

การออกแบบวงจรกรองความถี่เพื่อแยกซินทิลเลชันและเรนแอทเทนชูน์ในสัญญาณดาวเทียม

ON DESIGN OF FILTER FOR EXTRACTING SCINTILLATION AND RAIN  
ATTENUATION FROM RAW DATA



เลขที่.....  
เลขทะเบียน.. 42316  
วัน, เดือน, ปี 16 พ.ค. 2545

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

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำออกใช้

21175

หัวข้อปริญญานิพนธ์

การออกแบบวงจรกรองความถี่เพื่อแยก ซิลทิลเลชัน  
และเรนแอทเทนนูเอชัน ในสัญญาณดาวเทียม  
On Design Of Filter For Extracting Scintillation  
And Rain Attenuation From Raw Data

นักศึกษา

นายสุพจน์ แสงศรี เลขประจำตัว 42015665  
นายสุวิทย์ นัมรัตน์ เลขประจำตัว 42015666

อาจารย์ที่ปรึกษา

ผศ. อรลาภ แสงอรุณ

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้  
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

อาจารย์ที่ปรึกษา

( ผศ.อรลาภ แสงอรุณ )

กรรมการ

( )

กรรมการ

( )

กรรมการ

( )

กรรมการ

( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

การออกแบบวงจรกรองความถี่เพื่อแยกชนิดทิลเลชั่นและเรนแอทเทนนูเอชันในสัญญาณดาวเทียม

On Design Of Filter For Extracting Scintillation

And Rain Attenuation From Raw Data

นักศึกษา

นายสุพจน์ แสงศรี

เลขประจำตัว 42015665

นายสุวิทย์ นัมรัตน์

เลขประจำตัว 42015666

อาจารย์ที่ปรึกษา

ผศ.อรสาภ แสงอรุณ

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2544

บทคัดย่อ

ในการวิเคราะห์ระดับการเปลี่ยนแปลงของสัญญาณดาวเทียม ในย่านความถี่เคยูพบว่าในขณะที่ฝนตกอาจเกิดปรากฏการณ์ชนิดทิลเลชั่นและปรากฏการณ์การลดทอนของสัญญาณขึ้นพร้อมกัน ปรากฏการณ์ทั้งสองนี้สามารถแยกออกจากกัน โดยใช้วงจรกรองความถี่ ปริญญานิพนธ์นี้ออกแบบวงจรกรองความถี่แบบดิจิทัล เอฟไออาร์ แบตเตอร์เวิร์ด เพื่อแยกปรากฏการณ์ดังกล่าวออกจากกัน โดยการเลียนแบบการทำงานด้วยคอมพิวเตอร์และการประยุกต์ใช้บอร์คดีเอสพี ซึ่งควบคุมการทำงานด้วยโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT ON DESIGN OF FILTER FOR EXTRACTING  
SCINTILLATION AND RAIN ATTENUATION FROM  
RAW DATA

NAME Mr.Supote Sangsri ID. 42015665  
Mr.Suwit Nummarut ID. 42015666

ADVISOR Asst.Prof. Ornlarp Sangaroon

DEPARTMENT INDUSTRIAL TECHNOLOGY

ACADEMIC Year 2001

ABSTRACT

Base on the analysis of satellite signal fading in Ku – Band, scintillation phenomena are concurrent with rain attenuation phenomena . Two phenomena can be separated by filtering. This project examined and extracting this phenomena by using FIR Butterworth Digital Filter. The producer of application , on simulation and DSP control board, can be concluded.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงไปด้วยดีทั้งด้านทฤษฎี การปฏิบัติและองค์ประกอบอื่นๆ อีกมากมาย พร้อมทั้งคำแนะนำและความช่วยเหลือจากอาจารย์ที่ปรึกษา ทั้งด้านเทคนิคต่างๆ ตลอดจนความร่วมมือของบุคคลหลายๆท่าน

ขอขอบคุณ ท่านอาจารย์ ผศ. อรลาภ แสงอรุณ และ ท่านอาจารย์ รศ. ชวลิต เบญจางคประเสริฐ ที่ได้ให้ความช่วยเหลือในทุกๆด้านเกี่ยวกับการทำโครงการในครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญาบัตรฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน



คณะผู้จัดทำ

นายสุพจน์ แสงศรี

นายสุวิทย์ นัมรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
บทที่ 1 ตัวกรองเชิงเลข	1
1.1 ตัวกรองเชิงเลข	1
1.2 การออกแบบตัวกรองป้อนกลับเชิงเลข	19
1.3 โครงสร้างของวงจรถองเชิงเลข	40
1.4 การออกแบบตัวกรองไม่ป้อนกลับเชิงเลข	45
บทที่ 2 ทฤษฎีวิชาการประมวลผลสัญญาณดิจิทัล (DSP TMS320C542)	60
2.1 รายละเอียดของบอร์ด DSK	60
2.2 ระบบที่ต้องการ	61
2.3 วงจรสมบูรณ	62
2.4 เกี่ยวกับชิพ TMS320C542	62
2.5 TLC320AC01C	63
2.6 ติดตั้งซอฟต์แวร์	64
2.7 เริ่มต้นการใช้งาน	64
2.8 การใช้งานทั่วไป	65
2.9 ส่วนสำคัญของ CPU	69
2.10 สถาปัตยกรรมภายใน	72
2.11 Memory	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

บทที่ 3	รายละเอียดเกี่ยวกับโครงการ	96
3.1	โครงสร้างทางฮาร์ดแวร์	96
3.2	รายละเอียดทางด้านซอฟต์แวร์	97
3.3	การออกแบบ FIR Filter ที่ใช้ในโครงการ	112
บทที่ 4	การทดลอง	116
4.1	Frequency Response Of Analog Interface Chip	116
4.2	FIR Low Pass Filter	119
4.3	FIR High Pass Filter	121
บทที่ 5	สรุปผลการทดลอง	125
5.1	สรุปผล	125
5.2	แนวทางการพัฒนาและประยุกต์ใช้งาน	125
5.3	ปัญหาที่เกิดขึ้นและแนวทางแก้ไข	127
ภาคผนวก		
เอกสารอ้างอิง		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

รูปที่	หน้า
รูป 1-1 แผนภาพวงจรรองเชิงเลขของสมการ (1.1)	5
รูป 2-2 แสดงผลตอบสนองความถี่ของอินทิเกรเตอร์เชิงเลข	7
รูป 1-3 แสดงการหาผลตอบสนองความถี่ของตัวกรองเชิงเลขโดยวิธีกราฟ	9
รูป 1-4 ตัวแบบแทนการสุ่มตัวอย่างสัญญาณเชิงอุปมาน	10
รูป 1-5 แสดงสัญญาณสุ่มตัวอย่าง	11
รูป 1-6 แสดงสเปกตรัมของสัญญาณจากการสุ่มตัวอย่าง	13
รูป 1-7 แสดงผลตำแหน่งโพลจากการสุ่มตัวอย่างระบบที่มีโพลที่ $s = s_k$	16
รูป 1-8 แสดงตัวอย่าง โพล ซีโร และ สเปกตรัมของระบบเชิงอุปมาน	16
รูป 1-9 แสดงโพล ซีโร และ สเปกตรัมของระบบเชิงเลข	17
รูป 1-10 แสดงความสัมพันธ์ของโพล $s_k$ ในระบบ S และ Z	18
รูป 1-11 แสดงความสัมพันธ์ระหว่างตัวแปร ลาปลาซ กับตัวแปร Z	18
รูป 1-12 แสดงการส่งค่าจินตภาพจากระนาบ S ไปยังเส้นรอบวงกลมในระนาบ Z	22
รูป 1-13 แสดงการส่งค่าทางด้านซ้ายของระนาบ S เข้าไปในวงกลมของระนาบ Z	22
รูป 1-14 แสดงข้อกำหนดในการออกแบบวงจรรองผ่านความถี่ต่ำ	23
รูป 1-15 แสดงผลตอบสนองแอมพลิจูดอุดมคติ	24
รูป 1-16 แสดงลำดับขั้นตอนการออกแบบตัวกรองโดยใช้การแปลงความถี่	25
รูป 1-17 แสดงแผนภาพการแปลงความถี่จากตัวกรองผ่านความถี่ต่ำ เป็นตัวกรองผ่านแถบความถี่	25
รูป 1-18 แผนแสดงการส่งโดยใช้การแปลงเชิงเส้นคู่	30
รูป 1-19 แสดงคุณสมบัติเป็นคาบของค่าความถี่เชิงเลขจากการแปลงเชิงเส้นคู่	30
รูป 1-20 รูปร่างทั่วไปของตัวกรองเชิงอุปมาน	31
รูป 1-21 ซีโร โพล Plot ของ $H(s)$ และ $L(-s^2)$	32
รูป 1-22 แสดง Nonideal loss characteristic	32
รูป 1-23 แสดงค่า Loss characteristic ของ Fourth-order normalized chebyshev filter	33
รูป 1-24 แสดงการ plot ค่า F และ W ที่สัมพันธ์กับแกน $\theta$	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
รูป 1-25 แสดง ซีโร-โพล plot ของตัวกรองเชบีเชฟ 15 <sup>th</sup> -order	40
รูป 1-26 โครงสร้างวงจรป้อนกลับเชิงเลข โดยตรง	42
รูป 1-27 โครงสร้างรูปแบบโดยตรง 2 หรือ แบบบัญญัติ	45
รูป 1-28 แสดงคุณสมบัติของตัวกรองหน่วงเวลา	46
รูป 1-29 แสดงผลตอบสนองเฟสและแอมพลิจูดของตัวกรองหน่วงเวลา	48
รูป 1-30 ผลตอบสนองแอมพลิจูดอคมคิต	51
รูป 1-31 สเปกตรัมของฮานและแฮมมิงวินโดส์	54
รูป 1-32 สเปกตรัมของวินโดว์สี่เหลี่ยม และฮานวินโดว์กรณี N = 11	55
รูป 1-33 แสดงผลตอบสนองความถี่จากการใช้วินโดว์แบบต่างๆ	59
รูป 2-1 ขนาดและอัตราส่วนของ TMS320C542 DSK CIRCUIT BOARD	60
รูป 2-2 อุปกรณ์โดยทั่วไปของ TMS320C542 DSK BOARD	61
รูป 2-3 แสดงลักษณะของ IC # TLC320AC01C	63
รูป 2-4 DSK Block Diagram	70
รูป 2-5 Memory Map of the DSK	76
รูป 2-6 Block Diagram แสดงโครงสร้างภายในทั้งหมดของ C5x	77
รูป 2-7 แสดง Block Diagram การทำงานของ Timer ใน C5x	93
รูป 2-8 การต่อใช้งาน Board DSK	95
รูป 3-1 แสดงโครงสร้างทางฮาร์ดแวร์ของโครงการ	96
รูป 3-2 บล็อกไดอะแกรมของระบบ DSP ที่ใช้ในโครงการ	97
รูป 3-3 แสดงคุณสมบัติของตัวกรองอคมคิตแบบต่างๆ	100
รูป 3-4 แสดงพล็อตฮาร์ดของโปรแกรมที่ใช้คำนวณตัวกรองบน TMS	105
รูป 4-1 กราฟแสดงการตอบสนองความถี่	117
รูป 4-2 แสดงสัญญาณ Input และ Output ที่ความถี่ 2.4 KHz	118
รูป 4-3 แสดงสัญญาณ Input และ Output ที่ความถี่ 6 KHz ที่เป็นจุด Cut off	118
รูป 4-4 แสดงสัญญาณ Input และ Output ที่ความถี่ 7 KHz	119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
รูป 4-5 แสดงสัญญาณ input $\sin 2 \text{ KHz} + \sin 283 \text{ Hz}$ และสัญญาณ output ที่ $FC = 0.4 \text{ Hz}$ ของ FIR lowpass filter	121
รูป 4-6 แสดงสัญญาณ input ที่ความถี่ $\sin 5 \text{ KHz} + \sin 545 \text{ Hz}$ และสัญญาณ output ที่ $FC = 0.4$ ของ HIGH PASS FILTER	122
รูป 4-7 แสดงซิงทิลเลชั่นแยกออกจากสัญญาณดาวเทียม โดย High Pass Filter	123
รูป 4-8 แสดงเรนเอทเทนนูเอชั่นแยกออกจากสัญญาณดาวเทียม โดย Low Pass Filter	123



## สารบัญตาราง

ตารางที่		หน้า
ตารางที่ 1-1	สรุปคุณสมบัติของวินโดว์	56
ตารางที่ 2-1	แสดงรายการของการใช้งาน C54X ในด้านต่างๆ	66
ตารางที่ 2-2	แสดงภาพรวมของความสามารถในรุ่น C54x	68
ตารางที่ 2-3	Serial Port Register	72
ตารางที่ 2-4	Memory – Mapped Register And I/O Ports	81
ตารางที่ 2-5	Local Data Memory Configure Control	89
ตารางที่ 2-6	On-chip Single-Access RAM Configuration Control	89
ตารางที่ 2-7	การใช้งาน Bit ต่างๆของ CWSR	91
ตารางที่ 2-8	ความสัมพันธ์ของค่า Wait State กับค่าใน PDWSR และ IOWSR	91
ตารางที่ 3-1	แสดงหน้าต่างแบบต่างๆ	99
ตารางที่ 4-1	การตอบสนองความถี่	120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### ตัวกรองเชิงเลข

#### 1.1 ตัวกรองเชิงเลข

ตัวกรองอาจแบ่งออกเป็น 2 ประเภทคือ ตัวกรองไม่ป้อนกลับแบบเชิงเลขและตัวป้อนกลับเชิงเลข ตัวกรองทั้งสองประเภทนี้มีคุณสมบัติ และการออกแบบที่แตกต่างกันออกไป เราจะศึกษาถึงคุณสมบัติทั่วไปของการกรองทั้งสองประเภท รวมทั้งข้อเปรียบเทียบระหว่างการกรองทั้งสองประเภท นอกจากการกรองเชิงเลขที่มีการประยุกต์ใช้กับระบบวงจรอิเล็กทรอนิกส์ ซึ่งส่วนใหญ่เป็นการกรองในระบบเวลาจริงแล้ว การประมวลผลการกรองเชิงเลขในระบบเวลาไม่จริง (non-real time system) ได้ นอกจากนั้นในบทนี้ยังได้กล่าวถึง ทฤษฎีการสุ่มตัวอย่างสัญญาณและผลจากการสุ่มตัวอย่างสัญญาณด้วย

##### 1.1.1 การกรองเชิงเลข

ในระยะ 2 ทศวรรษที่ผ่านมา ได้มีการนำเอาหลักการของการกรองประมวลผลสัญญาณเชิงเลข ไปประยุกต์ใช้กันอย่างกว้างขวาง โดยเฉพาะกับการประยุกต์ใช้ในเกือบทุกสาขาวิชาของสาขา วิศวกรรมอิเล็กทรอนิกส์ และ วิศวกรรมไฟฟ้า ยกตัวอย่างเช่น ในการสื่อสารข้อมูล ในระบบเรดาร์โซนาร์ ในระบบเครื่องมือวัด และระบบควบคุมในอุตสาหกรรมต่างๆ เหตุผลที่เกิดพัฒนาการและมีประยุกต์ใช้กันมากนี้ ก็เนื่องมาจากหลักการของการประมวลผลสัญญาณเชิงเลข เหมาะสมเป็นอย่างยิ่งที่จะนำเอาคอมพิวเตอร์หรือไมโครโปรเซสเซอร์ ซึ่งในปัจจุบันมีราคาถูกลงเป็นอย่างมาก มาเป็นตัวประมวลผลสัญญาณ ผลที่ได้ก็คือเราได้ระบบการประมวลผลสัญญาณที่มีประสิทธิภาพ ความเที่ยงตรง ความแม่นยำ และความเชื่อถือได้สูง นอกจากนี้ยังมีความคล่องตัวกว่าระบบการประมวลผลสัญญาณเชิงอุปมานมากด้วย

นอกจากการประมวลผลสัญญาณเชิงเลขในลักษณะมิติเดียว หรือ 1-D(one dimensional) ยังมีการประมวลผลสัญญาณในลักษณะ สองมิติ หรือ 2-D (Two Dimension) ที่มีการประยุกต์ใช้งานของระบบการประมวลผลสัญญาณเชิงเลขกันอยู่มากเช่นกัน เช่น การประมวลผลเชิงเลขของ ภาพรังสีเอกซ์ (X-ray image) การวิเคราะห์ภาพถ่ายทางอากาศ (aerial photograph) เพื่อตรวจดูถึงปรากฏการณ์ต่างๆของโลก การวิเคราะห์ภาพถ่ายสถานะทางอากาศจากดาวเทียม (satellite weather photograph) การทำให้สัญญาณโทรภาพจากยานสำรวจอวกาศมีความชัดเจนขึ้น การวิเคราะห์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณความไหวสะเทือน (seismic signal) ที่มีใช้ในการตรวจหาแหล่งน้ำมันหรือ การตรวจวัดสั่นสะเทือนของพื้นโลก (earth quake) เหล่านี้เป็นตัวอย่างของลักษณะการประมวลผลสัญญาณเชิงเลขทั้งสิ้น ซึ่งกล่าวโดยทั่วไปแล้วเราสามารถนำเอาหลักการของ การประมวลผลสัญญาณมิติเดียว มาประยุกต์ใช้กับการประมวลผลสัญญาณสองมิติได้

ระบบการประมวลผลสัญญาณเชิงเลขที่ได้กล่าวถึงไปนั้น การกรองเชิงเลข เป็นกระบวนการประมวลผลสัญญาณอันหนึ่งที่มีความสำคัญ โดยที่ การกรองเชิงเลข มักจะเกี่ยวเนื่องกับการกำจัดสัญญาณรบกวน (noise) และ ลดความผิดเพี้ยนของสัญญาณ การที่ในปัจจุบัน ระบบการกรองเชิงเลขมีบทบาทสำคัญมากขึ้นก็อาจเนื่องมาจาก

- (ก) การที่มีวิธีการออกแบบตัวกรองที่ง่ายแต่ใช้งานได้อย่างมีประสิทธิภาพ
  - (ข) พัฒนาการของการออกแบบวงจร VLSI ที่ทำให้ การคูณตัวเลข การเก็บข้อมูล ของสัญญาณเชิงเลข ทำได้รวดเร็วขึ้นมาก และ
  - (ค) การที่มี คอมพิวเตอร์ฮาร์ดแวร์ และซอฟต์แวร์ที่มีประสิทธิภาพสูง รวมถึงการที่มีระบบไมโครโปรเซสเซอร์ และระบบวงจรเชิงเลข ที่มีประสิทธิภาพสูงด้วย
- ด้วยเหตุผลเหล่านี้ ทำให้การกรองเชิงเลขถูกนำไปประยุกต์ใช้ในการประมวลผลสัญญาณที่สำคัญๆ เช่น การสื่อสาร โทรคมนาคม ระบบเรดาร์ โซนาร์ หรือระบบอากาศยาน เป็นต้น
- จากที่กล่าวมาแล้วว่าการกรองเชิงเลขเป็นกระบวนการที่ไปตัดแปลงสเปกตรัมของสัญญาณให้มีสเปกตรัมเป็นไปตามข้อกำหนดที่ต้องการ หรือ กำหนดให้ซึ่งอาจเป็นการ เพิ่มค่า หรือ ลดทอน ค่าขนาดของสัญญาณในแถบความถี่ที่ต้องการออกมาใช้งาน ก็ได้ ซึ่งกระบวนการเหล่านี้ จุดประสงค์หลักก็เพื่อต้องการ (ก) ลดความผิดเพี้ยนของสัญญาณ หรือลดสิ่งรบกวนที่อาจเป็นผลมาจากการส่งสัญญาณไปใน สายส่งหรือตัวกลางที่มีคุณภาพไม่สมบูรณ์ หรือมีสิ่งรบกวนมาก หรือ (ข) ต้องการแยกสัญญาณออกจากสัญญาณใน โดเมนเวลาออกเป็นองค์ประกอบของสัญญาณใน โดเมนความถี่ หรือ (ง) ใช้ในการคิโมดูเลทสัญญาณ (demodulation) หรือ (จ) ใช้ในการแปลงสัญญาณเชิงเต็มหน่วยออกเป็นองค์ประกอบทางความถี่ เป็นต้น

ตัวกรองเชิงเลขดังที่เคยได้กล่าวมาข้างต้นว่า เป็นระบบการประมวลผลแบบเวลายืนยงเชิงเส้นแบบหนึ่ง เนื่องจากการที่มีประยุกต์ใช้อย่างมากมายในสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ ดังนั้นวิธีการออกแบบ (design) และการสร้าง (implementation) ที่สามารถนำไปใช้งานได้จริงในทางปฏิบัติ จึงได้รับความสนใจและมีการพัฒนากันอย่างกว้างขวางนั้น อาจมาจากข้อได้เปรียบหลายประการดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) ผลตอบสนองความถี่ของตัวกรอง สามารถออกแบบให้มีความใกล้เคียงกับผลตอบสนองความถี่ที่กำหนดให้ หรือ ผลตอบสนองความถี่ที่ต้องการได้ นอกจากนี้การออกแบบตัวกรองให้มี ผลตอบสนองเฟสเชิงเส้น ก็ทำได้ง่าย

(ข) คุณสมบัติของตัวกรองที่ออกแบบและสร้างแล้วจะไม่ ขยับเลื่อน (drift) ไปตามสภาวะแวดล้อม หรือตามอุณหภูมิ หรือระยะเวลาการใช้งาน นอกจากนี้ยังสามารถใช้งานย่านความถี่ต่ำได้เป็นอย่างดี

(ค) การประยุกต์ใช้งานเป็นตัวกรองแบบ ปรับตัวได้(adaptive) ทำได้ง่าย

(ง) ผู้ออกแบบสามารถออกแบบ โดยกำหนดถึง ความยาวของคำ (wordlength) ของตัวเลขฐานสองที่ต้องการใช้ และยังสามารถออกแบบให้มีผลตอบสนองความถี่ ตามที่ต้องการได้

(จ) จนถึงปัจจุบัน ถ้าพิจารณาในแง่ของเสถียรภาพของตัวกรอง ความเชื่อถือ ราคา หรือขนาดของตัวกรองเชิงเลข สิ่งเหล่านี้กำลังได้รับการพัฒนาและปรับปรุง และมีแนวโน้มว่าจะให้ผลลัพธ์ที่ดีกว่าของตัวกรองเชิงอุปมานมาก

อย่างไรก็ตามถ้าพิจารณาในแง่ของการสร้างใช้งาน โดยเฉพาะการสร้างใช้งานโดยใช้ เลขฐานสองที่มีความยาวจำกัด (finite wordlength) นั้น การออกแบบและการสังเคราะห์ตัวกรองเชิงเลขแบบนี้มีความยุ่งยากมากขึ้น เนื่องจากต้องนำผลของปรากฏการณ์ที่ไม่เป็นเชิงเส้น ดังสามารถเขียนสรุปได้ดังนี้คือ

(ก) เสถียรภาพของตัวกรองที่ได้รับการออกแบบและตรวจสอบมาเป็นอย่างดี เพื่อให้ได้ผลตอบสนองความถี่ตามที่ต้องการเสถียร เมื่อนำไปสร้างใช้งานอาจเกิดการไม่เสถียรขึ้นได้

(ข) การคำนวณต่างๆ โดยเฉพาะ การคูณตัวเลข เราต้องทำการปัดเศษ (rounding) หรือ ตัดปลาย (truncate) ตัวเลขหลังการคูณแต่ละครั้ง ผลนี้อาจทำให้เกิด สิ่งรบกวนหรือสัญญาณรบกวนที่สัญญาณออกได้ ถ้าในตัวกรองมีการคูณตัวเลขมากสัญญาณรบกวนจากการคูณก็มากด้วย

(ค) เมื่อนำไปสร้างใช้งานโดยใช้ตัวเลขฐานสอง โดยเฉพาะการใช้ตัวเลขแบบส่วนเติมเต็ม มากจนตัวเลขเลยไปทับปิดเครื่องหมาย(sign bit) หรือเกิดการล้น หรือ โอเวอร์โฟลล์ (overflow) ขึ้น การล้นนี้อาจทำให้ที่สัญญาณออกของตัวกรองเกิดการกวัดแกว่งขึ้นได้ กรณีนี้ทำให้ตัวกรองกลายเป็นวงจรกวัดแกว่งไป

ปรากฏการณ์ต่างๆที่ได้กล่าวมานี้ ในทางปฏิบัติอาจเกิดหรือไม่เกิดขึ้นก็ได้ ขึ้นอยู่กับว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำไปสร้างโดยใช้ความยาวค่าจำกัดนั้น ฟังก์ชันถ่ายโอนของตัวกรองถูกสร้างโดยใช้โครงสร้าง (structure) ของตัวกรองแบบใด ซึ่งเมื่อถึงจุดนี้เราจำเป็นต้องศึกษาเพื่อค้นคว้าหาโครงสร้างของตัวกรองที่สามารถหรือ อย่างน้อยเกิดผลของปรากฏการณ์ไม่เป็นเชิงเส้นน้อยที่สุด เราจะได้กล่าวถึงโครงสร้างของตัวกรองเหล่านี้อีกครั้งในภายหลัง

ดังนั้นสำหรับการออกแบบสร้างตัวกรองเชิงเลข โดยให้มีคุณสมบัติ ผลตอบสนองความถี่ตามอุดมคติ (ideal response) หรือ มีผลตอบสนองความถี่ตามต้องการนั้น อาจกล่าวโดยทั่วไปได้ว่าผู้ออกแบบและสร้างต้องนำองค์ประกอบสำคัญต่างๆ มาร่วมพิจารณาด้วย ซึ่งเขียนสรุปได้ดังนี้

- (ก) เป็นการกำหนดคุณสมบัติของตัวกรอง ในโดเมนความถี่ เช่น ค่าขนาด หรือค่าความหน่วงกลุ่มหรือว่าเป็นการกำหนดคุณสมบัติของตัวกรองใน โดเมนเวลาเช่นผลตอบสนองอิมพัลส์
- (ข) สำหรับตัวกรองป้อนกลับแบบเชิงเลข ที่เราเขียนฟังก์ชันถ่ายโอนของมัน เป็นอัตราส่วนของพหุนาม (polynomial) สองพหุนามนั้น ต้องออกแบบให้ได้ตัวกรองที่มีเสถียรภาพดี
- (ค) เมื่อได้ฟังก์ชันถ่ายโอนแล้วเราต้องทำการเลือกโครงสร้างตัวกรองให้เหมาะสมเพื่อที่ว่าเมื่อนำไปสร้างใช้งานโดยใช้ตัวเลขของความยาวของค่าจำกัดแล้ว ไม่เกิดปรากฏการณ์ไม่เป็นเชิงเส้นขึ้น หรือถ้าหลีกเลี่ยงไม่ได้ก็ให้เกิดปรากฏการณ์ไม่เป็นเชิงเส้นขึ้นน้อยที่สุด

## 1.1.2 ตัวกรองป้อนกลับเชิงเลข (Recursive digital filter)

### 1.1.2.1 การอินทิเกรตเชิงเลข

ตัวอย่างของตัวกรองเชิงเลขอีกรูปแบบหนึ่งที่มีการใช้งานกัน ก็คือในระเบียบวิธีเชิงเลข (numerical method) ก็คือ การอินทิเกรตเชิงเลข (numerical integration) ซึ่งหลายเรามาพิจารณาการอินทิเกรตเชิงเลข แบบสี่เหลี่ยมคางหมู (trapezoidal) ซึ่งลำดับขั้นตอนวิธีของมัน เขียนเป็นสมการตามลำดับได้ คือ

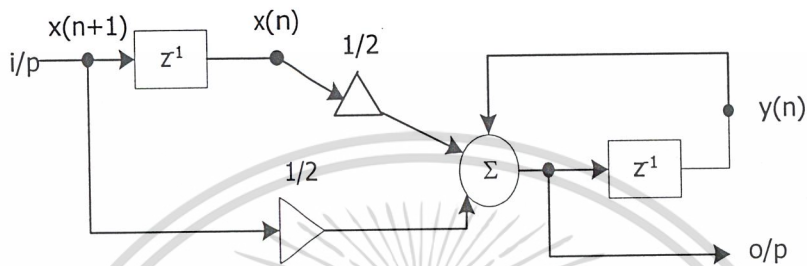
$$Y(0) = 0$$

$$\begin{aligned} y(1) &= y(0) + (1/2)\{x(1) + x(0)\} \\ y(2) &= y(1) + (1/2)\{x(2) + x(1)\} \end{aligned} \quad (1.1)$$

$$y(n+1) = y(n) + (1/2)\{x(n+1) + x(n)\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราให้แต่ละชิ้น (strip) ของสี่เหลี่ยมคางหมู มีค่าเป็น 1 หน่วย จะเห็นได้ว่าวงจรรองเชิงเลขที่ใช้คำนวณสมการ (1.1) อาจเขียนแทนด้วยแผนภาพของวงจรรองเชิงเลขดังแสดงในรูป (1-1) ได้



รูป 1-1 แผนภาพวงจรรองเชิงเลขของสมการ (1.1)

สำหรับการอินทิเกรตเชิงเลขแบบอื่นนั้น จะขอเพียงสมการโดยไม่เขียนแผนภาพของตัวกรองประกอบเช่นการอินทิเกรต โดยใช้กฎของซิมป์สัน (Simpson's rule) คือ

$$y(n+1) = y(n-1) + (1/3)\{x(n+1) + 4x(n) + x(n-1)\} \quad (1.2)$$

หรือการอินทิเกรต โดยใช้กฎของทิก (Tick's rule)

$$y(n+1) = y(n-1) + 0.3584x(n+1) + 1.2832x(n) + 0.3584x(n-1) \quad (1.3)$$

หรือ การอินทิเกรต โดยใช้กฎของจุดกึ่งกลาง (mid-point rule) เป็นต้น

$$y(n+1) = y(n) + x(n+1/2) \quad (1.4)$$

การอินทิเกรตแบบต่างๆ ที่ได้กล่าวมานี้ อาจกล่าวได้ว่าเป็นแบบหนึ่งของ การประมวลผลสัญญาณ โดยใช้ตัวกรองป้อนกลับเชิงเลข ซึ่งตัวป้อนกลับแบบนี้ อาจเขียนแทนได้ด้วยสมการทั่วไปคือ

$$y(n) = \sum_{k=-\infty}^{\infty} a_k x(n-k) - \sum_{k=1}^{\infty} b_k y(n-k) \quad (1.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $b_0=1$  และสมการ (1.5) เป็นสมการที่สามารถนำไปสร้างเป็นวงจรกรองได้ หรือเป็นตัวกรองภาพ เมื่อให้  $b_k=0$  เมื่อ  $k < 1$  แต่สำหรับกรณีค่า  $b_k \neq 0$  เมื่อ  $k < 1$  จะสามารถนำไปสร้างไปเป็นตัวกรองได้เช่นเดียวกัน แต่เป็นวงจรกรองแบบไม่เป็นระบบเวลาจริง ซึ่งตัวกรองแบบนี้มีประโยชน์ประยุกต์มากเช่นกัน

ที่นี้เรามาตรวจสอบสนองความถี่ของการอินทิเกรตเชิงเลขตามสมการ (1.2) หรือ (1.4) ว่าเป็นอย่างไร โดยการคำนวณจะแสดงเฉพาะกรณีแบบสี่เหลี่ยมคางหมูเท่านั้น ส่วนผลตอบสนองความถี่ของตัวกรองแบบอื่น ก็หาได้โดยใช้วิธีการเดียวกัน จากรูป (1-1) สัญญาณออกของวงจรเขียนได้เป็น

$$y(n+1) = y(n) + (1/2)\{x(n+1) + x(n)\}$$

ทำการแปลงแซด เข้ากับทั้งสองข้างของสมการนี้ แล้วทำการแทนค่า  $Z = \exp(jw)$  จะได้

$$Y(w)\exp(jw) = Y(w) + (1/2)\{X(w)\exp(jw) + X(w)\}$$

ทำการหารตลอดด้วยพจน์  $X(w)$  ได้ผลตอบสนองความถี่ของวงจรกรองเป็น

$$H(w)\exp(jw) = H(w) + (1/2)\{\exp(jw)\} \quad (1.6)$$

หรือ

$$\begin{aligned} H(w) &= (1/2)\{\exp(jw) + 1\} / \{\exp(jw) - 1\} \\ &= \cos(w/2) / (2j \sin(w/2)) \end{aligned} \quad (1.7)$$

และสำหรับ ตัวอินทิเกรตอุดมคติ ให้ผลตอบสนองความถี่เป็น

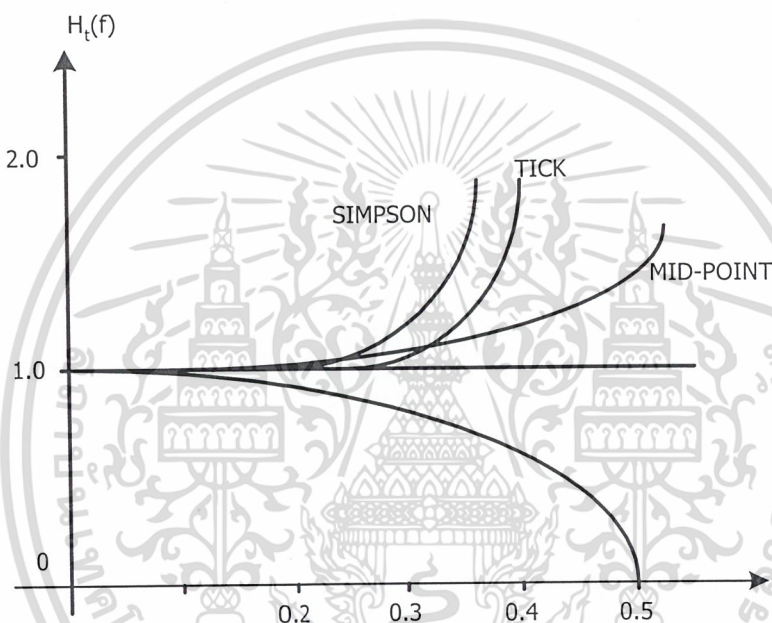
$$H_c(w) = 1 / jw \quad (1.8)$$

เพราะฉะนั้น เมื่อนำมาหาอัตราส่วนระหว่าง  $H(w)$  กับ  $H_c(w)$  จะได้

$$H(w) / H_c(w) = (w/2) \cot(w/2) \quad (1.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งผลที่ได้แสดงไว้ในรูป (1-2) เห็นได้ชัดว่าผลตอบสนองความถี่ของการอินทิเกรตแบบสี่เหลี่ยมคางหมูเมื่อเทียบกับตัวอินทิเกรตจริงแล้ว มีค่าผิดพลาดมาก โดยเฉพาะในย่านความถี่สูงค่าผิดพลาดก็เพิ่มมากขึ้น ในรูป (1-2) ได้วาดผลตอบสนองความถี่ของการอินทิเกรตแบบอื่นไว้ด้วย เห็นได้ว่าแบบอื่น แต่ก็มีข้อเสียคล้ายแบบซิมป์สัน คือ ถ้าความถี่สูงขึ้นค่าความผิดพลาดมาก และอาจมีค่าผิดพลาดเข้าสู่อนันต์ได้ ดังนั้นเมื่อเทียบในตลอดย่านความถี่ทั้งหมด อาจกล่าวได้ว่า แบบจุดกึ่งกลางให้ผิดพลาดน้อยที่สุด



รูป 1-2 แสดงผลตอบสนองความถี่ของอินทิเกรเตอร์เชิงเลข

#### 1.1.2.2 ตัวกรองป้อนกลับเชิงเลขทั่วไป

โดยทั่วไปถ้าให้สัมประสิทธิ์ของตัวกรองเชิงเลขมีจำนวนจำกัด จากสมการ (1.5) เราอาจเขียนสมการผลต่างสืบเนื่องอันดับที่  $M$  ( $M^{\text{th}}$  order difference equation) สำหรับตัวกรองระบบเวลาจริงได้

$$y(nT) = \sum_{k=0}^N a_k x(nT - kT) - \sum_{k=1}^M b_k y(nT - kT) \quad (1.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในที่นี้เขียนให้  $b_0 = 1$  และเมื่อทำการแปลงแซดทั้งสองข้างจะได้

$$Z\{y(nT)\} + Z\left\{\sum_{k=1}^N b_k y(nT - kT)\right\} = Z\left\{\sum_{k=0}^N a_k x(nT - kT)\right\} \quad (1.11)$$

หรือ ได้ฟังก์ชันถ่ายโอนเป็น

$$H(z) = Y(z) / X(z) = \sum_{k=0}^N a_k z^{-k} / \left\{1 + \sum_{k=1}^M b_k z^{-k}\right\} \quad (1.12)$$

โดยที่ถ้าให้  $b_0 = 1$  และโดยการแทนค่าให้  $Z = \exp(jw)$  เราจะได้ผลตอบสนองความถี่ของวงจรกรอง  
ป้อนกลับเชิงเลขแบบทั่วไป คือ

$$H(e^{jwT}) = H(w) = \sum_{k=0}^N a_k e^{-jkwT} / \left\{1 + \sum_{k=1}^M b_k z^{-k}\right\} \quad (1.13)$$

### 1.1.2.3 โพล และ ซีโร ของฟังก์ชันถ่ายโอน

พหุนามเศษ และพหุนามส่วน ของฟังก์ชันถ่ายโอน ของวงจรป้อนกลับเชิงเลขทั่วไป ในสมการ (2.11) อาจเขียนแยกตัวประกอบในพจน์ของ โพล (pole) และ ซีโร (zero) ได้เช่นเดียวกับระบบเชิงอุปมาน สมมติให้  $H(z)$  มี โพล อยู่  $M$  โพล ที่ตำแหน่ง  $Z = P_1, P_2, P_3, \dots, P_M$  และมี ซีโร อยู่  $N$  ตัว ที่ตำแหน่ง

$Z = Z_1, Z_2, \dots, Z_M$  ดังนั้นฟังก์ชันถ่ายโอนของระบบเขียนใหม่ได้คือ

$$H(z) = A \prod_{i=1}^N (1 - z_i z^{-1}) / \prod_{i=1}^M (1 - P_i z^{-1}) \quad (1.14)$$

โดยที่  $A$  แทนค่าขยายของระบบ

ดูตัวอย่างสำหรับระบบที่มี 2 ซีโร ที่  $Z_1$  และ  $Z_2$  และมี 3 โพลที่  $P_1, P_2$  และ  $P_3$  เราสามารถหาผลตอบสนองความถี่ของวงจรกรองได้โดยวิธีการพหุคูณ โดยทำให้  $R_1$  และ  $R_2$  เป็นระยะห่างจากซีโร  $Z_1$  และ  $Z_2$  ถึงจุดบนวงกลมหนึ่งหน่วยตามลำดับ และถ้าให้  $R_3, R_4$  และ  $R_5$  หาผลตอบสนองแอมพลิจูด

เอกสารนี้จะได้ออกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

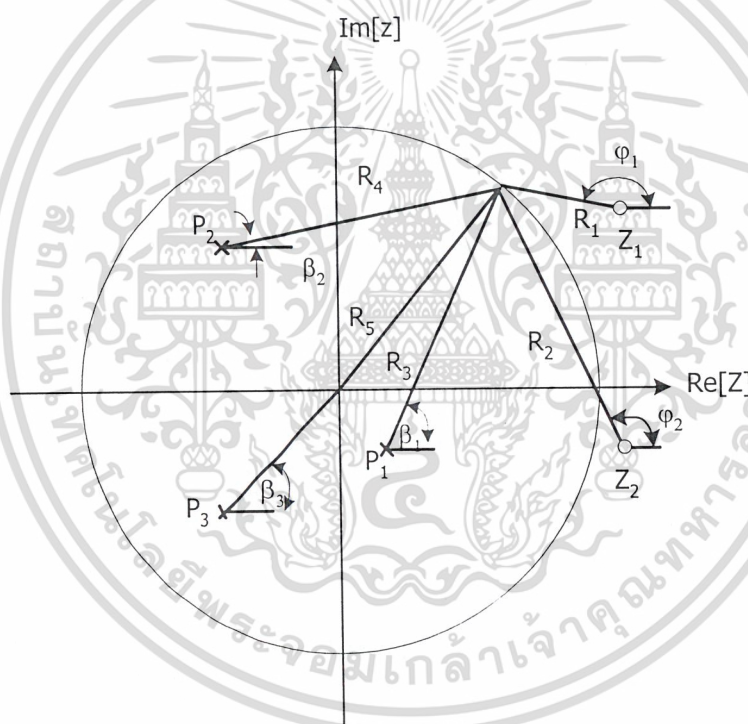
$$|H(z)| = \frac{|z - z_2| |z - z_1|}{|z - p_1| |z - p_2| |z - p_3|} \quad (1.15)$$

นั่นคือ  $|H(e^{j\omega T})| = R_1 R_2 / (R_3 \cdot R_4 \cdot R_5)$  (1.16)

และผลตอบสนองเฟสเป็น

$$\theta = \psi_1 + \psi_2 - (\beta_1 + \beta_2 + \beta_3) \quad (1.17)$$

ค่าต่างๆตามสมการ (1.16) และ (1.17) ได้แสดงไว้ในรูป (1-3)



รูป 1-3 แสดงการหาผลตอบสนองความถี่ของตัวกรองเชิงเลขโดยวิธีกราฟ

#### 1.1.2.4 เสถียรภาพของตัวป้อนกลับเชิงเลข

ดังที่เคยกล่าวในหัวข้อที่แล้วว่า ฟังก์ชันถ่ายโอนของวงจรรองเชิงเลขสามารถเขียนอยู่ในพจน์ของ โพล และ ซีโร ได้ ดังนั้นเราอาจพิจารณาเสถียรภาพของวงจรรองโดยดูที่ตำแหน่งโพล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสมการ (1.14) ซึ่งจะเห็นได้ว่าวงจรกรองป้อนกลับเชิงเลขจะเสถียรก็ต่อเมื่อโพลของฟังก์ชันถ่ายโอน วางตัวอยู่ในวงกลมหนึ่งหน่วยของระนาบเชิงซ้อน  $Z$  (หรือวางตัวอยู่นอกวงกลมหนึ่งหน่วยของระนาบเชิงซ้อน  $Z^{-1}$ ) หรือ

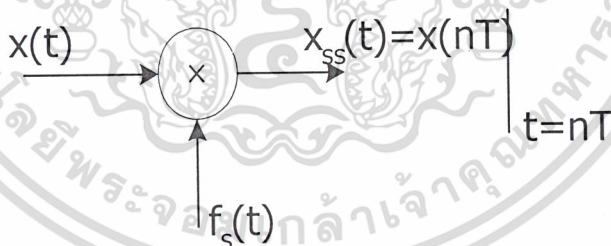
$$|z| = 1$$

จึงอาจกล่าวได้ว่า ถ้าหากฟังก์ชันถ่ายโอนของตัวกรองเชิงเลขอยู่ในพจน์ของตัวกรองอันดับหนึ่งที่อยู่ตรงกันดังสมการ (1.13) แล้ว ตัวกรองเชิงเลขแบบเหตุภาพ จะเสถียร เมื่อโพลทุกตัวของตัวกรอง มีตำแหน่งอยู่ภายในวงกลมหนึ่งหน่วยของระนาบเชิงซ้อน

### 1.1.3 การสุ่มตัวอย่าง และ ผลจากการสุ่ม

#### 1.1.3.1 การสุ่มตัวอย่าง

รูป (1-4) แสดงการสุ่มตัวอย่างในโดเมนเวลาก็คือ เราทำการคูณสัญญาณเชิงอุปมาน  $x(t)$  ด้วยลำดับของอิมพัลส์หนึ่งหน่วย  $f_s(t)$  ตามรูป (1-5) โดยที่อิมพัลส์แต่ละตัวสมมติให้มีความห่างกันเท่ากับ  $T$  วินาที โดยทั่วไปลำดับ อิมพัลส์หนึ่งหน่วย หรือ  $f_s(t)$



รูป 1-4 ตัวแบบแทนการสุ่มตัวอย่างสัญญาณเชิงอุปมาน

โดยที่อิมพัลส์แต่ละตัวสมมติให้มีความห่างกันเท่ากับ  $T$  วินาที โดยทั่วไปลำดับ อิมพัลส์หนึ่งหน่วย หรือ  $f_s(t)$  อาจเขียนแทนด้วยสมการ

$$f_s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (1.18)$$

เนื่องจาก  $f_s(t)$  เป็นสัญญาณที่มีลักษณะเป็นคาบ เพราะฉะนั้นจึงสามารถเขียนแทน  $f_s(t)$  ได้ด้วยอนุกรมฟูรีเยอร์ คือ

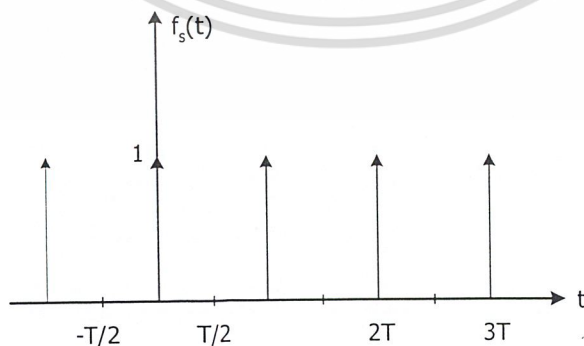
$$f_s(t) = \sum_{n=-\infty}^{\infty} C_n \cdot \exp(j2n\pi/T) = \sum_{n=-\infty}^{\infty} C_n \cdot \exp(jn\omega_s t) \quad (1.19)$$

โดยที่  $\omega_s = 2\pi/T$  และ  $C_n$  คือค่าขนาดเชิงซ้อน (complex amplitude) ของตัวประกอบความถี่เชิงซ้อน (complex frequency component)  $\omega_s$  และ  $C_n$  หาค่าได้ตามลำดับ คือ

$$\begin{aligned} C_n &= (1/T) \int_{-T/2}^{T/2} f_s(t) \cdot \exp(-jn\omega_s t) dt \\ C_n &= (1/T) \int_{-T/2}^{T/2} \delta(t) \cdot \exp(-jn\omega_s t) dt \\ C_n &= (1/T) \int_{-T/2}^{T/2} \{ \exp(-jn\omega_s t) \} \Big|_{-T/2}^{T/2} = 1/T \end{aligned} \quad (1.20)$$

เมื่อแทนค่า  $C_n$  จาก (1.20) กับในสมการ (1.19) จะได้

$$f_s(t) = (1/T) \sum_{n=-\infty}^{\infty} \exp(jn\omega_s t) \quad (1.21)$$



รูป 1-5 แสดงสัญญาณสุ่มตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำ การแปลงฟูริเยอร์ เพื่อทำการหาค่าสเปกตรัมความถี่ของ  $f_s(t)$  จะได้

$$F_s(w) = Z\{f_s(t)\} = (2\pi / T) \sum_{-\infty}^{\infty} \delta(w - nw_s) \quad (1.22)$$

นั่นคือ ถ้าพิจารณาจากสมการ (1.18) และ (1.19) จะเห็นได้ว่า

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) \rightarrow w_o \sum_{n=-\infty}^{\infty} \delta(w - nw_s) \quad (1.23)$$

บรรทัดข้างบนนี้เห็นได้ชัดว่า เมื่อพิจารณาในโดเมนความถี่ สเปกตรัมความถี่ของตัวส่งสัญญาณ  $f_s(t)$  เป็นอิมพัลส์ที่วางตัวห่างเท่าๆกัน ไปบนแกนความถี่เช่นกัน ดังแสดงในรูป (1-6b)

ถ้าเขียนให้  $X_{ss}(t)$  เป็นสัญญาณที่ได้จากการสุ่มตัวอย่าง ดังนั้น

$$x_{ss}(t) = F_s(t) \cdot x(t) \quad (1.24)$$

และถ้าให้  $x(w)$  เป็นสเปกตรัมความถี่ของ  $x(t)$  เนื่องจากโดเมนเวลาสัญญาณที่ได้จากการสุ่มตัวอย่าง เป็นการคูณกับสัญญาณ ดังนั้นใน โดเมนความถี่ จึงเป็นการประสานของสเปกตรัม  $f_s(t)$  และ  $x(t)$

$$X_{ss}(t) = F_s(w) x(w) \quad (1.25)$$

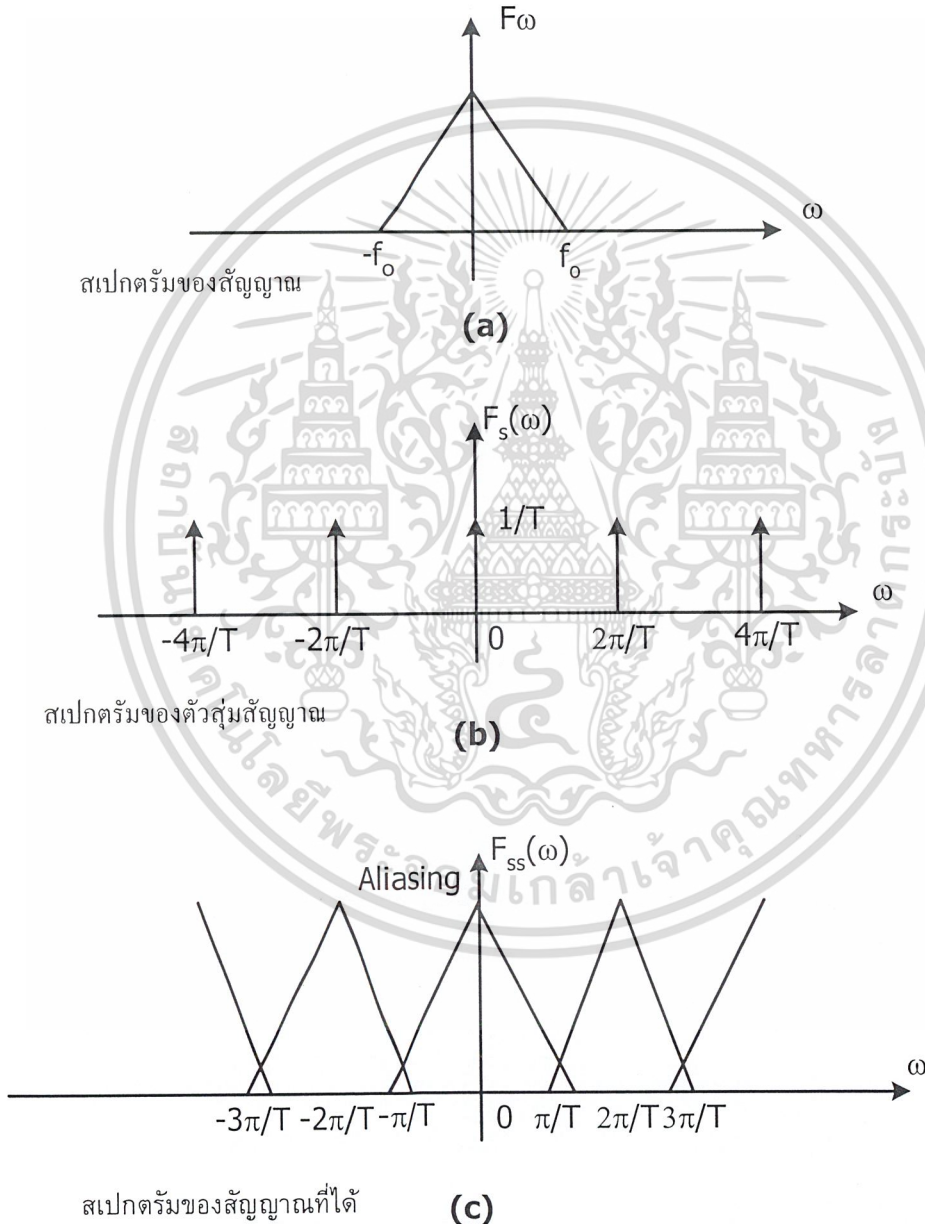
โดยที่สัญลักษณ์ \* แทนการคูณโดยการประสาน ผลการประสานนี้ได้แสดงเป็นภาพดังในรูป (1-6c) ซึ่งเห็นได้ชัดว่า จากการที่สัญญาณเชิงอุปมานที่มีสเปกตรัมความถี่เป็นช่วงเดียวโดดๆ และสมมติให้สเปกตรัมนี้มีแถบความถี่ปฏิบัติงานเป็น  $2w_o$  เมื่อสัญญาณนี้ถูกสุ่มตัวอย่าง  $f_s(t)$  ผลลัพธ์ที่ได้จะเห็นได้ว่าสเปกตรัมของสัญญาณที่ได้จากการสุ่มตัวอย่าง  $X_{ss}(w)$  เป็นการนำเอาสเปกตรัมของ  $x(w)$  มาวางเรียงห่างเท่าๆกันไปตลอดบนแกนความถี่  $w$

ข้อสังเกต จากรูป (1-6c) ถ้าความถี่ของตัวส่งสัญญาณ  $f_{sn}(t)$  มีค่าต่ำกว่า  $w_o$  มีค่าเท่าหรือมี  $T$  มาก จะทำให้เกิดความผิดเพี้ยนไปของสเปกตรัมของสัญญาณเดิมได้ ผลนี้มีชื่อเรียกว่า ผลการเอเลตซิง (aliasing effect) นั่นหมายถึงว่า ความถี่ของตัวส่งสัญญาณสุ่มต้องมีค่าสูงพอจึงจะไม่ให้เกิดผลการเอเลตซิงได้ ซึ่งค่าความถี่ต่ำที่สุดที่ทำให้ไม่เกิดผลก็คือ

$$w_s = 2w_o \quad (1.26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และค่าความถี่ค่านี้มีชื่อเรียกว่า ความถี่ในควิสต์ หรือบางครั้งเรียกว่าอัตราในควิสต์ก็ได้ รูป (1-6) บอกเรานัยหนึ่งว่า คือผลตอบสนองความถี่ของวงจรกรองเชิงเลข จะมีผลตอบสนองความถี่ที่มีลักษณะเป็นคาบ คือเริ่มซ้ำค่าเดิม ที่มีค่าความถี่เป็น  $2\pi/T$  หรือ  $\omega_s/2$  ซึ่งมีค่าความถี่นี้เรียกว่า ค่าความถี่พับ ผลจากการที่ผลตอบสนองความถี่มีลักษณะเป็นคาบก็เนื่องมาจากกาสุ่มตัวอย่างนั่นเอง



รูป 1-6 แสดงสเปกตรัมของสัญญาณจากการสุ่มตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.1.3.2 ความสัมพันธ์ของโพลของระบบเชิงอุปมาน และ โพลของระบบเชิงเลข

จากที่ได้เคยกล่าวไว้แล้วว่า ระบบเชิงอุปมาน ทั่วไปสามารถเขียนอธิบาย ได้ด้วยสมการเชิงอนุพันธ์ ซึ่งคำตอบของสมการเชิงอนุพันธ์นี้อยู่ในรูปแบบของส่วนประกอบของพจน์ที่กำลังตั้งนั้น สัญญาณ  $x(t)$  ใดๆอาจเขียนอยู่ในรูปแบบ

$$x(t) = a_1 \exp(s_1 t) + a_2 \exp(s_2 t) + \dots + a_k \exp(s_k t) = \sum_{m=1}^k a_m \exp(s_m t) \quad (1.27)$$

โดยที่  $s_k$  เป็นค่าโพลในโดเมนความถี่ของสัญญาณ ซึ่งโพลนี้อาจเป็นค่าจริงหรือค่าสังยุค ก็ได้ เมื่อเราทำการสุ่ม  $x'(t)$  หรือ

$$x'_{ss}(t) = \sum_{-\infty}^{\infty} a_k \exp(s_k t) \delta(t - nT) \quad (1.28)$$

โดยอาศัยคุณสมบัติของการแปลงลาปลาซ ของลำดับอิมพัลส์ที่หนองออกไป คือ

$$Z\{\delta(t - nT)\} = \exp(-s_n T) \quad (1.29)$$

ดังนั้น ถ้าเราทำการแปลงลาปลาซของสมการ (1.28) และประยุกต์ใช้สมการ (1.29) จะได้

$$\begin{aligned} Z(x'_{ss}(t)) &= x'_{ss}(s) \\ &= a_k \{1 + \exp(s_k T) \cdot \exp(-sT) + \exp(2s_k T) \cdot \exp(-2sT) + \dots\} \end{aligned} \quad (1.30)$$

สเปกตรัมที่ได้จากการสุ่มตัวอย่าง  $x'_{ss}(s)$  เขียนในรูปปิด (closed form) ได้เป็น

$$\begin{aligned} x'_{ss}(s) &= a_k / \{1 - \exp(s_k T) \cdot \exp(-sT)\} \\ &= a_k / \{1 - \exp([s_k - s]T)\} \end{aligned} \quad (1.31)$$

ซึ่งโพลของ  $x'_{ss}(s)$  หาได้โดยการให้พจน์ส่วนเป็นศูนย์ และหาค่าจะได้

$$\begin{aligned} \text{ดังนั้น} \quad & \exp\{(s_k - s)T\} = 1 \\ & (s_k - s)T = 0 \pm jn2\pi \\ & s = s_k \pm j2n\pi / T \end{aligned} \quad (1.32)$$

โดยที่  $-\infty < n < +\infty$  ในขณะเดียวกัน การแปลงลาปลาซ ของระบบโพลเดี่ยว  $x'(t)$  เป็น

$$Z\{x'(t)\} = Z\{a_k \cdot \exp(s_k T)\} = a_k / (s - s_k) \quad (1.33)$$

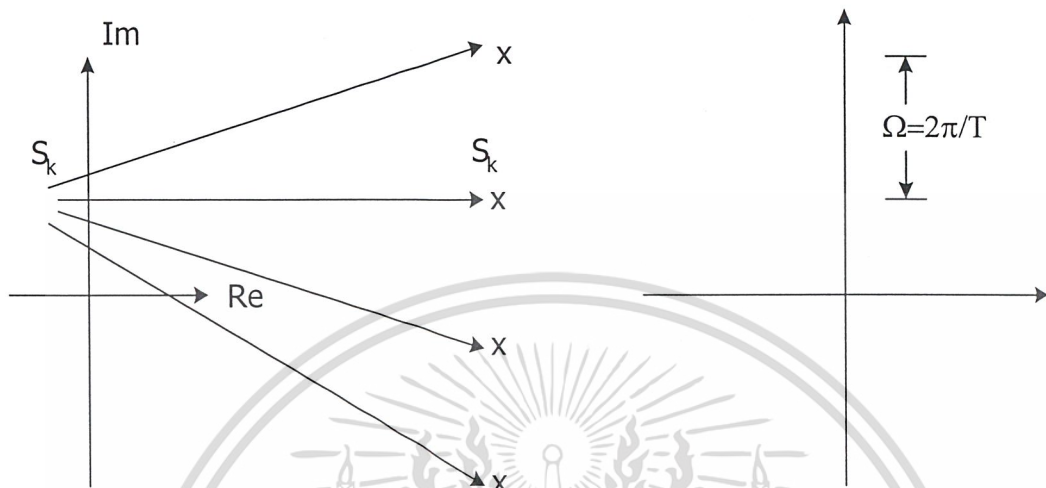
หรือ  $x'(t)$  มีโพลอยู่ที่

$$s = s_k \quad (1.34)$$

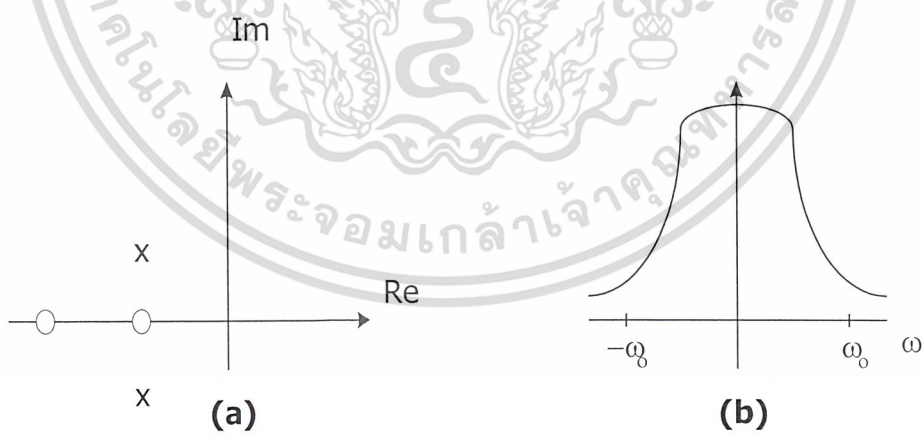
พิจารณาความสัมพันธ์ระหว่าง พจน์  $s_k$  จากสมการ (1.34) ซึ่งเป็นตำแหน่งโพลของระบบเชิงอุปมาน กับ  $s$  ในสมการ (1.32) ซึ่งเป็นตำแหน่งโพลของระบบเชิงเลขเห็นได้ชัดตามรูป (1-7) ว่าจากระบบเชิงอุปมานที่มีโพลเพียงตัวเดียว เมื่อทำการลุ่มสัญญาณเพื่อประมวลในระบบเชิงเต็มหน่วย หรือระบบเชิงเลข โพล  $s_k$  จะถูกส่ง (map) ไปเป็นโพลจำนวนอนันต์โพลที่แต่ละโพลวางห่างกันด้วยระบบ  $\Omega = 2\pi/T$  ดังนั้นสำหรับระบบเชิงอุปมานทั่วไปของ

$$x(t) = \sum_{k=1}^m a_k \cdot \exp(s_k t)$$

ที่มีชุดของโพล และ ซีโร ตามรูป (1-8a) และมีสเปกตรัมตามรูป (1-8b) เมื่อมีการลุ่มสัญญาณชุดของโพล และซีโรนี้ก็จะถูกส่งไปบนระนาบ  $Z$  เป็นจำนวนอนันต์ชุด (1-9a) และสเปกตรัมของระบบเชิงลุ่มตัวอย่างก็จะมีคุณสมบัติตามดังรูป (1-9b) ซึ่งผลก็คือผลอันเดียวกันกับรูป (1-6c)



(a) โพลของระบบเชิงอุปมาน (b) โพลของระบบเชิงเลข  
รูป 1-7 แสดงผลตำแหน่งโพลจากการสุ่มตัวอย่างระบบที่มีโพลที่  $s = s_k$

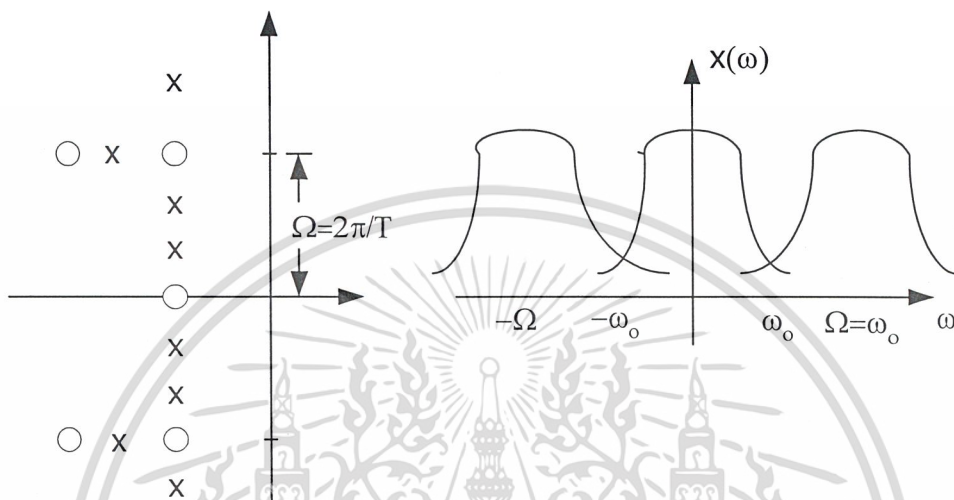


รูป 1-8 แสดงตัวอย่าง โพล ซีโร และ สเปกตรัมของระบบเชิงอุปมาน

1.1.3.3 ความสัมพันธ์ของระบบเอส และระบบแซด

จากสมการ (1.31) เห็นได้ว่า การทำการแปลงลาปลาซของสัญญาณที่ได้จากการสุ่มตัวอย่างสัญญาณเชิงซ้อน  $\exp(s_k t)$  เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_{ss}(s) = 1 / \{1 - \exp(-sT) \cdot \exp(s_k T)\}$$



รูป 1-9 แสดงโพล ซีโร และ สเปกตรัมของระบบเชิงเลข

ซึ่งผลจากการลุ่มตัวอย่างทำให้ได้โพลเป็นจำนวนอนันต์โพล บนระนาบเชิงซ้อนฮอส การวิเคราะห์ระบบโดยใช้ ระบายเชิงซ้อนฮอส ที่มีโพลจำนวนอนันต์โพลของระบบเชิงเลขทำได้ยุ่งยาก ดังนั้นเพื่อความสะดวกจึงได้ ทำการแทนให้

$$Z = \exp(sT) \text{ หรือ } Z^{-1} = \exp(-sT) \tag{1.35}$$

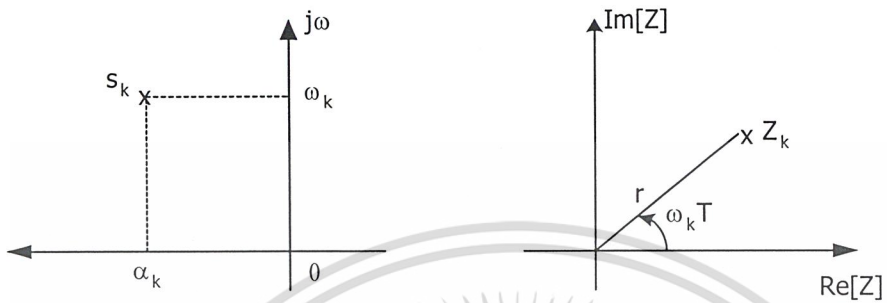
ซึ่งมีค่าขนาน และ เฟส ของตัวแปรแซด สำหรับโพลที่ตำแหน่ง  $s_k$  คือ

$$Z = \exp(s_k T) = \exp(\{\alpha + jw\}T) \tag{1.36}$$

$$= |Z| \exp(j\theta) \tag{1.37}$$

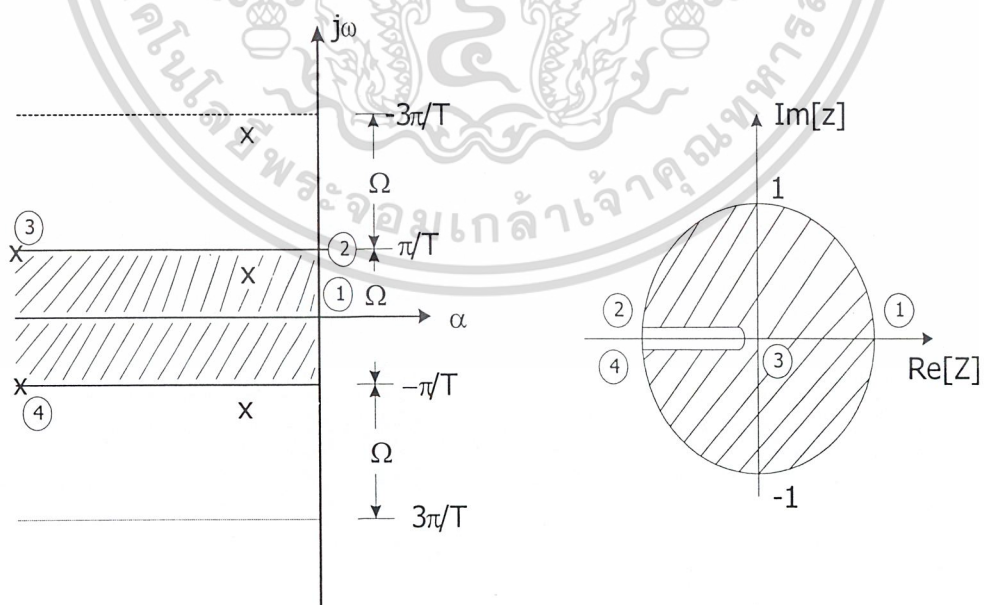
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $|Z| = \exp(\alpha T)$  และ  $\angle Z = \omega T$  และความสัมพันธ์แสดงได้ในรูป (1-10)



รูป 1-10 แสดงความสัมพันธ์ของ โพล  $s_k$  ในระบบ S และ Z

การแทนตัวแปรตามสมการ (1.35) ถ้าแสดงโดยทางกราฟ (1-11) ก็คือ การส่งค่าในแถบที่มีความกว้าง  $2\pi/T$  ในระนาบเชิงซ้อนไปยังระนาบเชิงซ้อน โดยที่ค่าทางด้านซ้ายของระนาบเอต



รูป 1-11 แสดงความสัมพันธ์ระหว่างตัวแปร ลาปลาซ กับตัวแปร Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นบริเวณของตำแหน่งโพลที่มีเสถียรภาพ จะถูกส่งเข้าไปภายในวงกลมรัศมีหนึ่งหน่วย เห็นได้ว่าการส่งจากระนาบเอสไปยังจุดในระนาบแซค จะไม่เป็น การส่งแบบหนึ่งต่อหนึ่ง (one-to-one mapping) นั่นคือแถบขนาดกว้าง  $2\pi/T$  จะซ้อนทับไปบนระนาบแซค หรือจำนวนโพลอนันต์ โพลถูกส่งเข้าไปแบบซ้อนทับกันในวงกลมหนึ่งหน่วย ระนาบแซค กลายเป็นโพลเพียงตัวเดียว ซึ่งทำให้ง่ายต่อการพิจารณาเสถียรภาพของตัวกรอง

## 1.2 การออกแบบตัวกรองป้อนกลับเชิงเลข

ตามที่ได้ทราบแล้วว่าการออกแบบตัวกรองนั้น เราต้องการให้ได้ตัวกรองตอบสนองต่อความถี่ (ถ้าเป็นไปได้) ให้มีทั้งผลตอบสนองแอมพลิจูด และผลตอบสนองเฟส ตามผลตอบสนองอุดมคติแอมพลิจูดอย่างเดียวกันพอ สำหรับการออกแบบตัวกรองให้มีผลตอบสนองแอมพลิจูดอย่างเดียวกันนั้นโดยทั่วไป ตัวกรองป้อนกลับเชิงเลขใช้อันดับของตัวกรองต่ำกว่าตัวกรองไม่ป้อนกลับเชิงเลขมาก ซึ่งข้อดีของการที่มีตัวกรองอันดับต่ำก็คือ เป็นการประหยัดในเทอมของเวลาในการคำนวณ และขนาดของหน่วยความจำที่ต้องใช้ ส่วนข้อเสียนี้จะทำให้การประมวลผลสัญญาณกับสัญญาณที่ต้องการคงรูปร่างของสัญญาณเดิมไว้ ผิดเพี้ยนไป เช่น ในกรณีประมวลผลสัญญาณ ECG เป็นต้น ข้อเสียนี้อาจแก้ไขโดยเพิ่มส่วนของวงจรปรับเฟสเท่า (phase equalizer) เข้าไป

ในหัวข้อนี้เป็นการกล่าวถึงการออกแบบ ตัวกรองป้อนกลับเชิงเลข ซึ่งมีแนวทางการออกแบบแตกต่างไปจากวิธีการมาข้างต้น และการออกแบบค่อนข้างยุ่งยากกว่า ทั้งนี้เนื่องจากตัวกรองแบบนี้มีการนำเอาสัญญาณออกมาใช้ในการคำนวณหาสัญญาณออกลำดับถัดไป หรือมีการป้อนกลับสัญญาณ ในการนี้การออกแบบจึงต้องคำนึงถึงเสถียรภาพของตัวกรองด้วย โดยทั่วไปวิธีการออกแบบตัวป้อนกลับเชิงเลข 2 วิธีใหญ่คือ วิธีการออกแบบโดยทำการแปลง จากฟังก์ชันถ่ายโอนของตัวกรองเชิงอุปมานที่มีเสถียรภาพดี วิธีการนี้ถ้าหากเรามีการแปลงที่ดี การออกแบบก็ไม่ต้องคำนึงถึงเสถียรภาพของตัวกรองอีก โดยที่ได้ผลตอบสนองแอมพลิจูดตามต้องการ วิธีนี้เหมาะสำหรับผู้ที่มีความรู้หรือ เชี่ยวชาญการออกแบบตัวกรองเชิงอุปมานดีแล้ว สำหรับการออกแบบตัวกรองเชิงอุปมาน จะขอนำมากล่าวโดยสังเขป เนื่องจากได้มีการค้นคว้าวิจัย และรวบรวมเป็นหลักการไว้เป็นอย่างดี ผู้ที่สนใจสามารถหาอ่านได้ทั่วไป ส่วนวิธีที่สองเป็นการออกแบบตัวกรองป้อนกลับเชิงเลขในโดเมนแซคโดยตรงซึ่งวิธีนี้ยังรวมถึง การออกแบบโดยใช้คอมพิวเตอร์ช่วย ดังจะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.2.1 คุณสมบัติของการแปลงตัวแปร

เพื่อที่จะเป็นประโยชน์แก่ผู้ทำการศึกษาวិชาการออกแบบวงจรกรองโดยผ่าน การแปลงตัวแปรซึ่งอาจถือได้ว่าเป็น การแปลงความถี่ (frequency transformation) แบบหนึ่ง โดยเฉพาะผลการแปลงที่สามารถทำได้โดยยังคงคุณสมบัติ ทั้งผลตอบสนองเฟส และ ผลตอบสนองแอมพลิจูด ของตัวกรองเชิงอุปมานที่เป็นตัวกรองต้นแบบไว้ได้ ซึ่งในปัจจุบันเราสามารถทำการแปลงโดยรับขาคผลตอบสนองแอมพลิจูดเพียงอย่างเดียว ดังนั้นในหัวข้อนี้จะกล่าวถึงหลักการของผลการแปลงว่าเป็นอย่างไร

#### 1.2.1.1 ตัวกรองเชิงเลข และตัวกรองเชิงอุปมาน

โดยทั่วไป ฟังก์ชันถ่ายโอนของตัวกรองป้อนกลับเชิงเลข สามารถเขียนได้คือ

$$H(z) = N(z^{-1}) / D(z^{-1}) = \sum_{i=0}^M a_i z^{-i} / \sum_{i=0}^N b_i z^{-i} \quad (1.38)$$

และถ้าหากเป็นตัวกรองแบบเหตุกภาพ เราเขียนให้  $b_0=1$  หรือ

$$H(z) = \sum_{i=0}^M a_i z^{-i} / \left\{ 1 + \sum_{i=1}^N b_i z^{-i} \right\} \quad (1.39)$$

จากสองสมการนี้ ถ้าหากในเบื้องต้นเราทำการแปลงตัว  $z^{-1}$  ด้วยตัวแปร  $s$  โดยตรง แล้วจะได้สมการที่คล้ายกัน ซึ่งเป็นฟังก์ชันถ่ายโอนของตัวกรองเชิงอุปมาน

$$H(s) = \sum_{i=0}^M a_i s^i / \sum_{i=1}^N b_i s^i \quad (1.40)$$

โดยทั่วไปทั้งสมการ (1.39) และ (1.40) นั้น ค่า  $M \leq N$  เพื่อที่จะทำให้จำนวนของโพลมากกว่าจำนวนซีโร เพื่อป้องกันไม่ให้เกิดค่าขยายเป็นอนันต์ที่ค่าความถี่อนันต์

ในทางทฤษฎีแล้ว การแปลงจากสมการ (1.39) ไปเป็นสมการ (1.40) หรือแปลงในทางกลับกัน มักไม่นิยมทำโดยทำการแทนตัวแปรโดยตรง เพราะการทำเช่นนี้อาจทำให้ คุณสมบัติของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวกรอง จากโดเมนหนึ่งเมื่อทำ การส่งไปยังอีกโดเมนหนึ่ง มีคุณสมบัติผิดเพี้ยนไป นอกจากนี้ เสถียรภาพของตัวกรองอาจเสียไปด้วย ดังนั้น การส่งจากโดเมนหนึ่งไปยังอีกโดเมนหนึ่ง จึงควรมี เงื่อนไขกำหนดไว้ ซึ่งเราจำทำการศึกษาต่อไปนี้

### 1.2.1.2 การส่งจากโดเมนเอส ไป โดเมนแซด

เพื่อที่จะให้การส่ง จากโดเมนเชิงอุปมานเชิงเลข โดยที่เสถียรภาพของตัวกรอง และคุณสมบัติเชิงความถี่ ซึ่งอาจเป็นผลตอบสนองแอมพลิจูด หรือ ผลตอบสนองเฟส หรือทั้งสองผลตอบสนอง ไม่เปลี่ยนแปลง หรือเปลี่ยนไปน้อย การส่งควรสอดคล้องตามเงื่อนไข 2 ประการ คือ

๑. เงื่อนไขที่ 1 เงื่อนไขนี้ทำให้การส่งจาก โดเมนเอส ไปยัง โดเมนแซด ทำได้โดยที่ยังรักษาคุณสมบัติเชิงความถี่ของตัวกรองเชิงอุปมาไว้ เงื่อนไขนี้มีว่า “ค่าบนแกนจินตภาพ ในระนาบเชิงซ้อนเอส (หรือ  $S = jw$  สำหรับ  $-\infty < w < \infty$ ) จะถูกส่งไปยังค่าบนเส้นรอบวงกลมหนึ่งหน่วย ในระนาบเชิงซ้อนแซด (หรือ  $Z = \exp \{jwT\} = \exp \{je\}$  สำหรับ  $-\pi < \theta < \pi$ ) “ หรือเขียนได้เป็น

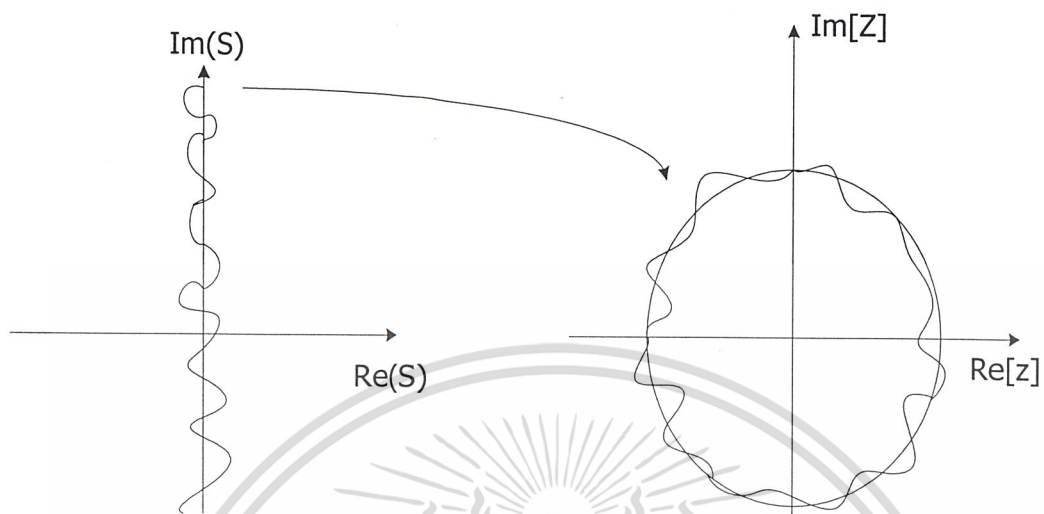
$$\{s = jw | -\infty < w < \infty\} \rightarrow \{z = \exp(j\theta) | -\pi < \theta < \pi\} \quad (1.41)$$

คุณสมบัติเงื่อนไขของการส่งนี้ได้แสดงไว้ในรูป (1-12)

๒. เงื่อนไขที่ 2 เป็นเงื่อนไขที่รักษาเสถียรภาพของวงจรกรองไว้ ซึ่งทำให้เราสามารถออกแบบตัวกรองเชิงเลขที่เสถียรได้ เงื่อนไขนี้มีอยู่ว่า “ค่าบนด้านซ้ายทั้งหมดของ ระนาบเชิงซ้อนเอส (หรือ  $R_0[S] < 0$ ) จะถูกส่งเข้าไปยังบริเวณในวงกลมหนึ่งหน่วย ของระนาบเชิงซ้อนแซด (หรือ  $|Z| < 1$ ) “ หรือเขียนได้ว่า

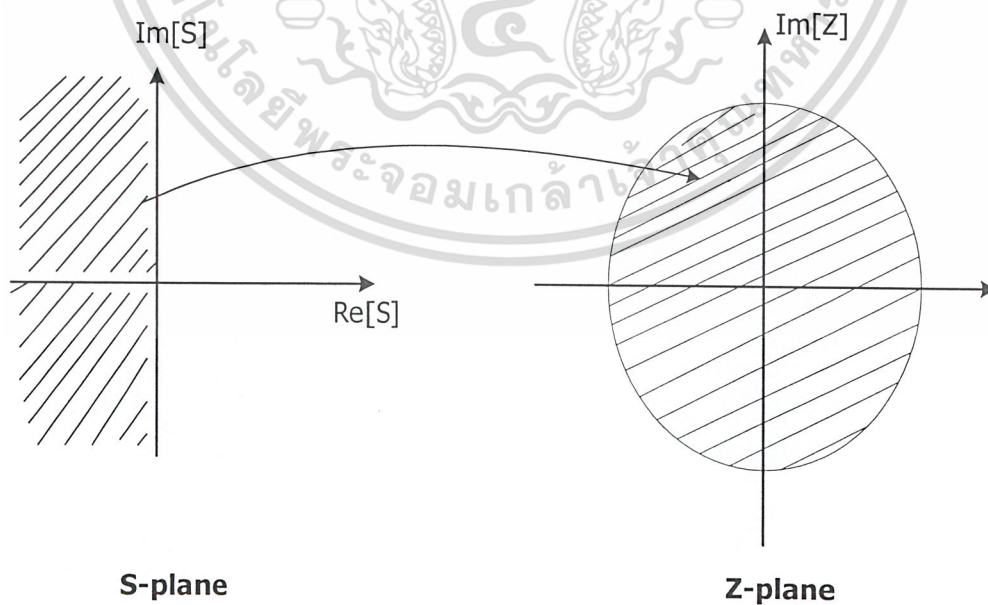
$$\{s | R_0[s] < 0\} \rightarrow \{z | |z| < 1\} \quad (1.42)$$

ดังสามารถแสดงความสัมพันธ์นี้ได้ตามรูป (1-12)



รูป 1-12 แสดงการส่งค่าจินตภาพจากระนาบ S ไปยังเส้นรอบวงกลมในระนาบ Z

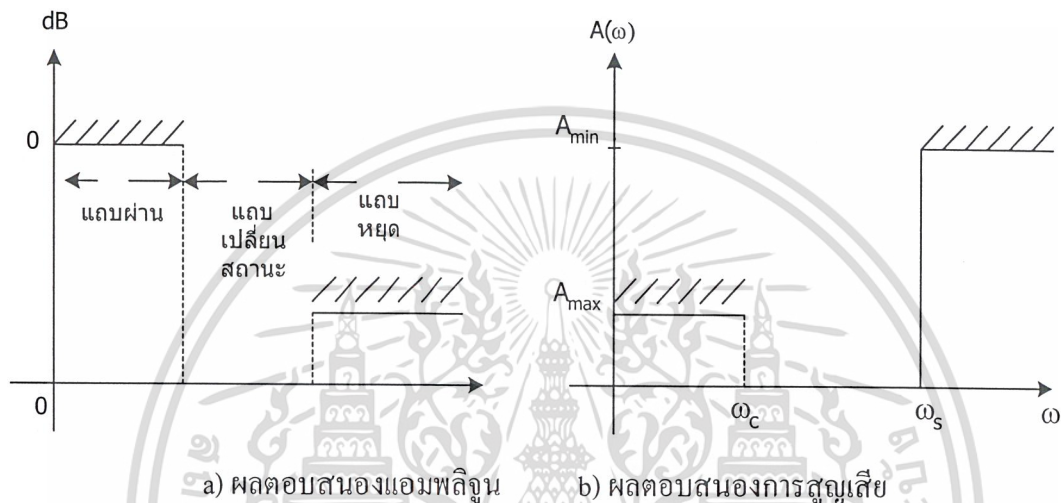
มีการเปลี่ยนแปลงเพียงชนิดเดียว คือ การแปลงเชิงเส้นคู่ที่เป็นไปตามเงื่อนไขทั้ง 2 ประการ



รูป 1-13 แสดงการส่งค่าทางด้านซ้ายของระนาบ S เข้าไปในวงกลมของระนาบ Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อนี้ขอก้าวถึงวิธีการออกแบบตัวกรองเชิงอุปมาน ซึ่งเป็นการประมาณเพื่อการหาฟังก์ชันถ่ายโอน ของตัวกรอง 2 แบบด้วยกันคือ ตัวกรองบัตเตอร์เวิร์ท (Butterworth filter) และตัวกรองเชบิเชฟ (Chebyshev filter) ตัวกรองทั้งสองแบบนี้เป็นตัวกรองที่นิยมใช้กันมาก เนื่องจากการออกแบบทำได้ง่าย และมีวิธีการออกแบบที่เป็นมาตรฐาน

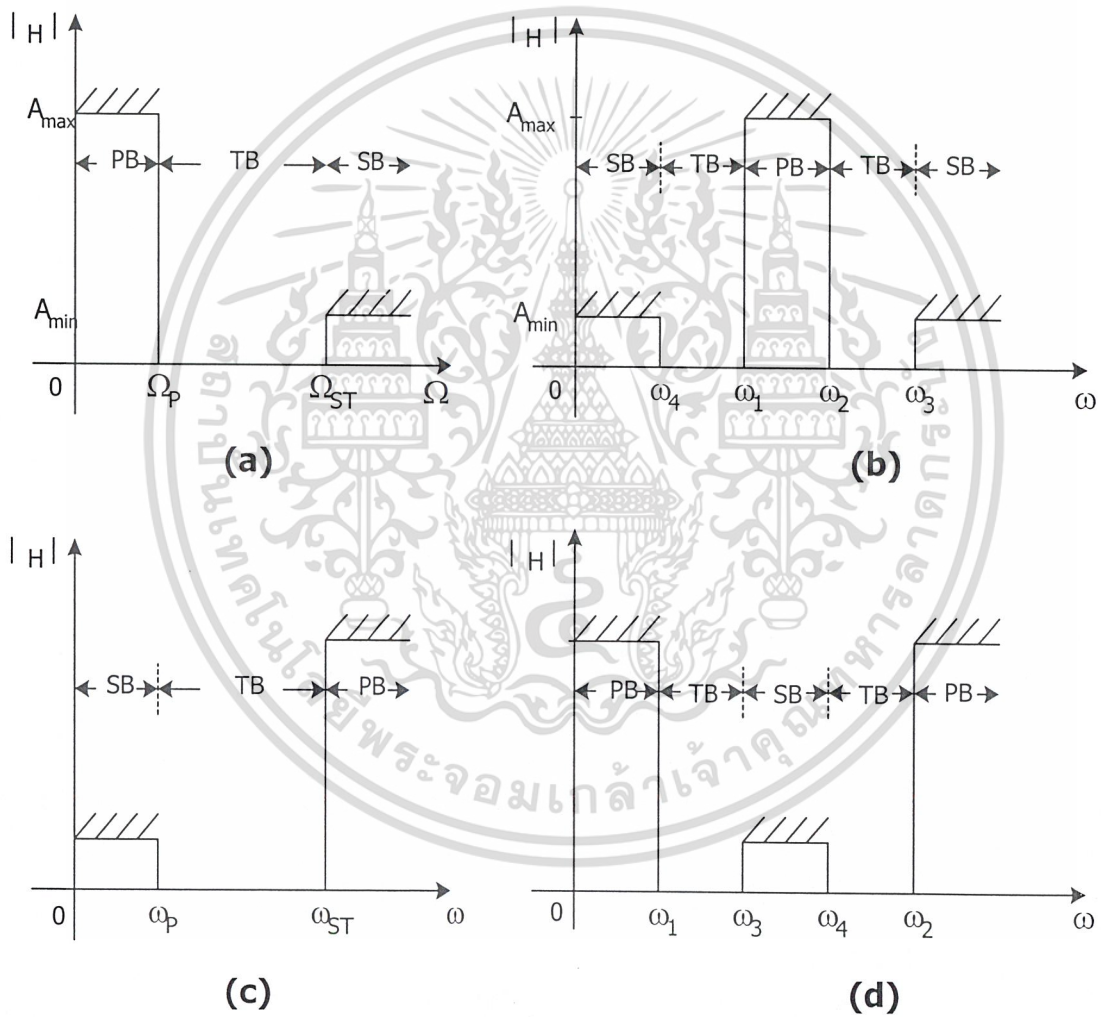


รูป 1-14 แสดงข้อกำหนดในการออกแบบวงจรกรองผ่านความถี่ต่ำ

ในการออกแบบตัวกรอง ให้มีคุณสมบัติตามต้องการนั้น โดยทั่วไปในเบื้องต้นจะทำการออกแบบตัวกรองผ่านความถี่ต่ำนอร์มอลไลซ์ หรือตัวกรองผ่านความถี่ต่ำบรรทัดฐานก่อน จากนั้นจึงค่อยทำการแปลงคุณสมบัติให้เป็นตัวกรองที่ต้องการ ซึ่งคุณสมบัติของตัวกรองผ่านความถี่ต่ำบรรทัดฐานนี้อาจกำหนดให้อยู่ในรูปแบบของ ผลตอบสนองแอมพลิจูด ตามรูป (1-14a) หรือกำหนดเป็นรูปแบบผลตอบสนองการสูญเสีย (loss characteristic) ตามรูป (1-14b) โดยที่มีข้อนำสังเกตว่า ผลตอบสนองทั้งสองแบบนี้ได้มีการทำเป็นบรรทัดฐาน ให้มี ค่าการขยายสูงสุด หรือ ค่าลดทอนสูงสุด เป็น 1 เท่า หรือ  $0$  dB เสมอ ส่วนค่า ความถี่ตัด หรือ  $\omega_c$  หรือ ที่เราเรียกว่าช่วงความถี่นี้ว่า แถบผ่าน ซึ่งเป็นช่วงความถี่ที่สัญญาณสามารถผ่านไปได้ และต้องการให้มีการสูญเสียเป็นค่า  $A_{max}$  ประการที่สองในช่วงความถี่ระหว่าง  $\omega = \omega_c$  ถึง  $\omega = \omega_{ST}$  หรือเรียกช่วงความถี่นี้ว่า แถบเปลี่ยนสถานะ ให้มีค่าการสูญเสียอยู่ในระหว่าง  $A_{max}$  ถึง  $A_{min}$  ส่วนในช่วงความถี่  $\omega = \omega_{ST}$  ถึงความถี่  $\omega = \infty$  หรือเรียกว่า แถบหยุด จะให้มีค่าการสูญเสียเป็น  $A_{min}$  ต่อไปนี้เราจะนำวิธีประมาณค่าของ ตัวกรองแบบบัตเตอร์เวิร์ท และตัวกรองแบบเชบิเชฟ มาทำการออกแบบเพื่อให้มีคุณสมบัติ ใกล้เคียงกับผลตอบสนองตามรูป (1-14) ที่สุด

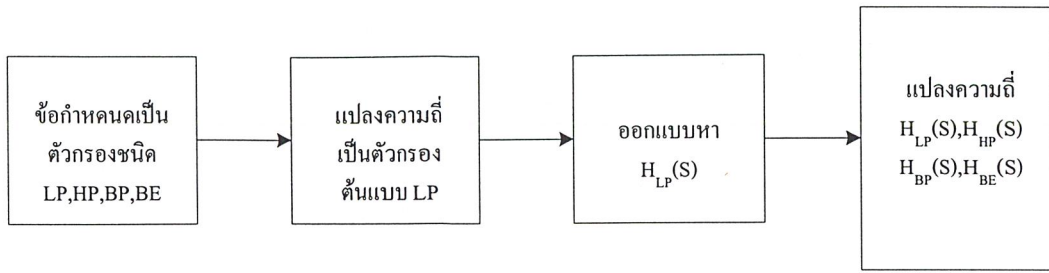
### 1.2.3 การแปลงความถี่

ในทางปฏิบัติเราอาจต้องการตัวกรองแบบอื่น อาทิเช่น ตัวกรองผ่านความถี่สูง (high pass filter หรือ อาจเขียนย่อเป็น HP) หรือตัวกรองผ่านแถบความถี่ (band pass filter หรือ BP) หรือตัวกรองกำจัดแถบความถี่ (band elimination filter หรือ BE) ก็ได้ ซึ่งตัวกรองเหล่านี้ผลตอบสนองอุดมคติของมันได้แสดงไว้ในรูป (1-16) โดยที่ตามรูปได้ใช้ สัญลักษณ์ SB แทน แถบหยุด PB แทน แถบผ่าน TB แทน แถบเปลี่ยนสถานะ การออกแบบตัวกรองเหล่านี้



รูป 1-15 แสดงผลตอบสนองแอมพลิจูดอุดมคติของ  
 a) ตัวกรองผ่านความถี่ต่ำแบบ b) ตัวกรองผ่านแถบความถี่  
 c) ตัวกรองผ่านความถี่สูง d) ตัวกรองกำจัดแถบความถี่

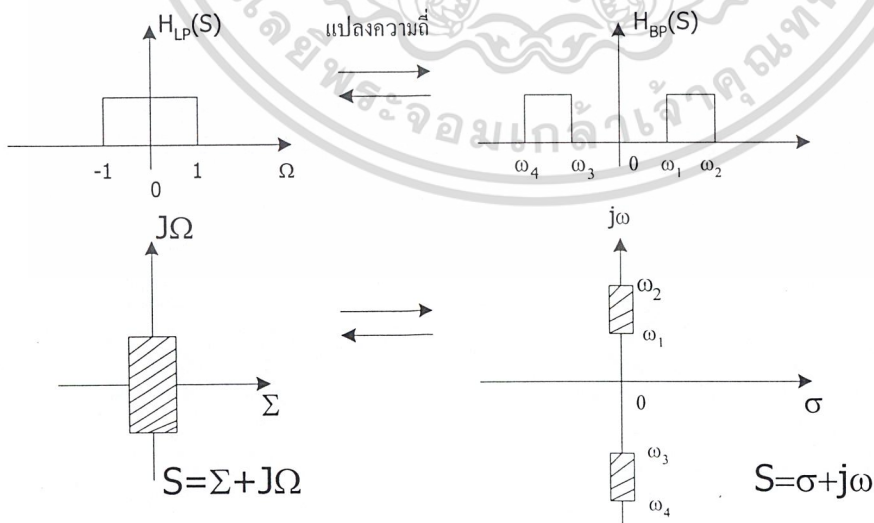
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1-16 แสดงลำดับขั้นตอนการออกแบบตัวกรองโดยใช้การแปลงความถี่

ไม่จำเป็นต้องทำการออกแบบโดยตรงอาจทำได้โดยทางอ้อม ด้วยวิธีการตามแผนในรูปที่ (1-16) ซึ่งประกอบไปด้วย 4 ขั้นตอน คือ

- 1) กำหนดคุณสมบัติของตัวกรอง เช่น เป็นตัวกรองชนิดใด มีค่า  $A_{max}$ ,  $A_{min}$  และค่าความถี่ต่างๆ
- 2) ทำการแปลงความถี่จากคุณสมบัติของตัวกรองที่ต้องการ ไปเป็นคุณสมบัติของตัวกรองผ่านความถี่ต่ำแบบ ตามรูป (1-15a)
- 3) ใช้วิธีการตามหัวข้อ (1.42) ออกแบบตัวกรองผ่านความถี่ต่ำต้นแบบ ที่มีค่าความถี่  $\Omega_c=1$  และได้  $H(S)$
- 4) จากตัวกรองต้นแบบทำการ แปลงความถี่  $\Omega$  ให้เป็นตัวกรองที่มีคุณสมบัติตามที่กำหนดในข้อ (1) ซึ่งทำให้ได้  $H_{LP}(S)$ ,  $H_{BP}(S)$  หรือ  $H_{BE}(S)$



รูป 1-17 แสดงแผนภาพการแปลงความถี่จากตัวกรองผ่านความถี่ต่ำเป็นตัวกรองผ่านแถบความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 1-17 แสดงแผนภาพการแปลงความถี่จากตัวกรองผ่านความถี่ต่ำเป็นตัวกรองผ่านแถบความถี่ หรือในทางกลับกัน โดยที่ตามรูปนั้น การวาดคุณสมบัติการตอบสนองความถี่ของตัวกรองได้ขยายให้ครอบคลุม คุณสมบัติตอบสนองความถี่ทางด้านแกนลบด้วย ในรูป (1-17b) เห็นได้ว่าการแปลงความถี่ ก็คือ การแปลงตัวแปรเชิงซ้อน  $s = \Sigma + j\Omega$  ไปเป็นตัวแปรเชิงซ้อน  $S = \sigma + jw$

การแปลงแบบ การแปลงผ่านความถี่ต่ำเป็นผ่านความถี่ต่ำ ( low - pass to low - pass transformation) ของตัวกรองบัตเตอร์เวิร์ธ และเชบีเชฟ ตามลำดับ กรณีนี้อาจเขียนเป็นสูตรของการแปลงได้ว่า

$$H_{LP}(s) = H_{LP}(s) \Big|_{s=s/w_c} \quad (1.43)$$

โดยที่ ความถี่  $\Omega_c = 1$  และ  $\Omega$  ถูกแปลงเป็น  $w/w_c$

สำหรับกรณีอื่นขอเขียนไว้โดยไม่มีกรณีสอง คือ

(ก) การแปลงผ่านความถี่ต่ำเป็นผ่านความถี่สูง (low - pass to high - pass transformation) ใช้สูตร

$$H_{LP}(s) = H_{LP}(s) \Big|_{s=w_c/s} \quad (1.44)$$

(ข) การแปลงผ่านความถี่ต่ำเป็นผ่านแถบความถี่ (low - pass to band - pass transformation) ใช้สูตร

$$H_{LP}(s) = H_{LP}(s) \Big|_{s=(s^2+w_0^2)/Bs} \quad (1.45)$$

โดยที่ ตัวแปรความถี่ของตัวกรองต้นแบบ  $\Omega = 1$  และ  $\Omega_{ST} = (w_4 - w_3)/(w_2 - w_1)$  และ  $w_0 = (w_1 w_2)^{1/2}$  เป็นค่า ตัวกลางเรขาคณิต (geometric mean) ของแถบความถี่ผ่าน และ  $B = w_2 - w_1$  ซึ่งเป็นค่า ความกว้างของแถบผ่านความถี่

(ค) การแปลงผ่านความถี่ต่ำเป็นกำจัดแถบความถี่ (low - pass to band - elimination transformation) ใช้สูตร

$$H_{BE}(s) = H_{LP}(s) \Big|_{s=Bs/(s^2 - w_0^2)} \quad (1.46)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.2.4 การแปลงเชิงเส้นคู่

การแปลงจากตัวแปร  $s$  ไปเป็นตัวแปร  $z$  ในหัวข้อที่แล้ว ยังมีคุณสมบัติไม่ครอบคลุมตามเงื่อนไขในหัวข้อ (1.2.1) นอกจากนี้ผลการแปลงอิมพัลส์ขึ้นยงก็ยังมีปัญหาเรื่องปรากฏการณ์เอเลสซิง อันเนื่องมาจากการส่งไม่เป็นแบบหนึ่งต่อหนึ่ง จึงได้มีการค้นหาสมการในการแปลใหม่ วิธีหนึ่งก็คือการใช้ การประมาณการค่าอินทิเกรตด้วย กฎการอินทิเกรตแบบสี่เหลี่ยมคางหมู และทำการประมาณสมการเชิงอนุพันธ์ด้วยการใช้ ผลต่างสี่เหลี่ยมคางหมู ผลของการแปลงที่ได้จากการประมาณค่านี้เรียกว่า การแปลงเชิงเส้นคู่ (bilinear transform) ซึ่งสามารถแสดงให้เห็นจริงได้ จากการพิจารณาจริงเชิงอนุมาณดังต่อไปนี้

สมมุติให้วงจรกรองเชิงอนุมาณแบบ โพลเดี่ยวที่มีฟังก์ชันถ่ายโอนเป็น

$$H(s) = a/(s + a) \quad (1.47)$$

ซึ่งสมการเชิงอนุพันธ์ที่สัมพันธ์กับสมการนี้เป็น

$$\frac{dy}{dx} = -ay + ax \quad (1.48)$$

ถ้าหากเราทำการอินทิเกรตจากเวลา  $t_1 = 0$  ถึงเวลา  $t_1 = kT$  ทั้งสองข้างของสมการเขียนได้เป็น

$$\int_0^{kT} \frac{dy(t)}{dt} dt = -a \int_0^{kT} y(t) dt + a \int_0^{kT} x(t) dt$$

หรือ

$$\int y(kT) - y((k-1)T) = -a \int_{(k-1)T}^{kT} y(t) dt + a \int_{(k-1)T}^{kT} x(t) dt$$

จากนั้นทำการประมาณค่าการอินทิเกรตด้วย กฎการอินทิเกรตแบบสี่เหลี่ยมคางหมู ได้

$$y(kT) - y((k-1)T) = -(aT/2)\{y(kT) + y((k-1)T)\} + (aT/2)\{x(kT) + x((k-1)T)\} \quad (1.49)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแปลงเซต แล้วจัดพจน์เพื่อเขียนฟังก์ชันถ่ายโอนได้

$$H(z) = Y(z)/X(z) = a/\{(2/T)(\{1 - z^{-1}\}/\{1 + z^{-1}\}) + a\} \quad (1.50)$$

ซึ่งเมื่อเทียบการเปรียบเทียบสัมประสิทธิ์ระหว่าง สมการ (1.47) และ (1.48) จะได้

$$S = (2/T)(\{1 - z^{-1}\}/\{1 + z^{-1}\}) \quad (1.51a)$$

และ

$$z = (2 + sT)/(s - sT) \quad (1.51b)$$

ซึ่งการแปลงตามสมการ (1.51) นี้มีชื่อเรียกว่า การแปลงเชิงเส้นคู่

คุณสมบัติการแปลงเชิงเส้นคู่

เรามาดูว่า การแปลงเชิงเส้นคู่ จะมีคุณสมบัติสอดคล้องกับเงื่อนไขของการแปลงความถี่ตามหัวข้อ (1.2.1) หรือไม่ โดยขอเริ่มจากสมการ (1.51b) ซึ่งเราอาจเขียนเป็นตัวแปรแซด ได้เป็น

$$Z = (2/T + s)/(s/T - s)$$

และเมื่อแทน  $S = \sigma + jw$  แล้วเขียนเป็นตัวแปร  $Z$  ให้อยู่ในรูปแบบพิกัดเชิงขั้ว

$$Z = r \exp(j\theta)$$

โดยที่  $r$  แทนค่าขนาด และ  $\theta$  แทนค่ามุม โดยที่

$$r = \left[ \frac{(2/T + \sigma)^2}{(2/T - \sigma)^2 + w^2} \right]^{1/2} \quad (1.52)$$

และ

$$\theta = \tan^{-1}\{w/(2/T + \sigma)\} + \tan^{-1}\{w/2(2/T - \sigma)\} \quad (1.53)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลนี้สามารถแยกพิจารณาได้เป็น 3 กรณี คือ

1) กรณี  $\sigma < \phi$  ทำให้  $r > 1$  นั่นคือ แสดงว่าค่าบนด้านขวาของ ระนาบเอส จะถูกส่งไปบนบริเวณนอกวงกลมหนึ่งหน่วย หรือ นอก  $|Z| = 1$  ของระนาบเซต

2) กรณี  $\sigma = \phi$  ทำให้  $r > 1$  นั่นคือ  $\theta = 2 \tan^{-1}(wT/2)$  แสดงว่า ค่าบนแกน  $jw$  ของระนาบเอส ถูกส่งไปบนเส้นรอบวง  $|Z| = 1$  ของระนาบเซต และ กรณีนี้อาจพิจารณาอย่างลงได้เป็น 3 ส่วน คือ ส่วนแรกที่มีความถี่  $w = \phi$  นั่นคือ จุดกำเนิด  $(\theta, \phi)$  ของระนาบเอส ถูกส่งไปยังจุด  $(1, \phi)$  ของระนาบเซต ส่วนที่สองและสามเป็นจุดใกล้กับบริเวณที่ความถี่  $w$  เข้าใกล้  $+\infty$  และ  $w$  เข้าใกล้  $-\infty$  ของแกน  $jw$  ของระนาบเอสถูกส่งไปยังครึ่งบนและล่าง ตามลำดับของวงกลม  $|Z| = 1$

3) กรณี  $\sigma < \phi$  ทำให้  $r = 1$  นั่นคือค่าทางด้านซ้ายของระนาบเอส ถูกส่งไปภายในวงกลมหนึ่งหน่วยของระนาบเซต

คุณสมบัติทั้ง 3 กรณี ได้แสดงไว้ในรูป (1-18) ซึ่งตามรูปได้แสดงถึงการส่งจุดต่างๆจากระนาบเอสไปยังระนาบเซตด้วย ที่นี้เราเขียนให้  $w_a$  แทน ค่าความถี่เชิงมุม ของผลตอบสนองความถี่ของตัวกรองเชิงเลขที่ได้จากการแปลงเชิงเส้นคู่ เราสามารถเขียนความสัมพันธ์ดังต่อไปนี้

$$S = jw_a \quad (1.54)$$

$$Z = \exp(jw_d T) \quad (1.55)$$

และจากผลของการแปลงเชิงเส้นคู่ ความสัมพันธ์ของตัวแปร  $S$  และ  $Z$  จะเป็น

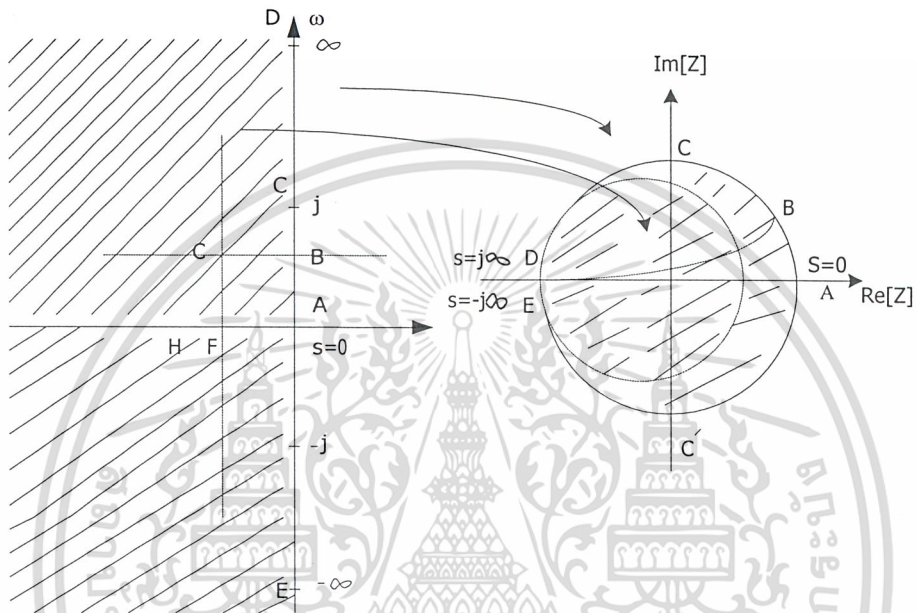
$$S = (2/T) \{(Z-1)/(Z+1)\} \quad (1.56)$$

เมื่อแทนค่าสมการ (1.54) และ (1.55) ลงในสมการ (1.56)

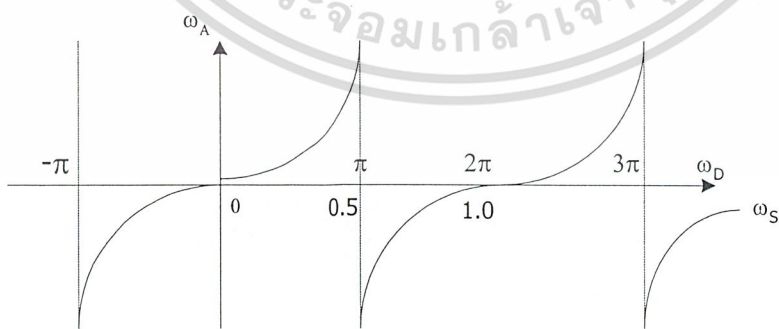
$$\begin{aligned} jw_a &= (2/T) \{ \{ \exp(-jw_d T) - 1 \} / \{ \exp(jw_d T) + 1 \} \} \\ &= (2/T) \left\{ \frac{\exp(jw_d T / 2) - \exp(-jw_d T / 2)}{\exp(jw_d T / 2) + \exp(-jw_d T / 2)} \right\} \\ &= (2j/T) \tan(w_d T / 2) \\ w_a &= (2/T) \tan(w_d T / 2) \end{aligned} \quad (1.57)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ ตามสมการ (1.57) ซึ่งเป็นฟังก์ชันแบบแทนเจนต์นั้นแสดงไว้ในรูป (1-19) เห็นได้ว่า ผลตอบสนองความถี่ของตัวกรองเชิงเลขมีลักษณะเป็นคาบ และเริ่มซ้ำค่าเมื่อความถี่  $\omega_D$  มากกว่า  $\pi$  เรเดียน และจากคุณสมบัติความเป็นคาบนี้ ทำให้ตัวกรองเชิงเลขสามารถใช้งานได้ในช่วงความถี่  $\phi < \omega_D < \pi$  เท่านั้น



รูป 1-18 แผนแสดงการส่งโดยใช้การแปลงเชิงเส้นคู่

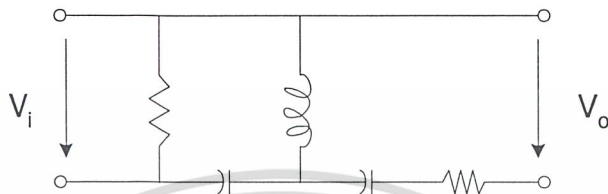


รูป 1-19 แสดงคุณสมบัติเป็นคาบของค่าความถี่เชิงเลขจากการแปลงเชิงเส้นคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2.5 ตัวกรองเชบีเชฟ

ความรู้พื้นฐาน



รูป 1-20 รูปร่างทั่วไปของตัวกรองเชิงอุปมาน

จากรูป 1-20 จะได้ฟังก์ชันถ่ายโอนคือ

$$\frac{V_o(s)}{V_i(s)} = H(s) = \frac{N(s)}{D(s)} \quad (1.58)$$

โดยที่  $V_i(s)$  และ  $V_o(s)$  เป็นการแปลงลาปลาซ ของ  $V_i(t)$  และ  $V_o(t)$  และ  $H(s)$  คือฟังก์ชันถ่ายโอนและ  $N(s)$  และ  $D(s)$  เป็นพหุนามในรูป  $(s = \sigma + j\omega)$  ค่าการลดทอนของตัวกรองจะถูกอธิบายดังนี้คือ

$$A(\omega) = 20 \log \frac{|V_i(j\omega)|}{|V_o(j\omega)|} = 20 \log \frac{1}{|H(j\omega)|} = 10 \log L(\omega^2)$$

โดยที่

$$L(\omega^2) = \frac{1}{H(j\omega)H(-j\omega)} \quad (1.59)$$

$A(\omega)$  เราเรียกว่า Loss characteristic ส่วน  $(\omega)$  หรือ การเลื่อนทางเฟส (phase shift) และ group delay  $(\omega)$  หาได้จาก

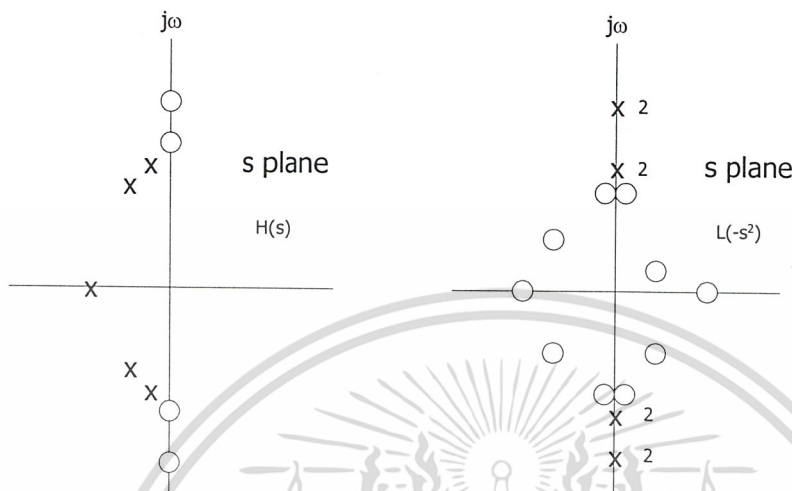
$$(\omega) = \arg H(j\omega) \text{ และ } \tau(\omega) = \frac{d(\omega)}{d\omega}$$

ถ้าเราแทนค่าของ  $\omega = s/j$  ในสมการ (1.59) จะได้

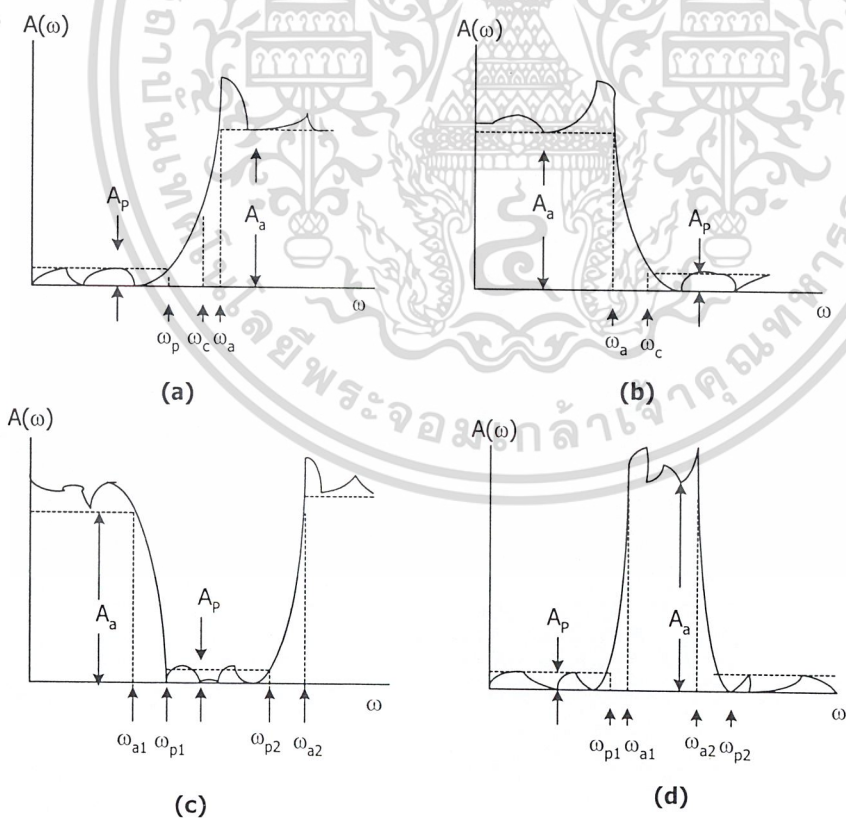
$$L(-s^2) = \frac{D(s)D(-s)}{N(s)N(-s)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราเรียก  $L(-s^2)$  ว่าเป็นฟังก์ชันการสูญเสีย ของตัวกรองซึ่งซีโรของมันก็คือ โพลของ  $H(s)$  และ  $H(-s)$  ส่วนโพลของ  $H(s)$  ส่วนโพลของมันก็คือ ซีโรของ  $H(s)$  และ  $H(-s)$  ตามรูปที่ 1-21



รูป 1-21 ซีโร โพล Plot ของ  $H(s)$  และ  $L(-s^2)$



รูป 1-22 แสดง Nonideal loss characteristic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ (a) Lowpass (b) Highpass  
(c) Bandpass (d) bandstop

จากรูป 1-22

$\omega_p$  คือ passband edge

$\omega_s$  คือ stopband edge

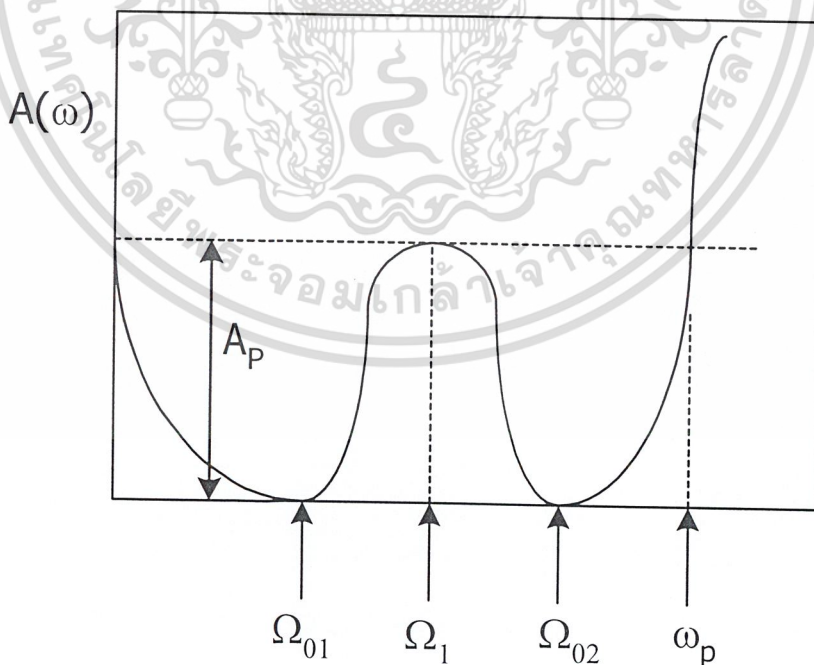
$A_p$  คือ maximum passband edge

$A_s$  คือ minimum stopband edge

$\omega_c$  คือ ขอบเขตระหว่าง passband และ stopband ซึ่ง  $|A(\omega)|$  เท่ากับ (3dB)

ตัวกรองเชบีเชฟ

เป็นตัวอย่างที่มี Pass band loss อยู่ระหว่าง 0 และค่า  $A_p$  จากรูป 1-23 เป็นรูปแสดงค่า loss characteristic ของ fourth-order normalized chebychev filter



รูป 1-23 แสดงค่า Loss characteristic ของ Fourth-order normalized chebyshev filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $W_p = 1$  ค่า Loss จะหาได้จาก

$$\begin{aligned} A(w) &= 10 \log L(w^2) \\ L(w^2) &= 1 + \varepsilon^2 F^2(w) \\ \varepsilon^2 &= 10^{0.1A_p} - 1 \end{aligned}$$

$F(w)$  หรือ  $L(-s^2)$  เป็นพหุนาม ซึ่งรูปของ normalized transfer function จะอยู่ในรูป

$$H_N(s) = \frac{H_0}{D(s)}$$

โดยที่  $H_0$  เป็นค่าคงที่

วิธีการหา  $H_N(s)$  จะมี 3 ขั้นตอนคือ

1. หาค่าของ  $F(w)$  ที่จะทำให้ได้ Loss characteristic ตามที่ต้องการ
2. ค่าของ  $L(-s^2)$  ซึ่งจะเป็ โพลของ  $H_N(s)$
3. หาค่าซีโร  $L(-s^2)$  ซึ่งจะเป็ โพลของ  $H_N(s)$

จากรูปเราจะได้ คุณสมบัติของ  $F(w)$  และ  $L(W^2)$  ดังนี้

1.  $F(w)=0$  ถ้า  $w = \pm\Omega_{01}, \pm\Omega_{02}$
2.  $F^2(w) = 1$  ถ้า  $w = 0, \pm\Omega_{11}, \pm 1$
3.  $dL(w^2)/dw$  ถ้า  $w=0, \pm\Omega_{01}, \pm\Omega_{11}, \pm\Omega_{02}$

จากคุณสมบัติ ข้อ (1) จะได้  $F(w)$  เป็นพหุนามอยู่ในรูป

$$F(w) = M_1(w^2 - \Omega_{01}^2)(w^2 - \Omega_{02}^2)$$

ในที่นี้  $M_1, M_2, \dots, M_7$  จะเป็นค่าคงที่ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคุณสมบัติข้อ 2 จะได้  $1 - F^2(w)$  จะมี ซีโร ที่  $w = 0, \pm\Omega, \pm 1$  ดังนั้นอนุพันธ์ของ  $1 - F^2(w)$  เทียบกับ  $w$  คือ

$$\frac{d}{dw}[1 - F^2(w)] = -2F(w) \frac{dF(w)}{dw} = -\frac{1}{\varepsilon^2} \frac{dL(w^2)}{dw} \quad (1.60)$$

จะมีซีโรที่  $w = \pm\Omega_{01}, \pm\Omega_1, \pm\Omega_{02}$  ตามคุณสมบัติข้อที่ 3 จะได้  $1 - F^2(w)$  นั้นจะมี ซีโรอย่างน้อย 2 ค่า คือที่  $2 = 0, \Omega_1$  นั่นคือเราสามารถเขียนได้ด้วย

$$1 - F^2(w) = M_2 w^2 (w^2 - \hat{\Omega}_1)(w^2 - 1)$$

จากสมการที่ (1.60) และคุณสมบัติข้อที่ 1 และ 3 จะได้

$$\frac{dF(w)}{dw} = \frac{1}{2\varepsilon^2 F(w)} \frac{dL(w^2)}{dw} = M_3 w (w^2 - \hat{\Omega}_1)$$

จากข้างบนเราสามารถจัดเป็นรูปสมการดิฟเฟอเรนเชียลได้ดังนี้

$$\left[ \frac{dF(w)}{dw} \right]^2 = \frac{M_4 [1 - F^2(w)]}{1 - w^2}$$

ซึ่งสามารถจัดให้อยู่ในรูปอินทิเกรตแบบจำกัดขอบเขตได้ คือ

$$M_5 \int_0^F \frac{dx}{\sqrt{1-x^2}} + M_6 = \int_0^w \frac{dy}{\sqrt{1-y^2}}$$

ซึ่ง  $F$  และ  $w$  จะมีความสัมพันธ์กันโดยสมการ

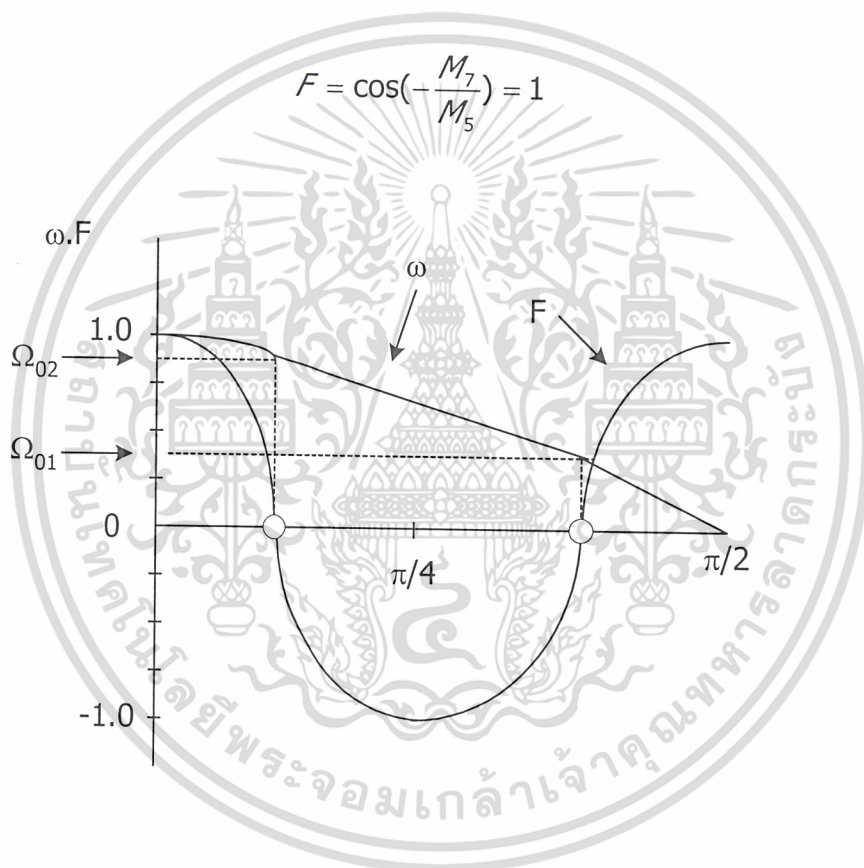
$$M_5 \cos^{-1} F + M_7 = \cos^{-1} w = \theta$$

$\therefore$  ถ้าให้ค่าของ  $\theta$  มาจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w = \cos \theta \text{ และ } F = \cos\left(\frac{\theta}{M_5} - \frac{M_7}{M_5}\right) \quad (1.61)$$

ต่อไปจะเป็นการหาค่า  $M_5$  และ ค่า  $M_7$  ซึ่งจะเห็นว่า ถ้า  $w = 0$  แล้ว จะได้  $\theta = \pi/2$  และถ้า  $w = 1$  แล้วจะได้  $\theta = 0$  ตามรูป 1-24 ส่วน  $F$  จะสัมพันธ์กับ  $F(w)$  คือมันจะมี 2 โพล ในระยะ  $0 \leq \theta \leq \pi/2$  และมีขนาดเป็น 1 ถ้า  $\theta = 0, \pi/2$  ซึ่งรูปร่างของ  $F$  จะแสดงในรูป 1-24 เราจะเห็นได้ว่าถ้า  $\theta = 0$  แล้วจะได้



รูป 1-24 แสดงการ plot ค่า  $F$  และ  $w$  ที่สัมพันธ์กับแกน  $\theta$

หรือ  $M_5 = 0$  แล้ว 1 คาบ ของ  $F$  จะเป็น  $1/4$  เท่าของคาบ  $w$  นั่นคือ

$$2\pi M_5 = \frac{\pi}{2} \text{ หรือ } M_5 = \frac{1}{4}$$

∴ จะได้  $F(w)$  จากสมการที่ 2.12.24 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$F(w) = \cos(4 \cos^{-1} w)$$

และสำหรับ nth – order ของ การประมาณแบบเชบีเชฟ จะได้

นั่นคือ

$$\begin{aligned} F(w) &= T_n(w) = \cos(n \cos^{-1} w) \\ \mathcal{L}(w) &= 1 + \varepsilon [\cos(n \cos^{-1} w)]^2 \end{aligned} \quad (1.62)$$

สมการนี้สำหรับค่า  $|w| \leq 1$  เท่านั้น ส่วนสำหรับ ถ้า  $|w| \geq 1$  ค่าของ  $\cos^{-1}$  จะเป็นจำนวนเชิงซ้อนคือ

$$\cos^{-1} w = j\theta \quad (1.63)$$

และเนื่องจาก

$$w = \cos j\theta = \frac{1}{2} (e^{j(j\theta)} - e^{-j(j\theta)}) = \cosh \theta$$

เราจะได้

$$\theta = \cosh^{-1} w$$

แทนค่า  $\theta$  ลงในสมการ (1.63) จะได้

$$\cos^{-1} w = j \cosh^{-1} w$$

และ

$$\cos(n \cos^{-1} w) = \cos(jn \cosh^{-1} w) = \cosh(jn \cosh^{-1} w)$$

ดังนั้นสำหรับ  $|w| > 1$  สมการที่ (1.62) จะกลายเป็น

$$\mathcal{L}(w^2) = 1 + \varepsilon^2 [\cosh(jn \cosh^{-1} w)]^2 \quad (1.64)$$

สรุปแล้วค่าการสูญเสียในการประมาณของโลพาสฟีลเตอร์แบบเชบีเชฟ จะหาได้จาก

$$A(w) = 10 \log [1 + \varepsilon^2 T_n^2(w)]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

$$T_n(w) = \begin{cases} \cos(n \cos^{-1} w) & \text{for } |w| \leq 1 \\ \cosh(n \cosh^{-1} w) & \text{for } |w| > 1 \end{cases}$$

Normalize Transfer Function ของการประมาณแบบเชบีเชฟ

ถ้าเราแทนค่า  $w=s/j$  ลงในสมการที่ (1.64) จะได้

$$L(-s^2) = 1 + \varepsilon^2 \left[ \cosh\left(n \cos^{-1} \frac{2}{j}\right) \right]^2$$

และถ้าเราให้  $s_k = \sigma_k + j\nu_k$  เป็น ซีโรของ  $L(-s^2)$  เราสามารถเขียนได้ว่า

$$\mu_k + j\nu_k = \cosh^{-1}(-j\sigma_k + w_k) \quad (1.65a)$$

$$\cosh[n(\mu_k + j\nu_k)] = \pm \frac{j}{\varepsilon} \quad (1.65b)$$

จากสมการ (1.65a) จะได้

$$-j\sigma_k + w_k = \cosh(\mu_k + j\nu_k) = \cosh \mu_k \cos \nu_k + j \sin \mu_k \sin \nu_k$$

$$\sigma_k = -\sinh \mu_k \sin \nu_k \quad (1.66)$$

และ

$$w_k = \cosh \mu_k \cos \nu_k \quad (1.67)$$

ในการทำงานเดียวกัน จากสมการที่ (1.65b) จะได้

$$\cosh n\mu_k \cos n\nu_k + j \sin n\mu_k \sin n\nu_k = \pm \frac{j}{\varepsilon}$$

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\cosh n u_k \cos n v_k = 0 \quad (1.68a)$$

และ

$$\sinh n u_k \sin n v_k = \pm \frac{j}{\varepsilon} \quad (1.68b)$$

ตามสมการที่ (1.68a) จะได้

$$v_k = \frac{(2k-1)\pi}{2n} \text{ โดยที่ } k = 1, 2, \dots, n \quad (1.69)$$

และเนื่องจาก  $\sin(nv_k) = \pm 1$  และจากสมการที่ (1.68b) จะได้

$$u_k = \pm \frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon} \quad (1.70)$$

ดังนั้นจากสมการที่ (1.68), (1.69), (1.70) จะได้

$$\sigma_k = \pm \sinh\left(\frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon}\right) \sin \frac{(2k-1)\pi}{2n}$$

$$w_k = \cosh\left(\frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon}\right) \cos \frac{(2k-1)\pi}{2n}$$

โดยที่  $k = 1, 2, \dots, n$  และ

$$\frac{\sigma_k^2}{\sinh^2 u} + \frac{w_k^2}{\cosh^2 u} = 1$$

ซึ่ง ซีโรของ  $L(-s^2)$  จะอยู่บนวงรี ดังที่ได้แสดงไว้ในรูปที่ 1-25

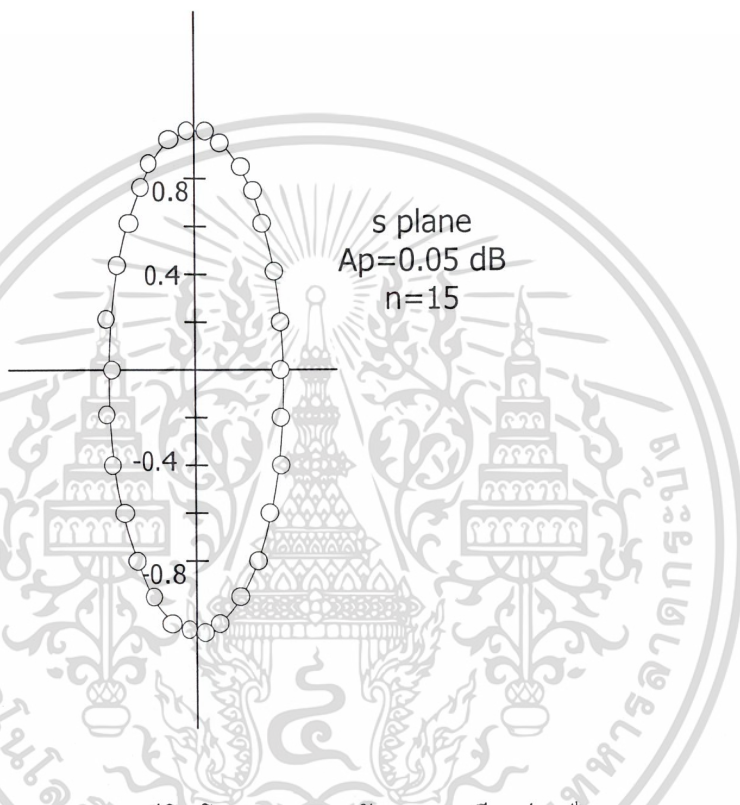
Normalized Transfer Function สามารถเขียนอยู่ในรูป

$$H_N(s) = \frac{H_0}{\prod_{i=1}^n (s - p_i)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $p_i$  สำหรับ  $i = 1, 2, \dots, n$  คือ ซีโรของ  $L(-s^2)$  ที่อยู่ด้านซ้าย ของระนาบ  $S$  ค่าคงที่  $H_0$  หาได้จาก

$$H_0 = \begin{cases} 10^{-0.05A} \prod_{j=1}^N (-p_j); n=even \\ \prod_{i=1}^N (-p_i); n=odd \end{cases}$$



รูป 1-25 แสดง ซีโร-โพล plot ของตัวกรองเชบีเชฟ 15<sup>th</sup>-order

### 1.3 โครงสร้างของวงจรถองเชิงเลข

การที่จะออกแบบให้ตัวกรองมีคุณสมบัติตามต้องการนั้น ในเบื้องต้นเราต้องกำหนดคุณสมบัติของตัวกรองขึ้นมา การกำหนดคุณสมบัติอาจอยู่ในรูป ผลตอบสนองความถี่ ซึ่งเป็นคุณสมบัติในโดเมนความถี่ หรืออาจกำหนดในรูป ผลตอบสนองอิมพัลส์ ซึ่งเป็นคุณสมบัติในโดเมนเวลา แล้วจากผลตอบสนองของระบบที่กำหนดขึ้นนี้เราใช้เป็นข้อกำหนดในการออกแบบ ซึ่งก็มีวิธีการออกแบบตัวกรองได้หลายวิธีการ อย่างไรก็ตามทุกวิธีการออกแบบได้ผลลัพธ์เป็นฟังก์ชัน ที่มีชื่อเรียกว่า ฟังก์ชันถ่ายโอน หรือ  $H(Z) = N(Z) / D(Z)$  ขึ้นมา และ ฟังก์ชันถ่ายโอนนี้จะเป็นฟังก์ชันที่ให้คุณสมบัติการกรองตามต้องการ ในหัวข้อนี้เราจะดูถึงวิธีการนำเอาฟังก์ชันถ่ายโอนมาสร้างเป็นโครงสร้างวงจรถองเชิงเลขแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปสำหรับฟังก์ชันถ่ายโอนอันดับหนึ่ง สามารถนำไปสร้างเป็น โครงสร้างวงจรกรองได้หลายรูปแบบ ลักษณะของรูปแบบขึ้นอยู่กับทางเลือกใช้ ระเบียบวิธีการทำให้เป็นจริง (realization method) ซึ่งเป็น กระบวนการ หรือกรรมวิธี ในการแปลฟังก์ชันถ่ายโอนให้เป็น โครงข่ายหรือ โครงสร้างวงจร ส่วนการเลือกจะนำโครงสร้างแบบใดไปใช้สร้างตัวกรองนั้น ไม่สามารถกำหนดลงไปได้อย่างแน่นอนได้ ทั้งนี้เนื่องจากมีองค์ประกอบหลายประการที่ต้องนำมาพิจารณา ร่วมด้วย เช่น ความเร็วในการคำนวณ สภาพไวต่อการควอนไทซ์สัมประสิทธิ์ (sensitivity to coefficient quantization) ระดับของสัญญาณรบกวนที่จุดสัญญาณออก จากการควอนไทซ์ซึ่งอาจ เป็นการ บิดเบือน หรือการตัดปลาย ของผลการคำนวณทางคณิตศาสตร์ หรือปรากฏการณ์ไม่เป็นเชิงเส้นที่อาจเกิดขึ้นได้ เป็นต้น

ในที่นี้เมื่อกล่าวถึงคำว่า โครงสร้าง (structure) เราจะเขียนแทนด้วยแผนภาพ ที่ประกอบ ขึ้นด้วย ตัวบวกสัญญาณ ตัวคูณสัญญาณ ตัวหน่วงสัญญาณ ส่วนทิศทางการไหลของสัญญาณจะ แทนด้วยทิศทางของหัวลูกศร ซึ่งแผนภาพที่ได้นี้แทน การคำนวณ ที่อาจสร้างเป็น วงจรเฉพาะหรือ ฮาร์ดแวร์หรือเขียนเป็นโปรแกรม หรือเป็นส่วนผสมระหว่างวงจรและโปรแกรมก็ได้

### 1.3.1 โครงสร้างของวงจรกรองป้อนกลับเชิงเลข

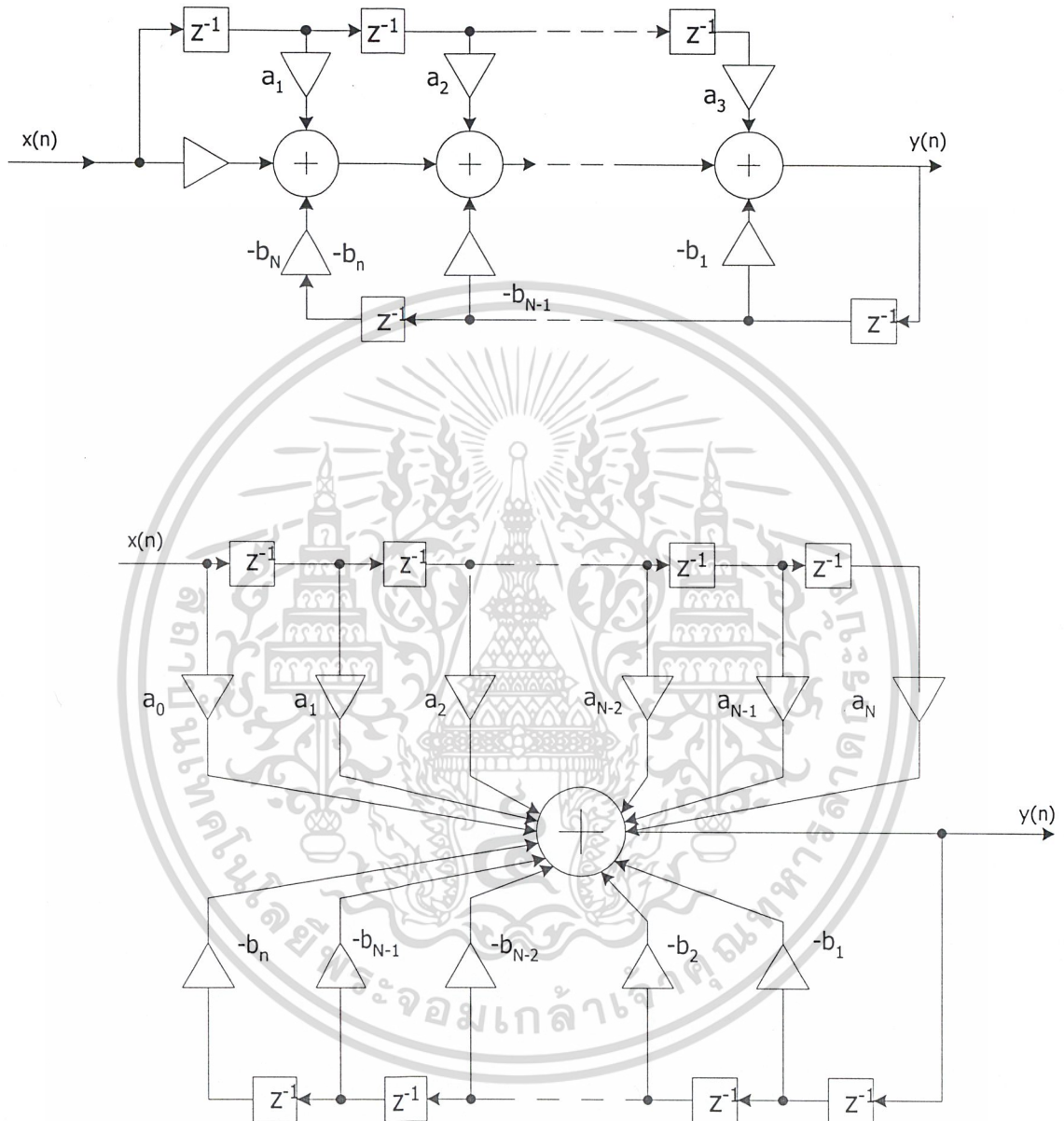
ตอนนี้มาพิจารณาถึงตัวป้อนกลับเชิงเลขซึ่งอาจเขียนอธิบายได้ด้วย สมการผลต่างสืบเนื่อง

$$y(n) = \sum_{i=0}^M a_i x(n-i) + \sum_{i=1}^N b_i y(n-i) \quad (1.71)$$

และเมื่อทำการแปลงโดยใช้การแปลงแซด ได้ฟังก์ชันถ่ายโอนตัวกรองนี้ เป็นสมการนี้ เป็น

$$H(Z) = Y(Z)/X(Z) = \sum_{i=0}^M a_i Z^{-i} / \sum_{i=0}^N b_i Z^{-i}$$

สามารถนำไปสร้างวงจรกรองเชิงเลขได้ และมีวิธีการหรือโครงสร้าง อยู่หลายแบบด้วยกันอย่างไรก็ตามจะขอกกล่าวถึงเฉพาะ โครงสร้างแบบโดยตรง 1 และ โครงสร้างแบบโดยตรง 2



รูป 1-26 โครงสร้างวงจรป้อนกลับเชิงเลข โดยตรง

(a) เขียนแยกตัวบวกสัญญาณ ส่วน (b) รวมตัวบวกสัญญาณเป็นตัวเดียว

### 1.3.1 โครงสร้างแบบโดยตรง 1 หรือ DF1

ที่เรียกชื่อ โครงสร้างแบบโดยตรง (direct form I) เพราะการสร้างวงจรนั้นสร้างโดยเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตรงมาจาก สมการผลต่างสืบเนื่อง โดยที่เลือก  $b_0 = 1$  รูป (1-26a) แสดงถึงแผนภาพของโครงสร้างของตัวกรองแบบนี้ และในรูป (1-26b) เป็นแผนภาพของวงจรเดียวกัน แต่ได้เขียนโดยรวมเอาวงจรบวกมาไว้ในส่วนเดียวกัน

ข้อเสีย ของโครงสร้างแบบนี้ก็คือ ต้องใช้ อุปกรณ์หน่วยสัญญาณ หรือ ชิฟต์รีจิสเตอร์ถึงสองชุด กล่าวคือ ชุดหนึ่งไว้บรรจุลำดับสัญญาณเข้า  $x(n-i)$  อีกชุดหนึ่งไว้ลำดับสัญญาณออก  $y(n-i)$  ข้อเสีย อีกประการหนึ่งคือ โครงสร้างของวงจรกรองแบบนี้ ไม่เหมาะสำหรับตัวกรองที่มีโพลวางตัวอยู่ใกล้เส้นรอบวงของวงกลมหนึ่งหน่วย ทั้งนี้เนื่องมาจากปรากฏการณ์ไม่เป็นเชิงเส้น เช่น สิ่งหรือสัญญาณรบกวนจากผลของการควอนไทซ์ผลคูณ และสภาพไวของตัวกรองจากการควอนไทซ์ค่าสัมประสิทธิ์ของวงจรกรองมีค่ามาก ดังนั้นโดยทั่วไปนิยมสร้างเป็นวงจรที่มีอันดับต่ำ เช่น ตัวกรองอันดับหนึ่ง หรืออันดับสอง เป็นต้น เนื่องจากการควบคุมการเกิดปรากฏการณ์ไม่เป็นเชิงเส้นทำได้ง่าย สังเกตจากรูป (1-26) จะเห็นได้ว่าโครงสร้างแบบนี้ ลำดับสัญญาณเข้า ถูกคูณด้วยสัมประสิทธิ์  $a_i$  ซึ่งเป็นค่าสัมประสิทธิ์ในส่วนของ ชิโร ของฟังก์ชันถ่ายโอน ดังนั้นจึงอาจมีชื่อเรียกอีกอย่างว่า โครงสร้างแบบ ชิโรก่อนโพล (zeros before poles) กล่าวโดยสรุปก็คือ การสร้างตัวกรองแบบนี้ทำได้โดยการสร้างฮาร์ดแวร์ หรือวงจร หรือเขียนโปรแกรมเพื่อคำนวณสมการ (1.71)

### 1.3.2 โครงสร้างรูปแบบโดยตรง 2 หรือ DF2

หนังสือบางเล่มเรียก โครงสร้าง แบบ โดยตรง 2 (direct form II) นี้ว่า โครงสร้างแบบ แบบบัญญัติ (canonic form) ทั้งนี้เพราะว่าโครงสร้างวงจรแบบนี้ใช้จำนวนอุปกรณ์ ตัวคูณสัญญาณ ตัวหน่วยสัญญาณ และตัวบวกสัญญาณน้อยที่สุด อย่างไรก็ตามก็อาจมีโครงสร้างแบบนี้เริ่มโดย การเขียน  $H(Z)$  ออกเป็น 2 วงจรคูณ หรือต่อкасцепт กันคือ

$$\begin{aligned} H(Z) &= Y(Z) / X(Z) = \left\{ 1 / \sum_{i=0}^N b_i Z^{-i} \right\} \cdot \left\{ \sum_{i=0}^M a_i Z^{-i} \right\} \\ &= H_1(Z) \cdot H_2(Z) \end{aligned} \quad (1.72)$$

โดยที่  $H_1(Z)$  เป็นวงจรกรองที่มีโพลเพียงอย่างเดียว และ  $H_2(Z)$  เป็นวงจรที่มีชิโรเพียงอย่างเดียว ถ้าเรานิยามได้ว่า

$$H_1(Z) = W(Z)/X(Z) = 1/\sum_{i=0}^M b_i Z^{-i} \quad (1.73)$$

และ

$$H_2 = Y(Z)/W(Z) = \sum_{i=0}^M a_i Z^{-i} \quad (1.74)$$

แล้วนำสองสมการนี้มา เขียนในรูปสมการผลต่างสืบเนื่องได้ ซึ่งการคำนวณทำได้ตามลำดับ คือ

$$w(n) = x(n) - \sum_{i=1}^M b_i w(n-i) \quad (1.75)$$

และ

$$y(n) = \sum_{i=0}^M a_i w(n-i) \quad (1.76)$$

โดยที่  $b_0 = 1$  เมื่อนำสองสมการนี้ไปเขียนเป็นแผนภาพจะได้ดังรูป (1-27a) และ เนื่องจากสัญญาณในชุดตัวหนังสือสัญญาณในวงจรป้อนกลับ และสัญญาณในชุดตัวหนังสือสัญญาณในส่วนวงจรป้อนตรง เป็นสัญญาณเดียวกันคือ  $w(n)$  ดังนั้นจึงสามารถเขียนชุดตัวหนังสือสัญญาณทั้งสองเข้าเป็นชุดเดียวกัน ดังรูป (1-27b) ซึ่งทำให้เห็นว่าโครงสร้างแบบบัญญัตินี้ใช้ตัวหนังสือสัญญาณเพียง  $N$  ตัว สำหรับกรณีตัวกรองอันดับ  $N$

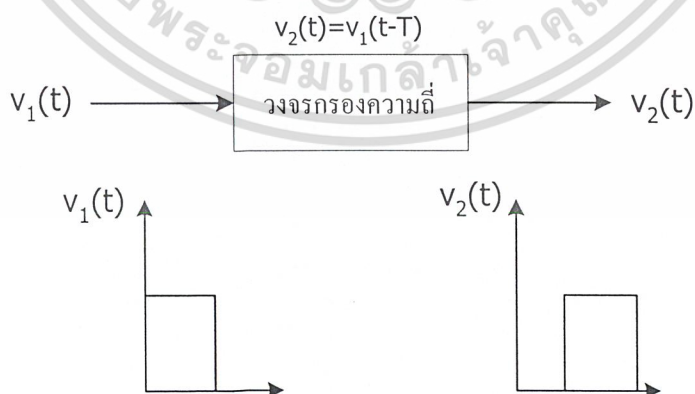
สังเกตตามโครงสร้างของตัวกรองแบบนี้ตามรูป (1-27b) จะเห็นได้ว่า การคำนวณสัญญาณออก  $y(n)$  นั้น ในเบื้องแรกต้องทำการคำนวณหา ลำดับสัญญาณ  $w(n)$  ซึ่งเป็น สัญญาณระหว่างกลางที่ได้แทนในสมการ (1.75) ก่อน แล้วนำค่าของสัญญาณระหว่างกลางที่ได้แทนในสมการ (1.76) เพื่อคำนวณหา  $y(n)$  ตามต้องการ ข้อดีของโครงสร้างแบบโดยตรง 2 ก็คือคุณสมบัติความไม่เป็นเชิงเส้น เช่น สิ่งรบกวนจากผลการคูณ หรือสภาพไวของสัมประสิทธิ์ คล้ายกับแบบโดยตรง 1 แต่ข้อดีดีกว่าโครงสร้างแบบโดยตรง 1 ตรงที่ว่าโอกาสเกิด ล้น หรือ โอเวอร์โพล์ ของการบวกตัวเลขฐานสองเป็นไปได้มากกว่า ทั้งนี้เนื่องจากสัญญาณออกของโครงสร้างแบบโดยตรง 2 เป็นการเอาลำดับ  $w(n)$  มาคูณค่ากับสัมประสิทธิ์ของตัวกรอง แล้วนำมาบวกกัน โดยที่ไม่ได้มีการป้อนกลับของสัญญาณออก ดังนั้นโอกาสการเกิด โอเวอร์โพล์ จึงมีได้มากกว่า



แก่ผลตอบสนองแอมพลิจูดมากกว่า นั่นคือ ออกแบบให้มีผลตอบสนองแอมพลิจูดตามต้องการโดยยอมให้ผลตอบสนองเฟสเป็นอย่างไรก็ได้ แต่ในงานบางลักษณะนั้น เช่น การประมวลผลสัญญาณภาพ เป็นต้น ผลตอบสนองเฟสมีความสำคัญมาก หรือ ในการประมวลผลสัญญาณอีซีจี ( ECG : electrocrdiogram) ที่ขนาดและลักษณะรูปคลื่นของสัญญาณ และระยะเวลาในการเกิดสัญญาณมีความสำคัญมาก ในสถานการณ์แบบนี้ ถ้าใช้ตัวกรองที่มีผลตอบสนองเฟสไม่เป็นเชิงเส้นมาทำการประมวลผลสัญญาณ ก็จะทำให้ค่าขนาด ลักษณะรูปคลื่น หรือเวลาในการเกิดสัญญาณนี้ผิดเพี้ยนไปได้ ดังได้กล่าวไว้ข้างแล้วว่่าข้อดีของตัวกรองไม่ป้อนกลับเชิงเลข คือสามารถออกแบบให้มีผลตอบสนองเฟสเป็นเชิงเส้นได้ง่าย ในเรื่องนี้ เราจะพิจารณาถึงคุณสมบัติของตัวกรอง และการออกแบบตัวกรองแบบนี้โดยละเอียด

### 1.4.1 วงจรกรองแบบหน่วงเวลา (Delay filters)

เพื่อให้ความกระจ่างเกี่ยวกับ คุณสมบัติการมีผลตอบสนองเฟสเป็นเชิงเส้น เรามาดูความรู้เกี่ยวกับวงจรกรองแบบหนึ่งที่มีชื่อเรียกว่า วงจรกรองหน่วงเวลา วงจรกรองแบบนี้มีคุณสมบัติ ดังแสดงไว้ในรูป 1-28 คือยอมให้สัญญาณผ่านออกไปได้โดยที่ ลักษณะรูปคลื่นของสัญญาณออก ไม่ผิดเพี้ยน เพียงแต่เวลาของการเกิดรูปคลื่นจะมีเวลาหน่วงออกไป  $T$  วินาที โดยที่  $T$  เป็นค่าเวลาคงตัวที่กำหนดให้ ถ้าให้  $v_1(t)$  เป็นสัญญาณออกจากวงจรกรอง และ  $v_2(t)$  เป็นสัญญาณเข้าของวงจรกรอง เราสามารถเขียนความสัมพันธ์ของทั้งสองสัญญาณนี้ได้เป็น



รูป 1-28 แสดงคุณสมบัติของตัวกรองหน่วงเวลา

จากการวิเคราะห์โดยการประยุกต์ใช้ ผลการแปลงฟูรีเยอร์ ทำให้รู้ว่าสัญญาณสี่เหลี่ยมตามรูป (1-27) สามารถแตกย่อยออกเป็น สัญญาณไซน์จำนวนอนันต์มาประกอบรวมกัน ดังนั้นสามารถศึกษาถึงผลตอบสนองเฟสของวงจรโดยนำพิจารณาแต่ละความถี่ได้ ถ้าเรานำสัญญาณไซน์ที่มีความถี่เดียวคือที่ความถี่  $\omega$  มาพิจารณา หรือให้สัญญาณเข้าเป็น

$$v_1(t) = A\sin(\omega t + \phi) \quad (1.77)$$

โดยอาศัยความสัมพันธ์ตามสมการ (1.77) เราเขียนสัญญาณออกในกรณีนี้ได้เป็น

$$v_2 = A\sin(\omega\{t - T\} + \phi) \quad (1.78)$$

หรือ

$$v_2 = A\sin(\omega t - \omega\tau + \phi) \quad (1.79)$$

และถ้าให้  $\theta$  เป็นค่าเฟส ที่ต่างกันระหว่าง  $v_1$  และ  $v_2$  จะได้

$$\theta = -\omega\tau \quad (1.80)$$

ซึ่งจะเห็นว่า เฟส มีค่าเท่ากับค่าคงตัว  $T$  คูณกับความถี่  $\omega$  หรือ วงจรมีค่าแปรเป็นเชิงเส้นกับความถี่ สำหรับสัญญาณไซน์ตัวอื่น ก็สามารถพิสูจน์ได้ในทำนองเดียวกันว่า วงจรกรองให้ผลตอบสนองเป็นแบบหน่วงเวลาตามสมการ (1.77) ได้ก็ต่อเมื่อมีผลตอบสนองความถี่เชิงเส้น กับความถี่ ดังสมการ (1.80) โดยที่  $\omega$  เป็นค่าความถี่ใดๆ

คุณสมบัติของวงจรนี้อาจศึกษาให้ละเอียดลงไปโดยการเขียนความสัมพันธ์ของสัญญาณเข้าและสัญญาณออกจากวงจรกรองใหม่ในรูปของ เฟสเซอร์ (phasor) คือ

$$V_1(\tau) = A\angle\phi \text{ และ } V_2(t) = A\angle(\phi - \omega\tau)$$

และฟังก์ชันถ่ายโอนเขียนได้เป็น

$$T(s) = V_2(s)/V_1(s) = 1\angle -\omega\tau \quad (1.81)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ

$$T(s) = \exp(-j\omega\tau) \quad (1.82)$$

ซึ่งพจน์  $s = j\omega$  และ  $T(s)$  ก็จะเป็นจำนวนเชิงซ้อน

$$T(\omega) = R(\omega) + jU(\omega) \quad (1.83)$$

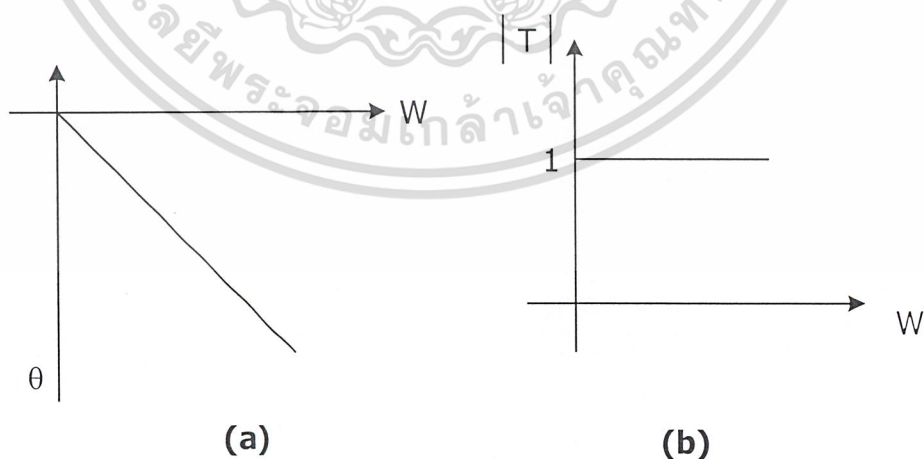
เพราะฉะนั้นผลตอบสนองความถี่ ซึ่งแบ่งพิจารณาเป็น 2 ส่วน คือผลตอบสนองแอมพลิจูด และผลตอบสนองเฟส หาได้ตามลำดับโดย

$$\text{ผลตอบสนองแอมพลิจูด} = |V_2(s)/V_1(s)| = |T(\exp\{-j\omega\tau\})| \quad (1.84)$$

และ

$$\text{ผลตอบสนองเฟส} = \theta(\omega) = \tan^{-1}\{U(\omega)/R(\omega)\} \quad (1.85)$$

ผลตอบสนองความถี่ของ วงจรกรองหน่วยเวลา ได้แสดงไว้ในรูป (1-29a) ซึ่งนอกจากแสดงให้เห็นในรูป (1-29b) ว่ามีผลตอบสนองเฟสเป็นเชิงเส้นแล้ว ยังให้ผลตอบสนองแอมพลิจูดเป็นแบบตัวกรองผ่านทุกความถี่ (all pass filter) ด้วย



รูปที่ 1-29 แสดง

(a) ผลตอบสนองเฟส และ (b) ผลตอบสนองแอมพลิจูดของตัวกรองหน่วยเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกล่าวถึง คุณสมบัติของวงจรกรอง นั้นยังมีคำอีกสองคำที่มักใช้ในการบ่งถึงคุณสมบัติของผลตอบสนองด้วยเฟสด้วย คำแรกก็คือ ความหน่วงเฟส (phase delay) จะเขียนแทนด้วย  $\tau_p$  ซึ่งนิยามให้เป็นอัตราส่วนของเฟสต่อความถี่ หรือ

$$\tau = -\theta(\omega) / \omega \quad (1.86)$$

เทอมที่สองคือ ความหน่วงกลุ่ม (group หรือ signal หรือ envelope delay) จะเขียนแทนด้วย  $\tau_G$  ซึ่งนิยามให้เป็นค่าอนุพันธ์ของเฟสเมื่อเทียบกับความถี่ หรือ

$$\tau_G = -d\theta(\omega) / d\omega \quad (1.87)$$

ถ้าหากนำนิยามทั้งสองนี้มาประยุกต์ใช้กับ ผลตอบสนองเฟสของวงจรกรองหน่วงเวลา ที่มีคุณสมบัติตามสมการ (1.77) แล้วจะเห็นได้ว่า ตัวกรองหน่วงเวลามีผลตอบสนองเฟสเป็นแบบความหน่วงเฟสคงตัว (constant phase delay) และ ความหน่วงกลุ่มคงตัว (constant group delay) ด้วย ซึ่งตัวกรองโดยทั่วไปอาจให้คุณสมบัติเพียง ความหน่วงเฟสคงตัว หรือ ความหน่วงกลุ่มคงตัวเพียงอย่างเดียวก็ได้

#### 1.4.2 การออกแบบโดยใช้อนุกรมฟูรีเยอร์จำนวนพจน์คงตัว

ตัวกรองไม่ป้อนกลับนี้สามารถเขียนได้ตามสมการทั่วไป (1.88) ซึ่งจะเห็นว่าความยาวของตัวกรองจำกัด หรือฮิพตรีจิสเตอร์ มีจำนวนจำกัด ถ้าให้  $x(n)$  เป็นลำดับสัญญาณเข้าและให้  $y(n)$  เป็นลำดับสัญญาณออก เพราะฉะนั้นเขียนสมการผลต่างสืบเนื่องได้

$$\begin{aligned} y(n) &= \sum_{k=-M}^{+M} C_k \cdot x(n-k) \\ &= C_0 x(n) + \sum_{k=-1}^{-M} C_k \cdot x(n-k) + \sum_{k=1}^{+M} C_k \cdot x(n-k) \\ &= C_0 x(n) + \sum_{k=1}^{+M} \{C_{-k} \cdot x(n+k) + C_k \cdot x(n-k)\} \end{aligned} \quad (1.88)$$

โดยการสมมติให้  $T = 1$  และประยุกต์ใช้ การแปลงแซด เพื่อเขียนฟังก์ชันถ่ายโอนจะได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(Z) = c_0 + \sum_{k=1}^M \{c_{-k} z^k + c_k z^{-k}\} = \sum_{k=-M}^M \{c_k z^{-k}\}$$

หรือ

$$H(\omega) = c_0 + \sum_{k=1}^M \{c_{-k} e^{j\omega k} + c_k e^{-j\omega k}\} = \sum_{k=-M}^M \{c_k \cdot e^{-j\omega k}\}$$

ถ้าพิจารณากรณีตัวกรองแบบสมมาตรคู่หรือ  $C_k = C_{-k}$

$$H(\omega) = c_0 + \sum_{k=1}^M c_k (e^{j\omega k} + e^{-j\omega k}) \quad (1.89)$$

$$= c_0 + \sum_{k=1}^M 2c_k \left( \frac{e^{j\omega k} + e^{-j\omega k}}{2} \right) \quad (1.90)$$

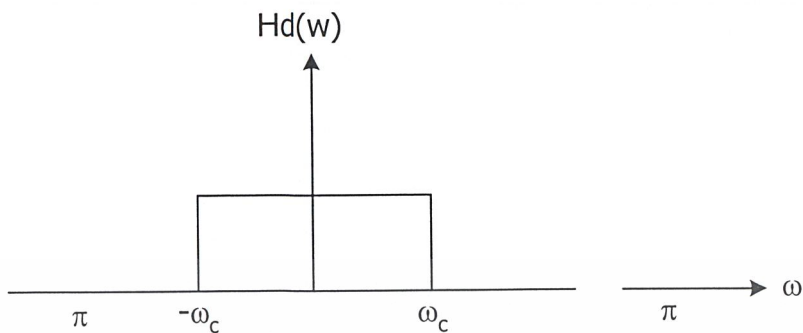
$$H(\omega) = a_0 + \sum_{k=1}^M a_k \cos \omega k$$

โดยที่กำหนดให้  $a_0 = c_0$  และ  $a_k = 2c_k$  และ  $k > 0$

ผลที่ได้จากสมการ (1.90) เห็นได้ชัดว่า ตัวกรองไม่ป้อนกลับเชิงเลขคณิตสมมาตรคู่ นั้นสามารถเขียนอธิบายได้โดยใช้ อนุกรมฟูรีเยอร์ ที่ประกอบด้วยสัมประสิทธิ์เฉพาะโคไซน์เท่านั้น และค่าสัมประสิทธิ์ของอนุกรมฟูรีเยอร์  $a_k$  ตามสมการ (1.90) ก็จะเป็นค่าสัมประสิทธิ์ของตัวกรองด้วยในทำนองเดียวกันเราอาจแสดงได้ว่า สำหรับตัวกรองไม่ป้อนกลับเชิงเลขคณิตสมมาตรคู่ สามารถเขียนแทนด้วยอนุกรมฟูรีเยอร์ที่มีเฉพาะไซน์เท่านั้น และ ค่าสัมประสิทธิ์จะเป็นค่าจินตภาพเสมอ

ที่นี้มาดูหลักการออกแบบ โดยปกติแล้วการออกแบบตัวกรองสัญญาณนั้น กระทำโดยการทำการประมาณค่า ให้ตัวกรองที่ออกแบบมีผลตอบสนองความถี่ได้ใกล้เคียงกับผลตอบสนองอุดมคติดังรูป (1-29) ให้มากที่สุด โดยตามรูป  $\omega_c$  เป็นค่าความถี่ตัด (cut off frequency) จากผลตอบสนองความถี่อุดมคติ  $H_d(\omega)$  สามารถกระจายให้อยู่ในพจน์ของโดเมนเวลาโดยการประยุกต์ใช้ การแปลงฟูรีเยอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1-30 ผลตอบสนองแอมพลิจูดอุดมคติ

$$H_d(\omega) = \sum_{-\infty}^{\infty} h_d(k) e^{-jk\omega T} \quad (1.91)$$

โดยในกรณีนี้ ผลตอบสนองอิมพัลส์  $h_d(k)$  หาได้จาก

$$h_d(k) = (1/2\pi) \int_{-\omega_s/2}^{\omega_s/2} H_d(\omega) e^{jk\omega T} d\omega \quad (1.92)$$

แต่จากสมการ (1.90) ผลตอบสนองความถี่ของตัวกรองไม่ย้อนกลับเชิงเลขมีค่าเป็น

$$h_d(k) = (1/2\pi) \int_{-\omega_s/2}^{\omega_s/2} H_d(\omega) e^{jk\omega T} d\omega \quad (1.93)$$

เมื่อเปรียบเทียบกับสมการ (1.91) และ(1.93) จะเห็นได้ว่าผลตอบสนองความถี่ของตัวกรองใกล้เคียงอุดมคติก็ต่อเมื่อ  $M = \infty$  แต่ในทางปฏิบัติเป็นไปได้ยาก เราจึงทำการตัดปลาย อนุกรมฟูรีเยร์ให้เหลือเฉพาะเท่าที่จำเป็น และจัดให้

$$c_k = h_d(k) \quad (1.94)$$

โดยที่  $k = -M, \dots, 0, \dots, M$  ส่วนความยาวของตัวกรอง  $N_s = (2M+1)$  เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.4.3 การออกแบบโดยใช้วินโดว์

เราได้แสดงให้เห็นว่าการถ่วงน้ำหนักอนุกรมฟูรีเยอร์ให้เหมาะสมสามารถทำให้ลูกคลื่นที่เกิดบนผลตอบสนองความถี่ลดน้อยลงไปได้ โดยทั่วไปการถ่วงน้ำหนักอาจทำได้หลายวิธีซึ่งจะกล่าวอย่างละเอียดในหัวข้อนี้ จากเดิมที่เราตัดปลายอนุกรมฟูรีเยอร์จำนวนอนันต์พจน์ผ่านหน้าต่างนี้ออกไป อนุกรมนี้จะถูกถ่วงน้ำหนัก หรือถูกตัดปลายตามต้องการ

ถ้าให้  $H_d(w)$  แทนผลตอบสนองความถี่อุดมคติ และผลตอบสนองนี้สามารถเขียนแทนด้วยอนุกรมฟูรีเยอร์ความยาวอนันต์พจน์ ได้คือ

$$H_d(w) = \sum_{-\infty}^{\infty} h_d(n) e^{-jn\omega T} \quad (1.95)$$

โดยที่  $h_d(n)$  หาได้จาก

$$h_d(n) = (1/2\pi) \int_{-\pi}^{\pi} H_d(\omega) \cdot e^{jn\omega T} d\omega \quad (1.96)$$

หรือ

$$h_d(n) = (1/w_s) \int_{\omega_s/2}^{\omega_s/2} H_d(w) \cdot e^{jn\omega T} d\omega \quad (1.97)$$

โดยที่  $w_s$  เป็นค่าความถี่ในการสุ่มตัวอย่าง ตามที่ทราบกันแล้วว่าการนำสมการ (1.95) และ (1.96) ไปใช้ในการออกแบบตัวกรองไม่ป้อนกลับเชิงเลข ซึ่งต้องตัดปลายอนุกรมในสมการ (1.95) ให้เป็นอนุกรมจำนวนพจน์จำกัด หรือ

$$H_d = \sum_{n=(N-1)/2}^{(N-1)/2} h_d(n) e^{-jn\omega T} \quad (1.98)$$

ซึ่งผลลัพธ์ก็คือ ทำให้เกิดลูกคลื่นบนผลตอบสนองแอมพลิจูด แต่ก็ยังไม่สามารถนำสมการ (1.97) ไปใช้งานได้ เพราะว่าผลตอบสนองอิมพัลส์เริ่มจากลำดับที่  $n = (N-1)/2$  ซึ่งสำหรับระบบเวลาจริงไม่สามารถสร้างลำดับเวลาที่เป็นลบได้ ข้อนี้แก้ไขได้โดยการเลื่อน หรือห้วงผลตอบสนองอิมพัลส์ออกไป โดยให้เริ่มจากค่า  $n = 0$  หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(\omega) = \sum_{n=0}^{N-1} h(n) e^{-jn\pi T} \quad (1.99)$$

ซึ่งการทำแบบนี้มีผลก็คือ ทำให้ตัวกรองมีผลตอบสนองเฟสเป็นเชิงเส้น

จากสมการ (1.99) จะเห็นได้ว่าการที่เราตัดปลายอนุกรมฟูรีเยอร์ก็คือ การที่เราทำการเลือกให้

$$\begin{aligned} h(n) &= h_d(n) \quad \text{เมื่อ } 0 \leq n \leq N-1 \\ &= 0 \quad \text{เมื่อ } n \text{ เป็นค่าอื่น} \end{aligned} \quad (1.100)$$

ซึ่งสมการนี้ถ้าหากเขียนให้อยู่ในรูปสมการทั่วไปคือ การนำเอาลำดับ  $h_d(n)$  มาคูณกับค่าลำดับจำกัด  $w(n)$  หรือ

$$\begin{aligned} h(n) &= 1 \quad \text{เมื่อ } 0 \leq n \leq N-1 \\ &= 0 \quad \text{เมื่อ } n \text{ เป็นค่าอื่น} \end{aligned} \quad (1.101)$$

#### 1.4.4 วินโดว์แบบ ฮาน แฮมมิง และ แบล็คแมน

ฮานวินโดว์ และ แฮมมิงวินโดว์ สามารถเขียนเป็นสมการของลำดับได้ โดยให้  $T = 1$  วินาที ดังนี้

$$\begin{aligned} W_H(n) &= \alpha + 1(1 - \alpha) \cos(2\pi n / (N - 1)); |n| \leq (N - 1) / 2 \\ &= 0 \quad ; \text{เมื่อ } n \text{ กรณีอื่น} \end{aligned} \quad (1.102)$$

โดยที่  $\alpha = 0.5$  สำหรับกรณีฮานวินโดว์ และ  $\alpha = 0.54$  สำหรับกรณีแฮมมิงวินโดว์ ผลตอบสนองความถี่ของวินโดว์ทั้งสองแบบ สามารถหาได้ตามลำดับดังนี้

สมการ (1.102) สามารถเขียนในพจน์ของวินโดว์สี่เหลี่ยม ได้คือ

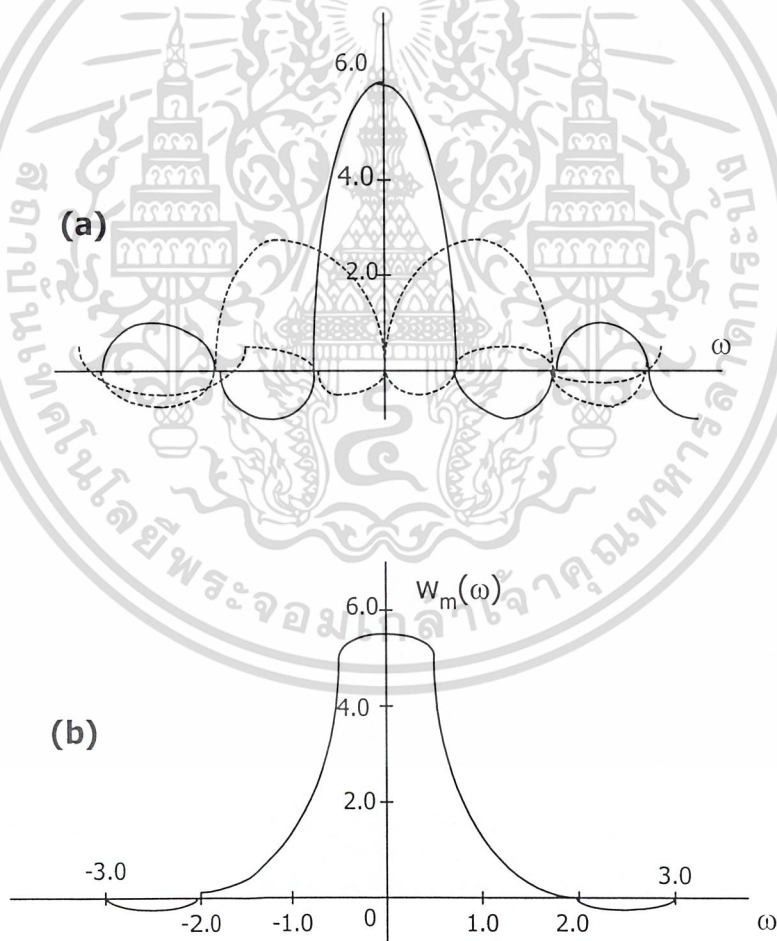
$$\omega_H(n) = \omega_R(\alpha + (1 - \alpha) \cos(2\pi n / (N - 1))) \quad (1.103)$$

เมื่อประยุกต์ใช้ผลการแปลง Z กับลำดับในสมการ (1.103) ได้ผลตอบสนองสมการความถี่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

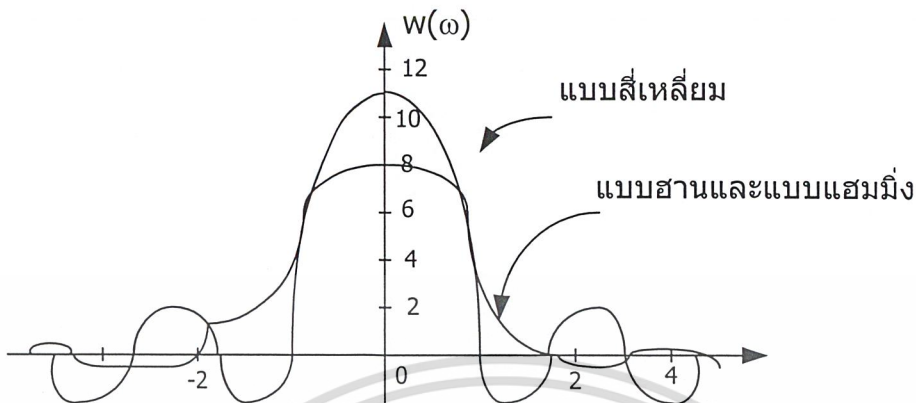
$$\begin{aligned}\omega_H(\exp[j\omega T]) &= Z[\omega_H(n)]_{z=e^{j\omega T}} \\ &= \alpha\omega_R(\exp[j\omega T]) + \{(1-\alpha)/2\}\omega_R(\exp[j(\omega T - 2\pi)/(N-1)])\end{aligned}$$

สเปกตรัมของวินโดว์นี้วาดได้ในรูป 1-31 ซึ่งจะเห็นได้ว่า เทอมที่ 2 และเทอมที่ 3 ของสมการ (1.104) เป็นการถ่วงน้ำหนัก  $w_R(e^{j\omega T})$  แล้วเลื่อนไปทางขวาและซ้าย  $2\pi/(N-1)T$  ลำดับการอย่างนี้จุดประสงค์ก็เพื่อไปหักล้าง กับไซด์โพล (side lobe) ทั้งสองข้างแล้วไปเพิ่มขนาดความกว้างของโพลหลักขึ้นมา ผลการรวมดังรูป (1-31b) เห็นได้ชัดว่าโพลหลักเพิ่มขึ้น และถูกคลื่นลดลงไปอย่างมาก ความกว้างของโพลหลักของวินโดว์นี้มีค่าโดยประมาณเท่ากับ  $4w_a/N$



รูป 1-31 สเปกตรัมของฮานและแฮมมิงวินโดส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1-32 สเปกตรัมของวินโดว์สี่เหลี่ยม และฮานวินโดว์กรณี  $N = 11$

ในรูป (1-32) นั้นเป็นการเปรียบเทียบกันระหว่างสเปกตรัมความถี่ของ วินโดว์สี่เหลี่ยม ฮานวินโดว์ และแฮมมิงวินโดว์ ซึ่งเห็นได้ชัดว่าแบบสี่เหลี่ยม พลังงานในส่วนความถี่นอกโหลบลหลักมีอยู่มาก ในแบบฮานและแฮมมิง พลังงานได้ถ่ายเข้าไปใน โหลบลหลักเกือบ 99%

สำหรับ แบล็คแมนวินโดว์สามารถเขียนได้คือ

$$\omega_s = 0.42 + 0.5 \cos(2\pi n / [N - 1]) + 0.08 \cos(4\pi n / [N - 1]); |n| \leq (N - 1) / 2$$

$$= 0$$

(1.104)

สเปกตรัมความถี่ไม่ได้หาค่าไว้ แต่วินโดว์นี้ให้คุณสมบัติดีกว่า แบบฮาน และแบบแฮมมิง คือ สามารถถ่ายพลังงานเข้าไปในโหลบลหลักได้มากกว่า ดังได้แสดงคุณสมบัติเพื่อการเปรียบเทียบไว้ในตาราง 2-1 จากตาราง ในวินโดว์ทุกแบบนี้จนถึงแม้  $N$  จะมีค่าเพิ่มขึ้น ค่าอัตราถูกล้นมีค่าประมาณคงที่ คือไม่แปรไปตามค่าของ  $N$

ชนิดของ วินโดว์	Main – Lobe Width	Ripple Ratio (%) N=31	Peak Amplitude Of Side Lobe (dB)	Minimum Stop Band Attenuation (dB)
สี่เหลี่ยม	$2w_s/N$	21.80	-13	-21
ฮาน	$4w_s/N$	2.67	-31	-44
แฮมมิ่ง	$4w_s/N$	0.82	-41	-53
แบล็คแมน	$6w_s/N$	0.12	-57	-74

วินโดว์	Ripple Ratio		
	N=11	N=21	N=31
สี่เหลี่ยม	22.34	21.89	21.8
ฮาน	2.62	2.67	2.67
แฮมมิ่ง	1.67	0.93	0.82
แบล็คแมน	0.08	0.12	0.12

ตารางที่ 2-1 สรุปคุณสมบัติของวินโดว์

โดยที่  $w_s$  คือความถี่ในการสุ่ม

#### 1.4.5 การออกแบบโดยทำวินโดว์

เราได้กล่าวถึงคุณสมบัติของวินโดว์แบบต่างๆ มาแล้ว คราวนี้มาดูถึงวิธีการออกแบบโดยใช้วินโดว์ในโดเมนเวลานั้น ผลตอบสนองความถี่ของตัวกรองความถี่ต่ำอุดมคติ จะเป็นดังรูป (1-33a) หรือ

$$\begin{aligned}
 H(e^{j\omega T}) &= 1; |\omega| \leq \omega_c \\
 &= 0; \omega_c \leq |\omega| \leq \omega_s / s
 \end{aligned}
 \tag{1.105}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $\omega_c$  คือค่า ความถี่ตัด และ  $\omega_s$  คือค่า ความถี่ของสัญญาณสุ่มตัวอย่าง และการหาความสัมพันธ์ของผลตอบสนองความถี่ของตัวกรอง กับ ผลตอบสนองอิมพัลส์ของตัวกรอง ทำโดยอาศัยการกระจายอนุกรมฟูรีเยอร์ คือ

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h(nT)e^{-jn\omega T} \quad (1.106)$$

และ

$$h(nT) = (1/\omega_s) \int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega T}) e^{jn\omega T} d\omega \quad (1.107)$$

เมื่อแทนค่าจากสมการ (1.105) ลงในสมการ (1.107) จะได้

$$h(nT) = (1/\omega_s) \int_{-\omega_c}^{\omega_c} e^{jn\omega T} d\omega \quad (1.108)$$

$$h(nT) = (1/n\pi T) \{\sin n\omega_c T\}$$

ซึ่งเป็นผลตอบสนองอิมพัลส์ของตัวกรองความถี่ต่ำอุดมคติ

จากสมการ (1.106) เราสามารถแปลงให้อยู่ในรูปของ ตัวแปรแซด โดยการแทนให้  $e^{j\omega T} = Z$  เพราะฉะนั้น

$$H(Z) = \sum_{n=-\infty}^{\infty} h(nT)Z^{-n} \quad (1.109)$$

ดังได้กล่าวไว้ในทางปฏิบัติ ตัวกรองนี้มีผลตอบสนองอิมพัลส์จำกัด เพราะฉะนั้นถ้าเราตัดปลายอนุกรมฟูรีเยอร์โดยให้

$$h(nT) = 0; |n| > (N-1)/2$$

หรือจาก (1.109) จะได้เป็น

$$H(Z) = h(0) + \sum_{n=1}^{(N-1)/2} \{h(-nT)Z^n + h(nT)Z^{-n}\} \quad (1.110)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เนื่องจากตัวกรองนี้ ต้องทำให้เป็นตัวกรองเหตุกภาพ ซึ่งทำได้โดยการคูณ (1.109) ด้วย พจน์  $z^{-(N-1)/2}$  และถ้าให้  $H'(Z)$  เป็นตัวกรองที่ได้ หรือ

$$H'(Z) = Z^{-(N-1)/2} \quad (1.111)$$

การคูณด้วยพจน์นี้เข้าไปจะไม่ทำให้ ผลตอบสนองแอมพลิจูด เปลี่ยนแปลงไปแต่อย่างใด แต่ทำให้ ผลตอบสนองทางเฟส ของตัวกรองมีคุณสมบัติเป็นเชิงเส้นขึ้น และจาก(1.110) และ (1.111) ได้

$$H'(Z) = Z^{-(N-1)/2} \left\{ \sum_{n=0}^{(N-1)/2} h(nT)(Z^n + Z^{-n}) \right\} \quad (1.112)$$

ในทำนองเดียวกันโดยการใช้สมการ (1.110) ซึ่งเป็นผลตอบสนองความถี่ของตัวกรอง ที่ได้จากการ ตัดปลายอนุกรมฟูรีเยอร์โดยใช้วิธีการทำวินโดว์ เราจะได้

$$H'(z) = Z^{-(N-1)/2} \left\{ \sum_{n=0}^{(N-1)/2} \omega(nT)h(nT)(Z^n + Z^{-n}) \right\} \quad (1.113)$$

$$= Z^{-(N-1)/2} \left\{ \sum_{n=0}^{(N-1)/2} (a_n/2)(z^n + z^{-n}) \right\} \quad (1.114)$$

โดยการเปรียบเทียบสมการ (1.113) กับ (1.114) และโดยให้  $a_n$  เป็นสัมประสิทธิ์ของตัวกรองไม่ ป้อนกลับเชิงเลขที่ต้องการ ซึ่งจะมีค่าคือ

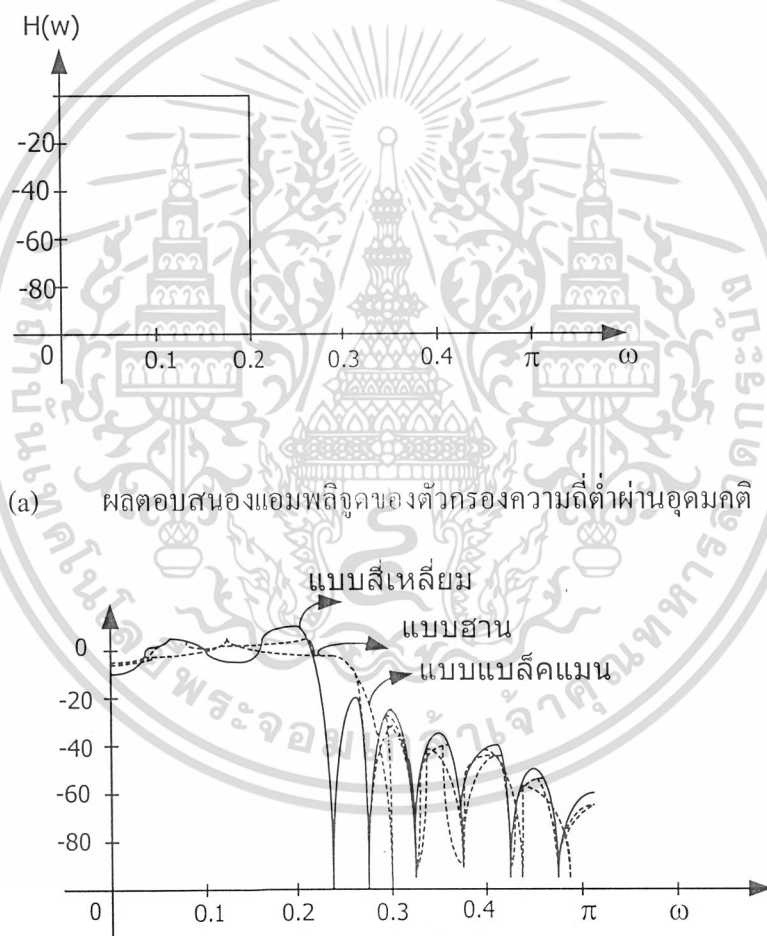
$$\begin{aligned} \text{และ} \quad a_0 &= \omega(0) \cdot h(0) \\ a_n &= 2 \cdot \omega(nT) \cdot h(nT) \end{aligned} \quad (1.115)$$

โดยที่  $\omega(nT)$  แทนผลตอบสนองอิมพัลส์ ของวินโดว์แบบต่างๆ เช่นดังสมการ (1.102) และ (1.106) ซึ่งเป็นผลตอบสนองอิมพัลส์ของวินโดว์แบบฮาน แฮมมิง และ แบล็คแมน ตามลำดับ ผลตอบสนองแอมพลิจูดสำหรับกรณี  $\omega_c = 2$  เรเดียน/วินาที,  $\omega_n = 10$  เรเดียน/วินาที และ  $N = 2$  ได้แสดงไว้ในรูป (1-33b) โดยที่ผลตอบสนองแอมพลิจูดมีค่าเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$M(\omega) = \left| \sum_{n=0}^{(N-1)/2} a_n \cos n\omega T \right| \quad (1.116)$$

จะเห็นได้ว่าการเปรียบเทียบว่า เมื่อมีการทำการถ่วงน้ำหนักโดยใช้วินโดว์ ลูกคลื่นในแถบผ่านความถี่ หรือแถบผ่าน (pass band) จะลดลง แต่ค่าลดทอนน้อยที่สุดในแถบหยุดก่อน (minimum stopband attenuation) และแถบเปลี่ยนสถานะ จะเพิ่มขึ้น



รูป 1-33 แสดงผลตอบสนองความถี่ จากการใช้วินโดส์แบบต่างๆ

รูป 1-33 แสดงผลตอบสนองความถี่ จากการใช้วินโดส์แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

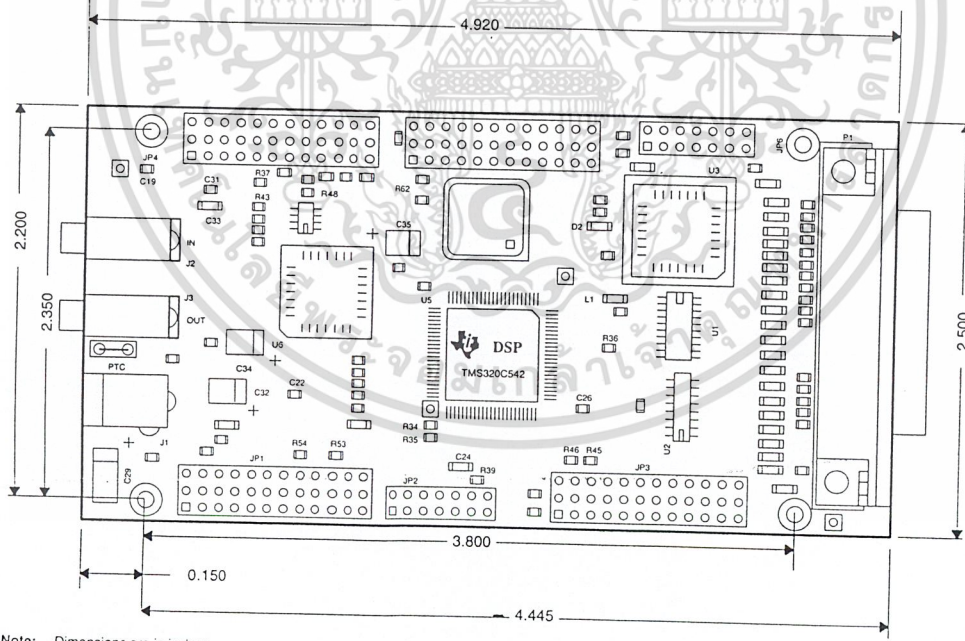
### ทฤษฎีวิชาการประมวลผลสัญญาณดิจิทัล (DSP TMS320C542)

#### 2.1 รายละเอียดของบอร์ด DSK

รูปร่างภายนอกของบอร์ด DSK อุปกรณ์โดยส่วนใหญ่เป็นแบบติดตั้งบนผิวหน้าลงแผ่นวงจรพิมพ์ จะมีเพียงตัวเก็บประจุค่าสูงและไอซี เรกูเลเตอร์ที่เป็นธรรมดา บนบอร์ด DSK จะมีชิพที่เป็นหัวใจหลักๆอยู่ 2 อย่าง คือ TMS320C542 ซึ่งเป็น DSP โปรเซสเซอร์ และชิพแปลงสัญญาณ

อนาลอกเป็นดิจิทัลและแปลงจากดิจิทัลเป็นอนาลอก เบอร์ TLC 320AC01C

ขนาดของบอร์ดอยู่ที่ 2.5 \* 4.92 นิ้ว มีคอนเน็คเตอร์ชนิด D ตัวเมียเพื่อใช้ในการเชื่อมต่อกับคอมพิวเตอร์ผ่านทางบอร์ด Serial Port 1 ชุด มีแจ็ค RCA สำหรับรับและส่งสัญญาณอนาลอก ดังรูปที่ (2-1) และรูปที่ (2-2)



Note: Dimensions are in inches.

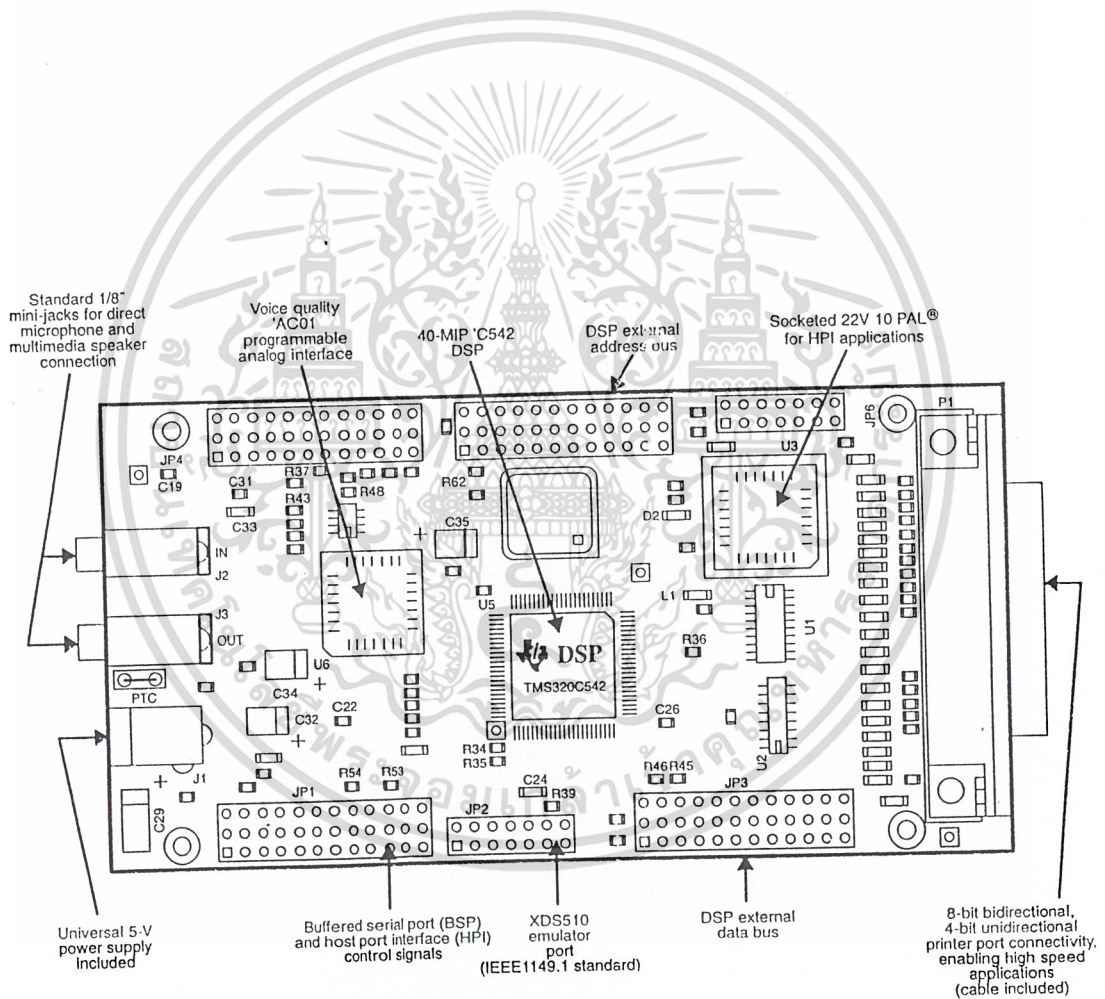
รูปที่ 2-1 ขนาดและอัตราส่วนของ TMS320C542 DSK CIRCUIT BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ระบบที่ต้องการ

ระบบที่เครื่องคอมพิวเตอร์ที่จะใช้กับชุด DSK นี้จะต้องมีคุณสมบัติดังนี้

- เป็นเครื่อง IBM PC/AT หรือเทียบเท่า
- หน่วยความจำ RAM อย่างน้อย 4M bytes
- มีฟลอปปีดิส และ ฮาร์ดดิส
- Monitor : Color VGA



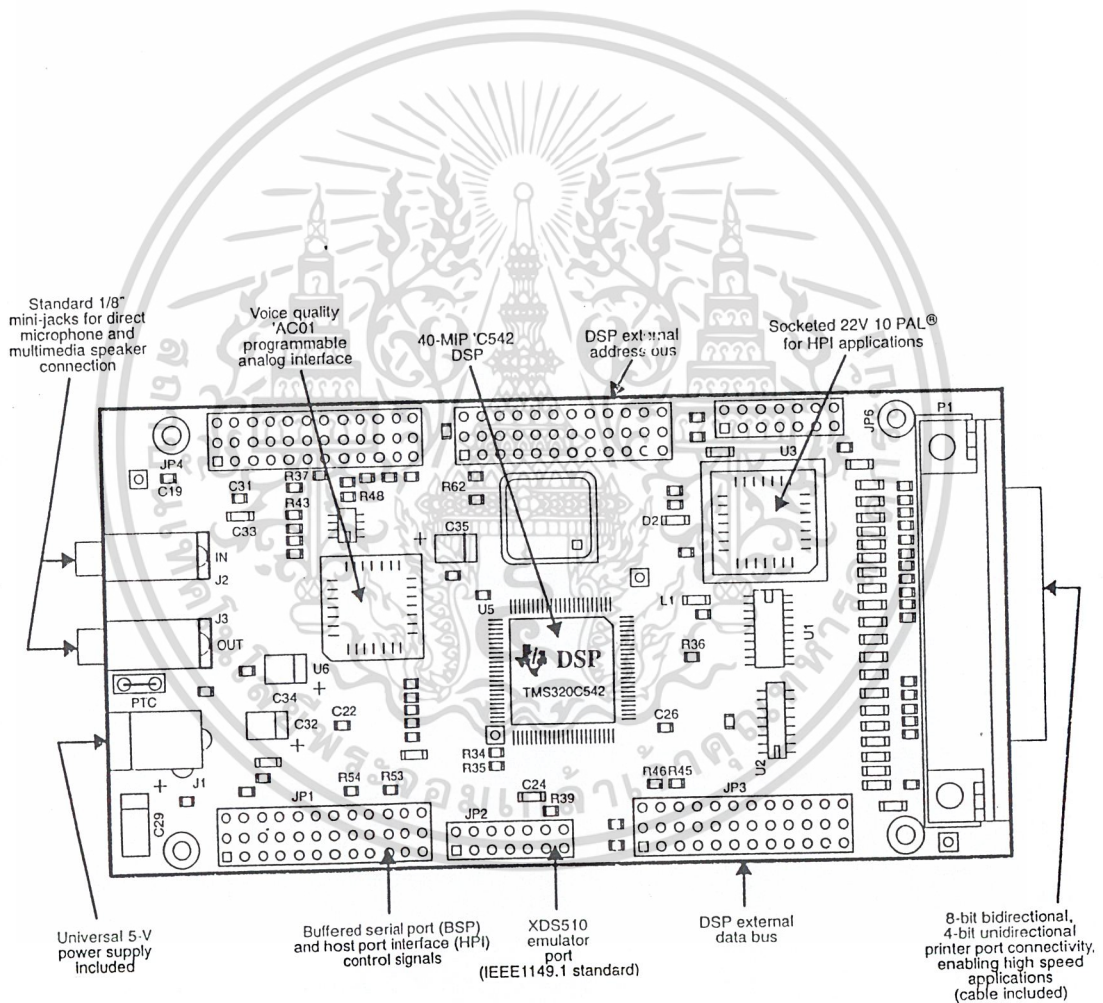
รูปที่ 2-2 อุปกรณ์โดยทั่วไปของ TMS320C542 DSK BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ระบบที่ต้องการ

ระบบที่เครื่องคอมพิวเตอร์ที่จะใช้กับชุด DSK นี้จะต้องมีคุณสมบัติดังนี้

- เป็นเครื่อง IBM PC/AT หรือเทียบเท่า
- หน่วยความจำ RAM อย่างน้อย 4M bytes
- มีฟลอปปีดิส และ ฮาร์ดดิส
- Monitor : Color VGA



รูปที่ 2-2 อุปกรณ์โดยทั่วไปของ TMS320C542 DSP BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 วงจรสมบูรณ

บอร์ด DSK มีวงจรที่สมบูรณในการเชื่อมตอระหวางชิพ DSK กับอุปกรณ์ตอร่วมอื่นๆ ในสวนลางเป็นภาคจ่ายไฟของบอร์ด DSK หัวใจสำคัญของบอร์ดอยูที่ U5 ชิพ DSP เบอร์ TMS320C542 เป็นไมโครโปรเซสเซอร์ตัวหนึ่งที่ใชงานดานการประมวลผลสัญญาณดิจิทัลโดยเฉพาะ ทำหน้าที่ประมวลผลข้อมูลทั้งหมดบนบอร์ด DSK นี้ โดยตองทำงานร่วมนกับซอฟต์แวร์

U5 ทำงานได้โดยอาศัยสัญญาณนาฬิกาจากโมดูลกำเนิดสัญญาณนาฬิกา 40 เมกะเฮิร์ต U4 SQ3300 โดยมี X1 เซรามิกเรโซเนเตอร์ควบคุมความถี่ของสัญญาณนาฬิกาให้มีความถี่ตรงอีกชั้นหนึ่ง

U6 ชิพ ADC/DAC เบอร์ TLC320AC01C ของเท็กซัสอินสตรูเมนต์ ทำหน้าที่รับสัญญาณอนาลอกทางดานอินพุทที่จ่ายเข้ามาทาง J2 แล้วแปลงเป็นข้อมูลทางดิจิทัลส่งไปประมวลผลตอที่ U4 และเมื่อตองการส่งสัญญาณออก U4 จะส่งข้อมูลดิจิทัลมายัง U3 เพื่อทำการเปลี่ยนแปลงเป็นสัญญาณอนาลอกจ่ายออกไปทาง J3 ความละเอียดของการแปลงสัญญาณดิจิทัลจะมีขนาด 14 บิต สามารถกำหนดอัตราแซมปลิงได้สูงสุด 19,200 แซมปลิงใน 1 วินาที

สวนมากภาคจ่ายไฟบอร์ด DSK ตองการไฟสลับ 5 โวลท์ จากภายนอกจ่ายเข้ามาทาง J1 และมีไดโอด D1, D7 กับคาปาซิเตอร์ C1, C2 = 25 โวลท์ 1000 ไมโครฟารัด

### 2.4 เกี่ยวกับชิพ TMS320C542

TMS320C542 เป็นหัวใจหลักของบอร์ด DSK เป็นผลงานของ เท็กซัสอินสตรูเมนต์ เป็นไมโครโปรเซสเซอร์ตัวหนึ่งที่ถูกออกแบบมาให้ใชงานดาน DSP โดยเฉพาะ มีคาบเวลาของไซเคิลของคำสั่งเป็น 100 นาโนวินาที ในการทำงานคำสั่งแต่ละตัวจะใช้เวลาหนึ่งไซเคิล ทำให้สามารถประมวลผลได้ถึง 10 ล้านคำสั่งตอนาที จึงส่งผลให้ชิพนี้สามารถทำการประมวลผลแบบเวลาจริง (real time) ได้ทัน

ในตัว TMS320C542 มีหน่วยความจำความเร็วสูงสุดขนาด 192 กิโลเวิร์ด โดยที่แต่ละเวิร์ดจะมีขนาด 16 บิต และมีรอมภายในขนาด 128 เวิร์ด สามารถอ้างอิงหน่วยความจำภายนอกได้ถึง 128 กิโลเวิร์ด แบ่งเป็นหน่วยความจำข้อมูลและหน่วยความจำโปรแกรมอย่างละ 64 กิโลเวิร์ด ALU (Arithmetic Logic Unit) และแอกคิวมูลเตอร์ ขนาด 40 บิตที่สามารถคูณเลขขนาด 40 บิต ได้ภายใน 1 แมกซีนไซเคิลทั้งนี้เนื่องจากตัว ALU มีวงจรคูณแบบ  $16 \times 16$  บิตฮาร์ดไวร์ นอกจากนี้ยังมีตัวตั้งเวลาและส่วนกำเนิดสัญญาณนาฬิกาอยูภายในชิพ สวนของการอินเตอร์เฟสเป็นแบบขนาน 16 บิต แบ่งเป็นอินพุท 16 แชนแนล เอาท์พุท 16 แชนแนล วงจรติดตอหน่วยความจำร่วมน (Global Data Memory Interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใชงานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ดานการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตองอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

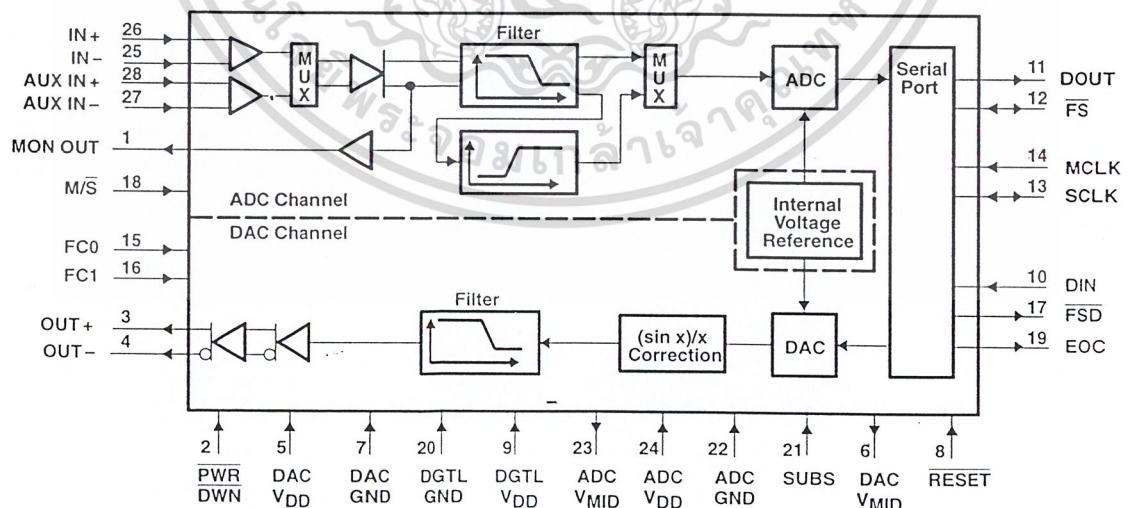
ตัว TMS320C542 สามารถต่อใช้งานเพียงตัวเดียวหรือจะต่อใช้งานขนานกันหลายตัวที่เรียกว่าเป็นระบบ multi-processor system ก็ได้

ในการใช้งานกับหน่วยความจำภายนอกไม่ว่าจะเป็นอีพรอมหรือรอมตัวชิพจะทำงานได้เข้ามาจนถึงได้เน้นให้ใช้หน่วยความจำแบบภายในชิพเป็นที่เก็บโปรแกรมแทน ในการใช้งานให้ดาวน์โหลดโปรแกรมจากหน่วยความจำภายนอกมาเก็บในชิพแล้วจึงรันโปรแกรมทำงานโดย TMS320C542 ได้เพิ่ม 'Block Transfer' ที่ใช้ในการโหลดข้อมูลจากหน่วยความจำภายนอกมาที่ละบล็อค แล้วมาเก็บไว้ในแรมของชิพเอง ด้วยวิธีการนี้จะช่วยให้การทำงานเร็วมากขึ้น

การอ้างแอดเดรสของ TMS320C542 มีด้วยกัน 3 โหมด คือ โดยตรง (direct), ทางอ้อม (indirect), และทันทีทันใด (immediate) นอกจากนี้ยังมีรีจิสเตอร์เสริมอีก 8 ตัว

## 2.5 TLC320AC01C

IC เบอร์ TLC320AC01 จะเป็น analog interface circuit (AIC) เป็น IC ที่ใช้ประมวลผลในย่านความถี่เสียง เป็นตัวเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอลและแปลงจากดิจิตอลเป็นอนาล็อกรวมอยู่ใน IC ตัวเดียวซึ่งภายในจะประกอบด้วย Bandpass Filter, Switch Capacitor, 14 bits ADC และ DAC วงจรกรองความถี่ต่ำทางด้าน OUTPUT



รูป 2-3 แสดงลักษณะของ IC # TLC320AC01C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หน้าที่หลักของ TLC320AC01

1. เปลี่ยนสัญญาณอนาล็อกเป็นความถี่ดิจิทัล
2. ควบคุมสัญญาณที่จะส่งไปประมวลผลยังตัวประมวลผลแบบดิจิทัลหรือ Microprocessor
3. เปลี่ยนสัญญาณดิจิทัลกลับไปเป็นอนาล็อก

TLC320AC01 ประกอบด้วยการประมวลผลสัญญาณสองช่อง ส่วนของ ADC และส่วนของ DAC และส่วนของการควบคุมดิจิทัล โดยแต่ละส่วนจะแยกออกจากกัน ข้อมูลที่รับเข้ามาจากส่วนของ ADC และข้อมูลที่ส่งมาจาก DAC จะเกิดขึ้นในเวลาเดียวกัน

การใช้งานของ TLC320AC01 สามารถใช้เป็นโมเด็ม การประมวลผลสัญญาณเสียง ใช้ในการต่อสัญญาณอนาล็อกสำหรับ DSPs การควบคุมกระบวนการทางอุตสาหกรรม การวิเคราะห์ Spectrum

### 2.6 ติดตั้งซอฟต์แวร์

ก่อนใช้งานต้องทำการติดตั้งซอฟต์แวร์เสียก่อน โดยในชุด DSK นี้จะมีแผ่นดิสก์ 2 ชุด ชุดแรกจะเป็นแผ่น USER SOFTWARE อันเป็นแผ่นที่ใช้งานจริง ส่วนอีกชุดหนึ่งเป็น TMS320 Development Flow and Programmer's Interface ซึ่งบรรจุแนะนำวิธีการใช้งาน และพัฒนา TMS320C542

การติดตั้งจะติดตั้งทั้ง 2 แผ่นลงในฮาร์ดดิสก์ก็ได้ หรือจะติดตั้งเฉพาะแผ่นใช้งานก็ได้ วิธีการง่ายๆ เริ่มจากสร้างไดเรกทอรีของ DSK เสียก่อนให้ชื่อว่า DSKTOOL จากนั้นก็อปไฟล์ทุกไดเรกทอรีย่อยในแผ่น USE SOFTWARE ลงในฮาร์ดดิสก์ เพียงเท่านี้ก็ใช้งานได้แล้ว

ส่วนแผ่นแนะนำนั้น จะมีลักษณะเหมือนโปรแกรมฟรีเซนต์ชัน คือเป็นกราฟฟิกที่สวยงามแนะนำขั้นตอนต่างๆอย่างละเอียด ถ้าหากมีที่ว่างในฮาร์ดดิสก์ก็น่าจะติดตั้งลงไปด้วย

### 2.7 เริ่มต้นการใช้งาน

หลังจากติดตั้งโปรแกรมเรียบร้อยแล้ว ต่อสายระหว่างบอร์ด DSK กับคอมพิวเตอร์ จ่ายแรงดันไฟสลับ 5 โวลต์ให้แก่บอร์ด DSK สังเกตดูหลอด LED (D1) ถ้าหลอดสว่างแสดงว่าบอร์ด DSK ทำงานแล้วจากนั้นทำการทดลองรันโปรแกรม จากขั้นตอนการพัฒนาโปรแกรม ต้องเริ่มจากการเขียนโปรแกรมภาษาแอสเซมบลี แล้วทำการคอมไพล์โดยแอสเซมเบลอร์ จากนั้นจึงนำโปรแกรมมารัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของซอฟต์แวร์ทั้งหมด สามารถพิมพ์ออกมาอ่านรายละเอียดได้โดยพิมพ์ไฟล์ที่มีนามสกุล .DOC ทั้งหมดหรือถ้าต้องการศึกษาโปรแกรมภาษาแอสเซมบลีให้พิมพ์ไฟล์ นามสกุล .ASM

ชิพ TMS320C542 ของบริษัท Texas Instruments จะทำการประมวลผลสัญญาณแบบจำนวนเต็ม (Fixed-point) ซึ่งจะต้องทำการแปลงจำนวนทศนิยมมาเป็นจำนวนเต็ม ก่อนจะนำไปประมวลผลซึ่งชิพรุ่นใหม่นั้นสามารถประมวลผลสัญญาณแบบทศนิยม(Float-point) ได้เลยโดยไม่ต้องทำการแปลงเป็นจำนวนเต็ม

ชิพ DSP ตระกูล Fixed-Point ของบริษัท Texas Instruments มีการพัฒนามาหลายรุ่นซึ่งทั้งหมดจะมีชุดคำสั่งซึ่งสามารถใช้ร่วมกันได้โดยคำสั่งของรุ่นเก่าจะยังคงใช้ได้กับ ชิพ ตัวใหม่ ซึ่งสามารถทำให้ผู้ศึกษาเพิ่มเติม ไม่มากก็จะทำให้สามารถใช้ ชิพ รุ่นใหม่ๆ ได้ไม่ยาก และหนึ่งในนั้นคือ ชิพ ในตระกูล C5x จะมีหลายรุ่นซึ่งจะแตกต่างกันในส่วนของ Memory ที่บรรจุมาภายในเพื่อให้ผู้ใช้เลือกนำไปใช้งานในลักษณะต่างๆกันแต่ในส่วนหลักๆของการทำงานภายในจะเหมือนกัน โดยโครงการนี้จะนำชิพรุ่น C542 มาใช้งานแต่ในส่วนของทฤษฎีในบทนี้ จะเรียกรวมๆว่า C54x

## 2.8 การใช้งานทั่วไป

Automotive	Consumer	Control
Adaptive ride control	Digital radios/TVs	Disk drive control
Antiskid brakes	Educational toys	Engine control
Cellular telephones	Music synthesizers	Laser printer control
Digital radios	Power tools	Motor control
Engine Control	Rader detectors	Robotics control
Global positioning	Solid-state answering machines	Servo control
Navigation		
Vibration analysis		
Voice commands		
General-Purpose	Graphics/Imaging	Industrial
Adaptive filtering	3-D rotation	Numeric control
Convolution	Animation/digital map	Power-line monitoring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Correlation	Homomorphic processing	Robotics
Digital filtering	Pattern recognition	Security access
Fast Fourier transforms	Image enhancement	
Hilbert transforms	Image compression/transmission	
Waveform generation	Robot vision	
Windowing	Workstations	

<b>Instrumentation</b>	<b>Medical</b>	<b>Military</b>
Digital filtering	Diagnostic equipment	Image processing
Function generation	Fetal monitoring	Missile guidance
Pattern matching	Hearing aids	Navigation
Phase-locked loops	Patient monitoring	Radar processing
Seismic processing	Prosthetics	Radio freq. Modems
Spectrum analysis	Ultrasound equipment	Secure Commu
Transient analysis		Sonar processing
<b>Telecommunications</b>		<b>Voice/Speech</b>
1200 to 19200 bps modems	DTMF encoding/decoding	Speech enhancement
Adaptive equalizers	Echo cancellation	Speech recognition
ADPCM transcoders	Fax	Speech synthesis
Cellular telephones	Line repeaters	Speaker verification
Channel multiplexing	Speaker phones	Speech vocoding
Data encryption	Spread spectrum Communications	Voice mail
Digital PBXs	Video conferencing	Text-to-Speech
Digital speech interpolation(DSI) x.25 Packet Switching		
Personal digital assistants(PDA)	Personal Communications systems(PCS)	

### ตารางที่ 2-1 แสดงรายการของการใช้งาน C54X ในด้านต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยลักษณะการทำงานแบบ Real Time และความสามารถในการใช้งานที่หลากหลายทำให้ชิพ รุ่น C54x ใช้งานได้อย่างกว้างขวางในการแก้ปัญหาต่างๆที่มีการทำงานแบบ Signal – Processing เช่น การถอดรหัส หรือการกรองสัญญาณไม่เพียงเท่านั้นในรุ่น C54x ยังออกแบบมาเพื่อรองรับการทำงานที่มีการทำงานหลายอย่างในเวลาเดียวกันด้วย

ในรุ่น C54x ยังประกอบไปด้วย C541, C542 ถึง C58 ซึ่งทั้งสามตัวเป็นสิ่งประดิษฐ์ประเภท Static CMOS ที่มีสถาปัตยกรรมภายในอยู่บนพื้นฐานของรุ่น C25 ด้วยการผสมผสานกันของสถาปัตยกรรมขั้นสูงที่มีการแยกแยะระหว่าง Program – Bus และ Memory – Bus ด้วยการเพิ่มอุปกรณ์สนับสนุนลงบนชิพ การสร้าง On – chip Memory ที่มีขนาดใหญ่ขึ้นและคำสั่งพิเศษที่มีความสามารถสูงขึ้น ซึ่งเป็นพื้นฐานให้การทำงานที่มีความยืดหยุ่นและความเร็วสูงขึ้น

TMS320C542 ถูกออกแบบให้มีความเร็วในการทำงานมากกว่า 28 MIPS ในอนาคต คาดว่า จะมีการพัฒนาขึ้นอีก และสามารถตอบสนองความต้องการของตลาดได้มากขึ้น ในรุ่น C54x ออกแบบให้มีประโยชน์ใช้สอยได้ดังนี้

- ใช้หลักการในการออกแบบที่มีความเร็วสูงสุดและใช้งานง่าย
- ใช้เทคโนโลยีขั้นสูงเพื่อเพิ่มความเร็วในการทำงาน
- มีความ Compatible (เข้ากันได้) กับรุ่นก่อนๆไม่ว่าจะเป็น C1x, C2x ทำให้สามารถทำการ Upgrade ได้
- มีการปรับปรุงชุดคำสั่งเพื่อความเร็วในการทำงานแบบ Algorithm (ขั้นตอนวิธี) ที่มีประสิทธิภาพมากขึ้นและเพิ่มความสามารถในการทำงานกับภาษาชั้นสูง
- ใช้เทคนิคแบบใหม่ในการออกแบบช่วยให้ประหยัดกำลังงานในการทำงานและทำให้การกระจายของคลื่นวิทยุไปสู่ภายนอกน้อยลง
- ใช้สถาปัตยกรรมขั้นสูงเพื่อเพิ่มความเร็วในการทำงานและสามารถทำงานได้หลายๆอย่างในเวลาเดียวกัน

ตารางที่ 2-2 แสดงภาพรวมของความสามารถในรุ่น C54x โดยแสดงถึงความจุ ของ On – Chip RAM และ ROM จำนวนของ Serial และ Parallel I/O port , Execution Time ของแต่ละ Machine Cycle และชนิดของ Package รวมทั้งจำนวนขาทั้งหมดของ ชิพ

TMS320 Device	On-chip memory			I/O Port		Cycle Time (ns)	Package type
	RAM		ROM	Serial	Parallel		
	Data	Data+Prog	Prog				
TMS320c541	1K	9K	2K	2	64K	50/35	142-pin plastic
TMS320c542	1K	1K	8K	2	64K	50/35	142-pin plastic
TMS320c543	1K	3K	16K	2	64K	50/35	142-pin plastic

### ตารางที่ 2-2 แสดงภาพรวมของความสามารถในรุ่น C54x

รายละเอียดของรุ่น C54x แสดงไว้ข้างล่างนี้ โดยแสดงลักษณะที่สำคัญของแต่ละเบอร์ โดยชื่อของ ชิพ จะแสดงอยู่ในวงเล็บ

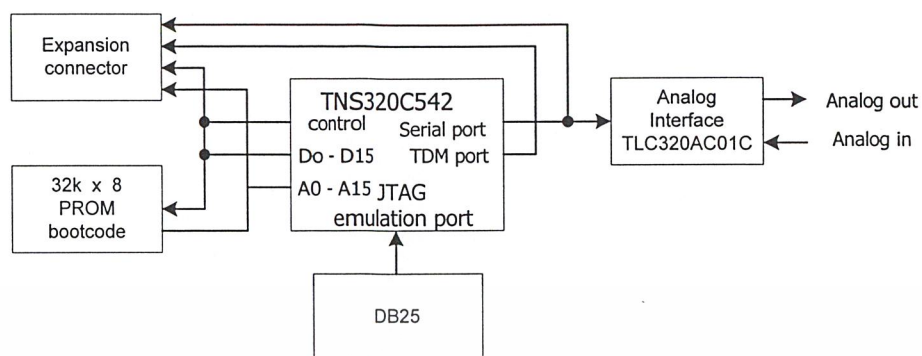
- ใช้เวลาทำงานตามคำสั่งแบบ Fixed-Point ด้วยความเร็ว 35-50 ns
- มีคำสั่งที่ Compatible กับรุ่น C1x,C2x
- RAM – Based Memory Operation (C50)
- 9k×16 bit on-chip program/data RAM (C50)
- 2k×16 bit on-chip boot ROM (C50)
- 1056k×16 bit dual-access on-chip data RAM
- 32 bit ALU,32 bit ACC(accumulator) และ 32 bit ACCB(Accumulator Buffer)
- 16 bit PLU (Parallel Logic Unit)
- 16×16 bit parallel multiplier พร้อมทั้ง 32 bit product capability
- มีความสามารถในการคูณด้วยรอบคำสั่งเดียว
- 8 Auxiliary Register
- 11 Context – switch register (Shadow register) เพื่อใช้ในการ Interrupt
- 0-16 bit shifter และ 64 bit incremental

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2 circular buffer สำหรับการอ้าง address แบบ circular มีคำสั่ง single – instruction repeat and block repeat
- มีคำสั่ง move แบบ block เพื่อการจัดการข้อมูลดีขึ้น
- Full-Duplex Synchronous serial port เพื่อการสื่อสารระหว่าง C5x กับ อุปกรณ์ภายนอก
- มี TMD (Time Division Multiple Access) Serial port
- 64k parallel i/o port และมี 16 port ที่สามารถ mapped ไปยัง memory
- 16 wait state generator ควบคุมการทำงานด้วย software
- เชื่อมต่อกับอุปกรณ์ภายนอกแบบ DMA ได้สะดวก
- ภายในประกอบด้วย 4-deep pipeline สำหรับ delayed branch
- index-addressing mode
- Bit-Reverse index – addressing mode สำหรับ radix-2 FFT
- มีความสามารถหารได้ภายใน clock ลูกเดียว
- มี clock generator ภายในชิพ
- JTAG boundary scans logic (IEEE standard, 1149.1)
- 5 volt static CMOS พร้อมด้วย power – down mode 2 mode

## 2.9 ส่วนสำคัญของ CPU

การพัฒนาขึ้นมาเป็นรุ่น C5x ยังใช้ Source Code ที่ Compatible กับรุ่น C1x และ C2x เมื่อมีการพัฒนา Program จึงทำได้ง่าย ในการพัฒนารุ่นต่างๆของ C5x นี้ยังมีการสร้าง 32 bit accumulator buffer (ACCB) เพิ่มเติมขึ้นมาอีกด้วย เพื่อสนับสนุนการทำงานของ ACC ใน Control function แบบใหม่มี PLU ซึ่งเป็นอิสระในการทำงาน ดังนั้นจึงสามารถทำงานในแบบสมการ Boolean ได้ดีขึ้น และมีการบริการการ interrupt ด้วย IRS จึงทำงานได้เร็วขึ้น การจัดการข้อมูลได้มีการปรับปรุงโดยมีการใช้คำสั่ง block move และมีคำสั่ง memory-mapped ในรุ่น C5x มี register สำหรับทำการ mapped memory ทั้งหมด 28 ตัว และมี register สำหรับทำการ mapped I/O ทั้งหมด 16 ตัว



รูปที่ 2-4 DSK Block Diagram

### 2.9.1 On – chip ROM

C50 มี  $2K \times 16$  bit maskable programmable ROM หน่วยความจำนี้ใช้สำหรับการ Boot เครื่องในตอนเริ่มต้น ในกรณีที่ ROM ภายนอกมีความเร็วต่ำ โดยการเลือก ROM สามารถเลือกได้ ในขณะที่ Reset ด้วยขา MP /  $\overline{MC}$  ในครั้งแรกที่ทำการ Boot เครื่อง โดย Boot ROM สามารถที่จะทำการยกเลิกการใช้งานด้วยการควบคุม bit ใน PMST Register

### 2.9.2 On – chip data RAM

อุปกรณ์ทั้งหมดในรุ่น C5x จะมี RAM ขนาด  $1056 \times 16$  bit โดย RAM ในส่วนนี้มีความสามารถในการทำงานได้ถึง 2 ครั้งใน 1 machine cycle (dual – access RAM) โดยใช้สำหรับเก็บข้อมูล โดยเฉพาะแต่ก็สามารถเก็บ Program ในพื้นที่ส่วนนี้ได้ถ้าต้องการ

### 2.9.3 On – chip Program / Data RAM

C50 มี  $9K \times 16$  bit On – chip RAM ซึ่งพื้นที่หน่วยความจำส่วนนี้สามารถจัดโครงสร้างด้วย software ให้เป็นพื้นที่ Program หรือ data หรือ เป็นทั้งพื้นที่ program และ data ก็ได้

### 2.9.4 On – chip Memory Security

C5x มีระบบป้องกันข้อมูลที่บรรจุอยู่ใน On – chip Memory เมื่อมีการ Set ค่าของ Register ต่างๆ ที่เกี่ยวข้องกับการแบ่งหน่วยความจำแล้วจะไม่มีคำสั่งจากภายนอกเข้าไปใช้งานในพื้นที่หน่วยความจำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.5 Address – Mapped Software Wait – State Generator

การกำหนด Wait – State เมื่อติดต่อกับหน่วยความจำหรืออุปกรณ์ I/O สามารถกระทำได้ โดยการ Set ค่า Register ที่เกี่ยวข้องกับ Software โดยไม่จำเป็นที่ต้องต่อ Hardware เพิ่มเติม ในส่วนนี้ประกอบไปด้วยวงจร Wait – State Generators จำนวน 16 ชุดด้วยกัน เพื่อทำการสร้าง Wait – State 0,1,2,3 หรือ 7 ในการติดต่อกับหน่วยความจำภายนอกและอุปกรณ์ I/O จะสามารถกำหนดขอบเขตเพื่อที่จะทำการ Map กับ Wait-State ได้ด้วยขนาด 16K – word ต่อ 1 วงจร

### 2.9.6 Parallel I/O Ports

ชิพ ในรุ่น C5x จะมี I/O port ทั้งหมด 64 K-port โดยที่ 16 port ใดๆ ใน 64 K-port สามารถ Map ลงไปยัง Data Memory ได้ port เหล่านี้สามารถกำหนด Address ได้ด้วยคำสั่ง IN และ OUT และ I/O port ที่ถูก Map ไปยัง Data Memory สามารถทำงานได้ด้วยคำสั่งอ่านและเขียนหน่วยความจำแบบปกติได้

### 2.9.7 I/O Ports

ชิพ ในรุ่น C5x ประกอบไปด้วย Serial port ความเร็วสูงจำนวน 2 port ซึ่งสามารถทำงานได้ด้วยความเร็วถึง  $\frac{1}{4}$  ของ Machine Cycle (CLKOUT1) โดยที่ Serial port ตัวแรกทำงานแบบ Synchronous Full – Duplex และมี Buffer ภายในสำหรับการรับและการส่งข้อมูล ส่วน Serial port อีกตัวหนึ่งเป็นแบบ Full – Duplex ที่สามารถกำหนดการทำงานให้เป็นแบบ Synchronous หรือ TDM (Time Division Multiple Access) ก็ได้ซึ่ง TDM จะใช้งานทั่วไปในระบบ Multi – Processor

### 2.9.8 Hardware Timer

ชิพ ในรุ่น C5x จะมีวงจร Timing ขนาด 16 bit พร้อมด้วย 4 bit Pre-scalar ความถี่ Clock จะมีอัตราตั้งแต่  $\frac{1}{2}$  ถึง  $\frac{1}{32}$  ของ Machine Cycle ขึ้นอยู่กับการ Program Divide – Down Ratio วงจร Timer สามารถควบคุมให้หยุด, เริ่มต้นใหม่, Reset หรือ Disable ด้วย Status bit พิเศษที่เป็นวงจร Timer

Registers	Description
SPC	Serial port control register
DXR	Data transmit register
DRR	Data transmit register
XSR	Transmit shift register
RSR	Receive shift register

ตารางที่ 2-3 Serial Port Register

### 2.9.9 User – Maskable Interrupts

ชิพ ในรุ่น C5x มีสาย External – Interrupt จำนวน 4 เส้น ดังนั้นอุปกรณ์ภายนอกแต่ละตัวสามารถควบคุม TMS320 ได้ และอุปกรณ์ภายในจะสามารถควบคุมการ Interrupt ได้ 5 ช่อง จาก Timer 1 ช่อง และ Serial อีก 4 ช่องด้วย

### 2.9.10 JTAG Scanning Logic

ชิพ ในรุ่น C5x มีความสามารถในการตรวจสอบการทำงานอย่างต่อเนื่องเกี่ยวกับการทำงานของอุปกรณ์ภายนอกที่เชื่อมต่ออยู่และการทำงานภายใน CPU เองได้อย่างดี โดยที่ JTAG จะเชื่อมต่ออยู่กับวงจร Scanning Logic อื่นๆ ภายใน CPU ชิพ ในตระกูล C5x จึงสามารถทำงานเป็นวงจรตรวจสอบโดยอ่านค่าจากขาสัญญาณ JTAG Serial Scan และขาสัญญาณ Emulation – Dedicated

### 2.9.11 Packages

ชิพ ในรุ่น C5x มีลักษณะ Packages เป็น QFP (Quad Flat Pack Package) 132 pin โดยมีจุดประสงค์ให้กินพื้นที่ในบอร์ดให้น้อยที่สุด โดยใช้พื้นฐานเดียวกันกับการออกแบบ ชิพ C25

## 2.10 สถาปัตยกรรมภายใน

โครงสร้างสถาปัตยกรรมภายในของ TMS320 DSP ประกอบไปด้วยส่วนพื้นฐาน 3 ส่วนด้วยกัน

- Central Processing Unit (CPU)
- Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## - Peripheral Interfacing Circuit

ซึ่งในส่วนนี้กล่าวถึง สถาปัตยกรรมและการทำงานของ CPU C51 ที่มีความสามารถในการทำงานทางคณิตศาสตร์ด้วยความเร็วสูง ด้วยสถาปัตยกรรมการทำงานแบบขนานของมัน

### โครงสร้างสถาปัตยกรรมโดยรวม

ชิพในรุ่น C5x เป็นชิพที่มีความสามารถสูงในการทำงานด้าน Digital Signal Processing โดยออกแบบให้มีความคล้ายคลึงกับ ชิพ ในรุ่น C25 ใช้สถาปัตยกรรมการทำงานแบบแยก Memory Bus ระหว่าง Program Bus และ Data Bus ให้มีการทำงานแยกกันต่างหาก เพื่อความเร็วสูงสุดในการทำงาน โดยมีคำสั่ง Transfer รองรับการทำงานทั้งสองส่วน

C5x สามารถใช้งานในการทำ 2 complement โดยการใส่ 32 bit ALU และ Accumulator ALU มีลักษณะเป็น General Purpose ที่ใช้ขนาด 16 bit word จาก Data memory หรืออ้างแบบ Immediate จากคำสั่งหรือเก็บผลลัพธ์จากการคูณในขนาด 32 bit ด้วยการเพิ่มเติมการทำงาน ALU ทำให้ CPU สามารถทำคำสั่ง Boolean ได้ด้วย

Accumulator จะมีหน้าที่เป็น เอาท์พุท ของ ALU และในขณะเดียวกันมันก็เป็นทั้งอินพุทของ ALU ด้วย Accumulator มีขนาด 32 bit โดยแบ่งเป็น 16 bit บน ( bit 31 ถึง bit 16) และ 16 bit ล่าง ( bit 15 ถึง bit 0 ) ในชุดคำสั่งของ C5x ก็ยังมีที่เก็บข้อมูลชั่วคราวสำหรับ Accumulator ด้วยซึ่งมีขนาด 32 bit เท่ากัน มีชื่อเรียกว่า ACCB ในการเพิ่มความสามารถของ ALU มีการเพิ่มหน่วย PLU ขึ้นมาซึ่งสามารถทำคำสั่งเกี่ยวกับ Logic ได้ โดยการทำงานเกี่ยวกับ bit จะไม่ส่งผลกระทบต่อ Accumulator ALU สามารถใช้ในการโอนย้าย bit ทำให้สามารถควบคุม bit และแก้ไข bit ได้อย่างรวดเร็ว เพื่อใช้ในการ set bit, clear bit, และ test bit สำหรับ status register ต่างๆ

การคูณเลข 16 bit  $\times$  2 จำนวนแบบ 2-Complement โดยให้ผลลัพธ์ออกมาเป็นเลข 32 bit สามารถกระทำได้ภายใน 1 รอบคำสั่ง ส่วนของการคูณประกอบไปด้วย 3 ส่วนสำคัญคือ Multiplier Array ,PREG (Product Register) และ TREGO (Temporary Register) โดยที่ TERGO มีขนาด 16 bit และ PREG มีเลข 32 bit ในการคูณนั้นค่าของตัวคูณอาจจะนำมาจาก DATA Memory, Program Memory เมื่อใช้คำสั่ง MAC/MACD/MADS/MADD หรืออาจเป็นค่า Immediate ภายในคำสั่ง MPY# ก็ได้ ด้วยการคูณที่มีความสูงนี้ ทำให้ DSP มีการทำงานที่มีประสิทธิภาพสูงในการทำงาน เช่นการ Filter หรือ Correlation

ในส่วนของการ Scaling Shifter ภายใน C5x มีลักษณะที่เชื่อมต่อระหว่าง Data Bus และ ALU โดยมีอินพุท ขนาด 16 bit ทางด้าน Data Bus และ เอาท์พุท ขนาด 32 bit ทางด้าน ALU เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scaling Shifter มีความสามารถในการเลื่อน bit ไปทางซ้ายได้ตั้งแต่ 0-16 bit จาก Input Data ที่เข้ามา ตามค่าที่กำหนดภายใน TREG1 โดยค่าทางหลังจากการทำการเลื่อน bit แล้วจะถูกแทนที่ด้วยค่าศูนย์ ส่วนค่า MSB สามารถให้เป็น 0 หรือ Sign ก็ได้ ขึ้นอยู่กับสถานะของ Sign-Extension Mode bit (SXM) ของ status Register ST1 การเพิ่มความสามารถในการ Shift นี้ทำให้ Processor มีความสามารถเพิ่มขึ้นในด้าน Numerical – Scaling , Bit – Extraction , Extended – Arithmetic และ Overflow - Prevention

C5x มี Hardware Stack 8 ระดับ เพื่อป้องกันข้อมูลขณะ Interrupt และขณะทำคำสั่ง Call โดยขณะทำการ Interrupt Register หลักอันได้แก่ ACC, ACCB, ARCR, INDEX, PMSTPREG, STO และ ST1 TREGS จะถูก Push ลงบน Stack ระดับแรกสุด และจะถูก POP ขึ้นมาเมื่อมีการ Interrupt Return หรือกลับจากการ Call

## 2.11 Memory

หน่วยความจำทั้งหมดของ ชิพ ในรุ่น C5x มีขนาดสูงสุด 224 k 16 – bit word พื้นที่ในหน่วยความจำแบ่งเป็น 4 ส่วนดังนี้คือ พื้นที่ Program พื้นที่ Program ขนาด 64 K – words, Local Data ขนาด 64 K – words, Global Data ขนาด 32 K-words และ I/O Port อีก 64 K ในสถาปัตยกรรมแบบขนาน จะทำให้อุปกรณ์แยกการทำงานได้ 3 ส่วน ได้แก่ Fetch, Read และ Write ซึ่ง โครงสร้างและการทำงานอธิบายได้ดังนี้

### 2.11.1 Memory Space Chip

ใน C5x มีสถาปัตยกรรมหลายแบบ Memory Space ซึ่งสามารถทำงานใน Bus แบบขนาน ทั้งหมด 3 Bus สิ่งนี้เองที่ทำให้ CPU สามารถทำงานเกี่ยวกับ Program และ Data ได้ในเวลาเดียวกัน Parallel Bus ทั้งสาม ได้แก่ PAB ( Program Read/Write Bus), DAB1 (Data Read Bus) และ DAB2 (Data Wire Bus) แต่ละ Bus ทำงานในพื้นที่หน่วยความจำที่แตกต่างกัน หน่วยความจำใน C5x แบ่งออกเป็นสี่ส่วนที่แยกเป็นอิสระจากกันคือ ส่วน Program , Local Data และ Input /Output Port ซึ่งสามารถ Mapped ให้อยู่ภายใน ชิพ หรืออยู่ข้างนอก ชิพ ก็ได้

Program Space จะเก็บค่าของคำสั่งที่จะทำการ Executed , Local Data Space เก็บค่าของ Data ที่ถูกใช้โดยคำสั่ง Global Data Space มีความสามารถในการ Share ข้อมูลระหว่าง Processor หรือสามารถจองไว้ใช้เพื่อเป็น Data Space ก็ได้ ส่วน I/O Space จะเชื่อมต่อไปยังอุปกรณ์ภายนอกในรูปแบบของ Memory Map ภายนอกและยังสามารถจองพื้นที่นี้ไว้ใช้เป็นพื้นที่เก็บ Data แบบ Extra ได้ด้วย

### 2.11.2 Program memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิพ ในรุ่น C5x สามารถต่อหน่วยความจำเพิ่มเติมได้ถึง 64 K 16 bit word เมื่อต่อหน่วยความจำดังกล่าวเพิ่มขึ้นแล้ว จะทำให้ C5x มีหน่วยความจำทั้งหมดคือ On-Chip ROM , Single – Access Program / Data Ram และ Dual – Access RAM ด้วยการใช้ Software จะทำให้สามารถจัดโครงสร้างของหน่วยความจำแต่ละส่วนใหญให้อยู่ภายในหรือภายนอก Address Map ก็ได้เมื่อ Map ไปยัง Program Space ชิพ จะประมวลผลข้อมูลจำกัดอยู่ภายในขอบเขตของมันโดยอัตโนมัติเมื่อ CPU สร้าง Address นอกเหนือไปจากขอบเขต ชิพ จะเปลี่ยนแปลงการทำงานไปประมวลผลข้อมูลจากอุปกรณ์ภายนอกโดยอัตโนมัติ

a) การจัดโครงสร้างของ Program space

หลังจากที่ทำการ Reset การจัดโครงสร้างของข้อมูล จะถูกกำหนดด้วยขา MC/MP ถ้าแรงดันที่ขานี้มีค่า High C51 จะจัดโครงสร้างข้อมูลเป็น Microprocessor และ On-Chip ROM จะไม่มีการนำมาใช้งานถ้าขานี้มีแรงดันเป็น Low C51 จะจัดโครงสร้างข้อมูลเป็น Microcomputer คือ On-Chip ROM จะถูกนำมาใช้งาน ดังนั้นถ้าต้องการนำ C51 ไปใช้งานเป็น Microcomputer มันจะเริ่มทำงานด้วยคำสั่งบน On-Chip ROM ไม่เช่นนั้นมันจะเริ่มการทำงานที่หน่วยความจำภายนอกในขณะที่ Program เริ่มทำงานที่หน่วยความจำภายนอกในขณะที่ Program เริ่มทำงานเราสามารถทำการเปลี่ยนแปลงค่าของ MP/MC ได้ใน PMST Register โดยการทำตามคำสั่งดังเช่นแสดงไว้ในบรรทัดข้างล่างนี้

OPL#8,PMST ;Remove boot ROM from program space

b) Program Memory Address Mapped

ค่าของ Reset, Interrupt และ Trap Vector จะถูกเก็บไว้ในพื้นที่ Program ซึ่งพื้นที่ในส่วนนี้จะอยู่ในตอนต้นของเนื้อที่หน่วยความจำ ในขณะที่ Reset ข้อมูลใน Vector Mapped จะมีค่าชี้ไปยัง Address oh

c) Local Data Memory

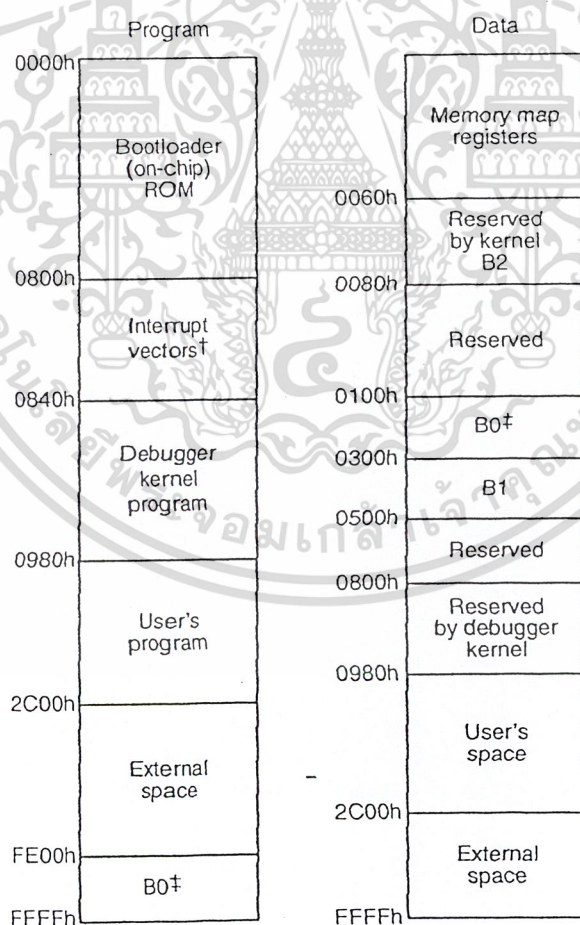
ขนาดของ Local Data Memory สามารถขยายได้ถึง 64K 16 bit-word โดยในชิพ C51 สร้างเนื้อที่ขนาด 1 K ไว้บนชิพ c50C50 มี 2K 16 bit maskable programmable ROM หน่วยความจำนี้ใช้สำหรับการ Boot เครื่องในตอนแรก ROM สามารถเลือกได้ในขณะ reset ด้วยขา mp/mc เราสามารถ set การทำงานได้ด้วยการควบคุม bit ใน PMST

RAM ขนาด  $1056 \times 16$  bit ซึ่งส่วนนี้จะสามารถทำงานแบบ dual-access สำหรับเก็บข้อมูลและสามารถใช้เก็บ program ในส่วนนี้ได้ด้วย

RAM  $9k \times 16$ - bit- on ชิป หน่วยความจำนี้สามารถใช้เป็นที่เก็บ program and data ก็ได้ Wait-state Generator จำนวน 16 ชุด เพื่อทำการสร้าง wait-state ในการติดต่อกับหน่วยความจำภายนอกและอุปกรณ์ I/O จะกำหนดขอบเขตที่จะทำการ map กับ wait-state ได้

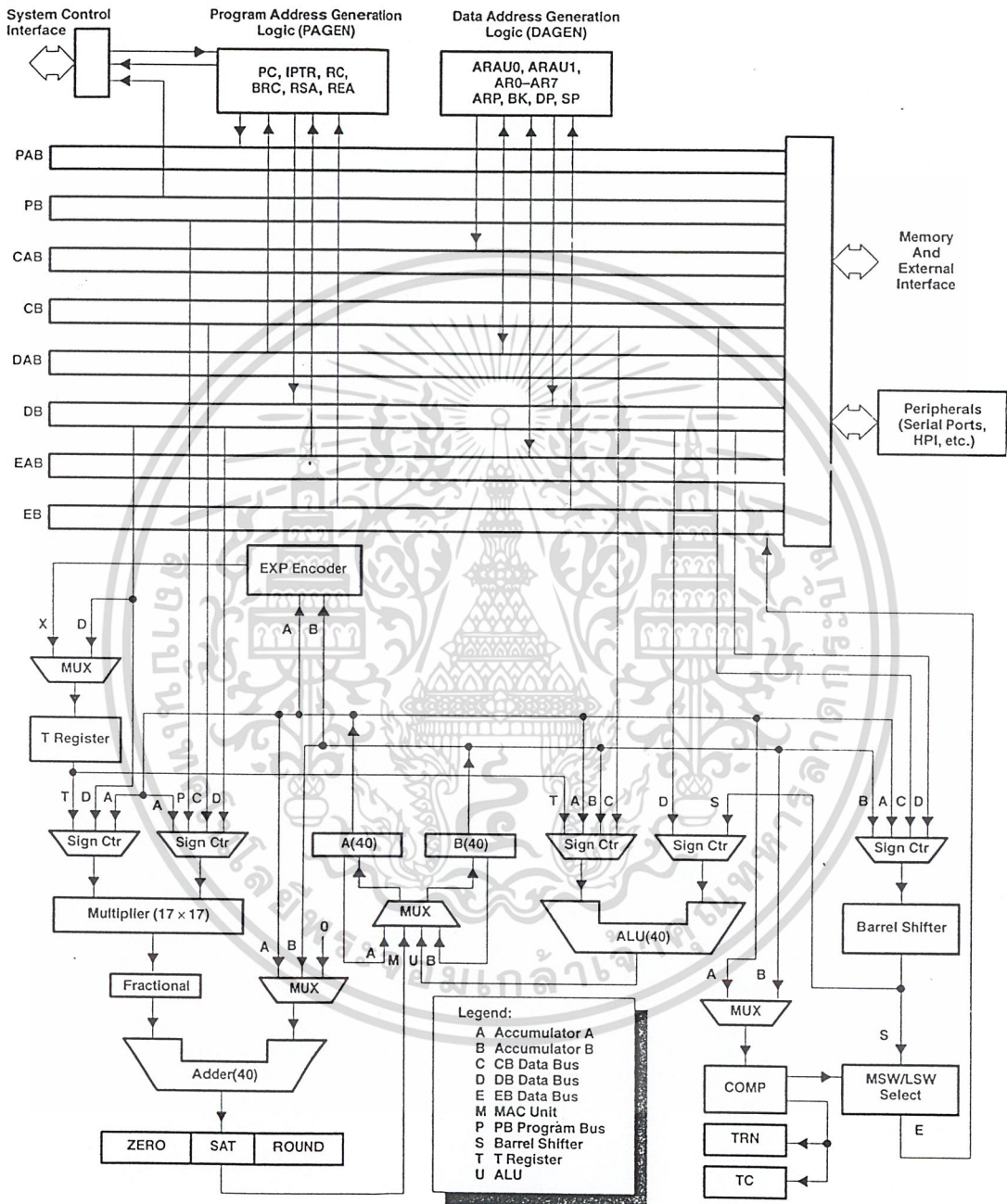
### 2.11.3 Local Data Memory

ขนาดของ Local Data Memory สามารถขยายได้ถึง 64 K 16 bit-word โดยใน ชิป C51 จะสร้างเนื้อที่ขนาด 1 K ไวบนชิป เป็นแบบ SARAM ชิป ทั้งหมดในรุ่น C5x มีขนาดของหน่วยความจำ Dual – Access Space RAM (DARAM) ขนาด 1056 word โดยสามารถนำมาจัดโครงสร้างใหม่ด้วย Software ให้อยู่ภายในหรือภายนอก Local Data Address Map ก็ได้ เมื่อหน่วยความจำส่วนนี้ถูก Map ลงใน Data Space ชิป จะทำการประมวลผลข้อมูล โดยอยู่ภายในขอบเขตที่กำหนด ถ้ามีการกำหนดพื้นที่ใช้งานเกินขอบเขต Processor จะเปลี่ยนไปทำงานในพื้นที่ Address ภายนอกโดยอัตโนมัติ



รูปที่ 2-5 Memory Map of the DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 Block Diagram แสดงโครงสร้างภายในทั้งหมดของ C5x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.11.4 Global Memory

สำหรับการทำงานแบบ Multi-Processing ชิป ในรุ่น C5x มีความสามารถในการกำหนด Memory Space เพื่อใช้เป็นทางผ่านสำหรับสื่อสารข้อมูลระหว่าง Processor โดยอาศัยขาสัญญาณ BR และ Ready Global Memory เป็น Memory ที่จัดให้มีการใช้งานร่วมกันระหว่าง Processor โดยอาศัยขาสัญญาณ BR และ Ready Global Memory เป็น Memory ที่จัดให้มีการใช้งานร่วมกันระหว่าง Processor มากกว่าหนึ่งตัว เพื่อที่ว่ากรใช้งานร่วมกันจะทำได้โดยสะดวก Global Memory Address Space ถูกแบ่งออกเป็นส่วนๆ คือส่วน Local และ Global ในส่วน Local จะถูกใช้งานในหน้าที่ที่ต้องแยกอิสระจากกันและในส่วน Global จะใช้เพื่อติดต่อสื่อสารกันกับ Processor ตัวอื่นๆ

### 2.11.5 อินพุต/เอาต์พุต Space

C5x มีพื้นที่รองรับอุปกรณ์ I/O ขนาด 64K 16 bit เช่น Digital – to – Analog ( D-A)

Memory – Mapped Core Processor Register			
Name	Address		Description
	Dec	Hex	
-----	0-3	0-3	Reserved
IMR	4	4	Interrupt Mask Register
GREG	5	5	Global Memory Allocation Register
IFR	6	6	Interrupt Flag Register
PMST	7	7	Processor Mode Status Register
RPTC	8	8	Repeat Counter Register
BRCR	9	9	Block Repeat Counter Register
PASR	10	A	Block Repeat Program Address Start Register
PAER	11	B	Block Repeat Program Address End Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TREG0	12	C	Temporary Register Use for Multiplicand
-------	----	---	--

TREG1	13	D	Temporary Register Use for Dynamic Shift Count (5 bits only)
TREG2	14	E	Temporary Register Use as Bit pointer In Dynamic Bit Test(4 bits only)
DBMR	15	F	Dynamic Bit Manipulation Register
AR0	16	10	Auxiliary Register Zero
AR1	17	11	Auxiliary Register One
AR2	18	12	Auxiliary Register Two
AR3	19	13	Auxiliary Register Three
AR4	20	14	Auxiliary Register Four
AR5	21	15	Auxiliary Register Five
AR6	22	16	Auxiliary Register Six
AR7	23	17	Auxiliary Register Seven
INDEX	24	18	Index Register
ARCR	25	19	Auxiliary Register Compare Register
CBSR1	26	1A	Circular Buffer 1 Start Register
CBER1	27	1B	Circular Buffer 1 End Register
CBSR2	28	1C	Circular Buffer 2 Start Register
CBER2	29	1D	Circular Buffer 2 End Register
CBCR	30	1E	- Circular Buffer Control Register
BMAR	31	1F	Block Move Address Register
Memory – Mapped Peripheral Register			
DRR	32	20	Data Receive Register
DXR	33	21	Data Transmit Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPC	34	22	Serial Port Control Register
-----	35	23	Reserved
TIM	36	24	Timer Register

PRD	37	25	Period Register
TCR	38	26	Timer Control Register
-----	39	27	Reserved
PDWSR	40	28	Program/Data S/W Wait-State Register
LOWSR	41	29	I/O S/W Wait-State Register
CWSR	42	2A	S/W Wait-State Control Register
-----	43-47	2B-2F	Reserved
TRCV	48	30	TDM Data Receive Register
TDXR	49	31	TDM Transmit Data Register
TSPC	50	32	TDM Serial Port Control Register
TCSR	51	33	TDM Channel Select Register
TRTA	52	34	TDM Receive/Transmit Address Register
TRAD	53	35	TDM Received Address Register
Name	Address		Description
	Dec	Hex	
-----	54 - 79	36 - 4F	Reserved
Memory – Mapped I/O Ports			
PA0	80	50	I/O Port 50h
PA1	81	51	I/O Port 51h
PA2	82	52	I/O Port 52h
PA3	83	53	I/O Port 53h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PA4	84	54	I/O Port 54h
PA5	85	55	I/O Port 55h
PA6	86	56	I/O Port 56h
PA7	87	57	I/O Port 57h

PA8	88	58	I/O Port 58h
PA9	89	59	I/O Port 59h
PA10	90	5A	I/O Port 5Ah
PA11	91	5B	I/O Port 5Bh
PA12	92	5C	I/O Port 5Ch
PA13	93	5D	I/O Port 5Dh
PA14	94	5E	I/O Port 5Eh
PA15	95	5F	I/O Port 5Fh

ตารางที่ 2-4 Memory – Mapped Register And I/O Ports

การทำงานติดต่อกับ Parallel / Data Memory แต่พื้นที่ใช้งานของ I/O จะถูกแยกออกจาก Program / Data Memory ด้วยสัญญาณ IS Port ทั้งหมดจำนวน 65536 Ports สามารถทำงานได้ด้วยคำสั่ง IN และ OUT

16 Address จาก 64 K I/O Port จะถูก Mapped ลงบนพื้นที่ของ Data Memory ทำให้ I/O Port สามารถเขียนหรืออ่านได้ด้วยคำสั่ง IN และ OUT เช่นเดียวกับการประมวลผลคำสั่งอื่นที่อยู่ใน Data Space ดังตัวอย่างต่อไปนี้ ซึ่งอธิบายการอ้างตำแหน่งแบบ Direct Address เพื่อทำงานเกี่ยวกับอุปกรณ์ I/O ในตำแหน่ง 51h

#### 2.11.6 Software Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C5x ซึ่คือรูปแบบ Source Code ให้ Compatible กับ C1x และ C2x และมีโครงสร้างที่พัฒนาขึ้นเพื่อปรับปรุงประสิทธิภาพในการทำงานและการใช้งานได้กว้างขวางมากขึ้น ชุดคำสั่งใน C5x จะมีเพิ่มขึ้นเพื่อรองรับ Hardware รวมทั้งเพิ่มการทำ Data Movement และ Memory Mapped Register รวมไปถึงรองรับ Parallel Logic Unit (PLU) เพื่อการทำงานเกี่ยวกับคณิตศาสตร์ Boolean , 32 bit Accumulator Buffer และ register ที่ใช้สำหรับบริการการ Interrupt เนื่องจาก On-Chip DARAM

### 1. Processor Initialization

ก่อนทำการประมวลผล Algorithm มีความจำเป็นที่จะต้องกำหนดค่าเริ่มต้นต่างๆให้แก่ Processor โดยทั่วไปการ Initialization จะกระทำทุกครั้งหลังจากการที่มีการ Reset โดยหลังทำการ Reset แล้ว IPTR bit ของ PMST จะถูก Clear ดังนั้น Vector จากการ Mapping จะชี้ไปที่ Page 0 ของพื้นที่ Program Memory ซึ่งพื้นที่ดังกล่าวโดยปกติแล้วบรรจุคำสั่ง Branch เพื่อชี้ไปยังตำแหน่งของ Initialization Routine ส่วนใน Hardware จะมีการ Disable Interrupt โดยอัตโนมัติ โครงสร้างของ Processor หลังจาก Reset ที่จะต้องทำการ Initialization ใหม่ได้แก่ส่วนต่างๆดังต่อไปนี้

Memory-Mapped และ Peripheral Control Register

Interrupt Structure (INTM)

Mode Control (OVM,SXM,PM,A VIX,NDX,TRM)

Memory Control (RAM,OVLY,CNF)

Auxiliary Register และ Auxiliary Pointer (ARP)

Data Memory Page Pointer (DP)

โดยที่ OVM (Overflow Mode), TC (Test/Control Flax), IMR (Interrupt Mask Register),Auxiliary Register Pointer(ARP),Auxiliary Register Pointer Buffer (ARB) และ Data Memory Page Pointer (DP) ไม่ต้องทำการ Initial ใหม่หลังทำการ Reset

### 2. Interrupt

C5x ประกอบไปด้วย External Maskable Interrupt จำนวน 4 ช่อง และ Non-Maskable Interrupt อีกช่องหนึ่งไว้ให้กับอุปกรณ์ภายนอก และยังมี Interrupt ภายใน ซึ่งได้แก่ Interrupt ที่เกิดจาก Serial Port, Timer และ Interrupt ที่เกิดจาก Software ในคำสั่ง Interrupt อันได้แก่ INTR,TRAP และ NMI

C5x มีความสามารถในการสร้าง Software Interrupt โดยใช้คำสั่ง INTR ซึ่งจะทำได้ทั้ง

ส่วนจาก 32 Interrupt Service Routine ทำการบริการ Interrupt โดยที่ 20 ISR ถูกจองไว้เพื่อบริการ แก่ External Interrupt, Peripheral Interrupt และสงวนไว้ใช้ในอนาคต ส่วน Interrupt Vector ที่ เหลืออีก 12 Location สามารถนำไปใช้ได้โดยผู้เขียน Program ในคำสั่ง INTR สามารถอ้าง Interrupt ใดๆก็ได้ภายใน 32 Location

Context Saving และ Restoring Function จะทำงานเมื่อ Interrupt Trap ถูก Execute และ 8 Deep Hardware Stack จะถูกใช้เพื่อเก็บค่า Address ที่จะต้องกลับมาทำงานต่อหลังจากการบริการ การ Interrupt เสร็จสิ้นเรียบร้อยแล้ว ดังนั้นจะมี One Deep Stack ( หรือ Shadow Register) เพื่อเก็บ ค่าของแต่ละ Register ดังต่อไปนี้

ACC	Accumulator
PREG	Product Register
STO	Status Register 0 ( INTM not restored)
ST1	Status Register 1 (XF not restored)
PMST	Processor Mode Status Register
TERG0	Temporary Register for Multiplier
TREG1	Temporary Register for Shift count
TREG2	Temporary Register for Bit test
INDX	Indirect Address Index Register
ARCR	Auxiliary Register compare Register

เมื่อได้รับสัญญาณ Interrupt Trap Register ทั้งหมดจะถูก Push ลงบน One-Deep Stack และ Shadow Register เหล่านี้จะถูก POP เมื่อมีการกลับจาก Interrupt ด้วยคำสั่ง RET1 หรือ RETE

### 3. Software Stack

C5x มี Hardware Stack ภายในจำนวน 8 Deep ด้วยกันเพื่อใช้ในการเก็บค่า Address ในขณะที่ทำการ Subroutine และ Interrupt

คำสั่ง PUSH และ POP มีการเพิ่มความสามารถของคำสั่งขึ้นมาเป็นคำสั่งใหม่อีก คำสั่งได้แก่ PSHD และ POPD ทำให้ Stack สามารถ Store และ Recover ได้จาก Data Memory เมื่อ มี Stack ถูกใช้ไปทั้งหมด 7 ค่าและมีการเขียนค่า Stack ทับลงไปอีกมากกว่า 2 ค่า โดยที่ค่าเก่ายังไม่ ถูก POP ขึ้นมาจำเป็นที่จะต้องมีการขยาย Stack

เนื่องจากคำสั่ง Call ใช้ Stack ในการเก็บค่า Program Counter คำสั่ง BACC จะมี

ประโยชน์ในการเข้าสู่ Main Program โดยไม่จำเป็นต้องเก็บค่าของ Program Counter ไว้ใน Memory

#### 4. Logical and Arithmetic Operations

PLU จะช่วยทำให้การส่งผ่านข้อมูลข้ามไปยัง Data Memory ทำได้ โดยไม่ส่งผลกระทบต่อค่าภายใน Accumulator หรือ Product Register ทำให้สามารถเคลื่อนย้ายค่าไปยังตำแหน่งใดๆภายใน Data Memory ได้โดยตรงกับ Source Operand

C5x ประกอบไปด้วยเงื่อนไขต่างๆ เพื่อทำการตรวจสอบก่อนที่จะกระโดดไปทำโปรแกรมอื่นๆบางส่วนจาก 13 เงื่อนไข โดยสามารถทดสอบแต่ละเงื่อนไขหรือทดสอบเงื่อนไขรวมกันได้ โดยใช้คำสั่ง CC,RETC,XC และ BCND

Search Algorithm ด้วยคำสั่ง CRGT ด้วยการใช้คำสั่ง CRGT และ RPTB เราสามารถค้นหาค่าสูงสุดภายใน Data Block และตำแหน่งของมันได้ การหาค่าต่ำสุดสามารถใช้คำสั่ง CRLT ค้นหาได้ในลักษณะเดียวกัน

#### 5. Matrix Multiplication Using Nested Loops

C5x มีคำสั่ง 3 คำสั่ง คำสั่งแรกคือ RPT ( Single-Instruction Repeat) สามารถทำคำสั่งซ้ำได้ n คำสั่ง RPTB ( Repeat Block) ทำงานด้วยจำนวน Loop ที่ตั้งไว้ใน BRCR Register ส่วนคำสั่ง BANZ (Branch if AR Not Zero) เป็นอีกคำสั่งที่สามารถกำหนดจำนวน loop ได้โดยใช้ Auxiliary Register

#### 6. Circular Buffers

Circular Addressing เป็นหัวข้อสำคัญในชุดคำสั่งของ C5x การทำงานแบบ Convolution, Correlation และ FIR Filter สามารถทำงานได้โดยการใช้ circular Buffer ภายใน Memory C5x มีการรองรับการทำงานแบบต่อเนื่องกันระหว่าง 2 Buffer โดยใช้ Auxiliary Register โดยมี CBSR1,CBSR2,CBER1,CBER2 และ CBCR ทำหน้าที่เป็น Memory – Mapped Register ที่ควบคุมการทำงานของ Circular Buffer

โดยก่อนที่จะเริ่มทำงานจะต้องมีการ load ค่า Address เริ่มต้นและสุดท้ายสุดใน Buffer Register และ Auxiliary Register ทำงานเป็น Pointer จะต้องทำการ Initial ค่าด้วย

## 7. Extended-Precision Arithmetic

การวิเคราะห์ตัวเลข, การคำนวณ Floating-Point หรือการทำงานอื่นๆ อาจมีความจำเป็นที่จะต้องใช้ตัวเลขที่มีค่ามากกว่า 32 bit เนื่องจาก C5x เป็น 16/32 bit Fixed – Point Processor ดังนั้น ในการคำนวณจึงต้องอาศัยการขยาย bit ด้วย Software เข้าช่วยในการทำงาน

### 7.1 การคูณ

มีคำสั่งในการใช้งาน เช่น MAC MACD MPY ซึ่งจะมีหน้าที่ในการคูณเลขแบบ 16 bit และผลลัพธ์ที่ได้จากการคำนวณ จะมีขนาด 32 bit ไว้ใน PREG (Product Register)

### 7.2 การหาร

มีคำสั่งในการใช้งาน เช่น SUB SUBC เป็นคำสั่งในการลบหลายๆครั้ง โดยการกำหนดให้ตัวตั้งและตัวหารมีค่าเป็นบวก 16 bit จะทำการลบทั้งหมด 16 ครั้ง แล้วจึงเก็บค่าผลหารที่มีขนาด 16 bit ไว้ในครึ่งล่างของ ACC ( accumulator) และเก็บอีก 16 bit ไว้ในส่วนครึ่งบนของ ACC การใช้คำสั่ง SUBC ทำได้ด้วยการลบหลายๆครั้ง ตัวตั้งจะถูกชิฟต์จนกระทั่งตัวหารถูกลบจนหมด แล้วเปลี่ยนเป็นค่าลบ ในการที่ไม่ทำให้คำตอบมีค่าเป็นลบค่าหนึ่งจะถูกเลื่อนเข้าไปแทนที่ในครึ่งล่างของ ACC การหารที่กล่าวมานั้นตัวตั้งและตัวหารจะต้องมีค่าเป็นบวก โดยใช้คำสั่ง Absolute

## 2.12 REGISTER ที่สำคัญกำหนดสถานะที่สำคัญใน C5x

การทำงานของหลายๆส่วนจะถูกกำหนดโดย Register ทั้งหมด 4 ตัว คือ ST0, ST1, CBCR และ PMST การเปลี่ยนแปลงค่า การเขียน และการเก็บของเดิมไว้เมื่อมีการเรียกใช้ subroutine หรือเมื่อเกิดการ interrupt โดย Register PMST และ CBCR จะอยู่ใน Data Memory ดังนั้นจึงถูกกระทำโดย ALU และ PLU

ส่วนค่าใน ST0 และ ST1 จะถูกเขียนและอ่านโดยคำสั่ง LST ตามลำดับไม่สามารถเขียนและอ่านเหมือน PMST กับ CBCR ได้แต่คำสั่ง LST ไม่สามารถกำหนด Bit INTM ได้แต่ไม่ทั้งหมดสามารถใช้คำสั่ง SETC, CLRC เพื่อทำให้เป็น 1 และ 0 ได้ตามลำดับ

ST0 (status register 0) มีขนาด 16 บิต

15	14	13	12	11	10	9	8	0
APR		OV	OVM	1	INTM	DP		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ARP Auxiliary Register Pointer 3 bit เป็นตัวชี้ Register AR เพื่อใช้ในการอ้าง Address แบบ indirect เมื่อทำการเก็บค่าของ ARP ค่าของ ARP เดิมจะถูกเก็บใน ARB โดยค่าใน ARP สามารถเปลี่ยนได้หลายทางโดย การเปลี่ยนค่า เกี่ยวกับการอ้าง memory ของหลายคำสั่งหรือคำสั่ง MAR
- OV Overflow Flag bit จะเป็น 1 เมื่อเกิดการ Overflow ขึ้นที่ ALU สามารถเซตบิตนี้จะเป็น 1 ด้วยคำสั่ง BCND หรือ LST
- OVM Overflow Mode bit ถ้าเป็น 1 จะมีการ check การเกิด Overflow จาก Time และ Counter ด้วยสามารถใช้คำสั่ง SETC CLRC และ LST ทำการเปลี่ยนแปลงค่า bit นี้ได้
- INTM Interrupt Bit Mode เมื่อ bit นี้เป็น 0 จะมีเฉพาะ Interrupt แบบ Non-Maskable เท่านั้นที่สามารถทำงานได้ แต่เมื่อเป็น 1 ทุกๆ Interrupt จะสามารถทำงานได้ โดยเมื่อทำการ Reset bit นี้จะเป็น 1 สามารถทำการเปลี่ยนแปลงค่าคำสั่งด้วย SETC,CLRC เหมือน bit อื่นๆ แต่ไม่สามารถใช้คำสั่ง LST กับ bit นี้ได้โดยการเกิด Interrupt Maskable จะทำให้ bit นี้เป็น 0 และเมื่อมีการ Interrupt Service Routine ด้วยคำสั่ง RETE จะทำให้ bit กลายเป็น 1
- DP Data Memory Page Pointer ค่า 9 bit ในนี้จะเป็นค่าที่ 7 bit ในคำสั่งในการอ้าง Memory แบบ Direct โดย 9 bit นี้จะเป็น MSB ในการอ้าง Memory ซึ่งมี 16 bit ค่านี้สามารถเปลี่ยนแปลงโดยคำสั่ง LDP หรืออ่านโดยคำสั่ง LST

### ST1 ( Status Register 1 ) มีขนาด 16 บิต

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ARB	CNF	TC	SXM	C	1	1	HM	1	XF	1	1	PM		

ARB Auxiliary Register Pointer Buffer เป็นตัวกำหนดค่าสำรองของ ARP

CNF On-Chip RAM Configuration Control Bit ถ้าเป็น 0 แสดงว่า RAM สามารถเป็นได้ทั้ง Program และ Data ถ้าเป็น 1 แสดงว่า RAM ใช้เป็นส่วนของ Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียงอย่างเดียว สามารถกระทำด้วยคำสั่ง SETC,CLRC,LST#1 เมื่อ RESET จะเป็น 0

- TC Test Control/flag Bit โดย bit สามารถเก็บผลของ ALU และ PLU ในการ Test bit การทำงานของชิพจะมีผลต่อคำสั่ง BITT,CMPR,CPL,OPL,NORM และ XPL TC=0 เมื่อใช้คำสั่ง CLRC TC TC=1 เมื่อใช้คำสั่ง SETC TC
- SXM Sign Extension Mode จะมีผลต่อเครื่องหมายการกระทำทางคณิตศาสตร์ SXM = 1 จะมีการเช็คเครื่องหมายทางคณิตศาสตร์โดยคำสั่ง CLRC SXM SXM = 0 จะไม่คำนึงถึงเครื่องหมายทางคณิตศาสตร์โดยคำสั่ง SETC SXM
- C Carry Bit เกิดจากการกระทำทางคณิตศาสตร์ที่มีการยืม การทดในการคำนวณ C=0 เมื่อผลของการลบเกิดการยืมหรือผลจากการบวกที่ไม่ทำให้เกิด carry bit โดยใช้คำสั่ง CLRC C C=1 เมื่อผลของการบวกเกิดการทด หรือผลของการลบที่ไม่เกิดการยืม โดยใช้คำสั่ง SETC C
- HM Hold Mode Bit HM = 0 เมื่อใช้คำสั่ง CLRC HM จะกระทำใน Program Memory ก่อนจนหมดแล้วทำการเช็คขาที่ทำการติดต่อกับสัญญาณภายนอกให้เป็น high impedance HM = 1 เมื่อใช้คำสั่ง SETC HM จะหยุดกระทำทุกอย่างแล้วทำการ active ขา hold
- XM XF Pin Status Bit เป็นขาแสดงสถานะของขา XF สามารถควบคุมอุปกรณ์ภายนอก XF=0 เมื่อใช้คำสั่ง CLRC XF XF = 1 เมื่อใช้คำสั่ง SETC XF ในการกระทำ interrupt จะไม่มีการเก็บค่าบิตนี้ไว้ ถ้าใน interrupt Routine มีการคืนค่าเดิมหลังการ Return หลังการ interrupt
- PM Product Shift Mode จะกำหนดเงื่อนไขการเก็บผลการคูณซึ่งจะมีการ shift หรือไม่ ถ้าเป็น 00 ผลคูณจะไม่มี shift ถ้าเป็น 01 ผลคูณจะมีการ shift left ไป 1 บิต ถ้าเป็น 11 ผลคูณจะมีการ shift left ไป 4 บิตแล้วทำการแทน LSB ด้วย 0 ถ้าเป็น 11 ผลคูณจะมีการ shift left ไป 6 bit จะมีผลกรณีที่มีการเก็บค่าไปยัง Data Memory ด้วย สามารถเขียนด้วยคำสั่ง SPM และ LST#1

### CBCR (Circular Buffer Control Register)

15	8	7	6	5	4	3	2	1	0
XXXX		CENB2		CAR2		CENB1		CAR1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CBSR1	เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าตำแหน่งแอดเดรสเริ่มต้นของ Circular Buffer #1
CBER1	เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าตำแหน่งแอดเดรสสุดท้ายของ Circular Buffer#1
CBSR2	เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าตำแหน่งแอดเดรสเริ่มต้นของ Circular Buffer#2
CBER2	เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าตำแหน่งแอดเดรสสุดท้ายของ Circular Buffer#2
CENB1	Circular Buffer 1 ทำงานที่ 1 และไม่ทำงานที่ 0
CENB2	Circular Buffer 2 ทำงานที่ 1 และไม่ทำงานที่ 0
CAR1	รีจิสเตอร์ชี้ตำแหน่งแอดเดรสใน Circular Buffer 1 (AR0-AR7)
CAR2	รีจิสเตอร์ชี้ตำแหน่งแอดเดรสใน Circular Buffer 2 (AR0-AR7)

การใช้งาน Circular Buffer ใน TMS320c50 สามารถทำได้โดยกำหนดแอดเดรสเริ่มต้นและกำหนดแอดเดรสสุดท้าย ซึ่งจะต้องกำหนดรีจิสเตอร์ช่วยชี้ตำแหน่งแอดเดรส ( AR0 ถึง AR7 ตัวใดตัวหนึ่ง)

#### PMST (Processor Mode Status Register)

	10	9	8	7	6	5	4	3	2	1	0
IPTR	0	0	0	AVIS	0	OVLY	RAM	MP/MC	NDX	TRM	BRAF

IPTR	Interrupt vector pointer bits เป็นบิตที่ใช้สำหรับเคิลียร์ Vector ที่อยู่ในแอดเดรสใน Program memory
AVIS	เป็นบิตที่กำหนดให้ enable/disable program address ภายในมองเห็น address pins AVIS=1 จะไม่มีการเปลี่ยนแปลงของขาแอดเดรส เมื่อมีการอ้างแอดเดรสภายใน AVIS=0 การเปลี่ยนแปลงภายใน จะแสดงที่ขาแอดเดรส
OVLY	Ram Over Bit นี้จะทำการกำหนด RAM ส่วนที่สำหรับเก็บ โปรแกรมสามารถกำหนดได้ด้วยตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ตารางที่ 2-5 Local Data Memory Configure Control

RAM เมื่อบิตนี้เป็น 1 จะกำหนดให้ RAM ภายในเป็นส่วนของการเก็บโปรแกรม ถ้าเป็น 0 จะไม่ใช่ สามารถแสดงการใช้ RAM ดังตาราง

OVLY	RAM	ON-chip SARAM CONFIGURATION
0	0	Disabled
0	1	Mapped into program space
1	0	Mapped into data space
1	1	Mapped into both program and data space

ตารางที่ 2-6 On-chip Single-Access RAM Configuration Control

MP/MC Microprocessor/Microcomputer Bit เมื่อเป็น 0 จะมีการใช้ ROM ภายในเมื่อเป็น 1 จะไม่มีการใช้ ROM ภายใน โดยจะอ่านสถานะจากขา  $\overline{CS}$  ในการ Reset ครั้งแรกจากนั้นการ Reset ครั้งต่อไปจะไม่เป็นผล

NDX Enable Extra Index Register ถ้าเป็น 0 จะทำงานในโหมดการอ้าง Address แบบ indirect ที่เหมือนกับตระกูล C2x ซึ่งก็คือใช้ในการอ้าง Address แบบ Indirect แบบ Indexing หรือการเปรียบเทียบค่า AR ได้เฉพาะ ARO เท่านั้น แต่ในการทำงานของ C5x จะสามารถทำได้หมดเมื่อกำหนดให้ bit นี้เท่ากับ 1 โดยจะเป็น 0 เมื่อทำการ Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRM	Enable Multiple TERGs เป็น bit กำหนดการทำงานของ TREG TREG TREG ให้ทำงานในโหมดต่างๆคือถ้าเป็น 0 จะทำงานโหมดที่เหมือนกับ C2x ซึ่งทั้ง 3 Register จะมีผลเหมือนกับคำสั่งของ C2x แต่ถ้าเป็น 1 จะสามารถแยกใช้ T-Register แต่ละตัวเป็น Pre-Scaling หรือใช้กับคำสั่ง BITT ได้โดย bit นี้จะเป็น 0 เมื่อทำการ reset
BRAF	Block Repeat Active Flag เป็น bit แสดงการทำงานของ Block Repeat ซึ่งจะเป็น 1 เมื่อมีการทำงานโดยเมื่อทำการ Reset bit นี้จะเป็น 0

### 2.13 Register อื่นๆที่สำคัญ

Software wait-state Register ในการใช้งาน C5x ต่อกับอุปกรณ์ภายนอกถ้าอุปกรณ์ภายนอกมีการทำงานช้ากว่าการอ้างถึงของ C5x มักจะอนุญาตรอให้มีการทำงานของอุปกรณ์ภายนอกซึ่งสามารถกำหนดได้ด้วย Software โดยมี Register ต่างๆ มากำหนดค่าพวกนี้ซึ่งจะมีทั้งหมด 3 ตัวคือ PDWSR ( 16 BIT) และ CWSR ( 5 bit ) ซึ่ง PDWSR จะเป็นตัวกำหนดการแบ่ง Memory ออกเป็น 4 Block อย่างละ 16K ทั้งของ Program Memory และ Data Memory, ส่วน LOWSR จะควบคุมทั้งหมด 16 ส่วน CWSR จะเป็นตัวกำหนด Range ของ Wait State โดย bit Big ใน CWSR จะเป็นตัวกำหนด โดยถ้าเป็น 0 จะกำหนด Wait State แยกเป็น port เดียว แต่ถ้าเป็น 1 จะกำหนดเป็น block อย่างละ 8k

โดยที่ Range ของ Wait State สามารถกำหนดได้ถึง 7 Machine Cycle เพื่อให้การติดต่อกับอุปกรณ์ภายนอกสามารถควบคุมได้โดยผ่านทางขา Ready

การแบ่ง bit ใน Register ทั้งสองออกเป็นคู่ๆ เพื่อกำหนด Wait State ให้ทั้งสองส่วนของ Program และ Data ( ใน PDWSR) ซึ่งสามารถมีได้ 4 ค่าของ Wait State โดยจะเป็นค่าอะไรต้องดูจากค่าใน CRCSR ประกอบเช่นใน bit 0 กับ 1 จะเป็นตัวกำหนด Wait State ของ Program MEMORY จากช่วง Address 000h ถึง 3fffh และส่วนของ I/O สามารถกำหนดได้ 2 แบบ ขึ้นอยู่กับ bit Big ใน CWSR โดยที่ถ้า bit Big เป็น 0 จะเป็นตัวกำหนด I/O ที่แบ่งเป็นช่วงๆเหมือน memory โดยค่าของ Wait State แต่ละค่าจะดูจากใน CWSR อีกครั้ง การกำหนดค่าของ Wait State นั้นจะเป็นการทำกับอุปกรณ์ภายนอกเท่านั้นจะไม่มีผลกับ Memory ภายในแต่อย่างใด

โดยที่การกำหนดค่าใน CWSR จะทำการกำหนดค่าของ Wait State ให้กับส่วนต่างๆดังตารางที่ 7

โดยที่ค่าของ bit ต่างๆ จะมีผลต่อค่าของ Wait State ดังตารางที่ 2-8 (ต้องดูค่าที่กำหนดใน PDWSR และ IOWSR ประกอบด้วย)

โดยหลังการ Reset Wait State ทุกๆ ค่าจะกลายเป็น 7 Machine Cycle โดยค่าใน Bit Big จะเป็น 0

N ( Bit Position in CWSR)	SPACE
0	Program
1	Data
2	I/O (lower – haif:Port 0-Port7 if BIG =0,0000h-7FFFh if BIG=1
3	I/O (lower – haif:Port 8-PortF if BIG =0,8000h-0FFFFh if BIG=1
	BIG mode bit

ตารางที่ 2-7 การใช้งาน Bit ต่างๆของ CWSR

Wait-State Field of PDWSR or LOWSR (Binary Value)	No of Wait States (CWSR Bit n = 0)	No of Wait states ( CWSR Bit n = 1)
00	0	0
01	1	1
10	2	2
11	3	3

ตารางที่ 2-8 ความสัมพันธ์ของค่า Wait State กับค่าใน PDWSR และ IOWSR

## 2.14 Register เกี่ยวกับ Interrupt

ในการใช้งานอุปกรณ์ประเภท Microprocessor ลักษณะการทำงานที่สำคัญอย่างหนึ่งคือการ Interrupt เราต้องทราบการทำงานของ Interrupt เพราะมีความจำเป็นที่ต้องการแต่ขั้นตอนการออกแบบ Hardware และการเขียนโปรแกรม โดย C5x มี Register ที่ทำการเกี่ยวกับการ Interrupt อยู่หลายตัวซึ่งอยู่ในรูป Register PMST คือ IPTR ซึ่งได้กล่าวถึงมาแล้วที่เหลืออีก 2 ตัวคือ IFR และ IMR ซึ่งมีรายละเอียดแต่ละตัวดังนี้ (ดูจากรายละเอียดเพิ่มเติมในส่วนของการจัดการ Interrupt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IFR Register ( Interrupt Flag Register ) ขนาด 9 bit ใน Data Memory ถือเป็น Memory Map Register ใช้ในการแสดงสถานะของ Maskable Interrupt ทั้ง 9 ตัว โดยจะเป็น 1 เมื่อ Interrupt นั้นทำงานโดยสามารถตั้ง Clear Interrupt จะทำให้มันเป็น 1 อัตโนมัติและจะ Clear อัตโนมัติเมื่อมีการตอบรับนั้นโดยจะทำการ Clear พร้อมกับ CPU สร้างสัญญาณ Interrupt Acknowledge โดยทั้งหมดจะเป็น 0 เมื่อทำการ Reset ค่า bit ต่างๆ ที่ทำการควบคุมการ Interrupt ใน register นี้จะสามารถแสดงได้ดังนี้

8	7	6	5	4	3	2	1	0
INT4	TXINT	TRINT	XINT	RINT	TINT	INT3	INT2	INT1

IMR ( Interrupt Mark Register ) ขนาด 9 bit ใน Data Memory ถือเป็น Memory Map Register เหมือนกับ FIR เป็น register ที่จะเป็นตัวกำหนดการใช้งาน Interrupt นั้น โดยสามารถเขียนและอ่านได้เหมือน Data Memory หรือคำสั่งเกี่ยวกับ Memory Map Register เช่น SAMM

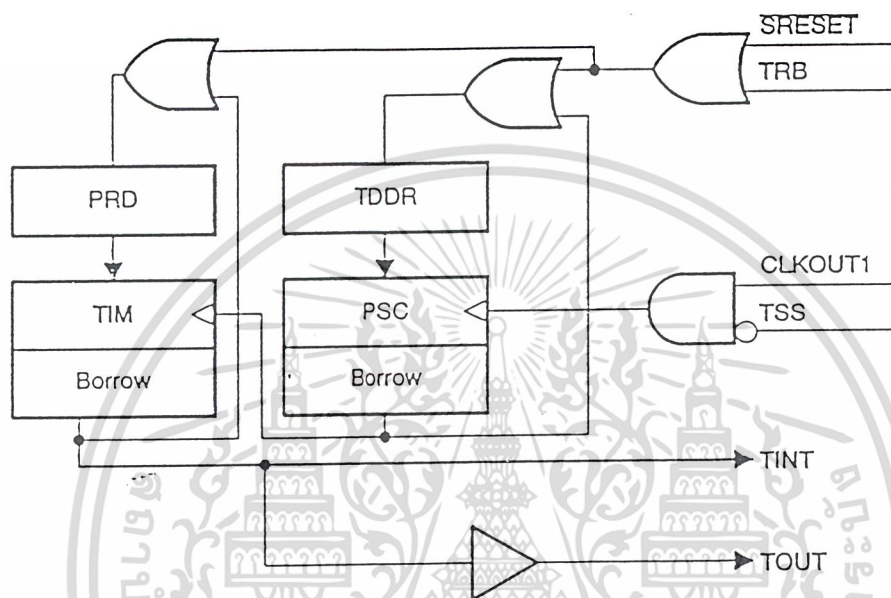
### 2.15 Register ควบคุมการทำงานของอุปกรณ์ร่วม (Peripheral Interfacing Circuit)

SPC Register ( Serial Port Control Register ) ขนาด 16 bit ที่ใช้ในการควบคุมการทำงานของ Serial Port ที่จะทำการติดต่อกับอุปกรณ์ภายนอกแบบอนุกรมโดย อยู่ในส่วนของ Memory Map Register โดยมีการกำหนดการทำงานต่างๆเหมือนกับ Data Memory และ Memory map Register อื่นๆ

TCR Register ( Time Control Register)  
 ใน c5x มี Hardware Timer ขนาด 16 bit โดยมี 4 bit pre-scalar สามารถกำหนดการ control ผ่านทาง TCR Register ซึ่งถือเป็น Memory Map Register สามารถทำการเขียนการอ่านและเขียนเหมือนกับ Memory ทั่วไป

โดยที่อัตราการนับจะสามารถกำหนดได้ใน PDR และ PSC ซึ่ง Timer จะทำการนับโดยนับลงทุกๆ Machine Cycle โดยเมื่อนับถึง 0 แล้วจะทำการสร้างสัญญาณ Interrupt ของ Timer ขึ้นมาพร้อมกับการเปลี่ยนค่าที่หา Tout โดยอัตราการหารหรือการ Interrupt จะเป็น

$$T_{out} = 1/t_c(TDDR+1)(PRD+1)$$



รูปที่ 2-7 แสดง Block Diagram การทำงานของ Timer ใน C5x

โดยค่า PRD จะสามารถเขียนใน Register นี้โดยตรงแต่ค่าของ TDDR จะทำการเขียนผ่าน Register TCR : ซึ่งรายละเอียดของ Register TCR จะสามารถแสดงได้ดังนี้

รายละเอียดของ Register TCR

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

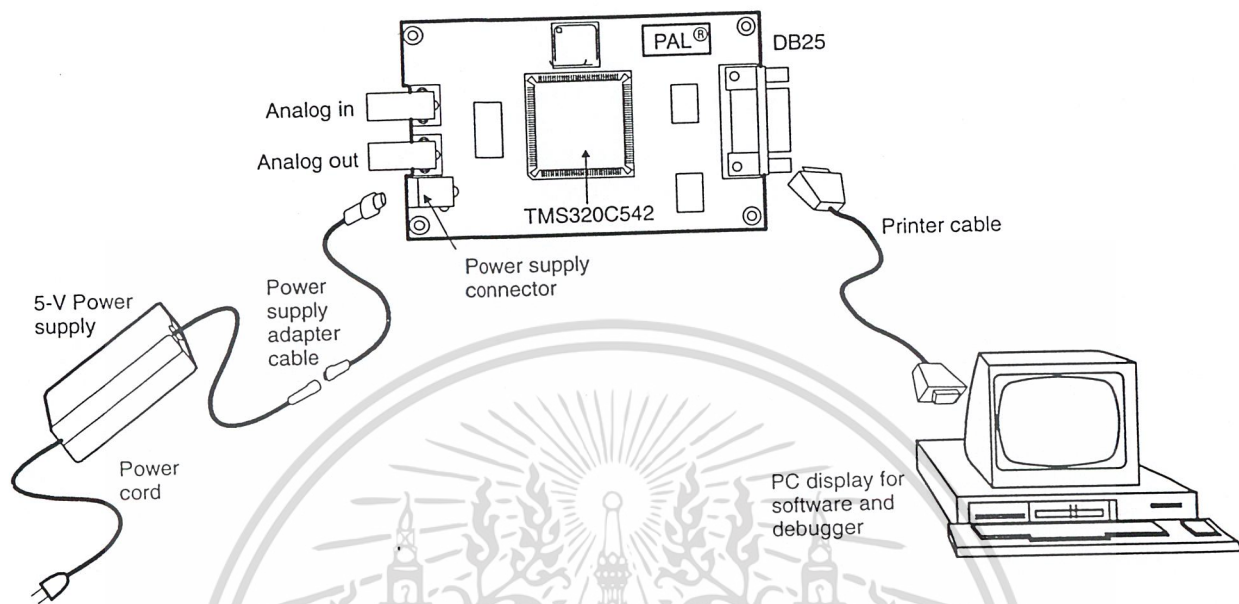
TDDR	(Timer Division Down Ratio) เป็นตัวกำหนดอัตราการหาร โดยจะโหลดค่าจาก TDDR ไปเก็บใน SPC แล้วทำการนับลงโดยขณะนับ จะไม่เกี่ยวกับค่าใน TDDR
TSS	(Time Slope Status bit) เป็นตัวกำหนดการ start stop ของ timer TSS = 1 จะเป็นการ stop timer TSS =0 จะเป็นการ start timer
TRB	(Timer reset) จะเป็นบิตที่ทำการสั่ง Reset time ซึ่งจะทำการโหลดค่าจาก TDDR ไปเก็บใน PSC แล้วจะทำการนับใหม่ โดยขณะอ่าน bit จะเป็น 0 ตลอด
PSC	จะทำการนับลงมาเรื่อยๆตามการทำงานของ CPU โดยไม่สามารถอ่านได้ต้องทำให้มันหยุดการทำงานก่อนโดยการให้ TSS=1

## 2.16 TMS320C5x DSK Starter Kit

ส่วนสำคัญในโครงการนี้เป็นชุดพัฒนาโปรแกรม สำหรับ ชิพ DSP ตระกูล TMS320C5 ซึ่งเป็นชิพ Digital Processing ตระกูล Fixed-Point ซึ่งนำมารวมกันกับ ชิพ Analog Interface เป็นบอร์ดที่ใช้ในการทดลองเกี่ยวกับ DSP ในชุดทดลองนี้ประกอบด้วย

ส่วนของ Debugger ที่ใช้พัฒนาโปรแกรม โดยการทำแก้ไขผ่านทาง Computer PC (โดยติดต่อกันผ่านทาง ชิพ Analog Interface เบอร์ TLC 32040 ซึ่งภายในประกอบด้วย Analog to Digital และ Digital to Analog ขนาด 14 ซึ่ง ชิพ สามารถโปรแกรม การทำงานของมันผ่านทาง Register ภายในโดยที่มันติดต่อกับ TMS320C5x แบบ Serial ทั้งรับและส่งข้อมูล ทั้งการโปรแกรมค่าต่างๆใน Register

ส่วนที่สำคัญที่สุดคือ ชิพ DSP TMS320C50 เป็นส่วนที่นำมาทำการรับค่าเข้ามาเพื่อกระทำการคำนวณแบบต่างๆตามโปรแกรมโดยภายในจะมีส่วนของ ROM และ RAM ในการเก็บ Program และ Data เพื่อทำการเข้าถึงรวดเร็วเพราะในการประมวลผลสัญญาณแบบ Real Time ต้องใช้ความเร็วของอุปกรณ์สูง ส่วนของการเก็บ Program และ Data (10K ram) ก็มีมากพอสำหรับ Application ขนาดใหญ่ทางด้าน Signal Processing



รูปที่ 2-8 การต่อใช้งาน Board DSK

ใน Memory Map เป็นส่วนที่สำคัญอีกส่วนหนึ่งที่เราจะต้องทราบเกี่ยวกับพื้นที่ที่จะทำการเขียนโปรแกรม, พื้นที่ในการใช้ interrupt และพื้นที่ในการเก็บหน่วยความจำกับส่วนอื่นๆที่จำเป็นในการเขียนโปรแกรม

นอกจากนี้ยังมีความสะดวกสบายในการพัฒนาโปรแกรม เพราะเป็นชุดทดลองสำหรับการเริ่มต้นการเรียนรู้การทำงานของ DSP ของบริษัท ซิฟ มาสำหรับผู้เริ่มต้นการศึกษาและยังมี Program ตัวอย่างซึ่งเป็น Program เกี่ยวกับการนำไปใช้งานด้านต่างๆ (สามารถ Download ได้เพิ่มเติมที่ [WWW.TI.COM](http://WWW.TI.COM)) รวมทั้งมีส่วนของการรองรับการขยาย Hardware ซึ่งสามารถทำได้สะดวก

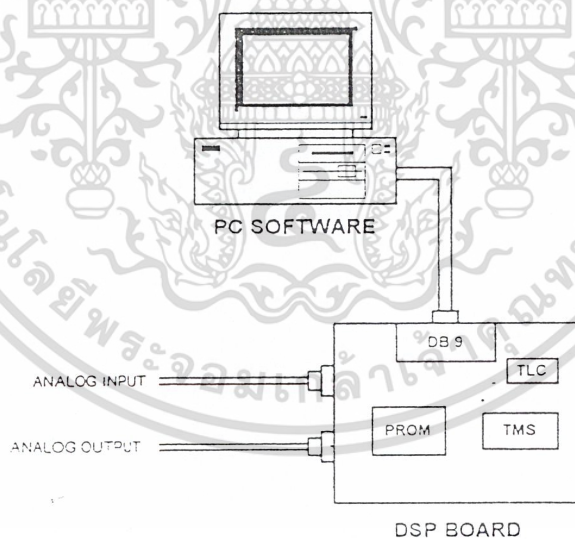
### บทที่ 3

#### รายละเอียดเกี่ยวกับโครงการงาน

ในโครงการงานนี้เป็นการใช้บอร์ด DSP ( Digital Signal Processing) มาประยุกต์ใช้งานด้านดิจิตอลฟิลเตอร์ ซึ่งดิจิตอลฟิลเตอร์นั้นมีด้วยกันหลายชนิด ไม่ว่าจะเป็นแบบ FIR ( Finite Impulse Response Filter) หรือแบบ IIR ( Infinite Impulse Response Filter) และอาจแบ่งเป็นตัวกรองป้อนกลับและแบบไม่ป้อนกลับก็ได้ ในการประยุกต์ใช้บอร์ด DSP นี้ จะใช้การออกแบบวงจรแบบ FIR ซึ่งวงจรถูกออกแบบ FIR นี้มีข้อดีอยู่ 3 ประการคือ

1. เป็นระบบที่เสถียร (stable)
2. เป็นระบบที่เป็นจริง (realizable)
3. สามารถออกแบบให้เป็นเฟสเชิงเส้นได้ง่าย (linear phase)

#### 3.1 โครงสร้างทางฮาร์ดแวร์



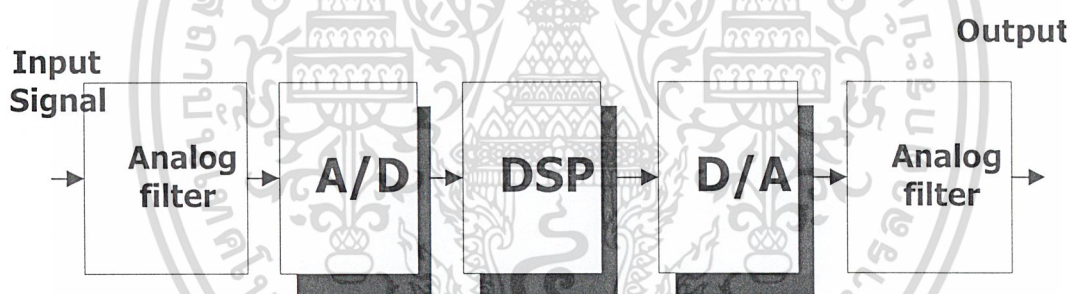
รูปที่ 3-1 แสดงโครงสร้างทางฮาร์ดแวร์ ของโครงการงาน

จากโครงสร้างของบอร์ดเราสามารถเขียนเป็นบล็อกไดอะแกรมของระบบ DSP ที่ใช้ในโครงการงานดังรูปที่ 3-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากระบบ DSP ตามบล็อกไดอะแกรมจะประกอบด้วย สัญญาณอนาล็อกเข้ามาจะผ่านวงจรกรองแบบแบนด์พาส เพื่อที่จะกำจัดช่วงความถี่ของสัญญาณอินพุท และยังช่วยโปรเซสเซอร์ไม่ให้ทำงานนอกเหนือความถี่ทำงาน จากนั้นจะส่งต่อไปยังวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล (ADC) แล้วส่งไปให้ระบบประมวลผลสัญญาณดิจิตอลต่อไป ระบบประมวลผลสัญญาณดิจิตอล (DSP Processor : TMS320C542) ก็จะนำเอาอัลกอริทึมที่มีอยู่มากระทำกับข้อมูลสัญญาณ จากนั้นก็จะส่งข้อมูลสัญญาณที่ถูกประมวลผลแล้ว ไปให้ส่วนของวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อก (DAC) เพื่อเปลี่ยนสัญญาณข้อมูลกลับไปเป็นสัญญาณอนาล็อกที่ต้องการ และเพื่อให้เอาท์พุทเป็นสัญญาณอนาล็อกมากขึ้นจะต้องผ่าน วงจรกรองสัญญาณให้เรียบ (Low Pass Filter) และจะขจัดสัญญาณความถี่สูงที่ไม่ต้องการออกไป

จากระบบข้างต้นนั้นถูกควบคุมด้วยพีซีโฮสต์ (PC Host) โดยจะควบคุมทางด้าน การเปลี่ยนแปลง และคำสั่งให้ทางโปรแกรม กำหนดค่าที่จำเป็นต่อ โปรเซสเซอร์



รูปที่ 3-2 บล็อกไดอะแกรมของระบบ DSP ที่ใช้ในโครงการ

### 3.2 รายละเอียดทางด้านซอฟต์แวร์

เนื่องด้วยโครงงานนี้เป็นการออกแบบวงจรกรองชนิดต่างๆ ซึ่งสามารถเปลี่ยนค่าพารามิเตอร์ ของวงจรกรองได้ เช่นการเปลี่ยนชนิดของวงจรกรอง การเปลี่ยนความถี่คัทออฟและอื่นๆ จึงจำเป็นต้องมีโปรแกรมสำหรับการทำงานต่างๆ ดังที่กล่าวมา ซึ่งจะแบ่งออกเป็น 2 ส่วนคือ

1. ส่วนโปรแกรมคำนวณฟิลเตอร์ บน PC
2. ส่วนโปรแกรมคำนวณฟิลเตอร์ บน TMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 โปรแกรมคำนวณฟิลเตอร์บน PC

การออกแบบ FIR ที่นำมาใช้นี้จะสมการของคอนโวลูชัน ซัม (Convolution Sum)

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (3.1)$$

โดยที่  $h(n)$ ,  $n=0,1,\dots,N-1$  จะเป็นการตอบสนองอิมพัลส์ (Impulse response) ของวงจรกรองที่เราจะออกแบบ ก่อนอื่นเราต้องกำหนดการตอบสนองความถี่ (Frequency response) ของวงจรกรอง ที่เราจะออกแบบก่อน โดย

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n} \quad (3.2)$$

จากสมการ (3.2) หาเมกนิจูดของการตอบสนองความถี่  $|H(e^{j\omega})|$  และหาเฟส ของ  $H(e^{j\omega})$  ได้ เป็น

$$H(e^{j\omega}) = \beta - \alpha\omega \quad ; \omega \geq 0 \quad (3.3)$$

ในการหา  $h(n)$  จะสามารถแบ่งออกได้โดย

$$h(n) = h(N-1-n) \quad ; 0 \leq n \leq N-1$$

$$\beta=0, \quad \alpha=(N-1)/2 \quad \text{ซึ่งจะเป็นแบบสมมาตร}$$

$$\text{และ } h(n) = -h(N-1-n) \quad ; 0 \leq n \leq N-1$$

$$\beta=\pm\pi/2 \text{ และ } \alpha = (N-1)/2 \quad \text{จะเป็นแบบไม่สมมาตร}$$

จากการพิจารณาว่าการออกแบบหา  $h(n)$  โดยใช้แบบสมมาตรจะลดการคำนวณลงถึง 50 %

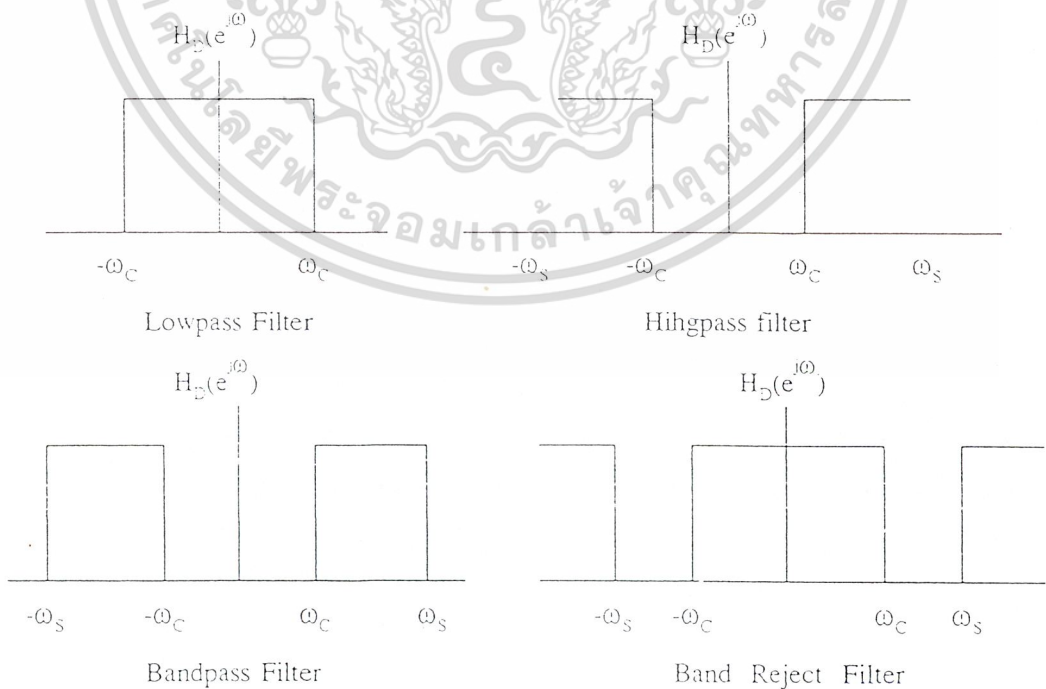
ต่อมาในการออกแบบของ FIR จำเป็นต้องมีการใช้ วิธีที่เรียกว่า วิธีหน้าต่าง (window method) เพื่อให้ตัวกรอง ที่ได้เป็นตัวกรองที่ใกล้เคียงกับตัวกรอง ในอุดมคติมากที่สุด ซึ่งการใช้หน้าต่างมีหลายวิธี ดังแสดงในตารางที่ 3-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bartlett	$\frac{2n}{N-1}, 0 \leq n \leq \frac{N-1}{2}$ $2 - \frac{2n}{N-1}, \frac{N-1}{2} \leq n \leq N-1$
Hanning	$0.5 \left( 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right)$
Hamming	$0.54 - 0.46 \cos \left( \frac{2\pi n}{N-1} \right)$
Blackman	$0.42 - 0.5 \cos \left( \frac{2\pi n}{N-1} \right) + 0.08 \cos \left( \frac{4\pi n}{N-1} \right)$
Kaiser	$\frac{I_0 \left[ \beta \sqrt{1 - \left( 1 - \frac{2n}{N-1} \right)^2} \right]}{I_0[\beta]}$ <p><math>I_0</math> เป็นค่าจากฟังก์ชันเบสเซล (Bessel function) อันดับที่ 0</p>

ตารางที่ 3-1 แสดงหน้าต่างแบบต่างๆ

การออกแบบต้องกำหนดตัวกรองแบบอุดมคติ  $H_D(e^{j\omega})$  ก่อน ดังรูปที่ 3-3



รูปที่ 3-3 แสดงคุณสมบัติของตัวกรองอุดมคติแบบต่างๆ

เอกสารนี้เป็นเอกสารที่ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดให้เป็น ยูนิตีแมกนิจูด (unity magnitude) (=1)  
จะได้

$$H_D(e^{j\omega}) = \begin{cases} e^{-j\alpha\omega} & ; |\omega| \leq \omega_c \\ 0 & ; \omega_c < |\omega| \leq \pi \end{cases} \quad (3.4)$$

โดยที่  $\omega_c$  = Corner Frequency เราจะหา  $h(n)$  ได้จาก

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega \quad (3.5)$$

การหา  $h(n)$  จำเป็นต้องกำหนดค่า  $N$  เพื่อให้การครอบคลุมการคำนวณการคำนวณการคำนวณ

$$h(n) = \begin{cases} h_d(n) & ; 0 \leq n \leq N-1 \\ 0 & ; \text{Otherwise} \end{cases} \quad (3.6)$$

และ 
$$\alpha = \frac{N-1}{2}$$

วิธีนี้จะเรียกว่า การใช้วิธีวินโดว์ ถ้ากำหนดให้ฟังก์ชันวินโดว์ เป็น  $w(n)$  จะเขียนใหม่ได้  
เป็น

$$h(n) = h_d(n) \cdot w(n) \quad (3.7)$$

โดยที่

$$w(n) = \begin{cases} \text{Some symmetric function over} & ; 0 \leq n \leq N-1 \\ 0 & ; \text{Otherwise} \end{cases}$$

ในการออกแบบมักเลือก  $w(n)$  ตามการลดทอนสัญญาณที่ช่วงหยุด (stopband) ว่ามากน้อยเพียงใด

### ตัวอย่างที่ 3.1

ออกแบบวงจรกรองโลว์พาสฟิลเตอร์ ที่มีอัตราแซมปลิง (sampling rate) = 8 kHz

ความถี่แซมปลิง : 8 kHz

ช่วงผ่านความถี่: 0 – 1 kHz

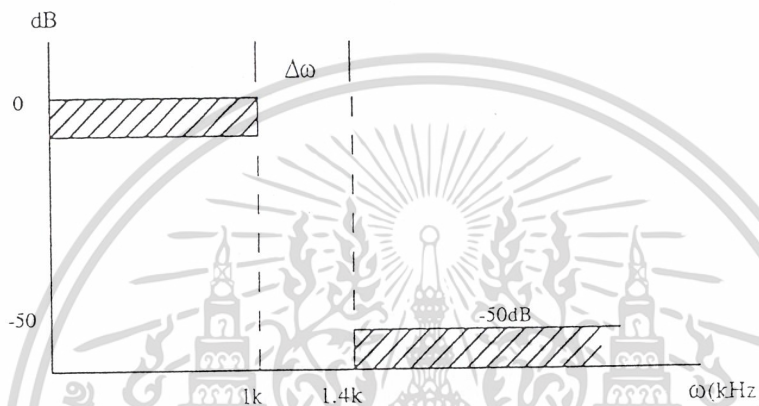
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงหยุด: 1.4 – 4 kHz

การลดทอนสัญญาณในช่วงหยุด : 50 dB

จากตารางที่ 3-2 จะเห็นว่าทำให้การลดทอนสัญญาณในช่วงหยุด มีค่า 50 dB จะใช้  $w(n)$  ได้เมื่อแฮมมิง หรือ แบดลิกแมน ในที่นี้เลือกใช้หน้าต่างแฮมมิง

หาใช้ความถี่ที่เปลี่ยนแปลง จากรูปที่แสดงข้างล่างนี้



$$\Delta\omega = \frac{2\pi(1.4\text{kHz} - 1\text{kHz})}{8\text{kHz}} = 0.1\pi$$

หาขนาดของ  $N$  ที่ใช้ จากตาราง

$$N = \frac{6.6\pi}{\Delta\pi} = \frac{6.6\pi}{0.1\pi} = 66$$

ซึ่งจะเลือกใช้  $N = 67$  (สมมาตร) และจะได้

$$\alpha = \frac{N-1}{2} = 33$$

คำนวณหา

$$\omega_c = \frac{(1\text{kHz} + 1.4\text{kHz})\pi}{8\text{kHz}}$$

$$= 0.3\pi$$

ซึ่งจะได้

$$H_D(e^{j\omega}) = \begin{cases} e^{-j33\omega} & ; |\omega| \leq 0.3\pi \\ 0 & ; 0.3\pi \leq \omega \leq \pi \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้สมการหาค่า

$$h_a(n) = \frac{\sin[0.3(n-33)]}{\pi(n-33)}$$

กำหนดให้  $w(n) = 0.54 - 0.46 \cos(2\pi n/66)$       $0 \leq n \leq 66$

จะได้ 
$$h(n) = \frac{\sin[0.3\pi(n-33)]}{\pi(n-33)} [0.54 - 0.46 \cos(2\pi n / 60)] \quad ; 0 \leq n \leq 66$$

นำ  $h(n)$  มาใช้ในการคำนวณบน PC เพื่อหาค่า  $h(n)$  ที่  $0 \leq n \leq 66$  ดังตาราง

Filter Length = 67	
$h(0) = -2.384527E-04 = h(66)$	$h(17) = 6.570477E-03 = h(49)$
$h(1) = -7.765305E-04 = h(65)$	$h(18) = 1.284836E-02 = h(48)$
$h(2) = -7.336193E-04 = h(64)$	$h(19) = 8.665973E-03 = h(47)$
$h(3) = -3.645995E-09 = h(63)$	$h(20) = -5.224195E-03 = h(46)$
$h(4) = 1.002993E-03 = h(62)$	$h(21) = -1.844362E-02 = h(45)$
$h(5) = 1.417815E-03 = h(61)$	$h(22) = -1.802628E-02 = h(44)$
$h(6) = 5.574883E-04 = h(60)$	$h(23) = -3.472296E-08 = h(43)$
$h(7) = -1.283883E-03 = h(59)$	$h(24) = 2.407036E-02 = h(42)$
$h(8) = -2.636652E-03 = h(58)$	$h(25) = 3.303238E-02 = h(41)$
$h(9) = -2.384527E-03 = h(57)$	$h(26) = 1.266898E-02 = h(40)$
$h(10) = 1.168261E-03 = h(56)$	$h(27) = -2.890585E-02 = h(39)$
$h(11) = 4.265746E-03 = h(55)$	$h(28) = -6.040657E-02 = h(38)$
$h(12) = 4.278595E-03 = h(54)$	$h(29) = -4.523324E-02 = h(37)$
$h(13) = 1.676480E-08 = h(53)$	$h(30) = 3.217674E-02 = h(36)$
$h(14) = -5.849049E-03 = h(52)$	$h(31) = 1.501070E-01 = h(35)$
$h(15) = -7.980914E-03 = h(51)$	$h(32) = 2.569817E-01 = h(34)$
$h(16) = -2.997856E-03 = h(50)$	$h(33) = 3.000000E-01 = h(33)$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของตัวกรอง ในตัวอย่างที่ 3.1  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมบน PC จะจัดการเก็บตัวกรอง  $h(n)$  เพื่อส่งให้บอร์ดเก็บเป็นหน่วยความจำ ข้อมูลของ TMS อีกที สำหรับ โปรแกรมการคำนวณหา ค่าสัมประสิทธิ์ของ  $h(n)$  (โลว์พาส)

เนื่องจากจะต้องมีการแสดงผลของตัวกรอง ที่ออกแบบมา จึงต้องทำการพล็อตกราฟของ ผลตอบสนองความถี่โดยพล็อต จากการหาค่าของ  $H_D(e^{j\omega})$

เพื่อให้การคำนวณและออกแบบตัวกรอง ให้ได้ผลตอบสนองที่กำหนดไว้แน่นอนขึ้น เรา เลือกใช้วิธี วินโดว์ (window methods) โดยเลือกใช้ ไคเซอร์วินโดว์ จากข้อจำกัดที่ว่า

- ความกว้างของ โหลบลึก (main lobe) แปรตามค่า  $1/N$  หมายถึงว่าถ้าต้องการออกแบบตัวกรองให้มีแถบเปลี่ยนความถี่ค่าแคบ ต้องใช้  $N$  ค่ามากๆ

- ค่าขนาดของ โหลบข้าง (side lobe) ทำให้เกิดลูกคลื่นบนผลตอบสนองความถี่ไม่แปรตามค่า  $N$  แต่ขึ้นอยู่กับชนิดของวินโดว์ที่ใช้

ดังนั้นการออกแบบตัวกรอง ให้มีค่าลดทอนในแถบหยุดน้อยที่สุด (minimum stopband attenuation) ตามต้องการอาจทำได้โดยต้องเลือกใช้ชนิดของวินโดว์ให้เหมาะสม

ไคเซอร์ (J.F. Kaiser) ใช้ฟังก์ชันทรงคล้ายทรงกลมแบนข้าง (prolate spheroidal function) มาทำการประเมินค่า ผลที่ได้คือ ได้วินโดว์ที่มีคุณสมบัติแลกเปลี่ยน (trade off) ของ โหลบลึกกับค่าขนาดของโหลบข้าง โดยมีพารามิเตอร์ เป็นตัวควบคุมเพื่อปรับค่าขนาดโหลบข้าง เมื่อเทียบกับค่าขนาดของ โหลบลึกได้

การออกแบบโดยทั่วไปพารามิเตอร์  $\delta_p$  และ  $\delta_s$  เป็นตัวกำหนดคุณสมบัติของค่าขนาดลูกคลื่นในแถบผ่าน (pass band) และแถบหยุด (stop band) ตามลำดับ ถ้ากำหนดให้  $A_p$  เป็นค่าขนาดของลูกคลื่นในแถบผ่าน  $A_s$  เป็นค่าลดทอนในแถบหยุดน้อยที่สุด และ  $B_p$  เป็นค่าความกว้างของแถบเปลี่ยนสถานะจะเขียนได้ดังนี้

$$A_p = 20 \log \left[ \frac{1 + \delta_p}{1 - \delta_p} \right]$$

$$A_a = -20 \log \delta_s$$

$$B_t = \omega_a - \omega_p$$

ถ้าสมมติให้  $A_p'$  และ  $A_a'$  เป็นค่าขนาดของลูกคลื่นในแถบผ่าน และให้ค่าลดทอนในแถบหยุดน้อยที่สุดขั้นตอนการออกแบบมีดังนี้

- 1.) หาผลตอบสนองอิมพัลส์  $h(n)$  โดยประยุกต์ใช้โปรแกรมฟูรีเยอร์ กับผลตอบสนองออดมคคคือ

$$h_c(\omega) = 1 \quad ; \quad |\omega| < \omega_c$$

$$= 0 \quad ; \quad \omega_c < |\omega| < \frac{\omega_s}{2}$$

โดย  $\omega_c = \frac{\omega_p + \omega_s}{2}$

- 2.) เลือกพารามิเตอร์จาก  $A_p < A_p'$  และ  $A_a > A_a'$  โดยเลือก

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

- 3.) คำนวณหา  $A_a$  ตามค่า  $\delta = \delta_p = \delta_s$

- 4.) เลือกพารามิเตอร์  $\alpha$

$$\alpha = 0 \quad \text{เมื่อ } A_a < 21$$

$$= 0.5842(A_a - 21)^{0.4} + 0.07886(A_a - 21) \quad \text{เมื่อ } 21 < A_a < 50$$

$$= 0.1102(A_a - 8.7) \quad \text{เมื่อ } A_a > 50$$

- 5.) เลือกพารามิเตอร์  $D$  ให้สอดคล้องกับ

$$D = 0.9222 \quad \text{เมื่อ } A_a < 21$$

$$= \frac{A_a - 7.95}{14.36} \quad \text{เมื่อ } A_a > 21$$

- 6.) คำนวณค่า  $N$  ที่เป็นเลขคี่ที่ต่ำสุด

$$N \geq \frac{\omega_s D}{B_t} + 1$$

- 7.) คำนวณ  $h(n) = h_d(n) w_k(n)$  สำหรับ  $n \leq \frac{(N-1)}{2}$

- 8.) ฟังก์ชันถ่ายโอนของตัวกรองเขียนเป็น

$$H(\omega) = h(0) + 2 \sum_{n=1}^{(N-1)/2} h(n) \cos 2n\omega t$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 การคำนวณฟิลเตอร์บน TMS

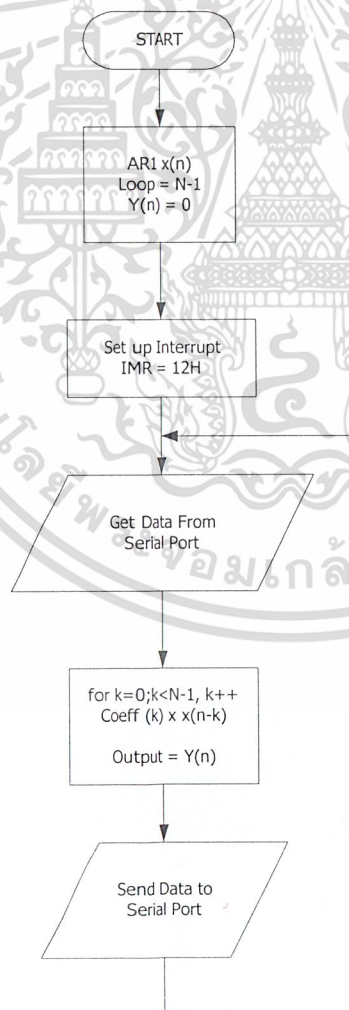
เนื่องจากโปรแกรมบน TMS ต้องเป็นแอสเซมบลี ของตัว TMS320C542 ซึ่งโปรแกรมจะกำหนดไว้เลยโดยเป็นการคำนวณหาคอนโวลูชัน จากสมการ

$$y(i) = \sum_{k=0}^{N-1} b_k x(i-k) \quad \text{หรือ}$$

$$y(i) = b_0 x(i) + b_1 x(i-1) + b_2 x(i-2) + b_3 x(i-3) + \dots + b_{N-1} x(i-(N-1))$$

โดยที่  $b_0, b_1, \dots, b_{N-1}$  เป็นสัมประสิทธิ์ของวงจรกรอง ซึ่งคำนวณมาจาก PC ในการเปลี่ยนแปลงรูปแบบของตัวกรองก็เปลี่ยนแต่ค่าสัมประสิทธิ์เท่านั้น ซึ่งจะทำให้การเปลี่ยนได้บน PC เป็นการง่ายต่อการใช้งาน ส่วนสัญญาณ อินพุต  $x(i-k)$  จะรับมาจาก TLC320AC01C ซึ่งเป็นวงจรการเชื่อมต่อแบบอนาลอก โดยกำหนดความถี่แซมปลิง ได้ ( $\leq 19.2\text{kHz}$ )

สำหรับโปรแกรมที่ใช้คำนวณแสดงเป็นโฟลว์ชาร์ต ได้ดัง รูปที่ 3-4



รูปที่ 3-4 แสดงโฟลว์ชาร์ตของโปรแกรมที่ใช้คำนวณตัวกรองบน TMS

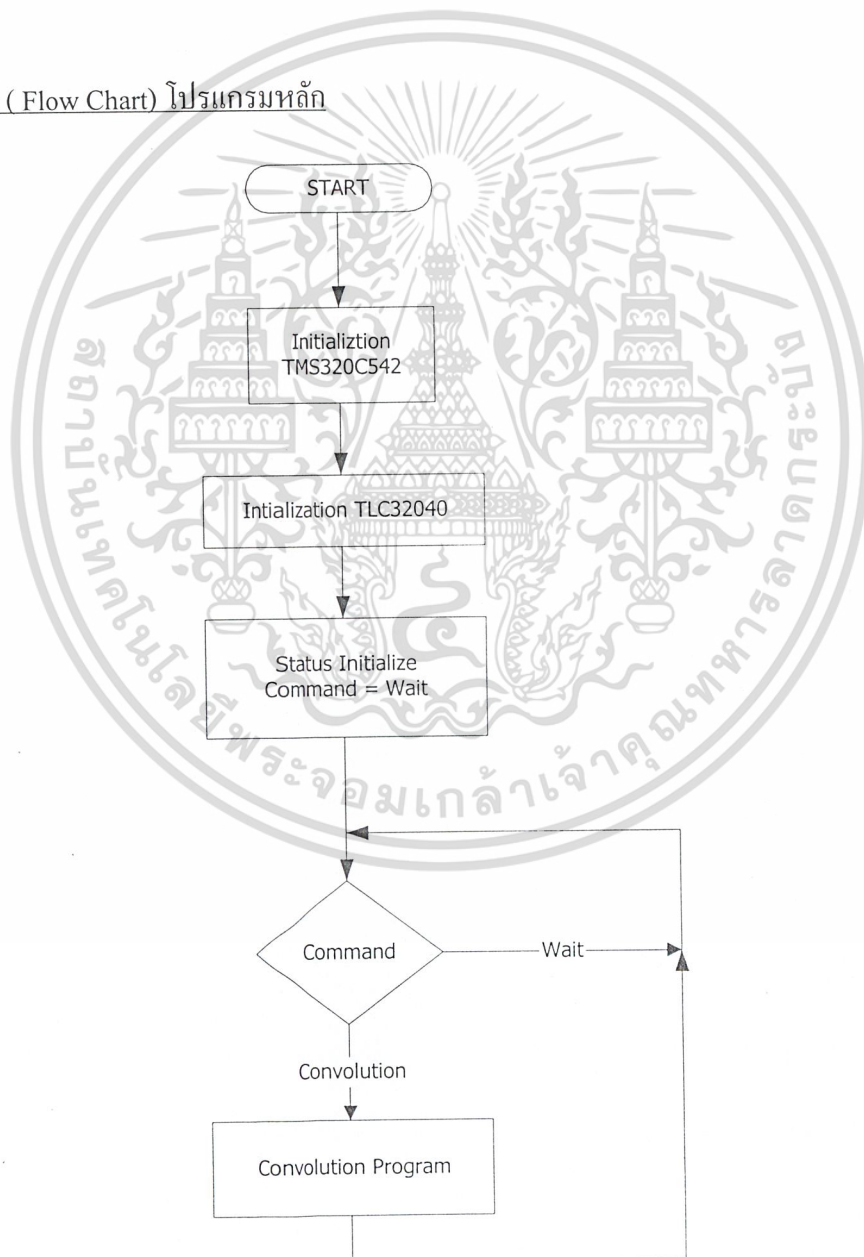
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของโปรแกรมบน TMS320C542 สามารถแบ่งได้ดังนี้

- 1) โปรแกรมการกำหนดค่าเริ่มต้น ( Initialization Processor) TMS320C542
- 2) โปรแกรมการกำหนดค่าเริ่มต้น ( Initialization) TLC320AC01C
- 3) โปรแกรมคอนโวลูชัน
- 4) โปรแกรมรับ/ส่ง ข้อมูล Parallel Port
- 5) โปรแกรมควบคุมทดสอบคำสั่ง ( Test Command)

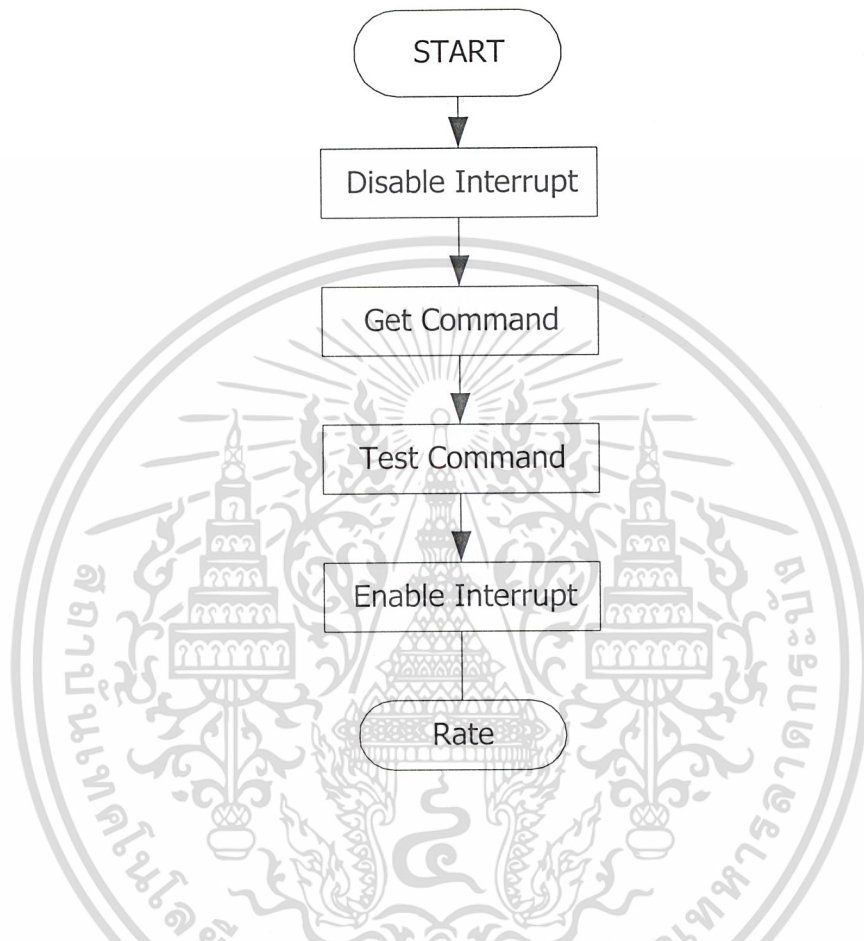
จากโปรแกรมทั้งหมด สามารถนำมาเขียนเป็น โปรแกรมหลักและอธิบายได้ดังนี้

### โฟลว์ชาร์ต ( Flow Chart) โปรแกรมหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

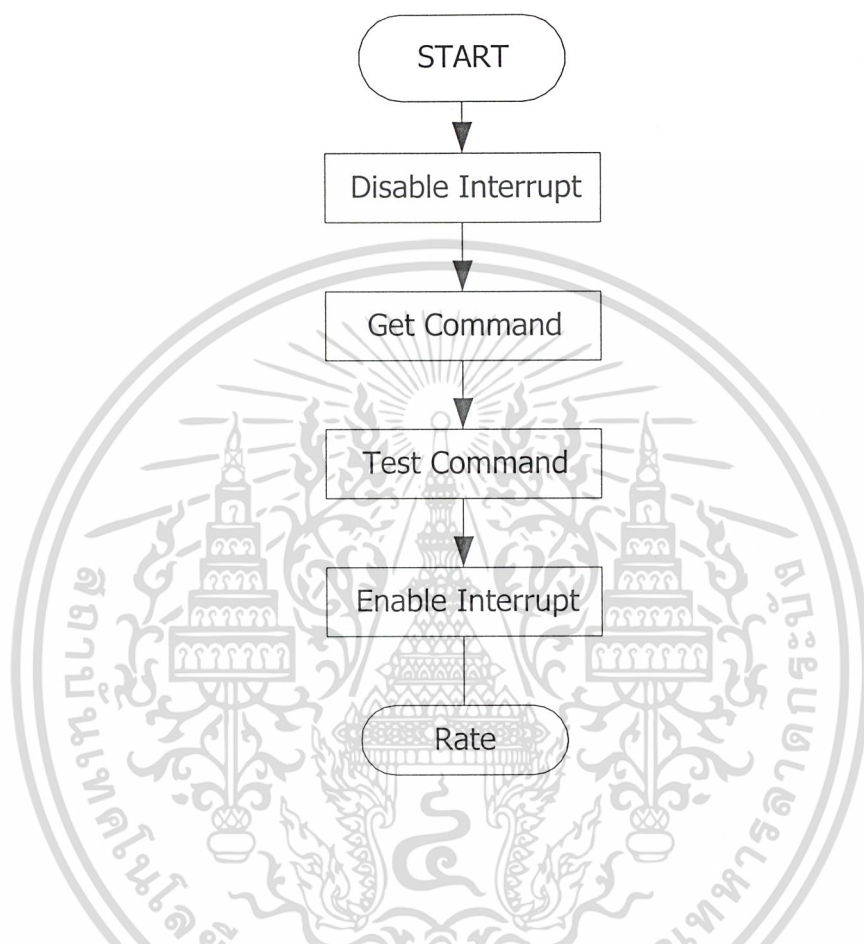
## โปรแกรมบริการอินเตอร์รัพต์



เนื่องจากใช้อินเตอร์รัพต์ในการติดต่อกับพอร์ตอนุกรม ของคอมพิวเตอร์ผ่านทาง RS232 ซึ่งเราจะใช้ อินเทอร์รัพต์ หมายเลข 1 (INT1) จะเป็นตัวคอยรับข้อมูลมา ทั้งโปรแกรมหลักและโปรแกรมบริการอินเตอร์รัพต์ จะอธิบายดังนี้คือ เมื่อเริ่มโปรแกรม จะทำการกำหนดค่าเริ่มต้นข้อมูลบน TM320C542 ที่จำเป็นในการใช้ แล้วจะเซตค่าที่กำหนดการใช้งานให้กับ TLC320AC01C ซึ่งเป็น IC ที่ทำหน้าที่แปลงสัญญาณอนาลอก เป็น ดิจิตอล (A/D) และแปลงสัญญาณดิจิตอล เป็นอนาลอก (D/A) และกำหนดอัตราการแซมปลิง เนื่องจากต่อมาเราต้องการทำคอน โวลูชันเพื่อทำตัวกรองแก้สัญญาณที่เข้า และส่งกลับไปอีกที ข้อมูลหรือสัมประสิทธิ์ต่างๆ นั้นจะรับมาจาก PC ดังนั้นจึงจำเป็นต้องรอเพื่อทดสอบคำสั่งว่าครบหรือพร้อมหรือยัง ในส่วนของ โปรแกรมอินเทอร์รัพต์จะเป็นการรับข้อมูลจาก PC มาเก็บใช้เป็นค่าสัมประสิทธิ์ ซึ่งจะอธิบายละเอียดตามหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

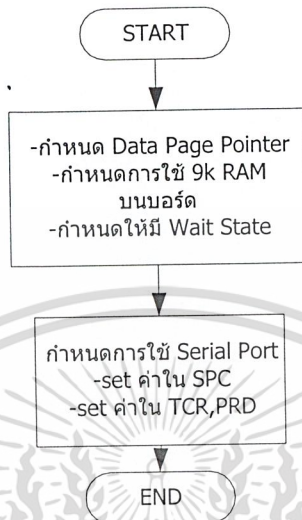
## โปรแกรมบริการอินเตอร์รัพต์



เนื่องจากใช้อินเตอร์รัพต์ ในการติดต่อกับพอร์ตอนุกรม ของคอมพิวเตอร์ผ่านทาง RS232 ซึ่งเราจะใช้ อินเทอร์รัพ หมายเลข 1 (*INT1*) จะเป็นตัวคอยรับข้อมูลมา ทั้งโปรแกรมหลักและโปรแกรมบริการอินเตอร์รัพ จะอธิบายดังนี้คือ เมื่อเริ่ม โปรแกรม จะทำการกำหนดค่าเริ่มต้นข้อมูลบน TM320C542 ที่จำเป็นในการใช้ แล้วจะเซตค่าที่กำหนดการใช้งานให้กับ TLC320AC01C ซึ่งเป็น IC ที่ทำหน้าที่แปลงสัญญาณอนาลอก เป็น ดิจิตอล (A/D) และแปลงสัญญาณดิจิตอล เป็นอนาลอก (D/A) และกำหนดอัตราการแซมปลิง เนื่องจากต่อมาเราต้องการทำคอนโวลูชันเพื่อทำตัวกรองแก้สัญญาณที่เข้า และส่งกลับไปอีกที ข้อมูลหรือสัมประสิทธิ์ต่างๆ นั้นจะรับมาจาก PC ดังนั้นจึงจำเป็นต้องรอเพื่อทดสอบคำสั่งว่าครบหรือพร้อมหรือยัง ในส่วนของ โปรแกรมอินเตอร์รัพต์จะเป็นการรับข้อมูลจาก PC มาเก็บ ใช้เป็นค่าสัมประสิทธิ์ ซึ่งจะอธิบายละเอียดตามหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. โปรแกรมการกำหนดค่าเริ่มต้น (Initialization) TMS320C542



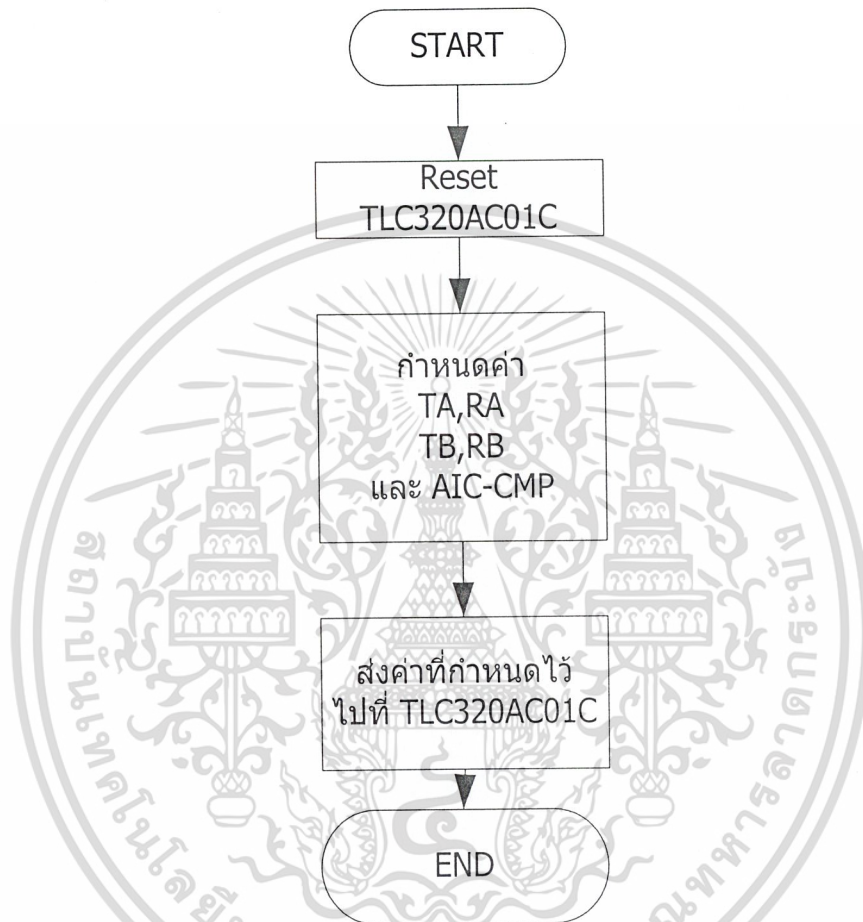
ซึ่งสามารถเขียน โปรแกรมหลักได้ดังนี้

```

; TMS320C542 INITIALIZATION ;
; This routine initializes the C5x register, internal RAM and ;
; external RAM from xxxx to FFFF ;
;-----;
        .mmregs
start:   intm = 1           ; disable all interrupts
        dcall AC01INIT    ; initialize Analog interface.
        DP = #0
        Nop
        Pmst = #01a0h     ; Interrupt pointer maps vectors to address 180h
        Sp = #0ffah       ; stack pointer located in Communications Kernal
        Imr = #240h       ; unmask TDM RINT and HPIINT (host port interface)
        Intm = 0          ; enable all interrupts
  
```

WAIT: goto WAIT ; wait for receive interrupt.  
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. โปรแกรมการกำหนดค่าเริ่มต้น (Initialization) TLC320AC01C



ซึ่งสามารถเขียน โปรแกรมหลักได้ดังนี้

```

;-----;
;          Set TLC320AC01C          ;
;-----;

```

AC01INIT:

```

Xf=0          ; reset ac01
Intm = 1      ; disable all int service routines
Tcr = #10h    ; stop timer
Imr = #280h   ; unmask TXINT and HPIINT
Tspc = #0008h ; stop TDM serial port

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

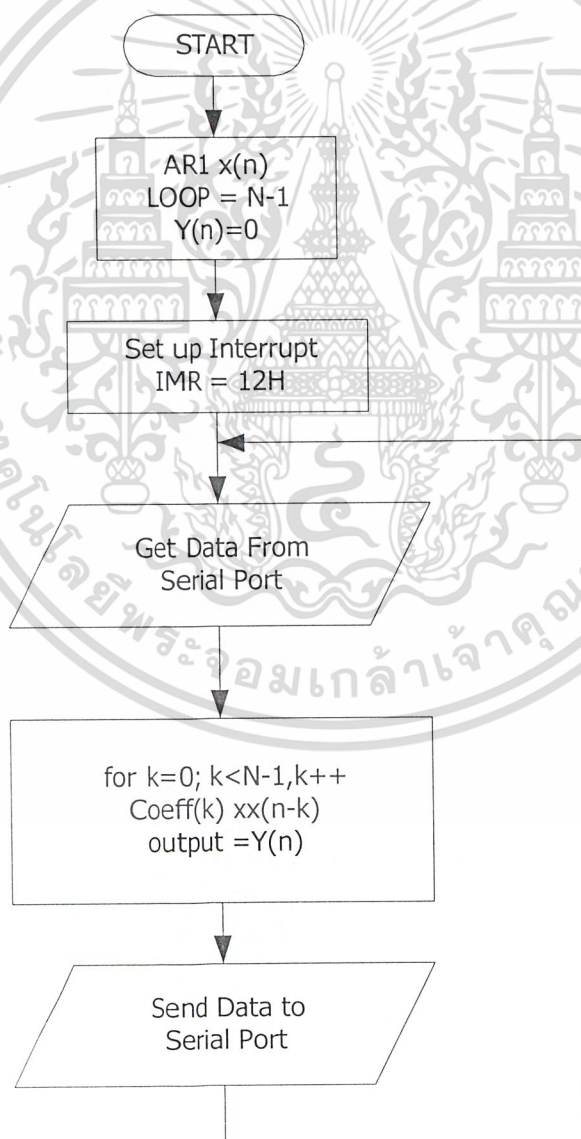
```

Tdxr = #0h           ; send 0 as first xmit word
Tspc = #00c8h       ; reset and start TDM serial port
Xf = 1               ; release ac01 from reset

```

-----;

3) โปรแกรมการคอนโวลูชัน  
สำหรับโปรแกรมที่ใช้คำนวณแสดงเป็นพล็อตชาร์ต ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4) โปรแกรมรับส่งผ่าน RS232

ในการส่งข้อมูลทาง PC จะกำหนดรูปแบบดังนี้

- ใช้พอร์ต ขนาน

รูปแบบของข้อมูล

START	D0	D1	D2	D3	D4	D5	D6	D7	STOP	STOP
BIT										

จะใช้บิตเริ่มต้นเป็นสถานะต่ำ เป็นการเปิดอินเทอร์รัพ เบอร์ INT1 ของ TMS320C542 การกำหนด baud rate จะคำนวณทาง TMS โดยการเก็บค่าลงใน ARS Baud rate = 9600 bit/sec ซึ่ง 1 bit = 104.1666  $\mu$ sec จะได้ 8 bit เป็น 833.333  $\mu$ sec เราจะใช้คำสั่ง NOP เพื่อกำหนดค่าจำนวน ARS คำสั่ง NOP 1 คำสั่ง ใช้เวลาทำงาน 50 nsec เพราะฉะนั้น 833.333  $\mu$ sec ใช้ทำงานคำสั่ง NOP เป็น 16,666 คำสั่ง AR5 จะเก็บค่า 1 bit ดังนั้นจะได้ ARS = 823H

ในการติดต่อ RS232 TMS320C542 จะใช้ขา BIO เป็นตัวรับ และขา XF เป็นตัวส่ง ในการใช้งานถ้าส่ง "1" จะทำการ setc แต่ถ้าส่ง "0" จะทำการ clrc บิตนั้นๆ ซึ่งขา BIO และ XF จะใช้ควบคุมระดับบิตได้

#### 5) โปรแกรมควบคุมทดสอบคำสั่ง (Test Command)

เนื่องจากจะติดต่อกับ TMS320C542 เพื่อทำงานในขั้นตอนต่างๆ โดยจะกำหนดดังนี้

- Command 01h

เป็นการส่งค่า A, B, โดย PC จะส่งค่าดังกล่าวให้แก่ TMS320C542 เมื่อมีการ

เปลี่ยน sampling rate

- Command 02h

เป็นการส่งค่าสัมประสิทธิ์ให้กับ TMS โดยจะส่งค่า N และ Coeff ที่คำนวณจาก PC โดย

จะเป็นการ down load ส่งไปที่ TMS ตามแอดเดรส ที่กำหนด

- Command 05h

เพื่อเป็นการเริ่มทำการคอนโวลูชัน เมื่อรับค่า N, Coeff และ A, B เรียบร้อย

แล้ว

- Command 07h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อการรับส่ง A, B และ N, Coeff ยังไม่สมบูรณ์จะต้องรอจนกว่าเสร็จสิ้นการ  
บอกให้รอ (wait)

- Command 08h หรืออื่นๆนอกจากที่กำหนดมา  
เมื่อมีการทดสอบคำสั่ง (Test command) ไม่ตรงกับค่าก่อนหน้าแล้ว แสดงว่า  
TMS320C542 จะผิดพลาด คือ โปรแกรมที่ได้กำหนดนั้นออกจากโปรแกรม จำเป็นต้องมีการรีเซต  
ใหม่ โดยจะกระโดดไปที่แอดเดรสเริ่มต้น 0800h

ในการทดสอบคำสั่ง จะตรวจสอบเมื่อมีการรับส่งครั้งแรกโดยรูปแบบ (Form) ของการรับ  
ส่ง ผลของโปรแกรมต่างๆจะขึ้นต้นด้วย command แล้วตามด้วยข้อมูลที่ Command นั้นระบุ

### 3.3 การออกแบบ FIR Filter ที่ใช้ในโครงการ

ในการออกแบบจะกำหนดฟังก์ชันนวินโดว์ (window function) เพื่อให้มีประสิทธิภาพ  
ในการกำจัดช่วงผ่านแถบความถี่ (Bandpass) ให้ต่ำลง มีวินโดว์ที่ใช้มีดังนี้

- หน้าต่างสี่เหลี่ยม (Rectangular window)
- หน้าต่างบาร์ตเลต (Bartlett window)
- หน้าต่างแฮมมิง (Hamming window)
- หน้าต่างแฮนนิง (Hanning window)
- หน้าต่างแบล็คแมน (Blackman window)
- หน้าต่างไคเซอร์ (Kaiser window)

ซึ่งแต่ละฟังก์ชันวินโดว์ (window function) จะมีขนาดสำหรับการลดทอน

สัญญาณช่วงหยุด (Bandstop) ได้ต่างกัน

ในการออกแบบจะเลือกใช้สมการคอนโวลูชัน (Convolution)

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

เมื่อ  $h(k)$  เป็นผลตอบสนองอิมพัลส์ (Impulse Response) ที่ได้จากการกำหนด  $H(e^{j\omega})$  ที่ออกแบบมา จะ  
เก็บไว้ที่หน่วยความจำข้อมูล (data memory) ของ TMS320C542 ซึ่งมี N จำนวน

$x(n)$  เป็นอินพุตที่รับมา

$y(n)$  เป็นเอาต์พุตที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากขั้นตอนการหา  $h(k)$  จะต้องหาจาก การกำหนดผลตอบสนองความถี่ ของ  
ฟิลเตอร์จากสูตร

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

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} h_d(e^{j\omega}) e^{j\omega n} d\omega$$

และนำมาคูณกับฟังก์ชันวินโดว์ ( $w(n)$ )

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

ซึ่งจะได้  $h(n)$  ทั้งหมด  $N$  จำนวน

โปรแกรมสำหรับการคำนวณ

#### 1.) โปรแกรมบน PC คำนวณหา $h(n)$

ดังที่กล่าวมาการคำนวณหา  $h(n)$  จะต้องกำหนดผลตอบสนองความถี่ก่อน  $H(e^{j\omega})$  ซึ่งจะ  
กำหนดจากโปรแกรมด้วย

- $f_c$  เป็นความถี่คัทออฟของฟิลเตอร์
- $f_s$  เป็นความถี่แซมปลิง
- วินโดว์ กำหนดฟังก์ชันวินโดว์ที่ใช้

การคำนวณหา  $h(n)$

1. ฟิลเตอร์ FIR ความถี่ต่ำ (lowpass)

$$h_d(n) = \left( \frac{2f_c}{F} \right) \frac{\sin 2\pi f_c |F|}{2\pi n f_c |F|}; n > 0$$

$$= \frac{2f_c}{F}; n = 0$$

$$f_c = \frac{1}{2} (f_p + f_s); \Delta F = f_s - f_p$$

2. ฟิลเตอร์ FIR ความถี่สูง (highpass)

$$h_d(n) = -\left(\frac{2f_c}{F}\right) \frac{\sin 2\pi n f_c / F}{2\pi n f_c / F}; n > 0$$

$$= 1 - \frac{2f_c}{F}; n = 0$$

3. ฟิลเตอร์ FIR ช่วงหยุด (stopband)

$$h_d(n) = \frac{1}{n\pi} \left[ \sin\left(\frac{2\pi n f_{c1}}{F}\right) - \sin\left(\frac{2\pi n f_{c2}}{F}\right) \right]; n > 0$$

$$= \frac{2}{F} (f_{c1} - f_{c2}) + 1; n = 0$$

4. ฟิลเตอร์ FIR ช่วงหยุด (stopband)

$$h_o(n) = \frac{1}{n\pi} \left[ \sin\left(\frac{2\pi n f_c}{F}\right) - \sin\left(\frac{2\pi n f_{c2}}{F}\right) \right]; n > 0$$

$$= \frac{2}{F} (f_{c1} - f_{c2}) + 1; n = 0$$

## 2.) โปรแกรมบน TMS320C542

ซึ่งโปรแกรมบน TMS320C542 เป็นภาษาแอสเซมบลีของ CPU นี้จะเป็นการทำการคอนโวลูชัน สัญญาณระหว่าง  $h(n)$  กับ  $x(n)$  ดังสูตร

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

จากสมการ

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \dots + h(N-1)x(n-(N-1))$$

1. จะต้องเก็บ  $h(0)$  ถึง  $h(N-1)$  ที่หน่วยความจำข้อมูล (Data memory) ใช้กำหนดเป็นที่

Coeff

2. อินพุตข้อมูล (Data input)  $x(n)$  เมื่อเริ่มโปรแกรมจะรับมา 1 ค่า

$$Y(0) = h(0)x(0) + h(1)x(-1) + \dots + h(N-1)x(-(N-1))$$

ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$x(0)$  เป็นค่าที่รับมา

$x(-1)$  เป็นค่าที่รับก่อน  $x(0)$

$x(-2)$  เป็นค่าที่รับมาก่อน  $x(-1)$

$x(-(N-1))$  เป็นค่าที่รับมาก่อน  $x(-(N-2))$

3. เมื่อรับมาเรื่อยๆ  $x(0)$  จะต้องเลื่อนมาที่  $x(-1)$  และ  $x(-1)$  จะต้องเลื่อนมาที่  $x(-2)$  ไปจนถึง  $x(-(N-2))$  เลื่อนมาที่  $x(-(N-1))$  และต่อ  $x(-(N-1))$  สุดท้ายปล่อยให้หายไป

4. แต่ละอินพุต เข้ามาจะมีการคูณกันถึง  $N$  ครั้ง และบวกกัน  $N-1$  ครั้ง เพราะฉะนั้นจะต้องใช้ความเร็วมาก ในการเขียน โปรแกรม TMS320C542 มีคำสั่งสำหรับ DSP โดยเฉพาะ



## บทที่ 4 การทดลอง

จากที่ผ่านมาเราได้ทำการ Simulate การทดลองโดยใช้โปรแกรม Matlab ส่วนในบทนี้เราจะทำการเขียนโปรแกรมลงบนบอร์ด DSP เพื่อดูผลตอบสนองต่อสัญญาณจริงๆ เพื่อเทียบกับผลการทดลอง

### 4.1 Frequency Response Of Analog Interface Chip

ก่อนทำการเขียนโปรแกรมลงบน Chip DSP เพื่อให้ทำงานเป็น Filter และทำการวิเคราะห์ผลตอบสนองในรูปแบบต่างๆ เราต้องทราบว่าที่ภาคสุดท้ายของ Digital To Analog Converter ใน IC Analog Interface (TLC320AC01c) จะมี Low-pass Filter อยู่ ทำให้การวิเคราะห์ผลตอบสนองเชิงความถี่ของ Filter ที่เราทำการออกแบบเกิดการผิดพลาดได้ ดังนั้นเราต้องหาผลตอบสนองทางความถี่ของ Filter ตัวนี้เสียก่อน โดยทำการเขียนโปรแกรมให้รับค่าเข้ามาโดยไม่ต้องทำการประมวลผลใดๆ แล้วทำการส่งออกไป โดยโปรแกรมแสดงไว้ในโปรแกรมที่ 1

โดยความถี่ Sampling จะไม่มีผลต่อ Low-pass Filter เพราะเราจะเปลี่ยนแปลงเฉพาะค่าใน Register A,B ซึ่งไม่มีผลต่อการตอบสนองต่อความถี่ของ Low-pass Filter (Switch Capacitor Filter) เพราะ Clock ที่ใช้ในส่วนนี้จะมาจากการหาร Master Clock กับค่าใน Register A , B (ดูรายละเอียดการทำงานของ TLC320AC01C Interface Analog จาก TLC320AC01C Single Supply Analog Interface Circuit Data Manual) จากนั้นทำการรันโปรแกรมที่ 1 แล้วเก็บค่าผลตอบสนองที่ความถี่ต่างๆ ซึ่งจะได้ดังตารางข้างล่าง โดย Amplitude ของ Input เท่ากับ 1.00 โวลท์

Frequency (Khz)	Amplitude (Output)
2.00	4 V
2.50	4 V
3.00	4 V
3.50	4 V
4.00	4 V
4.50	4v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.00	4V
5.50	4V
6.00	3.84 V
6.50	3.6 V
7.00	1.04 V
7.50	640 mV
8.00	400 mV

ตารางที่ 4-1 การตอบสนองความถี่

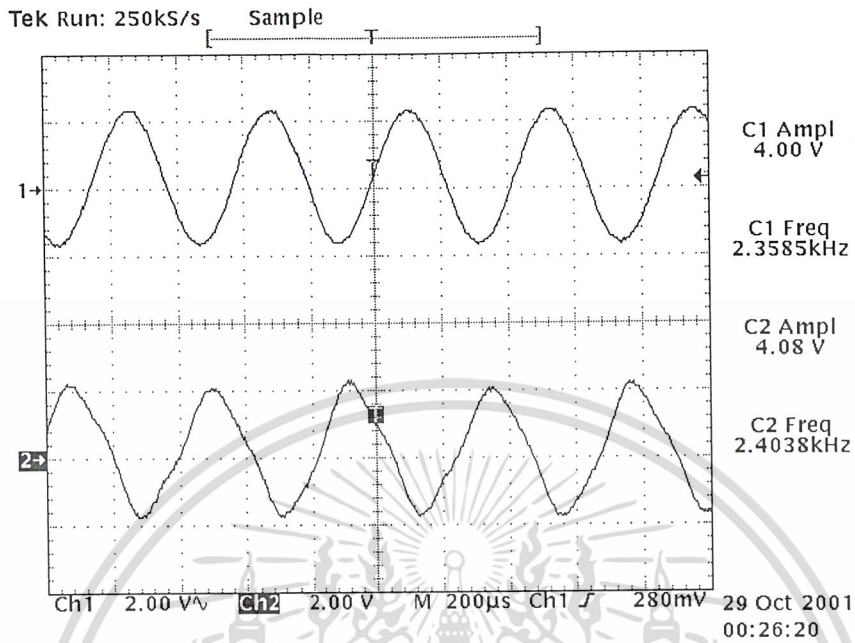
จากตารางสามารถ Plot graph ได้ดังนี้



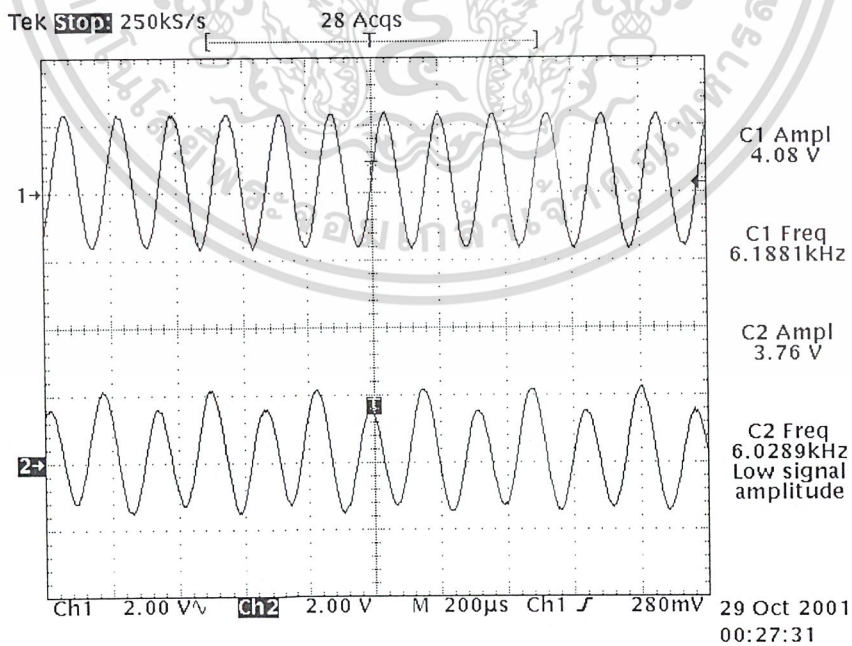
รูปที่ 4-1 กราฟแสดงการตอบสนองความถี่

จากกราฟจะเห็นว่าความถี่ Cutoff ของ Low - Pass Filter จะอยู่ที่ 6 KHz ซึ่งเวลาทดลองจะต้องอยู่ในช่วงไม่เกินความถี่นี้ อีกอย่างคือการตอบสนองเชิง Phase ของมัน ซึ่งสามารถแสดงโดยรูปของการตอบสนองที่ความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

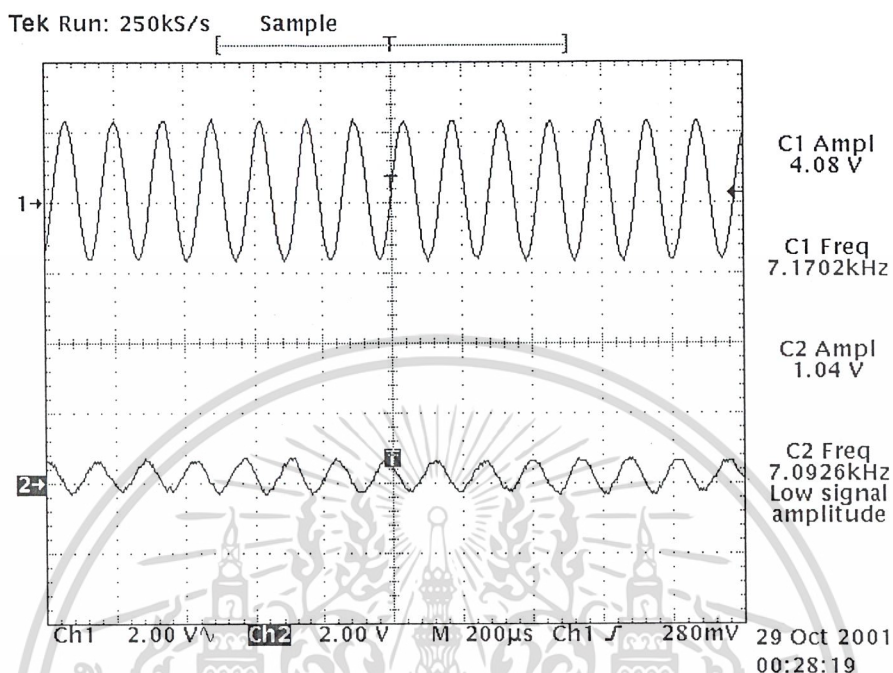


รูปที่ 4-2 แสดงสัญญาณ Input และ Output ที่ความถี่ 2.4 KHz



รูปที่ 4-3 แสดงสัญญาณ Input และ Output ที่ความถี่ 6 KHz ที่เป็นจุด Cut off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 แสดงสัญญาณ Input และ Output ที่ความถี่ 7 KHz

จากรูปแสดงถึงการตอบสนองทาง Amplitude และ Phase จะเห็นว่ามี การ Shift Phase ไปของแรงดัน Output โดยจากการทดลองเปลี่ยนค่าความถี่พบว่า จะมีการ Shift Phase ไปประมาณ 90 องศาที่ความถี่ประมาณทุกๆ 600 Hz

#### 4.2 FIR Low Pass Filter

ในขั้นตอนนี้จะนำเอาสมการของ Low – Pass Filter มาทำการเขียนโปรแกรมเพื่อดูผลการตอบสนองของ Transform Function ที่เราทำการ Simulate ในตอนแรก โดยดูผลการตอบสนองว่าจะเหมือนที่ทำการ Simulate หรือไม่

จากสมการ

$$Y(Z) = h_p X(Z) + h_1 Z^{-1} X(Z) + h_{(2)} Z^{-2} X(Z) + \dots h_{(n-1)} Z^{N-1} X(Z)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเราจะกำหนดความถี่การ Sampling เท่ากับ 8Hz (ซึ่งเรากำหนดได้จาก กำหนดจาก ค่า Register A,B ใน โปรแกรม) ซึ่งสามารถหาความถี่ Sampling ได้จากสูตร

$$f_s = \frac{f_{MCLK}}{2 \times A \times B} \quad (4.1)$$

โดย  $f_s$  คือ ความถี่ในการ Sampling

$f_{MCLK}$  คือ AIC Master Clock มีค่า 10.368 MHz AIC Shift Clock มีค่า 2.592 MHz หรือ Instruction Cycle 7.72 MHz Master Clock มีค่า 20 MHz หรือ Instruction Cycle 20 MHz

A คือ จำนวนเต็มที่มีค่าระหว่าง  $4 < A < 31$

B คือ จำนวนเต็มที่มีค่าระหว่าง  $2 < B < 63$

เพราะฉะนั้นเราทำการแทนค่าจากสูตร

$$f_s = \frac{f_{MCLK}}{2 \times A \times B}$$

$$f_s = \frac{10 \text{ MHz}}{2 \times 16 \times 20}$$

$$f_s = 8 \text{ Hz}$$

จากนั้นทำการหาค่า  $h(n)$  ซึ่งค่า

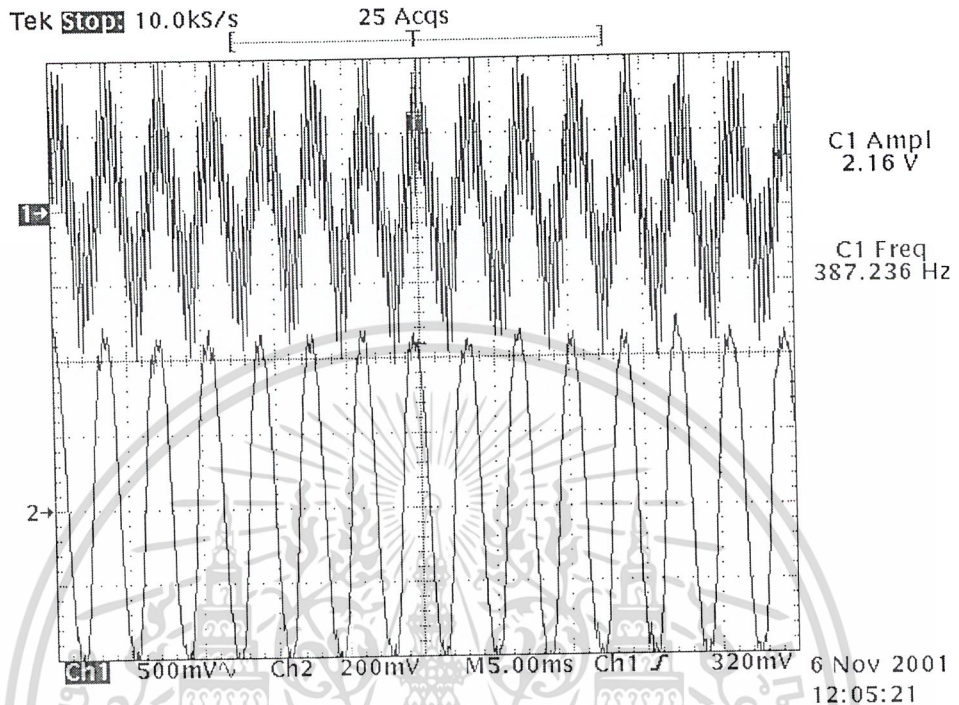
$$h(n) = w(n) * h_d(n)$$

$$h(0) = 55$$

$$h(1) = 57$$

จากนั้นนำค่าที่ได้ไปเขียนค่าสัมประสิทธิ์ของ filter ใน โปรแกรม

จากการรัน โปรแกรมจะได้ผลดังรูปข้างล่าง



รูปที่ 4-5 แสดงสัญญาณ input  $\sin 2 \text{ KHz} + \sin 283 \text{ Hz}$  และสัญญาณ output ที่  $F_c = 0.4 \text{ Hz}$  ของ FIR lowpass filter

### 4.3 FIR High Pass Filter

ในขั้นตอนนี้จะนำเอาสมการของ High pass Filter มาทำการเขียนโปรแกรมเพื่อดูผลการตอบสนองของ Transform Function ที่เราทำการ Simulate ในตอนแรก โดยดูผลการตอบสนองว่าจะเหมือนกับทำการ Simulate หรือไม่

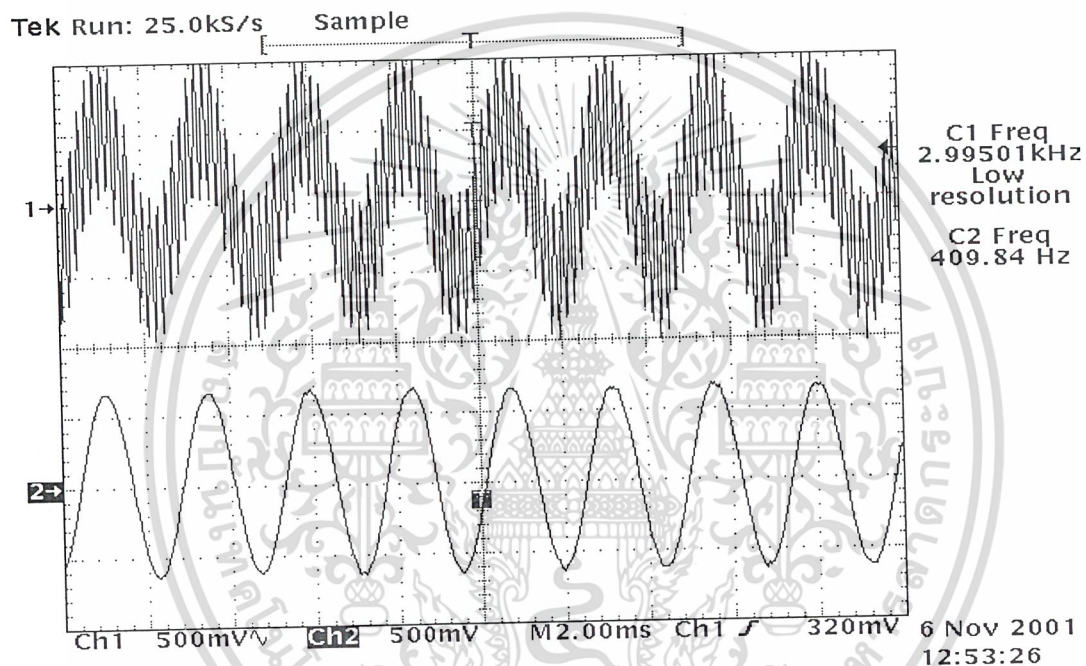
จากสมการ

$$Y(Z) = h_0X(Z) + h_1Z^{-1}X(Z) + h_2Z^{-2}X(Z) + \dots h_{(n-1)}Z^{N-1}X(Z)$$

โดยเราจะกำหนดความถี่การ Sampling เท่ากับ 8Hz (ซึ่งเรากำหนดได้จาก กำหนดจาก ค่า Register A,B ในโปรแกรม) จากนั้นทำการหาค่าสัมประสิทธิ์ของ High Pass Filter เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

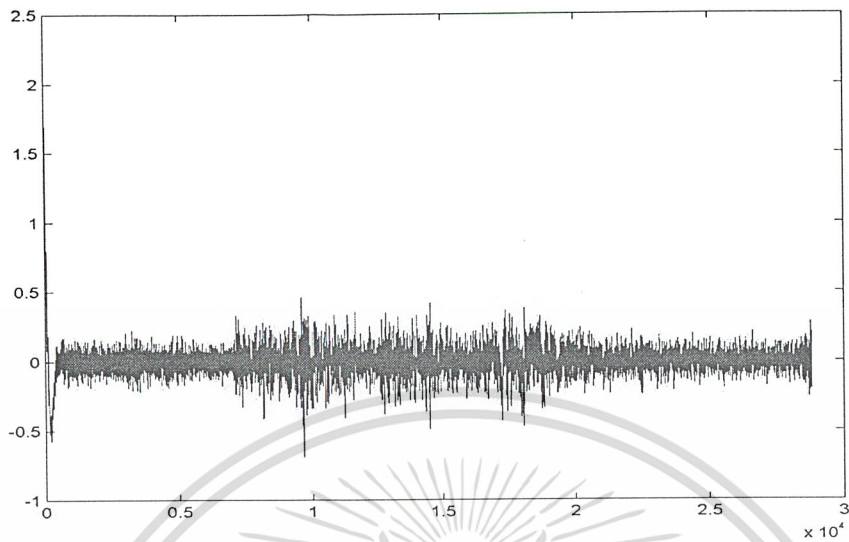
จากนั้นเรานำค่าสัมประสิทธิ์ไปเขียนลงโปรแกรมและเมื่อลองรันโปรแกรมจะได้ผลดังรูปข้างล่าง



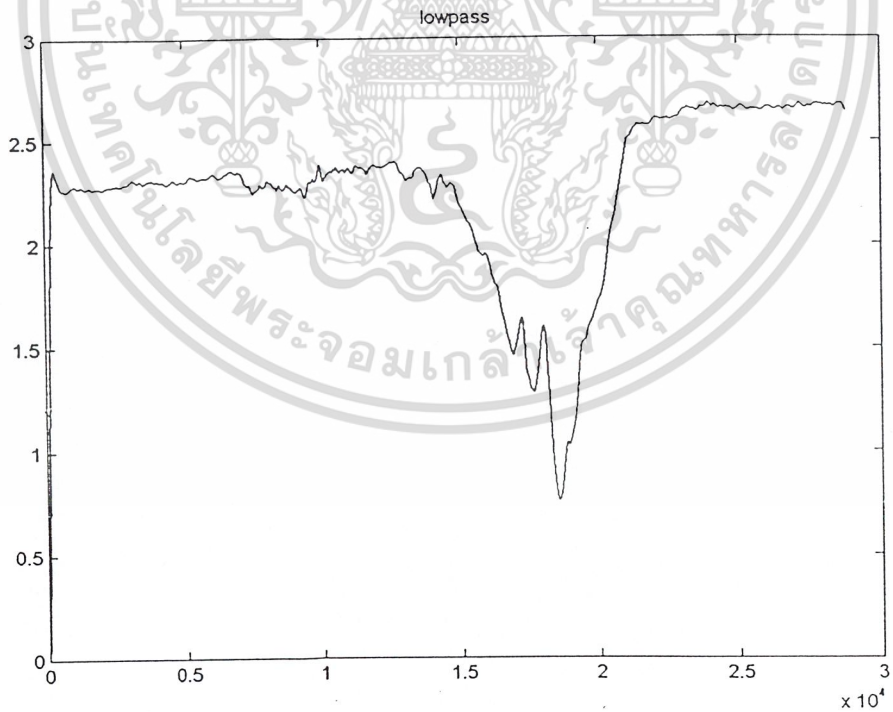
รูปที่ 4-6 แสดงสัญญาณ input ที่ ความถี่  $\sin 5 \text{ KHz} + \sin 545 \text{ Hz}$  และสัญญาณ output ที่  $F_c = 0.4$  ของ HIGH PASS FILTER

หลังจากนั้นจะนำบอร์ด DSP ไปใช้งานจริงเพื่อทำการแยก Scintillation and Rain Attenuation ออกจากสัญญาณดาวเทียม เพื่อนำไปทำการวิเคราะห์ แต่เนื่องจากสัญญาณดาวเทียมมีระดับสัญญาณที่ต่ำมากทำให้บอร์ด DSP ไม่สามารถประมวลผลได้ ดังนั้นเราต้องทำการออกแบบวงจร Amplifier เพื่อทำการขยายระดับสัญญาณซึ่งต้องการการขยายประมาณ 100 เท่า แต่ที่ทำได้ประมาณ 10 เท่า ซึ่งบอร์ด DSP ยังไม่สามารถทำงานได้ ดังนั้นเราจึงใช้โปรแกรม MATLAB แทนเพื่อทำการแยกซินทิลเลชันและเรนเอทเทนนูเอชันแทนเพื่อนำสัญญาณไปวิเคราะห์ ซึ่งจะได้ตามรูปข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 แสดง Scintillation ที่แยกออกจากสัญญาณดาวเทียม โดย High Pass Filter



รูปที่ 4-8 แสดง Rain Attenuation ที่แยกออกจากสัญญาณดาวเทียม โดย Low Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลที่ได้ในการแยก Scintillation and Rain Attenuation โดยการใช้โปรแกรม MATLAB ซึ่งจะเห็นได้ว่าโปรแกรม MATLAB สามารถแยกได้จริง Scintillation and Rain Attenuation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลอง

#### 5.1 สรุปผลการทดลอง

จากการศึกษาวิเคราะห์และออกแบบวงจรกรองความถี่ดิจิทัลแบบ FIR ทั้ง Low – Pass และ High – Pass Filter จากผลที่ Simulation จาก Computer โดยใช้โปรแกรม Matlab นั้น เมื่อเรานำมาเปรียบเทียบกับผลที่ได้จากการออกแบบโดยเขียนโปรแกรมลงใน DSP TMS320C542 จะเห็นว่าผลที่ได้จะเหมือนกัน

ดังนั้นจะเห็นได้ว่าจากที่เราทำการออกแบบผลที่ได้จะเป็นไปตามทฤษฎีทุกอย่างโดยผลที่ได้จากเขียนโปรแกรมที่ใช้ Low-Pass Filter จะแยก Scintillation ออกจากสัญญาณดาวเทียม และใช้ High – Pass Filter จะแยก Rain Attenuation ออกจากสัญญาณดาวเทียม

#### 5.2 แนวทางในการพัฒนาและประยุกต์ใช้งาน

ทำการทดลองโดยการทำการสร้างการ์ด Interface กับคอมพิวเตอร์ PC ที่มีรูปแบบการคำนวณในรูปของ 32 bits และใช้การเขียนโปรแกรมด้วยภาษาชั้นสูงในคอมพิวเตอร์ โดยเลือกใช้ D/A และ A/D ที่มี Sampling Rate สูงๆ จะสามารถทำการทดลองในเงื่อนไขต่างๆ ได้กว้างขึ้น เช่น สามารถทดลองในช่วงความถี่กว้างขึ้น มีการคำนวณในรูปแบบที่ทำการประมวลค่าน้อยลง แต่จะสามารถที่จะนำไปใช้งานจริงได้ยาก

#### 5.3 ปัญหาที่เกิดขึ้นและ แนวทางแก้ไข

ในการทดลองปัญหาที่เกิดขึ้นประการหนึ่งจะขึ้นอยู่กับอุปกรณ์ประกอบการทดลอง ซึ่งต้องใช้ความละเอียดมากในการปรับ โดยค่าที่เปลี่ยนแปลงไปในการปรับแต่ละครั้งจะมีผลกับ Output ที่ได้เสมอ อีกประการหนึ่ง การเขียนโปรแกรมที่ติดต่อกับบอร์ด DSP ต้องศึกษาในโครงสร้างของบอร์ดให้เข้าใจ อาทิ เช่น หน่วยความจำ, คำสั่งต่าง ๆ ก่อนที่จะนำไปใช้งาน ในโครงการนี้ใช้ภาษาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสเซมบลี ซึ่งยากต่อความเข้าใจ แต่จะมีข้อดีที่สามารถทำงานได้เร็ว แต่ส่วนนี้สามารถใช้ภาษาซี ซึ่งใช้กันแพร่หลาย และมีผู้เชี่ยวชาญมากกว่าทดแทนได้ โดยผู้ใช้จะสามารถ download C-Compiler พร้อมทั้ง คู่มือ User's Guide ได้จาก Web site ของบริษัท Texas Instruments

อุปสรรคที่สำคัญคือเราไม่สามารถที่จะนำสัญญาณจริงของสัญญาณดาวเทียมนำมาต่อใช้งานกับบอร์ด DSP ได้ เนื่องจากสัญญาณมีค่าระดับต่ำเกินไป ทำให้บอร์ดไม่สามารถประมวลผลได้ จึงจำเป็นต้องมีการสร้างวงจรขยายขึ้น เพื่อที่จะขยายสัญญาณให้แรงขึ้น ซึ่งจากการทดลองต้องขยายสัญญาณประมาณร้อยเท่า และต้องพิจารณาเรื่อง Impedance matching ด้วย สำหรับการพัฒนาต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมทดสอบ Lowpass Filter ของบอร์ด DSP

```
*****
; File: FirstApp.ASM -> First Application program for the 'C54x DSKplus
;
; *****

.width 80
.length 55
.title "FirstApp program"
.mmregs
.setsect ".text", 0x500,0
.setsect "vectors", 0x180,0

.sect "vectors"
.copy "c:\dskplus\inits\vectors.asm"
.text
start:
call AC01INIT
pmst = #01a0h ; set up iptr
sp = #0ffah ; init stack pointer.
ar2 = #1200h ; pointer to receive buffer.
*ar2+ = data(#0bh) ; store to rcv buffer
imr = #280h
intm = 0 ; ready to rcv int's

wait nop
goto wait

; ----- Receive Interrupt Routine -----

XINT:
b = trcv ; load acc b with input
b = #0FFCh & b
*ar2+ = data(#0bh) ; store to rcv buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tdxr = b          ; transmit the data.
TC = (@ar2 == #01280h)
if (TC) goto restrt          ; stop if rcv buffer is at 1280h
return_enable

restrt
ar2 = #1200h          ; set intm bit ...no int's
; hpic = #0ah          ; flag host task completed
return_enable
; ----- end ISR -----

.copy "c:\dskplus\inits\ac01init.asm"
.end

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE NAME: lowpass filter

```
.title "Low pass filter"

.mmregs

.width 80

.length 55

.setsect ".text",0x1800,0           ; these assembler directives specify
.setsect ".data",0x0200,1         ; the absolute addresses of different
.setsect "vectors",0x0180,0       ; sections of the assembly code
.setsect "filter",0x2700,0

.sect "filter"
.copy "lp_coeff.asm"

.sect "vectors"                   ; interrupt vector table stored at
.copy "lp_vecs.asm"              ; location 0x0180

.data                             ; delayed data stored at 0x0200
seed .word 07e6dh                 ; seed for random variable
temp .word 0
XN .word 0                        ; 80 data locations for 80
XN1 .word 0                       ; stage delay line.
XNLAST .word 0
OUTPUT .word 0                   ; extra word for the bit bucket

.text

start: intm = 1                   ; disable all interrupts
      dcall AC01INIT              ; initialize Analog interface.
      DP = #0
      nop
      pmst = #01a0h              ; Interrupt pointer maps vectors to address 180h
      sp = #0ffah                ; stack pointer located in Communications Kernal
      imr = #240h                ; unmask TDM RINT and HPIINT(host port interface)
      intm = 0                   ; enable all interrupts
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAIT:

```
goto WAIT ; wait for receive interrupt.
```

```
receive: DP = #XN ; This sets Data Memory Page Pointer  
; to page XN, which is defined  
; earlier in the program.
```

```
A =trcv ; LOAD ACCUMULATOR WITH WORD  
; RECEIVED FROM AIC!
```

```
@XN = A << 0 ; STORE THE VALUE OF RECEIVE  
; WORD TO VARIABLE XN!
```

```
AR0 = #XNLAST ; LOAD AR0 WITH ADDRESS OF LAST  
; DELAY ELEMENT!
```

```
A = #0 ; ZERO ACCUMULATOR A!
```

```
repeat(#1) ; Repeat next instructions 2 times.
```

```
macd(*AR0-,h0,A) ; Compute FIR output.
```

```
@OUTPUT = hi(A) << 0 ; Store the filtered input into  
; variable OUTPUT.
```

```
A = @OUTPUT << 0 ; OUTPUT ==>Accumulator A
```

```
A = #0FFFC & A ; TWO LSB's MUST BE ZERO FOR AIC!
```

```
tdxr = A ; SEND TO TRANSMIT REGISTER!
```

```
return_enable ; Enable interrupts and return  
; from interrupt.
```

transmit:

```
return_enable ; Enable interrupts and return  
; from interrupt.
```

```
;-----END-----
```

```
.copy "lp_ac01.asm"
```

```
nop
```

```
nop
```

```
nop
```

```
.end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FILE NAME: highpass filter

```
.title "high pass filter"

.mmregs

.width 80

.length 55

.setsect ".text",0x1800,0 ; these assembler directives specify
.setsect ".data",0x0200,1 ; the absolute addresses of different
.setsect "vectors",0x0180,0 ; sections of the assembly code
.setsect "filter",0x2700,0

.sect "filter"
.copy "lp_coeff.asm"

.sect "vectors" ; interrupt vector table stored at
.copy "lp_vecs.asm" ; location 0x0180

.data ; delayed data stored at 0x0200
seed .word 07e6dh ; seed for random variable
temp .word 0
XN .word 0 ; 80 data locations for 80
XN1 .word 0 ; stage delay line.
XNLAST .word 0 ;
OUTPUT .word 0 ; extra word for the bit bucket

.text

start: intm = 1 ; disable all interrupts
dcall AC01INIT ; initialize Analog interface.
DP = #0
nop
pmst = #01a0h ; Interrupt pointer maps vectors to address 180h
sp = #0ffah ; stack pointer located in Communications Kernal
imr = #240h ; unmask TDM RINT and HPIINT(host port interface)
intm = 0 ; enable all interrupts
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAIT:

```
goto WAIT ; wait for receive interrupt.
```

```
receive: DP = #XN ; This sets Data Memory Page Pointer  
; to page XN, which is defined  
; earlier in the program.
```

```
A =trcv ; LOAD ACCUMULATOR WITH WORD  
; RECEIVED FROM AIC!
```

```
@XN = A << 0 ; STORE THE VALUE OF RECEIVE  
; WORD TO VARIABLE XN!
```

```
AR0 = #XNLAST ; LOAD AR0 WITH ADDRESS OF LAST  
; DELAY ELEMENT!
```

```
A = #0 ; ZERO ACCUMULATOR A!
```

```
repeat(#1) ; Repeat next instructions 2 times:
```

```
macd(*AR0-,h0,A) ; Compute FIR output.
```

```
@OUTPUT = hi(A) << 0 ; Store the filtered input into  
; variable OUTPUT.
```

```
A = @OUTPUT << 0 ; OUTPUT ==>Accumulator A
```

```
A = #0FFFCh & A ; TWO LSB's MUST BE ZERO FOR AIC!
```

```
tdxr = A ; SEND TO TRANSMIT REGISTER!
```

```
return_enable ; Enable interrupts and return  
; from interrupt.
```

transmit:

```
return_enable ; Enable interrupts and return  
; from interrupt.
```

```
;-----END-----
```

```
.copy "lp_ac01.asm"
```

```
nop
```

```
nop
```

```
nop
```

```
.end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; File: AC01 Initialization Routine

.width 80

.length 55

.title "AC01 Initialization Program"

.mmregs

```
*
* 0000 0000 0000 0001 -> initialize Register 1 (A Register)
* 0000 0000 0000 0010 -> initialize Register 2 (B Register)
* 0000 0000 0000 0100 -> initialize Register 3 (A' Register)
* 0000 0000 0000 1000 -> initialize Register 4 (Amplifier Gain-Select)
* 0000 0000 0001 0000 -> initialize Register 5 (Analog Configuration)
* 0000 0000 0010 0000 -> initialize Register 6 (Digital Configuration)
* 0000 0000 0100 0000 -> initialize Register 7 (Frame-Sync Delay)
* 0000 0000 1000 0000 -> initialize Register 8 (Fram-Sync number)
*
```

REGISTER .set 1bh ; Powerup default values:

REG1 .set 124h ;\* 112h

REG2 .set 20fh ;\* 212h

REG3 .set 300h ; 300h

REG4 .set 409h ;\* 405h

REG5 .set 503h ;\* 501h

REG6 .set 600h ; 600h

REG7 .set 700h ; 700h

REG8 .set 801h ; 801h

AC01INIT:

xf = 0 ; reset ac01

intm = 1 ; disable all int service routines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tcr = #10h          ; stop timer
imr = #280h        ; unmask TXINT and HPIINT
tspc = #0008h      ; stop TDM serial port
tdxr = #0h         ; send 0 as first xmit word
tspc = #00c8h      ; reset and start TDM serial port
xf = 1             ; release ac01 from reset

```

```

; ----- Register init's -----

```

```

.eval REGISTER & 1h, SELECT ; if REG1 then include this source
.if SELECT = 1h
a = #REG1 ; load Acc A with REG1 value
call REQ2 ; Call REQ2 subroutine
.endif

.eval REGISTER & 2h, SELECT ; if REG2 then include this source
.if SELECT = 2h
a = #REG2
call REQ2
.endif

.eval REGISTER & 4h, SELECT ; if REG3 then include this source
.if SELECT = 4h
a = #REG3
call REQ2
.endif

.eval REGISTER & 8h, SELECT ; if REG4 then include this source
.if SELECT = 8h
a = #REG4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call REQ2
.endif

.eval REGISTER & 10h, SELECT ; if REG5 then include this source
.if SELECT = 10h
a = #REG5
call REQ2
.endif

.eval REGISTER & 20h, SELECT ; if REG6 then include this source
.if SELECT = 20h
a = #REG6
call REQ2
.endif

.eval REGISTER & 40h, SELECT ; if REG7 then include this source
.if SELECT = 40h
a = #REG7
call REQ2
.endif

.eval REGISTER & 80h, SELECT ; if REG8 then include this source
.if SELECT = 80h
a = #REG8
call REQ2
.endif
return

```

REQ2

```

ifr = #080h ; clear flag from IFR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tdxr = #03h                ; request secondary when AC01 starts

idle(1)                    ; wait for primary to xmit

tdxr = a                    ; send register value to serial port

ifr = #080h                ; clear flag from IFR

idle(1)                    ; wait for secondary to xmit

tdxr = #0h                  ; send neutral state in case last init

ifr = #080h                ; clear flag from IFR

idle(1)                    ; wait for neutral state to xmit

return                      ; return from subroutine

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\*

; File: LP\_VECS.ASM -> Vector Table for the 'C54x DSKplus

; \*\*\*\*\*

```
.title "Vector Table"

.mmregs
.width 80
.length 55

reset    goto #80h          ;00; RESET * DO NOT MODIFY IF USING DEBUGGER *
         nop
         nop
nmi      return_enable     ;04; non-maskable external interrupt
         nop
         nop
         nop
trap2    goto #88h          ;08; trap2 * DO NOT MODIFY IF USING DEBUGGER *
         nop
         nop
         .space 52*16      ;0C-3F: vectors for software interrupts 18-30
int0     return_enable     ;40; external interrupt int0
         nop
         nop
         nop
int1     return_enable     ;44; external interrupt int1
         nop
         nop
         nop
int2     return_enable     ;48; external interrupt int2
         nop
         nop
         nop
tint     return_enable     ;4C; internal timer interrupt
         nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

nop
brint  return_enable    ;50; BSP receive interrupt
nop
nop
nop
bxint  return_enable    ;54; BSP transmit interrupt
nop
nop
nop
trint  dgoto receive    ;58; TDM receive interrupt
nop
nop
txint  return_enable    ;5C; TDM transmit interrupt
nop
nop
int3   return_enable    ;60; external interrupt int3
nop
nop
nop
hpiint goto #0e4h        ;64; HPIint * DO NOT MODIFY IF USING DEBUGGER *
nop
nop
.space 24*16            ;68-7F; reserved area

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****
*
* File: LP_COEFF.ASM -Filter coeffs for the lowpass filter.
*
*****

```

```

lp_coeff      .sect      "filter"

TA            .word      16
TB            .word      16

AIC_CTR      .word      0ch
coeff        .word      5219

;Filter Coefficient Generator lopass @ fcut=0.4hz
h0           .word      55
h1           .word      57

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\*\*\*\*\*

\* \*  
\* File: LP\_COEFF.ASM -Filter coeffs for the highpass filter. \*  
\* \*

\*\*\*\*\*

lp\_coeff .sect "filter"

TA .word 16

TB .word 16

AIC\_CTR .word 0ch

coeff .word 5219

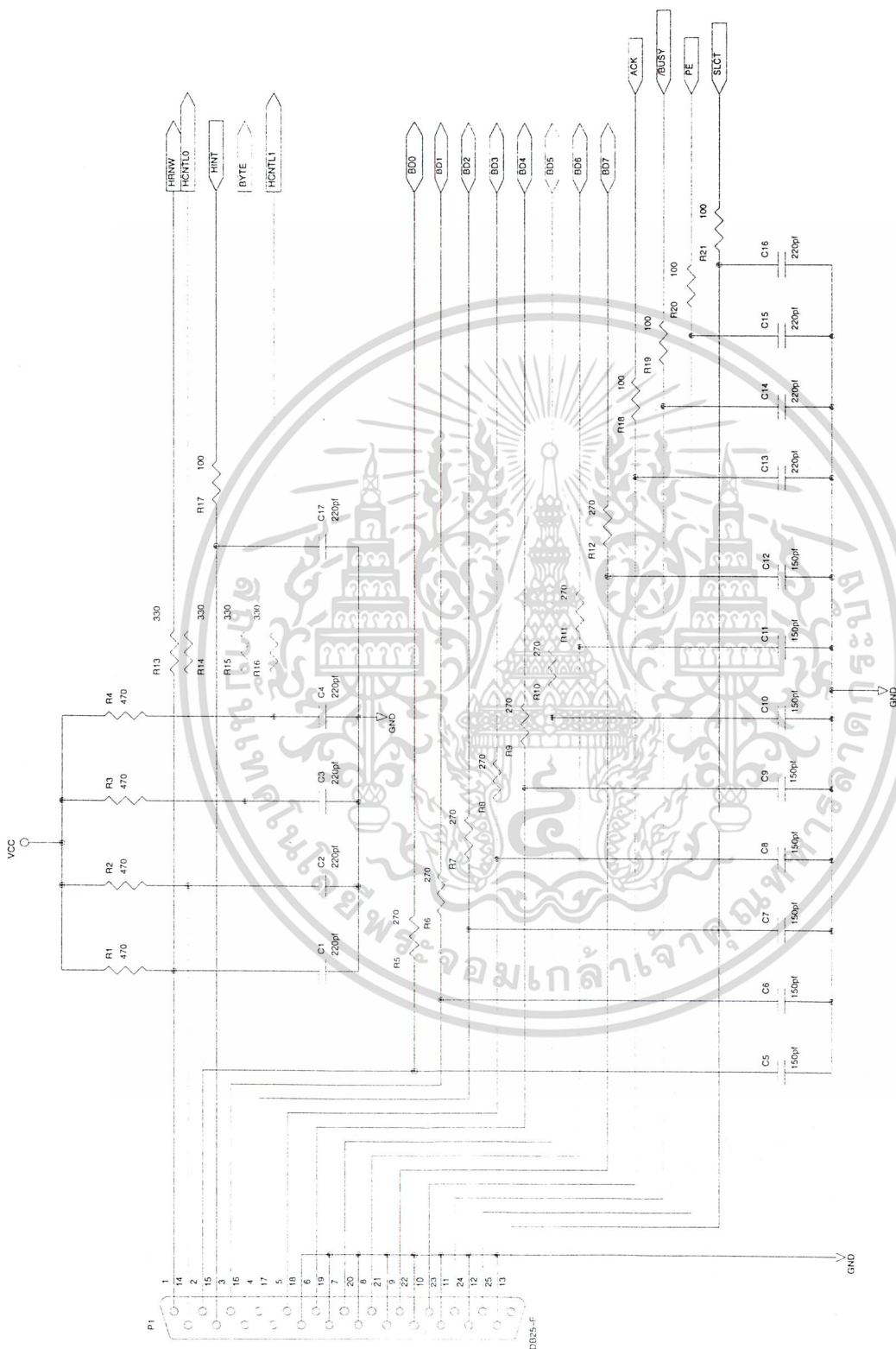
;Filter Coefficient Generator lopass @ fcut=0.4 hz

h0 .word -55

h1 .word -57

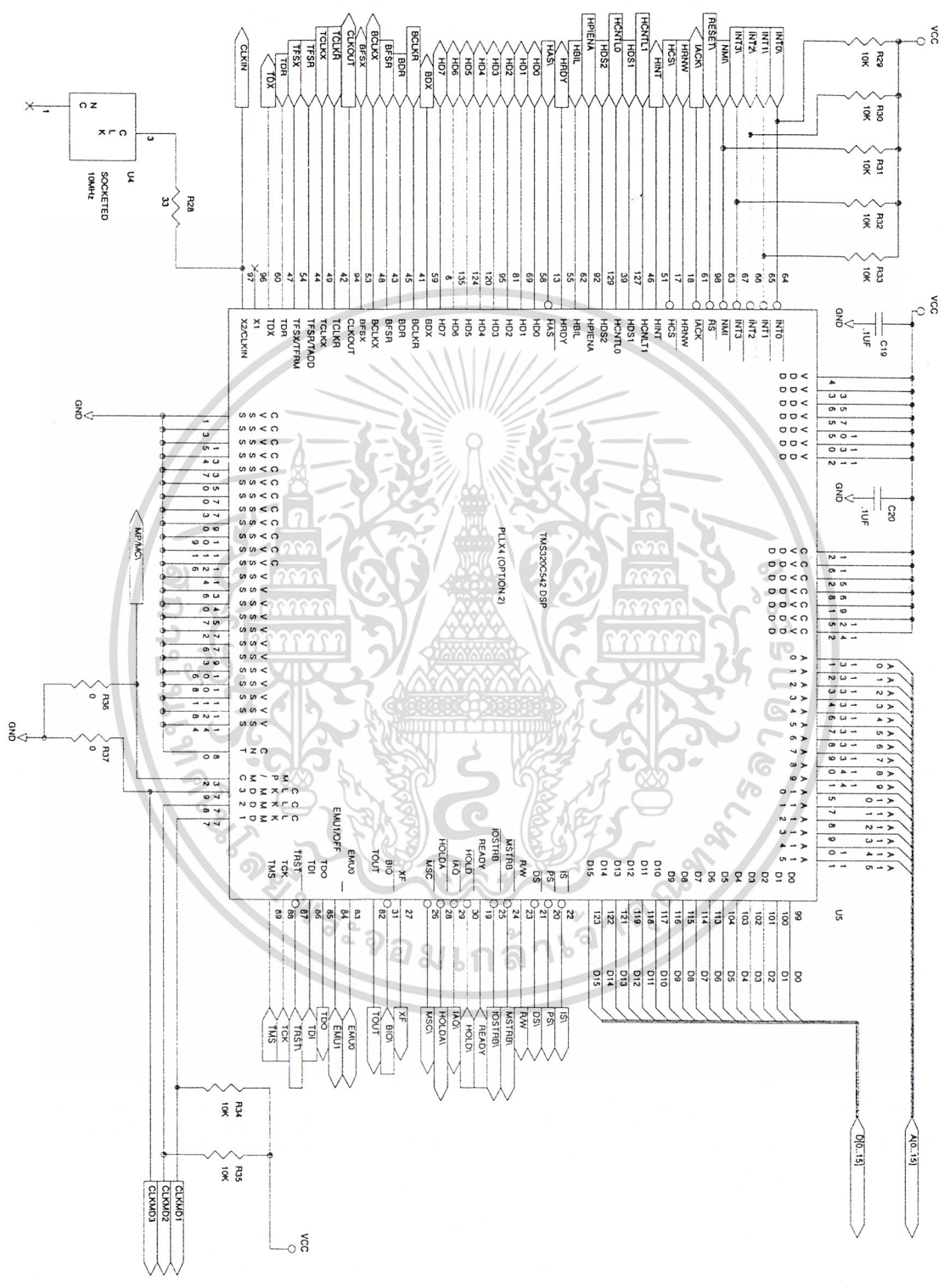


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

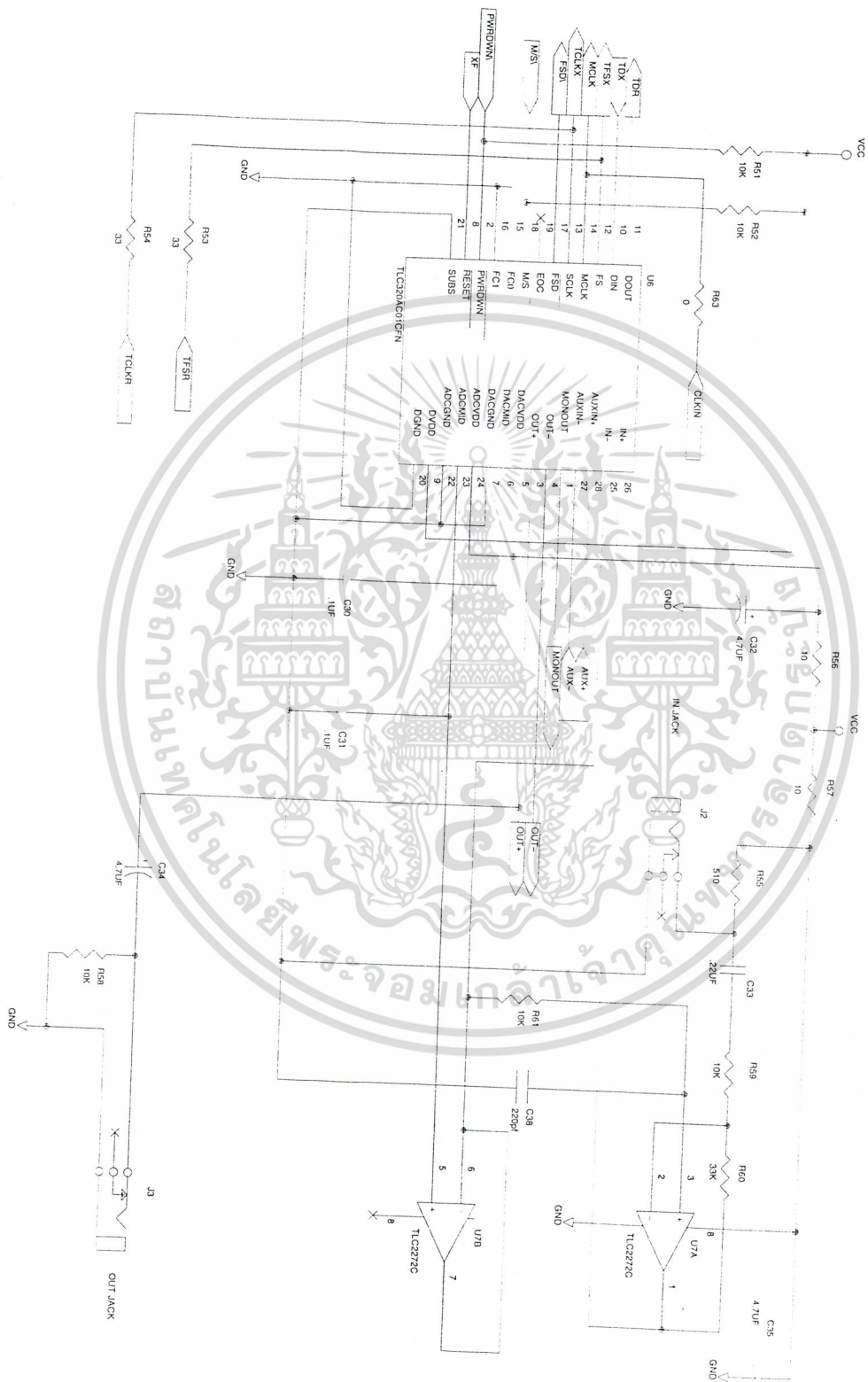


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

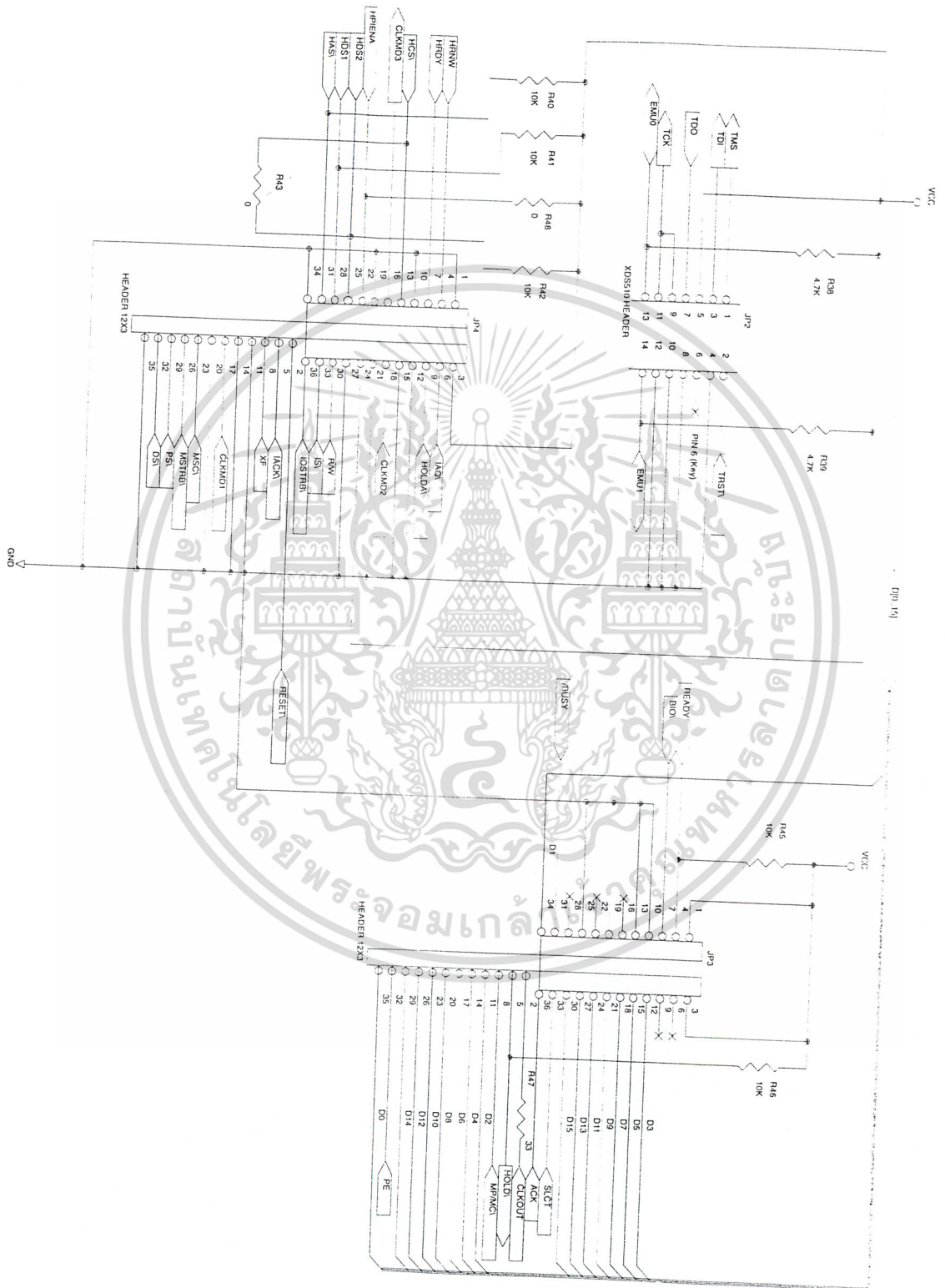




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

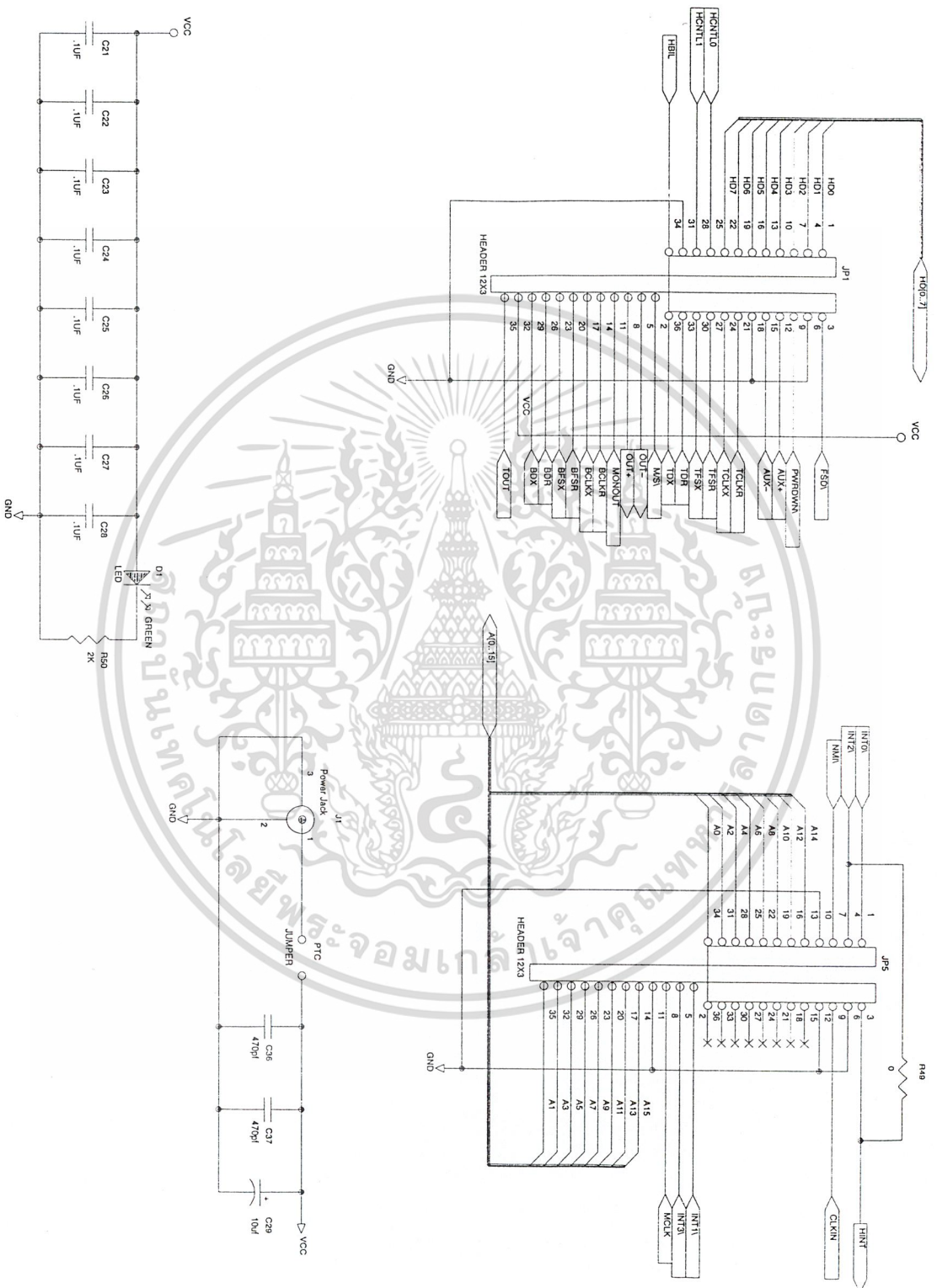


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



D10 151

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- (1) สุธรรม ศรีเกษม, "MATLAB" เพื่อการแก้ปัญหาวิศวกรรม", ศรีอนันต์การพิมพ์, กรุงเทพฯ
- (2) อรรถสิทธิ์ หล้าสกุล, "Digital Signal Processing(DSP) Fundamentals"
- (3) Texas Instruments, "TMS320C54x DSP CPU and Peripherals", Custom Printing Company  
Owensville , Missouri
- (4) Texas Instruments, " TLC320AC01C Signal – Supply Analog Interface Circuit", Custom  
Printing Company Owensville , Missouri



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้