



## การใช้งาน DSP ในงานสื่อสาร

Using DSP in communication



วัน เดือน ปี..... 18.ค.ค. 2541  
เลขทะเบียน..... 039177  
เลขเรียกหนังสือ..... T 40178 0825 ก

ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง 039177

รายงานเรื่อง           การใช้งาน DSP ในงานสื่อสาร  
( Using DSP in communication)  
จัดทำโดย           นาย วีรชัย           ปิยควาณิช           รหัส 38013246  
                                  นาย สมบูรณ์           กลิ่นจันทร์กลิ่น           รหัส 38013251  
อาจารย์ที่ปรึกษา    อ. ชินภัทร           นันทจิวากรชัย



รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

( ชินภัทร นันทจิวากรชัย )

วันที่ 26 / ๕๓ / 41.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ขอขอบคุณ ผู้มีพระคุณและครูบาอาจารย์ทุกท่านที่ได้ช่วยสั่งสอนวิชาต่างๆ จนกระทั่ง  
กระผมมีความรู้มีความสามารถ และขอขอบคุณอาจารย์ทุกท่านที่ได้เสียสละเวลาอันมีค่าที่ช่วยให้  
คำปรึกษาและชี้แนะรวมถึงว่ากล่าวตักเตือนเสมอมา รวมถึงอาจารย์ที่ปรึกษา อ.ชินภัทร นันทจิวา  
กรชัย ที่ให้โอกาสในการเรียนรู้ สิ่งที่เป็นประโยชน์ต่อพวกกระผมทั้งในตอนนี้และในอนาคต รวมทั้ง  
คอยให้คำแนะนำ ให้คำปรึกษาในเรื่องต่างๆอย่างเป็นกันเองตลอดมา ขอขอบคุณครับ





( นาย วีรชัย ปิยควาณิช )



( นาย สมบูรณ์ กลิ่นจันทร์กลิ่น )

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การใช้งาน DSP ในงานสื่อสาร

นาย วีรชัย ปิยควาณิช  
นาย สมบูรณ์ กลิ่นจันทร์กลิ่น  
อ. ชินภัทร นันทจิวารัชย  
ปีการศึกษา 2540

### บทคัดย่อ

ปัจจุบันการสื่อสารเป็นสิ่งจำเป็นอย่างยิ่งในการดำรงชีวิต ทั้งทางด้านการศึกษาทางด้านอุตสาหกรรมหรือธุรกิจ การสื่อสารที่ใช้สายโทรศัพท์ในปัจจุบันมีหลักการมากมายซึ่งมีทั้งข้อดีข้อเสียโครงการนี้ได้ใช้หลักการของทฤษฎีเวฟเลิตทรานฟอร์มในการสื่อสาร ในการส่งจะใช้หลักการของการแปลงกลับเวฟเลิตทรานฟอร์มและในส่วนรับจะใช้การแปลงเวฟเลิตทรานฟอร์ม การส่งข้อมูลโดยใช้สายโทรศัพท์มีข้อจำกัดทางด้านความถี่ดังนั้นการออกแบบการทำงานจึงจำเป็นต้องนำมาพิจารณาการทำงานจะใช้การประมวลผลสัญญาณเชิงเลขซึ่งควบคุมโดยการเขียนโปรแกรม ตัวประมวลผลคือชุดทดลอง TMS320C5X ของบริษัทเท็กซัสอินสตรูเมนต์ การศึกษาอยู่ในขั้นค้นคว้าและวิจัย ผลของการทำงานจะมีประสิทธิภาพยังไม่ดีพอแต่สามารถปรับปรุงแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Using DSP in Communication

Mr. Weerachai Piyakawanit

Mr. Somboon Klinchanklan

Educational Year 1997

## Abstract

In present communication is necessary for life. Either educations or industrail economics must to use it. There are many method to send information via telephone line. It have advantage and disadvantage. In this project we apply new method that use Wavelets Transformation. When we send intellegence use inverse wavelets transformation and the reciever use wavelets transform method. When we design circuits must to consider the bandwidth because in telephone line is limited. The Debugger is used to control operating of Digital Signal Processing. The processors TMS320C5S of Texas Instruments Incorporates. At this time this project in reserch and delvelope. Then the results is not good performance. But it easy to correct.

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV-VI
สารบัญรูป	VII-VIII
บทที่ 1	
บทนำ	1
1.1 การทำงานในการส่งและรับข้อมูลโดยใช้ DSP	1
1.1.1 Personal Computer 1 & 2	2
1.1.2 DSP TMS320C5X A/D or D/A convertor	2
1.1.3 Telephon stationary & line ( Lowpass filter)	3
1.2 ขอบเขตการทำงานของโครงการ	3
1.3 ประโยชน์ที่จะได้รับจากโครงการ	4
1.3.1 ประโยชน์ทางด้านวิชาการ	4
1.3.2 ประโยชน์ที่เกิดกับผู้ทำโครงการ	4
1.3.3 ประโยชน์ที่เกิดกับผู้ที่จะศึกษาโครงการนี้ภายหลัง	4
บทที่ 2	
ทฤษฎีพื้นฐานของการแปลงเวฟเล็ด	5
2.1 ลักษณะของเวฟเล็ด	5
2.2 ทฤษฎีพื้นฐานเวฟเล็ดในการสเกลและการเลื่อนตำแหน่ง	8
2.3 ความเป็นมาของการแปลงเวฟเล็ด	13
2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ดและการแปลงฟูรีเยร์ในลักษณะของฟังก์ชัน หน้าต่าง	16
2.5 ลักษณะของเวฟเล็ดแม่แบบต่างๆ	16
2.6 การแปลงเวฟเล็ด	18
2.6.1 การแปลงเวฟเล็ดแบบต่อเนื่อง	18
2.6.2 การแปลงเวฟเล็ดแบบแบ่งช่วง	19

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การแปลงเวฟเล็ตโดยใช้หลักการของ Analysis Filter Banks	24
2.8 Octave Synthesis Filter Banks	25
บทที่ 3	
การพิสูจนทฤษฎี Wavelets Transform เพื่อใช้ในการสื่อสาร	27
3.1 ทดสอบทฤษฎี Wavelets Transform ที่นำมาใช้	27
3.2 เปลี่ยน function Wavelets ให้เหมาะสม	30
3.3 การทดสอบสัญญาณเมื่อสัญญาณเกิดการ lagging	40
3.4 นำสัญญาณผ่าน Lowpass Filter	50
3.4.1 T, U From Workspace	51
3.4.2 Auto-Scale Graph & Graph1	51
3.4.3 Analog Butterworth LP Filter	51
3.4.4 y to Workspace	51
3.5 การหาข้อมูลกลับจากสัญญาณที่ผ่าน Lowpass Filter	54
บทที่ 4	
ผลการทดลอง	62
4.1 การส่งและรับข้อมูลโดยใช้เครื่องคอมพิวเตอร์เป็นหลัก	62
4.1.1 การส่งข้อมูลโดยใช้เครื่องคอมพิวเตอร์	62
4.1.2 การรับข้อมูลโดยใช้เครื่องคอมพิวเตอร์	65
4.1.3 โปรแกรมการนับจำนวนของเลข 1 ที่รับเข้ามาเทียบกับข้อมูลที่ทำการส่งไป	68
4.1.4 ฮาร์ดแวร์ในส่วนของ การติดต่อระหว่างเครื่องคอมพิวเตอร์และบอร์ด TMS320C5X	73
4.2 การใช้บอร์ด TMS320C5X ประมวลผลสัญญาณ	74
4.2.1 ผลการทดลองทางด้านส่งข้อมูล	74
4.2.2 ผลการทดลองทางด้านรับ	81
บทที่ 5	
สรุปและวิจารณ์ผลการทดลอง	87
5.1 สรุปผลการทดลองทางการส่ง	87
5.2 สรุปผลการทดลองทางการรับ	88

	หน้า
5.3 แนวทางการแก้ไขปัญหาและปรับปรุง	91
ภาคผนวก	92
บรรณานุกรม	117



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

	หน้า
รูปที่ 1.1 Block Diagram แสดงระบบการทำงาน	1
รูปที่ 1.2 ภาพแสดงขอบเขตการทำงานของโครงการโดยใช้ Flow Chart	3
รูปที่ 2.1 แสดงตัวอย่างลักษณะของ Morlet : $g(t) = e^{-\frac{t^2}{64}} \cos(t)$	5
รูปที่ 2.2 แสดงลักษณะเงื่อนไขทั้งสอง functions wavelet	6
รูปที่ 2.3 แสดงลักษณะของเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า a,b ต่างๆกัน	7
รูปที่ 2.4 แสดงลักษณะของการแตกกระจายสัญญาณและรวมกับสัญญาณของเวฟเล็ต	8
รูปที่ 2.5 แสดงลักษณะของสัญญาณที่มีหลายความถี่ที่ตำแหน่งเวลาต่างๆกัน	9
รูปที่ 2.6 แสดงการพิจารณาเวฟเล็ตในมิติและสภาวะต่างๆ	10
รูปที่ 2.7 แสดงลักษณะของผลของการ correlation ระหว่าง $f(t), g((t-b)/a)$	12
รูปที่ 2.8 แสดงลักษณะความแตกต่างของสัญญาณ output ระหว่าง FT กับ WT	12
รูปที่ 2.9 แสดงลักษณะของระนาบเวลาความถี่แบบคงที่	14
รูปที่ 2.10 แสดงลักษณะการแตกกระจายสัญญาณ WT	15
รูปที่ 2.11 แสดงลักษณะเวฟเล็ตแม่ชนิดต่างๆ	17
รูปที่ 2.12 แสดงลักษณะของ Filter Banks ในแบบของ STFT	18
รูปที่ 2.13 แสดงลักษณะของระนาบของเวลาและความถี่ของการแปลงเวฟเล็ต	19
รูปที่ 2.14 แสดงลักษณะการแตกกระจายของ Space vector	22
รูปที่ 2.15 แสดงลักษณะการแตกกระจายสัญญาณ $f_j$ ไปยังระดับความละเอียดต่างๆกัน	22
รูปที่ 2.16 แสดงลักษณะการแตกกระจายสัญญาณ	23
รูปที่ 2.17 Block Diagram ของการแตกกระจายสัญญาณ	24
รูปที่ 2.18 แสดงลักษณะของการแยก frequency Band ของ octave filter banks	25
รูปที่ 2.19 แสดงลักษณะของ Bands seperator และ Octave synthesis filter banks	26
รูปที่ 3.1 แสดง library ของ Simulink ใน software Matlab	50
รูปที่ 3.2 แสดง Block Diagram ที่ใช้สำหรับจำลองการส่งข้อมูลผ่าน LP filter	50
รูปที่ 3.3 แสดงสัญญาณก่อนที่จะทำการผ่านวงจรของความถี่ต่ำ	52
รูปที่ 3.4 แสดงสัญญาณหลังผ่านวงจรของความถี่ต่ำ	53
รูปที่ 3.5 ภาพแสดงสัญญาณที่ถูก Sampling แล้วส่งไปตามสายโทรศัพท์	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6	แสดงสัญญาณหลังผ่านวงจรของความถี่ต่ำ	56
รูปที่ 3.7	แสดงการเปลี่ยนแปลงสัญญาณก่อนและหลังผ่านวงจรของความถี่ต่ำ	57
รูปที่ 4.1	แสดง Flow chart การส่งข้อมูลจากคอมพิวเตอร์ให้บอร์ด TMS320C5X	64
รูปที่ 4.2	แสดง Flow chart ของขั้นตอนการรับข้อมูลจากบอร์ด TMS320C5X	67
รูปที่ 4.3	แสดง Flow chart ของขั้นตอนการทำงานในการเช็คจำนวนบิตที่เป็นหนึ่ง	69
รูปที่ 4.4	แสดง Flow chart ของขั้นตอนการทำงานเพื่อนับเทียบจำนวน bit ที่คลาดเคลื่อน	72
รูปที่ 4.5	แสดงภาพสัญญาณโดยรวมที่ส่งออกมาจาก TMS320C5X โดยใช้ข้อมูลคือ $41_{16}$	80
รูปที่ 4.6	แสดงรายละเอียดสัญญาณที่ถูกส่งออกมาจาก TMS320C5X โดยมีข้อมูลคือ $41_{16}$	80
รูปที่ 4.7	ภาพแสดงสัญญาณทางด้านส่งและรับของ TMS320C5X	82
รูปวงจรในภาคผนวก	แสดงวงจรที่ใช้ในการติดต่อระหว่างเครื่องคอมพิวเตอร์และบอร์ด TMS320C5X	92
ตารางที่ 3.1	แสดงความถี่ Sampling ค่าต่างๆ	54



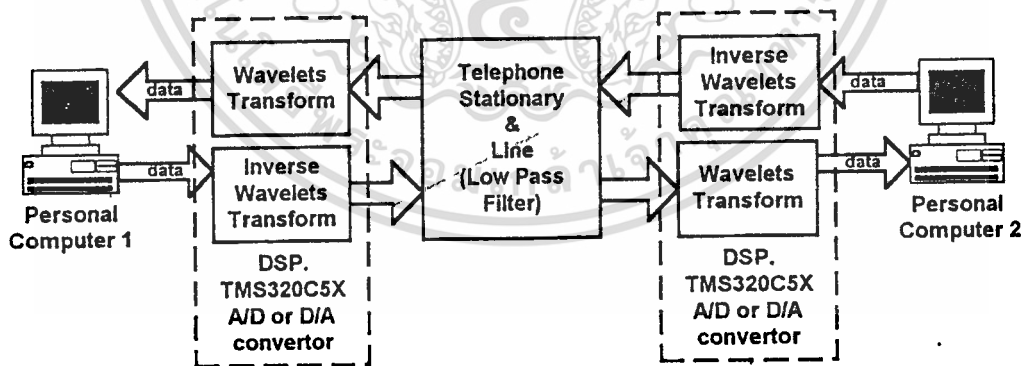
# บทที่ 1

## บทนำ

ปัจจุบันเครื่องมือที่ใช้ในการสื่อสารมีความสำคัญมาก ไม่ว่าจะเป็นหน่วยงานหรือองค์กรใด ก็จำเป็นต้องใช้สื่อสารและโดยเฉพาะอย่างยิ่งการติดต่อสื่อสารโดยใช้คอมพิวเตอร์ ในปัจจุบันการติดต่อสื่อสารโดยใช้คอมพิวเตอร์สามารถติดต่อได้ทั่วโลกไม่ว่าเราจะอยู่ที่ไหนซึ่งเป็นสิ่งที่ช่วยอำนวยความสะดวกและกลายเป็นสิ่งจำเป็นต่อการดำเนินงานทุกสิ่งอย่างไปแล้ว โครงการนี้จะทำการศึกษาวิจัยเกี่ยวกับการติดต่อสื่อสารคอมพิวเตอร์โดยผ่านทางสายโทรศัพท์ ในโครงการนี้จะใช้ Digital Signal Processing ในการส่งและรับสัญญาณ และใช้ทฤษฎี Wavelets Transform ในการเปลี่ยนข้อมูลเพื่อส่งและรับให้มีประสิทธิภาพดีที่สุด ในบทที่ 1 นี้จะขออธิบายเป็นส่วนๆดังนี้

### 1.1 การทำงานในการส่งและรับข้อมูลโดยใช้ DSP.

การทำงานของระบบการสื่อสารในโครงการนี้จะขออธิบายเป็น Block Diagram เพื่อความสะดวกและง่ายต่อการเข้าใจ Block Diagram ด้านล่างจะแสดงถึงการทำงานทั้งการส่งและรับข้อมูล



รูปที่ 1.1 Block Diagram แสดงระบบการทำงาน

จาก Block Diagram เมื่อเครื่องคอมพิวเตอร์ 1 ต้องการส่งข้อมูลไปให้เครื่องคอมพิวเตอร์ 2 ข้อมูลที่ต้องการส่งนั้นจะถูกเปลี่ยนโดยใช้ Inverse Wavelets Transform และทำเป็น Analog ส่งไปตามสายโทรศัพท์หรือผ่านทวนสายเมื่อถึงปลายทางก็จะเปลี่ยนเป็น Digital และแปลงสัญญาณนั้นกลับมาเป็นข้อมูลโดยใช้ Wavelets Transform หากเครื่องคอมพิวเตอร์ 2 ต้องการส่งข้อมูลไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังเครื่องคอมพิวเตอร์ 1 ก็ทำเช่นเดียวกัน จะขออธิบายส่วนต่างๆใน Block Diagram เป็นข้อๆ ดังนี้

### 1.1.1 Personal Computer 1 & 2

เครื่องคอมพิวเตอร์นี้เป็นตัวส่งและตัวรับข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง เพื่อส่งข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่งเพื่อนำไปใช้งานต่างๆที่เหมาะสม การติดต่อระหว่างคอมพิวเตอร์ 2 เครื่องหรือมากกว่าเราอาจใช้ระบบ LAN (Local Area Network) หากเราใช้การเชื่อมต่อแบบนี้มีข้อเสียที่ว่าต้องใช้การลงทุนสูงทั้งในด้าน software และ Hardware แต่มีข้อดีคือสามารถใช้ติดต่อระหว่างเครื่องคอมพิวเตอร์ด้วยกันเป็นจำนวนมากได้แม้ว่าเครื่องคอมพิวเตอร์ทั้งสองจะอยู่ไกลกัน วิธีการติดต่อระหว่างคอมพิวเตอร์ 2 เครื่องเราอาจใช้วิธีการ Link โดยการใช้สายที่ทำไว้เฉพาะต่อที่ Printer port ของคอมพิวเตอร์ทั้งคู่ วิธีนี้เป็นวิธีการที่ง่ายแต่มีข้อเสียคือไม่สามารถติดต่อสื่อสารกันในระยะทางไกลๆได้และสามารถติดต่อกันได้แค่ 2 เครื่อง อีกวิธีการหนึ่งที่นิยมใช้โดยทั่วไปมีข้อดีของระบบ LAN และ การ Link เข้าด้วยกันนั่นคือการใช้ MODEM (MODulation and DEModulation) นั่นคือมันสามารถติดต่อได้ไกลและเชื่อมต่อได้ง่ายและราคาถูก โดยทั่วไปในการใช้ MODEM จะใช้หลักการของ QPSK เพื่อการส่งข้อมูลแต่การใช้หลักการ QPSK จะทำให้เกิดความถี่สูงขึ้นและจะใช้ย่านความถี่เพียงความถี่เดียวในการส่งข้อมูลทำให้ในสายโทรศัพท์ที่มี Bandwidth ประมาณ 3 KHz เราไม่สามารถนำ Bandwidth ที่เหลือจากการส่งข้อมูลมาใช้งานได้ ทำให้ไม่สามารถใช้ Bandwidth ได้อย่างเต็มที่ดังนั้นการใช้ Wavelets Transform เข้ามาช่วยในการแปลงข้อมูลจะทำให้เราสามารถส่งข้อมูลไปได้เต็ม Bandwidth ในสายโทรศัพท์

### 1.1.2 DSP. TMS320C5X A/D or D/A Converter

เมื่อเครื่องคอมพิวเตอร์เครื่องหนึ่งต้องการส่งข้อมูลไปยังเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง การส่งข้อมูลนี้จะผ่านกระบวนการเพื่อให้ได้สัญญาณที่เหมาะสมในการส่ง กระบวนการนั้นจะทำตามทฤษฎีของ Wavelets Transform ซึ่งเราใช้ Chip TMS320C5X เป็นตัวประมวลผลข้อมูลนั้นๆ หลังจากนั้นก็จะผ่าน D/A Converter เพื่อเปลี่ยนข้อมูลนั้นให้เป็นสัญญาณทาง Analog และส่งต่อโดยผ่านทางสายโทรศัพท์และสถานีชุมสายต่างๆ ไปยังปลายทางเมื่อถึงปลายทางเราก็จะนำสัญญาณนั้นผ่าน A/D Converter เพื่อเปลี่ยนจากสัญญาณที่ส่งมาให้เป็นข้อมูลทาง Digital และเช่นเดียวกับการส่งเราจะใช้ Chip TMS320C5X มาประมวลผลข้อมูลนั้นตามทฤษฎี Wavelets Transform สุดท้ายเราก็จะได้ข้อมูลตามต้องการ ส่วนนี้เป็นส่วนที่มีความสำคัญมากและการทำงานในส่วนนี้ทั้งหมดจะถูกกำหนดด้วย Software ทั้งหมดดังนั้นจึงง่ายต่อการแก้ไขและปรับปรุงในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.1.3 Telephone Line & Telephone Stationary ( Low Pass Filter )

องค์ประกอบในส่วนนี้ทั้งหมดเป็นสิ่งที่ถูกสร้างและติดตั้งไว้แล้วตามสถานที่ต่างๆ และคุณสมบัติของสายโทรศัพท์ที่สำคัญในระบบนี้ก็คือมี Bandwidth 3 KHz ซึ่งหากเราทำการเปรียบเทียบการทำงานของส่วนนี้ก็เปรียบเสมือนวงจรกรองความถี่ต่ำ ( Low Pass Filter ) ทำให้สัญญาณ Analog ที่เราจะทำการส่งในระบบนี้ ต้องมีความถี่ต่ำกว่า 3 KHz

## 1.2 ขอบเขตการทำงานของโครงการ

จะขอแสดงขอบเขตการทำงานของโครงการนี้ตามรูป Flow chart ด้านล่าง



รูปที่ 1.2 ภาพแสดงขอบเขตการทำงานของโครงการโดยใช้ Flow chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ประโยชน์ที่จะได้รับจากโครงการ

ประโยชน์ที่จะได้รับจากโครงการนี้มีหลายอย่างจะขอแยกแยะออกเป็นข้อๆดังนี้

#### 1.3.1 ประโยชน์ทางด้านวิชาการ

เป็นการค้นคว้าและวิจัยวิชาการใหม่ๆเพื่อศึกษาและประยุกต์การใช้งานจากทฤษฎี Wavelets Transform ซึ่งเป็นทฤษฎีที่ใช้กันอย่างแพร่หลายทั้งในด้าน Image Processing และอีกมากมาย แต่โครงการนี้จะนำมาใช้ในการลดข้อมูลเพื่อการติดต่อสื่อสารซึ่งเป็นสิ่งที่ท้าทายและน่าทดลอง ซึ่งสามารถทำให้เกิดแนวความคิดใหม่ๆได้

#### 1.3.2 ประโยชน์ที่เกิดกับผู้ทำโครงการ

ทำให้ผู้ที่ศึกษาและทำ มีความรู้เพิ่มมากขึ้นและมีแนวความคิดใหม่ๆในการทำงานในระหว่างทำโครงการเต็มไปด้วยสิ่งที่จะต้องทดลองเนื่องจากมีตัวแปรต่างๆที่มีความสำคัญต่อระบบและไม่ทราบว่าตัวแปรไหนจะทำให้เกิดประสิทธิภาพสูงที่สุดในระบบ มีเพียงวิธีเดียวคือต้องทดลองโดยใช้ Software (Mathlab) และคำนวณทางคณิตศาสตร์พื้นฐาน นอกจากนี้ยังได้ศึกษาการทำงานของ Chip Digital Signal Processing TMS320C5X และศึกษาการเขียนโปรแกรมในการสั่งให้ DSP. นั้นทำงานตามที่เราต้องการซึ่งเป็นความรู้ทางด้าน Microprocessor ซึ่งปัจจุบันมีความจำเป็นอย่างยิ่งทางด้านวิศวกรรม

#### 1.3.3 ประโยชน์ที่เกิดกับผู้ที่จะศึกษาโครงการนี้ภายหลัง

โครงการนี้ยังสามารถมีการปรับปรุงแก้ไขให้การทำงานมีประสิทธิภาพสูงกว่านี้อีกมากซึ่งทำให้เกิดผลดีในด้านต่างๆ เมื่อมีผู้ที่จะมาศึกษาภายหลังนอกจากจะได้รับประโยชน์เช่นเดียวกับที่ผู้ทำได้รับแล้ว ยังได้ความรู้เพิ่มเติมในการที่เขาอาจจะคิดแก้ไขและปรับปรุงโดยใช้วิธีต่างๆที่ต้องใช้แนวความคิดของตัวเองและทฤษฎีต่างๆมาเพิ่มเติม

## บทที่ 2

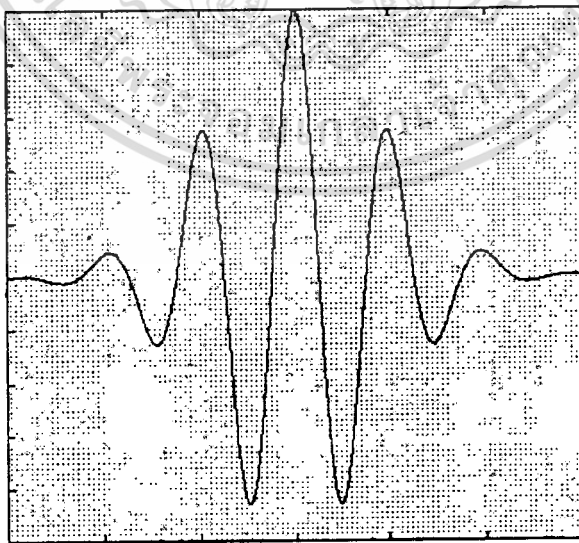
### ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต

ในบทนี้จะเป็นการอธิบายหลักการของการแปลงเวฟเล็ต(Wavelet Transform :WT)อย่างง่าย โดยจะกล่าวถึง ความหมาย ลักษณะทั่วไป ลักษณะทางฟิสิกส์ ความแตกต่างระหว่าง WT กับการแปลงข้อมูลแบบอื่นๆ เหตุผลและความจำเป็นในการพัฒนาสู่ WT เป็นต้น

#### 2.1 ลักษณะของเวฟเล็ต

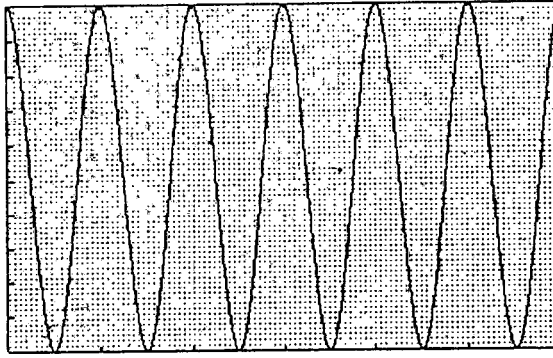
ทฤษฎีเวฟเล็ต(Wavelet Theory)เป็นคณิตศาสตร์ที่ใช้อธิบายการสร้างโมเดลของสัญญาณระบบ หรือ ขบวนการทางฟิสิกส์ใดๆ ซึ่งประกอบขึ้นด้วยเซตของสัญญาณเฉพาะหรือระบบย่อยๆ มารวมกันเป็นสัญญาณหรือระบบนั้นๆ สัญญาณเฉพาะนี้จะเป็นคลื่นเล็กๆ ซึ่งถูกเรียกว่า "เวฟเล็ต"(Wavelets) ลักษณะของเวฟเล็ตจะเป็นคลื่นที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง (Oscillatory) และขนาดของคลื่นจะลดลงสู่ศูนย์อย่างรวดเร็วทั้งสองด้าน ดังแสดงในรูปที่ 2.1

ในรูปที่ 2.1 เป็นตัวอย่างของเวฟเล็ตชนิดหนึ่งเรียกว่าเวฟเล็ตแบบ Morlet สังเกตว่า มีคุณสมบัติของเวฟเล็ตตามที่กล่าวมาทั้ง 2 เงื่อนไข คือ มีการออสซิลเลทในลักษณะของสัญญาณซายนูซอยด์ล(Sinusoidals) ดังแสดงในรูปที่ 2.2 ก) รวมกับเงื่อนไขของการลดลงอย่างรวดเร็วตามลักษณะฟังก์ชันหน้าต่าง(Windows Function) ดังแสดงในรูปที่ 2.2 ข)

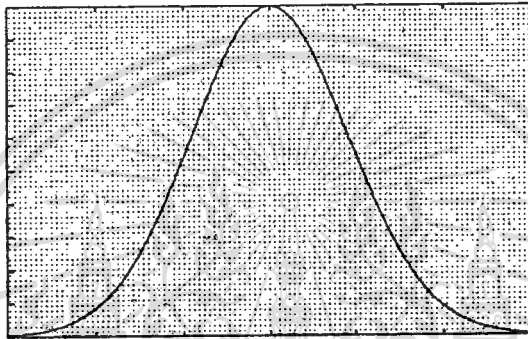


รูปที่ 2.1 แสดงตัวอย่างลักษณะของ Morlet :  $g(t) = e^{-\frac{t^2}{64}} \cos(t)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) แสดงลักษณะของสัญญาณ Sinusoidals Signal :  $\cos(t)$



ข) แสดงลักษณะของ Window function =  $e^{-\frac{t^2}{64}}$

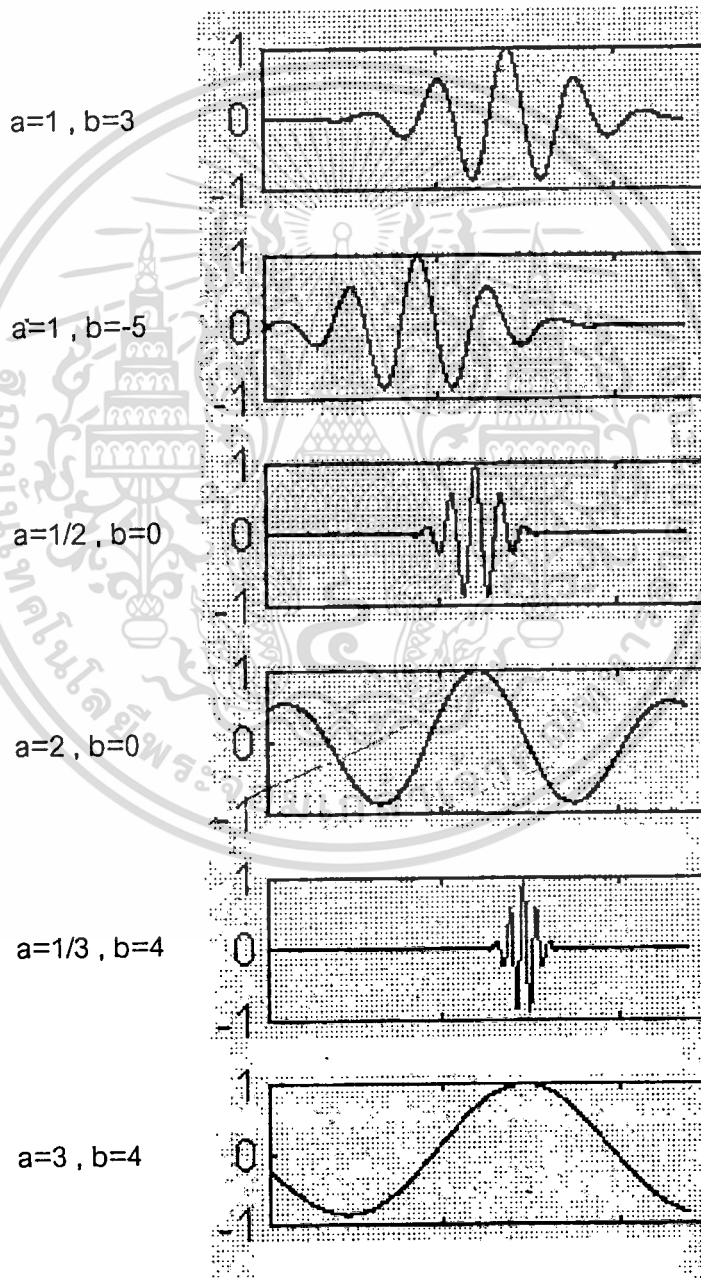
รูปที่ 2.2 แสดง ลักษณะเงื่อนไขทั้งสองของฟังก์ชันเวฟเล็ต

เมื่อทั้งสองเงื่อนไขเกิดขึ้นพร้อมกัน ก็จะทำให้เกิดคลื่นเล็กๆ ที่เรียกว่า "เวฟเล็ต" ดังนั้นการคูณกันของฟังก์ชันออสซิลเลท (Ossilate function) และฟังก์ชันการลดลง (Decay function) จะให้ผลเป็นคลื่นเวฟเล็ต ดังแสดงในรูปที่ 2.1

การนำเวฟเล็ตหลายๆ อันมารวมกันเป็นเซต เพื่อจะใช้ในการอธิบายสัญญาณใดๆ โดยที่คลื่นเวฟเล็ตแต่ละตัวจะมีโครงสร้างมาจากฟังก์ชันเดียวกัน ซึ่งฟังก์ชันนี้จะเป็นเวฟเล็ตต้นกำเนิดที่เรียกว่า เวฟเล็ตแม่ (Mother Wavelets) คลื่นเวฟเล็ตแต่ละอันจะอยู่ในเซตของเวฟเล็ตนี้ โดยคลื่นแต่ละคลื่นจะเกิดจาก "การสเกล" (Scaling : " $a$ ") และ "การเลื่อนตำแหน่ง" (Translation or Shifting : " $b$ ") ของเวฟเล็ตแม่ตัวนี้ การสเกลจะหมายถึง การหดเข้า (Compressing) หรือการขยายออก (Dilation) ในที่นี้จะใช้ " $a$ " แทนพารามิเตอร์ของการหดหรือขยาย ซึ่งจะเป็นการเปลี่ยนความถี่ของเวฟเล็ตแม่นั่นเอง ส่วนการเลื่อนตำแหน่งกำหนดโดยพารามิเตอร์ " $b$ " จะหมายถึงการเลื่อนตำแหน่งการเกิดของคลื่นเวฟเล็ตบนแกนเวลา ในรูปที่ 2.3 จะแสดงให้เห็นถึงเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า " $a$ ", " $b$ " ต่างๆ กัน โดยที่ยังคงอยู่ในเซตของเวฟเล็ตที่สัมพันธ์กับเวฟเล็ตแม่ในรูปที่ 2.1 ซึ่งจะสังเกตเห็นว่าคลื่นเวฟเล็ตแต่ละคลื่นจะมีรูปร่างเหมือนเดิม โดยยังไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คงมีจำนวนไซเคิลของการออสซิลเลทเท่าเดิม จะเปลี่ยนไปเฉพาะความถี่และตำแหน่งการเกิดเท่านั้น นอกจากนี้เวฟเลิตที่ถูกสเกลจะต้องทำการ Normalize ด้วย  $1/\sqrt{a}$  เสมอ เพื่อให้เวฟเลิตที่ถูกสเกลไปแล้วยังคงมีพลังงานเช่นเดียวกับเวฟเลิตแม่ ดังนั้นถ้าให้  $g(t)$  เป็นฟังก์ชันเวฟเลิตแม่ จะสามารถเขียนเป็นสมการทั่วไปของเวฟเลิตที่ตำแหน่ง " $a$ ", " $b$ " ใดๆ ซึ่งสัมพันธ์กับเวฟเลิตแม่ได้ดังนี้

$$g_{b,a}(t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) \quad (2.1)$$

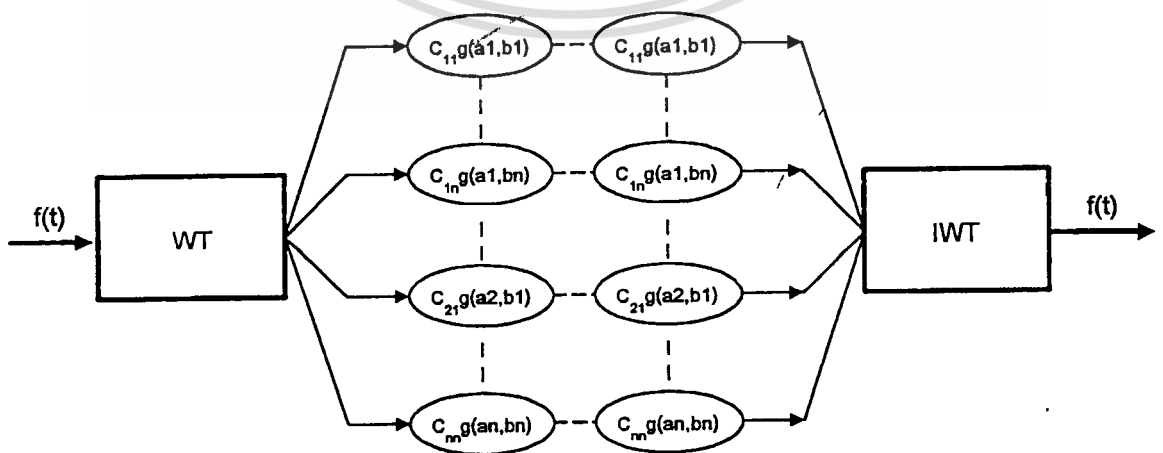


เอกสารนี้รูปที่ 2.3 แสดงลักษณะของเวฟเลิตแม่ที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า  $a, b$  ต่างๆ กัน ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ทฤษฎีพื้นฐานเวฟเล็ตในการสเกลและการเลื่อนตำแหน่ง

หากสามารถอธิบายลักษณะของสิ่งใดสิ่งหนึ่งได้จากลักษณะของสมาชิกเล็กๆ ที่เป็นองค์ประกอบแล้ว ทฤษฎีเวฟเล็ตก็เป็นอีกวิธีหนึ่งที่ใช้ในการอธิบายลักษณะของสิ่งต่างๆ หรือระบบใดๆ ได้เช่นเดียวกัน เช่น การนำเวฟเล็ตไปใช้อธิบายการแก้ปัญหาสมการดิฟเฟอเรนเชียลที่จำลองระบบใดระบบหนึ่ง การเด่นของหัวใจและการไหลเวียนของเลือดผ่านร่างกาย เป็นต้น ในการที่จะใช้สมาชิมาอธิบายสิ่งเหล่านี้ ก็จะมีวิธีการหาค่าสมาชิกเหล่านี้หลายวิธี การเลือกที่จะใช้วิธีใดก็ขึ้นอยู่กับว่าจะนำไปประยุกต์ใช้กับงานนั้นแล้วได้ประสิทธิภาพเพียงใด เช่น การใช้ระบบเลขฐานสิบจะใช้ในระบบบัญชีได้ เหมาะสมกว่าระบบเลขฐานสองหรือฐานสิบหก ในขณะที่ระบบเลขฐานสองจะให้ข้ออธิบายในระบบคอมพิวเตอร์ได้ดีกว่า ดังนั้นการใช้ทฤษฎีเวฟเล็ตให้ได้ดีก็จะขึ้นอยู่กับงานที่จะนำไปประยุกต์ใช้ด้วย

ทฤษฎีเวฟเล็ตจะใช้ในการอธิบายสิ่งใดสิ่งหนึ่ง เหมือนการแตกสิ่งเหล่านั้นออกเป็นส่วนประกอบเล็กๆ ที่สัมพันธ์กันคล้ายๆ กับภาพต่อ (Jig-saw) โดยที่ชิ้นส่วนเหล่านี้จะอยู่ในรูปของเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งและมีค่าถ่วงน้ำหนักต่างๆ กันขึ้นอยู่กับเวฟเล็ตแต่ละตัว ดังนั้นจึงเปรียบเสมือนว่าสัญญาณใดๆ จะสามารถสร้างขึ้นมาได้ โดยมีคลื่นเวฟเล็ตเป็นฟังก์ชันพื้นฐาน (Basis function : b.f.) หรือองค์ประกอบพื้นฐาน ขบวนการแตกกระจายชิ้นส่วนนี้เรียกว่า การแตกกระจายเวฟเล็ต (Wavelet Decomposition) ซึ่งก็คือ การทำ WT นั่นเอง ในทำนองเดียวกัน การรวมกลับเวฟเล็ต (Wavelet Reconstruction) ซึ่งเป็นการแปลงกลับเวฟเล็ต (Inverse Wavelet Transform : IWT) ก็จะเป็นการนำส่วนประกอบย่อยๆ เหล่านี้มาเรียงต่อกันเพื่อประกอบเป็นสัญญาณเดิม ดังแสดงในรูปที่ 2.4 เป็นการแตกกระจายสัญญาณ  $f(t)$  ออกเป็นคลื่นเวฟเล็ตที่มีความถี่และตำแหน่งต่างกันจำนวน  $N \times N$  ตัว



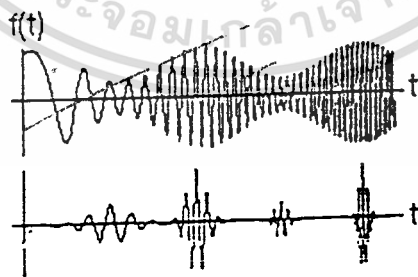
รูปที่ 2.4 แสดงลักษณะของการแตกกระจายสัญญาณและรวมกลับสัญญาณของเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WT สามารถเทียบเคียงกับขบวนการทางฟิสิกส์เพื่อให้เกิดความเข้าใจมากยิ่งขึ้น โดยเทียบเคียงกับระบบกล้องดูดาว ถ้าให้ท้องฟ้าเป็นสัญญาณอินพุตและเลนส์เป็นเวฟเลิตแม่ ในกรณีที่ต้องการดูภาพท้องฟ้าทั้งหมด โดยไม่ต้องมีการเลื่อนตำแหน่งของกล้อง จำเป็นจะต้องใช้โฟกัสที่หยาบมาก เพื่อให้ได้ภาพที่ครอบคลุมท้องฟ้าทั้งหมด ซึ่งภาพที่ได้จะไม่ให้รายละเอียดอะไรเลย แต่ถ้าทำการซูม(สเกล) เข้าไป จะทำให้เห็นรายละเอียดต่างๆ ของท้องฟ้ามากขึ้น เช่น กลุ่มดาว กาแล็คซี่ ฯลฯ แต่ก็เห็นรายละเอียดเป็นบางส่วนของท้องฟ้า ถ้าต้องการเห็นท้องฟ้าทั้งหมด ก็จะต้องทำการเลื่อนกล้องไปยังตำแหน่งต่างๆ ยิ่งถ้าต้องการรายละเอียดของท้องฟ้ามากเท่าไร ก็จะต้องมีการซูมลึกเข้าไปและจะต้องเลื่อนตำแหน่งของกล้องละเอียดมากยิ่งขึ้น จึงจะได้ภาพท้องฟ้าทั้งหมด

ดังนั้นภาพแต่ละภาพก็จะเป็นสัมประสิทธิ์ของการทำ WT ที่พิจารณาผ่านเลนส์ หรือ เวฟเลิตแม่ที่ค่าการสเกล (Scale :  $a$ ) (โฟกัส หรือ ความละเอียด) และ ตำแหน่งการมองดู (Translation :  $b$ ) ในขณะนั้น ถ้านำเซตของภาพทั้งหมดที่ตำแหน่งของ  $a, b$  ต่างๆ กันมารวมกัน ก็จะเป็นภาพท้องฟ้าได้(หมายถึง เมื่อนำสัญญาณเล็กๆ ที่ระดับความละเอียด  $a$  ซึ่งมี  $b$  หลายๆ ตำแหน่งมารวมกัน จะเกิดเป็นสัญญาณอินพุตที่ระดับความละเอียดนั้น เมื่อนำสัญญาณอินพุตที่ทุกความละเอียดมารวมกันก็จะได้เป็นสัญญาณอินพุตจริง : Multiresolution Analysis)

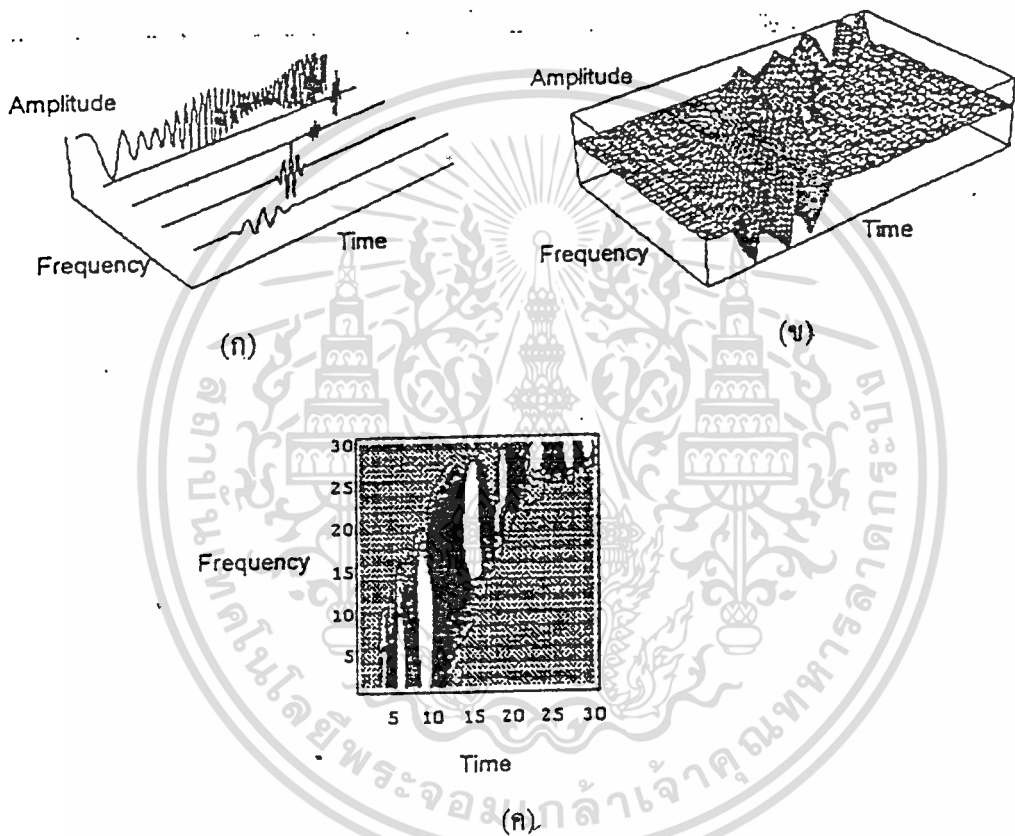
ถ้าให้  $f(t)$  เป็นสัญญาณที่ประกอบด้วยคลื่นหลายความถี่ที่เกิดขึ้นบนตำแหน่งต่างๆ บนแกนเวลา ดังแสดงในรูปที่ 2.5 ภาพบน ส่วนในภาพด้านล่างจะแสดงถึงการดึงสัญญาณ  $f(t)$  ที่ตำแหน่งเวลาใดๆ ออกมา



รูปที่ 2.5 แสดงลักษณะของสัญญาณที่มีหลาย ความถี่ที่ตำแหน่งเวลาต่างๆ กัน

ถ้าทำการพิจารณาในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา ในอีกแกนหนึ่งเป็นแกนความถี่ จะสามารถทำการแยกสัญญาณที่ตำแหน่งเวลาและมีความถี่ใดๆ ได้แสดงในรูปที่ 2.6 ก.)  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ในรูปที่ 2.6 ข.) จะพิจารณาที่ค่าเวลาและค่าความถี่ที่มีความต่อเนื่อง ซึ่งก็จะได้เป็นสัญญาณพื้นผิวที่ต่อเนื่องมากขึ้น ในรูปที่ 2.6 ค.) จะเป็นการพิจารณามองจากด้านบนของรูปที่ 2.6 ข) ซึ่งจะเห็นว่าค่าความเข้มจะแทนขนาดของสัญญาณ



รูปที่ 2.6

- ก.) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติแบบแบ่งช่วง  
 ข.) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติแบบต่อเนื่อง  
 ค.) แสดงลักษณะของสัญญาณเมื่อพิจารณาจากด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสเกลและการเลื่อนตำแหน่งจะถูกเรียกรวมกันว่า *Affine Operation* ซึ่งจะสามารถเขียนเป็น สมการคณิตศาสตร์ในลักษณะการส่ง (Mapping) ตัวแปรอิสระ  $t$  ไปยังตัวแปรอิสระตัวใหม่  $t'$  ดังแสดงในสมการที่ 2.2

$$t' = \frac{t-b}{a} \quad (2.2)$$

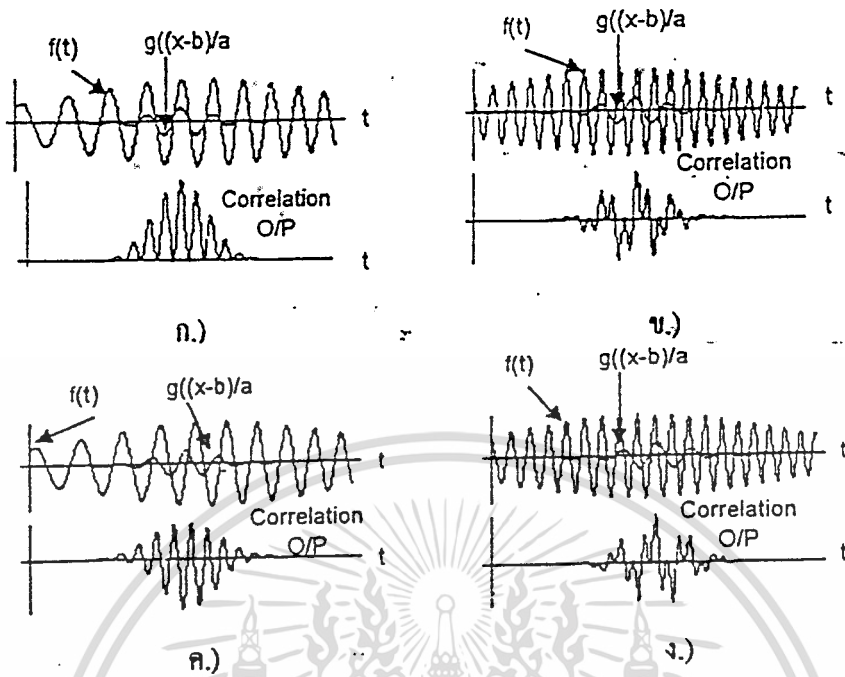
จากการที่สามารถทำการสร้างเซตของเวฟเล็ตได้จากการสเกลและการเลื่อนตำแหน่งของเวฟเล็ตแม่ ดังนั้นจะสามารถเขียนสมการสำหรับการแปลงเวฟเล็ตแบบต่อเนื่อง (Continuous Wavelets Transform : CWT) ที่ใช้ในการกระจาย (Decomposition) สัญญาณของฟังก์ชัน  $f(t)$  ด้วยเวฟเล็ตแม่  $g(t)$  เป็นดังนี้

$$(W_g f)(a, b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) g^* \left( \frac{t-b}{a} \right) dt \quad (2.3)$$

โดยที่ "\*" หมายถึง Complex Conjugate "a" เป็นพารามิเตอร์ของการสเกล และ "b" จะเป็นพารามิเตอร์ของการเลื่อนตำแหน่ง สัญลักษณ์  $W_g f(a, b)$  จะหมายถึง สัญญาณอินพุตที่ขึ้นอยู่กับตัวแปรอิสระ  $t$  จะถูกส่ง (Map) ไปเป็นฟังก์ชัน 2 มิติที่ขึ้นอยู่กับตัวแปรอิสระ  $a$  และ  $b$  โดยที่  $W_g f(a, b)$  จะเป็นค่าสัมประสิทธิ์ของเวฟเล็ต (Wavelet Coefficient) เวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่งจะใช้เป็นตัวแทนสัญญาณได้ สัมประสิทธิ์ของเวฟเล็ตที่ตำแหน่ง  $a, b$  จะเป็นค่าแสดงความเหมือนกัน (Degree of Correlation) ระหว่างเวฟเล็ตแม่ที่ถูกสเกล ( $a$ ) กับสัญญาณอินพุตที่ตำแหน่งเวลานั้น ( $b$ ) ดังแสดงในรูปที่ 2.7

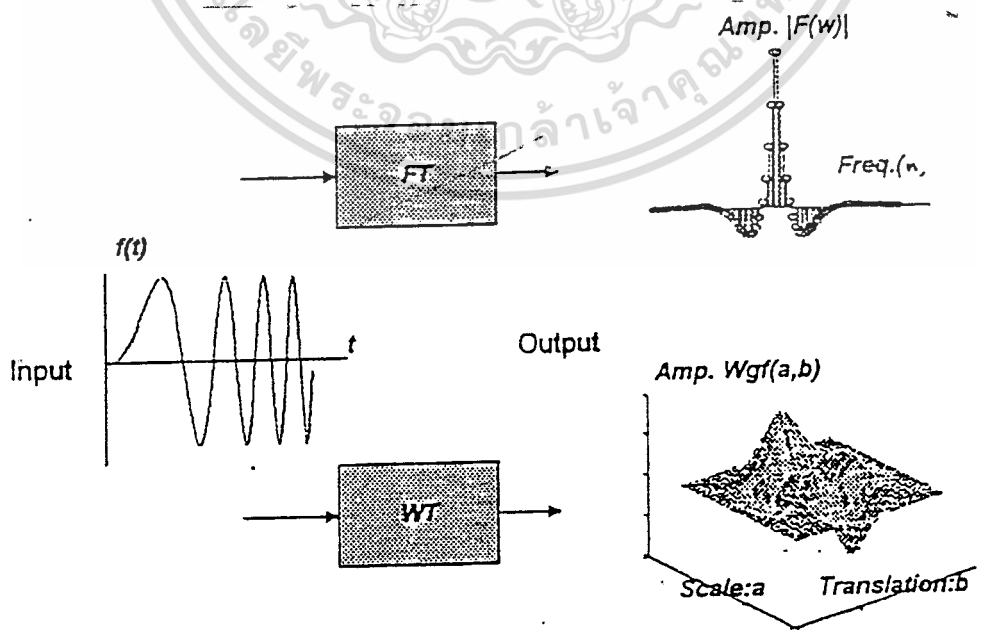
พิจารณารูปที่ 2.7 ก.) เปรียบเทียบกับรูปที่ 2.7 ข.) จะเห็นว่าในรูปที่ 2.7 ก.) คลื่นเวฟเล็ตจะมีความถี่และเฟสใกล้เคียงกับสัญญาณอินพุตมาก ทำให้ปรากฏคลื่นเอ้าท์พุทออกมามีค่าสูง แต่ในรูปที่ 2.7 ข.) คลื่นอินพุตจะมีค่าความถี่สูงกว่าคลื่นเวฟเล็ตทำให้เอ้าท์พุทมีค่าต่ำกว่ามาก และในขณะเดียวกันเมื่อลองพิจารณาเปรียบเทียบรูปที่ 2.7 ก.) กับ 2.7 ค.) ซึ่งสัญญาณอินพุตที่ความถี่เดียวกันแต่ทำการเปรียบเทียบที่ตำแหน่งต่างกัน โดยในรูปที่ 2.7 ก.) จะทำที่ตำแหน่งที่เฟสตรงกัน ซึ่งก็จะให้ผลของการ Correlation ที่สูงกว่าในรูปที่ 2.7 ค.) ซึ่งกระทำที่ตำแหน่งเฟสไม่ตรงกัน ส่วนในรูปที่ 2.7 ง.) ก็คล้ายกับรูปที่ 2.7 ข.) เพียงแต่เลื่อนตำแหน่งการเปรียบเทียบออกไป 180 องศา ซึ่งจะสังเกตเห็นว่าจะให้ผลลัพธ์ที่ต่างกัน ดังนั้นจึงสามารถสรุปได้ว่า ถ้าคลื่นเวฟเล็ตมีความคล้ายกับสัญญาณอินพุตมากเท่าไรทั้งในด้านของความถี่ ลักษณะสัญญาณและตำแหน่งเวลาในการเปรียบเทียบ ก็จะทำให้ได้ค่าของสัมประสิทธิ์ของการทรานส์ฟอร์มที่มากนั่นเอง เพราะฉะนั้น การทำ WT จะเป็นการอธิบายสัญญาณ  $f(t)$  ในรูปแบบของเวฟเล็ตโดเมนส์ (Wavelet Domain) เทียบกับคลื่นเวฟเล็ตที่ค่า  $a, b$  ไตๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงลักษณะของผลของการ Correlation ระหว่าง  $f(t), g((t-b)/a)$

ในรูปที่ 2.8 จะแสดงการเปรียบเทียบลักษณะผลลัพธ์เมื่อนำสัญญาณอินพุต 1 มิติ มาทำการแปลงฟูรีเยร์ (Fourier Transform :FT) กับการใช้ WT จะเห็นว่า FT เป็นการส่งสัญญาณจากโดเมนส์เวลา (1 มิติ) ไปเป็นโดเมนส์ความถี่ (1 มิติ) แต่ WT จะเป็นการส่งไปเป็น " ระนาบเวลาและความถี่ " (Time-frequency plane) ซึ่งมี 2 มิติ



รูปที่ 2.8 แสดงลักษณะความแตกต่างของสัญญาณเอาร์ทพุทระหว่าง FT กับ WT

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 "ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้"

## 2.3 ความเป็นมาของการแปลงเวฟเล็ด

FT เป็นการทรานส์ฟอร์มที่นิยมใช้กันมาก ซึ่งเป็นการแปลง สัญญาณ  $f(t)$  จากโดเมนส์เวลาไปเป็นสัญญาณ  $f(\omega)$  ซึ่งอยู่ในโดเมนส์ความถี่ หรือที่เรียกกันทั่วไปว่าเป็นการหาสเปกตรัม (Spectrum) ของสัญญาณ  $f(t)$  นั้นเอง โดยมีสมการเป็น

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (2.4)$$

จะสังเกตพบว่าข้อมูลที่ได้จากการทรานส์ฟอร์มจะมีตลอดย่านความถี่  $(-\infty, \infty)$  ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของสัญญาณอินพุตที่ช่วงเวลาใดๆ ก็ส่งผลกระทบต่อตลอดย่านของสเปกตรัม  $f(\omega)$  ด้วยนอกจากนี้การใช้ FT ยังไม่สามารถเลือกทำการวิเคราะห์สัญญาณเฉพาะช่วงเวลาหรือเฉพาะช่วงความถี่ใดๆ ซึ่งบางครั้งต้องการวิเคราะห์สัญญาณเพียงแค่ว่าเฉพาะช่วงเวลาหรือช่วงความถี่ที่เปลี่ยนแปลงที่ไม่คงที่ (Non-Stationary Signal) ซึ่งถ้าใช้ FT จะทำให้สิ้นเปลืองเวลาในการคำนวณ เนื่องจากมีข้อมูลมากเท่านั้น ดังนั้น FT จึงให้ข้อมูลที่มากเกินไปจนความจำเป็น นอกจากนั้นในทางปฏิบัติสัญญาณอินพุตจะมีมากเกินไปในการคำนวณใหม่แต่ละครั้ง ดังนั้น FT จึงเหมาะสำหรับสัญญาณที่มีการเปลี่ยนแปลงที่คงที่ (Stationary Signal)

Gabor Transform : GT เป็นทรานส์ฟอร์มที่ได้พัฒนาขึ้นเป็นแบบแรก เพื่อแก้ปัญหาที่กล่าวมาข้างต้น โดยจะเป็นการทำ FT เฉพาะช่วงเวลาและช่วงความถี่ที่กำหนดผ่านฟังก์ชันหน้าต่าง (Window function : w.f.) ซึ่ง w.f. นี้จะใช้ Gaussian function เท่านั้น

$$g_\alpha(t) = \frac{1}{\sqrt{2\pi}\alpha} e^{-\frac{t^2}{4\alpha}} \quad (2.5)$$

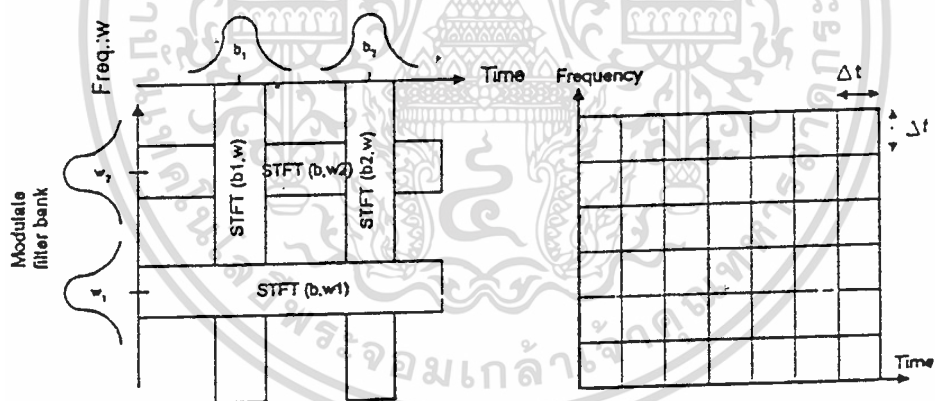
$$(G_b^\alpha f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t} f(t)) g_\alpha(t-b) dt \quad (2.6)$$

สมการที่ 2.5 เป็น Gaussian window function และสมการที่ 2.6 เป็นสมการของ GT จะเห็นว่าการทรานส์ฟอร์มแบบนี้จะสามารถเลือกตำแหน่งการวิเคราะห์สัญญาณ โดยกำหนดที่พารามิเตอร์  $b$  และเลือกช่วงความถี่ด้วยพารามิเตอร์  $\alpha$  แต่เนื่องจาก GT ใช้ w.f. แบบเดียว ซึ่งอาจจะไม่เหมาะสมกับการประยุกต์ใช้งานในด้านต่างๆ ดังนั้นจึงได้มีการพัฒนาต่อมาเป็น Short Time Fourier Transform : STFT ซึ่งจะสามารถทำการเลือก w.f.( $w(t)$ )ที่จะใช้ในการวิเคราะห์สัญญาณได้ ทำให้เกิดความยืดหยุ่นและสามารถเลือกใช้ให้เหมาะสมกับสัญญาณอินพุตที่จะวิเคราะห์ได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(G_{\alpha}f)(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t} f(t)) \overline{w(t-b)} dt \quad (2.7)$$

สมการที่ 2.7 จะเป็นสมการ STFT โดยที่ตำแหน่งเวลาในการวิเคราะห์จะกำหนดโดยค่า "b" และช่วงความถี่ในการวิเคราะห์กำหนดด้วยความถี่ของ w.f. นั้นๆ แต่จะสังเกตเห็นว่าทั้ง GT และ STFT จะมีลักษณะของช่วงการวิเคราะห์ที่คงที่ดังแสดงในรูปที่ 2.9 ช่วงเวลาและช่วงความถี่คงที่ (Fixed resolution transform) ทั้งนี้เนื่องจาก w.f. ไม่มีการเปลี่ยนแปลงคุณสมบัติขณะทำการทรานฟอร์ม ซึ่งไม่เหมาะสมในทางปฏิบัติ เนื่องจากสัญญาณในทางปฏิบัติ นั้น สัญญาณความถี่สูงจะมีช่วงคาบเวลาที่แคบ ในขณะที่สัญญาณความถี่ต่ำจะมีช่วงคาบเวลาที่กว้างกว่า ดังนั้น GT และ STFT อาจจะให้ข้อมูลที่ได้จากการทรานฟอร์มออกมามากหรือน้อยเกินไป (ต้องทำการทรานฟอร์มหลายช่วงเวลามาประกอบกัน จึงจะได้ข้อมูลครบถ้วนในช่วงเวลาที่ต้องการ) ดังนั้นช่วงเวลาควรจะต้องการเปลี่ยนแปลงสัมพันธ์กับช่วงความถี่ที่จะทำการวิเคราะห์ โดยที่ช่วงความถี่ที่สูงขึ้นจะมีช่วงเวลาในการวิเคราะห์ที่น้อยลง และในขณะที่เป็นการวิเคราะห์ช่วงความถี่ต่ำจะต้องใช้ช่วงเวลาที่ยาวกว่าในการวิเคราะห์สัญญาณ



รูปที่ 2.9

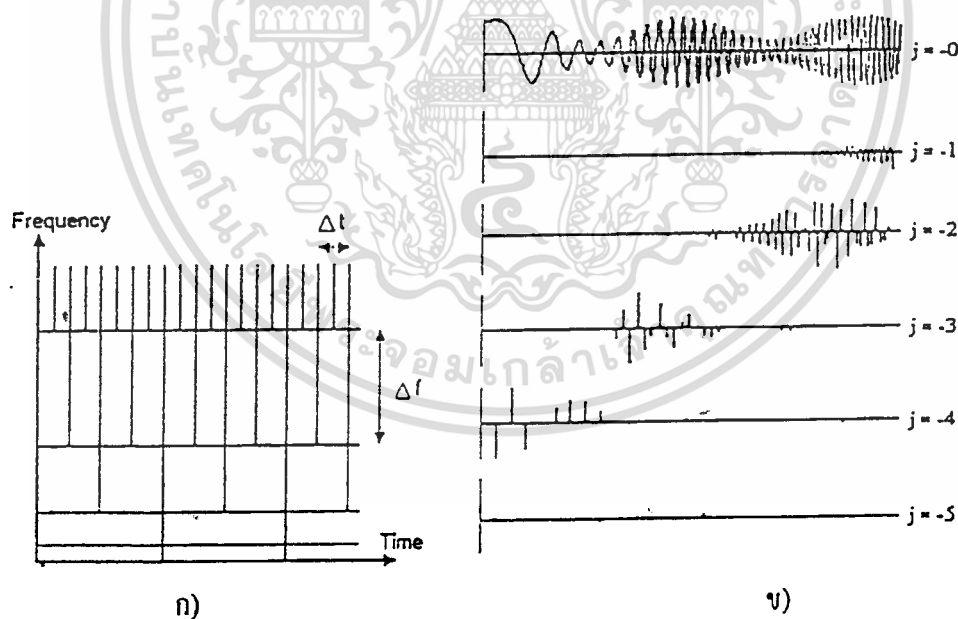
แสดงลักษณะของระนาบเวลาความถี่แบบคงที่ (Constant Time-Frequency plane)

จากเหตุผลดังกล่าวข้างต้นทำให้มีการพัฒนาเป็นการทรานฟอร์มในรูปแบบใหม่ที่เรียกว่า "การแปลงเวฟเล็ต" ซึ่งจะเป็นทรานฟอร์มผ่าน w.f. แต่ w.f. จะสามารถเลื่อนตำแหน่งและปรับเปลี่ยนช่วงเวลาให้เหมาะสมกับช่วงความถี่ที่จะวิเคราะห์ได้ โดยเรียก w.f. แบบนี้ว่า "ฟังก์ชันเวฟเล็ต" (Wavelet function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(W_g f)(a, b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) * \left(\frac{t-b}{a}\right) dt \quad (2.8)$$

$g(t)$  จะเป็นฟังก์ชันเวฟเล็ดแม่ ซึ่งจะถูกลื่อนตำแหน่งและถูกสเกล(เปลี่ยนความถี่) โดยพารามิเตอร์ “ $b$ ” และ “ $a$ ” ตามลำดับ โดยที่ช่วงเวลาและความถี่ในการทรานส์ฟอร์มจะสัมพันธ์กันในลักษณะที่กล่าวมาแล้วข้างต้น ดังแสดงในรูปที่ 2.10 ก.) ซึ่งจะเห็นว่าระนาบเวลาและความถี่จะมีลักษณะการเปลี่ยนแปลงช่วงเวลาและความถี่ที่สัมพันธ์กันคือ ในช่วงความถี่สูงจะใช้ช่วงเวลาในการวิเคราะห์เพียงสั้นๆ ( $\Delta f$  สูง  $\Delta t$  ต่ำ) แต่ในช่วงความถี่ต่ำจะมีช่วงเวลาในการวิเคราะห์ที่ยาวนานขึ้น ( $\Delta f$  ต่ำ  $\Delta t$  สูง) ซึ่งจะเหมาะสมกับลักษณะสัญญาณในทางปฏิบัติ ส่วนในรูปที่ 2.10 ข.) จะเป็นการแสดงลักษณะการแตกกระจายสัญญาณออกไปในระดับความละเอียด(ความถี่) ที่ต่างกันออกไป ขอให้พิจารณาเปรียบเทียบกับรูปที่ 2.7 จะพบว่ามีความสัมพันธ์กัน ในระดับ  $j = -1$  จะมีเฉพาะส่วนความถี่สูงและที่ระดับที่  $j$  ต่ำลงสัญญาณจะมีค่าความถี่ที่ต่ำลงด้วยสอดคล้องกับภาพของ ระนาบเวลาและความถี่ นอกจากนี้  $g(t)$  ยังมีได้หลายรูปแบบ แต่ต้องเป็นไปตามคุณสมบัติของ WT ดังนั้นจึงทำให้ WT มีความยืดหยุ่นและสามารถเลือกใช้ให้เหมาะสมกับงานที่นำไปใช้ได้เป็นอย่างดี



รูปที่ 2.10

ก.) แสดงลักษณะของระนาบเวลาและความถี่สำหรับ WT

ข.) แสดงลักษณะการแตกกระจายสัญญาณไปยังระดับความละเอียดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาทั้งหมดเป็นการพัฒนาการทรวนฟอร์มจาก FT มาสู่ WT ที่มีประสิทธิภาพที่สูงขึ้น

## 2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ตและการแปลงฟูรีเยร์ในลักษณะของฟังก์ชันหน้าต่าง

จากที่ได้กล่าวมาบ้างแล้วในหัวข้อที่ผ่านมา ในหัวข้อนี้จะกล่าวเปรียบเทียบ FT และ WT ในลักษณะของฟังก์ชันหน้าต่างให้ละเอียดขึ้น จากการที่ WT จะมีความสัมพันธ์กับ FT และอนุกรมฟูรีเยร์ (Fourier Series : FS) โดยที่ FT จะเป็นการอธิบายสัญญาณในลักษณะของผลรวมของการถ่วงน้ำหนักของฟังก์ชันเอกโพเนนเชียลที่ความถี่ต่างๆ กัน น้ำหนักที่แต่ละความถี่จะเป็นค่าสัมประสิทธิ์ของการทำ FT แต่ในกรณีของ WT จะเป็นการนำเสนอฟังก์ชันในรูปแบบการประกอบกันของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง ดังนั้นเวฟเล็ตแม่จะเปรียบได้กับเอกโพเนนเชียลฟังก์ชัน ส่วนการสเกลและการเลื่อนตำแหน่งจะเทียบได้กับการเลื่อนความถี่ (Frequency shifting) ขณะที่พื้นผิว 2 มิติของสัมประสิทธิ์เวฟเล็ตจะใช้แทนสัมประสิทธิ์ของฟูรีเยร์ที่เป็น 1 มิติ ดังนั้นถ้าให้เวฟเล็ตแม่เท่ากับ  $g(t) = e^{it}$  และให้ค่าการสเกล  $a = 1/\omega$  การเลื่อนตำแหน่ง  $b = 0$  จะสามารถเขียนเป็นสมการ WT ได้เป็น

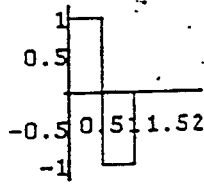
$$\begin{aligned} (W_g f)(a, b) &= W_g f\left(\frac{1}{\omega}, b\right) \\ &= \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt = F(\omega) \end{aligned} \quad (2.9)$$

ซึ่งจะเป็นสมการของ FT นั่นเอง ดังนั้นจะเทียบได้ว่า การสเกลเป็นส่วนกลับของการเปลี่ยนความถี่ที่จะพิจารณานั้นเอง จากสมการ FT จะเห็นว่า  $b = 0$  เสมอ และเนื่องจากฟังก์ชันหน้าต่างแบบเอกโพเนนเชียลทำการเลื่อนตำแหน่งไม่ได้ เพราะเป็นสัญญาณที่ไม่มีจุดเริ่มต้นและจุดสิ้นสุด  $(-\infty, \infty)$  จึงทำการเลื่อนตำแหน่งไม่ได้ต่างจากฟังก์ชันหน้าต่างของ WT ที่มีจุดเริ่มต้นและจุดสิ้นสุด หรือมีขอบเขตที่แน่นอน (Compact support) ดังนั้นจึงสามารถยกไปวางตำแหน่งใดก็ได้ทำให้สามารถเลือกตำแหน่งที่จะทำการพิจารณาสัญญาณได้

## 2.5 ลักษณะของเวฟเล็ตแม่แบบต่างๆ

เนื่องจาก w.f. ของ WT นั้นสามารถมีได้หลายรูปแบบ เพื่อให้สามารถเลือกใช้งานได้อย่างเหมาะสม แต่จะต้องมีเงื่อนไขตามคุณสมบัติของ WT ในหัวข้อนี้จะแสดงลักษณะของฟังก์ชันเวฟเล็ตแม่ชนิดต่างๆ ที่นิยมใช้ในปัจจุบัน

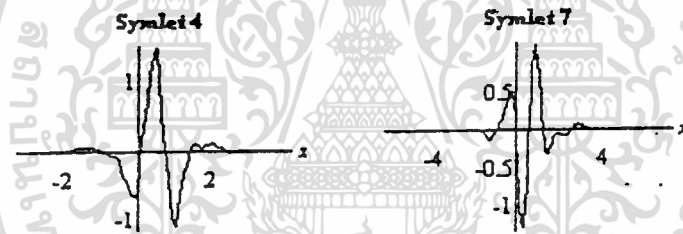
Harr wavelet



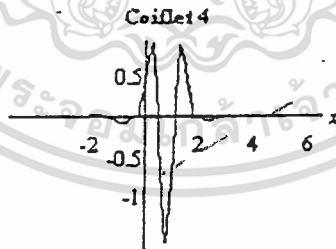
Daubechies wavelet



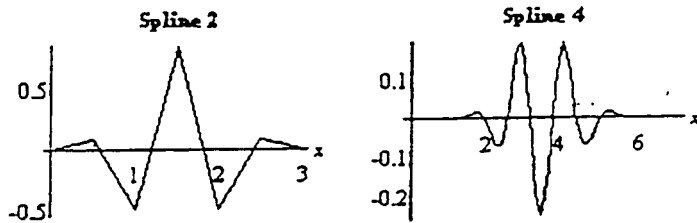
Symlet wavelet



Coiflet wavelet



Spline wavelet



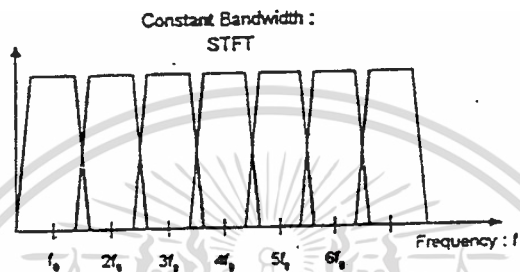
รูปที่ 2.11 แสดงลักษณะเวฟเล็ตแม่ชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การแปลงเวฟเล็ิต

ในที่นี้จะขอแบ่งการแปลงเวฟเล็ิตออกเป็น 2 แบบ คือ การแปลงเวฟเล็ิตแบบต่อเนื่องและ การแปลงเวฟเล็ิตแบบแบ่งช่วง

### 2.6.1 การแปลงเวฟเล็ิตแบบต่อเนื่อง



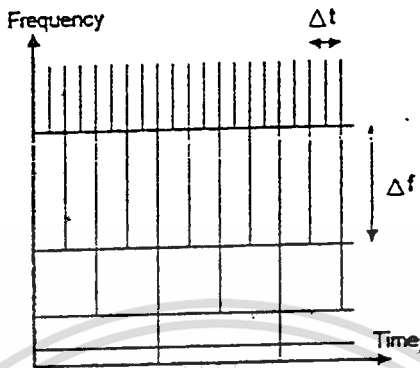
รูปที่ 2.12

แสดงลักษณะของ Filter banks ในแบบของ STFT

จากการวิเคราะห์สัญญาณด้วย STFT จะสังเกตเห็นว่า Time-frequency window จะมีพื้นที่ในการวิเคราะห์ที่คงที่เสมอไม่ว่าจะเป็นการวิเคราะห์ที่ความถี่และช่วงเวลาใดๆ ขนาดช่วงความกว้างของความถี่และช่วงเวลาจะขึ้นอยู่กับลักษณะของ w.f. ที่ใช้ ซึ่งจะไม่มีการเปลี่ยนคุณสมบัติในขณะการทรานฟอร์ม แต่ในทางปฏิบัติความถี่ของสัญญาณจะเป็นสัดส่วนโดยตรงกับจำนวนไซเคิลต่อหนึ่งหน่วยเวลา ดังนั้นที่ความถี่สูงจะมีการเปลี่ยนแปลงสัญญาณในช่วงเวลาที่แคบต่างกับสัญญาณความถี่ต่ำที่ต้องใช้ช่วงเวลานานกว่าดังนั้นถ้าใช้ STFT และเลือก w.f. ที่มีช่วงความถี่และช่วงเวลาไม่เหมาะสม เช่นช่วงเวลาที่แคบเกินไปก็จะทำให้ต้องมีการทำการทรานฟอร์มหลายครั้งทีละตำแหน่งเพื่อให้ได้ข้อมูลครบในช่วงเวลาทั้งหมดที่ต้องการหรือถ้าความกว้างของช่วงเวลาของ w.f. อาจจะมีมากเกินไปทำให้ได้รับข้อมูลมากเกินไปทำให้ได้รับข้อมูลมากเกินไปจนความจำเป็นดังนั้น w.f. ที่ดีความกว้างของช่วงเวลาและช่วงความถี่จะต้องมีการเปลี่ยนแปลงเป็นสัดส่วนผกผันที่สัมพันธ์กัน นั่นคือควรจะให้ Time-window ที่แคบแต่มี Frequency-window ที่กว้างเมื่อทำการวิเคราะห์สัญญาณที่ความถี่สูงและ Time-window ที่กว้างแต่มี Frequency-window ที่แคบเมื่อใช้วิเคราะห์สัญญาณที่ความถี่ต่ำ ซึ่งจะทำให้ได้รับข่าวสารที่ครบถ้วนและใช้เวลาในการทรานฟอร์มที่น้อยลงจากที่กล่าวมาทั้งหมด STFT จึงไม่เหมาะสมสำหรับการวิเคราะห์สัญญาณที่มีการเปลี่ยนความถี่สูงและต่ำในสัญญาณตัวเดียวกัน แต่ใน WT จะอาศัย w.f. ที่มีคุณสมบัติที่สามารถเปลี่ยนแปลงคุณสมบัติการเป็นเอกลักษณ์สำหรับงานเพื่อการศึกษาเท่านั้น ไม่นับว่าให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สมบัติของช่วงเวลาและช่วงความถี่ที่เหมาะสมสำหรับการวิเคราะห์สัญญาณที่ช่วงความถี่และช่วงเวลาที่สนใจซึ่ง w.f. ที่มีคุณสมบัติดังกล่าว เรียกว่า "ฟังก์ชันเวฟเล็ต"



รูปที่ 2.13

แสดงลักษณะของระนาบเวลาและความถี่ของการแปลงเวฟเล็ต

จากรูปที่ 2.13 จะเห็นว่า  $\psi(t)$  (Basis Wavelet) ที่มีค่าความถี่สูง ( $\alpha$  มีค่าน้อยๆ) จะทำให้ช่วงความกว้างของ Time-window ต่ำลง แต่จะส่งผลให้ความกว้างทางความถี่ของ Frequency-window เพิ่มขึ้น และ ในขณะที่  $\alpha$  มีค่ามากขึ้น ก็จะมีผลต่อการเปลี่ยนแปลงในทางตรงกันข้ามกัน ซึ่งการเปลี่ยนแปลงในลักษณะนี้จะเหมาะสมกับสัญญาณในทางปฏิบัติ

### 2.6.2 การแปลงเวฟเล็ตแบบแบ่งช่วง

ในหัวข้อนี้จะอธิบายการทำ WT กับสัญญาณแบบแบ่งช่วง (Discrete signal) ซึ่งเป็นสัญญาณในทางปฏิบัติที่จะถูกวิเคราะห์ด้วยระบบคอมพิวเตอร์ สัญญาณอินพุตจะมีลักษณะเป็นข้อมูลการสุ่ม (Sample data)

#### ➤ การวิเคราะห์สัญญาณแบบหลายระดับความละเอียด (Signal Analysis with Multiresolution)

ก่อนที่จะศึกษาลักษณะสมบัติของการวิเคราะห์สัญญาณแบบ Multiresolution :MRA จะอธิบายถึงทฤษฎีพื้นฐานของสเปซของเวกเตอร์ (Vector space) ก่อน เนื่องจากเป็นสิ่งสำคัญที่มีความเกี่ยวข้องกับการวิเคราะห์แบบ MRA

## สเปซของเวกเตอร์

ความหมายของ "สเปซของเวกเตอร์" ในสาขาวิเคราะห์สัญญาณ คือ สเปซหรือปริภูมิของสัญญาณใดๆ ที่เกิดจากการรวมกันของสัญญาณพื้นฐานย่อยๆ (Basis function : b.f.)

ถ้ากำหนดให้  $a, b$  เป็นจำนวนจริงใดๆ และ  $\hat{i}, \hat{j}$  เป็นเวกเตอร์หนึ่งหน่วย (Unit vector) ในสเปซของเวกเตอร์  $V$  จะได้ว่า  $a\hat{i} + b\hat{j} \in V$  ซึ่งหมายความว่าเวกเตอร์ใดๆ ก็ตามจะต้องอยู่ในสเปซของเวกเตอร์  $V$  เสมอ ดังนั้นถ้าเวกเตอร์ใดประกอบขึ้นจากหลายเวกเตอร์หนึ่งหน่วยจะได้ว่าเวกเตอร์นั้นยังคงอยู่ในสเปซของเวกเตอร์นี้ นั่นคือ  $a\hat{i} + b\hat{j} + c\hat{k} + \dots \in V$  เวกเตอร์หนึ่งหน่วย  $\hat{i}, \hat{j}$  จะมีลักษณะเป็นฟังก์ชันที่เป็นเชิงตั้งฉากปกติ (Orthonormal: o.n.) ซึ่งกันและกัน หมายความว่า  $\hat{i}, \hat{j}$  จะอยู่กันต่างมิติ หรือ อาจจะกล่าวได้ว่าอยู่บนแกนต่างกัน ดังนั้นถ้าพิจารณาจากผลของ Inner product ระหว่าง  $\hat{i}, \hat{j}$  จะมีค่าเป็นศูนย์เสมอ แต่ถ้าเป็นการ Inner product ระหว่างตัวมันซึ่งเป็นการหาขนาดของมันด้วยจะมีค่าเป็นหนึ่งเสมอ ในขณะที่  $a, b$  เป็นสัมประสิทธิ์ที่จะแสดงขนาดในแต่ละทิศทางของเวกเตอร์หนึ่งหน่วย

ดังนั้นถ้าพิจารณาในลักษณะของสัญญาณ อาจจะมีมองว่า  $\hat{i}, \hat{j}$  คือ b.f. ที่เป็นสัญญาณเล็กๆ ซึ่งสามารถนำมาประกอบกันเป็นสัญญาณใดๆ ดังนั้นถ้าให้  $U_i$  เป็น o.n.b.f. ใดๆ และ  $C_i$  เป็นค่าสัมประสิทธิ์ของ  $U_i$  จะได้ว่า สัญญาณ  $v$  ใดๆ จะประกอบขึ้นจาก o.n.b.f. ที่มีขนาดต่างๆ มารวมกัน แสดงในสมการที่ 2.10

$$v = \sum_i C_i U_i \quad ; v \in V \quad (2.10)$$

ในการประมวลผลสัญญาณด้วยคอมพิวเตอร์นั้น สัญญาณดังกล่าวควรจะมีจำนวน b.f. ที่จำกัด (Finite basis function) เนื่องจากข้อจำกัดของทรัพยากรของระบบ ถ้าให้  $k$  เท่ากับจำนวนของ b.f. ที่น้อยที่สุดที่ประกอบขึ้นมาเป็นสัญญาณชนิดนั้น ก็จะทำให้ได้สัญญาณ  $v$  ที่มีขนาดที่จำกัด (Finite Dimension =  $C_1 U_1, C_2 U_2, \dots, C_k U_k$ )

## การวิเคราะห์สัญญาณแบบหลายระดับความละเอียด

สมมติให้  $V^j$  เป็นสเปซของเวกเตอร์ที่มี  $j$  แสดงถึงระดับความละเอียดและจำนวนของ b.f. ที่ประกอบขึ้นเป็นฟังก์ชันนั้น ถ้า  $j$  มีค่าสูงขึ้นก็แสดงว่าที่ระดับความละเอียดสูงขึ้นจะมีจำนวนของ b.f. มากขึ้นด้วย ทำให้สัญญาณที่เกิดจากการประกอบกันจาก b.f. เหล่านี้มีความละเอียดมากยิ่งขึ้นด้วย ดังนั้นจึงอาจกล่าวได้ว่า  $j$  เป็นค่าที่แสดงถึงระดับความละเอียดของสัญญาณนั่นเอง จากข้อ

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเหล่านี้ทำให้สามารถสรุปเป็นลักษณะสมบัติของการวิเคราะห์สัญญาณที่ระดับความละเอียดต่างๆ ได้ดังนี้

$$1) V^{-\infty} \dots \subset V^{-1} \subset V^0 \subset V^1 \dots \subset V^{\infty}$$

$$2) \text{Close}_{L^2} \left( \bigcup_{j \in \mathbb{Z}} V^j \right) = L^2(\mathbb{R}) \quad ; \mathbb{R} = \text{เซตของจำนวนเต็ม}$$

$$3) \left( \bigcap_{j \in \mathbb{Z}} V^j \right) = \{0\}$$

$$4) V^j + W^j = V^{j+1} \quad ; j \in \mathbb{Z}; \mathbb{Z} = \text{เซตของจำนวนเต็ม}$$

$$5) f(x) \in V^j \Leftrightarrow f(2x) \in V^{j+1} \quad ; j \in \mathbb{Z}$$

ความหมายของคุณสมบัติแต่ละข้อจะอธิบายได้ดังนี้

(1) หมายถึง สเปซของเวกเตอร์ที่มีระดับความละเอียดสูงกว่าจะสามารถพิจารณาให้มีระดับความละเอียดต่ำลงได้ โดยมีความสัมพันธ์กันเป็นโครงข่ายในลักษณะของเซตย่อย ซึ่งจะหมายถึง สัญญาณใดๆก็ตามจะสามารถพิจารณาที่ระดับความละเอียดใดๆ ก็ได้ โดยที่ยังคงอยู่ภายใต้สัญญาณเดิมนั่นเอง

(2) ฟังก์ชันทุกฟังก์ชันใน  $L^2(\mathbb{R})$  สามารถถูกประมาณให้อยู่ในสเปซของเวกเตอร์ที่ระดับความละเอียด  $j$  ใดๆ ก็ได้ในลักษณะของการโปรเจกชันสัญญาณ  $f(t)$  ลงใน  $V^j$  การประมาณฟังก์ชันไปที่ระดับ  $j$  ใดๆ จะทำให้ข่าวสารของ  $f(t)$  สูญหายไปตามระดับความละเอียดนั้นๆ แต่ถ้าระดับความละเอียดเพิ่มขึ้นถึงอนันต์สัญญาณที่ประมาณนั้นยังคงลู่เข้าสู่สัญญาณต้นกำเนิด  $f(t) \in L^2(\mathbb{R})$

(3) การโปรเจกต์แต่ละครั้งที่ระดับ  $j$  น้อยลงเรื่อยๆ จะทำให้สัญญาณที่ประมาณมีข่าวสารที่สูญหายไปมากขึ้น ดังนั้นรายละเอียดของสัญญาณจะน้อยลงเรื่อยๆ จนลู่เข้าสู่ศูนย์

(4) การโปรเจกต์สัญญาณจากระดับหนึ่งไปยังระดับที่ต่ำกว่าหนึ่งระดับ จะทำให้ข่าวสารหรือพลังงานของสัญญาณส่วนหนึ่งหายไป หรือ อาจจะมองว่าแถบความถี่ของสัญญาณบางส่วนหายไป พลังงานหรือสัญญาณในแบนด์ความถี่ที่หายไปนี้ จะถูกรวมอยู่ใน Complementary subspace :  $W^j$

(5) คุณสมบัติของ Two scale property จะเป็นการลดระดับความละเอียดของสัญญาณลงมาหนึ่งระดับซึ่งหมายความว่าสัญญาณที่อยู่ใน  $V^{j+1}$  จะมีความถี่เป็นสองเท่าของสัญญาณใน  $V^j$

จากลักษณะสมบัติทั้ง 5 ข้อ แสดงว่าถ้ากำหนดให้  $f(t) \in L^2(\mathbb{R})$  สัญญาณ  $f(t)$  อันเดียวกันแต่อยู่ที่สเปซต่างกันจะมีจำนวนของ b.f. ที่ต่างกัน โดยระดับสเปซที่สูง ( $j$  มาก) จะมีจำนวนของ b.f. มาก ซึ่งจะหมายถึงสัญญาณมีความละเอียดสูงนั่นเอง

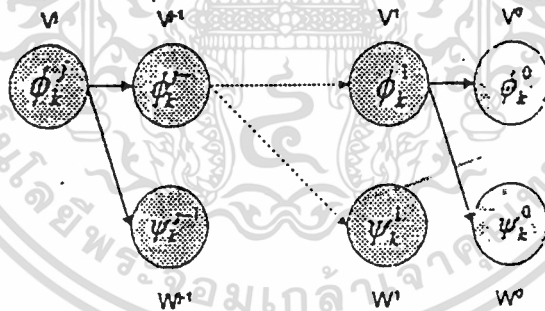
จากการที่มี b.f. ประกอบกันเป็นสัญญาณการประมาณที่ระดับความละเอียด  $j$  ภายในสเปซ  $V^j$  จะเรียก b.f. เหล่านี้ว่า "ฟังก์ชันสเกลลิง" (Scaling function) :  $\phi(t)$  สัญญาณเหล่านี้จะเกิดที่ตำแหน่งเวลาที่ต่างๆ กันของสเปซ และมีความถี่เท่ากันภายในสเปซเดียวกัน และความถี่จะเปลี่ยนแปลงขึ้นอยู่กับระดับความละเอียดของสเปซนั้นๆ ดังนั้นสามารถเขียนเป็นสมการฟังก์ชันสเกลลิงที่เกิดขึ้นที่ตำแหน่งใดๆ ภายในสเปซได้ดังแสดงในสมการที่ 2.11

$$\phi(t) = \phi(t-k) \quad ; k \in \mathbb{Z} \quad (2.11)$$

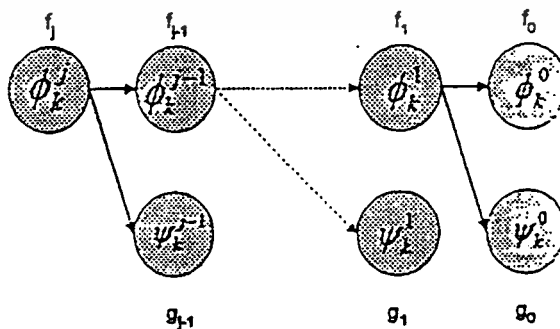
สมมติให้  $f(t) \in V^{j+1}$  สามารถแตกกระจายให้  $f(t)$  ให้มีความละเอียดน้อยลงได้จากสมการ  $V^j + W^j = V^{j+1}$  ซึ่งในขณะเดียวกัน  $V^j$  สามารถแตกต่อไปได้เรื่อยๆ จนกระทั่ง  $j=0$  ดังนั้นจะได้เป็นความสัมพันธ์ว่า

$$V^{j+1} = V^0 + W^0 + W^1 + \dots + W^j$$

ในทำนองเดียวกัน  $f_{j+1}$  ก็จะสามารถแตกกระจายเป็น  $f_j$  และ  $g_j$  ซึ่งก็จะสามารถเขียนเป็นแผนภูมิแสดงการแตกกระจายสเปซและสัญญาณได้ดังแสดงในรูปที่ 2.14 และ 2.15 ตามลำดับ



รูปที่ 2.14 แสดงลักษณะการแตกกระจายของสเปซเวกเตอร์



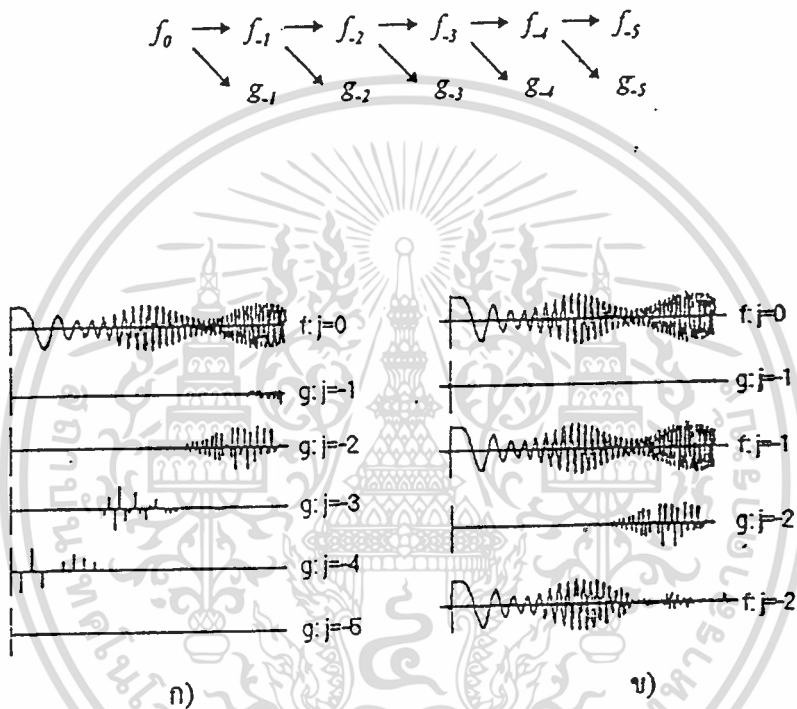
รูปที่ 2.15 แสดงลักษณะการแตกกระจายสัญญาณ  $f_j$  ไปยังระดับความละเอียดต่างๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$f$  และ  $g$  ที่ลดระดับความละเอียดลงมาจะมีความถี่ของ b.f. ลดลงครึ่งละสองเท่าเสมอตามลักษณะข้อที่ 5 ของ MRA

**ตัวอย่าง** ถ้าสมมติให้  $f_0$  เป็นระดับความละเอียดสูงสุดของสัญญาณใดๆ ในสเปซ  $V^0$  ถ้าต้องการทำการแตกกระจาย  $f_0$  ออกไป 5 ระดับ ก็จะได้ลักษณะของสัญญาณดังแสดงในรูปที่ 2.16



รูปที่ 2.16

ก) แสดงลักษณะของสัญญาณ  $f_0$  ซึ่งถูกแตกออกไปเป็น  $g_j(t)$  5 ระดับ

ข) แสดงลักษณะของการแตกกระจายสัญญาณที่ระดับ  $j = -1, -2$

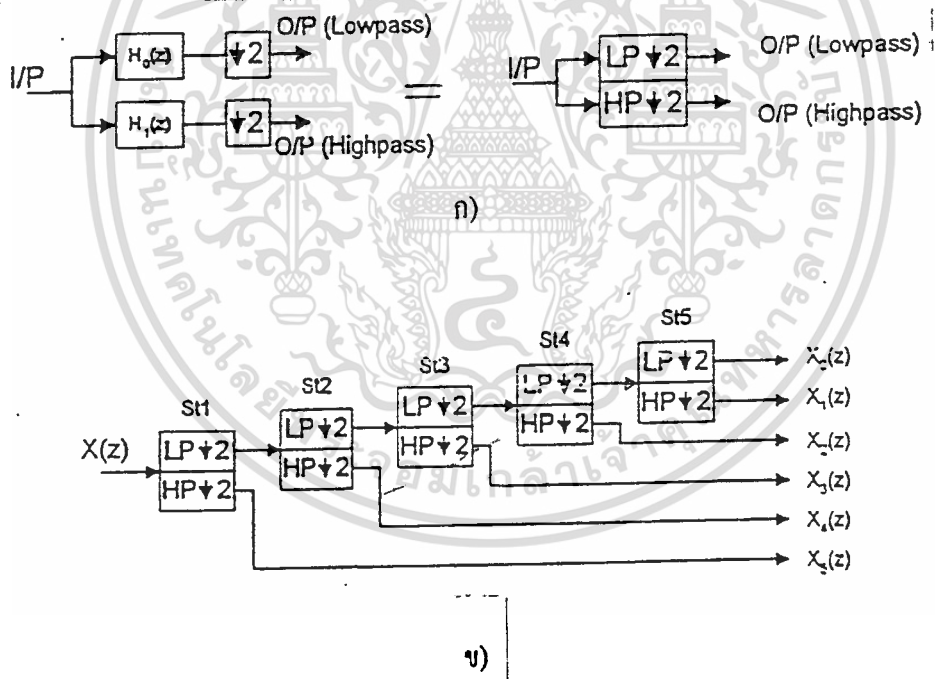
$W^j$  ที่ระดับความละเอียดใดๆ จะมีลักษณะเชิงตั้งฉาก (Orthogonal) ซึ่งกันและกันเสมอ เนื่องจากสัญญาณภายในสเปซของเวฟเลตที่ระดับ  $j$  ต่างกันจะไม่มีสัญญาณเกี่ยวข้องกันเลย ดังแสดงในรูปที่ 2.16 ดังนั้นจะสามารถเปลี่ยนการบวกกันแบบตรงๆ ของสเปซ (Direct sum) เป็นการบวกในลักษณะแบบการบวกเชิงตั้งฉาก (Orthogonal Sum) ซึ่งเขียนเป็นความสัมพันธ์ใหม่ได้ดังนี้

$$V^{j+1} = V^0 \oplus W^0 \oplus W^1 \oplus \dots \oplus W^j$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 การแปลงเวฟเล็ดโดยใช้หลักการของ Analysis Filter Banks (Octave Analysis Filter Banks)

Two channel filter banks เป็นการแยกสัญญาณอินพุตออกเป็น 2 ส่วน โดยแบ่งเป็นส่วนความถี่ต่ำและส่วนความถี่สูง ดังนั้น Two channel filter banks จึงประกอบด้วยวงจรที่เป็น lowpass filter :  $H_0(z)$  และ Complementary highpass filter :  $H_1(z)$  ซึ่งในส่วนของ highpass จะใช้ในการแยกความถี่ในแบนด์ที่ต้องการออกมา (Band-seperating filter) ดังแสดงในรูปที่ 2.17 และรูปที่ 2.18 ลักษณะของ Octave analysis filter banks จะเป็นแบบ Tree structure ซึ่งเป็นการนำเอา Two channel filter banks มาเรียงต่อกัน โดยใช้สัญญาณเอาท์พุทในส่วนความถี่ต่ำจากวงจร Lowpass filter มาทำการแยกแบนด์ความถี่ออกอีกครั้งหนึ่งดังแสดงในรูปที่ 2.18



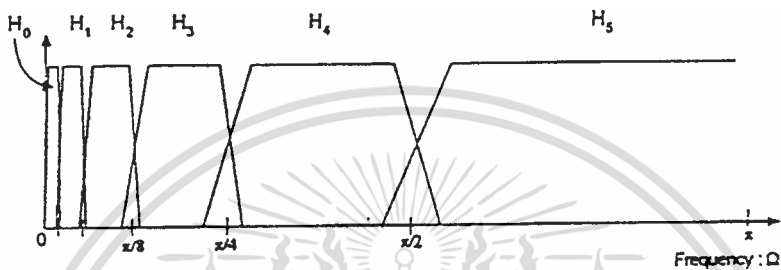
รูปที่ 2.17

ก) แสดงสัญลักษณ์ของ Two channel analysis filter banks หรือ Bands seperator

ข) แสดงลักษณะของ Tree structured octave analysis filter banks

ใน Two channel filter banks ชุดแรก จะทำการแยกสัญญาณความถี่สูงครั้งด้านบน ( $X_5(z)$ ) และสัญญาณความถี่ต่ำครั้งด้านล่างออกจากกัน จากนั้นจะนำสัญญาณความถี่ต่ำครั้งด้านล่างมาแยกสัญญาณความถี่สูงครั้งด้านล่างออกอีกครั้งหนึ่ง จากนั้นจะนำสัญญาณความถี่ต่ำครั้งด้านล่างมาแยกสัญญาณความถี่สูงครั้งด้านล่างออกอีกครั้งหนึ่ง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแยกแบนด์ความถี่ออกครึ่งหนึ่งออกอีกครั้ง โดยให้ครึ่งบนทางความถี่สูงเป็นสัญญาณ  $X_4(z)$  และจะนำส่วนความถี่ต่ำไปทำการแยกแบนด์ต่อไปเรื่อยๆ ตามระดับที่ต้องการ ในรูปที่ 2.18 จะแสดงให้เห็นถึงลักษณะของแบนด์ความถี่ที่ถูกแบ่งออกไปเรื่อยๆ สัมพันธ์กับอัตราการสุ่มของสัญญาณอินพุท  $X(z)$  อัตราสุ่มจะลดลงครึ่งสอง ซึ่งจะหมายถึงจำนวนข้อมูลสุ่มจะลดลงสองเท่าในแต่ละสเตทนั่นเอง Transfer function ของ Filter banks ในแต่ละระดับจะหาได้จากสมการ  $H_l(z)=X_l(z)/X(z)$  , $l=0...5$

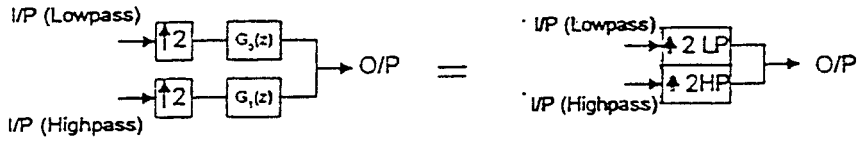


รูปที่ 2.18 แสดงลักษณะของการแยก Frequency bands ของ Octave filter bank

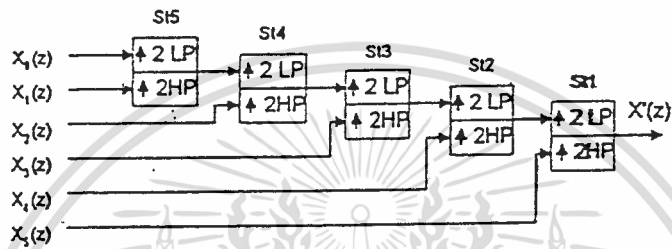
Cut off frequency มีค่าลดลงครึ่งละสองเท่า ดังนั้นจึงเรียกว่า Octave filter banks และจาก รูปที่ 2.18 จะสังเกตว่า Absolute transition bandwidth (เอทท์พูท Lowpass) จะลดลงในอัตราของสองเท่า โดยลดลงไปในทางความถี่ที่ต่ำลง และ Transfer function อื่นๆ จะมีลักษณะเป็น Bandpass ดังนั้น Filter banks จึงสามารถในการแยกแบนด์ความถี่ต่างๆ ออกมา(Band-seperating filter)

## 2.8 Octave Synthesis Filter Banks

รูปที่ 2.19 จะแสดงลักษณะของ Octave synthesis filter banks ซึ่งจะเป็นโครงสร้างที่กลับกันของ Octave analysis filter banks ในรูปที่ 2.17 ถ้าทำการเชื่อมต่อ Filter banks ทั้งสองแบบเข้าด้วยกัน จะได้สัญญาณเอทท์พูท  $X'(z)$  ออกมาซึ่งจะเป็นค่าประมาณของ  $X(z)$  ถ้ารูปแบบของ Transfer function  $H_0(z), H_1(z), G_0(z)$  และ  $G_1(z)$  ทั้ง 4 ชุดเป็นแบบ Perfectly reconstructing two channel filter banks แล้ว Octave filterbanks จะเป็น Perfectly reconstructing นั่นคือ  $X'(z)$  เท่ากับ  $X(z)$



ก)



ข)

รูปที่ 2.19

ก) แสดงสัญลักษณ์ของ two-channel synthesis filter banks หรือ Bands separator

ข) แสดง Octave synthesis filter banks

## บทที่ 3

### การพิสูจนท์ฤษฎี Wavelets Transform เพื่อใช้ในการสื่อสาร

จากการที่ได้ศึกษาทฤษฎีโดยคร่าวๆจากบทที่ 2 แล้วในบทนี้ก็จะเป็นการนำเอาทฤษฎีในบทที่ 2 มาใช้และก็มี การประยุกต์นิดหน่อยเพื่อให้เกิดความเหมาะสมกับการใช้งาน ดังนั้นในบทนี้ส่วนมากเป็นการทดลองในแบบต่างๆที่จะเหมือนกับการทำงานเมื่อเรานำไปใช้งานจริง เช่นทำการแปลงข้อมูลที่เราอ่านได้จากเอาต์พุทของคอมพิวเตอร์ให้เป็นค่า Coefficient ต่างๆและนำไปผ่านกระบวนการเพื่อให้ได้สัญญาณที่เหมาะสม การทดสอบการทำงานเมื่อสัญญาณมีการ lagging แล้วเราสามารถที่จะหาผลลัพธ์จากสัญญาณนั้นได้หรือไม่ เมื่อสัญญาณผ่าน lowpass filter แล้วลักษณะของสัญญาณจะเป็นอย่างไรและยังสามารถที่จะกู้ข้อมูลนั้นได้หรือไม่

ดังนั้นเนื้อหาในบทนี้จะมีมากมายและมีหลายประเด็นทางผู้ทำก็จะแบ่งแยกให้เป็นหัวข้อย่อยต่างๆเพื่อให้สะดวกแก่การพิจารณาและศึกษา

#### 3.1 ทดสอบทฤษฎี Wavelets Transform ที่นำมาใช้

เราจะทำการทดลองว่าทฤษฎีที่เราใช้นั้นสามารถแปลงข้อมูลแล้วทำการส่งไปได้หรือไม่ และสามารถเปลี่ยนกลับมาเป็นข้อมูลเหมือนเดิมได้หรือไม่ ในตอนแรกนี้จะใช้ Haar wavelet เป็น wavelet ที่ใช้เป็นฟังก์ชันของ high pass และ low pass ซึ่งในที่นี้ค่าของ low pass คือ  $[1 \ 1]$  ส่วนค่าของ high pass หาได้โดยการกลับค่าฟังก์ชันของ lowpass แล้วคูณ  $-1$  หน้าสัมประสิทธิ์ตัวที่ 2 เพื่อที่จะทำให้ค่าสัมประสิทธิ์ของ low pass และ high pass เมื่อทำเป็นกราฟแล้วรูปกราฟทั้ง 2 ไม่สามารถที่จะซ้อนทับซึ่งกันและกันได้ไม่ว่าจะใช้วิธีใดก็ตาม

จากนั้นได้ทำการทดลองเขียนโปรแกรมโดยใช้ Matlab ซึ่งใช้ Haar wavelet

#### โปรแกรมที่เขียนเพื่อทดสอบทฤษฎีพื้นฐาน

ส่วนของการแปลงข้อมูลไป

$$a = [9 \ 7 \ 3 \ 5]$$

$$g = [1 \ -1]/\text{sqrt}(2);$$

$$g = [0.7071 \ -0.7071]$$

$$h = [1 \ 1]/\text{sqrt}(2);$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
h = [ 0.7071  0.7071 ]
```

```
h1 = conv(a,h);
```

```
h1 = [ 6.3640  11.3137  8.4853  5.6569  2.1213 ]
```

```
g1=conv(a,g);
```

```
g1 = [ 6.3640  -1.4142  -1.4142  -1.4142  -2.1213 ]
```

```
h1d = h1(2:2:5);
```

```
h1d = [ 11.3137  5.6569 ]
```

```
g1d = g1(2:2:5);
```

```
g1d = [ -1.4142  -1.4142 ]
```

```
b = h1d;
```

```
h2 = conv(b,h);
```

```
h2 = [ 8.0000  12.0000  4.0000 ]
```

```
g2 = conv(b,g);
```

```
g2 = [ 8.0000  -4.0000  -4.0000 ]
```

```
h2d = h2(2:2:3);
```

```
h2d = [ 12.0000 ]
```

```
g2d = g2(2:2:3);
```

```
g2d = [ -4.0000 ]
```

ส่วนของการแปลงข้อมูลกลับ

```
h2du(1:2:2) = h2d;
```

```
h2du = [ 12.0000 ]
```

```
g2du(1:2:2) = g2d;
```

```
g2du = [ -4.0000 ]
```

```
ih2 = conv(h2du,flipr(h));
```

```
ih2 = [ 8.4853  8.4853 ]
```

```
ig2 = conv(g2du,flipr(g));
```

```
ig2 = [ 2.8284  -2.8284 ]
```

```
ib = ih2+ig2;
```

```
ib = [ 11.3137  5.6569 ]
```

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$h1du(1:2:4) = ib;$$

$$h1du = [ 11.3137 \quad 0 \quad 5.6569 ]$$

$$g1du(1:2:4) = g1d;$$

$$g1du = [ -1.4142 \quad 0 \quad 1.4142 ]$$

$$ih1 = \text{conv}(h1du, \text{fliplr}(h));$$

$$ih1 = [ 8.0000 \quad 8.0000 \quad 4.0000 \quad 4.0000 ]$$

$$ig1 = \text{conv}(g1du, \text{fliplr}(g));$$

$$ig1 = [ 1.0000 \quad -1.0000 \quad -1.0000 \quad 1.0000 ]$$

$$ia = ih1 + ig1$$

$$ia = [ 9.0000 \quad 7.0000 \quad 3.0000 \quad 5.0000 ]$$

จากโปรแกรมจะขออธิบายเป็นขั้นๆ ดังนี้คือตัวแปร  $a$  คือ ข้อมูลที่นำมาใช้ทดสอบ  $g$  และ  $h$  คือ function wavelet transform highpass และ lowpass ตามลำดับตัวแปร  $h1$  และ  $g1$  คือ ค่าที่ได้จากการนำเอาข้อมูล  $a$  มาทำการ Convolution กับฟังก์ชัน wavelet transform  $h$  และ  $g$  ส่วนค่า  $h1d$  และ  $g1d$  คือ ค่าของข้อมูล  $h1$  และ  $g1$  ที่ผ่านการ down sampling นำค่า Coefficient ของความถี่สูง ( $g1d$ ) เก็บไว้ก่อนเพื่อที่จะนำมาใช้ในการหาข้อมูลกลับภายหลัง นำค่า Coefficient ของความถี่ต่ำ ( $h1d$ ) มาทำการประมวลผลต่อไป กำหนดให้  $b = h1d$  จากนั้นนำค่า  $b$  ไปทำการ Convolution กับ  $h$  และ  $g$  เหมือนครั้งแรกแต่จะได้ค่าเป็น  $h2$  และ  $g2$  แทนและเช่นเดียวกัน  $h2d$  และ  $g2d$  คือ ค่าของข้อมูลที่ได้จากการนำเอา  $h2$  และ  $g2$  มาทำการ down sampling

ซึ่งค่าสัมประสิทธิ์เริ่มต้นที่ใช้คือ  $a = [9 \ 7 \ 3 \ 5]$  ซึ่งเมื่อผ่านการกระทำตามโปรแกรมแล้ว จะได้ค่าของสัมประสิทธิ์ตัวสุดท้าย 2 ตัวคือ  $h2d$  และ  $g2d$  ( $h2d = 12, g2d = -4$ ) เมื่อสามารถที่จะทำการรวมค่าของสัมประสิทธิ์เข้าด้วยกันได้แล้ว จากนั้นเราก็ต้องทำการพิสูจน์ว่าเราสามารถที่จะทำการแยกข้อมูลที่ผ่านการรวมแล้วให้เป็นข้อมูลตัวเดิมได้

ตัวแปร  $h2du$  และ  $g2du$  คือ ค่าที่ได้จากการนำเอา  $h2d$  และ  $g2d$  มาทำการ up sampling  $ih2$  และ  $ig2$  คือ การนำเอา  $h2du, g2du$  มา Convolution กับ  $h, g$  ที่ผ่านการ flip แล้ว เพื่อที่เราจะได้ค่า inverse ของ  $h2$  และ  $g2$  และค่า  $ib$  จะมีค่าเท่ากับ  $b$  ของโปรแกรมในครั้งแรก  $h1du$  และ  $g1du$  คือ ค่าที่ได้จากการนำเอาค่า  $ib$  และ  $g1d$  มาทำการ up sampling ตัวแปร  $ih1$  และ  $ig1$  คือ ค่าที่ได้จากการนำเอา  $h1du$  และ  $g1du$  มาทำการ Convolution กับ  $h$  กับ  $g$  ที่ผ่านการ flip แล้วตามลำดับ  $ia$  คือ ค่าที่ได้จากการรวม  $ih1$  และ  $ig1$  เข้าด้วยกัน ซึ่งจะมีค่าเท่ากับ  $a$

เราจะได้อำนาจ  $ia=[9 \ 7 \ 3 \ 5]$  ซึ่งจากการที่ได้ทำการทดลองสามารถที่จะหาค่าของข้อมูลได้ เหมือนกับข้อมูลในครั้งแรกที่ใส่เข้าไป เป็นอันว่าเราสามารถที่จะใช้ทฤษฎี Wavelets Transform ทำการแปลงข้อมูลไปแล้วก็แปลงข้อมูลนั้นกลับมาได้

### 3.2 เปลี่ยน function wavelet ให้เหมาะสม

จากหัวข้อแรกแสดงให้เราเห็นแล้วว่าการแปลงข้อมูลใดๆก็แล้วแต่เราจำเป็นจะต้องมี function wavelets อยู่ 2 ชุด และ function ทั้งสองนั้นก็จะเป็น Inverse ซึ่งกันและกันไม่สามารถแทนกันได้และเป็นอิสระต่อกันโดยสมบูรณ์ แต่ข้อจำกัดของ function นี้ก็คือ เราไม่สามารถที่จะใช้ function wavelets ได้ function หนึ่ง เพื่อทำการแปลงข้อมูลได้ทุกๆรูปแบบถึงแม้จะทำได้ก็จะเป็นไม่ดีดังนั้นเราจึงควรหา function wavelets ที่เหมาะสมมาใช้โดยได้ทำการเปลี่ยนมาใช้ wavelet ตัวใหม่ที่มีชื่อ Daubechies 4 แทน Haar wavelet ซึ่งสามารถที่จะให้ความถูกต้องแม่นยำได้มากกว่าเมื่อนำไปใช้กับข้อมูลจำนวนมากๆ

จากนั้นก็ทำการทดลองเขียนโปรแกรมโดยใช้ wavelet ตัวใหม่คือ Daubechies 4 ซึ่งจะขอยกตัวอย่างโปรแกรมที่เขียนไว้ดังนี้

โปรแกรมที่ใช้เพื่อทดสอบว่า Daubechies 4 เหมาะสมกับการใช้งาน

$$h=(\text{sqrt}(2)/8)*[(1+\text{sqrt}(3)) (3+\text{sqrt}(3)) (3-\text{sqrt}(3)) (1-\text{sqrt}(3))]$$

$$h = [ 0.4830 \ 0.8365 \ 0.2241 \ -0.1294 ]$$

$$k=\text{fliplr}(h);$$

$$k = [ -0.1294 \ 0.2241 \ 0.8365 \ 0.4830 ]$$

$$g=[ k(1) \ k(2)*(-1) \ k(3) \ k(4)*(-1) ];$$

$$g = [ -0.1294 \ -0.2241 \ 0.8365 \ -0.4830 ]$$

$$h4d=[1]$$

$$g4d=[-1]$$

$$h4du(1:2:2)=h4d;$$

$$h4du = [ - 1 ]$$

$$g4du(1:2:2)=g4d;$$

$$g4du = [ - -1 ]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ih4=conv(h4du,flipr(h));
```

```
ih4 = [ -0.1294  0.2241  0.8365  0.4830 ]
```

```
ig4=conv(g4du,flipr(g));
```

```
ig4 = [ 0.4830 -0.8365  0.2241  0.1294 ]
```

```
id= ih4 + ig4;
```

```
id = [ 0.3536 -0.6124  1.0606  0.6124
```

```
g3d=[1 -1]
```

```
h3du(1:2:8)=id;
```

```
h3du = [ - 0.3536    0 -0.6124    0 1.0606    0 0.6124 ]
```

```
g3du(1:2:4)=g3d;
```

```
g3du = [ - 1  0 -1 ]
```

```
g3du=[0 0 g3du 0 0];
```

```
g3du = [ - 0  0  1  0 -1  0  0 ]
```

```
ih3=conv(h3du,flipr(h));
```

```
ih3 = [ -0.0458  0.0792  0.3750  0.0335 -0.6495 -0.0581  0.8079
        0.6495  0.5123  0.2958 ]
```

```
ig3=conv(g3du,flipr(g));
```

```
ig3 = [ 0  0 -0.4830  0.8365  0.2589 -0.9659  0.2241 -0.1294  0  0 ]
```

```
ic = ih3 + ig3;
```

```
ic = [ -0.0458  0.0792 -0.1080  0.8700 -0.3906 -1.0240  1.0320
        0.7789  0.5123  0.2958 ]
```

```
g2d=[1 -1 1 -1]
```

```
h2du(1:2:20)=ic;
```

```
h2du = [ - -0.0458  0  0.0792  0 -0.1080  0  0.8700  0 -0.3906  0 -1.0240  0
          1.0320  0  0.7789  0  0.5123  0  0.2958 ]
```

```
g2du(1:2:8)=g2d;
```

```
g2du = [ - 1  0 -1  0  1  0 -1 ]
```

```
g2du=[0 0 0 0 0 g2du 0 0 0 0 0];
```

```
g2du = [ - 0  0  0  0  0  0  0  1  0 -1  0  1  0 -1  0  0  0  0  0
```

```
0 ]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ih2=conv(h2du,flipr(h));
```

```
ih2 = [ 0.0059 -0.0103 -0.0485 -0.0043 0.0803 0.0141 -0.2029 0.1428 0.7783
        0.3327 -0.1942 -0.4181 -0.9901 -0.2633 0.7625 0.6730 0.5853 0.4910
        0.3902 0.3137 0.2474 0.1429 ]
```

```
ig2=conv(g2du,flipr(g));
```

```
ig2 = [ 0 0 0 0 0 0 -0.4830 0.8365
        0.2589 -0.9659 0.2589 0.9659 0.2589 -0.9659 0.2241 0.1294
        0 0 0 0 0 0 ]
```

```
ib =ih2+ig2;
```

```
ib = [ 0.0059 -0.0103 -0.0485 -0.0043 0.0803 0.0141 -0.6859 0.9793
        1.0372 -0.6332 -0.4531 0.5478 -0.7312 -1.2292 0.9866 0.8024
        0.5853 0.4910 0.3902 0.3137 0.2474 0.1429 ]
```

```
g1d=[1 -1 1 -1 1 -1 1 -1]
```

```
h1du(1:2:44)=ib;
```

```
h1du = [ - 0.0059 0 -0.0103 0 -0.0485 0 -0.0043 0 0.0803
          0 0.0141 0 -0.6859 0 0.9793 0 1.0372 0 -0.6332
          0 -0.4531 0 0.5478 0 -0.7312 0 -1.2292 0 0.9866
          0 0.8024 0 0.5853 0 0.4910 0 0.3902 0 0.3137
          0 0.2474 0 0.1429 ]
```

```
g1du(1:2:16)=g1d;
```

```
g1du = [ - 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 ]
```

```
g1du=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 g1du 0 0 0 0 0 0 0 0 0 0 0 0];
```

```
g1du = [ - 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
          -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 ]
```

```
ih1=conv(h1du,flipr(h));
```

```
ih1 = [ -0.0008 0.0013 0.0063 0.0006 -0.0023 -0.0158 -0.0400
        -0.0244 -0.0140 0.0159 0.0653 0.0419 0.1005 -0.1469
        -0.7005 -0.1118 0.6850 0.7055 0.9496 0.3591 -0.4710
        -0.4074 -0.4499 -0.0961 0.5528 0.1007 -0.4526 -0.6287
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-1.1559 -0.3726 0.7215 0.6564 0.5955 0.5187 0.4260
0.3927 0.3602 0.3246 0.2858 0.2588 0.2304 0.2070
0.1885 0.1515 0.1195 0.0690 ]
```

```
ig1=conv(g1du,flipr(g));
```

```
ig1 = [ 0      0      0      0      0      0      0
        0      0      0      0      0      0      0
       -0.4830 0.8365 0.2589 -0.9659 -0.2589 0.9659 0.2589
       -0.9659 -0.2589 0.9659 0.2589 -0.9659 -0.2589 0.9659
        0.2589 -0.9659 0.2241 0.1294 0      0      0
        0      0      0      0      0      0      0
        0      0      0      0 ]
```

```
ia=ih1 + ig1;
```

```
ia = [-0.0008 0.0013 0.0063 0.0006 -0.0023 -0.0158 -0.0400
      -0.0244 -0.0140 0.0159 0.0653 0.0419 0.1005 -0.1469
      -1.1835 0.7247 0.9439 -0.2604 0.6907 1.3250 -0.2121
      -1.3733 -0.7088 0.8698 0.8117 -0.8652 -0.7115 0.3372
      -0.8970 -1.3385 0.9456 0.7858 0.5955 0.5187 0.4260
      0.3927 0.3602 0.3246 0.2858 0.2588 0.2304 0.2070
      0.1885 0.1515 0.1195 0.0690 ]
```

จากโปรแกรมข้างบนเป็นการเขียนโปรแกรมโดยใช้ Wavelet ตัวใหม่ที่ชื่อว่า Daubechies 4 ซึ่งค่า h และ g คือ ค่าของ Daubechies 4 ที่ความถี่ต่ำและความถี่สูงตามลำดับ ซึ่ง h4d และ g4d คือ ค่าของ Coefficient ที่ต้องการส่งไป จากนั้นนำค่า h4d และ g4d มาทำการ up sampling ซึ่งจะทำให้ได้ Coefficient ตัวใหม่ คือ h4du และ g4du ต่อจากนั้นนำ h4du และ g4du มาทำการ Convolution กับ h และ g ที่ผ่านการ flip แล้ว จะได้ Coefficient ตัวใหม่เป็น ih4 และ ig4 ตามลำดับ ถัดจากนั้นนำเอา ih4 และ ig4 มารวมกันจะได้ Coefficient ตัวใหม่เป็น id หลังจากนั้นนำค่า Coefficient ตัวใหม่ที่ต้องการส่ง คือ g3d มาทำการประมวลผลต่อไป นำค่าของ id และ g3d มาทำการ up sampling จะได้ค่า Coefficient ตัวใหม่เป็น h3du และ g3du และเนื่องจาก g3du มีจำนวนขนาดของข้อมูลน้อยกว่า h3du จึงต้องมีการเติม 0 เพิ่มข้างหน้าและข้างหลังของ g3du เพื่อจะได้มีขนาดเท่ากับ h3du โดยที่การเติม 0 จะต้องทำให้ข้อมูลเดิมนั้นอยู่ตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับต่อไป นำ  $h_{3du}$  และ  $g_{3du}$  มาทำการ Convolution กับ  $h$  และ  $g$  ที่ผ่านการ flip เรียบร้อยแล้ว ก็จะได้ค่า Coefficient ตัวใหม่คือ  $ih_3$  และ  $ig_3$  ตามลำดับ นำค่าของ  $ih_3$  และ  $ig_3$  มาทำการรวมกันก็จะได้ค่า Coefficient ตัวใหม่เป็น  $ic$

ต่อจากนั้นนำค่า  $g_{2d}$  ซึ่งเป็นค่า Coefficient ที่ต้องการจะส่งมาทำการประมวลผลต่อไป เอาค่า  $ic$  และ  $g_{2d}$  มาทำการ up sampling ก็จะได้ค่าเป็น  $h_{2du}$  และ  $g_{2du}$  ตามลำดับ เนื่องจากว่าขนาดของข้อมูลของ  $g_{2du}$  เล็กกว่า ขนาดของ  $h_{2du}$  จึงต้องทำการเพิ่มขนาดให้มีขนาดเท่ากันก่อน โดยการเอาข้อมูลเดิมไว้ตรงกลางแล้วทำการเติม 0 ทั้งข้างหน้าและข้างหลังให้มีขนาดเท่ากับ  $h_{2du}$  จากนั้นนำค่า  $h_{2du}$  และ  $g_{2du}$  มาทำการ Convolution กับ  $h$  และ  $g$  ที่ผ่านการ flip แล้วก็จะได้ค่าของ  $ih_2$  และ  $ig_2$  ตามลำดับ ต่อจากนั้นทำการรวมค่า  $ih_2$  และ  $ig_2$  ก็จะได้ค่า Coefficient ตัวใหม่เป็น  $ib$

หลังจากนั้นนำค่า  $g_{1d}$  ซึ่งเป็นค่าของ Coefficient ที่ต้องการจะส่งมาทำการประมวลผลต่อไป นำค่า  $ib$  และ  $g_{1d}$  มาทำการ up sampling ก็จะได้ค่า Coefficient เป็น  $h_{1du}$  และ  $g_{1du}$  ตามลำดับและเนื่องจากว่าขนาดของ  $g_{1du}$  มีขนาดเล็กกว่าขนาดของ  $h_{1du}$  จึงต้องทำการเติม 0 ที่ข้างหน้าและข้างหลังของ  $g_{1du}$  โดยการทำให้ข้อมูลเดิมอยู่ตรงกลางเช่นเดียวกับที่ผ่านมามีขนาดเท่ากับ  $h_{1du}$  ถัดจากนั้นนำค่า  $h_{1du}$  และ  $g_{1du}$  ไปทำการ Convolution กับ  $h$  และ  $g$  ที่ผ่านการ flip แล้วจะได้ค่า Coefficient ตัวใหม่เป็น  $ih_1$  และ  $ig_1$  ซึ่งผลของการรวมกันของ  $ih_1$  และ  $ig_1$  จะทำให้ได้ค่า Coefficient ตัวใหม่คือ  $ia$  ซึ่งก็คือ ข้อมูลที่ทำการกระจายเสร็จเรียบร้อยแล้วที่จะทำการส่งแล้ว

จากโปรแกรมข้างบนเป็นการเขียนโปรแกรมโดยใช้ Daubechies 4 เพื่อใช้ในการส่งข้อมูล ซึ่งจากโปรแกรมข้างบน ค่า Coefficient ที่เราต้องการส่งคือ  $h_{4d}=[1], g_{4d}=[-1], g_{3d}=[1 \ -1], g_{2d}=[1 \ -1 \ 1 \ -1], g_{1d}=[1 \ -1 \ 1 \ -1 \ 1 \ -1]$

จากนั้นก็ทำตามวิธีการที่ได้ศึกษามากก็สามารถที่จะได้ข้อมูลสุดท้ายที่เราต้องการส่งคือ  $ia$  ซึ่งค่า  $ia$  ที่ได้มีค่าดังนี้

$ia =$

Columns 1 through 7

-0.0008 0.0013 0.0063 0.0006 -0.0023 -0.0158 -0.0400

Columns 8 through 14

-0.0244 -0.0140 0.0159 0.0653 0.0419 0.1005 -0.1469

Columns 15 through 21

-1.1834 0.7248 0.9439 -0.2604 0.6907 1.3249 -0.2123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Columns 22 through 28

-1.3734 -0.7088 0.8698 0.8116 -0.8653 -0.7115 0.3372

Columns 29 through 35

-0.8970 -1.3384 0.9457 0.7859 0.5956 0.5188 0.4261

Columns 36 through 42

0.3927 0.3602 0.3246 0.2858 0.2588 0.2304 0.2070

Columns 43 through 46

0.1885 0.1515 0.1195 0.0690

ดังนั้นเราจะได้ข้อมูลที่สามารถส่งได้แล้วคือ ia เมื่อได้ข้อมูลดังกล่าวออกมาแล้วจากนั้นก็  
ต้องนำข้อมูลที่ได้ไปทำการแปลงกลับเพื่อพิสูจน์ให้ได้ก่อนว่าเราสามารถนำข้อมูลเดิมกลับออกมาได้  
ซึ่งจะแสดงให้เห็นต่อไป

โปรแกรมในส่วนรับข้อมูล

x=ia;

```
x = [-0.0008  0.0013  0.0063  0.0006 -0.0023 -0.0158 -0.0400
      -0.0244 -0.0140  0.0159  0.0653  0.0419  0.1005 -0.1469
      -1.1835  0.7247  0.9439 -0.2604  0.6907  1.3250 -0.2121
      -1.3733 -0.7088  0.8698  0.8117 -0.8652 -0.7115  0.3372
      -0.8970 -1.3385  0.9456  0.7858  0.5955  0.5187  0.4260
      0.3927  0.3602  0.3246  0.2858  0.2588  0.2304  0.2070
      0.1885  0.1515  0.1195  0.0690 ]
```

rg1=conv(x,g);

```
rg1 = [ 0.0001  0.0000 -0.0018  0.0000  0.0048  0.0000  0.0065
        0.0000 -0.0186  0.0000 -0.0119  0.0000  0.0246  0.0000
        0.2499  0.0000 -1.2036  1.0000  0.4085 -1.0000  0.4341
        1.0000 -0.4179 -1.0000 -0.2296  1.0000  0.5449 -1.0000
        -0.1368  1.0000 -0.7356 -1.0000  1.1843  0.0000 -0.0528
        0.0000 -0.0288  0.0000  0.0019  0.0000 -0.0055  0.0000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-0.0030 0.0000 0.0083 0.0000 0.0113 0.0000 -0.0333 ]
```

```
rh1=conv(x,h);
```

```
rh1 = [ -0.0004 0.0000 0.0040 0.0059 0.0006 -0.0103 -0.0332
```

```
-0.0485 -0.0341 -0.0043 0.0449 0.0803 0.0962 0.0141
```

```
-0.6774 -0.6859 0.8159 0.9793 0.2335 1.0372 1.1944
```

```
-0.6332 -1.7101 -0.4531 1.1385 0.5477 -0.9980 -0.7312
```

```
-0.1986 -1.2292 -0.9076 0.9866 1.3300 0.8024 0.6715
```

```
0.5853 0.5309 0.4910 0.4395 0.3902 0.3498 0.3137
```

```
0.2823 0.2474 0.1999 0.1429 0.0649 0.0000 -0.0089 ]
```

```
rg1d=rg1(2:2:49); % high pass coefficient(1)
```

```
rg1d = [ 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
0.0000 1.0000 -1.0000 1.0000 -1.0000 1.0000 -1.0000
```

```
1.0000 -1.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
0.0000 0.0000 0.0000 ]
```

```
rh1d=rh1(2:2:49);
```

```
rh1d = [ 0.0000 0.0059 -0.0103 -0.0485 -0.0043 0.0803 0.0141
```

```
-0.6859 0.9793 1.0372 -0.6332 -0.4531 0.5477 -0.7312
```

```
-1.2292 0.9866 0.8024 0.5853 0.4910 0.3902 0.3137
```

```
0.2474 0.1429 0.0000 ]
```

```
rg1d=rg1d(9:16)
```

```
rg1d = [ 1.0000 -1.0000 1.0000 -1.0000 1.0000 -1.0000 1.0000
```

```
-1.0000 ]
```

```
rc=rh1d;
```

```
rc = [ 0.0000 0.0059 -0.0103 -0.0485 -0.0043 0.0803 0.0141
```

```
-0.6859 0.9793 1.0372 -0.6332 -0.4531 0.5477 -0.7312
```

```
-1.2292 0.9866 0.8024 0.5853 0.4910 0.3902 0.3137
```

```
0.2474 0.1429 0.0000 ]
```

```
rc=rc(2:23);
```

```
rc = [ 0.0059 -0.0103 -0.0485 -0.0043 0.0803 0.0141 -0.6859
```

```
0.9793 1.0372 -0.6332 -0.4531 0.5477 -0.7312 -1.2292
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0.9866 0.8024 0.5853 0.4910 0.3902 0.3137 0.2474
0.1429 ]

```

```

pause;

```

```

rg2=conv(rc,g);

```

```

rg2 = [ -0.0008 0.0000 0.0135 0.0000 -0.0451 0.0000 0.1548
0.0000 -0.9342 1.0000 0.5952 -1.0000 -0.1013 1.0000
-0.7284 -1.0000 1.1634 0.0000 -0.0585 0.0000 -0.0130
0.0000 0.0234 0.0000 -0.0690 ]

```

```

rh2=conv(rc,h);

```

```

rh2 = [ 0.0029 0.0000 -0.0307 -0.0458 0.0256 0.0792 -0.3010
-0.1080 1.1646 0.8700 -0.6428 -0.3906 0.0854 -1.0240
-0.7864 1.0320 1.3341 0.7789 0.6265 0.5123 0.4058
0.2958 0.1344 0.0000 -0.0185 ]

```

```

rg2d=rg2(2:2:25); %high pass coefficient(2)

```

```

rg2d = [ 0.0000 0.0000 0.0000 0.0000 1.0000 -1.0000 1.0000
-1.0000 0.0000 0.0000 0.0000 0.0000 ]

```

```

rh2d=rh2(2:2:25);

```

```

rh2d = [ 0.0000 -0.0458 0.0792 -0.1080 0.8700 -0.3906 -1.0240
1.0320 0.7789 0.5123 0.2958 0.0000 ]

```

```

rg2d=rg2d(5:8)

```

```

rg2d = [ 1.0000 -1.0000 1.0000 -1.0000 ]

```

```

rb=rh2d;

```

```

rb = [ 0.0000 -0.0458 0.0792 -0.1080 0.8700 -0.3906 -1.0240
1.0320 0.7789 0.5123 0.2958 0.0000 ]

```

```

rb=rh2d(2:11);

```

```

rb = [ -0.0458 0.0792 -0.1080 0.8700 -0.3906 -1.0240 1.0320
0.7789 0.5123 0.2958 ]

```

```

pause;

```

```

rg3=conv(rb,g);

```

```

rg3 = [ 0.0059 0.0000 -0.0421 0.0000 -0.2730 1.0000 -0.6510

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

-1.0000  1.1170  0.0000 -0.0140  0.0000 -0.1429 ]
rh3=conv(rb,h);
rh3 = [ -0.0221  0.0000  0.0039  0.3536  0.5047 -0.6124 -0.5582
        1.0606  1.2627  0.6124  0.2614  0.0000 -0.0383 ]
rg3d=rg3(2:2:13); % high pass coefficient(3)
rg3d = [ 0.0000  0.0000  1.0000 -1.0000  0.0000  0.0000 ]
rh3d=rh3(2:2:13);
rh3d = [ 0.0000  0.3536 -0.6124  1.0606  0.6124  0.0000 ]
rg3d=rg3d(3:4)
rg3d = [ 1.0000 -1.0000 ]
ra=rh3d;
ra = [ 0.0000  0.3536 -0.6124  1.0606  0.6124  0.0000 ]
ra=rh3d(2:5);
ra = [ 0.3536 -0.6124  1.0606  0.6124 ]
pause;
rg4=conv(ra,g);
rg4 = [ -0.0458  0.0000  0.2958 -0.9999  1.0457  0.0000 -0.2958 ]
rh4=conv(ra,h);
rh4 = [ 0.1708  0.0000  0.0792  0.9999  0.8292  0.0000 -0.0792 ]
format bank
rg4d=rg4(2:2:7); % high pass coefficient(5)
rg4d = [ 0.0000 -0.9999  0.0000 ]
rh4d=rh4(2:2:7);
rh4d = [ 0.0000  0.9999  0.0000 ]
rg4d=rg4d(2)
rg4d = [ -0.9999 ]
rh4d=rh4d(2)
rh4d = [ 0.9999 ]
format

```

จากโปรแกรมข้างบนเป็นส่วนของการรับข้อมูล ซึ่งข้อมูลของชุดส่งคือ ia ซึ่งต่อไปนี้จะขออธิบายการทำงานของโปรแกรมในส่วนรับข้อมูล เริ่มต้นด้วยการกำหนดให้  $x = ia$  จากนั้นนำ  $x$  มาทำการ Convolution กับ  $h$  และ  $g$  ก็จะได้ค่า  $rh1$  และ  $rg1$  ตามลำดับ จากนั้นนำค่า  $rh1$  และ  $rg1$  มาทำการ down sampling ก็จะได้  $rh1d$  และ  $rg1d$  หลังจากนั้นทำการเลือกช่วงของข้อมูลที่ต้องการ ซึ่งค่าของ  $rg1d$  ที่ได้จะมีค่าเท่ากับ  $g1d$  ที่ทำการส่ง

กำหนดให้  $rc$  เท่ากับ  $rh1d$  จากนั้นก็ทำการเลือกช่วงของข้อมูลที่ต้องการ ต่อจากนั้นนำค่า  $rc$  ไปทำการ Convolution กับ  $h$  และ  $g$  ซึ่งจะได้ค่าเป็น  $rh2$  และ  $rg2$  ตามลำดับ หลังจากนั้นนำค่า  $rh2$  และ  $rg2$  มาทำการ down sampling ก็จะได้ค่า Coefficient ใหม่ออกมาเป็น  $rh2d$  และ  $rg2d$  จากนั้นเลือกช่วง  $rg2d$  ที่เป็นข้อมูลที่ต้องการออกมาซึ่งจะเท่ากับค่าของ  $g2d$  ที่เป็นข้อมูลในการส่ง

กำหนดให้  $rb$  เท่ากับ  $rh2d$  จากนั้นทำการเลือกช่วงข้อมูลของ  $rb$  ที่ต้องการ ต่อไปก็นำ  $rb$  ไปทำการ Convolution กับ  $h$  และ  $g$  ก็จะได้ค่าออกมาเป็น  $rh3$  และ  $rg3$  ต่อจากนั้นนำค่า  $rh3$  และ  $rg3$  ที่ได้มาทำการ down sampling ก็จะได้ค่า Coefficient ออกมาเป็น  $rh3d$  และ  $rg3d$  ตามลำดับ ต่อจากนั้นทำการเลือกช่วงของข้อมูลที่ต้องการออกมา กำหนดให้  $ra$  เท่ากับ  $rh3d$  แล้วก็ทำการเลือกช่วงข้อมูลของ  $ra$  ที่ต้องการ หลังจากนั้นนำ  $ra$  ไปทำการ Convolution กับ  $h$  และ  $g$  ก็จะได้ค่าเป็น  $rh4$  และ  $rg4$  ตามลำดับ จากนั้นนำ  $rh4$  และ  $rg4$  ไปทำการ down sampling ก็จะได้ Coefficient เป็น  $rh4d$  และ  $rg4d$  ตามลำดับ ต่อจากนั้นก็ทำการเลือกช่วงของข้อมูลที่ต้องการซึ่งเมื่อทำการเขียนโปรแกรมเสร็จแล้ว เมื่อเราทำการ Run โปรแกรม จะได้ค่า Coefficient ต่างๆ ออกมาดังนี้

$$rg1d = 1.0000 \quad -1.0000 \quad 1.0000 \quad -1.0000 \quad 1.0000 \quad -1.0000 \quad 1.000 \quad -1.0000$$

$$rg2d = 1.0000 \quad -1.0000 \quad 1.0000 \quad -1.0000$$

$$rg3d = 1.0000 \quad -1.0000$$

$$rg4d = -0.9999$$

$$rh4d = 0.9999$$

ซึ่งจะเห็นได้ว่าค่า Coefficient ที่ได้ออกมา  $rg1d, rg2d$  และ  $rg3d$  มีค่าตรงกับค่าของ Coefficient ของข้อมูลที่ทำกรส่งในครั้งแรกและค่า Coefficient ของ  $rg4d$  และ  $rh4d$  มีค่าใกล้เคียงกับข้อมูลที่ใส่ส่งในครั้งแรกมาก ประมาณได้ว่าเท่ากัน เป็นอันว่าเราสามารถที่จะวิเคราะห์ได้ว่าเราพอที่จะทำการส่งข้อมูลโดยผ่านกรรมวิธีของ Daubechies 4 ได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ผ่านมาเป็นอันสรุปได้ว่า สามารถทำการส่งและรับข้อมูลได้ ต่อจากนั้นก็จะต้องทำการวิเคราะห์ในกรณีของการที่มีการเลื่อนของข้อมูล - เนื่องจากในการส่งข้อมูลจริงๆ เมื่อส่งข้อมูลออกไปแล้ว เมื่อข้อมูล ส่งไปผ่าน Low pass filter แล้วจะมีการเลื่อนของข้อมูล จากนั้นก็ทำการวิเคราะห์ว่าสามารถที่จะหาข้อมูลเดิมออกมาได้หรือไม่ ซึ่งจะทำการแสดงให้เห็นต่อไป

### 3.3 การทดสอบสัญญาณเมื่อสัญญาณเกิดการ lagging

ในการส่งสัญญาณไปในสายโทรศัพท์ย่อมต้องเกิดการสูญเสียหรือลดทอนไปในสาย ทำให้สัญญาณที่เราส่งนั้นมีขนาดลดลงและอาจมีการเลื่อนไป (lag) ดังนั้นเราต้องพิสูจน์ให้ได้ว่าทฤษฎีของ Wavelets Transform นี้ยังสามารถนำข้อมูลของเรากลับมาเหมือนเดิมได้โดยไม่ผิดพลาดจากการทดลองเราจะได้ผลการทดลองดังนี้

เรามีข้อมูลคือ [ 1 0 1 1 1 0 1 0 1 1 0 1 0 1 0 1 ] และนำข้อมูลเหล่านี้มาแยกเป็น Coefficient ต่างๆเพื่อทำการส่งโดยให้

$h4d=1, g4d=0, g3d=1, g2d=1, g1d=1$  ตามลำดับ

โปรแกรมที่ใช้ทดสอบการเกิด lagging ของสัญญาณโดยใช้ Matlab

```
h = (sqrt(2)/8)*[ 1+sqrt(3) (3+sqrt(3)) (3-sqrt(3)) (1-sqrt(3)) ]
```

```
h = [ 0.4830 0.8365 0.2241 -0.1294 ]
```

```
k = fliplr(h);
```

```
g = [ k(1) k(2)*(-1) k(3) k(4)*(-1) ];
```

```
g = [ -0.1294 -0.2241 0.8365 -0.4830 ]
```

```
h4d = [1] %coefficient(5)
```

```
g4d = [0] %coefficient(4)
```

```
h4du(1:2:2) = h4d;
```

```
h4du = [ 1 ]
```

```
g4du(1:2:2) = g4d;
```

```
g4du = [ 0 ]
```

```
ih4 = conv(h4du,fliplr(h));
```

```
ih4 = [ -0.1294 0.2241 0.8365 0.4830 ]
```

```
ig4 = conv(g4du,fliplr(g));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ig4 = [ 0 0 0 0 ]
```

```
id = ih4 + ig4;
```

```
id = [-0.1294 0.2241 0.8365 0.4830 ]
```

```
g3d = [1 1] %coefficient(3)
```

```
h3du(1:2:8) = id;
```

```
h3du = [-0.1294 0 0.2241 0 0.8365 0 0.4830 ]
```

```
g3du(1:2:4) = g3d;
```

```
g3du = [ 1 0 1 ]
```

```
g3du = [ 0 0 g3du 0 0 ];
```

```
g3du = [ 0 0 1 0 1 0 0 ]
```

```
ih3 = conv(h3du,fliplr(h));
```

```
ih3 = [ 0.0167 -0.0290 -0.1373 -0.0123 0.0792 0.2958 0.6373 0.5123 0.4040  
0.2333 ]
```

```
ig3 = conv(g3du,fliplr(g));
```

```
ig3 = [ 0 0 -0.4830 0.8365 -0.7071 0.7071 -0.2241 -0.1294 0 0 ]
```

```
ic = ih3 + ig3;
```

```
ic = [ 0.0167 -0.0290 -0.6202 0.8243 -0.6279 1.0029 0.4131 0.3829 0.4040  
0.2333 ]
```

```
g2d = [1 0 1 0] %coefficient(2)
```

```
g2du = [1 0 1 0]
```

```
h2du(1:2:20) = ic;
```

```
h2du = [ 0.0167 0 -0.0290 0 -0.6202 0 0.8243 0 -0.6279 0 1.0029 0  
0.4131 0 0.3829 0 0.4040 0 0.2333 ]
```

```
g2du(1:2:8) = g2d;
```

```
g2du = [1 0 0 0 1 0 0 ]
```

```
g2du=[ 0 0 0 0 0 0 g2du 0 0 0 0 0 ];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

g2du = [0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0]
ih2 = conv(h2du,flipr(h));
ih2 = [-0.0022 0.0038 0.0178 0.0016 0.0560 -0.1530 -0.6255 -0.1148 0.7708
0.2574 -0.6550 -0.0784 0.7854 0.5769 0.2960 0.2853 0.2680 0.2755 0.3078
0.2474 0.1951 0.1127]

ig2 = conv(g2du,flipr(g));
ig2 = [0 0 0 0 0 0 -0.4830 0.8365 -0.2241 -0.1294 -0.4830 0.8365 -
0.2241 -0.1294 0 0 0 0 0 0 0]
ib = ih2+ig2;
ib = [-0.0022 0.0038 0.0178 0.0016 0.0560 -0.1530 -1.1085 0.7217 0.5466
0.1279 -1.1380 0.7581 0.5613 0.4475 0.2960 0.2853 0.2680 0.2755 0.3078
0.2474 0.1951 0.1127]

g1d = [1 1 0 1 0 1 1] %coefficient(1)

h1du(1:2:44) = ib;
h1du = [-0.0022 0 0.0038 0 0.0178 0 0.0016 0 0.0560 0 -0.1530 0 -
1.1085 0 0.7217 0 0.5466 0 0.1279 0 -1.1380 0 0.7581 0 0.5613 0
0.4475 0 0.2960 0 0.2853 0 0.2680 0 0.2755 0 0.3078 0 0.2474 0
0.1951 0 0.1127]
g1du(1:2:16) = g1d;
g1du = [ 1 0 1 0 0 0 1 0 0 0 1 0 0 0 1]
g1du = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
g1du = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

ih1 = conv(h1du,flipr(h));
ih1 = [0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.0206 -0.3736 0.5330 0.4711 0.4407
0.2927 0.2543 -0.1933 -1.0500 -0.3797 0.5615 0.4919 0.4116 0.3714 0.3361

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0.2825 0.2107 0.2069 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 0.1486 0.1195 0.0942 0.0544 ]
```

```
ig1 = conv(g1du,flipr(g));
```

```
ig1 = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.4830 0.8365 -0.7071
0.7071 -0.2241 -0.1294 -0.4830 0.8365 -0.2241 -0.1294 -0.4830 0.8365 -
0.2241 -0.1294 -0.4830 0.8365 -0.2241 -0.1294 0 0 0 0 0 0 0 0 0 0
0 0 0 0 ]
```

```
ia=ih1 + ig1;
```

จะได้สัญญาณที่พร้อมจะส่งมีค่าคือ

```
ia = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741 1.1782 0.2165
0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875 0.2420 -0.1469
1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 0.1486 0.1195 0.0942 0.0544 ]
```

สัญญาณที่ได้หากเราทำการทดลองว่ามันเกิดการ lag ขึ้น 1 columns ดังนั้นข้อมูล columns ที่ 46 จะถูกตัดทิ้งและ columns 1 จะมีข้อมูลอื่นเข้ามาแทน ในที่นี้ใช้ ค่า 0 เข้ามาแทน ข้อมูลที่ได้จะกลายเป็น

```
ia' = [ 0.0000 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147
0.0089 -0.0059 0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -
0.1741 1.1782 0.2165 0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785
1.3284 0.1875 0.2420 -0.1469 1.1190 -0.0134 0.0775 0.2040 0.1979
0.1885 0.1912 0.1906 0.2020 0.2254 0.2041 0.1817 0.1632 0.1486
0.1195 0.0942 ]
```

หลังจากนั้นเราก็ทำการทดลองดูว่าวิธีการที่สามารถนำข้อมูลได้กลับมาเหมือนเดิมนั้นจะต้องทำอย่างไร ได้ทำการทดลองเขียน Matlab และทดลองเปลี่ยนตัวแปรต่างๆเพื่อให้ได้ข้อมูลกลับมาเหมือนเดิมในที่สุดก็สามารถนำข้อมูลกลับมาได้โดยทำตามวิธีการนี้

โปรแกรมที่ใช้สำหรับแก้การ lagging เพื่อสามารถหาข้อมูลกลับมาเหมือนเดิมได้

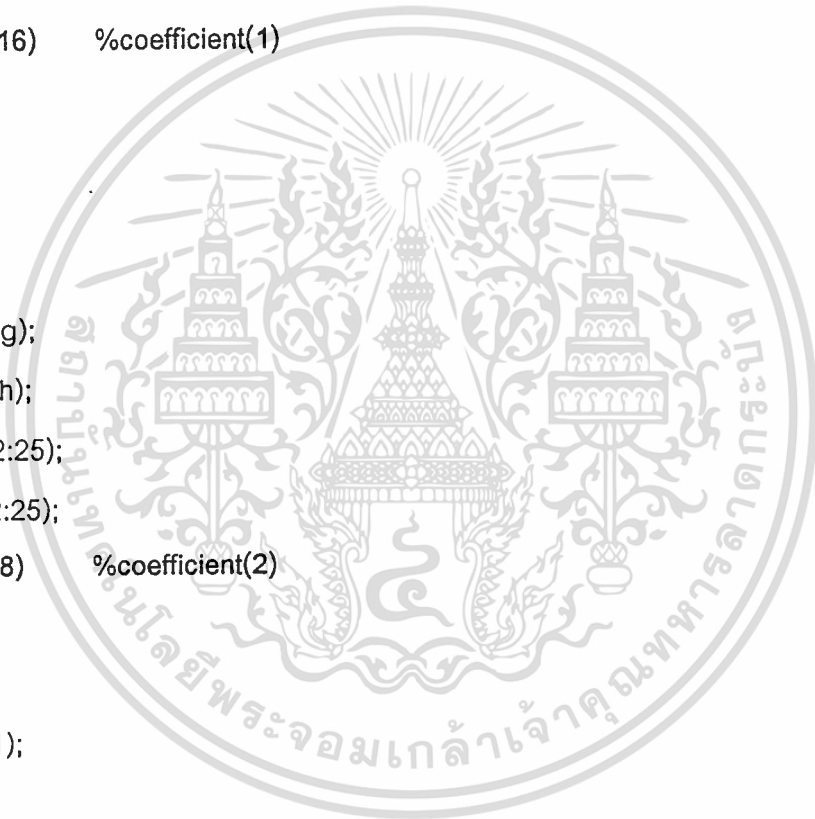
```

format bank
x=x(2:46)
rg1=conv(x,g);
rh1=conv(x,h);
rg1d=rg1(2:2:49);
rh1d=rh1(2:2:49);
rg1d=rg1d(9:16)    %coefficient(1)
pause;
rc=rh1d;
rc=rc(2:23);

rg2=conv(rc,g);
rh2=conv(rc,h);
rg2d=rg2(2:2:25);
rh2d=rh2(2:2:25);
rg2d=rg2d(5:8)    %coefficient(2)
pause;
rb=rh2d;
rb=rb(2:11);

rg3=conv(rb,g);
rh3=conv(rb,h);
rg3d=rg3(2:2:13);
rh3d=rh3(2:2:13);
rg3d=rg3d(3:4)    %coefficient(3)
pause;

```



```

ra=rh3d;
ra=rh3d(2:5);
rg4=conv(ra,g);
rh4=conv(ra,h);
rg4d=rg4(2:2:7);
rh4d=rh4(2:2:7);
rg4d=rg4d(2)    %coefficient(4)
rh4d=rh4d(2)    %coefficient(5)

```

ค่าของ Coefficient ต่างๆหลังจากที่ผ่านการกระทำตามคำสั่งด้านบนแล้วจะได้

```

rg1d = 1.00    1.00    0.00    1.00    0.00    1.00    0.00    1.00
rg2d = 1.00    0.00    1.00    0.00
rg3d = 1.00    1.00
rg4d = 0.00
rh4d = 1.00

```

จะเห็นได้ว่า Coefficient ที่ได้มีค่าตรงกับ Coefficient ของ input แสดงให้เห็นว่าเราสามารถที่ใช้ทฤษฎี Wavelets Transform ได้แม้ว่าในระบบของเราจะเกิดการเลื่อน (lag) และในระบบจริงนั้นเราไม่สามารถทราบได้เลยว่าจะเกิดการเลื่อนมากเท่าไร ดังนั้นก่อนที่เราจะทำการส่งสัญญาณเราต้องมีการตรวจสอบก่อนเสมอว่าระบบของเรานั้นมีการเลื่อนข้อมูลมากเท่าไร เราจึงใช้การทดลองนี้เป็นพื้นฐานในการตรวจสอบโดยที่เราจะส่งข้อมูลที่ทั้งภาครับและที่ภาคส่งนั้นรู้อยู่แล้วว่าสัญญาณที่ส่งมานั้นมี Coefficient เป็นอย่างไร ภาครับจะทำการรับสัญญาณและไปผ่านกระบวนการที่ 1 ซึ่งเขียนโปรแกรมให้แปลงสัญญาณที่มีการเลื่อน 1 ตำแหน่งและนำ Coefficient ที่ได้มาเปรียบเทียบกับ Coefficient ที่รู้อยู่แล้วหากไม่ตรงกันก็เปลี่ยนกระบวนการเป็นกระบวนการที่ 2, 3, ... โดยมีการเลื่อน สัญญาณที่เข้ามาเป็น 2, 3, ... ไปเรื่อยๆจนกว่าจะเจอกระบวนการที่สามารถหาค่า Coefficient ที่ตรงได้เราก็จะทราบว่าระบบเรานั้นจะใช้กระบวนการไหนเป็นกระบวนการหลักในการแปลงสัญญาณกลับ

โปรแกรมด้านล่างต่อไปนี้จะแสดงตัวอย่างการ lagging ซึ่งมีการ lag ถึง 4 ตำแหน่งแต่ยังสามารถนำข้อมูลกลับมาได้เหมือนเดิม ข้อมูลทางด้านส่งจะเหมือนกับตัวอย่างด้านบนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ia = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741 1.1782 0.2165
0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875 0.2420 -0.1469
1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 0.1486 0.1195 0.0942 0.0544 ]
```

เมื่อถูกทำให้ lagging ไป 4 ตำแหน่ง

```
ia' = [ 0 0 0 0 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147
0.0089 -0.0059 0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741
1.1782 0.2165 0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875
0.2420 -0.1469 1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906
0.2020 0.2254 0.2041 0.1817 0.1632 ]
```

ผ่านโปรแกรมที่ทำการนำข้อมูลกลับเมื่อข้อมูลเลื่อนไป 4 ตัวโดยให้ ia' = x

```
x = x(5:46)
```

```
x = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741 1.1782 0.2165
0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875 0.2420 -0.1469
1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 ]
```

```
rg1 = conv(x,g);
```

```
rg1 = [ 0.0000 0.0000 0.0007 0.0000 -0.0017 0.0000 -0.0024 0.0000 0.0083
-0.0001 -0.0208 0.0000 0.0489 0.0000 0.2832 -0.0001 -1.1832 1.0000 -
0.6613 1.0000 -0.3950 0.0001 -0.2494 0.9999 -1.2726 0.0001 -0.0104 1.0000
-0.5200 0.0000 -0.4888 1.0000 -0.5955 0.0000 0.0645 0.0000 -0.0054
0.0000 -0.0073 0.0000 0.0217 0.0000 0.0168 0.0488 -0.0788 ]
```

```

rh1 = conv(x,h);
rh1 = [ 0.0001  0.0000 -0.0015 -0.0022 -0.0002  0.0037  0.0122  0.0178  0.0071
0.0016  0.0408  0.0559  0.0145 -0.1531 -0.9915 -1.1084  0.0079  0.7217  0.9912
0.5465 -0.0778  0.1279 -0.1498 -1.1380 -0.7567  0.7581  1.2852  0.5613  0.0016
0.4476  0.8653  0.2960  0.0156  0.2853  0.2923  0.2680  0.2686  0.2755  0.2958
0.3077  0.2829  0.2474  0.1508  0.0131 -0.0211 ]
rg1d = rg1(2:2:45); %22 columns
rg1d = [ 0.0000  0.0000  0.0000  0.0000 -0.0001  0.0000  0.0000 -0.0001  1.0000
1.0000  0.0001  0.9999  0.0001  1.0000  0.0000  1.0000  0.0000  0.0000  0.0000
0.0000  0.0000  0.0488 ]
rh1d = rh1(2:2:45); %22 columns
rh1d = [ 0.0000 -0.0022  0.0037  0.0178  0.0016  0.0559 -0.1531 -1.1084  0.7217
0.5465  0.1279 -1.1380  0.7581  0.5613  0.4476  0.2960  0.2853  0.2680  0.2755
0.3077  0.2474  0.0131 ]
rg1dr = rg1d(9:16) %coefficient(1)
rg1dr = [ 1.0000  1.0000  0.0001  0.9999  0.0001  1.0000  0.0000  1.0000 ]
pause;

rc = rh1d;
rc = rc(2:21);
rc = [ -0.0022  0.0037  0.0178  0.0016  0.0559 -0.1531 -1.1084  0.7217  0.5465
0.1279 -1.1380  0.7581  0.5613  0.4476  0.2960  0.2853  0.2680  0.2755  0.3077
0.2474 ]

rg2 = conv(rc,g);
rg2 = [ 0.0003  0.0000 -0.0050  0.0000  0.0055  0.0000  0.2238 -0.0001 -1.0857
-1.0001  0.2272  0.0000 -1.2563  1.0001 -0.0353  0.0000 -0.0672  0.0000 -0.0152
0.0000  0.0689  0.0583 -0.1195 ]

rh2 = conv(rc,h);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rh2 = [ -0.0010  0.0000  0.0112  0.0168  0.0318  -0.0291  -0.6511  -0.6201  0.6391
0.8241  -0.4136  -0.6279  0.6336  1.0028  0.5450  0.4131  0.3765  0.3829  0.4022
0.4040  0.2403  0.0156  -0.0320 ]
rg2d = rg2(2:2:23);    %11 columns
rg2d = [ 0.0000  0.0000  0.0000  -0.0001  1.0001  0.0000  1.0001  0.0000  0.0000
0.0000  0.0583 ]
rh2d = rh2(2:2:23);    %11 columns
rh2d = [ 0.0000  0.0168  -0.0291  -0.6201  0.8241  -0.6279  1.0028  0.4131  0.3829
0.4040  0.0156 ]
rg2dr = rg2d(5:8)      %coefficient(2)
rg2dr = [ 1.0001  0.0000  1.0001  0.0000 ]
pause;

rb = rh2d;
rb = rh2d(2:10);
rb = [ 0.0168  -0.0291  -0.6201  0.8241  -0.6279  1.0028  0.4131  0.3829  0.4040 ]

rg3 = conv(rb,g);
rg3 = [ -0.0022  0.0000  0.1008  -0.0001  -0.6081  0.9999  -1.2015  1.0000  -0.2769
0.0302  0.1530  -0.1951 ]
rh3 = conv(rb,h);
rh3 = [ 0.0081  0.0000  -0.3201  -0.1294  0.2509  0.2241  0.7910  0.8364  0.4782
0.3703  0.0410  -0.0523 ]
rg3d = rg3(2:2:12);    %6 columns
rg3d = [ 0.0000  -0.0001  0.9999  1.0000  0.0302  -0.1951 ]
rh3d = rh3(2:2:12);    %6 columns
rh3d = [ 0.0000  -0.1294  0.2241  0.8364  0.3703  -0.0523 ]
rg3dr = rg3d(3:4)      %coefficient(3)
rg3dr = [ 0.9999  1.0000 ]
pause;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ra = rh3d;
ra = rh3d(2:5);
ra = [-0.1294  0.2241  0.8364  0.3703 ]

rg4 = conv(ra,g);
rg4 = [ 0.0167  0.0000 -0.2667  0.0146  0.5085 -0.0943 -0.1788 ]
rh4 = conv(ra,h);
rh4 = [-0.0625  0.0000  0.5625  0.9455  0.4682 -0.0253 -0.0479 ]
rg4d = rg4(2:2:7);
rg4d = [ 0.0146 ]
rh4d = rh4(2:2:7);
rh4d = [ 0.9455 ]
format bank
rg4dr=rg4d(2) %coefficient(4)
rg4dr = [ 0.01 ]
rh4dr=rh4d(2) %coefficient(5)
rh4dr = [ 0.95 ]

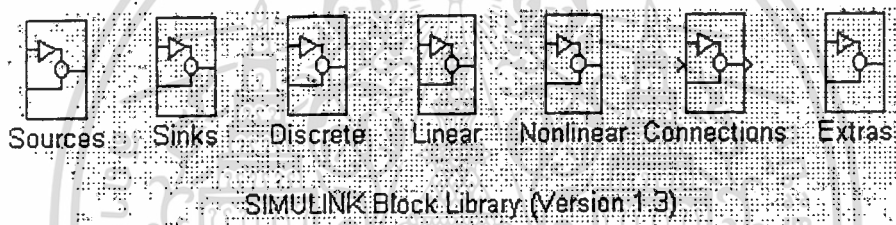
```

จากคำตอบของ Coefficient ของแต่ละตัวเราก็สามารถสรุปได้ว่าแม้ว่าในทางปฏิบัติจริงเมื่อเราทำการแปลงข้อมูล digital โดยใช้ Wavelets Transform แล้วและผ่าน D/A Converter เราจะได้สัญญาณ Analog ที่พร้อมจะส่ง เมื่อส่งไปตามสายโทรศัพท์อาจเกิดการ lagging ขึ้นเมื่อถึงปลายทางเราก็สามารถนำข้อมูลกลับมาได้อย่างแน่นอนโดยใช้หลักการที่เราได้ทำการทดลองและแสดงดังด้านบน

### 3.4 นำสัญญาณผ่าน LowPass Filter

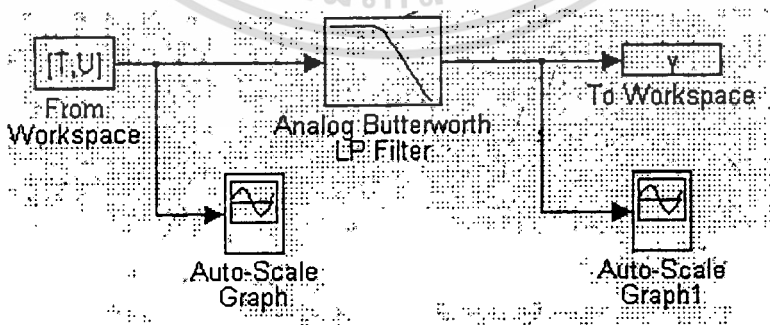
หลังจากที่เราสามารถที่จะพิสูจน์ได้แล้วว่าเมื่อสัญญาณเกิดการ lagging เรายังสามารถนำข้อมูลที่ต้องการกลับมาได้และอีกประเด็นหนึ่งที่สำคัญในการทำงานของระบบนั้นก็คือการที่สัญญาณที่เราส่งไปนั้นต้องผ่านสายโทรศัพท์ซึ่งมีลักษณะการทำงานเป็นเหมือนวงจรกรองความถี่ต่ำเนื่องจากคุณสมบัติของสายโทรศัพท์เองนั้นมี Bandwidth เพียงแค่ 3 KHz ดังนั้นเราจึงจำเป็นต้องทำให้สัญญาณที่เราส่งออกไปนั้นมีความถี่ที่ต่ำกว่า 3 KHz เพื่อให้สามารถส่งข้อมูลได้

ก่อนที่เราจะสามารถทำการทดลองนี้ได้เราต้องศึกษา software matlab ก่อนซึ่งมีตัวเลือกให้เราสามารถทำการทดลองส่งสัญญาณของเราโดยผ่าน Block ซึ่งเป็น Lowpass Filter และเราสามารถกำหนดความถี่คัทออฟเองได้ ตัวเลือกนั้นมีชื่อว่า Simulink ซึ่งมี library ต่างๆมากมายของทั้ง Sources และ Sinks และอื่นๆดังแสดงให้เห็นดังรูปที่ 3.1



รูปที่ 3.1 แสดง library ของ Simulink ใน Software Matlab

หลังจากที่ได้ทำการศึกษา library ต่างๆก็ทำให้สามารถสร้าง Block Diagram จำลองการทำงานเมื่อสัญญาณผ่านวงจรกรองแรงดันต่ำได้ดังรูปที่ 3.2



รูปที่ 3.2 แสดง Block Diagram ที่ใช้สำหรับจำลองการส่งข้อมูลผ่าน LP Filter

จากรูปจะขออธิบาย Block ต่างเพื่อความเข้าใจที่ละ Block ดังนี้

#### 3.4.1 T, U From Workspace

เป็นการกำหนดค่าของสัญญาณ (U) ณ เวลา (T) ต่างๆ เพื่อให้มีการประมวลผลแต่ละจุดของสัญญาณที่เวลาต่างๆกันเช่นเมื่อเรามีสัญญาณต่อเนื่องอยู่ 46 ค่า เราก็ต้องมีค่า T เท่ากับ 46 ค่า เช่นกันหากเราจะทำการทวีสัญญาณเป็นสองเท่า . เพื่อที่จะทำการส่ง ให้มีความถี่ของสัญญาณลดลงครึ่งหนึ่งให้สามารถผ่านวงจรกรองความถี่ต่ำได้ค่าของสัญญาณ 1 ค่า เราก็จะใช้เวลา 2 เวลาที่ห่างเท่ากันแสดงสัญญาณต่อ 1 ค่า ของสัญญาณหรือนั่นก็คือเวลาของการ Sampling เท่าเดิมแต่ค่าของสัญญาณ 1 ค่าจะใช้การ Sampling 2 ครั้ง

#### 3.4.2 Auto-Scale Graph & Graph1

เป็น library ที่ใช้สำหรับแสดงภาพของสัญญาณที่เวลา T ต่างๆทำให้เรารู้ว่าสัญญาณทางด้าน input แตกต่างกับสัญญาณทางด้าน output อย่างไรเมื่อผ่านวงจรกรองความถี่ต่ำ

#### 3.4.3 Analog Butterworth LP Filter

library นี้เปรียบเสมือนคุณสมบัติของสายโทรศัพท์โดยใน Block นี้เราสามารถกำหนดความถี่คัทออฟของสัญญาณได้โดยมีหน่วยเป็น (rad/sec) ดังนั้นเมื่อความถี่คัทออฟของเราอยู่ที่ 3 KHz เราก็จะกำหนดค่าที่ Block นี้ประมาณเท่ากับ 18850 rad/sec

#### 3.4.4 y To Workspace

เป็น library ที่จะนำค่าของสัญญาณเมื่อผ่าน lowpass filter แล้วไปคำนวณต่อใน workspace โดยใช้ตัวแปร y เป็นค่าของสัญญาณนั้นและเราสามารถนำค่าที่ได้นั้นไปคำนวณหาข้อมูลกลับอีกครั้งหนึ่ง

เมื่อเข้าใจการทำงานอย่างคร่าวๆของ Block แต่ละ Block แล้วก็จะแสดงผลการทดลองให้เห็นว่าสามารถที่จะจำลองการทำงานได้หรือไม่ เราจะใช้ข้อมูลเช่นเดียวกับหัวข้อ 3.3 เพื่อที่จะสามารถเปรียบเทียบกันได้ง่าย ข้อมูลที่เราต้องการส่งคือ

[ 1 0 11 1010 11010101 ]

และเมื่อผ่านกระบวนการของ Inverse Wavelet Transform เพื่อเปลี่ยนข้อมูลให้เป็นสัญญาณที่เหมาะสมกับการส่งเราจะได้สัญญาณคือ

```
ia = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741 1.1782 0.2165
0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875 0.2420 -0.1469
1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 0.1486 0.1195 0.0942 0.0544 ]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเราจะเปลี่ยนสัญญาณนี้ให้อยู่ในรูปที่สามารถนำไปใช้ใน Block ของ T, U From Workspace นั่นคือเราต้องสร้างตัวแปร T ขึ้นมาและทำค่าของ ia ให้เป็นค่า U แต่มีเงื่อนไขที่สำคัญข้อหนึ่งคือต้องเปลี่ยนให้อยู่ในรูปที่เป็น 1 column แต่จะมีทั้ง row ก็ได้ลักษณะการเปลี่ยนจะเป็นดังนี้

```
T=(0:0.000335:0.030485)'
```

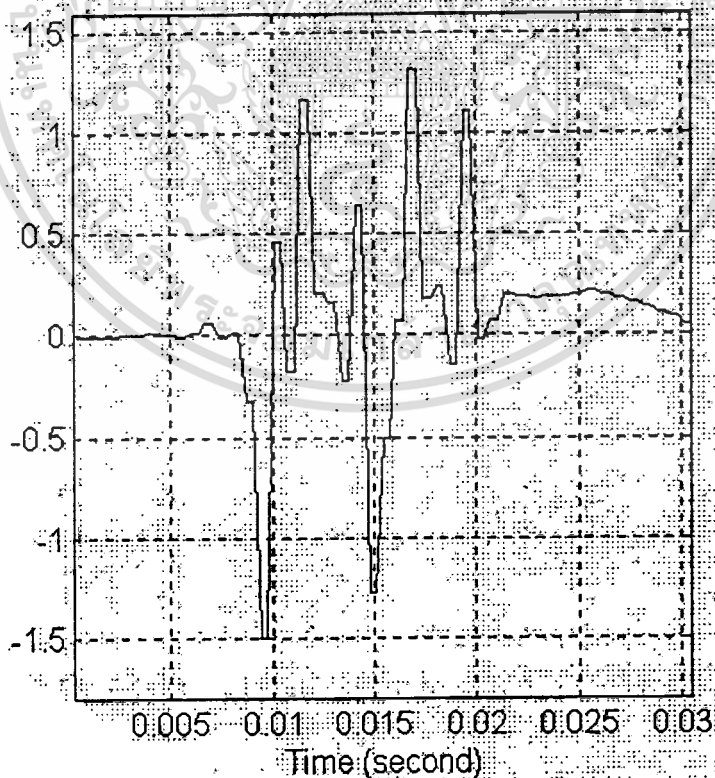
เนื่องจากสัญญาณของเรามีระดับต่างกัน 46 ค่าและเพื่อให้สามารถผ่าน lowpass filter เราจะทำการทวิสัญญาณให้เป็น 2 เท่าดังนั้นหากเรากำหนดให้ค่าช่วงเวลาในการ Sampling 1 ครั้งมีเวลาเท่ากับ 0.000335 sec นั่นก็คือเราใช้ความถี่ sampling ต่ำประมาณ 3 KHz เราก็จะใช้เวลาทั้งหมด 0.030485 sec เพื่อที่จะแสดงสัญญาณทั้ง 46 ค่าค่าละ 2 Sampling

```
ia1=[ia(1) ia(1:0.5:46)]';
```

เป็นการทำให้สัญญาณมีการทวิเป็น 2 เท่า

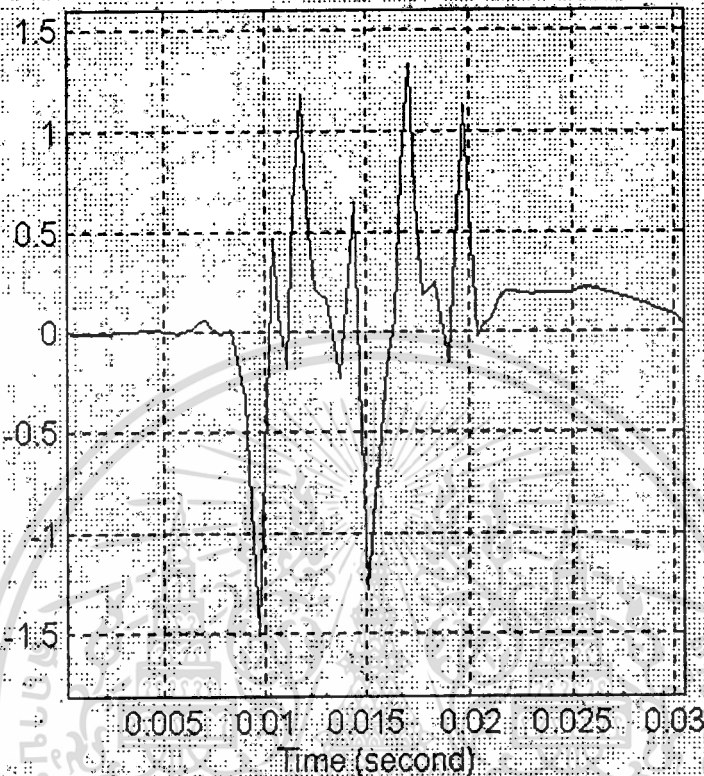
```
U=ia1
```

ต่อไปเป็นการแสดงรูปที่ 3.3 plot ออกมาจาก Auto-Graph ( สัญญาณ input ) & Auto-Graph1 ( สัญญาณ output )



รูปที่ 3.3 แสดงสัญญาณก่อนที่จะทำการผ่านวงจรกรองความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงผลของสัญญาณหลังจากผ่านวงจรรองความถี่ต่ำ

จากภาพจะสังเกตเห็นได้อย่างชัดเจนว่าความถี่สูงจะถูกกำจัดออกไปแต่ว่ารูปร่างของสัญญาณเดิมยังคงอยู่ทำให้เราดูแล้วคิดว่าน่าจะสามารถหาข้อมูลกลับมาได้จากสัญญาณที่ผ่านวงจรรองความถี่ต่ำแล้ว และเราจะได้ระดับของสัญญาณที่ output ออกมาเป็นค่า  $y$  และค่า  $y$  ที่เราได้นี้เราจะนำกลับมาหาข้อมูล ส่วนวิธีการพิสูจน์หาข้อมูลจะกล่าวถึงในหัวข้อต่อไป

แต่ในความเป็นจริงเวลาที่ใช้ในการ sampling สัญญาณ output จะมีเวลาน้อยกว่า 0.000335 sec มากเนื่องจาก Chip DSP ที่เราใช้นั้นเราสามารถที่จะเลือกความถี่ในการ Sampling ได้ซึ่งความถี่นี้จะสามารถแสดงได้ตามตารางที่ 3.1 ด้านล่าง

$F_s$ (Hz)	$20 \log \frac{\sin \pi(f / f_s)}{\pi(f / f_s)}$ ( $f = 3000$ Hz) (dB)
7200	-2.64
8000	-2.11
9600	-1.44
14400	-0.63
19200	-0.35

ตารางที่ 3.1 แสดงความถี่ Sampling ค่าต่างๆ

จากการทดลองจะพบว่าหากว่าเราใช้ความถี่ Sampling ที่มีค่าสูงมาก สัญญาณที่เราจะได้รับที่ปลายทางสามารถที่จะนำมาแปลง Wavelets Transform และได้ข้อมูลออกมาได้อย่างเที่ยงตรงมากยิ่งขึ้น ดังนั้นในการทดลองหัวข้อนี้ทำให้เราทราบว่าสัญญาณที่ได้จากการแปลง Inverse Wavelets Transform เพื่อทำการส่งนั้นเราสามารถที่จะทำการลดความถี่ของสัญญาณโดยอาศัยหลักการเพิ่มความถี่ sampling ให้สูงขึ้นและสัญญาณนั้นสามารถที่จะส่งผ่านสายโทรศัพท์ได้อย่างแน่นอน

### 3.5 การหาข้อมูลกลับจากสัญญาณที่ผ่าน LowPass Filter

จากข้อ 3.4 เราทราบว่าสัญญาณที่เราได้ทำการ Inverse Wavelets Transform แล้วสามารถที่จะทำให้มีความถี่ต่ำและสามารถส่งผ่านสายโทรศัพท์ได้แต่ประเด็นที่สำคัญที่สุดของการพิสูจน์ทฤษฎีนี้คือ เราสามารถหาข้อมูลจากสัญญาณที่ถูกกรองความถี่ต่ำมาแล้วนั้นได้หรือไม่ หากการพิสูจน์ในข้อนี้ไม่สามารถหาวิธีที่จะทำให้สัญญาณกลับคืนมาได้ก็เป็นอันว่าการทดลองที่ทำมาทั้งหมดนั้นไม่เป็นผลสำเร็จแต่ในขั้นต้นนั้นเรามีการทดสอบทฤษฎีคร่าวๆมาแล้วและข้อมูลที่ได้นั้นก็เป็นการสนับสนุนว่าสามารถที่จะทำได้จริงดังนั้นการทดลองในหัวข้อนี้จะนำเอาสัญญาณที่ผ่านกระบวนการส่งและกรองความถี่ต่ำมาแล้วมาแปลง Wavelets Transform กลับดูว่าข้อมูลที่ได้มีความแม่นยำมากเพียงไรและสามารถนำไปใช้งานได้หรือไม่

จากหัวข้อ 3.4 หากเราให้ระบบของเราทำงานที่ความถี่ Sampling เท่ากับ 19.2 KHz ค่าของเวลาในการทำงานในหนึ่ง Sampling จะเท่ากับ 0.0000521 sec เมื่อสัญญาณมีซีกบวก และลบความถี่ Sampling ก็จะลดลงอย่างน้อยครึ่งหนึ่งเหลือความถี่ประมาณ 9.6 KHz ทำให้เกิดการ Sampling เป็น 2 ครั้งต่อสัญญาณ 1 ค่าดังนั้นหากเราต้องการให้มีความถี่ต่ำกว่า 3 KHz เราต้องหารด้วย 4 จะเหลือความถี่ที่จะส่งจริงเท่ากับ 2.4 KHz ก็คือการทำ การ Sampling 4 ครั้งต่อ output 1 ค่าซึ่งยิ่งทำให้มีผลดีต่อการส่งดังนั้นเราจะได้ว่าสัญญาณ 1 ค่าเรา ต้องทำการ Sampling ทั้งหมดเป็น 8 ครั้ง เราจะนำหลักการนี้ทำการเขียนโปรแกรม เพื่อทดสอบ ทฤษฎีดูว่าสามารถที่จะทำได้หรือไม่

โปรแกรมที่ใช้เพื่อทดสอบหาข้อมูลกลับหลังจากสัญญาณผ่าน Low Pass Filter แล้ว

สัญญาณที่ได้จากการแปลงข้อมูลและพร้อมที่จะส่งเราจะใช้สัญญาณเดียวกันกับหัวข้อที่ 3.3 และ 3.4 นั่นคือ

```
ia = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0089 -0.0059
0.0133 0.0666 -0.0073 0.0154 -0.3224 -1.5036 0.4629 -0.1741 1.1782 0.2165
0.1633 -0.2287 0.6432 -1.2742 -0.5091 0.0785 1.3284 0.1875 0.2420 -0.1469
1.1190 -0.0134 0.0775 0.2040 0.1979 0.1885 0.1912 0.1906 0.2020 0.2254
0.2041 0.1817 0.1632 0.1486 0.1195 0.0942 0.0544 ]
```

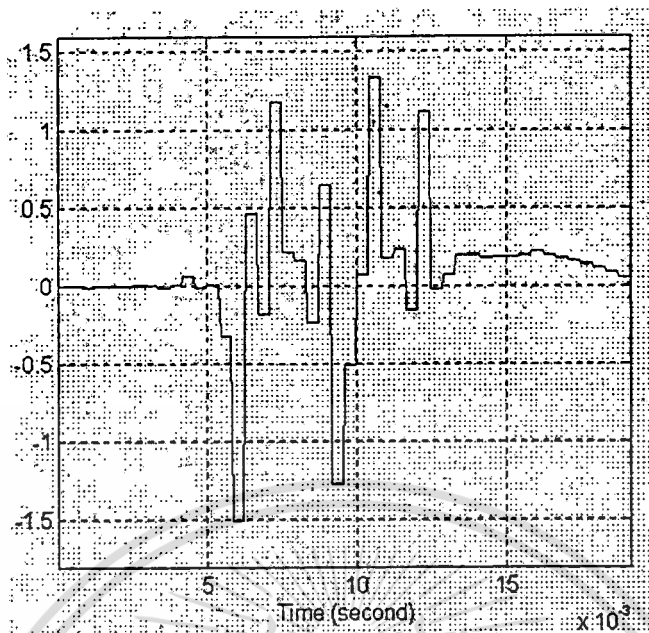
หลังจากนั้นเราก็ต้องกำหนดเวลาในการทำงานเพื่อที่จะทำให้ระบบมีการทำงานที่เหมือนกันกับในทางปฏิบัติ และสามารถนำไปใส่ใน Block Diagram เพื่อที่จะทำการจำลองการทำงานได้

```
T = [ 0:0.0000521:0.01916759 ]'
```

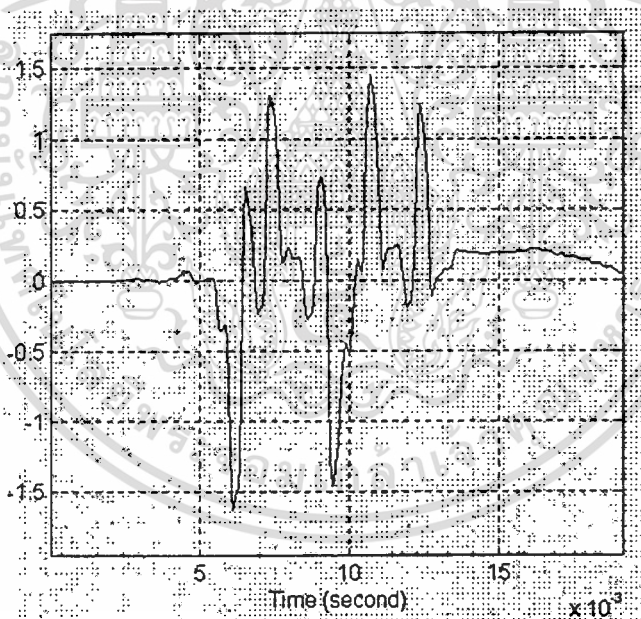
เราจะได้เวลาที่ใช้ในการ Sampling เท่ากับ 368 ครั้งหลังจากนั้นเราก็จะนำสัญญาณ ia มาทวิจำนวนให้มี 8 เท่าเพื่อที่จะได้มีความถี่เหมาะในการส่ง

```
U = [ ia(1) ia(1) ia(1) ia(1) ia(1:0.125:46) ia(46) ia(46) ia(46) ]'
```

เราก็จะได้ข้อมูลทั้งหมดเป็น 368 ค่าซึ่งใช้เวลาส่งเพียง 0.01916759 sec ในการส่งข้อมูลทั้งหมด แสดงข้อมูลเป็นกราฟได้ดังรูปที่ 3.5 และรูปที่ 3.6 ซึ่งอยู่ถัดไปเป็นรูปแสดงสัญญาณหลังจากที่ผ่านวงจรกรองความถี่ต่ำแล้ว



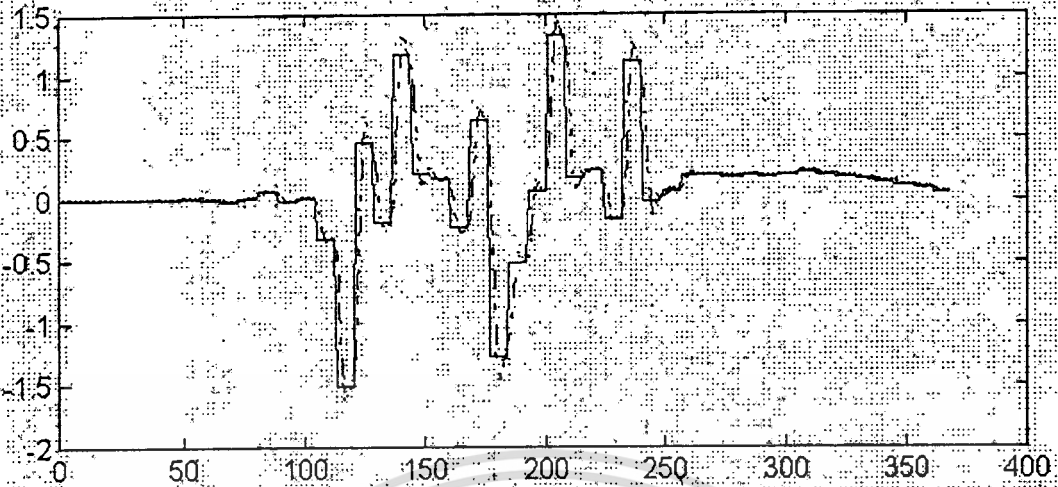
รูปที่ 3.5 ภาพแสดงสัญญาณที่ถูก Sampling แล้วส่งไปตามสายโทรศัพท์



รูปที่ 3.6 ภาพแสดงสัญญาณหลังจากผ่านวงจรรองความถี่ต่ำ

เมื่อเรานำสัญญาณทั้งสองมา plot เพื่อเปรียบเทียบกันเราสามารถเห็นการเปลี่ยนแปลงต่างๆของสัญญาณได้ดียิ่งขึ้นดูรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงการเปรียบเทียบสัญญาณก่อนและหลังผ่านวงจรรองความถี่ต่ำ

จากรูปที่ 3.7 เราจะเห็นได้ว่าสัญญาณของ output นั้นเกิดการ lagging และจะไม่เป็นระดับแบบสัญญาณ input แต่จะมียอดแหลมและค่าของสัญญาณ output คือค่า  $y$  จะมีค่าทั้งหมด 368 ค่าและวิธีที่เราจะนำมาคำนวณหาก็คือการแยกสัญญาณ output  $y$  นั้นออกเป็นสัญญาณย่อยๆ 8 สัญญาณ (เมตริกซ์) ดังนี้

$$A = y(1:8:368)$$

$$A = \begin{bmatrix} 0.0000 & 0.0002 & -0.0007 & -0.0021 & -0.0002 & 0.0012 & 0.0065 & 0.0139 & 0.0076 \\ -0.0037 & 0.0179 & 0.0581 & -0.0032 & -0.0173 & -0.4259 & -1.2842 & 0.3502 & -0.0297 & 1.0507 \\ 0.2362 & 0.1282 & -0.1351 & 0.4374 & -1.1504 & -0.4721 & 0.1811 & 1.1856 & 0.2224 & 0.2044 \\ -0.0159 & 0.9774 & 0.0247 & 0.0881 & 0.1999 & 0.1971 & 0.1890 & 0.1911 & 0.1917 & 0.2039 \\ 0.2227 & 0.2025 & 0.1806 & 0.1623 & 0.1462 & 0.1179 & 0.0911 & & & \end{bmatrix}$$

$$B = y(2:8:368)$$

$$B = \begin{bmatrix} 0.0001 & 0.0000 & -0.0012 & -0.0015 & 0.0001 & 0.0026 & 0.0090 & 0.0125 & 0.0035 \\ 0.0013 & 0.0327 & 0.0388 & 0.0020 & -0.1091 & -0.7527 & -0.7662 & 0.2046 & 0.3313 & 0.8081 \\ 0.2089 & 0.0196 & 0.0966 & -0.0725 & -0.9695 & -0.3027 & 0.5311 & 0.8939 & 0.2217 & 0.0981 \\ 0.3233 & 0.6872 & 0.0340 & 0.1227 & 0.2002 & 0.1945 & 0.1896 & 0.1909 & 0.1948 & 0.2105 \\ 0.2173 & 0.196 & 0.1752 & 0.1580 & 0.1381 & 0.1105 & 0.0799 & & & \end{bmatrix}$$

$$C = y(3:8:368)$$

$$C = \begin{bmatrix} 0.0002 & -0.0003 & -0.0018 & -0.0008 & 0.0005 & 0.0044 & 0.0123 & 0.0105 & -0.0018 \\ 0.0080 & 0.0520 & 0.0131 & 0.0090 & -0.2297 & -1.1803 & -0.0800 & 0.0051 & 0.8066 & 0.4836 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1762 -0.1218 0.4029 -0.7460 -0.7235 -0.0819 0.9873 0.5042 0.2249 -0.0407  
 0.7706 0.3004 0.0504 0.1689 0.1999 0.1911 0.1904 0.1908 0.1989 0.2190  
 0.2100 0.1878 0.1682 0.1525 0.1274 0.1011 0.0652 ]

D = y(4:8:368)

D = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0090 -0.0059  
 0.0131 0.0665 -0.0066 0.0147 -0.3209 -1.5015 0.4455 -0.1564 1.1685 0.2308  
 0.1564 -0.2279 0.6366 -1.2597 -0.5270 0.0815 1.3283 0.2022 0.2332 -0.1453  
 1.1108 0.0010 0.0686 0.2038 0.1990 0.1885 0.1911 0.1906 0.2020 0.2254  
 0.2044 0.1816 0.1631 0.1485 0.1195 0.0941 0.0543 ]

E = y(4:8:368)

E = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0058 0.0147 0.0090 -0.0059  
 0.0131 0.0665 -0.0066 0.0147 -0.3209 -1.5015 0.4455 -0.1564 1.1685 0.2308  
 0.1564 -0.2279 0.6366 -1.2597 -0.5270 0.0815 1.3283 0.2022 0.2332 -0.1453  
 1.1108 0.0010 0.0686 0.2038 0.1990 0.1885 0.1911 0.1906 0.2020 0.2254  
 0.2044 0.1816 0.1631 0.1485 0.1195 0.0941 0.0543 ]

F = y(5:8:368)

F = [ 0.0003 -0.0006 -0.0025 0.0000 0.0009 0.0063 0.0156 0.0084 -0.0074  
 0.0151 0.0721 -0.0144 0.0173 -0.3562 -1.6241 0.6540 -0.2273 1.3107 0.1273  
 0.1527 -0.2684 0.7287 -1.4621 -0.4427 0.1417 1.4573 0.0798 0.2412 -0.1857  
 1.2441 -0.1202 0.0805 0.2171 0.1980 0.1875 0.1914 0.1906 0.2031 0.2278  
 0.2021 0.1793 0.1612 0.1470 0.1165 0.0915 0.0503 ]

G = y(6:8:368)

G = [ 0.0003 -0.0006 -0.0024 0.0000 0.0009 0.0062 0.0154 0.0084 -0.0071  
 0.0148 0.0708 -0.0131 0.0171 -0.3491 -1.5974 0.6174 -0.2225 1.2843 0.1417  
 0.1580 -0.2596 0.7118 -1.4250 -0.4505 0.1260 1.4278 0.0983 0.2451 -0.1775  
 1.2188 -0.1018 0.0835 0.2142 0.1975 0.1877 0.1914 0.1906 0.2029 0.2273  
 0.2024 0.1799 0.1617 0.1474 0.1172 0.0922 0.0512 ]

H = y(7:8:368)

H = [ 0.0003 -0.0005 -0.0023 -0.0002 0.0008 0.0059 0.0148 0.0088 -0.0062  
 0.0137 0.0675 -0.0087 0.0159 -0.3285 -1.5241 0.5002 -0.1889 1.2033 0.1969

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.1639 -0.2355 0.6595 -1.3100 -0.4921 0.0879 1.3496 0.1648 0.2449 -0.1538  
 1.1423 -0.0359 0.0810 0.2063 0.1976 0.1883 0.1913 0.1906 0.2022 0.2258  
 0.2037 0.1813 0.1629 0.1484 0.1190 0.0938 0.0537 ]

$l = y(8:8:368)$

$l = [ 0.0003 -0.0005 -0.0022 -0.0002 0.0008 0.0057 0.0145 0.0089 -0.0055$   
 $0.0132 0.0650 -0.0057 0.0134 -0.3208 -1.4669 0.4185 -0.1570 1.1446 0.2349$   
 $0.1643 -0.2155 0.6151 -1.2296 -0.5194 0.0703 1.2938 0.2106 0.2410 -0.1320$   
 $1.0860 0.0097 0.0783 0.2010 0.1977 0.1887 0.1912 0.1907 0.2019 0.2248$   
 $0.2044 0.1821 0.1636 0.1488 0.1200 0.0946 0.0552 ]$

หลังจากเราได้สัญญาณทั้ง 8 แล้วนำแต่ละตัวมาผ่านกระบวนการเวฟเล็ตทรานฟอร์มเพื่อดูว่าสัญญาณแต่ละสัญญาณนั้นสามารถให้ข้อมูลได้เป็นอย่างไรและสัญญาณไหนสามารถค้นหาข้อมูลกลับได้ดีที่สุด

ข้อมูลที่ส่งไปคือ

$h4d=1,$   $g4d=0,$   $g3d=1\ 1,$   $g2d=1\ 0\ 1\ 0,$   $g1d=1\ 1\ 0\ 1\ 0\ 1\ 0\ 1$

หากหาข้อมูลจากสัญญาณ A เราจะได้

$rg1d = [ -0.9432 -0.4601 -0.3211 -0.1241 -1.0912 0.0850 -0.4303 -0.3343 ]$

$rg2d = [ 0.3060 -0.3553 -0.6829 -0.7156 ]$

$rg3d = [ 1.1658 0.9177 ]$

$rg4d = [ 0.04 ]$

$rh4d = [ 1.05 ]$

หากหาข้อมูลจากสัญญาณ B เราจะได้

$rg1d = [ -0.3661 -0.0298 -0.2324 0.2131 -0.7769 0.3614 -0.3101 0.0646 ]$

$rg2d = [ 0.5241 -0.2531 -0.1743 -0.5064 ]$

$rg3d = [ 1.1185 0.9457 ]$

$rg4d = [ 0.03 ]$

$rh4d = [ 1.03 ]$

หากหาข้อมูลจากสัญญาณ C เราจะได้

$rg1d = [ 0.3976 0.5411 -0.1110 0.6567 -0.3543 0.7233 -0.1460 0.5904 ]$

$rg2d = [ 0.8040 -0.1152 0.4929 -0.2266 ]$

$rg3d = [ 1.0543 0.9797 ]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$rg4d = [ 0.01 ]$$

$$rh4d = [ 1.01 ]$$

หากหาข้อมูลจากสัญญาณ D เราจะได้

$$rg1d = [ 0.9817 \quad 0.9810 \quad -0.0119 \quad 0.9931 \quad -0.0226 \quad 0.9963 \quad -0.0138 \quad 0.9903 ]$$

$$rg2d = [ 1.0078 \quad -0.0069 \quad 0.9953 \quad -0.0100 ]$$

$$rg3d = [ 1.0036 \quad 1.0029 ]$$

$$rg4d = [ 0.00 ]$$

$$rh4d = [ 1.00 ]$$

หากหาข้อมูลจากสัญญาณ E เราจะได้

$$rg1d = [ 0.9817 \quad 0.9810 \quad -0.0119 \quad 0.9931 \quad -0.0226 \quad 0.9963 \quad -0.0138 \quad 0.9903 ]$$

$$rg2d = [ 1.0078 \quad -0.0069 \quad 0.9953 \quad -0.0100 ]$$

$$rg3d = [ 1.0036 \quad 1.0029 ]$$

$$rg4d = [ 0.00 ]$$

$$rh4d = [ 1.00 ]$$

หากหาข้อมูลจากสัญญาณ F เราจะได้

$$rg1d = [ 1.2128 \quad 1.1579 \quad 0.0321 \quad 1.1242 \quad 0.1155 \quad 1.1015 \quad 0.0439 \quad 1.1469 ]$$

$$rg2d = [ 1.0805 \quad 0.0379 \quad 1.1880 \quad 0.0776 ]$$

$$rg3d = [ 0.9824 \quad 1.0100 ]$$

$$rg4d = [ 0.00 ]$$

$$rh4d = [ 0.99 ]$$

หากหาข้อมูลจากสัญญาณ G เราจะได้

$$rg1d = [ 1.1716 \quad 1.1296 \quad 0.0298 \quad 1.0985 \quad 0.0984 \quad 1.0798 \quad 0.0396 \quad 1.1173 ]$$

$$rg2d = [ 1.0588 \quad 0.0322 \quad 1.1468 \quad 0.0639 ]$$

$$rg3d = [ 0.9849 \quad 1.0065 ]$$

$$rg4d = [ 0.00 ]$$

$$rh4d = [ 1.00 ]$$

หากหาข้อมูลจากสัญญาณ H เราจะได้

$$rg1d = [ 1.0411 \quad 1.0325 \quad 0.0094 \quad 1.0227 \quad 0.0267 \quad 1.0179 \quad 0.0119 \quad 1.0274 ],$$

$$rg2d = [ 1.0106 \quad 0.0086 \quad 1.0325 \quad 0.0161 ]$$

$$rg3d = [ 0.9959 \quad 1.0007 ]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rg4d = [ 0.00 ]

rh4d = [ 1.00 ]

หากหาข้อมูลจากสัญญาณ I เราจะได้

rg1d = [ 0.9456 0.9594 -0.0073 0.9614 -0.0238 0.9699 -0.0111 0.9599 ]

rg2d = [ 0.9806 -0.0082 0.9599 -0.0131 ]

rg3d = [ 0.9978 0.9953 ]

rg4d = [ 0.00 ]

rh4d = [ 1.00 ]

จากการทดลองพบว่า การเลือกสัญญาณที่อยู่ในช่วงกลางระหว่าง A ถึง I ค่าที่ดีที่สุดคือค่าสัญญาณ F, G และ H เพราะค่าที่ได้มีความใกล้เคียง กับข้อมูลที่ส่งไปมากที่สุดและจากการทดลองนี้ทำให้เราทราบว่าสัญญาณเกิดการ lagging อย่างแน่นอนเพราะสัญญาณ A - E คือสัญญาณในช่วงแรกที่ได้รับได้หลังจากมีการส่ง แต่ไม่สามารถนำกลับมาหาข้อมูลได้ต้องใช้หลักการของการ lagging ทำการทดสอบเพื่อที่จะรู้ว่าสัญญาณมีการ lagging ที่ Sampling ในการทดลองโดยใช้ Matlab นี้เราพบว่ามีการ lagging ไปถึง 5 Sampling

จากผลการทดลองในหัวข้อนี้สามารถให้คำตอบกับเราได้ว่าการนำทฤษฎีเวฟเล็ตทรานฟอร์มมาใช้เพื่อทำการสื่อสารข้อมูลของคอมพิวเตอร์โดยผ่านทางสายโทรศัพท์สามารถนำมาใช้ได้เป็นอย่างดีส่วนการที่จะทำให้ข้อมูลของเราส่งได้มากและเร็วเท่าไรนั้นก็ขึ้นอยู่กับข้อจำกัดหลายอย่างเช่นการออกแบบ, Chip DSP เองหรือ แม้กระทั่ง Software ที่เป็นโปรแกรมเพื่อจะสั่งให้ DSP ทำงานตามขั้นตอนหากเขียนโปรแกรมไม่ดีก็อาจทำให้เสียเวลาในการคำนวณสูงมีผลทำให้ส่งข้อมูลได้ช้า

## บทที่ 4

### ผลการทดลอง

การที่เราจะทำให้ระบบการสื่อสารสามารถติดต่อกันได้อย่างมีประสิทธิภาพนั้น จำเป็นจะต้องมีการวางระบบการสื่อสารที่มีประสิทธิภาพ เช่นเดียวกันในโครงการที่กำลังกระทำอยู่นี้จะไม่สามารถสำเร็จลุล่วงลงได้เลยถ้าไม่มีการวางระบบรวมถึงวิธีการติดต่อที่มีประสิทธิภาพ

ซึ่งในโครงการนี้ได้ใช้การติดต่อระหว่างเครื่องคอมพิวเตอร์และบอร์ด TMS320C5X ซึ่งจะต้องใช้ทั้งการส่งข้อมูลและรับข้อมูล นอกจากนี้แล้วก็จะเป็นส่วนของการติดต่อระหว่างบอร์ดทดลอง TMS320C5X กับ บอร์ด TMS320C5X ซึ่งจะได้ขอกกล่าวต่อไปโดยจะแบ่งเป็น 2 ส่วนคือ

- 1.) การใช้คอมพิวเตอร์เป็นหลัก จะใช้คอมพิวเตอร์เป็นตัวส่งข้อมูลและใช้คอมพิวเตอร์เป็นตัวรับข้อมูลจากบอร์ด TMS320C5X
- 2.) การใช้บอร์ด TMS320C5X เป็นหลักในการประมวลผลจะใช้ในการติดต่อระหว่างบอร์ด TMS320C5X กับบอร์ด TMS320C5X ซึ่งจะเป็นการรับส่งข้อมูลที่ผ่านการกระทำกระบวนการทางคณิตศาสตร์ที่เรียกว่า การแปลงเวฟเล็ตทรานฟอร์ม

ต่อไปจะขอกกล่าวโดยใช้คอมพิวเตอร์เป็นหลัก ซึ่งจะประกอบไปด้วยการใช้คอมพิวเตอร์เป็นตัวส่งข้อมูลและการใช้คอมพิวเตอร์เป็นตัวรับข้อมูล ในการทำโครงการนี้เราได้ใช้ภาษาซีเพื่อควบคุมการส่งและรับข้อมูลของคอมพิวเตอร์ ซึ่งจะขอกกล่าวถึงขั้นตอนในการรับและส่งดังต่อไปนี้

#### 4.1 การส่งและรับข้อมูลโดยใช้เครื่องคอมพิวเตอร์เป็นหลัก

##### 4.1.1 การส่งข้อมูลโดยใช้เครื่องคอมพิวเตอร์

ในการส่งข้อมูลโดยใช้ภาษาซีในโครงการนี้เราจะต้องทำการติดต่อกับพอร์ตของคอมพิวเตอร์ 3 พอร์ต คือ พอร์ต 0378 พอร์ต 0379 และพอร์ต 037A พอร์ตทั้ง 3 นี้จะทำหน้าที่เป็นตัวส่งและรับข้อมูลรวมถึงการส่งและรับสัญญาณชิงโครนัส ขั้นตอนการทำงานของโปรแกรมจะเป็นดังต่อไปนี้ เริ่มต้นด้วยการเปิดไฟล์โดยการป้อนชื่อไฟล์ที่จะทำการส่ง เมื่อทำการป้อนชื่อไฟล์เสร็จโปรแกรมก็จะทำการรับชื่อไฟล์ไปเช็คความสามารถทำการเปิดไฟล์ชื่อดังกล่าวได้หรือไม่ถ้าไม่สามารถทำการเปิดไฟล์ดังกล่าวได้ก็จะทำการฟ้องว่าไม่สามารถทำการเปิดไฟล์ได้ โดยจะฟ้องว่า Error in open file แต่ถ้าสามารถทำการเปิดไฟล์ดังกล่าวได้โปรแกรมก็จะทำงานต่อไป จากนั้นก็จะทำการเช็คตัวอักขระที่พอยเตอร์ fp ซึ่งอยู่เป็น eof ของโปรแกรมหรือเปล่า โดยใช้ฟังก์ชัน feof(fp) ในภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่โดยไม่หวังกำไร  
ไม่หวังกำไรใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซี ถ้าเป็น eof ของโปรแกรมก็จะหลุดจากลูปแล้วก็จะไปทำตามกระบวนการต่อไป แต่ถ้าตัวอักษรที่รับเข้ามาไม่ได้เป็น eof ก็จะทำงานในลูปต่อไป หลังจากนั้นก็จะทำการนำตัวอักษรที่ fp ซ้ำอยู่ไปเก็บไว้ในตัวแปร ch แล้วก็ทำการเช็คในตัวที่ fp ซ้ำอยู่ Error หรือไม่ ถ้า Error โปรแกรมจะฟ้องว่า file error แต่ถ้าไม่ก็จะทำงานต่อไป โดยจะทำการเก็บข้อมูลในตัวแปร ch ไว้ใน ตัวแปร value1 จากนั้นก็ทำการแสดงข้อมูลดังกล่าวที่หน้าจอภาพเพื่อแสดงให้รู้ว่าทำงานไปถึงไหนแล้วและในขณะเดียวกันก็ทำการนำข้อมูลที่อยู่ใน value1 ส่งออกที่พอร์ต 0378 ต่อจากนั้นก็ทำการส่ง 0xFF และ 0x00 ออกที่พอร์ต 037A เพื่อให้ทำหน้าที่เป็น CLOCK ไปคอนโทรลให้ฮาร์ดแวร์ทำงาน และต่อไปก็จะทำการรับข้อมูลจากพอร์ต 0379 เพื่อเช็คดูว่าบอร์ด TMS320C5X พร้อมทั้งจะรับข้อมูลตัวต่อไปหรือยังโดยที่จะทำการวนรับข้อมูลจากพอร์ต 0379 จนกว่าจะได้ค่าที่ถูกต้องแล้วจึงไปวนลูปไปทำซ้ำกับข้างบนอีกทีหนึ่งคือ ตรงที่เริ่มเช็คค่าถึง eof ของโปรแกรมหรือยัง หลังจากนั้นก็จะทำซ้ำไปเรื่อยๆ จนกว่าจะอ่านข้อมูลในไฟล์จนหมด เป็นอันว่าจบการทำงานของโปรแกรมในส่วนของส่งข้อมูลให้บอร์ด TMS320C5X

ผลการทดลองที่ได้จากการทำงานของโปรแกรมภาษาซีคือเรามีเพิ่มข้อมูลที่กำหนดอย่างแน่นอนแล้วเพื่อทำการส่งไปให้บอร์ด TMS320C5X จากนั้นก็ Run โปรแกรมควบคุมการทำงานของ TMS320C5X เพื่อทำงานตามการ interrupt ที่ถูกควบคุมมานำค่าที่ได้มาเก็บไว้ใน memory ส่วนใดส่วนหนึ่งหลังจากที่ทำการรับข้อมูลเสร็จเรียบร้อยแล้วเราจะทำการตรวจสอบข้อมูลที่รับเข้ามาได้ว่ามีค่าเป็นอะไรบ้างแสดงตัวอย่างที่รับได้คือ

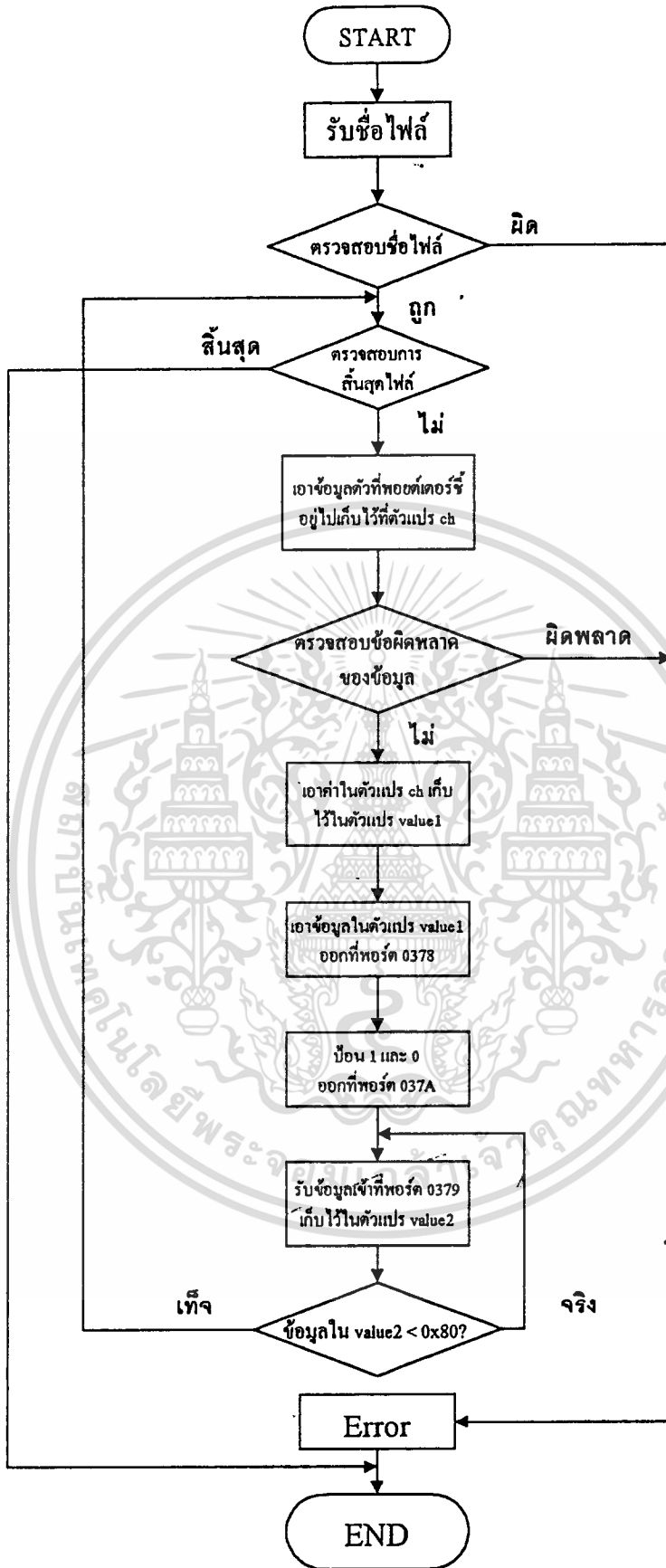
Using DSP in communication ← File ดั้งแบบที่ทำการส่งมาจากคอมพิวเตอร์  
ค่าที่อ่านได้จาก memory ของบอร์ดคือ

.U.s.i.n.g...D.S.P...i.n...c.o.m.m.u.n.i.c.a.t.i.o.n.d.a.

พิจารณาค่าที่ได้เช่นที่ตัวแรกเราได้ .U ใน memory จะมีค่าคือ 0055<sub>16</sub> จุดค่าแรกก็คือ 00 ส่วนตัว U คือค่า 55 นั่นเอง

พิจารณาค่าสุดท้ายเราจะเห็นว่าข้อมูลควรจะหมดที่ .n แต่ที่ต้องมีค่า .d.a ทั้งนี้เนื่องจากเรามีการ Enter ที่ท้ายบรรทัดดังนั้นจึงมีค่า .d.a เป็นตัวบอกว่ามีการเริ่มขึ้นบรรทัดใหม่

จากผลที่ได้เราทำการส่งข้อมูลโดยใช้ข้อมูลต้นแบบเป็นเพิ่มอื่นที่กำหนดขึ้นมาแล้วนำมาเปรียบเทียบปรากฏว่าสามารถตรวจสอบแล้วพบความถูกต้องทุกครั้งที่เราส่งข้อมูล ภาพในหน้าถัดไปเป็นการแสดง Flow chart ของขั้นตอนการทำงานในช่วงที่เครื่องคอมพิวเตอร์ทำการส่งข้อมูลให้กับบอร์ด TMS320C5X



รูปที่ 4.1 แสดง Flow Chart การส่งข้อมูลจากคอมพิวเตอร์ให้บอร์ด TMS320C5X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 การรับข้อมูลโดยใช้เครื่องคอมพิวเตอร์

การรับข้อมูลโดยใช้เครื่องคอมพิวเตอร์เป็นการรับข้อมูลที่ผ่านการประมวลผลโดยบอร์ด TMS320C5X เรียบร้อยแล้วและส่งข้อมูลมาให้เครื่องคอมพิวเตอร์เพื่อตรวจสอบว่าได้ตรงกับข้อมูลที่ทำการส่งไปครั้งแรกให้กับบอร์ด TMS320C5X หรือไม่ ซึ่งในการรับข้อมูลจากบอร์ด TMS320C5X ก็ยังคงทำการรับและส่งข้อมูลและสัญญาณซิงโครนัส กับพอร์ต 0378 พอร์ต 0379 และพอร์ต 037A อยู่เช่นเดิม จากนั้นจะกล่าวถึงหลักการในการทำงานของโปรแกรมภาครับ เริ่มต้นจากการทำการเปิดไฟล์เพื่อทำการเขียนข้อมูลว่าสามารถทำการเปิดไฟล์ได้หรือไม่ ถ้าไม่สามารถทำการเปิดไฟล์เพื่อเขียนได้ก็จะฟ้องว่า Error in opening file แต่ถ้าสามารถทำการเปิดไฟล์ดังกล่าวได้ก็จะทำงานตามโปรแกรมต่อไป จากนั้นก็ทำการส่งข้อมูล 0xFF ออกที่พอร์ต 0378 เพื่อทำการ clear .ให้สามารถรับข้อมูลที่ port นั้นได้ จากนั้นก็ทำการป้อนค่า 0x03 และ 0x04 ออกที่พอร์ต 037A เพื่อส่งไปเพื่อทำหน้าที่เป็น CLOCK เพื่อเป็นตัวไปบอกว่าพร้อมที่จะรับข้อมูล จากนั้นก็จะทำการรับข้อมูลจากพอร์ต 0379 หลังจากนั้นก็จะทำการนำค่าดังกล่าวที่รับมาได้ไปทำการ AND กับค่าที่ตั้งไว้เพื่อเช็คความถูกต้องคือ ค่า 0x10 ต่อจากนั้นก็ทำการเช็คค่าที่ได้มันตรงกับค่า 0x10 หรือไม่ ถ้าผลการคำนวณที่ตั้งไว้ยังไม่ถูกต้องก็จะวนรับค่าต่อไป แล้วทำการประมวลผลซ้ำๆ กันไปจนกว่าจะได้ที่ถูกต้อง ในการรับค่าของข้อมูลในครั้งแรกนี้จะทำการวนรับค่าที่พอร์ต 0379 จนกว่าจะได้รับค่าที่ถูกต้องจึงจะนำค่าของข้อมูลที่รับมาได้ไปทำการเก็บในไฟล์ ในการนำค่าของข้อมูลที่จะนำไปเก็บไว้ในไฟล์ก็จะทำการเช็คด้วยว่าค่าที่รับมานั้นเป็น eof ของไฟล์หรือไม่ ถ้าไม่ใช่ก็ทำการเก็บข้อมูลได้เลย แต่ถ้าใช่ก็จะฟ้องว่า Error in writing file เมื่อเสร็จแล้วก็จะเริ่มทำการรับข้อมูลในครั้งต่อไป แต่ในการวนรับค่าของข้อมูลในครั้งต่อไป ก็จะทำให้เช็คดูว่าค่าของข้อมูลที่รับมาจากพอร์ต 0379 นั้น เมื่อนำค่าที่รับมาได้มันไปทำการ AND กับค่า 0x10 แล้ว ได้ค่าออกมาเป็น 0x10 หรือไม่ถ้าใช่ก็จะหลุดจากลูปแล้วทำการเก็บข้อมูลต่อไป แต่ว่าค่าที่ผ่านการคำนวณที่ได้มาอย่างคงมีค่าไม่ถูกต้องตามที่ตั้งไว้ ก็จะทำให้เช็คด้วยว่าค่าที่ผ่านการคำนวณที่ถูกต้องนั้นขาดหายไปนานกว่าที่ตั้งไว้หรือไม่ ถ้านานเกินกว่าช่วงเวลาที่ตั้งไว้มากก็จะทำการปิดไฟล์ แต่ถ้าทำการรับข้อมูลที่ถูกต้องเข้ามาได้ แล้วก็ทำการรับค่าจากพอร์ตที่รับส่งข้อมูลต่อไป(พอร์ต 0378) แล้วก็ไปทำการเก็บข้อมูลไว้ในไฟล์ต่อไป สาเหตุที่ต้องทำการเช็คระยะเวลาเนื่องจากว่าข้อมูลบางข้อมูลที่ทำการรับมานั้นข้อมูลที่รับมาไม่สามารถตรวจเจอ eof ของไฟล์ได้ จึงมีไว้เพื่อไม่ให้โปรแกรมต้องทำงานต่อไปโดยไม่จำเป็น แต่ถ้าข้อมูลที่รับมาได้มีตัว eof ของข้อมูลมาด้วยโปรแกรมก็จะทำการเช็คค่าดังกล่าวด้วย ซึ่งถ้าสามารถทำการเช็คค่าดังกล่าวได้โปรแกรมก็จะทำการปิดไฟล์ทันที

ในการทดลองในครั้งแรกๆ นั้นยังไม่ทราบว่ากรรับข้อมูลมาจากบอร์ด TMS320C5X นั้นจะส่งข้อมูลการสิ้นสุดไฟล์มาให้ด้วยหรือไม่ จึงได้ทำการเช็คค่าของเลขศูนย์ที่รับเข้ามาได้คือ ถ้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถทำการรับค่า 00<sub>16</sub> ที่ให้ค่าเป็นเลขศูนย์ได้ติดต่อกันตั้งแต่ 5 ตัวขึ้นไปโปรแกรมก็จะหยุดการทำงาน ในการทดลองในครั้งนั้นก็จะทำการรับค่าของข้อมูลเข้ามา 5 ตัว แล้วนำค่าของข้อมูลที่รับมาได้มาทำการบวกกันถ้าผลบวกเป็นศูนย์ก็จะทำการปิดไฟล์ได้เลย แต่ถ้าผลบวกที่ได้ยังไม่เท่ากับศูนย์ก็จะทำการนำข้อมูลตัวแรกที่รับเข้ามาไปทำการเก็บไว้ในไฟล์ที่เปิดออกมาก่อน จากนั้นก็จะทำการเลื่อนตำแหน่งของข้อมูลที่หรืออีก 4 ตัว เลื่อนมาไว้ข้างหน้า ต่อจากนั้นก็ทำการรับข้อมูลเพิ่มเข้ามาอีก 1 ตัวมาเก็บไว้ในข้อมูลที่จะทำการเช็คตัวสุดท้ายที่ถูกเลื่อนไปเรียบร้อยแล้ว ทำการเช็คต่อไปว่าผลบวกของทั้ง 5 ค่าที่ได้เป็นศูนย์หรือไม่ ถ้าเป็นศูนย์ก็ทำการปิดไฟล์และถ้ายังไม่เท่ากับศูนย์ก็ทำการรับข้อมูลต่อไปเรื่อยๆ จนกว่าค่าผลบวกที่ได้จะเป็นศูนย์

ผลการทดลองที่ได้จากการรับข้อมูลจากบอร์ด TMS320C5X เราทำการทดลองโดยใส่ข้อมูลเข้าไปไว้ใน memory ของบอร์ดแล้วทำการส่งข้อมูลนั้นออกมาให้กับเครื่องคอมพิวเตอร์โดยใช้การควบคุมการทำงานจากโปรแกรมภาษาซีแล้วทำการอ่านค่าที่คอมพิวเตอร์รับเข้าไปผ่านทางจอภาพตรวจสอบความถูกต้องขั้นต้นใช้ข้อมูลที่มีความแตกต่างกันทุกตัวอักษรดังผลด้านล่าง

Weerachai Piyakawanit

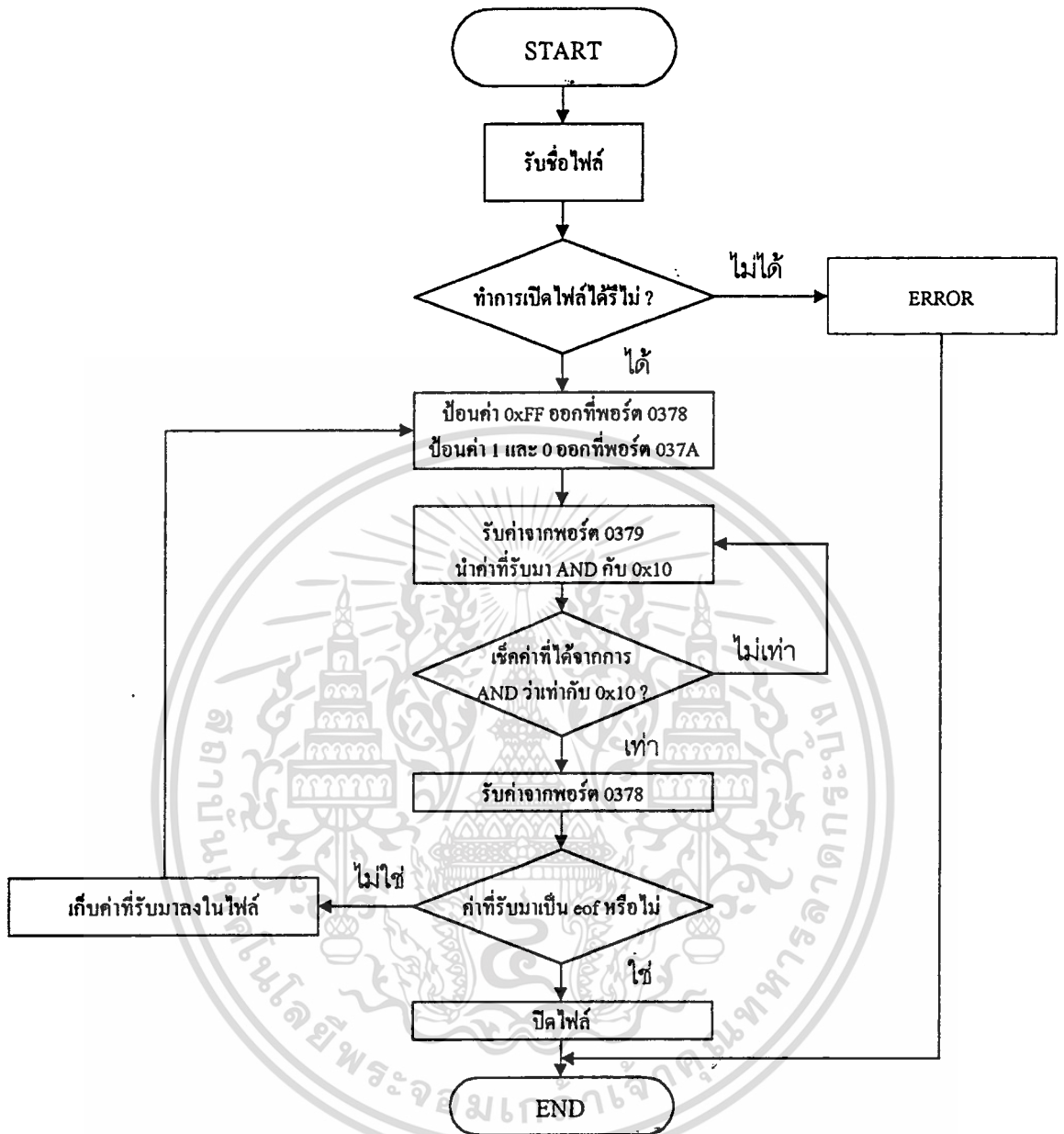
ข้อมูลที่ทำการส่งไปโดยไม่ซ้ำกัน

ข้อมูลที่รับเป็น file ของคอมพิวเตอร์คือ Weerachai Piyakawanit ค่าที่ได้ถูกต้องทุก bit  
FFFFFFFFFFFFFFFF

ข้อมูลที่ทำการส่งไปโดยซ้ำๆกัน

ข้อมูลที่รับเป็น file ของคอมพิวเตอร์คือ FFFFFFFFFFFFFFFFFF ค่าที่ได้ถูกต้องทุก bit

ดังนั้นในขั้นตอนนี้ก็ทำการทดลองซ้ำแล้วซ้ำอีกและสลับกันระหว่างข้อมูลซ้ำกับไม่ซ้ำก็ยังปรากฏว่าผลการทดลองที่ได้ถูกต้องทุกครั้งไม่ผิดเพี้ยนไปเลยในขั้นตอนนี้จึงมาความถูกต้อง 100% ภาพในหน้าถัดไปจะแสดงให้เห็นถึงขั้นตอนการทำงานในกระบวนการรับข้อมูลมากจากบอร์ดแล้วนำไปจัดเก็บเป็นแฟ้มข้อมูล



รูปที่ 4.2 แสดง Flow Chart ของขั้นตอนการรับข้อมูลจากบอร์ด TMS320C5X

เมื่อทำการรับข้อมูลที่ส่งมาจากบอร์ด TMS320C5X เรียบร้อยแล้ว ต่อจากนั้นก็ทำการเปิดไฟล์ของข้อมูลที่รับมาได้เปรียบเทียบกับไฟล์ต้นแบบเพื่อตรวจสอบความสามารถของการประมวลผลที่ผ่านกระบวนการของเวฟเลิตทราานฟอร์มมาเรียบร้อยแล้วว่าถูกต้องเพียงใด ในการทำการเปรียบเทียบความถูกต้องของการรับข้อมูลที่ได้มากับข้อมูลต้นแบบนี้จะทำการนับจำนวนของเลข 1 ที่รับเข้ามาได้เปรียบเทียบกับจำนวนของเลข 1 ของข้อมูลต้นแบบ ในการเปรียบเทียบความถูกต้องของข้อมูลที่ทำการรับเข้ามาได้จะเป็น 2 แบบคือ การนับจำนวนของเลข 1 ของข้อมูลต้นแบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบทั้งหมดว่ามีจำนวนเลข 1 กี่ตัว จากนั้นก็ทำการนับจำนวนของเลข 1 ของข้อมูลที่ได้รับเข้ามาได้ว่า มีจำนวนกี่ตัว แล้วนำผลของจำนวนเลข 1 ที่ได้ของข้อมูลต้นแบบและข้อมูลที่ได้รับได้มาทำการเปรียบเทียบกัน ส่วนแบบที่ 2 จะทำการนับจำนวนความถูกต้องของข้อมูลที่ได้รับเข้ามาได้ว่ามีความถูกต้องกี่บิต โดยจะแบ่งเป็นถูก 4 บิต ถูก 3 บิต ถูก 2 บิต ถูก 1 บิต และไม่ถูกเลย ซึ่งขั้นตอนในการนับมีรายละเอียดดังนี้

#### 4.1.3 โปรแกรมการนับจำนวนของเลข 1 ที่รับเข้ามาเทียบกับข้อมูลที่ทำการส่งไป

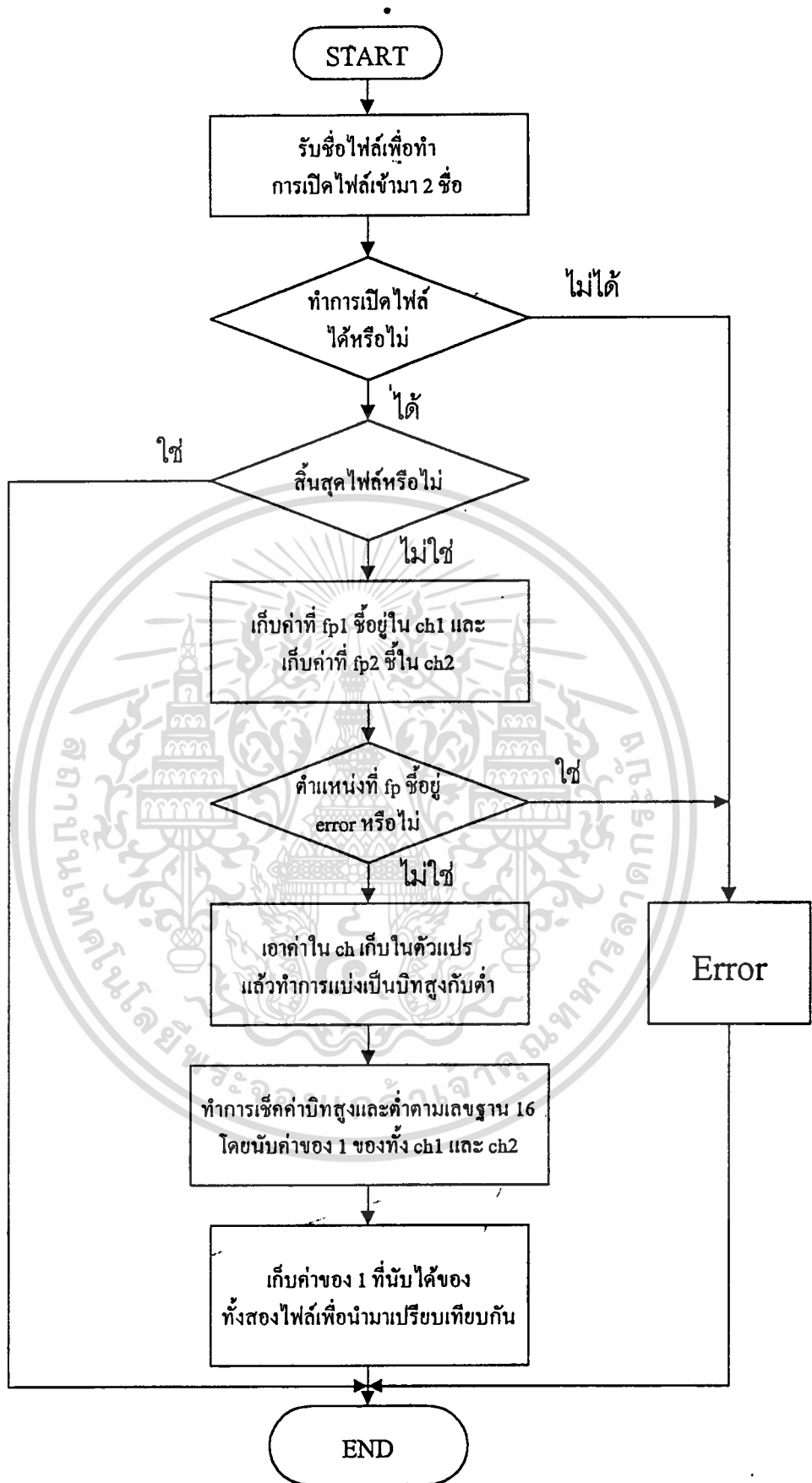
โปรแกรมที่ใช้ในการนับจำนวนของเลข 1 ในการทดลองเพื่อหาจำนวนความถูกต้องของข้อมูลที่ได้รับมาได้เทียบกับจำนวนของเลข 1 ที่อยู่ในข้อมูลที่ทำการส่งไป เพื่อที่ได้สามารถหาเปอร์เซ็นต์ของความผิดพลาดได้ ในทางกลับกันก็จะเป็นการหาเปอร์เซ็นต์ของความถูกต้องว่ามีความถูกต้องมากน้อยเพียงใด

ในการทดลองโครงการนี้ได้ทำการเขียนโปรแกรมเพื่อทำการนับจำนวนของเลข 1 โดยทำการนับเป็น 2 วิธี คือ จะทำการนับจำนวนเลข 1 ทั้งจากข้อมูลต้นแบบที่ทำการส่งและข้อมูลที่ได้รับทั้งหมด แล้วนำมาเปรียบเทียบกัน อีกวิธีหนึ่งคือ ทำการเช็คค่าของข้อมูลโดยการเปรียบเทียบหาความถูกต้องในแต่ละบิตของข้อมูลที่ทำการแบ่งเป็นชุดๆละ 4 บิต ต่อไปก็จะกล่าวถึงหลักการคิดของแต่ละแบบโดยละเอียดต่อไป

วิธีแรกเป็นโปรแกรมที่จะทำการนับจำนวนของเลข 1 ของข้อมูลต้นแบบที่ทำการส่งและข้อมูลที่ได้รับได้ทั้งหมดแล้วนำมาเปรียบเทียบกัน การทำงานของโปรแกรมอธิบายได้ตาม flow chart ดังต่อไปนี้

เริ่มต้นจากการรับชื่อไฟล์ของไฟล์ข้อมูลที่เป็นข้อมูลของต้นแบบที่ส่งและข้อมูลที่ได้รับได้ขึ้นมา แล้วทำการเช็คว่าการเปิดไฟล์ทั้งสองได้หรือไม่ ถ้าไม่ได้โปรแกรมจะฟ้องว่า error แต่ถ้าทำการเปิดไฟล์ทั้งสองได้ก็จะทำการเช็คต่อไปว่าตำแหน่งที่ fp ของไฟล์ทั้งสองที่ตั้งอยู่เป็นจุดสิ้นสุดไฟล์หรือไม่ ถ้าใช่ก็จบการทำงานของโปรแกรม แต่ถ้าไม่ใช่ก็จะทำการเก็บค่าที่ fp ของฝั่งส่งข้อมูลและฝั่งรับข้อมูลที่อยู่ไปเก็บไว้ในตัวแปร ch1 และ ch2 ตามลำดับ หลังจากนั้นก็ทำการเช็คต่อไปว่าตำแหน่งที่ fp ของฝั่งส่งข้อมูลและฝั่งรับข้อมูลนั้น error หรือไม่ ถ้า error ก็จะมีฟ้องว่า error แล้วก็จบโปรแกรม แต่ถ้าไม่ก็จะนำค่าที่อยู่ในตัวแปร ch1 และ ch2 ไปทำการแยกเป็นสองส่วน โดยแบ่งเป็นบิตสูงและบิตต่ำโดยวิธีการเลื่อนบิตของข้อมูล จากนั้นก็ทำการเทียบค่ากับค่าในเลขฐาน 16 ก็จะสามารถรู้ได้ว่าค่านั้นประกอบไปด้วยเลข 1 จำนวนกี่ตัวก็ทำการนับไปเรื่อยๆ โดยแบ่งเป็นฝั่งส่งข้อมูลและรับข้อมูล จนกว่าโปรแกรมจะเช็คได้ว่าถึงจุดสิ้นสุดของไฟล์แล้วก็เป็นจบการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง Flow Chart ของขั้นตอนการทำงานในการเช็คจำนวนบิตที่เป็นหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนโปรแกรมอีกวิธีหนึ่งเป็นการนับจำนวนความถูกต้องของข้อมูลที่ได้รับมาได้เทียบกับข้อมูลต้นแบบที่ส่ง โดยจะทำการเช็คความถูกต้อง 4 บิทก็ครั้ง 3 บิทก็ครั้ง 2 บิทก็ครั้ง 1 บิทก็ครั้งและไม่ถูกเลยก็ครั้ง ต่อไปก็จะขอกกล่าวโดยละเอียดเพื่อเป็นการอธิบายถึงหลักการและขั้นตอนในการทำงานของโปรแกรม

การทำงานของโปรแกรมเริ่มต้นด้วยการรับชื่อไฟล์ของข้อมูลทางฝั่งส่งข้อมูลและรับข้อมูลเข้ามาประมวลผลก่อนว่าสามารถที่จะทำการเปิดไฟล์ทั้งสองได้หรือไม่ ถ้าไม่สามารถที่จะทำการเปิดไฟล์ได้ไฟล์หนึ่งในสองไฟล์นั้นได้ก็เป็นอันว่าเกิดข้อผิดพลาดขึ้นในการป้อนชื่อไฟล์ โปรแกรมก็จะทำการฟ้องว่า error in open file ถ้าทำการป้อนชื่อไฟล์ถูกต้องทั้งสองไฟล์และสามารถทำการเปิดไฟล์ได้โปรแกรมก็จะทำงานต่อไป โปรแกรมนี้จะทำการเปิดไฟล์ขึ้นมาพร้อมกัน 2 ไฟล์โดยที่ไฟล์ที่หนึ่งจะใช้ fp1 เป็นตัวชี้ตำแหน่งของข้อมูล และไฟล์ที่สองจะใช้ fp2 เป็นตัวชี้ตำแหน่งของข้อมูลในการทำงานของโปรแกรม 1 รอบจะทำการรับข้อมูล 2 ครั้ง จึงได้กำหนดตัวแปร x ขึ้นมาโดยที่ในตอนเริ่มแรกจะกำหนดให้ x มีค่าเท่ากับศูนย์และในโปรแกรมจะทำการเช็คค่าของ x ทุกครั้งเพื่อที่จะได้สามารถทราบถึงตำแหน่งของการรับข้อมูลว่าเป็นการรับข้อมูลเข้ามาที่ตัวแรกหรือตัวที่สองเพื่อความสะดวกในการนำค่าของข้อมูลไปใช้ในการประมวลผลต่อไป

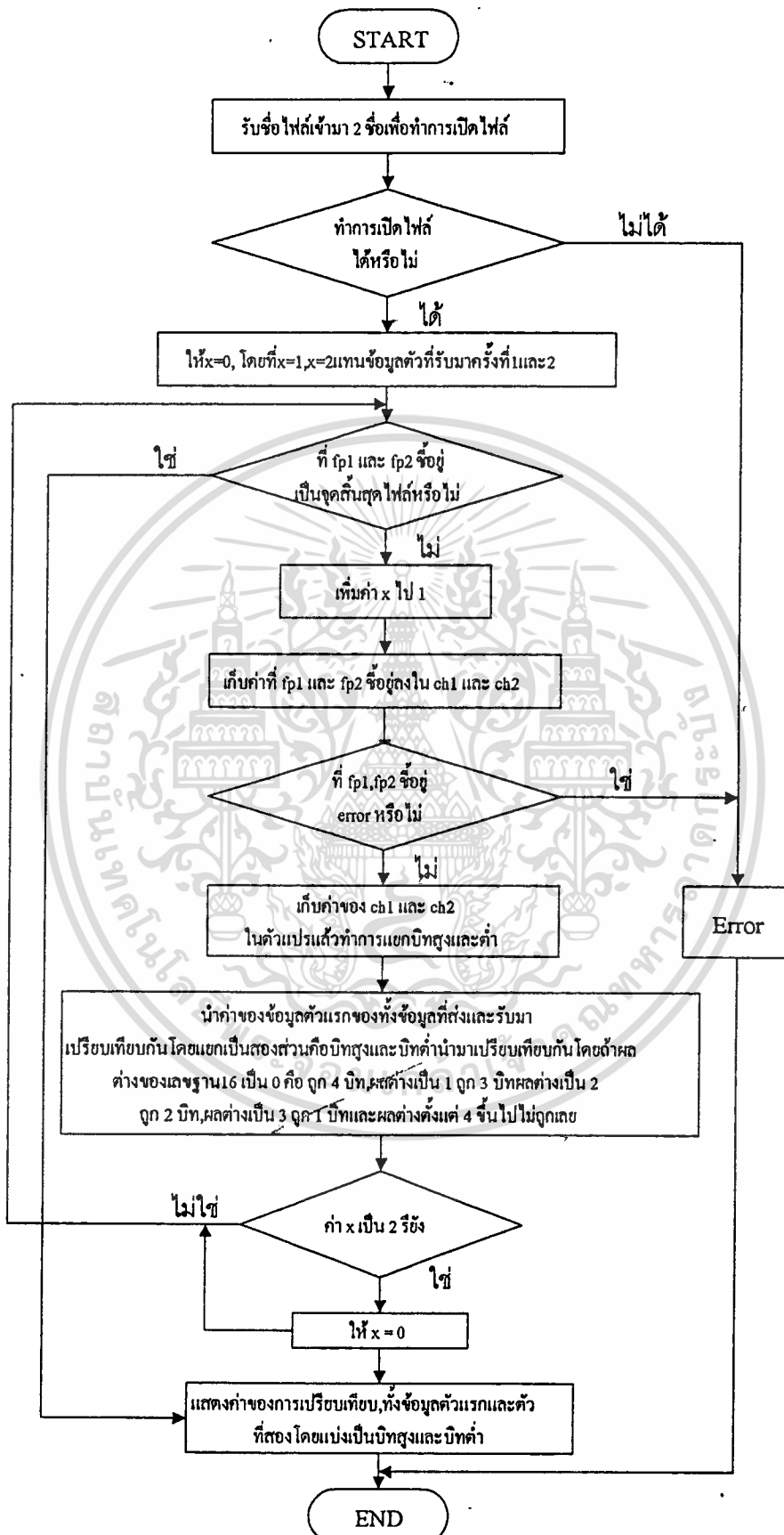
หลังจากการกำหนดค่าของตัวแปร x ให้มีค่าเท่ากับศูนย์แล้ว ต่อไปก็จะทำการเช็คว่าในขณะที่ตำแหน่งที่ fp1 และ fp2 ชี้อยู่มีค่าเป็น eof ของไฟล์หรือไม่ ถ้าใช่ก็เป็นอันว่าจบการทำงานของโปรแกรมในลูปนี้ก็จะไปทำงานในลูปต่อไป ถ้าค่าที่ fp1 และ fp2 ชี้อยู่ยังไม่มีค่าเป็น eof ของไฟล์โปรแกรมก็จะยังคงทำงานในลูปนี้ต่อไป จากนั้นก็จะนำค่าของข้อมูลที่ fp1 และ fp2 ชี้อยู่ไปเก็บไว้ในตัวแปร ch1 และ ch2 ต่อไปตามลำดับ เสร็จแล้วก็จะทำการเช็คค่าที่ตำแหน่งที่ fp1 และ fp2 ชี้อยู่ว่า error หรือเปล่า ถ้าเกิดการ error ขึ้น โปรแกรมก็จะฟ้องว่าเกิดไฟล์ error ขึ้นก็จะออกจากการทำงานของโปรแกรมไป ถ้าไม่ error ก็จะทำการเพิ่มค่าให้ตัวแปร x จากเดิมที่เป็นศูนย์ก็จะกลายเป็น 1 หลังจากนั้นก็จะทำการแยกข้อมูลตัวแรกของฝั่งส่งข้อมูลและฝั่งรับข้อมูล โดยการ AND ข้อมูลที่รับเข้ามาได้ด้วย 0XF0 แล้วเลื่อนไปทางขวา 4 บิท และ AND ข้อมูลตัวเดิมที่รับมาด้วย 0X0F อีกครั้งหนึ่งก็เป็นอันว่าสามารถทำการแยกข้อมูลออกจากกันได้แล้วโดยที่ในตอนนี้แต่ละตัวเสมือนว่ามีค่าให้คิดเพียง 4 บิท ในการกระทำการแยกข้อมูลและทำการเลื่อนบิทนี้จะทำทั้งข้อมูลของฝั่งส่งและรับไปพร้อมๆ กันเพื่อนำไปเปรียบเทียบกัน หลังจากนั้นก็จะนำเอาตัวของบิทที่มีค่าสูงไปทำการเปรียบเทียบกับบิทตัวที่มีค่าสูงและทำการเปรียบเทียบบิทที่มีค่าต่ำกับบิทที่มีค่าต่ำ ในการเปรียบเทียบจะทำการเปรียบเทียบก่อนว่าค่าที่นำมาเปรียบเทียบกันนั้น(ค่า 4 บิท) ค่าไหนมีค่าสูงกว่ากันเมื่อรู้แล้วก็เอาค่าที่สูงเป็นตัวตั้งหลังจากนั้นก็ทำการลบกันถ้าผลต่างที่ได้จากการลบบมีค่าเป็นศูนย์แสดงว่าค่าที่เปรียบเทียบกันอยู่นั้นถูกทั้ง 4 บิท, ถ้าผลต่างของการลบบมีค่าเป็นหนึ่งแสดงว่าถูก 3

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต, ถ้าผลต่างของการลบมีค่าเป็นสองแสดงว่าถูก 2 บิต, ถ้าผลต่างของการลบมีค่าเป็นสามแสดงว่าถูก 1 บิตและถ้าผลต่างของการลบมีค่าตั้งแต่สี่ขึ้นไปแสดงว่าไม่มีบิตที่ถูกเลย ในการรับค่าครั้งแรกนี้สามารถทำการเช็คข้อมูลของตัวที่รับเข้ามาตัวแรกได้ทั้งบิตสูงและบิตต่ำเป็นที่เรียบร้อยแล้ว หลังจากนั้นก็จะทำการเช็คค่าของ  $x$  ว่าเป็นสองร้อยถ้ายังก็จะไปทำการรับค่าของข้อมูลตัวถัดไปซึ่งเป็นตัวที่สองของการรับ

โปรแกรมก็จะทำการเช็คค่าต่างๆ ที่กล่าวมาแล้วอีกครั้งในครั้งนี้อ่า  $x$  มีค่าเป็นสองก็คือว่าเป็นข้อมูลตัวที่สอง ต่อไปก็จะไปทำตามกระบวนการต่อไปโดยการแยกข้อมูลและเลื่อนบิตเพื่อหาค่าความถูกต้องของการรับและส่งข้อมูลของข้อมูลตัวที่สอง โปรแกรมก็จะทำงานต่อไปจนกระทั่งเช็คเจอตัว eof ของไฟล์โปรแกรมก็จะหยุดการทำงานในลูปลงแล้วจะไปทำการแสดงค่าของจำนวนความถูกต้องของแต่ละบิตต่อไป ภาพในหน้าถัดไปจะแสดงถึง Flow Chart ของการทำงานในการเช็ค bit ในขั้นตอนนี้





รูปที่ 4.4 แสดง Flow Chart ขั้นตอนการทำงานเพื่อนับเทียบจำนวน bit ที่คลาดเคลื่อน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับบริการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่ออนุญาตให้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองที่ได้ในขั้นตอนนี้คือการที่เราสร้างไฟล์ไว้เพื่อเป็นfile อ้างอิงจากนั้นก็ทำการนับ 1 เก็บไว้ว่าเพิ่มต้นแบบมีค่าเท่าไร จากนั้นก็นำข้อมูลที่รับได้มาทำการนับ ผลปรากฏว่าทำการนับความแตกต่างในเมื่อส่ง 0 มา 100,000 ตัวจะมีค่า 1 เกิดขึ้นมาประมาณ 1199 bits

#### 4.1.4 ฮาร์ดแวร์ในส่วนของการติดต่อระหว่างเครื่องคอมพิวเตอร์และบอร์ด TMS320C5X

ในการติดต่อระหว่างเครื่องคอมพิวเตอร์และบอร์ด TMS320C5X จะแบ่งเป็น 2 แบบคือ แบบแรกคือ ช่วงที่เครื่องคอมพิวเตอร์ส่งข้อมูลให้กับบอร์ด TMS320C5X และแบบที่สองคือ ช่วงที่เครื่องคอมพิวเตอร์รับข้อมูลจากบอร์ด TMS320C5X ซึ่งต่อไปนี้จะได้กล่าวถึงการทำงานของฮาร์ดแวร์เพื่อที่จะทำให้สามารถเข้าใจถึงระบบการรับและส่งข้อมูลได้อย่างดียิ่งขึ้น

##### ช่วงที่เครื่องคอมพิวเตอร์ส่งข้อมูลให้กับบอร์ด TMS320C5X

เมื่อเครื่องคอมพิวเตอร์ต้องการที่จะส่งข้อมูลให้กับบอร์ด TMS320C5X เครื่องคอมพิวเตอร์จะทำการส่งข้อมูลออกไปที่พอร์ต 0378 ข้อมูลที่ทำการส่งไปจะเข้าไปอยู่ในไอซี 74F245 จากนั้นเครื่องคอมพิวเตอร์จะส่งสัญญาณสโตปออกไป ซึ่งในการส่งจะส่ง 0 กับ 1 ไปเพื่อใช้สัญญาณสโตปไปทำเป็น CLOCK ให้กับ D FLIP-FLOP จากนั้นสัญญาณที่ออกที่ขา  $\bar{Q}$  จะมีค่าเป็นศูนย์ซึ่งจะทำการส่งสัญญาณไปบอกบอร์ด TMS320C5X ว่าขอทำการอินเตอร์รัพเพื่อที่จะส่งข้อมูลให้บอร์ด TMS320C5X อ่านข้อมูลเข้าไป เมื่อบอร์ด TMS320C5X รับสัญญาณอินเตอร์รัพเข้าไปแล้วก็จะทำการส่งสัญญาณเพื่อที่ทำการอ่านข้อมูลออกมาที่แอดเดรส  $Y_0$  ของไอซี 74F138 เพื่อไปทำการ ENABLE ไอซี 74F245 เพื่อรับข้อมูลเข้าไป เมื่อบอร์ด TMS320C5X ทำการรับข้อมูลเสร็จแล้วก็จะส่งสัญญาณออกมาที่แอดเดรส  $Y_3$  ของไอซี 74F138 เพื่อไปทำการเคลียร์ D FLIP-FLOP เพื่อที่จะบอกเครื่องคอมพิวเตอร์รับข้อมูลเสร็จแล้วและก็จะส่งสัญญาณไปบอกเครื่องคอมพิวเตอร์ว่าพร้อมที่จะรับข้อมูลตัวต่อไปแล้ว

##### ช่วงที่เครื่องคอมพิวเตอร์รับข้อมูลจากบอร์ด TMS320C5X

ในการรับข้อมูลจากบอร์ด TMS320C5X เข้าไปในเครื่องคอมพิวเตอร์นั้นก่อนที่จะทำการรับข้อมูลเครื่องคอมพิวเตอร์จะต้องทำการส่งข้อมูล 0xFF ออกมาที่พอร์ต 0378 ก่อนเพื่อให้ศักยภาพที่ขาของไอซี 74F374 มีสถานะเป็น high เพื่อป้องกันการเกิดความผิดพลาดจากกระแส sink และกระแส source ต่อจากนั้นก็ทำการส่ง 1 และ 0 ออกที่ขา error เพื่อทำหน้าที่เป็น CLOCK ให้กับ D FLIP-

FLOP จากนั้นขา  $Q$  ก็จะมีสถานะเป็น high ส่วนขา  $\bar{Q}$  ก็จะมีสถานะเป็น low เพื่อส่งไปเป็น สัญญาณอินเตอร์รัพท์เพื่อไปบอกบอร์ด TMS320C5X ว่าพร้อมที่จะรับข้อมูลจากบอร์ด TMS320C5X แล้ว หลังจากนั้นบอร์ด TMS320C5X ก็จะส่งสัญญาณเพื่อเขียนข้อมูลออกมาที่แอดเดรส  $Y_6$  ของไอซี 74F138 เพื่อไปทำการเคลียร์ D FLIP-FLOP ในขณะที่เดียวกันข้อมูลจากบอร์ด TMS320C5X ก็จะถูกส่งไปยังไอซี 74F374 ในขณะที่สัญญาณที่ขาของ  $Q$  จะเป็น low สัญญาณนี้ก็จะไปทำการทริกขา output select ของไอซี 74F374 และในขณะเดียวกันสัญญาณที่ขา  $\bar{Q}$  จะมีค่าเป็น high คอมพิวเตอร์ก็จะทำการรับสัญญาณนี้ที่ขา SLCT ซึ่งสัญญาณนี้จะเป็นตัวไปบอกคอมพิวเตอร์ว่ามีข้อมูลถูกส่งออกมาแล้ว จากนั้นคอมพิวเตอร์ก็จะทำการรับค่าของข้อมูลโดยผ่านพอร์ต 0378

จากกระบวนการทั้งหมดผลการทดลองที่ได้คือสามารถส่งข้อมูลให้และรับข้อมูลจากบอร์ด TMS320C5X เพื่อติดต่อระหว่างเครื่องคอมพิวเตอร์ได้อย่างถูกต้องทุกประการ

## 4.2 การใช้บอร์ด TMS320C5X ประมวลผลสัญญาณ

จากการทำงานในการติดต่อระหว่างบอร์ด DSP TMS320C5X โดยผ่านทางสายโทรศัพท์ สามารถที่จะทำการทดลองโดยผลการทดลองที่ได้มีดังนี้

### 4.2.1 ผลการทดลองทางด้านส่งข้อมูล

ทางด้าน การส่งข้อมูลเราสามารถแยกพิจารณาในรายละเอียดได้เป็น 3 ส่วนคือ

- การนำ Wavelet function ที่เป็นเลขทศนิยมให้อยู่ในรูปแบบของเลขฐาน 16 หรือ Q Number
- การนำข้อมูลและค่าที่ได้จากการแปลงมาทำการคำนวณตามทฤษฎี Inverse Wavelet Transform

การนำ Wavelet function ที่เป็นเลขทศนิยมให้อยู่ในรูปแบบของเลขฐาน 16 หรือ Q Number Coefficient ที่นำมาใช้ในการทดลองนี้ได้กล่าวไปแล้วในบทที่ 3 ว่าเราจะใช้ Coefficient ของ Wavelet คือ Daubechies's 4 ซึ่งมีค่าดังนี้

$$H = [ \quad -0.129 \quad 0.2241 \quad 0.8365 \quad 0.4830 \quad ]$$

$$G = [ \quad -0.4830 \quad 0.8365 \quad -0.2241 \quad -0.1294 \quad ]$$

$$H^* = [ \quad 0.4830 \quad 0.8365 \quad 0.2241 \quad -0.1294 \quad ]$$

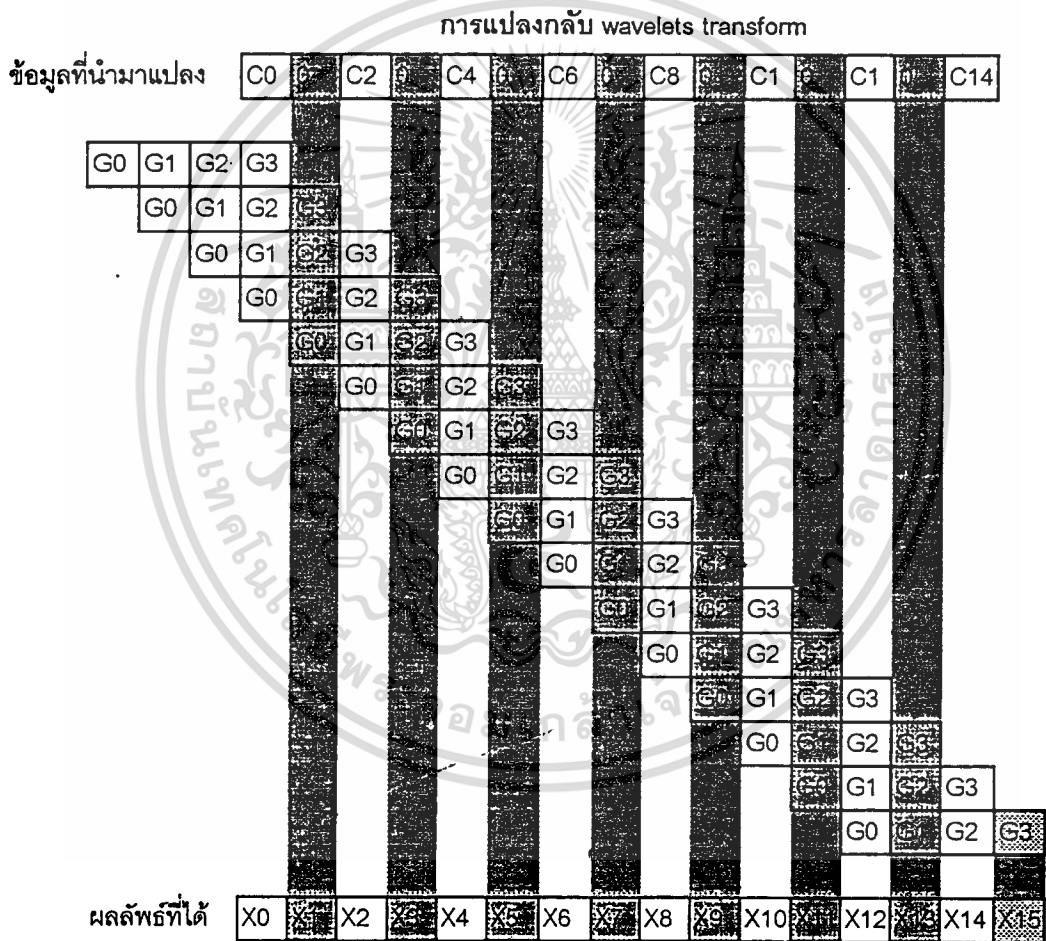
$$G^* = [ \quad -0.1294 \quad -0.2241 \quad 0.8365 \quad -0.4830 \quad ]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### การนำข้อมูลและค่าที่ได้จากการแปลงมาทำการคำนวณตามทฤษฎี Inverse Wavelet Transform

ขั้นตอนนี้เป็นกระบวนการที่ยากที่สุดในการทำการประมวลผลทางด้าน การส่ง เนื่องจากต้องเข้าใจทฤษฎีของ Inverse Wavelet Transform เป็นอย่างดีและต้องเข้าใจกระบวนการทำงานของชุดคำสั่งในการควบคุมบนบอร์ด TMS320C5X ให้เป็นอย่างดีและนำมาประยุกต์ใช้ให้ได้ตรงตามที่ได้ทำการทดลองไว้ใน MathLab ซึ่งค่าที่ได้จะนำมาเปรียบเทียบทุกขั้นตอนและหาสาเหตุของการเกิดปัญหาและแก้ไขปัญหานั้น ดังนั้นก่อนอื่นต้องมาทำความเข้าใจกับทฤษฎีของ Inverse Wavelet Transform ก่อนเพื่อจะสามารถเข้าใจกระบวนการทั้งหมด สามารถแสดงได้ดังนี้



จากสมการด้านบนทำให้เห็นว่าการคูณที่เปลี่ยนแปลงอยู่ดังนั้นหากเราทำการจัดรูปใหม่เราจะได้สมการที่เหมาะสมดังนี้

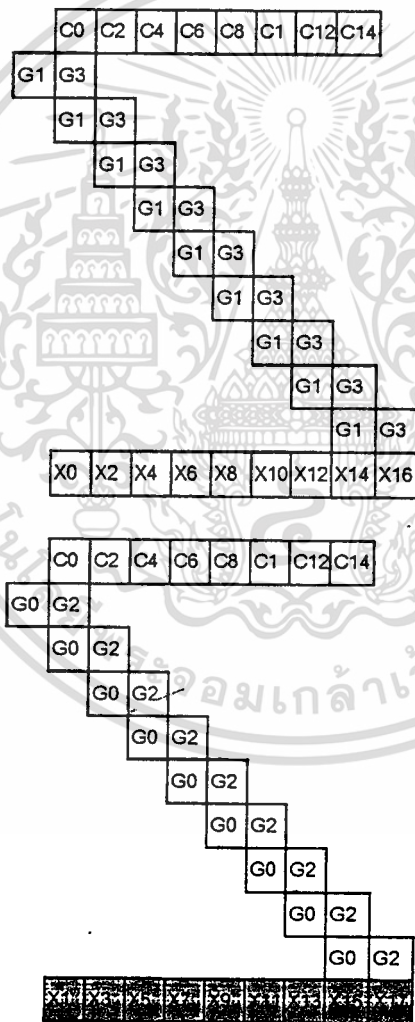
$$X_0 = C_0 \cdot G_3$$

$$X_1 = C_0 \cdot G_2$$

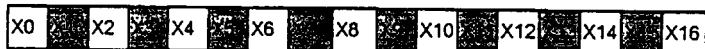
$$X_2 = C_0 \cdot G_1 + C_2 \cdot G_3$$

$$X_3 = C_0 \cdot G_0 + C_2 \cdot G_2$$

ดังนั้นขั้นตอนการทำ Convolution แบบใหม่ที่เราจะนำไปใช้ก็คือ



ผลลัพธ์ที่ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการทำงานใน TMS320C5X นี้เพื่อที่จะทำให้ได้ตามกระบวนการเรามีขั้นตอนการทำงานดังนี้

1. ทำการแยก Wavelet function ออกเป็นข้อมูลตำแหน่งคู่กับตำแหน่งคี่
2. Convolution ของ Wavelet coefficient กับ Wavelet function ที่แยกออกเป็นสองส่วนนี้
3. ทำการเรียงสลับค่าที่ได้จากการ Convolution ทั้งสอง

ขบวนการดังกล่าวข้างต้นนี้จะมีข้อดีคือ

1. จำนวนครั้งในการคูณเลขน้อยกว่าขบวนการตามปกติ ครั้งหนึ่ง
2. การแยกค่าของ Wavelet function เป็นตำแหน่งคู่และคี่ซึ่งเป็นวิธีการเดียวกันกับการทำ Up Sampling ดังนั้นจึงเปรียบเทียบได้ว่า วิธีการใหม่นี้จะมีการทำ Up Sampling และคำนวณค่าภายในครั้งเดียว
3. การที่สัญญาณถูกแยกและการ Convolution ที่แยกออกเป็นสองส่วนจะทำให้การพัฒนาให้ขบวนการนี้เป็นการทำงานแบบ Parallel processing ได้โดยง่ายในกรณีที่มีหน่วยประมวลผลหลายตัว

เมื่อเรานำกระบวนการทั้งหมดนี้ไปประมวลผลในบอร์ด TMS320C5X เรายังต้องมีขบวนการที่ต้องคำนึงถึงอีกอย่างหนึ่งก็คือการเลื่อนจุดทศนิยม ในการส่งข้อมูลจาก Computer เพื่อนำไปประมวลผลที่ TMS320C5X ข้อมูลที่ถูกส่งออกไปจะมีขนาด 8 bits เราจะนำค่าข้อมูลที่ได้รับมาจาก Computer นี้มาแยกเป็น 4 bits 2 ชุดแต่ละชุดคือ Coefficient ที่เราจะนำไปประมวลผลกับ Wavelet function ดังนั้น Coefficient จะมีค่าสูงสุดคือ  $F_{16}$  หรือ  $15_{10}$  จากในตอนต้นที่เรากำหนดให้มีการแปลง Wavelet function ให้อยู่ในรูป Q10 สาเหตุก็เนื่องมาจากการที่เรากำหนดค่า Coefficient สูงสุดของเราคือค่า 15 นั่นเองเพราะเมื่อเราทำการคูณค่า Coefficient กับ Wavelet function ที่แปลงเป็น Q Number สูงกว่า Q10 เช่น Q11 เป็นต้นไปจะทำให้ผลที่ได้จากการคูณในบอร์ด TMS320C5X นี้ผิดไปเนื่องจากจะทำให้ค่าที่ควรจะเป็นบวกกลายเป็นลบเพื่อความเข้าใจจะแสดงดังตัวอย่างด้านล่าง

Convolution ครั้งแรก  $(0.8365 * 15) + (-0.1294 * 15) = 10.6065_{10}$  เราจะเห็นว่าค่าที่ได้ ออกมาเป็นค่า 10.6065 หากเราใช้ Q11 ค่าที่ได้ควรจะมีค่า  $(A5EF)_{16}$  แต่เราลองมาดูว่า TMS จะคำนวณได้ค่าเท่าไร

$$0.8365_{10} = 0680_{16}, 15 = F_{16}, -0.1294_{10} = FEF8_{16}$$

$$\text{Convolution ครั้งแรก } (0680 * F)_{16} + (FEF8 * F)_{16} = 15208_{16} \neq A5EF_{16}$$

ซึ่งจะรู้ได้ทันทีเลยว่าค่าที่ได้จากการแปลงเป็น Q11 ค่าที่ได้ก็มีการผิดพลาดเกิดขึ้นตั้งแต่ในกระบวนการแรกแล้วดังนั้นเราจึงใช้การแปลงที่ Q10 และเมื่อเรานำค่า Q10 กับ Q10 มาคูณกันทำให้เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

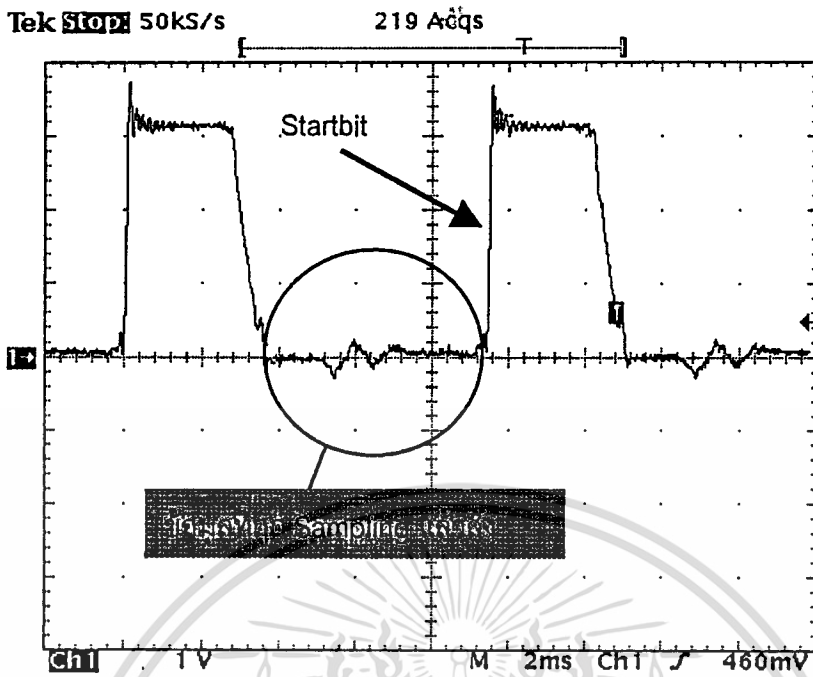
ทราบว่าเราต้องเพิ่มจุดทศนิยมจากเดิมไปอีก 10 ค่าแต่ในทางปฏิบัติเราจะใช้การเลื่อน bit กลับมาอีก 10 bit แทนเพื่อความเข้าใจจะแสดงตัวอย่างให้เห็นดังนี้

ในฐานสิบหากเรามีค่า  $0.5 * 0.25$  ในการคูณค่าที่ได้คือ 0.125 เมื่อเทียบกับค่าปกติที่ไม่มีจุดทศนิยมคือ  $5 * 25 = 125$  จะแตกต่างจากกันเพียงเราเพิ่มจุดทศนิยมเข้าไปอีก 3 จุดเท่านั้นตามจำนวนของตัวตั้งกับตัวคูณ

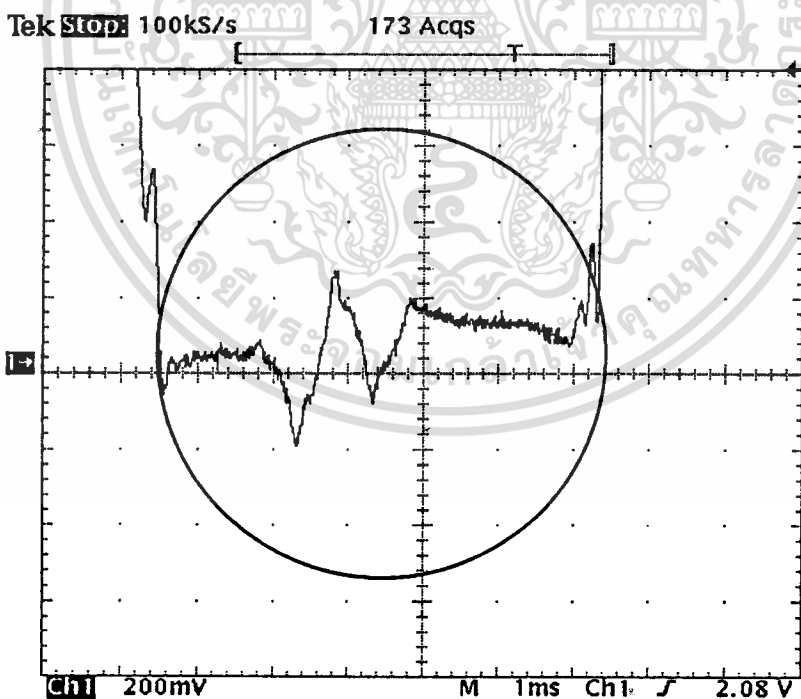
ในฐานสิบหกหากเราใช้ค่าเดียวกันคือ  $0.5 * 0.25$  จะได้เป็น  $(0200 * 0100)_{16}$  คำตอบคือ  $2000_{16}$  แทนที่เราจะเพิ่มจุดทศนิยมเหมือนกับในฐานสิบเราใช้การเลื่อนข้อมูลกลับมาเข้าหา LSB 10 ครั้งค่าที่เราได้สุดท้ายคือ  $0080_{16}$  ทำเป็นฐานสิบก็คือค่า 0.125 เท่ากัน

ดังนั้นเมื่อเราทำการคูณในกระบวนการ Inverse Wavelet Transform หลายๆขั้นแน่นอนว่าเราจะต้องนำค่าที่ถูกแปลงเป็น Q10 สองค่ามาคูณกันอย่างแน่นอนดังนั้นเราจึงจำเป็นต้องทำการเลื่อน bit กลับเพื่อที่จะทำให้กระบวนการประมวลผลนั้นถูกต้อง และหากเราทำตามเงื่อนไขที่กำหนดให้ได้ความถูกต้องทั้งหมดแล้วภาพในหน้าถัดไปจะแสดงกระบวนการทำการสร้างข้อมูลสัญญาณโดยวิธี Inverse Wavelet Transform ทั้งหมด

จากการเขียนโปรแกรม Debugger เพื่อให้บอร์ด TMS320C5X ทำงานได้ตามกระบวนการแล้วเราก็จะทำการส่งข้อมูลออกไปผ่านสายโทรศัพท์ไปยังปลายทางแต่หลักการในการคำนวณกลับจำเป็นต้องทำการ Sampling สัญญาณกลับมาเช่นเดียวกันเนื่องจากเราไม่ทราบว่าเมื่อไหร่จะส่งข้อมูลทำให้ไม่รู้ว่าเมื่อไหร่ฝ่ายรับจึงนำข้อมูลที่ได้มาประมวลผลดังนั้นในทางปฏิบัติจึงจำเป็นต้องทำการส่ง Start Bit ไปด้วยเพื่อเป็นการกำหนดว่ามีข้อมูลเริ่มเข้ามาแล้วดังนั้นในการส่งข้อมูลเราจึงจำเป็นต้องเสียเวลาส่วนหนึ่งให้กับการส่งสัญญาณ Star bit ผลการทดลองที่ได้จากการทำการแก้ไขจนคิดว่าผลได้ตามที่ต้องการแล้วแสดงในภาพที่อยู่ในหน้าถัดไป



รูปที่ 4.5 แสดงภาพสัญญาณโดยรวมที่ส่งออกมาจาก TMS320C5X โดยใช้ข้อมูลคือ  $41_{16}$



รูปที่ 4.6 แสดงรายละเอียดสัญญาณที่ถูกส่งออกมาจาก TMS320C5X โดยมีข้อมูลคือ  $41_{16}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.2 ผลการทดลองทางด้านรับ

ทางด้านรับก็สามารถทำการแยกพิจารณาเป็นส่วนๆ ได้เช่นเดียวกับกับทางด้านส่งแต่ทางด้านรับจะมีความยุ่งยากมากกว่าและต้องทำการทดลองหลายอย่างเพื่อที่จะทำการตรวจสอบจนได้ความเหมาะสม

- การตรวจสอบสัญญาณเพื่อหา Start bit แล้วเริ่มรับข้อมูลเข้ามาเก็บ
- การแปลงสัญญาณที่ได้นั้นโดยวิธีการ Wavelet Transform
- ทำการปิดทศนิยมจากค่าที่คำนวณได้

**การตรวจสอบสัญญาณเพื่อหา Start bit แล้วเริ่มรับข้อมูลมาเก็บเพื่อนำไปประมวลผล**

ในขั้นตอนนี้เกี่ยวกับการ Sampling สัญญาณทางด้านภาครับดังนั้นสิ่งที่ทำการทดลองในขั้นตอนนี้คือการปรับแต่งความถี่ Sampling ของบอร์ด TMSC320C5X แล้วทำการรับข้อมูลเก็บไว้ใน memory ทำการตรวจสอบข้อมูลใน memory ว่ามีค่าเป็นอย่างไรทำอย่างไรจึงมีการรับข้อมูลได้เที่ยงตรงที่สุดผลการทดลองตอนนี้คือเราต้องใช้ความถี่ Sampling สูงกว่า ทางภาคส่ง

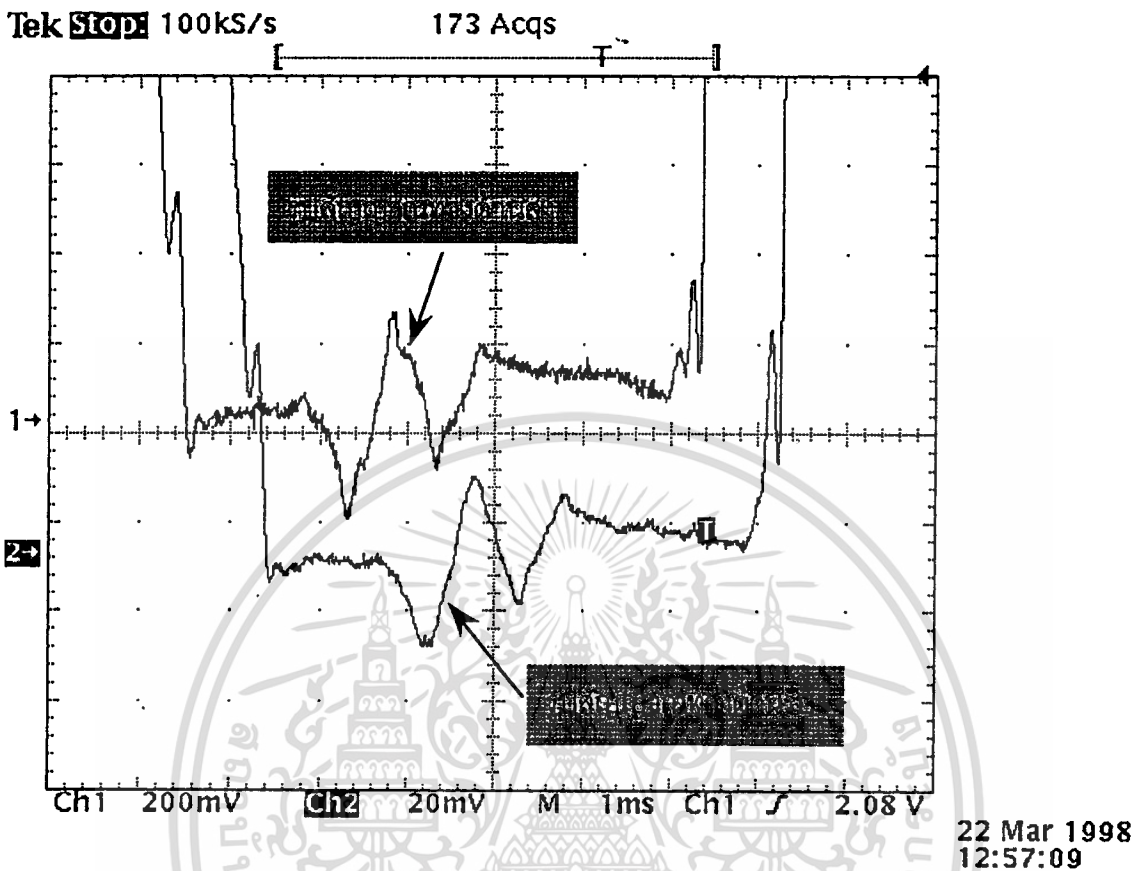
ภาคส่งใช้ความถี่ Sampling  $\approx 8228$  Hz

ภาครับใช้ความถี่ Sampling  $\approx 8361$  Hz

จากการใช้ความถี่ Sampling ทางภาครับสูงกว่าทางภาคส่งนี้เองเราจะได้ค่าที่ตรงกับค่าที่ทำการส่งมาทางด้านส่งมากที่สุดโดยสังเกตข้อมูลที่เป็นค่าสูงสุดกับค่าต่ำสุดว่ามีตำแหน่งที่ตรงกันหรือไม่จากการทดลองเปลี่ยนความถี่ทั้งทางด้านส่งและทางด้านรับปรากฏว่าค่าที่ได้ดีที่สุดคือค่านี้

แต่ผลของการ Sampling ทางภาครับที่มีความถี่สูงว่านี้ก็ยังคงมีคือการทำที่เราเก็บข้อมูลได้ถูกต้องในช่วงเริ่มต้นของข้อมูลแต่ในช่วงท้ายๆ จะทำให้ข้อมูลไม่ตรงและจะทำการประมวลผลเสร็จก่อนที่รูปคลื่นของสัญญาณจริงจะถูกส่งมาหมดทำให้ค่า Coefficient ที่ได้จากการประมวลผลในตอนแรกนั้นมีความคลาดเคลื่อนมากส่วนการหาค่า Coefficient ที่อยู่ในสุดจะสามารถได้ค่าที่ถูกต้องมากกว่า

ผลการทดลองในขั้นตอนนี้จะขอแสดงให้เห็นถึงการรับข้อมูลทางด้านรับโดยใช้ภาพสัญญาณซึ่งอยู่ในหน้าถัดไป



รูปที่ 4.7 ภาพแสดงสัญญาณทางด้านส่งและรับของ TMS320C5X

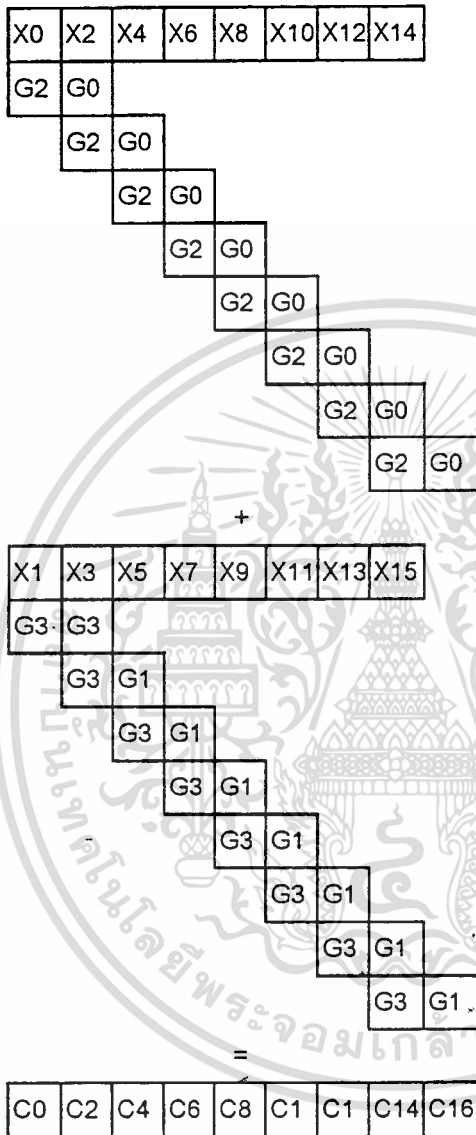
จากภาพแสดงให้เห็นถึงว่าเมื่อส่งข้อมูลมาและทำการรับจะมีการเลื่อนเฟสของข้อมูลอย่างเห็นได้ชัดและทางด้าน Amplitude ก็มีขนาดลดลงทั้งนี้เพราะในขั้นตอนที่เราส่งข้อมูลจะไม่มีการ repeat สัญญาณเลยมีเพียงการลดทอนสัญญาณลงอย่างเดียว ค่าที่ได้จากการ Sampling จึงมีค่าผิดไปจากเดิมด้วยเป็นผลทำให้การคำนวณมีความผิดพลาดอย่างแน่นอน

#### การแปลงสัญญาณที่ได้นั้นโดยวิธีการ Wavelet Transform

จากหลักการเดียวกับทางด้านส่งในขั้นตอนการแปลงเพื่อหา Coefficient ที่ส่งมาคือมีการกระทำพื้นฐานอยู่ 2 แบบคือการทำ Convolution และการทำ Down Sampling เช่นเดียวกันกับการส่งเราจะทำการลดการประมวลเพื่อป้องกันการซ้ำซ้อนที่จะเกิดขึ้น นั่นคือเราจะทำกระบวนการ Convolution และการทำ Down Sampling ไปในคราวเดียวกัน



ดังนั้นขั้นตอนการทำ Convolution แบบใหม่ที่เราจะนำไปใช้ก็คือ



ซึ่งสามารถสรุปเป็นขั้นตอนได้ดังนี้

1. ทำการแยกสัญญาณที่รับเข้ามาและ Wavelet function ออกเป็นข้อมูลตำแหน่งคู่ กับตำแหน่งคี่
2. ทำการ Convolution โดยให้ข้อมูลตำแหน่งคู่กระทำกับตำแหน่งคู่และตำแหน่งคี่ก็เช่นเดียวกัน
3. รวมผลลัพธ์ที่ได้จากการ Convolution โดยเริ่มให้ค่าแรกกับค่าแรกรวมกันจนถึงค่าสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ Wavelet Transform ด้วยวิธีการที่กล่าวนี้ก็มีข้อดีเช่นเดียวกับการแปลงที่ได้ อธิบายไว้แล้วในภาคส่งผลการทดลองในขบวนการนี้ทราบได้จากการวัดค่าที่รับได้นั้นคือวัดค่า Bit error rate โดยทำการส่งข้อมูล  $00_{16}$  จำนวน 100,000 ตัวซึ่งเราจะทำการส่งทีละ 8 bits จากผลการทดลองเราวัดได้ว่าทำการส่ง 1 ครั้งใช้เวลาโดยประมาณ 5.8 msec รวม Start bit เมื่อทำการส่ง 100,000 จึงใช้เวลาทั้งหมด  $12,500 * 5.8 \text{ msec} = 72.5 \text{ sec}$

ส่งครั้งที่หนึ่งผิด	1.254%
ส่งครั้งที่สองผิด	1.396%
ส่งครั้งที่สามผิด	1.292%
ส่งครั้งที่สี่ผิด	1.017%
ส่งครั้งที่ห้าผิด	1.124%
ส่งครั้งที่หกผิด	1.204%
ส่งครั้งที่เจ็ดผิด	1.301%
ส่งครั้งที่แปดผิด	1.006%
ส่งครั้งที่เก้าผิด	1.153%
ส่งครั้งที่สิบผิด	1.247%
รวมเฉลี่ยสิบครั้ง =	1.1994%

ค่าเฉลี่ยของ Bit error rate ที่ได้คือ 16.54 bit/sec ต่อการส่ง 10 ครั้งๆละ 100,000 ตัว

### ทำการปิดทศนิยมจากค่าที่คำนวณได้

การปิดทศนิยมหลังจากที่ได้ข้อมูลแล้วเพื่อทำการหาค่า Coefficient ทางด้านรับให้ตรงกับค่าที่ส่งมา จากที่ได้กล่าวมาแล้วในขั้นต้นว่าการคำนวณจำเป็นต้องมีการแปลงค่าของ Wavelet function เป็นเลข Q Number ดังนั้นเมื่อทำการหาค่ากลับเราจำเป็นต้องหาค่าโดยการเปลี่ยนค่า Q Number ที่ใช้ในที่นี้คือค่า Q10 ให้เป็นเลขธรรมดาโดยก่อนที่เราจะทำกลับคืนนั้นเราต้องทำการเช็ค ว่าที่ ค่าของ Q10 นั้นค่าที่คำนวณได้มีค่าเกิน 0.5 หรือไม่หากค่าที่ได้จากการคำนวณมีค่าเกิน 0.5 จะทำการบวกข้อมูลเพิ่มอีก 1 ค่าทำให้ผลลัพธ์ที่ได้นั้นมีค่าสูงขึ้น 1 ค่าตามที่ได้กำหนดแต่ก็มีความผิดพลาดที่อาจเกิดขึ้นได้ระหว่างค่า  $15_{10}$  หรือ  $F_{15}$  กล่าวคือเมื่อค่าที่ได้มีค่าเกิน  $15.5_{10}$  โดยหลักการที่เราวางไว้ในเบื้องต้นจะทำให้ค่าที่ได้เป็น  $16_{10}$  แต่ค่าของข้อมูลที่เราได้ในฐานสิบหกคือ  $0_{16}$  ทั้งนี้เพราะเราทำการเลื่อน bit กลับมาและเอาค่าเพียง 4 bits ที่อยู่ทางด้านหน้าของทศนิยมที่เรากำหนดเอาไว้ การแก้ไขปัญหานี้ในส่วนนี้ก็ต้องมีโปรแกรมตรวจสอบว่าค่าที่รับมาได้นี้เมื่อรับค่าเป็น  $F_{16}$  แล้วไม่ต้องทำการปิดค่าหรือบวกเพิ่มขึ้นไปอีกหนึ่งหลักให้เก็บค่านั้นได้เลยก็จะทำให้เราสามารถที่จะแยกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่แตกต่างระหว่าง  $0_{10}$  กับ  $15_{10}$  ได้อย่างชัดเจนแต่ในการทดลองจริงค่าที่เราปิดทศนิยมแล้วนี้ก็ยังเกิด error ทั้งนี้ก็เป็นผลต่อเนื่องมาจากการ Sampling ในเบื้องต้นและการคำนวณ Wavelet Transform ในเบื้องต้นทำให้ค่าที่ได้มีความผิดพลาดมาแล้ว 1 หรือ 2 bits ดังนั้นไม่ว่าเราจะทำการปิดทศนิยมให้ถูกต้องเท่าไรก็ยังคงเป็นไปได้ว่าเรายังรับข้อมูลที่ผิดได้อยู่ผลการทดลองที่ได้ในขั้นตอนนี้จะแสดงเป็นภาพโดยใช้การแสดงเป็นเลขฐานสิบหก

0 0 [redacted] 1 1 0 0 0 0 0 0 0 0 0 0 = 3F00(hex)  
 = 15.75(dec)

เมื่อทำการเช็คจะได้ว่าควรปิดทศนิยมเพราะมีค่าเกิน 0.5  
 ดังนั้นค่าที่ได้ใหม่หลังจากปิดคือ

0 1 [redacted] 1 1 0 0 0 0 0 0 0 0 0 0 = 4300(hex)  
 = 16.75(dec)

ทำการเลื่อนข้อมูลกลับ 10 ตำแหน่งเพื่อให้ได้ข้อมูลที่เป็น Q0

0 0 0 0 0 0 0 0 0 0 0 0 1 [redacted] = 0010(hex)  
 = 0(dec)  
 ไม่ทำการพิจารณาเพราะใช้เพียง 4 bits

นอกจากนี้การผิดพลาดยังสามารถเกิดขึ้นได้อีกหนึ่งจุดก็คือในการแสดงค่าของบอร์ด TMS320C5X นี้ค่า  $0_{10}$  สามารถเป็นไปได้ ทั้ง  $0000_{16}$  และ  $FFFF_{16}$  ทำให้เมื่อเราส่งข้อมูลที่เป็นศูนย์มาทั้งหมดค่าที่เราได้อาจกลายเป็นค่า  $15_{10}$  หรือ  $F_{16}$  ก็ได้ดังนั้นเพื่อความถูกต้องเราจึงต้องทำการตรวจสอบ Sign Bit เพื่อที่จะได้รู้ทันทีหากการคำนวณของ TMS320C5X นั้นได้ค่าออกมาเป็น  $0_{10}$  แต่ใช้การแสดงค่าที่ได้ออกมาเป็น  $FFFF_{16}$  เราก็จะสามารถทำการส่งได้ทันทีว่าค่านี้เป็นค่า 0 อย่างแน่นอนขั้นตอนนี้ขอแสดงเป็นเลขฐานสิบหกดังนี้

[redacted] 1 [redacted] 0 0 0 0 0 0 0 0 0 0 0 0 = FC00(hex)  
 = (-1)(dec)

sign flag ซึ่งโดยความเป็นจริงแล้วบอร์ด TMS320C5X ต้องการส่งข้อมูลออกมาเป็นศูนย์ที่ทราบเนื่องจาก Coefficient ที่เราส่งไปนั้นมีค่าเป็นบวกเพียงอย่างเดียวไม่มีค่าล ดังนั้นเราจึงทำการ AND ด้วย 0 ทั้งหมดแล้วก็ส่งไปที่ output ค่าที่ได้คือ

0 0 0 0 0 0 0 0 0 0 0 0 0 [redacted] = 0000(hex)  
 = 0(dec)

## บทที่ 5

### สรุปและวิจารณ์ผลการทดลอง

จากผลการทดลองที่ได้มา ผลที่ออกมาเป็นไปตามที่ตั้งเป้าหมายเอาไว้ สามารถทำการรับและส่งข้อมูลได้ ถึงแม้ว่าผลของการรับข้อมูลที่ทำการส่งมาเมื่อนำมาทำการเปรียบเทียบกับข้อมูลต้นแบบแล้วจะยังคงไม่สามารถที่จะรับข้อมูลให้สามารถถูกต้อง 100% จากผลที่ได้จะขอนำมาแยกสรุปเป็นส่วนต่างดังนี้

#### 5.1 สรุปผลการทดลองทางการส่ง

ผลที่ได้จากด้านส่งข้อมูลมี 2 ส่วนที่เราจะนำมาพิจารณาได้คือ ผลการทดลองของเครื่องคอมพิวเตอร์ที่ทำการส่งข้อมูลให้กับ TMS320C5X โดยผ่านทาง LPT port ซึ่งเป็น port ขนานผลที่ได้ในส่วนนี้เราสามารถส่งข้อมูลไปยังบอร์ด TMS320C5X ได้อย่างถูกต้องจากการทดลองได้ทำการส่งทั้งข้อมูลที่มีการซ้ำกันและข้อมูลที่ไม่ซ้ำกัน เมื่อทำการตรวจสอบที่บอร์ด TMS320C5X แล้วพบว่าข้อมูลที่รับมาได้นั้นถูกต้องทุกครั้งแต่ก่อนที่จะสามารถทำการส่งข้อมูลได้อย่างถูกต้องจะต้องมีการปรับแต่งทางด้านวงจรที่ประกอบด้วย D-Flip Flop, Latch, Decoder/Demultiplexer เพื่อให้การทำงานติดต่อกันระหว่างเครื่องคอมพิวเตอร์และตัวบอร์ด TMS320C5X เป็นไปอย่างสอดคล้องซึ่งกันและกัน อีกส่วนหนึ่งก็คือผลการทดลองที่ได้จากการประมวลผลของบอร์ด TMS320C5X เพื่อให้ได้สัญญาณอะนาล็อก (Analog) โดยใช้กระบวนการประมวลผลของทฤษฎี Inverse Wavelet Transform ผลการทดลองในส่วนนี้เราจะได้สัญญาณรูปคลื่นออกมาซึ่งเมื่อเทียบกับค่าของการคำนวณจาก Matlab แล้วผลที่ได้จะมีความคล้ายคลึงกันมากอาจจะมี error ซึ่งเกิดขึ้นได้จาก

1. การแปลงจากค่าทศนิยมในเลขฐานสิบเป็น ระบบ Q Number
2. การที่นำมาประมวลผลโดยมีการคูณแล้วบวกหลายๆครั้งค่าที่ได้จะมีการปัดขึ้นทำให้ค่าที่คำนวณสูงกว่าการคำนวณจาก Matlab

สรุปโดยรวมแล้วในภาคส่งมีผลการทดลองที่เป็นไปตามเป้าหมายและมีความถูกต้องและสามารถแก้ไขปัญหาที่เกิด error ขึ้นได้

## 5.2 สรุปผลการทดลองทางด้านรับ

เช่นเดียวกับกับทางด้านส่งเราสามารถที่จะทำการสรุปผลทางด้านรับเป็นส่วนของการประมวลผลของบอร์ด TMS320C5X และส่วนของการรับข้อมูลไปตรวจสอบและเก็บโดยคอมพิวเตอร์ ในส่วนแรกคือการรับข้อมูลและนำไปประมวลผลของบอร์ด TMS320C5X ผลการทดลองที่ได้ในส่วนนี้ไม่สามารถบรรลุจุดประสงค์ได้แต่ก็อยู่ในเกณฑ์ที่ยอมรับสามารถแยกปัญหาที่เกิดขึ้นในส่วนนี้ได้ดังนี้

- ไม่สามารถ Sampling สัญญาณให้ได้ตรงตามที่ภาคส่งฯสัญญาณมา
- เกิด error ขึ้นในการประมวลผลซึ่งสืบเนื่องมาจากค่าที่ได้จากการ Sampling และการเกิด error ในขบวนการประมวลผลเอง
- การปิดทศนิยมในขั้นตอนสุดท้ายเพื่อให้ข้อมูลที่ได้นั้นถูกต้องมากที่สุด

การแก้ไขปัญหาที่ได้กระทำเพื่อให้ได้ผลการทดลองที่ดีที่สุด

- เรื่องของการ Sampling ได้แก้ไขปัญหาโดยให้ด้านส่งก่อนที่จะส่งข้อมูลทำการส่ง Start bit ก่อนเพื่อที่จะทำให้ทางด้านรับหลังจากตรวจพบ Start bit แล้วหลังจาก Start bit ผ่านไปก็จะทำการเริ่มเก็บข้อมูล แต่ผลที่ได้ก็ยังไม่สามารถทำให้ค่าที่ได้จากการส่งและรับมีความใกล้เคียงกัน ต่อมาจึงทำการขยายสัญญาณทางด้านส่งเพื่อทำให้ขนาดของสัญญาณมีค่าสูงเป็นการยกกระดืบ ให้ค่าของสัญญาณมีความสูงกว่า Noise ที่เกิดขึ้นในกระบวนการส่งผลที่ได้สามารถรับค่าได้เที่ยงตรงมากยิ่งขึ้นแต่การแก้ปัญหานี้ต้องยกเลิกไปเนื่องจากเมื่อลองส่งข้อมูลที่มีค่าสูงเช่น  $FF_{16}$  ค่าที่ได้จะเกิดความเพี้ยนเกิดขึ้นโดยที่ค่าที่ได้สูงสุดควรจะเป็นค่าบวกแต่เมื่อทำการส่งค่าออกไปจริงๆแล้วค่านั้นจะออกมาเป็นลบเนื่องจากเกินขีดแรงดันบวกสูงสุดที่ตัวบอร์ดจะทำการส่งได้จึงได้มีการแก้ไขปัญหาดังกล่าวด้วยวิธีการใหม่คือการเปลี่ยนความถี่ Sampling ทางด้านรับให้มีความถี่สูงกว่าทางด้านส่งเล็กน้อย จากการแก้ไขที่จุดนี้เองทำให้ผลการทดลองที่ได้ถูกต้องมากยิ่งขึ้นแต่ว่าผลเสียที่เกิดตามมาก็คือ ทางด้านรับจะรับข้อมูลเข้าไปประมวลผลไม่หมดทั้งรูปคลื่นของสัญญาณที่ส่งมาให้เนื่องจากเมื่อเพิ่มความถี่ Sampling ก็จะทำให้ค่าที่เก็บนั้นถูกในเบื้องต้นและจะทำการล้นข้อมูลที่รับเข้ามาจนเก็บครบตามจำนวนที่กำหนดดังนั้นค่าที่ได้จึงมีความถูกต้องช่วงแรกแต่ช่วงหลังมีความผิดพลาดแต่เมื่อเทียบผลการทดลองกับการแก้ไขปัญหาโดยวิธีอื่นแล้วเป็นผลที่ได้ถูกต้องมากที่สุดดังนั้นจึงใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้ามาแล้วทำการเปรียบเทียบกับไฟล์ต้นแบบว่ามีความถูกต้องมากเพียงใด จึงได้ทำการทดลองหา ค่าของความถูกต้องในระดับบิท จากที่กล่าวมาแล้วว่าในการเช็คค่าความถูกต้องในระดับของบิท นั้น จะทำการเช็คค่าความถูกต้องตั้งแต่ถูก 4 บิท ถูก 3 บิท ถูก 2 บิท ถูก 1 บิทและไม่ถูกเลย จาก ผลการทดลองที่ได้จะเห็นได้ว่าค่าของความถูกต้องก็จะมีค่าที่มากกว่าความผิดพลาดที่เกิดขึ้น ใน การทดลองได้ทดลองทำการส่งข้อมูลที่เป็นจำนวนประมาณ 1000 ตัว ผลของการทดลองที่ได้ในการ เช็คค่าความถูกต้องแบบเช็คบิทนี้ ค่าที่ได้จากการเปรียบเทียบความถูกต้องที่ทำการนับเป็นจำนวน ครั้ง จะเห็นว่าจำนวนของความถูกต้องเป็นจำนวนครั้งของความถูกต้อง 4 บิทรวมกับความถูกต้อง 3 บิท มีค่าประมาณ 900 กว่าครั้ง จากการทดลองจะเห็นว่ามีความถูกต้องมากอยู่ในเกณฑ์ที่น่าพอใจ ซึ่งการผิดพลาดทั้ง 4 บิทจะมีค่าประมาณ 5% ซึ่งอาจกล่าวได้ว่าความผิดพลาดอยู่ในระดับที่ไม่ สูงมาก ซึ่งยังคงทำให้สามารถเห็นลักษณะของข้อมูลเดิมได้ การที่มีข้อมูลผิดพลาดขึ้นในการรับข้อมูลนั้นอาจเกิดจากความผิดพลาดอันเนื่องมาจากค่าของการคำนวณในบอร์ด TMS320C5X มีการ บิดค่าและตัดค่าของข้อมูลทำให้เกิดการ error ขึ้น นอกจากนั้นยังอาจเกิดจากสาเหตุของการที่ ขนาดของตัวเวฟเล็ทมีขนาดสั้นเกินไปจึงทำให้ค่าที่รับเข้ามาได้ยังไม่ถูกต้องถึง 100% สาเหตุดังกล่าวนี้แก้ไขได้โดยการใช้เวฟเล็ทที่มีขนาดยาวขึ้นเพื่อที่จะทำให้ความถูกต้องของข้อมูลมีมากขึ้น นอกจากนี้แล้วความผิดพลาดที่เกิดขึ้นของการรับข้อมูลอาจจะเกิดมาจากการที่การ Sampling ของข้อมูลในฝั่งรับ Sampling ได้ไม่ตรงกับจุดที่ต้องการ เนื่องจากมีการเลื่อนของข้อมูลทำให้เกิดความผิดพลาดขึ้น ค่าที่ได้จากการ Sampling ก็จะมีการเลื่อนไปทำให้ได้ค่ามาไม่ตรงกับที่ต้องการ จากนั้น เมื่อทำการนำค่าที่รับมาจากการ Sampling ที่ไม่ตรงจุดไปทำการประมวลผลต่อไปโดยกระบวนการ ทางทฤษฎีของเวฟเล็ททรานส์ฟอร์มซึ่งในการประมวลผลในบอร์ด TMS320C5X ที่จะต้องมีการบิดค่า และตัดค่าดังที่กล่าวมาแล้ว ก็จะทำให้เกิดความผิดพลาดเพิ่มขึ้นไปอีก นอกจากนี้แล้วปัญหาที่เกิดขึ้นนี้อาจจะมาจากการลดทอนของสายส่งทำให้มีการลดทอนขนาดของสัญญาณบางส่วนไปทำให้เกิดความผิดพลาดขึ้น ซึ่งปัญหาดังกล่าวเหล่านี้ยังส่งผลถึงความสามารถในการรับและส่งข้อมูลบาง ตัวอีกด้วย ข้อมูลบางตัวที่ทำการทดลองในการรับและส่งข้อมูลจะมีความถูกต้องสูงและก็มีข้อมูล บางตัวที่จะมีความถูกต้องต่ำ จากสาเหตุของการที่ข้อมูลบางตัวมีความถูกต้องต่ำอาจจะเป็นเพราะ ว่าค่าของเลขฐาน 16 ของเลขนั้นมีผลต่อการรับและส่งข้อมูล ซึ่งทำให้ค่าของข้อมูลที่ทำการรับนั้นผิดพลาดเสมอ นอกจากนี้แล้วในการทดลองถ้ามีการนำค่า Coefficient ของเวฟเล็ทแม่มาใช้ในการ คำนวณผิด ซึ่งอาจจะเป็นการสลับตำแหน่งของเวฟเล็ทแม่ผิดก็จะทำให้ผลที่ได้แทนที่จะถูกต้องกลับ กลายเป็นผิดไป ซึ่งในการสลับตำแหน่งของเวฟเล็ทแม่จะต้องกระทำให้ถูกต้อง เพราะผลที่ได้จาก การทดลองแทนที่จะถูกกลับกลายเป็นผิดไปเลยก็ได้ซึ่งผลอาจจะมาจากการที่นำเอาค่าเวฟเล็ทแม่มาใช้ งานผิดสลับกันแทนที่จะได้ค่าของความถี่ต่ำออกมากลับได้ค่าของความถี่สูงออกมาทำให้ค่าของข้อ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ใช้ในเชิงพาณิชย์นั้น การคัด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลที่ส่งไปได้ไม่ตรงกับค่าของข้อมูลที่ได้รับเข้ามา ส่วนการทดลองที่ทำการรับค่าของเลขศูนย์ที่ส่งมาแล้วทำการเช็คค่ารับเลข 1 ได้ก็ครั้งเป็นการทดลองเพื่อหา bit error rate จากการทดลองมีความถูกต้องมากผิดพลาดเพียงเล็กน้อยเท่านั้น

### 5.3 แนวทางการแก้ไขปัญหาและปรับปรุง

สาเหตุที่ต้องทำการแก้ไขและปรับปรุงมีดังนี้

1. ใช้ Wavelet function ที่ทำให้การประมวลผลแล้วได้สัญญาณที่สั้นเกินไป
2. สัญญาณมีการเลื่อนเฟสและมีการลดทอนทางแอมพลิจูดดังนั้นเมื่อใช้ความถี่ Sampling ใกล้เคียงกันระหว่างฝ่ายส่งและฝ่ายรับแล้วจะทำให้เกิดความผิดพลาดได้ ควรแก้ไขโดยใช้อัตรา Sampling ที่ภาครับมากกว่าภาคส่ง 2 เท่าของฝ่ายส่งแต่การประมวลผลจะซับซ้อนมากยิ่งขึ้น

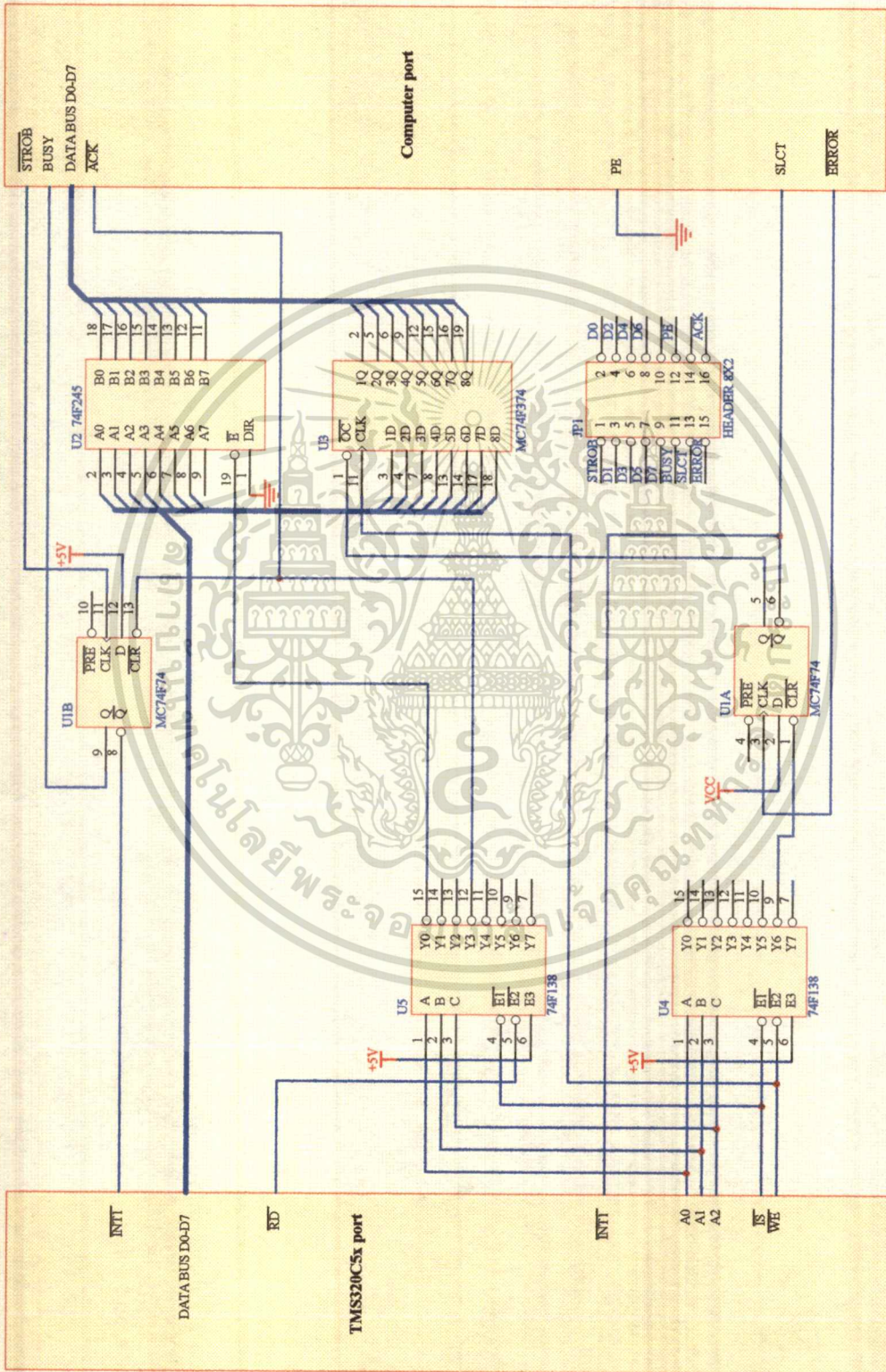
การแก้ไขและปรับปรุง

1. เปลี่ยน Wavelet Function เพื่อเพิ่มขนาดสัญญาณให้ยาวขึ้น
2. ใช้ความถี่ Sampling ฝ่ายรับเป็น 2 เท่า
3. ส่งข้อมูลติดต่อกันไปเพื่อให้ส่งได้เร็วขึ้นไม่ต้องมี Start bit
4. ก่อนการส่งและรับควรมีการตรวจสอบการลดทอนของสายส่งเสียก่อนแล้วทำการนำเปอร์เซ็นต์การลดทอนไปเพิ่มที่ฝ่ายส่ง



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Hardware Interfacing between TMS320C5x and Computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## File name : Realout.c

```

#include"stdio.h"
#include"stdlib.h"
#define port1 0x0378
#define port2 0x0379
#define port3 0x037A
main()
{
char ch,file_name[20];
int x,value1,value2,value3;
FILE *fp;
clrscr();

if((fp = fopen("test.txt","r+b")) == NULL)
{
printf("Error in open file\n");
exit(1);
}

while(!feof(fp))
{
ch = getc(fp);
if(ferror(fp))
{

```

```

printf("file error");
exit(1);
}

value1 = ch;
printf("%c",ch);
outportb(port1,value1);
value3 = 0xFF;

outportb(port3,value3);
outportb(port3,0x00);
value2 = inportb(port2);
while(value2 < 0x80)
{
value2 = inportb(port2);
}
fclose(fp);
}

```

## File name : Rub-use1.c

```

#include"stdio.h"
#include"dos.h"
#include"conio.h"
#include"stdlib.h"
#define port1 0x0378
#define port2 0x0379
#define port3 0x037A
void main()
{
/*-----first-----*/

char ch,file_name[20];
int y,z,value1,value2,value3;

```

```

FILE *fp;

/*-----second-----*/

char file_name1[20],file_name2[20];
unsigned char ch1,ch2,t[2],one[2],two[2];
int p,q,x;
float i1,i2,i3,i4,j1,j2,j3,j4;
FILE *fp1;
FILE *fp2;

clrscr();

```

```

if((fp = fopen("test1.txt","w+b")) == NULL)
{
    printf("Error in opening file\n");
    exit(1);
}

    outportb(port1,0xFF);    /* START */
    outportb(port3,0x03);
    outportb(port3,0x04);
    value2 = inportb(port2); /* check SLCT */
    y = value2 & 0x10;
    while(y != 0x10 )
        {
            value2 = inportb(port2);
            y = value2 & 0x10;
        }

    value1 = inportb(port1);
    ch = value1; /* < NEW> */
    printf("%c",ch);
    if(putc(ch,fp)==EOF)
        {
            printf("Error in writing file\n");
            exit(1);
        }

do {
    z=0;
    outportb(port3,0x03);
    outportb(port3,0x04);
    value2 = inportb(port2); /* check SLCT */
    y = value2 & 0x10;
    z=0;
    while(y != 0x10 )
        {
            z++;
            while(z == 10000)
                {
                    goto stop;
                }
            value2 = inportb(port2);
            y = value2 & 0x10;
            z=0;
            value1 = inportb(port1);
            ch = value1; /* <NEW> */
            printf("%c",ch);
            if(putc(ch,fp)==EOF)
                {
                    printf("Error in writing file\n");
                    exit(1);
                }
            }while(value1 != EOF);
        }while(value1 != EOF);

    stop:
    /* printf("File name 1 : ");
    gets(file_name1);*/

    if((fp1 = fopen("test.txt","r+b")) == NULL)
        {
            printf("Error in open file\n");
            exit(1);
        }

    /* printf("File name 2 : ");
    gets(file_name2);*/

    if((fp2 = fopen("test1.txt","r+b")) == NULL)
        {
            printf("Error in open file\n");
            exit(1);
        }

    i1=i2=i3=i4=j1=j2=j3=j4=0;

```

```

while(!feof(fp1)){
ch1 = getc(fp1);
if(ferror(fp1))
{
printf("file error");
exit(1);
}

/*printf("%x",ch1);*/
t[0] = ch1;
one[0] = t[0] & 0xF0;
one[0] = one[0] >> 4;
one[1] = t[0] & 0x0F;

for(p=0;p<2;++p)
{
if(one[p] == 0x0F)
{
i4 = i4 + 4;
}
else if(one[p]==0x07 || one[p]==0x0B || one[p]==0x0D || one[p]==0x0E)
{
i3 = i3 + 3;
}
else
if(one[p]==0x03 || one[p]==0x05 || one[p]==0x06 || one[p]==0x09 || one[p]==0x0A || one[p]==0x0C)
{
i2 = i2 + 2;
}
else
if(one[p]==0x01 || one[p]==0x02 || one[p]==0x04 || one[p]==0x08)
{
i1 = i1 + 1;
}
}
}

while(!feof(fp2)){
ch2 = getc(fp2);
if(ferror(fp2))
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("file error");
exit(1);
}

/*printf("%x",ch2);*/
t[1] = ch2;
two[0] = t[1] & 0xF0;
two[0] = two[0] >> 4;
two[1] = t[1] & 0x0F;
for(q=0;q<2;++q)
{
if(two[q] == 0x0F)
{
j4 = j4 + 4;
}
else
if((two[q]==0x07 || two[q]==0x0B || two[q]==0x0D || two[q]==0x0E)
{
j3 = j3 + 3;
}
else
if((two[q]==0x03 || two[q]==0x05 || two[q]==0x06 || two[q]==0x09 || two[q]==0x0A || two[q]==0x0C)
{
j2 = j2 + 2;
}
else
if((two[q]==0x01 || two[q]==0x02 || two[q]==0x04 || two[q]==0x08)
{
j1 = j1 + 1;
}
}
}

printf("\n07\n");
printf("The number 1 of sources is = %fn",i1+i2+i3+i4);
printf("The number 1 of received is = %fn",j1+j2+j3+j4);
fclose(fp);
fclose(fp1);
fclose(fp2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## File name : Rub-use2.c

```

#include"stdio.h"
#include"dos.h"
#include"conio.h"
#include"stdlib.h"
#define port1 0x0378
#define port2 0x0379
#define port3 0x037A
void main()
{
char ch,file_name[20];
int i,y,z,value1,value2,value3;

FILE *fp;

char file_name1[20],file_name2[20];
unsigned char ch1,ch2,t1[2],sou1[2],rec1[2];
unsigned char test1[2];
int x,p,q;
float
i0,i1,i2,i3,i4,j0,j1,j2,j3,j4,k0,k1,k2,k3,k4,l0,l1,l2,l3,l4;

FILE *fp1;
FILE *fp2;

clrscr();

if((fp = fopen("test1.txt","w+b")) == NULL)
{
printf("Error in opening file\n");
exit(1);
}

outport(port1,0xFF); /* START */
outportb(port3,0x03);
outportb(port3,0x04);
value2 = inportb(port2); /* check
SLCT */

y = value2 & 0x10;

while(y != 0x10 )
{
value2 = inportb(port2);
y = value2 & 0x10;
}

value1 = inportb(port1);
ch = value1; /* <NEW> */
printf("%c",ch);
if(putc(ch,fp)==EOF)
{
printf("Error in writing file\n");
exit(1);
}

do {
z=0;
outportb(port3,0x03);
outportb(port3,0x04);
value2 = inportb(port2); /* check SLCT */
y = value2 & 0x10;
z=0;
while(y != 0x10 )
{
z++;
while(z == 10000)
{
goto stop;
}

value2 = inportb(port2);
y = value2 & 0x10;
}

z=0;
value1 = inportb(port1);
ch = value1; /* <NEW> */
printf("%c",ch);
z=0;
if(putc(ch,fp)==EOF)
{
printf("Error in writing file\n");
exit(1);
}
}

```

```

    }
    )while(value1 != EOF);
stop:

i0=i1=i2=i3=i4=j0=j1=j2=j3=j4=0;
k0=k1=k2=k3=k4=l0=l1=l2=l3=l4=0;
if((fp1 = fopen("test.txt","r+b")) == NULL)
{
    printf("Error in open file\n");
    exit(1);
}

if((fp2 = fopen("test1.txt","r+b")) == NULL)
{
    printf("Error in open file\n");
    exit(1);
}
x = 0;
while(!feof(fp1) && !feof(fp2))
{
    ch1 =getc(fp1);
    ch2 =getc(fp2);
    if(ferror(fp1) && ferror(fp2))
    {
        printf("File error");
        exit(1);
    }
    /* printf("%x",ch1);*/

    x++;
    t1[0] = ch1;
    sou1[0] = t1[0] & 0xF0;
    sou1[0] = sou1[0] >> 4;
    sou1[1] = t1[0] & 0x0F;

    t1[1] = ch2;
    rec1[0] = t1[1] & 0xF0;
    rec1[0] = rec1[0] >> 4;
    rec1[1] = t1[1] & 0x0F;

```

```

if(x == 1)
{
    if(sou1[0] >= rec1[0])
    {
        test1[0] = sou1[0] - rec1[0];
        if(test1[0] == 0x00)
        {
            i4 = i4 + 1;
        }
        else
        if(test1[0] == 0x01)
        {
            i3 = i3 + 1;
        }
        else
        if(test1[0] == 0x02)
        {
            i2 = i2 + 1;
        }
        else
        if(test1[0] == 0x03)
        {
            i1 = i1 + 1;
        }
        else
        {
            i0 = i0 + 1;
        }
    }

    if(sou1[0] < rec1[0])
    {
        test1[0] = rec1[0] - sou1[0];
        if(test1[0] == 0x00)
        {
            i4 = i4 + 1;
        }
        else
        if(test1[0] == 0x01)
        {

```

```

        i3 = i3 + 1;
    }
else
if(test1[0] == 0x02)
{
    i2 = i2 + 1;
}
else
if(test1[0] == 0x03)
{
    i1 = i1 + 1;
}
else
{
    i0 = i0 + 1;
}
}
if(x == 2)
{
if(sou1[0] >= rec1[0])
{
    test1[0] = sou1[0] - rec1[0];
    if(test1[0] == 0x00)
    {
        k4 = k4 + 1;
    }
    else
if(test1[0] == 0x01)
{
        k3 = k3 + 1;
    }
    else
if(test1[0] == 0x02)
{
        k2 = k2 + 1;
    }
    else
if(test1[0] == 0x03)
{
        k1 = k1 + 1;
    }
    else
{
        k0 = k0 + 1;
    }
}
}
if(x == 1)
{
    if(sou1[1] >= rec1[1])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

    }
}

if(sou1[1] < rec1[1])
{
    test1[1] = rec1[1] - sou1[1];
    if(test1[1] == 0x00)
    {
        i4 = i4 + 1;
    }
    else
    if(test1[1] == 0x01)
    {
        i3 = i3 + 1;
    }
    else
    if(test1[1] == 0x02)
    {
        i2 = i2 + 1;
    }
}
else
if(test1[1] == 0x03)
{
    i1 = i1 + 1;
}
else
{
    i0 = i0 + 1;
}
}
}

if(x == 2)
{
    x = 0;
}
}

printf("007\n");

```

```

printf("The first character with high bit corrected 4 bit is = %.2f\n",i4);
printf("The first character with high bit corrected 3 bit is = %.2f\n",i3);
printf("The first character with high bit corrected 2 bit is = %.2f\n",i2);
printf("The first character with high bit corrected 1 bit is = %.2f\n",i1);
printf("The first character with high bit is noncorrected is = %.2f\n\n",i0);
printf("The first character with low bit corrected 4 bit is = %.2f\n",j4);
printf("The first character with low bit corrected 3 bit is = %.2f\n",j3);
printf("The first character with low bit corrected 2 bit is = %.2f\n",j2);
printf("The first character with low bit corrected 1 bit is = %.2f\n",j1);
printf("The first character with low bit is noncorrected is = %.2f\n\n",j0);
printf("The second character with high bit corrected 4 bit is = %.2f\n",k4);
printf("The second character with high bit corrected 3 bit is = %.2f\n",k3);
printf("The second character with high bit corrected 2 bit is = %.2f\n",k2);
printf("The second character with high bit corrected 1 bit is = %.2f\n",k1);
printf("The second character with high bit is noncorrected is = %.2f\n\n",k0);
printf("The second character with low bit corrected 4 bit is = %.2f\n",l4);
printf("The second character with low bit corrected 3 bit is = %.2f\n",l3);
printf("The second character with low bit corrected 2 bit is = %.2f\n",l2);
printf("The second character with low bit corrected 1 bit is = %.2f\n",l1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("The second character with low bit is noncorrected is = %.2f",i0);
fclose(fp);
fclose(fp1);
fclose(fp2);
}

```

### Filename : Rub-use3.c

```

#include"stdio.h"
#include"dos.h"
#include"conio.h"
#include"stdlib.h"
#define port1 0x0378
#define port2 0x0379
#define port3 0x037A
void main()
{
/*-----first-----*/

char ch,file_name[20];
int y,z,value1,value2,value3;
FILE *fp;

/*-----second-----*/

char file_name1[20],file_name2[20];
unsigned char ch1,ch2,t[2],one[2],two[2];
int p,q,x;
float i1,i2,i3,i4;
FILE *fp1;

clrscr();

if((fp = fopen("test1.txt","w+b")) == NULL)
{
printf("Error in opening file\n");
exit(1);
}

output(port1,0xFF); /* START */
output(port3,0x03);
output(port3,0x04);
value2 = inportb(port2); /* check SLCT */
y = value2 & 0x10;
z=0;
while(y != 0x10 )
{
value1 = inportb(port1);
ch = value1; /* < NEW> */
printf("%c",ch);
if(putc(ch,fp)==EOF)
{
printf("Error in writing file\n");
exit(1);
}
do {
z=0;
output(port3,0x03);
output(port3,0x04);
value2 = inportb(port2); /* check SLCT */
y = value2 & 0x10;
z=0;
while(y != 0x10 )
{
z++;
while(z == 10000)

```

```

        goto stop;
    }
    value2 = inportb(port2);
    y = value2 & 0x10;
}
z=0;
value1 = inportb(port1);

    ch = value1;    /* <NEW> */
    printf("%c",ch);
    z=0;
    if(putc(ch,fp)==EOF)
    {
        printf("Error in writing file\n");
        exit(1);
    }
}while(value1 != EOF);

stop:
if((fp1 = fopen("test1.txt","r+b")) == NULL)
{
    printf("Error in open file\n");
    exit(1);
}

if(one[p]==0x07 || one[p]==0x0B || one[p]==0x0D || one[p]==0x0E)
{
    i3 = i3 + 3;
}
else
if(one[p]==0x03 || one[p]==0x05 || one[p]==0x06 || one[p]==0x09 || one[p]==0x0A || one[p]==0x0C)
{
    i2 = i2 + 2;
}
else
if(one[p]==0x01 || one[p]==0x02 || one[p]==0x04 || one[p]==0x08)
{
    i1 = i1 + 1;
}
}

        i1=i2=i3=i4=0;

        while(!feof(fp1)){
            ch1 = getc(fp1);
            if(ferror(fp1))
            {
                printf("file error");
                exit(1);
            }
            /*printf("%x",ch1);*/
            t[0] = ch1;
            one[0] = t[0] & 0xF0;
            one[0] = one[0] >> 4;
            one[1] = t[0] & 0x0F;

            for(p=0;p<2;++p)
            {
                if(one[p] == 0x0F)
                {
                    i4 = i4 + 4;
                }
                else
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }  
}  
  
printf("\007\n");  
printf("The number 1 of receive is = %f\n",i1+i2+i3+i4);  
fclose(fp);  
fclose(fp1);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        .ps      0a00h
        .entry
START:  SETC     INTM      ;Disable interrupts
        LDP     #0        ;Set data page pointer
        OPL    #0&34h,PMST
        LACC   #0
        SAMM   CWSR      ;Set software wait state to 0
        SAMM   PDWSR

* Reset AIC by writing to PA2 (address >52) to DSK

        SPLK   #020h,TCR    ; set timer to generate 10 MHz clock to AIC
        SPLK   #01h,PRD     ; set period register to 1
        SPLK   #0&h,SPC     ; set serial port for non continuous mode
        SPLK   #0C&h,SPC    ; enable serial port
        SPLK   #022h,IMR    ; Using XINT sync TA & RA
        CALL   AICINIT     ; Initialize AIC and enable interrupts

;
;Start Program to derive data from computer and process to send it
;
        LAR    AR2,#30D4h
RSTART: SETC   INTM
        LACC   #0
        SAMM   DXR
        MAR    *,AR2
        BANZ   WAVELET,*-
        B      END

*****
* Wavelet Transform method *
*****
*----- Prepare Coefficient 1 and 2 -----*
WAVELET: SETC   INTM
        LAR    AR0,#01500h
        LDP    #COEFF0
        MAR    *,AR0
        LACC   *,0,AR0
        AND    #00Fh
        SACL   COEFF1
        LACC   **+,0,AR0
        AND    #00F0h
        RPT    #3
        SFR
        SACL   COEFF2
*----- G1 * COEFF1 -----*
        LAR    AR1,#1010h    ; First address XG10
        ZAP    #1
        RPT
        MAC    G3,COEFF1
        MAR    *,AR1
        SACL   *+
        AND    #0
        APAC
        MAR    **+,AR1
        SACL   *-
        ZAP
        RPT    #1
        MAC    G2,COEFF1
        SACL   *+
        AND    #0
        APAC
        MAR    **+,AR1
        SACL   *                ; Final address XG13
*----- H1 * COEFF1 -----*
        LAR    AR1,#1010h    ; First address A0
        ZAP
        RPT    #1
        MAC    H3,COEFF2
        MAR    *,AR1
        ADD    *
        SACL   *+
        AND    #0
        APAC
        MAR    **+,AR1
        ADD    *
        SACL   *-

        ZAP
        RPT    #1
        MAC    H2,COEFF2

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SACL      *+
AND       #0
APAC
MAR       *+,AR1
ADD       *
SACL      *           ; Final address A3

```

\*----- Prepare Coefficient 31 and 32 -----\*

```

ZAP
MAR       * ,AR0
LACC      * ,0 ,AR0
AND       #000Fh
SACL      COEFF31
LACC      * ,0 ,AR0
AND       #00F0h
RPT       #3
SFR
SACL      COEFF32

```

\*----- G2 \* Coefficient 31 and 32 -----\*

```

LAR       AR1, #1022h           ; First address XG22
ZAP
RPT       #1
MAC       G3, COEFF32
MAR       * ,AR1
SACL      *+
AND       #0
RPT       #1
MAC       G3, COEFF31
MAR       *+, AR1
SACL      *+
AND       #0
APAC
MAR       *+, AR1
SACL      *           ; Final address XG26

LAR       AR1, #1023h           ; First address XG23
ZAP
RPT       #1
MAC       G2, COEFF32
MAR       * ,AR1
SACL      *+
AND       #0
RPT       #1
MAC       G2, COEFF31
MAR       *+, AR1
SACL      *+
AND       #0
APAC
MAR       *+, AR1
SACL      *           ; Final address XG27

```

\*----- H2 \* CONVOLUTION -----\*

```

LAR       AR1, #1010h           ; Address A0
LAR       AR5, #1030h           ; First address B0
LAR       AR3, #4               ; Repeat 4 time
LOOP1:
ZAP
MAR       * ,AR1
RPT       #1
MAC       H3, *
RPT       #4
SFR
MAR       *+, AR5
SACL      *+
AND       #0
MAR       *+, AR3
BANZ     LOOP1, *-, AR5
APAC
RPT       #4
SFR
SACL      *

LAR       AR1, #1010h           ; Address A0
LAR       AR5, #1031h           ; First address B1
LAR       AR3, #4               ; Repeat 4 time
ZAP
LOOP2:
MAR       * ,AR1
RPT       #1

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAC      H2,*
RPT      #9
SFR
MAR      *+,ARS
SACL     *+
AND      #0
MAR      *+,AR3
BANZ     LOOP2,*-,ARS
APAC
RPT      #9
SFR
SACL     *

```

```

LAR      AR3,#1022h ; Address of XG22
LAR      AR4,#1032h ; Address of XH22
LAR      AR5,#5
SUMA1:  MAR      *-AR3
LACC     *+,0,AR4
ADD      *
SACL     *+,0,ARS
BANZ     SUMA1,*-

```

\*----- H3 \* CONVOLUTION -----\*

```

LAR      AR6,#1040h ; First address keep H3*
LAR      AR5,#1030h ; First address 80
LAR      AR4,#09h ; In loop 9 time
LOOP3:  ZAP
MAR      *-ARS
RPT      #1
MAC      H3,*
RPT      #9
SFR
MAR      *+,AR6
SACL     *+
AND      #0
MAR      *+,AR4
BANZ     LOOP3,*-,AR6
APAC
RPT      #9
SFR
SACL     *

```

```

LAR      AR6,#1041h ; Secound address keep H3*
LAR      AR5,#1030h ; First address 80
LAR      AR4,#09h ; In loop 9 time
LOOP4:  ZAP
MAR      *-ARS
RPT      #1
MAC      H2,*
RPT      #9
SFR
MAR      *+,AR6
SACL     *+
AND      #0
MAR      *+,AR4
BANZ     LOOP4,*-,AR6
APAC
RPT      #9
SFR
SACL     *

```

\*----- H4 \* CONVOLUTION -----\*

```

LAR      AR6,#1060h ; First address keep H4 *
LAR      AR5,#1040h ; First address C0
LAR      AR4,#015h ; In loop 15 time
LOOP5:  ZAP
MAR      *-ARS
RPT      #1
MAC      H3,*
RPT      #13
SFR
MAR      *+,AR6
SACL     *+
AND      #0
MAR      *+,AR4
BANZ     LOOP5,*-,AR6
APAC
RPT      #13

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SFR
SACL      *

LAR       AR6,#1061h    ; Secound address keep H4*
LAR       AR5,#1040h    ; First address C0
LAR       AR4,#015h     ; In loop 15 time

LOOP6:    MAR          *,AR5
          RPT          #1
          MAC          H2,*
          RPT          #13
          SFR
          MAR          *+,AR6
          SACL        *+
          AND          #0
          MAR          *+,AR4
          BANZ        LOOP6,*-,AR6
          APAC
          RPT          #13
          SFR
          SACL        *

```

```

-----*
;SEND DATA TO OUTPUT
LDP       #0
SPLK     #020h,IMR      ; Using XINT syn TX & RX
CLRC     OVM           ; OVM = 0
SPM      0             ; PM = 0

```

```

*----- Start Bit for check -----*

```

```

LOOP7:    LAR         AR4,#2000h
          LAR         AR5,#05h
          LAR         AR6,#015h
          CLRC        INTM
          IDLE
          MAR         *,AR6
          BANZ        LOOP7,*-,AR4
          MAR         *+,AR4
          LOOP6:    CLRC        INTM
          IDLE
          MAR         *,AR4
          MAR         *+,AR5
          BANZ        LOOP6,*-

```

```

LAR       AR4,#1060h    ; First address of DATA
LACC     #106Dh        ; Final address of DATA
SAMM    ARCR
AND      #0
LSEND:   CLRC        INTM      ; Send DATA
          IDLE
          SETC        INTM
          MAR         *+,AR4
          CHPR        2
          BCND        LSEND,NTC

```

```

CLEARM:  LAR         AR1,#1000h
          LACC        #106Dh
          SAMM        ARCR
          MAR         *,AR1
          LACC        *
          AND          #0
          SACL        *+,0,AR1
          CHPR        2
          BCND        CLEARM,NTC

```

```

B        RSTART      ; Return to Start

```

```

;----- End of main program -----;

```

```

*****
* Analog Interface Circuit INITIALIZATION *
*****

```

```

AICINIT: MAR          *,AR0
          LACC        #0&0h      ; Pulse AIC reset by setting it low
          SACH        DXR
          SACL        GREG
          LAR         AR0,#0FFFFh

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการปฏิบัติงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RPT    #100          ; and taking it high after 10000 cycles
LACC   *,0,ARD      ; (.5ms at 50ns)
SACH   GREG
;-----
LDP    #TA
SETC   SXM          ;
LACC   TA,9        ; Initialized TA and RA register
ADD    RA,2         ;
CALL   AIC_2ND     ;
;-----
LDP    #TA
LACC   TB,9        ; Initialized TB and RB register
ADD    RB,2         ;
ADD    #02h        ;
CALL   AIC_2ND     ;
;-----
LDP    #AIC_CTR
LACC   AIC_CTR,2   ; Initialized control register
ADD    #03h        ;
CALL   AIC_2ND     ;
RET

```

AIC\_2ND:

```

LDP    #0
SACH   DXR          ;
CLRC   INTM
IDLE
ADD    #6h,15      ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
SACH   DXR          ;
IDLE
SACL   DXR          ;
IDLE
LACL   #0           ;
SACL   DXR          ; make sure the word got sent
IDLE
SETC   INTM
RET

```

END: .end

Filename : SendA.asm ทำการเปลี่ยนบรรทัดที่ .ps 01500h เป็น  
.word 0041h  
.word 0042h

Filename : Au9.asm

.mmregs

```

*****
* Initialize variable and memory *
*****

```

```

.ps    01501h
.word  0042h

```

```

*****
* Function Wavelet Daubechie4 *
*****

```

```

.ds    0e00h
H3:    .q10    -0.1294
H1:    .q10    0.8365
H2:    .q10    0.2241
H0:    .q10    0.483
G3:    .q10    -0.483
G1:    .q10    -0.2241
G2:    .q10    0.8365
G0:    .q10    -0.1294

```

```

.ds    0f00h          ; Start address for set program
TA     .word    10     ; LP SCF = 518.4 KHz ; Cutoff = 12.96 KHz
RA     .word    10     ; BP SCF = 518.4 KHz
TB     .word    63     ; D/A Fs = 8 KHz

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเอกสารลิขสิทธิ์สงวนไว้สำหรับหน่วยงานราชการเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

WAIT:   LAR      AR5,#2b00h
        MAR      *,AR5
        IN       *,0003h      ; Tell PC "I am ready to recieve"
        CLRC    INTM         ; Enable Interrupt
        IDLE    ; Wait Interrupt
    
```

```

*****
* Wavelet Transform method *
*****
    
```

\*----- Prepare Coefficient 1 and 2 -----\*

```

SETC    INTM
LAR     AR0,#01500h
LDP     #COEFF0
MAR     *,AR0
LACC    *,0,AR0
AND     #000Fh
SACL   COEFF1
LACC    *+,0,AR0
AND     #00F0h
RPT     #3
SFR
SACL   COEFF2
    
```

\*----- G1 \* COEFF1 -----\*

```

LAR     AR1,#1010h      ; First address XG10
ZAP
RPT     #1
MAC     G1,COEFF1
MAR     *,AR1
SACL   *+
AND     #0
APAC
MAR     *+,AR1
SACL   *-

ZAP
RPT     #1
MAC     G2,COEFF1
SACL   *+
AND     #0
APAC
MAR     *+,AR1
SACL   *
        ; Final address XG13
    
```

\*----- H1 \* COEFF1 -----\*

```

LAR     AR1,#1010h      ; First address A0
ZAP
RPT     #1
MAC     H1,COEFF2
MAR     *,AR1
ADD
SACL   *+
AND     #0
APAC
MAR     *+,AR1
ADD
SACL   *-

ZAP
RPT     #1
MAC     H2,COEFF2
ADD
SACL   *+
AND     #0
APAC
MAR     *+,AR1
ADD
SACL   *
        ; Final address A3
    
```

\*----- Prepare Coefficient 31 and 32 -----\*

```

ZAP
MAR     *,AR0
LACC    *,0,AR0
AND     #000Fh
SACL   COEFF31
LACC    *,0,AR0
    
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND      #00F0h
RPT      #3
SFR
SACL     C0EFF32

```

```
*----- G2 * Coefficient 31 and 32 -----*
```

```

LAR      AR1,#1022h      ; First address XG22
ZAP
RPT      #1
MAC      G3,C0EFF32
MAR      *-AR1
SACL     *+
AND      #0
RPT      #1
MAC      G3,C0EFF31
MAR      *+AR1
SACL     *+
AND      #0
APAC
MAR      *+AR1
SACL     *                ; Final address XG26

```

```

LAR      AR1,#1023h      ; First address XG23
ZAP
RPT      #1
MAC      G2,C0EFF32
MAR      *-AR1
SACL     *+
AND      #0
RPT      #1
MAC      G2,C0EFF31
MAR      *+AR1
SACL     *+
AND      #0
APAC
MAR      *+AR1
SACL     *                ; Final address XG27

```

```
*----- H2 * CONVOLUTION -----*
```

```

LAR      AR1,#1010h      ; Address A0
LAR      AR2,#1030h      ; First address B0
LAR      AR3,#4          ; Repeat 4 time
ZAP
LOOP1:  MAR      *-AR1
RPT      #1
MAC      H3,*
RPT      #4
SFR
MAR      *+AR2
SACL     *+
AND      #0
MAR      *+AR3
BANZ    LOOP1,*-AR2
APAC
RPT      #4
SFR
SACL     *

```

```

LAR      AR1,#1010h      ; Address A0
LAR      AR2,#1031h      ; First address B1
LAR      AR3,#4          ; Repeat 4 time
ZAP
LOOP2:  MAR      *-AR1
RPT      #1
MAC      H2,*
RPT      #4
SFR
MAR      *+AR2
SACL     *+
AND      #0
MAR      *+AR3
BANZ    LOOP2,*-AR2
APAC
RPT      #4
SFR
SACL     *

```

```

LAR      AR3,#1022h      ; Address of XG22
LAR      AR4,#1032h      ; Address of XH22

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท เทคโนโลยีการช่าง จำกัด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SUMA1:  LAR      AR5, #5
        MAR      *, AR3
        LACC     ++, D, AR4
        ADD      *
        SACL     ++, D, AR5
        BANZ     SUMA1, *-

```

```
*----- H3 * CONVOLUTION -----*
```

```

        LAR      AR6, #1040h    ; First address keep H3*
        LAR      AR5, #1030h    ; First address B0
        LAR      AR4, #09h      ; In loop 9 time
LOOP3:  ZAP
        MAR      *, AR5
        RPT      #1
        MAC      H3, *
        RPT      #9
        SFR
        MAR      ++, AR6
        SACL     ++
        AND      #0
        MAR      ++, AR4
        BANZ     LOOP3, *-, AR6
        APAC
        RPT      #9
        SFR
        SACL     *

```

```

        LAR      AR6, #1041h    ; Secound address keep H3*
        LAR      AR5, #1030h    ; First address B0
        LAR      AR4, #09h      ; In loop 9 time
LOOP4:  ZAP
        MAR      *, AR5
        RPT      #1
        MAC      H2, *
        RPT      #9
        SFR
        MAR      ++, AR6
        SACL     ++
        AND      #0
        MAR      ++, AR4
        BANZ     LOOP4, *-, AR6
        APAC
        RPT      #9
        SFR
        SACL     *

```

```
*----- H4 * CONVOLUTION -----*
```

```

        LAR      AR6, #1060h    ; First address keep H4 *
        LAR      AR5, #1040h    ; First address C0
        LAR      AR4, #015h     ; In loop 15 time
LOOP5:  ZAP
        MAR      *, AR5
        RPT      #1
        MAC      H3, *
        RPT      #13
        SFR
        MAR      ++, AR6
        SACL     ++
        AND      #0
        MAR      ++, AR4
        BANZ     LOOP5, *-, AR6
        APAC
        RPT      #13
        SFR
        SACL     *

```

```

        LAR      AR6, #1061h    ; Secound address keep H4*
        LAR      AR5, #1040h    ; First address C0
        LAR      AR4, #015h     ; In loop 15 time
LOOP6:  ZAP
        MAR      *, AR5
        RPT      #1
        MAC      H2, *
        RPT      #13
        SFR
        MAR      ++, AR6
        SACL     ++
        AND      #0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAR      *+,AR4
BANZ    LOOP6,*-,AR6
APAC
RPT     #13
SFR
SACL    *

```

```

;SEND DATA TO OUTPUT

```

```

LDP     #0
SPLK   #020h,IMR      ; Using XINT syn TX & RX
CLRC   0VM           ; 0VM = 0
SPM    0             ; PM = 0

```

```

*----- Start Bit for check -----*

```

```

LOOP7:  LAR      AR4,#2000h
        LAR      AR5,#05h
        LAR      AR6,#015h
        CLRC    INTM
        IDLE
        MAR     *+,AR6
        BANZ   LOOP7,*-,AR4
LOOP8:  MAR     *+,AR4
        CLRC    INTM
        IDLE
        MAR     *+,AR4
        MAR     *+,AR5
        BANZ   LOOP8,*-

```

```

        LAR      AR4,#1060h      ; First address of DATA
        LACC    #106Dh          ; Final address of DATA
        SAMM   ARCR
        AND    #0
NSEND:  LAR      AR2,#0h         ; 0 time for one DATA
LSEND:  CLRC    INTM           ; Send DATA
        IDLE
        SETC   INTM
        MAR     *+,AR2
        BANZ   LSEND,*-,AR4
        MAR     *+,AR4
        CMPR   2
        BCND   NSEND,NTC

```

```

CLEARM: LAR      AR1,#1000h
        LACC    #106Dh
        SAMM   ARCR
        MAR     *+,AR1
        LACC    *
        AND    #0
        SACL   *+,0,AR1
        CMPR   2
        BCND   CLEARM,NTC

```

```

B      RSTART      ; Return to Start

```

```

;----- End of main program -----;

```

```

*****
*   Analog Interface Circuit INITIALIZATION   *
*****

```

```

AICINIT: MAR     *+,AR0
        LACC    #080h          ; Pulse AIC reset by setting it low
        SACH   DXR
        SACL   GREG
        LAR    ARD,#0FFFFh
        RPT    #100           ; and taking it high after 10000 cycles
        LACC   *+,0,AR0       ; (.5ms at 50ns)
        SACH   GREG
;-----
        LDP    #TA
        SETC   SXM
        LACC   TA,9          ; Initialized TA and RA register
        ADD   RA,2
        CALL   AIC_2ND
;-----
        LDP    #TA
        LACC   TB,9          ; Initialized TB and RB register

```

เอกสารนี้เป็นเอกสารที่ TB.9 ไว้สำหรับการใช้ในระบบที่รองรับการนำเอาข้อมูลเหล่านี้ไปประมวลผลให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD    RB,2           ;
ADD    #02h          ;
CALL   AIC_2ND       ;
;-----
LDP    *AIC_CTR
LACC   AIC_CTR,2     ; Initialized control register
ADD    #03h          ;
CALL   AIC_2ND       ;
RET                                         ;

```

AIC\_2ND:

```

LDP    #0
SACH   DXR           ;
CLRC   INTM
IDLE
ADD    #6h,15        ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
SACH   DXR           ;
IDLE
SACL   DXR           ;
IDLE
LACL   #0            ;
SACL   DXR           ; make sure the word got sent
IDLE
SETC   INTM
RET                                         ;
.end

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. C.K. Chui., " An Introduction to Wavelets. ", Academic Press.Boston 1992
2. C. Makkun, P. Koosirivanichakom, K Chitsakul and M. Sangworasil, " Editing Character of curve by Wavelets. " Proceeding of the REGIONAL SYMPOSIUM ON TELECOMMUNICATIONS ELECTRONICS CIRCUIT AND SYSTEMS, KMITL Bangkok, july 1996,pp. C49-54.
3. Douglas V. Hall , " Microprocessors And Interfacing Programming and Hardware ", Singapore, 1991
4. Nico M. Temme., " Wavelets An Elementary Treatment of Theory and Applications" , SanDiego: Academic, 1993 ,pp.1-11.
5. R.K. Young., " WAVELET THEORY AND ITS APPLICATIONS ", Kluwer Academic Publishers. Boston 1993
6. Texas Instruments , " TMS320C5x Users's Guide " , Houston, Texas, Januar 1993
7. Texas Instruments , " TMS320C5x DSP Starter Kit User's Guide " , Dallas, Texas , 1994
8. The Mathworks, Incorporations , " Matlab High-Performance Numeric Computation ans Visualization Software " , Natick, Mass., 1992
9. ธันวา ศรีประโม่ง , " การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม " , หอนงจอก, กรุงเทพฯ, พ.ศ. 2539
10. มนต์รี พยনারอดลาวัณย์ , " การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี " , ยานนาวา, กรุงเทพฯ พ.ศ. 2535