค่าไปทางซ้าย(เลื่อนไปที่ ตำแหน่งที่สูดขึ้นของหน่วยความจำข้อมูล)ทีละหนึ่งดังสมการที่ 4.6 ในการเลื่อนข้อมูลอินพุทนี้เราสามารถแสดงการเลื่อนได้ดังรูปที่ 4-3



รูปที่ 4-3 แสดงความสัมพันธ์ของค่าในหน่วยความจำข้อมูล

ในการเขียนโปรแกรมนั้น ผู้เขียนอาจจะลดขนาดโปรแกรมที่เขียนในหน่วยความจำโปรแกรมโดยการใช้ประโยชน์จากการอ้างข้อมูลแบบทางอ้อมที่มีในTMS320C26 ซึ่งเป็นการใช้ Auxiliary Register ในการเพิ่มค่าหรือลดค่าอย่างอัตโนมัติ โดยเราจะเขียนโปรแกรมในลักษณะลูป ถ้าลำดับข้อมูลของอินพุธมีการเก็บดังรูปที่ 4-2 และ ผลตอบสนองอิมพัลล์มีการจัดเก็บดังรูปที่ 4-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 แสดงถึงการจัดเก็บผลการตอบสนองอิมพัลส์ในหน่วยความจำข้อมูล

ในการเขียนโปรแกรมแบบลูปโดยการใช้การอ้างถึงข้อมูลแบบทางอ้อมนั้นจะประโยชน์การลดค่าอย่างอัตโนมัติและการใช้คำสั่ง BANZ เพื่อที่จะควบคุมลูปและสร้างตำแหน่งของข้อมูลชุดใหม่ในการเข้าถึงหน่วยความจำข้อมูล ซึ่งในการใช้โปรแกรมในลักษณะลูปนี้จะมีข้อดีคือขนาดของโปรแกรมในหน่วยความจำโปรแกรมจะมีค่าน้อยกว่าการเขียนโดยวิธีตรง ๆ แต่ในการเขียนแบบตรงๆนั้นจะสามารถทำงานได้เร็วกว่าการเขียนแบบลูปเพราะว่าจะต้องมีการประกาศส่วนของในการควบคุมลูป ดังนั้นผู้ออกแบบควรระวังในเรื่องนี้เป็นอย่างยิ่ง

การประยุกต์การใช้งานของวงจรกรองความถี่แบบ FIR นั้นจะแบบการประมวลผลแบบเวลาจริง (real time processing) ซึ่งการใช้งานกับการทดลองนี้จะกำหนดให้มีอัตราการสุ่มข้อมูลประมาณ 10 KHz ซึ่งคิดเป็นเวลาในการสุ่มสัญญาณในแต่ละข้อมูลเป็น 100 μS แต่TMS320C26 สามารถประมวลผลวัฏจักรละ 100 nS ดังนั้นเราสามารถเขียนโปรแกรมได้สูงสุดที่ 1000 วัฏจักร

จากคุณสมบัติของ TMS320C26 ที่มีการจัดให้มีหน่วยความจำข้อมูลขนาด 1568 word ใน program / data memory และสามรถแบ่งได้เป็น 4 กลุ่ม คือ B0, B1, B2, B3 โดยที่ B0, B1, B3 มีขนาด 512 word และ B2 มีขนาด 32 word และในกลุ่มที่ B0, B1, B3 สามารถเป็นได้ทั้งหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล โดยใช้คำสั่ง CONF ในการการเลือกชนิดของหน่วยความจำตารางที่ 4-1

CONF Mode Decoding Table

CNF1	CNF0	B0	B1	B2	B3
0	0	data	data	data	data
0	1	program	data	data	data
1	0	program	program	data	data
1	1	program	program	data	program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-1 ตารางการควบคุมการเลือกชนิดของหน่วยความจำในTMS320C26

และการจะใช้ข้อดีของคำสั่ง MACD (multiply and accumulate with data move) โดยที่เรากำหนดให้ B0 จะเป็นหน่วยความจำโปรแกรมโดยใช้คำสั่ง CONF 1 และใช้คำสั่ง MACD เพียงคำสั่งทำงานเป็นหน่วยความจำที่อยู่ภายในตัวซิบ ซึ่งจะทำให้การประมวลผลเร็วขึ้น และช่วยในการขยายหน่วยความได้เพิ่มขึ้นอีก และจากการใช้คำสั่ง RPTK ควบคู่กับ MACD ในการเขียนโปรแกรมวนลูปช่วย จะได้ว่า

RPTK NM1

MACD (pma),(dma)

จากคำสั่งชุดข้างบนจะได้ว่า ในคำสั่ง RPTK เป็นการโหลดค่า 8 บิตแบบทันที และค่าจะลดลงที่ละหนึ่ง เก็บใน repeat counter ซึ่งจะทำให้การทำคำสั่งถัดไปเป็นจำนวน N ครั้งโดยที่ N เป็นความยาวของวงจรกรองความถี่ และคำสั่ง MACD (pma), (dma) จะมีขบวนการตามขั้นตอนในด้านล่างนี้

1. โหลด program counter ด้วยค่า pma
2. คูณค่าในหน่วยความจำข้อมูลที่ตำแหน่ง dma ด้วยค่าในหน่วยความจำโปรแกรมที่ตำแหน่ง pma
3. บวกค่าที่ผลที่ได้ก่อนหน้านี้ในAccumulator
4. ก็อบปี้ค่าในหน่วยความจำข้อมูลไปยังตำแหน่งของข้อมูลที่สูงกว่า ในการเลื่อนค่านี้เป็นขบวนการเขียน Z^{-1} delay
5. เพิ่มค่าใน program counter เพื่อที่จะคูณและบวกเลขในตำแหน่งที่ถัดไปของผลตอบสนองอิมพัลส์

ซึ่งเป็นการรวมการใช้งานระหว่างคำสั่ง LTP กับ MPY ในคำสั่งเดียว

4.5 การสร้างโปรแกรมของตัวกรองความถี่แบบ IIR ด้วย TMS320C26

จากที่ได้อธิบายมาแล้วในหัวข้อที่ 4.4 ซึ่งเป็นการเขียนโปรแกรมแบบ FIR นั้นเราจะนำมาประยุกต์ต่อในการใช้งานการเขียนโปรแกรมแบบ IIR แต่จากสมการที่ 4.3 ค่าสัมประสิทธิ์ a_k มีค่าอย่างต่ำเพียงหนึ่งค่าที่ไม่เท่ากับศูนย์ ดังนั้นถ้าเราเขียนอยู่รูปของการแปลง Z ของผลตอบสนองอิมพัลส์ของวงจรกรองความถี่เป็นIIR เราจะได้ความสัมพันธ์ดังนี้

$$H(Z)=\frac{Y(Z)}{X(Z)}=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{k=1}^{N} a_k z^{-k}} \qquad (4.7)$$

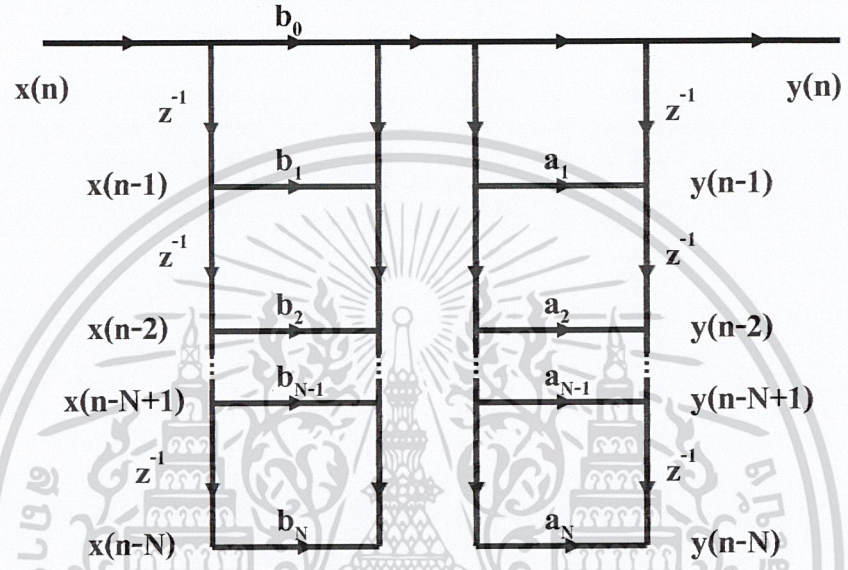
โดยที่ $H(Z), Y(Z)$ และ $X(Z)$ เป็นการแปลง Z ของ $h(n), y(n)$ และ $x(n)$ ตามลำดับ

โดยที่เราจะเขียนโปรแกรมให้มีโครงสร้างแบบ โครงสร้างแบบตรง แบบเทียมเดี่ยว ในรายงานนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

4.6 การสร้างโปรแกรมแบบโครงสร้างแบบตรงแบบที่ 2 ของวงจรกรองความถี่แบบ IIR (Direct-Form II IIR Filter)

จากสมการที่ 4.1 และ 4.7 เราจะแทนโครงสร้างดังรูปที่ 4-5



รูปที่ 4-5 โครงสร้างแบบตรงแบบที่ 1

เพื่อความสะดวกถ้าเราสมมุติให้ $M = N$ เราจะได้โครงสร้างแบบตรงแบบที่ 1(direct-form I) ซึ่งเป็นสมการผลต่างลำดับที่ N (N-order difference equation) โดยที่ สัมประสิทธิ์ของวงจรจะได้จากการต่อแบบตรงกับวงจรหน่วงเวลา

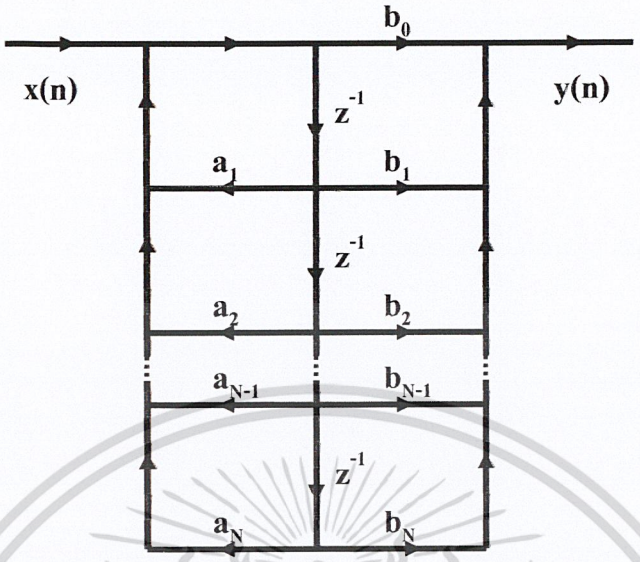
จากสมการผลต่าง

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (4.8)$$

ซึ่งแสดงให้เห็นว่าเอาท์พุทของวงจรกรองความถี่จะเป็นผลบวกของน้ำหนักค่าอินพุทและเอาท์พุทที่ผ่านมา จากวิธีการที่เหมือนกับวงจรกรองความถี่แบบ FIR เราสามารถให้วิธีการเดียวกันกับการเขียนโปรแกรมแบบ IIR

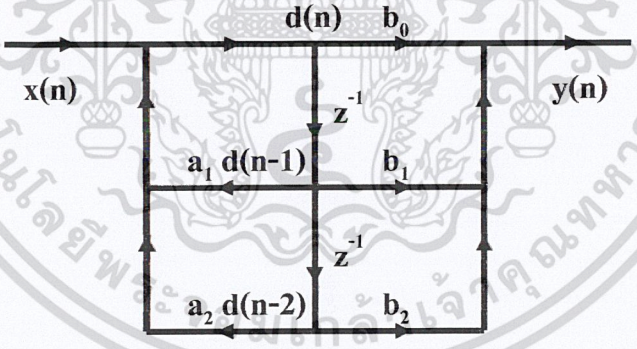
จากรูปที่ 11 เป็น Network flowgraph ของของสมการที่ 4.8 เราจะเขียนให้ให้อยู่ในรูปใหม่คือรูปที่ 4-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 โครงสร้างแบบตรงแบบที่ 2

จากรูปที่ 4-6 เป็นโครงสร้างแบบตรงแบบที่ 2 (direct-form II) ซึ่งมีวงจรการหน่วงเวลาน้อยลงจะช่วยในการลดจำนวนข้อมูลการจำน้อยลงสำหรับการคำนวณ โครงสร้างนี้ทำให้จำนวนการใช้หน่วยความจำข้อมูลน้อยลงจะใช้ประโยชน์ในการเขียนโปรแกรมวงจรกรองความถี่แบบ IIR



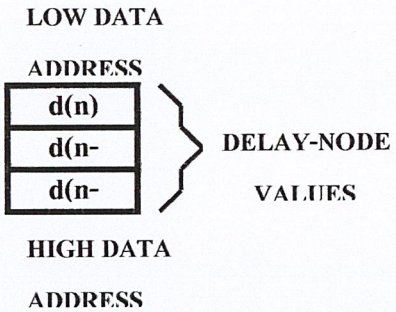
รูปที่ 4-7 วงจรกรองความถี่แบบ IIR ลำดับที่ 2 โครงสร้างแบบตรงแบบที่ 2

จากรูปที่ 4-7 เป็นโครงสร้างแบบตรงแบบที่ 2 ของวงจรกรองความถี่แบบ IIR ลำดับที่ 2 จะเป็นตัวอย่างในการออกแบบ เราจะได้สมการผลต่างของรูปที่ 4-7 เป็น

$$\begin{aligned} d(n) &= x(n) + a_1 d(n-1) + a_2 d(n-2) \\ y(n) &= b_0 d(n) + b_1 d(n-1) + b_2 d(n-2) \end{aligned} \tag{4.9}$$

จากสมการที่ 4.9 นี้และรูปที่ 4-7 $d(n)$ จะผลตอบสนองร่วมระหว่างค่าของวงจรกับจุดที่มีการหน่วงเวลา (delay-node values) โดยที่รีจิสเตอร์หน่วงเวลาที่ 0 แทนด้วย $d(n)$, รีจิสเตอร์หน่วงเวลาที่ 1 แทนด้วย $d(n-1)$ และ $d(n-2)$ จะแทนด้วยรีจิสเตอร์หน่วงเวลาที่ 2 ค่าของการหน่วงเวลานั้นจะถูกเก็บในหน่วยความจำข้อมูลดังรูปที่ 4-8

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 การเก็บของค่าของการหน่วงเวลาในหน่วยความจำข้อมูล

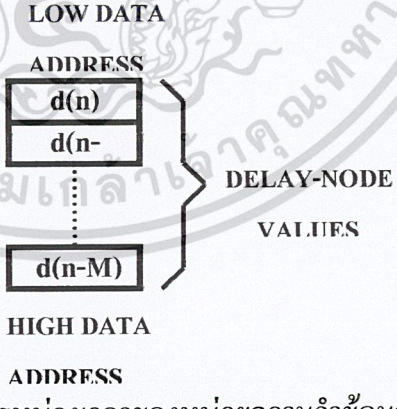
แต่ละขั้นจะเริ่มด้วยการคูณ ผลที่ได้จากผลการคูณก่อนหน้าจะบวกเข้ากับ Accumulator จากนั้นค่าการหน่วงเวลาจะถูกเลื่อนไปตำแหน่งที่สูงกว่าในหน่วยความจำข้อมูล ทั้งหมดสามารถใช้เพียง 2 คำสั่งคู่กัน คือ LTD DNM1

MPY B1

เมื่อ DNM1 แทนด้วย $d(n-1)$ และ B1 แทนด้วย b_1 ในสมการที่ 4.9

เมื่อผลการคูณสุดท้ายเสร็จแล้วและผลการบวกถูกเก็บใน Accumulator เรียบร้อยแล้วค่าจะถูกเก็บเข้าไปในหน่วยความจำ ในขณะที่ $y(n)$ จากสมการที่ 9 และ รูปที่ 4-7 เป็นค่าที่หาได้จากค่าของการหน่วงเวลา $d(n)$ และค่าการหน่วงเวลาก่อนหน้า ถือว่าเป็นระบบป้อนกลับโดยการใช้คำสั่ง

ลำดับค่าการหน่วงเวลาของการจัดที่อยู่ในหน่วยความจำข้อมูลดังรูปที่ 4-8 ซึ่งเป็นการเขียนโปรแกรมอย่างง่ายเพื่อใช้ในการคำนวณ และ ใช้หน่วยความจำข้อมูลที่น้อยที่สุด และโครงสร้างนี้มีความคล่องตัวและสามารถขยายสำหรับวงจรกรองความถี่แบบลำดับสูงได้



รูปที่ 4-9 การเก็บของค่าการหน่วงเวลาของหน่วยความจำข้อมูล

จากรูปที่ 4-9 แสดงให้เห็นความสำคัญของลำดับของค่าการหน่วงเวลาจากโครงสร้างแบบตรงแบบที่ 2 ในกรณี $M \geq N$ อันดับของวงจรกรองความถี่หาได้จาก M หรือ N ที่ค่ามากที่สุด

เราอาจจะใช้โปรแกรมจาก FIR จากคำสั่ง RPTK และ คำสั่ง MACD เมื่อช่วยในการเขียนโปรแกรม จากที่เราได้ออกแบบการเขียนโปรแกรมแบบโครงสร้างแบบตรงแบบที่ 2 โดยที่เป็นวงจรกรองความถี่อันดับ 2 จะถูกเรียกว่า Biquad implementation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

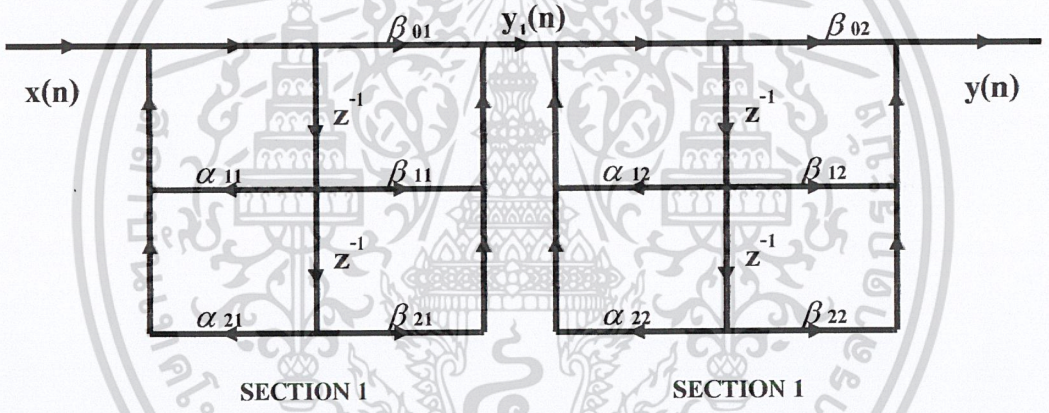
4.7 การออกแบบวงจรกรองความถี่แบบ IIR แบบเทียมเดียว (Cascade-Form IIR filter)

ในหัวข้อนี้จะเป็นการสร้างและการเขียนโปรแกรมของวงจรกรองความถี่แบบ IIR แบบเทียมเดียว (Cascade-form IIR filter) การสร้างโปรแกรมแบบเทียมเดียวนั้นเราจะปรับปรุงผลของวงจรกรองความถี่ IIR แบบตรงแบบที่ 2

การแปลง Z ของผลการตอบสนองอิมพัลล์ของวงจรกรองความถี่ IIR จากสมการที่ 4.7 เขียนในรูปของแบบของอันดับที่ 2 จะได้ว่า

$$H(Z)=\prod_{k=1}^{N/2}\frac{\beta_{0k}+\beta_{1k}Z^{-1}+\beta_{2k}Z^{-2}}{1-\alpha_{1k}Z^{-1}-\alpha_{2k}Z^{-2}} \tag{4.10}$$

โดยที่วงจรเป็นวงจรที่ต่ออนุกรมกันของวงจรกรองความถี่ IIR อันดับที่ 2 แบบวงจร Biquads เราสามารถจะแสดงได้ดังรูป 4-10



รูปที่ 4-10 วงจรกรองความถี่ IIR อันดับ 4 แบบ Cascade

ในแต่ละส่วนของจะเป็นวงจรกรองความถี่แบบตรง อันดับ 2 ซึ่งแสดงสมการที่ 4-10 ซึ่งเอาท์พุทที่ได้จากส่วนที่ 1 จะเป็น อินพุทในส่วนที่ 2 และส่งต่อกันไปเป็นทอดๆ

สมการผลต่างสำหรับแบบส่วนที่ Cascade กันนั้นจะเขียนได้เป็น

$$\begin{aligned} d_i(n) &= y_{i-1}(n)+\alpha_{1i}d_i(n-1)+\alpha_{2i}d(n-2)\\ y_i(n) &= \beta_{0i}d_i(n)+\beta_{1i}d_i(n-1)+\beta_{2i}d(n-2) \end{aligned} \tag{4.11}$$

เมื่อ i = 1, 2, …, N/2

y_{i-1} = ค่าอินพุทของส่วนที่ i

$d_i(n)$ = ค่าหน่วงเวลาแต่ละค่าในส่วนที่ i

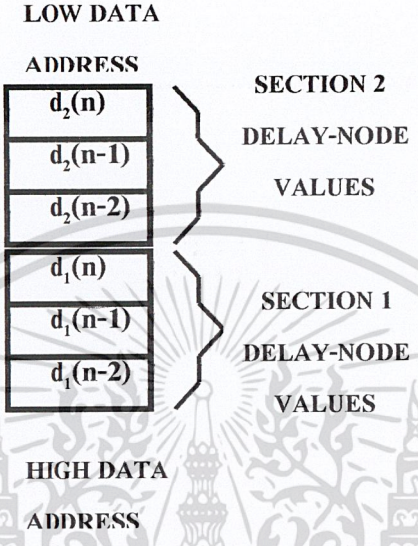
$y_i(n)$ = ค่าเอาท์พุทของส่วนที่ i

$y_0(n)=x(n)$ = ค่าอินพุทเข้าวงจรกรอง

$y_{N/2}=y(n)$ = ค่าเอาท์พุทของวงจรกรองความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรกรองความถี่ IIR ที่มีการ cascade กัน 2 ส่วน ดังรูปที่ 4-10 โดยที่ค่าการหน่วงเวลาถูกเก็บในหน่วยความจำข้อมูลดังรูปที่ 4-11



รูปที่ 4-11 การจัดหน่วยความจำของค่าการหน่วงเวลา แบบ Cascade

การเขียนโปรแกรมแบบ Cascade อันดับ 4 มีขั้นตอนในการสร้างดังต่อไปนี้

1. โหลดค่าอินพุทใหม่เข้ามา $x(n)$
2. ลักษณะ โปรแกรมส่วนแรกจะเป็นเช่นเดียวกับวงจรกรองความถี่ IIR อันดับ 2
3. ค่าเอาท์พุทที่ออกจากวงจรส่วนที่ 1 จะอยู่ผ่านใน Accumulator
4. สร้างส่วนที่ 2 โดยมีโปรแกรมเช่นเดียวกันกับส่วน 1 โดยที่ ค่าเอาท์พุทของส่วนที่ 1 ยังภายใน Accumulator
5. เอาท์พุทของส่วนที่ 2 จะเป็นเอาท์พุทของวงจรกรองความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลอง

วัตถุประสงค์

เพื่อให้นักศึกษาสังเกตถึงปรากฏการณ์ของการสุ่มสัญญาณ (Sampling) ที่เป็นไปตามทฤษฎี Nyquist Sampling rate ที่มีความถี่เป็นสองเท่าของความถี่สัญญาณเชิงอุปมาน (frequency analog signal)

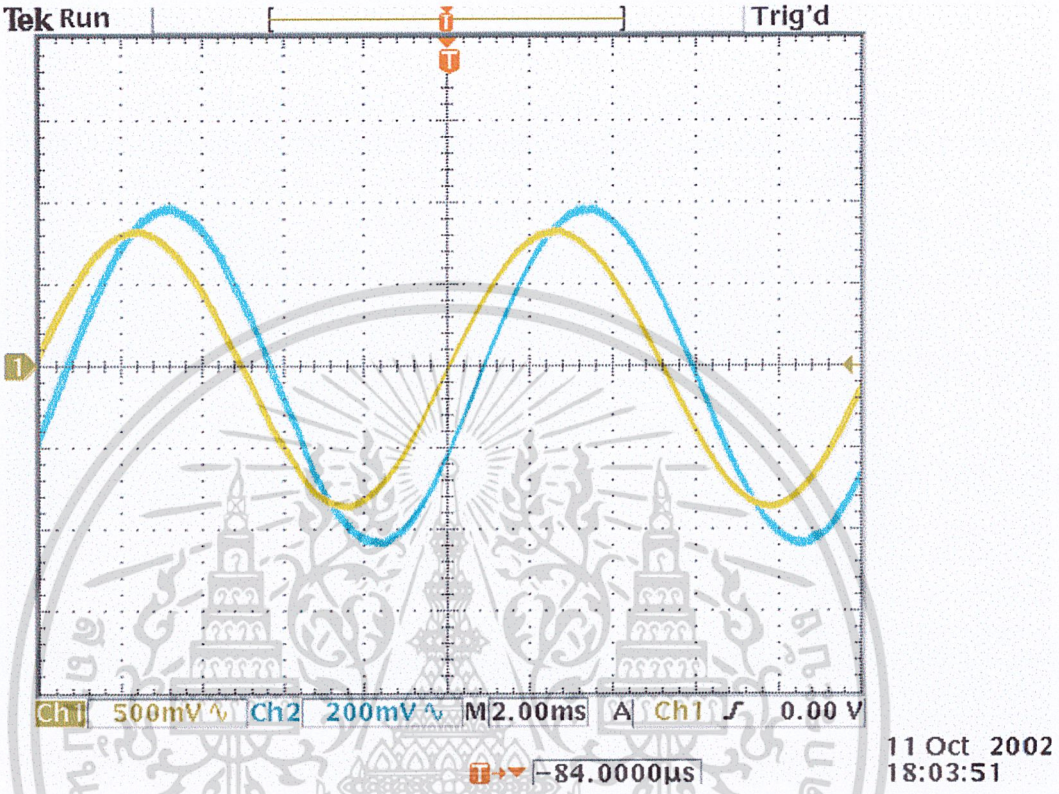
วิธีทำการทดลอง

1. ต่อบอร์ดเข้ากับ Oscilloscope และ Signal generator
2. ปรับให้ Signal generator ให้เป็น $2V_{P-P}$, 1 kHz เป็นสัญญาณเป็นรูป Sine wave
3. ปรับ Sampling rate โดยการเปลี่ยนค่าของ TA,TB กำหนดให้ Sampling rate เริ่มต้น เป็น 10kHz
4. ปรับความถี่ของ signal generator จาก ความถี่ 0 ถึง 5 kHz บันทึกผล Output ที่ ความถี่ต่ำ และที่ $f_S - f_0$ เป็นfold-backแรก และ $k \cdot (f_S - f_O)$ ถัดไป
5. ทำการ run program แล้วบันทึกผลการทดลอง
6. เปลี่ยนค่า Sampling rate แล้วทำการทดลองเช่นเดียวกับ ข้อที่ 4

ข้อสังเกต ในการทดลองข้างต้น เราจะไม่มีการใช้ antialliasing filter ในส่วนของ A/D แต่ในส่วนของการสร้างสัญญาณคืนมีการใช้ lowpass $\left(f_c = \frac{f_S}{2} \right)$ กำหนดโดย TA/TB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

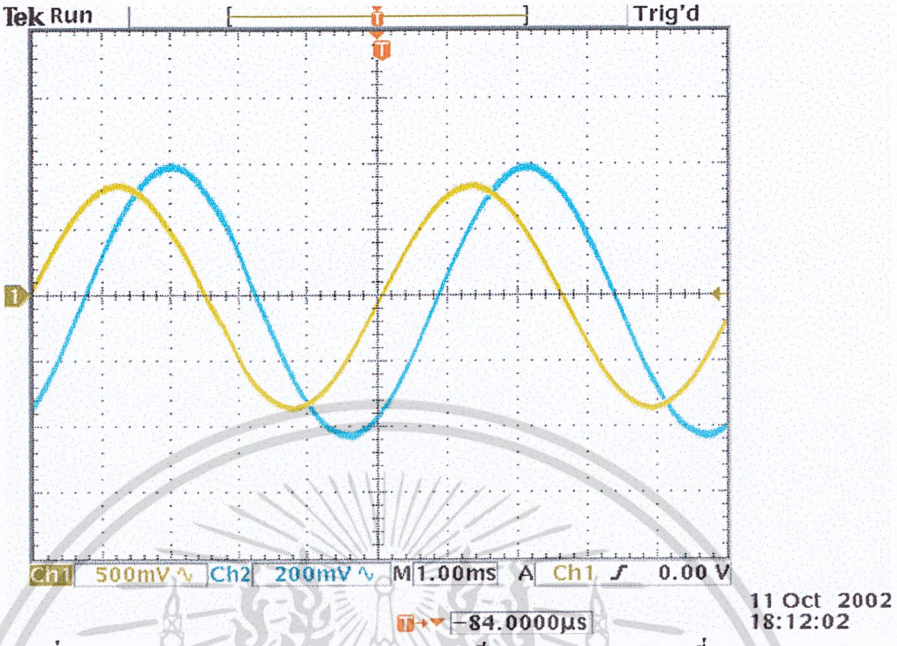
5.1 ผลการทดลองเรื่องการสุ่มสัญญาณ



รูปที่ 5-1 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 100Hz

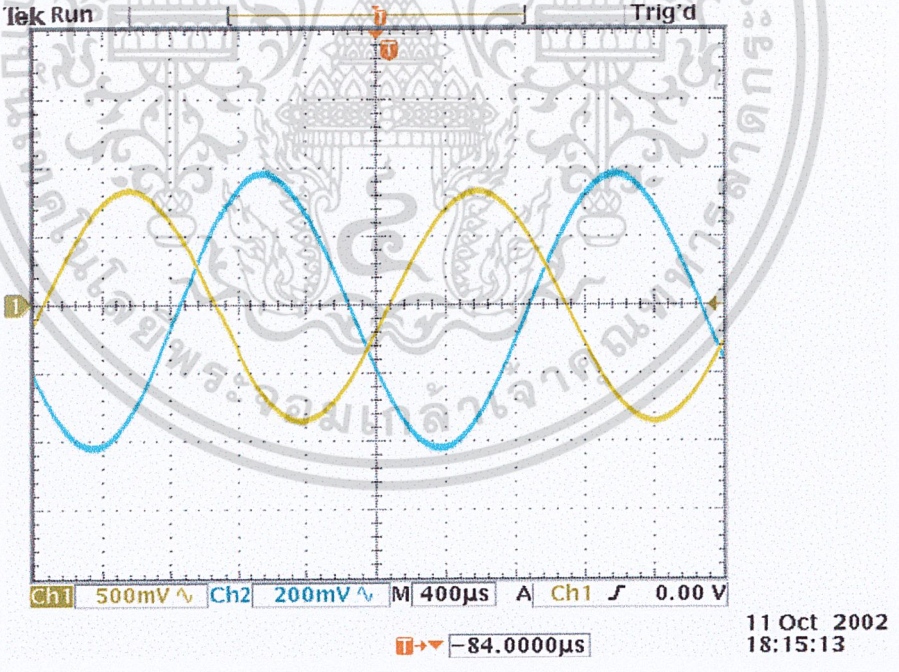
โดย CH 1 เป็น input และ CH 2 เป็น output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 500Hz

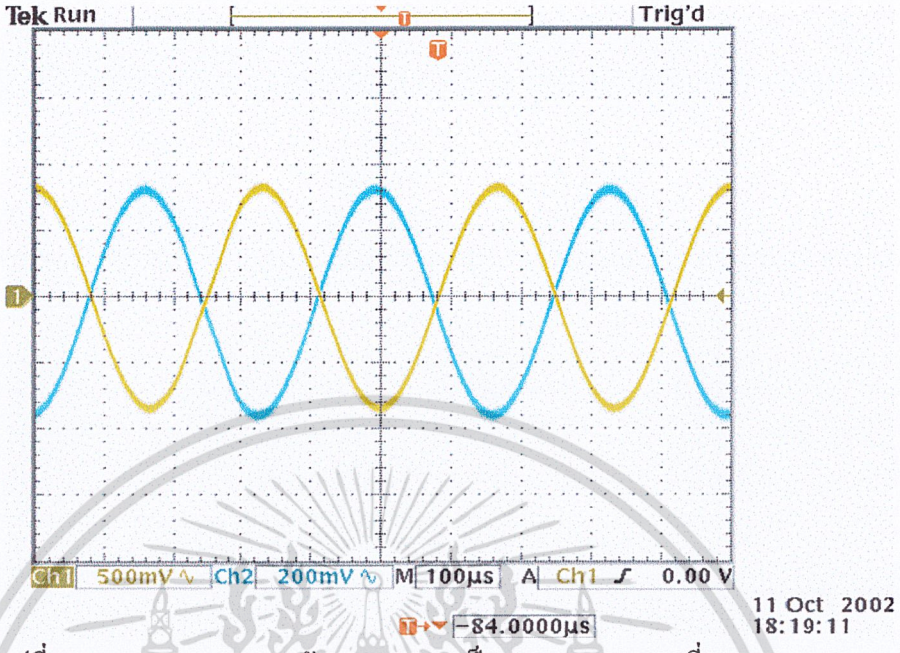
โดย CH 1 เป็น input และ CH 2 เป็น output



รูปที่ 5-3 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 1kHz

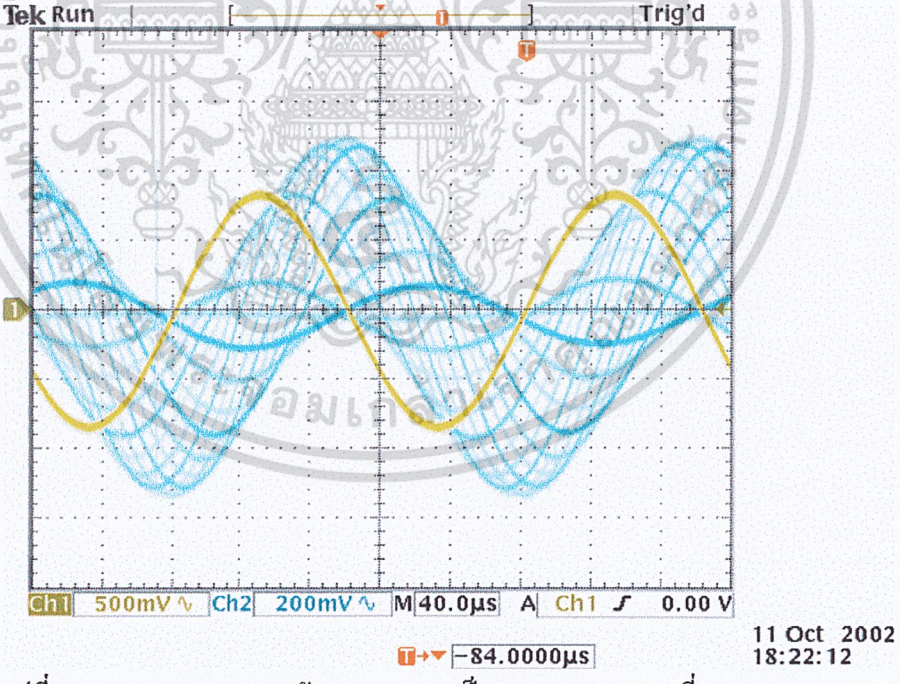
โดย CH 1 เป็น input และ CH 2 เป็น output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 4 kHz

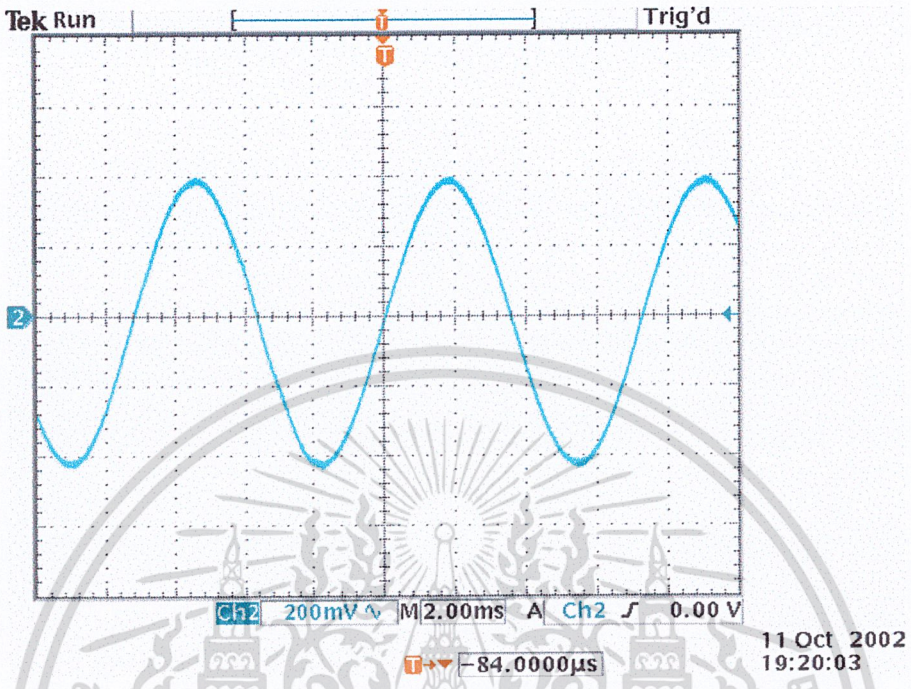
โดย CH 1 เป็น input และ CH 2 เป็น output



รูปที่ 5-5 ผลการทดลองของสัญญาณ input เป็น Sine wave ความถี่ 5 kHz

โดย CH 1 เป็น input และ CH 2 เป็น output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 รูปของ output ที่เกิดจากสัญญาณ Sine wave 10kHz

โดย CH 1 เป็น input และ CH 2 เป็น output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ผลการทดลอง (ส่วนของ FIR และ IIR)

ส่วนนี้จะกล่าวถึงผลการทดลองที่แสดงออกมามาในรูปของกราฟ ซึ่งแบ่งได้เป็น 2หัวข้อใหญ่ๆ ดังนี้

1. ผลการทดลองของตัวกรองผลการตอบสนองอิมพัลส์จำกัดที่ได้จากโปรแกรม MATLAB

2. ผลการทดลองของตัวกรองผลการตอบสนองอิมพัลส์ไม่จำกัดที่ได้จากโปรแกรม MATLAB

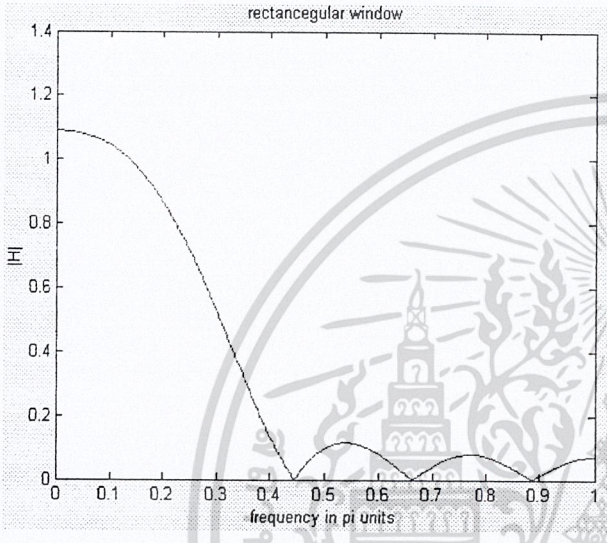
โดยการทดลองนี้เราใช้ความถี่ของการสุ่มตัวอย่างมีค่าเป็นเท่ากับ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

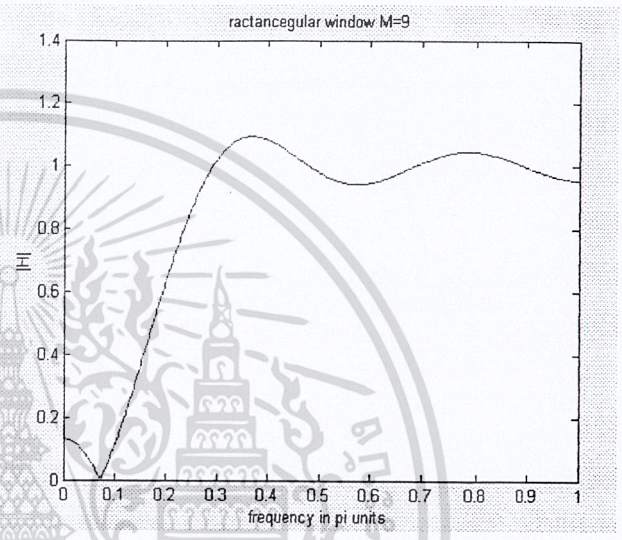
5.2.1 ผลการทดลองของตัวกรองผลการตอบสนองอิมพัลส์จำกัดที่ได้จากโปรแกรม MATLAB

1. โดยใช้ rectancegular window

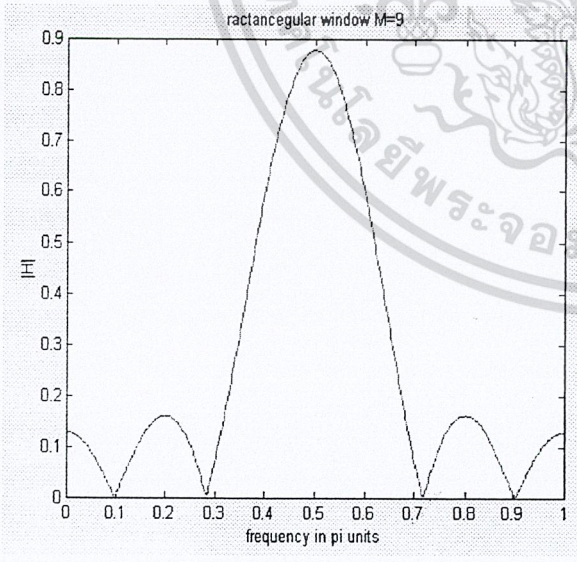
ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2,3 kHz ($\omega = 0.4\pi, 0.6\pi$)



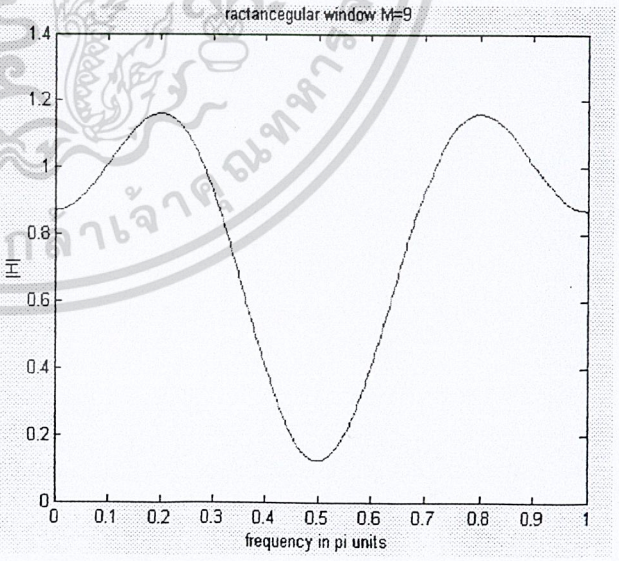
Magnitude response of 6 length lowpass



Magnitude response of 9 length highpass



Magnitude response of 9 length bandpass



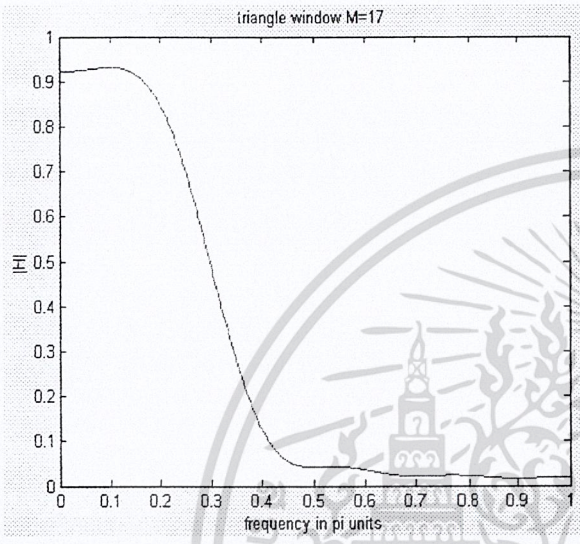
Magnitude response of 9 length bandstop

รูปที่ 5-7 Magnitude response ของตัวกรองความถี่แบบ retancegular window

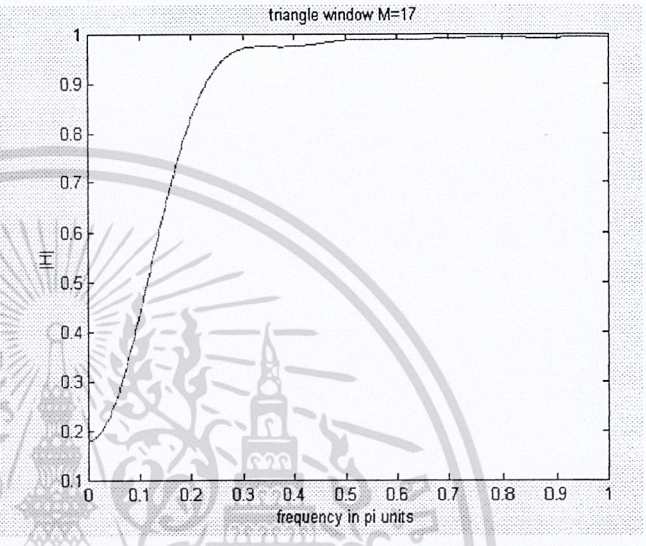
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โดยใช้ triangle window

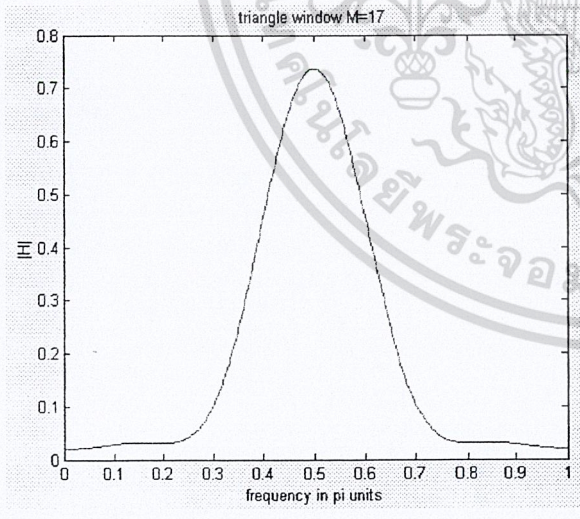
ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2,3 kHz ($\omega = 0.4\pi, 0.6\pi$)



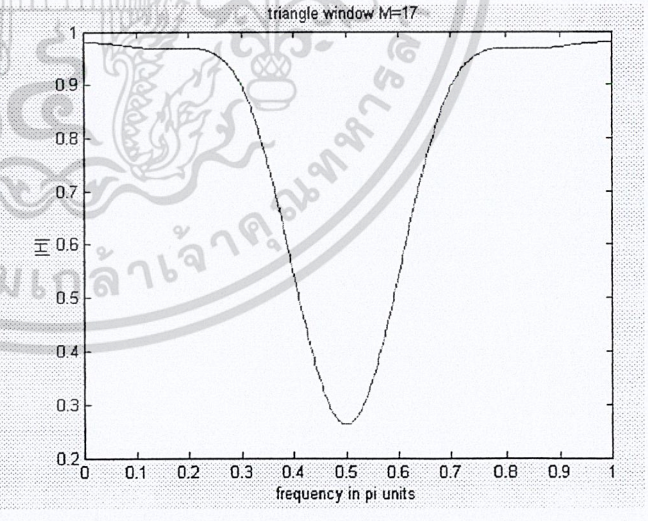
Magnitude response of 17 length lowpass



Magnitude response of 17 length highpass



Magnitude response of 17 length bandpass



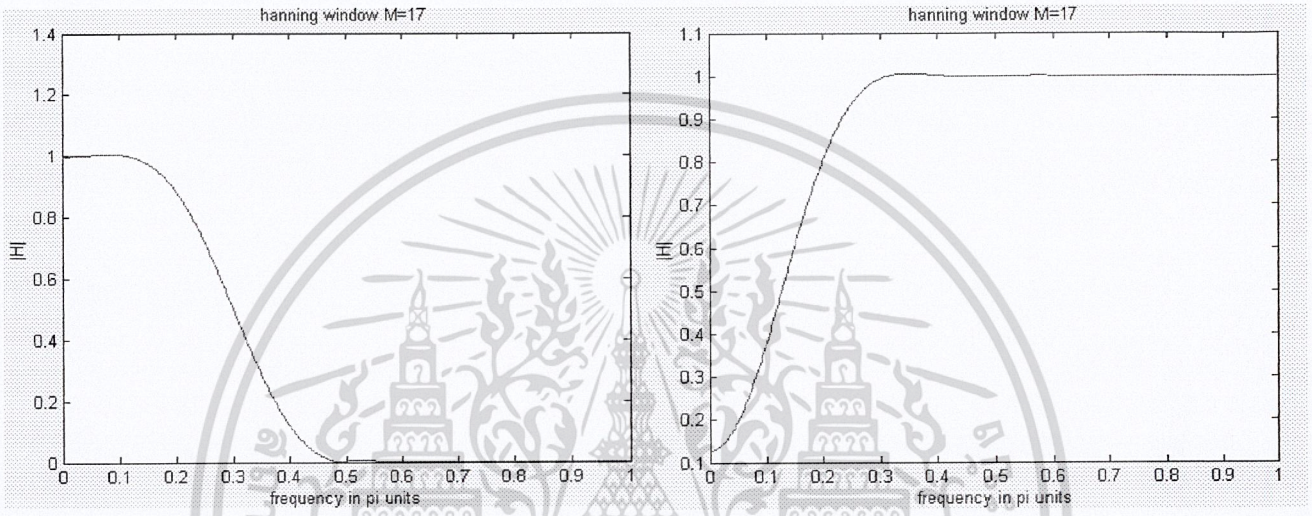
Magnitude response of 17 length bandstop

รูปที่ 5-8 Magnitude response ของตัวกรองความถี่แบบ triangle window

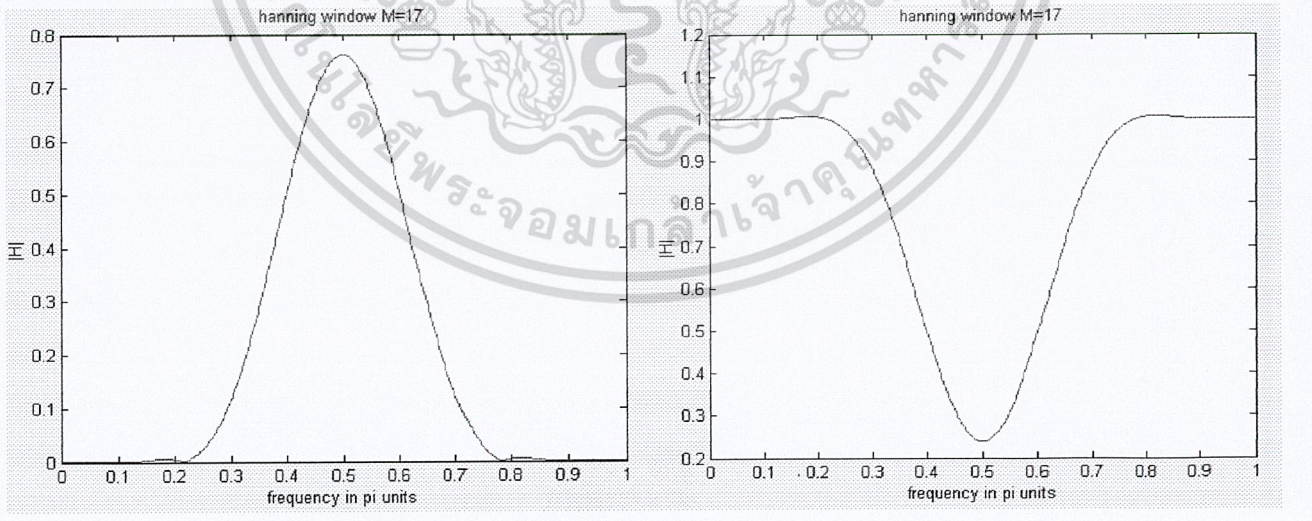
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โดยใช้ hanning window

ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2,3 kHz ($\omega = 0.4\pi, 0.6\pi$)



Magnitude response of 17 length lowpass Magnitude response of 17 length highpass



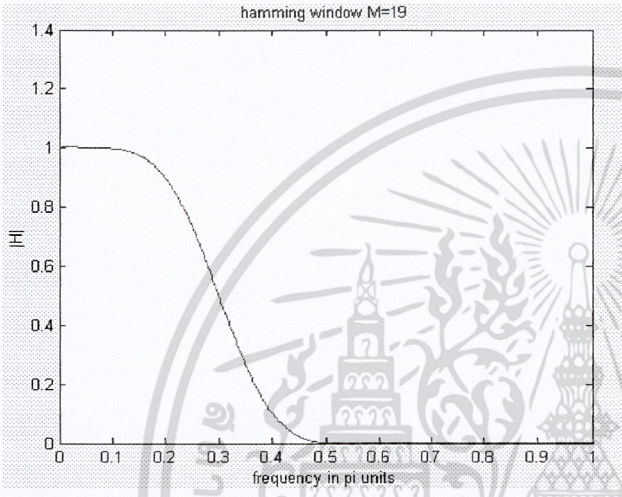
Magnitude response of 17 length bandpass Magnitude response of 17 length bandstop

รูปที่ 5-9 Magnitude response ของตัวกรองความถี่แบบ hanning window

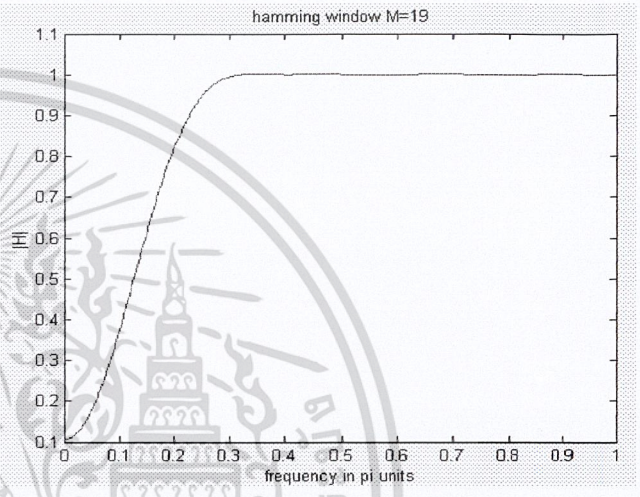
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โดยใช้ hamming window

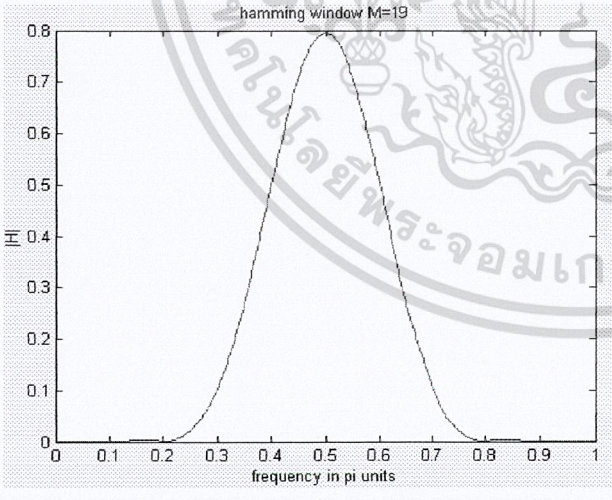
ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2,3 kHz ($\omega = 0.4\pi, 0.6\pi$)



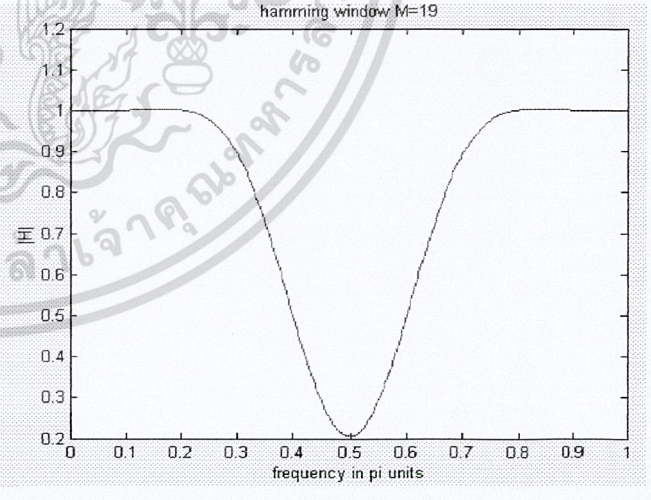
Magnitude response of 19 length lowpass



Magnitude response of 19 length highpass



Magnitude response of 19 length bandpass



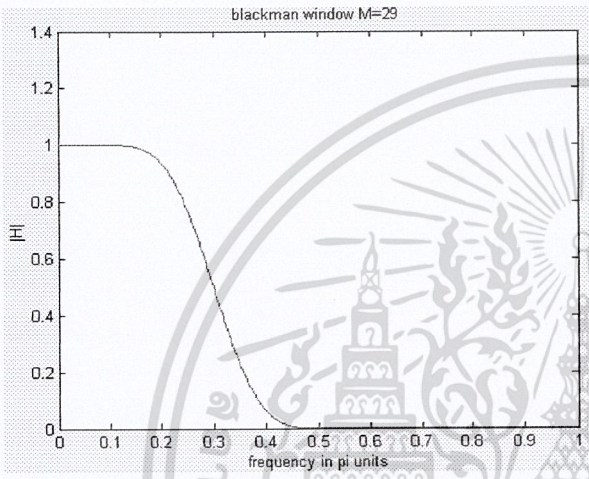
Magnitude response of 19 length bandstop

รูปที่ 5-10 Magnitude response ของตัวกรองความถี่แบบ hamming window

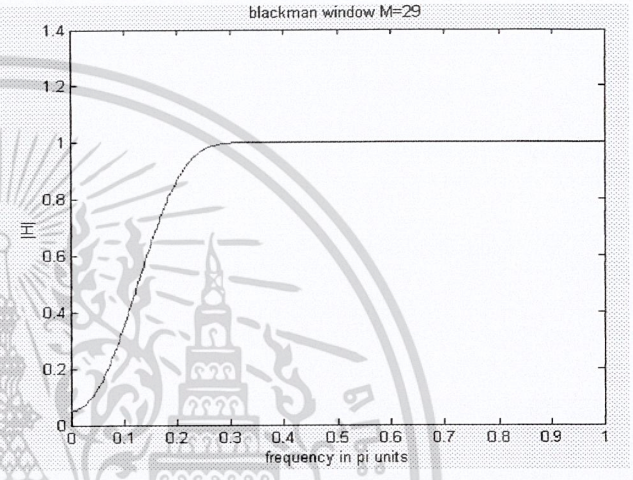
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โดยใช้ blackman window

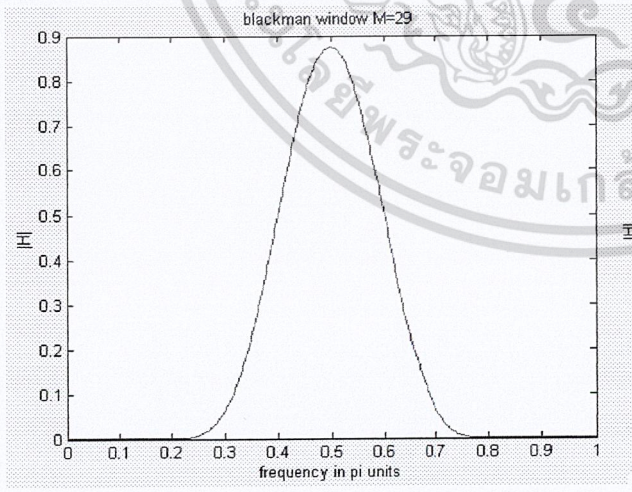
ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2, 3 kHz ($\omega = 0.4\pi, 0.6\pi$)



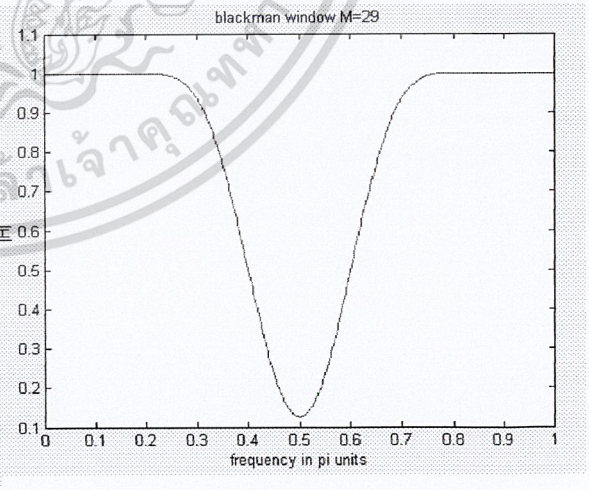
Magnitude response of 19 length lowpass



Magnitude response of 19 length highpass



Magnitude response of 29 length bandpass



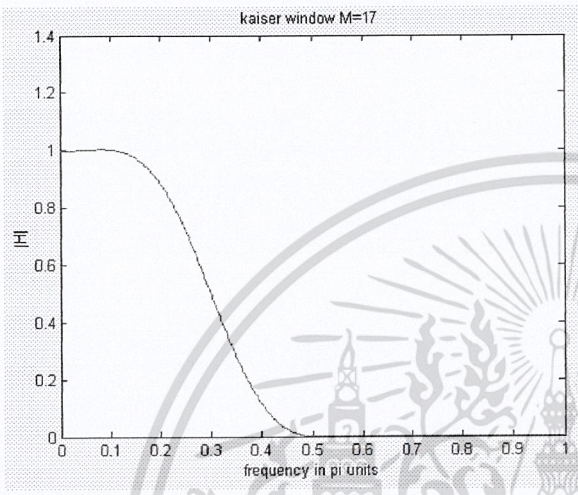
Magnitude response of 29 length bandstop

รูปที่ 5-11 Magnitude response ของตัวกรองความถี่แบบ blackman window

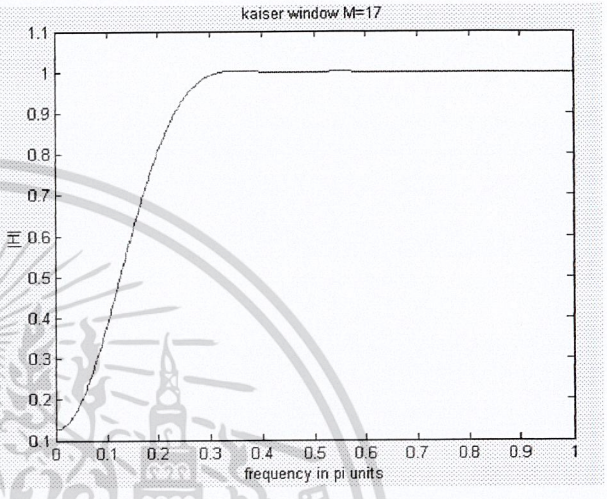
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. โดยใช้ Kaiser window

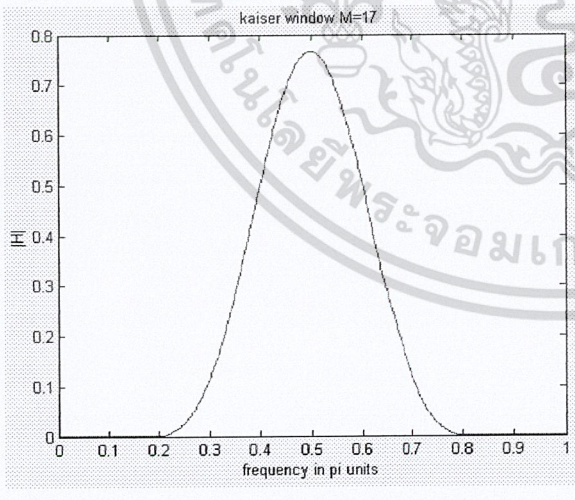
ในการทดลองนี้ ตัวกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ 1.5 kHz ($\omega = 0.4\pi$) และ ตัวกรองความถี่สูงผ่านจะมีความถี่คัทออฟ 875 kHz ($\omega = 0.125\pi$) และ ตัวกรองช่วงความถี่และช่วงลดทอนความถี่เป็น จะมีคัทออฟเป็น 2,3 kHz ($\omega = 0.4\pi, 0.6\pi$)



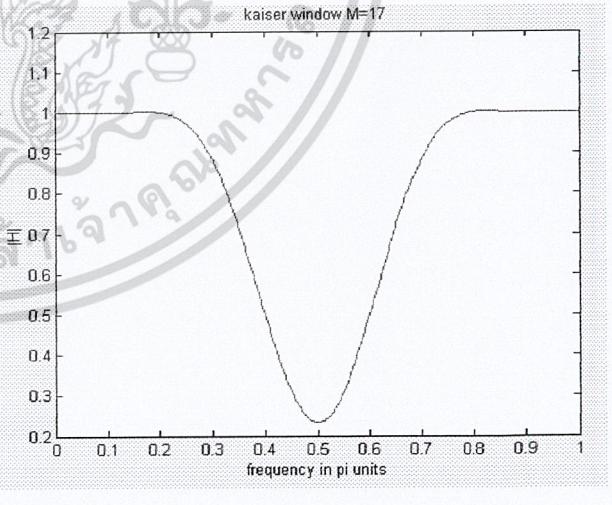
Magnitude response of 17 length lowpass



Magnitude response of 17 length highpass



Magnitude response of 17 length bandpass



Magnitude response of 17 length bandstop

รูปที่ 5-12 Magnitude response ของตัวกรองความถี่แบบ Kaiser window

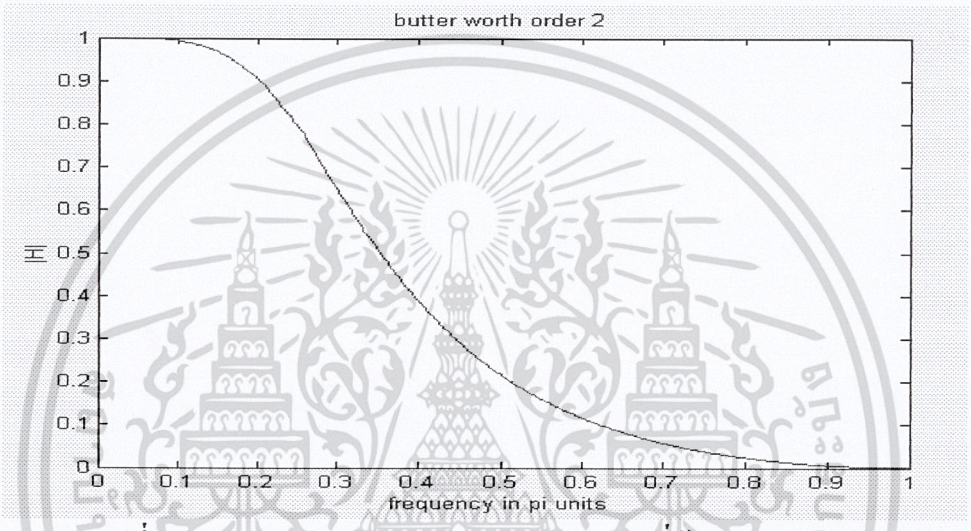
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ผลการทดลองของตัวกรองผลการตอบสนองอิมพัลส์ไม่จำกัดที่ได้จากโปรแกรม MATLAB โดยแปลงจากวงจรเชิงอุปมานให้อยู่ในรูปของ วงจงกรองความถี่เชิงเลขโดยวิธี Bilinear transform

1. การออกแบบวงจกรองความถี่ต้นแบบแบบความถี่ต่ำผ่าน

- butterworth

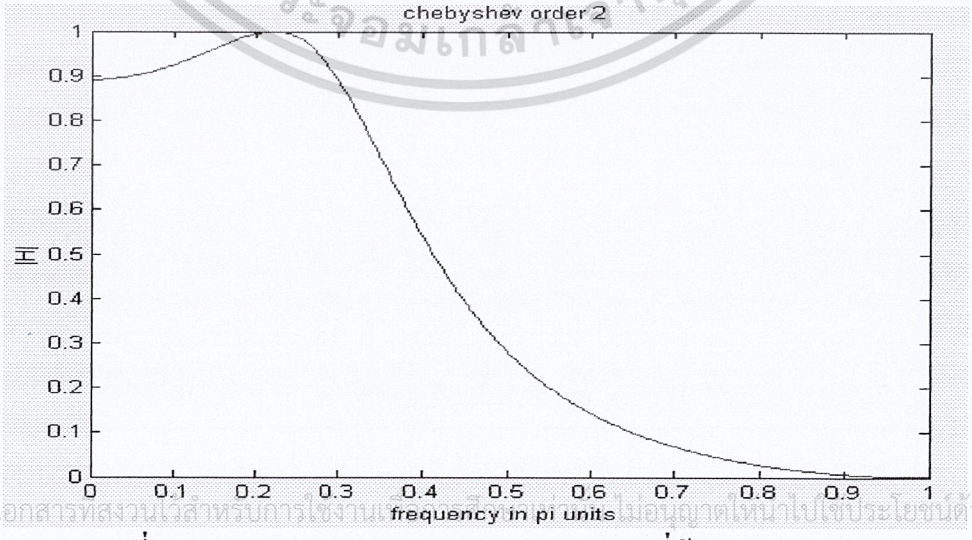
ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation เป็น 15 dBจะได้ว่า



รูปที่ 5-13 Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ butterworth

- chevbyshev

ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation stopband เป็น 15 dBจะได้ว่า

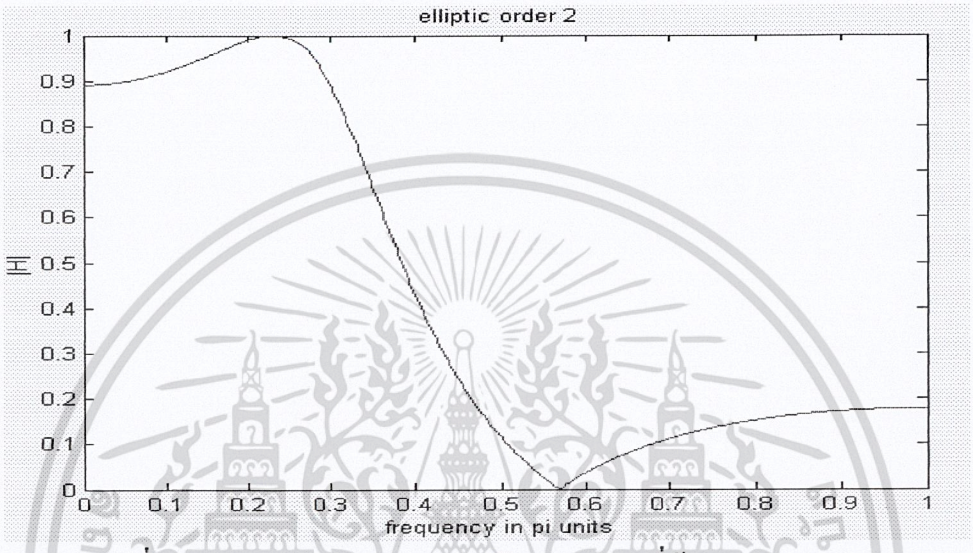


รูปที่ 5-14 Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ chebyshev

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-elliptic

ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation stopband เป็น 15 dBจะได้ว่า

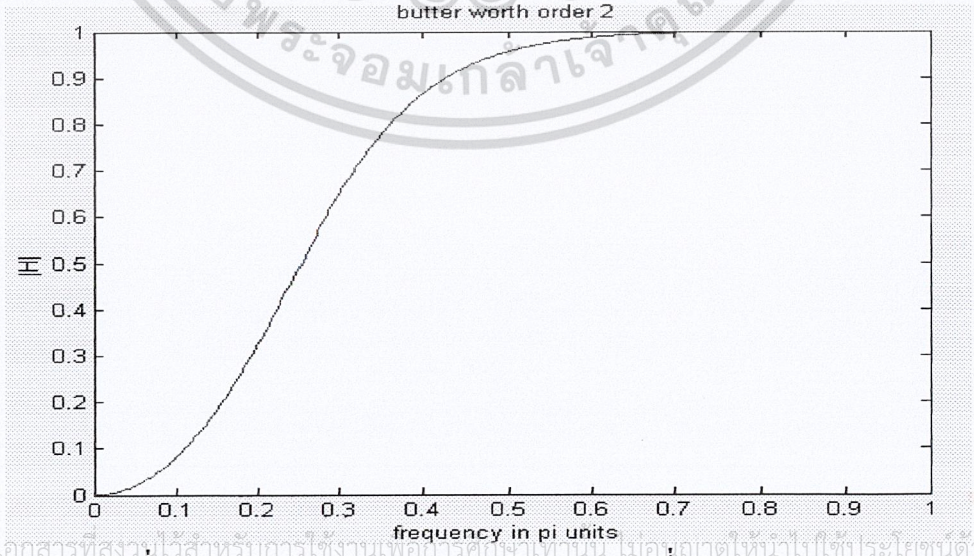


รูปที่ 5-15Magnitude response ของตัวกรองความถี่ต้นแบบ แบบ elliptic

2. การออกแบบวงจรกรองความถี่แบบความถี่สูงผ่าน

- butterworth highpass

ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation stopband เป็น 15 dBจะได้ว่า

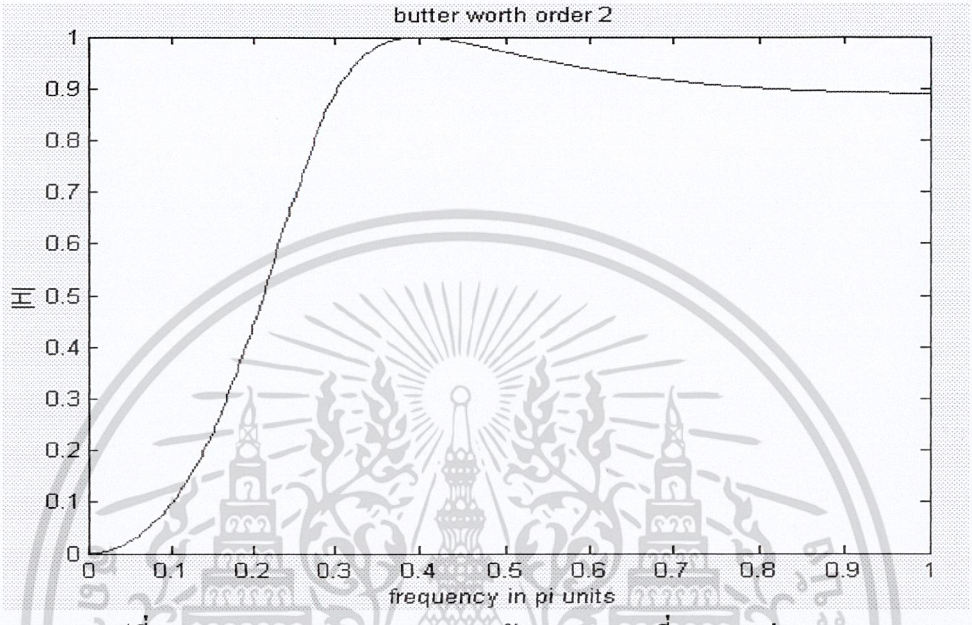


รูปที่ 5-16 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ butterworth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- chebyshev highpass

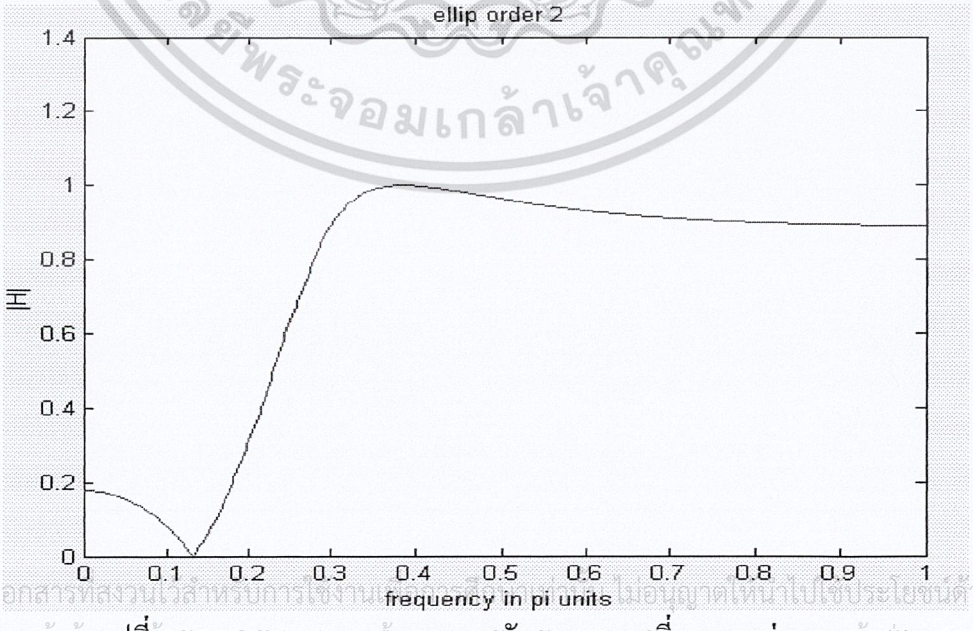
ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation stopband เป็น 15 dBจะได้ว่า



รูปที่ 5-17 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ chebyshev

-elliptic highpass

ในการทดลองนี้เป็นการออกแบบวงจรกรองความถี่แบบ IIR อันดับ 2 และความถี่คัทออฟเป็น 1.5 kHz (0.3π) และripple passband เป็น 1 dB และ attenuation stopband เป็น 15 dBจะได้ว่า



รูปที่ 5-18 Magnitude response ของตัวกรองความถี่แบบสูงผ่านแบบ ellip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์

ผลการทดลองของฟิลเตอร์แต่ละชนิดสามารถนำมาเปรียบเทียบกันได้ดังนี้

ตารางที่ 6-1 ตารางเปรียบเทียบจุดคัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบจำกัด

ชนิดของฟิลเตอร์	อันดับ	จุดคัทออฟ (kHz)	จุดคัทออฟจากการทดลอง (kHz)
Lowpass Kaiser	17	1.5	1.39
Lowpass hanning	17	1.5	1.4
Lowpass hamming	19	1.5	1.3
Lowpass Blackman	29	1.5	1.3
Lowpass rectangular	9	1.5	1.1
Lowpass triangle	17	1.5	1.4
Highpass Kaiser	17	0.875	0.550
Highpass hanning	17	0.875	0.500
Highpass hamming	19	0.875	0.503
Highpass Blackman	29	0.875	0.560
Highpass rectangular	9	0.875	0.625
Highpass triangle	17	0.875	0.500
Bandpass Kaiser	17	2 ,3	1.5 , หาค่าไม่ได้
Bandpass hanning	17	2 ,3	1.5 , หาค่าไม่ได้
Bandpass hamming	19	2 ,3	1.6 ,หาค่าไม่ได้
Bandpass Blackman	29	2 ,3	หาค่าไม่ได้,หาค่าไม่ได้
Bandpass rectangular	9	2 ,3	1.5 ,หาค่าไม่ได้
Bandpass triangle	17	2 ,3	1.5 , หาค่าไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bandstop Kaiser	17	1.75 , 3.25	1.5 , หาค่าไม่ได้
Bandstop hanning	17	1.75 , 3.25	1.5 , หาค่าไม่ได้
Bandstop hamming	19	1.75 , 3.25	1.6 ,หาค่าไม่ได้
Bandstop Blackman	29	1.75 , 3.25	หาค่าไม่ได้,หาค่าไม่ได้
Bandstop rectangular	9	1.75 , 3.25	1.5 ,หาค่าไม่ได้
Bandstop triangle	17	1.75 , 3.25	1.5 , หาค่าไม่ได้

จากผลการทดลองในตารางที่เปรียบเทียบไว้ด้านบนพบจุดคัทออฟของตัวกรองต่างๆ มีค่าคลาดเคลื่อนไปจากที่ออกแบบไว้ อาจเนื่องมาจากค่าสัมประสิทธิ์ที่ใช้ยังมีความละเอียดไม่มากนัก นอกจากนี้ยังพบว่าในการทดลองยังมีความแตกต่างกันระหว่างตัวกรองที่มีอันดับไม่เท่ากันเพื่อมีการออกแบบให้ transition band ที่ ค่าเท่ากัน ผลที่ได้จะมีสัญญาณ noise ที่เกิดจากการ weight ค่าสัมประสิทธิ์ทำให้เกิดการขยาย noise เกิดขึ้น และเพื่อป้องกันความเสียหายของบอร์ดทดลองต้องมีการปรับค่าให้มีสัญญาณต่ำๆจึงทำให้เห็น noise ได้จากการทดลอง และ noise เกิดจากการ switch ของ AIC ที่ทำงาน sampling สัญญาณ ในกรณีของ bandpass และ stoppass ที่หาค่าไม่ได้เนื่องจากผลของ alising ที่ 5kHz จึงไม่เห็น upper cutoff frequency

ตารางที่ 6-2 เปรียบเทียบความถี่คัทออฟที่ได้จากการออกแบบกับที่ได้จากการทดลองของตัวกรองผลตอบสนองอิมพัลส์แบบอนันต์

ชนิดของฟิลเตอร์	ค่าความถี่คัทออฟที่ออกแบบ (kHz)	ค่าความถี่คัทออฟที่ได้จากการทดลอง (kHz)
Butterworth Lowpass filter	1.5	1.2
Chebyshev Lowpass filter	1.5	1.3
Elliptic Lowpass filter	1.5	1.4
Butterworth Highpass filter	1.5	หาค่าไม่ได้
Chebyshev Highpass filter	1.5	หาค่าไม่ได้
Elliptic Highpass filter	1.5	หาค่าไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิจารณ์การทดลอง

จากผลการทดลองที่ได้เมื่อนำผลมาเปรียบเทียบกับที่ออกแบบไว้ตอนแรกแล้วพบว่ามีความคลาดเคลื่อนเกิดขึ้น เนื่องจากสัมประสิทธิ์ที่คำนวณได้จาก MATLAB มีค่าทศนิยมเพียง 4 ตำแหน่งเท่านั้น ซึ่งอาจทำให้ผลที่ได้มีความผิดพลาดไปบ้างแต่ก็ถือว่ายังได้ผลเป็นที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรม mmr.asm

```
;--------------------------------
;     memory map register
;     mmr.asm
;--------------------------------
drr    .set   0
dxr    .set   1
tim    .set   2
prd    .set   3
imr    .set   4
greg   .set   5
;--------------------------------
```

2.โปรแกรมการทดลองเรื่องการสุ่มสัญญาณ

```
     .include "mmr.asm"
     .ds    0400h
;---------------------------------
; adjust sampling rate by  ta,tb
;
;     ta=18,tb=40  sampling frequency=7.2
kHz
;     ta=12,tb=42  sampling frequency=10 kHz
;     ta=4,tb=26   sampling frequency=50 kHz
;     ta=4,tb=16   sampling frequency=80 kHz
;---------------------------------
ta     .word  12
tb     .word  42
aic_cmd .word  0003h
;---------------------------------
; interupt vector
;---------------------------------
     .ps    0fa0ah
     eint   ;rint
     ret
     eint   ;xint
     ret
;---------------------------------
;  main program
;---------------------------------
     .ps    0fb00h
     .entry
start:  sxf
     ssxm
     sovm
     ldpk   0
     fort   0
     rtxm
     sfsm
     lack   80h
     sach   dxr
     sacl   greg    ;reset AIC by BR
     conf   1
;---------------------------------
aic_rs: lac    imr     ;open transmit interupt
     ork    020h
     sacl   imr
     ldpk   08h
     lac    aic_cmd
     call   aic_2nd
     lac    tb,2
     add    tb,9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      addk   2
      call   aic_2nd
      lac    ta,2
      add    ta,9
      call   aic_2nd
      ldpk   0
      lack   imr   ;open recieve interupt
      ork    010h
      sacl   imr
      b      main
loop: idle
main: lac    drr
      sacl   dxr
      b      loop
;--------------------------------------
aic_2nd:
      ldpk   0
      adlk   6,15
      idle
      sach   dxr
      idle

      sacl   dxr
      idle
      sacl   dxr,2
      idle
      ldpk   08h
      ret
;----------------------------------------
```

3. โปรแกรมสร้าง pulse

```
dxr   .set   01h
tim   .set   02h
prd   .set   03h
imr   .set   04h
;--------------------------------------
max   .set   07ffch
min   .set   00000h
;--------------------------------------
;       interupt vector
;--------------------------------------
      .ps    0fa08h
      b      time
      .ps    0fa0ch
      eint
      ret
;--------------------------------------
;       main program
;--------------------------------------
      .ps    0fb00h
      .entry
      ldpk   0
      ssxm
      lalk   20000
      sacl   prd
      lack   008h
      or     imr
      sacl   imr
      eint
loop: lalk   max
      sacl   dxr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        idle
        lalk    min
        sacl    dxr
        idle
        b       loop
;---------------------------------------
time:   eint
        ret
;---------------------------------------
```

4. โปรแกรมสร้าง sine

```
dxr     .set    1
imr     .set    4
greg    .set    5
;--------------------------------
;       define  word
;--------------------------------
        .ds     0400h
;--------------------------------
;       ta=12,tb=42   sampling=10kHz
;       ta=4,tb=26    sampling=50kHz
;--------------------------------
ta      .word   10
tb      .word   16
aic_cmd .word   03h
;--------------------------------
        .ds     500h    ;dp=10
f(0)    .word   0
f(1)    .word   3135
f(2)    .word   5791
f(3)    .word   7568
f(4)    .word   8191
f(5)    .word   7568
f(6)    .word   5791
f(7)    .word   3135
f(8)    .word   0
f(9)    .word   -3135
f(10)   .word   -5791
f(11)   .word   -7568
f(12)   .word   -8191
f(13)   .word   -7568
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
f(14)  .word  -5791
f(15)  .word  -3135
;--------------------------------
;      interupt vector
;--------------------------------
       .ps    0fa0ch
       eint   ;xint
       ret
;--------------------------------
;      program
;--------------------------------
       .ps    0fb00h
       .entry
start: ;ssxm
       sovm
       fort   0
       rtxm
       sfsm
       lack   08h
       sach   dxr
       sacl   greg
       conf   1
;--------------------------------
       lack   20h    ;open  transmit serial
interupt
       or     imr
       sacl   imr
;--------------------------------
       ldpk   8
       lac    aic_cmd ;send command to aic
       call   aic_2nd
;-------------------------
       lac    tb,2
       add    tb,9
       addk   2
       call   aic_2nd
;-------------------------
       lac    ta,2
       add    ta,9
       call   aic_2nd

       lack   0000h
       sacl   dxr
;-------------------------
       ldpk   0
       larp   0
       lrlk   ar1,050fh
main:  lrlk   ar0,0500h
       larp   0
loop:  lac    *+,0,ar1
       sacl   dxr,2
       idle
       larp   1
       cmpr   3
       bbnz   loop,ar0
end:   b      main
;--------------------------------
aic_2nd:
       ldpk   0
       adlk   6,15   ;set acc_hi=3
       idle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sach   dxr
idle

sacl   dxr
idle

sacl   dxr,2
ldpk   08h
ret
;---------------------------------
```

5. โปรแกรมสร้างวงจรกรองความถี่แบบ FIR

```
bcmd    .set 0xfa10        ;
bxmit   .set 0xfa12        ;
bxmit16 .set 0xfa14    ;
brecv   .set 0xfa18    ;
brecv16 .set 0xfa1a    ;
pma     .set 0xfbea        ;
        .ds  0x60          ;setup data in upper
1/2 b2
ta      .word 8            ;10kHz
tb      .word 63           ;
aic_cmd .word 0x0003   ;
        .include "mmregs.asm";
        .ps  0xfa0a        ;
        eint
        ret
        eint
        ret                ;Xint is only timing
,so just reture
        .ps  0xfbe0        ;this is the array of
coeff's
        .word   -766       ;h[0]=0.15 x 16384 =
2^14
        .word    537       ;h[1]=0.19
        .word   2480       ;h[2]=0.20
        .word   4218       ;h[3]=0.19
        .word   4915       ;h[4]=0.15
        .word   4218       ;h[5]
        .word   2480       ;h[6]
        .word    537       ;h[7]
        .word   -766       ;h[8]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .ps  0xfb00      ;
        .entry
start:  sxf              ;
        ssxm
        sovm
        ldpk    0
        fort    0
        rtxm
        sfsm
        lack    0x80     ;
        sach    dxr
        sacl    greg
        lrlk    ar0,0xffff
        rptk    250
        lac     *,0,ar0
        conf    1
aic_rs: lack    0x28
        sacl    imr
        ldpk    0
        lac     aic_cmd
        call    aic2nd
        lac     tb,2
        add     tb,9
        addk    2
        call    aic2nd
        lac     ta,2
        add     ta,9
        call    aic2nd
        ldpk    0
        lack    0x18
        sacl    imr
        conf    1

        spm     0
        sovm    0
        ldpk    0
sub:    idle    ; wait for receiver interrupt
        lac     drr     ; read the new data
        ldpk    0x8     ; we have indirect
addressing
        sacl    do      ; store it into array
        mpyk    0       ; make p reg 0
        zac             ; make accumulator 0
        lrlk    ar0,0x408 ; address d(n-(N-
1))  (change)
        rptk    9       ; repeat N-1 to
multiply and acumulate N time (change)
        macd    0xfbe0,*-       ;muliply and
accumulate with the data move
        sach    yn,1
        lac     yn
        ldpk    0
        sacl    dxr,1   ;the shift is because
of the 2'compl arith
        b sub
aic2nd: ldpk    0
        adlk    6,15    ;set accu_hi = 3 for
seconsdary xmit
        idle
        sach    dxr
        idle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sacl    dxr
        idle
        sacl    dxr,3
        ldpk    0
    ret

        .ds     0x400
do:     .word   0
        .word   1
        .word   2
        .word   3
        .word   4
        .word   5
        .word   6
        .word   7
        .word   8
        .word   9
        .word   10
yn:     .word   11
        .end
;-----------------------------------------
```

6. โปรแกรมการสร้างความถี่แบบ IIR

```
cmd     .set    0xfa10  ;
bxmit   .set    0xfa12  ;
bxmit16 .set    0xfa14  ;
brecv   .set    0xfa16  ;
brecv16 .set    0xfa18  ;
bcxmit  .set    0xfa1a  ;
pma     .set    0xfbea  ;
        .ds     0x0420  ;
ta      .word   12      ;fs = 10khz
tb      .word   42
aic_cmd .word   0x0003  ;
        .include "mmregs.asm"
        .ps     0xfa0a  ;
        eint
        ret
        eint    ; XINT is only for time, so
just return
        ret
;-----------------------------------------
---
        .ps     0xfb00  ;
        .entry          ;
start:  sxf
        ssxm
        sovm
        ldpk    0
        fort    0
        rtxm
        sfsm
        lack    0x80    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sach    dxr
        sacl    greg
        lrlk    ar0,0xffff
        rptk    250
        lac     *,0,ar0
        conf    1
aic_rs: lack    0x28
        sacl    imr     ;
        ldpk    0x08
        lac     aic_cmd
        call    aic2nd
        lac     tb,2
        add     tb,9
        addk    2
        call    aic2nd
        lac     ta,2
        lac     ta,9
        call    aic2nd
        ldpk    0
        lack    0x28
        sacl    imr
        conf    1

        spm     0
        sovm    0
        larp    ar0
        ssxm                    ;set
signextension
        larp    ar0
        lark    ar0,0x400
        zac
        ldpk    0x8
        sacl    un
        sacl    unm1
        sacl    unm2
        ldpk    0
sub:    idle
        lac     drr,15          ;read the new
data
        ldpk    0x8
; we implement here the direct II of IIR filter
main:   mpyk    0               ;clear P reg
        lt      unm1            ;tr = unm1
        mpy     a1              ; pr =
umm1*a1
        apac                    ; acc last
product acc = unm1*a1+2^15*x[n]
        lt      unm2
        mpy a2    ; pr = unm2*a2
        apac            ; acc last product acc
= unm2*a2+unm1*a2+2^15*x[n]
        sach    un              ;shift
1,storage upper 16 bit un is the temporary
stroage un=unm2*a2+unm1*a2+2^15*x[n]
        zac
        lt      unm2
        mpy b2
        ltd     unm1
        mpy     b1              ; we still
have tr =unm1 so pr = b1*unm1
        ltd     un
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mpy b0              ; tr=b0*un
        apac                ; acc =
acc+b0*un=b1*unm1+b0*un
        sach    yn          ; prepare to output
y[n]
        lac     yn
        ldpk    0
        sacl    dxr,2
        b       sub         ;repeat
forever
;---------------------------------
aic2nd: ldpk    0
        adlk    6,15
        idle
        sach    dxr
        idle
        sacl    dxr
        idle
        sacl    dxr,2
        ldpk    0x08
        ret
;---------------------------------
        .ds     0x0400
un              .word   0
unm1    .word   1       ;storage for u(n-k)
unm2    .word   2       ;
spar    .word   0
b0              .word   3853            ;
2^14 = 0.5*2^15   (map)
b1              .word   7707    ;
b2              .word   3853            ;
a1              .word   26997   ;
a2              .word   -9640
yn              .word   0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เรื่อง สามารถทำสำเร็จได้ด้วยดีก็เพราะความช่วยเหลือเป็นอย่างดีจากท่านอาจารย์ที่ปรึกษาที่ช่วยเหลือในการให้ความรู้และข้อมูลต่างๆ , อาจารย์ทุกๆ ท่านซึ่งกรุณาช่วยให้คำปรึกษา ให้ยืมหนังสือเพื่อศึกษาค้นคว้าข้อมูลและกรุณาตอบคำถามเป็นอย่างดี , พี่ๆซึ่งให้ทั้งกำลังใจและความรู้ใหม่ๆ รวมทั้งเพื่อนๆทุกคนที่ช่วยเหลือเป็นอย่างดี ทางผู้จัดทำจึงขอถือโอกาสนี้ขอแสดงความขอบคุณมายัง อาจารย์ เทอดศักดิ์ ลิ่วหาทอง , อาจารย์ทุกท่าน , พี่ ๆ และเพื่อนๆทุกๆ คนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. รศ.ดร. สมศักดิ์ ชุ่มช่วย , "การประมวลผลสัญญาณเชิงเลขเบื้องต้น" , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 239 หน้า , 2545

2. ศ.ดร. วัลลภ สุระกำพลธร , "การประมวลผลสัญญาณเชิงเลข" , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 384 หน้า , 2533

3. รศ. ดร. มนัส สังวรศิลป์ , วรรัตน์ ภัทรอมรกุล , "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์" , อินโฟ เพรส , 508 หน้า , 2543

4. Viney K. Ingle and G. Proakis , "Digital Signal Processing using MATLAB" , Bookware Companion Series , 418 p. , 2000

5. Texus Instuments Incorporated, "TMS320C2X User's Guide", 1993

6. Texus Instuments Incorporated, "TMS320C2X DSP Starter Kit User's Guide", 1996

7. AI Lovrich and Ray Simar , Jr. , "Implementation FIR/IIR Filters with TMS32010/TMS32020",Texas Instument,1989

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้