



ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

ชื่อหัวข้อ การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ตโดย DSP TMS320C31
Speech Compression with Wavelet Transform by DSP TMS320C31

ชื่อนักศึกษา 1. นางสาวจริลักษณ์ ประสิทธิ์มณีรัตน์ รหัสประจำตัว 42035290
2. นายบรรยงค์ ค้างนุ่ม รหัสประจำตัว 42035306

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์อำพล ทองระอา

อาจารย์ที่ปรึกษาร่วม อาจารย์วรวิทย์ สมหา

คณะกรรมการสอบปริญญาโท		ลายมือชื่อ
1. อาจารย์อำพล	ทองระอา	
2. อาจารย์วรวิทย์	สมหา	
3. อาจารย์สุชิน	อาจหาญ	
4. อาจารย์กิติพงศ์	มะโน	
5. อาจารย์ไพบุลย์	พวงวงศ์ตระกูล	

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 25 พฤศจิกายน พ.ศ. 2543 เวลา 15.00 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 3 เดือน 11 พ.ศ. 2543



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ตโดย DSP TMS320C31

SPEECH COMPRESSION WITH WAVELET TRANSFORM BY DSP

TMS320C31



นางสาวจรีลักษณ์ ประสิทธิ์มณีรัตน์
นายบรรยงค์ คิ้วงุ่ม

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาคครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขหมึก.....

เลขทะเบียน..... 40157

วัน, เดือน, ปี 17 ส.ค. 2544

b..... 11092130
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
Speech Compression with Wevelet Transform by DSP TMS320C31

วัตถุประสงค์

1. เพื่อศึกษากระบวนการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยการใช้ DSP TMS320C31
2. เพื่อออกแบบวงจรลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
3. เพื่อสร้างวงจรลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
4. เพื่อนำระบบการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31 ไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. มีความเข้าใจหลักการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยการใช้ DSP TMS320C31 ได้
2. ได้วงจรลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
3. ได้ระบบการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
4. ได้เครื่องบันทึกเสียงที่มีการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31 ไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดย DSP TMS320C31
นักศึกษา	นางสาวจริลักษณ์ ประสิทธิ์มณีรัตน์ นายบรรยงก์ ค้วงนุ้ม
อาจารย์ที่ปรึกษา	อาจารย์อำพล ทองระอา
อาจารย์ที่ปรึกษาร่วม	อาจารย์รววิทย์ สมหา
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการบีบอัดข้อมูลเสียงพูดด้วยเวฟเล็ต โดย DSP TMS320C31 ในการลดขนาดข้อมูลเสียงพูด โดยใช้วิธีการแปลงเวฟเล็ต กระบวนการการแปลงเวฟเล็ตกระทำให้เกิดผลลัพธ์ขึ้นมา 2 ส่วนคือค่าสัมประสิทธิ์ด้านความถี่ต่ำและค่าสัมประสิทธิ์ด้านความถี่สูง โดยทั่วไปค่าสัมประสิทธิ์ด้านความถี่สูงมีความสำคัญน้อยกว่าสัมประสิทธิ์ด้านความถี่ต่ำ จึงสามารถละทิ้งได้ ซึ่งเรียกขบวนการนี้ว่า การลดขนาดข้อมูลเสียงพูด แล้วจึงทำการแปลงกลับเวฟเล็ต ผลที่ได้คือซึ่งเรียกว่าการเล่นกลับเสียง ในปริญญานิพนธ์ฉบับนี้ได้ใช้วิธี Multi Resolution Wavelet ในระดับหนึ่ง ทำให้เกิดการลดข้อมูลเสียงได้ร้อยละ 50 และจากผลการทดลองสามารถรับฟังเสียงพูดได้ ซึ่งอาจนำผลที่ได้ไปใช้ในเครื่องบันทึกสัญญาณเสียงในอนาคต

Thesis	Speech Compression with Wevelet Transform by DSP TMS320C31
Students	Miss.Jarecrak Prasitmaneerut Mr.Bungong Duangnoom
Advisor	Mr.Amphon Thongra-ar
Co-Advisor	Mr.Worawit Somha
Education level	Bachelor of Science in Industrial Education
Program in	Electronics and computer
Academic year	2000

ABSTRACT

This thesis presents Speech Compression with Wavelet Transform by DSP TMS320C31. Algorithm of Wavelet can two section is Wavelet Transform and Inverse Wavelet Transform. It is used to speech compression. The result of Wavelet Transform is approximations coefficients and details coefficients. The details coefficients is leave because it is important less than approximations coefficients, this process called data compression. Then take the result of data compression to Inverse Wavelet Transform, the result is reduced speech size. This thesis uses multi resolution Level 1, So speech size is reduced 50 percent but it can heard be speech sound.

กิตติกรรมประกาศ

ปริญาานิพนธ์ฉบับนี้ถูกล่วงไปด้วยดี เนื่องจากความร่วมมือของสมาชิกภายในกลุ่มและการสนับสนุนทางด้านทุนทรัพย์จากทางภาควิชาครุศาสตร์วิศวกรรม ของขอพระคุณ อาจารย์ อัมพล ทองระอาและคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความรู้ คำแนะนำ และแนวทางการแก้ปัญหา ในการจัดทำปริญาานิพนธ์ สุดท้ายสิ่งที่ระลึกอยู่เสมอคือ บิดา มารดา และ ครู-อาจารย์ ตั้งแต่อดีตจนถึงปัจจุบัน ที่เป็นผู้ให้การสนับสนุนด้านการศึกษา และเป็นผู้ให้กำลังใจด้วยดีตลอดมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	V
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ชี้ดความสามารถ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	3
2.1 กล่าวนำการแปลงเวฟเล็ด	3
2.2 พื้นฐานของทฤษฎีเวฟเล็ด	6
2.2.1 ตัวแปลงเวฟเล็ดและคำจำกัดความ	9
2.2.2 การแปลงกลับเวฟเล็ด	12
2.3 การกระจายและรวบรวมข้อมูลด้วยเวฟเล็ด	13
2.3.1 การแปลงเวฟเล็ด	13
2.3.2 การแปลงกลับเวฟเล็ด	17
2.4 วิธีการที่มีประสิทธิภาพในการแปลงเวฟเล็ด	19
2.5 ตัวประมวลผลสัญญาณเชิงเลข	23
2.5.1 สถาปัตยกรรมและหน่วยความจำ	23
2.5.2 รีจิสเตอร์	25
2.5.3 การติดต่ออินพุตและเอาต์พุตของ DSK	27
2.5.4 AIC (Analog Interface Circuit)	27
2.5.5 การติดต่อระหว่าง PC กับ TMS320C31	30
2.5.6 ซอฟต์แวร์ Development Tools	34
2.5.7 Compiling C Code	35

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	37
3.1 ส่วนของซอฟต์แวร์	38
3.1.1 โปรแกรม DWT	39
3.1.2 โปรแกรม IDWT	38
3.1.3 โปรแกรมการลดขนาดข้อมูลเสียงพูด	39
3.1.4 โปรแกรมควบคุมการทำงานบอร์ด DSK	42
3.2 ส่วนของฮาร์ดแวร์	45
3.2.1 หลักการทำงานและคุณสมบัติ PRE AMP	45
3.2.2 การต่อหน่วยความจำภายนอก	45
3.2.3 วงจรควบคุมการทำงาน	47
บทที่ 4 การทดลอง และผลการทดลอง	49
4.1 การทดลองด้านของซอฟต์แวร์	49
4.1.1 โปรแกรมนำข้อมูลอินพุตสู่เอาต์พุต	49
4.1.2 โปรแกรมนำข้อมูลเก็บในหน่วยความจำภายนอก	50
4.1.3 ทดสอบ โปรแกรมการแปลงเวฟเล็ทและแปลงกลับเวฟเล็ท	51
4.2 ผลการทดลองส่วนของฮาร์ดแวร์	52
4.2.1 การทดสอบต่อหน่วยความจำภายนอก	52
4.3 รวมการทำงานของ ซอฟต์แวร์และฮาร์ดแวร์	53
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	55
5.1 บทสรุป	55
5.2 ปัญหาที่เกิดขึ้นในการจัดทำโครงการ	55
5.3 แนวทางการแก้ไขและพัฒนา	56
ภาคผนวก ก	58
ภาคผนวก ข	74
บรรณานุกรม	77
ประวัติผู้แต่ง	78

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ขาต่างๆ ของชิพ AIC	29
ตารางที่ 3.1 ตำแหน่งรีจิสเตอร์ต่าง ๆ	46



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 Morlet Mother Wavelet	3
รูปที่ 2.2 สัญญาณของ $y = \sin(5x)$	4
รูปที่ 2.3 สัญญาณของ $Y = \text{Emp}(-5x^2)$	4
รูปที่ 2.4 Scaling and Translation of Mother Wavelet	6
รูปที่ 2.5 การแปลงสัญญาณ $f(x)$ ให้อยู่ใน โดเมนของเวฟเลต g	8
รูปที่ 2.6 ขบวนการแปลงเวฟเลต	11
รูปที่ 2.7 ขบวนการแปลงกลับเวฟเลต	13
รูปที่ 2.8 การแปลงเวฟเลต	14
รูปที่ 2.9 สัญญาณอินพุตที่ใช้ทดสอบจำนวน 3920 จุดข้อมูล	14
รูปที่ 2.10 ลักษณะของ Dabuchies 2	15
รูปที่ 2.11 (ก) เป็นสัญญาณส่วนของความถี่ (Approximation Coefficients)	16
(ข) สัญญาณความถี่สูง (Detail Coefficients)	16
รูปที่ 2.12 การแตกกระจายเวฟเลตของสัญญาณอินพุต	17
รูปที่ 2.13 (ก) สัมประสิทธิ์ด้านความถี่ต่ำ	18
(ข) สัมประสิทธิ์ด้านความถี่สูง	18
รูปที่ 2.14 (ก) สัญญาณคั่นแบบ	19
(ข) สัญญาณที่ได้จากการรวมกลับเวฟเลต	19
รูปที่ 2.15 การทำ Down Sampling	20
รูปที่ 2.16 การกลับเวฟเลตของค่าสัมประสิทธิ์ C	21
รูปที่ 2.17 สถาปัตยกรรม TMS320C31	24
รูปที่ 2.18 ตำแหน่งหน่วยความจำ	26
รูปที่ 2.19 อินพุตและเอาต์พุตของระบบ	27
รูปที่ 2.20 ตัวถังชีพ (AIC)	28
รูปที่ 2.21 การทำงานภายในของ AIC ชีพ	28
รูปที่ 2.22 การเชื่อมต่อ Parallel Port เข้าไป DSK บอร์ด	31
รูปที่ 2.23 พื้นฐานหน้าจอ Debugger	32
รูปที่ 2.24 ส่วนประกอบ TMS320C3x/C4x เกี่ยวกับซอฟต์แวร์ Development	33
รูปที่ 2.25 การใช้ C Compiler	34

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่ 3.1 ระบบการทำงาน	36
รูปที่ 3.2 วงจร PRE AMP	37
รูปที่ 3.3 ตำแหน่งที่ต่อหน่วยความจำ	38
รูปที่ 3.4 การต่อวงจรหน่วยความจำภายนอก	40
รูปที่ 3.5 จำนวนบิต	41
รูปที่ 3.6 วงจรควบคุมการทำงาน	41
รูปที่ 3.7 ขบวนการลดขนาดของข้อมูล	45
รูปที่ 3.8 แผนผังโปรแกรมควบคุมการทำงานบอร์ด DSK	48
รูปที่ 4.1 การทดสอบหน่วยความจำภายนอก	49
รูปที่ 4.2 สัญญาณสัญญาณอินพุตและสัญญาณเอาต์พุตของวงจรควบคุมการทำงาน	50
รูปที่ 4.3 สัญญาณอินพุตและผลการทดลอง โปรแกรมนำข้อมูลจากอินพุตสู่เอาต์พุต	51
รูปที่ 4.4 สัญญาณอินพุตและผลการทดลอง โปรแกรมนำข้อมูลเก็บในหน่วยความจำภายนอก	52
รูปที่ 4.5 สัญญาณสัญญาณอินพุตและสัญญาณที่แปลงเวฟได้และลดขนาดข้อมูล	53
รูปที่ 4.6 ผลการทดสอบการรวมกันของทั้งหมด	54
รูปที่ 4.7 ภาพถ่ายบอร์ดตัวอย่างของ DSK TMS320C31	55
รูปที่ 4.8 การทดสอบบอร์ดตัวอย่างของ DSK TMS320C31	55

บทที่ 1

บทนำ

1.1 ความสำคัญของปริยญาณิพนธ์

ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทมากขึ้น ได้มีการประยุกต์ใช้งานในด้านต่างๆ รวมทั้งการสื่อสารข้อมูลและใช้ในการบันทึกข้อมูลต่างๆ เป็นต้น และข้อมูลส่วนหนึ่งที่ใช้กับคอมพิวเตอร์นี้จะเป็นข้อมูลเสียงพูด โดยมีแนวโน้มที่จะมีการใช้งานเพิ่มมากขึ้นเนื่องจากข้อมูลเสียงพูดสามารถสื่อความความหมายได้ดีโดยทางอวัจนะ แต่การใช้งานข้อมูลเสียงพูดนี้มีข้อเสียคือ ปริมาณของข้อมูลที่มีจำนวนมาก จะเป็นอุปสรรคในการสื่อสารข้อมูลทำงานได้ช้าและสิ้นเปลืองเนื้อที่ ในการจัดเก็บ

ดังนั้นขบวนการลดขนาดของข้อมูลเสียงพูดจึงเป็นส่วนที่สำคัญที่จะเพิ่มความสามารถในการทำงานของอุปกรณ์ที่มีอยู่ให้มีประสิทธิภาพ ซึ่งจะช่วยให้ประหยัดค่าใช้จ่ายในการเพิ่มความสามารถของอุปกรณ์ให้มีประสิทธิภาพมากยิ่งขึ้น ด้วยเหตุนี้จึงเกิดแนวคิดที่จะลดขนาดของข้อมูลเสียงพูด เพื่อเพิ่มเนื้อที่ในการจัดเก็บ สามารถจัดเก็บข้อมูลเสียงพูดได้มากขึ้นและช่วยลดค่าใช้จ่าย

ในส่วนของโครงการนี้จะนำเสนอวิธีลดข้อมูลเสียงพูดโดยใช้อัลกอริทึมการแปลงเวฟเล็ต (Wavelet) โดยใช้ตัวประมวลผลคือ การประมวลผลสัญญาณเชิงเลข (DSP ย่อมาจาก Digital Signal Processor) โดยเป็นบอร์ดรุ่น DSK (DSP Starter Kit) TMS320C31

1.2 ขีดความสามารถของโครงการ

1. สามารถลดขนาดข้อมูลเสียงพูดได้
2. สามารถแปลงข้อมูลเสียงพูดที่ลดขนาดแล้วกลับเป็นสัญญาณเสียงได้
3. ใช้โปรแกรมภาษาซี ในการเขียนโปรแกรมการแปลงเวฟเล็ต
4. ใช้อุปกรณ์ DSK (DSP Starter Kit) TMS320C31 เป็นตัวประมวลผลการบีบอัดข้อมูลเสียงพูด

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญยานิพนธ์ ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจในแต่ละบทประกอบด้วยเนื้อหาต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหา ดังนี้, การแปลงเวฟเล็ต, การแปลงกลับเวฟเล็ต, ตัวประมวลผลสัญญาณเชิงเลข (TMS320C31)

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงเนื้อหาที่เกี่ยวกับวงจรต่างๆ และโปรแกรมที่ใช้ในโครงการนี้ คือ อัลกอริทึมเวฟเล็ต การใช้งาน บอร์ด DSK TMS320C31

บทที่ 4 การทดลองและผลการทดลอง ผลการทดลองส่วนของซอฟต์แวร์และผลการทดลองส่วนของฮาร์ดแวร์

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา ขั้นตอนการสรุปผล ปัญหาที่เกิดขึ้น, แนวทางการแก้ไข และแนวทางในการพัฒนาโครงการให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก ผังการทำงานและโปรแกรม

ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์

บทที่ 2

ทฤษฎีและหลักการ

2.1 การแปลงเวฟเล็ต (Wavelet Transform)

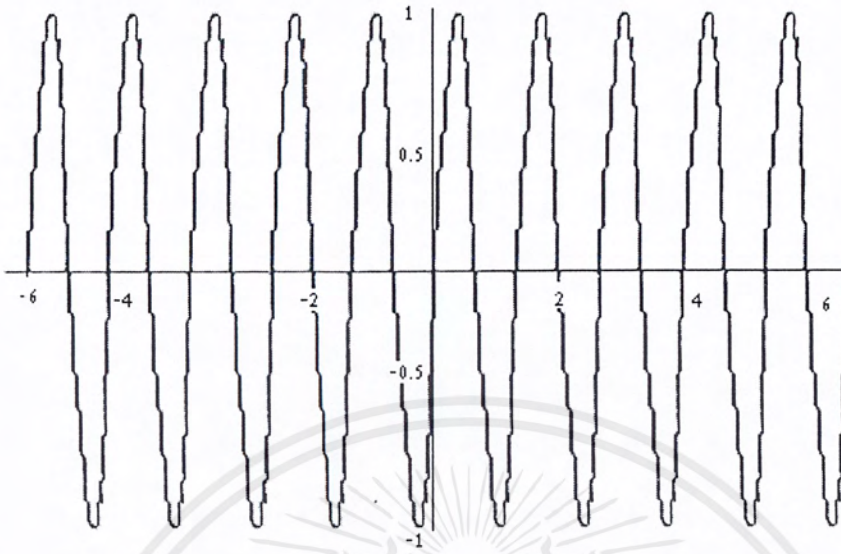
ทฤษฎีเวฟเล็ต (Wavelet Theory) เป็นขบวนการคณิตศาสตร์ ที่ใช้อธิบายถึงโมเดลของสัญญาณระบบ หรือขบวนการโดยใช้เซตสัญญาณ ที่มีลักษณะเฉพาะเป็นองค์ประกอบพื้นฐาน โดยเรียกว่า เซตของเวฟเล็ต (Wavelet Set) สมาชิกในเซตของเวฟเล็ตนี้ จะมีคุณสมบัติสำคัญคือ เป็นสัญญาณที่เกิดจากคั่นแบบอันเดียวกัน คั่นแบบสัญญาณในเซตของเวฟเล็ตนี้เรียกว่า เวฟเล็ตแม่ (Mother Wavelet)

เวฟเล็ตจะมีคุณสมบัติเป็นสัญญาณที่เกิดขึ้นในช่วงเวลาสั้นๆ หรือกล่าวคือ เวฟเล็ตเป็นสัญญาณที่เกิดขึ้นอย่างต่อเนื่องและมีขนาดที่ลดลงสู่ศูนย์ในทั้งสองด้าน ดังแสดงในรูปที่ 2.1

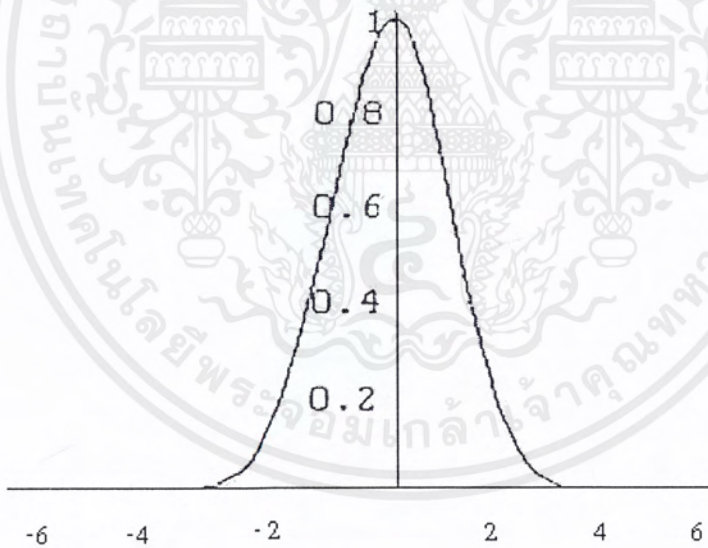


รูปที่ 2.1 Morlet Mother Wavelet

สัญญาณในรูปที่ 2.1 จะประกอบด้วยลักษณะทั้งสองคือ การเกิดขึ้นอย่างต่อเนื่อง ซึ่งได้จากสัญญาณชายน์ ในรูปที่ 2.2 และการลู่เข้าสู่ค่าศูนย์ของสัญญาณในทั้งสองด้านจาก ฟังก์ชันวินโดว์ (Window Function) ดังรูปที่ 2.3 เมื่อฟังก์ชันทั้งสองนี้มาคูณเข้าด้วยกันจะได้ สัญญาณของเวฟเล็ต ดังรูปที่ 2.1



รูปที่ 2.2 สัญญาณของ $y = \sin(5x)$



รูปที่ 2.3 สัญญาณของ $Y = \text{Exp}(-5x^2)$

จากเวฟเลตแม่ที่ได้ จะถูกนำไปสร้างเป็นสมาชิกตัวอื่นๆ ในเซตของเวฟเลตที่ต้องการนำไปอธิบายถึงสัญญาณ ขบวนการหรือระบบต่างๆ โดยสมาชิกตัวอื่นๆ นี้จะได้มาจาก เวฟเลตแม่ที่ถูก “สเกล” (Scaling : a) และเลื่อนตำแหน่ง (Translation : b) ไปตามแกนเวลา ถ้าให้ $g(t)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นฟังก์ชันของเวฟเล็ทแม่แล้ว การสเกลด้วยพารามิเตอร์ของ “a” และ “การเลื่อนตำแหน่ง” ด้วยพารามิเตอร์ของ “b” จะสามารถหาได้จากสมการ

$$g_{a,b}(t) = g\left(\frac{t-b}{a}\right) \quad (2.1)$$

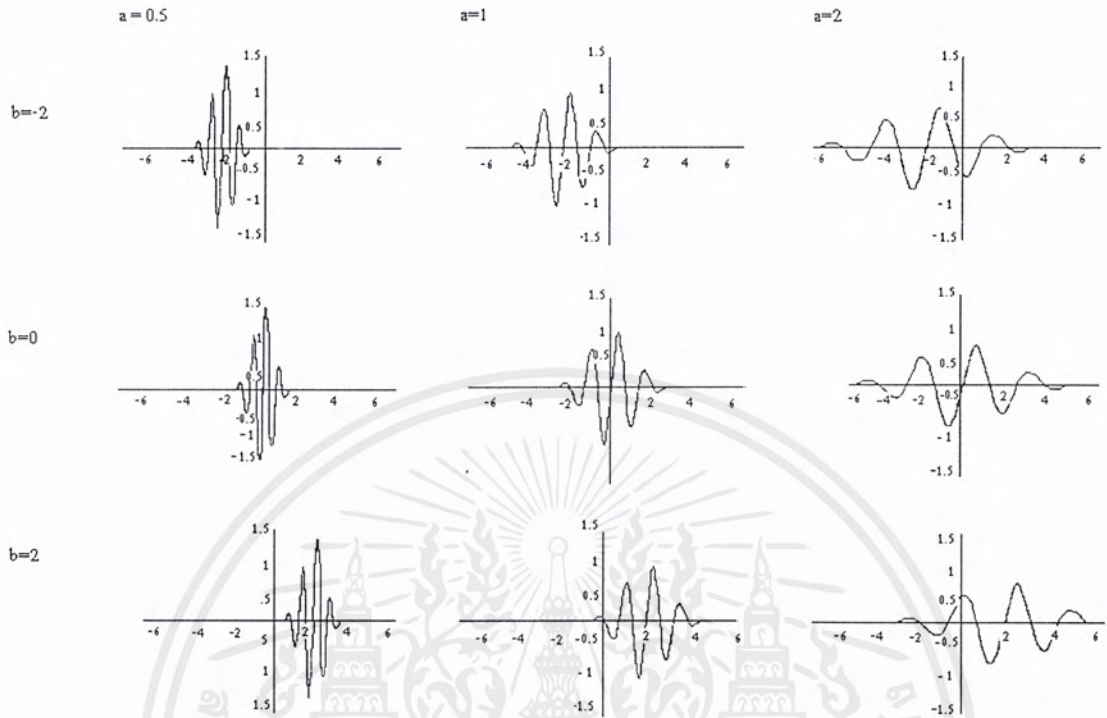
เมื่อ a ค่าอัตราส่วนของการ Scaling

เมื่อ b การเลื่อนตำแหน่ง

ซึ่งการทำ Scaling และ Translation ตามแกนเวลานี้ จะรวมเรียกว่า Affine Operation (การทำ Linear Mapping ร่วมกับการทำ Translation) และเพื่อให้สัญญาณที่ถูกสเกลแล้วมีค่าพลังงานของสัญญาณที่เท่าเดิม จึงจะต้องทำการโดยปกติ (Normalization) สัญญาณที่ได้นี้ด้วย $\frac{1}{\sqrt{a}}$ เสมอจะได้

$$g_{a,b}(t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) \quad (2.2)$$

จากรูปที่ 2.4 จะแสดงตัวอย่างของสมาชิกในเซตของเวฟเล็ทที่มีเวฟเล็ทแม่ตามตัวอย่างที่ได้ กล่าวมาแล้ว



รูปที่ 2.4 Scaling and Translation of Mother Wavelet

จากตัวอย่าง จะสามารถสังเกตได้ว่าลักษณะและจำนวนลูกคลื่นของสัญญาณนั้นจะเหมือนกันทั้งหมด สมาชิกแต่ละตัวแตกต่างกันที่ ขนาดและตำแหน่งทางแกนเวลาเท่านั้น

2.2 พื้นฐานของทฤษฎีเวฟเล็ต

ทฤษฎีของเวฟเล็ตเป็นการอ้างถึงอย่างหนึ่ง เช่นเดียวกับระบบตัวเลขที่ใช้แทนจำนวนของสิ่งต่าง ๆ ทฤษฎีของเวฟเล็ตจะสามารถใช้อธิบายถึงต่าง ๆ ได้ว่าจะ เป็น ราคาหุ้น ผลเฉลยของสมการดิฟเฟอเรนเชียล การเต้นของหัวใจ เป็นต้น และนอกเหนือจากทฤษฎีของเวฟเล็ตยังสามารถใช้อธิบายถึงการทำงานของระบบต่างๆ ได้

ทฤษฎีของเวฟเล็ตซึ่งเป็นเครื่องมือคณิตศาสตร์อย่างหนึ่งที่สามารถนำไปประยุกต์ใช้งาน ได้สิ่งต่างๆ โดยมีวัตถุประสงค์ที่จะใช้อธิบาย ระบบหรือสัญญาณเหล่านั้นได้ดีกว่าวิธีการอื่นๆ การที่จะนำทฤษฎีของเวฟเล็ตไปใช้งาน จึงอยู่บนพื้นฐานที่ว่า การนำมาใช้งานนั้น ประสิทธิภาพเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีของเวฟเล็ตจะอธิบายสิ่งใดสิ่งหนึ่ง โดยการแยกสิ่งเหล่านี้ออกเป็นส่วนประกอบย่อยๆ ที่มีความสัมพันธ์กัน โดยที่ส่วนย่อยๆ เหล่านี้ก็คือ เวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งไป และจะมีค่าสัมประสิทธิ์ (Coefficient) คู่อยู่กับเวฟเล็ตแต่ละตัว

ขบวนการในการแยกสัญญาณออกเป็นส่วนย่อยๆ หรือการหาค่าของสัมประสิทธิ์นี้จะเรียกว่า “การแตกกระจายเวฟเล็ต” (Wavelet Decomposition) หรือการแปลงเวฟเล็ต (Wavelet Transform) และในทางกลับกัน การนำค่าสัมประสิทธิ์แยกออกเป็นส่วนย่อยๆ มารวมกลับเพื่อได้สัญญาณเดิมที่ต้องการ จะเรียกว่า “การรวมกลับเวฟเล็ต” (Wavelet Reconstruction) หรือ “การแปลงกลับแบบเวฟเล็ต” (Inverse Wavelet Transform)

การแปลงเวฟเล็ตแบบต่อเนื่องของฟังก์ชัน $f(x)$ จะสามารถแสดงได้โดยสมการทางคณิตศาสตร์คือ

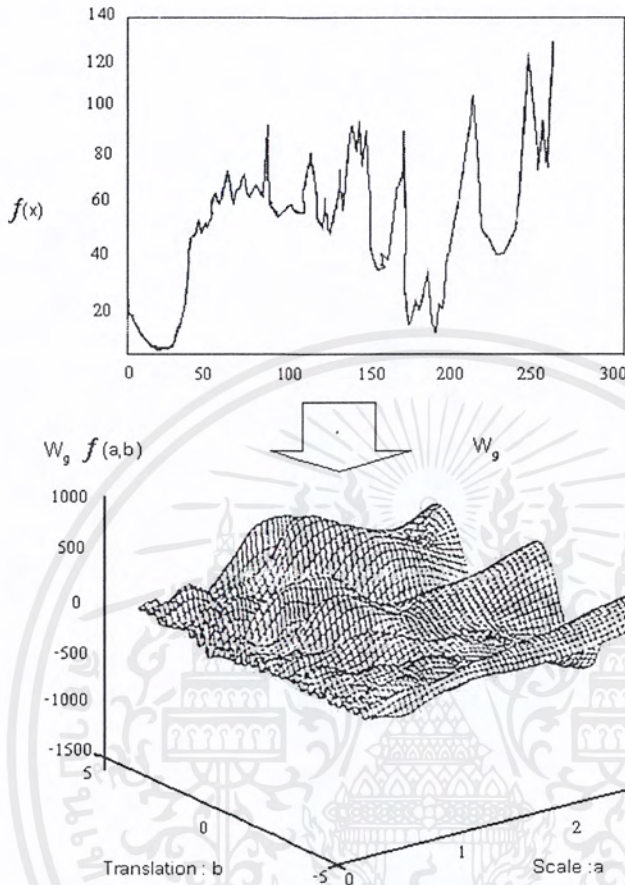
$$W_g[f(x)](a, b) = |a|^{-2} \int f(x) g^* \left(\frac{x-b}{a} \right) dx \quad (2.3)$$

เมื่อ $g(x)$ ฟังก์ชันของเวฟเล็ตแม่และ $g^*(x)$ คือ Complex Conjugate ของ $g(x)$

a คือ Scaling

b คือ Translation

และจะใช้สัญลักษณ์ $W_g[f(x)](a, b)$ แทนค่าสัมประสิทธิ์ที่ได้เพื่อความสะดวก จากข้างต้นจะเห็นได้ว่าการแปลงเวฟเล็ตจะเป็นการส่งผ่านฟังก์ชัน ที่มีตัวแปรอิสระ x ไปยังปริภูมิ (Space) ของฟังก์ชันที่เป็นสองมิติ ของตัวแปรอิสระคือ a และ b ตามลำดับ ค่าสัมประสิทธิ์ของเวฟเล็ตที่ a, b นี้ จะแสดงถึง สหสัมพันธ์ (Degree of Correlation) ระหว่างเวฟเล็ตแม่ ที่สเกล a และตำแหน่ง b กับสัญญาณอินพุต $f(x)$ ที่ตำแหน่งเดียวกัน โดยสัญญาณทั้งสองนี้มีความเหมือนกันมาก ค่าสัมประสิทธิ์ที่ได้ก็จะมากตามไปด้วยและเซตของค่าสัมประสิทธิ์ทั้งหมดที่ได้ ($W_g f(a, b)$ ที่ a, b ใด ๆ) จะเรียกว่า โดเมนของเวฟเล็ต (Wavelet Domain) ซึ่งจะเป็นการอธิบายถึงฟังก์ชัน $f(x)$ โดยใช้เวฟเล็ตแม่ ดังรูปที่ 2.1 จะแสดงการแปลงเวฟเล็ตของสัญญาณ $f(x)$ ให้อยู่ในโดเมนของเวฟเล็ต g



รูปที่ 2.5 การแปลงสัญญาณ $f(x)$ ให้อยู่ในโดเมนของเวฟเล็ต g

จะสังเกตเห็นว่าสำหรับการแปลงเวฟเล็ตนั้น ส่วนของเวฟเล็ตแม่ที่สเกลและตำแหน่งต่าง ๆ (จากตัวอย่างในรูปที่ 2.4) จะมีจำนวนของลูกคลื่นสัญญาณที่เท่ากัน เปรียบเสมือนกับฟังก์ชันวินโดว์ ที่มีขนาดเปลี่ยนไปตามความถี่ ถ้าค่าของสเกล (a) น้อยกว่าหนึ่ง ขนาดของวินโดว์ นี้จะเล็กลง จำนวนลูกคลื่นต่อหนึ่งหน่วยเวลาก็จะมาก จำนวนลูกคลื่นสัญญาณที่มาก ก็จะหมายถึงความถี่สูงนั่นเอง แต่ถ้าจำนวนลูกคลื่นต่อหนึ่งหน่วยเวลามีค่าน้อยจะหมายถึงความถี่ต่ำ จะเห็นได้ว่าค่าของช่วงเวลา กับความถี่จะมีความสัมพันธ์กัน การแปลงเวฟเล็ตจะกำหนดช่วงเวลาที่ยาวขึ้น ในการวิเคราะห์ ความถี่สูง ช่วงเวลาที่ยาวขึ้น สำหรับความถี่ต่ำลงมา

2.2.1 ตัวแปลงเวฟเล็ตและคำจำกัดความ

ก่อนที่จะอธิบายถึงขบวนการแปลงเวฟเล็ตนั้น จะต้องกำหนดถึงฟังก์ชันที่สามารถนำมาใช้เป็นเวฟเล็ตแม่ได้ จากที่ได้กล่าวมาแล้ว ฟังก์ชันที่จะเป็นเวฟเล็ตได้นั้น จะต้องมีคุณสมบัติของการแกว่งไปมา และลู่เข้าสู่ศูนย์ ซึ่งฟังก์ชันที่จะสามารถใช้เป็นเวฟเล็ตแม่ จะต้องมีคุณสมบัติทั้งสองข้อนี้ ทัวไปแล้วสัญญาณต่างๆ นั้นจะมีคุณสมบัติทั้งสองข้ออยู่แล้ว จะกล่าวได้ว่าฟังก์ชันที่สามารถหาค่าพลังงานได้ (Finite energy function : $L^2(\mathbb{R})$) g จะใช้เป็นเวฟเล็ตแม่ได้เมื่อ

$$c_g = \int_{-\infty}^{\infty} \frac{|G(\omega)|^2}{|\omega|} d\omega < \infty \tag{2.4}$$

เมื่อ $G(\omega)$ 푸ริเยร์ ทรานฟอร์ม (Fourier Transform) ของ g

C_g ค่าพลังงานของ g

ตัวแปลงเวฟเล็ต W_g จะเป็นการ Mapping สัญญาณที่เป็น Finite Energy ของ $L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R} \setminus \{0\} \times \mathbb{R})$ หรือจะอธิบายได้คือ สัญญาณที่เป็น Finite Energy ใน เวลา (Time) หรือ สเปก โดเมน (Space Domain) การแปลงเวฟเล็ตของฟังก์ชัน $f(a, b)$ ด้วยฟังก์ชันของเวฟเล็ตแม่ g จะกำหนดโดย

$$\begin{aligned} \text{Wavelet Coefficient} &= W_g f(a,b) \\ &= |a|^{\frac{1}{2}} \int f(x) g^* \left(\frac{x-b}{a} \right) dx \\ &= \left\langle f, \frac{1}{\sqrt{|a|}} g \left(\frac{x-b}{a} \right) \right\rangle \\ &= \langle f, g_{a,b} \rangle = \langle f, U(a,b)g \rangle \end{aligned} \tag{2.5}$$

เมื่อ g^* แสดงถึง Complex Conjugate

$\langle f, g_{1,2} \rangle$ Inner Product

$$U(a,b): g(x) \rightarrow \frac{1}{\sqrt{|a|}} g \left(\frac{x-b}{a} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 ซึ่งเป็นตัวอย่างของการแปลงเวฟเลต ในตัวอย่างนี้ ฟังก์ชัน $g(x)$ จะต้องมีความสมบัติของเวฟเลต เชื่อกของเวฟเลตจะสร้างได้จาก $g_{a,b}$ ที่กำหนดโดย Unitary Affine Mapping

$$U(a,b): g(x) \rightarrow \frac{1}{\sqrt{|a|}} g\left(\frac{x-b}{a}\right) \text{ หรือจะสามารถค่า } b \text{ ค่าอัตราส่วนของการสเกลนี้ จึงอยู่ใน}$$

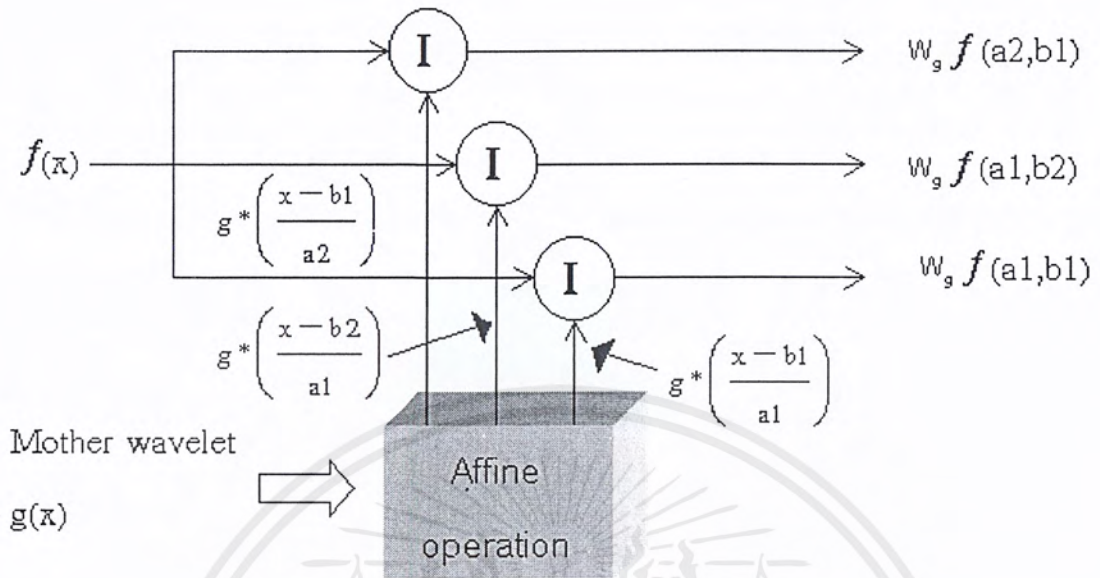
เชื่อกของเลขจำนวนจริงที่ไม่เท่ากับ $0 (R \setminus \{0\})$ ส่วนอธิบายได้ง่ายๆ คือ $g_{a,b}$ เกิดจากฟังก์ชันของเวฟเลตแม่ ที่ถูกสเกล ด้วยอัตราส่วน a และเลื่อนไปด้วยคำว่า “Unitary” นั้นหมายถึง Energy Normalization ซึ่งได้จาก $\frac{1}{\sqrt{|a|}}$ การทำ ถึง Energy Normalization นี้จะทำให้เวฟเลตแม่ที่ถูกสเกล แล้วนั้นยังคงมีค่าขนาดของพลังงานที่เท่าเดิม หรืออาจใช้เวฟเลตแม่ ที่มีค่าพลังงานเท่ากับ หนึ่งหน่วย ($c_g = 1$) ก็ได้

การที่ค่าพลังงานของเวฟเลตแม่มีค่าเป็นหนึ่งหน่วย จะทำให้เกิดคุณสมบัติที่สำคัญ คือค่าพลังงานของสัมประสิทธิ์ที่ได้ในเวฟเลตโดเมน จะเท่ากับค่าพลังงานของสัญญาณที่อยู่ใน โดเมนของเวลา หรือในปริภูมิ และยังทำให้ ค่าสัมประสิทธิ์เวฟเลตที่ (a, b) ใดๆ สามารถที่จะนำมาเปรียบเทียบกันได้

สำหรับการแปลงเวฟเลตแบบต่อเนื่อง (Continuous Wavelet Transform) นั้น คุณสมบัติที่ต้องการของฟังก์ชันที่จะนำมาเป็นเวฟเลตแม่นั้น จะมีเพียงดังที่กล่าวข้างต้น ซึ่งการเลือกใช้ฟังก์ชันของเวฟเลตแม่ ก็จะขึ้นอยู่กับ การประยุกต์ใช้งานที่ต้องการ

เชื่อกของค่าสัมประสิทธิ์เวฟเลตนี้ จะสามารถใช้เป็นตัวแทนของสัญญาณและจากการแปลงผกผันของค่าสัมประสิทธิ์ที่ได้ โดยใช้เชื่อกของเวฟเลตที่ได้ ด้วยเวฟเลตแม่ตัวเดิม ก็จะสามารถนำมาใช้ในการแปลงผกผันได้ (Inverse Transform)

ค่าสัมประสิทธิ์เวฟเลต แต่ละตัวจะเปรียบเสมือนเป็นส่วนประกอบย่อยๆ ของสัญญาณ (ฟังก์ชัน) และเมื่อรวมส่วนประกอบย่อยๆ เหล่านี้เข้าด้วยกัน จะได้สัญญาณกลับออกมาในการที่จะหาค่าของสัมประสิทธิ์เวฟเลตที่จะเป็นตัวแทนของสัญญาณ สัญญาณนั้นถูกฉาย (Projection) ลงไปยังส่วนประกอบย่อยในเวฟเลตเชื่อก ซึ่งผลลัพธ์จะได้เป็นค่าจำนวนจริง หรือจำนวนเชิงซ้อนที่เรียกว่า ค่าสัมประสิทธิ์เวฟเลต (Wavelet Coefficient) ของสัญญาณ หรือฟังก์ชัน ถึง f ที่สัมพันธ์กับเวฟเลตแม่ g หรือแสดงด้วย $W_g f(a,b)$ ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 ขบวนการแปลงเวฟเล็ต

นอกเหนือจากที่อธิบายได้ด้วยสมการที่ 2.5 แล้วขบวนการฉาย จะสามารถเรียกได้คือ ขบวนการ Correlation ซึ่งเป็นการเปรียบเทียบว่าสัญญาณ ทั้งสองมีความเหมือนกันมากเพียงใด สัญญาณ f จะถูกนำมาเปรียบเทียบกับ เวฟเล็ตแม่ g ที่ถูกสเกล และเลื่อนออกไป จากในรูปที่ 2.6 ขบวนการของฉายหรือ Correlation ด้วยสัญญาณ I อยู่ภายในวงกลม ข้อมูลเอาต์พุตที่ได้ จะเป็นค่าสัมประสิทธิ์เวฟเล็ต $W_g f(a, b)$ ที่ต้องการ

ในขบวนการแปลงเวฟเล็ตที่แสดงในรูปที่ 2.6 จะสามารถเรียกได้อีกอย่างหนึ่งว่า เป็น Analysis Filter โดยมันจะทำการแตกกระจายสัญญาณออกเป็นส่วนย่อย ซึ่งจะถูกนำไปใช้ในการวิเคราะห์ต่อไป ขบวนการแปลงกลับเวฟเล็ต หรือ Synthesis Filter จะเป็นการนำส่วนประกอบย่อยนี้来做ได้ดีกว่า

2.23 การแปลงกลับเวฟเลต

การแปลงกลับเวฟเลต ที่ได้จากขบวนการแปลงเวฟเลต ตามสมการที่ 2.5 นั้น จะสามารถทำได้ดังนี้ขบวนการแสดงคังรูปที่ 2.17 ซึ่งในการแปลงกลับเวฟเลตนี้จะได้ฟังก์ชันเดิมกลับออกมา

ถ้า $f(x)$ และ $g(x)$ เป็นฟังก์ชันที่สามารถหาค่าพลังงานได้แล้ว (อยู่ในรูป $L^2(\mathbb{R})$) และ $g(x)$ เป็นฟังก์ชันที่มีคุณสมบัติเวฟเลตแม่ได้แล้ว $g_{a,b}(t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right)$ ($a, b \in (\mathbb{R} \setminus \{0\} \times \mathbb{R})$)

W_g^{-1} ขบวนการแปลงกลับเวฟเลต Mapping จะเป็นการ ฟื้นฟูของค่าสัมประสิทธิ์เวฟเลต มาเป็นสัญญาณหนึ่งมิติของเวลาคือ $W_g^{-1} : L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R} \setminus \{0\} \times \mathbb{R})$ แสดงได้ดังนี้

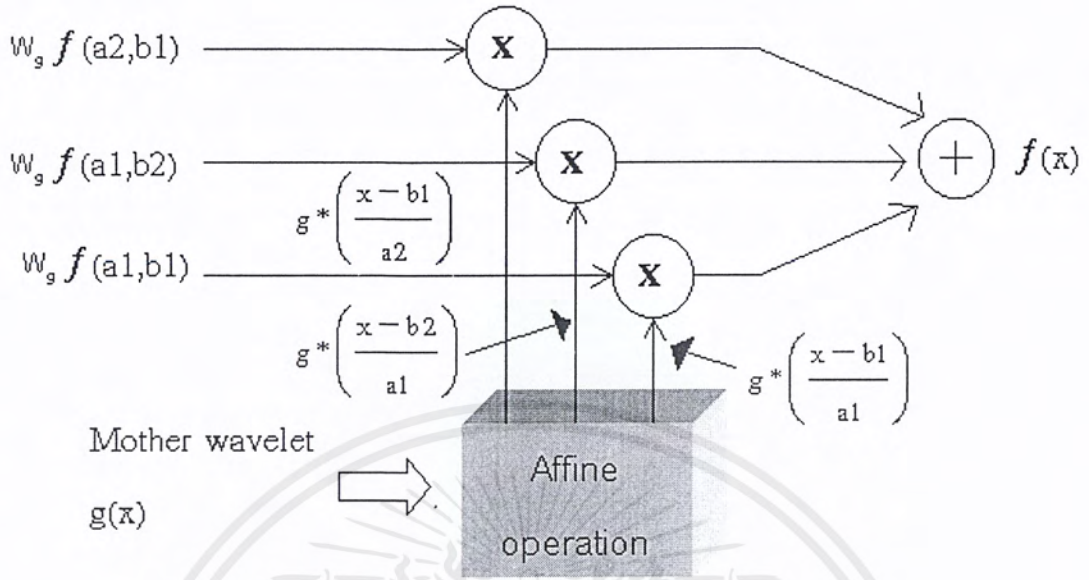
$$W_g^{-1} : W_g f(a, b) \longrightarrow f(x)$$

$$f(x) = \frac{1}{c_g} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_g f(a, b) \frac{1}{\sqrt{|a|}} g\left(\frac{x-b}{a}\right) \frac{db da}{a^2} \quad (2.6)$$

การแปลงกลับเวฟเลตจะสร้างฟังก์ชันหรือสัญญาณ ต้นฉบับขึ้นมาจากการรวมกันของเวฟเลตแม่ (Mother Wavelet) ที่ Scale-Translation ต่างๆ ที่คุณอยู่กับค่าถ่วงน้ำหนัก (ค่าสัมประสิทธิ์เวฟเลต คือ $W_g f(a, b)$) และสำหรับในกรณีที่มีการแปลงกลับเวฟเลตนี้ใช้เวฟเลตแม่ที่ แตกต่างไปจากเวฟเลตแม่ที่ใช้ในขบวนการแปลงเวฟเลต ผลลัพธ์ที่ได้จะเป็นสัญญาณหรือฟังก์ชันที่แตกต่างไปจากฟังก์ชันหรือสัญญาณต้นฉบับ คือ

$$f(x) = \frac{1}{c_g} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_g f(a, b) \frac{1}{\sqrt{|a|}} g\left(\frac{x-b}{a}\right) \frac{db da}{a^2} = f(x) \quad (2.7)$$

ถ้า $g(2x) \neq g(x)$



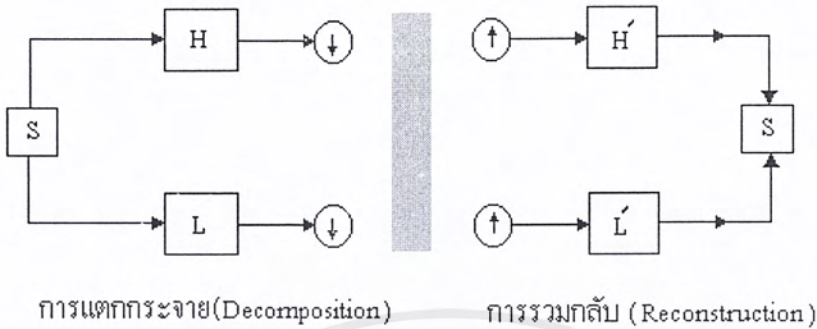
รูปที่ 2.7 ขบวนการแปลงกลับเวฟเล็ต

2.3 การกระจายและรวบรวมข้อมูลด้วยเวฟเล็ต

ในหัวข้อนี้จะเป็นการแสดงถึงการกระจายและรวมข้อมูลด้วยเวฟเล็ต โดยในทางปฏิบัติใช้โปรแกรม Matlab V5.1 เป็นเครื่องมือ

2.3.1 การแปลงเวฟเล็ต

การแปลงเวฟเล็ตหรือการแตกกระจายข้อมูล ที่ใช้นั้นจะเป็นการแปลงเวฟเล็ตในแบบของอนุกรมเวฟเล็ตที่ไม่ต่อเนื่อง (Discrete Time Wavelet Series) เนื่องมาจากการแปลงแบบนี้ ข้อมูลที่ได้ จะไม่มีความซ้ำซ้อนกันจึงเหมาะสมที่จะนำมาใช้เพื่อลดขนาดข้อมูล ซึ่งการแปลงเวฟเล็ตนี้จะสามารถกระทำได้ด้วยวิธี การของมัลติรีโซลูชันอนาไลซิส (Multi Resolution Signal Analysis) ซึ่งจะประกอบไปด้วยส่วนต่างๆ คือ



รูปที่ 2.8 การแปลงเวฟเล็ต

สัญญาณอินพุต ที่ใช้ในการประมวลผลสัญญาณเชิงเลขนี้ จะเป็นสัญญาณแบบเต็มหน่วยเวลา ที่ได้ขบวนการ Sampling (สัญญาณสุ่ม) ดังรูป 2.9 จะเป็นตัวอย่างของสัญญาณ ที่ใช้ในการทดสอบ



รูปที่ 2.9 สัญญาณอินพุต ที่ใช้ทดสอบจำนวน 3920 จุดข้อมูล

ตัวกรอง (Filter) สามารถที่สร้างได้จากคอนโวลูชัน (Convolution) ของสัญญาณ กับฟังก์ชันในการโอนย้าย (Transfer Function)

ซึ่งในการใช้งานนี้ จะใช้ตัวกรองทั้งหมด 2 ตัว คือ G, H

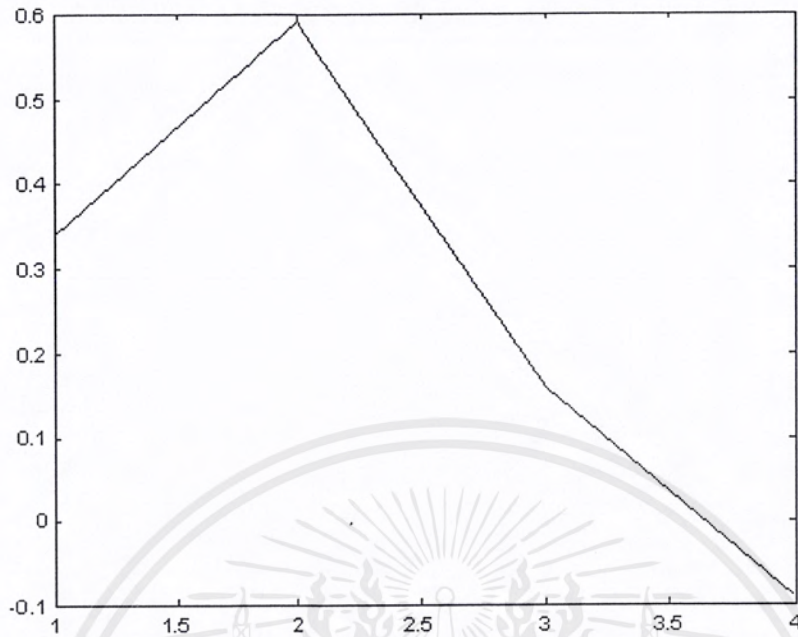
ตัวกรอง $g(k)$ ที่จะใช้ก็คือ ฟังก์ชันของเวฟเล็ตแม่ ในการทดลองจะเลือกใช้ ฟังก์ชันที่เป็นเวฟเล็ตแม่ คือ Daubechies 2

ตัวกรองแต่ละตัวจะสามารถหาได้จากการคำนวณคือ

$$G = g(k) = [0.3415 \quad 0.5915 \quad 0.1585 \quad -0.0915]$$

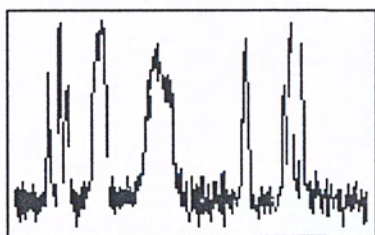
$$H = h(k) = [-0.0915 \quad -0.1585 \quad 0.5915 \quad -0.3415]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



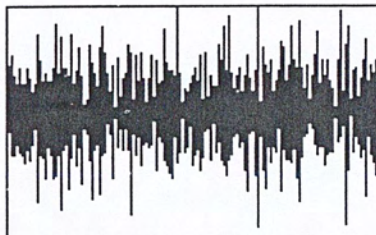
รูปที่ 2.10 ลักษณะของ Dabuchies 2 ซึ่งประกอบด้วย 4 จุดข้อมูลคือ [0.3415 0.5915 0.1585 0.0915]

การทำดาวนแซมปลิง (Down Sampling) เป็นการลดอัตราการสุ่มของสัญญาณสัญญาณลง การลดอัตราการสุ่มนี้ จะกระทำกับสัญญาณที่ตัวกรองมาแล้ว เพื่อเป็นการลดความซ้ำซ้อนกันของข้อมูลที่ได้จากตัวกรองทั้งสอง ซึ่งจะทำให้สัญญาณ ที่ได้จากตัวกรองนี้ มีจำนวนจุดข้อมูลลดลงครึ่งหนึ่ง สัมประสิทธิ์เวฟเลต สัมประสิทธิ์เวฟเลตที่ได้นี้จะแบ่งออกเป็นสองส่วนตัวกรองที่ใช้คือ Approximation Coefficients (G) ความถี่ต่ำ และ Detail Coefficients (H) ความถี่สูง ดังรูปที่ 2.11



Approximation Coefficient

(ก)



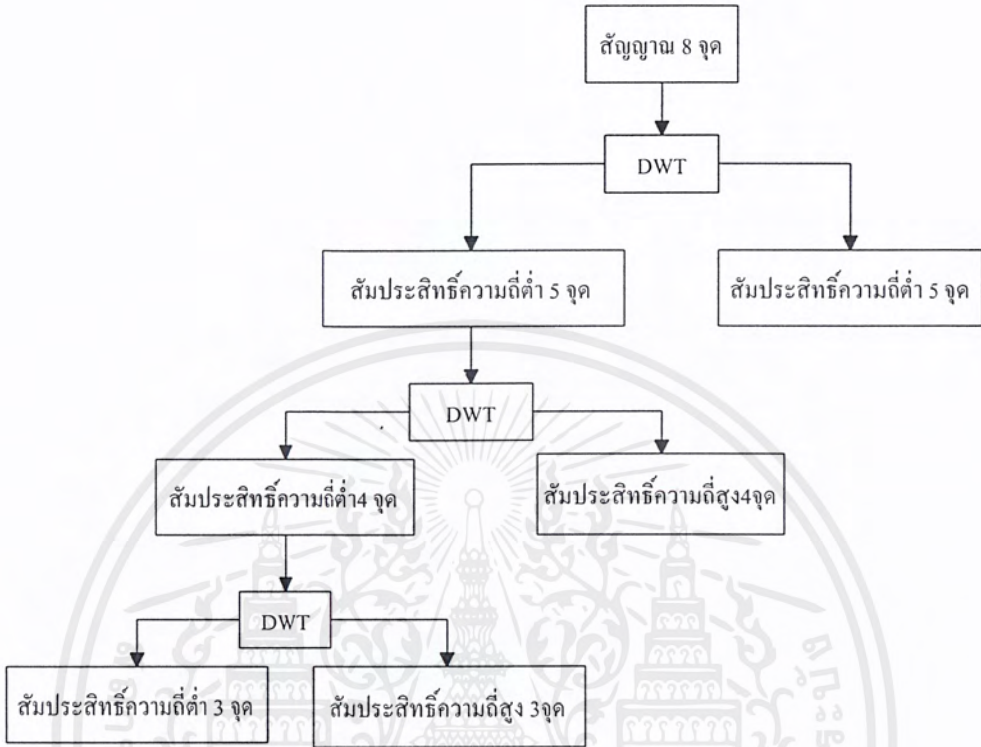
detail Coefficient

(ข)

รูปที่ 2.11 (ก) เป็นสัญญาณส่วนของความถี่ (Approximation Coefficients)

(ข) สัญญาณความถี่สูง (Detail Coefficients)

จากการทดลองข้างต้นจะเป็นการแปลงหรือแตกกระจายเวฟเล็ฟหนึ่งระดับของสัญญาณ ซึ่งการแตกกระจายสัญญาณจะสามารถกระทำต่อไปในระดับความละเอียดที่ต่ำลงมาได้ โดยทำการแตกกระจายค่าสัมประสิทธิ์ที่ได้จากส่วนตัวกรองก็จะแตกกระจายดังรูปที่ 2.12 ซึ่งเรียกว่า แบบ Multi Resolution Decomposition

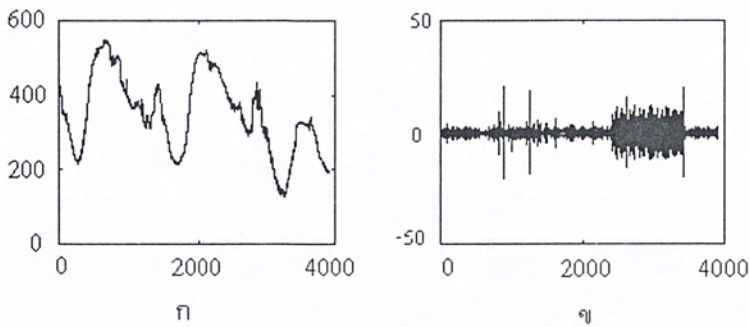


รูปที่ 2.12 การแตกกระจายเวฟเล็ทของสัญญาณอินพุต

ทั้งนี้การกระจายเวฟเล็ทจะกระทำต่อไปได้เฉพาะในกรณีที่ขนาดของสัญญาณ มีจำนวนจุดของข้อมูลเป็นจำนวนคู่เท่านั้น จากรูป 2.12 จะมีจำนวนข้อมูล 8 จุด จึงสามารถแตกกระจายเวฟเล็ทได้ 3 ครั้ง

2.3.2 การแปลงกลับเวฟเล็ท

การแปลงกลับเวฟเล็ทจะสามารถทำได้จาก ค่าสัมประสิทธิ์เวฟเล็ท ซึ่งถ้าเวฟเล็ทแม่ที่ใช้เป็นตัวเดียวกันแล้ว สัญญาณเอาท์พุทที่ได้จะเหมือนกับสัญญาณอินพุททุกประการ โดยมีขั้นตอนดังนี้ การทำ Up Sampling โดยการแทรกข้อมูลศูนย์เพิ่มลงไปนข้อมูลของค่าสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ท เพื่อเป็นการเพิ่มความถี่ของสัญญาณให้สูงขึ้น สัญญาณหลังผ่านการทำ Up Sampling แล้วจะได้ดังรูปที่ 2.13



รูปที่ 2.13 (ก) สัมประสิทธิ์ด้านความถี่ต่ำ
(ข) สัมประสิทธิ์ด้านความถี่สูง

ส่วนของตัวกรอง สามารถที่สร้างได้จากการ Convolution ของสัญญาณกับ Transfer Function ซึ่งในการใช้งานนี้จะต้องใช้ตัวกรอง \tilde{G}, \tilde{H}

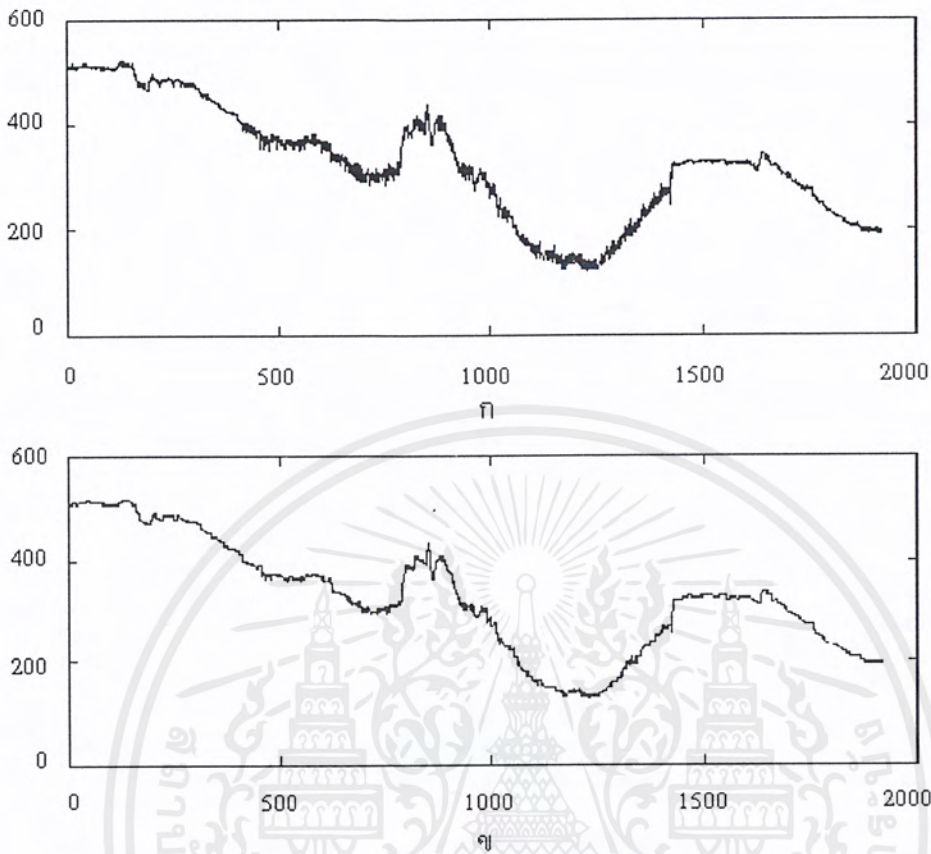
ตัวกรอง $g(k)$ ที่ใช้ก็คือ Function ของเวฟเล็ทแม่ ในการทดลองจะเลือกใช้ ฟังก์ชันที่เป็นเวฟเล็ทแม่คือ Daubechies 2

ตัวกรองแต่ละจะสามารถหาได้จาก การคำนวณคือ

$$\tilde{G} = \tilde{g}(k) = g(N - k) = [-0.0915 \quad 0.1585 \quad 0.5915 \quad 0.3415]$$

$$\tilde{H} = \tilde{h}(k) = h(N - k) = [-0.3415 \quad 0.5915 \quad -0.1585 \quad -0.0915]$$

สัญญาณเอาต์พุต จะสร้างกลับมาได้จากการผลรวมของสัญญาณใช้เป็นอินพุตในขบวนการแปลงเวฟเล็ท ซึ่งความผิดพลาดที่เกิดขึ้นนี้แสดงได้ดังรูปที่ 2.14



รูปที่ 2.14 (ก) สัญญาณต้นแบบ

(ข) สัญญาณที่ได้จากการรวมกลับเวฟเล็ต

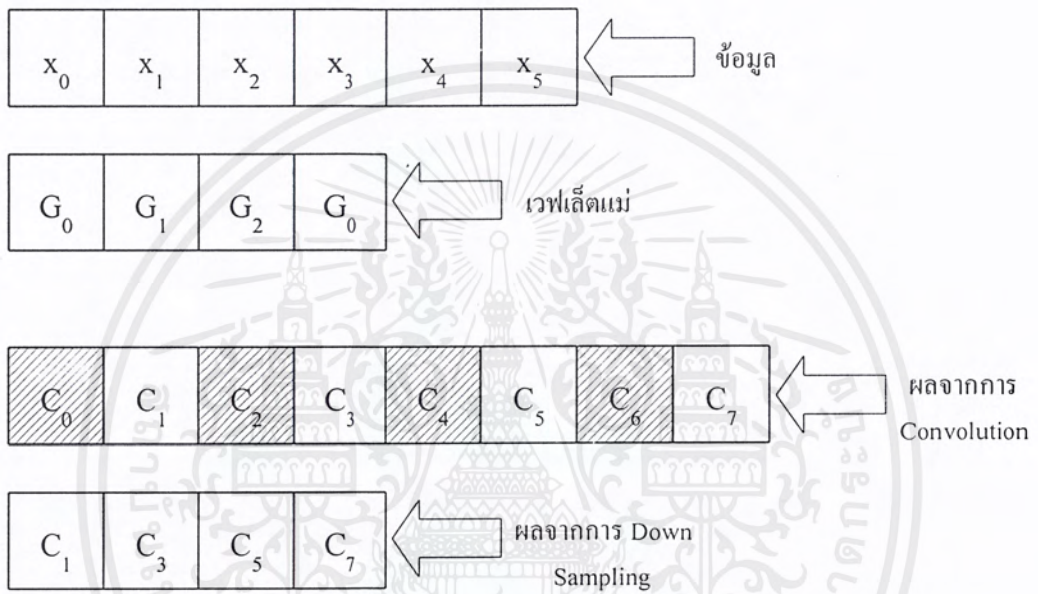
จากการทดลองที่ได้จะเห็นว่า ค่าความผิดพลาดระหว่างสัญญาณทั้งสองจะมีอยู่น้อยมาก ซึ่งสาเหตุที่ทำให้สัญญาณเอาต์พุตมีความแตกต่างจากอินพุตนั้นเนื่องมาจากความผิดพลาดในการปิดเศษของการคำนวณ (Round of Error)

2.4 วิธีการที่มีประสิทธิภาพในการแปลงเวฟเล็ต

ขบวนการแปลงเวฟเล็ต Multi Resolution Wavelet Transform ที่ได้แสดงข้างต้นนี้เป็นขบวนการที่อธิบายไว้ เป็นวิธีที่ง่ายแต่ยังขาดประสิทธิภาพ เนื่องจากจะมีการคำนวณมากเกินไป จำเป็น คือ ในการ Convolution ของสัญญาณกับฟังก์ชันของเวฟเล็ต ซึ่งผลลัพธ์ที่ได้จะถูกนำมาใช้งานเพียงครั้งหนึ่งเท่านั้น จากการทำ Decimation by 2 ดังนั้นในการใช้งานจริงของขบวนการแปลงเวฟเล็ตจึงต้องถูกปรับปรุงโดยแทนที่จะต้องทำการ Convolution ของสัญญาณทุก ๆ จุด ก็จะทำเพียงเฉพาะจุดที่ต้องการนำไปใช้เท่านั้นซึ่งก็จะทำให้ เวลาที่ต้องใช้ในการคำนวณนี้ลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งหนึ่ง การแปลงเวฟเล็ต ขบวนการ Multi Resolution Wavelet Transform นี้ประกอบด้วย การกระทำพื้นฐานอยู่ 2 แบบ คือ การทำ Convolution และการทำ Down Sampling สัญญาณอินพุต A_0 จะถูก Convolution กับฟังก์ชันของ Mother Wavelet (G) และส่วนกลับ ของ Mother Wavelet (h) จากสัญญาณที่ผ่านการกรอง หรือ Convolution นี้แล้ว จะถูกนำมาผ่านการ Down Sampling เพื่อเป็นการลดความซ้ำซ้อนกันของข้อมูล ขบวนการ Down Sampling นี้ จะเป็นการตัดเอาข้อมูลบางส่วนของสัญญาณทิ้งไป



รูปที่ 2.15 การทำ Down Sampling

จากรูป 2.15 นำข้อมูลมา Convolution กับเวฟเล็ตแม่ ดังสมการ

$$C_0 = X_0 G_0$$

$$C_1 = X_0 G_1 + X_1 G_0$$

$$C_2 = X_0 G_2 + X_1 G_1 + X_2 G_0$$

$$C_3 = X_0 G_3 + X_1 G_2 + X_2 G_1 + X_3 G_0$$

$$C_4 = X_1 G_3 + X_2 G_2 + X_3 G_1 + X_4 G_0$$

$$C_5 = X_2 G_3 + X_3 G_2 + X_4 G_1 + X_5 G_0$$

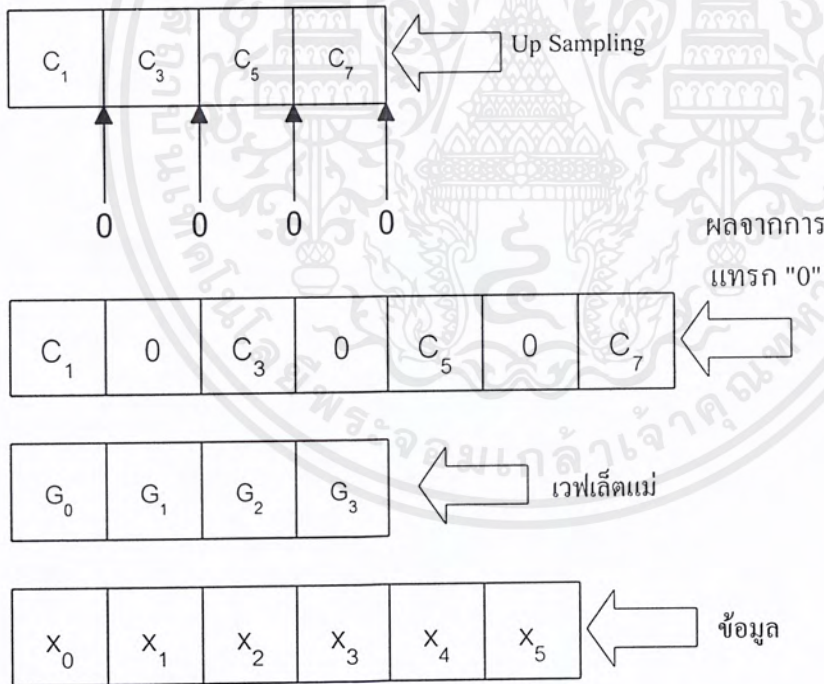
$$C_6 = X_3 G_3 + X_4 G_2 + X_5 G_1 + 0$$

$$C_n = X_n G_n + X_{(n+1)} G_{(n-1)}$$

สัมประสิทธิ์ C_n ถูก Down Sampling โคนการตัดเอาสัญญาณที่ตำแหน่ง $C_2, C_4, C_6, C_8, \dots, C_{2N}$ ทิ้งไป จะเห็นว่าข้อมูลที่คำนวณได้จากการทำ Convolution บางส่วน ไม่ได้ถูกนำมาใช้งาน ดังนั้นในการปรับปรุงจึงสามารถกระทำได้คือ ให้ทำการ Convolution สัญญาณ โดยการเลื่อนตำแหน่งที่ละ 2 แทนขบวนการเดิม ก็จะได้ผลลัพธ์เช่นเดียวกับวิธีการตามปกติ แต่จะใช้เวลาในการคำนวณลดลงครึ่งหนึ่ง ดังรูปที่ 2.16

แต่การทำตามวิธีที่กล่าวมานี้ เมื่อนำมาประยุกต์ใช้ในตัวประมวลผลสัญญาณเชิงเลข (DSP Processor) แล้ว อาจทำได้ไม่สะดวกนัก เนื่องจากชุดคำสั่งที่มีในตัวประมวลผลสัญญาณเชิงเลข ส่วนใหญ่จะถูกออกแบบไว้สำหรับการทำ Convolution ตามปกติ

การแปลงกลับเวฟเลต ทำเช่นเดียวกับการแปลงเวฟเลต ซึ่งจะเป็นพื้นฐาน คือ การทำ Convolution และการทำ Up Sampling จาก Wavelet Coefficient (c) ที่เข้ามา จะถูก Up Sampling โดยการ เติมข้อมูล "0" แทรกเข้าไป จากนั้นจึงทำการ Convolution กับ ส่วนกลับของ Wavelet Function ซึ่งขบวนการนี้สามารถได้ดังรูปที่ 2.16



รูปที่ 2.16 การกลับเวฟเลตของค่าสัมประสิทธิ์ C

จากรูปที่ 2.16 สามารถเขียนเป็นสมการได้ดังนี้

$$X_0 = C_1 G_0$$

$$X_1 = C_1 G_1 + 0G_0$$

$$X_2 = C_1 G_2 + 0G_1 + C_3 G_0$$

$$X_3 = C_1 G_3 + 0G_2 + C_3 G_1 + 0G_0$$

$$X_4 = 0G_3 + C_1 G_2 + 0G_1 + C_3 G_0$$

$$X_5 = C_3 G_3 + 0G_2 + C_5 G_1 + 0G_0$$

ซึ่งจะมีการคูณค่าของ Wavelet Function กับข้อมูล “0” ซึ่งเป็นสิ่งไม่จำเป็น ดังนั้นจึงตัด

ออกไปเป็น

$$X_0 = C_1 G_0$$

$$X_1 = C_1 G_1$$

$$X_2 = C_1 G_2 + C_3 G_0$$

$$X_3 = C_1 G_3 + C_3 G_1$$

$$X_4 = C_1 G_2 + C_3 G_0$$

$$X_5 = C_3 G_3 + C_5 G_1$$

ซึ่งคือ การ Convolution ของ Wavelet Coefficient กับ Wavelet Function ตำแหน่งคู่ (G_0, G_2) สำหรับตำแหน่งของ x เป็นคู่ (2_n) และที่ตำแหน่งของ x ที่เป็นคี่ ก็จะเป็นการ Convolution ของ Wavelet Coefficient กับ Wavelet Function ตำแหน่งคี่ (G_1, G_3) แล้วจึงนำข้อมูลทั้งสองส่วนนี้ มาจัดเรียงเข้าด้วยกัน

2.5 ตัวประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)

ใช้บอร์ด DSK (DSP Starter Kit) TMS320C31 เป็นไมโครโปรเซสเซอร์ที่ต้องใช้กับชุดคำสั่งพิเศษและสถาปัตยกรรมเหมาะกับการประมวลผลสัญญาณเชิงเลข

คุณสมบัติของบอร์ด DSK

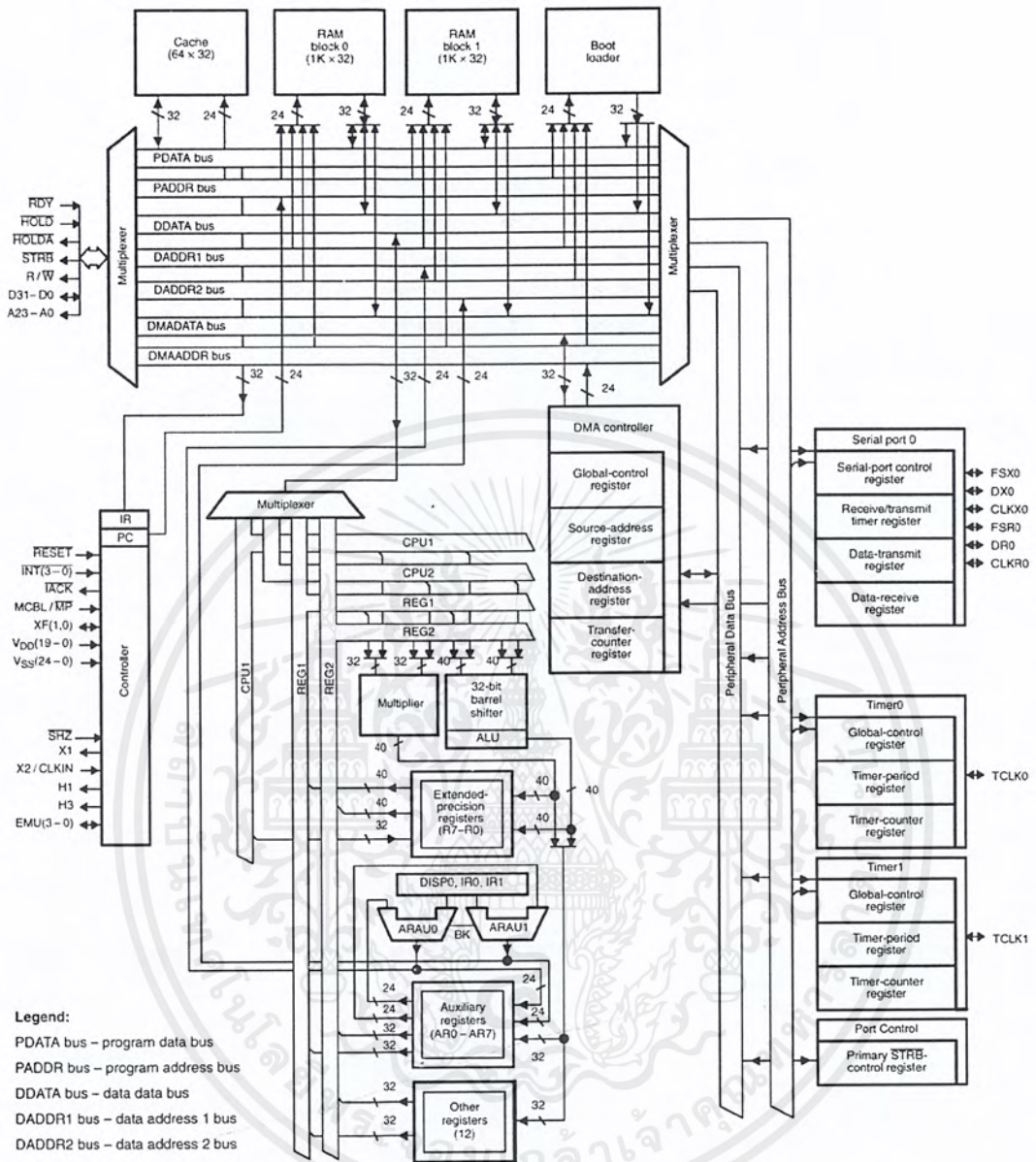
1. TMS320C31 เป็นตัวประมวลผลแบบ Floating-Point จะประกอบไปด้วย AIC ย่อมาจาก Analog Interface Circuit Chip ซึ่งมีไว้สำหรับทำการโปรแกรม ADC (ตัวเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก) กับ DAC (ตัวเปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล) และอินพุตและเอาต์พุต พัลลิตอร์, สัญญาณทั้งหมดบนซีพียู
2. บอร์ด DSK จะต่อพอร์ตนาน (เครื่องพิมพ์) เพื่อติดต่อกับคอมพิวเตอร์
3. สามารถใช้งานร่วมกับออสซิลโลสโคป, เครื่องกำเนิดสัญญาณ, ลำโพงและ Signal/Spectrum Analyzer ได้
4. สามารถโปรแกรมด้วยภาษาแอสเซมบลีและภาษาซีได้
5. ใช้กับแหล่งจ่ายไฟตรง 7.5-12 V

2.5.1 สถาปัตยกรรมและหน่วยความจำ TMS320C31

TMS320C31 มีหน่วยความจำของเวิร์ด (Words) เท่ากับ 2K (32 Bit) จากหน่วยความจำภายในและมี 16 ลائنเวิร์ดสำหรับตำแหน่งหน่วยความจำภายนอก ซึ่งประกอบไปด้วย ส่วนของโปรแกรม, ส่วนของข้อมูลและ Input/Output

TMS320C31 เป็นสมาชิกในตระกูล C3x เป็น Floating-Point (จุดทศนิยม) มีขนาด 32 บิต โดยมีหน่วยความจำสามารถแยกได้ดังนี้ ส่วนโปรแกรม Boot-Loader (ส่วนของการเริ่มต้นโปรแกรม), หน่วยความจำภายใน 2K (RAM) ในสถาปัตยกรรมสามารถแบ่งได้ดังนี้

1. Arithmetic Logic Unit (ALU)
2. 32-Bit Barrel Shifter
3. Internal Buses (CPU1/CPU2 and REG1/REG2)
4. Auxiliary Register Arithmetic Units (ARAUs)
5. CPU Register File



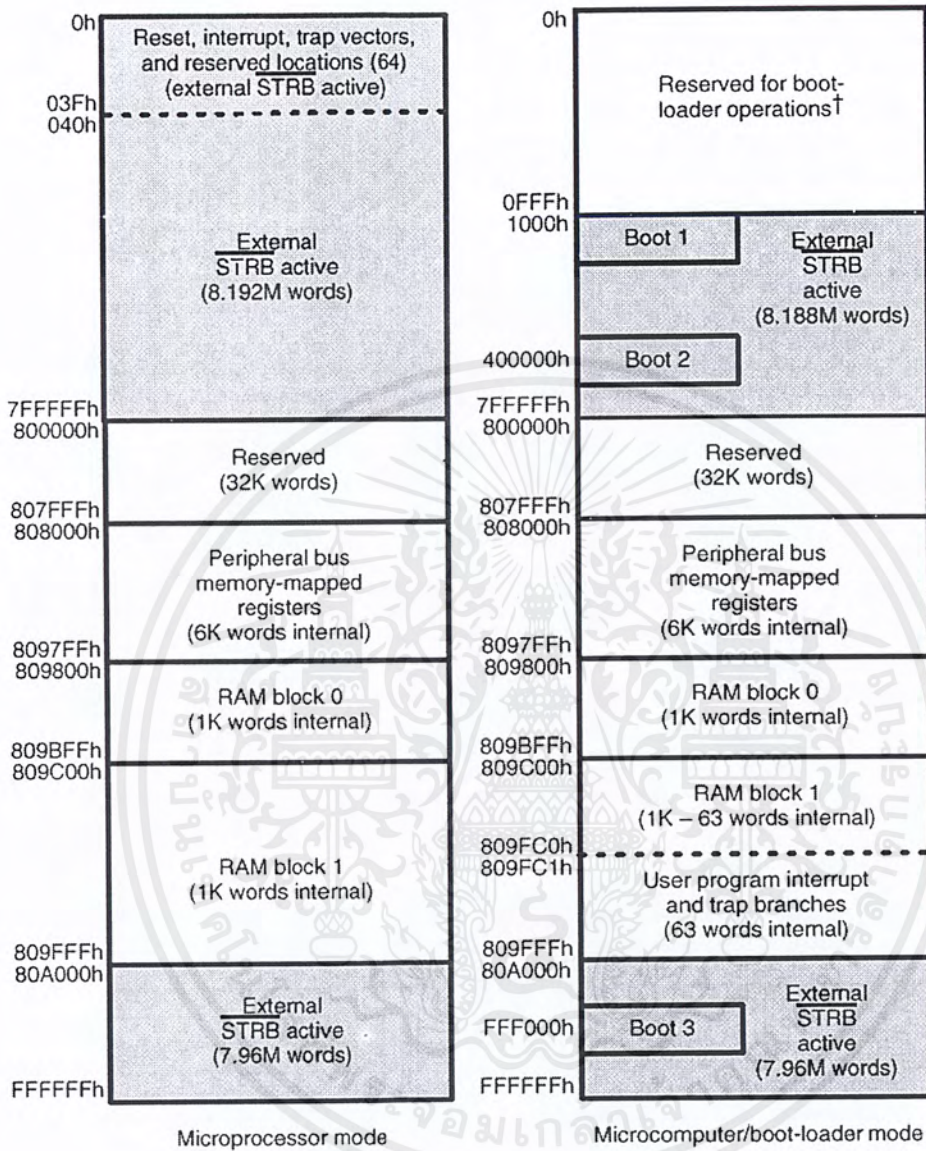
รูปที่ 2.17 สถาปัตยกรรม TMS320C31

จำนวนเต็ม (Fixed-Point) มีขนาด 24 บิตและค่าทศนิยม (Floating-Point) มีขนาด 32 บิต สำหรับ Floating-Point และ Fixed-Point จะมีความเร็ว 33 ns ต่อคำสั่งหนึ่ง Cycle, การคูณและ ALU จะคำนวณใน Cycle เดียว เมื่อไรมีการกระทำ Multiplier โดย Floating-Point Multiplication, Inputs มีขนาด 32 บิตแต่ผลลัพธ์มีขนาด 40 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 รีจิสเตอร์

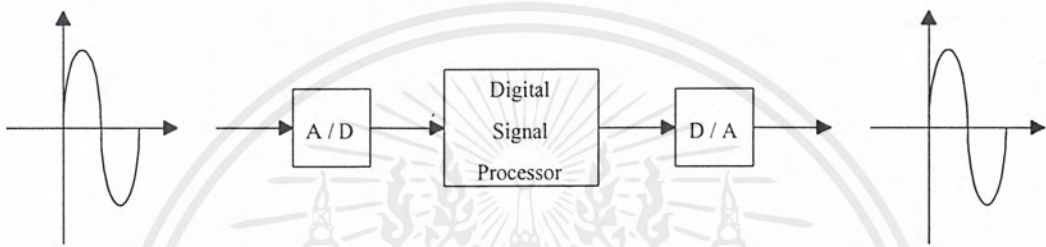
1. R0-R7 มีทั้งหมด 8 ตัว เป็นรีจิสเตอร์ขนาด 40 บิต สามารถเก็บผลลัพธ์ที่เป็นจำนวนเต็ม 32 บิต และสำหรับตัวเลขทศนิยมเก็บ 40 บิต
2. AR0 - AR7 เป็นรีจิสเตอร์ช่วย
3. IR0 และ IR1 ใช้สำหรับชี้ตำแหน่งหน่วยความจำ
4. ST ใช้บอกสถานะของ CPU
5. SP เป็น Stack Pointer
6. BK (Specify the Book)
7. IE, IF, IOF คือ อินเตอร์รัปต์ Enable, อินเตอร์รัปต์ Flag, I/O Flag (IOF)
8. RC (Repeat Count) ใช้เกี่ยวกับTime
9. RS และ RE จะประกอบไปด้วยตำแหน่งเริ่มต้นกับตำแหน่งสุดท้าย
10. PC (Program Counter)



รูปที่ 2.18 ตำแหน่งหน่วยความจำ

2.5.3 การติดต่ออินพุตและเอาต์พุตของ DSK ชั้นพื้นฐาน

การติดต่ออินพุตและเอาต์พุตของ DSK ชั้นพื้นฐานแสดงในรูปที่ 2.18 ซึ่งประกอบไปด้วย แอนาล็อกอินพุตและแอนาล็อกเอาต์พุต ในส่วนอินพุตคือ Antialiasing Filter ซึ่งจะมี A/D คือวงจรแปลงแอนาล็อกเป็นดิจิตอลแล้วจึงจะนำสัญญาณที่ได้ไปประมวลผลโดยวงจรประมวลผลสัญญาณเชิงเลข ซึ่งระดับสูงสุดของสัญญาณอินพุตจะถูกจำกัดด้วยตัว A/D ดังแสดงในรูป 2.17



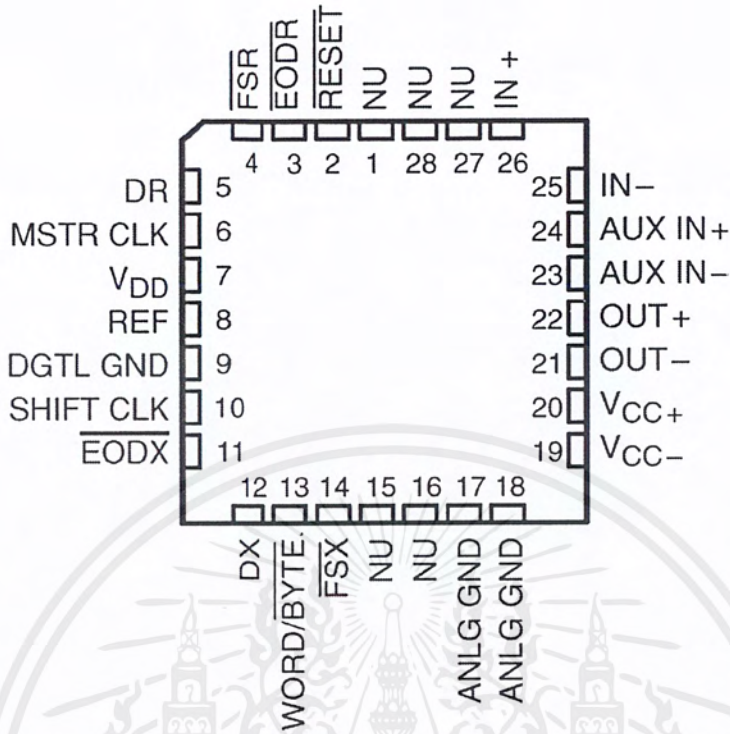
รูปที่ 2.19 อินพุตและเอาต์พุตของระบบ

หลังจากสัญญาณผ่านการประมวลผลแล้ว ผลลัพธ์ที่ต้องการจะต้องผ่าน D/A คือวงจรแปลงดิจิตอลเป็นแอนาล็อก

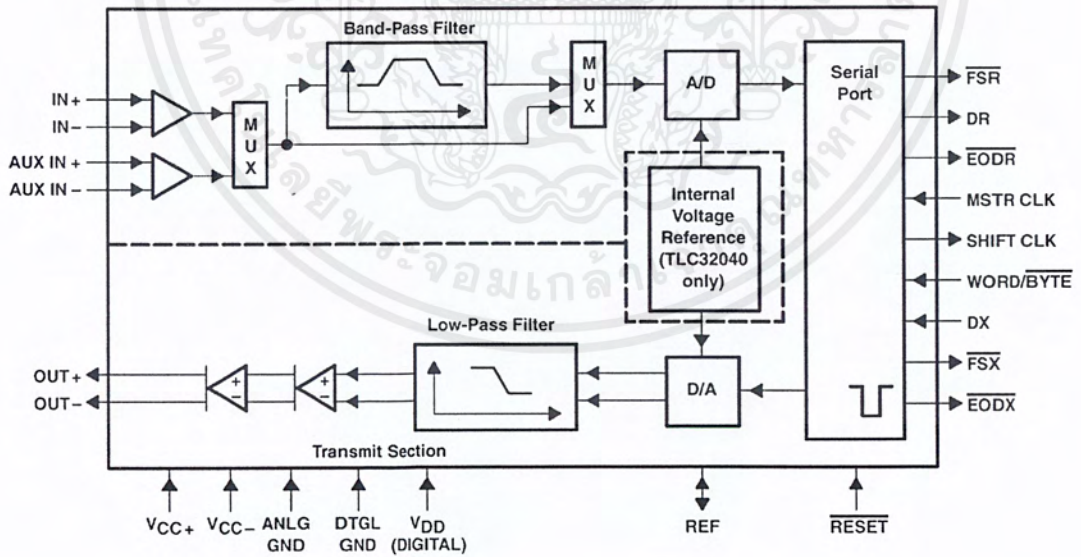
2.5.4 AIC (Analog Interface Circuit)

มีคุณสมบัติ ดังนี้

- 1) มีขนาด 14 บิต Dynamic Range ADC และ DAC
- 2) ADC และ DAC มีอัตราการสุ่ม 19,200 Samples ต่อวินาที
- 3) มี Switched-Capacitor Antialiasing เป็น Input Filter และ Output Filter
- 4) เป็นชุด Serial Port สำหรับต่อโดยตรง เพื่อนำไปประมวลผลสัญญาณเชิงเลข
- 5) มี Synchronous และ Asynchronous ADC และ DAC Convert
- 6) เป็นชุด Port ส่วนติดต่อ SN74299
- 7) เป็นเทคโนโลยี CMOS ซึ่งแสดงในรูปที่ 2.20



รูปที่ 2.20 ตัวถังชิพ (AIC)



รูปที่ 2.21 การทำงานภายในของ AIC ชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ขาต่างๆ ของชิพ AIC

ชื่อ	หน้าที่
ANLG GND	อยู่ที่ ขา 17, 18 เป็น Analog Ground อยู่ ภายใน Analog Circuits จะไม่เชื่อมต่อกับ DGTL GND
AUX IN+	อยู่ที่ขา 24 เป็น Noninverting Auxiliary Analog Input สามารถเปลี่ยนค่าในการกรองความถี่แบบ Band Pass และ A /D
AUX IN-	อยู่ที่ขา 23 เป็น Inverting Auxiliary Analog Input (อยู่ข้างบน AUX IN +)
DGTL GND	อยู่ที่ขา 9 Digital Ground อยู่ ภายใน Logic Circuits ไม่ เชื่อมต่อกับ ANLG GND
DR	อยู่ที่ขา 5 เป็นเอาต์พุต DR ถูกใช้เพื่อส่งไปข้อมูลจาก ADC ไปยัง AIC
DX	อยู่ที่ขา 12 เป็นอินพุต DX ถูกใช้เพื่อรับข้อมูลจาก DAC Input และหน่วยควบคุม ไปยัง AIC ทาง port
EODR	อยู่ที่ขา 3 เป็นเอาต์พุต เพื่อแสดงการจบของข้อมูลและเป็นชุด Port ของ Timing Diagrams ระหว่างโหมดค่า
EODR	ไบต์ต่ำหลังไบต์แรกเป็น Transmitted จาก AIC สู่ TMS320C31 เป็นชุด Port และ Kept เป็นลอจิก LOW กระทั่งไบต์ที่สองได้เป็น Transmitted TMS320C3 เพื่อบอกความแตกต่างระหว่างไบต์สอง ซึ่งก็คือวินาที
EODR	จะปรากฏ หลัง Secondary ของการสื่อสาร
EODX	อยู่ที่ขา 11เป็นเอาต์พุตเป็นการแสดงว่าจบของข้อมูลที่ส่งไปยังชุด Port การ Timing Diagram ระหว่าง โหมดค่าที่ Timer, EODX คือ ไบต์ต่ำ
SR 4	เป็นเอาต์พุต เป็นเฟรม sync ที่ใช้ รับข้อมูลในชุด Transmission โหมด ซึ่งถูกอ้างถึงในค่า FSR คือ Held เป็นลอจิก LOW ระหว่างบิต เมื่อเกิดการ Transmission เมื่อไร FSR เป็นไบต์ต่ำ TMS320C31 เมื่อนั้นชุด Port เริ่มต้นการรับบิตจาก AIC
FFSX	อยู่ที่ขา 14 เป็นเอาต์พุตเฟรม Sync ส่งไป เมื่อไร FSX เป็นไบต์ต่ำ เมื่อนั้นชุด Port เริ่มต้นการส่งไปกลับสู่ AIC
DX	โหมดค่าสั่งชุด Transmission ซึ่งถูกอ้างถึงในค่า FSX คือ Held ในบิตต่ำระหว่างบิต Transmission (เห็นเป็นชุด Port การเวลาและภายในการเวลา ค่าที่ตั้งไว้ Diagrams)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ) ขาต่างๆ ของชิพ AIC

IN +	อยู่ที่ขา 26 เป็นอินพุต Non Inverting Input เป็นตัวขยาย
IN-	อยู่ที่ ขา 25 เป็นอินพุต Inverting Input เป็นตัวขยาย
MS TR CLK	อยู่ที่ขา 6 เป็นอินพุต Master Clock เมื่อถูกใช้ Key Logic Signals ที่ AIC Switched-Capacitor, A/D และ D/A ค่าเวลาที่ตั้งไว้ภายใน Diagram จะให้เอาต์พุตเป็นสัญญาณ โดยความถี่สัญญาณขึ้นอยู่กับการ โปรแกรม
OUT +	อยู่ที่ขา 22 เป็นเอาต์พุต Non-Inverting ส่งออกสัญญาณเป็น Analog ส่งออก Power Amplifier
OUT -	อยู่ที่ขา 21 เป็นเอาต์พุต Inverting ส่งออกสัญญาณเป็น Analog ส่ง Power Amplifier
REF	อยู่ที่ขา 8 เป็นแรงดันอ้างอิงสำหรับ TLC32040 สำหรับ TLC32040 และ TLC32041 แรงดันภายนอก
V +	อยู่ที่ขา 20 เป็น Positive Analog จ่ายแรงดัน 5 V ถึง +5%
V -	อยู่ที่ 19 เป็น Analog จ่ายแรงดัน - 5 V ถึง +5%

2.5.5 การติดต่อระหว่าง PC (Personal Computer) กับ TMS320c31

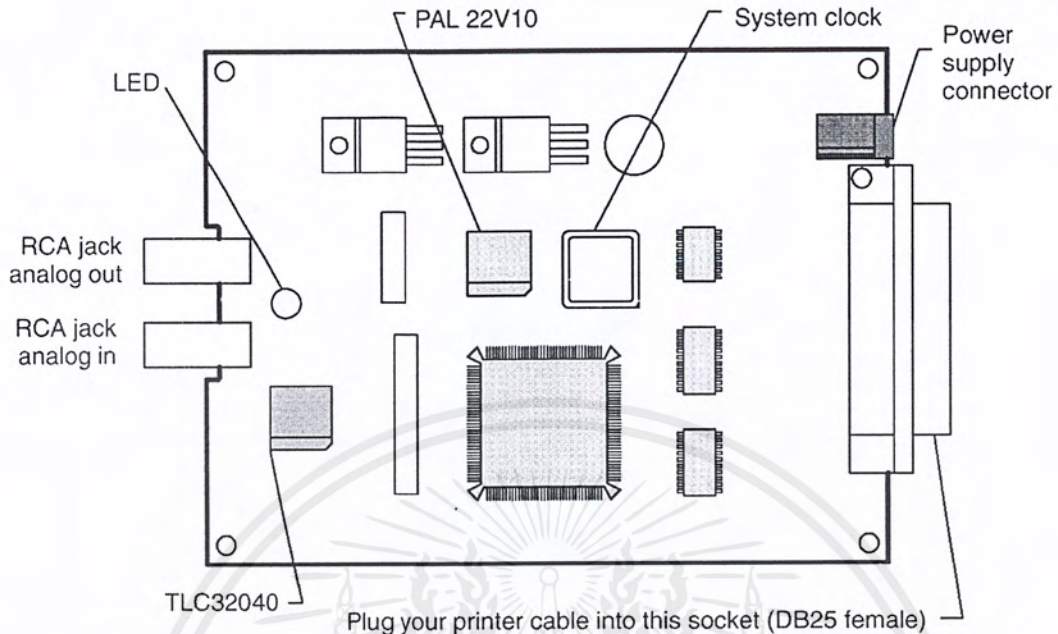
การติดต่อระหว่างคอมพิวเตอร์กับ TMS320c31 จะเริ่มต้นจากการ Load Kernel จากคอมพิวเตอร์ ผ่านพอร์ตขนาน ก่อนที่ Kernel จะ Load TMS320c31 ต้องการการรีเซ็ตจากคอมพิวเตอร์ ผ่านพอร์ตขนาน โดยในการ Boot (การเริ่มต้น) จะโหลดจากตำแหน่งเริ่มต้น 0xFFFF000

การเชื่อมต่อ บอร์ด DSK กับ PC

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับ DSK บอร์ดต้องใช้ในการเชื่อมต่อพอร์ตขนาน (เครื่องพิมพ์)

ขั้นตอน 1

1. ปิดสวิตซ์เครื่องคอมพิวเตอร์
2. เชื่อมต่อ Parallel Port (เครื่องคอมพิวเตอร์) ตู่ Parallel Communication Port (DSK บอร์ด)
3. ต่อแหล่งจ่ายไฟตรง 7 – 12 Vdc หรือ 6 – 9 Vac



รูปที่ 2.22 การเชื่อมต่อ Parallel Port เข้าไป DSK บอร์ด

4. เสียบแหล่งจ่ายเข้าที่รูเสียบ (Jack)
5. เปิดเครื่องคอมพิวเตอร์
6. LED เกิดแสงสว่างทั้งสีแดงหรือสีเขียว

ขั้นตอนที่ 2 : การติดตั้ง DSK ซอฟต์แวร์

ส่วนนี้อธิบายติดตั้ง Debugger ซอฟต์แวร์

1. สร้างไดเรกทอรี ชื่อ Dsktools เอกสารจะบรรจุ อยู่ใน DSK Assembler และซอฟต์แวร์ Debugger

2. ทำการ Copy โปรแกรม

ขั้นตอน 3 : แก้ไขเพิ่ม Config.sys

เมื่อเริ่มการใช้ Debugger ต้องเกิดการแก้ไข Config.sys และเพิ่ม file

ขั้นตอน 4 : การแก้ไข PATH Statement

ทำให้แน่ใจ Debugger และ Assembler เป็นอ้างอิง Directory ในคอมพิวเตอร์ต้องแก้ไข PATH statement เพื่อค้นหาไดเรกทอรีของ Dsktools เช่น

```
PATH=C:\dsktools; pathname2; pathname3
```

การติดตั้ง DSK Environment บน DOS Command (ตัวอย่าง autoexec.bat File)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


PATH statement →

```

DATE
TIME
ECHO OFF
PATH=c:\dos;c:\dsktools
CLS

```

ถ้าแก้ไข autoexec.bat

autoexe 

ขั้นตอน 5 : ทดสอบการติดตั้ง

เพื่อให้แน่ใจในการติดตั้ง DSK บอร์ด, Assembler, และ Debugger, ใ้คำสั่งต่อไปนี้ที่ระบบพร้อมเริ่มต้น DSK

debugger:

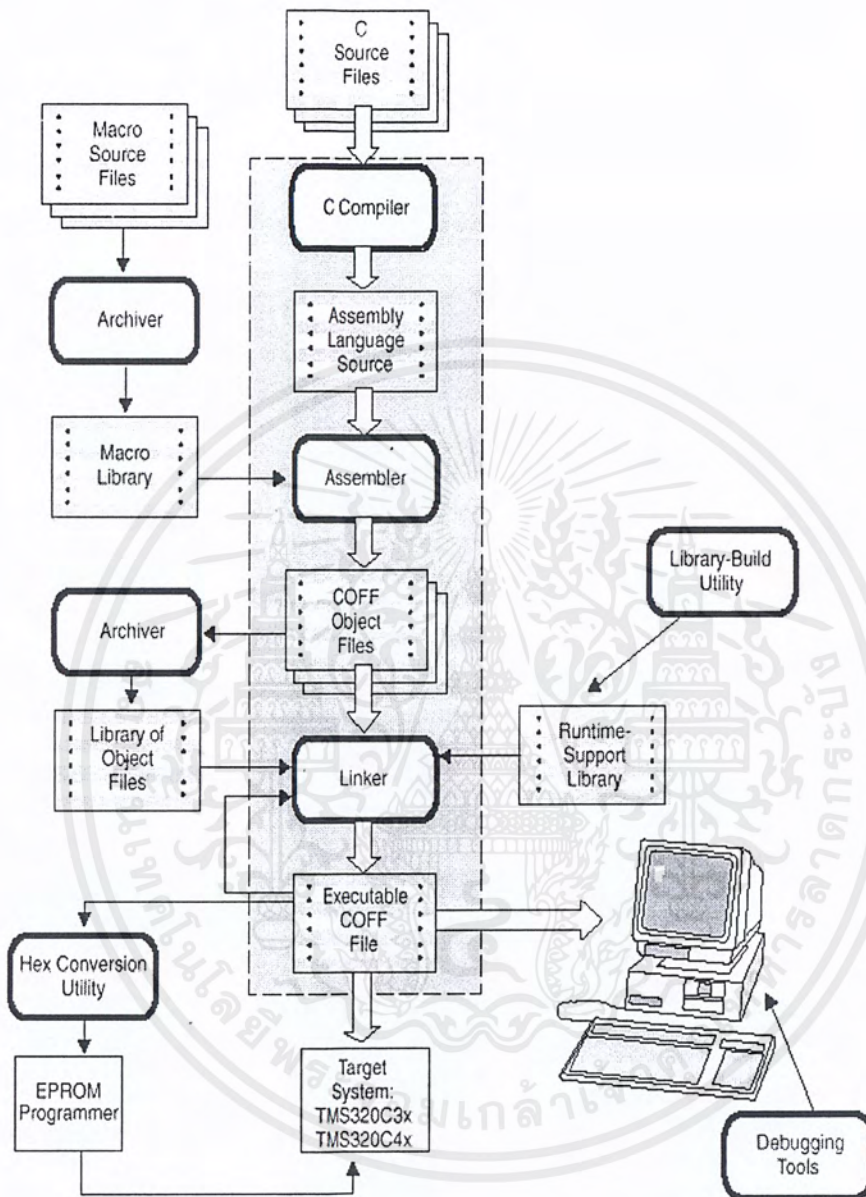
dsk3d

หลังจากใ้คำสั่ง dsk3d จะเห็นหน้าจอดังรูปที่ 2.22

DISASSEMBLY				C31 DSP STARTERS KIT			
809c03	50700080	start	LDIU 00080h,DP	PC	00809c03	SP	008098de
809c04	08349c2c		LDI @09c2cH,SP	R0	00000000	R1	00000000
809c05	07608000		LDF 0.000000e+00,R0	R2	00000000	R3	00000000
809c06	c610c1c0		LDI *AR0,R0 LDI *AR	R4	00000000	R5	00000000
809c07	c610c1c0		LDI *AR0,R0 LDI *AR	R6	00000000	R7	00000000
809c08	08600100		LDI 256,R0	AR0	00000000	AR1	00000000
809c09	09a09c00		LSH @09c00H,R0	AR2	00000000	AR3	00000000
809c0a	61809c0e		BRD jump	AR4	00000000	AR5	00000000
809c0b	07618000		LDF 0.000000e+00,R1	AR6	00000000	AR7	00000000
809c0c	07628000		LDF 0.000000e+00,R2	IR0	00000000	IR1	00000000
809c0d	07630000		LDF 1.000000e+00,R3	ST	00000000	RC	00000000
809c0e	07640000	jump	LDF 1.000000e+00,R4	RS	00000000	RE	00000000
809c0f	087b0003	loop	LDI 3,RC	DP	00000000	BK	00000000
809c10	64809c1a		RPTB block	IE	00000000	IF	00000000
809c11	02640001		ADDI 1,R4				
COMMAND				MEMORY			
Texas Instruments 1994				809800	00000007	fffffffc	00809802 00809827
				809804	0080982c	00809839 0080983c	0080983f
				809808	00809843	00809842 00809868	0080989a
				80980c	008098a9	10800000 0f350000	0f300000
				809810	0f200000	0f320000 0f280000	0f290000
				809814	1a770004	6a050006 628098a9	50700080
load testa							

รูปที่ 2.23 พื้นฐานหน้าจอ Debugger

2.5.6 เครื่องซอฟต์แวร์ Development Tools



รูปที่ 2.24 ส่วนประกอบ TMS320C3x/C4x เกี่ยวกับซอฟต์แวร์ Development

เครื่องมือ ที่แสดงในรูปที่ 2.23

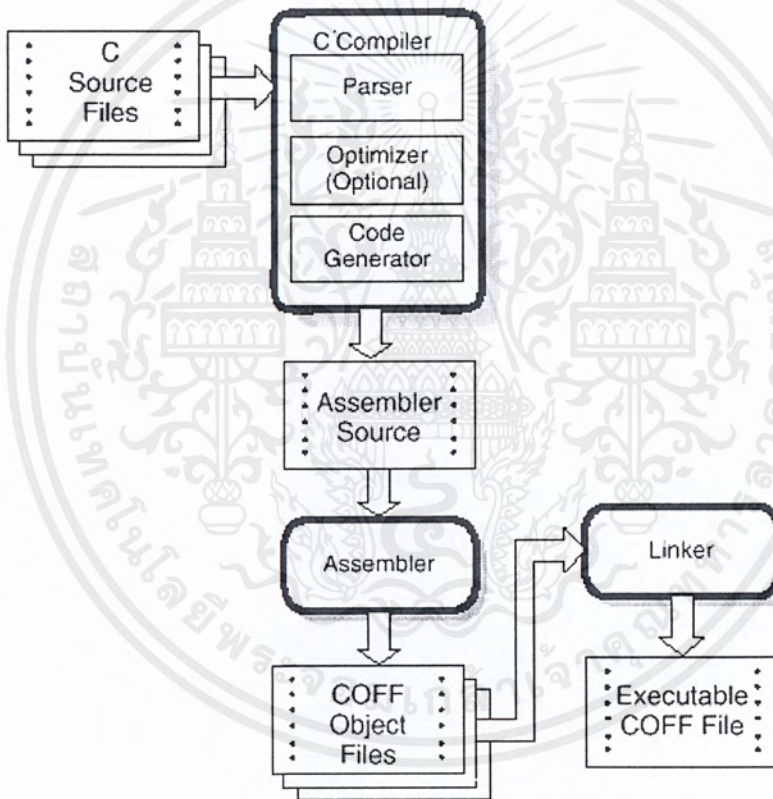
- C Compiler จะเป็นข้อมูลรหัสซึ่งจะใช้กับ TMS320C3x หรือ TMS320C4x หรือภาษา Assembly โปรแกรม Shell (cl30), Optimizer (opt30), Interlist (clist) จะอยู่ใน ชุด C Compiler
- โปรแกรม Shell เป็นโปรแกรมช่วยให้ตรวจสอบโดยอัตโนมัติ, Assemble และ Link Source Modules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Optimizer แก้ไขปรับปรุงโปรแกรมภาษา C
- Interlist เครื่องมือ Interlists C จะแปลงข้อมูลประโยคไปเป็นภาษา Assembly
- Assembler จะแปล Assembly ภาษาข้อมูลเข้าไปในเครื่อง

2.5.7 Compiling C Code

cl30 เป็น โปรแกรม shell คือเครื่องมือใช้ Compile, Assemble และ ข้อกำหนดเพิ่มเติมการเชื่อมต่อ Compiler จะประกอบด้วย Parser, Optimizer และ Generator Code, Assembler ซึ่งทำให้เกิด แฟ้ม COFF



รูปที่ 2.25 การใช้ C Compiler

Linker สามารถ Invoked (อ้าง) เป็นส่วนของกระบวนการ Larger หรือสามารถ Compiler และ Assembler Invoking ของ C Compiler เพื่อใช้ในการ Compiler ได้

```
cl30 [-options] [filenames] [-z [link_options]] [object files]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cl30 คือ คำสั่งในการ Invokes ของ Compiler

Options คือ ข้อกำหนดที่มีผลต่อวิธี Compiler ของ กระบวนการ Input Files

-z คือ ข้อกำหนดให้ Linker เป็น หน่วยควบคุมข้อกำหนดทางเชื่อมต่อ เช่น

> Cl30 symtab

เมื่อ Enter แล้วจะเกิดการ Compiler จะข้อความด้านล่าง

[symtab]

TMS320C3x/4x ANSI C Compiler Version x.xx

Copyright (c) 1987–1997, Texas Instruments Incorporated

"symtab.c":==> main

"symtab.c":==> lookup

TMS320C3x/4x ANSI C Codegen Version x.xx

Copyright (c) 1987–1997, Texas Instruments Incorporated

"symtab.c":==> main

"symtab.c":==> lookup

TMS320C3x/4x COFF Assembler Version x.xx

Copyright (c) 1987–1997, Texas Instruments Incorporated

PASS 1

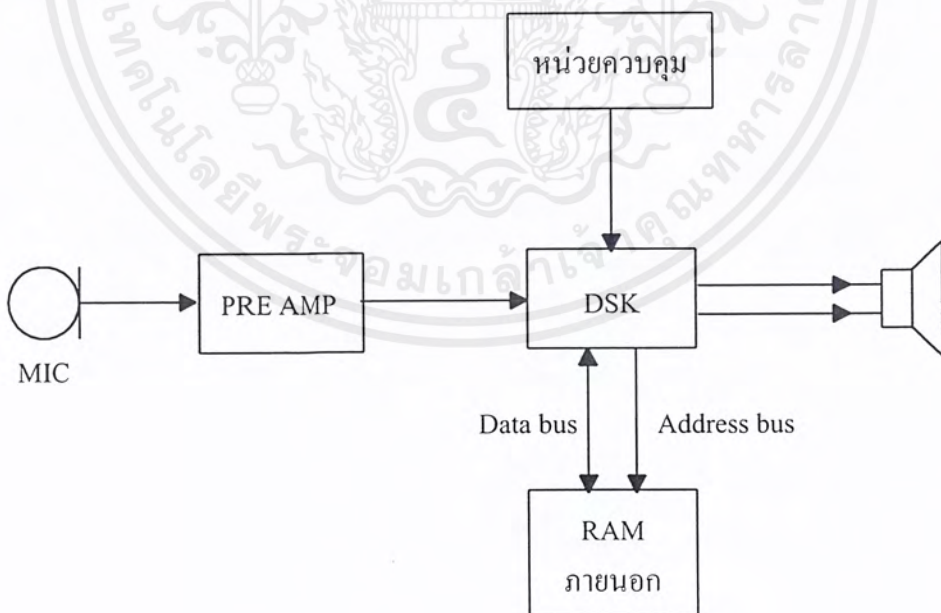
PASS 2

No Errors, No Warnings

บทที่ 3

การออกแบบ การสร้างและการทำงาน

โครงการเรื่องการบีบอัดข้อมูลเสียงพูดด้วยเวฟเล็ตได้เลือกใช้บอร์ด DSK (Dsp Starter Kit) TMS320C31 ในการประมวลผลสัญญาณเสียงต้องการตัวประมวลผลที่มีความเร็วสูง จึงเป็นเหตุผลที่เลือกใช้ TMS320C31 ซึ่งมีจุดเด่นอยู่หลายประการ เช่น เป็นตัวประมวลผลที่มีความเร็วสูง, สามารถคำนวณในระดับจุดทศนิยม, สามารถประมวลผลสัญญาณที่เวลาจริง (Real Time) และยังสามารถพัฒนาโปรแกรมบน PC (Personal Computer) แล้วส่งไปยังบอร์ด DSK จึงสามารถแก้ไขและพัฒนาโปรแกรมได้อย่างสะดวกและใช้เวลาไม่มาก ซึ่งโครงการนี้ประกอบไปด้วย ส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ ส่วนของซอฟต์แวร์ประกอบไปด้วย การแปลงเวฟเล็ต, การแปลงกลับเวฟเล็ต, การติดต่ออินพุต (Input) และเอาต์พุต (Output) ของ DSK, โปรแกรมควบคุมการทำงานของบอร์ด DSK ส่วนของฮาร์ดแวร์ประกอบไปด้วยวงจร PRE AMP (Pre Amplifier), การต่อหน่วยความจำภายนอก (RAM) และการต่อสวิทช์เพื่อควบคุมการทำงาน โดยการทำงานของระบบจะแสดงดังรูปที่ 3.1



รูปที่ 3.1 ระบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแรกของระบบการทำงานคือรับอินพุตสัญญาณเสียงพูดจาก MIC (Microphone) ซึ่ง MIC จะทำหน้าที่เปลี่ยนสัญญาณเสียงที่ได้ให้เป็นสัญญาณไฟฟ้า จากนั้นทำการขยายโดย PRE AMP ซึ่งทำหน้าที่ขยายสัญญาณเสียง ทำให้มีสัญญาณมีขนาดใหญ่เหมาะสมกับการประมวลผล แล้วส่งไปยัง DSK TMS320C31 หน่วยควบคุมประกอบไปด้วย 2 ส่วน คือ สวิตช์ตัวที่ 1 (Play/REC) และ สวิตช์ตัวที่ 2 (Playback) เมื่อสวิตช์ที่ 1 เลือก REC (บันทึก) จะนำข้อมูลเสียงพูดที่ได้ไปประมวลผลแปลงเวฟเล็ท (DWT) หลังจากนั้นนำข้อมูลที่ได้จากการประมวลผลบันทึกลงหน่วยความจำภายนอก (RAM) ต่อมาตรวจสอบการกดสวิตช์ตัวที่ 2 ซึ่งทำหน้าที่ ย้อนกลับไปทำการบันทึกใหม่ ถ้าเกิดการเลือก PLAY โปรแกรมจะทำการนำข้อมูลในหน่วยความจำภายนอก (RAM) ทำการประมวลผลการแปลงกลับเวฟเล็ท (IDWT) นำข้อมูลเสียงพูดที่ได้ ออกสู่ลำโพง

3.1 ส่วนของฮาร์ดแวร์

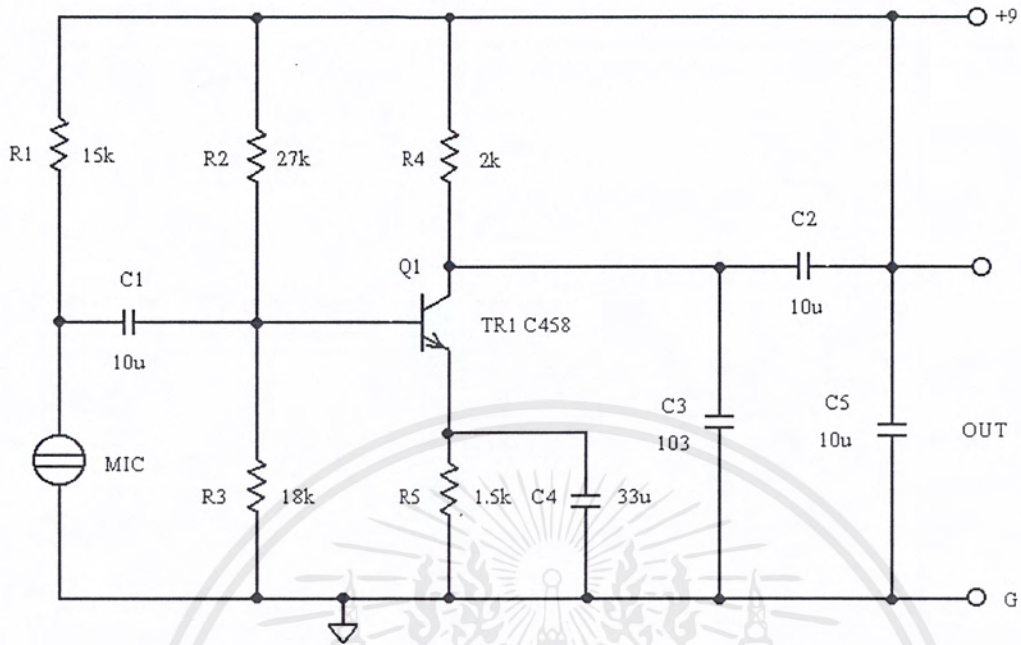
3.1.1 PRE AMP

3.1.2 การต่อหน่วยความจำภายนอก

3.1.3 วงจรควบคุมการทำงาน

3.1.1 PRE AMP

TR1 (Transistor 1) ต่อแบบวงจรขยายคลาส A สัญญาณจาก MIC จะผ่าน C1 (Capacitor1) มาเข้าขา B ของ TR1 เพื่อทำการขยายสัญญาณออกทางขา C (Collector) ของ TR1 สัญญาณที่ทำการขยายแล้วจะผ่าน C2 ไปที่ OUT R1 (Resister 1) จะมีหน้าที่ เป็น โหลดของ MIC C3 ทำหน้าที่บายพาสความถี่สูง ดังเช่นรูปที่ 3.2



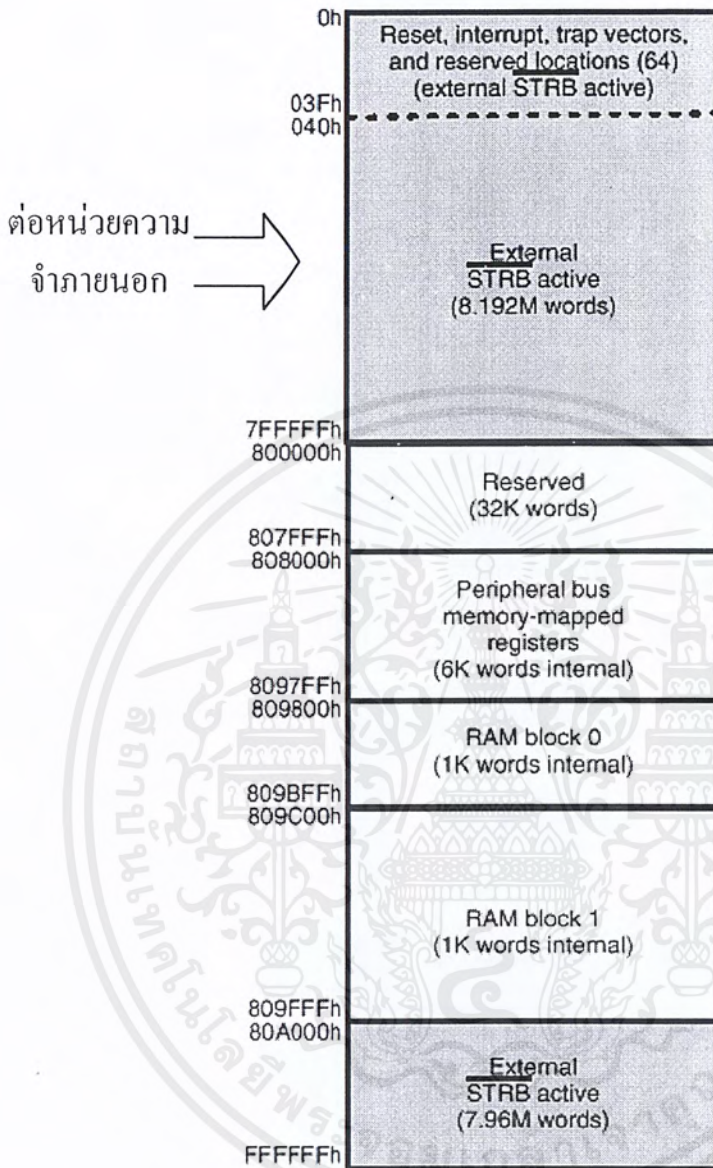
รูปที่ 3.2 วงจร PRE AMP

3.1.2 การต่อหน่วยความจำภายนอก (RAM)

ในการต่อหน่วยความจำ จำเป็นต้องใช้ขาสัญญาณของ TMS320C31 ดังนี้

1. บัสข้อมูล (Data Bus) ขนาด 32 บิต
2. บัสตำแหน่ง (Address Bus) ขนาด 24 บิต
3. ขาสัญญาณควบคุม เช่น STRB, R/W

สำหรับการต่อหน่วยความจำภายนอก จะต้องเลือกตำแหน่งหน่วยความจำที่สงวนไว้ ดังรูปที่ 3.3 ในโครงการเลือกตำแหน่งที่ 200000H โดยขนาดหน่วยความจำ 32 kword



รูปที่ 3.3 ตำแหน่งที่ต่อหน่วยความจำ

สำหรับหน่วยความจำภายนอก ใช้ไอซี 62256 มีขนาด 32 บิต ใช้ 4 ตัว โดย TMS320C31 จัดสรรชุดในการต่อหน่วยความจำภายนอก 4 ชุด ชุดละ 32 ขา นั่นก็คือ JP2-3 และ JP5-6 ซึ่งจะอยู่ที่ขอบของบอร์ด ซึ่งจะประกอบไปด้วย Address (ตำแหน่ง), Data(ข้อมูล), V+, GND, R/W, INTO-3 และ STRB ดังนี้

JP2 ขาที่ 1, 2, 31, 32 เป็นขา GND

JP2 ขาที่ 21 เป็นขา R/W

JP2 ขาที่ 22 เป็นขา STRB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

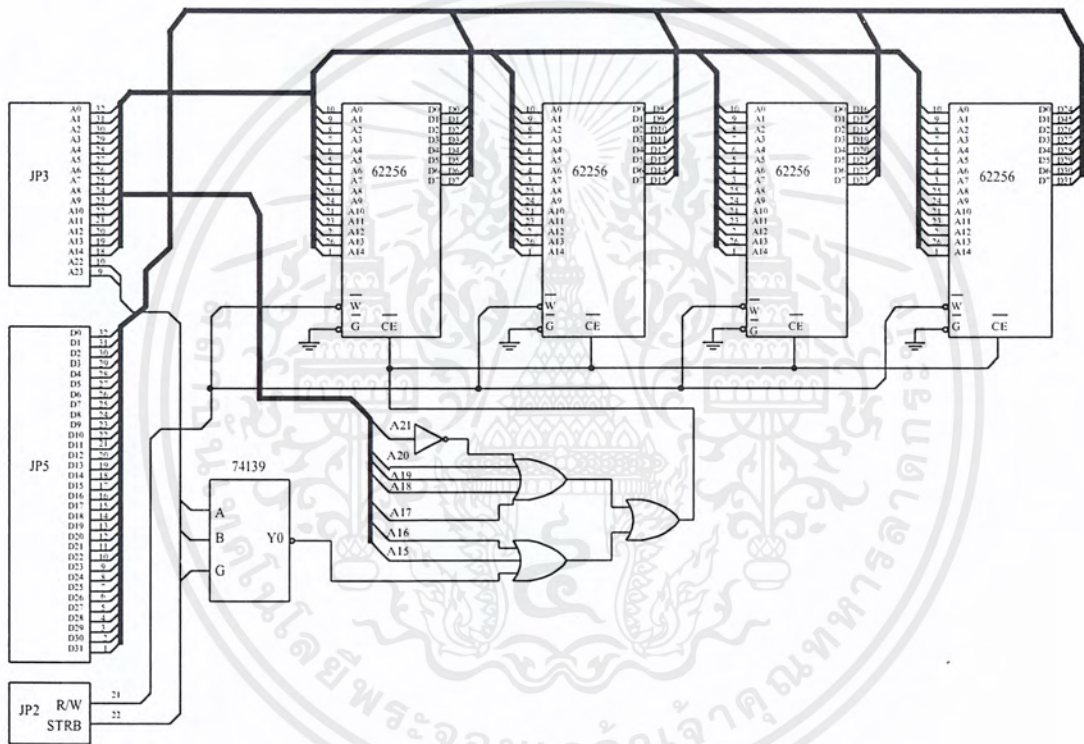
JP3 ขาที่ 9-32 เป็นขา Address A8-A23

JP5 ขาที่ 1-32 เป็นขา Data D0-D31

JP6 ขาที่ 1, 2, 31, 32 เป็นขา GND

JP6 ขาที่ 23, 24 เป็นขา V+, V-

ใช้ SRAM เบอร์ 62256 ซึ่งมีขนาด 32 K bit ใช้ 4 ตัว โดยให้ ขา Data 8 บิตนั้นก็คือ ตัวแรก ต่อ D0-D7, ตัวที่ 2 ต่อ D8-D15, ตัวที่ 3 ต่อ D16-D23, ตัวที่ 4 ต่อ D24-D31 ใช้ IC74373 Decode ที่ตำแหน่ง 200000 H ลักษณะการประกอบวงจรเป็นดังรูปที่ 3.4

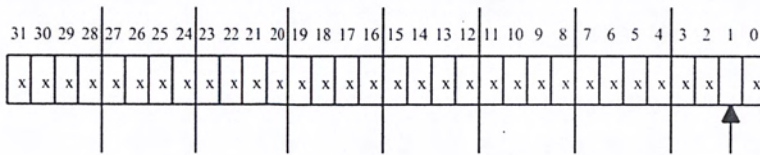


รูปที่ 3.4 การต่อวงจรหน่วยความจำภายนอก

3.1.3 วงจรควบคุมการทำงาน

การออกแบบวงจรควบคุมจะใช้สวิตช์คือให้เลือก 2 ทางคือ Play/REC ถ้าเลือก Play ทำการรับอินพุตเป็น 0 ถ้าเลือก REC ได้รับอินพุตเป็น 1 ในการดีโค้ดจะใช้ตำแหน่งหน่วยความจำที่ 400000H โดยใช้ 74373 โดยแลตค่าเอาไว้ ในการเขียนโปรแกรมจะเช็คว่าผู้ใช้งานต้องการ Play/REC ถ้าเป็น REC จะทำการ AND บิตสุดท้ายลักษณะดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 จำนวนบิต

เราสนใจ 4 บิตสุดท้าย ที่ตำแหน่งลูกศรชี้อยู่ ดังรูป เรานำมา AND กับ 0x000002 แล้วทำการเขียนโปรแกรมควบคุม



รูปที่ 3.6 วงจรควบคุมการทำงาน

3.2 ส่วนของซอฟต์แวร์

- 3.2.1 โปรแกรม DWT (Discrete Wavelet Transform)
- 3.2.2 โปรแกรม IDWT (Invert Discrete Wavelet Transform)
- 3.2.3 โปรแกรมการลดขนาดข้อมูลเสียงพูด
- 3.2.4 โปรแกรมควบคุมการทำงานบอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 โปรแกรม DWT (Discrete Wavelet Transform)

ในหัวข้อนี้เป็นการแปลงเวฟเล็ต ในการออกแบบและสร้างส่วนนี้ได้ใช้โปรแกรม Matlab V5.1 เป็นเครื่องมือ และได้เขียนโปรแกรมเลียนแบบการทำงาน ด้วยภาษาซี ซึ่งอยู่ในภาคผนวก ก การใช้ฟังก์ชันในการแปลงเวฟเล็ตของ MatLab

DWT (Discrete Wavelet Transform)

```
>x=[1,2,3,4,5]          /*Input*/
```

```
x =
```

```
1 2 3 4 5
```

```
> load db2;             /*Dabuchies 2*/
```

```
> db2
```

```
db2 =
```

```
0.3415 0.5915 0.1585 -0.0915
```

```
> [ca,cd]=dwt(x,'db2')
```

```
ca =
```

```
-0.0347 2.3108 5.9157 2.4148
```

```
cd =
```

```
-0.1294 0.0000 2.8978 -0.6470
```

สรุปขั้นตอน dwt ได้ดังนี้

1. ทำการแยกสัญญาณอินพุต และ Wavelet Function ออกเป็นข้อมูลตำแหน่งคู่กับตำแหน่ง
2. ทำการ Convolution ของสัญญาณที่แยกออกเป็นสองส่วนนี้กับเวฟเล็ตแม่ (Dabuchies 2)
3. รวมผลลัพธ์ที่ได้จากการ Convolution
4. ทำการ Down Sampling

3.2.2 โปรแกรม IDWT (Invert Discrete Wavelet Transform)

ในหัวข้อนี้เป็นการแปลงกลับเวฟเล็ต ในการทดลองส่วนนี้จะทำโดยใช้โปรแกรม Matlab V5.1 เช่นเดียวกันและได้เขียนโปรแกรมเลียนแบบการทำงาน Tool ของ Matlab ด้วยภาษาซี ซึ่งอยู่ในภาคผนวก ก

การใช้ฟังก์ชันในการแปลงเวฟเล็ตของ MatLab

IDWT (Invert Discrete Wavelet Transform)

```
> s=idwt(ca,cd,'db2')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

s =

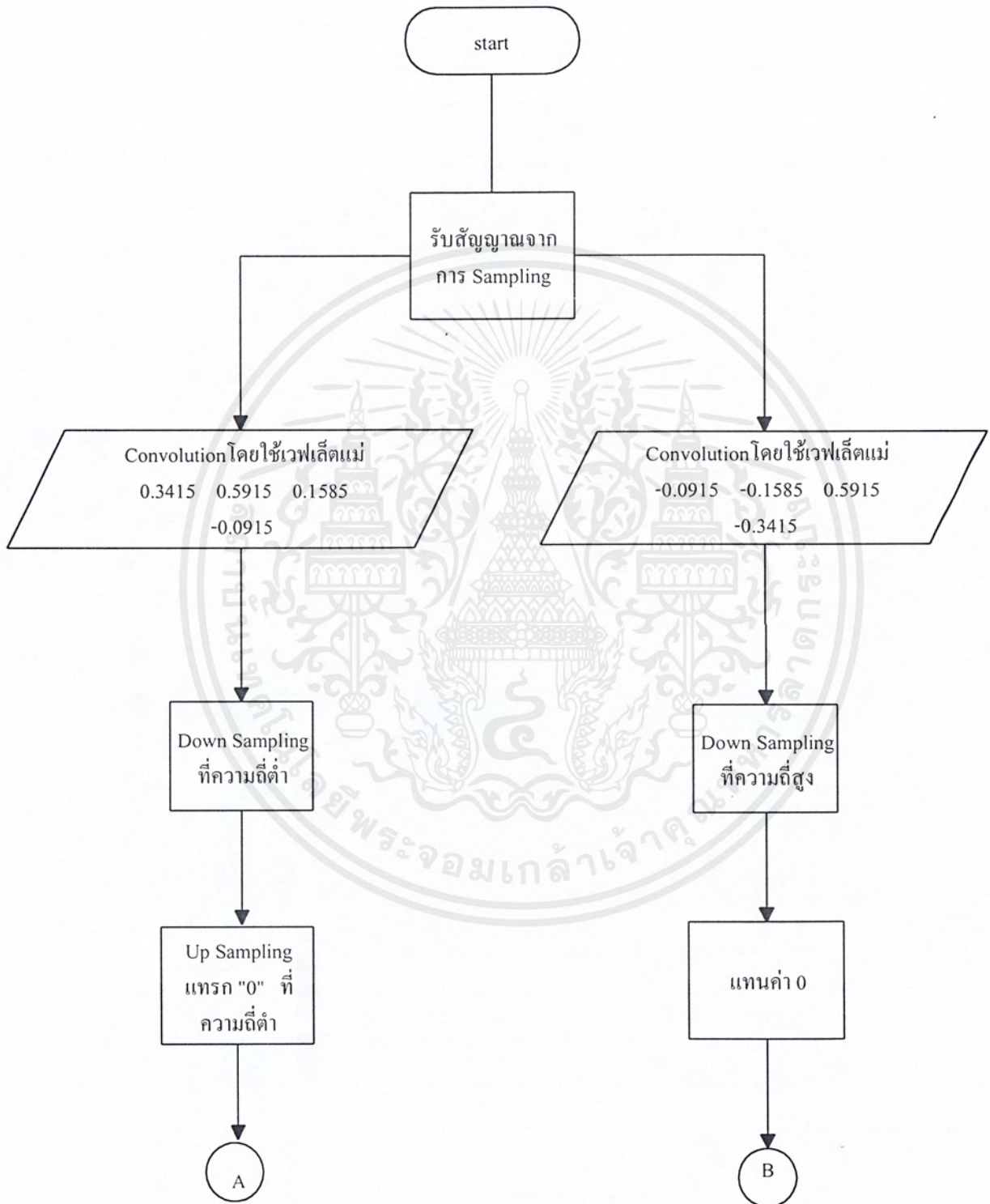
1.0000 2.0000 3.0000 4.0000 5.0000 0.0000

สรุปขั้นตอน idwt ได้ดังนี้

1. แทรกศูนย์ระหว่างค่าของข้อมูล
2. ทำการ Convolution ของ Wavelet Coefficient กับ Wavelet Function ที่แยกออกเป็นสองส่วนนี้กับเวฟเลตแม่
3. ทำการเรียงสลับของสัญญาณที่ได้จากการ Convolution ทั้งสอง

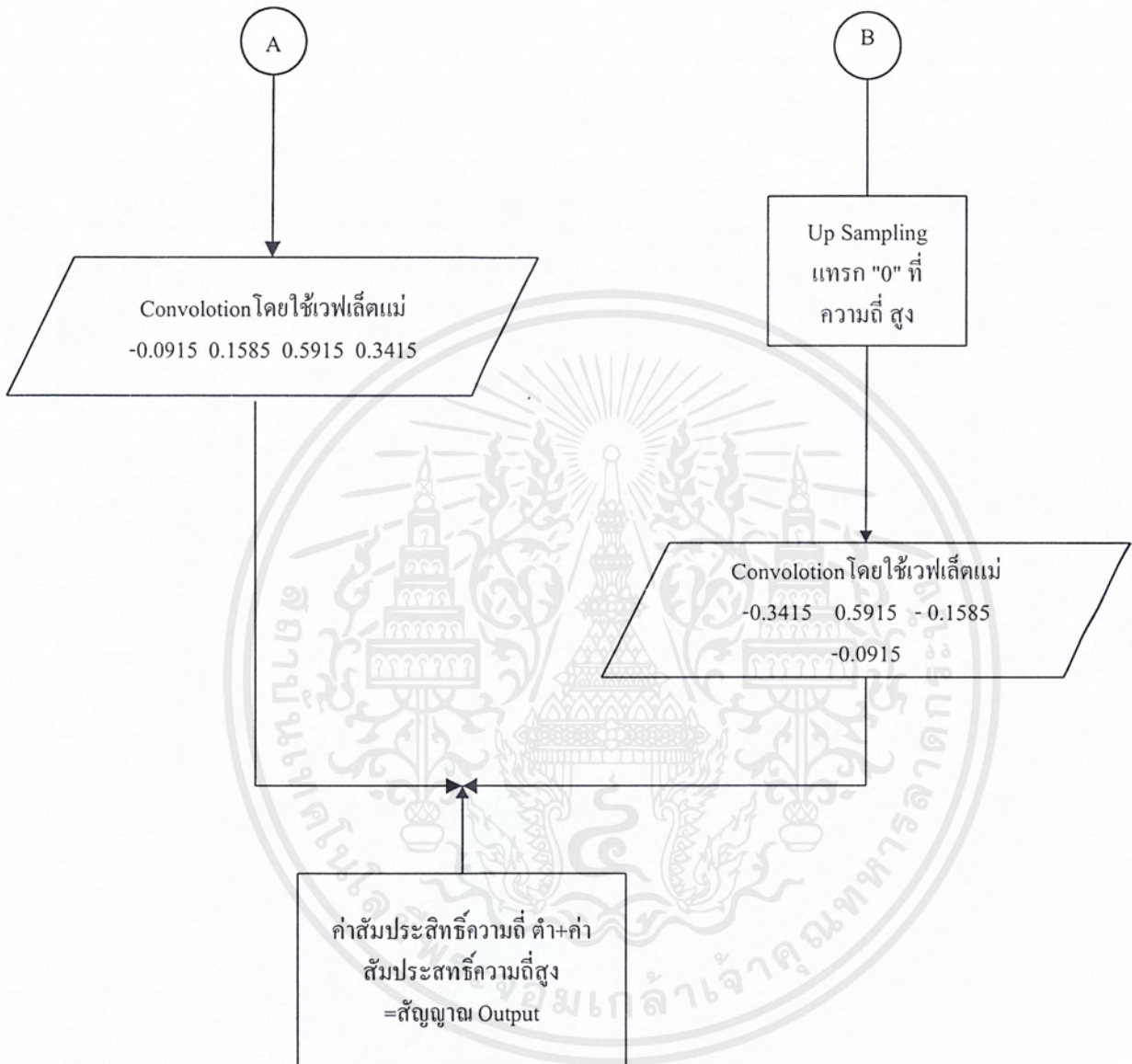
3.2.3 โปรแกรมการลดขนาดข้อมูลเสียงพูด

ในการลดขนาดข้อมูลเสียงพูดสามารถทำได้ในช่วงการแปลงเวฟเลต ผลลัพธ์จากการแปลงเวฟเลตจะได้ค่าสัมประสิทธิ์ด้านความถี่ต่ำและความถี่สูง ซึ่งค่าสัมประสิทธิ์ด้านความถี่สูงเป็นส่วนประกอบที่มีความสำคัญน้อยจึงสามารถตัดทิ้งได้ โดยใช้วิธีการของ Multi Resolution Wavelet ใน Level 1 เมื่อนำค่าสัมประสิทธิ์ที่ได้มาทำการแปลงกลับเวฟเลตทั้งด้านความถี่ต่ำและความถี่สูงมารวมกัน ซึ่งจะได้ผลลัพธ์ที่มีความเพี้ยนไม่มากสามารถยอมรับได้



รูปที่ 3.7 ขบวนการลดขนาดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 (ต่อ) ขบวนการลดขนาดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โปรแกรมควบคุมการทำงานบอร์ด DSK

การติดต่อสื่อสารระหว่างอินพุตและเอาต์พุตของ DSK จะมี AIC เป็นตัวจัดการซึ่งได้อธิบายไปแล้วในบทที่ 2 ในการติดต่อกับ AIC ต้องมีการตั้งค่าเริ่มต้นให้รีจิสเตอร์ที่เกี่ยวข้องและต้องรู้ตำแหน่งรีจิสเตอร์ต่างๆ ดังตารางที่ 3.1

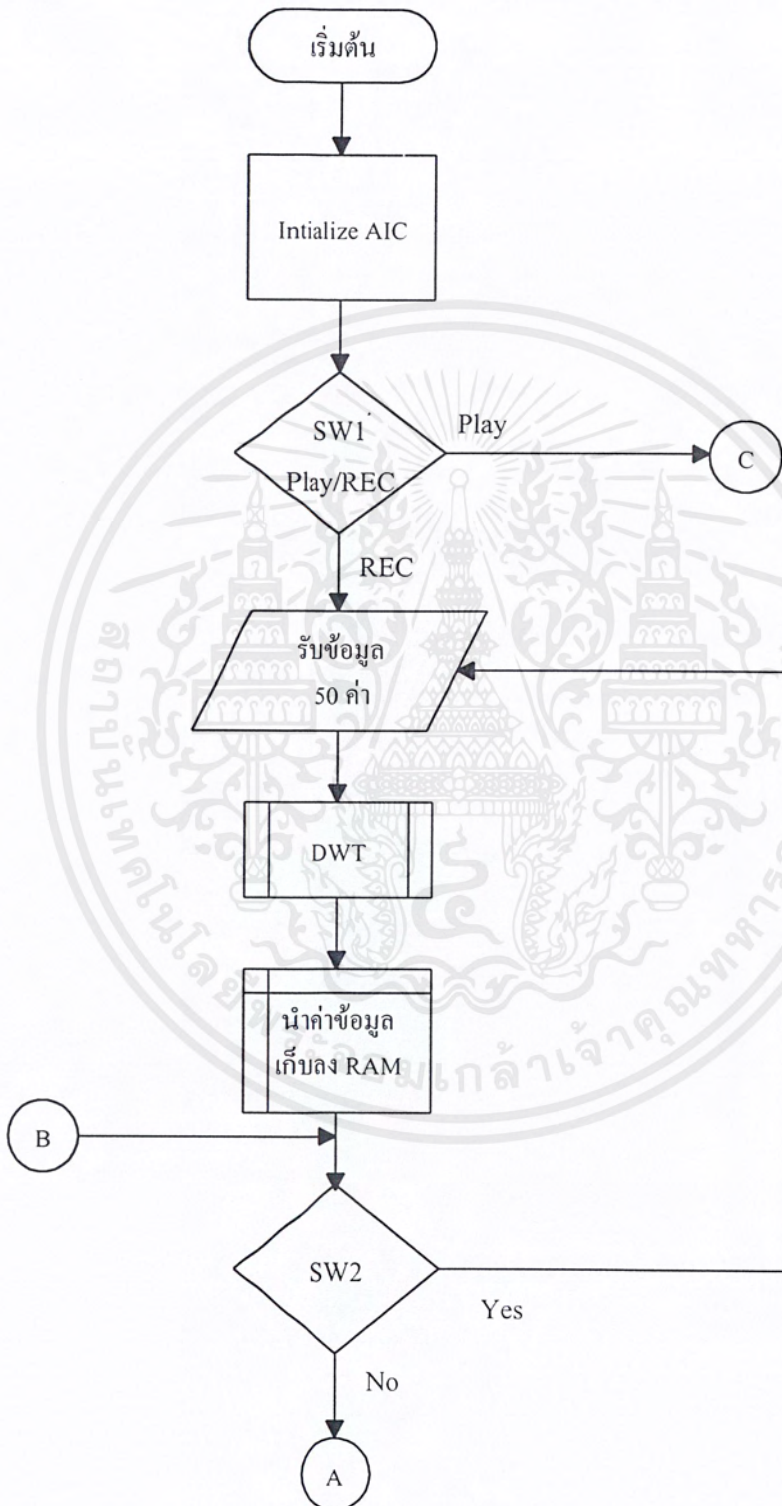
ตารางที่ 3.1 ตำแหน่งรีจิสเตอร์ต่าง ๆ

Register	Address	Command
Timer 0 period	0x808028	Load 0x1
Timer 0 global control	0x808020	Load 0x3c1
I/O flag	IOF	Load 0x2
SPO transmit port control	0x808042	Load 0x131
SPO receive port control	0x808043	Load 0x131
SPO global control	0x808040	Load 0x0E970300
SPO data transmit	0x808048	Load 0x0
I/O	IOF	Load 0x6
Interrupt flag	IF	Load 0x0
Interrupt enable	IE	OR 0x10
Status register	ST	OR 0x2000

ทำการศึกษาโปรแกรม AICCOMC.C โดยโปรแกรมนี้จะเป็นการติดต่อสื่อสารระหว่าง Input และ Output สามารถสรุปขั้นตอนได้ดังนี้

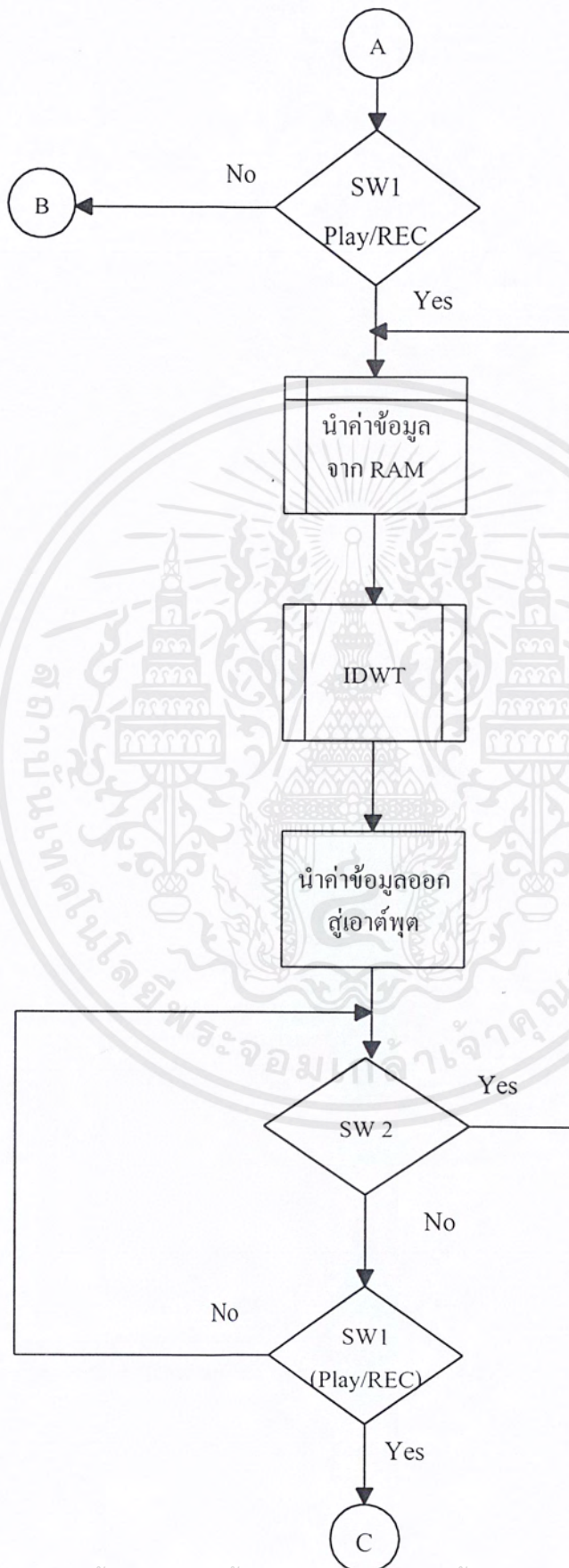
1. Set Timer Period ที่ตำแหน่ง 0x808028
2. Set รีจิสเตอร์ Timer Control ที่ตำแหน่ง 0x808020
3. Set IOF (Input / Output Flag) ให้เป็น low (ต่ำ) เพื่อไปรีเซ็ต AIC
4. Set Xmit Port Control ที่ตำแหน่ง 0x808042 (SPO Transmit Port Control)
5. Set Receive Port Control (พอร์ตรับ) ที่ตำแหน่ง 0x808043
6. Set Serial Port Global Register ที่ตำแหน่ง 0x808040

โปรแกรมควบคุมการทำงานบอร์ด DSK



รูปที่ 3.8 แผนผังโปรแกรมควบคุมการทำงานบอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.8 (ต่อ) แผนผังโปรแกรมควบคุมการทำงานบอร์ด DSK
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

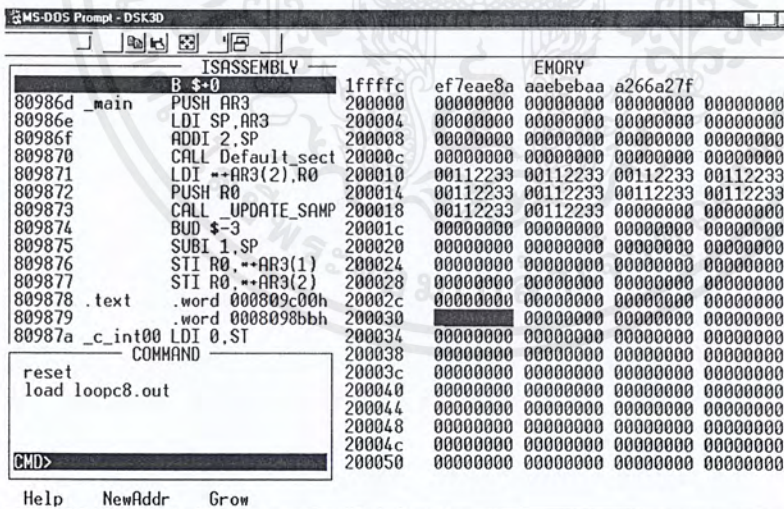
การทดลองและผลการทดลอง

โครงการนี้เป็นการบีบข้อมูลเสียงด้วยเวฟเล็ตด้วย DSP จากบทที่ 3 ได้ทำการออกแบบการทำงานของระบบ โดยโครงการนี้สามารถแบ่งการทดลองของระบบออกเป็น 2 ส่วนใหญ่ คือการทดลองด้านซอฟต์แวร์และการทดลองด้านฮาร์ดแวร์

4.1 ฮาร์ดแวร์ สามารถแบ่งการทำงานออกเป็นดังนี้

4.1.1 การทดสอบหน่วยความจำภายนอก (RAM)

จากขั้นตอนในการออกแบบในบทที่ 3 จึงทำการต่อวงจร ซึ่งในการทดสอบในเบื้องต้นได้ใส่ LED ที่ขา CE เพื่อตรวจสอบว่าเกิดการอ่านหรือเขียนข้อมูลในหน่วยความจำหรือไม่ หลังจากนั้นจึงเขียนโปรแกรมมาตรวจสอบ RAM โดยให้ตำแหน่งที่ 200010 – 200019 เป็น 00112233 โดยโปรแกรมจะอยู่ในภาคผนวก แสดงไว้ในรูปที่ 4.1

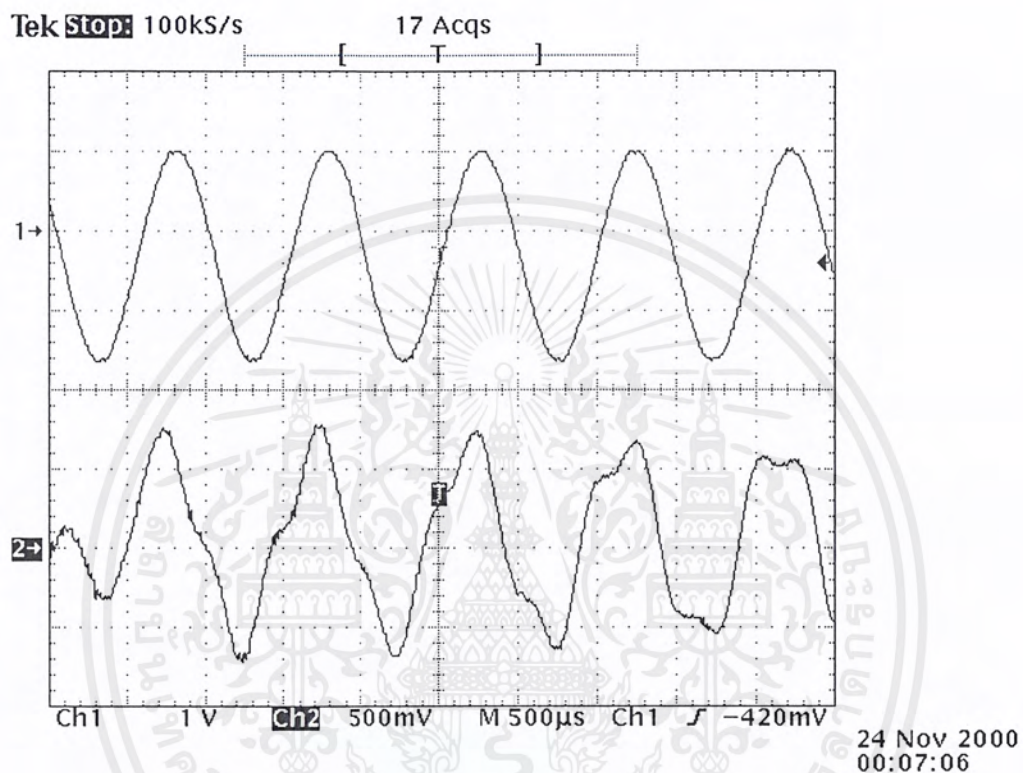


```
MS-DOS Prompt - DSK3D
ISASSEMBLY
B $+0
80986d _main PUSH AR3 200000 ef7eae8a aaebebaa a266a27f
80986e LDI SP,AR3 200004 00000000 00000000 00000000 00000000
80986f ADDI 2,SP 200008 00000000 00000000 00000000 00000000
809870 CALL Default_sect 20000c 00000000 00000000 00000000 00000000
809871 LDI **AR3(2),R0 200010 00112233 00112233 00112233 00112233
809872 PUSH R0 200014 00112233 00112233 00112233 00112233
809873 CALL _UPDATE_SAMP 200018 00112233 00112233 00000000 00000000
809874 BUD $-3 20001c 00000000 00000000 00000000 00000000
809875 SUBI 1,SP 200020 00000000 00000000 00000000 00000000
809876 STI R0,**AR3(1) 200024 00000000 00000000 00000000 00000000
809877 STI R0,**AR3(2) 200028 00000000 00000000 00000000 00000000
809878 .text .word 000809c00h 20002c 00000000 00000000 00000000 00000000
809879 .word 0008098bbh 200030 00000000 00000000 00000000 00000000
80987a _c_int00 LDI 0,ST 200034 00000000 00000000 00000000 00000000
COMMAND
reset 200038 00000000 00000000 00000000 00000000
load loopc8.out 20003c 00000000 00000000 00000000 00000000
200040 00000000 00000000 00000000 00000000
200044 00000000 00000000 00000000 00000000
200048 00000000 00000000 00000000 00000000
20004c 00000000 00000000 00000000 00000000
200050 00000000 00000000 00000000 00000000
CMD>
Help NewAddr Grow
```

รูปที่ 4.1 การทดสอบหน่วยความจำภายนอก

4.1.2 ทดสอบวงจรควบคุมการทำงาน

วงจรควบคุมการทำงาน ได้ใช้สวิตช์เป็นตัวเลือกการประมวลผล โดยมีการลดขนาดของข้อมูลจากขั้นตอนในการออกแบบในบทที่ 3 ได้ผลลัพธ์ดังรูปที่ 4.2

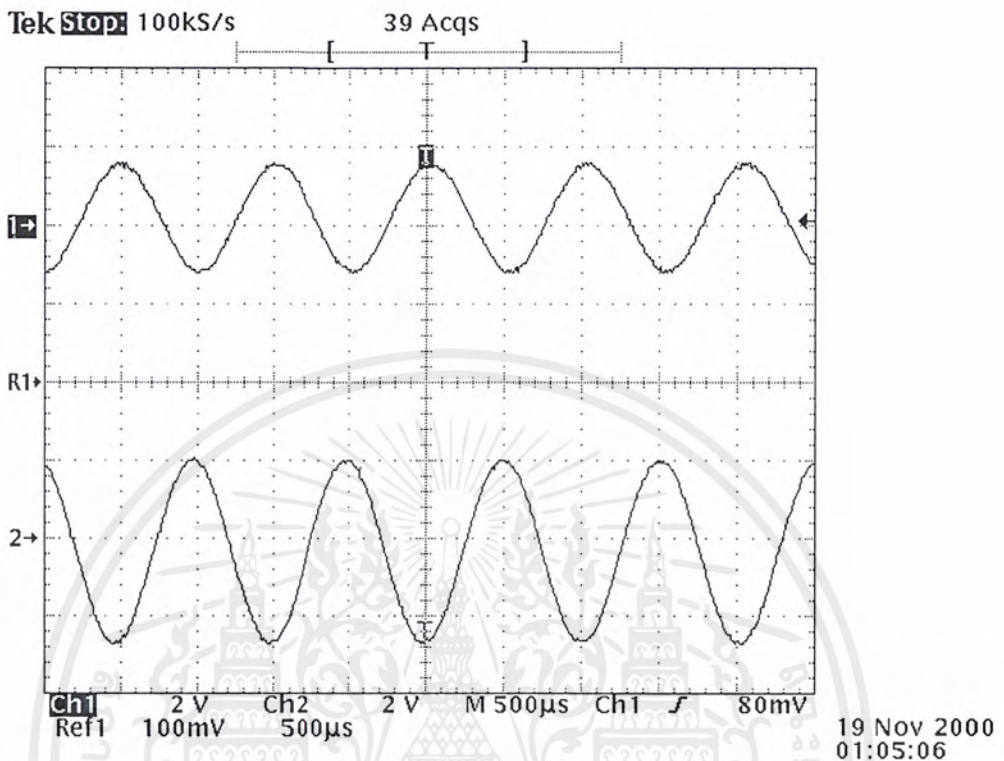


รูปที่ 4.2 ด้านบนคือสัญญาณสัญญาณอินพุต ด้านล่างคือสัญญาณเอาต์พุตของวงจร คุมการทำงาน

4.2 การทดลองด้านซอฟต์แวร์ สามารถแบ่งการทำงานออกเป็นดังนี้

4.2.1 โปรแกรมนำข้อมูลจากอินพุตสู่เอาต์พุต

จากโปรแกรมตัวอย่าง AICCOM.C ในบทที่ 3 ได้นำมาพัฒนาได้ดังโปรแกรมที่ 1 ในภาคผนวก ก หลังจากนั้นได้ทำการทดลอง ได้ผลดังรูปที่ 4.3

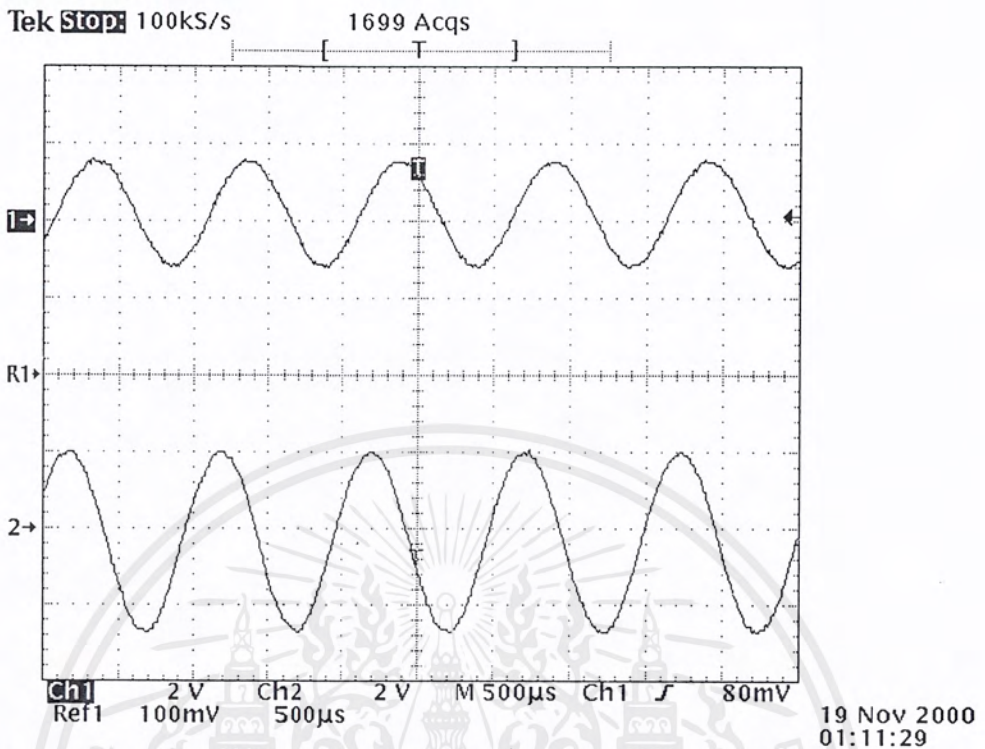


รูปที่ 4.3 ด้านบนคือ สัญญาณอินพุตและส่วนด้านล่างคือผลการทดลองโปรแกรมนำข้อมูลจากอินพุตสู่เอาต์พุต

จากรูป 4.3 ด้านบนเป็นสัญญาณอินพุต ส่วนรูปด้านล่างเป็นสัญญาณเอาต์พุต ในการทดลองได้นำสัญญาณไซน์ ป้อนเป็นอินพุตมีความถี่ 1 kHz โปรแกรมจะทำการติดต่อกับ AIC จึงได้ สัญญาณเอาต์พุต ตามรูปที่ 4.3

4.2.2 โปรแกรมนำข้อมูลเก็บในหน่วยความจำภายนอก

จากขั้นตอนในการออกแบบในบทที่ 3 จึงนำมาเขียนโปรแกรมได้ดังโปรแกรมที่ 2 ในภาคผนวก ก โดยโปรแกรมนำข้อมูลอินพุตเก็บในหน่วยความจำภายนอกก่อนออกสู่เอาต์พุต หลังจากนั้นได้ทำการทดลองโดยตำแหน่งของหน่วยความจำภายนอกอยู่ที่ 200000H ได้ผลดังรูปที่ 4.4

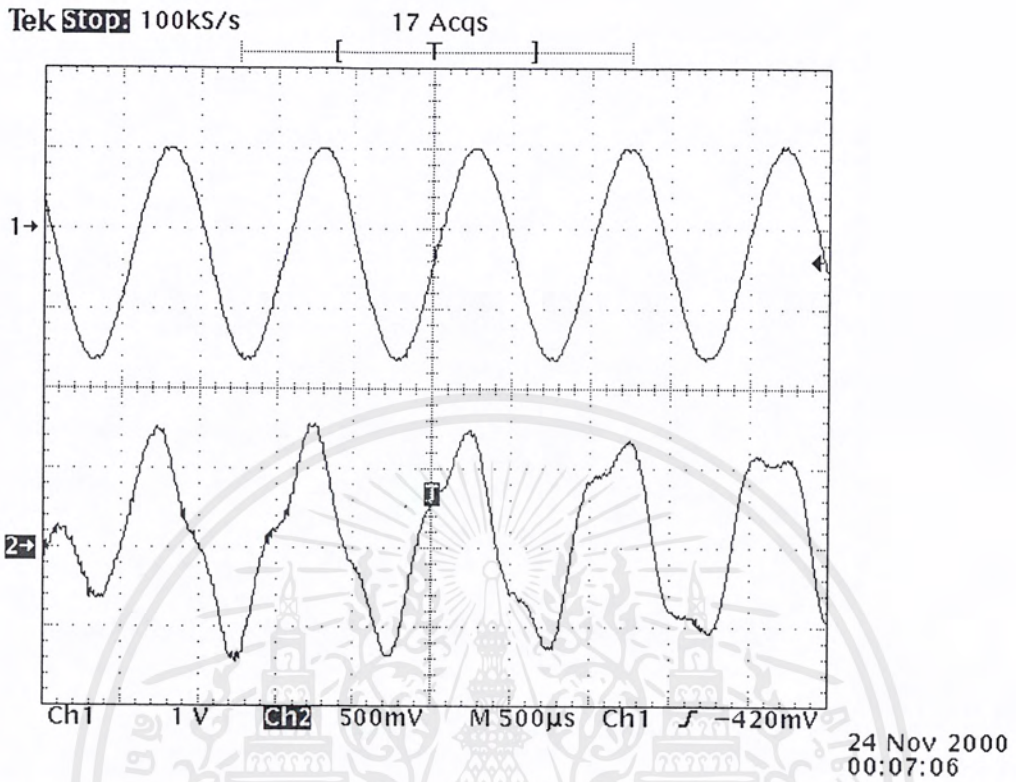


รูปที่ 4.4 ด้านบนคือ สัญญาณอินพุต ด้านล่างคือ ผลการทดลองโปรแกรมนำข้อมูลเก็บในหน่วยความจำภายนอก

4.2.3 ทดสอบโปรแกรมการแปลงเวฟเล็ดและแปลงกลับเวฟเล็ด

โปรแกรมนำข้อมูลเสียงจากอินพุต มากระทำการแปลงเวฟเล็ดแล้วจึงเก็บในหน่วยความจำภายนอก ทำการแทรกศูนย์และแปลงกลับเวฟเล็ด ก่อนออกสู่เอาต์พุต

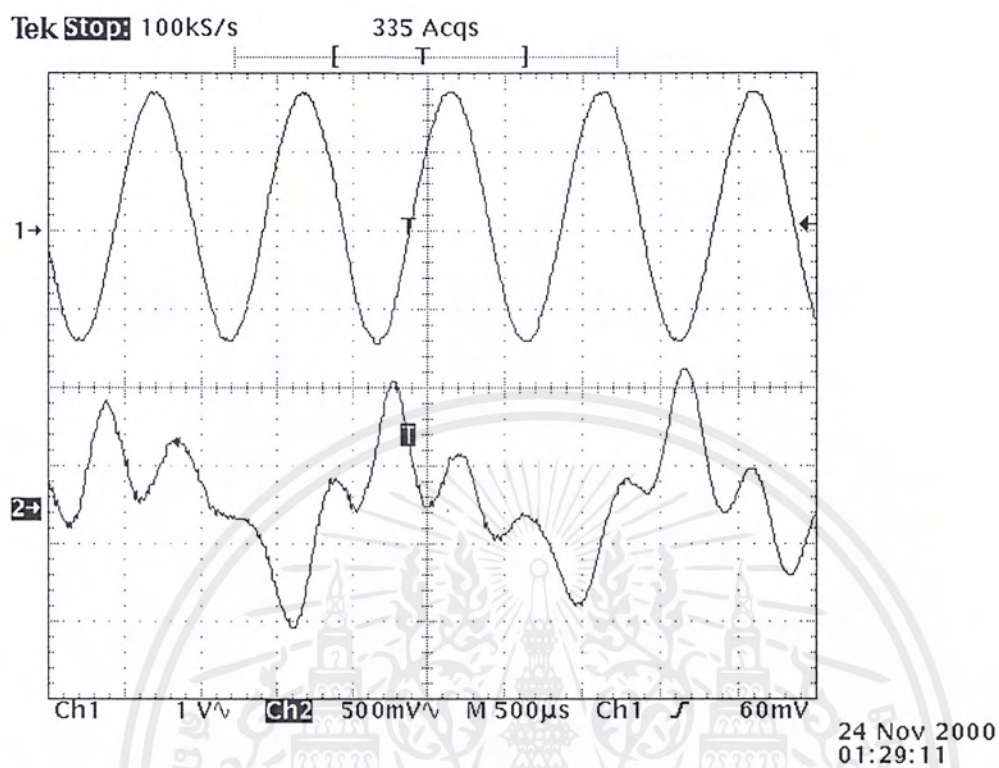
จากขั้นตอนในการออกแบบในบทที่ 3 จึงนำมาเขียนโปรแกรมได้ดังโปรแกรมที่ 2 ในภาคผนวก ก หลังจากนั้นได้ทำการทดลองโดยตำแหน่งของหน่วยความจำภายนอกอยู่ที่ 20000H ได้ผลดังรูปที่ 4.5



รูปที่ 4.5 ด้านบนคือสัญญาณสัญญาณอินพุต ด้านล่างคือ ที่ทำการแปลงเวฟเล็ดและลดขนาดข้อมูล

4.3 รวมการทำงานของซอฟต์แวร์และฮาร์ดแวร์

หลังจากทดสอบทั้งหมดแล้ว ทำการป้อนสัญญาณเสียง ซึ่งได้ผลการทดลองดังรูปที่ 4.6

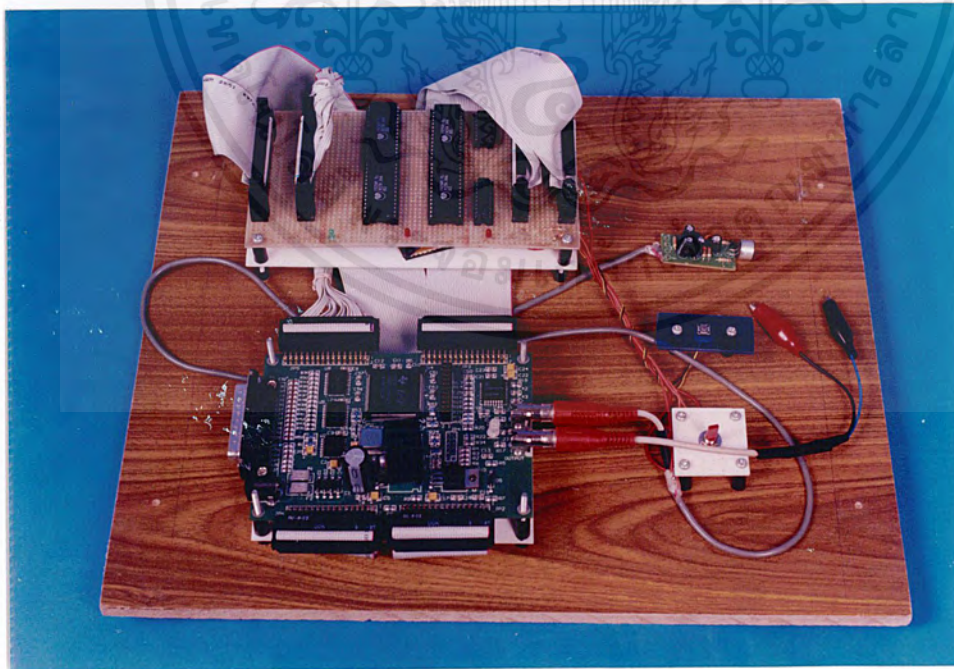


รูปที่ 4.6 ผลการทดสอบการรวมกันของทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ภาพถ่ายบอร์ดตัวอย่างของ DSK TMS320C31



รูปที่ 4.8 การทดสอบบอร์ดตัวอย่างของ DSK TMS320C31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

ในการออกแบบสร้างวงจรในการบีบอัดข้อมูลเสียงพูดด้วยเวฟเล็ตโดยใช้ตัวประมวลผล DSK TMS320C31 โดยใช้อัลกอริทึมของเวฟเล็ต ในการแปลงเวฟเล็ตจะแยกเป็นค่าสัมประสิทธิ์ที่ ความถี่ต่ำและค่าสัมประสิทธิ์ที่ความถี่สูง โดยการแตกกระจายสัมประสิทธิ์ ซึ่งในความถี่สูงนี้เป็นส่วนประกอบของข้อมูลที่มีความสำคัญน้อยซึ่งสามารถตัดทิ้งไปได้ นั่นก็คือการลดขนาดของข้อมูล

การแปลงกลับเวฟเล็ตก็คือการรวมสัมประสิทธิ์ด้านความถี่สูงและความถี่ต่ำซึ่งจะกลับมาเป็นสัญญาณต้นแบบ อาจมีค่าผิดพลาดบางส่วนแต่สามารถยอมรับได้ ในโครงงานนี้ได้ใช้วิธีการของ Multi Resolution Wavelet โดยใช้ Level 1 และได้เลือกสัมประสิทธิ์เฉพาะด้านความถี่ต่ำ ดังนั้นสามารถลดขนาดของข้อมูลได้ร้อยละ 50 โดยเสียงพูดที่ได้ทางเอาต์พุตสามารถรับฟังได้

ส่วนของ DSK TMS320C31 เป็นตัวประมวลผลสัญญาณที่มีความเร็วสูง และการทำงานจะเป็นแบบ Real Time) และยังสามารถพัฒนาโปรแกรมบนคอมพิวเตอร์ แล้วส่งไปยังบอร์ด DSK จึงสามารถแก้ไขและพัฒนาโปรแกรมได้อย่างสะดวกและใช้เวลาไม่มาก ซึ่งในการทำงานของ DSK TMS320C31 เหมาะกับการประมวลผลสัญญาณเชิงเลข

5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่ 1 การอ้างตำแหน่งหน่วยความจำภายนอก (RAM) ไม่สามารถอ้างที่ 8 บิตสุดท้ายได้

แนวทางแก้ไข 1 ครั้งแรกใช้ตัวแปรชนิด Unsigned Int แก้ไขโดยใช้ Long Int

ปัญหาที่ 2 ในการบีบอัดสัญญาณมีผลทำให้สัญญาณเอาต์พุตผิดเพี้ยนทางความถี่

แนวทางแก้ไข 2 แก้ไขโปรแกรมการ Convolution ใหม่

ปัญหาที่ 3 แรงดันที่ออกจาก JP6 ไม่ใช่แรงดันเอาต์พุตที่ออกจากไอซีเร็กกูเลท 7805 แต่เป็นแรงดันที่ออกจากขาอินพุตไอซีเร็กกูเลท 7805 จึงทำให้มีแรงดันสูงกว่า 5 โวลต์ เป็นผลทำให้ RAM ภายนอก เสียหาย

แนวทางแก้ไข 3 ต่อวงจร ไอซี 7805 เพิ่มที่วงจร RAM ภายนอก

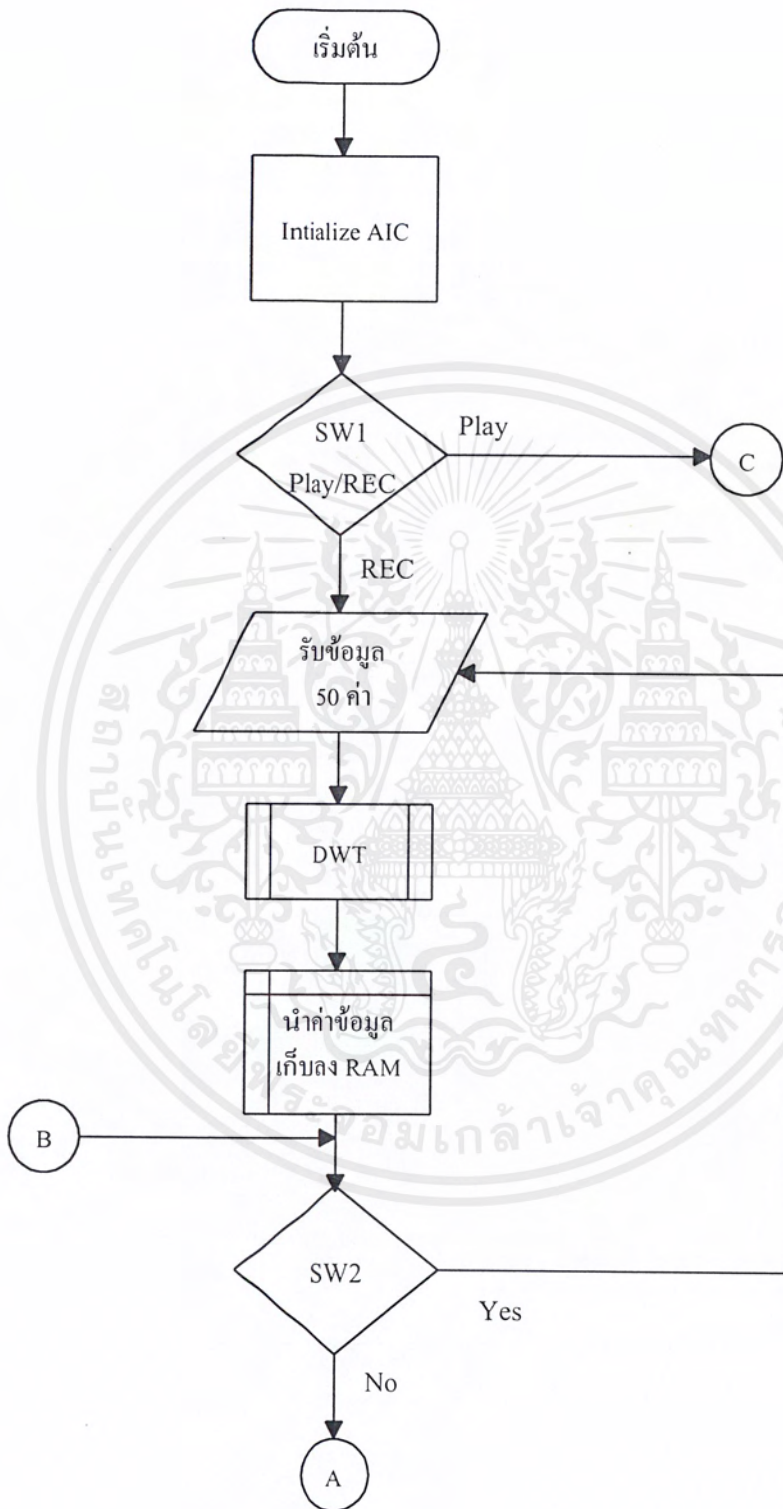
5.3 แนวทางการพัฒนา

1. วงจรบันทึกเสียงสามารถบันทึกเสียงพูดได้นานกว่า 8 วินาที
แนวทางการพัฒนา สามารถต่อหน่วยความจำภายนอก (RAM) เพิ่มเติมเพื่อเก็บข้อมูลได้มากขึ้น
2. วงจรบันทึกเสียงสามารถลดขนาดของข้อมูลได้มากกว่าร้อยละ 50
แนวทางการพัฒนา ใช้การแปลงเวฟเลตแบบ Multi Resolution Wavelet โดยใช้ Level ที่ 2 ขึ้นไป

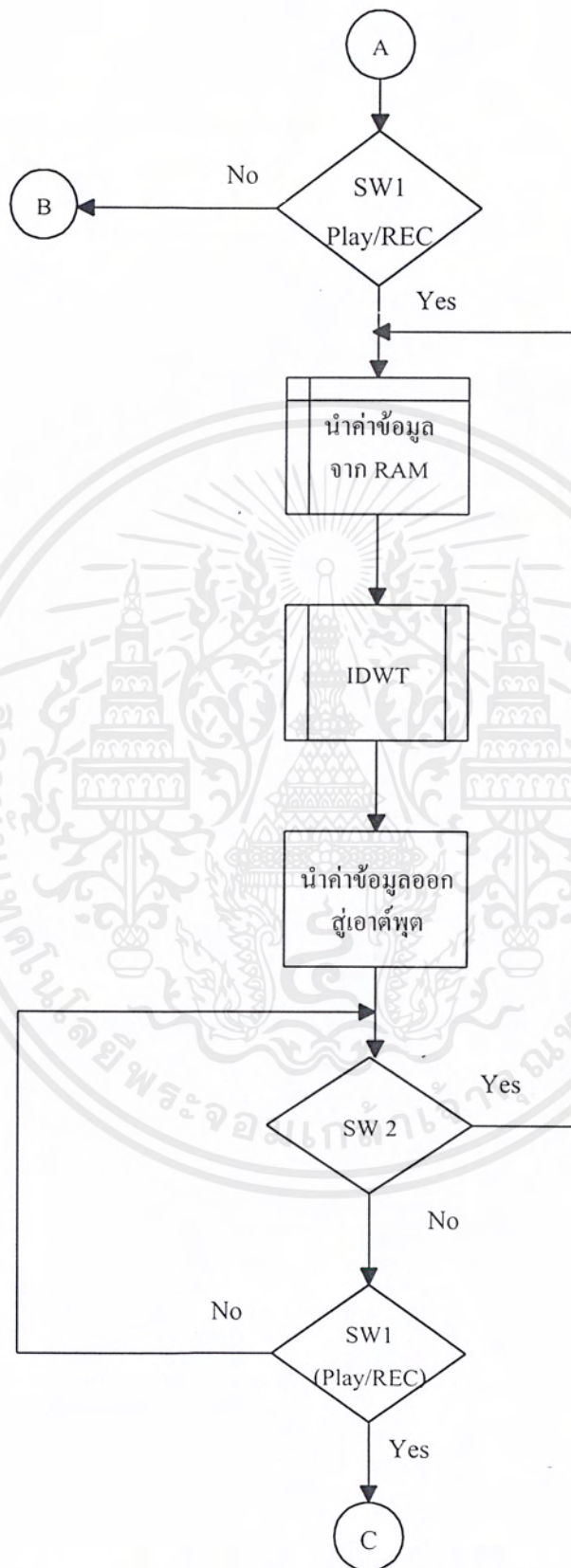




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ผังงานโปรแกรมควบคุมการทำงานบอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*AICCOMC.C – COMMUNICATION ROUTINES FOR AIC*/
#define TWAIT while (!(PBASE[0x40] & 0x2)) /*wait till XMIT buffer clear*/
extern int AICSEC[4];          /*array defined in main prog */
volatile int *PBASE = (volatile int *) 0x808000; /*peripherals base addr*/
float ld_a[4]={0.4830, 0.8365 ,0.2241, -0.1294},hd_d[4]={-0.1294,-0.2241,0.8365,-0.4830},
      lo[4] = {-0.1294, 0.2241, 0.8365, 0.4830},ho[4] = {-0.4830, 0.8365, -0.2241, -0.1294};
int i,j,a1,a2,a3,a4,a5,a6,b1,b2,b3,h6[100],s6;
float f1,f2,f3,h1[70],h2[70],h3[70],h4[70],h5[70],ca[70],cd[70];
void AICSET()                  /*function to initialize AIC */
{
volatile int loop;           /*declare local variables */
PBASE[0x28] = 0x00000001;     /*set timer period */
PBASE[0x20] = 0x000003C1;     /*set timer control register */
asm(" LDI  00000062h,IOF"); /*set IOF low to reset AIC */
for (loop = 0; loop < 90; loop++); /*keep IOF low for a while */
PBASE[0x42] = 0x00000131;     /*set xmit port control */
PBASE[0x43] = 0x00000131;     /*set receive port control */
PBASE[0x40] = 0x0E970300;     /*set serial port global reg */
PBASE[0x48] = 0x00000000;     /*clear xmit register */
asm(" OR  00000006h,IOF"); /*set IOF high to enable AIC */
for (loop = 0; loop < 4; loop++) /*loop to configure AIC */
{
TWAIT;                       /*wait till XMIT buffer clear */
PBASE[0x48] = 0x3;           /*enable secondary comm */
TWAIT;                       /*wait till XMIT buffer clear */
PBASE[0x48] = AICSEC[loop];  /*secondary command for SP0 */
} }

void AICSET_I()               /*configure AIC, enable TINTO */
{

```

```

AICSET();          /*function to configure AIC */
asm(" LDI  00000000h,IF"); /*clear IF Register */
asm(" OR   00000010h,IE"); /*enable EXINT0 CPU interrupt */
asm(" OR   00002000h,ST"); /*global interrupt enable */
}
void delay ()
{
int delay1,delay2;
for(delay1=0;delay1<200;delay1++)delay2=delay1;
return;
}

void dwt (void )
{
int ii;
a4=a3;
f1=0;
f2=0;
i=0;
ii=0;
for(a1=0;a1<(4+a3);a1++)
{ a6=a4;
for(a5=0;a5<4;a5++,a6++)
{
f2+=hd_d[a5]*ca[a6];
f1+=ld_a[a5]*ca[a6];
}

if(a1%2==0)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    h1[i]=f1;    /* (a3+1)/2-i */
    h2[i]=f2;
    i++;
}
f1=0;
f2=0;
a4--;
}
}
void insert (void)
{
    b2=2*i;
    h3[b2]=0;
    h4[b2]=0;
    b2--;
    for(b1=i-1;b1>=1;b1--)
    {
        h3[b2]=h1[b1];
        h4[b2]=0;
        b2--;
        h3[b2]=0;
        h4[b2]=0;
        b2--;
    }
}

void idwt (void)
{
    a4=2*i;
    f1=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

f2=0;
b3=0;
for(a1=2*i+1;a1<50;a1++)
{
    h3[a1]=0;
    h4[a1]=0;
    h5[a1]=0;
}

for(a1=0;a1<(2*i)+4;a1++)
{
    a6=a4;
    for(a5=0;a5<4;a5++,a6++)
    {
        f1+=ld_a[a5]*h3[a6];
        f2+=hd_d[a5]*h4[a6]; /*h3[a6];*/
    }
    f3=f1+f2;
    if (f3!=0)
    {
        h5[b3]=f3;
        b3++;
    }
    f1=0;
    f2=0;
    a4--;
}
return;
}

int UPDATE_SAMPLE(int output) /*function to update sample */
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int input,s1,s2,s4,z1,z2,z3;

float z4;

float *BAS,*BAS1;

long int *SW,i1,s3;

SW = ( long int *) 0x0c00000;

BAS = ( float *) 0x0200000;

BAS1= ( float *) 0x0200000;

for(;;)
{
y: for(z1=0;z1<70;z1++)
{
h1[z1]=0;
h2[z1]=0;
h3[z1]=0;
h4[z1]=0;
h5[z1]=0;
}
BAS = ( float *) 0x0200000;
for(z2=0;z2<1260;z2++) /*654*/ /*4861 7*/ /* 1560 21*/
{
for(a3=0;a3<50;a3++) /*5 6553*/
{
    TWAIT;
    Ca[a3]=PBASE[0x4C]<<16>>18;
    PBASE[0x48]=0x0;
}
dwt();
for(z1=0;z1<i;z1++,BAS++) *BAS=(float )h1[z1];
for(z3=0;z3<70;z3++)
{

```

```

    h1[a3]=0;
    ca[a3]=0;
    h2[a3]=0;
}
}
for(;;)
{
    i1 = *SW;
    s3 = i1 & 0x00000001;
    if(s3==0)
    {
        while(s3!=0x00000001)
        {
            s3 = 0;
            i1 = *SW;
            s3 = i1 & 0x00000001;
        }
        goto y;
    }
    else
    {
        i1 = *SW;
        s3 = i1 & 0x00000002;
        if(s3==0x00000002)
            goto x;
    }
}
x: BAS = ( float *) 0x0200000;
BAS1= ( float *) 0x0200000;
for(z1=0;z1<1260;z1++) /*654*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    for(z3=0;z3<70;z3++)
    {
        h1[z3]=0;
        h2[z3]=0;
        ca[z3]=0;
        h3[z3]=0;
        h4[z3]=0;
        h5[z3]=0;
    }
    for (a3=0;a3<i;a3++,BAS++)
    {
        h1[a3] = (float)*BAS;
    }
    insert();
    idwt();
    for(s3=(b3-50);s3<b3;s3++)
    {
        output=(float)h5[s3];
        TWAIT;
        PBASE[0x48] = output << 2;
        s1 =PBASE[0x4C]<<16>>18;

    }
}

for(;;)
{
    i1 = *SW;
    s3 = i1 & 0x00000001;
    if(s3==0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    while(s3!=0x00000001)
    {
        s3 = 0;
        i1 = *SW;
        s3 = i1 & 0x00000001;
    }
    goto x;
}
else {
    i1 = *SW;
    s3 = i1 & 0x00000002;
    if(s3==0)
        goto y;
} } }
PBASE[0x48] = output << 2; /*left shift and output sample*/
input = PBASE[0x4C] << 16 >> 18; /*input sample and sign extend*/
return(input); /*return new sample */

```

รูปที่ ก.2 โปรแกรมควบคุมการทำงาน

```

#include<stdio.h>
#include<conio.h>
main(){
float i_d[4]={0.4830, 0.8365 ,0.2241, -0.1294},
    h_d[4]={ -0.1294,-0.2241,0.8365,-0.483},
    i_r[4]={-0.1294,0.2241,0.8365,0.483},
    h_r[4]={-0.4830, 0.8365 ,-0.2241, -0.1294},
    s[100]={0},a[100]={0},ca[100]={0},b[100]={0} ,da[100]={0},da1[100]={0}
    , da2[100]={0},cd[100]={0},cd1[100]={0},ca1[100]={0},ca2[100]={0};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j,x,c;
clrscr();

printf("Length : ");scanf("%d",&x); /*Input signal*/
for (i=0;i<x;i++) {
printf("Value %d: ",i); scanf("%f",&s[i]); }
for (i=0;i<=x+2;i++) {
for (j=0;j<=3&&j<=i;j++) { /*Convolution at low*/
a[i]=a[i]+i_d[j] * s[i-j]; }
printf("conv Lo %.4f \n",a[i]); }
getch();
j=0;
for(i=0;i<x*2;i+=2)
{ ca[j]=a[i+1]; j++; }
printf("\n"); /*Down sample*/
for(i=0;i<(x+3)/2;i++)
{ printf("Down Sample %.4f\n",ca[i]); } getch();printf("\n");
for (i=0;i<=x+2;i++) {
for (j=0;j<=3&&j<=i;j++) {
b[i]=b[i]+h_d[j] * s[i-j]; } /*Convolution at high*/
printf("conv Hi %d %.4f \n",i,b[i]); } getch();
printf("\n");
j=0;
for(i=0;i<x*2;i+=2)
{ cd[j]=b[i+1]; j++; } /*Down Sample*/
printf("\n");
for(i=0;i<(x+3)/2;i++)
{ printf("Down Sample %.4f\n",cd[i]); } getch();
for(i=x*2+1;i>0;i-=2)
{ ca[i]=ca[i/2];
ca[i-1]=0; } /*Up sample Low*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n");
for(i=0;i<x*2-2;i++) {
printf("Up Sample at low %.4f\n",ca[i]); }
getch();

for (i=0;i<=(x*2)+3;i++) {
for (j=0;j<=3&&j<=i;j++) {
da[i]=da[i]+i_r[j] * ca[i-j]; }
printf("\n conv at low %d = %.4f ",i,da[i]); } getch();
printf("\n");
for(i=x*2+1;i>0;i-=2)
{ cd[i]=cd[i/2];
cd[i-1]=0; }
printf("\n"); /*Up Sample High*/
for(i=0;i<x*2-2;i++) {
printf("Up Sample at High %.4f\n",cd[i]); }
getch();

for (i=0;i<=(x*2)+3;i++) {
for (j=0;j<=3&&j<=i;j++) {
da1[i]=da1[i]+h_r[j] * cd[i-j]; }
printf("\n conv at High %d = %.4f ",i,da1[i]); }getch();
printf("\n");
for (i=0;i<x*2+1;i++){
da2[i]=da[i]+da1[i];
printf("\nconv at low + conv at High = %.4f",da2[i]);}getch();
printf("\n");
for (i=0;i<x*2+1;i++){
if (da2[i]<0) {
da2[i]=da2[i+1]; } /*wkeep*/
else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n%.4F",da2[i]); } }
getch();
}

```

รูปที่ ก.3 โปรแกรมการแปลงเวฟเล็ต

```

/*AICCOMC.C – COMUNICATION ROUTINES FOR AIC*/
#define TWAIT while (!(PBASE[0x40] & 0x2)) /*wait till XMIT buffer clear*/
extern int AICSEC[4]; /*array defined in main prog */
volatile int *PBASE = (volatile int *) 0x808000; /*peripherals base addr*/
void AICSET() /*function to initialize AIC */
{
volatile int loop; /*declare local variables */
PBASE[0x28] = 0x00000001; /*set timer period */
PBASE[0x20] = 0x000003C1; /*set timer control register */
asm(" LDI 00000062h,IOF"); /*set IOF low to reset AIC */
for (loop = 0; loop < 90; loop++); /*keep IOF low for a while */
PBASE[0x42] = 0x00000131; /*set xmit port control */
PBASE[0x43] = 0x00000131; /*set receive port control */
PBASE[0x40] = 0x0E970300; /*set serial port global reg */
PBASE[0x48] = 0x00000000; /*clear xmit register */
asm(" OR 00000006h,IOF"); /*set IOF high to enable AIC */
for (loop = 0; loop < 4; loop++) /*loop to configure AIC */
{
TWAIT; /*wait till XMIT buffer clear */
PBASE[0x48] = 0x3; /*enable secondary comm */
TWAIT; /*wait till XMIT buffer clear */
PBASE[0x48] = AICSEC[loop]; /*secondary command for SP0 */
}
}

```

```

void AICSET_I()          /*configure AIC, enable TINTO */
{
    AICSET();           /*function to configure AIC */
    asm(" LDI    00000000h,IF"); /*clear IF Register */
    asm(" OR     00000010h,IE"); /*enable EXINT0 CPU interrupt */
    asm(" OR     00002000h,ST"); /*global interrupt enable */
}

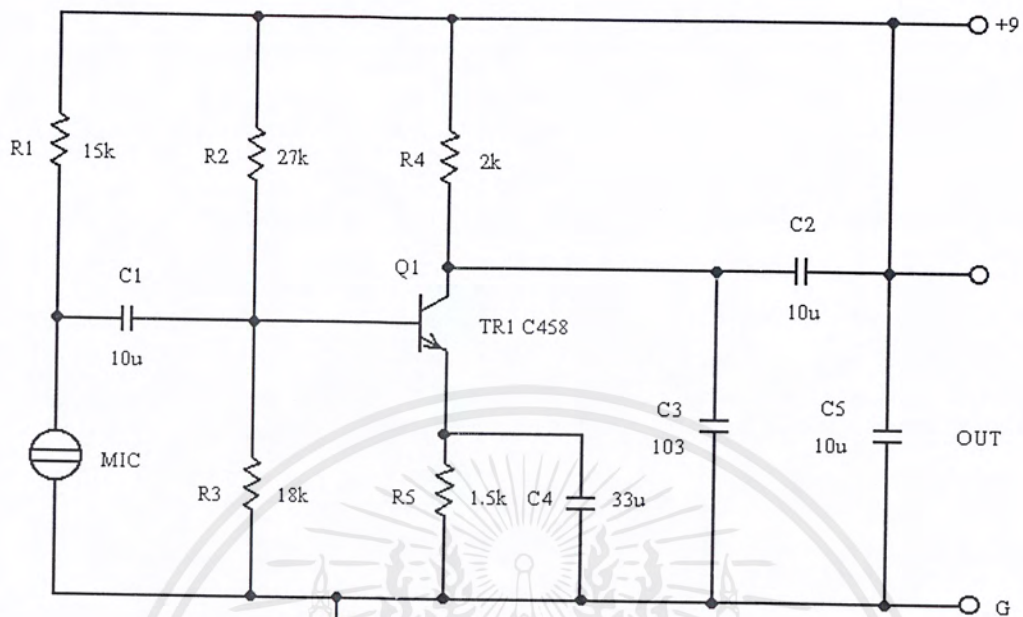
int UPDATE_SAMPLE(int output) /*function to update sample */
{
    int input,input1,n1,n2;    /*declare local variables */
    long int *a1,a2,s3,*s4;
    s4=(long int *)0x0200000;
    TWAIT;                    /*wait till XMIT buffer clear */
    PBASE[0x48] = output << 2; /*left shift and output sample*/
    input1 = PBASE[0x4C] << 16 >> 18; /*input sample and sign extend*/
    for(n1=0;n1<1000;n1++,s4++) *s4 = 0;
    s4=(long int *)0x0200010;
    for(n1=0;n1<10;n1++,s4++)
    {
        *s4=0x00112233; }
    for(;;);
    return(input);           /*return new sample */
}

```

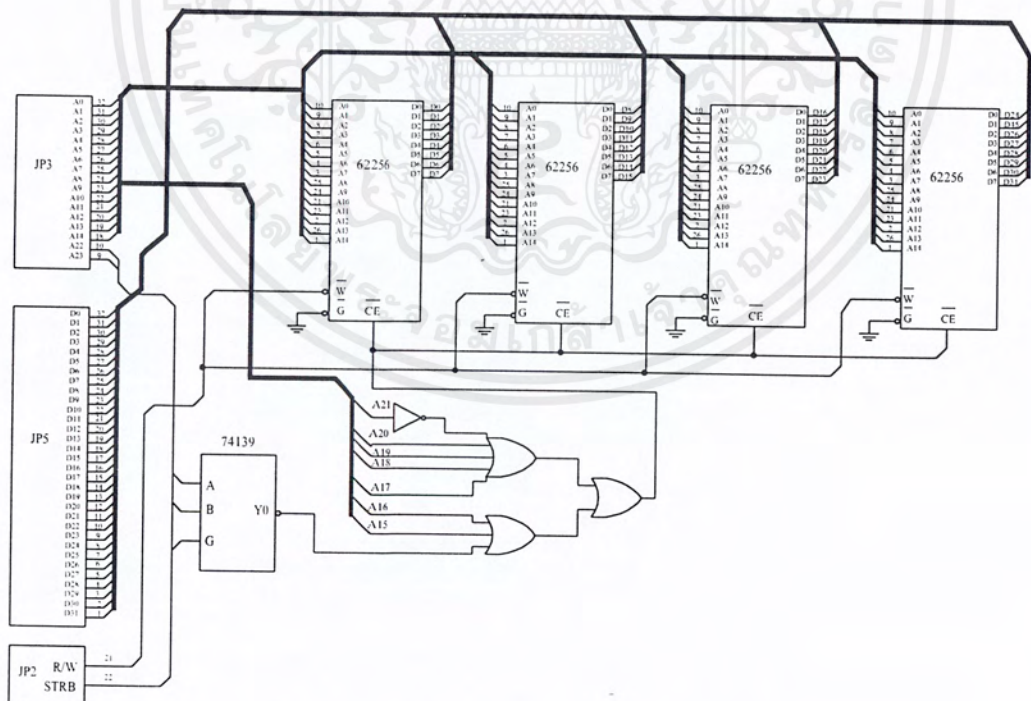
รูปที่ ก.4 โปรแกรมการตรวจสอบหน่วยความจำภายนอก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

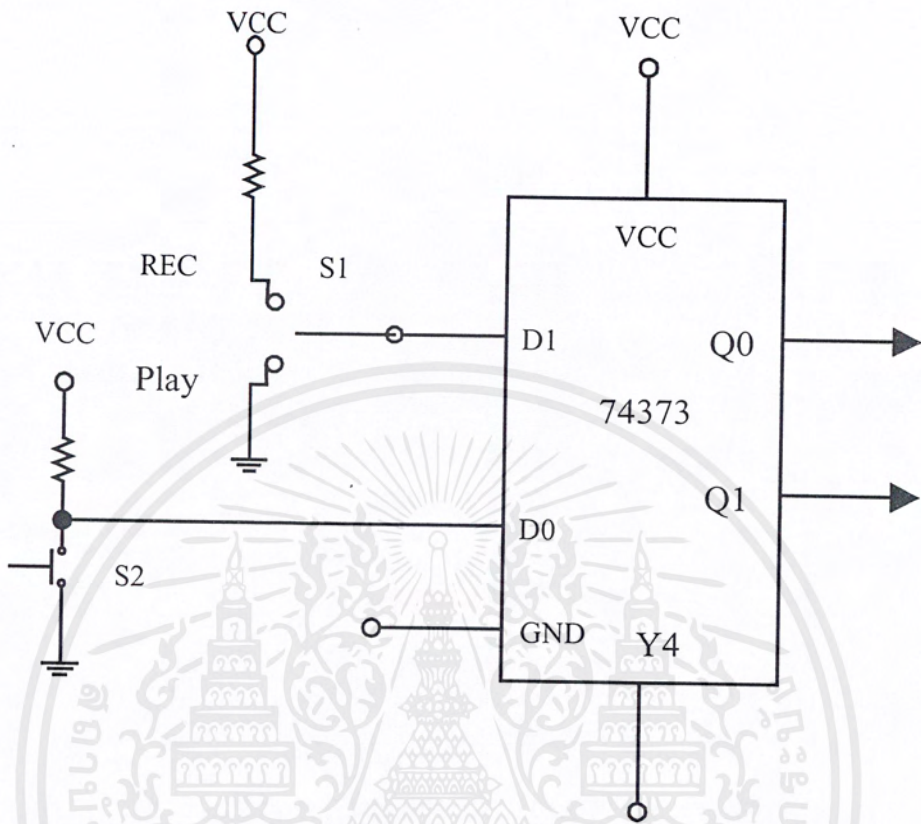


รูปที่ ข.1 วงจร PRE AMP



รูปที่ ข.2 การต่อวงจรหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 วงจรควบคุมการทำงาน

บรรณานุกรม

ชินภัทร นันทจิวงกรชัย. “การลดข้อมูลภาพโดยใช้เวฟเล็ตทรานฟอร์ม.” วิทยานิพนธ์วิศวกรรม
ศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอม
เกล้าเจ้าคุณทหารลาดกระบัง. 2542

CHASSAING, R. **Digital Signal Processing Laboratory Experiment Using C and the
TMS320C31.** New York : University of Massachusetts. 1999



ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวจรีลักษณ์ ประสิทธิ์มณีรัตน์
วันเดือนปีเกิด	23 มิถุนายน 2521
สถานที่เกิด	จังหวัดชลบุรี
ภูมิลำเนาเดิม	66/26 ม.5 ต.ห้วยกะปิ อ.เมือง จ.ชลบุรี
ที่อยู่ปัจจุบัน	หอ เสาวณี หมู่ 4 แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	02-3268456

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านห้วยกะปิ
มัธยมศึกษาตอนต้น	โรงเรียนอ่างศิลาพิทยาคม
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคสัตหีบ
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	วิทยาลัยเทคนิคสัตหีบ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
คติพจน์	เส้นทางทุกความฝันไม่ได้โรยด้วยกลีบกุหลาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานิพนธ์	นาย บรรยงค์ ค้วงนุ้ม
วันเดือนปีเกิด	17 พฤศจิกายน 2522
สถานที่เกิด	จังหวัดสิงห์บุรี
ภูมิลำเนาเดิม	26 ม.12 ต.คงคอน อ.สรรคบุรี จ.ชัยนาท
ที่อยู่ปัจจุบัน	หอ เสาวณี หมู่ 4 แขวงลำปลาตีว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	02-3268456

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเทพรัตน์
มัธยมศึกษาตอนต้น	โรงเรียนครูประชาสรรค์
ประกาศนียบัตรวิชาชีพ(ปวช.)	วิทยาลัยเทคนิคชัยนาท
ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.)	วิทยาลัยเทคนิคท่าหลวงซิเมนต์ไทยอนุสรณ์
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
กิตติพจน์	อ้วนวันนี้ดีกว่าอ้วนหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้