

ชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50

EXPERIMENT FOR FIR FILTER BY DSP TMS320C50



ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

รพ.  
ท.ว.๕  
8524

เลขหม.....  
เลขทะเบียน..... 46430  
จัน, เดือน, ปี..... 1 เม.ย. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

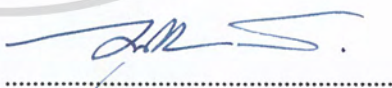
หัวข้อปริญญานิพนธ์ ชุดทดลองตัวกรองเชิงเลขแบบ เอฟไออาร์ด้วยDSP TMS320C50  
TITLE EXPERIMENT FOR FIR FILTER BY DSP TMS320C50  
นักศึกษา นายทรงฤทธิ์ ขอมสระน้อย รหัสประจำตัว 43015725  
นายเอกชัย ชันทอง รหัสประจำตัว 43015759  
อาจารย์ผู้ควบคุมปริญญานิพนธ์ อ.สมภพ แก้วมีชัย  
อ.บุญชนะ ภูระหงษ์  
ระดับการศึกษา ปริญญาอุตสาหกรรมศาสตรบัณฑิต  
ภาควิชา วิศวกรรมสารสนเทศ  
ปีการศึกษา 2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิตคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



(อ.สมภพ แก้วมีชัย)

อาจารย์ผู้ควบคุมปริญญานิพนธ์



(อ.บุญชนะ ภูระหงษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

**ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ชุดทดลองตัวกรองเชิงเลขแบบ เอฟไออาร์ด้วยDSP TMS320C50  
นักศึกษา นายทรงฤทธิ์ ขอมสรระน้อย รหัสประจำตัว 43015725  
นายเอกชัย ชันทอง รหัสประจำตัว 43015759  
อาจารย์ผู้ควบคุมปริญญานิพนธ์ อ.สมภพ แก้วมีชัย  
อ.บุญยชนะ ภูระหงษ์  
ระดับการศึกษา ปริญญาอุตสาหกรรมศาสตรบัณฑิต  
ภาควิชา วิศวกรรมสารสนเทศ  
ปีการศึกษา 2544

#### บทคัดย่อ

ระบบประมวลผลสัญญาณดิจิทัล ซึ่งพบได้กว้างขวางในการนำไปใช้ เช่น ในระบบสื่อสารข้อมูล ระบบควบคุม โซนาร์ เรดาร์ ระบบนำร่อง ระบบประมวลผลสัญญาณภาพ โมเด็ม ระบบโทรศัพท์ เป็นต้น

สำหรับโครงการนี้ ได้ศึกษาถึงทฤษฎีของระบบประมวลผลสัญญาณดิจิทัล โครงสร้างและคำสั่งที่ใช้สำหรับบอร์ดประมวลผล TMS320C50 โดยบอร์ดประมวลผลได้เพิ่มส่วนของวงจรอนาล็อกอินเทอร์เฟซ การสื่อสารกับคอมพิวเตอร์ผ่านทางพอร์ทอนุกรม RS232 ในโครงการนี้เราใช้ภาษาซีร่วมกับแอสเซมบลีในการออกแบบโปรแกรม

ในโครงการจะทำการวิเคราะห์สมการโดยใช้ MATLAB เพื่อหาค่าพารามิเตอร์ของสัญญาณ แล้วนำมาเขียนโปรแกรม สำหรับตัวกรองความถี่ต่ำผ่าน (Lowpass filter) ตัวกรองความถี่สูงผ่าน (Highpass filter) ตัวกรองความถี่แถบผ่าน (Bandpass filter) และตัวกรองความถี่แถบหยุด (Bandstop filter)

**PROJECT TITLE**      EXPERIMENT FOR FIR FILTER BY DSP TMS320C50

**STUDENT**             Mr.Songrit      Komsranoi      No. 43015725

                                 Mr.Ekkachai      Khantong      No. 43015759

**ADVISOR**             Mr.Somphop      Kaewmeechai

                                 Mr.Boonchana      Purahong

**COURE**                Bachelor of Industrial Technology in Electronics

**DEPARTMENT**        Information Engineering

**YEAR**                 2001

#### ABSTRACT

Digital signal processing are found in a wide range of applications such as in telecommunications, control system, sonar, radar, navigator system, image processing, modems, cellular phones, etc.

For this project, studied the theories of DSP system the architecture and the instructions available for the TMS320C50 processor. The DSP board includes an analog interface circuit (AIC) chip that connects to the serial port on the C50. The computer is communication are via the RS232 port . This project we use C and Assembly language for design program.

The project analyzed equation using MATLAB to determine the parameter values of simulated signal and then applied it to program with actual signal. For filter by be made to lowpass filter, highpass filter, bandpass filter and bandstop filter.

## กิตติกรรมประกาศ

โครงการชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50 เกิดขึ้นได้  
เนื่องจาก อาจารย์สมภพ แก้วมีชัย และ อาจารย์บุญณัยชนะ ภูระหงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ได้ให้คำปรึกษาและให้โอกาสในการทำโครงการชิ้นนี้ และได้อำนวยความสะดวกในการใช้เครื่องมือในการทดลองโครงการ ตลอดจนอาจารย์ทุกท่าน ที่ได้ให้คำปรึกษาและคำชี้แนะ หอสมุดกลาง  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับแหล่งศึกษาค้นคว้าข้อมูล เพื่อนๆ  
พี่ๆ ทุกคนที่ให้ความสะดวกในการทำโครงการ คุณพ่อ คุณแม่ ที่ให้การสนับสนุนมาตลอด ทาง  
ผู้จัดทำจึงขอขอบพระคุณมา ณ โอกาสนี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตของโครงการ	1
1.3 ขั้นตอนการดำเนินโครงการ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 การแปลงสัญญาณเสียงจากอนาล็อกเป็นดิจิทัล	3
2.1.1 การสุ่มสัญญาณ (sampling)	3
2.1.2 ทฤษฎีการสุ่มสัญญาณ (Sampling Theorem)	5
2.1.2.1 การใช้ตัวกรองเพื่อป้องกัน aliasing (Anti-Aliasing Filter)	6
2.1.2.2 การสุ่มโดยใช้ความถี่เกินกว่า $2f_{\max}$ มากๆ	6
2.2 ผลตอบสนองต่ออิมพัลส์ของระบบ (Impulse Response)	7
2.3 ผลการประสาน (Convolution)	8
2.4 การออกแบบโดยวิธีหน้าต่าง (Window Method)	10
2.4.1 หน้าต่างแบบสี่เหลี่ยม (Rectangular Window)	16
2.4.2 หน้าต่างฮานนิงและแฮมมิง (Hanning and Hamming Window)	18
2.4.3 หน้าต่างไคเซอร์ (Kaiser Window)	18
2.5 ระบบตัวเลขในการประมวลผล	20
2.5.1 ระบบเลขจำนวนเต็ม (Fixed-point Number System)	20
2.5.2 ระบบเลขจำนวนเต็มแบบมีเครื่องหมาย	23
2.5.2.1 แบบ sign-magnitude	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
2.5.2.2 แบบ 2's complement	23
2.6 ความคลาดเคลื่อนจากการใช้ระบบเลขจำนวนเต็ม	25
2.6.1 การแบ่งขั้นสัญญาณ (Signal Quantization)	25
2.6.2 ความคลาดเคลื่อนจากการปัดเศษสัมประสิทธิ์ (Coefficient Rounding)	30
2.7 ข้อมูลโดยทั่วไปของ TMS320C50	31
2.8 สถาปัตยกรรมของ TMS320C50	31
2.9 ลักษณะโดยทั่วไปของ TMS320C50	32
2.10 กลุ่มรีจิสเตอร์ที่สำคัญ	34
2.11 Data-Addressing Modes	36
2.12 การอินเทอร์รัพต์ (Interrupt)	37
2.13 ลักษณะทางกายภาพของบอร์ดทดลอง DSK TMS320C50	38
2.14 ดีบั๊กเกอร์ของบอร์ดทดลอง DSP TMS320C50 (DSK Debugger)	38
2.15 การใช้งาน DSK TMS320C50 Debugger	39
2.16 การโปรแกรมภาษาแอสเซมบลีบน TMS320C50	39
2.16.1 Accumulator (ACC) และ Accumulator buffer (ACCB) Register กับการบวกเลข	39
2.16.2 Temporary และ Product Register กับการคูณ	41
2.16.3 การทดลองโดยใช้ คำสั่ง MAC และ RPT	42
2.16.4 Auxiliary Register (AR 0..7) กับการเก็บค่าการนับ (Loop Counter)	43
2.16.5 การใช้งาน Circular Buffer	44
2.17 สรุปรูปแบบการเขียน โปรแกรมภาษาแอสเซมบลีบน TMS320C50	47
2.18 การโปรแกรมภาษา C บน TMS320C50	48
บทที่ 3 การออกแบบ	52
3.1 การใช้งานจอแอลซีดี (LCD)	52
3.1.1 การเชื่อมต่อและควบคุม	52
3.1.2 คำสั่งควบคุมการทำงานของแอลซีดี	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
3.2 การออกแบบการเชื่อมต่อกับ Key Board	57
3.3 การออกแบบตัวกรองแบบ FIR โดยวิธีหน้าต่างแบบต่างๆ	58
3.3.1 หน้าต่างแบบสี่เหลี่ยม (Rectangular Window)	58
3.3.2 หน้าต่างแบบแฮมมิง (Hamming Window)	66
3.3.3 หน้าต่างแบบแบล็กแมน (Blackman Window)	70
3.4 การออกแบบคอนโวลูชัน	74
3.5 การออกแบบโปรแกรมเพื่อหาค่าสัมประสิทธิ์ของผลตอบสนอง $h(n)$	76
3.6 การออกแบบโปรแกรมในการประมวลผลตัวกรองและการควบคุมอุปกรณ์ แอลซีดี และ คีย์บอร์ด	84
บทที่ 4 ผลการทดลอง	86
4.1 ผลการทดลองการกรองความถี่แบบต่างๆ	86
4.1.1 การกรองความถี่ต่ำผ่าน	86
4.1.2 การกรองความถี่สูงผ่าน	89
4.1.3 การกรองความถี่แถบผ่าน	91
4.1.4 การกรองความถี่แถบหยุด	93
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	97
บรรณานุกรม	
ภาคผนวก ก โปรแกรมในการออกแบบฟิลเตอร์	
ภาคผนวก ข รูปวงจรและส่วนต่างๆ ของบอร์ด	

## สารบัญรูป

	หน้า
รูปที่ 2.1 วงจรการสุ่มด้วยสวิตช์อุดมคติ	3
รูปที่ 2.2 สัญญาณต่าง ๆ ในการสุ่มทั้งภาคเวลาและความถี่	5
รูปที่ 2.3 แสดงสัญญาณในเชิงความถี่ที่เกิดขึ้นเมื่อใช้ $f_s$ ต่ำกว่า $2f_{max}$	6
รูปที่ 2.4 ผลตอบสนองเชิงความถี่ของตัวกรองป้องกัน aliasing	6
รูปที่ 2.5 สเปกตรัมของสัญญาณที่เกิดขึ้นเมื่อสุ่มด้วยความถี่สูงกว่า $2f_{max}$	7
รูปที่ 2.6 แสดงการคอนโวลูชันของสัญญาณ	10
รูปที่ 2.7 ผลตอบสนองเชิงความถี่ของตัวกรองอุดมคติ	11
รูปที่ 2.8 ค่าต่าง ๆ ในการระบุคุณลักษณะเฉพาะของผลตอบสนองเชิงความถี่ ของตัวกรอง FIR	15
รูปที่ 2.9 ผลลัพธ์ที่ได้จากตัวอย่างที่ 2.2	17
รูปที่ 2.10 ลักษณะพิเศษในการออกแบบ หน้าต่างโคเชอร์	19
รูปที่ 2.11 แสดงการแบ่งขั้นสัญญาณ	26
รูปที่ 2.12 แสดงค่า $SNR_{AD}$ ในฟังก์ชันของจำนวนบิต (B) และ loading factor (LF)	29
รูปที่ 2.13 แสดงขาต่าง ๆ ของ TMS320C50	33
รูปที่ 2.14 แสดงบล็อกไดอะแกรมของ TMS320C50	35
รูปที่ 2.15 แสดงลักษณะของบอร์ดทดลอง DSK TMS320C50	38
รูปที่ 2.16 แสดงหน้าต่างของ DSK TMS320C50 DEBUGGER	38
รูปที่ 3.1 การเชื่อมต่อ LCD ไปใช้งาน	53
รูปที่ 3.2 การต่อคีย์บอร์ดร่วมกับ TMS320C50	57
รูปที่ 3.3 ผลตอบสนองเชิงความถี่ต่ำของ FIR โดยวิธี หน้าต่างแบบสี่เหลี่ยม	59
รูปที่ 3.4 ผลตอบสนองเชิงความถี่สูงของ FIR โดยวิธี หน้าต่างแบบสี่เหลี่ยม	61
รูปที่ 3.5 ผลตอบสนองเชิงความถี่แถบผ่านของ FIR โดยวิธี หน้าต่างแบบสี่เหลี่ยม	63
รูปที่ 3.6 ผลตอบสนองเชิงความถี่แถบหยุดของ FIR โดยวิธี หน้าต่างแบบสี่เหลี่ยม	65
รูปที่ 3.7 ผลตอบสนองเชิงความถี่ต่ำของ FIR โดยวิธี หน้าต่างแบบแฮมมิง	66
รูปที่ 3.8 ผลตอบสนองเชิงความถี่สูงของ FIR โดยวิธี หน้าต่างแบบแฮมมิง	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.9 ผลตอบสนองเชิงความถี่แถบผ่านของ FIR โดยวิธี หน้าต่างแบบแฮมมิง	68
รูปที่ 3.10 ผลตอบสนองเชิงความถี่แถบหยุดของ FIR โดยวิธี หน้าต่างแบบ แฮมมิง	69
รูปที่ 3.11 ผลตอบสนองเชิงความถี่ต่ำของ FIR โดยวิธี หน้าต่างแบบแบล็คแมน	70
รูปที่ 3.12 ผลตอบสนองเชิงความถี่สูงของ FIR โดยวิธี หน้าต่างแบบแบล็คแมน	71
รูปที่ 3.13 ผลตอบสนองเชิงความถี่แถบผ่านของ FIR โดยวิธี หน้าต่างแบบแบล็คแมน	72
รูปที่ 3.14 ผลตอบสนองเชิงความถี่แถบหยุดของ FIR โดยวิธี หน้าต่างแบบแบล็คแมน	73
รูปที่ 3.15 แสดงโพลวัชาร์ทของ โปรแกรมการ คอนโวลูชันบน TMS320C50	75
รูปที่ 3.16 แสดงโพลวัชาร์ทของ โปรแกรมประมวลผลบน TMS320C50	85
รูปที่ 4.1 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 200 Hz	86
รูปที่ 4.2 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 600 Hz	87
รูปที่ 4.3 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 800 Hz	87
รูปที่ 4.4 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 1 KHz	88
รูปที่ 4.5 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 1.5 KHz	88
รูปที่ 4.6 สัญญาณเอาต์พุตตัวกรองต่ำผ่านเมื่ออินพุตมีความถี่ 2 KHz	89
รูปที่ 4.7 สัญญาณเอาต์พุตตัวกรองสูงผ่านเมื่ออินพุตมีความถี่ 1 KHz	89
รูปที่ 4.8 สัญญาณเอาต์พุตตัวกรองสูงผ่านเมื่ออินพุตมีความถี่ 2 KHz	90
รูปที่ 4.9 สัญญาณเอาต์พุตตัวกรองสูงผ่านเมื่ออินพุตมีความถี่ 2.5 KHz	90
รูปที่ 4.10 สัญญาณเอาต์พุตตัวกรองสูงผ่านเมื่ออินพุตมีความถี่ 3 KHz	91
รูปที่ 4.11 สัญญาณเอาต์พุตตัวกรองแถบผ่านเมื่ออินพุตมีความถี่ 1 KHz	91
รูปที่ 4.12 สัญญาณเอาต์พุตตัวกรองแถบผ่านเมื่ออินพุตมีความถี่ 2 KHz	92
รูปที่ 4.13 สัญญาณเอาต์พุตตัวกรองแถบผ่านเมื่ออินพุตมีความถี่ 3 KHz	92
รูปที่ 4.14 สัญญาณเอาต์พุตตัวกรองแถบผ่านเมื่ออินพุตมีความถี่ 3.5 KHz	93
รูปที่ 4.15 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 500 Hz	93
รูปที่ 4.16 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 1 KHz	94
รูปที่ 4.17 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 1.2 KHz	94
รูปที่ 4.18 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 2 KHz	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.19 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 3 KHz	95
รูปที่ 4.20 สัญญาณเอาต์พุตตัวกรองแถบหยุดเมื่ออินพุตมีความถี่ 3.5 KHz	96



## สารบัญตาราง

	หน้า
ตารางที่ 2.1 ผลตอบสนองต่ออิมพัลส์ของตัวกรองอุดมคติ	12
ตารางที่ 2.2 หน้าต่างแบบต่าง ๆ และค่าพารามิเตอร์ที่สำคัญ	16
ตารางที่ 2.3 รูปแบบเลข 2's complement (กรณี 8 บิต) กับค่าที่แทนได้	24
ตารางที่ 2.4 แสดงตำแหน่งและลำดับความสำคัญของอินเตอร์รัพท์ของ TMS320C50	37
ตารางที่ 3.1 การกำหนดสถานะของบิต S/C และ R/L และการทำงาน	56
ตารางที่ 3.2 ค่าสัมประสิทธิ์ของผลตอบสนองความถี่ต่ำผ่านแบบแบล็กแมนวินโดว์	77
ตารางที่ 3.3 ค่าสัมประสิทธิ์ของผลตอบสนองความถี่สูงผ่านแบบแบล็กแมนวินโดว์	79
ตารางที่ 3.4 ค่าสัมประสิทธิ์ของผลตอบสนองความถี่แถบผ่านแบบแบล็กแมนวินโดว์	81
ตารางที่ 3.5 ค่าสัมประสิทธิ์ของผลตอบสนองความถี่แถบหยุดแบบแบล็กแมนวินโดว์	83

# บทที่ 1

## บทนำ

ระบบการประมวลผลดิจิทัลในปัจจุบัน มีการนำไปใช้อย่างกว้างขวางซึ่งมีประสิทธิภาพมากในการประมวลผล ได้แก่ ระบบสื่อสารข้อมูล ระบบควบคุม โซนาร์ เรดาร์ ระบบนำร่อง ระบบประมวลผลสัญญาณภาพ โมเด็ม ระบบโทรศัพท์ เป็นต้น ซึ่งระบบประมวลผลดิจิทัลที่เป็นที่นิยมใช้กันมากคือตระกูล TMS320 ของบริษัท เท็กซัสอินSTRUMENT

ข้อดีของตัวประมวลผลสัญญาณดิจิทัล ที่เหนือกว่าการใช้วงจรรอนาล็อก ได้แก่ ความสามารถในการโปรแกรมได้ทำให้ง่ายต่อการออกแบบ การเปลี่ยนแปลงแก้ไขและทดสอบ สำหรับวงจรรอนาล็อกถ้าต้องการเปลี่ยนแปลงคุณสมบัติบางอย่าง อาจจะหมายถึงการออกแบบวงจรใหม่ ความถูกต้องแม่นยำที่คิดว่า ความถูกต้องของการประมวลผลสัญญาณดิจิทัล ขึ้นอยู่กับจำนวนบิตที่ใช้แทนสัญญาณ และพารามิเตอร์ต่างๆ ซึ่งมีความยืดหยุ่น และควบคุมง่าย

ในโครงการ “ ชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50 ” ใช้งานเกี่ยวกับการประมวลผลตัวกรอง(Filter) โดยการวิเคราะห์สัญญาณ โดยใช้โปรแกรม MATLAB ในการหาค่าพารามิเตอร์ต่างๆของตัวกรองLowpass filter,Highpass filter,Bandpass filter และ Bandstop filter แล้วนำมาเขียน โปรแกรมใช้งานจริงกับตัวบอร์ดประมวลผลสัญญาณดิจิทัล

### 1.1 วัตถุประสงค์

1. เพื่อศึกษาทฤษฎีการทำงานของตัวกรองความถี่แบบ เอฟ ไอ อาร์ โดยจำลองการทำงานด้วยโปรแกรม MATLAB
2. เพื่อนำผลที่ได้จากการจำลองการทำงานด้วยโปรแกรม MATLAB ไปสร้างเป็นชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50

### 1.2 ขอบเขตของโครงการ

สร้างชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50 ประกอบไปด้วยตัวกรอง ความถี่ต่ำผ่าน (Lowpass filter) ความถี่สูงผ่าน (Highpass filter) ความถี่แถบผ่าน (Bandpass filter) ความถี่แถบหยุด (Bandstop filter) โดยวิธีหน้าต่าง (window method) แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขั้นตอนการดำเนินโครงการ

1. ออกแบบตัวกรองความถี่แบบ เอฟ ไอ อาร์ ด้วยโปรแกรม MATLAB และหาค่าสัมประสิทธิ์ของตัวกรอง ความถี่ต่ำผ่าน (Lowpass filter) ความถี่สูงผ่าน (Highpass filter) ความถี่แถบผ่าน (Bandpass filter) ความถี่แถบหยุด (Bandstop filter) โดยวิธีหน้าต่าง (window method) แบบต่างๆ
2. นำสัมประสิทธิ์ที่ได้จากการออกแบบไปประมวลผลบนบอร์ด DSP เพื่อสร้างเป็นชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์
3. ทดสอบผลการทดลองด้วยออสซิลโลสโคป
4. สรุปผลการทดลอง



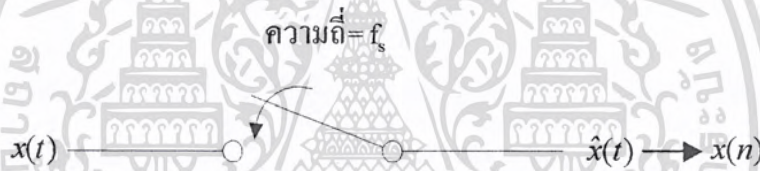
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 การแปลงสัญญาณจากอนาล็อกเป็นดิจิทัล (Analog To Digital Converter)

หลักการที่แปลงสัญญาณจากอนาล็อกเป็นดิจิทัลแบ่งได้เป็น 2 ขั้นตอน คือ

**2.1.1 การสุ่มสัญญาณ (sampling)** สัญญาณขาเข้าเป็นสัญญาณอนาล็อก  $x(t)$  ส่วนสัญญาณขาออกเป็นสัญญาณที่ไม่ต่อเนื่อง  $x(n)$  พารามิเตอร์ วงจรสุ่มสัญญาณนี้ก็คือ ค่าอัตราการสุ่ม (sampling rate) หรือ ความถี่ในการสุ่ม ใช้สัญลักษณ์คือ  $f_s$  ค่านี้จะเป็นตัวกำหนดว่า วงจรสุ่มจะสุ่มสัญญาณด้วยอัตราที่ครั้งต่อวินาที หรือ กิโลเฮิรซ์ (Hz) วงจรที่แสดงหลักการง่าย ๆ ดังรูปที่ 2.1



รูปที่ 2.1 วงจรการสุ่มด้วยสวิทช์อุดมคติ

ทำการวิเคราะห์หาสเปกตรัมของสัญญาณ  $\hat{x}(t)$  ว่ามีความสอดคล้องกับสัญญาณสเปกตรัมของ  $x(t)$  อย่างไร โดยถ้านิยามว่ามี สัญญาณอิมพัลส์ (Impulse signal) หรือเขียนแทนด้วยสัญลักษณ์ว่า  $\delta(t)$  เป็นสัญญาณที่มีค่าเท่ากับ 1 ที่เวลา  $t=0$  และเป็น 0 ที่เวลาอื่น ๆ ดังนี้

$$\delta(t) = \begin{cases} 1, & t=0 \\ 0, & t=other \end{cases} \quad (2.1)$$

เราอาจมองว่าการสุ่ม คือการนำเอาสัญญาณขาเข้า มาคูณกับสัญญาณอิมพัลส์หลาย ๆ ลูก ที่มีคาบเท่ากับ  $T$  โดยนิยามให้อิมพัลส์หลาย ๆ ลูกนี้เป็น  $s(t)$  สามารถเขียนสมการเป็นผลรวมของสัญญาณอิมพัลส์ที่ตำแหน่งเวลาต่าง ๆ  $\dots, -2T, -T, 0, T, 2T, \dots$  ดังนี้

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำ  $s(t)$  มาคูณกับ  $x(t)$  จะได้สัญญาณขาออกของตัวสุ่ม คือ

$$\hat{x}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.3)$$

โดย เราสามารถนิยามให้เป็นสัญญาณไม่ต่อเนื่อง หรือเป็น “ ลำดับของค่า ” โดยใช้ค่า  $n$  เป็นตัวชี้แทน  $t$  จะได้ ความสัมพันธ์ของ  $x(t)$  และ  $x(n)$  ดังนี้

$$x(n) = x(t)|_{t=nT} \quad (2.4)$$

ต่อไปทำการวิเคราะห์ในภาคความถี่ว่าเป็นอย่างไร สมมติว่า  $x(t)$  เมื่อใช้การแปลงฟูเรียร์ แล้วได้สเปกตรัมเป็น  $X(f)$  และสัญญาณ  $S(f)$  เป็นสเปกตรัมที่ได้จากการแปลงฟูเรียร์ของ  $s(t)$  ดังสมการ คือ

$$S(f) = f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \quad (2.5)$$

สำหรับสเปกตรัมของสัญญาณขาออก  $\hat{x}(t)$  จะได้เป็น  $\hat{X}(f)$  หาได้โดยกฎที่ว่า การคูณในเชิงเวลา เท่ากับการทำคอนโวลูชัน (convolution) ในเชิงความถี่ โดยใช้ สัญลักษณ์ \* แทนการทำคอนโวลูชัน

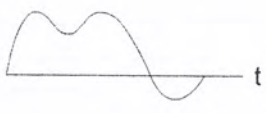

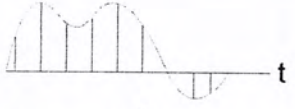
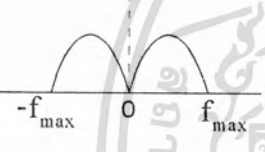
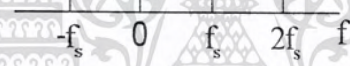
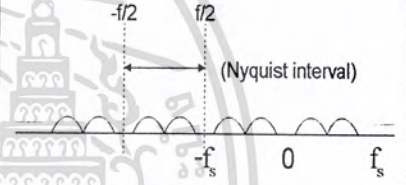
$$\begin{aligned} \text{จาก} \quad & \hat{x}(t) = x(t)s(t) \\ \text{จะได้} \quad & \hat{X}(f) = X(f) * S(f) \end{aligned}$$

เมื่อแทน  $S(f)$  ลงไป แล้วจัดให้อยู่ในรูปร่างง่ายจะได้

$$\begin{aligned} \hat{X}(f) &= X(f) * f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \\ &= f_s \sum_{n=-\infty}^{\infty} X(f) * \delta(f - nf_s) \\ \hat{X}(f) &= f_s \sum_{n=-\infty}^{\infty} X(f - nf_s) \end{aligned} \quad (2.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสมการที่ 2.6 นี้ ประกอบด้วยผลบวกของเทอม  $X(f - nf_s)$  ซึ่งเทอมนี้ ก็คือ สเปกตรัมของสัญญาณขาเข้าที่เลื่อนจุดศูนย์กลางไปอยู่ที่ตำแหน่งความถี่  $nf_s$  โดย  $n$  เป็นจำนวนเต็มตั้งแต่  $-\infty$  ถึง  $+\infty$

$x(t)$	$s(t)$	$\hat{x}(t)$ หรือ $x(n)$
	 $s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$	 $\hat{x}(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$
$ X(f) $	$S(f)$	$ \hat{X}(f) $
	 $S(f) = f_s \sum_{n=-\infty}^{\infty} \delta(f - nf_s)$	 $\hat{X}(f) = f_s \sum_{n=-\infty}^{\infty} X(f - nf_s)$

รูปที่ 2.2 สัญญาณต่างๆ ในการสุ่มทั้งภาคเวลาและความถี่

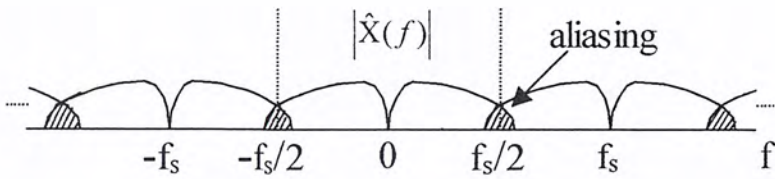
### 2.1.2 ทฤษฎีการสุ่มสัญญาณ (Sampling Theorem)

ทฤษฎีการสุ่มสัญญาณระบุว่า ถ้าสัญญาณที่ต้องการสุ่มมีความถี่สูงสุด  $f_{\max}$  เพื่อให้ได้สัญญาณที่สุ่มแล้วเป็นตัวแทนที่ถูกต้องของสัญญาณอินพุท ความถี่ที่ใช้ในการสุ่มจะต้องมีค่ามากกว่าสองเท่าของความถี่สูงสุดในสัญญาณ นั่นคือ

$$f_s > 2f_{\max} \quad (2.7)$$

ซึ่งเรียก  $2f_{\max}$  ว่า “ความถี่ไนควิสต์” (Nyquist frequency) และถ้าเราใช้  $f_s$  ต่ำกว่าค่าความถี่ไนควิสต์ จะทำให้เกิดความผิดพลาดในการสุ่มที่เรียกว่า aliasing ซึ่งเป็นองค์ประกอบความถี่ที่ซ้อนทับกันดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

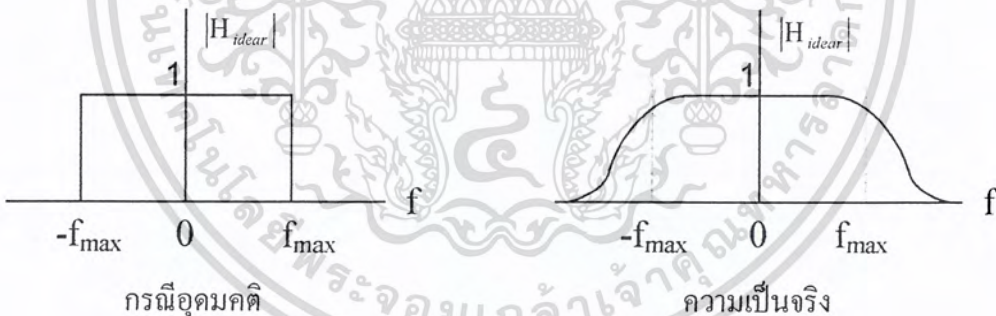


รูปที่ 2.3 แสดงสัญญาณในเชิงความถี่ที่เกิดขึ้น เมื่อใช้  $f_s$  ต่ำกว่า  $2f_{\max}$

เราต้องพยายาม ไม่ให้เกิด aliasing ขณะทำการประมวลผลสัญญาณ โดยแก้ไขได้สองวิธี คือ

### 2.1.2.1 การใช้ตัวกรองเพื่อป้องกัน aliasing (Anti-Aliasing Filter)

ในกรณีที่เราไม่มั่นใจว่าสัญญาณที่จะทำการสุ่ม ไม่มีความถี่จำกัดอยู่ที่  $f_{\max}$  เช่น อาจมีสัญญาณรบกวนความถี่สูง หรือสัญญาณอื่นๆ ปนอยู่ด้วย วิธีแก้ทำได้โดยการใช้ตัวกรองอนาล็อกแบบผ่านความถี่ต่ำ เพื่อจำกัดความถี่ของสัญญาณให้อยู่ในช่วงความถี่ที่สนใจเท่านั้น (ต่ำกว่า  $f_{\max}$ ) นั่นคือ เราต้องการตัวกรองผ่านความถี่ต่ำที่มีความถี่ตัดที่  $f_{\max}$  ดังแสดงผลตอบสนองเชิงความถี่แบบอุดมคติของตัวกรองนี้ ในรูปซ้ายมือของรูปที่ 2.4



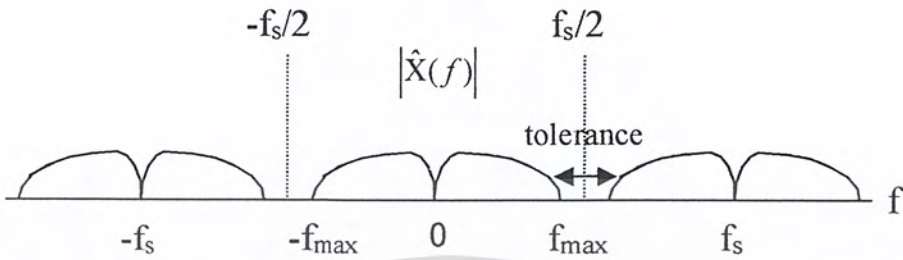
รูปที่ 2.4 ผลตอบสนองเชิงความถี่ของตัวกรองป้องกัน aliasing

### 2.1.2.2 การสุ่มโดยใช้ความถี่เกินกว่า $2f_{\max}$ มาก ๆ

หรือเรียกว่าการทำ Oversampling การสุ่มด้วยความถี่สูงขึ้นเป็นวิธีที่ใช้ป้องกัน aliasing เสริมจากวิธีแรก เช่นเดียวกัน ถ้าพิจารณาในเชิงความถี่จะเห็นว่า เมื่อ มีค่าสูงขึ้นจะมีช่วงความถี่ที่เพื่อให้เกิด aliasing ได้กว้างขึ้น (ช่วงเฟื่อนี้คือ ย่าน tolerance ซึ่งเป็นย่านที่ไม่มีความถี่ที่เราสนใจอยู่) ดังนั้นการสุ่มด้วยความถี่ที่สูงกว่าอัตราในควิสท์ มาก ๆ จึงทำให้เราสามารถในตัวป้องกัน aliasing ที่ไม่ต้องมีคุณสมบัติดีมากนักในทางปฏิบัติจึงมักใช้  $f_s \geq 2.5f_{\max}$  เพื่อชดเชยผลของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ตัวกรอง Anti-Aliasing Filter ไม่เป็นอุดมคติ สำหรับ  $f_s$  สูงสุดที่ใช้ก็ขึ้นอยู่กับขีดจำกัดด้านความเร็วของตัวแปลงอนาล็อกเป็นดิจิทัล และตัวประมวลผลที่เลือกใช้



รูปที่ 2.5 สเปกตรัม ของสัญญาณที่เกิดขึ้นเมื่อสุ่มด้วยควมถี่สูงกว่า  $2f_{max}$  มาก ๆ

## 2.2 ผลการตอบสนองต่ออิมพัลส์ของระบบ (Impulse Response)

สมการของผลตอบสนองอิมพัลส์ (Impulse Response) นั้นจะได้จากสมการ ทัวไปของผลตอบสนองในหน่วยของสัญญาณ (unit sample response) ซึ่งมาจากที่ให้อินพุตเป็น  $x(n) = \delta(n)$  แก่ระบบ และจะได้เอาท์พุทออกมา ซึ่งหมายถึงทรานส์เฟอร์ฟังก์ชัน (Transfer function) ของระบบ ในการพิจารณาระบบๆหนึ่งซึ่งมีเอาท์พุทเป็น

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

เนื่องจากเราต้องการหาผลตอบสนองในหน่วยของสัญญาณเราจึงให้ยูนิคแซมเปิลเป็น

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

ดังนั้นจะได้เอาท์พุทซึ่งเรียกว่า ผลตอบสนองในหนึ่งหน่วยของสัญญาณซึ่งกำหนดให้

$$y(n) = h(n) \text{ เมื่อ } x(n) = \delta(n) \quad (2.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากสมการ (2.9) เมื่อเราแทน  $y(n) = h(n)$  และ  $x(n) = \delta(n)$  จะได้

$$\begin{aligned} y(1) &= h(1) = b(1) \\ y(0) &= h(0) = b(0) \\ &\cdot \quad \cdot \quad \cdot \\ &\cdot \quad \cdot \quad \cdot \\ &\cdot \quad \cdot \quad \cdot \\ y(L) &= h(L) = b(L) \end{aligned}$$

จะเห็นได้ว่า

$$y(n) = h(n) = \begin{cases} b_n, n=0,1,2,3,\dots,L \\ 0, n<0, n>L \end{cases}$$

รูปแบบทั่วไปของผลตอบสนองในหน่วยของสัญญาณคือ

$$\begin{aligned} h(n) &= \sum_{k=0}^L b_k \delta(n-k) \\ &= \sum_{k=0}^L h(k) \delta(n-k) \end{aligned} \quad (2.10)$$

### 2.3 ผลการประสาน (Convolution)

จากข้างต้น ได้อธิบายเกี่ยวกับสัญญาณลักษณะต่างๆ ในทางดิจิทัลแล้ว ซึ่งในหัวข้อจะพิจารณา การนำสัญญาณดิจิทัลผ่านเข้าไปในระบบต่างๆ กำหนดให้  $x(n)$  เป็นอินพุท ที่ผ่านระบบที่ต้องการ  $y(n)$  คือผลที่ได้จากการนำอินพุทผ่านระบบ  $h(n)$  เนื่องจาก  $h(n)$  สามารถอธิบายได้ด้วยผลตอบสนองอิมพัลส์

ซึ่งหลักการของผลตอบสนองของยูนิตอิมพัลส์มีสมการดังนี้

$$h(n) = R\delta(n) \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเมื่อได้ผลตอบสนองของยูนิตอิมพัลส์ที่ต้องการแล้วจะได้เอาท์พุทคือ

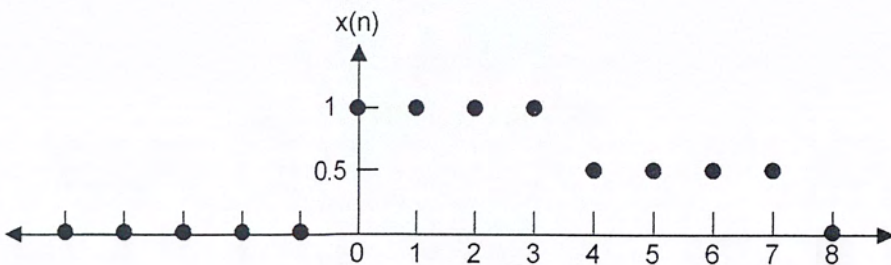
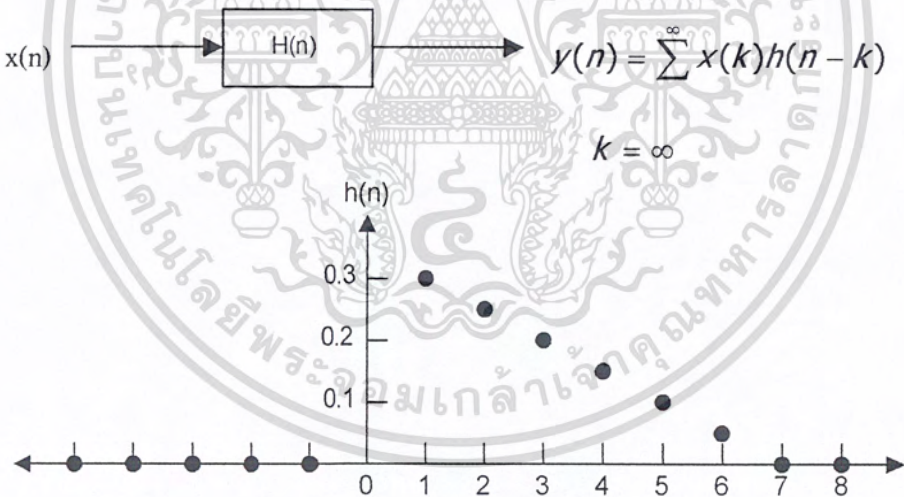
$$y(n) = R[x(n)] = \sum_{-\infty}^{\infty} x(k)h(n-k) \quad (2.12)$$

หรือ

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

สมการที่ได้นี้จะเป็นสมการของคอนโวลูชัน ระหว่างสัญญาณอินพุท และผลตอบสนองอิมพัลส์ของระบบ

ตัวอย่างการคอนโวลูชัน



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5
-----	-----	-----	-----	-----	-----	-----	-----

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(0) = 1(0.3) = 0.3$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(1) = 1(0.25)+1(0.3) = 0.55$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(2) = 1(0.2)+1(0.25)+1(0.3) = 0.75$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(3) = 1(0.15)+1(0.2)+1(0.25)+1(0.3) = 0.9$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(4) = 1(0.1)+1(0.15)+1(0.2)+1(0.25)+0.5(0.3) = 0.85$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

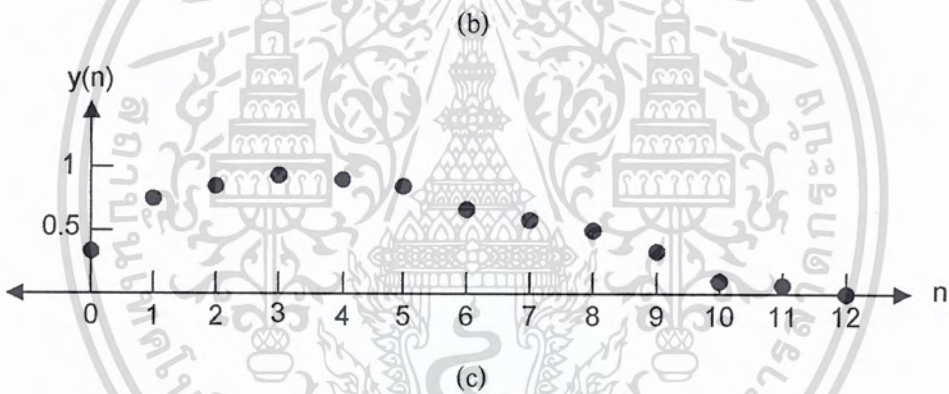
 $y(5) = 1(0.05)+1(0.1)+1(0.15)+1(0.2)+0.5(0.25)+0.5(0.3) = 0.775$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(6) = 1(0.05)+1(0.1)+1(0.15)+0.5(0.2)+0.5(0.25)+0.5(0.3) = 0.675$ 
  
 $y(7) = 0.61y(8) = 0.4y(9) = 0.25y(10) = 0.15y(11) = 0.075$ 
  

0.05	0.1	0.15	0.2	0.25	0.3
------	-----	------	-----	------	-----

 $y(12) = 0.05(0.5)0.025$



รูปที่ 2.6 แสดงการคอนโวลูชันของสัญญาณ

จากรูปที่ 2.6 จะเห็นว่า  $y(n]$  ได้มาจากการเลื่อนตำแหน่งของ  $h(n]$  ในการเลื่อนตำแหน่งจะเลื่อนไปทางขวาของรูปภาพ และจะได้ output ตามตำแหน่งของสัญญาณจะมีค่าเท่ากับ จำนวนลำดับของ input บวกด้วยจำนวนลำดับของ  $h(n]$  ลบด้วย 1 ดังนั้นจากรูป 2.6 จะได้จำนวนเอทพุทคือ 13 ลำดับ

## 2.4 การออกแบบโดยวิธีหน้าต่าง (Window Method)

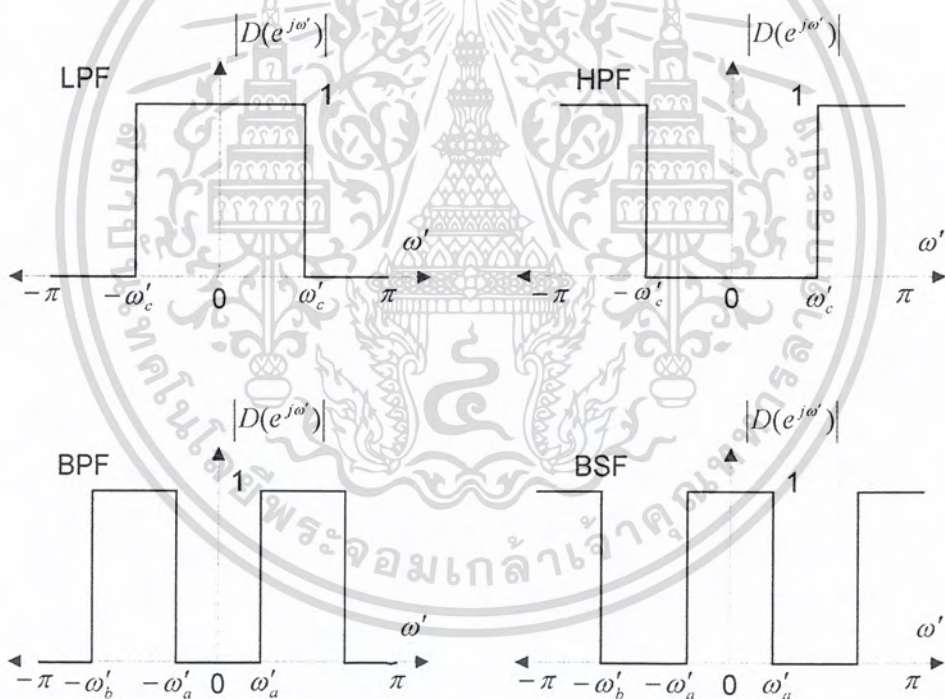
ในส่วนนี้จะได้อธิบายถึง การหาค่าสัมประสิทธิ์ของตัวกรอง FIR เมื่อกำหนดคุณลักษณะเฉพาะของตัวกรอง ซึ่งคุณลักษณะเฉพาะนี้ ส่วนใหญ่จะเป็นการกำหนดลักษณะของผลตอบสนอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชิงความถี่ที่ต้องการ ได้แก่ ความถี่ตัด, ความคมของตัวกรอง, การลดทอนในแถบหยุด และอื่นๆ เราจะทำการออกแบบโดยคำนึงถึงความถี่ดิจิทัล  $\omega'$  ที่มีย่านความถี่ที่สนใจในช่วง  $-\pi$  ถึง  $\pi$  หรือ  $f'$  ในช่วง  $-1$  ถึง  $1$

วิธีหน้าต่างเป็นวิธีพื้นฐานที่สุดวิธีหนึ่งที่จะใช้หาสัมประสิทธิ์ของตัวกรอง เป็นวิธีที่ง่ายต่อการออกแบบ และสามารถใช้ออกแบบตัวกรองแบบต่างๆ ได้ ไม่ว่าจะเป็นแบบความถี่ต่ำผ่าน (LPF), ความถี่สูงผ่าน (HPF), ความถี่แถบผ่าน (BPF), หรือความถี่แถบหยุด (BSF)

เราจะเริ่มออกแบบโดยใช้ต้นแบบจากตัวกรองอุดมคติ ผลตอบสนองเชิงความถี่ของตัวกรองอุดมคติทั้งสี่แบบแสดงอยู่ในรูปที่ 2.7 โดยมีความถี่ตัด (cutoff frequency) เท่ากับ  $\omega'_c$  สำหรับแบบผ่านต่ำ และผ่านสูง ส่วนแบบผ่านแถบความถี่ และตัดแถบความถี่มีความถี่ตัดของแถบความถี่ที่  $\omega'_a$  และ  $\omega'_b$  ดังแสดงในรูป



รูปที่ 2.7 ผลตอบสนองเชิงความถี่ของตัวกรองอุดมคติ

สมมติให้  $d(n)$  แทนผลตอบสนองต่ออิมพัลส์ และ  $D(e^{j\omega'})$  แทนผลตอบสนองเชิงความถี่ สำหรับตัวกรองแบบผ่านต่ำอุดมคติ ดังในรูป 2.7 เราจะหาผลตอบสนองต่ออิมพัลส์ของตัวกรองอุดมคติได้โดยการแปลง IDTFT ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 d(n) &= \text{IDTFT}\{D\} \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} D(e^{j\omega'}) e^{j\omega' n} d\omega' \\
 &= \frac{1}{2\pi} \int_{-\omega'_c}^{\omega'_c} (1) e^{j\omega' n} d\omega' \\
 &= \left[ \frac{e^{j\omega' n}}{2\pi j n} \right]_{-\omega'_c}^{\omega'_c} \\
 &= \frac{e^{j\omega'_c n} - e^{-j\omega'_c n}}{2\pi j n} \\
 d(n) &= \frac{\sin(\omega'_c n)}{\pi n}, -\infty < n < \infty
 \end{aligned} \tag{2.13}$$

สมการนี้มีปัญหาที่  $n=0$  เพราะจะได้ว่า  $d(n)$  มีค่าเป็นเศษศูนย์ส่วนศูนย์ ซึ่งเราสามารถหาค่า  $d(0)$  ได้โดยใช้ทฤษฎีบทของโลปีตัล จะได้ว่า

$$d(0) = \frac{\lim_{n \rightarrow 0} \frac{d(\sin(\omega'_c n))}{dn}}{\lim_{n \rightarrow 0} \frac{d(\pi n)}{dn}} = \frac{\omega'_c}{\pi} \tag{2.14}$$

ในการทำงานเดียวกันเราสามารถหา  $d(n)$  สำหรับตัวกรองแบบอื่นๆ ได้โดยใช้ IDTFT กระทำกับผลตอบสนองเชิงความถี่ของตัวกรองนั้นๆ ตารางที่ 2.1 ได้สรุปค่าของ  $d(n)$  สำหรับตัวกรองแบบต่างๆ ไว้ เพื่อใช้ในการออกแบบต่อไป

ชนิดของตัวกรอง	$d(n), -\infty < n < \infty$	$d(n)$
ผ่านต่ำ (LPF)	$\frac{\sin(\omega'_c n)}{\pi n}$	$\frac{\omega'_c}{\pi}$
ผ่านสูง (HPF)	$\delta(n) - \frac{\sin(\omega'_c n)}{\pi n}$	$1 - \frac{\omega'_c}{\pi}$
ผ่านแถบความถี่ (BPF)	$\frac{\sin(\omega'_b n) - \sin(\omega'_a n)}{\pi n}$	$\frac{\omega'_b}{\pi} - \frac{\omega'_a}{\pi}$
ตัดแถบความถี่ (BSF)	$\delta(n) - \frac{\sin(\omega'_b n) - \sin(\omega'_a n)}{\pi n}$	$1 - \frac{\omega'_b}{\pi} + \frac{\omega'_a}{\pi}$

ตารางที่ 2.1 ผลตอบสนองต่ออิมพัลส์ของตัวกรองอุดมคติต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกขาดให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราอาจหา  $d(n)$  ของตัวกรองแบบอื่นๆได้จากตัวกรอง LPF และตัวกรองแบบผ่านทุกความถี่ ตัวกรองแบบผ่านทุกความถี่ในที่นี้ คือ  $D(e^{j\omega'}) = 1$  หรือ  $d(n) = \delta(n)$  ตัวอย่างเช่น ถ้าต้องการหา  $d(n)$  ของ HPF จาก  $d(n)$  ของ LPF เราสามารถใช้ผลตอบสนองเชิงความถี่ของ LPF บวกกับผลตอบสนองเชิงความถี่ของ HPF ที่มีความถี่ตัดตรงกัน จะได้เป็นตัวกรองผ่านทุกความถี่ ดังนี้

$$D_{LP}(e^{j\omega'}) + D_{HP}(e^{j\omega'}) = 1 \quad (2.15)$$

เมื่อแปลง  $z$  ทั้งสมการจะได้ความสัมพันธ์ของ  $d(n)$  เป็น

$$d_{LP}(n) + d_{HP}(n) = \delta(n) \quad (2.16)$$

ยังมีความสัมพันธ์อื่นในลักษณะเดียวกันนี้ก็คือ ผลตอบสนองเชิงความถี่ของ HPF บวกกับผลตอบสนองเชิงความถี่ของ BSF ที่มีความถี่ตัดตรงกัน จะได้เป็นตัวกรองผ่านทุกความถี่ ดังนี้

$$D_{BP}(e^{j\omega'}) + D_{BS}(e^{j\omega'}) = 1 \quad (2.17)$$

แปลง  $z$  ผลผันได้

$$d_{BP}(n) + d_{BS}(n) = \delta(n) \quad (2.18)$$

และผลตอบสนองเชิงความถี่ของ BPF ก็สามารถหาได้จากการนำเอาผลตอบสนองเชิงความถี่ของ LPF ที่มีความถี่ตัดเท่ากับความถี่ตัดด้านต่ำของ BPF ลบออกจากผลตอบสนองเชิงความถี่ของ LPF ที่มีความถี่ตัดเท่ากับความถี่ตัดด้านสูงของ BPF ดังนี้

$$D_{BP}(e^{j\omega'}) = D_{LP}(e^{j\omega'}) \Big|_{\omega'_c = \omega'_b} - D_{LP}(e^{j\omega'}) \Big|_{\omega'_c = \omega'_a} \quad (2.19)$$

แปลง  $z$  ผลผันได้

$$d_{BP}(n) = d_{LP}(n) \Big|_{\omega'_c = \omega'_b} - d_{LP}(n) \Big|_{\omega'_c = \omega'_a} \quad (2.20)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะใช้ค่า  $d(n)$  ในตารางที่ 2.1 เพื่อเป็นผลตอบสนองต่ออิมพัลส์ของตัวกรองที่ต้องการ แต่อย่างไรก็ตาม  $d(n)$  ที่ได้นี้ไม่สามารถสร้างได้จริงในทางปฏิบัติ เนื่องจากมีความยาวไม่จำกัด และเป็นแบบสองด้าน กล่าวคือ  $d(n)$  มีค่าตั้งแต่  $n$  เป็น  $-\infty$  จนถึง  $\infty$  เราจะต้องใช้เทคนิคบางอย่างเพื่อให้  $d(n)$  เหล่านี้มาใช้ได้ ซึ่งเทคนิคนี้ก็คือ วิธีหน้าต่างนั่นเอง

ก่อนอื่นมาทำความรู้จักกับผลตอบสนองเชิงความถี่ก่อน ซึ่งมีรูปร่างดังรูปที่ 2.7 ค่าที่จะใช้กำหนดเป็นคุณลักษณะเฉพาะของวิธีหน้าต่าง ได้แก่

ความพลิวของแถบผ่าน (pass-band ripple,  $\delta_{pass}$ ) คือ ค่าสูงสุดที่ขนาดของแถบผ่านแกว่งออกห่างจากค่าหนึ่ง บางครั้งวัดเป็น dB โดยใช้

$$A_{pass} = 20 \log \frac{1 + \delta_{pass}}{1 - \delta_{pass}} \text{ (dB)} \quad (2.21)$$

การลดทอนของแถบหยุด (stop band attenuation,  $A_{stop}$ ) คือ จำนวนเท่าที่แถบหยุดลดทอนลงจากค่าหนึ่ง วัดเป็น dB โดยมีความสัมพันธ์กับความพลิวของแถบหยุดคือ

$$A_{stop} = -20 \log \delta_{stop} \text{ (dB)} \quad (2.22)$$

ความกว้างของแถบเปลี่ยน (transition band width,  $\Delta f$ )

ความถี่ตัด (cutoff frequency,  $f_c$ ) คือ ค่าความถี่ที่ขนาดลดลงประมาณ 0.5 หรืออยู่ที่ประมาณครึ่งหนึ่งของแถบเปลี่ยน

การออกแบบโดยวิธีหน้าต่าง สามารถแบ่งเป็นขั้นตอนย่อยได้ดังนี้

1 ใช้ค่าความพลิวของแถบผ่าน หรือการลดทอนของแถบหยุดอย่างใดอย่างหนึ่งเพื่อเลือกชนิดของหน้าต่างที่สามารถใช้ได้จากรายการที่ 2.2 ถ้ามีข้อกำหนดทั้งสองอย่าง ให้เปลี่ยน  $A_{stop}$  เป็น  $\delta_{stop}$  ก่อน แล้วเปรียบเทียบ  $\delta_{pass}$  กับ  $\delta_{stop}$  ว่าค่าใดน้อยกว่ากัน ถ้า  $\delta_{pass}$  น้อยกว่าให้ใช้  $\delta_{pass}$  เป็นตัวเลือกหน้าต่าง แต่ถ้า  $\delta_{stop}$  น้อยกว่าให้ใช้  $A_{stop}$  เป็นตัวเลือกหน้าต่าง โดยมีหลักการว่า  $\delta_{pass}$  ในตารางต้องน้อยกว่าที่ต้องการ และ  $A_{stop}$  ในตารางต้องมากกว่าที่ต้องการ

2 ใช้ค่าความกว้างของแถบเปลี่ยนหาค่าอันดับของตัวกรองที่ต้องการใช้ โดยใช้ความสัมพันธ์ระหว่าง  $\Delta f$  กับ  $N$  ที่แสดงไว้ในตารางที่ 2.2 จากนั้น คำนวณฟังก์ชันหน้าต่างที่ต้องการใช้ จะได้

$$w(n), \text{ ที่ } n = 0, 1, \dots, N-1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

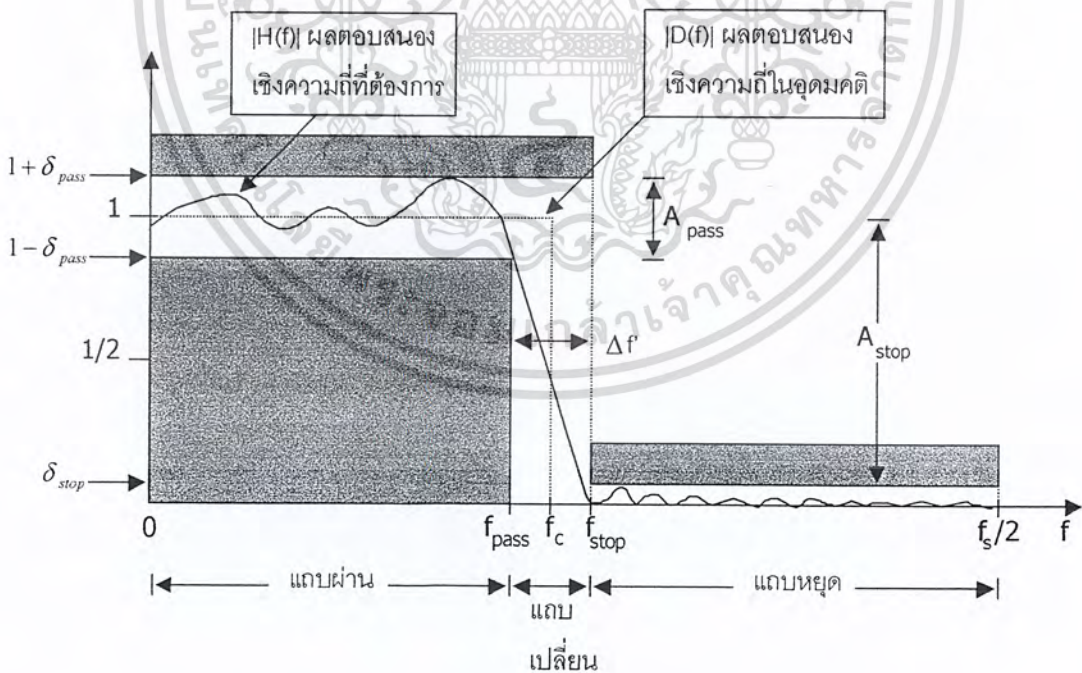
จากตารางสังเกตได้ว่าอันดับของตัวกรองจะเป็นสัดส่วนผกผันกับ  $\Delta f$  นั่นคือ เราสามารถปรับปรุง  $\Delta f$  ให้แคบลงได้โดยการเพิ่มค่า  $N$  แต่เราไม่สามารถปรับปรุงค่า  $\delta_{stop}$  และ  $A_{stop}$  ให้ดีขึ้นได้ เพราะค่า  $\delta_{stop}$  และ  $A_{stop}$  จะมีขนาดคงที่ สำหรับหน้าต่างแบบต่างๆ ยกเว้นหน้าต่างแบบ Kaiser

3 ใช้ชนิดของตัวกรอง (LPF, HPF, ...) และความถี่ตัดที่ต้องการ เลือกผลตอบสนองต่ออิมพัลส์  $d(n)$  ที่ถูกต้องจากตารางที่ 2.1

4 เลื่อน  $d(n)$  ให้ล้าหลังลง  $M$  ตำแหน่ง โดย  $M=(N-1)/2$  จะได้สัญญาณเป็น  $d(n-M)$  จากนั้นคูณเข้ากับฟังก์ชันหน้าต่าง  $w(n)$  ที่ได้จากข้อ 2 ซึ่งจะได้เป็นผลตอบสนองต่ออิมพัลส์ที่มีความยาว  $N$  จุด และเป็นแบบคอซัล ดังนี้

$$h(n) = d(n-M)w(n), n = 0, 1, \dots, N-1 \quad (2.23)$$

การออกแบบนี้จะต้องใช้  $N$  เป็นจำนวนคี่ ซึ่งจะได้เป็นตัวกรอง FIR ที่มีเฟสเชิงเส้น และมีสมมาตรชนิดที่ 1 เพราะทั้ง  $d(n)$  และ  $w(n)$  มีสมมาตรรอบจุดกึ่งกลางทั้งคู่



รูปที่ 2.8 ค่าต่างๆในการระบุคุณลักษณะเฉพาะของผลตอบสนองเชิงความถี่ของตัวกรอง FIR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง	$\delta_{pass}$	$A_{stop}$ $-20\log\delta_{stop}$ (dB)	$\Delta f$ (normalized)	$w(n), n=0,1,\dots,N-1$ $(M = \frac{N-1}{2})$
สี่เหลี่ยม (rectangular)	8.9%	21	$2/N$	1
ฮานนิง (Hanning)	0.63%	44	$4/N$	$0.5 - 0.5\cos(\frac{2\pi n}{N-1})$
แฮมมิง (Hamming)	0.22%	53	$4/N$	$0.54 - 0.64\cos(\frac{2\pi n}{N-1})$
แบล็กแมน (Blackman)	0.02%	74	$6/N$	$0.42 - 0.5\cos(\frac{2\pi n}{N-1}) + 0.08\cos(\frac{4\pi n}{N-1})$
ไคเซอร์ (Kaiser)	ปรับได้	ปรับได้	$\frac{A-7.95}{14.36(N-1)}$	$\frac{I_0(\alpha\sqrt{1-(n-M)^2/M^2})}{I_0(\alpha)}$

ตารางที่ 2.2 หน้าต่างแบบต่างๆ และค่าพารามิเตอร์ที่สำคัญ

#### 2.4.1 หน้าต่างแบบสี่เหลี่ยม (Rectangular Window)

จากตารางที่ 2.1 ใช้  $d(n)$  สำหรับ LPF คือ  $d(n) = \frac{\sin(\omega'_c n)}{\pi n}$  เลื่อน  $d(n)$  ให้ล้าหลังไป  $M$  ตำแหน่ง จะได้  $d(n-M)$  คือ

$$d(n-M) = \frac{\sin(\omega'_c(n-M))}{\pi(n-M)}$$

สำหรับหน้าต่างแบบสี่เหลี่ยม เราจะได้  $w(n) = 1, n=0, 1, \dots, N-1$  ดังนั้น หา  $h(n)$  ได้โดยใช้สมการที่ 2.23 คือ

$$\begin{aligned} h(n) &= d(n-M)w(n) \\ &= \frac{\sin(\omega'_c(n-M))}{\pi(n-M)}, n=0, 1, \dots, N-1 \\ &= \frac{\sin(\pi/2(n-22))}{\pi(n-22)}, n=0, 1, \dots, 44 \end{aligned}$$

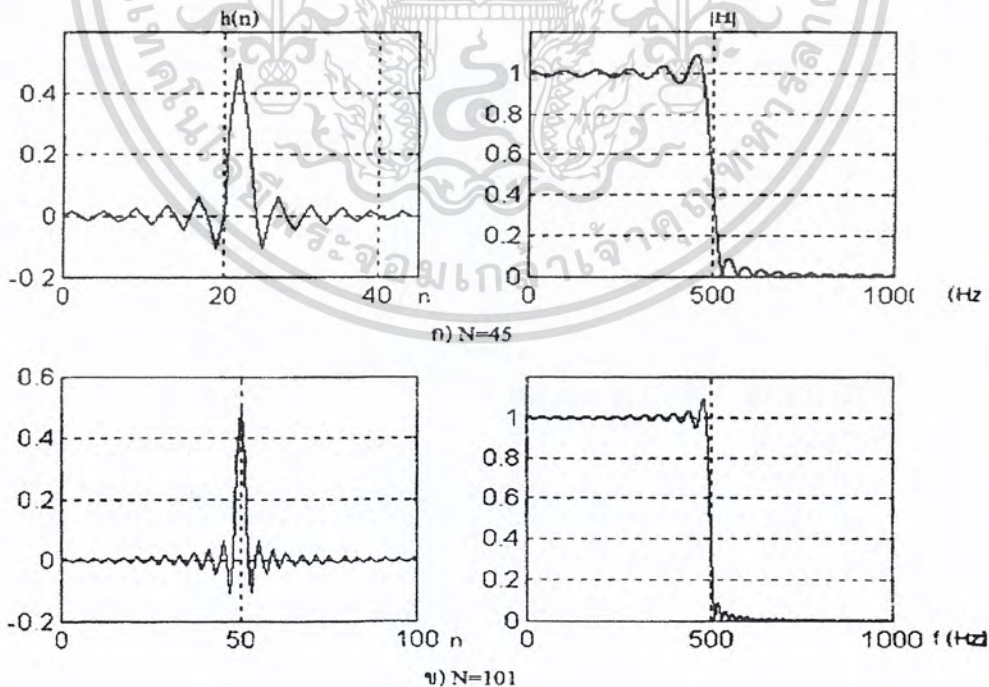
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับผลตอบสนองเชิงความถี่ของระบบ หาได้โดยหาสเปกตรัมของสัญญาณ  $h(n)$  ซึ่งก็คือ  $H(e^{j\omega'}) = DTFT(h(n))$  ซึ่งสามารถประมาณได้โดยการใช้ FFT ซึ่งจะได้  $H(k) = FFT(h(n))$  ก็ได้ในรูปที่ 2.9 เราใช้โปรแกรม freques และวาดออกมาในช่วงความถี่ที่สนใจ คือ  $f = (0, f_s/2)$  ได้รูปที่ 2.9

จะเห็นได้ว่าผลตอบสนองเชิงความถี่มีลักษณะเปลี่ยนไปจากอุดมคติ โดยมีย่านแถบเปลี่ยนเกิดขึ้น และมีการแกว่งขึ้นลง หรือ ความพลัวในช่วงแถบผ่าน และแถบหยุด ปรัชการณณ์ที่เกิดความพลัวขึ้นนี้ เรียกว่า ปรัชการณณ์ของ Gibbs

เมื่อเราเพิ่ม  $N$  มากขึ้นๆ ก็ได้ผลตอบสนองเชิงความถี่ที่เข้าใกล้อุดมคติมากขึ้นๆ เช่นกัน ลองใช้ค่า  $N$  เป็น 101 และวาด  $h(n)$  และ  $|H|$  ที่ได้ในรูปที่ 2.9 เมื่อเปรียบเทียบกับผลที่ได้ของกรณี  $N = 45$  จะเห็นได้ว่า แถบเปลี่ยนมีการบีบตัวแคบลง และความพลัวของแถบผ่านและแถบหยุดก็มีการบีบตัวแคบลง แต่ความสูงของความพลัวมีขนาดค่อนข้างคงที่ ซึ่งจากตารางที่ 2.2 เราจะได้ว่า  $\delta_{pass} \approx \delta_{stop} \approx 0.089$  สำหรับหน้าต่างแฮมมิงซึ่งพบว่า หน้าต่างทุกชนิดจะให้  $\delta$  มีค่าค่อนข้างคงที่โดยไม่ขึ้นกับ  $N$  นี้ ถ้าเราต้องการ  $\delta$  ที่เล็กลง จำเป็นต้องเปลี่ยนชนิดของหน้าต่างที่ใช้

หลังจากเพิ่มค่า  $N$  จะเห็นได้ว่า  $\Delta f$  มีขนาดแคบลง ซึ่งเมื่อคำนวณหาค่า  $\Delta f$  จากสูตรในตารางจะได้  $\Delta f = \Delta f_s \quad f_s = 2f_s/N = 2(2000)/101 = 40 \text{ Hz}$  ซึ่งใกล้เคียงกับค่าที่ได้ในรูป



รูปที่ 2.9 ผลลัพธ์ที่ได้จากตัวอย่างที่ 2.2 (ใช้หน้าต่างแบบสี่เหลี่ยม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.2 หน้าต่างฮานนิงและแฮมมิง (Hanning and Hamming Window)

หน้าต่างแฮมมิงนี้จะเป็นการปรับปรุงลักษณะของ Window เพื่อพยายามให้ได้ผลที่ดีทั้งส่วนของ Transition region และ Side-lobe โดยเพิ่มส่วนที่เป็น DC และลักษณะของความลาดชันของ Window จะได้เป็นสูตร Hanning window คือ

$$w(n) = 0.5 - 0.5 \cos\left(\frac{n\pi}{M+1}\right), \dots - M \leq n \leq M \quad (2.24)$$

$$w(n) = 0, \dots \text{other}$$

จะเห็นได้ว่าจะเป็นความลาดชันที่มีลักษณะ cosine ซึ่งมีความเป็น smooth มากกว่าแบบ Triangular และ เพิ่มค่าของ DC เข้า ต่อมาได้มีผู้นำมาพัฒนาโดยการปรับอัตราส่วนของ DC และขนาดของ M และทำให้ได้รูปแบบของ Freq-response ที่ดีขึ้น เรียกว่า Hamming window จากสูตร

$$w(n) = 0.54 - 0.46 \cos\left(\frac{n\pi}{M}\right), \dots - M \leq n \leq M \quad (2.25)$$

$$w(n) = 0, \dots \text{other}$$

ด้วยทั้งสองแบบ ไม่ค่อยแตกต่างกันมากนัก จึงสามารถนำมาเขียนเป็นสูตรรวมได้เป็น

$$w(n) = A + B \cos\left(\frac{n\pi}{C}\right), \dots - M \leq n \leq M \quad (2.26)$$

$$w(n) = 0, \dots \text{other}$$

### 2.4.3 หน้าต่างไคเซอร์ (Kaiser Window)

หน้าต่างไคเซอร์มีลักษณะพิเศษกว่าหน้าต่างแบบอื่นๆ ตรงที่สามารถปรับค่าความพลีวของแถบผ่าน และการลดทอนของแถบหยุดได้ ซึ่งทำให้มีความยืดหยุ่นในการออกแบบมากกว่าแบบอื่นๆ ทำให้มักจะได้ค่า N ต่ำกว่าหน้าต่างแบบอื่นที่มีลักษณะเฉพาะเดียวกันมีสูตรดังนี้

$$w(n) = \frac{I_0\left(\alpha \sqrt{1 - \left(\frac{n}{M}\right)^2}\right)}{I_0(\alpha)}, \dots - M \leq n \leq M \quad (2.27)$$

$$w(n) = 0, \dots \text{other}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ  $I_0$  นั้นเป็น Modified Bessel function ที่ Zero order จากสมการเมื่อเราให้ค่าของ  $n=0$  จะให้ผลที่สุดเป็นหนึ่ง นั่นคือที่จุด Center จะมีค่าเป็น unity เสมอ ค่าของ  $\alpha$  จะเป็นค่าที่ใช้ควบคุมความลาดชันของตัว window จากจุด Center สู่ edge แต่ละข้าง หากค่าของ  $\alpha$  มีค่าเป็นศูนย์ ก็จะมีลักษณะเป็น Rectangular window และหากมีค่าเป็น 5.44 ก็จะมีลักษณะคล้ายกับ Hamming filter นั่นเอง เราสามารถมองเห็นข้อกำหนดของการออกแบบแบบนี้ได้ โดยสังเกตรูปที่ 2.10



รูปที่ 2.10 ลักษณะพิเศษในการออกแบบ แบบหน้าต่างโคเชอร์

จากรูปลักษณะของ Ideal filter  $H_D(\omega_{k-1})$  ที่มีส่วนของขนาด Ripple ที่สามารถยอมรับได้ ( $\pm\delta$ ) ที่ส่วนของ Pass-band และ Stop-band และความกว้างที่ยอมรับได้ของช่วง Transition region ในการออกแบบจะพยายามไม่ไห้ส่วนใดๆของ filter ทั้ส่วนที่แรงแง ไว้โดยใช้ค่าจำนวนของ  $M$  ให้น้อยที่สุด ค่าพารามิเตอร์ของ  $\alpha$  สามารถสรุปย่อสำหรับค่า -dB ที่ต้องการดังต่อไปนี้

$$\begin{aligned} \alpha &= 0.1101(A - 8.7) && \text{เมื่อ } A \geq 50 \\ \alpha &= 0.584(A - 21)^{0.4} + 0.07886(A - 21) && \text{เมื่อ } A \geq 50 \\ \alpha &= 0.0 && \text{เมื่อ } A \geq 21 \end{aligned}$$

ในส่วน of ค่า  $A$  และ Transition region ( $\Delta$ ) ที่มีความสัมพันธ์กับค่าของ  $M$  ดังนี้

$$M \geq \frac{A - 7.95}{28.72\Delta} \quad (2.28)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบหาค่าของ  $M$  มีค่ามากขึ้นไป เราสามารถปรับขนาดของ Ripple ให้มากขึ้นเพื่อให้ค่าของ  $M$  อยู่ในค่าที่ยอมรับได้ ส่วนค่าของ  $I_o$  สามารถเขียนเป็นค่า Power series ได้เป็น

$$I_o(x) = 1 + \sum_{n=1}^{\infty} \left[ \left( \frac{x}{2} \right)^2 \frac{1}{n!} \right]^2 \quad (2.29)$$

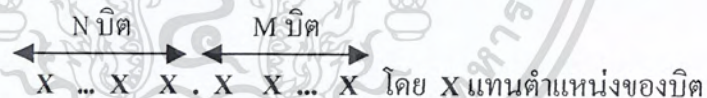
$$I_o(x) = 1 + \frac{x^2}{4} + \frac{x^2}{4} \cdot \frac{x^2}{4} \cdot \frac{1}{2^2} + \frac{x^4}{16} \cdot \frac{x^2}{4} \cdot \frac{1}{3^3} + \dots$$

## 2.5 ระบบตัวเลขในการประมวลผล

### 2.5.1 ระบบเลขจำนวนเต็ม (Fixed-Point Number System)

ระบบเลขจำนวนเต็มจะมีตำแหน่งจุดทศนิยมคงที่ เนื่องจากเมื่อแปลงแล้วมองดูเหมือนเลขจำนวนเต็ม สมมติเราจะใช้เลขฐานสองจำนวน 8 บิต แทนค่าของตัวเลขซึ่งเลข 8 บิตนี้ เริ่มต้นที่ค่า  $0000000_2$  และสิ้นสุดที่ค่า  $1111111_2$  มีจำนวนรวมทั้งสิ้น 256 ค่า หรือ  $2^8$  ค่า ในการนำเลขฐานสองนี้ไปแทนค่าที่จะใช้ ทำได้โดยการสมมติตำแหน่งจุดทศนิยมขึ้นมา

กำหนดให้  $N$  คือ จำนวนบิตที่อยู่ก่อนหน้าจุดทศนิยม และ  $M$  คือ จำนวนบิตที่อยู่ด้านหลังจุดทศนิยมหรือเขียนเป็นสัญลักษณ์ว่า



ค่าที่แต่ละบิตแทนคือ  $2^{N-1} \dots 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \dots 2^{-M}$   
 และค่าที่แทนได้ คือ 0 และนับขึ้นไปทีละ  $2^{-M}$  จนกระทั่งถึง  $2^N - 2^{-M}$  ซึ่งเป็นค่าสุดท้าย

ตัวอย่างเช่น สำหรับเลข 8 บิต ถ้า  $N=8, M=0$  หรือ จุดทศนิยมอยู่ขวามือสุด จะได้

เขียนเป็นสัญลักษณ์คือ  $X \ X \ X \ X \ X \ X \ X \ X$ .  
 ค่าของแต่ละบิต  $2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$   
 ค่าที่สามารถแทนได้คือ 0, 1, 2, ..., 255

ถ้า  $N=0, M=8$  หรือ จุดทศนิยมอยู่ขวามือสุด จะได้

เขียนเป็นสัญลักษณ์คือ  $.X \ X \ X \ X \ X \ X \ X \ X$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของแต่ละบิต คือ  $2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5} 2^{-6} 2^{-7} 2^{-8}$

ค่าที่สามารถแทนได้ คือ  $0, 2^{-8}, 2(2^{-8}), 3(2^{-8}), \dots, 1-2^{-8}$

ซึ่งเท่ากับ  $0.00390625, 0.0078125, 0.01171875, \dots, 0.99609375$

ถ้า  $N=3, M=5$  จะได้

เขียนเป็นสัญลักษณ์ คือ  $X X X . X X X X X$

ค่าของแต่ละบิต คือ  $2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5}$

ค่าที่สามารถแทนได้ คือ  $0, 2^{-5}, 2(2^{-5}), 3(2^{-5}), \dots, 2^3-2^{-5}$

ซึ่งเท่ากับ  $0.03125, 0.0625, 0.09375, \dots, 3.96875$

จะเห็นได้ว่า ตำแหน่งของจุดทศนิยมมีผลต่อค่าที่แทนได้ ดังนั้น ในระบบการใช้เลขจำนวนเต็มเพื่อแทนค่าตัวเลขทศนิยม นอกจากระบุจำนวนบิตที่จะใช้แล้ว ยังต้องบอกว่าใช้จุดทศนิยมอยู่ที่ตำแหน่งไหน หรือบอกว่า  $N$  กับ  $M$  มีค่าเท่ากับเท่าไร เมื่อเลือกตำแหน่งจุดทศนิยมแล้ว ให้คงตำแหน่งไว้ที่ตำแหน่งนั้นตลอดการคำนวณหนึ่งๆ

ในกรณีที่ค่าที่ต้องการแทนไม่ลงตัวพอดีเป็นค่าที่สามารถแทนได้ เราจะต้องทำการปัดเศษให้เป็นค่าที่ใกล้เคียงกับค่าที่ต้องการแทนมากที่สุด ซึ่งวิธีทำมีดังนี้ คือ

ถ้าให้  $a$  เป็นค่าที่ต้องการแทน และ  $0 \leq a < 2^N$  และสมมติให้  $a_{fix}$  คือ ค่าหลังจากเปลี่ยนเป็นรูปแบบจำนวนเต็มแล้ว เราสามารถหา  $a_{fix}$  ซึ่งเป็นตัวแทนที่ใกล้ค่าของ  $a$  มากที่สุด ดังนี้

$$a_{fix} = \text{round}(a \times 2^M) \quad (2.30)$$

โดยนิยามให้  $\text{round}()$  คือ ฟังก์ชันในการปัดเศษเป็นจำนวนเต็ม ถ้าค่าที่ใส่ให้มีส่วนทศนิยมที่น้อยกว่า 0.5 ก็จะตัดทิ้ง แต่ถ้ามากกว่าหรือเท่ากับ 0.5 ก็จะปัดเพิ่มเป็น 1

การปัดเศษนี้จะทำให้เกิดความคลาดเคลื่อนขึ้น โดยค่าที่  $a_{fix}$  แทน จะเป็น  $\frac{a_{fix}}{2^M}$  ซึ่งไม่เท่ากับค่า  $a$  เริ่มต้นที่ต้องการ ดังนั้นในการแปลงเป็นเลขจำนวนเต็ม ค่าความคลาดเคลื่อนที่เกิดขึ้น คือ

$$err = \frac{a_{fix}}{2^M} - a \quad (2.31)$$

ตัวอย่างที่ 2.1 การแปลงเลข 0.66 เป็นจำนวนเต็ม 8 บิต โดยทำ 2 กรณี คือ กรณีที่

$M=8$  และ  $M=7$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณี  $M=8$ ,  $a=0.66$  จะได้

$$\begin{aligned} a_{\text{fix}} &= \text{round}(0.66 \times 2^8) \\ &= \text{round}(168.96) \\ &= 169 \end{aligned}$$

ค่า  $a=169$  นี้สามารถแทนด้วยเลขฐานสองที่มีจำนวน 8 บิต ได้แน่นอน ซึ่งในที่นี้ได้  $a=10101001_2$  ในข้อต่อไปจะย่อโดยตอบเป็นเลขฐานสิบเท่านั้น

ค่า  $a_{\text{fix}}=169$  นี้ เทียบเท่ากับค่าทศนิยมที่แทนอยู่ คือ  $\frac{a_{\text{fix}}}{2^8} = 0.66015625$  ซึ่งหมายถึงว่าค่า  $a=0.66$  ที่แรกนั้นไม่สามารถแทนได้พอดีด้วยรูปแบบจำนวนเต็ม 8 บิต แต่ค่าที่ใกล้เคียงที่สุดที่สามารถแทนได้ คือ  $0.66015625$  ซึ่งตรงกับค่า  $a_{\text{fix}}=169$  ดังนั้นความคลาดเคลื่อนที่เกิดขึ้นในการแทนค่าตัวเลขนี้ ก็คือ

$$\text{err} = \frac{a_{\text{fix}}}{2^8} - a = 0.00015625$$

กรณี  $M=7$ ,  $a=0.66$  จะได้

$$\begin{aligned} a_{\text{fix}} &= \text{round}(0.66 \times 2^7) \\ &= 84 \end{aligned}$$

และค่าความคลาดเคลื่อนที่เกิดขึ้น คือ

$$\text{err} = \frac{a_{\text{fix}}}{2^7} - a = -0.00375$$

จะเห็นได้ว่าในกรณี  $M=7$  สามารถแทนตัวเลขได้ในช่วงที่กว้างกว่า คือแทนค่าได้ตั้งแต่ 0 ถึง 2 แต่กรณี  $M=8$  แทนได้เพียง 0 ถึง 1 แต่ทั้งสองกรณีถือว่ามีความละเอียดในการแทนค่าเท่ากัน เพราะใช้จำนวนบิตรวมเท่ากัน กล่าวคือ ถึงแม้กรณี  $M=7$  จะแทนค่าได้ในช่วงใหญ่กว่าเท่าตัว แต่มันก็มีการแบ่งขั้นที่หยากกว่า คือ มีขนาดของขั้นที่ใหญ่กว่าเท่าตัว การเลือกว่าจะใช้  $M$  เท่ากับเท่าไร จึงเหมาะกับข้อมูลชุดหนึ่งๆ ก็ขึ้นอยู่กับว่าค่าสูงสุด และต่ำสุดในข้อมูลเป็นเท่าไร

ขนาดของขั้น จะมีค่าเท่ากับค่าของบิตขวาสุด (Least Significant Bit) ซึ่งคือ  $2^{-M}$  และความคลาดเคลื่อนที่เกิดขึ้นในกรณีของการปิดเศษโดยวิธี rounding ก็จะมีค่าขอบเขตไม่เกินครึ่งหนึ่งของขั้น นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

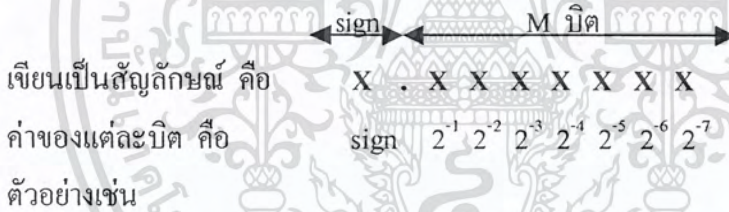
$$\begin{aligned}
 & - \text{ขนาดขั้น}/2 < \text{err} < \text{ขนาดขั้น}/2 \\
 & \text{หรือ } -2^{M+1} < \text{err} < 2^{M+1}
 \end{aligned} \tag{2.32}$$

ดังนั้น กรณี  $M=8$  จะแทนค่าได้ในช่วง  $0 \leq a < 1$  และ  $\text{err}$  ไม่เกิน  $\pm 2^{-9}$   
 และ กรณี  $M=7$  จะแทนค่าได้ในช่วง  $0 \leq a < 2$  และ  $\text{err}$  ไม่เกิน  $\pm 2^{-8}$

### 2.5.2 ระบบเลขจำนวนเต็มแบบมีเครื่องหมาย

ที่ผ่านมาเป็นการใช้ระบบเลขจำนวนเต็มแทนเฉพาะเลขจำนวนบวกเท่านั้น สำหรับการแทนทั้งจำนวนบวก และลบก็สามารถทำได้ โดยมีรูปแบบใหญ่ๆ สองแบบ คือ

1) แบบ **sign-magnitude** มีรูปแบบเหมือนเลขบวก แต่เพิ่มบิตพิเศษขึ้นมาอีกหนึ่งบิต (โดยทั่วไปคือบิตซ้ายสุด) เป็นบิตบอกเครื่องหมาย (sign bit) โดยที่เมื่อบิตเครื่องหมายเป็น 0 หมายถึงเลขบวก และถ้าเป็น 1 หมายถึงเลขลบ ขอยกตัวอย่างเลขแบบ sign-magnitude ในกรณีที่  $N=0$ , และ  $M=7$  เมื่อรวมบิตเครื่องหมายอีก 1 บิต จะต้องใช้จำนวนบิตทั้งสิ้น 8 บิต ดังนี้

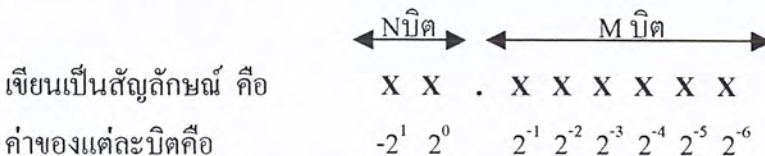


$01010100_2 = 84_{10}$  มีค่าเท่ากับ  $0.65625$  (จากค่าคอบของตัวอย่าง 2.1)

และ  $11010100_2 = -84_{10}$  ก็จะมีค่าเท่ากับ  $-0.65625$  เป็นต้น

จะเห็นได้ว่ารูปแบบ sign-magnitude สามารถเข้าใจได้ง่าย โดยเพียงแค่เติมบิตเครื่องหมายลงไปข้างหน้ารูปแบบปกติเท่านั้น อย่างไรก็ตาม รูปแบบนี้ไม่เป็นที่นิยมในการใช้ เนื่องจากการบวกลบเลขทำได้ไม่สะดวก และไม่ดีเท่ารูปแบบ 2's complement ที่จะได้กล่าวถึงต่อไป

2) แบบ **2's complement** เป็นแบบที่นิยมใช้กันมากที่สุด โดยกำหนดบิตซ้ายมือสุดมีค่าเป็น  $-2^{N-1}$  ซึ่งเป็นบิตเดียวที่มีค่าติดลบ บิตอื่นๆ มีความหมายเหมือนเดิม ตัวอย่างเช่น เลข 2's complement ขนาด 8 บิต ที่มี  $N=2$  และ  $M=6$  จะมีรูปแบบ และค่าของบิตต่างๆ ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงตัวอย่างของค่าที่แทนได้สำหรับกรณีเลข 2's complement ขนาด 8 บิตที่มี  $N=1, 2$ , และ 8 ไต่ตั้งแต่ค่าลบเป็นค่าบวก สังเกตขนาดของขั้นในแต่ละกรณียังมีค่าเท่ากับ บิตซ้ายมือสุด คือ  $2^{-M}$  เหมือนกับกรณีการแทนเลขบวกอย่างเดียว

ฐานสอง	ฐานสิบ (บวก)	ค่าที่แทน (กรณี $N=8$ )	ค่าที่แทน (กรณี $N=1$ )	ค่าที่แทน (กรณี $N=2$ )
1000 0000	128	-128	-1 (ค่าลบมากที่สุด)	-2
1000 0001	129	-127	$-1+2^{-7} = -0.9921875$	$-2+2^{-6} = -1.984375$
1000 0010	130	-126	$-1+2(2^{-7}) = -0.984375$	$-2+2(2^{-6}) = -1.96875$
...	...	...	...	...
1111 1111	255	-1	$-2^{-7} = -0.0078125$	$-2^{-6} = -0.015625$
0000 0000	0	0	0	0
0000 0001	1	1	$2^{-7} = 0.0078125$	$2^{-6} = 0.015625$
0000 0010	2	2	$2(2^{-7}) = 0.015625$	$2(2^{-6}) = 0.03125$
...	...	...	...	...
0111 1111	127	127	$1-2^{-7} = 0.9921875$ (ค่าบวกมากที่สุด)	$2-2^{-6} = 1.984375$

ตารางที่ 2.3 รูปแบบเลข 2's complement (กรณี 8 บิต) กับค่าที่แทนได้

จะเห็นได้ว่า บิตซ้ายมือสุดยังสามารถมองเป็นบิตเครื่องหมายได้ เพราะถ้าบิตซ้ายมือสุด เป็น 1 จะได้เลขจำนวนนั้นเป็นค่าลบแน่นอน และ เป็น 0 ก็จะได้ เป็นค่าบวกแน่นอน

ในระบบประมวลผลสัญญาณ นิยมใช้รูปแบบ 2' complement ที่มีจุดทศนิยมหลังบิตซ้ายมือสุด 1 บิต ( $N=1$ ) ซึ่งจะแทนเลขได้จาก -1 ถึงประมาณ 1 และผลก็คือ การคูณเลขสองจำนวนเข้าด้วยกันจะไม่เกิดโอเวอร์โฟล เพราะผลที่ได้จะไม่มีทางเกิดช่วง -1 ถึง 1

ตัวอย่างที่ 2.2 การแปลงเลข -0.595 เป็นจำนวนเต็มแบบ 2's complement 8 บิต โดยใช้  $M=7$

ในกรณีของ 2's complement ถ้าให้  $a$  เป็นค่าที่ต้องการแทน และ  $-2^{N-1} \leq a < 2^{N-1}$  สมมติให้  $a_{fix}$  คือ ค่าหลังจากการเปลี่ยนเป็นรูปแบบจำนวนเต็มแล้ว เราสามารถหา  $a_{fix}$  ซึ่งแทนค่า  $a$  ได้ใกล้เคียงที่สุด ด้วยสมการเดียวกับ 2.32 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_{fix} = \text{round}(a \times 2^M) \quad (2.33)$$

ในข้อนี้  $M=7$  และ  $N=1$  ซึ่งจะสามารถแทนค่า  $a$  ได้ระหว่าง  $-1$  ถึง  $1$  ซึ่ง  $a = -0.595$  ดังนั้นจะได้

$$\begin{aligned} a_{fix} &= \text{round}(-0.595 \times 2^7) \\ &= \text{round}(-76.16) \\ &= -76 \end{aligned}$$

ค่า  $-76$  สามารถแปลงกลับทศนิยมได้เป็น  $-76/2^7 = -0.59375$  ซึ่งมีความคลาดเคลื่อนจากค่า  $a$  เท่ากับ

$$\frac{a_{fix}}{2^7} - a = -0.59375 - (-0.595) = 0.00125$$

## 2.6 ความคลาดเคลื่อนจากการใช้ระบบเลขจำนวนเต็ม

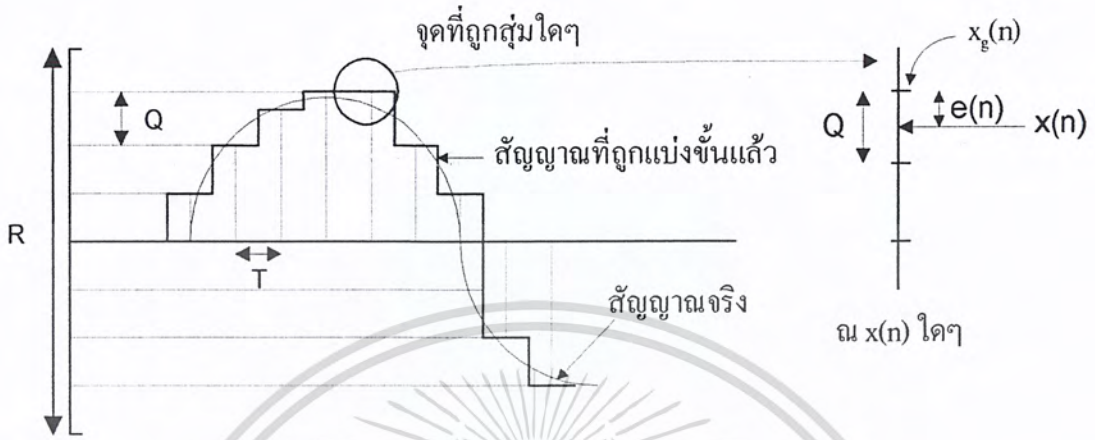
### 2.6.1 การแบ่งขั้นสัญญาณ (Signal Quantization)

กระบวนการที่เกิดความคลาดเคลื่อนในระบบคือ การแบ่งขั้นสัญญาณ ซึ่งกระทำหลังจากการสุ่มสัญญาณ การแบ่งขั้นสัญญาณ หมายถึง การแทนค่าสัญญาณที่ถูกสุ่มมาซึ่งมีความละเอียดไม่จำกัด (เพราะมาจากระดับสัญญาณอนาล็อก) ด้วยระบบเลขฐานสองที่มีจำนวนบิตจำกัด ซึ่งก็จะเกิดความคลาดเคลื่อนจากการแทนค่าขึ้น การแบ่งขั้นสัญญาณนี้ในทางปฏิบัติรวมอยู่ในตัวแปลงอนาล็อกเป็นดิจิตอล (A/D) แต่ในทางทฤษฎีแยกการแปลงสัญญาณอนาล็อกเป็นดิจิตอลเป็นสองกระบวนการ คือ การสุ่มซึ่งทำให้ได้สัญญาณไม่ต่อเนื่อง และการแบ่งขั้นสัญญาณซึ่งทำให้ได้สัญญาณที่มีความละเอียดทางขนาดจำกัดลง

ในที่นี้จะอธิบายเฉพาะกรณีที่ใช้ระบบเลขจำนวนเต็ม (fixed-point) ซึ่งจะให้ขนาดของขั้นคงที่ตลอดช่วงที่แทนค่าสัญญาณ การใช้จำนวนบิตยิ่งมากเท่าไรก็จะได้การแบ่งขั้นที่ละเอียด และแทนสัญญาณจริงได้ถูกต้องมากเท่านั้น ถ้าเราให้  $B$  เป็นจำนวนบิตที่ใช้แทน  $1$  ค่าของสัญญาณ  $R$  เป็นช่วงของค่าสัญญาณที่สามารถแทนได้ และ  $Q$  แทนเป็นความกว้างของขั้น จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{จำนวนขั้นทั้งหมด} = \frac{R}{Q} = 2^B \quad (2.34)$$



รูปที่ 2.11 แสดงการแบ่งขั้นสัญญาณ (เส้นประในแกนตั้งเป็นตำแหน่งที่มีการสุ่ม)

สัญญาณก่อนที่แบ่งขั้น จะต้องมียุ่ในช่วง  $-R/2$  ถึง  $R/2$  ถ้าเราพิจารณาตัวอย่างสัญญาณที่จุดหนึ่งๆ คือ  $x(n)$  ซึ่งหลังจากการแบ่งขั้นแล้วจะมีค่าเปลี่ยนไปเพื่อให้ตรงกับขั้นที่แทนค่าได้ สมมติว่าได้ค่าใหม่เป็น  $x_q(n)$  ถ้าเราใช้กฎเกณฑ์การปัดเศษหาขั้นที่ใกล้ที่สุด (rounding) จะได้ว่า  $x(n)$  ที่มีค่าอยู่ระหว่าง  $x_q(n) - Q/2$  ถึง  $x_q(n) + Q/2$  จะถูกปัดเป็นค่า  $x_q(n)$  ซึ่งก็จะทำให้ผลต่างระหว่าง  $x_q(n)$  และ  $x(n)$  มีค่าไม่เกิน  $Q/2$  หรือครึ่งหนึ่งของขั้น

เรานิยามผลต่างนี้เป็น  $e(n)$  ซึ่งเป็นฟังก์ชันของความคลาดเคลื่อนที่เกิดขึ้นจากการแบ่งขั้นสัญญาณ จะได้

$$e(n) = x_q(n) - x(n) \quad (2.35)$$

$$\text{และ} \quad -Q/2 < e(n) < Q/2 \quad (2.36)$$

จากสมการค่าที่จะแทนเป็นจุดในสัญญาณ เพราะฉะนั้นค่าคลาดเคลื่อนก็จะเกิดขึ้นที่ทุกๆ จุดของสัญญาณ และค่า  $Q$  หาได้จากสมการที่ 2.35 โดยจะมีหน่วยเป็นแรงดันของสัญญาณอนาล็อก

ถ้าสมมติให้  $x_q(n)$  ซึ่งเป็นสัญญาณหลังจากที่ถูกแทนค่าด้วยจำนวนเต็ม เราอาจมองว่าความคลาดเคลื่อนที่เกิดขึ้นกับ  $x_q(n)$  เป็นเหมือนสัญญาณรบกวนที่ถูกเติมเข้าไปในสัญญาณเริ่มแรกนั่นคือ  $x_q(n) = x(n) + e(n)$  ดังนั้น  $e(n)$  จึงมีอีกชื่อหนึ่งว่า เป็นสัญญาณรบกวนที่เกิดจากการแบ่งขั้นสัญญาณ (quantization noise)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน  $e(n)$  นี้เป็นฟังก์ชันที่มีค่าไม่แน่นอน (random) ชนิดหนึ่ง เพราะเราไม่สามารถรู้ล่วงหน้าได้ว่าแต่ละค่าของ  $e(n)$  จะเป็นเท่าไรแน่ (เพราะไม่รู้ว่าสัญญาณที่จะแทนจะมีรูปร่างอย่างไร) แต่รู้ว่า สัญญาณความคลาดเคลื่อนที่จะเกิดจะมีค่าอยู่ระหว่าง  $-Q/2$  ถึง  $Q/2$  เท่านั้นเพื่อให้ง่ายต่อการวิเคราะห์ เราจะสมมติให้ค่าของ  $e(n)$  ที่มีการกระจายตัวสม่ำเสมอ (uniform distribution) ในช่วงของค่าระหว่าง  $-Q/2$  ถึง  $Q/2$  หรือมีฟังก์ชันการกระจายตัวของโอกาส (probability density function) เป็น

$$p(e) = \begin{cases} 1/Q, & -Q/2 < e < Q/2 \\ 0, & e = \text{other} \end{cases} \quad (2.37)$$

จากทฤษฎีความน่าจะเป็น เราจะได้ว่าค่าเฉลี่ยของ  $e$  มีค่าเท่ากับ 0 และค่าเฉลี่ยของกำลังสองของ  $e$  มีค่าเท่ากับ  $Q^2/12$  ซึ่งหาได้จาก

$$P_q = \int_{-Q/2}^{Q/2} e^2 p(e) \cdot de = \frac{Q^2}{12} \quad (2.38)$$

ค่าเฉลี่ยของกำลังสองของสัญญาณนี้ ก็คือ ค่ากำลังเฉลี่ยนั่นเอง ดังนั้น ขอใช้สัญลักษณ์แทนค่านี้ว่า  $P_q$  ซึ่งในที่นี้เราหาค่ากำลังเฉลี่ยของสัญญาณรบกวนจากการแบ่งขั้นสัญญาณได้เท่ากับ  $Q^2/12$  สังเกตว่ากำลังของสัญญาณรบกวนจะแปรตาม  $Q^2$  ซึ่งแปรผกผันกับจำนวนบิต

ถ้าเราทราบว่ากำลังเฉลี่ยของสัญญาณมีค่าเท่ากับ  $P_s$  เราสามารถหาค่าอัตราส่วนระหว่างกำลังของสัญญาณต่อกำลังของสัญญาณรบกวน หรือ SNR ที่เนื่องมาจากสัญญาณรบกวนของการแบ่งขั้นสัญญาณได้ คือ

$$SNR_{A/D} = \frac{P_s}{P_q}$$

$$\text{หรือ } SNR_{A/D} \text{ (dB)} = P_s \text{ (dB)} - P_q \text{ (dB)} \quad (2.39)$$

$P_s$  อาจหาได้จาก  $v_{rms}^2$  ซึ่งค่า  $v_{rms}$  ของสัญญาณหนึ่งๆ นั้นขึ้นกับลักษณะรูปร่างของสัญญาณในการวิเคราะห์ต่อไป เราอาจนิยามค่าคุณลักษณะของสัญญาณค่าหนึ่ง เพื่อบ่งบอกถึงค่า  $v_{rms}$  ของสัญญาณ ซึ่งค่านี้ก็คือ loading factor หรือ LF ซึ่งเป็นค่าอัตราส่วนระหว่าง  $v_{rms}$  ต่อ  $v_p$  (Peak Voltage) ของสัญญาณหนึ่งๆ นั่นคือ

$$LF = \frac{V_{rms}}{V_p} \quad (2.40)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่มีระดับสัญญาณสูงสุดอยู่ห่างจากระดับสัญญาณ ณ เวลาอื่นๆ มาก จะมีค่า LF ที่ต่ำกว่าสัญญาณที่มีระดับสัญญาณสูงสุดอยู่พอๆ กับระดับสัญญาณ ณ เวลาอื่นๆ เช่น สัญญาณชานนี้  $v_{rms} = \frac{V_p}{\sqrt{2}}$  ดังนั้น จะมี  $LF = \frac{1}{\sqrt{2}} = 0.707$  ส่วนสัญญาณสี่เหลี่ยมมี  $v_{rms} = v_p$  ดังนั้น จะมี  $LF = 1$  ซึ่ง เป็นค่าที่มากที่สุดเท่าที่จะเป็นไปได้สำหรับสัญญาณใดๆ สำหรับสัญญาณเสียงคนพูด มีค่า LF ประมาณ 0.15 - 0.2 เป็นต้น

จากนิยามของ LF เราสามารถหาค่ากำลังของสัญญาณได้เป็น

$$P_s = v_{rms}^2 = (LF)^2 v_p^2 \quad (2.41)$$

สมมติว่าเราสามารถรู้ค่าสูงสุดของสัญญาณ คือ รู้ว่าสัญญาณมีค่าอยู่ระหว่าง  $-v_p$  และ  $+v_p$  และสามารถเลือกค่าในช่วงนี้เป็นช่วงที่ควมแปลงแอนาลอกเป็นดิจิตอลสามารถแทนค่าได้พอดี นั่นคือ ค่า R ในรูปที่ 2.12 มีค่าเท่ากับ  $2v_p$  เมื่อแทนค่า R ลงในสมการที่ 2.35 จะได้ว่า ขนาดของ 1 ขั้วของสัญญาณมีค่าเป็น

$$Q = \frac{2v_p}{2^B} \quad (2.42)$$

ดังนั้นได้ SNR อันเนื่องมาจากการแบ่งขั้วสัญญาณ มีค่าเป็น

$$\begin{aligned} SNR_{A/D} &= \frac{P_s}{P_q} = \frac{(LF)^2 v_p^2}{\left(\frac{2v_p}{2^B}\right)^2} \\ &= 3(LF)^2 (2^{2B}) \quad \text{แปลงเป็น dB โดยใส่ } 10\log() \text{ จะได้} \\ SNR_{A/D} &= 10 \log\{3(LF)^2 (2^{2B})\} dB \\ &= (2B)10 \log(2) + 10 \log(3) + 20 \log(LF) dB \\ &= 6.02B + 4.77 + 20 \log(LF) dB \end{aligned} \quad (2.43)$$

ข้อสังเกต และพิจารณาสำหรับ ค่า  $SNR_{A/D}$  ที่ได้จากสมการที่ 2.42 นี้ คือ

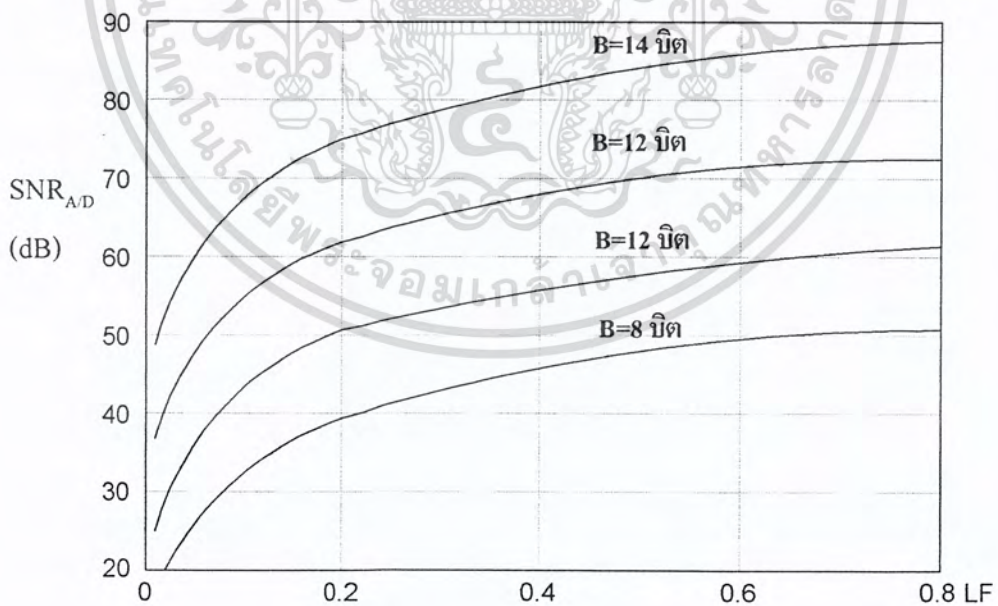
- 1)  $SNR_{A/D}$  มีค่าแปรตามจำนวนบิต (B) โดย SNR จะมีค่ามากขึ้น 6dB ต่อ 1 บิตที่เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2)  $SNR_{A/D}$  ยังแปรตามค่า LF ดังแสดงในรูปที่ 2.13 ซึ่งวาดกราฟระหว่าง  $SNR_{A/D}$  กับ LF ที่จำนวนบิตต่างๆ

3) ค่า  $SNR_{A/D}$  ที่ได้นี้เป็นค่าที่มากเกินไปกว่าที่เราจะได้จริงในทางปฏิบัติ เนื่องจากการหาค่า  $SNR_{A/D}$  นี้ เราได้สมมติให้ค่าสูงสุดของสัญญาณ เป็นค่าสูงสุดที่ A/D สามารถแปลงค่าได้พอดี แต่ในทางปฏิบัติเราอาจไม่รู้ค่าสูงสุดของสัญญาณ จึงมักเพื่อให้ค่าสูงสุดที่ A/D สามารถแปลงได้ มีค่ามากกว่าช่วงที่เราใช้งานเล็กน้อย นอกจากนี้ภายในวงจร A/D เองในทางปฏิบัติก็จะมี ความคลาดเคลื่อนในการแปลงค่าด้วย ซึ่งก็จะส่งผลให้มีสัญญาณรบกวนอันเนื่องมาจากความคลาดเคลื่อนของวงจร A/D เองปนเข้ามาอีก

4) การเลือกว่า จะใช้ข้อมูลกี่บิตแทน 1 ค่าสัญญาณ นอกจากขึ้นกับลักษณะของงานว่า สามารถยอมรับสัญญาณรบกวนได้ที่เท่าไรแล้ว ก็ยังขึ้นกับ SNR ของสัญญาณอนาล็อกขาเข้าด้วย สัญญาณขาเข้า โดยทั่วไป มักจะมีสัญญาณรบกวนปนมาจากแหล่งอื่นอยู่แล้ว ดังนั้น จะมีค่า SNR อยู่ค่าหนึ่ง ถ้าเราใช้จำนวนบิตที่ให้  $SNR_{A/D}$  สูงกว่า SNR ของสัญญาณขาเข้ามากๆ ก็อาจ จะไม่มีประโยชน์มากนักเทียบกับต้นทุนที่เพิ่มขึ้น เพราะไม่ว่าจำนวนบิตจะเป็นเท่าไรก็ตาม SNR ของสัญญาณขาออกก็จะยังถูกจำกัดด้วย SNR เริ่มต้นของสัญญาณขาเข้า



รูปที่ 2.12 แสดงค่า  $SNR_{A/D}$  ในฟังก์ชันของจำนวนบิต (B) และ loading factor (LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2 ความคลาดเคลื่อนจากการปัดเศษสัมประสิทธิ์ (Coefficient Rounding)

ความคลาดเคลื่อนจากการปัดเศษสัมประสิทธิ์ เกิดจากการที่เรานำค่าสัมประสิทธิ์ของระบบที่ได้ออกแบบไว้ไปใช้กับระบบเลขจำนวนเต็ม ซึ่งต้องแทนค่าสัมประสิทธิ์ทุกตัวด้วยจำนวนเต็มที่มีจำนวนบิตจำกัด ซึ่งก็แน่นอนว่าจะต้องเกิดความคลาดเคลื่อนกับค่าสัมประสิทธิ์เหล่านั้น การแทนค่าสัมประสิทธิ์ด้วยจำนวนเต็มนี้ เรียกอีกอย่างหนึ่งว่า การปัดเศษสัมประสิทธิ์เพราะเป็นการปัดให้ค่าสัมประสิทธิ์ไปลงที่ค่าที่สามารถแทนค่าได้ด้วยจำนวนเต็ม

ความคลาดเคลื่อนที่เกิดขึ้นกับสัมประสิทธิ์จะส่งผลถึงความคลาดเคลื่อนของลักษณะของระบบจากที่เคยออกแบบไว้ ได้แก่ ผลตอบสนองเชิงความถี่ที่มีรูปร่างเปลี่ยนไป ความถี่ตัดมีค่าเปลี่ยนไป ผลดังกล่าวจะมีมากขึ้นอยู่กับ

- 1) จำนวนบิตที่ใช้ จำนวนบิตน้อยลงย่อมส่งผลกระทบต่อมาก
- 2) ชนิดของตัวกรอง ผลของความคลาดเคลื่อนต่อตัวกรอง FIR จะมากกว่าตัวกรอง BPF และ BSF ที่มีแถบความถี่แคบๆ มีแนวโน้มที่จะได้รับผลกระทบมากกว่าตัวกรองแบบอื่นๆ
- 3) อันดับของตัวกรอง ผลกระทำต่อตัวกรองที่มีอันดับสูงจะมากกว่าตัวกรองที่มีอันดับต่ำ ซึ่งเราสามารถแก้ไขได้โดยการแตกตัวกรองที่มีอันดับสูง ให้อยู่ในโครงสร้างอนุกรม หรือขนาดของตัวกรองที่มีอันดับสอง

ความคลาดเคลื่อนของสัมประสิทธิ์ทำให้คุณลักษณะของระบบเปลี่ยนไป แต่จะเปลี่ยนไปอย่างไรมากน้อยแค่ไหน และอยู่ในภาวะที่ยอมรับได้หรือไม่ นั้น ยังไม่สามารถหาสูตรทั่วไปที่จะบอกได้ แนวทางที่ดีที่สุดคือ ลองปัดเศษสัมประสิทธิ์ดูเป็นเฉพาะกรณีๆ ไป แล้วนำสัมประสิทธิ์ชุดใหม่ที่ได้หลังจากการปัดเศษไปตรวจสอบดู เช่น ถ้าเป็นตัวกรองดิจิทัลก็ตรวจสอบว่าผลตอบสนองเชิงความถี่เปลี่ยนไปอย่างไร

สมมติ ในระบบมีสัมประสิทธิ์ค่าหนึ่งเท่ากับ  $a$  และเราต้องการนำไปใช้กับตัวเลข 16 บิต (2's complement) ก่อนอื่นต้องเลือกตำแหน่งจุดทศนิยมของเลขจำนวนเต็มก่อน ในกรณีที่ค่าสัมประสิทธิ์ทุกค่ามีค่าอยู่ระหว่าง  $-1$  ถึง  $1$  เราสามารถเลือกให้  $N=1$  และ  $M=15$  ได้ แต่ถ้ามีสัมประสิทธิ์บางตัวที่มีค่าเกินช่วง  $-1$  ถึง  $1$  ก็อาจต้องเลือกให้  $N$  มากกว่า  $1$

หลังจากเลือกจำนวนบิตและตำแหน่งของจุดทศนิยมแล้ว ก็สามารถหาค่าของสัมประสิทธิ์ที่จะมีค่าเปลี่ยนไปได้ สมมติให้  $\hat{a}$  คือค่าของสัมประสิทธิ์หลังการปัดเศษ จะได้ว่า

$$\hat{a} = \text{round}(a \times 2^M) / 2^M \quad (2.44)$$

เราจะใช้สัมประสิทธิ์นี้กระทำกับสัมประสิทธิ์ทุกค่า และจะได้สัมประสิทธิ์ชุดใหม่ คือชุดหลังการปิดเศษ ซึ่งจะนำไปวิเคราะห์หาผลตอบสนองเชิงความถี่ หรือลักษณะอื่นๆ ต่อไป

## 2.7 ข้อมูลโดยทั่วไปของ TMS320C50

- ใช้เทคโนโลยีแบบ CMOS
- ชิพ DSP ตระกูล C5X สามารถประมวลผลได้มากกว่า 28 MIPS (Million Instruction Per Second)
- วงรอบคำสั่งใช้เวลา 50 ns
- มี RAM ภายในชิพขนาด 9 กิโลเวิร์ด
- มี ROM ภายในชิพขนาด 2 กิโลเวิร์ด
- สามารถต่อหน่วยความจำโปรแกรม / ข้อมูล ภายนอกได้สูงสุด 224 กิโลเวิร์ด
- มีหน่วยประมวลผลทางคณิตศาสตร์/แอดคิวมูเลเตอร์/แอดคิวมูเลเตอร์บัฟเฟอร์ขนาด 32 บิต
- สามารถรองรับข้อมูลการคูณเลขสองจำนวนขนาด 16 บิต ได้ผลลัพธ์ที่ได้จะมีขนาด 32 บิต
- มีออกซิลเลอรีรีจิสเตอร์ (Auxiliary) มากถึงแปดรีจิสเตอร์ (AR0-AR7)
- มีฮาร์ดแวร์สเตจ 8 ระดับ
- มีเซอร์คิวลาร์บัฟเฟอร์ (Circular Buffer) จำนวนสองชุด
- มีอินพุต/เอาต์พุต แบบขนาน ขนาด 64 กิโลไบต์ จำนวน 16 พอร์ต
- มีอินพุต/เอาต์พุต แบบอนุกรม 2 พอร์ต
- มีชุดคำสั่งที่สามารถเข้าได้กับตระกูล TMS320C2X
- มีตัวกำเนิดสัญญาณนาฬิกาภายในชิพ
- สามารถเรียกอินเตอร์รัปต์ (External Interrupt) จากภายในได้ 4 อินเตอร์รัปต์ ( $\overline{INT0} - \overline{INT4}$ )

## 2.8 สถาปัตยกรรมของ TMS320C50

โครงสร้างภายในของ TMS320C50 จะประกอบด้วยสามส่วนหลักๆ คือ

- 1) หน่วยประมวลผลกลาง (CPU) ใช้ในการประมวลผลทางคณิตศาสตร์โดยส่วน ALU (Arithmetic Logic Unit) ในการคำนวณทางตรรกะโดยส่วน PLU (Parallel Logic Unit) และใช้ในการควบคุมส่วนต่างๆของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หน่วยความจำใช้สำหรับเก็บข้อมูลหรือโปรแกรม

3) เพอริเฟอรัลอินเตอร์เฟส (Peripheral-interface Circuit) เป็นส่วนที่ต่ออุปกรณ์ภายนอก และมีหน้าที่ทำให้อุปกรณ์เหล่านั้นทำงานสอดคล้องกัน

TMS320C50 จะทำการคำนวณทางคณิตศาสตร์ด้วยวิธีทำ 2S-Complement โดยใช้หน่วยประมวลผล และแอสคิมมูลเตอร์ขนาด 32 บิต โดยที่ แอสคิมมูลเตอร์สามารถแบ่งใช้งานได้เป็นบิตสูงและบิตต่ำ มีคำสั่งในการใช้งานแยกระหว่างบิตต่ำและบิตสูงด้วย เพื่อความรวดเร็วก็มีแอสคิมมูลเตอร์เพื่อเก็บข้อมูลชั่วคราว คือแอสคิมมูลเตอร์บัฟเฟอร์ มีขนาด 32บิต เช่นเดียวกับแอสคิมมูลเตอร์

ยังมี Parallel Logic Unit (PLU) เพื่อการประมวลผลทางตรรกะ โดยไม่ขึ้นอยู่กับความจุของแอสคิมมูลเตอร์ ในส่วน PLU นี้จะจัดบิตให้ผู้ใช้ โปรแกรมตามต้องการได้ เพื่อความรวดเร็วและง่ายในการใช้งาน ในการเซตบิต เล็ยบิต และเทสบิต โดยการควบคุมและเช็คสถานะการทำงานของรีจิสเตอร์

ในการคูณเลข 16 บิตสองจำนวน โดยใช้การประมวลผลแบบ 2S-Complement จะได้ผลลัพธ์เป็น 32 บิต การคูณกันจะใช้รีจิสเตอร์สามตัว ได้แก่ ตัวที่จะนำมาคูณอาจเป็น TREG1 หรือ TREG2 ก็ได้ PREG (Product Register) ไว้สำหรับเก็บผลการคูณ และ TREG0 ซึ่งจะนำมาไว้เก็บค่าตัวตั้งในการคูณ

มี Hardware Stack ทั้งหมด 8 ระดับ เพื่อการเก็บค่าโปรแกรมเคาเตอร์ (Program Counter) ระหว่างที่มีการเรียกอินเตอร์รัป (Interrupt) และการเรียก Subroutine ในการอินเตอร์รัปจะเก็บค่าของรีจิสเตอร์ต่างๆ ได้แก่ ACC, ACCB, ARCR, INDX, PREG, ST0, ST และ TREGs ลงในหน่วยความจำสแต็ก (stack) และจะนำค่ารีจิสเตอร์เหล่านี้ กลับคืนมาเมื่ออินเตอร์รัปเสร็จสิ้น

การต่อหน่วยความจำมีสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ คือ ข้อมูลขนาด 16 บิต (D15-D0) แอสคิมมูลขนาด 16 บิต (A15-A0) สัญญาณเลือกชนิดของหน่วยความจำที่จะติดต่อก็คือ  $\overline{DS}$ ,  $\overline{PS}$ ,  $\overline{IS}$  สัญญาณการอ่าน/เขียน ( $R/\overline{W}$ ) และสัญญาณกำหนดไทม์มิงในการติดต่อ ( $\overline{STRB}$ )

พอร์ตอนุกรม มีลักษณะฟูลดูเพล็กซ์ (FuLL-Duplex) สามารถติดต่อกับอุปกรณ์ที่รับส่งข้อมูลแบบอนุกรมได้โดยตรง โดยที่รีจิสเตอร์ของพอร์ตอนุกรมทั้งสองรีจิสเตอร์ไว้คอยรับส่งข้อมูล และสามารถรับส่งข้อมูลได้มากที่สุด 14 บิต

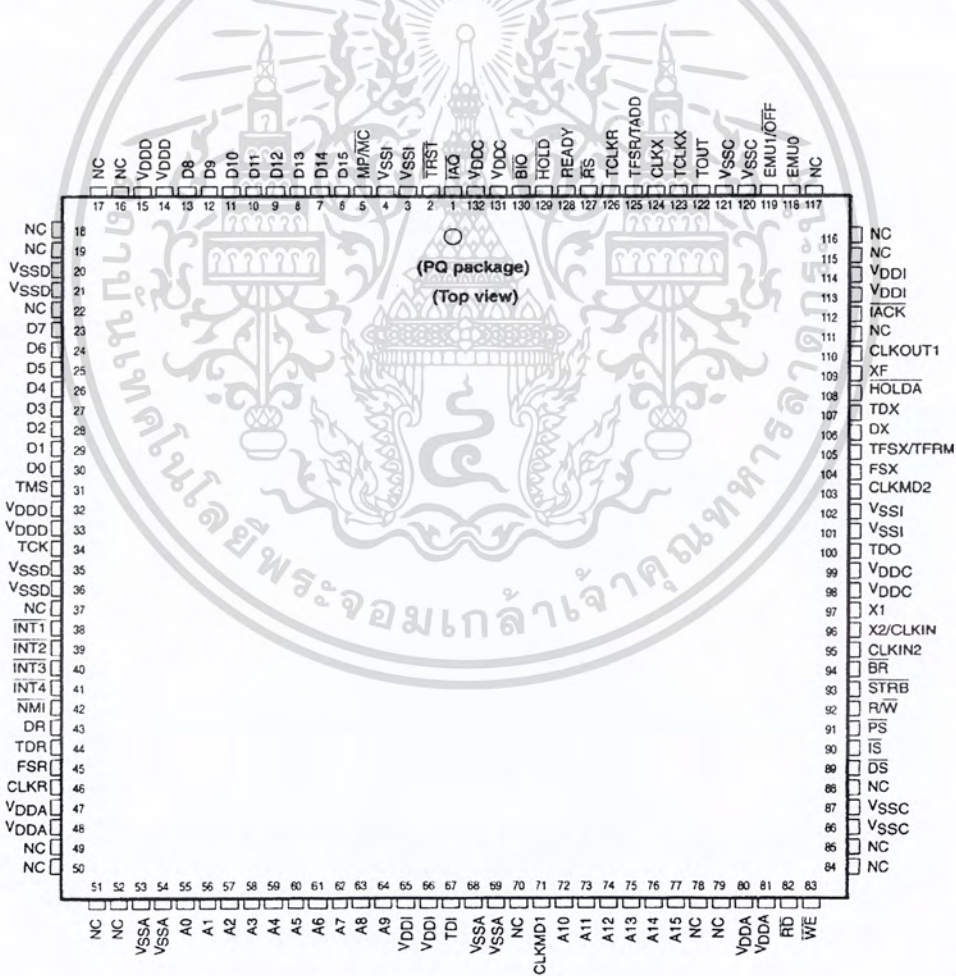
## 2.9 ลักษณะโดยทั่วไปของ TMS320C50

เป็นชิปที่มีขาทั้งหมด 132 ขา เป็นแบบ QFP(quad flat pack) ดังรูปที่ 2.17 มีขาสัญญาณแอสคิมมูลและขาสัญญาณข้อมูลอย่างละ 16 ขา เพื่อความเร็วในการส่งรับข้อมูลการส่งข้อมูลจะเป็นแบบขนาน จะมีการทำงาน 2 โหมดคือ ไมโครโปรเซสเซอร์ (Microprocessor) และโหมดไมโครเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ (Microcomputer) สามารถเลือกการทำงานในแต่ละโหมดได้ที่ขา  $MP/\overline{MC}$  ถ้ามีสัญญาณแรงดันต่ำเข้ามาจะทำงานที่โหมดไมโครคอมพิวเตอร์เราสามารถควบคุมการทำงานให้เป็นโหมดใดก็ได้โดยการผ่านทางรีจิสเตอร์ PMST

จากรูปที่ 2.13 จะเห็นว่าผู้ใช้สามารถอินเทอร์รับจากภายนอกได้ทั้งหมด 4 อินเทอร์รับต์ ( $\overline{INT1} - \overline{INT4}$ ) ขาที่ 38-41 ตามลำดับ

สัญญาณ  $\overline{DS}, \overline{PS}$  และ  $\overline{IS}$  เป็นสัญญาณที่เลือกใช้หน่วยความจำข้อมูลหน่วยความจำโปรแกรม หรือ อินพุท/เอาต์พุทพอร์ท เมื่อ TMS320C50 ส่งสัญญาณ  $\overline{DS}, \overline{PS}$  และ  $\overline{IS}$  ออกไป จะได้รับสัญญาณ Ready กลับมา เป็นสัญญาณบอกว่า หน่วยความจำหรืออุปกรณ์อินพุท/เอาต์พุท นั้นว่างหรือไม่ ถ้าไม่ว่าง TMS320C50 จะรออีก 1 สัญญาณนาฬิกาแล้วจะส่งสัญญาณ  $R/\overline{W}$  ออกไปเพื่อบอกหน่วยความจำหรืออุปกรณ์ อินพุท/เอาต์พุท ว่าต้องการอ่านหรือเขียน โปรแกรม



รูปที่ 2.13 แสดงขาต่างๆ ของ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับและส่งข้อมูลผ่านทาง ซีเรียลพอร์ต จะส่งข้อมูลผ่านเข้าทางขา RD และข้อมูลออกทางขา DX ในระหว่างการรับข้อมูลจากขา DR ไปให้กับ RSR (Serial port receive shift register) จะต้องมีการป้อนสัญญาณนาฬิกาเข้ามาสำหรับทำหน้าที่เลื่อนข้อมูล สัญญาณนาฬิกาที่ว่านั้นคือสัญญาณนาฬิกาจากขา CLKR ในทำนองเดียวกันการส่งข้อมูลจาก XSR (Serial port transmit shift register) ไปให้กับ DX ก็จะต้องมีสัญญาณนาฬิกาจากขา CLKX มาช่วยในการเลื่อนข้อมูล สัญญาณที่สำคัญในการส่งข้อมูลและรับข้อมูลผ่านทางซีเรียลพอร์ตคือ สัญญาณ FSR (Frame synchronization pulse for receive) และ FSX (Frame synchronization pulse for transmit input/output) ที่ขอบล่างของทั้งสองสัญญาณเกิดขึ้นถึงจะเกิดการเลื่อนข้อมูลที่ RSR และ XSR

## 2.10 กลุ่มรีจิสเตอร์ที่สำคัญ

- แอคคิวมูเลเตอร์ (ACC) มีขนาด 32 บิต สามารถแบ่งใช้งานได้เป็นบิตสูง (ACCH) และ บิตต่ำ (ACCL)
- ออกซิลเลอรีรีจิสเตอร์ มีทั้งหมด 8 ตัว แต่ละตัวมีขนาด 16 บิต ใช้ในการอ้างอิงตำแหน่งแอด (AR0-AR7) เครื่องทางอ้อมใช้เป็นตัวชี้ตำแหน่งของ Circular Buffer
- ออกซิลเลอรีรีจิสเตอร์ ใช้สำหรับเปรียบเทียบค่าที่อยู่ในออกซิลเลอรีรีจิสเตอร์ โดยผลของการคอมแพริจิสเตอร์ (ARCR) เปรียบเทียบจะถูกแสดงใน บิต TC ของรีจิสเตอร์ ST0
- อินดีกรีจิสเตอร์ (INDX) สามารถถูกใช้โดยส่วน ARAU ไว้สำหรับเพิ่มค่าหรือลด ค่าที่อยู่ใน ARn
- เซอร์คิวลาร์บัฟเฟอร์ (Circular Buffer) จะประกอบไปด้วย CBSR1, CBSR2 สำหรับกำหนดตำแหน่งเริ่มต้นของ Circular Buffer ตัวที่ 1 และ 2, CBER1, CBER2 สำหรับกำหนดตำแหน่งสุดท้ายของ Circular Buffer ตัวที่ 1 และ 2 ส่วน CBCR จะใช้เป็นตัวควบคุมการทำงานของ Circular Buffer ทั้ง 2 ตัว
- Temporary Register ได้แก่ TREG0, TREG1, TREG2 โดยที่ TREG0 จะใช้เป็นตัวตั้ง ในการคูณ ส่วน TEG1 อาจใช้เป็นตัวคูณหรือตัวกำหนดการ shift ก็ได้ ส่วน TREG2 ใช้เป็นตัวบอกตำแหน่งของบิตที่ต้องการใช้เปรียบเทียบในคำสั่ง Bitt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 2.11 Data-Addressing Modes

ในการสร้างคำสั่งที่มีประสิทธิภาพของ TMS320C50 แต่ละคำสั่งที่การอ้างอิงแอดเดรส Operand ได้หลายโหมด ซึ่งจะทำให้หลายวิธีดังนี้

### 1) Direct Addressing Mode

เป็นการนำค่าที่อยู่ในตำแหน่งที่อยู่ของหน่วยความจำมากระทำ เช่น คำสั่ง ADD 01h เป็นการนำค่าคงที่ที่อยู่ในตำแหน่ง 01h มาบวกกับค่าที่อยู่ใน ACC และนำค่าที่บวกกันได้มาเก็บไว้ใน ACC

### 2) Indirect Addressing Mode

เป็นการกำหนดตำแหน่งที่อยู่ของหน่วยความจำของ TMS320C50 โดยทางอ้อม วิธีการระบุตำแหน่งของหน่วยความจำที่ต้องการติดต่อวิธีนี้จะใช้ รีจิสเตอร์เป็นตัวชี้ ได้แก่ออกซิลลารี รีจิสเตอร์ เช่น คำสั่ง ADD \* เป็นการนำค่าตำแหน่งของหน่วยความจำที่ออกซิลลารีรีจิสเตอร์ ปัจจุบันชี้อยู่ มาบวกกับค่าใน Accumulator และนำผลลัพธ์ที่ได้มาเก็บใน Accumulator

### 3) Immediate Addressing Mode เป็นการกระทำกับค่าคงที่ แบ่งได้ 2 แบบ คือ

#### 3.1) Short Immediate Addressing Mode

เป็นคำสั่งที่ใช้กับค่าคงที่โดยตรง คำสั่งนี้จะมีการกำหนดค่าคงที่ในส่วนโอเปอเรนด์ แต่ค่าคงที่จะมีค่าไม่เกิน 256 เช่น คำสั่ง ADD #0FFh เป็นการนำค่าตำแหน่ง 0FFh มาบวกกับค่าที่อยู่ใน Accumulator และนำผลลัพธ์ที่ได้มาเก็บใน Accumulator

#### 3.2) Long Immediate Addressing Mode

ลักษณะการทำงานคล้ายกับแบบ Short Immediate Addressing Mode แตกต่างกันเพียงแต่ค่าที่นำมากระทำมีค่าไม่เกิน 65536 เช่น คำสั่ง ADD #01234h

### 4) Dedicated-Register Addressing Mode

วิธีนี้จะนำมาใช้เมื่อต้องการเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำ เช่น BLDD #02345,012h คำสั่งนี้เป็นการย้ายข้อมูลระหว่างหน่วยความจำข้อมูลตำแหน่ง 02345h ไปไว้ที่ตำแหน่ง 012h โดยที่เครื่องหมาย # จะหมายถึงการอ้างตำแหน่งของหน่วยความจำไม่ใช่หมายถึงข้อมูล

### 5) Memory-Mapped Addressing Mode

เหมาะสำหรับการอ้างตำแหน่งแอดเดรสด้วยรีจิสเตอร์ โดยใช้รีจิสเตอร์ทั่วไปได้ เช่น คำสั่ง LAMM DBMR, #300h เป็นการย้ายข้อมูลจากตำแหน่ง 300h ไปไว้ที่ตำแหน่งของ DBMR ชี้อยู่

### 6) Circular Addressing Mode

เหมาะสำหรับการอ้างอิงถึงข้อมูลที่มีหลาย ๆ ค่า ที่มีอยู่แล้วในหน่วยความจำและต้องการนำกลับมาใช้บ่อย ๆ เช่น การสร้างคำสัญญาเก็บไว้ใน Look up table เมื่อมีการอ้างถึงตำแหน่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลในหน่วยความจำ จะใช้ Circular Buffer เป็นตัวชี้ ดังตัวอย่างคือ SPLK #200h, CBSR1 จะเป็นการกำหนดค่าให้แก่รีจิสเตอร์ CBSR1 เริ่มต้นชี้ตำแหน่งที่ 200h ในหน่วยความจำของ Circular Buffer

## 2.12 การอินเทอร์รัพต์ (Interrupt)

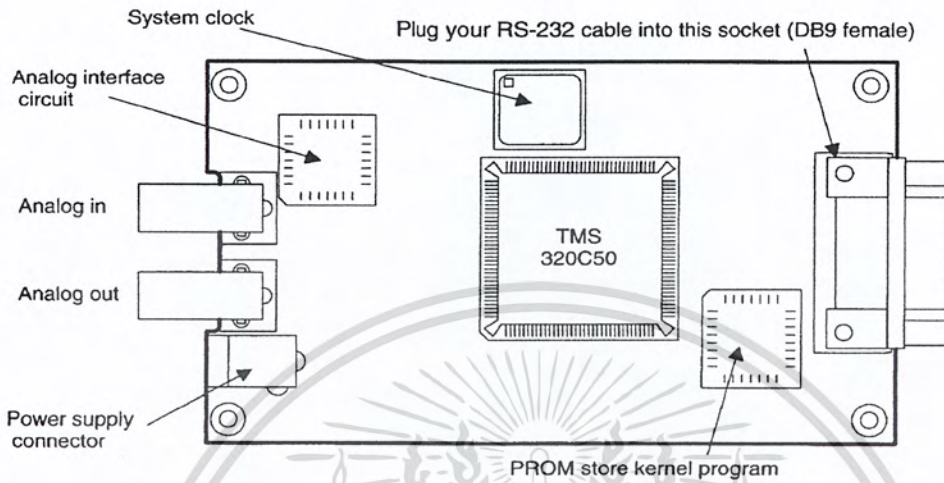
สำหรับ TMS320C50 จะรองรับการอินเทอร์รัพต์ จากผู้ใช้ทั้งหมด 16 อินเทอร์รัพต์ โดยมี 4 อินเทอร์รัพต์ที่เป็นการถูกเรียกใช้จากภายนอก และอีก 12 อินเทอร์รัพต์จะเป็นการอินเทอร์รัพต์จากภายใน โดยมีการจัดลำดับความสำคัญของอินเทอร์รัพต์ดังตารางที่ 2.4 ในการเรียกใช้อินเทอร์รัพต์ จะใช้คำสั่ง branch

Name †	Location		Priority	Function
	Dec	Hex		
RS	0	0	1 (highest)	reset signal
INT1	2	2	3	user interrupt #1
INT2	4	4	4	user interrupt #2
INT3	6	6	5	user interrupt #3
INT4	8	8	6	user interrupt #4
INT5	10	A	7	user interrupt #5
INT6	12	C	8	user interrupt #6
INT7	14	E	9	user interrupt #7
INT8	16	10	10	user interrupt #8
INT9	18	12	11	user interrupt #9
INT10	20	14	12	user interrupt #10
INT11	22	16	13	user interrupt #11
INT12	24	18	14	user interrupt #12
INT13	26	1A	15	user interrupt #13
INT14	28	1C	16	user interrupt #14
INT15	30	1E	17	user interrupt #15
INT16	32	20	18	user interrupt #16
TRAP	34	22	N/A	TRAP Instruction vector
NMI	36	24	2	nonmaskable interrupt

ตารางที่ 2.4 แสดงตำแหน่งและลำดับความสำคัญของอินเทอร์รัพต์ของ TMS320C50

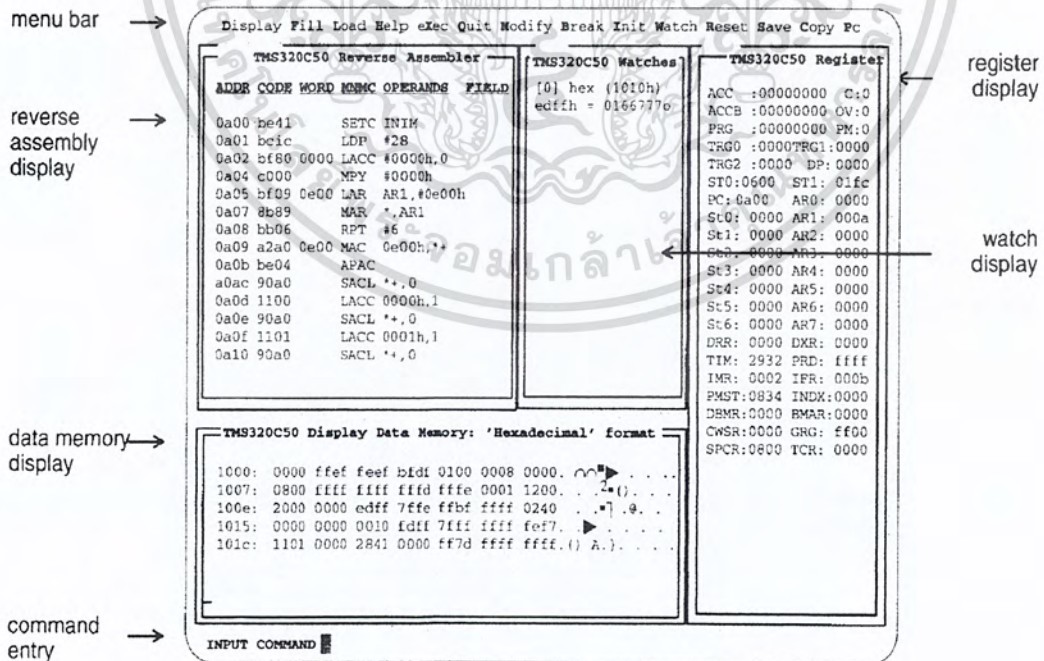
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.13 ลักษณะทางกายภาพของบอร์ดทดลอง DSK TMS320C50



รูปที่ 2.15 แสดงลักษณะของบอร์ดทดลอง DSK TMS320C50

## 2.14 ดีบั๊กเกอร์ของบอร์ดทดลอง DSK TMS320C50 (DSK Debugger)



รูปที่ 2.16 แสดงหน้าต่างของ DSK TMS320C50 Debugger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.15 การใช้งาน DSK TMS320C50 Debugger มีขั้นตอนดังนี้

ขั้นที่ 1 กรณีที่เป็นโปรแกรมภาษาแอสเซมบลี ทำการคอมไพล์ (assemble) ไฟล์ที่มีนามสกุล .asm ให้เป็นไฟล์นามสกุล .dsk โดยการเข้าไปที่ DOS แล้วใช้โปรแกรม DSK5A.EXE โดยพิมพ์ดังนี้ dsk5a \*.asm ถ้าไม่มีข้อผิดพลาดจะได้ไฟล์ที่มีนามสกุล .dsk

กรณีที่เป็นโปรแกรมภาษา C ทำการคอมไพล์ (compiler) ไฟล์ที่มีนามสกุล .c ให้เป็นไฟล์นามสกุล .OUT โดยการเข้าไปที่ DOS แล้วใช้โปรแกรม DSPC.BAT โดยพิมพ์ดังนี้ DSPC \*.OUT ถ้าไม่มีข้อผิดพลาดจะได้ไฟล์ที่มีนามสกุล .OUT

ขั้นที่ 2 ต่อไฟเลี้ยงเข้าบอร์ดทดลอง และต่อสาย RS-232 จากบอร์ดเข้าคอมพิวเตอร์ จากนั้นเรียกโปรแกรม Debugger (DSK5D.EXE) โดยพิมพ์ DSK5D ที่ DOS prompt ดังนี้

DSK5D (สำหรับเครื่องคอมพิวเตอร์ที่ใช้ พอร์ต COM1)

DSK5D C3 (สำหรับเครื่องคอมพิวเตอร์ที่ใช้ พอร์ต COM3)

ถ้าไม่มีข้อผิดพลาดจะแสดงหน้าต่างดังรูปที่ 2.16

ขั้นที่ 3 โหลดไฟล์นามสกุล .DSK (กรณีที่เป็นภาษาแอสเซมบลี) โดยการกดคำสั่ง LD (Load .dsk) หรือ .OUT (กรณีที่เป็นโปรแกรมภาษา c) โดยการกดคำสั่ง LC (Load .OUT) เข้าไปในดีบั๊กเกอร์ แล้วพิมพ์ชื่อไฟล์ จากนั้นกด ENTER 2 ครั้ง โปรแกรมจะถูกโหลดเข้าสู่ หน้าต่าง Debugger

ขั้นที่ 4 ทดลอง RUN โปรแกรมทีละคำสั่ง โดยการกดปุ่ม Space bar ที่คีย์บอร์ด หรือ RUN โปรแกรมทั้งหมด โดยการพิมพ์คำสั่ง XR (execute)

## 2.16 การโปรแกรมภาษาแอสเซมบลีบน TMS320C50

คำสั่งที่สำคัญกับการใช้งานภาษาแอสเซมบลีบน TMS320C50 สรุปได้ดังนี้

**2.16.1 Accumulator (ACC) และ Accumulator buffer (ACCB) Register** กับการบวกและลบ

Accumulator (ACC) Register มีขนาด 32 บิต ใช้ในการเก็บข้อมูลในการคำนวณต่าง ๆ สามารถแบ่งใช้งานได้เป็นบิตสูง คือ ACCH และบิตต่ำ คือ ACCL การบวกเลขจะใช้คำสั่ง ADD โดยจะบวกกับค่าที่กำหนดแล้วนำผลลัพธ์ไปเก็บใน ACC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Accumulator buffer (ACCB) Register มีขนาด 32 บิต ใช้ในการเก็บสำรองข้อมูลทีมาจาก ACC หากเมื่อต้องการนำ ACC ไปใช้งานอื่น การบวกเลขจะใช้คำสั่ง ADDB โดยจะบวกค่าใน ACCB กับค่าใน ACC แล้วนำผลลัพธ์ไปเก็บใน ACC ดังตัวอย่างดังนี้

- เป็นการบวก 1238h กับ 2228h ในฐานสิบหกแล้วเก็บผลลัพธ์ใน ACC

```
.mmregs
.ps      0a00      ; กำหนดค่า Address เริ่มต้น โปรแกรม
CLRC    C         ; เคลียร์ Carry Flag
CLRC    OV        ; เคลียร์ OverFlow Flag
step1 : LACC     #038h ; เก็บค่า 1238h ไว้ใน ACC
step2 : ADD      #028h ; บวกค่า 2228h กับ ค่าใน ACC เก็บผลลัพธ์ไว้ใน
                ; ACC
.end
```

- ทดลองการบวกค่าดังนี้  $1234h + 6789h + 3456h$

```
.mmregs
.ps      0a00      ; กำหนดค่า Address เริ่มต้น โปรแกรม
CLRC    C         ; เคลียร์ Carry Flag
CLRC    OV        ; เคลียร์ OverFlow Flag
step1 : LACC     #1234h ; เก็บค่า 1234h ไว้ใน ACC
step2 : ADD      #6789h ; บวกค่า 6789h กับ ค่าใน ACC เก็บผลลัพธ์ไว้ใน
                ; ACC
step3 : SACB                    ; พักค่าจาก ACC ไว้ใน ACCB
step4 : LACC     #3456h ; เก็บค่า 3456h ไว้ใน ACC
step5 : ADDB                    ; บวกค่า ใน ACCB กับ ค่าใน ACC เก็บผลลัพธ์ไว้ใน
                ; ACC
.end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทดลองการลบค่าดังนี้ 1234h - 4h

```
.mmregs
.ps      0a00      ; กำหนดค่า Address เริ่มต้นโปรแกรม
CLRC    C        ; เคลียร์ Carry Flag
CLRC    OV       ; เคลียร์ OverFlow Flag
step1 : LACC     #1234h ; เก็บค่า 1234h ไว้ใน ACC
step2 : SUB      #4h   ; ลบค่า 4h ออกจาก ค่าใน ACC เก็บผลลัพธ์ไว้ใน ACC
.end
```

### 2.16.2 Temporary และ Product Register ก็บการคูณ

Temporary Register เป็น รีจิสเตอร์ 16 บิต ได้แก่ TREG0, TREG1 และ TREG2 โดยที่ TREG0 จะใช้เป็นตัวเก็บตัวตั้ง ในการคูณ ส่วน TREG1 อาจใช้เป็นตัวคูณหรือเป็นตัวกำหนดการ Shift ก็ได้ ส่วน TREG2 ใช้เป็นตัวบอกตำแหน่งของบิตที่ต้องการใช้เปรียบเทียบในคำสั่ง BITT Product Register เป็น รีจิสเตอร์ 32 บิต ได้แก่ ใช้สำหรับเก็บผลของการคูณ โดยใช้คำสั่ง MPY ซึ่งจะเป็นการ คูณค่าตัวคูณกับตัวตั้งที่เก็บใน TREG ผลลัพธ์ที่ได้จะเก็บใน PREG แล้วนำผลลัพธ์นี้ไปเก็บใน ACC อีกที โดยใช้คำสั่ง LTP ดังตัวอย่าง

- การคูณ 5h กับ 1234h ในฐานสิบหกแล้วเก็บผลลัพธ์ใน ACC

```
.mmregs
.ps      0a00      ; กำหนดค่า Address เริ่มต้นโปรแกรม
step1 : LACC     #1234h ; เก็บค่า 1234h ไว้ใน Accumulator
step2 : SAMM    TREG0 ; เก็บค่า 1234h ไว้ใน TREG0
step3 : MPY     #5      ; คูณค่า 5h เข้ากับ TREG0 ผลลัพธ์เก็บใน
                        ; PREG
step4 : LTP     *       ; เก็บค่าจาก PREG ไว้ใน Accumulator
.end
```

- ทดลองการคูณค่าดังนี้  $(A \times C) + (3 \times F) + 6$

```
.mmregs
.ps      0a00      ; กำหนดค่า Address เริ่มต้นโปรแกรม
CLRC    C        ; เคลียร์ Carry Flag
step1 : LACC    #A      ; เก็บค่า Ah ไว้ใน Accumulator
step2 : SAMM   TREG0   ; เก็บค่า Ah ไว้ใน TREG0
step3 : MPY    #C      ; คูณค่า Ch เข้ากับ TREG0 ผลลัพธ์เก็บใน PREG
step4 : LTP    *      ; เก็บค่าจาก PREG ไว้ใน Accumulator
step5 : SACB                   ; นำค่าจาก ACC เก็บใน ACCB
step6 : LACC    #3      ; เก็บค่า 3h ไว้ใน Accumulator
step7 : SAMM   TREG0   ; เก็บค่า 3h ไว้ใน TREG0
step8 : MPY    #F      ; คูณค่า Fh เข้ากับ TREG0 ผลลัพธ์เก็บใน PREG
step9 : LTP    *      ; เก็บค่าจาก PREG ไว้ใน Accumulator
step10 : ADDB                   ; บวกค่าใน ACC กับ ค่าใน ACCB ผลลัพธ์เก็บใน ACC
step11 : ADD    #6      ; บวกค่าใน ACC กับ ผลลัพธ์เก็บใน ACC
.end
```

### 2.16.3 การทดลองโดยใช้ คำสั่ง MAC และ RPT

คำสั่ง MAC ใช้สำหรับคูณและบวก (Multiply and Accumulate) ส่วนคำสั่ง RPT ใช้สำหรับสั่งให้คำสั่งที่ตามหลังมันทำงานซ้ำหลายๆรอบ เมื่อนำคำสั่งทั้งสองมาใช้งานร่วมกันจะสามารถใช้คำนวณการกระทำต่างๆ ในการประมวลผลสัญญาณได้อย่างมีประสิทธิภาพ การทดลองข้างล่างเป็นการนำค่า 4 ค่าในตาราง (ซึ่งเริ่มต้นโดย Table) มาคูณกับ 2 (ซึ่งโดย Const) แล้วเก็บผลคูณที่ได้สะสมไว้ใน Accumulator ซึ่งขั้นตอนการคำนวณจะเป็น คือ  $ACC = (2 \times 1) + (2 \times 2) + (2 \times 3) + (2 \times 4)$

```
.mmregs
.ds      0f00h      ; กำหนดแอดเดรสข้อมูลเริ่มต้นตารางที่ 0f00h
TABLE : .word    1
        .word    2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.word 3
.word 4
CONST : .word 2

.ps 0a00h
step1 : LACC #0h
step2 : LDP #CONST
step3 : RPT #4 ; ตั้งให้ทวนคำสั่งต่อจากนี้ 4 ครั้ง
step4 : MAC TABLE, CONST ; เอาค่าใน TABLE คูณกับ CONST แล้วสะสม
        ผลลัพธ์ใน ACC การคูณแต่ละครั้งจะดึงเอาค่า
        ถัดไปของ TABLE ออกมา

.END

```

#### 2.16.4 Auxiliary Register (AR 0..7) กับการเก็บค่าการนับ (Loop Counter)

1. เป็นการให้ AR0 เป็นตัวนับเพื่อใช้ในการวนลูป และลดค่าจนเป็น 0 โดยตรวจสอบด้วยคำสั่ง BANZ แล้วจึงออกจากลูป

```

.mregs
.ps 0a00 ; กำหนดค่า Address เริ่มต้นโปรแกรม
CLRC C ; เคลียร์ Carry Flag
step1 : LACC #1h ; เก็บค่า 1h ไว้ใน ACC
step2 : LAR AR0, #3h ; เก็บค่า 3h ไว้ใน AR0 เป็นตัวนับ
step3 : ROR ; หมุนค่าใน ACC ไปทางขวาร่วมกับ Carry
step4 : BANZ step3,*-, AR0 ; กระโดดไปยังตำแหน่ง step3 ถ้า AR0 ไม่
        เท่ากับและลดค่าใน AR0 ลง 1
step5 : LAR AR0, #3h ; เก็บค่า 3h ไว้ใน AR0 เป็นตัวนับ
step6 : ROL ; หมุนค่าใน ACC ไปทางซ้ายร่วมกับ Carry
step7 : BANZ step6,*-, AR0 ; กระโดดไปยังตำแหน่ง step6 ถ้า AR0 ไม่
        เท่ากับ 0 และลดค่าใน AR0 ลง 1
step8 : B step2 ; กระโดดไปตำแหน่ง step2 เพื่อทำซ้ำทั้งหมด

.end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เป็นการคำนวณ แฟลคเตอร์เรียล ของ 6 โดยให้ AR2 เป็นตัวนับเพื่อใช้ในการวนลูป และลดค่าจนเป็น 0 โดยตรวจสอบด้วยคำสั่ง BANZ แล้วจึงออกจากลูป

```
.mmregs
.PS      0A00h      ; กำหนดค่า Address เริ่มต้นโปรแกรม
step1 : LACC      #01h      ; เก็บค่า 1h ไว้ใน ACC
step2 : LAR       AR2,#06h   ; เก็บค่า 6h ไว้ใน AR2 เป็นตัวนับ
step3 : SAR       AR2,AR5    ; เก็บค่าข้อมูลใน AR2 ไว้ใน AR5 เป็นคูณ
step4 : SAMM      TREG0     ; เก็บค่าใน ACC ไว้ใน TREG0 เพื่อเป็นตัวตั้ง
step5 : MPY       AR5        ; คูณค่าใน AR5 เข้ากับ TREG0 ผลลัพธ์เก็บใน
                          ; PREG
step6 : LTP       *          ; เก็บค่าจาก PREG ไว้ใน Accumulator
step7 : MAR       AR2
step8 : MAR       *         ; จัดตำแหน่ง ตัวนับและลดค่าลงหนึ่ง
step9 : BANZ      STEP3,*    ; ตรวจสอบตัวนับถ้าไม่เป็น 0 ให้ไปที่ STEP 3
.END
```

### 2.16.5 การใช้งาน Circular Buffer

มีอยู่สองตัวคือ Circular Buffer #1 และ Circular Buffer #2 การเรียกใช้งานตัวใดหรือทั้งสองตัวสามารถทำได้โดย การกำหนดค่าในรีจิสเตอร์ควบคุม คือ CBCR (Circular Buffer Control Register) ดังจะได้กล่าวต่อไป ข้อมูลที่อยู่ใน Circular Buffer จะถูกดึงออกมาใช้งานในลักษณะเป็นลำดับ เช่น ถูกเรียกออกมาประมวลที่แอดเดรสเรียงกันไป หรือถอยหลังที่แอดเดรส โดยมีตัวชี้แอดเดรส (Address Pointer) เป็นตัวบอกตำแหน่ง ลักษณะพิเศษก็คือ เมื่อข้อมูลตำแหน่งสุดท้ายถูกเรียกออกมาแล้ว จะทำให้ตัวชี้ชี้มายังตำแหน่งเริ่มต้นอีกครั้ง เหมือนกับว่าข้อมูลแรกกับสุดท้ายเชื่อมต่อกันอยู่ และสามารถอ่านข้อมูลไปได้อย่างไม่มีสิ้นสุด

Circular Buffer สามารถนำไปประยุกต์ใช้งานที่มีลักษณะเป็นรายคาบ เช่น การสร้างสัญญาณ หรือการสังเคราะห์สัญญาณ (รายคาบ) โดยเก็บรูปร่างของสัญญาณ 1 คาบไว้ใน Circular Buffer จากนั้นก็อ่านออกมาส่งให้ D/A ต่อไป และอีกตัวอย่างหนึ่ง คือ การใช้เก็บค่าสัมประสิทธิ์ของผลตอบสนองต่ออิมพัลส์ของตัวกรอง FIR เพื่อใช้อ่านออกมาทำคอนโวลูชันกับสัญญาณขาเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวที่บอกตำแหน่งแอดเดรสของข้อมูล จะใช้ Auxiliary Register ( $AR_0 - AR_7$ ) เช่น ที่แบบเพิ่มขึ้น หรือลดลงก็ได้

### รีจิสเตอร์ควบคุมการทำงานของ Circular Buffer (Circular Buffer Control Register)

ประกอบด้วยรีจิสเตอร์ 5 ตัว คือ

**1. Circular Buffer Start Address 1 : CBSR1**

เป็นรีจิสเตอร์ 16 บิต ใช้เก็บตำแหน่งแอดเดรสเริ่มต้นของ Circular Buffer #1

**2. Circular Buffer End Address 1 : CBER1**

เป็นรีจิสเตอร์ 16 บิต ใช้เก็บตำแหน่งแอดเดรสสุดท้ายของ Circular Buffer #1

**3. Circular Buffer Start Address 2 : CBSR2**

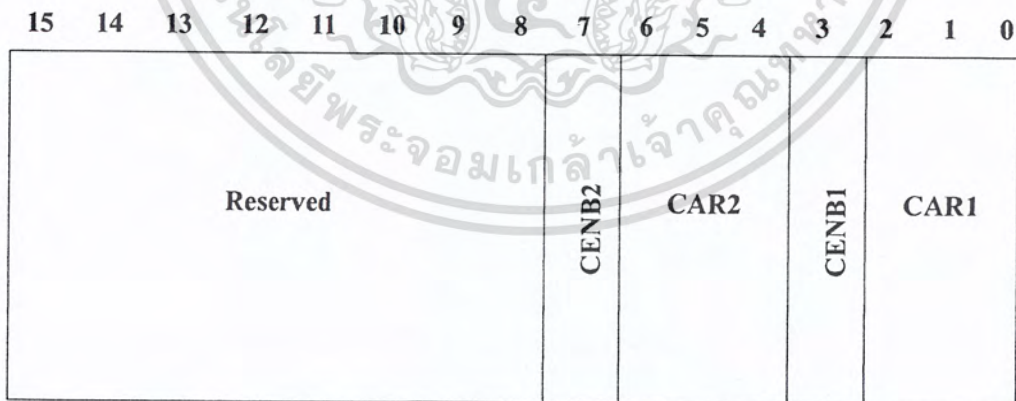
เป็นรีจิสเตอร์ 16 บิต ใช้เก็บตำแหน่งแอดเดรสเริ่มต้นของ Circular Buffer #2

**4. Circular Buffer End Address 2 : CBER2**

เป็นรีจิสเตอร์ 16 บิต ใช้เก็บตำแหน่งแอดเดรสสุดท้ายของ Circular Buffer #2

**5. Circular Buffer Control Register : CBCR**

เป็นรีจิสเตอร์ 8 บิต ใช้ Enable/Disable การทำงานของ Circular Buffer#1 และ Circular Buffer#2 ดังรูป



เมื่อ

CENB1 : Circular Buffer#1 ; Enable = 1 / Disable = 0

CENB2 : Circular Buffer#2 ; Enable = 1 / Disable = 0

CAR1 : Auxiliary Register ที่ใช้ตำแหน่งใน Circular Buffer#1

CAR2 : Auxiliary Register ที่ใช้ตำแหน่งใน Circular Buffer#2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเรียกใช้งาน Circular Buffer#1

CENB1 : Circular Buffer#1 ; Enable = 1

CENB2 : Circular Buffer#2 ; Disable = 0

CAR1 : ใช้ AR6 ซึ่งตำแหน่งใน Circular Buffer#1

CAR2 : ใช้ AR0 ซึ่งตำแหน่งใน Circular Buffer#2

7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	0

เพราะฉะนั้นต้องกำหนดค่าให้ CBCR = 0Eh

ตัวอย่าง การเรียกค่าจาก Look UP Table โดยการใช้งาน Circular Buffer

```
.mmregs
```

```
.ds 0300h
```

```
.include "s1.asm" ; เรียกไฟล์ s1.asm ซึ่งเก็บข้อมูลเป็น Table ดังแสดงท้ายการทดลองนี้
```

```
.ps 0a00h
```

```
;*****
```

```
;* ส่วนนี้ทำการ Initialize ค่าให้ circular buffer *
```

```
;*****
```

```
step1 : LACC #0300h
```

```
step2 : SAMM CBSR1 ; กำหนดให้ ค่าเริ่มต้น circular buffer ที่ตำแหน่ง 300h
```

```
step3 : SAMM AR7 ; ให้ AR7 เป็นตัวชี้แอดเดรสโดยเริ่มที่ 300h
```

```
step4 : ADD #4 ; กำหนดจำนวนแอดเดรสใน circular buffer จาก 300-304
```

```
step5 : SAMM CBER1 ; กำหนดให้ ค่าสุดท้าย circular buffer ที่ตำแหน่ง 304h
```

```
step6 : SPLK #0Fh,CBCR ; เงื่อนไขให้ circular buffer หมายเลข 1 ทำงาน และ AR7 เป็นตัวชี้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;*      ส่วนนี้ทดสอบการทำงานของ circular buffer      *
;*****

step7:  CALL  STEP9
step8:  B      STEP7:
step9:  MAR   *,AR7  ; ให้ AR7 ซีแอดเดรสใน circular buffer
step10: LACC  *+     ; นำข้อมูลใน circular buffer เก็บใน ACC และให้
                    AR7 ซีแอดเดรส ถัดไป
step11: SAMM  DXR   ; ส่งข้อมูลออก serial port
        RETE
        .END
;* ***** ส่วนนี้ เป็นไฟล์ SI.ASM *****
SINE .word 100h
      .word 200h
      .word 300h
      .word 400h
      .word 500h

```

## 2.17 สรุปรูปแบบการเขียนโปรแกรมภาษาแอสเซมบลีบน TMS320C50

```

.mmregs
.ds    0f00h
.include "sinetb.asm"
.ps    0080ah
rint:  B      RECEIVE
xint:  B      TRANSMIT
.ps    0a00h
.entry

WAIT:                                     ; a main program

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B WAIT ; รอจนกว่าจะมีการอินเตอร์รัพท์ A/D  
 RECEIVE: ; ส่วนของอินเตอร์รัพท์ serial port  
 LAMM DRR ; รับค่าอินพุท  
 .  
 . ; ใส่โปรแกรมที่ต้องการประมวลผลแบบเวลาจริงในส่วนนี้  
 SFL ; เลื่อนบิตไปทางซ้าย 2 บิตเพื่อให้ได้ 14 บิตก่อนส่งออกให้D/A  
 SFL  
 SAMM DXR ; ส่งค่าออกเอาท์พุท  
 RETE  
 TRANSMIT:  
 RETE  
 .END

## 2.18 การโปรแกรมภาษา C บน TMS320C50

การเขียนภาษา C บนบอร์ด TMS320C50 คล้ายกับการเขียนภาษา C โดยทั่วไป แต่จะมีรูปแบบแตกต่างออกไปในการกำหนดลักษณะทางฮาร์ดแวร์ให้กับบอร์ดโดยการใช้เฮดเดอร์ไฟล์ (header file) ที่เขียนขึ้นเฉพาะ (Run time support) ซึ่งจะใช้เฮดเดอร์ไฟล์ของภาษา C ทั่วไปไม่ได้ เช่น ioport.h จะกำหนดตำแหน่งของพอร์ต (port address)

ซึ่งเฮดเดอร์ไฟล์ที่สำคัญของ TMS320C50 มีดังนี้

assert.h limits.h stddef.h ctype.h  
 math.h stdlib.h errno.h setjmp.h  
 string.h float.h stdarg.h time.h  
 ioport.h evmaic5x.h

ซึ่งในการประกาศส่วนของเฮดเดอร์ไฟล์ นี้จะต้องแจ้งให้กับคอมไพเลอร์ทราบว่าเฮดเดอร์ไฟล์ นี้เก็บไว้ที่ไดเรกทอรีใด (Directory part) โดยการประกาศคำสั่งดังนี้ที่ คอสด

```
set c_dir = drive:\directory name1\ directory name2\...
```

รูปแบบ ของภาษา C บน TMS320C50 มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include " ioport.h"

#include " stdlib.h"

extern void aicinit();      /* ประกาศฟังก์ชันภายนอก */
void funtion();           /* ประกาศฟังก์ชันภายใน */
int *DRR = (int *) 0x20;   /* ประกาศแอดเดรสพอร์ท */
int *DXR = (int *) 0x21;

signed long ACC;

int i,x,y,xold1,xold2;

/*****
*
*      main function
*
*****/
main()
{
    xold1=xold2=0;
    function();
    aicinit();      /* initialize A/D, D/A */
    for(;;);       /* รอการอินเตอร์รัพท์จากซีเรียลพอร์ท */
}

/*****
*****/

*      A/D (Serial Port) Receive Interupt Service Routine
*

*****/

void c_int5(void)
{
    x = (*DRR >> 2); /* รับค่าอินพุทจาก A/D เก็บในตัวแปรและเลื่อนไปทางขวา 2 บิต */
    y = x + 2000;    /* ประมวลผล */
    *DXR = y << 2;  /* เลื่อนไปทางซ้าย 2 บิต แล้วส่งค่าออกเอาต์พุททาง D/A */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void funtion()
{
    .
    .
    .
}

```

การเชื่อม (Linking) ไฟล์รวมกับไฟล์อื่น สามารถทำได้โดย การเรียกใช้คำสั่ง ดังนี้

DSPLNK -V50 file

ไฟล์ที่สำคัญในการเชื่อมโยงมีดังนี้

1. ไฟล์นามสกุล .C
2. ไฟล์นามสกุล .CMD เป็นไฟล์ที่ใช้กำหนดคำสั่งและขนาดของหน่วยความจำที่ใช้ในโปรแกรม (Command file) เช่น DSK.CMD มีรูปแบบดังนี้

```

-stack 64
-l rts50.lib /* Runtime support library file*/
MEMORY
{
    PAGE 0: PROG: origin=0x1c00, length = 0x1000
    PAGE 0: RVEC: origin=0x080a, length = 0x0010
    PAGE 1: DATA: origin=0x0c00, length = 0x1000
}
SECTIONS
{
    vectors: >RVEC PAGE 0
    .text > PROG PAGE 0
    .cinit > PROG PAGE 0
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
.data > DATA PAGE 1
.bss > DATA PAGE 1
.stack > DATA PAGE 1
}
```

3. ไฟล์ที่อ้างตำแหน่งของอินเทอร์รัพท์ (Interrupt vector file) เช่น `vecs.asm` ตัวอย่างของไฟล์มีดังนี้

```
.title "vecs.asm"
.def IVECS
.ref _c_int5 ; อ้างถึงอินเทอร์รัพท์ที่ต้องใช้ในโปรแกรม
.sect "vectors"
IVECS b _c_int5 ; กำหนดอินเทอร์รัพท์เวกเตอร์
```

การเรียกฟังก์ชันที่เป็นภาษาแอสเซมบลี และการเขียนคำสั่งภาษาแอสเซมบลีร่วมกับภาษา C สามารถทำได้ดังนี้

การเรียกฟังก์ชันที่เป็นภาษาแอสเซมบลีจะใช้รูปแบบคือ `extern void funtion( );` ส่วนการเขียนคำสั่งภาษาแอสเซมบลีร่วมกับภาษา C จะใช้รูปแบบคือ `asm( " LAMM DRR ");` โดยห้ามเขียนคำสั่งติดกับเครื่องหมาย

## บทที่ 3

### การออกแบบ

โครงการนี้เป็นชุดทดลอง ตัวกรองเชิงเลขแบบ FIR ในการออกแบบจะแบ่งออกเป็น 2 ส่วน คือ ในส่วนแรกวงจรการอินเทอร์เฟซกับจอแอลซีดีและคีย์บอร์ด ส่วนที่สองการเขียนโปรแกรมด้วย MATLAB เพื่อหาค่าสัมประสิทธิ์อิมพัลส์ของ ตัวกรองความถี่ต่ำผ่าน (LPF), ตัวกรองความถี่สูงผ่าน (HPF), ตัวกรองความถี่แถบผ่าน (BPF), และ ตัวกรองความถี่แถบหยุด (BSP) ด้วยการออกแบบด้วยวิธีหน้าต่างแบบต่างๆ นำมาเปรียบเทียบผลตอบสนองของแต่ละหน้าต่าง ก่อนแล้วนำมาสร้างเป็นตัวกรองจริงบนบอร์ดประมวลผลสัญญาณ และสุดท้ายเป็นการออกแบบโปรแกรมเพื่อประมวลผลบน บอร์ด TMS320C50

#### 3.1 การใช้งานจอแอลซีดี (LCD)

ในระบบที่ใช้งานไมโครคอนโทรลเลอร์ทำงานแบบระบบโคเดเดี่ยว (Stand alone microcontroller applications) บ่อยครั้งที่จำเป็นต้องติดตั้งอุปกรณ์แสดงผลด้วย เพื่อแสดงข้อความ ตัวเลข ผลการวัดค่า หรือข้อมูลต่างๆ ในการสื่อสารกับผู้ใช้งานได้ และอุปกรณ์หนึ่งที่ถูกนำมาใช้งานอย่างกว้างขวางก็คือจอแสดงผลแอลซีดี ตัวอย่างที่เห็นกันได้บ่อยก็คือนำเอาแอลซีดีมาใช้ในการแสดงผลแบบเมนู เพื่อให้ผู้ใช้งานกดคีย์บอร์ดในการเลือกรูปแบบการทำงาน

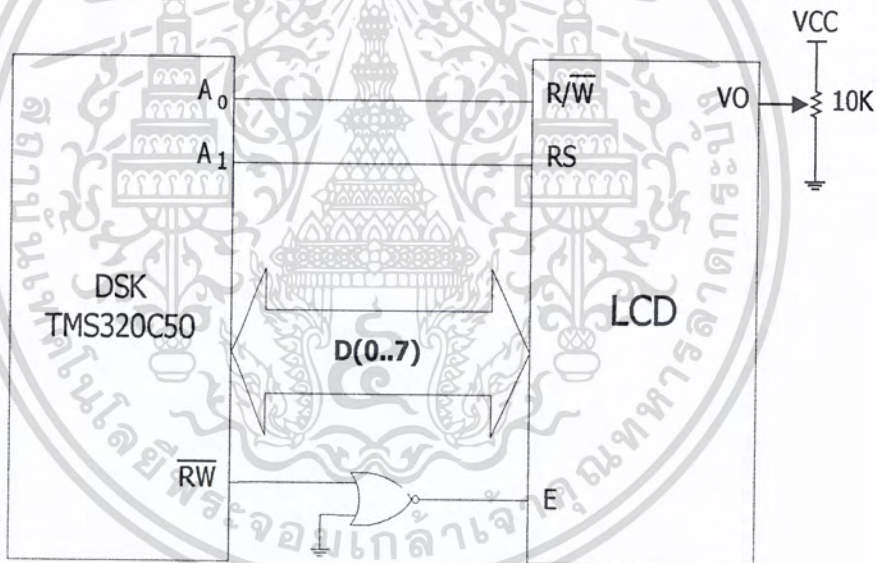
##### 3.1.1 การเชื่อมต่อและควบคุม

การเชื่อมต่อทางไฟฟ้าระหว่างตัวประมวลผล TMS320C50 และจอแสดงผลแอลซีดีใช้สายแพขนาด 14 เส้นต่อ ไปยังจอแสดงผลแอลซีดีที่กำหนดให้ใช้รุ่นใดก็ได้ที่ใช้อิซีคอนโทรลเลอร์อ้างอิงเป็นเบอร์ HD44780 หรือเบอร์อื่นที่เทียบเท่า ตัวอย่างเช่นรุ่น H2570 (ฮิตาชิ) LM016 (ชาร์ป) เป็นต้น ขาคู่ของแอลซีดีของแต่ละยี่ห้อไม่มีมาตรฐานที่แน่นอนเหมือนกัน การเชื่อมต่อขาต่างๆ ที่เชื่อมต่อไปยังแอลซีดีบนบอร์ดแสดงได้ดังรูปที่ 3.1 อาศัยรายละเอียดจากรูปนี้เพื่อเชื่อมต่อไปยังแอลซีดีและอ้างอิงในการตรวจสอบวงจรได้

การควบคุมการทำงานของแอลซีดีทำได้โดยอาศัยสัญญาณอินาเบิลซึ่งจะถูกส่งไปยังแอลซีดี หลังจากที่มีการกำหนดรูปแบบการทำงานว่าให้อยู่ในโหมดอ่านหรือเขียนไปยังตำแหน่งแอดเดรสที่เรากำหนด จากรูปที่ 3.1 จะเห็นได้ว่าขาสัญญาณเลือกโหมดการอ่านหรือการเขียน ( $R / \bar{W}$ ) ต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรโดยตรงกับขา  $A_0$  ของ DSP บอร์ด เมื่อระบุโหมดการอ่านหรือเขียนแล้วต้องกำหนดการทำงานของแอลซีดีด้วยว่าในโหมดการอ่านหรือการเขียนนั้นอยู่ในรูปของตัวอักษรหรือข้อมูล คำสั่งการกำหนดรูปแบบของข้อมูลทำได้ที่ขาสัญญาณ RS (Register Select) ของแอลซีดีที่เชื่อมต่อกับโดยตรงกับ  $A_1$  ของ TMS320C50 ถ้าหากค่าของ  $RS = A_1 = 1$  จะเป็นโหมดการเขียนข้อมูลอักขระ และถ้าค่าของ  $RS = A_1 = 0$  จะเป็นโหมดข้อมูลการอ่านคำสั่งควบคุมภายใน การส่งผ่านข้อมูลทั้งตัวอักขระและคำสั่งควบคุมนั้นจะถูกป้อนเข้ากับบัสข้อมูลแบบ 2 ทิศทางของระบบ จริงๆแล้วการใช้งานของจอแสดงผลแอลซีดีสามารถทำงานได้ทั้งในโหมด 4 บิต(D0-D3) หรือโหมด 8 บิต(D0-D7)ก็ได้ แต่เนื่องจากในระบบที่เราศึกษาอยู่เป็นระบบที่ทำงานแบบ 8 บิต ดังนั้นการเชื่อมต่อแอลซีดีจึงต่อใช้งานแบบ 8 บิตกับบัสข้อมูลได้พอดี และสามารถเขียนโปรแกรมให้ทำงานในโหมดส่งผ่านข้อมูล 8 บิตกับจอแอลซีดีได้ทันที



รูปที่ 3.1 เชื่อมต่อแอลซีดีไปใช้งาน

จากรูปที่ 3.1 จะมีการต่อตัวต้านทานปรับค่าได้เข้ากับจอแอลซีดีก็เพื่อที่จะปรับแรงดัน  $V_{cc}$  ที่ป้อนเป็นไฟเลี้ยงให้กับตัวแสดงผลแอลซีดี แรงดันนี้มีผลต่อการปรับความเข้ม (contrast) ในการแสดงผล เพื่อให้ผู้ใช้สามารถปรับความเข้ม ได้เหมาะสมกับสภาพแวดล้อมในขณะนั้น ทั้งความเข้มของแสงจากภายนอกและมุมมองของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 คำสั่งควบคุมการทำงานของแอลซีดี

ความสามารถหลักที่แอลซีดีนิยมถูกหยิบยกมาใช้งานก็เพราะมันสามารถแสดงผลตัวอักษร แอสกีได้ (ASCII character) ซึ่งมีอยู่ในคอนโทรลเลอร์ของแอลซีดีและในแอลซีดีแต่ละรุ่นก็มักใช้คอนโทรลเลอร์เบอร์เดียวกันเสมอ นั่นคือ คำสั่งควบคุมต่างๆ จึงใช้เหมือนกันในแอลซีดีแต่ละรุ่น ดังเช่นชุดคำสั่งที่จะกล่าวต่อไปซึ่งใช้งานได้กับแอลซีดีทั่วไป เช่นรุ่น H2570, LM16L, LM1612A เป็นต้น คำศัพท์เกี่ยวกับองค์ประกอบและการทำงานพื้นฐานของไมโครคอนโทรลเลอร์ในแอลซีดีก่อนการทำงานภายในแอลซีดีจะมีบัพเฟอร์อยู่ภายใน ซึ่งมีความจุประมาณ 80 ตัวอักษรที่มีชื่อเรียกว่า DD-RAM (Display Data RAM) มีตำแหน่งแอดเดรสอยู่ระหว่าง 00h ถึง 04Fh เช่นถ้าเป็นแอลซีดีขนาด 16 ตัวอักษร 1 บรรทัด DD-RAM จะเก็บข้อมูลตัวอักษรที่จะแสดงผลทั้ง 16 ตัวจะถูกเก็บอยู่ที่ตำแหน่งแอดเดรส 00h ถึง 0Fh โดยเริ่มต้นจากซ้ายของจอแสดงผล การทำให้เกิดช่องว่าง (window) ใน DD-RAM สามารถทำได้โดยการใช้คำสั่งชีพตัวอักษรในการแสดงผลหรืออีกวิธีที่ง่ายมากขึ้นก็คือในขณะที่มีข้อมูลตัวอักษรเก็บไว้ในแอดเดรสค่าหนึ่งการแสดงผลตัวอักษรต่อไปหรือทำให้เกิดช่องว่าง ทำได้โดยการกำหนดจุดเริ่มต้นของแอดเดรสใหม่แตกต่างไปจากเดิมไป 1 แอดเดรส DD-RAM ซึ่งก็ทำให้ได้ผลลัพธ์เช่นเดียวกันโดยทั่วไปการเขียนโปรแกรมเพื่อควบคุมการแสดงผลของแอลซีดีเมื่อมีการส่งข้อมูลอักษรไปยังแอลซีดี คือให้ทำการเลื่อนตัวอักษรเดิมไปและเลื่อนเคอร์เซอร์ตามไปด้วยเพื่อชี้ตำแหน่งของตัวอักษรต่อไป การเขียนโปรแกรมให้แสดงผลแบบนี้ทำให้ง่ายต่อการอ่านและเข้าใจของผู้ใช้งานได้มากที่สุด ในแอลซีดีมีหน่วยความจำส่วนหนึ่งที่สำคัญก็คือ CG-RAM ซึ่งทำหน้าที่เก็บข้อมูลรายละเอียดโครงสร้างของตัวอักษรที่สามารถนำมาแสดงผลได้ หน่วยความจำส่วนนี้ผู้ใช้งานสามารถที่จะสร้างตัวอักษรใดๆก็ได้ตามต้องการ แล้วเก็บลงในหน่วยความจำส่วนนี้สำหรับเรียกมาใช้งานได้ด้วยถ้าหากตัวอักษรที่มีอยู่แล้วนั้นไม่เพียงพอต่อการใช้งาน การใช้งานในส่วนของ CG-RAM ดูได้จากคำอธิบายของแอลซีดีรุ่นนั้นๆ การส่งคำสั่งควบคุมไปยังจอแอลซีดีสามารถกำหนดเป็นรหัสคำสั่งต่างๆ ส่งไปยังตำแหน่งแอดเดรสที่เรากำหนดโดยกำหนดให้ขาสัญญาณ RS มีค่าเป็น "0" และเซตโหมดให้เป็นการอ่านหรือการเขียนได้ด้วยขาสัญญาณ  $R/\overline{W}$  ต่อไปจะกล่าวถึงคำสั่งพื้นฐานที่ใช้งานบ่อยครั้ง ในการควบคุมการทำงานของแอลซีดี สำหรับเครื่องหมายดอกจัน (\*) แทนสถานะของบิตที่จะเป็นสถานะใดก็ได้ไม่สนใจและไม่มีความสัมพันธ์กับคำสั่งนั้นๆ

คำสั่งเคลียร์จอแสดงผล (Clear Display)

D7							D0
0	0	0	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคอนโทรลเลอร์ประมวลผล คำสั่งนี้จะทำให้ข้อมูลใน DD-RAM ถูกแทนที่ด้วยค่า 20h เท่ากับ space ในรหัสแอสกี ทำให้จอแสดงผลไม่ปรากฏตัวอักขระใดๆ บนจอภาพ เคอร์เซอร์จะถูกเซตอยู่ที่ตำแหน่งเริ่มต้นใหม่อีกครั้ง และยกเลิกผลจากการใช้คำสั่งเลื่อนข้อมูล (Display hift) ที่ผ่านมาแล้ว

คำสั่งเลื่อนเคอร์เซอร์ไปยังตำแหน่งเริ่มต้น (Return home)

D7							D0
0	0	0	0	0	0	1	*

คำสั่งนี้มีผลทำให้มีการรีเซตเคอร์เซอร์ให้กลับไปอยู่ที่ตำแหน่งเริ่มต้นใหม่และรีเซตคำสั่งเลื่อนข้อมูลที่ผ่านมาแล้ว โดยที่ข้อมูลตัวอักขระใน DD-RAM ไม่เกิดการเปลี่ยนแปลง นั่นคือตัวอักขระบนจอแสดงผลจะยังคงเหมือนเดิมไม่เปลี่ยนแปลง

คำสั่งกำหนดโหมดป้อนข้อมูล (Entry Mode Set)

D7							D0
0	0	0	0	0	1	I/D	S

คำสั่งถูกใช้ประโยชน์ในการกำหนดการทำงานของแอลซีดีหลังจากการส่งไบต์ไปยังจอแสดงผลบิต I/D (Increase/Decrease bit) ทำหน้าที่กำหนดการเพิ่ม (I/D = 1) หรือลด (I/D = 0) ค่าตำแหน่งแอดเดรสใน DD-RAM ลง 1 แอดเดรส โดยอัตโนมัติ เมื่อเกิดการอ่านหรือเขียนตัวอักขระค่าของตำแหน่งแอดเดรสนี้ถูกเก็บในแอดเดรสเคาท์เตอร์ (AC) บิต S (shift bit) เป็นบิตที่ใช้กำหนดลักษณะการเปลี่ยนข้อมูลอักขระอย่างอัตโนมัติ โดยถ้าบิต S = 1 เมื่อมีการส่งไบต์ข้อมูลให้เกิดขึ้นตัวเคอร์เซอร์จะอยู่กับที่ แต่อักขระข้อมูลเดิมจะถูกชิพออกไปทางซ้าย แต่ถ้าหากบิต S = 0 เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวา

คำสั่งควบคุมการแสดงผล (Display ON/OFF)

D7					D0		
0	0	0	0	1	D	C	B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งใช้สำหรับควบคุมการแสดงผลในการปิดหรือการเปิดการแสดงผลหน้าจอและเคอร์เซอร์ โดยปราศจากการเปลี่ยนแปลงข้อมูลใน DD-RAM ผู้เขียนโปรแกรมสามารถกำหนดหน้าจอแสดงผลให้ปิดหรือเปิดได้ด้วยบิต D โดยกำหนดให้บิต D = 1 เป็นการเปิดจอแสดงผลและถ้า D = 0 เป็นการปิดจอแสดงผล เช่นเดียวกันสำหรับบิต C = 1 และบิต C = 0 เป็นการควบคุมให้เคอร์เซอร์ปิดหรือเปิดตามลำดับ และบิต B ซึ่งเป็นบิตกำหนดว่าจะให้เคอร์เซอร์กะพริบหรือไม่

คำสั่งควบคุมการเลื่อนเคอร์เซอร์ ( Cursor or Display Shift )

D7				D0			
0	0	0	1	S/C	R/L	*	*

คำสั่งใช้สำหรับควบคุมการเลื่อนของเคอร์เซอร์และตัวอักขระที่แสดงผล ซึ่งมีความสำคัญและใช้งานบ่อยเนื่องจากผู้เขียนโปรแกรมต้องแสดงผลข้อความบนจอแสดงผลตามแนวนอน ผลที่เกิดจากการกำหนดสถานะของบิต S/C และ R/L แสดงในตารางที่ 3.1

S/C	R/L	การทำงาน
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย
0	1	เลื่อนเคอร์เซอร์ไปทางขวา
1	0	เลื่อนตัวอักขระไปทางซ้ายพร้อมทั้งเคอร์เซอร์ด้วย
1	1	เลื่อนตัวอักขระไปทางขวาพร้อมทั้งเคอร์เซอร์ด้วย

ตารางที่ 3.1 การกำหนดสถานะของบิต S/C และ R/L และการทำงาน

คำสั่งเซตฟังก์ชัน ( Function Set )

D7				D0			
0	0	1	DL	N	F	*	*

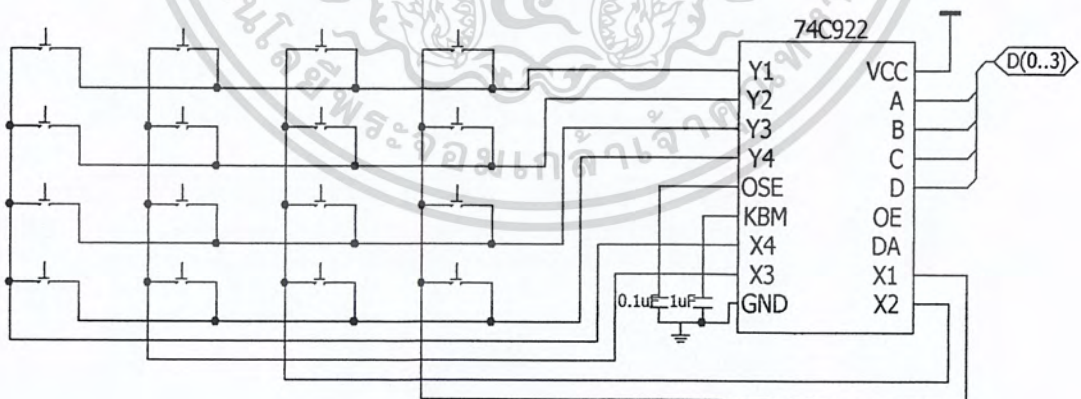
คำสั่งนี้ใช้สำหรับเซตโหมดการทำงานของแอลซีดีหลังจากการรีเซตการทำงานหรือเริ่มต้นทำงานระบบทุกครั้ง ในที่นี้เราจะใช้งานแอลซีดีในโหมด 8 บิตและใช้เพียง 1 บรรทัดเท่านั้น ความหมายของบิตต่างๆ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต LD = 1 หมายถึงการทำงานในโหมดอินเทอร์เฟซแบบ 8 บิต
- บิต LD = 0 หมายถึงการทำงานในโหมดอินเทอร์เฟซแบบ 4 บิต
- บิต N = 0 หมายถึงทำงานแบบ 1 บรรทัด
- บิต N = 1 หมายถึงทำงานแบบ 2 บรรทัด หรือมากกว่า
- บิต F = 0 หมายถึงทำงานในโหมดละเอียด 5 X 7 จุด
- บิต F = 1 หมายถึงทำงานในโหมดละเอียด 5 X 10 จุด

### 3.2 การออกแบบการเชื่อมต่อกับ Key Board

ใช้คีย์บอร์ดแบบเมตริกซ์ 4x4 คู่กับไอซีเบอร์ 74C922 ที่มีหน้าที่ในการเข้ารหัสของคีย์บอร์ด เมื่อมีการกดคีย์ ไอซีเบอร์ 74C922 จะเข้ารหัสการกดคีย์บอร์ดแต่ละคีย์เป็นเลขฐานสอง (0000 – 1111) ทางขา Y1 – Y4 และ X1 – X4 ของคีย์บอร์ด และข้อมูลจะออกทาง A, B, C และ D ซึ่งต่อกับขาข้อมูลของ TMS320C50 (D0 – D3) ขา OE (Output Enable) เป็นขาการกำหนดการทำงาน การแก้ไขบาวซ์ของคีย์บอร์ดจะต่อคาปาซิเตอร์ค่า 1 ไมโครฟารัด ที่ขา KBM (Key Bounce Mask) เพื่อทำการหน่วงเวลา 10 มิลลิวินาที เมื่อมีการกดคีย์และหลังการกดคีย์ เพื่อทำการตรวจสอบการกดคีย์ครั้งสุดท้าย ที่ขา OSC (Oscillator) จะต่อคาปาซิเตอร์ค่า 0.1 ไมโครฟารัด เพื่อกำหนดอัตราการสแกนตรวจสอบคีย์ที่มีความถี่ประมาณ 600Hz แสดงการต่อดังรูปที่ 3.2.



รูปที่ 3.2 การต่อคีย์บอร์ดร่วมกับ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การออกแบบตัวกรองแบบ FIR โดยวิธีหน้าต่างแบบต่างๆ

สำหรับตัวกรองแบบ FIR นั้น ในการออกแบบสิ่งที่เราต้องการ คือ ค่าของผลตอบสนองต่ออิมพัลส์ หรือ  $h(n)$  ของระบบ เราสามารถหาค่าผลตอบสนองต่ออิมพัลส์ของระบบต่างๆ ได้ โดยการใช้โปรแกรม MATLAB ดังต่อไปนี้

#### 3.3.1 หน้าต่างแบบสี่เหลี่ยม (Rectangular Window)

1) ตัวกรองความถี่ต่ำผ่านโดยผลตอบสนองแบบจำกัด มีความถี่คutoff 1 KHz และ ความถี่ในการสุ่ม 8 KHz

```

fc=1000;
fs=8000;
N=81;
wc=(fc*pi)/fs;
w=Boxcar(n);
for n=1:40
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wc/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(2,1,2);

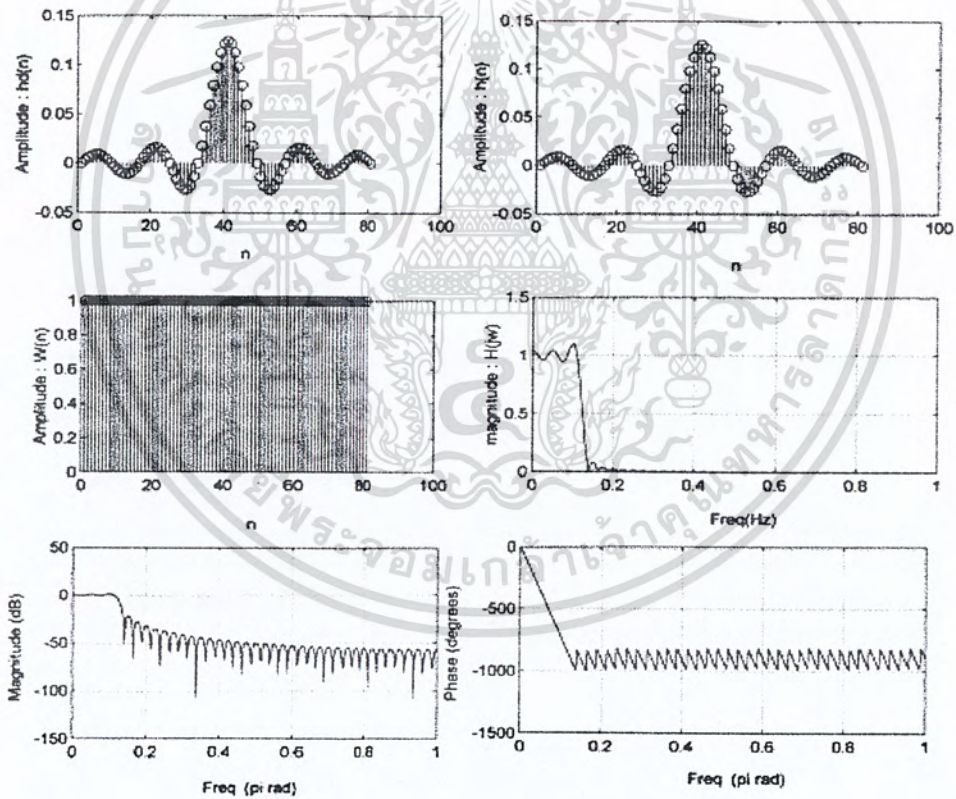
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(Hz)');
ylabel('magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

```



รูปที่ 3.3 ผลตอบสนองความถี่ต่ำของตัวกรองแบบจำกัดโดยวิธีหน้าต่างแบบสี่เหลี่ยม

ก.) ผลตอบสนอง  $hd(n)$  ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$

ค.) หน้าต่างสี่เหลี่ยม  $w(n)$  ง.) สเปกตรัม  $H(jw)$

จ.) สเปกตรัม (dB) ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ตัวกรองความถี่สูงผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ 3 KHz และ ความถี่ในการสุ่ม 8 KHz

```

fc=3000;
fs=8000;
N=81;
w=boxcar(N);
wc=(fc*pi)/fs;
for n=1:40
    hd(n)=-(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=1-(wc/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=-(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(211);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(212);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,20*log10(abs(X)))

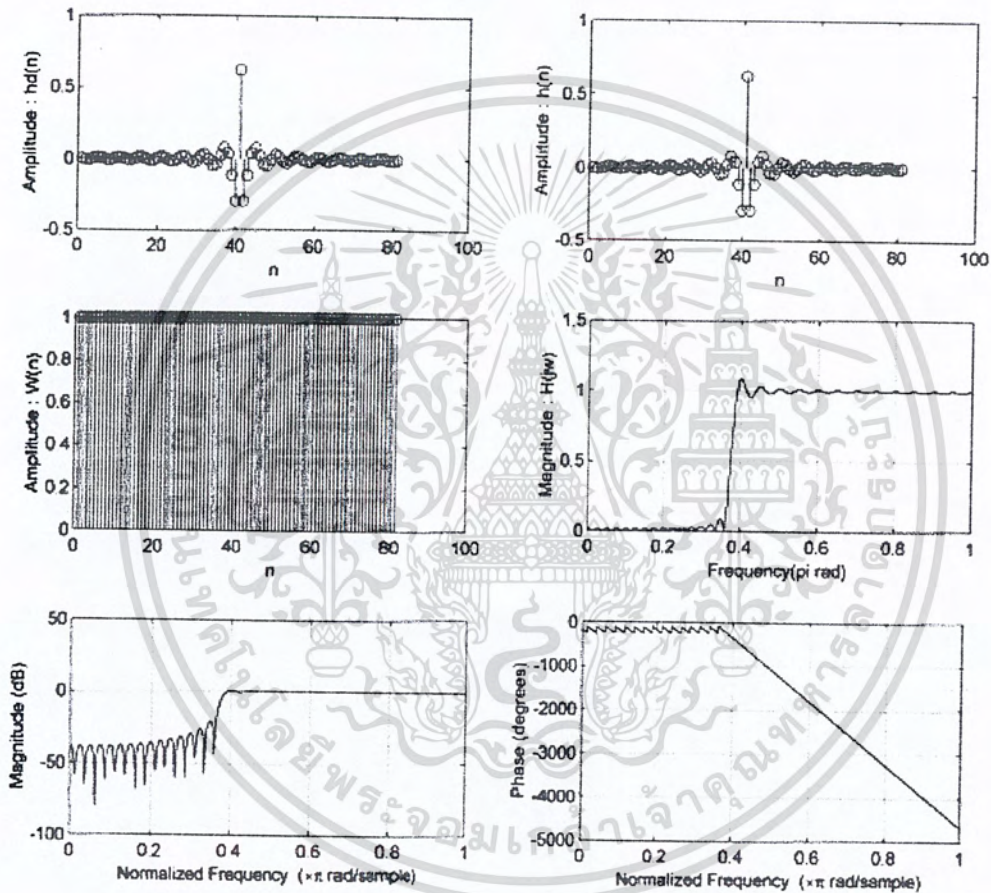
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

xlabel('Frequency(Hz)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

```



รูปที่ 3.4 ผลตอบสนองความถี่สูงผ่านของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบสี่เหลี่ยม

- ก.) ผลตอบสนอง  $hd(n)$     ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$     ง.) สเปกตรัม  $H(j\omega)$   
 จ.) สเปกตรัม (dB)    ฉ.) ผลตอบสนองทางเฟส

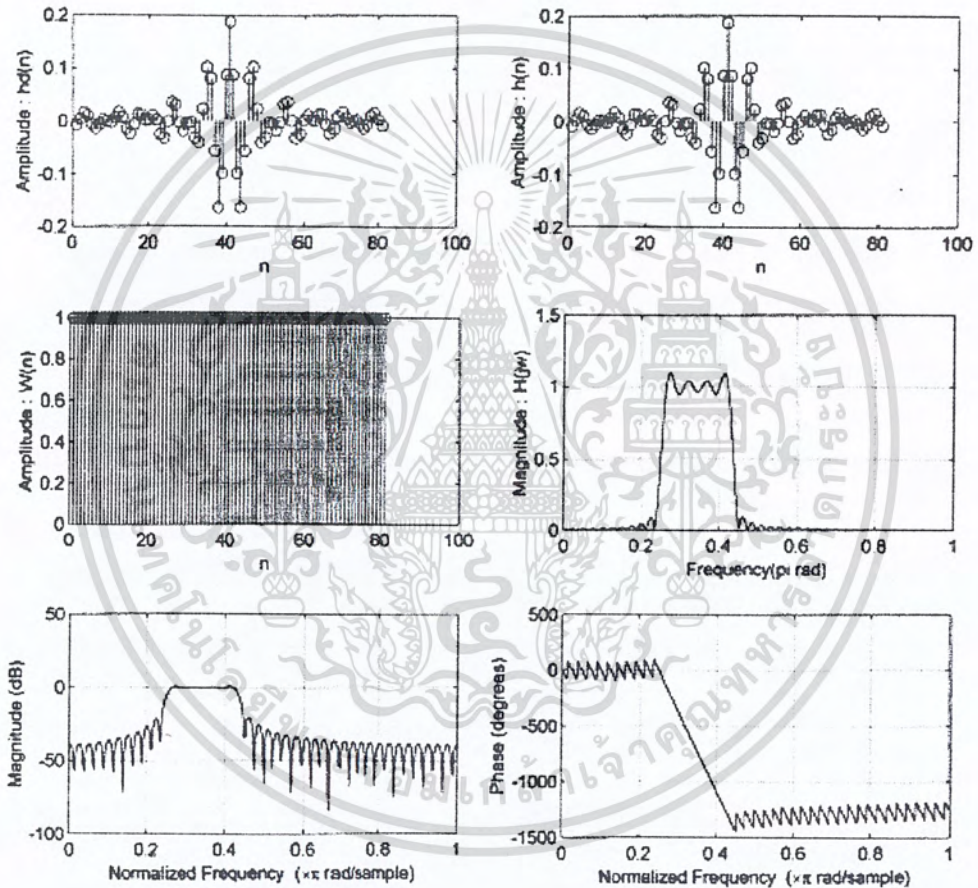
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตัวกรองความถี่แถบผ่านโดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ  $f_{cl} = 2 \text{ KHz}$  ,  $f_{ch} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$

```
fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=boxcar(N);
for n=1:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wch-wcl)/pi;
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd);
xlabel('n');ylabel('Amplitude : hd(n)');
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)
```



รูปที่ 3.5 ผลตอบสนองความถี่แถบผ่านของตัวกรองแบบจำกัดโดยวิธีหน้าต่างแบบสี่เหลี่ยม

- ก.) ผลตอบสนอง  $hd(n)$       ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$       ง.) สเปกตรัม  $H(jw)$   
 จ.) สเปกตรัม (dB)      ฉ.) ผลตอบสนองทางเฟส

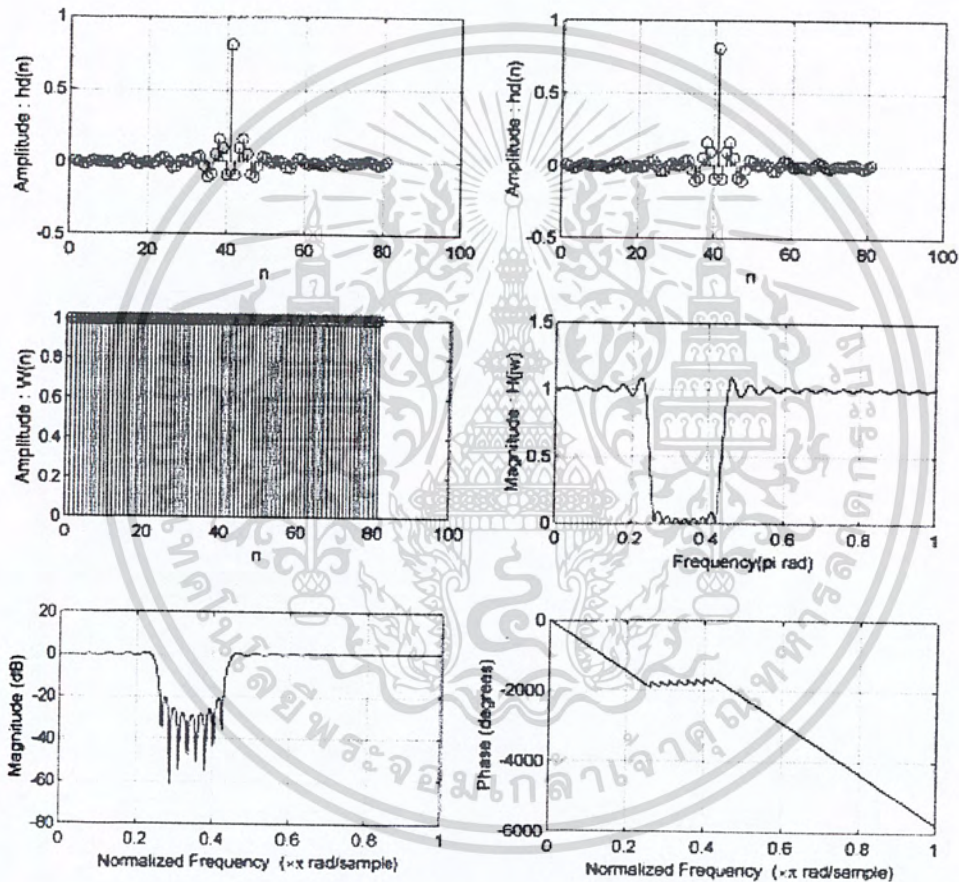
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตัวกรองความถี่แถบหยุดโดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ  $f_{cl} = 2 \text{ KHz}$  ,  
 $f_{ch} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$

```
fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=boxcar(N);
for n=1:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=1+((wcl-wch)/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)');grid on
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)
```



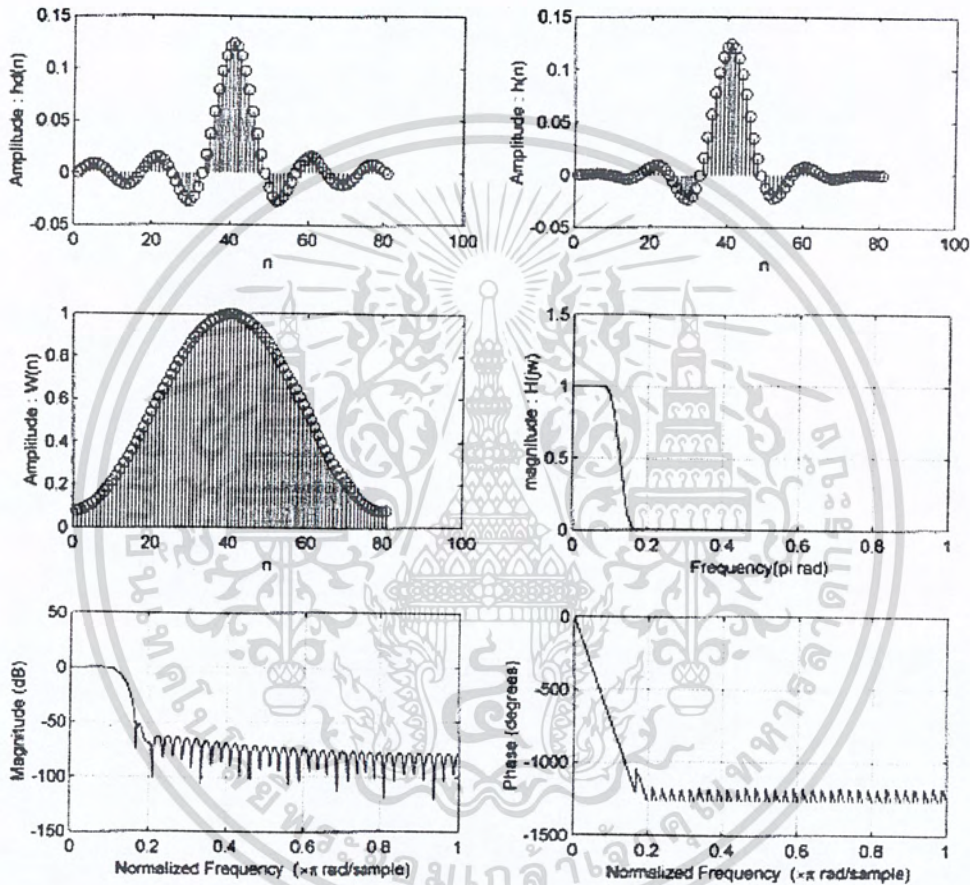
รูปที่ 3.6 ผลตอบสนองความถี่แถบหยุดของตัวกรองแบบจำกัดโดยวิธีหน้าต่างแบบสี่เหลี่ยม

- |                               |                                   |
|-------------------------------|-----------------------------------|
| ก.) ผลตอบสนอง $hd(n)$         | ข.) ผลตอบสนอง $h(n) = hd(n)*w(n)$ |
| ค.) หน้าต่างสี่เหลี่ยม $w(n)$ | ง.) สเปกตรัม $H(jw)$              |
| จ.) สเปกตรัม (dB)             | ฉ.) ผลตอบสนองทางเฟส               |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 หน้าต่างแบบแฮมมิง (Hamming Window)

1) ตัวกรองความถี่ต่ำผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คutoff 1KHz และความถี่ในการสุ่ม 8 KHz

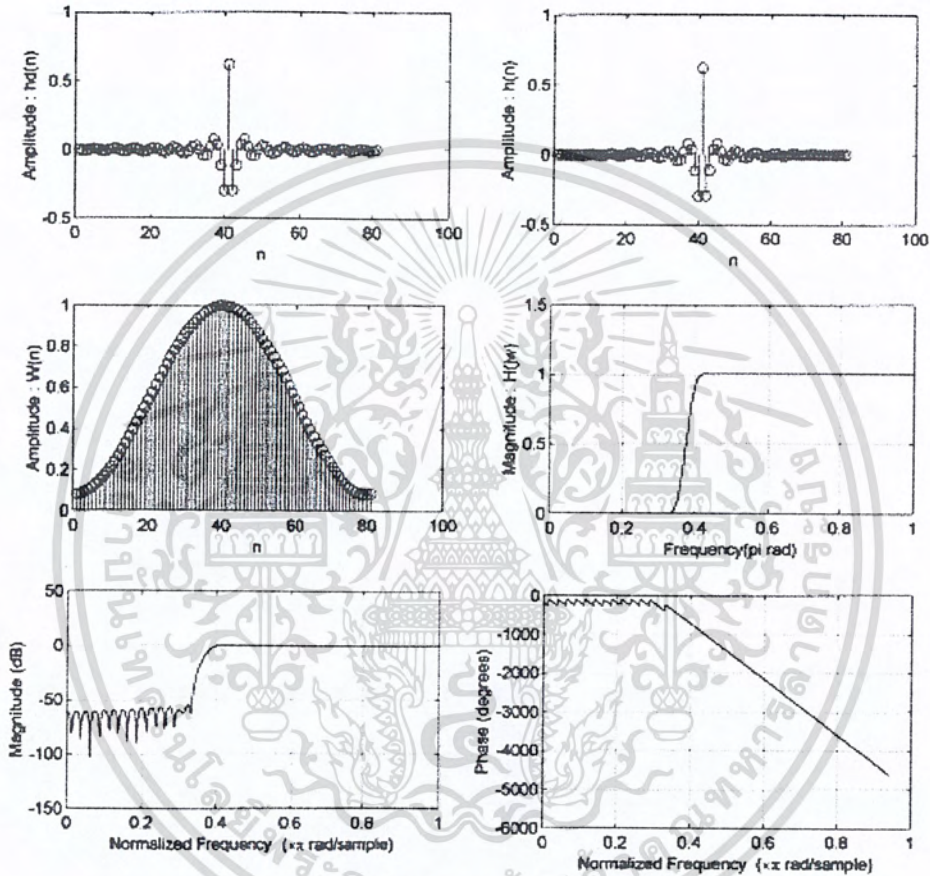


รูปที่ 3.7 ผลตอบสนองความถี่ต่ำผ่านของตัวกรองแบบจำกัดโดยวิธีหน้าต่างแบบแฮมมิง

- ก.) ผลตอบสนอง  $hd(n)$       ข.) ผลตอบสนอง  $h(n) = hd(n) * w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$       ง.) สเปคตรัม  $H(jw)$   
 จ.) สเปคตรัม (dB)      ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ตัวกรองความถี่สูงผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ 3 KHz และ ความถี่ในการสุ่ม 8 KHz

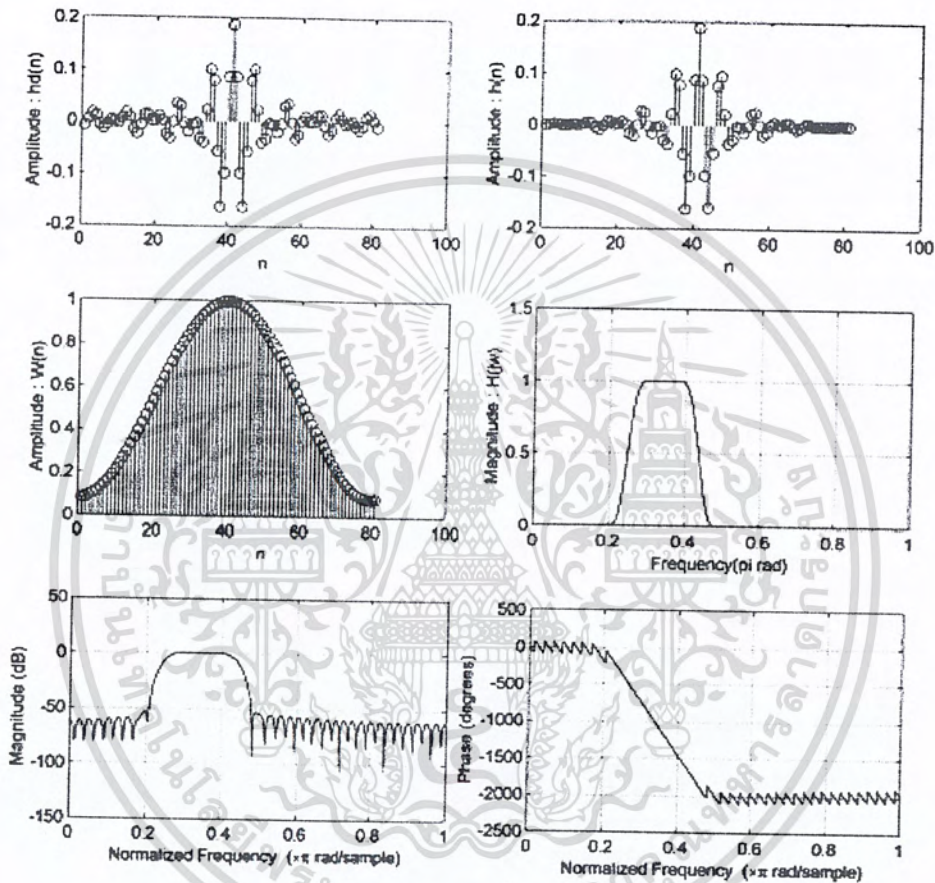


รูปที่ 3.8 ผลตอบสนองความถี่สูงผ่านของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแฮมมิง

- ก.) ผลตอบสนอง  $hd(n)$       ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$       ง.) สเปกตรัม  $H(j\omega)$   
 จ.) สเปกตรัม (dB)      ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตัวกรองความถี่แถบผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ  $f_{cl} = 2 \text{ KHz}$  ,  $f_{ch} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$

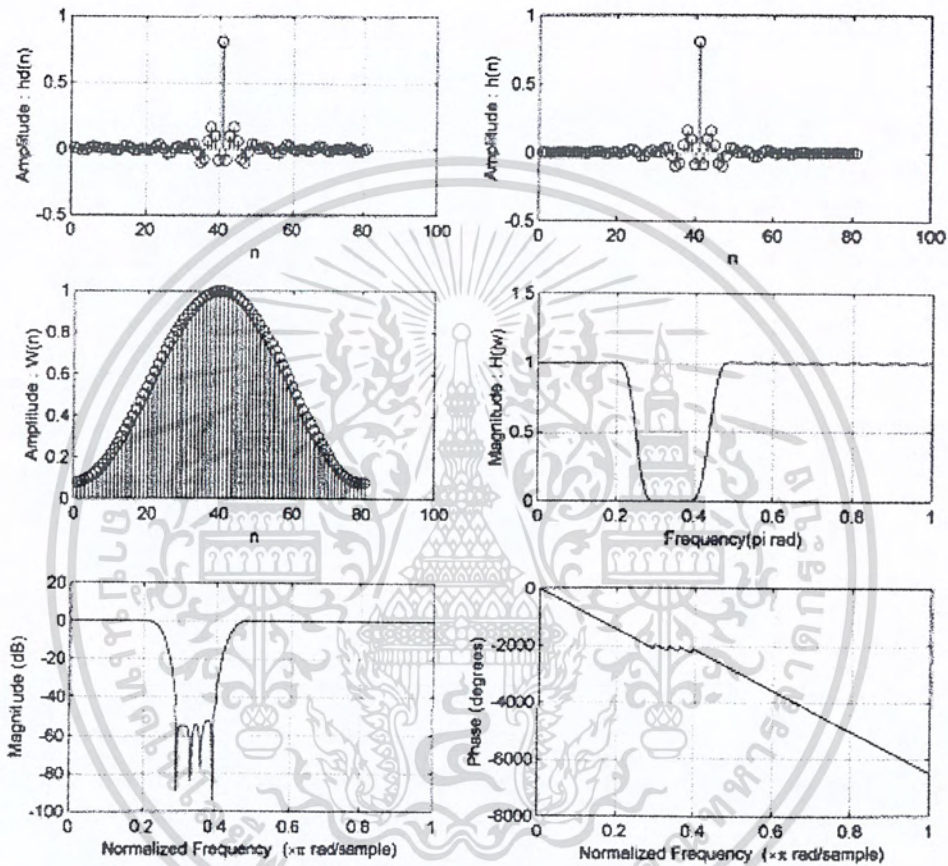


รูปที่ 3.9 ผลตอบสนองความถี่แถบผ่านของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแฮมมิง

- ก.) ผลตอบสนอง  $hd(n)$     ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$     ง.) สเปคตรัม  $H(jw)$   
 จ.) สเปคตรัม (dB)    ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตัวกรองความถี่แถบหยุดโดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ  $f_{cl} = 2 \text{ KHz}$ ,  $f_{ch} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$



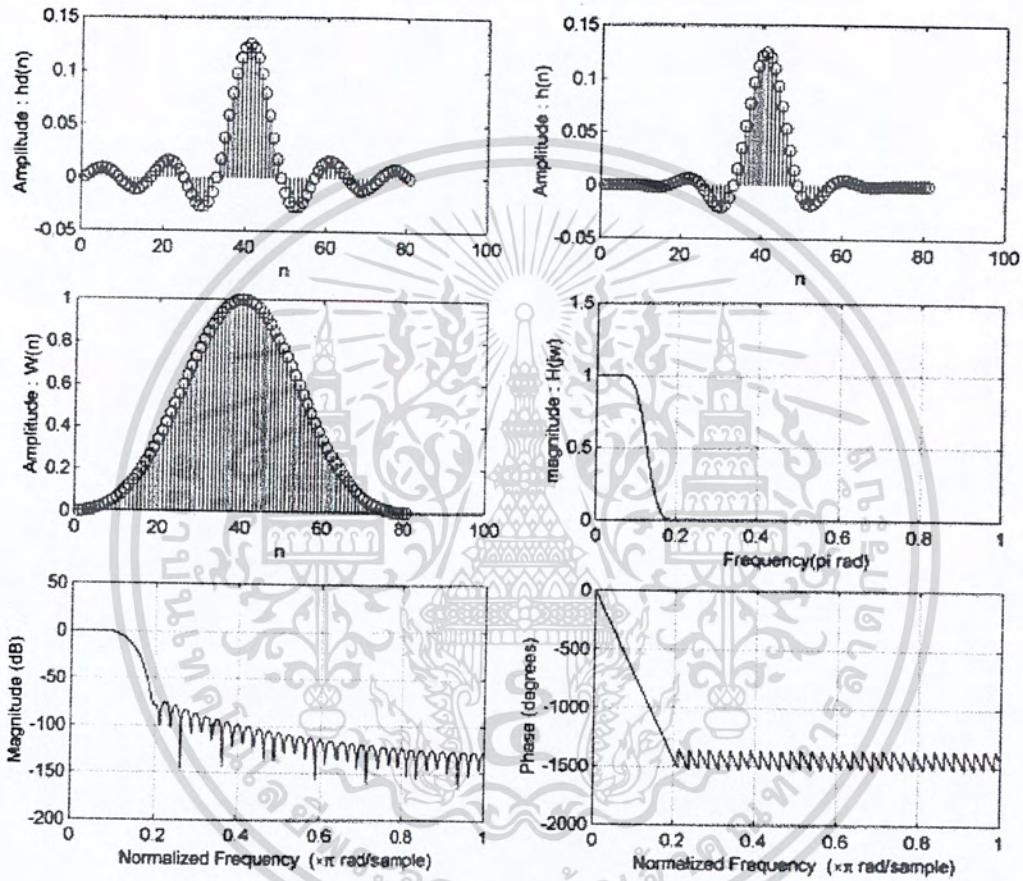
รูปที่ 3.10 ผลตอบสนองความถี่แถบหยุดของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแฮมมิง

- ก.) ผลตอบสนอง  $hd(n)$     ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$     ง.) สเปกตรัม  $H(jw)$   
 จ.) สเปกตรัม (dB)    ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 หน้าต่างแบบแบล็กแมน (Blackman Window)

1) ตัวกรองความถี่ต่ำผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คutoff 1KHz และความถี่ในการสุ่ม 8 KHz

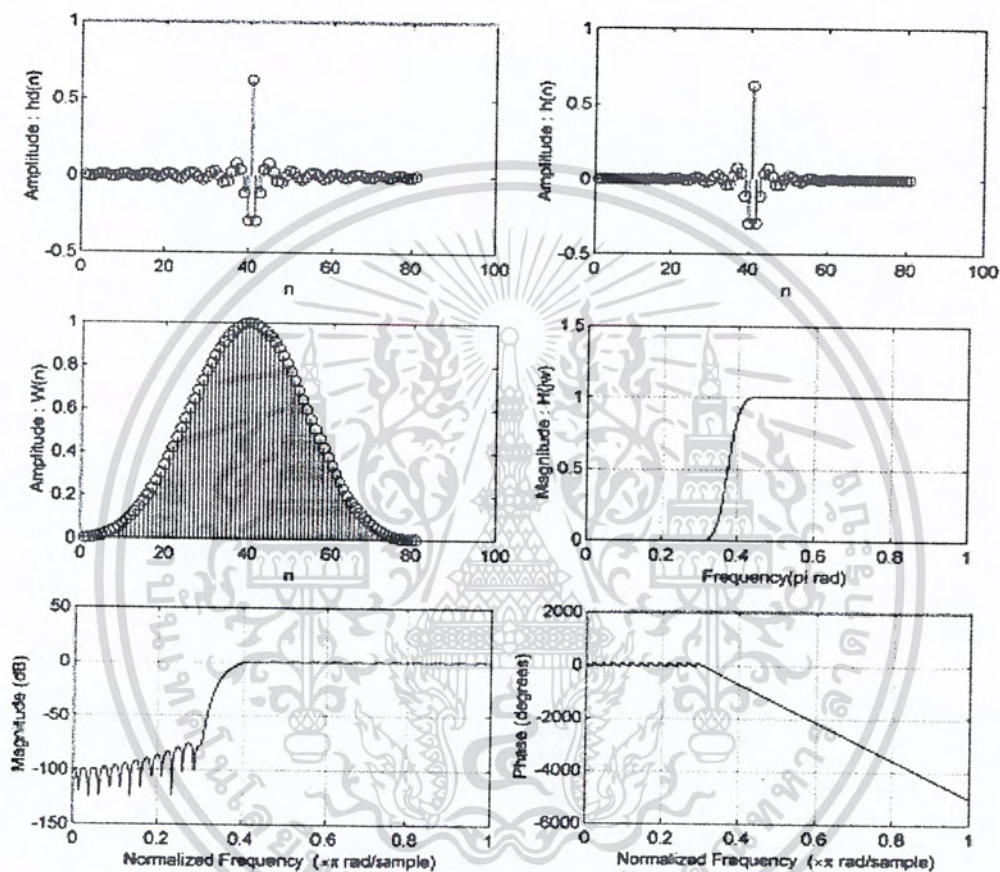


รูปที่ 3.11 ผลตอบสนองความถี่ต่ำผ่านของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแบล็กแมน

- ก.) ผลตอบสนอง  $hd(n)$       ข.) ผลตอบสนอง  $h(n) = hd(n) * w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$       ง.) สเปคตรัม  $H(j\omega)$   
 จ.) สเปคตรัม (dB)      ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ตัวกรองความถี่สูงผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ 3 KHz และ ความถี่ในการสุ่ม 8 KHz

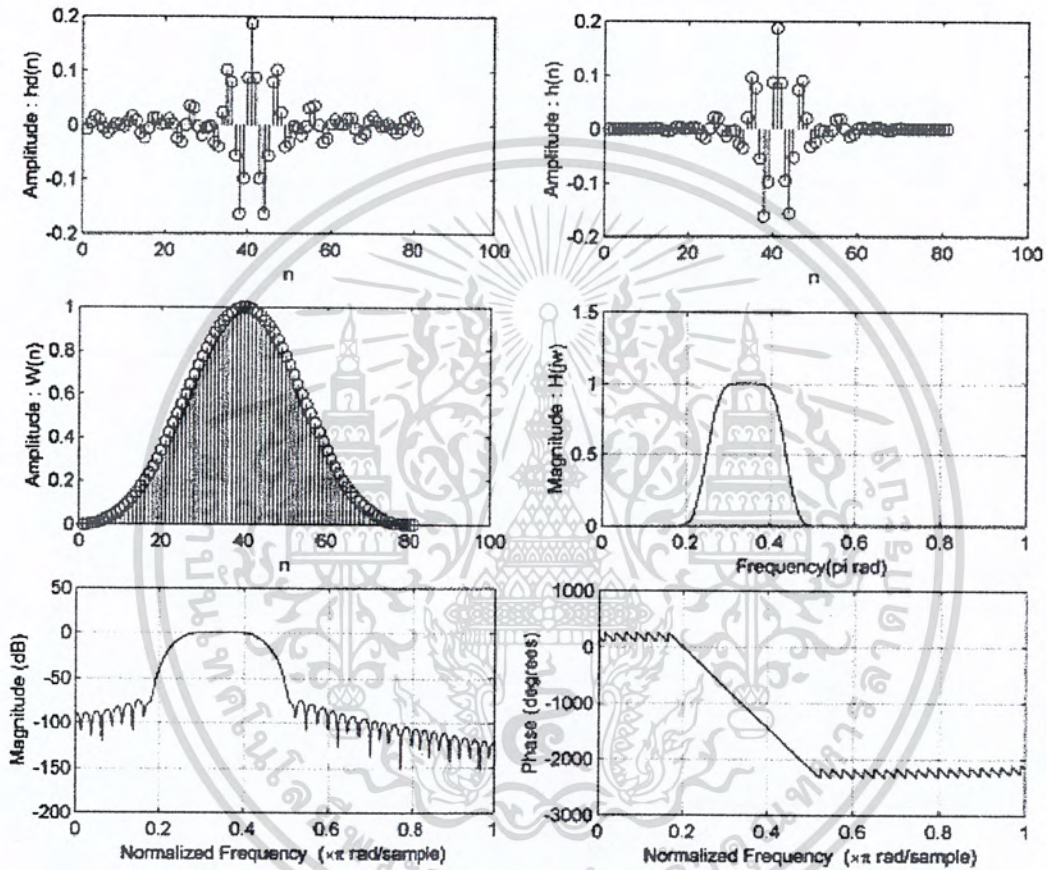


รูปที่ 3.12 ผลตอบสนองความถี่สูงผ่านของตัวกรองแบบจำกัดโดยวิธีหน้าต่างแบบแบล็คแมน

- ก.) ผลตอบสนอง  $hd(n)$       ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$   
 ค.) หน้าต่างสี่เหลี่ยม  $w(n)$       ง.) สเปกตรัม  $H(jw)$   
 จ.) สเปกตรัม (dB)      ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตัวกรองความถี่แถบผ่าน โดยผลตอบสนองแบบจำกัด มีความถี่คutoff  $f_{c1} = 2 \text{ KHz}$   
 $f_{c2} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$

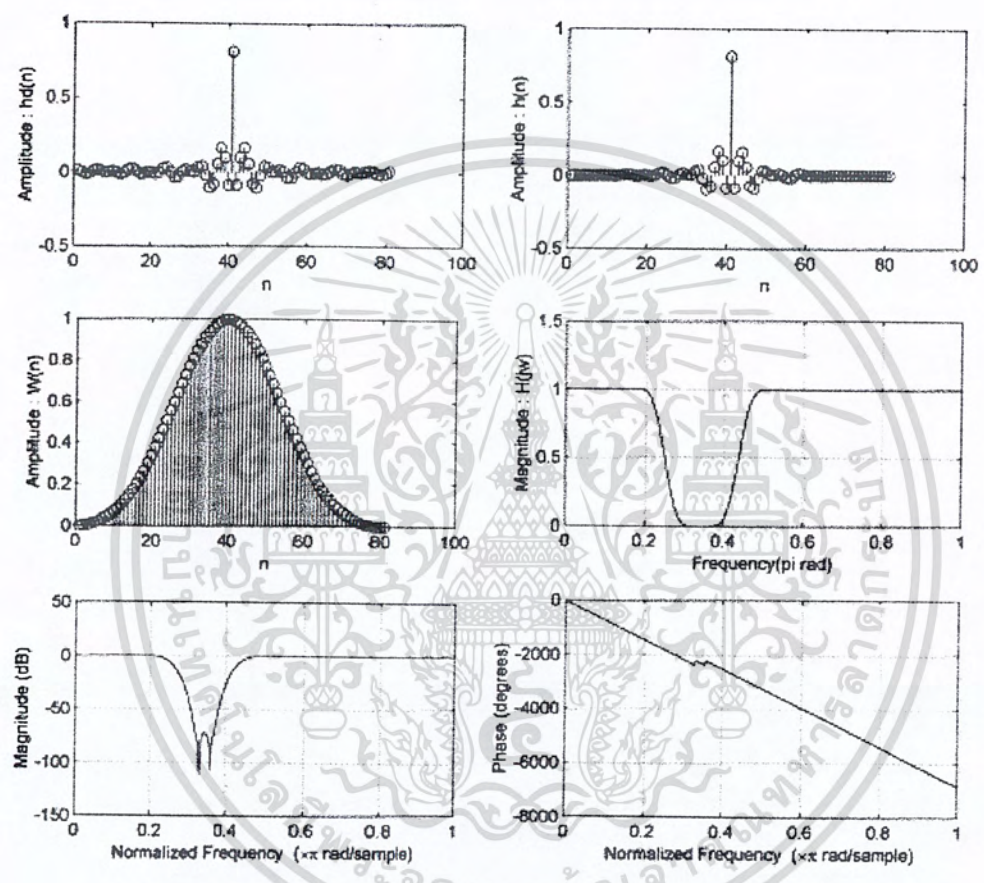


รูปที่ 3.13 ผลตอบสนองความถี่แถบผ่านของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแบ็ตติคแมน

- |                               |                                   |
|-------------------------------|-----------------------------------|
| ก.) ผลตอบสนอง $hd(n)$         | ข.) ผลตอบสนอง $h(n) = hd(n)*w(n)$ |
| ค.) หน้าต่างสี่เหลี่ยม $w(n)$ | ง.) สเปกตรัม $H(jw)$              |
| จ.) สเปกตรัม (dB)             | ฉ.) ผลตอบสนองทางเฟส               |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตัวกรองความถี่แถบหยุด โดยผลตอบสนองแบบจำกัด มีความถี่คัทออฟ  $f_{c1} = 2 \text{ KHz}$  ,  $f_{c2} = 3.5 \text{ KHz}$  และ ความถี่ในการสุ่ม  $f_s = 8 \text{ KHz}$



รูปที่ 3.14 ผลตอบสนองความถี่แถบหยุดของตัวกรองแบบจำกัด โดยวิธีหน้าต่างแบบแบล็คแมน

- ก.) ผลตอบสนอง  $hd(n)$     ข.) ผลตอบสนอง  $h(n) = hd(n)*w(n)$
- ค.) หน้าต่างสี่เหลี่ยม  $w(n)$     ง.) สเปกตรัม  $H(j\omega)$
- จ.) สเปกตรัม (dB)    ฉ.) ผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การออกแบบคอนโวลูชัน

ซึ่งโปรแกรมบน TMS320C50 ซีพียู จะเป็นการทำคอนโวลูชัน ของสัญญาณระหว่าง  $h(n)$  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)  $x(n)$  เมื่อเริ่ม โปรแกรมจะรันมาหนึ่งค่า

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

ซึ่ง

$x(0)$  เป็นค่าที่รันมา

$x(-1)$  เป็นค่าที่รับก่อน  $x(0)$

$x(-2)$  เป็นค่าที่รับมาก่อน  $x(-1)$

.

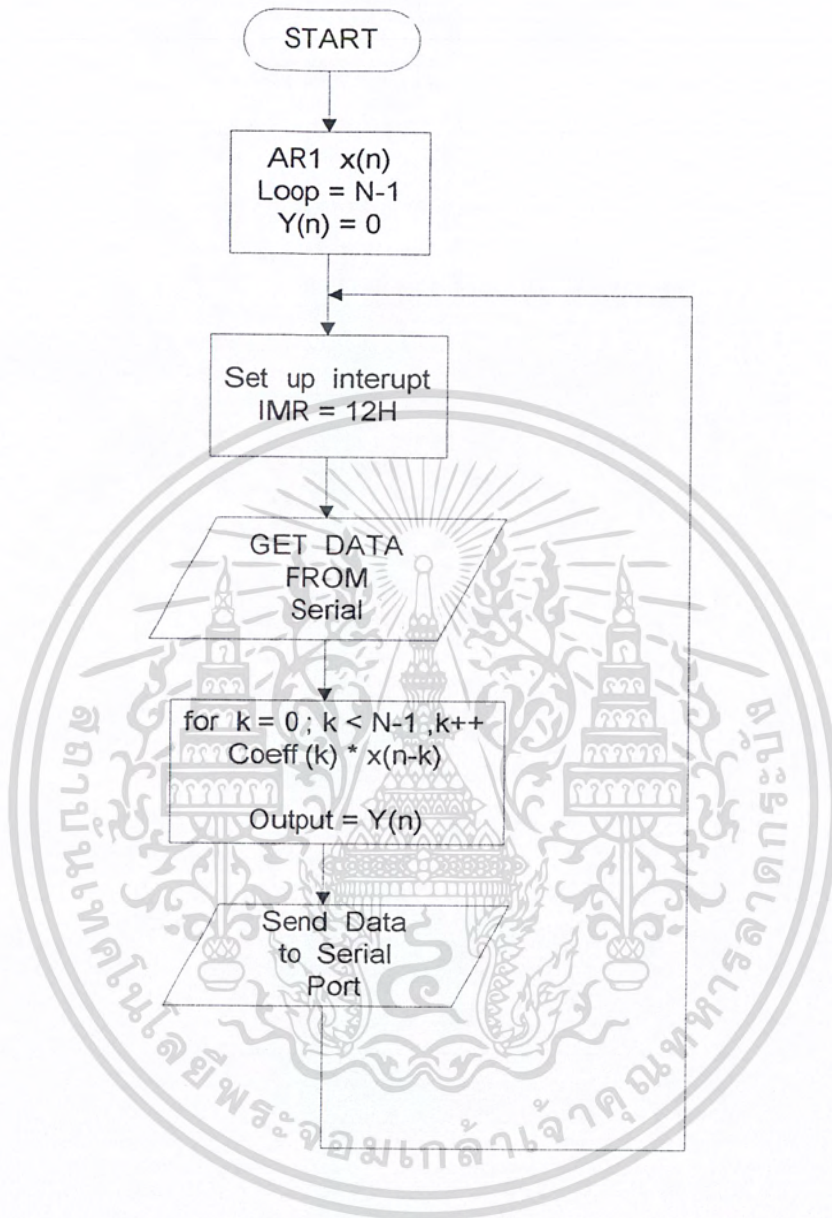
.

.

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

3) เมื่อรันมาเรื่อยๆ  $x(0)$  จะต้องเลื่อนมาที่  $x(-1)$  และ  $x(-1)$  จะต้องเลื่อนมาที่  $x(-2)$  จนถึง  $x(-(N-2))$  เลื่อนมาที่  $x(-(N-1))$  และต่อ  $x(-(N-1))$  สุดท้ายก่อนปล่อยออกไป

4) แต่ละอินพุต เข้ามาจะมีการคูณกันถึง  $N$  ครั้ง และบวกกัน  $N-1$  ครั้ง เพราะฉะนั้น จะต้องใช้ ความเร็วมาก ในการเขียนโปรแกรม



รูปที่ 3.15 แสดงโฟลว์ชาร์ท ของโปรแกรมการคอนโวลูชันบน TMS320C50

### 3.5 การออกแบบโปรแกรมเพื่อหาค่าสัมประสิทธิ์ของผลตอบสนอง $h(n)$

โดยใช้ภาษาซี แล้วทำการแปลงค่าจากเลขอิงคั้งนี้ (floating point) ให้เป็นเลขนวนเต็ม (fixed point) แล้วปัดเศษ เพื่อนำไปใช้ประมวลผลบน บอร์ด TMS320C50 เช่น ค่าเลขอิงคั้งนี้ ที่ได้คือ  $-0.0048$  สามารถแปลงให้เป็นเลขจำนวนเต็มคือ  $-0.0048 \times 2^{15} = -157$  ดังการทดลองใช้ โปรแกรมคำนวณหาค่าสัมประสิทธิ์ของ ตัวกรองความถี่แถบผ่านต่าง ๆ ดังนี้

1. สัมประสิทธิ์ของตัวกรองความถี่ต่ำผ่าน โดยมี  $f_c = 1 \text{ KHz}$  ด้วยวิธีหน้าต่างแบบแบ ล็คแมน ได้ผลดังนี้

สัมประสิทธิ์	ค่า	ค่า
$h(n)$	สัมประสิทธิ์ จริง	สัมประสิทธิ์ ที่นำมาใช้
$h(0)$	0	0
$h(1)$	-0.0048	-157
$h(2)$	-0.008	-261
$h(3)$	-0.0082	-268
$h(4)$	-0.0052	-170
$h(5)$	0	0
$h(6)$	0.0055	180
$h(7)$	0.0092	301
$h(8)$	0.0095	310
$h(9)$	0.006	198
$h(10)$	0	0
$h(11)$	-0.0065	-211
$h(12)$	-0.0108	-354
$h(13)$	-0.0112	-367
$h(14)$	-0.0072	-236
$h(15)$	0	0
$h(16)$	0.0078	255
$h(17)$	0.0132	431
$h(18)$	-0.0138	-451
$h(19)$	-0.0089	-292
$h(20)$	0	0
$h(21)$	-0.0098	-323
$h(22)$	-0.0168	-551
$h(23)$	-0.0178	-584
$h(24)$	-0.0117	-383
$h(25)$	0	0
$h(26)$	0.0134	438
$h(27)$	0.0233	763
$h(28)$	0.0252	827
$h(29)$	0.017	557
$h(30)$	0	0
$h(31)$	-0.0208	-681
$h(32)$	-0.0378	-1240
$h(33)$	-0.0432	-1417
$h(34)$	-0.0312	-1022
$h(35)$	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h(36)	0.0468	1533
h(37)	0.1009	3307
h(38)	0.1514	4960
h(39)	0.1871	6131
h(40)	0.2	6554
h(41)	0.1871	6131
h(42)	0.1514	4960
h(43)	0.1009	3307
h(44)	0.0468	1533
h(45)	0	0
h(46)	-0.0312	-1022
h(47)	-0.0432	-1417
h(48)	-0.0378	-1240
h(49)	-0.0208	-681
h(50)	0	0
h(51)	0.017	557
h(52)	0.0252	827
h(53)	0.0233	763
h(54)	0.0134	438
h(55)	0	0
h(56)	-0.0117	-383
h(57)	-0.0178	-584
h(58)	-0.0168	-551
h(59)	-0.0098	-323
h(60)	0	0
h(61)	0.0089	292
h(62)	0.0138	451
h(63)	0.0132	431
h(64)	0.0078	255
h(65)	0	0
h(66)	-0.0072	-236
h(67)	-0.0112	-367
h(68)	-0.0108	-354
h(69)	-0.0065	-211
h(70)	0	0
h(71)	0.006	198
h(72)	0.0095	310
h(73)	0.0092	301
h(74)	0.0055	100
h(75)	0	0
h(76)	-0.0052	-170
h(77)	-0.0082	-268
h(78)	-0.008	-261
h(79)	-0.48	-157
h(80)	0	0

ตารางที่ 3.2 ค่าสัมประสิทธิ์ของผลตอบ  
สนองความถี่ต่ำผ่านแบบแบล็คแมนวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สัมประสิทธิ์ของตัวกรอง  
ความถี่สูงผ่าน โดยมี  $f_c = 3$  KHz ด้วย  
วิธีหน้าต่างแบบแบล็คแมน ได้ผลดังนี้

สัมประสิทธิ์	ค่า สัมประสิทธิ์ จริง	ค่า สัมประสิทธิ์ ที่นำมาใช้
h(0)	0	0
h(1)	0	0
h(2)	-0.0001	-1
h(3)	-0.0003	-3
h(4)	0.0010	8
h(5)	0	0
h(6)	-0.0024	-19
h(7)	0.0021	17
h(8)	0.0029	24
h(9)	-0.0063	-52
h(10)	0	0
h(11)	0.0107	87
h(12)	-0.0084	-68
h(13)	-0.0105	-86
h(14)	0.0210	172
h(15)	0	0
h(16)	-0.0312	-256
h(17)	0.0233	191
h(18)	0.0279	228
h(19)	-0.0537	-440
h(20)	0	0
h(21)	0.0747	612

h(22)	-0.0541	-443
h(23)	-0.0632	-518
h(24)	0.1190	975
h(25)	0	0
h(26)	-0.1601	-1311
h(27)	0.1145	938
h(28)	0.1326	1086
h(29)	-0.2486	-2037
h(30)	0	0
h(31)	0.3373	2763
h(32)	-0.2450	-2007
h(33)	-0.2911	-2384
h(34)	0.5680	4653
h(35)	0	0
h(36)	-0.8966	-7345
h(37)	0.7521	6161
h(38)	1.1425	9359
h(39)	-3.7254	-30518
h(40)	3.9998	32767
h(41)	-3.7254	-30518
h(42)	1.1425	9359
h(43)	0.7521	6161
h(44)	-0.8966	-7345
h(45)	0	0
h(46)	0.5680	4653
h(47)	-0.2911	-2384
h(48)	-0.2450	-2007
h(49)	0.3373	2763
h(50)	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h(51)	-0.2486	-2037
h(52)	0.1326	1086
h(53)	0.1145	938
h(54)	-0.1601	-1311
h(55)	0	0
h(56)	0.1190	975
h(57)	-0.0632	-518
h(58)	-0.0541	-443
h(59)	0.0747	612
h(60)	0	0
h(61)	-0.0537	-440
h(62)	0.0279	228
h(63)	0.0233	191
h(64)	-0.0312	-256
h(65)	0	0
h(66)	0.0210	172
h(67)	-0.0105	-86
h(68)	-0.0084	-68
h(69)	0.0107	87
h(70)	0	0
h(71)	-0.0063	-52
h(72)	0.0029	24
h(73)	0.0021	17
h(74)	-0.0024	-19
h(75)	0	0
h(76)	0.0010	8
h(77)	-0.0003	-3
h(78)	-0.0001	-1
h(79)	0	0

h(80)	0	0
-------	---	---

ตารางที่ 3.3 ค่าสัมประสิทธิ์ของผลตอบ  
สนองความถี่สูงผ่านแบบเบตส์คแมนวินโดว์

3. สัมประสิทธิ์ของตัวกรอง  
ความถี่แถบผ่าน โดยมี  $f_{cl} = 3 \text{ KHz}$   
และ  $f_{ch} = 4 \text{ KHz}$  ด้วยวิธีหน้าต่างแบบ  
แบล็คแมน ได้ผลดังนี้

สัมประสิทธิ์	ค่า สัมประสิทธิ์ จริง	ค่า สัมประสิทธิ์ ที่นำมาใช้
h(0)	0	0
h(1)	0	0
h(2)	0.0001	1
h(3)	-0.0004	-6
h(4)	0.0003	5
h(5)	0.0013	22
h(6)	-0.0031	-50
h(7)	0.0008	13
h(8)	0.0061	100
h(9)	-0.0093	-153
h(10)	0	0
h(11)	0.0158	259
h(12)	-0.0175	-287
h(13)	-0.0040	-65
h(14)	0.0271	445
h(15)	-0.0216	-354
h(16)	-0.0096	-156
h(17)	0.0284	466
h(18)	-0.0138	-226
h(19)	-0.0064	-105
h(20)	0	0

h(21)	0.0089	146
h(22)	0.0268	439
h(23)	-0.0771	-1264
h(24)	0.0364	596
h(25)	0.1162	1904
h(26)	-0.2072	-3394
h(27)	0.0434	712
h(28)	0.2777	4549
h(29)	-0.3681	-6030
h(30)	0	0
h(31)	0.4993	8181
h(32)	-0.5132	-8408
h(33)	-0.1105	-1810
h(34)	0.7353	12047
h(35)	-0.5897	-9661
h(36)	-0.2740	-4489
h(37)	0.9180	15040
h(38)	-0.5649	-9255
h(39)	-0.4451	-7293
h(40)	1.1406	18688
h(41)	-0.4451	-7293
h(42)	-0.5649	-9255
h(43)	0.9180	15040
h(44)	-0.2740	-4489
h(45)	-0.5897	-9661
h(46)	0.7353	12047
h(47)	-0.1105	-1810
h(48)	-0.5132	-8408
h(49)	0.4993	8181

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h(50)	0	0
h(51)	-0.3681	-6030
h(52)	0.2777	4549
h(53)	0.0434	712
h(54)	-0.2072	-3394
h(55)	0.1162	1904
h(56)	0.0364	596
h(57)	-0.0771	-1264
h(58)	0.0268	439
h(59)	0.0089	146
h(60)	0	0
h(61)	-0.0064	-105
h(62)	-0.0138	-226
h(63)	0.0284	466
h(64)	-0.0096	-156
h(65)	-0.0216	-354
h(66)	0.0271	445
h(67)	-0.0040	-65
h(68)	-0.0175	-287
h(69)	0.0158	259
h(70)	0	0
h(71)	-0.0093	-153
h(72)	0.0061	100
h(73)	0.0008	13
h(74)	-0.0031	-50
h(75)	0.0013	22
h(76)	0.0003	5
h(77)	-0.0004	-6
h(78)	0.0001	1

h(79)	0	0
h(80)	0	0

ตารางที่ 3.4 ค่าสัมประสิทธิ์ของผลตอบ  
สนองความถี่แถบผ่านแบบแบสส์คแมนวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัมประสิทธิ์ของตัวกรอง  
ความถี่แถบหยุด โดยมี  $f_{cl} = 1 \text{ KHz}$   
และ  $f_{ch} = 3.5 \text{ KHz}$  ด้วยวิธีหน้าต่าง  
แบบแบล็คแมน ได้ผลดังนี้

สัมประสิทธิ์	ค่า สัมประสิทธิ์ จริง	ค่า สัมประสิทธิ์ ที่นำมาใช้
h(0)	0	0
h(1)	0	0
h(2)	-0.0004	-3
h(3)	-0.0003	-2
h(4)	0	0
h(5)	-0.0013	-11
h(6)	0.0023	19
h(7)	0.0036	29
h(8)	0	0
h(9)	0.0074	61
h(10)	0	0
h(11)	-0.0125	-103
h(12)	0	0
h(13)	-0.0179	-147
h(14)	-0.0207	-170
h(15)	0.0216	177
h(16)	0	0
h(17)	0.0203	167
h(18)	0.0722	591
h(19)	-0.100	-82
h(20)	0	0

h(21)	0.0139	114
h(22)	-0.1401	-1148
h(23)	-0.0552	-452
h(24)	0	0
h(25)	-0.1162	-952
h(26)	0.1583	1296
h(27)	0.1964	1609
h(28)	0	0
h(29)	0.2921	2393
h(30)	0	0
h(31)	-0.3963	-3246
h(32)	0	0
h(33)	-0.4991	-4089
h(34)	-0.5617	-4601
h(35)	0.5897	4831
h(36)	0	0
h(37)	0.6572	5384
h(38)	2.9578	24230
h(39)	-0.6933	-5679
h(40)	3.9998	32767
h(41)	-0.6933	-5679
h(42)	2.9578	24230
h(43)	0.6572	5384
h(44)	0	0
h(45)	0.5897	4831
h(46)	-0.5617	
h(47)	-0.4991	-4601
h(48)	0	0
h(49)	0.5897	-3246

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h(50)	0	0
h(51)	0.2921	2393
h(52)	0	0
h(53)	0.1964	1609
h(54)	0.1583	1296
h(55)	-0.1162	-952
h(56)	0	0
h(57)	-0.0552	-452
h(58)	-0.1401	-1148
h(59)	0.0139	114
h(60)	0	0
h(61)	-0.100	-82
h(62)	0.0722	591
h(63)	0.0203	167
h(64)	0	0
h(65)	0.0216	177
h(66)	-0.0207	-170
h(67)	-0.0179	-147
h(68)	0	0
h(69)	-0.0125	-103
h(70)	0	0
h(71)	0.0074	61
h(72)	0	0
h(73)	0.0036	29
h(74)	0.0023	19
h(75)	-0.0013	-11
h(76)	0	0
h(77)	-0.0003	-2
h(78)	-0.0004	-3

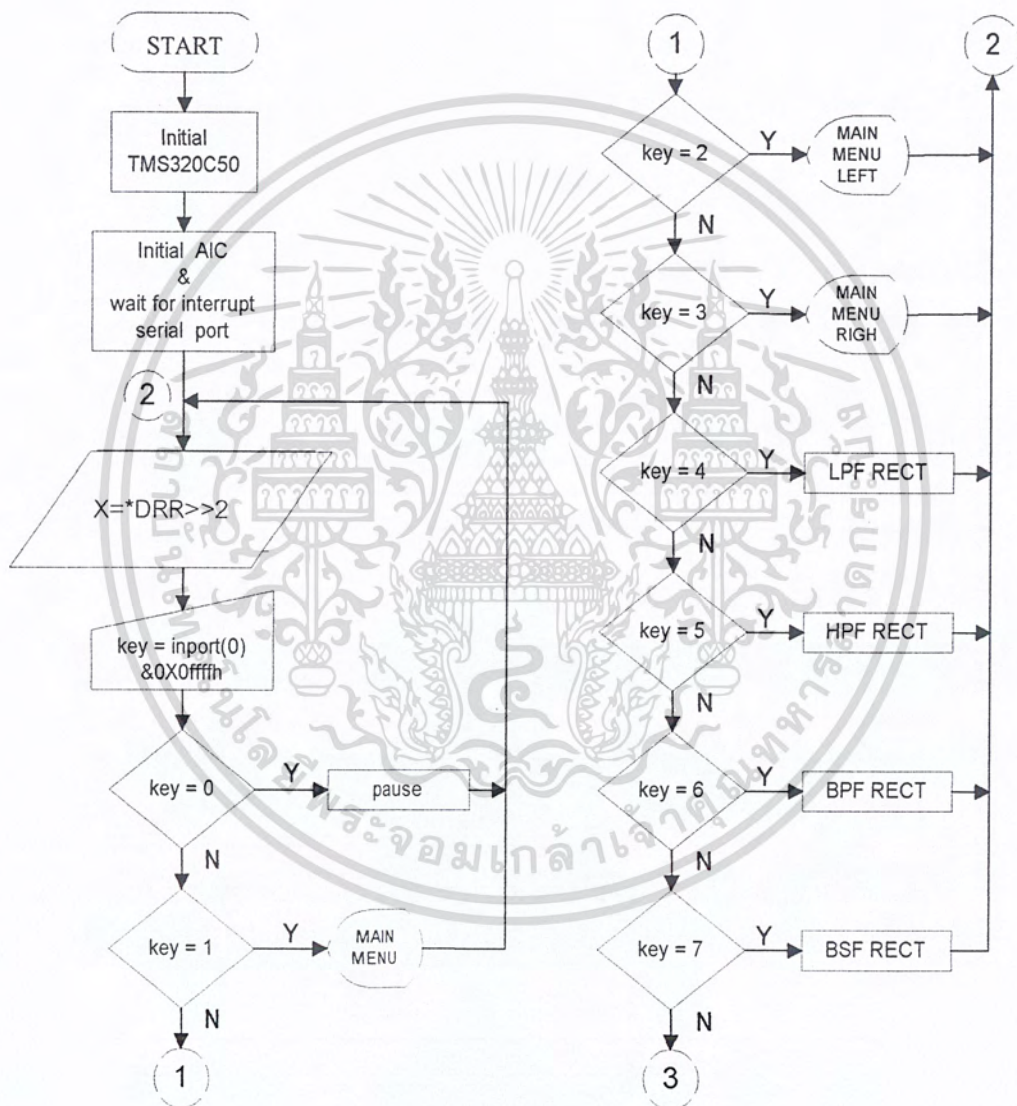
h(79)	0	0
h(80)	0	0

ตารางที่ 3.5 ค่าสัมประสิทธิ์ของผลตอบ  
สนองความถี่แถบหยุดแบบเบ็คแมนวินโดว์

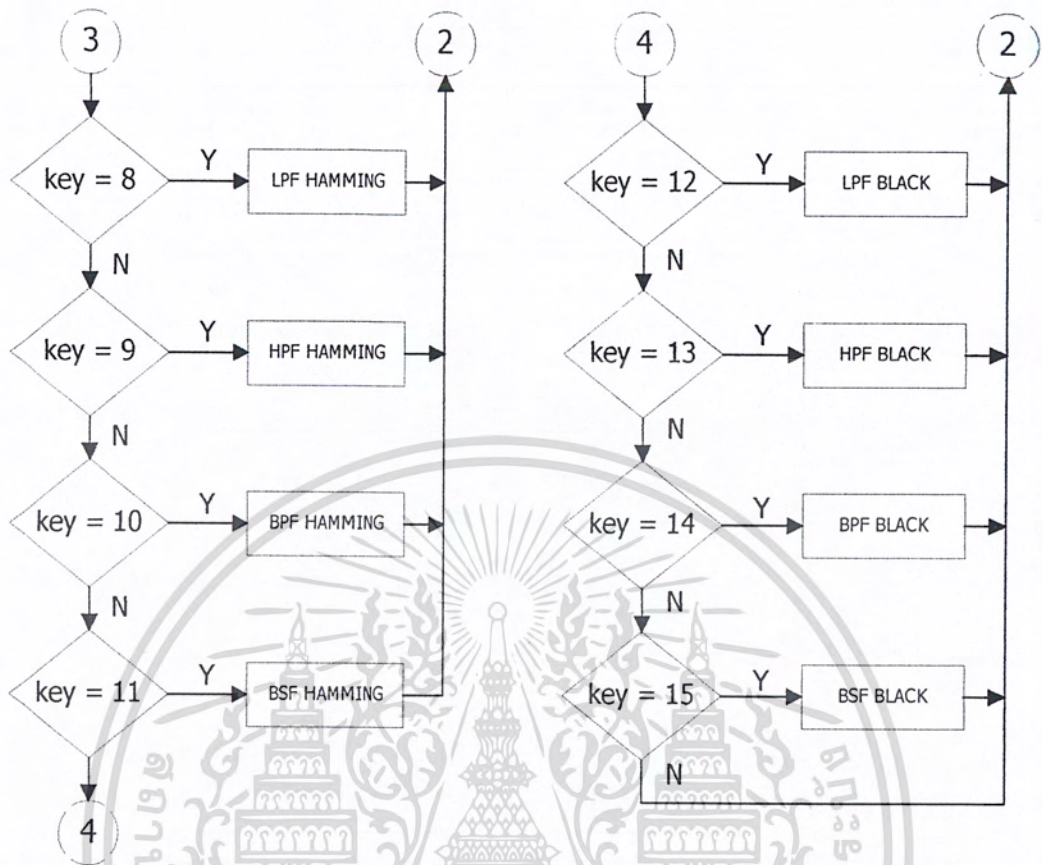
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 การออกแบบโปรแกรมในการประมวลผลตัวกรองและการควบคุมอุปกรณ์ แอล ซี ดี และคีย์บอร์ด

ในการออกแบบโปรแกรม จะใช้ภาษาซีและภาษาแอสเซมบลีร่วมกัน ดังแสดงในภาคผนวก และสามารถเขียนโปรแกรมได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงโฟลว์ชาร์ต ของโปรแกรมประมวลผลบน TMS320C50

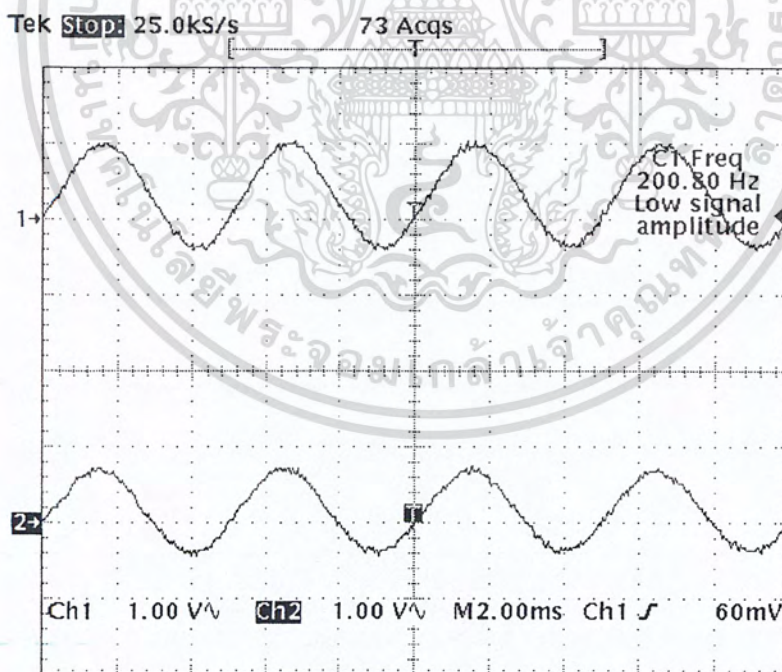
## บทที่ 4

### ผลการทดลอง

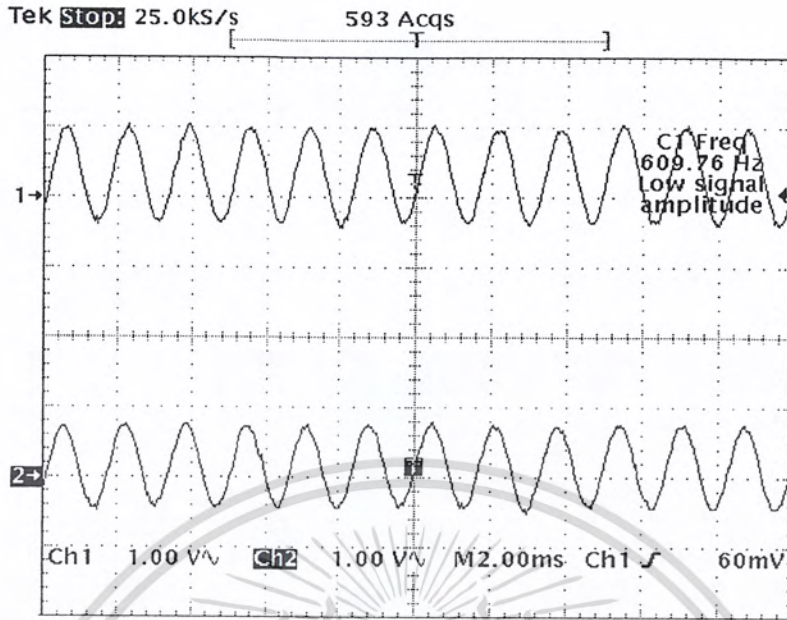
ในการทดลองเมื่อเราทำการออกแบบตัวกรองเชิงเลข FIR แล้ว นำมาเขียนโปรแกรมแล้วรันด้วยเครื่องคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม RS232 ทำการประมวลผลสัญญาณ ด้วยตัวประมวลผลสัญญาณดิจิทัล TMS320C50 โดยมีการเชื่อมต่อกับอุปกรณ์ต่างๆ ได้แก่ บอร์ด DSP Starter Kits TMS320C50, เครื่องกำเนิดสัญญาณ (Generator), และสโคป

#### 4.1 ผลการทดลองการกรองความถี่แบบต่างๆ

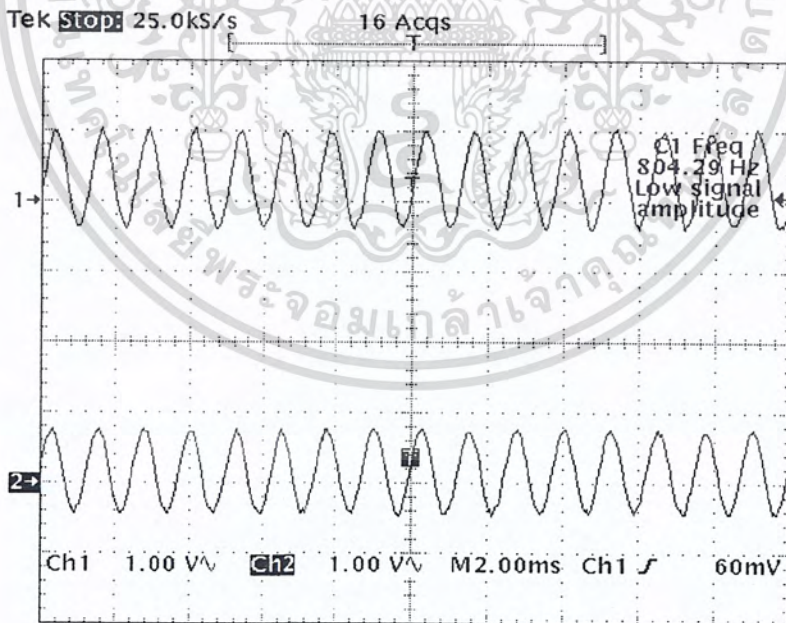
4.1.1 การกรองความถี่ต่ำผ่าน ดังภาคผนวก มีความถี่คัทออฟ 1 KHz โดยการป้อนสัญญาณอินพุตที่เป็นสัญญาณรูปไซน์



รูปที่ 4.1 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 200 Hz

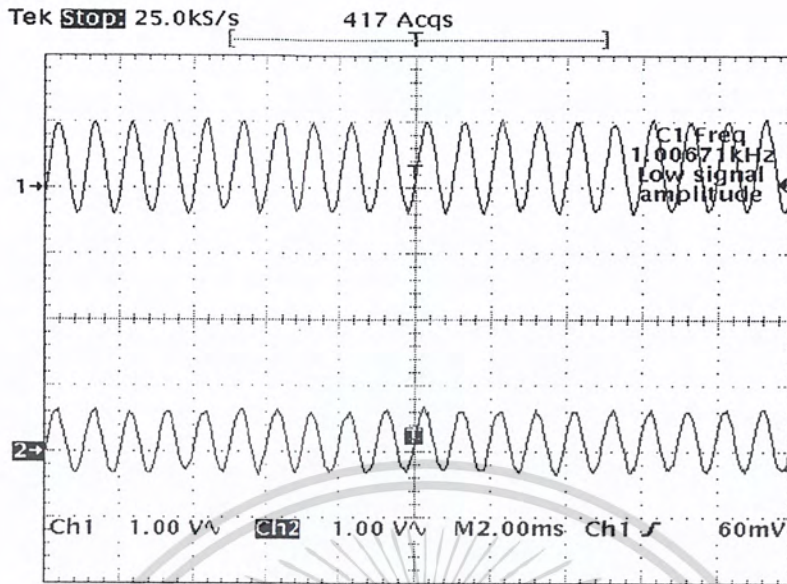


รูปที่ 4.2 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 600 Hz

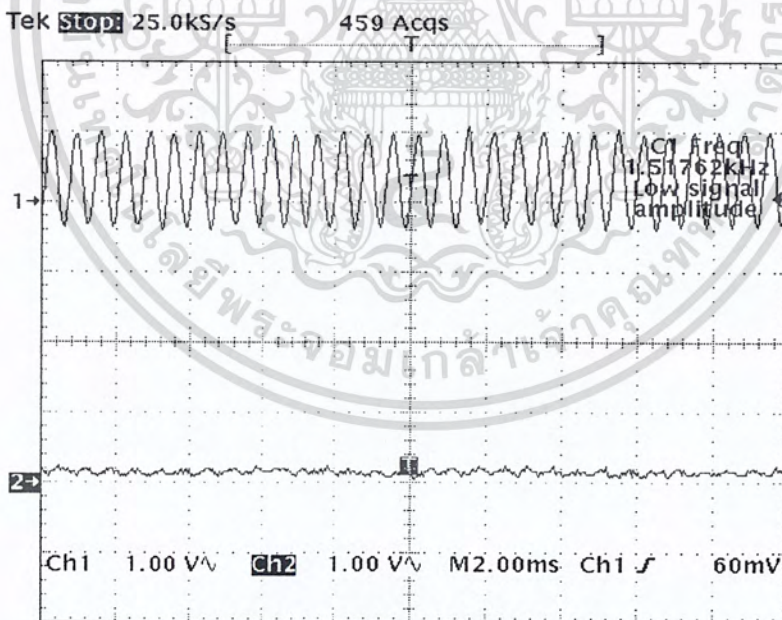


รูปที่ 4.3 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 800 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

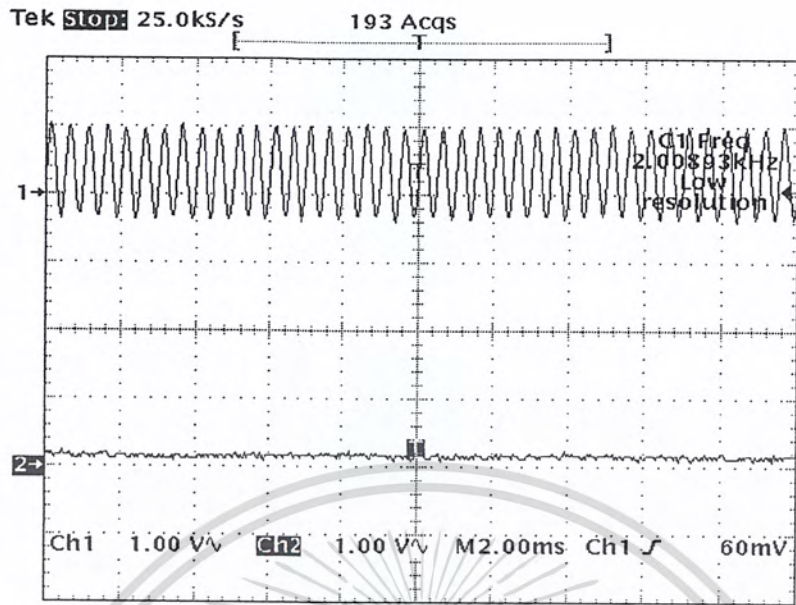


รูปที่ 4.4 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 1 KHz



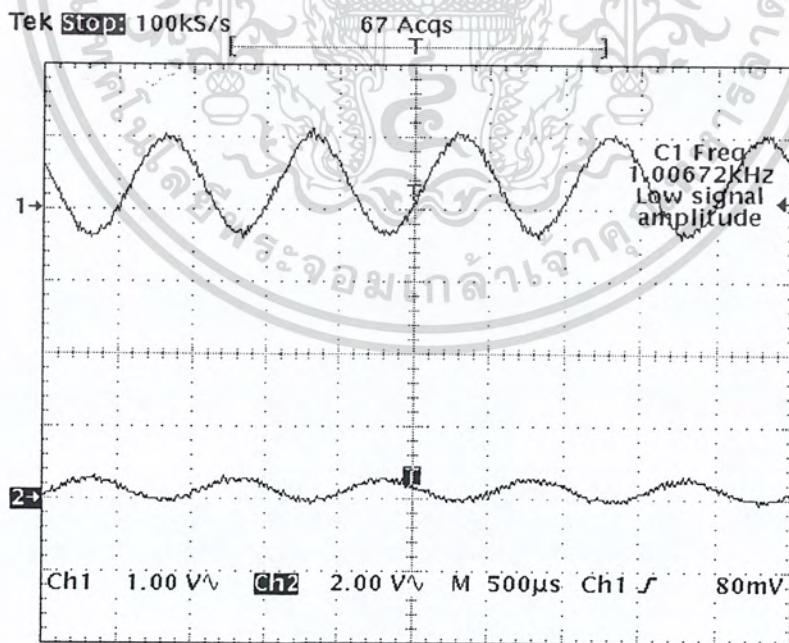
รูปที่ 4.5 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 1.5 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



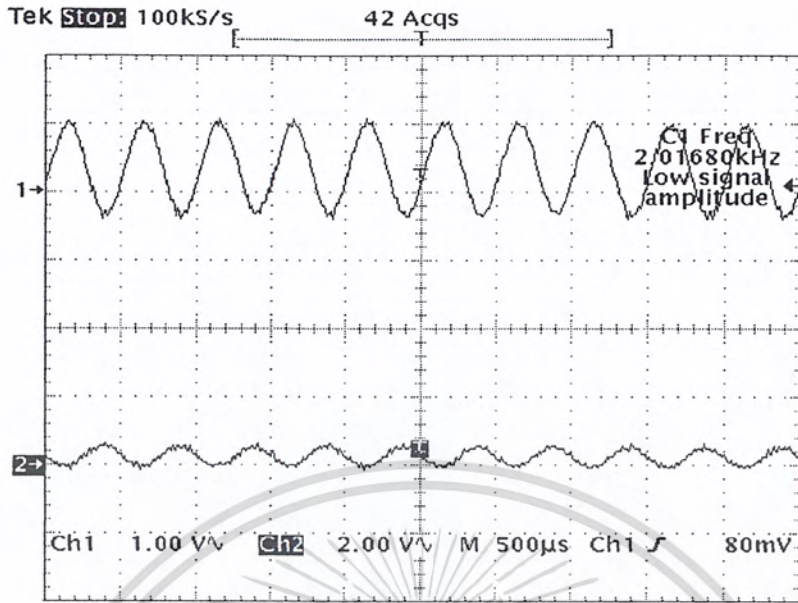
รูปที่ 4.6 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่ต่ำผ่านเมื่ออินพุตมีความถี่ 2 KHz

4.1.2 การกรองความถี่สูงผ่าน ตั้งโปรแกรมในภาคผนวก มีความถี่คัทออฟ 3 KHz โดยป้อนสัญญาณอินพุตเป็นสัญญาณรูปซายน์

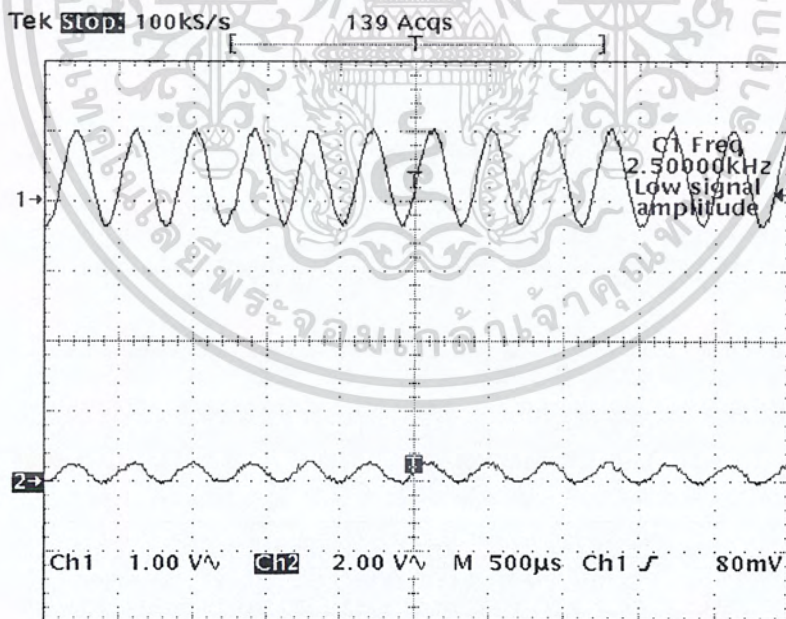


รูปที่ 4.7 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่สูงผ่านเมื่ออินพุตมีความถี่ 1 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

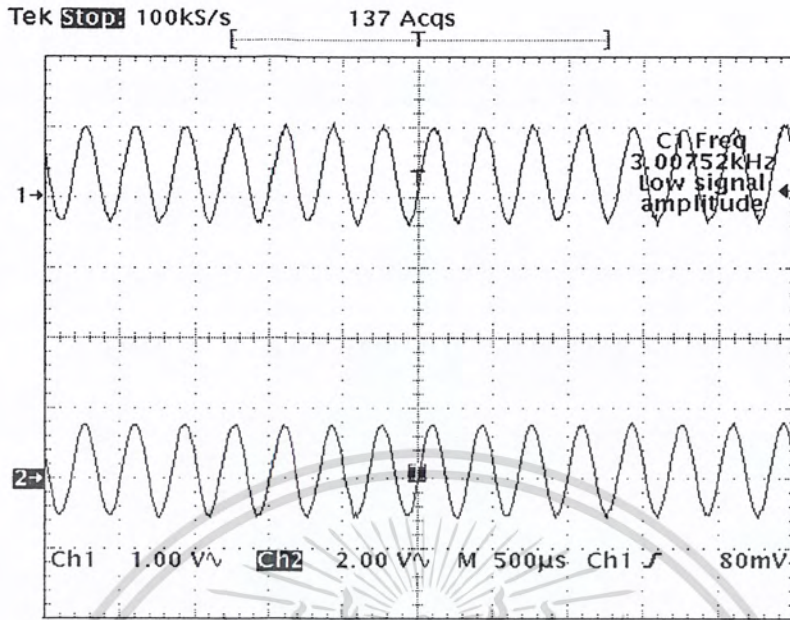


รูปที่ 4.8 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่สูงผ่านเมื่ออินพุตมีความถี่ 2 KHz



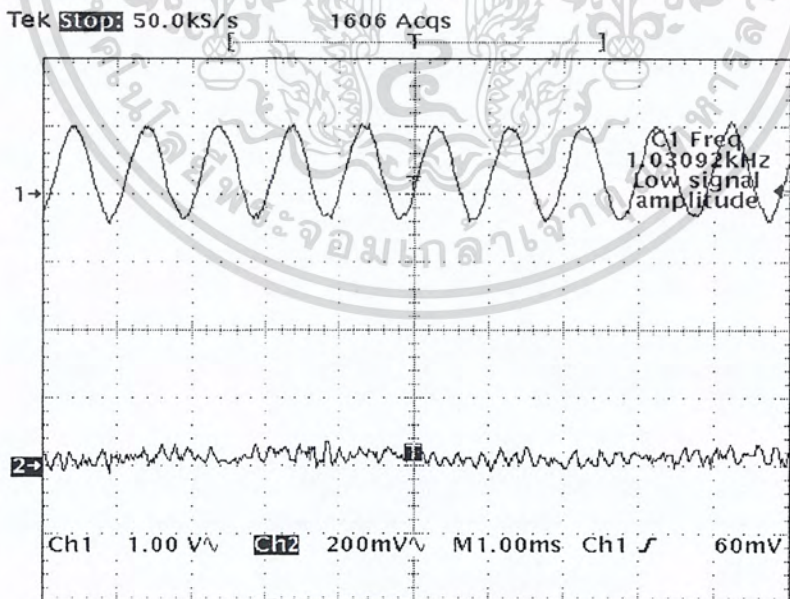
รูปที่ 4.9 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่สูงผ่านเมื่ออินพุตมีความถี่ 2.5 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

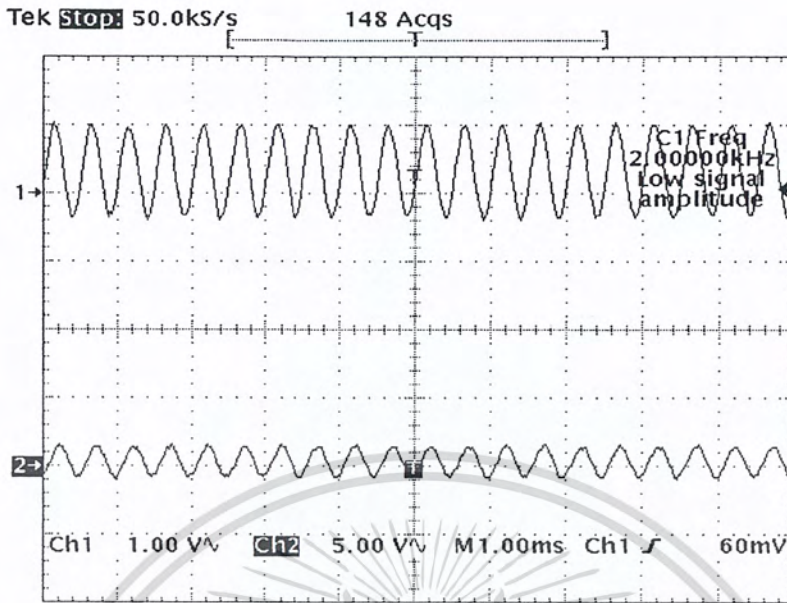


รูปที่ 4.10 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่สูงผ่านเมื่ออินพุตมีความถี่ 3 KHz

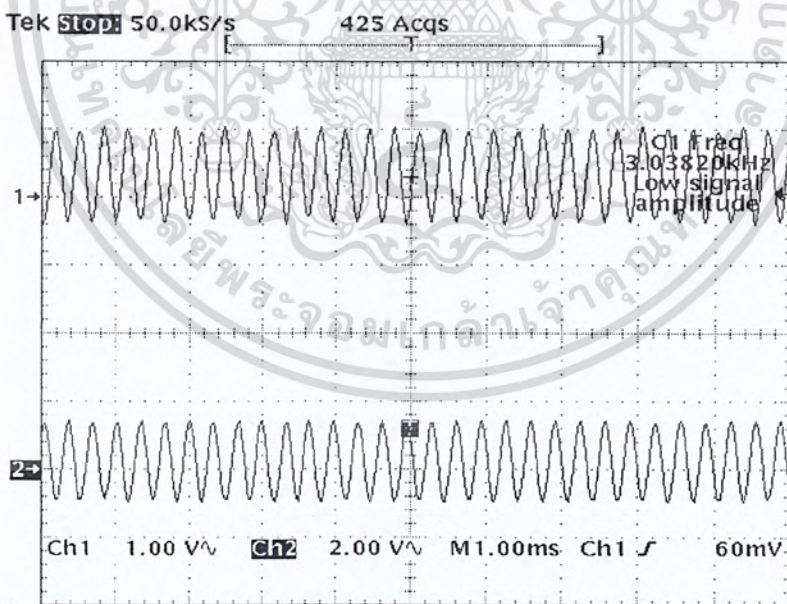
4.1.3 การกรองความถี่แถบผ่าน ดังโปรแกรมในภาคผนวก มีความถี่คัทออฟด้านต่ำ 2KHz และมีความถี่คัทออฟด้านสูง 3.5 KHz



รูปที่ 4.11 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบผ่านเมื่ออินพุตมีความถี่ 1 KHz

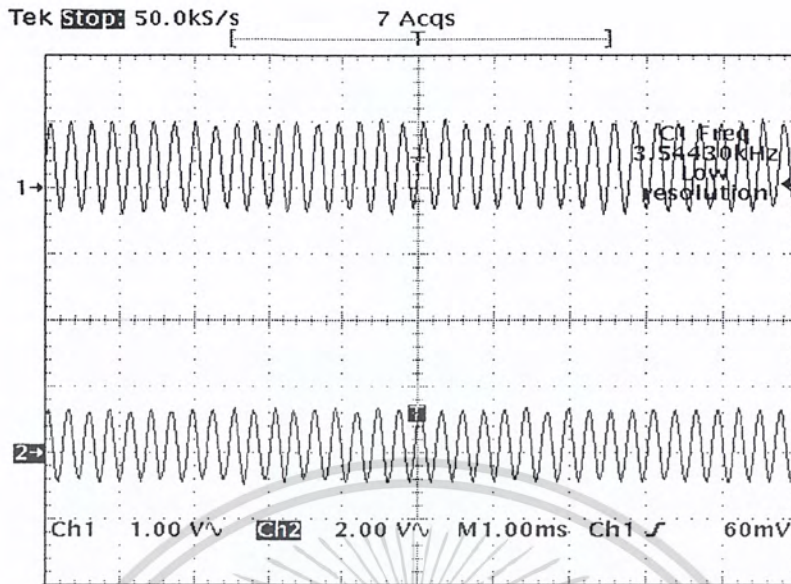


รูปที่ 4.12 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบผ่านเมื่ออินพุตมีความถี่ 2 KHz



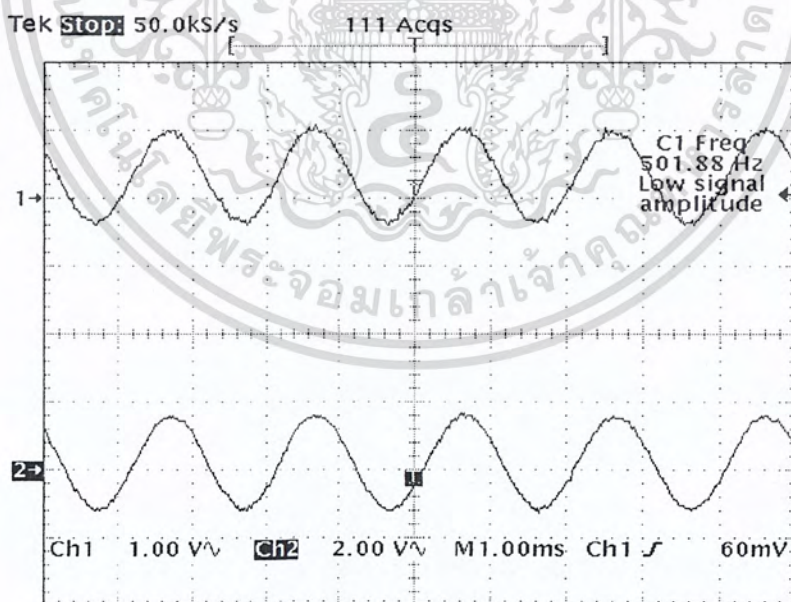
รูปที่ 4.13 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบผ่านเมื่ออินพุตมีความถี่ 3 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

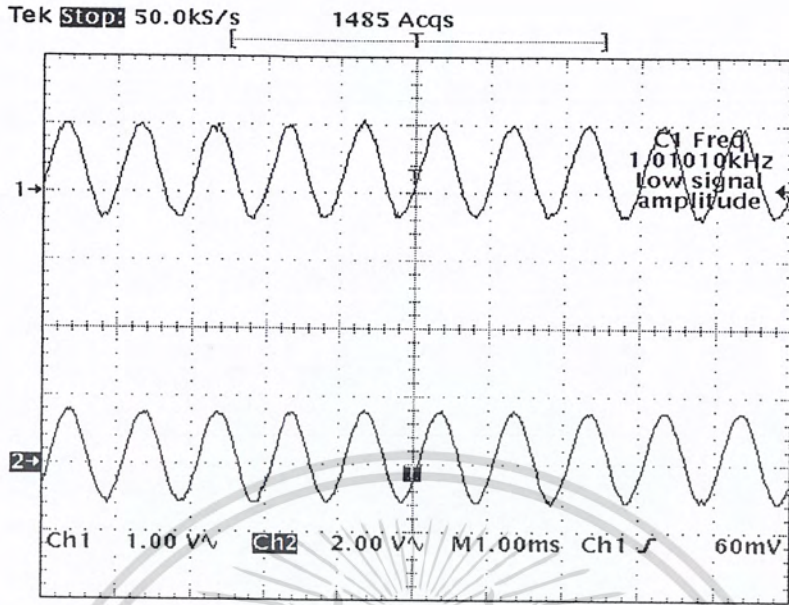


รูปที่ 4.14 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบผ่านเมื่ออินพุตมีความถี่ 3.5 KHz

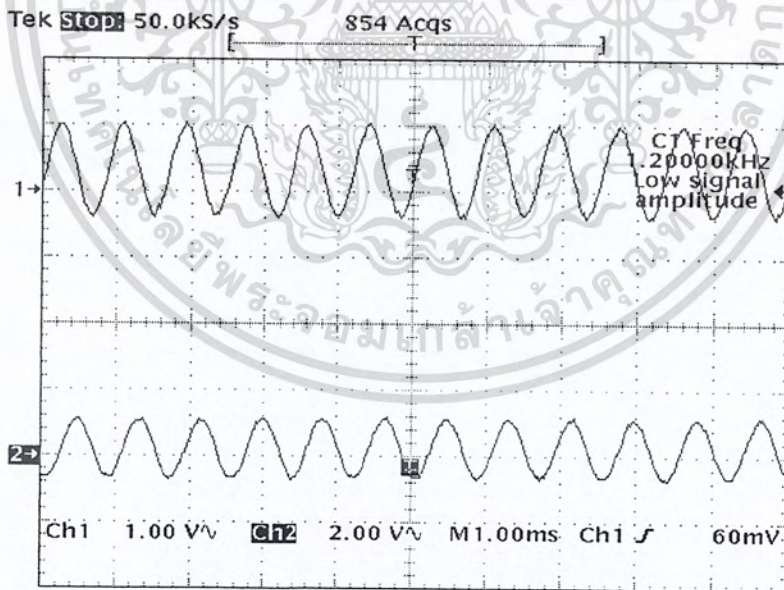
4.1.4 การกรองความถี่แถบหยุด ดังโปรแกรมในภาคผนวก มีความถี่คัทออฟด้านต่ำ 2 KHz และมีความถี่คัทออฟด้านสูง 3.5 KHz



รูปที่ 4.15 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 500 Hz

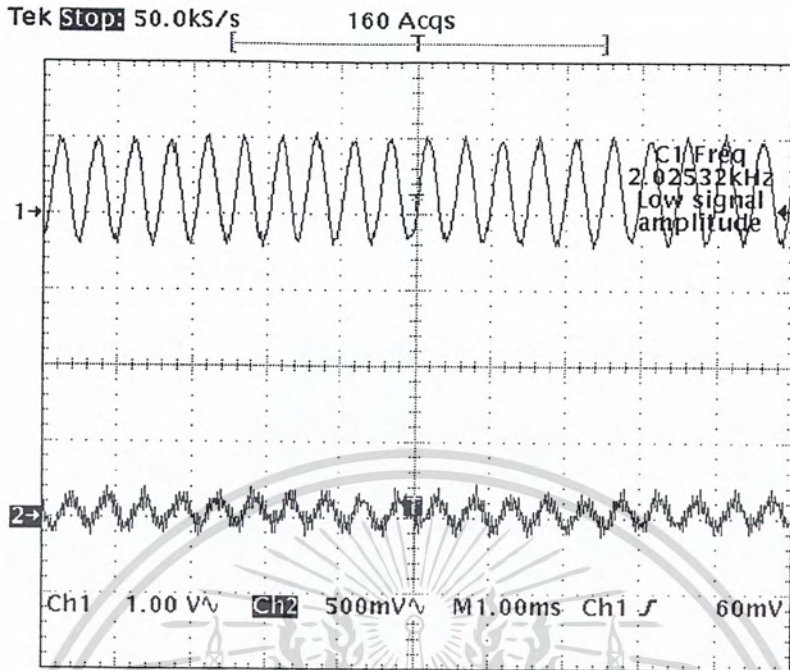


รูปที่ 4.16 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 1 KHz

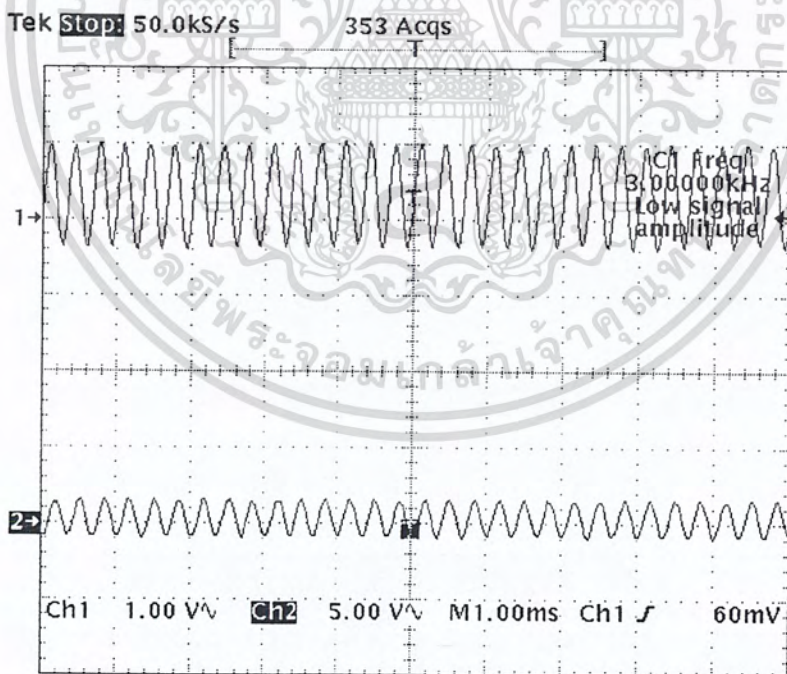


รูปที่ 4.17 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 1.2 KHz

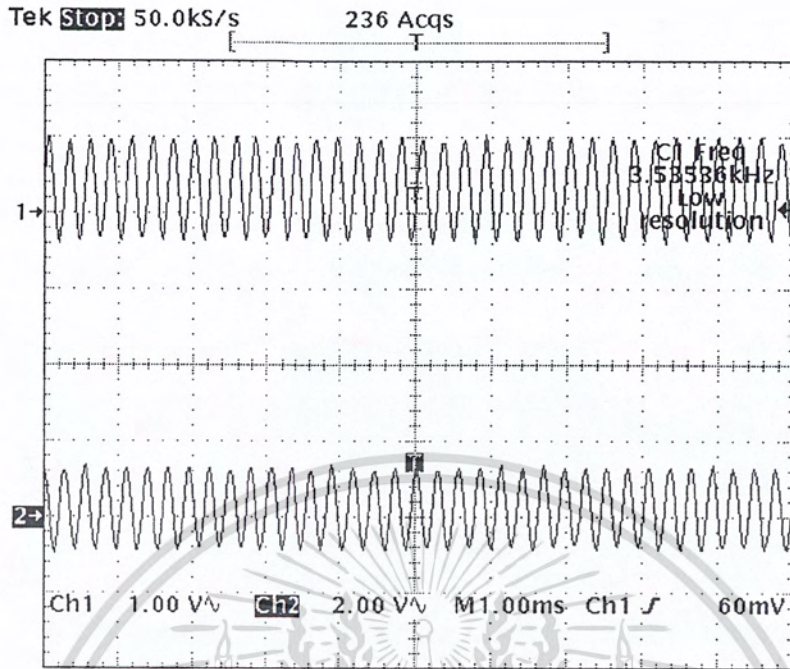
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 2 KHz



รูปที่ 4.19 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 3 KHz



รูปที่ 4.20 สัญญาณเอาต์พุตที่ผ่านตัวกรองความถี่แถบหยุดเมื่ออินพุตมีความถี่ 3.5 KHz

## บทที่ 5

### สรุปและวิจารณ์ผลการทดลอง

จากการทดลองการใช้บอร์ดทำตัวกรองเชิงเลข (Digital Filter) ซึ่งประกอบด้วยตัวกรองความถี่ต่ำผ่าน (Lowpass Filter), กรองความถี่สูงผ่าน (Highpass Filter), กรองความถี่แถบผ่าน (Bandpass Filter), และกรองความถี่แถบหยุด (Bandstop Filter) โดยการออกแบบด้วยวิธีหน้าต่างแบบแบล็กแมน (Blackman Window) จะให้ผลตอบสนองต่อความถี่ที่เราต้องการออกแบบได้ดีกว่าการออกแบบด้วยวิธี หน้าต่างแฮมมิง (Hamming Window) และวิธีการออกแบบแบบหน้าต่างสี่เหลี่ยม (Rectangular Window)

เนื่องจากบอร์ดทดลอง TMS320C50 เป็นบอร์ดประมวลผลแบบจำนวนเต็ม 16 บิต ดังนั้นผลการทดลองที่ได้ จึงมีความคลาดเคลื่อนมาก เมื่อเทียบกับผลที่ได้จากการคำนวณทางโปรแกรม MATLAB ซึ่งเป็นการประมวลผลแบบเลขอิงค์ซ์นี้ โดยความคลาดเคลื่อนเกิดขึ้นเนื่องจากการแปลงค่าสัมประสิทธิ์ของตัวกรองจากเลขอิงค์ซ์นี้ เป็นเลขจำนวนเต็มแล้วปิดเศษเพื่อให้ TMS320C50 ประมวลผลได้

ในการใช้งานบอร์ด TMS320C50 โดยการโปรแกรมด้วยภาษาซีนั้น จะทำให้การออกแบบและพัฒนา งานเป็นไปด้วยความสะดวกและรวดเร็วแต่จะมีความช้าในการประมวลผล ในขณะที่โปรแกรมภาษาแอสเซมบลีนั้น จะมีความเร็วในการประมวลผล แต่การออกแบบและพัฒนาที่มีความยากกว่าภาษาซี ดังนั้นเพื่อความยืดหยุ่นของข้อดีและข้อด้อยของภาษาทั้งสอง จึงควรออกแบบและใช้งานร่วมกัน

บอร์ด TMS320C50 ต้องการเวลาประมวลผลแบบเวลาจริงที่เร็ว ดังนั้นในการออกแบบค่าสัมประสิทธิ์ที่มีจำนวนมาก จะทำให้ความสามารถในการประมวลผลค่อยลง ซึ่งเป็นข้อเสียของ FIR ในข้อนี้เมื่อเทียบกับ IIR และจะเห็นได้ชัดเจนเมื่อการออกแบบโปรแกรมด้วยภาษาซี เพราะภาษาซี 1 คำสั่งจะต้องแปลงเป็น แอสเซมบลีหลายคำสั่ง

ในการพัฒนางานด้วยโปรแกรมภาษาซีที่มีขนาดใหญ่ จะต้องการหน่วยความจำโปรแกรมมากดังนั้นจึงควรพัฒนาให้ใช้หน่วยความจำภายนอกได้

## บรรณานุกรม

1. R.D. Strum and D.E. Kirk, "First Principle of Discrete System and Digital Signal Processing" Addison Wesley, 1988, USA.
2. TEXAS INSTRUMENTS, "TMS320C5X DSP Start Kit" TexasIntruments Incorporated , 1994,USA.
3. TEXAS INSTRUMENTS," TMS320C5X DSP User's Guide " Texas Intruments Incorporated , 1993,USA .
4. S.K.Mitra, Digital Signal Processing A Computer-Based Approach, McGraw-Hill, Singapore.
5. Reimer, J.B. nd K.S Lin, "TMS320 Digital Signal Processor in Speech Application. " Proceedings of SPEECH TECH ' 88, April 1988
6. พรชัย ภาวงษ์ศักดิ์, " การประมวลผลสัญญาณดิจิทัลเบื้องต้น " มหาวิทยาลัยเทคโนโลยีมห นคร, 2542



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 1. โปรแกรมหาค่าผลตอบสนอง โดย MATLAB

```

subplot(2,1,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(1),stem(w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xlabel('n');ylabel('Amplitude : W(n)')
% Lopass Filter by rectangular window
figure(2),stem(h)
method %
xlabel('n');ylabel('Amplitude : h(n)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[X,f]= freqz(h,1,512,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3),plot(f,abs(X))
xlabel('Frequency(Hz)');
ylabel('magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)
fc=1000;
fs=8000;
N=81;
wc=(fc*pi)/fs;
w=Boxcar(n);
for n=1:40
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi); %%%%%%%%%
    h(n)=hd(n)*w(n); %%%
end % Hipass Filter by Rectangular window method
for n=41
    %
    hd(n)=(wc/pi); %%%%%%%%%
    h(n)=hd(n)*w(n); %%%
end
for n=42:81
    fc=3000;
    fs=8000;
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    fs=8000;
    N=81;
    h(n)=hd(n)*w(n);
    w=boxcar(N);
end
subplot(2,1,1);
figure(1),stem(hd)
wc=(fc*pi)/fs;
for n=1:40
    hd(n)=-sin(wc*(n-41))/((n-41)*pi);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

h(n)=hd(n)*w(n);          %%%%%%%%%%
end                          %%%%
for n=41                     % Bandpass Filter by rectangular window
    hd(n)=1-(wc/pi);        method %
    h(n)=hd(n)*w(n);        %%%%%%%%%%
end                          %%%%

for n=42:81
    hd(n)=-(sin(wc*(n-41)))/((n-41)*pi);    fcl=2000;
    h(n)=hd(n)*w(n);                        fch=3500;
end                                          fs=8000;
subplot(211);                               N=81;
figure(1),stem(hd)                          wcl=(fcl*pi)/fs;
xlabel('n');ylabel('Amplitude : hd(n)')      wch=(fch*pi)/fs;
subplot(212);                               w=boxcar(N);
figure(1),stem(w)                            for n=1:81
                                              hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-
xlabel('n');ylabel('Amplitude : W(n)')      41)*pi);
figure(2),stem(h)                            h(n)=hd(n)*w(n);
xlabel('n');ylabel('Amplitude : h(n)')      end
[X,f]= freqz(h,1,512,2);                    for n=41
figure(3),plot(f,20*log10(abs(X)))          hd(n)=(wch-wcl)/pi;
xlabel('Frequency(Hz)');                    h(n)=hd(n)*w(n);
ylabel('Magnitude : H(jw)');                end
grid on                                     for n=42:81
figure(4),freqz(h,1,512)                    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-
                                              41)*pi);
                                              h(n)=hd(n)*w(n);
                                              end
                                              subplot(2,1,1);
                                              figure(1),stem(hd);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

xlabel('n');ylabel('Amplitude : hd(n)');
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bandstop Filter by Rectangular
window method %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=boxcar(N);
for n=1:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)');grid on
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

```

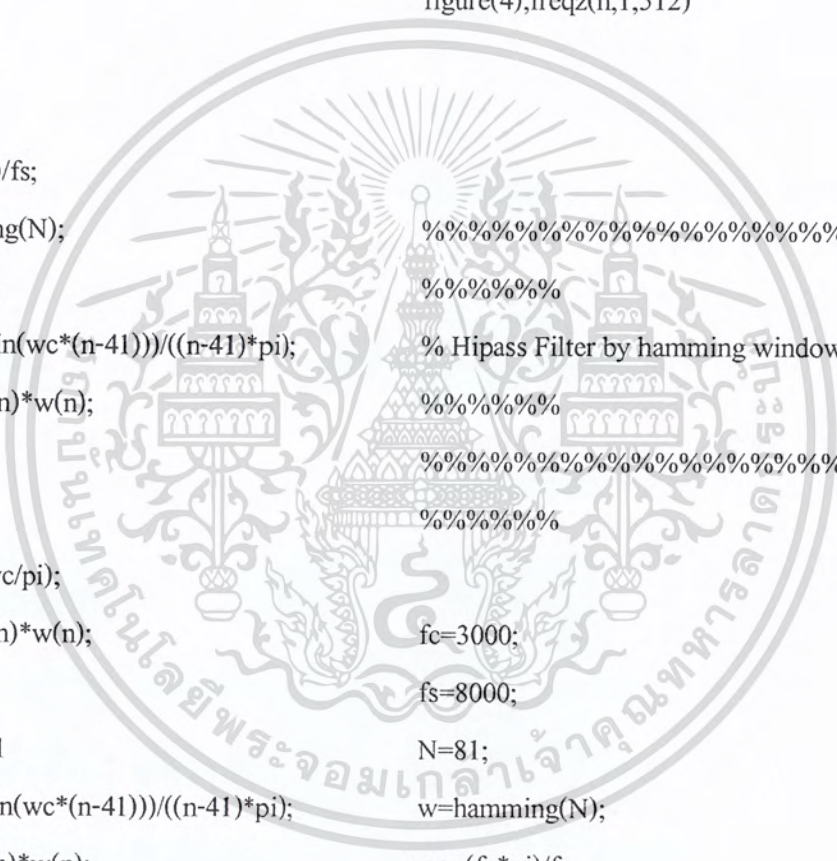
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lopass Filter by hamming window
method %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc=1000;
fs=8000;
N=81;
wc=(fc*pi)/fs;
w=hamming(N);
for n=1:40
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wc/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(2,1,2);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')

figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Hipass Filter by hamming window method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc=3000;
fs=8000;
N=81;
w=hamming(N);
for n=1:40
    hd(n)=-sin(wc*(n-41))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=1-(wc/pi);
    h(n)=hd(n)*w(n);
end

```



```

end
for n=42:81
    hd(n)=-((sin(wc*(n-41)))/((n-41)*pi));
    h(n)=hd(n)*w(n);
end
subplot(211);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(212);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bandpass Filter by hamming
window method %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=hamming(N);
for n=1:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wch-wcl)/pi;
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd);
xlabel('n');ylabel('Amplitude : hd(n)');
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))

```

```

xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

end
for n=41
    hd(n)=1+((wcl-wch)/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-
41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)');grid on
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% Bandstop Filter by hamming window
method %
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=hamming(N);
for n=1:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-
41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);

```

```

%%%%%%%%%%
%%%%%%%%%
% Lopass Filter by hanning window
method %
%%%%%%%%%%
%%%%%%%%%

fc=1000;
fs=8000;
N=81;
wc=(fc*pi)/fs;
for n=1:40
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));;
    h(n)=hd(n)*w(n);
end
for n=41
    % Hipass Filter by Hanning window method %
    hd(n)=(wc/pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));;
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));;
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(2,1,2);

figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
N2 = (N+1)/2;
f = [0:N-1]*fs/N2;
X = fft(h);
figure(3),plot(f(1:N2),abs(X(1:N2)),'k')
xlabel('Frequency(Hz)');
ylabel('Gain(dB) : H(jw)');
grid on
%%%%%%%%%%
%%%%%%%%%

```

```

hd(n)=1-(wc/pi); %%%%%%%%%%%
w(n)=0.5-(0.5*cos((2*pi*n)/(N-1))); %%%%%%
h(n)=hd(n)*w(n); % Bandpass Filter by hanning window method
end %
for n=42:81 %%%%%%%%%%%
    hd(n)=-(sin(wc*(n-41)))/((n-41)*pi); %%%%%%
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
subplot(211);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(212);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
N2 = (N+1)/2;
f = [0:N-1]*fs/N2;
X = fft(h);
figure(3),plot(f(1:N2),abs(X(1:N2)),'k')
xlabel('Frequency(Hz)');
ylabel('Magnitude : H(jw)');
grid on

fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
for n=1:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wch-wcl)/pi;
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end

```

```

subplot(2,1,1);
figure(1),stem(hd);
xlabel('n');ylabel('Amplitude : hd(n)');
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
N2 = (N+1)/2;
f = [0:N-1]*fs/N2;
X = fft(h);
figure(3),plot(f(1:N2),abs(X(1:N2)),'k')
xlabel('Frequency(Hz)');
ylabel('Magnitude : H(jw)');
grid on

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% Bandstop Filter by hanning window
method %
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%

fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;

wch=(fch*pi)/fs;
for n=1:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=1+((wcl-wch)/pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-41)*pi);
    w(n)=0.5-(0.5*cos((2*pi*n)/(N-1)));
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)');grid on
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
N2 = (N+1)/2;
f = [0:N-1]*fs/N2;
X = fft(h);
figure(3),plot(f(1:N2),abs(X(1:N2)),'k')

```

```

xlabel('Frequency(Hz)');
ylabel('Magnitude : H(jw)');
grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lopass Filter by blackman window
method %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc=1000;
fs=8000;
N=81;
wc=(fc*pi)/fs;
w=blackman(N);
for n=1:40
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=(wc/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wc*(n-41)))/((n-41)*pi);
    h(n)=hd(n)*w(n);
end

subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(2,1,2);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]=freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Hipass Filter by blackman window method
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc=3000;
fs=8000;
N=81;
w=blackman(N);
wc=(fc*pi)/fs;
for n=1:40

```

```

hd(n)=-sin(wc*(n-41))/((n-41)*pi);          %%%%%%%%%%
h(n)=hd(n)*w(n);                            %%%%%%%%%%
end                                           % Bandpass Filter by blackman window method
for n=41                                     %
    hd(n)=1-(wc/pi);                        %%%%%%%%%%
    h(n)=hd(n)*w(n);                        %%%%%%%%%%
end
for n=42:81                                 fcl=2000;
    hd(n)=-sin(wc*(n-41))/((n-41)*pi);     fch=3500;
    h(n)=hd(n)*w(n);                        fs=8000;
end                                           N=81;
subplot(211);                               wcl=(fcl*pi)/fs;
figure(1),stem(hd)                          wch=(fch*pi)/fs;
xlabel('n');ylabel('Amplitude : hd(n)')      w=blackman(N);
subplot(212);                               for n=1:81
figure(1),stem(w)                           hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-
xlabel('n');ylabel('Amplitude : W(n)')      41)*pi);
figure(2),stem(h)                           h(n)=hd(n)*w(n);
xlabel('n');ylabel('Amplitude : h(n)')     end
[X,f]= freqz(h,1,512,2);                   for n=41
figure(3),plot(f,abs(X))                   hd(n)=(wch-wcl)/pi;
xlabel('Frequency(pi rad)');              h(n)=hd(n)*w(n);
ylabel('Magnitude : H(jw)');              end
grid on                                     for n=42:81
figure(4),freqz(h,1,512)                  hd(n)=(sin(wch*(n-41))-sin(wcl*(n-41)))/((n-
                                           41)*pi);
                                           h(n)=hd(n)*w(n);
                                           end
                                           subplot(2,1,1);
                                           figure(1),stem(hd);

```

```

xlabel('n');ylabel('Amplitude : hd(n)');
xlabel('n');ylabel('Amplitude : hd(n)')
subplot(212);
figure(1),stem(w)
xlabel('n');ylabel('Amplitude : W(n)')
figure(2),stem(h)
xlabel('n');ylabel('Amplitude : h(n)')
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bandstop Filter by blackman
window method %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fcl=2000;
fch=3500;
fs=8000;
N=81;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
w=blackman(N);

subplot(2,1,2);
for n=1:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-
41)*pi);
    h(n)=hd(n)*w(n);
end
for n=41
    hd(n)=1+((wcl-wch)/pi);
    h(n)=hd(n)*w(n);
end
for n=42:81
    hd(n)=(sin(wcl*(n-41))-sin(wch*(n-41)))/((n-
41)*pi);
    h(n)=hd(n)*w(n);
end
subplot(2,1,1);
figure(1),stem(hd)
xlabel('n');ylabel('Amplitude : hd(n)');grid on
subplot(2,1,2);
figure(1),stem(w);
xlabel('n');ylabel('Amplitude : W(n)');
figure(2),stem(h);
xlabel('n');ylabel('Amplitude : h(n)');
[X,f]= freqz(h,1,512,2);
figure(3),plot(f,abs(X))
xlabel('Frequency(pi rad)');
ylabel('Magnitude : H(jw)');
grid on
figure(4),freqz(h,1,512)

```

2. โปรแกรมการหาค่าสัมประสิทธิ์ของตัวกรอง FIR โดยภาษาซีและการแปลงจากเลขอิงค์ซ์นี้ให้เป็นจำนวนเต็มเพื่อนำไปประมวลผลบน TMS320C50

```

#include <stdio.h>
#include <math.h>
#include <conio.h>
void sub();
void coeff(double* ,double*,double*
,int n_start,int n_end);
double hd[200],w[200],h
[200],pi,NORM[200],j[200];
int n,N,key1;
int fcL,fcH,fch,fcl,fs,key,n_st,n_en,M;
float wcL,wcH,wcl,wch;
char g,gg;
double hd1[200],hd1c[200],w1[200];
main()
{
fcL=1000;/* fcutoff to lowpass*/
fcH=3000;/* fcutoff to hiwpass*/
fcl=2000;/* fcutoff Lower to bandpass
and bandstop*/
fch=3500;/* fcutoff Upper to bandpass
and bandstop*/
fs=8000;
N=81;
M=(N-1)/2;
pi=3.141592654;
wcl=(fcl*pi)/fs;
wch=(fch*pi)/fs;
wcL=(fcL*pi)/fs;
wcH=(fcH*pi)/fs;
do /*main loop*/
{
clrscr();
printf("\n\n\n\t\tPLEASE CHOOSE
WINDOW METHOD OF FIR\n\n");
printf("\t\t\t1.) RECTANGULAR\n");
printf("\t\t\t2.) HANNING\n");
printf("\t\t\t3.) HAMMING\n");
printf("\t\t\t4.) BLACKMANG\n");
printf("\t\t\t5.) TO EXIT_ :");
scanf("\t\t\t%d",&key);
clrscr();
switch(key) /*switch main*/
{
case 1: for(n=0;n<N;n++)
{
w1[n]=1;
}
}
} /*switch main*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







```

printf("\n\n\t\t* Lopass
Filter Fc = 1 KHz ,Fs = 8 KHz *\n\n\n
");
coeff(w1,hd1,hd1c,n_st,n_en);break;
case 2: for(n=0;n<N;n++)
{
hd1[n]= -sin(wcH*(n-
(M)))/((n-(M))*pi);
}
for(n=M;n<M+1;n++)
{
hd1c[n]=1-(wch/pi);
}clrscr();
printf("\n\n\t\t
HAMMING WINDOW ");
printf("\n\n\t\t* Hipass
Filter Fc = 3 KHz ,Fs = 8 KHz *\n\n\n
");
coeff(w1,hd1,hd1c,n_st,n_en);break;
case 3: for(n=0;n<N;n++)
{
hd1[n]= (sin(wch*(n-
(M)))-sin(wcl*(n-(M))))/((n-(M))*pi);
}
for(n=M;n<M+1;n++)
{
hd1c[n]=(wch-wcl)/pi;
}clrscr();
printf("\n\n\t\t* Lopass
Filter Fc = 1 KHz ,Fs = 8 KHz *\n\n\n
");
coeff(w1,hd1,hd1c,n_st,n_en);break;
case 4: for(n=0;n<N;n++)
{
hd1[n]=(sin(wcl*(n-(M)))-sin
(wch*(n-(M))))/((n-(M))*pi);
}
for(n=M;n<M+1;n++)
{
hd1c[n]=1+((wcl-wch)/pi);
}clrscr();
printf("\n\n\t\t
HAMMING WINDOW ");
printf("\n\n\t\t* Bandstop Filter
Fcl = 2 KHz ,Fch = 3.5 KHz ,Fs = 8 KHz
*\n\n\n
");
coeff(w1,hd1,hd1c,n_st,n_en);break;
}g=getch();
}while(g!='q');break; /*end switch of
hamming(case : 3)*/
case 4: /*switch main of Blackman */
for(n=0;n<N;n++)
{

```





```

{
    g=getch();
    printf("\t\t h[%d]=%f :: NORM_h
[%d]=%f \n ",n,h[n],n,NORM[n]);
} if(n_st>n_en)
{
    printf("\t\t ERROR!! number of
n_start morethan n_end\n");
    printf("\t\t PLEASE TO
AGAIN\n ");
    printf("\t\t PRESS 'Q' TO
QUIT_ : ");
}
else if((n_st<n_en)||(n_st=n_en))
{
    printf("\t\t PRESS ANYKEY TO
CONTINUOUS\n");
    printf("\t\t PRESS 'Q' TO
QUIT_ : ");
}
    printf("\t\t\t\t PLEASE CHOOSE FILTER
TYPE \n\n ");
    printf("\t\t\t\t 1.) LOWPASS\n ");
    printf("\t\t\t\t 2.) HIPASS\n ");
    printf("\t\t\t\t 3.) BANDPASS\n ");
    printf("\t\t\t\t 4.) BANDSTOP ");
    scanf(" %d",&key1);
    printf("\t\t\t\t n_start of h(n)_:");
    scanf(" %d",&n_st);
    printf("\t\t\t\t n_end of h(n)_:");
    scanf(" %d",&n_en);
}
}

```

### 3. โปรแกรมภาษาซีและแอสเซมบลีที่ใช้ประมวลผลบน TMS320C50

```

#include "stdlib.h"
#include "ioports.h"
#include "TABLE.h"
extern void aicinit();
void outLCD(unsigned b);
int *DRR = (int *) 0x20;
int *DXR = (int *) 0x21;
void filter(int *h,int *dly,int X,int n);
void delay(unsigned long timer);
void delay1(unsigned long timer);
void delay2(unsigned long timer);
void InitLCD();
void SL();

```

```

void SR();
void SUB(char*,char*);
int x,i,j,p,pp,y,m,s,T;
unsigned long k ;
static unsigned key ;
void main()
{
    key = 0;
    SUB(w1,w2);
    SL();
    SR();
    delay1(10);
    delay1(10);
    delay1(10);
    delay1(10);
    aicinit();
    for(;;);
}
void c_int5(void)
{
    x=*DRR>>2;
    k=_in_port(0) & 0x000f;
    if(k == 1)
    {
        SUB(m1,m2);
        delay1(10);
        delay1(10);
    }
    else if(k == 2)
    {
        key = key-1;
        if(key<1)
        {
            key = 1;
        }
        if(key == 1)
        {
            SUB(LPF1,LPF2);
        }
        else if(key == 2)
        {
            SUB(HPF1,HPF2);
        }
        else if(key == 3)
        {
            SUB(BPF1,BPF2);
        }
        else if(key == 4)
        {
            SUB(BSF1,BSF2);
        }
        else if(key == 5)
        {
            SUB(LPF01,LPF02);
        }
        else if(key == 6)
        {
            SUB(HPF01,HPF02);
        }
        else if(key == 7)
        {

```

```

SUB(BPF01,BPF02);
}
else if(key == 8)
{
SUB(BSF01,BSF02);
}
else if(key == 9)
{
SUB(LPF001,LPF002);
}
else if(key == 10)
{
SUB(HPF001,HPF002);
}
else if(key == 11)
{
SUB(BPF001,BPF002);
}
else if(key == 12)
{
SUB(BSF001,BSF002);
}
}
delay1(1);
delay1(1);
delay1(1);
delay1(1);
delay1(1);
}
else if(k == 3)
{
key = key+1;
if(key>12)
{
key = 12;
}
if(key == 1)
{
SUB(LPF1,LPF2);
}
else if(key == 2)
{
SUB(HPF1,HPF2);
}
else if(key == 3)
{
SUB(BPF1,BPF2);
}
else if(key == 4)
{
SUB(BSF1,BSF2);
}
else if(key == 5)
{
SUB(LPF01,LPF02);
}
else if(key == 6)
{
SUB(HPF01,HPF02);
}
else if(key == 7)

```

```

{
SUB(BPF01,BPF02);
}
else if(key == 8)
{
SUB(BSF01,BSF02);
}
else if(key == 9)
{
SUB(LPF001,LPF002);
}
else if(key == 0)
{
SUB(HPF001,HPF002);
}
else if(key == 11)
{
SUB(BPF001,BPF002);
}
else if(key == 12)
{
SUB(BSF001,BSF002);
}
delay1(1);
delay1(1);
delay1(1);
delay1(1);
delay1(1);
}
if(k == 4)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm(" SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR ,AR0");
asm("RPT #80");
asm("MACD #LoR,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0ffch");
asm("SAMM DXR");
}
else if(k == 5)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #HiR,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT ");
asm("SFL");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 6)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #BPR ,*-");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 7)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0, #INPUTL");
asm("ZAP");
asm("MAR ,AR0");
asm("RPT #80");
asm("MACD #BSR ,*-");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR ");
}
else if(k == 8)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #LoH ,*-");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR ");
}
else if(k == 9)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");

```

```

asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80 ");
asm("MACD #HiH ,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 0x0a)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #BPH ,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 0x0b)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #BSH ,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR ");
}
else if(k == 0x0c)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #LoB ,*- ");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 0x0d)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #HiB ,*-");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR ");
}
else if(k == 0x0e)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #BPB ,*-");
}
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR");
}
else if(k == 0x0f)
{
asm("LDP #INPUT");
asm("LAMM DRR");
asm("SACL INPUT");
asm("LAR AR0,#INPUTL");
asm("ZAP");
asm("MAR *,AR0");
asm("RPT #80");
asm("MACD #BSB ,*-");
asm("APAC");
asm("SACH OUTPUT,1");
asm("LACC OUTPUT");
asm("SFL");
asm("AND #0fffch");
asm("SAMM DXR");
}
}
void delay(unsigned long timer)
{
unsigned long t;
for(t = 0;t<timer;t++)
{

```



```

for(i=0;i<16;i++)
    }
    {
        _out_port(8,0x18);delay(1);
        delay1(3);
    }
}
void SR()
{
    for(i=0;i<16;i++)
    {
        _out_port(8,0x1c);delay(1);
        delay1(3);
    }
}
โปรแกรมแฮคเกอร์ไฟล์ TABLE.H
/* Table of MainMenu */
char w1[16]={' ',' ','W','E','L','C','O','M','E',' ','T','O',' ',' '};
char w2[16]={'F','T','L','T','E','R',' ','B','Y',' ','C','S','O','D','S','K'};
char m1[16]={' ',' ','M','A','T','N',' ','M','E','N','U',' ',' ',' '};
char m2[16]={'S','E','L','E','C','T',' ','<','=',' ','O','R',' ','>'};
/*****
/* Table of rectangular window */
char LPF1[16]={' ','1',' ','L','P','F',' ',' ','R','E','C','T','A','N','G',' '};
char LPF2[16]={' ','F','c','u','t','=',' ','1',' ','K','H','z',' ','>'};
char HPF1[16]={' ','2',' ','H','P','F',' ',' ','R','E','C','T','A','N','G',' '};
char HPF2[16]={'<','=',' ','F','c','u','t','=',' ','3','K','H','z',' ','>'};
char BPF1[17]={' ','3',' ','B','P','F',' ',' ','R','E','C','T','A','N','G',' '};
char BPF2[16]={'<','=','F','L','=','2','K',' ','F','H','=','3',' ','5','K','>'};
char BSF1[16]={' ','4',' ','B','S','F',' ',' ','R','E','C','T','A','N','G',' '};
char BSF2[16]={'<','=','F','L','=','2','K',' ','F','H','=','3',' ','5','K','>'};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

asm(" .word 0");asm(" .word -148");asm(" .word -317");asm(" .word -479");
asm(" .word -600");asm(" .word -642");asm(" .word -571");asm(" .word -361");
asm(" .word -1");asm(" .word 503");asm(" .word 1121");asm(" .word 1808");
asm(" .word 2503");asm(" .word 3139");asm(" .word 3650");asm(" .word 3981");
asm(" .word 4096");asm(" .word 3981");asm(" .word 3650");asm(" .word 3139");
asm(" .word 2503");asm(" .word 1808");asm(" .word 1121");asm(" .word 503");
asm(" .word -1");asm(" .word -361");asm(" .word -571");asm(" .word -642");
asm(" .word -600");asm(" .word -479");asm(" .word -317");asm(" .word -148");
asm(" .word 0");asm(" .word 109");asm(" .word 172");asm(" .word 192");
asm(" .word 177");asm(" .word 138");asm(" .word 89");asm(" .word 40");
asm(" .word -1");asm(" .word -28");asm(" .word -42");asm(" .word -44");
asm(" .word -38");asm(" .word -28");asm(" .word -17");asm(" .word -7");
asm(" .word 0");asm(" .word 3");asm(" .word 4");asm(" .word 4");
asm(" .word 2");asm(" .word 1");asm(" .word 0");asm(" .word 0");
asm(" .word 0");
/*****/
/* Hipass fc = 3 KHz */
asm("HiB .word -1");asm(" .word -1");asm(" .word -1");asm(" .word 0");
asm(" .word 2");asm(" .word 1");asm(" .word -5");asm(" .word -9");
asm(" .word -1");asm(" .word 16");asm(" .word 16");asm(" .word -12");
asm(" .word -38");asm(" .word -19");asm(" .word 41");asm(" .word 66");
asm(" .word 0");asm(" .word -98");asm(" .word -90");asm(" .word 57");
asm(" .word 177");asm(" .word 79");asm(" .word -173");asm(" .word -264");
asm(" .word -1");asm(" .word 356");asm(" .word 316");asm(" .word -199");
asm(" .word -600");asm(" .word -266");asm(" .word 570");asm(" .word 870");
asm(" .word 0");asm(" .word -1216");asm(" .word -1122");asm(" .word 749");
asm(" .word 2503");asm(" .word 1300");asm(" .word -3651");asm(" .word -9613");
asm(" .word 20479");asm(" .word -9613");asm(" .word -3651");asm(" .word 1300");
asm(" .word 2503");asm(" .word 749");asm(" .word -1122");asm(" .word -1216");
asm(" .word 0");asm(" .word 870");asm(" .word 570");asm(" .word -266");

```

```
asm(" .word -600");asm(" .word -199");asm(" .word 316");asm(" .word 356");
asm(" .word -1");asm(" .word -264");asm(" .word -173");asm(" .word 79");
asm(" .word 177");asm(" .word 57");asm(" .word -90");asm(" .word -98");
asm(" .word 0");asm(" .word 66");asm(" .word 41");asm(" .word -19");
asm(" .word -38");asm(" .word -12");asm(" .word 16");asm(" .word 16");
asm(" .word -1");asm(" .word -9");asm(" .word -5");asm(" .word 1");
asm(" .word 2");asm(" .word 0");asm(" .word -1");asm(" .word -1");
asm(" .word 1");
```

```
/******
```

```
/* Bandpass Fcl = 2 KHz , Fch = 3.5 KHz */
```

```
asm("BPB .word 0");asm(" .word 0");asm(" .word 1");asm(" .word 1");
asm(" .word -2");asm(" .word -7");asm(" .word -5");asm(" .word 2");
asm(" .word -1");asm(" .word -5");asm(" .word 14");asm(" .word 45");
asm(" .word 26");asm(" .word -60");asm(" .word -113");asm(" .word -37");
asm(" .word 87");asm(" .word 94");asm(" .word 9");asm(" .word 22");
asm(" .word 125");asm(" .word 25");asm(" .word -339");asm(" .word -483");
asm(" .word 0");asm(" .word 651");asm(" .word 618");asm(" .word -65");
asm(" .word -424");asm(" .word -106");asm(" .word -62");asm(" .word -850");
asm(" .word -1108");asm(" .word 673");asm(" .word 3051");asm(" .word 2472");
asm(" .word -1771");asm(" .word -5229");asm(" .word -3187");asm(" .word 2847");
asm(" .word 6143");asm(" .word -7357");asm(" .word -3187");asm(" .word -5229");
asm(" .word -1771");asm(" .word 2472");asm(" .word 3051");asm(" .word 673");
asm(" .word -1108");asm(" .word -850");asm(" .word -62");asm(" .word -109");
asm(" .word -423");asm(" .word -65");asm(" .word 618");asm(" .word 651");
asm(" .word 0");asm(" .word -483");asm(" .word -339");asm(" .word 25");
asm(" .word 128");asm(" .word 22");asm(" .word 9");asm(" .word 94");
asm(" .word 87");asm(" .word -37");asm(" .word -113");asm(" .word -60");
asm(" .word 26");asm(" .word 45");asm(" .word 14");asm(" .word -5");
asm(" .word -1");asm(" .word 2");asm(" .word -5");asm(" .word -7");
asm(" .word -2");asm(" .word 1");asm(" .word 1");asm(" .word 0");
```

```

asm(" .word 0");
/*****/
/*Bandstop Fcl = 2 KHz , Fch = 3.5 KHz */
asm("BSB .word -1");asm(" .word -1");asm(" .word -2");asm(" .word -2");
asm(" .word 1");asm(" .word 6");asm(" .word 4");asm(" .word -3");
asm(" .word 0");asm(" .word 4");asm(" .word -15");asm(" .word -46");
asm(" .word -27");asm(" .word 59");asm(" .word 112");asm(" .word 36");
asm(" .word -88");asm(" .word -95");asm(" .word -10");asm(" .word -23");
asm(" .word -126");asm(" .word -26");asm(" .word 338");asm(" .word 482");
asm(" .word -1");asm(" .word -652");asm(" .word -619");asm(" .word 64");
asm(" .word 423");asm(" .word 105");asm(" .word 61");asm(" .word 849");
asm(" .word 1107");asm(" .word -674");asm(" .word -3052");asm(" .word -2473");
asm(" .word 1770");asm(" .word 5228");asm(" .word 3186");asm(" .word -2848");
asm(" .word 26624");asm(" .word -2848");asm(" .word 3186");asm(" .word 5228");
asm(" .word 1770");asm(" .word -2473");asm(" .word -3052");asm(" .word -674");
asm(" .word 1107");asm(" .word 849");asm(" .word 61");asm(" .word 105");
asm(" .word 423");asm(" .word 64");asm(" .word -619");asm(" .word -652");
asm(" .word -1");asm(" .word 482");asm(" .word 338");asm(" .word -26");
asm(" .word -126");asm(" .word -23");asm(" .word -10");asm(" .word -95");
asm(" .word -88");asm(" .word 36");asm(" .word 112");asm(" .word 59");
asm(" .word -27");asm(" .word -46");asm(" .word -15");asm(" .word 4");
asm(" .word 0");asm(" .word -3");asm(" .word 4");asm(" .word 6");
asm(" .word 1");asm(" .word -2");asm(" .word -2");asm(" .word -1");
asm(" .word -1");
/*****/
/*          Coefficient of Hamming window          */
/*****/
/*Coefficient of Lowpass */
asm("LoH .word -1");asm(" .word 8");asm(" .word 17");asm(" .word 26");
asm(" .word 33");asm(" .word 35");asm(" .word 32");asm(" .word 20");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

asm(" .word 0 ");asm(" .word -28 ");asm(" .word -60 ");asm(" .word -90");  
asm(" .word -112 ");asm(" .word -119 ");asm(" .word -104 ");asm(" .word -64");  
asm(" .word -1 ");asm(" .word 81 ");asm(" .word 168 ");asm(" .word 247");  
asm(" .word 300 ");asm(" .word 310 ");asm(" .word 265 ");asm(" .word 160");  
asm(" .word 0 ");asm(" .word -200 ");asm(" .word -412 ");asm(" .word -601");  
asm(" .word -730 ");asm(" .word -759 ");asm(" .word -657 ");asm(" .word -405");  
asm(" .word -1 ");asm(" .word 541 ");asm(" .word 1186 ");asm(" .word 1883");  
asm(" .word 2574 ");asm(" .word 3193 ");asm(" .word 3682 ");asm(" .word 0");  
asm(" .word 4090 ");asm(" .word 3968 ");asm(" .word 3640 ");asm(" .word 3139");  
asm(" .word 2516 ");asm(" .word 1830 ");asm(" .word 1145 ");asm(" .word 520");  
asm(" .word -1 ");asm(" .word -384 ");asm(" .word -619 ");asm(" .word -710");  
asm(" .word -679 ");asm(" .word -556 ");asm(" .word -378 ");asm(" .word -182");  
asm(" .word 0 ");asm(" .word 143 ");asm(" .word 236 ");asm(" .word 273");  
asm(" .word 262 ");asm(" .word 214 ");asm(" .word 145 ");asm(" .word 69");  
asm(" .word -1 ");asm(" .word -53 ");asm(" .word -86 ");asm(" .word -97");  
asm(" .word -90 ");asm(" .word -72 ");asm(" .word -47 ");asm(" .word -22");  
asm(" .word 0 ");asm(" .word 15 ");asm(" .word 24 ");asm(" .word 28");  
asm(" .word 26 ");asm(" .word 22 ");asm(" .word 15 ");asm(" .word 8");  
asm(" .word -1 ");  
/\*\*\*\*\*/  
/\*Coefficient of Hipass \*/  
asm("HiH .word 0");asm(" .word -21");asm(" .word -17");asm(" .word 10");  
asm(" .word 29");asm(" .word 13");asm(" .word -29");asm(" .word -44");  
asm(" .word -1");asm(" .word 59");asm(" .word 52");asm(" .word -34");  
asm(" .word -101");asm(" .word -45");asm(" .word 93");asm(" .word 140");  
asm(" .word 0");asm(" .word -182");asm(" .word -157");asm(" .word 95");  
asm(" .word 281");asm(" .word 121");asm(" .word -251");asm(" .word -367");  
asm(" .word -1");asm(" .word 460");asm(" .word 394");asm(" .word -240");  
asm(" .word -705");asm(" .word -305");asm(" .word 638");asm(" .word 952");  
asm(" .word 0");asm(" .word -1284");asm(" .word -1168");asm(" .word 770");

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

asm(" .word 2548");asm(" .word 1313");asm(" .word -3667");asm(" .word -9623");
asm(" .word 20479");asm(" .word -9623");asm(" .word -3667");asm(" .word 1313");
asm(" .word 2548");asm(" .word 770");asm(" .word -1168");asm(" .word -1284");
asm(" .word 0");asm(" .word 952");asm(" .word 638");asm(" .word -305");
asm(" .word -705");asm(" .word -240");asm(" .word 394");asm(" .word 460");
asm(" .word -1");asm(" .word -367");asm(" .word -251");asm(" .word 121");
asm(" .word 281");asm(" .word 95");asm(" .word -157");asm(" .word -182");
asm(" .word 0");asm(" .word 140");asm(" .word 93");asm(" .word -45");
asm(" .word -101");asm(" .word -34");asm(" .word 52");asm(" .word 59");
asm(" .word -1");asm(" .word -44");asm(" .word -29");asm(" .word 13");
asm(" .word 29");asm(" .word 10");asm(" .word -17");asm(" .word -21");
asm(" .word 0");
/*****/
/*Coefficient of Bandpass */
asm("BPH .word -22");asm(" .word 11");asm(" .word 48");asm(" .word 36");
asm(" .word -24");asm(" .word -60");asm(" .word -28");asm(" .word 14");
asm(" .word -1");asm(" .word -20");asm(" .word 51");asm(" .word 149");
asm(" .word 78");asm(" .word -162");asm(" .word -281");asm(" .word -85");
asm(" .word 188");asm(" .word 191");asm(" .word 18");asm(" .word 40");
asm(" .word 212");asm(" .word 41");asm(" .word -519");asm(" .word -707");
asm(" .word 0");asm(" .word 878");asm(" .word 803");asm(" .word -81");
asm(" .word -516");asm(" .word -125");asm(" .word -71");asm(" .word -954");
asm(" .word -1216");asm(" .word 724");asm(" .word 3227");asm(" .word 2574");
asm(" .word -1821");asm(" .word -5320");asm(" .word -3215");asm(" .word 0");
asm(" .word 6135");asm(" .word -7334");asm(" .word -3179");asm(" .word -5229");
asm(" .word -1780");asm(" .word 2501");asm(" .word 3117");asm(" .word 695");
asm(" .word -1161");asm(" .word -905");asm(" .word -67");asm(" .word -117");
asm(" .word -480");asm(" .word -75");asm(" .word 737");asm(" .word 800");
asm(" .word 0");asm(" .word -634");asm(" .word -462");asm(" .word 36");
asm(" .word 185");asm(" .word 35");asm(" .word 15");asm(" .word 162");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
asm(" .word 158");asm(" .word -71");asm(" .word -232");asm(" .word -132");
asm(" .word 63");asm(" .word 118");asm(" .word 40");asm(" .word -16");
asm(" .word -1");asm(" .word 11");asm(" .word -22");asm(" .word -48");
asm(" .word -19");asm(" .word 30");asm(" .word 42");asm(" .word 10");
asm(" .word -22");
```

```
/******
```

```
/*Coefficient of Bandstop*/
```

```
asm("BSH .word 21");asm(" .word -12");asm(" .word -49");asm(" .word -37");
asm(" .word 23");asm(" .word 59");asm(" .word 27");asm(" .word -15");
asm(" .word 0");asm(" .word 19");asm(" .word -52");asm(" .word -150");
asm(" .word -79");asm(" .word 161");asm(" .word 280");asm(" .word 84");
asm(" .word -189");asm(" .word -192");asm(" .word -19");asm(" .word -41");
asm(" .word -213");asm(" .word -42");asm(" .word 518");asm(" .word 706");
asm(" .word -1");asm(" .word -879");asm(" .word -804");asm(" .word 80");
asm(" .word 515");asm(" .word -124");asm(" .word 70");asm(" .word 953");
asm(" .word 1215");asm(" .word -725");asm(" .word -3228");asm(" .word -2575");
asm(" .word 1820");asm(" .word 5319");asm(" .word 2314");asm(" .word 0");
asm(" .word 26586");asm(" .word -10173");asm(" .word 3178");asm(" .word 5228");
asm(" .word 1779");asm(" .word -2502");asm(" .word -3118");asm(" .word -695");
asm(" .word 1160");asm(" .word 904");asm(" .word 66");asm(" .word 116");
asm(" .word 479");asm(" .word 74");asm(" .word -738");asm(" .word -801");
asm(" .word -1");asm(" .word 633");asm(" .word 461");asm(" .word -37");
asm(" .word -186");asm(" .word -36");asm(" .word -16");asm(" .word -163");
asm(" .word -159");asm(" .word 70");asm(" .word 231");asm(" .word 131");
asm(" .word -64");asm(" .word -119");asm(" .word -41");asm(" .word 15");
asm(" .word 0");asm(" .word -12");asm(" .word 21");asm(" .word 47");
asm(" .word 18");asm(" .word -31");asm(" .word -43");asm(" .word -11");
asm(" .word 21");
```

```

/*****/

/*      Coefficient of Rectangular window      */

/*****/

/*Coefficient of Lowpass */

asm("LoR .word -1");asm(" .word 102 ");asm(" .word 194 ");asm(" .word 260");
asm(" .word 289 ");asm(" .word 275 ");asm(" .word 216 ");asm(" .word 120");
asm(" .word 0 ");asm(" .word -129 ");asm(" .word -246 ");asm(" .word -333");
asm(" .word -373 ");asm(" .word -357 ");asm(" .word -284 ");asm(" .word -160");
asm(" .word -1 ");asm(" .word 173 ");asm(" .word 335 ");asm(" .word 458");
asm(" .word 521 ");asm(" .word 507 ");asm(" .word 409 ");asm(" .word 234");
asm(" .word 0 ");asm(" .word -267 ");asm(" .word -527 ");asm(" .word -742");
asm(" .word -870 ");asm(" .word -877 ");asm(" .word -738 ");asm(" .word -444");
asm(" .word -1 ");asm(" .word 570 ");asm(" .word 1229 ");asm(" .word 1927");
asm(" .word 2607 ");asm(" .word 3212 ");asm(" .word 3687 ");asm(" .word 3991");
asm(" .word 4096 ");asm(" .word 3991 ");asm(" .word 3687 ");asm(" .word 3212");
asm(" .word 2607 ");asm(" .word 1927 ");asm(" .word 1229 ");asm(" .word 570");
asm(" .word -1 ");asm(" .word -444 ");asm(" .word -738 ");asm(" .word -877");
asm(" .word -870 ");asm(" .word -742 ");asm(" .word -527 ");asm(" .word -267");
asm(" .word 0 ");asm(" .word 234 ");asm(" .word 409 ");asm(" .word 507");
asm(" .word 521 ");asm(" .word 458 ");asm(" .word 335 ");asm(" .word 173");
asm(" .word -1 ");asm(" .word -160 ");asm(" .word -280 ");asm(" .word -357");
asm(" .word -373 ");asm(" .word -333 ");asm(" .word -246 ");asm(" .word -129");
asm(" .word 0 ");asm(" .word 120 ");asm(" .word 216 ");asm(" .word 275");
asm(" .word 289 ");asm(" .word 260 ");asm(" .word 194 ");asm(" .word 102");
asm(" .word -1 ");

/*****/

/*Coefficient of Hipass */

asm("HiR .word 0");asm(" .word -248");asm(" .word -195 ");asm(" .word 107");
asm(" .word 289");asm(" .word 114");asm(" .word -217 ");asm(" .word -293");
asm(" .word -1");asm(" .word 310");asm(" .word 245");asm(" .word -138");

```

```

asm(" .word -373");asm(" .word -148");asm(" .word 283 ");asm(" .word 385");
asm(" .word 0");asm(" .word -419");asm(" .word -336 ");asm(" .word 190");
asm(" .word 512");asm(" .word 210");asm(" .word -410");asm(" .word -567");
asm(" .word -1");asm(" .word 642");asm(" .word 526");asm(" .word -308");
asm(" .word -870");asm(" .word -363");asm(" .word 737");asm(" .word 1070");
asm(" .word 0");asm(" .word -1377");asm(" .word -1230");asm(" .word 798");
asm(" .word 2607");asm(" .word 1330");asm(" .word -3688");asm(" .word 0");
asm(" .word 20479");asm(" .word 0");asm(" .word -3688");asm(" .word 1330");
asm(" .word 2607");asm(" .word 798");asm(" .word -1230");asm(" .word -1377");
asm(" .word 0");asm(" .word 1070");asm(" .word 737");asm(" .word -363");
asm(" .word -870");asm(" .word -308");asm(" .word 526");asm(" .word 642");
asm(" .word -1");asm(" .word -567");asm(" .word -410");asm(" .word 210");
asm(" .word 521");asm(" .word 190");asm(" .word -336");asm(" .word -419");
asm(" .word 0");asm(" .word 385");asm(" .word 283");asm(" .word -148");
asm(" .word -373");asm(" .word -138");asm(" .word 245");asm(" .word 310");
asm(" .word -1");asm(" .word -293");asm(" .word -217");asm(" .word 114");
asm(" .word 289");asm(" .word 107");asm(" .word -197");asm(" .word -248");
asm(" .word 0");
/*****/
/*Coefficient of Bandpass*/
asm("BPR .word -261");asm(" .word 136");asm(" .word 528 ");asm(" .word 355 ");
asm(" .word -205");asm(" .word -459");asm(" .word -190 ");asm(" .word 86 ");
asm(" .word -1");asm(" .word -93");asm(" .word 214 ");asm(" .word 553");
asm(" .word 263");asm(" .word -488");asm(" .word -772 ");asm(" .word -214");
asm(" .word 434");asm(" .word 409");asm(" .word 36 ");asm(" .word 75");
asm(" .word 368");asm(" .word 68");asm(" .word -802 ");asm(" .word -1036");
asm(" .word 0");asm(" .word 1173");asm(" .word 1030");asm(" .word -100");
asm(" .word -615");asm(" .word -144");asm(" .word -80 ");asm(" .word -1046");
asm(" .word -1304");asm(" .word 762");asm(" .word 3344");asm(" .word 2634");
asm(" .word -1844");asm(" .word -5350");asm(" .word -3220");asm(" .word 0");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
asm(" .word 6143");asm(" .word -7376");asm(" .word -3220");asm(" .word -5350");
asm(" .word -1844");asm(" .word 2634");asm(" .word 3344");asm(" .word 762 ");
asm(" .word -1304");asm(" .word -1046");asm(" .word -80");asm(" .word -144");
asm(" .word -615");asm(" .word -100");asm(" .word 1030");asm(" .word 1173");
asm(" .word 0");asm(" .word -1036");asm(" .word -802");asm(" .word 68");
asm(" .word 368");asm(" .word 75");asm(" .word 36");asm(" .word 409");
asm(" .word 434");asm(" .word -214");asm(" .word -772");asm(" .word -488");
asm(" .word 263");asm(" .word 553");asm(" .word 214");asm(" .word -93");
asm(" .word -1");asm(" .word 86");asm(" .word -190");asm(" .word -459");
asm(" .word -205");asm(" .word 355");asm(" .word 528");asm(" .word 136");
asm(" .word -261");
/*****/
/*Coefficient of Bandstop*/
asm("BSR .word 260");asm(" .word -137");asm(" .word -529");asm(" .word -365");
asm(" .word 204");asm(" .word 458");asm(" .word 189");asm(" .word -87");
asm(" .word 0");asm(" .word 92");asm(" .word -215");asm(" .word -554");
asm(" .word -264");asm(" .word 487");asm(" .word 771");asm(" .word 213");
asm(" .word -435");asm(" .word -410");asm(" .word -37");asm(" .word -76");
asm(" .word -369");asm(" .word -69");asm(" .word 801");asm(" .word 1035");
asm(" .word -1");asm(" .word -1174");asm(" .word -1031");asm(" .word 99");
asm(" .word 614");asm(" .word 143");asm(" .word 79");asm(" .word 1045");
asm(" .word 1303");asm(" .word -763");asm(" .word -3345");asm(" .word -2635");
asm(" .word 1843");asm(" .word 5349");asm(" .word 2319");asm(" .word 0");
asm(" .word 26624");asm(" .word -10230");asm(" .word 3219");asm(" .word 5349");
asm(" .word 1843");asm(" .word -2635");asm(" .word -3345");asm(" .word -763");
asm(" .word 1303");asm(" .word 1045");asm(" .word 79");asm(" .word 143");
asm(" .word 614");asm(" .word 99");asm(" .word -1031");asm(" .word -1174");
asm(" .word -1");asm(" .word 1035");asm(" .word 801");asm(" .word -69");
asm(" .word -369");asm(" .word -76");asm(" .word -37");asm(" .word -410");
asm(" .word -435");asm(" .word 213");asm(" .word 771");asm(" .word 487");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
asm(" .word -264");asm(" .word -554");asm(" .word -215");asm(" .word 92");
asm(" .word 0");asm(" .word -87");asm(" .word 189");asm(" .word 458");
asm(" .word 204");asm(" .word -356");asm(" .word -529");asm(" .word -137");
asm(" .word 260");
```

```
asm("INPUT .word 0,0,0,0,0,0,0,0,0,0 ");asm(" .word 0,0,0,0,0,0,0,0,0,0 ");
asm(" .word 0,0,0,0,0,0,0,0,0,0 ");asm(" .word 0,0,0,0,0,0,0,0,0,0 ");
asm(" .word 0,0,0,0,0,0,0,0,0,0 ");asm(" .word 0,0,0,0,0,0,0,0,0,0 ");
asm(" .word 0,0,0,0,0,0,0,0,0,0 ");asm(" .word 0,0,0,0,0,0,0,0,0,0 ");
asm("INPUTL .word 0");asm("OUTPUT .word 0");
```

โปรแกรม AICINIT.ASM

```
.file "aicinit.asm"
.globl _aicinit
.mmregs
_aicinit:
    POPD *+
    SAR AR0,*+
    SAR AR1,*
    LARK AR0,1
    LAR AR0,*0+,AR2
    TA .word 16
    RA .word 16
    TB .word 31
    RB .word 31
    AIC_CTRL .word 019h
    SETC INTM
    LDP #0
    OPL #0834h,PMST
    LACC #0
    SAMM CWSR
    SAMM PDWSR
    SETC SXM
    SPLK #022h,IMR
    SPLK #20h,TCR
    SPLK #01h,PRD
    MAR *,AR0
    LACC #0008h
    SACL SPC
    LACC #00c8h
    SACL SPC
```

```

LACC #080h
SACH DXR
SACL GREG
LAR AR0,#0FFFFh
RPT #10000
LACC *,0,AR0
SACH GREG
;-----
LDP #TA
SETC SXM
LACC TA,9
ADD RA,2
CALL AIC_2ND
;-----
LDP #TB
LACC TB,9
ADD RB,2
ADD #02h
CALL AIC_2ND
;-----
LDP #AIC_CTR
LACC AIC_CTR,2
ADD #03h
CALL AIC_2ND

SPLK #12h,IMR
CLRC OVM
SPM 0
CLRC INTM
LARP AR1
SBRK 2
LAR AR0,*-
PSHD *
RET
AIC_2ND:
LDP #0
SACH DXR
CLRC INTM
IDLE ADD #6h,15
SACH DXR
IDLE
SACL DXR
IDLE
LACL #0
SACL DXR
IDLE
SETC INTM
RET
.END

```

โปรแกรมการจองหน่วยความจำและการกำหนดอินเตอร์รัพท์เวคเตอร์

โปรแกรม **DSK.CMD**

-stack 64

MEMORY

{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAGE 0: PROG: origin=0x1c00, length = 0x1000

PAGE 0: RVEC: origin=0x080a, length = 0x0010

PAGE 1: DATA: origin=0x0c00, length = 0x1000

}

## SECTIONS

{

vectors: > RVEC PAGE 0

.text > PROG PAGE 0

.cinit > PROG PAGE 0

.data > DATA PAGE 1

.bss > DATA PAGE 1

.stack > DATA PAGE 1

}

โปรแกรม VEC.SASM

.title "vecs.asm"

.def IVECS

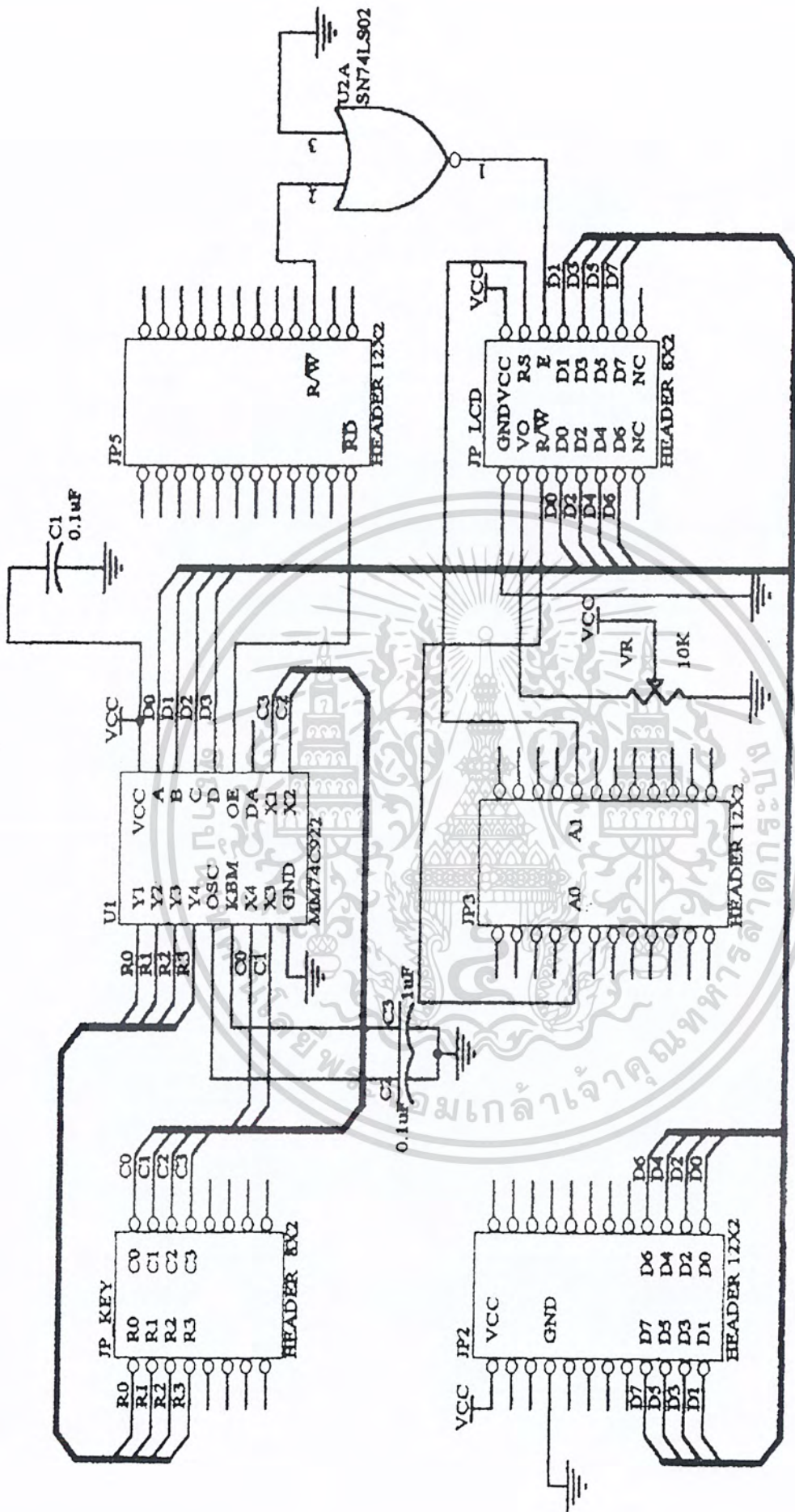
.ref \_c\_int5

.sect "vectors"

IVECS b \_c\_int5

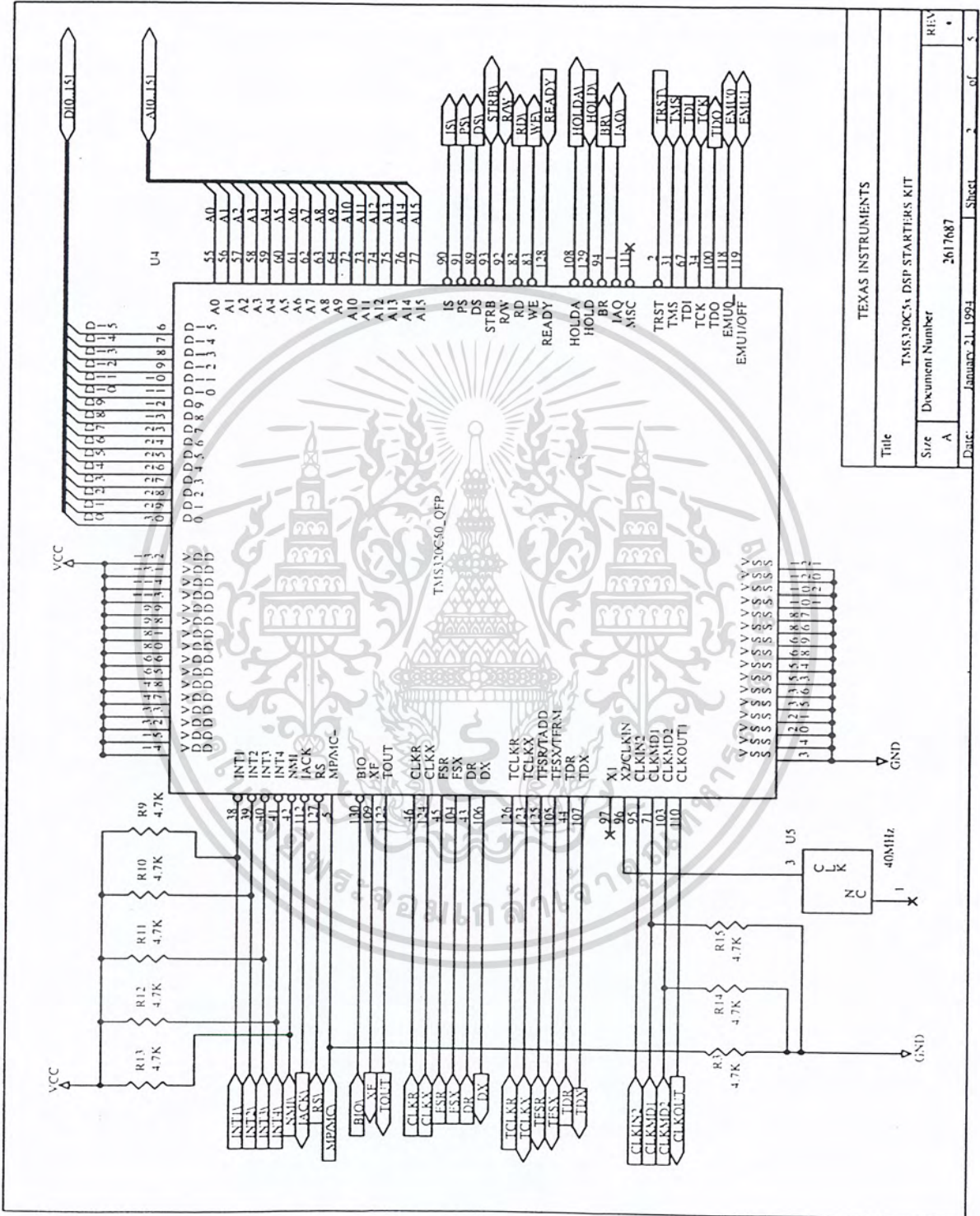


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



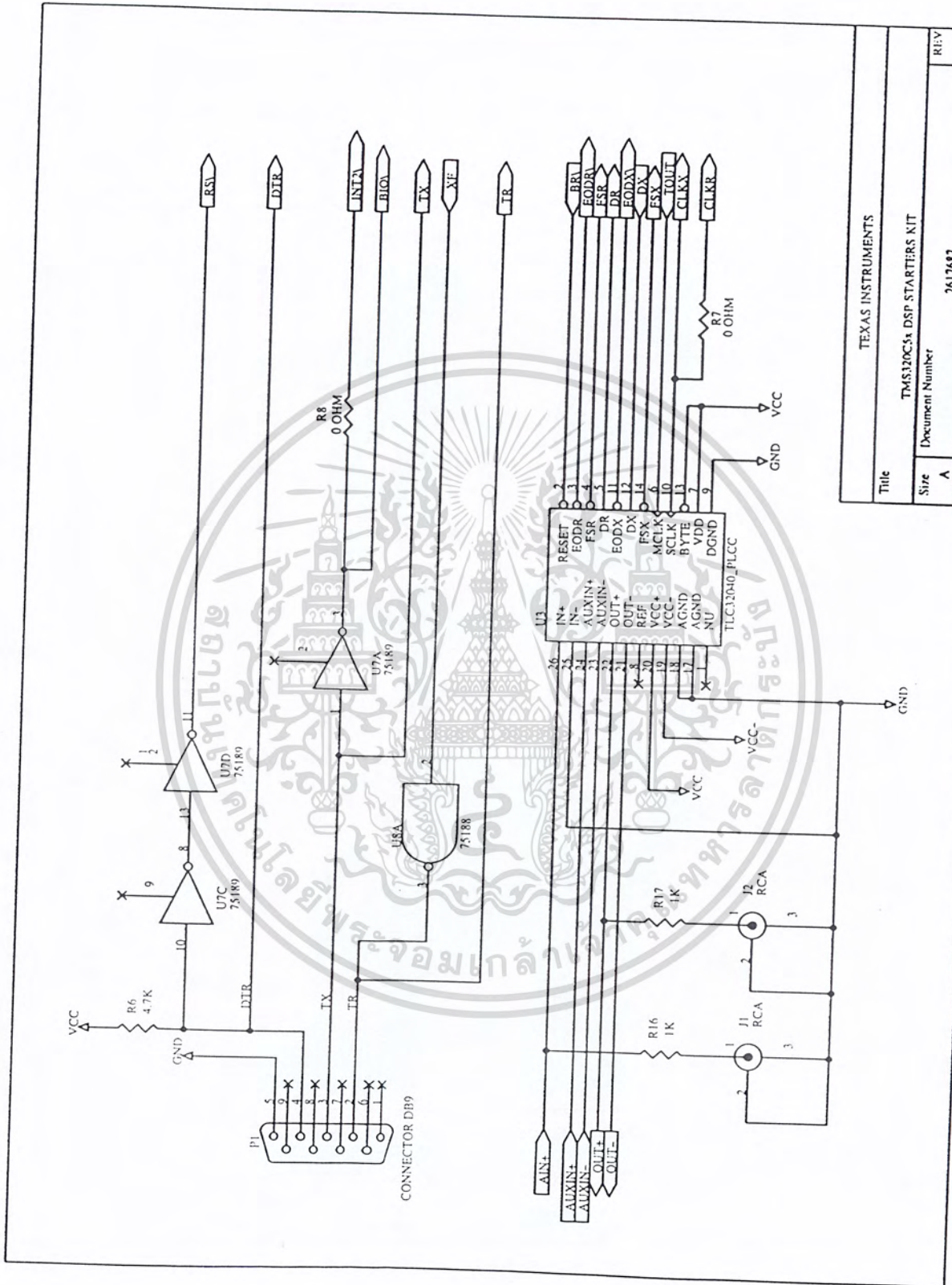
แสดงวงจรในโครงการ ชุดทดลองตัวกรองเชิงเลขแบบ เอฟ ไอ อาร์ ด้วย DSP TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



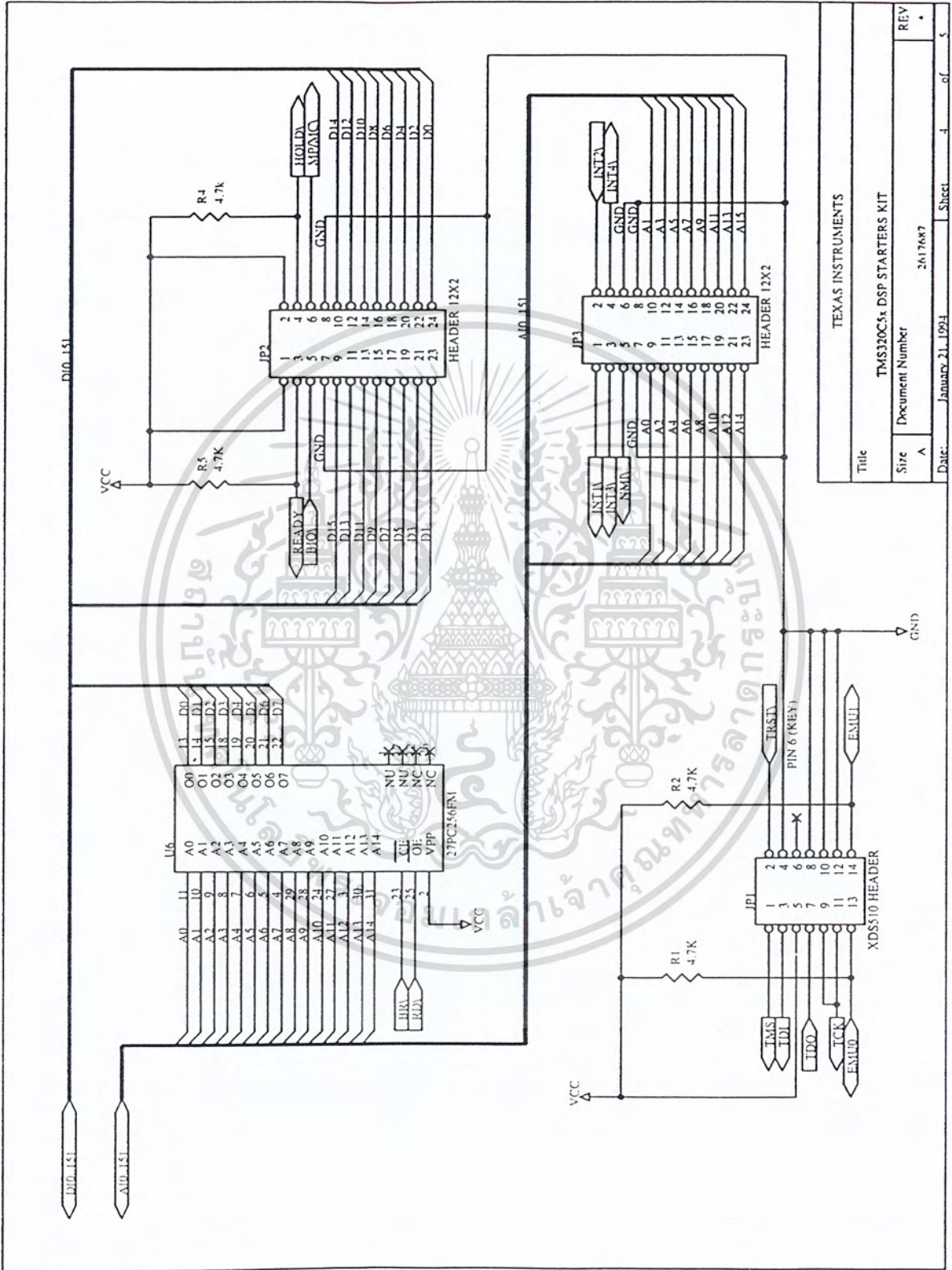
Title		TEXAS INSTRUMENTS	
Size		A	
Document Number		TMS320C55 DSP STARTERS KIT	
Date		January 21, 1994	
Sheet		2 of 5	
REV		2617687	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		TEXAS INSTRUMENTS	
Size		TMS320C5x DSP STARTERS KIT	
Document Number		2617687	
Date:	January 17, 1994	Sheet	3 of 5
REV	A		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		TEXAS INSTRUMENTS	
Size		TMS320C5x DSP STARTERS KIT	
Document Number		2617687	
Date:	January 21, 1994	Sheet	4 of 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

