สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตัวแปลงเวฟเลตไม่ต่อเนื่องโดยอุปกรณ์ FPGA

An FPGA-Based Implementation of Discrete Wavelet Transform

โดย

นาย สุรชัย แก้วศรีนาค

นาย อรรถพล วีระนพนันท์

นาย อัศนัย นิธิโรจนานนท์

เลขหมู่.....
เลขทะเบียน 72693
วัน,เดือน,ปี 21 มิ.ย. 2550

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปลงเวฟเลตไม่ต่อเนื่องโดยอุปกรณ์ FPGA

An FPGA-Based Implementation of Discrete Wavelet Transform

โดย

นาย สุรชัย	แก้วศรีนาค	46010868
นาย อรรถพล	วีระนพนันท์	46010940
นาย อัศนัย	นิธิโรจนานนท์	46010956

อาจารย์ที่ปรึกษา

ผศ. อัครพล ตรีรัตน์

อ. ศรวัฒน์ ชิวปรีชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ตัวแปลงเวฟเลตไม่ต่อเนื่องโดยอุปกรณ์ FPGA**

An FPGA-Based Implementation of Discrete Wavelet Transform

ผู้จัดทำ

นาย สุรชัย แก้วศรีนาค	**46010868**
นาย อรรถพล วีระนพนันท์	**46010940**
นาย อัศนัย นิธิโรจนานนท์	**46010956**

อาจารย์ที่ปรึกษา

(ผศ. อัครพล ตรีรัตน์)

อาจารย์ที่ปรึกษา

(อ. ศรวัฒน์ ชิวปรีชา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปลงเวฟเลตไม่ต่อเนื่องโดยอุปกรณ์ FPGA

An FPGA-Based Implementation of Discrete Wavelet Transform

โดย นาย สุรชัย แก้วศรีนาค 46010868

นาย อรรถพล วีระนพนันท์ 46010940

นาย อัศนัย นิธิโรจนานนท์ 46010956

อาจารย์ที่ปรึกษา ผศ.อัครพล ตรีรัตน์

อ. ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

การแปลงเวฟเลตไม่ต่อเนื่องเป็นเครื่องมือสำหรับการวิเคราะห์สัญญาณที่มีประสิทธิภาพ สามารถนำไปประยุกต์ใช้งานในทางปฏิบัติได้อย่างกว้างขวาง เนื่องจากเป็นวิธีการคำนวณระดับสูง ในปัจจุบันจึงสามารถสร้างตัวแปลงเวฟเลตให้ทำงานได้ใกล้เคียงกับความต้องการในการประมวลผลแบบเวลาจริงสำหรับการประยุกต์ใช้งานโดยทั่วไป

ปริญญานิพนธ์นี้นำเสนอการสร้างตัวแปลงเวฟเลตไม่ต่อเนื่องโดยใช้อุปกรณ์ลอจิกที่โปรแกรมได้ (FPGA)

ABSTRACT

The discrete wavelet transform has gained the reputation of being a very effective signal analysis tool for many practical application. However, due to its computation-intensive, current implements of the transform fall short of meeting real-time processing requirements of most application.

This thesis, describe an implementation of discrete wavelet transform using high-density field programmable logic devices

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีโดยได้รับความช่วยเหลือจากหลายท่าน คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ. อัครพล ตรีรัตน์ และ อ. ศรวัฒน์ ชิวปรีชา ที่ให้คำปรึกษา คำแนะนำทั้งทางทฤษฎีและปฏิบัติ ความอนุเคราะห์เครื่องมือและอุปกรณ์ที่ใช้ในการทดสอบทดลอง ตลอดจนชี้แนะแนวทางในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้น ขอขอบคุณนายมงคล แซ่จัง ที่สละเวลามาให้คำปรึกษาในการทำปริญญานิพนธ์เป็นอย่างดีมาโดยตลอด คณะผู้จัดทำระลึกอยู่เสมอว่าหากไม่ได้รับความช่วยเหลือจากบุคคลที่กล่าวมานั้น รายงานฉบับนี้ก็ไม่สามารถที่จะสำเร็จลุล่วงไปด้วยดี จึงขอขอบพระคุณมา ณ ที่นี้

ท้ายที่สุดนี้คณะผู้จัดทำขอขอบพระคุณบิดา มารดา ที่ให้การอบรมสั่งสอน เลี้ยงดู และให้โอกาสในการศึกษาอย่างเต็มที่ และขอบคุณเพื่อนๆ ที่ให้กำลังใจและช่วยเหลือกันอย่างเต็มที่เสมอมา

นาย สุรชัย แก้วศรีนาค
นาย อรรถพล วีระนพนันท์
นาย อัศนัย นิธิโรจนานนท์
12 มีนาคม 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 เนื้อหาของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เวฟเลตเบื้องต้น	3
2.1.1 ระบบของเวฟเลต	3
2.1.2 คุณสมบัติพิเศษของเวฟเลต	4
2.1.3 การกระจายเวฟเลตและการแปลงเวฟเลต	4
2.2 การแปลงเวฟเลตแบบไม่ต่อเนื่อง	5
2.3 ทฤษฎีมัลติรีโซลูชัน	6
2.3.1 สเกลลิ่งฟังก์ชัน	6
2.3.2 เวฟเลตฟังก์ชัน	7
2.4 ฟิลเตอร์แบงค์และการแปลงเวฟเลต	8
2.4.1 หลักการวิเคราะห์	8
2.4.1.1 การกรองสัญญาณและการสุ่มค่าตัวอย่าง	9
2.4.2 หลักการสังเคราะห์	13
2.4.2.1 การกรองสัญญาณและการเพิ่มค่าตัวอย่าง	14
2.5 ฟิลเตอร์แบงค์	15
2.5.1 การกำจัดการเกิดการซ้อนทับกันและผลที่ได้จากวงจรกรองความถี่	18
2.5.2 โครงสร้างของวงจรกรองความถี่รูปแบบโดยตรงและรูปแบบโพลีเฟส	19
2.5.2.1 วงจรกรองความถี่ที่มีโครงสร้างแบบโดยตรง	19
2.5.2.2 วงจรกรองความถี่ที่มีโครงสร้างแบบโพลีเฟส	21
2.6 โครงสร้างการแปลงเวฟเลตโดยใช้วิธีการ Lifting scheme	25
2.7 ภาษาวีเอชดีแอล	28
2.7.1 ประวัติความเป็นมาของภาษาวีเอชดีแอล	28
2.7.2 การออกแบบระบบดิจิตอล	29
2.7.3 การออกแบบจากบนลงล่าง	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.8 เอฟพีจีเอ 32
2.8.1 การออกแบบวงจรเชิงเลขด้วยชิพอุปกรณ์เอฟพีจีเอ 33
2.8.2 ปัจจัยที่ทำให้การออกแบบชิพอุปกรณ์เอฟพีจีเอง่ายและสะดวกรวดเร็ว 33
2.9 หลักการเบื้องต้นของโครงสร้างเลขคณิตกระจาย 34
2.9.1 ระบบตัวเลข 34
2.9.1.1 รูปแบบจำนวนโดยตรง 34
2.9.1.2 รูปแบบจำนวนอิงดรรชนี 36
2.9.2 ทฤษฎีเลขคณิตกระจาย 37
2.10 พอร์ตอนุกรม 42
2.10.1 การสื่อสารข้อมูล 42
2.10.1.1 การสื่อสารข้อมูลแบบขนาน 42
2.10.1.2 การสื่อสารข้อมูลแบบอนุกรม 42
2.10.2 การอินเตอร์เฟสตามมาตรฐาน RS-232 43

บทที่ 3 การคำนวณและการสร้าง 45
3.1 การออกแบบตัวแปลงเวฟเลต 45
3.1.1 สัมประสิทธิ์ที่ใช้สำหรับการแปลงเวฟเลตและการแปลงกลับเวฟเลต 45
3.1.2 การแปลงเวฟเลต 3 ระดับ 46
3.1.3 การแปลงกลับเวฟเลต 3 ระดับ 46
3.1.4 ส่วนประกอบของวงจรการแปลงเวฟเลตโดยใช้ Lifting scheme 47
3.1.4.1 วงจรหน่วงเวลา 47
3.1.4.2 วงจรฟอร์เวิร์ดเวลา 48
3.1.4.3 วงจรสุ่มค่าตัวอย่าง 48
3.1.4.4 วงจรเพิ่มค่าตัวอย่าง 48
3.1.4.5 วงจรบวก 49
3.1.4.6 วงจรคูณ 49
3.2 การออกแบบโครงสร้างเลขคณิตกระจาย 50
3.2.1 การประยุกต์ใช้โครงสร้างเลขคณิตกระจายกับการแปลงเวฟเลต ที่มีโครงสร้างแบบ Lifting scheme 50
3.2.2 โครงสร้างเลขคณิตกระจายแบบอนุกรม 51
3.2.2.1 วงจรเลื่อนข้อมูลแบบเข้าขนานออกอนุกรม 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.2.2.2 ตารางเปิดดู 52

3.2.2.3 วงจรบวกและลบ 52

3.2.2.4 วงจรสะสมค่า 53

3.2.3 โครงสร้างเลขคณิตกระจายแบบขนาน 53

3.2.3.1 การสร้างโครงสร้างเลขคณิตกระจายแบบขนาน 53

3.2.3.2 โครงสร้างวงจรกรองเอฟไออาร์โดยการใช้เลขคณิตกระจายแบบขนาน 54

3.3 การออกแบบอุปกรณ์รับส่งข้อมูลผ่านพอร์ทอนุกรม 57

3.3.1 การรับข้อมูลของอุปกรณ์ FPGA ที่ส่งมาจากคอมพิวเตอร์ผ่านพอร์ทอนุกรม 57

3.3.2 การส่งข้อมูลของอุปกรณ์ FPGA ไปยังคอมพิวเตอร์ผ่านพอร์ทอนุกรม 57

บทที่ 4 การทดลองและผลการทดลอง 58

4.1 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ 59

4.1.1 การแปลงเวฟเลต 59

4.1.2 การแปลงกลับเวฟเลต 59

4.1.3 ขั้นตอนการทดลอง 60

4.1.3.1 ผลการแปลงเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณไซน์ 65

4.1.3.2 ผลการแปลงกลับเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณไซน์ 66

4.1.3.3 ผลการแปลงเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม 68

4.1.3.4 ผลการแปลงกลับเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณสี่เหลี่ย 69

4.1.3.5 การแปลงเวฟเลตด้านสังเคราะห์เมื่อสัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม 71

4.2 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรม ModelSim XE III 6.0d 75

4.2.1 วงจรสุ่มค่าตัวอย่าง 75

4.2.2 วงจรเพิ่มค่าตัวอย่าง 76

4.2.3 วงจรบวก 76

4.2.4 วงจรหน่วงเวลา 1 เท่า 77

4.2.5 วงจรหน่วงเวลา 2 เท่า 77

4.2.6 วงจรหน่วงเวลา 3 เท่า 78

4.2.7 วงจรหน่วงเวลา 4 เท่า 78

4.2.8 วงจรหน่วงเวลา 5 เท่า 79

4.2.9 วงจรหน่วงเวลา 2 สัญญาณนาฬิกา 79

4.2.10 วงจรหน่วงเวลา 3 สัญญาณนาฬิกา 80

4.2.11 วงจรหน่วงเวลา 4 สัญญาณนาฬิกา 80

4.2.12 วงจรควบคุมการแปลงเวฟเลต 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

4.2.13 วงจรควบคุมการแปลงกลับเวฟเลต 81

4.3 การจำลองการทำงานวงจรเลขคณิตกระจายด้วยโปรแกรม ModelSim XE III 6.0d 82
4.3.1 การจำลองการทำงานของวงจรเลขคณิตกระจาย 82
4.3.1.1 วงจรบวกสะสมค่า 82
4.3.1.2 วงจรเลื่อนข้อมูล 82
4.3.1.3 หน่วยความจำการแปลงเวฟเลต 1 83
4.3.1.4 หน่วยความจำการแปลงเวฟเลต 2 83
4.3.1.5 หน่วยความจำการแปลงเวฟเลต 3 84
4.3.1.6 หน่วยความจำการแปลงเวฟเลต 4 84
4.3.1.7 หน่วยความจำการแปลงกลับเวฟเลต 1 85
4.3.1.8 หน่วยความจำการแปลงกลับเวฟเลต 2 85
4.3.1.9 หน่วยความจำการแปลงกลับเวฟเลต 3 86
4.3.1.10 หน่วยความจำการแปลงกลับเวฟเลต 4 86
4.3.1.11 วงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต 87
4.3.1.12 วงจรตัดบิตจาก 26 บิตเป็น 14 บิต 87
4.3.1.13 วงจรบวก 26 บิต 88
4.3.1.14 วงจรลบ 26 บิต 88
4.3.1.15 วงจรเลื่อนบิต 1 บิต 89
4.3.1.16 วงจรเลื่อนบิต 2 บิต 89
4.3.1.17 วงจรเลื่อนบิต 3 บิต 90
4.3.1.18 วงจรเลื่อนบิต 4 บิต 90
4.3.1.19 วงจรเลื่อนบิต 5 บิต 91
4.3.1.20 วงจรเลื่อนบิต 6 บิต 91
4.3.1.21 วงจรเลื่อนบิต 7 บิต 92
4.3.1.22 วงจรเลื่อนบิต 8 บิต 92
4.3.1.23 วงจรเลื่อนบิต 9 บิต 93
4.3.1.24 วงจรเลื่อนบิต 10 บิต 93
4.3.1.25 วงจรเลื่อนบิต 11 บิต 94
4.3.1.26 วงจรเลื่อนบิต 12 บิต 94
4.3.1.27 วงจรเลื่อนบิต 13 บิต 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

4.3.2 การเปรียบเทียบการทำงานของโครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน 95
4.3.3 การเปรียบเทียบประสิทธิภาพสำหรับการสร้างจริงระหว่างโครงสร้างเลขคณิต กระจายแบบอนุกรมและขนาน 96
4.3.4 การเปรียบเทียบการทำงานของตัวแปลงและตัวแปลงกลับเวฟเลตที่ใช้โครงสร้าง เลขคณิตกระจายแบบอนุกรมและขนาน 97
4.4 การเปรียบเทียบเพื่อหาค่าผิดพลาดของการแปลงเวฟเลตที่ได้จากการจำลองการทำงาน 98
4.4.1 การเปรียบเทียบการแปลงเวฟเลตเมื่ออินพุตเป็นสัญญาณไซน์ 98
4.4.2 การเปรียบเทียบการแปลงเวฟเลตเมื่ออินพุตเป็นสัญญาณเสียง 101
4.5 การจำลองการทำงานวงจรควบคุมการรับส่งข้อมูลใช้โปรแกรม ModelSim XE III 6.0d 104
4.5.1 วงจรหารความถี่ 104
4.5.2 วงจรมัลติเพล็กซ์ 104
4.5.3 วงจรตัดความยาวบิตข้อมูล 105
4.5.4 วงจรเพิ่มความยาวบิตข้อมูล 105
4.5.5 วงจรรับข้อมูลจากพอร์ทอนุกรม 106
4.5.6 วงจรส่งข้อมูลเข้าพอร์ทอนุกรม 106
4.5.7 วงจรเลือกการอ่านเขียนหน่วยความจำ 1 107
4.5.8 วงจรเลือกการอ่านเขียนหน่วยความจำ 2 108
4.5.9 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 1 (วงจรควบคุม 1) 109
4.5.10 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 1 (วงจรควบคุม 2) 110
4.5.11 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 2 (วงจรควบคุม 3) 110
4.5.12 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 2 (วงจรควบคุม 4) 111
4.5.13 วงจรควบคุมหลัก 112
4.6 การทดลองตัวแปลงเวฟเลตที่สร้างจริงลงอุปกรณ์ FPGA และแสดงผล ผ่านจอคอมพิวเตอร์ 112

บทที่ 5 บทวิจารณ์และบทสรุป 116
ภาคผนวก
กิตติกรรมประกาศ
หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ลักษณะของคลื่นเปรียบเทียบกับเวฟเลต	3
รูปที่ 2.2 การเลื่อนตำแหน่งและการปรับระดับของเวฟเลต	5
รูปที่ 2.3 การแบ่งพื้นที่ออกเป็นส่วนย่อยโดยการใช้สเกลลิ่งฟังก์ชัน	6
รูปที่ 2.4 สเกลลิ่งฟังก์ชันและเวฟเลต	7
รูปที่ 2.5 ตัวสุ่มค่าตัวอย่าง (Down Sampler or Decimator)	9
รูปที่ 2.6 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มตัวอย่างด้วยความถี่ f_s	9
รูปที่ 2.7 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มตัวอย่างด้วยความถี่ $2f_s$	10
รูปที่ 2.8 สเปกตรัมของสัญญาณอินพุตก่อนการลดค่าตัวอย่าง	11
รูปที่ 2.9 สเปกตรัมของสัญญาณเอาท์พุตหลังจากผ่านการลดค่าตัวอย่าง	11
รูปที่ 2.10 การแตกกิ่งก้านสาขาแบบสองแถบของการแปลงเวฟเลต	12
รูปที่ 2.11 การแตกกิ่งก้านสาขาแบบสองแถบสองขั้นของการแปลงเวฟเลต	12
รูปที่ 2.12 การแตกกิ่งก้านสาขาแบบสองแถบสามขั้นของการแปลงเวฟเลต	12
รูปที่ 2.13 แถบความถี่ของการแปลงเวฟเลต	13
รูปที่ 2.14 สเปกตรัมของสัญญาณอินพุตก่อนการเพิ่มค่าตัวอย่าง	14
รูปที่ 2.15 สเปกตรัมของสัญญาณเอาท์พุตเมื่อผ่านการเพิ่มค่าตัวอย่าง	14
รูปที่ 2.16 แสดงการแตกกิ่งก้านสาขาแบบสองแถบของการแปลงกลับเวฟเลต	15
รูปที่ 2.17 แสดงการแตกกิ่งก้านสาขาแบบสองแถบสองขั้นของการแปลงกลับเวฟเลต	15
รูปที่ 2.18 ฟิลเตอร์แบงค์	15
รูปที่ 2.19 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำและสูง	16
รูปที่ 2.20 การสลับเครื่องหมายในการหาค่าสัมประสิทธิ์	18
รูปที่ 2.21 ความสัมพันธ์ระหว่างสัมประสิทธิ์ทั้งการแปลงและการแปลงกลับ	19
รูปที่ 2.22 ฟิลเตอร์แบงค์เชิงตั้งฉากที่มีสัมประสิทธิ์ 4 ค่า	19
รูปที่ 2.23 บล็อกไดอะแกรมของวงจรกรองความถี่โครงสร้างโดยตรง	19
รูปที่ 2.24 บล็อกไดอะแกรมการสร้างจริงของโพลีเฟสของฟังก์ชันการส่งผ่าน	22
รูปที่ 2.25 การแยกโครงสร้างออกเป็นพจน์คี่และพจน์คู่	22
รูปที่ 2.26 บล็อกไดอะแกรมก่อนการเปลี่ยนโครงสร้าง	23
รูปที่ 2.27 บล็อกไดอะแกรมหลังการเปลี่ยนโครงสร้าง	23
รูปที่ 2.28 บล็อคไดอะแกรมของโครงสร้างแบบโพลีเฟส	23
รูปที่ 2.29 โครงสร้างของฟิลเตอร์แบบโพลีเฟสในการแปลง	24
รูปที่ 2.30 โครงสร้างของฟิลเตอร์แบบโพลีเฟสในส่วนการแปลงกลับ	24
รูปที่ 2.31 รูปแบบของการแปลงและการแปลงกลับของฟิลเตอร์แบงค์	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 2.32 บล็อกไดอะแกรมแทนเมตริกซ์ α 26
รูปที่ 2.33 บล็อกไดอะแกรมแทนเมตริกซ์ β 27
รูปที่ 2.34 บล็อกไดอะแกรมแทนเมตริกซ์ γ 27
รูปที่ 2.35 โครงสร้าง Lifting Scheme ที่แปลงมาจากโพลีเฟสเมตริกซ์ 28
รูปที่ 2.36 โครงสร้างทั่วไปของ Lifting Scheme ด้านการแปลง 28
รูปที่ 2.37 ขั้นตอนการออกแบบระบบดิจิตอล 29
รูปที่ 2.38 การออกแบบระบบเส้นทางข้อมูล 30
รูปที่ 2.39 ขั้นตอนการออกแบบจากบนลงล่าง 31
รูปที่ 2.40 ผังการแบ่งกลุ่มของวงจรรวมเอซิก 32
รูปที่ 2.41 การจัดรูปแบบจำนวนโดยตรงที่ประกอบด้วยบิตจำนวนเต็มและบิตเศษส่วน 34
รูปที่ 2.42 การจัดรูปแบบจำนวนโดยตรงที่มีแต่บิตเศษส่วน 35
รูปที่ 2.43 การจัดรูปแบบจำนวนอิงดรรชนี 36
รูปที่ 2.44 การคูณแบบเลขส่วนเติมเต็มสองโดยใช้เลขคณิตกระจาย 40
รูปที่ 2.45 บล็อกไดอะแกรมรูปแบบการสื่อสารข้อมูลแบบขนาน 42
รูปที่ 2.46 บล็อกไดอะแกรมรูปแบบการสื่อสารข้อมูลแบบอนุกรม 43
รูปที่ 2.47 แสดงลักษณะของคอนเน็กเตอร์แบบ DB-9 44
รูปที่ 3.1 บล็อกไดอะแกรมของการแปลงเวฟเลต 3 ระดับ 46
รูปที่ 3.2 โครงสร้างภายในของการแปลงเวฟเลตโดยใช้ lifting scheme 46
รูปที่ 3.3 บล็อกไดอะแกรมของการแปลงกลับเวฟเลต 3 ระดับ 46
รูปที่ 3.4 โครงสร้างภายในของการแปลงกลับเวฟเลตโดยใช้ lifting scheme 47
รูปที่ 3.5 บล็อกไดอะแกรมของวงจรหน่วงเวลา 47
รูปที่ 3.7 บล็อกไดอะแกรมของวงจรสุ่มค่าตัวอย่าง 48
รูปที่ 3.6 บล็อกไดอะแกรมของวงจรฟอร์เวิร์ดเวลา 48
รูปที่ 3.8 บล็อกไดอะแกรมของวงจรเพิ่มค่าตัวอย่าง 48
รูปที่ 3.9 บล็อกไดอะแกรมของวงจรบวก 49
รูปที่ 3.10 บล็อกไดอะแกรมของวงจรคูณ 49
รูปที่ 3.11 การแปลงเวฟเลตโดยนำหลักการของ DA 50
รูปที่ 3.12 การแปลงกลับเวฟเลตโดยนำหลักการของ DA มาใช้ 51
รูปที่ 3.13 วงจรเลื่อนข้อมูลแบบเข้าขนานออกอนุกรม 51
รูปที่ 3.14 วงจรบวกและลบ 52
รูปที่ 3.15 วงจรสะสมค่า 53
รูปที่ 3.16 โครงสร้างฟิลเตอร์แบงค์ (ก) การแปลงเวฟเลต 53
(ข) การแปลงกลับเวฟเลต 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 3.17 การสร้าง DA โดยใช้วงจรกรองเอฟไอาร์ดาร์บีซี 55
รูปที่ 3.18 การสร้าง DA ที่มีการแบ่งตารางเปิดดูของวงจรกรองเอฟไอาร์ดาร์บีซี 55
รูปที่ 3.19 วงจรกรองเอฟไอาร์ดาร์บีซี แบบ PDA (ก) บิตเดี่ยว (ข) 8 บิต 56
รูปที่ 3.20 บล็อกไดอะแกรมการรับข้อมูลของอุปกรณ์ FPGA ผ่านพอร์ทอนุกรม 57
รูปที่ 3.21 บล็อกไดอะแกรมการส่งข้อมูลของอุปกรณ์ FPGA ผ่านพอร์ทอนุกรม 57
รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของตัวแปลงเวฟเลต 58
รูปที่ 4.1 บล็อกไดอะแกรมของการแปลงเวฟเลต 59
รูปที่ 4.2 บล็อกไดอะแกรมของการแปลงกลับเวฟเลต 60
รูปที่ 4.3 หน้าต่างของแมทแลบส่วนอินเตอร์เฟสของโปรแกรม Wavelet Transform 60
รูปที่ 4.4 หน้าต่าง command ของโปรแกรมแมทแลบ 61
รูปที่ 4.5 คำสั่งที่ใช้เรียกส่วนของ GUI ขึ้นมา 61
รูปที่ 4.6 หน้าต่างของ GUI หลังพิมพ์คำสั่ง wavelet 2 62
รูปที่ 4.7 การเลือกชนิดของ mother wavelet ที่จะใช้ในการแปลงเวฟเลต 62
รูปที่ 4.8 การเลือกอันดับของ mother wavelet 63
รูปที่ 4.9 การเลือกสัญญาณอินพุตที่ใช้ในการแปลงเวฟเลต 63
รูปที่ 4.10 การเปรียบเทียบระหว่างสัญญาณอินพุตกับสัญญาณที่สร้างกลับคืน 64
รูปที่ 4.11 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1 65
รูปที่ 4.12 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2 65
รูปที่ 4.13 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3 66
รูปที่ 4.14 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1 66
รูปที่ 4.15 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 2 67
รูปที่ 4.16 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 3 67
รูปที่ 4.17 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare 67
รูปที่ 4.18 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1 68
รูปที่ 4.19 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.20 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3 69

รูปที่ 4.21 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1 69

รูปที่ 4.22 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 2 70

รูปที่ 4.23 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 3 70

รูปที่ 4.24 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare 70

รูปที่ 4.25 เมื่อสัญญาณอินพุตเป็นสัญญาณเสียง 71

รูปที่ 4.26 การเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่สร้างกลับคืนเมื่อกดปุ่ม Analysis 71

รูปที่ 4.27 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1 72

รูปที่ 4.28 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2 72

รูปที่ 4.29 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3 73

รูปที่ 4.30 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1 73

รูปที่ 4.31 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 2 74

รูปที่ 4.32 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 3 74

รูปที่ 4.33 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare 75

รูปที่ 4.34 ปุ่มที่ใช้ในการฟังเสียงของสัญญาณเสียงอินพุต และสัญญาณเสียงที่สร้างกลับคืน 75

รูปที่ 4.35 วงจรสุ่มค่าตัวอย่าง 75

รูปที่ 4.36 ผลการจำลองการทำงานของวงจรสุ่มค่าตัวอย่าง 75

รูปที่ 4.37 วงจรเพิ่มค่าตัวอย่าง 76

รูปที่ 4.38 ผลการจำลองการทำงานของวงจรเพิ่มค่าตัวอย่าง 76

รูปที่ 4.39 วงจรบวก 76

รูปที่ 4.40 ผลการจำลองการทำงานของวงจรบวก 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.41 วงจรหน่วงเวลา 1 เท่า 77
รูปที่ 4.42 ผลการจำลองการทำงานของวงจรหน่วงเวลา 1 เท่า 77
รูปที่ 4.43 วงจรหน่วงเวลา 2 เท่า 77
รูปที่ 4.44 ผลการจำลองการทำงานของวงจรหน่วงเวลา 2 เท่า 77
รูปที่ 4.45 วงจรหน่วงเวลา 3 เท่า 78
รูปที่ 4.46 ผลการจำลองการทำงานของวงจรหน่วงเวลา 3 เท่า 78
รูปที่ 4.47 วงจรหน่วงเวลา 4 เท่า 78
รูปที่ 4.48 ผลการจำลองการทำงานของวงจรหน่วงเวลา 4 เท่า 78
รูปที่ 4.49 วงจรหน่วงเวลา 5 เท่า 79
รูปที่ 4.50 ผลการจำลองการทำงานของวงจรหน่วงเวลา 5 เท่า 79
รูปที่ 4.51 วงจรหน่วงเวลา 2 สัญญาณนาฬิกา 79
รูปที่ 4.52 ผลการจำลองการทำงานของวงจรหน่วงเวลา 2 สัญญาณนาฬิกา 79
รูปที่ 4.53 วงจรหน่วงเวลา 3 สัญญาณนาฬิกา 80
รูปที่ 4.54 ผลการจำลองการทำงานของวงจรหน่วงเวลา 3 สัญญาณนาฬิกา 80
รูปที่ 4.55 วงจรหน่วงเวลา 4 สัญญาณนาฬิกา 80
รูปที่ 4.56 ผลการจำลองการทำงานของวงจรหน่วงเวลา 4 สัญญาณนาฬิกา 80
รูปที่ 4.57 วงจรควบคุมการแปลงเวฟเลต 81
รูปที่ 4.58 ผลการจำลองการทำงานของวงจรควบคุมการแปลงเวฟเลต 81
รูปที่ 4.59 วงจรควบคุมการแปลงกลับเวฟเลต 81
รูปที่ 4.60 ผลการจำลองการทำงานของวงจรควบคุมการแปลงกลับเวฟเลต 81
รูปที่ 4.61 วงจรบวกสะสมค่า 82
รูปที่ 4.62 ผลการจำลองการทำงานของวงจรวงจรบวกสะสมค่า 82
รูปที่ 4.63 วงจรเลื่อนข้อมูล 82
รูปที่ 4.64 ผลการจำลองการทำงานของวงจรเลื่อนข้อมูล 82
รูปที่ 4.65 หน่วยความจำการแปลงเวฟเลต 1 83
รูปที่ 4.66 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 1 83
รูปที่ 4.67 หน่วยความจำการแปลงเวฟเลต 2 83
รูปที่ 4.68 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 2 83
รูปที่ 4.69 หน่วยความจำการแปลงเวฟเลต 3 84
รูปที่ 4.70 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 3 84
รูปที่ 4.71 หน่วยความจำการแปลงเวฟเลต 4 84
รูปที่ 4.72 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 4 84
รูปที่ 4.73 หน่วยความจำการแปลงกลับเวฟเลต 1 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.74 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 2 85
รูปที่ 4.75 หน่วยความจำการแปลงกลับเวฟเลต 2 85
รูปที่ 4.76 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 2 85
รูปที่ 4.77 หน่วยความจำการแปลงกลับเวฟเลต 3 86
รูปที่ 4.78 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 3 86
รูปที่ 4.79 หน่วยความจำการแปลงกลับเวฟเลต 4 86
รูปที่ 4.80 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 4 86
รูปที่ 4.81 วงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต 87
รูปที่ 4.82 ผลการจำลองการทำงานของวงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต 87
รูปที่ 4.83 วงจรตัดบิตจาก 26 บิตเป็น 14 บิต 87
รูปที่ 4.84 ผลการจำลองการทำงานของวงจรตัดบิตจาก 26 บิตเป็น 14 บิต 88
รูปที่ 4.85 วงจรบวก 26 บิต 88
รูปที่ 4.86 ผลการจำลองการทำงานของวงจรบวก 26 บิต 88
รูปที่ 4.87 วงจรลบ 26 บิต 88
รูปที่ 4.88 ผลการจำลองการทำงานของวงจรลบ 26 บิต 88
รูปที่ 4.89 วงจรเลื่อนบิต 1 บิต 89
รูปที่ 4.90 ผลการจำลองการทำงานของวงจรเลื่อนบิต 1 บิต 89
รูปที่ 4.91 วงจรเลื่อนบิต 2 บิต 89
รูปที่ 4.92 ผลการจำลองการทำงานของวงจรเลื่อนบิต 2 บิต 89
รูปที่ 4.93 วงจรเลื่อนบิต 3 บิต 90
รูปที่ 4.94 ผลการจำลองการทำงานของวงจรเลื่อนบิต 3 บิต 90
รูปที่ 4.95 วงจรเลื่อนบิต 4 บิต 90
รูปที่ 4.96 ผลการจำลองการทำงานของวงจรเลื่อนบิต 4 บิต 90
รูปที่ 4.97 วงจรเลื่อนบิต 5 บิต 91
รูปที่ 4.98 ผลการจำลองการทำงานของวงจรเลื่อนบิต 5 บิต 91
รูปที่ 4.99 วงจรเลื่อนบิต 6 บิต 91
รูปที่ 4.100 ผลการจำลองการทำงานของวงจรเลื่อนบิต 6 บิต 91
รูปที่ 4.101 วงจรเลื่อนบิต 7 บิต 92
รูปที่ 4.102 ผลการจำลองการทำงานของวงจรเลื่อนบิต 7 บิต 92
รูปที่ 4.103 วงจรเลื่อนบิต 8 บิต 92
รูปที่ 4.104 ผลการจำลองการทำงานของวงจรเลื่อนบิต 8 บิต 92
รูปที่ 4.105 วงจรเลื่อนบิต 9 บิต 93
รูปที่ 4.106 ผลการจำลองการทำงานของวงจรเลื่อนบิต 9 บิต 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.107 วงจรเลื่อนบิต 10 บิต 93
รูปที่ 4.108 ผลการจำลองการทำงานของวงจรเลื่อนบิต 10 บิต 93
รูปที่ 4.109 วงจรเลื่อนบิต 11 บิต 94
รูปที่ 4.110 ผลการจำลองการทำงานของวงจรเลื่อนบิต 11 บิต 94
รูปที่ 4.110 วงจรเลื่อนบิต 12 บิต 94
รูปที่ 4.112 ผลการจำลองการทำงานของวงจรเลื่อนบิต 12 บิต 94
รูปที่ 4.113 วงจรเลื่อนบิต 13 บิต 95
รูปที่ 4.114 ผลการจำลองการทำงานของวงจรเลื่อนบิต 13 บิต 95
รูปที่ 4.115 วงจรการทดลองเปรียบเทียบการทำงานของ DA ทั้งสองแบบ 95
รูปที่ 4.116 ผลการทดลองเปรียบเทียบการทำงานของ DA ทั้งสองแบบ 96
รูปที่ 4.117 ผลการทดลองของตัวแปลงเวฟเลทที่ใช้ DA แบบอนุกรม 97
รูปที่ 4.118 ผลการทดลองของตัวแปลงเวฟเลทที่ใช้ DA แบบขนาน 97
รูปที่ 4.119 ผลการทดลองของตัวแปลงกลับเวฟเลทที่ใช้ DA แบบอนุกรม 98
รูปที่ 4.120 ผลการทดลองของตัวแปลงกลับเวฟเลทที่ใช้ DA แบบขนาน 98
รูปที่ 4.121 ผลการแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ 99
รูปที่ 4.122 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (SDA) โดยใช้โปรแกรม ModelSim XE III 6.0d 99
รูปที่ 4.123 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (PDA) โดยใช้โปรแกรม ModelSim XE III 6.0d 100
รูปที่ 4.124 การเปรียบเทียบสัญญาณที่สร้างคืนจากการแปลงเวฟเลตทั้ง 3 วิธีกับสัญญาณต้นฉบับ 100
รูปที่ 4.125 สัญญาณเสียงที่ใช้ทดลอง 101
รูปที่ 4.126 ผลการแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ 102
รูปที่ 4.127 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (SDA) โดยใช้โปรแกรม ModelSim XE III 6.0d 102
รูปที่ 4.128 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (PDA) โดยใช้โปรแกรม ModelSim XE III 6.0d 103
รูปที่ 4.129 การเปรียบเทียบสัญญาณที่สร้างคืนจากการแปลงเวฟเลตทั้ง 3 วิธีกับสัญญาณต้นฉบับ 103
รูปที่ 4.130 วงจรหารความถี่ 104
รูปที่ 4.131 ผลการจำลองการทำงานของวงจรหารความถี่ 104
รูปที่ 4.132 วงจรมัลติเพล็กซ์ 104
รูปที่ 4.133 ผลการจำลองการทำงานของวงจรมัลติเพล็กซ์ 105
รูปที่ 4.134 วงจรตัดความยาวบิตข้อมูล 105
รูปที่ 4.135 ผลการจำลองการทำงานของวงจรวงจรตัดความยาวบิตข้อมูล 105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.136 วงจรเพิ่มความยาวบิตข้อมูล 105
รูปที่ 4.137 ผลการจำลองการทำงานของวงจรเพิ่มความยาวบิตข้อมูล 106
รูปที่ 4.138 วงจรรับข้อมูลจากพอร์ทอนุกรม 106
รูปที่ 4.139 ผลการจำลองการทำงานของวงจรรับข้อมูลจากพอร์ทอนุกรม 106
รูปที่ 4.140 วงจรส่งข้อมูลเข้าพอร์ทอนุกรม 107
รูปที่ 4.141 ผลการจำลองการทำงานของวงจรส่งข้อมูลเข้าพอร์ทอนุกรม 107
รูปที่ 4.142 วงจรเลือกการอ่านเขียนหน่วยความจำ 1 107
รูปที่ 4.143 ผลการจำลองการทำงานของวงจรเลือกการอ่านเขียนหน่วยความจำ 1 108
รูปที่ 4.144 วงจรเลือกการอ่านเขียนหน่วยความจำ 2 108
รูปที่ 4.145 ผลการจำลองการทำงานของวงจรเลือกการอ่านเขียนหน่วยความจำ 2 109
รูปที่ 4.146 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 1 109
รูปที่ 4.147 ผลการจำลองการทำงานของวงจรควบคุมการเขียนหน่วยความจำชุดที่ 1 109
รูปที่ 4.148 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 1 110
รูปที่ 4.149 ผลการจำลองการทำงานของวงจรควบคุมการอ่านหน่วยความจำชุดที่ 1 110
รูปที่ 4.150 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 2 110
รูปที่ 4.151 ผลการจำลองการทำงานของวงจรควบคุมการเขียนหน่วยความจำชุดที่ 2 111
รูปที่ 4.152 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 2 111
รูปที่ 4.153 ผลการจำลองการทำงานของวงจรควบคุมการอ่านหน่วยความจำชุดที่ 2 111
รูปที่ 4.154 วงจรควบคุมหลัก 112
รูปที่ 4.155 ผลการจำลองการทำงานของวงจรควบคุมหลัก 112
รูปที่ 4.156 สัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม 113
รูปที่ 4.157 สัญญาณเอาท์พุตของสัญญาณสี่เหลี่ยมที่สร้างกลับคืน 113
รูปที่ 4.158 สัญญาณอินพุตเป็นสัญญาณรูปไซน์ 114
รูปที่ 4.159 สัญญาณเอาท์พุตของสัญญาณรูปไซน์ที่สร้างกลับคืน 114
รูปที่ 4.160 การเปรียบเทียบระหว่างสัญญาณเสียงอินพุตกับสัญญาณเสียงที่สร้างกลับคืน 115
รูปที่ 4.161 อุปกรณ์ FPGA ขณะทดสอบการทำงานจริง 115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์

ในปัจจุบันผู้คนมีการแลกเปลี่ยนข้อมูลข่าวสารในชีวิตประจำวันเป็นอันมาก ซึ่งข้อมูลข่าวสารโดยส่วนใหญ่แล้วจะอยู่ในรูปแบบของดิจิตอลทำให้สามารถจัดเก็บลงในอุปกรณ์หน่วยความจำได้อย่างสะดวก แต่เมื่อเวลาผ่านไปข้อมูลที่จัดเก็บเริ่มมีขนาดที่ใหญ่ขึ้น อุปกรณ์ที่ใช้เก็บข้อมูลจึงต้องเพิ่มขนาดความจุตามไปด้วย ทำให้เกิดค่าใช้จ่ายในการจัดหาอุปกรณ์ดังกล่าว เพื่อเป็นการลดความสิ้นเปลืองนี้จึงมีการพัฒนาวิธีการบีบอัดข้อมูลเพื่อลดขนาดของข้อมูลให้เล็กลงที่สุด โดยที่ยังคงสามารถแปลงกลับคืนมาเป็นข้อมูลเดิมได้อย่างถูกต้อง

วิธีการบีบอัดข้อมูลวิธีหนึ่งที่นักวิจัยทั่วโลกให้ความสนใจได้แก่ เวฟเลต (Wavelet) ซึ่งเป็นวิธีการที่มีประสิทธิภาพเป็นอย่างมาก กล่าวคือเป็นวิธีการที่สามารถบีบอัดข้อมูลให้มีขนาดที่เล็กลงได้มาก และสามารถสร้างข้อมูลกลับคืนได้อย่างสมบูรณ์ (Perfect Reconstruction) ซึ่งมีการนำประยุกต์ใช้งานอย่างกว้างขวาง เช่น การบีบอัดภาพแบบ JPEG การบีบอัดสัญญาณที่ส่งมาจากดาวเทียม

จากที่กล่าวข้างต้นจึงเป็นที่มาของปริญญานิพนธ์ฉบับนี้ ซึ่งจัดทำขึ้นเพื่อศึกษาหลักการและวิธีการประมวลผลสัญญาณของเวฟเลต เพื่อนำมาออกแบบและสร้างตัวแปลงเวฟเลตซึ่งทำหน้าที่บีบอัดสัญญาณอย่างมีประสิทธิภาพ นอกจากนี้การออกแบบและการสร้างยังมุ่งเน้นในด้านความเร็วของการประมวลผลของอุปกรณ์เป็นสำคัญ เพื่อลดระยะเวลาที่ใช้ในการบีบอัดข้อมูล

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 ศึกษาหลักการบีบอัดข้อมูลโดยวิธีการแปลงเวฟเลต (Wavelet Transform)

1.2.2 ศึกษาหลักการของวงจรกรองความถี่ที่มีโครงสร้างแบบลิฟท์ติ้ง (Lifting Scheme)

1.2.3 ศึกษาหลักการของเลขคณิตกระจายแบบอนุกรมและขนาน เพื่อนำมาประยุกต์ใช้ในการออกแบบวงจรดิจิตอล

1.2.4 จำลองการทำงานของวิธีการแปลงเวฟเลตที่มีโครงสร้างแบบลิฟท์ติ้ง โดยการออกแบบวงจรดิจิตอลที่เขียนด้วยภาษาวีเอชดีแอล

1.2.5 สร้างตัวแปลงเวฟเลตที่มีโครงสร้างแบบลิฟท์ติ้ง และใช้เลขคณิตกระจายแบบขนานโดยใช้อุปกรณ์ลอจิกที่โปรแกรมได้ (FPGA)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้นำเสนอการออกแบบและการสร้างตัวแปลงเวฟเลตเพื่อใช้ในการบีบอัดข้อมูล โดยตัวแปลงเวฟเลตจะใช้โครงสร้างแบบลิฟท์ติ้ง และเลขคณิตกระจายแบบขนานมาประยุกต์ใช้ การออกแบบจะใช้โปรแกรมแมทแลบ (MATLAB) และโปรแกรม Xilinx-Project Navigator ซึ่งใช้ภาษาวีเอชดีแอลเขียนบรรยายพฤติกรรมการทำงานของกระบวนการแปลงเวฟเลต และการจำลองการทำงานจะใช้โปรแกรมแมทแลบและโปรแกรม ModelSim XE III โดยการออกแบบตัวแปลงเวฟเลตที่มีโครงสร้างแบบลิฟท์ติ้งด้วยภาษาวีเอชดีแอลจะใช้ทั้งเลขคณิตกระจายแบบอนุกรมและแบบขนาน นำผลการจำลองการทำงานจากทั้งสองแบบมาเปรียบเทียบกัน รวมทั้งเปรียบเทียบกับผลที่ได้จากการจำลองการทำงานด้วยโปรแกรมแมทแลบ และใช้โปรแกรมที่เขียนจากภาษาวีเอชดีแอลมาโปรแกรมลงอุปกรณ์ FPGA

1.4 เนื้อหาของปริญญานิพนธ์

บทที่ 2 ทฤษฎีและหลักการของการแปลงเวฟเลต วงจรกรองความถี่ ความสัมพันธ์ของทฤษฎีทั้งสอง โครงสร้างของวงจรกรองความถี่แบบโพลีเฟส หลักการลิฟท์ติ้ง วิธีการแปลงเวฟเลตโดยใช้โครงสร้างแบบลิฟท์ติ้ง โครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน การนำมาประยุกต์ใช้กับโครงสร้างแบบลิฟท์ติ้ง ภาษาวีเอชดีแอลและอุปกรณ์ FPGA

บทที่ 3 การคำนวณและการสร้าง ประกอบด้วยการออกแบบส่วนประกอบของตัวแปลงเวฟเลต การออกแบบโครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน และการออกแบบส่วนรับส่งข้อมูลผ่านพอร์ทอนุกรม

บทที่ 4 ผลการทดลอง ประกอบด้วยผลการจำลองการทำงานจากโปรแกรมแมทแลบ โปรแกรม ModelSim ผลการทดสอบตัวแปลงเวฟเลตที่สร้างจริงลงอุปกรณ์ FPGA

บทที่ 5 บทวิจารณ์และบทสรุป

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

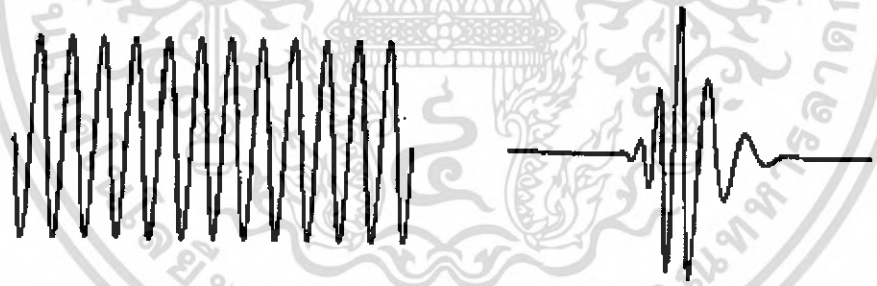
บทที่ 2
ทฤษฎีและหลักการ

2.1 เวฟเลตเบื้องต้น

เมื่อกล่าวถึงคลื่น (wave) จะมีลักษณะเป็นสัญญาณที่มีการเปลี่ยนแปลงอยู่ตลอดเวลาหรือที่เรียกว่าการออสซิเลต (oscillate) เช่น สัญญาณรูปไซน์ (sinusoid) ซึ่งจะใช้การวิเคราะห์โดยการแปลงฟูเรียร์(Fourier transform) คลื่นโดยทั่วไปจะมีการกระจายพลังงานอย่างไม่มีที่สิ้นสุดเนื่องจากสัญญาณเปลี่ยนแปลงอยู่ตลอดเวลา ถ้าเทียบกับเวฟเลต (wavelet) เปรียบเสมือนคลื่นเล็กๆ ที่เกิดขึ้นชั่วขณะหนึ่ง มีการเปลี่ยนแปลงแบบไม่คงที่ มีการแปรผันตามเวลา และมีค่าพลังงานรวมกันอยู่ในช่วงใดช่วงหนึ่ง สามารถวิเคราะห์ได้ทั้งในเทอมของเวลาและความถี่ในเวลาเดียวกัน โดยที่การกระจายเวฟเลต (wavelet expansion) สามารถเขียนให้อยู่ในรูปสมการที่ 2.1 ดังนี้

$$f(t)=\sum_{k}\sum_{j}a_{j,k}\psi_{j,k}(t) \tag{2.1}$$

โดยที่ $a_{j,k}$ แทนสัมประสิทธิ์การกระจาย และ $\psi_{j,k}(t)$ จะแทนฟังก์ชันการกระจายของเวฟเลต โดยที่เซตของสัมประสิทธิ์การกระจายนี้ว่า การแปลงเวฟเลตแบบไม่ต่อเนื่อง (Discrete Wavelet Transform) และสามารถแปลงกลับได้ตามสมการข้างต้น



รูปที่ 2.1 ลักษณะของคลื่นเปรียบเทียบกับเวฟเลต

2.1.1 ระบบของเวฟเลต (Wavelet System) มีคุณสมบัติโดยทั่วไป ดังนี้

1. ระบบของเวฟเลตจะมีลักษณะเป็นบล็อกที่สร้างขึ้น (building blocks) หรือสามารถแสดงเป็นสัญญาณหรือฟังก์ชันได้ และสามารถกระจายให้อยู่ในสองมิติได้
2. การกระจายของเวฟเลตทำให้สัญญาณมีการจำกัด (localization)อยู่ในช่วงเวลาและความถี่หนึ่ง ทำให้พลังงานของสัญญาณส่วนใหญ่สามารถแทนด้วยสัมประสิทธิ์การกระจายเพียงไม่กี่ตัว
3. การคำนวณสัมประสิทธิ์จากสัญญาณสามารถทำได้อย่างมีประสิทธิภาพ (efficiently)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกระจายแบบเวฟเลตจะแปลงให้อยู่ในสองมิติ ทำให้มีการจำกัดสัญญาณให้อยู่ในเทอมของเวลาและความถี่ ดังนั้นเวฟเลตจึงสามารถแสดงตำแหน่งของสัญญาณให้อยู่ในรูปของเวลาและความถี่ได้พร้อมๆกัน เช่นระดับเสียงดนตรี คือ โน้ตตัวหนึ่งสามารถบอกได้ทั้งระดับของเสียง (Tones) และความถี่ของเสียง (frequencies)

2.1.2 คุณสมบัติพิเศษของเวฟเลต (Specific Characteristics of Wavelet Systems)

1. เวฟเลตทุกประเภทสามารถสร้างได้จากสเกลลิ่งฟังก์ชันหรือเวฟเลตเพียงตัวเดียว โดยการแบ่งระดับ (scaling) และการเลื่อนตำแหน่ง (translation)โดยที่เรียกเวฟเลตต้นกำเนิดนี้ว่า เวฟเลตแม่ (Mother wavelet) หรือ $\psi(t)$ โดยที่ j จะแทนระดับต่างของสัญญาณ และ k จะแทนการเลื่อนตำแหน่ง ดังสมการที่ 2.2

$$\psi_{j,k}(t)=2^{j/2}\psi\left(2^{j}t-k\right) \tag{2.2}$$

2. เวฟเลตเกือบทุกแบบสามารถใช้เงื่อนไขมัลติรีโซลูชัน (multiresolution) หมายความว่าถ้าสัญญาณใดสัญญาณหนึ่งสามารถแสดงให้อยู่ในเทอมของฟังก์ชันสเกลลิ่ง (scaling function) $\varphi(t-k)$ ได้ สัญญาณนี้ก็สามารถแสดงให้อยู่ในรูป $\varphi(2t-k)$ ได้เช่นกัน หรืออีกนัยหนึ่งคือการกระจายของเวฟเลตจะแบ่งสัญญาณเดิมออกเป็นส่วนย่อย ๆ ได้

3. สัมประสิทธิ์การกระจายในส่วนย่อย ๆ สามารถคำนวณได้จากสัมประสิทธิ์ของส่วนที่ใหญ่กว่าตามวิธีการแตกกิ่งก้านสาขาหรือฟิลเตอร์แบงค์ (filter bank) ทำให้การคำนวณสัมประสิทธิ์การกระจายมีประสิทธิภาพมากยิ่งขึ้น

การวิเคราะห์เวฟเลตจะเหมาะสมกับสัญญาณที่เกิดขึ้นชั่วขณะหนึ่ง ซึ่งจากลักษณะจำเพาะอยู่เฉพาะส่วนของเวฟเลต ทำให้สัมประสิทธิ์การกระจายมีค่าน้อย ซึ่งจะเป็นประโยชน์กับการนำไปประยุกต์ใช้งาน

2.1.3 การกระจายเวฟเลตและการแปลงเวฟเลต (Wavelet expansions and wavelet transform)

สามารถพิสูจน์ได้ว่ามีประสิทธิภาพและสามารถนำไปใช้งานได้จริงในทุกสัญญาณตามเหตุผลดังต่อไปนี้

1. ขนาดของสัมประสิทธิ์การกระจายของเวฟเลต $a_{j,k}$ มีการลดลงอย่างรวดเร็วตามระดับของสัญญาณที่เพิ่มขึ้นตามพจน์ j และ k ดังนั้นจึงเหมาะแก่การใช้งานในด้านการบีบอัดข้อมูลภาพ (image compression) การลดสัญญาณรบกวน (denoising) และการตรวจจับสัญญาณ (detection)

2. การกระจายเวฟเลตทำให้ได้รูปร่างลักษณะของสัญญาณที่มีความถูกต้องและเฉพาะมากกว่าเนื่องจากจะทำให้ได้สัญญาณที่มีความจำเพาะมากกว่าและง่ายต่อการอธิบาย นอกจากนี้การแยกสัญญาณของเวฟเลตจะสามารถแยกได้ทั้งในเทอมของเวลาและความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เวฟเลตสามารถปรับตัวให้เข้ากับระบบหรือสัญญาณได้หลายรูปแบบ จึงสามารถนำไปประยุกต์ใช้งานได้หลายอย่าง

4. การคำนวณและการแปลงเวฟเลตเหมาะแก่การใช้งานในระบบดิจิตอลและคอมพิวเตอร์ เนื่องจากมีเพียงการคูณและการบวกเท่านั้น

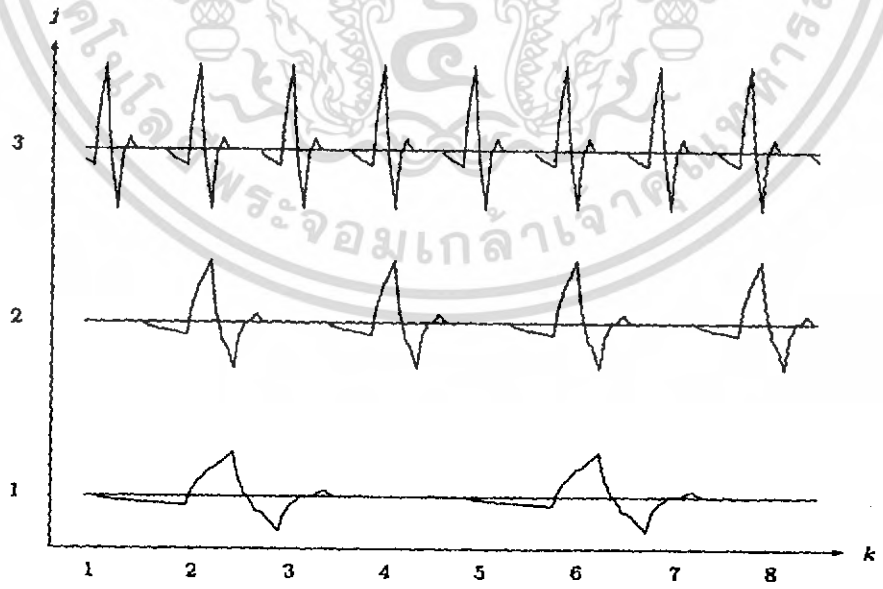
2.2 การแปลงเวฟเลตแบบไม่ต่อเนื่อง (The Discrete Wavelet Transform)

จุดประสงค์ของการแปลงเวฟเลต คือ การกระจายสัญญาณออกเป็นเซตของฟังก์ชันๆหนึ่ง โดยที่สามารถกระจายให้อยู่ในเทอมของเวลาและความถี่ ดังสมการที่ 2.3

$$f(t) = \sum_{j,k} a_{j,k} 2^{j/2} \psi\left(2^j t - k\right) \tag{2.3}$$

โดยที่ $a_{j,k}$ เป็นเซตของสัมประสิทธิ์การกระจายที่มีสองมิติ เรียกว่า การแปลงเวฟเลตแบบไม่ต่อเนื่อง หรือ Discrete wavelet transforms (DWT) จะเห็นได้ว่าการกระจายแบบนี้มีตัวระบุตำแหน่งอยู่สองตัวคือ j และ k โดยที่ตัวที่ใช้ในการเลื่อนตำแหน่งหรือ translation คือ k และตัวที่ใช้ในการแบ่งระดับหรือ scaling คือ j

เป้าหมายของการกระจายสัญญาณหรือฟังก์ชันทั่วไป คือ การใช้ประโยชน์จากสัมประสิทธิ์การกระจายให้คุ้มค่ามากที่สุด หรือพยายามให้ตัวสัมประสิทธิ์มีข้อมูลของสัญญาณเดิมให้มากที่สุด และพยายามให้จำนวนของสัมประสิทธิ์มีน้อยที่สุด หรือเป็นศูนย์มากที่สุด ซึ่งเหมาะแก่การใช้งานด้านการบีบอัดข้อมูล การลดสัญญาณรบกวน



รูปที่ 2.2 การเลื่อนตำแหน่งและการปรับระดับของเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทฤษฎีมัลติรีโซลูชัน (Multiresolution)

การวิเคราะห์โดยการแบ่งย่อยออกเป็นหลายส่วน สามารถอธิบายให้อยู่ในรูปของพื้นที่ย่อยๆที่เป็นสับเซตของพื้นที่ใหญ่ได้ดังนี้ โดยที่ L^2 จะแทนพื้นที่ทั้งหมดของสัญญาณ

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2 \tag{2.4}$$

หรือ

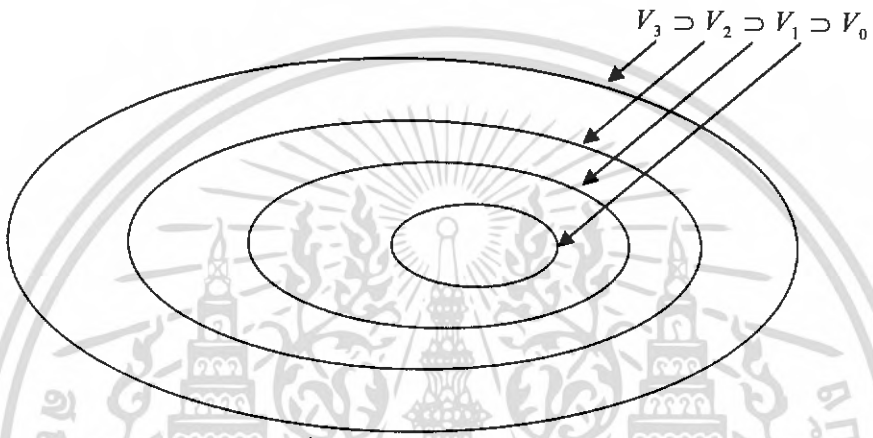
$$V_j \subset V_{j+1} \tag{2.5}$$

โดยที่

$$V_{-\infty} = \{0\} \qquad \text{และ} \qquad V_{\infty} = L^2 \tag{2.6}$$

และ

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1} \tag{2.7}$$



รูปที่ 2.3 การแบ่งพื้นที่ออกเป็นส่วนย่อยโดยการใช้สเกลลิ่งฟังก์ชัน

จากที่กล่าวมาข้างต้นสามารถหาค่าของสเกลลิ่งฟังก์ชันได้จาก

$$\varphi(t) = \sum_n h(n)\sqrt{2}\varphi(2t-n) \tag{2.8}$$

โดยที่ $h(n)$ หมายถึงสัมประสิทธิ์ของฟังก์ชันสเกลลิ่ง ค่า $\sqrt{2}$ จะรักษาระดับมาตรฐานของฟังก์ชันไว้

2.3.1 สเกลลิ่งฟังก์ชัน (The Scaling Function)

เป็นวิธีการหนึ่งในวิธีมัลติรีโซลูชัน โดยการแบ่งสัญญาณออกเป็นส่วนย่อย หรือเป็นระดับแล้วจึงอธิบายเวฟเลตในเทอมของสเกลลิ่งฟังก์ชัน ดังสมการที่ 2.9

$$f(t) = \sum_k a_k \varphi_k(t) \qquad \text{โดยที่} \quad f(t) \in V_0 \tag{2.9}$$

การเพิ่มขนาดของส่วนย่อยนี้สามารถทำได้โดยการเปลี่ยนค่าระดับในสเกลลิ่งฟังก์ชัน ซึ่งสามารถแสดงให้อยู่ในสองมิติ คือ การสเกลลิ่งและการทรานสเลชัน ได้ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k) \tag{2.10}$$

โดยที่ ถ้า $j>0$ จะทำให้มีการแบ่งมากขึ้น ทำให้ $\varphi_{j,k}(t)$ มีขนาดแคบหรือเล็กลง และมีการเลื่อนตำแหน่งที่น้อยลง ดังนั้นจะได้รายละเอียดดีขึ้น ในทางตรงกันข้ามถ้า $j<0$ ทำให้ $\varphi_{j,k}(t)$ จะขนาดกว้างขึ้นและเลื่อนตำแหน่งมากขึ้น สรุปได้ว่าในส่วนสเกลลิ่งฟังก์ชันจะทำให้ได้ข้อมูลแบบหยาบ

2.3.2 เวฟเลตฟังก์ชัน (The Wavelet Function)

เป็นอีกขั้นตอนหนึ่งในวิธีมัลติรีโซลูชัน โดยการกำหนดค่าความแตกต่างระหว่างสเกลลิ่งฟังก์ชันในแต่ละระดับ ดังนั้นจึงจะอธิบายให้อยู่ในเทอมของสเกลลิ่งฟังก์ชัน ข้อได้เปรียบของการทำสเกลลิ่งฟังก์ชันและเวฟเลตคือ การเป็นฟังก์ชันเชิงตั้งฉากซึ่งกันและกัน ทำให้ง่ายต่อการหาสัมประสิทธิ์ของแต่ละชุด โดยจะแทนเวฟเลตนี้ด้วยสัญญลักษณ์ W_j สามารถอธิบายได้ดังนี้

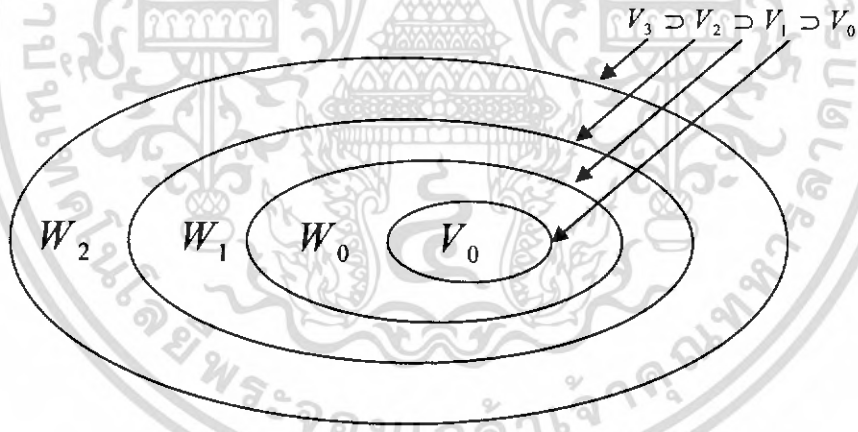
$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2 \tag{2.11}$$

ดังนั้น
$$V_1 = V_0 \oplus W_0 \tag{2.12}$$

ส่วนต่อไปคือ
$$V_2 = V_0 \oplus W_0 \oplus W_1 \tag{2.13}$$

สามารถอธิบายให้อยู่ในรูปทั่วไปได้ดังนี้

$$L^2 = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots \tag{2.14}$$



รูปที่ 2.4 สเกลลิ่งฟังก์ชันและเวฟเลต

จากที่อธิบายมาข้างต้น สามารถหาค่าเวฟเลตฟังก์ชันได้ดังนี้

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\varphi(2t - n) \tag{2.15}$$

โดยที่ $h_1(n)$ หมายถึงสัมประสิทธิ์ของฟังก์ชันเวฟเลต ค่า $\sqrt{2}$ จะรักษาระดับมาตรฐานของฟังก์ชันไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ฟิลเตอร์แบงค์และการแปลงเวฟเลต (Filter Banks and The Discrete Wavelet Transform)

ในหัวข้อนี้จะอธิบายถึงการแปลงเวฟเลต (Wavelet transform) ที่สามารถคำนวณได้จากฟิลเตอร์แบงค์ (filter bank) โดยประเด็นหลักจะอยู่ที่ความสัมพันธ์กันระหว่างการแปลงเวฟเลตและฟิลเตอร์แบงค์

2.4.1 หลักการวิเคราะห์ (Analysis-From Fine Scale to Coarse Scale)

การที่หาสัมประสิทธิ์ของการแปลงเวฟเลต จะหาได้จากความสัมพันธ์ของสัมประสิทธิ์การกระจายระหว่างระดับต่ำและระดับสูง หรืออาจกล่าวได้ว่าจากระดับที่มีความละเอียดไปสู่ระดับหยาบ โดยสามารถคำนวณได้จากสมการดังต่อไปนี้

$$\varphi(t)=\sum_n h(n)\sqrt{2}\varphi(2t-n) \tag{2.16}$$

เปลี่ยนระดับและเลื่อนตำแหน่ง โดยการแทนค่า $t=2^j t-k$

$$\varphi(2^j t-k)=\sum_n h(n)\sqrt{2}\varphi(2(2^j t-k)-n)=\sum_n h(n)\sqrt{2}\varphi(2^{j+1}t-2k-n) \tag{2.17}$$

แทนค่า $m=2k+n$

$$\varphi(2^j t-k)=\sum_m h(m-2k)\sqrt{2}\varphi(2^{j+1}t-m) \tag{2.18}$$

ถ้าแทนพจน์ที่เกิดจากการกระจายด้วย V_j

$$V_j=Span\left\{2^{j/2}\varphi(2^j t-k)\right\} \tag{2.19}$$

ดังนั้น จะได้ว่า

$$f(t)\in V_{j+1}\Rightarrow f(t)=\sum_k c_{j+1}(k)2^{(j+1)/2}\varphi(2^{j+1}t-k) \tag{2.20}$$

จาก $V_{j+1}=V_j\oplus W_j$

$$f(t)=\sum_k c_j(k)2^{j/2}\varphi(2^j t-k)+\sum_k d_j(k)2^{j/2}\psi(2^j t-k) \tag{2.21}$$

สามารถหาค่าสัมประสิทธิ์ของสเกลลิ่งฟังก์ชันได้ดังนี้

$$c_j(k)=\sum_m h(m-2k)c_{j+1}(m) \tag{2.22}$$

และจากความสัมพันธ์สามารถหาค่าสัมประสิทธิ์ของเวฟเลตได้ดังนี้

$$d_j(k)=\sum_m h_1(m-2k)c_{j+1}(m) \tag{2.23}$$

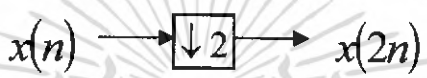
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 การกรองสัญญาณและการสุ่มค่าตัวอย่าง (Filtering and Down-Sampling or Decimating)

จากหลักการของการประมวลผลสัญญาณดิจิตอล การกรองสัญญาณสามารถทำได้โดยการคอนโวลูชั่นระหว่างสัญญาณอินพุตกับสัมประสิทธิ์ของวงจรกรองความถี่ (Filter coefficients or impulse response) โดยถ้าสมมติให้อินพุตคือ $x(n)$ และสัมประสิทธิ์ของวงจรกรองความถี่คือ $h(n)$ จะได้เอาต์พุตคือ $y(n)$ ดังสมการ

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k) \tag{2.24}$$

การสุ่มค่าตัวอย่างจะเป็นการนำอินพุต $x(n)$ สร้างออกมาเป็นเอาต์พุต $y(n)=x(2n)$ ดังรูป

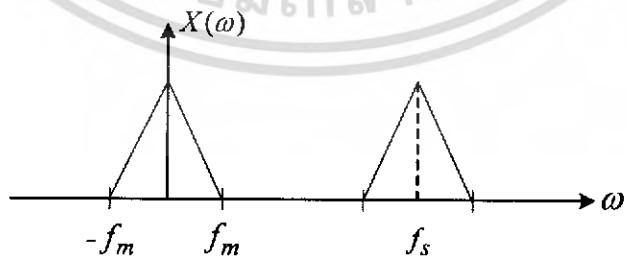


รูปที่ 2.5 ตัวสุ่มค่าตัวอย่าง (Down Sampler or Decimator)

จาก $X(\omega)=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n}$

$y(n)=x(2n)$; $Y(\omega)=\sum_{n=-\infty}^{\infty}x(2n)e^{-j\omega n}$ (2.25)

ถ้าความถี่สูงสุดของสัญญาณเป็น f_m และสุ่มตัวอย่างด้วยความถี่ f_s จะไม่ให้เกิดเอเลียสซิ่ง $f_s \geq 2f_m$ สุ่มตัวอย่างด้วย f_s จะได้สเปกตรัมของสัญญาณเป็น

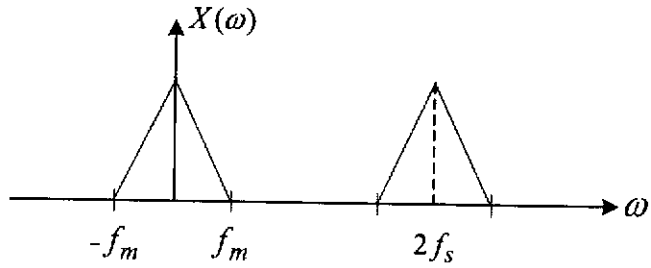


รูปที่ 2.6 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มตัวอย่างด้วยความถี่ f_s

ถ้านอร์มอลไลซ์ $f_s = 2\pi$ $\therefore$ $f_m = \dfrac{2\pi f_m}{f_s}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าสุ่มตัวอย่างด้วย $2f_s$ จะได้สเปกตรัมของสัญญาณเป็น



รูปที่ 2.7 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มตัวอย่างด้วยความถี่ $2f_s$

ถ้านอร์มอลไลซ์ $2f_s = 2\pi \quad \therefore \quad f_m = \dfrac{\pi f_m}{f_s}$

จะเห็นว่าสเปกตรัมของการสุ่มตัวอย่างด้วย $2f_s$ จะได้ $f_m = \dfrac{2\pi f_m}{f_s}$ และถ้าสุ่มตัวอย่าง

ด้วย f_s จะได้ $f_m = \dfrac{\pi f_m}{f_s}$

$\therefore$ การลดค่าตัวอย่างจาก $2f_s$ ไป f_s จะทำให้ f_m เพิ่มเป็น 2 เท่า (เกิดการขยายทางความถี่)

จาก $$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

$$= \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega(2n)} - \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j\omega(2n-1)} \qquad (2.26)$$

และ $$X(\omega-\pi) = \sum_{n=-\infty}^{\infty} x(2n)e^{-j(\omega-\pi)2n} - \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j(\omega-\pi)(2n-1)}$$

$$= \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega(2n)} + \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j\omega(2n-1)} \qquad (2.27)$$

นำสมการ $\dfrac{(2.26)+(2.27)}{2}$: $\dfrac{X(\omega)+X(\omega-\pi)}{2} = \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega(2n)}$

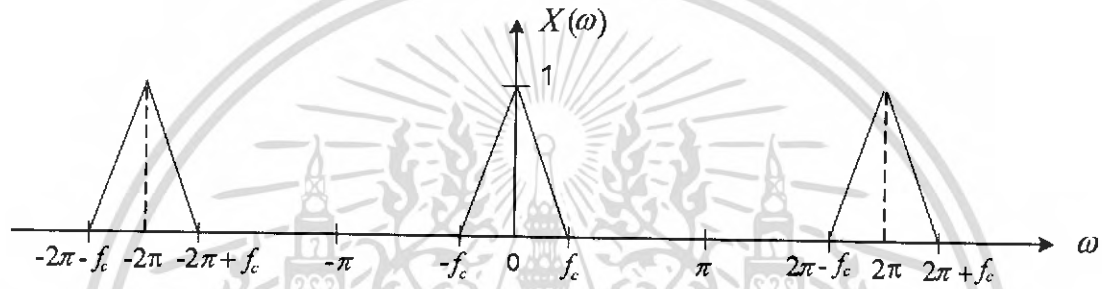
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า y(n) = x(2n)

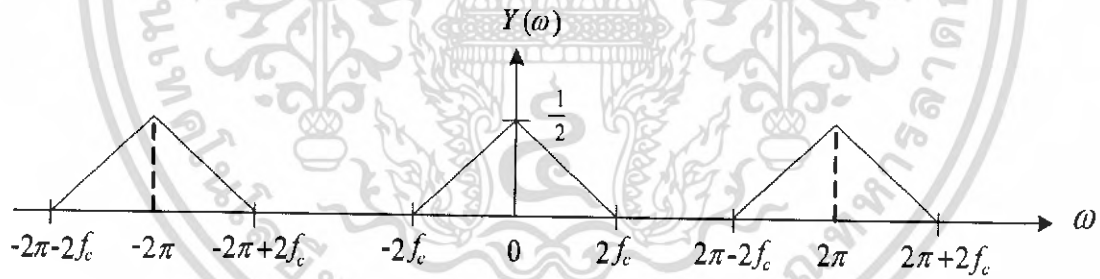
$$Y(\omega) = \sum_{n=-\infty}^{\infty} y(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x(2n)e^{-j\frac{\omega}{2}(2n)}$$

$$= \frac{X(\frac{\omega}{2}) + X(\frac{\omega}{2} - \pi)}{2} \qquad (2.28)$$

เมื่อนำไปพล็อตกราฟจะได้



รูปที่ 2.8 สเปกตรัมของสัญญาณอินพุตก่อนการลดค่าตัวอย่าง

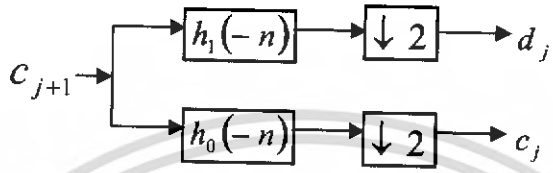


รูปที่ 2.9 สเปกตรัมของสัญญาณเอาท์พุตหลังจากผ่านการลดค่าตัวอย่าง

จากกราฟจะพบว่าการลดค่าตัวอย่างจะทำให้สเปกตรัมของสัญญาณอินพุตถูกขยายออกเป็น 2 เท่าและทำให้ขนาดลดลงครึ่งหนึ่ง ถ้าต้องการไม่ให้เกิดการเอเลียสซิ่งจึงต้องมีฟิลเตอร์กรองก่อนที่จะลดค่าตัวอย่าง (เป็น LPF ที่มี cut-off ที่ $\frac{\pi}{2}$) เรียกว่า decimation filter หรือ anti aliasing filter เพื่อป้องกันไม่ให้เกิดการซ้อนทับเนื่องจากการขยายของความถี่ขึ้น 2 เท่า $\left(f_c < \frac{\pi}{2} \right)$

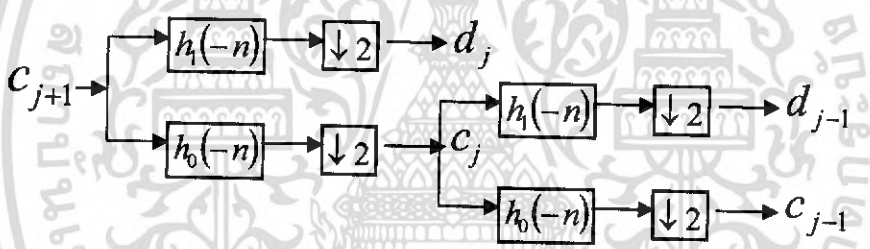
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (2.22) และ (2.23) จะเห็นได้ว่าสัมประสิทธิ์ของสเกลลิ่งและเวฟเลตในระดับที่ต่างกันเกิดจากการคอนโวลูชั่นระหว่างสัมประสิทธิ์การกระจายในระดับ j กับสัมประสิทธิ์ของวงจรกรองความถี่ $h_0(-n)$ และ $h_1(-n)$ จากนั้นจึงนำมาสุ่มค่าตัวอย่าง จึงจะได้สัมประสิทธิ์การกระจายในลำดับถัดไป คือ $j-1$ หรืออาจกล่าวได้ว่า สัมประสิทธิ์ในระดับ j ถูกกรองโดยวงจรกรองความถี่สองตัว ซึ่งมีสัมประสิทธิ์คือ $h_0(-n)$ และ $h_1(-n)$ และหลังจากผ่านการสุ่มค่าตัวอย่างก็จะได้สัมประสิทธิ์สเกลลิ่งและเวฟเลตในระดับที่ละเอียดน้อยกว่า ดังรูป



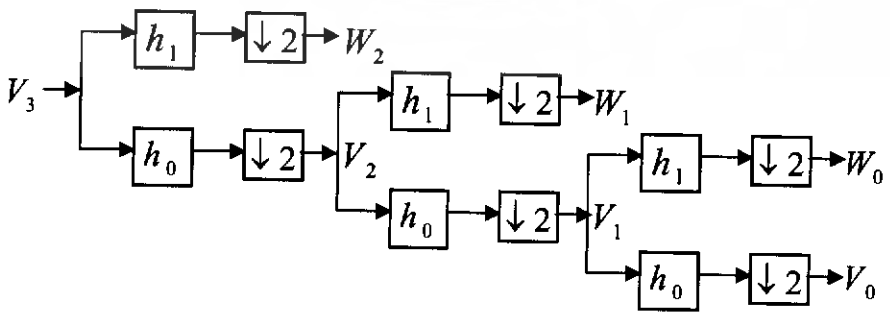
รูปที่ 2.10 การแตกกิ่งก้านสาขาแบบสองแถบของการแปลงเวฟเลต

การแตกออก การกรองสัญญาณและการสุ่มค่าตัวอย่างจะทำซ้ำในส่วนของสัมประสิทธิ์สเกลลิ่งเพื่อให้ได้โครงสร้างตามรูปที่ 2.12 ซึ่งเราเรียกว่า Iterating the filter bank



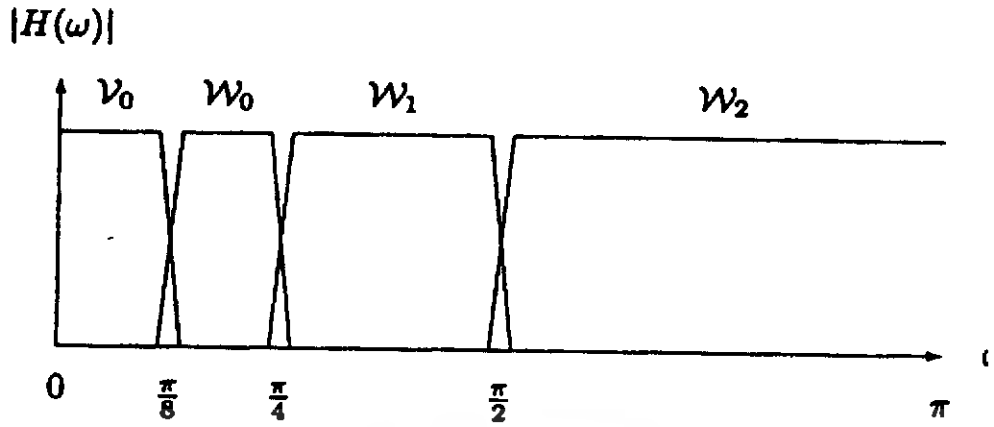
รูปที่ 2.11 การแตกกิ่งก้านสาขาแบบสองแถบสองขั้นของการแปลงเวฟเลต

ในขั้นแรกจะแบ่งส่วนของสเกลลิ่งฟังก์ชันออกเป็นส่วนของวงจรกรองความถี่ต่ำ (Lowpass Band) และส่วนของวงจรกรองความถี่สูง (Highpass Band) ซึ่งจะทำให้ได้สัมประสิทธิ์ของสเกลลิ่งและสัมประสิทธิ์ของเวฟเลตในระดับต่ำกว่า ส่วนในขั้นที่สองจะแบ่งส่วนของวงจรกรองความถี่ต่ำเดิมออกเป็นส่วนของวงจรกรองความถี่ต่ำและส่วนของวงจรกรองความถี่สูงอีกครั้ง ดังรูป



รูปที่ 2.12 การแตกกิ่งก้านสาขาแบบสองแถบสามขั้นของการแปลงเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แถบความถี่ของการแปลงเวฟเลต

2.4.2 หลักการสังเคราะห์ (Synthesis-From Coarse Scale to Fine Scale)

การที่จะนำสัญญาณเดิมกลับคืนมาสามารถทำได้โดยการรวมกันของสัมประสิทธิ์สเกลลิ่งฟังก์ชันและเวฟเลตในระดับที่มีความละเอียดน้อยกว่า โดยสามารถเขียนเป็นสมการที่อยู่ในรูปของสเกลลิ่งฟังก์ชันในระดับ $j+1$ ได้ดังนี้

$$f(t)=\sum_k c_{j+1}(k)2^{(j+1)/2}\varphi\left(2^{j+1}t-k\right) \tag{2.29}$$

หรืออาจจะเขียนในรูปของระดับถัดไป ซึ่งจะมีส่วนของเวฟเลตด้วยดังนี้

$$f(t)=\sum_k c_j(k)2^{j/2}\varphi\left(2^j t-k\right)+\sum_k d_j(k)2^{j/2}\psi\left(2^j t-k\right) \tag{2.30}$$

เมื่อแทนค่าสมการที่ (2.17) และ (2.15) ลงในสมการที่ (2.30) จะได้

$$f(t)=\sum_k c_j(k)\sum_n h(n)2^{(j+1)/2}\varphi\left(2^{j+1}t-2k-n\right)+\sum_k d_j(k)\sum_n h_1(n)2^{(j+1)/2}\varphi\left(2^{j+1}t-2k-n\right) \tag{2.31}$$

เนื่องจากฟังก์ชันทั้งหมดนี้เป็นฟังก์ชันเชิงตั้งฉาก จึงคูณสมการที่ (2.29) และ (2.31) ด้วย $\varphi\left(2^{j+1}t-k\right)$ และอินทิเกรตหาค่าสัมประสิทธิ์ได้ดังนี้

$$c_{j+1}(k)=\sum_m c_j(m)h(k-2m)+\sum_m d_j(m)h_1(k-2m) \tag{2.32}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.1 การกรองสัญญาณและการเพิ่มค่าตัวอย่าง (Filtering and Up-Sampling or Stretching)

ในส่วนของฟิลเตอร์แบงค์ด้านสังเคราะห์จะประกอบด้วยการเพิ่มค่าตัวอย่างและการกรองสัญญาณ ซึ่งหมายความว่าอินพุตของฟิลเตอร์จะมีศูนย์แทรกอยู่ระหว่างสัญญาณเดิม ดังสมการ

การเพิ่มค่าตัวอย่าง

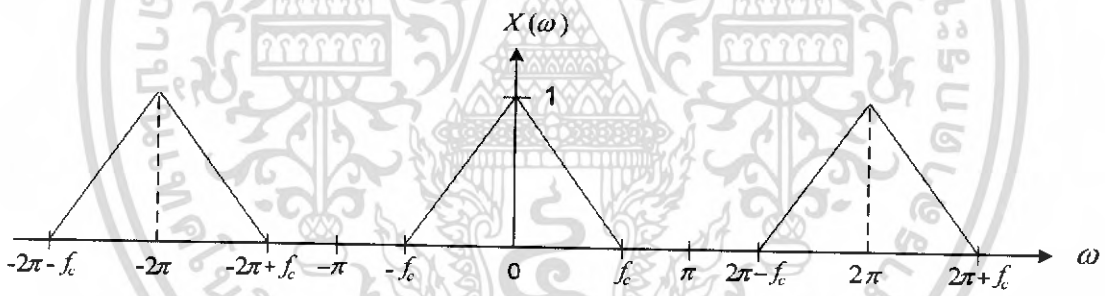
$$y(n) = \begin{cases} x(\frac{n}{2}) & n : \text{even} \\ 0 & n : \text{odd} \end{cases}$$

$$Y(\omega) = \sum_{n=-\infty}^{\infty} y(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} y(2n)e^{-j\omega(2n)} \qquad (2.33)$$

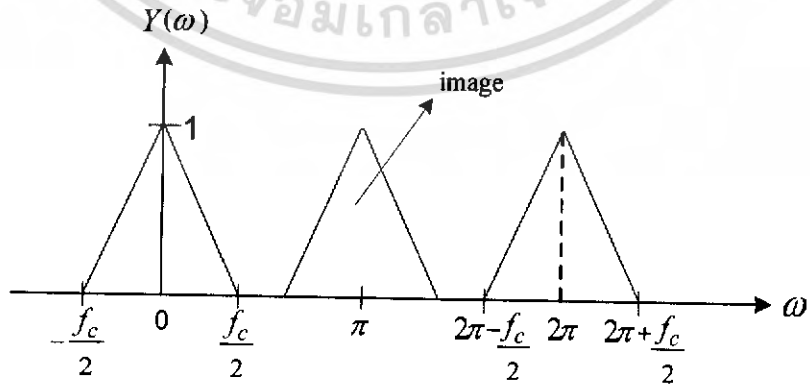
จาก $y(n) = x(n/2)$;

$$Y(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\omega n} = X(2\omega) \qquad (2.34)$$

เมื่อนำไปพล็อตกราฟจะได้



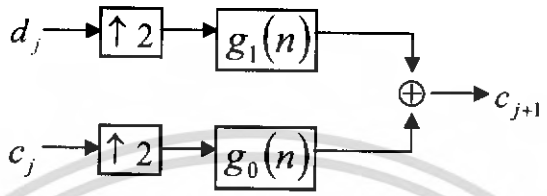
รูปที่ 2.14 สเปกตรัมของสัญญาณอินพุตก่อนการเพิ่มค่าตัวอย่าง



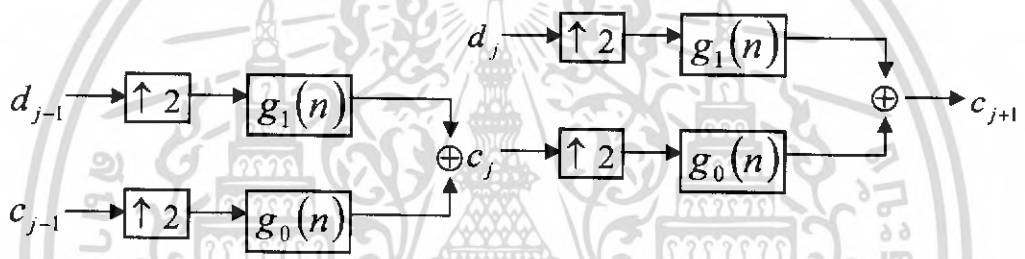
รูปที่ 2.15 สเปกตรัมของสัญญาณเอาท์พุตเมื่อผ่านการเพิ่มค่าตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟจะพบว่าการเพิ่มค่าตัวอย่างจะทำให้ความกว้างสเปกตรัมของสัญญาณลดลงเหลือครึ่งหนึ่งของสัญญาณเดิม และจะเกิดสเปกตรัมของอิมเมจขึ้น(เกิดจากการเติมศูนย์ในตำแหน่งคี่) ซึ่งต้องกรองสัญญาณอิมเมจออกโดยใช้วงจรกรองความถี่ต่ำ เพื่อทำให้สเปกตรัมเหมือนเดิมเรียกตัวกรองความถี่ต่ำนี้ว่า interpolation filter และจากสมการที่ (2.32) ซึ่งเกิดจากการเพิ่มค่าตัวอย่างในระดับ j จากนั้นจึงคอนโวลูชั่นกับสัมประสิทธิ์ของวงจรกรองความถี่ $g(n)$ ทั้งในส่วนของสเกลลิ่งฟังก์ชันและเวฟเลต แล้วนำมาบวกกันได้เป็นสัมประสิทธิ์ในระดับ $j+1$ ดังรูป



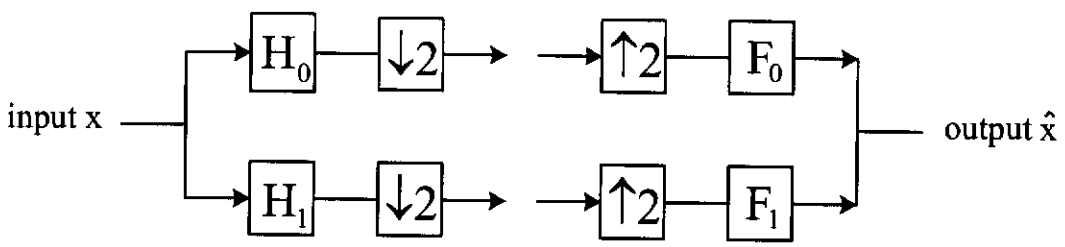
รูปที่ 2.16 การแตกกิ่งก้านสาขาแบบสองแถบของการแปลงกลับเวฟเลต



รูปที่ 2.17 การแตกกิ่งก้านสาขาแบบสองแถบสองขั้นของการแปลงกลับเวฟเลต

2.5 ฟิลเตอร์แบงค์ (Filter Bank)

ฟิลเตอร์แบงค์ คือ เซตของวงจรกรองความถี่ที่เชื่อมต่อกับตัวสุ่มค่าสัญญาณ หรือบางครั้งอาจเป็นตัวหน่วงสัญญาณ(Delay) โดยที่ตัวสุ่มค่าสัญญาณจะเป็นตัวที่ทำการลดขนาดของอินพุต (decimator) และตัวเพิ่มค่าสัญญาณจะเป็นตัวที่ทำการขยายขนาด (expander) ฟิลเตอร์แบงค์ทั้งการแปลงและการแปลงกลับการแปลงหรือวงจรกรองความถี่ด้านวิเคราะห์ (analysis filter) จะประกอบด้วยวงจรกรองความถี่ต่ำและวงจรกรองความถี่สูง โดยจะแทนด้วย H_0 และ H_1 ตามลำดับ ดังรูป

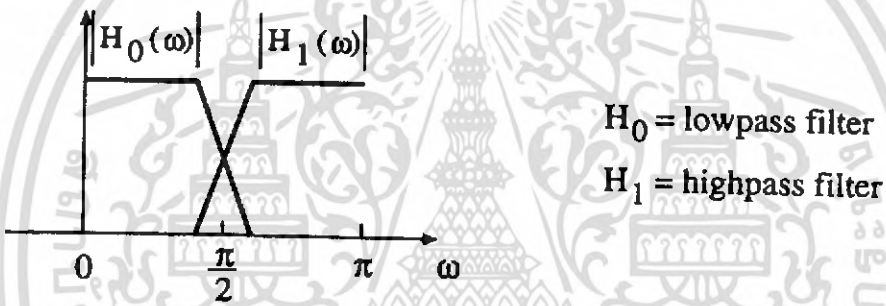


รูปที่ 2.18 ฟิลเตอร์แบงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยต่อไปจะอธิบายถึงการเลือก H_0, H_1, F_0, F_1 เพื่อให้ได้ตรงตามหลักการของการสร้างกลับคืนอย่างสมบูรณ์(Perfect reconstruction) ส่วนตรงที่เป็นช่องว่างในรูปจะหมายถึงการที่สัญญาณที่ถูกสุ่มค่าแล้วถูกนำไปเข้ารหัสเพื่อการส่งสัญญาณต่อไป โดยในจุดนี้สัญญาณจะถูกบีบอัดและอาจมีการสูญเสียของข้อมูลไปบ้าง แต่ในเรื่องการกู้กลับอย่างสมบูรณ์จะไม่เกี่ยวข้องกับการบีบอัดข้อมูลดังนั้นช่องว่างตรงนี้จึงไม่มี

ในการแสดงให้เห็นว่า H_0 คือวงจรกรองความถี่ต่ำ และ H_1 คือวงจรกรองความถี่สูง จะแสดงให้เห็นเป็นผลตอบสนองทางความถี่ ดังรูป ซึ่งจะเห็นได้ว่าทั้งสองวงจรไม่เป็นวงจรกรองความถี่ในอุดมคติ เนื่องจากผลตอบสนองมีการซ้อนทับกัน(Overlap) จึงทำให้เกิดการซ้อนทับกันของสเปกตรัม (aliasing) ในแต่ละช่องสัญญาณ รวมถึงยังเกิดการลดทอนสัญญาณในเชิงขนาดและเฟส แต่ในที่นี้จะกล่าวถึงเฉพาะการลดทอนสัญญาณในทางขนาดเท่านั้น (amplitude distortion) ดังนั้นในการแปลงกลับวงจรกรองความถี่ด้านสังเคราะห์ (synthesis filter) F_0, F_1 จึงจำเป็นต้องปรับให้เข้ากับวิเคราะห์ด้วยเพื่อที่จะกำจัดค่าผิดพลาดที่จะเกิดขึ้นในวงจรกรองความถี่ด้านวิเคราะห์



รูปที่ 2.19 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำและสูง

เป้าหมายของการหาเงื่อนไขในการกู้กลับอย่างสมบูรณ์ หมายความว่าฟิลเตอร์แบงค์จะต้องเป็นฟังก์ชันเชิงตั้งฉากคู่ (biorthogonal) คือ ทางด้านการแปลงกลับตั้งแต่ $F_0, F_1, \uparrow 2$ จะต้องเป็นส่วนกลับของการแปลง

หลักการของการกู้กลับอย่างสมบูรณ์ถือเป็นคุณสมบัติที่สำคัญมาก ซึ่งจะประกอบด้วย 2 เงื่อนไข คือ การกำจัดการเกิดซ้อนทับกันของสเปกตรัม และการป้องกันไม่ให้เกิดการผิดเพี้ยนของสัญญาณถ้าตัวดำเนินการสุ่มค่าตัวอย่างทั้ง $(\downarrow 2)$และ $(\uparrow 2)$ไม่ถูกแสดง การกู้กลับที่ปราศจากการหน่วงจะหมายถึง $F_0H_0 + F_1H_1 = I$ ซึ่งสามารถเขียนให้อยู่ในรูป Z-domain ได้ดังนี้

$$F_0(z)H_0(z) + F_1(z)H_1(z) = z^{-1} \tag{2.35}$$

ในที่นี้สามารถแทนค่า $-z$ ด้วย z เช่นเดียวกับการแทน $\omega + \pi$ ด้วย ω การรวมกันของ$(\downarrow 2)$ และ$(\uparrow 2)$ จะเป็นการกำจัดค่าศูนย์ออกจากพจน์คี่ จึงคงเหลือไว้เฉพาะพจน์คู่ ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(\downarrow 2)(\uparrow 2)H_0 x = \frac{1}{2}[H_0(z)X(z) + H_0(-z)X(-z)] \quad (2.36)$$

ในส่วนของการแปลงกลับจะคูณเทอมของ $H_0(-z)X(-z)$ ด้วย F_0 และคูณเทอม $H_1(-z)X(-z)$ ด้วย F_1 จะทำให้ได้เงื่อนไขของการกำจัดการเกิดซ้อนทับกันของสเปกตรัม(Aliasing cancellation) ดังสมการ

$$F_0(z)H_0(z) + F_1(-z)H_1(-z) = 0 \quad (2.37)$$

ถ้าสมมติให้ $H_0(z) = 1, H_1(z) = z^{-1}$ และ $F_0(z) = z^{-1}, F_1(z) = 1$ ในสมการ จะทำให้ได้ค่าเท่ากับ $2z^{-1}$ ดังสมการ ซึ่งจะเป็นการกำจัดความผิดเพี้ยนของสัญญาณ (No distortion condition)

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l} \quad (2.38)$$

จากรูปผลตอบสนองทางความถี่จะเห็นได้ว่า ในส่วนของวงจรกรองความถี่สูง $H_1 = 0$ ที่ $z = 1$ หรือที่ $\omega = 0$ ถ้าแทนค่าในสมการ จะได้ $F_0(1)H_0(1) = 2$ และถ้าทำการปรับค่า (Normalize) โดยการคูณด้วย $\sqrt{2}$ และแทนค่าสัมประสิทธิ์ของวงจรกรองความถี่ต่ำและสูงด้วย $C = \sqrt{2}H_0$ และ $D = \sqrt{2}H_1$ ผลรวมของสัมประสิทธิ์วงจรกรองความถี่ต่ำ $c(n) = \sqrt{2}h(n)$ จะเท่ากับ $\sqrt{2}$

ถ้าให้สัญญาณอินพุต คือ $x(n)$ และวงจรกรองความถี่ฝั่งส่ง คือ H_0 ถ้าอธิบายในเชิง Z-domain เมื่อผ่านการสุ่มค่าตัวอย่างและเพิ่มค่าตัวอย่างจะได้

$$(\downarrow 2) \qquad \frac{1}{2}\left[H_0\left(z^{\frac{1}{2}}\right)X\left(z^{\frac{1}{2}}\right) + H_0\left(-z^{\frac{1}{2}}\right)X\left(-z^{\frac{1}{2}}\right)\right] \quad (2.39)$$

$$(\uparrow 2) \qquad \frac{1}{2}[H_0(z)X(z) + H_0(-z)X(-z)] \quad (2.40)$$

เมื่อผ่านวงจรกรองความถี่จะคูณด้วย $F_0(z)$ ซึ่งจะทำให้ผลที่ออกจากวงจรกรองความถี่ต่ำและวงจรความถี่สูง ดังนี้

วงจรกรองความถี่ต่ำ $\frac{1}{2}F_0(z)[H_0(z)X(z) + H_0(-z)X(-z)]$ (2.41)

วงจรกรองความถี่สูง $\frac{1}{2}F_1(z)[H_1(z)X(z) + H_1(-z)X(-z)]$ (2.42)

หลังจากนั้นนำเอาต์พุตที่ได้จากทั้งสองวงจรมารวมเข้าด้วยกันเพื่อหา $\hat{x}(n)$ หรือในทาง Z-domain คือ $\hat{X}(z)$ ดังสมการ

$$\frac{1}{2}[F_0(z)H_0(z) + F_1(z)H_1(z)]X(z) + \frac{1}{2}[F_0(-z)H_0(-z) + F_1(-z)H_1(-z)]X(-z) \quad (2.43)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72693

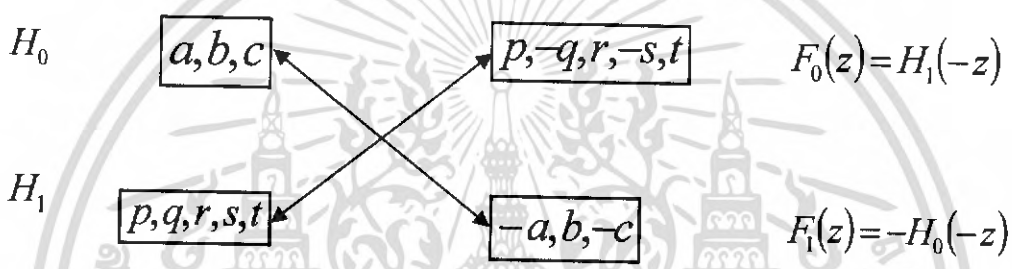
ในการกู้กลับอย่างสมบูรณ์ที่มีการหน่วงเวลา l ครั้ง $\hat{X}(z)$ จะต้องเปลี่ยนเป็น $z^{-l}X(z)$ ดังนั้น เทอมของการลดทอนจะต้องเป็น z^{-l} และเทอมของการซ้อนทับกันของสเปกตรัมจะต้องมีค่าเท่ากับศูนย์

2.5.1 การกำจัดการเกิดการซ้อนทับกันและผลที่ได้จากวงจรกรองความถี่ $P_0 = F_0H_0$

ในตอนนี้จะต้องออกแบบวงจรกรองความถี่สี่ตัว H_0, H_1, F_0, F_1 โดยอาศัยเงื่อนไขที่มีความเกี่ยวข้องกันระหว่างวงจรกรองความถี่ของการแปลงและการแปลงกลับ ดังนี้

$$F_0(z) = H_1(-z) \text{ และ } F_1(z) = -H_0(-z) \tag{2.44}$$

โดยสามารถอธิบายถึงความสัมพันธ์ระหว่าง F_0 กับ H_1 และระหว่าง F_1 กับ H_0 ได้โดยการใช้หลักการใส่เครื่องหมายสลับกัน(Alternating signs) ระหว่างสองฝั่ง ดังนี้



รูปที่ 2.20 การสลับเครื่องหมายในการหาค่าสัมประสิทธิ์

ผลที่ได้จากวงจรกรองความถี่ต่ำจะเท่ากับ $P_0(z) = F_0(z)H_0(z)$ ส่วนผลที่ได้จากวงจรกรองความถี่สูง คือ $P_1(z) = F_1(z)H_1(z)$ และจากเงื่อนไขความสัมพันธ์ที่กล่าวข้างต้น จะได้ว่า

$$P_1(z) = -H_0(-z)H_1(z) = -H_0(-z)F_0(-z) = -P_0(-z) \tag{2.45}$$

จากเงื่อนไขการกำจัดการลดทอนสัญญาณ $F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l}$ เขียนให้อยู่ในรูปของผลที่ได้จากวงจรกรองความถี่ ดังนี้

$$F_0(z)H_0(z) - F_0(-z)H_0(-z) = P_0(z) - P_0(-z) = 2z^{-l} \tag{2.46}$$

หลักการออกแบบวงจรกรองความถี่ทั้งการแปลงและการแปลงกลับสามารถลดได้เหลือ 2 ขั้นตอน ดังนี้

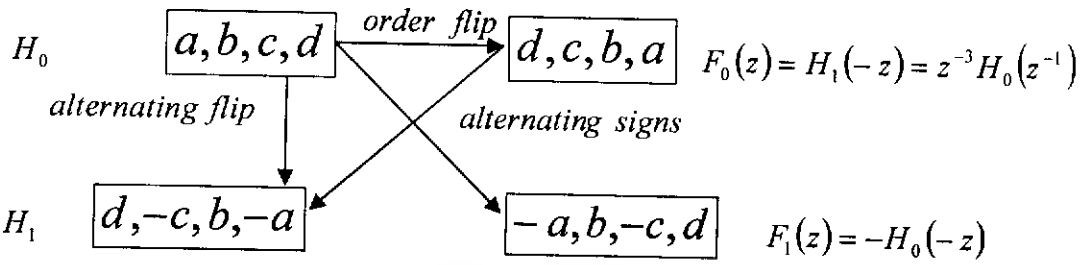
ขั้นแรก : ออกแบบวงจรกรองความถี่ต่ำให้ได้ผลตามสมการ (2.46)

ขั้นสอง : แสดงให้เห็นว่า P_0 เกิดจากผลคูณของ F_0H_0

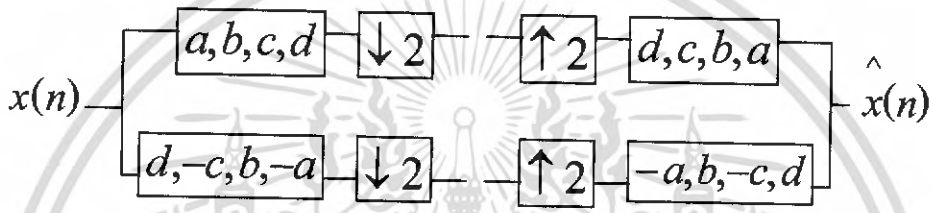
จากนั้นใช้สมการ (2.44) เพื่อหาค่า F_1 และ H_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบฟิลเตอร์แบงค์ทั้งการแปลงและการแปลงกลับให้ตรงตามหลักการการกู้กลับอย่างสมบูรณ์จะอยู่ภายใต้พื้นฐานของฟังก์ชันเชิงตั้งฉาก ดังนี้



รูปที่ 2.21 ความสัมพันธ์ระหว่างสัมประสิทธิ์การแปลงและการแปลงกลับ

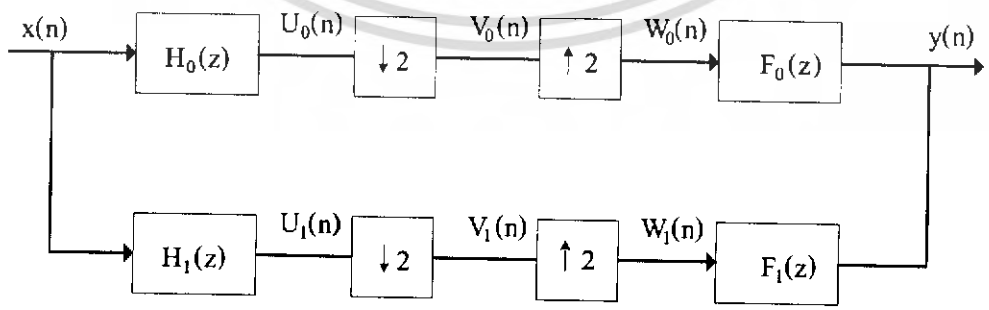


รูปที่ 2.22 ฟิลเตอร์แบงค์เชิงตั้งฉากที่มีสัมประสิทธิ์ 4 ค่า

2.5.2 โครงสร้างของวงจรกรองความถี่รูปแบบโดยตรงและรูปแบบโพลีเฟส (Direct Form and Polyphase Form Filters)

2.5.2.1 วงจรกรองความถี่ที่มีโครงสร้างแบบโดยตรง (Direct Form Filters)

วงจรกรองความถี่รูปแบบโดยตรงในส่วนการแปลงประกอบไปด้วย เซตของวงจรกรองความถี่ต่ำผ่านและวงจรกรองความถี่สูงผ่านตามด้วยการสุ่มค่าตัวอย่าง สำหรับวงจรกรองความถี่ในส่วนการแปลงกลับประกอบไปด้วยการเพิ่มค่าตัวอย่างตามด้วยวงจรกรองความถี่ต่ำผ่านและวงจรกรองความถี่สูงผ่าน โดยสามารถเขียนโครงสร้างโดยตรงตามบล็อกไดอะแกรมรูปที่ 2.23



รูปที่ 2.23 บล็อกไดอะแกรมของวงจรกรองความถี่โครงสร้างโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปบล็อกไดอะแกรมในส่วนการแปลง $x(n)$ คือสัญญาณอินพุต, H_0 และ F_0 เป็นวงจรกรองความถี่ต่ำผ่าน และความถี่สูงผ่านตามลำดับ $\downarrow 2$ หมายถึงการสุ่มค่าตัวอย่างลดลงครึ่งหนึ่งและ $\uparrow 2$ หมายถึง การเพิ่มค่าตัวอย่างเป็น 2 เท่า เช่นกัน อินพุต จะถูกกรองด้วยวงจรกรองความถี่ทั้งสองแบนด์ตามด้วย การสุ่มค่าตัวอย่างลดลงครึ่งหนึ่ง หลังจากนั้นจะได้ เอาต์พุตเป็น $y(n)$ ซึ่งอธิบายในเชิง z-domain ให้เห็นได้ดังสมการ

$$U_0(z) = H_0(z)X(z) \tag{2.47}$$

$$U_1(z) = H_1(z)X(z) \tag{2.48}$$

$$V_0(z) = \frac{1}{2}\left(U_0(z^{\frac{1}{2}}) + U_1(-z^{\frac{1}{2}})\right)$$

$$= \frac{1}{2}\left(H_0(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_0(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right) \tag{2.49}$$

$$V_1(z) = \frac{1}{2}\left(H_1(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_1(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right) \tag{2.50}$$

$$W_0(z) = V_0(z^2) = \frac{1}{2}\left(H_0(z)X(z) + H_0(-z)X(-z)\right) \tag{2.51}$$

$$W_1(z) = V_1(z^2) = \frac{1}{2}\left(H_1(z)X(z) + H_1(-z)X(-z)\right) \tag{2.52}$$

$$Y(z) = F_0(z)W_0(z) + F_1(z)W_1(z) \tag{2.53}$$

$$Y(z) = \frac{1}{2}F_0(z)\left(H_0(z)X(z) + H_0(-z)X(-z)\right)$$

$$+\frac{1}{2}F_1(z)\left(H_1(z)X(z) + H_1(-z)X(-z)\right) \tag{2.54}$$

จะเป็นการกู้กลับอย่างสมบูรณ์ก็ต่อเมื่อ $Y(z) = cz^{-n_0}X(z)$

จัดรูป $$Y(z) = \frac{1}{2}\left(F_0(z)H_0(z) + F_1(z)H_1(z)\right)X(z)$$

$$+\frac{1}{2}\left(F_0(z)H_0(-z) + F_1(z)H_1(-z)\right)X(-z) \tag{2.55}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเงื่อนไข คือ $\frac{1}{2}\left(F_0(z)H_0(-z)+F_1(z)H_1(-z)\right)=0$ (Aliasing cancellation)

และ $\frac{1}{2}\left(F_0(z)H_0(z)+F_1(z)H_1(z)\right)=cz^{-n_0}$ (No distortion condition)

เอาต์พุตของวงจรนั้นจะสามารถกู้กลับมา โดยในส่วนการแปลงกลับของวงจรและเพิ่มค่าตัวอย่างเป็น 2 เท่า แล้วผ่านวงจรกรองความถี่จะทำให้ได้สัญญาณเดิมกลับคืนมา

2.5.2.2 วงจรกรองความถี่ที่มีโครงสร้างแบบโพลีเฟส (Polyphase Structure Filters)

ตัวกรองความถี่แบบเอฟไอาร์ (FIR: Finite Impulse Response) ที่แสดงโดยการกระจายฟังก์ชันการส่งผ่านโพลีเฟส และมีผลลัพธ์เป็นโครงสร้างแบบขนาน ซึ่งเมื่อพิจารณาวิธีการนี้จากฟังก์ชันการส่งผ่าน $H(z)$ ที่มี 9 พจน์ดังสมการที่ 2.56

$$H(z)=h[0]+h[1]z^{-1}+h[2]z^{-2}+h[3]z^{-3}+h[4]z^{-4}+h[5]z^{-5}+h[6]z^{-6}+h[7]z^{-7}+h[8]z^{-8} \quad (2.56)$$

จากสมการที่ 2.56 สามารถแบ่งเป็น 2 เทอม คือ เทอมที่มีสัมประสิทธิ์เป็นพจน์กำลังคู่ และ เทอมที่มีสัมประสิทธิ์เป็นพจน์กำลังคี่ดังสมการที่ 2.57 และ 2.58

$$H(z)=(h[0]+h[2]z^{-2}+h[4]z^{-4}+h[6]z^{-6}+h[8]z^{-8})+(h[1]z^{-1}+h[3]z^{-3}+h[5]z^{-5}+h[7]z^{-7}) \quad (2.57)$$

$$H(z)=(h[0]+h[2]z^{-2}+h[4]z^{-4}+h[6]z^{-6}+h[8]z^{-8})+z^{-1}(h[1]+h[3]z^{-2}+h[5]z^{-4}+h[7]z^{-6}) \quad (2.58)$$

เมื่อกำหนดให้

$$H_e(z)=h[0]+h[2]z^{-2}+h[4]z^{-4}+h[6]z^{-6}+h[8]z^{-8}$$

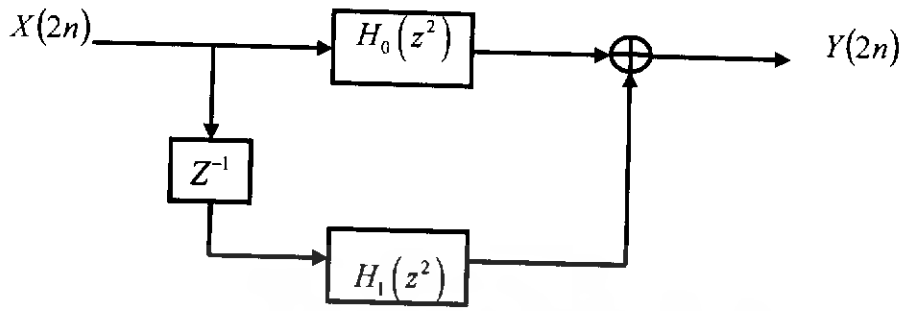
และ $$H_o(z)=h[1]+h[3]z^{-1}+h[5]z^{-2}+h[7]z^{-3} \quad (2.59)$$

จากสมการที่ 2.59 ทำให้จัดรูปแบบสมการที่ 2.58 ใหม่ ดังสมการที่ 2.60

$$H(z)=H_e(z^2)+z^{-1}H_o(z^2) \quad (2.60)$$

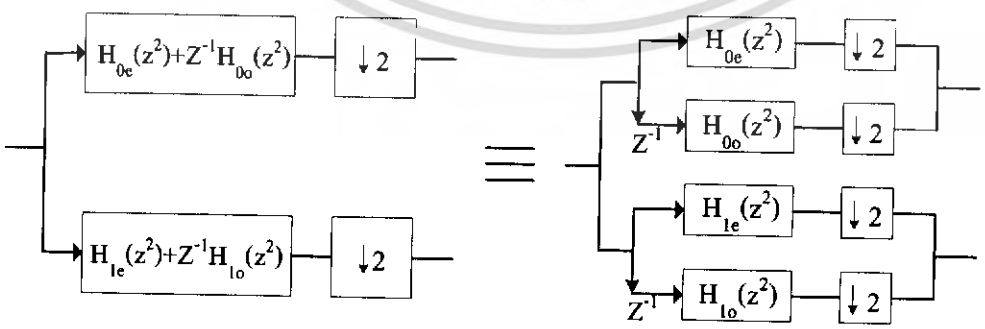
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวิธีการกระจาย $H(z)$ ดังสมการที่ 2.60 เรียกว่า การสร้างจริงของโพลีเฟส (Polyphase Realization) และจากสมการสามารถเขียนอยู่ในรูปของบล็อกไดอะแกรมดังรูปที่ 2.24



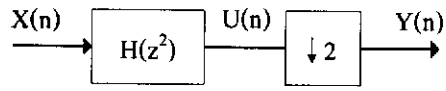
รูปที่ 2.24 บล็อกไดอะแกรมการสร้างจริงของโพลีเฟสของฟังก์ชันการส่งผ่าน

โครงสร้างรูปแบบโพลีเฟสจะมีข้อได้เปรียบจากรูปแบบโดยตรง สังเกตได้จากในส่วนของฟิลเตอร์แบงค์ของรูปแบบโดยตรง เมื่ออินพุตผ่านการกรองจะประกอบไปด้วย N แซมเปิลเมื่อทำการสุ่มค่าตัวอย่างลดลงครึ่งหนึ่งแล้วจะใช้เพียง N/2 เท่านั้น ส่วนที่เหลืออีก N/2 เป็นส่วนที่ไม่มีประโยชน์ และจะเห็นได้ว่าเอาต์พุตที่ผ่านวงจรกรองความถี่ต่ำผ่าน และผ่านการสุ่มค่าตัวอย่างแล้วจะประกอบด้วยจำนวนแซมเปิลของเฟสคู่ของอินพุตเวคเตอร์(X_{even}) ซึ่งคอนโวลูทกับสัมประสิทธิ์เฟสคู่ของวงจรกรองความถี่ต่ำผ่าน(H_{0even}) และจำนวนแซมเปิลของเฟสคี่ของอินพุตเวคเตอร์(X_{odd}) คอนโวลูทกับสัมประสิทธิ์เฟสคี่ของวงจรกรองความถี่ต่ำผ่าน(H_{0odd}) สำหรับรูปแบบโพลีเฟสสัญญาณอินพุตจะถูกแบ่งเป็นจำนวนแซมเปิลคู่และคี่โดยอัตโนมัติเมื่อสุ่มค่าตัวอย่างลดลงครึ่งหนึ่ง ในทำนองเดียวกันสัมประสิทธิ์ของวงจรกรองความถี่ก็จะถูกแบ่งเป็นส่วนคู่และคี่เช่นกัน ดังนั้น X_{even} คอนโวลูทกับ H_{0even} และ X_{odd} ก็จะคอนโวลูทกับ H_{0odd} ซึ่งเมื่อนำ 2 เฟสมาบวกกันทำให้ได้เอาต์พุตของความถี่ต่ำ ในทำนองเดียวกันสำหรับวงจรกรองความถี่สูงผ่าน โดยวงจรกรองความถี่สูงผ่านจะแบ่งเป็นเฟสคู่และคี่และสัมประสิทธิ์ของวงจรกรองความถี่สูงผ่านก็คือ H_{1even} และ H_{1odd} จากรูปที่ 2.24 สามารถแยกพจน์ของ $H_0(z^2)$ และพจน์ของ $H_1(z^2)$ ออกเป็นพจน์คี่และพจน์คู่ได้



รูปที่ 2.25 การแยกโครงสร้างออกเป็นพจน์คี่และพจน์คู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 บล็อคไดอะแกรมก่อนการเปลี่ยนโครงสร้าง

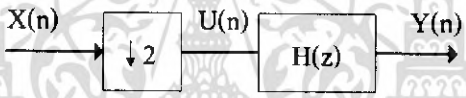
จากรูปที่ 2.26 $U(z) = H(z^2)X(z)$ (2.61)

$$Y(z) = \frac{1}{2}\left(U(z^{\frac{1}{2}}) + U(-z^{\frac{1}{2}})\right)$$

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

$$= \frac{1}{2}H(z)\left(X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})\right) \quad (2.62)$$

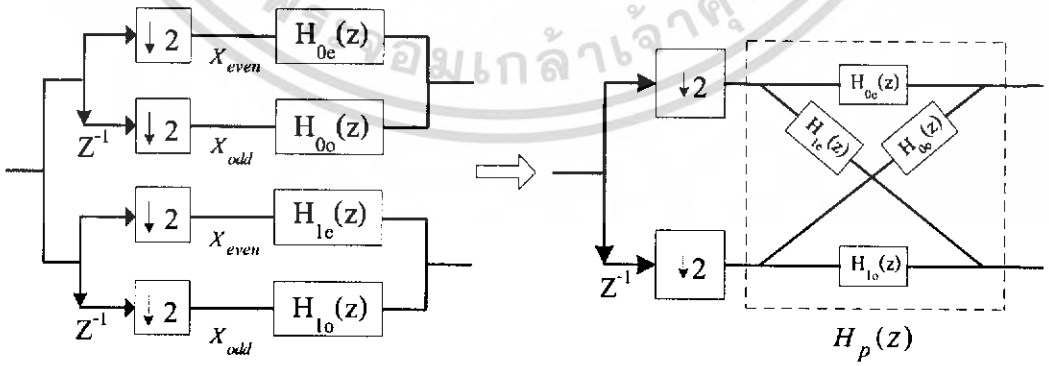
จากสมการข้างต้น สามารถเปลี่ยนจากบล็อกไดอะแกรมข้างต้นมาเป็นบล็อกไดอะแกรมดังรูป



รูปที่ 2.27 บล็อคไดอะแกรมหลังการเปลี่ยนโครงสร้าง

ระบบข้างล่างนี้เหมาะสมมากกว่าระบบข้างบนเมื่อจะนำไปสร้างลงอุปกรณ์ FPGA เพราะอินพุตที่เข้ามาคอนโวลูชั่นจะเหลือจำนวนเพียงครึ่งเดียวเมื่อเทียบกับแบบแรก ทำให้ลดเวลาในการประมวลผลลงและยังช่วยลดการใช้หน่วยความจำในการประมวลผลอีกด้วย

ดังนั้นสามารถเปลี่ยนโครงสร้างของโพลีเฟสให้เป็นดังรูปใหม่ได้



รูปที่ 2.28 บล็อคไดอะแกรมของโครงสร้างแบบโพลีเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $H_p(z)$ คือ โพลีเฟสเมตริกซ์

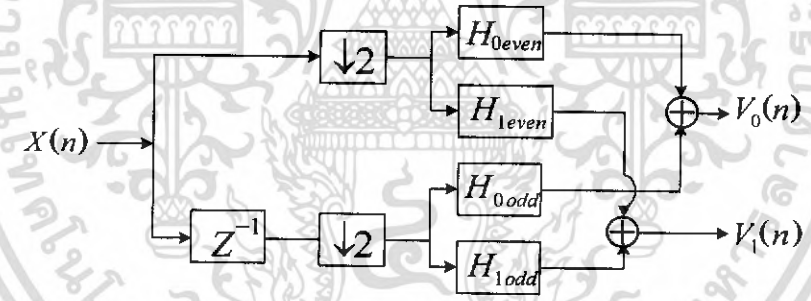
$$H_p(z) = \begin{bmatrix} H_{0e}(z) & H_{0o}(z) \\ H_{1e}(z) & H_{1o}(z) \end{bmatrix}$$

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} H_{0e}(z) & H_{0o}(z) \\ H_{1e}(z) & H_{1o}(z) \end{bmatrix} \begin{bmatrix} X_{even} \\ X_{odd} \end{bmatrix}$$

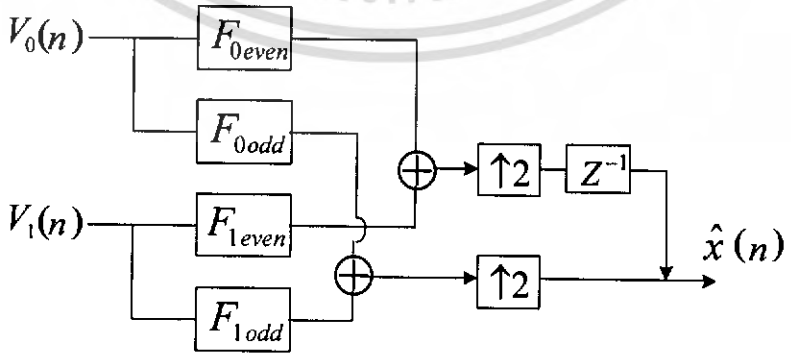
สมการเมตริกซ์ในส่วนการแปลงของโพลีเฟสจะเป็นไปตามสมการที่ 2.63

$$\begin{bmatrix} H_{0even} H_{0odd} \\ H_{1even} H_{1odd} \end{bmatrix} \times \begin{bmatrix} X_{even} \\ z^{-1} X_{odd} \end{bmatrix} = H_p \begin{bmatrix} X_{even} \\ z^{-1} X_{odd} \end{bmatrix} = \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} \quad (2.63)$$

เนื่องจากว่า เทอมคู่และเทอมคี่ถูกแบ่งให้เท่ากันแล้วนำไปกรองด้วยสัมประสิทธิ์ของวงจรกรองความถี่ที่เป็นคู่และคี่อย่างละครึ่งเช่นกัน ทำให้วงจรกรองความถี่สามารถปรับปรุงประสิทธิภาพของวงจรในทางขนาน ซึ่งแสดงให้เห็นดังรูปที่ 2.29 และ 2.30



รูปที่ 2.29 โครงสร้างของฟิลเตอร์แบบโพลีเฟสในส่วนการแปลง



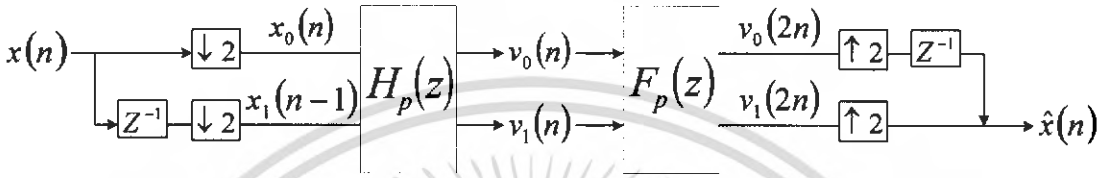
รูปที่ 2.30 โครงสร้างของฟิลเตอร์แบบโพลีเฟสในส่วนการแปลงกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.30 ซึ่งเป็นโครงสร้างของฟิลเตอร์แบบโพลีเฟสในส่วนสังเคราะห์ สามารถเขียนให้อยู่ในรูปของสมการ ดังนี้

$$F(z) = F_1(z^2) + z^{-1}F_0(z^2) \tag{2.64}$$

$$\hat{X}(z) = \begin{bmatrix} z^{-1} & 1 \end{bmatrix} \begin{bmatrix} F_{0,0}(z^2) & F_{1,0}(z^2) \\ F_{0,1}(z^2) & F_{1,1}(z^2) \end{bmatrix} \begin{bmatrix} V_0(z^2) \\ V_1(z^2) \end{bmatrix} \tag{2.65}$$



รูปที่ 2.31 รูปแบบของการแปลงและการแปลงกลับของฟิลเตอร์แบงค์

สำหรับรูปแบบโดยตรงส่วนการแปลงกลับของฟิลเตอร์แบงค์ อินพุตจะเพิ่มค่าตัวอย่างด้วยการบวก 0 แล้วจึงนำไปกรอง แต่ในรูปแบบโพลีเฟสนั้นอินพุตจะกรองก่อนแล้วจึงเพิ่มค่าตัวอย่าง เป็นผลทำให้ลดจำนวนของการคำนวณในการกรองความถี่ลงไปถึงครึ่งหนึ่ง ซึ่งลดลงครึ่งหนึ่งทั้งในการแปลงและการแปลงกลับของฟิลเตอร์แบงค์ ทำให้ประสิทธิภาพทั้งหมดเพิ่มขึ้น 50% ดังนั้นรูปแบบโพลีเฟสจึงแสดงให้เห็นประสิทธิภาพของอุปกรณ์ด้วย

2.6 โครงสร้างการแปลงเวฟเลตโดยใช้วิธีการ Lifting Scheme

สมมติสร้างสัญญาณ $f(t)$ โดยสุ่มที่ค่าแซมปลิ้งค่าหนึ่ง จะกำหนดให้ค่าสัญญาณตั้งต้นมีค่าเป็น $\lambda_0, k = f(k)$ โดย $k \in Z$ แทนสัญญาณที่เริ่มทดลอง ดังนั้นการที่จะแทนสัญญาณตัวนี้ด้วยค่าสัมประสิทธิ์เพียงไม่กี่ตัวหรือสัมประสิทธิ์มีค่าของแซมปลิ้งใหญ่กว่าเดิมเป็นสิ่งจำเป็นที่เอาไปใช้กับการบีบอัดข้อมูล ซึ่งในบางครั้งการใช้ค่าสัมประสิทธิ์ค่าน้อย มาแทนสัญญาณตัวหนึ่งจะเป็นไปได้ยากหรืออาจจะเป็นไปไม่ได้เลย แต่ก็ดีกว่าการประมาณสัญญาณโดยไม่สนใจค่าความผิดพลาดเลย ดังนั้นจึงต้องการการควบคุมที่แม่นยำในการแทนข้อมูลที่ผิดพลาดด้วยสัมประสิทธิ์จำนวนน้อย ด้วยการทำให้ค่าความแตกต่างระหว่างสัญญาณตั้งต้นกับสัญญาณที่ประมาณค่าออกมามีความแตกต่างกันน้อยที่สุด

การลดจำนวนสัมประสิทธิ์สัญญาณทำได้โดยการสุ่มเอาแต่จำนวนคู่และสามารถเขียนเป็นสมการใหม่เป็น

$$\lambda_{-1,k} = \lambda_{0,2k} \quad \text{โดย} \quad k \in Z \tag{2.66}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการทราบว่ามีจำนวนข้อมูลที่สูญหายเท่าไหร่ หรือต้องการข้อมูลพิเศษที่จะนำมาใช้กู้สัญญาณตั้งต้น $\{\lambda_{0,k}\}$ จาก $\{\lambda_{-1,k}\}$ โดยจะใช้สัมประสิทธิ์ $\{\gamma_{-1,k}\}$ มาเข้ารหัส ส่วนข้อมูลที่ต่างจากสัญญาณตั้งต้นกับสัญญาณที่ประมาณขึ้นมาจะให้เป็นสัมประสิทธิ์เวฟเลต อันที่จริงแล้วมีหลายวิธีที่ทำได้ ซึ่งขึ้นอยู่กับข้อมูลทางสถิติของสัญญาณ โดยวิธีที่เหมาะสมนั้นจะได้สัมประสิทธิ์เวฟเลตที่มีขนาดเล็กกว่า

โดยจะกล่าวว่าข้อมูลที่หายไปนั้นก็อยู่ในสัมประสิทธิ์จำนวนคี่นั่นเอง $\gamma_{-1,k} = \lambda_{0,2k+1}$ โดย $k \in Z$ ซึ่งตัวเลือกนี้ก็จะสอดคล้องกับวิธีการ Lazy wavelet ซึ่งเป็นวิธีการที่ไม่ยุ่งยากเพียงแต่สุ่มตัวอย่างสัญญาณให้อยู่ในรูปแซมเปิลคู่และคี่ สัมประสิทธิ์เวฟเลตจะมีขนาดเล็กในกรณีที่แซมเปิลคี่มีขนาดเล็กและจะต้องใช้วิธีการ Lazy wavelet มาร่วมด้วย

สำหรับวิธีการที่ซับซ้อนขึ้นในการกู้สัญญาณตั้งต้น $\{\lambda_{0,k}\}$ จากแซมเปิลย่อย $\{\lambda_{-1,k}\}$ จากแซมเปิลคู่ $\{\lambda_{0,2k}\}$ จากสมการข้างต้น จะพบว่า $\lambda_{0,2k} = \lambda_{1,k}$ แต่สามารถทำนายได้ว่าแซมเปิลคี่นั้นมาจาก $\{\lambda_{-1,k}\}$ เช่นกัน สมมติให้แซมเปิลที่อยู่ติดกันมีความสัมพันธ์กันโดยให้แซมเปิลคี่ $\lambda_{0,2k+1}$ เป็นค่าเฉลี่ยระหว่างสองแซมเปิลที่ติดกัน $\lambda_{-1,k}$ กับ $\lambda_{-1,k+1}$ โดยที่ไม่ได้ต้องการให้ถูกที่สุดเพราะสัมประสิทธิ์เวฟเลตจะเข้ารหัสค่าความแตกต่างของสัญญาณตั้งต้นและสัญญาณที่เราประมาณค่า ตามสมการ

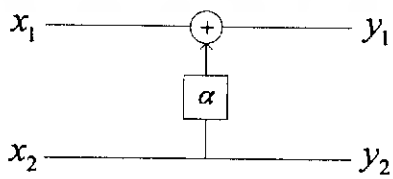
$$\gamma_{-1,k} = \lambda_{0,2k+1} - \frac{1}{2}\left(\lambda_{-1,k} + \lambda_{-1,k+1}\right) \tag{2.67}$$

ถ้าสัญญาณนั้นสัมพันธ์กันแบบใดก็ตาม ส่วนใหญ่แล้วค่าสัมประสิทธิ์เวฟเลตที่แทนตัวสัญญาณที่ต่างกันจะมีค่าน้อย ค่าสัมประสิทธิ์ตัวใดที่มีค่าต่ำกว่าค่าความผิดพลาดที่จุดเริ่มต้นอาจจะละทิ้ง โดยวิธีนี้เราจะได้สัมประสิทธิ์ที่แทนสัญญาณที่ดีกว่า ถ้าสัญญาณตั้งต้นกับสัญญาณที่แปลงค่าแล้วเหมือนกัน สัมประสิทธิ์เวฟเลตทั้งหมดจะมีค่าเป็นศูนย์

Lifting Scheme เป็นการแยกแฟคเตอร์โพลีเฟสเมตริก จากที่ได้กล่าวมาข้างต้นสามารถเขียนสมการโพลีเฟสในรูปเมตริกซ์ได้

กำหนดให้

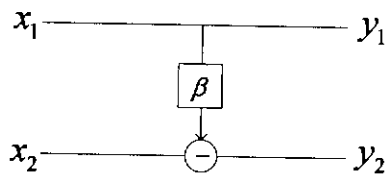
เมตริกซ์ $\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}$ แทนด้วยสัญลักษณ์



รูปที่ 2.32 บล็อกไดอะแกรมแทนเมตริกซ์ α

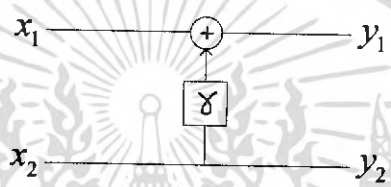
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตริกซ์ $\begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix}$ แทนด้วยสัญลักษณ์



รูปที่ 2.33 บล็อกไดอะแกรมแทนเมตริกซ์ β

เมตริกซ์ $\begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix}$ แทนด้วยสัญลักษณ์



รูปที่ 2.34 บล็อกไดอะแกรมแทนเมตริกซ์ γ

จากสมการของโพลีเฟสสามารถเขียนอยู่ในรูปเมตริก ให้เป็น $\begin{bmatrix} c_0 & -s_0 \\ s_0 & c_0 \end{bmatrix}$

นำเมตริกซ์ $\begin{bmatrix} c_0 & -s_0 \\ s_0 & c_0 \end{bmatrix}$ มาแยกแฟคเตอร์ให้อยู่ในรูปเมตริกซ์ย่อย α , β , γ

$$\begin{bmatrix} c_0 & -s_0 \\ s_0 & c_0 \end{bmatrix} = \begin{bmatrix} 1+\alpha\beta & \alpha+\gamma+\alpha\beta\gamma \\ \beta & 1+\beta\gamma \end{bmatrix} \tag{2.68}$$

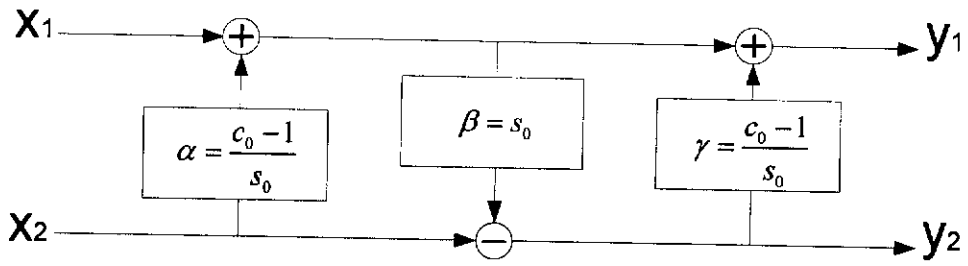
$$\begin{bmatrix} c_0 & -s_0 \\ s_0 & c_0 \end{bmatrix} = \begin{bmatrix} 1+\alpha\beta & \alpha+\gamma+\alpha\beta\gamma \\ \beta & 1+\beta\gamma \end{bmatrix} \tag{2.69}$$

สามารถหาค่า α , β , γ ดังนี้

$$\alpha = \frac{c_0 - 1}{s_0} \quad , \quad \beta = s_0 \quad \text{และ} \quad \gamma = \frac{c_0 - 1}{s_0}$$

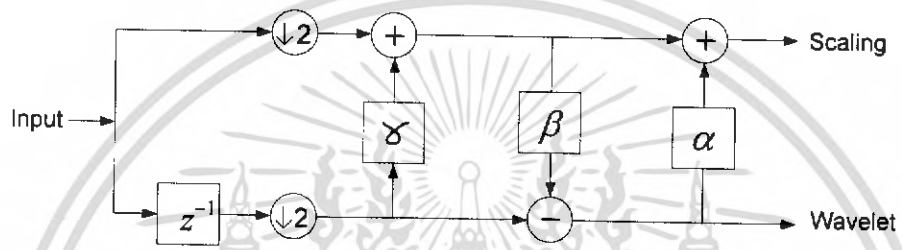
นำมาเขียนโครงสร้างใหม่ได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 โครงสร้าง Lifting Scheme ที่แปลงมาจากโพลีเฟสเมตริกซ์

นำมาเขียนโครงสร้างของ Lifting Scheme ทั่วไปได้เป็น



รูปที่ 2.36 โครงสร้างทั่วไปของ Lifting Scheme ด้านการแปลง

2.7 ภาษาวีเอชดีแอล

ความซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะ ส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบมาใช้ในกระบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้น อีกทั้งอุปกรณ์และวิธีการออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ เอชดีแอล (HDL: Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่องเพื่อช่วยในการปรับปรุงกระบวนการออกแบบระบบดิจิตอลให้เป็นไปอย่างมีประสิทธิภาพ

2.7.1 ประวัติความเป็นมาของภาษาวีเอชดีแอล

วีเอชดีแอล ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และสามารถเขียนได้หลายรูปแบบด้วยเหตุผลนี้จึงทำให้ภาษาวีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหาลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกรได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชั่นการทำงานของวงจรอย่างสังเขปโดยไม่คำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนี้วีเอชดีแอลยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด ดังนั้นวีเอชดีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง สำหรับมาตรฐานของภาษาที่ใช้บรรยายพฤติกรรมวงจรหรือฮาร์ดแวร์ สามารถสรุปได้ดังนี้

-ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอลและมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่องคอมพิวเตอร์โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก

- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้

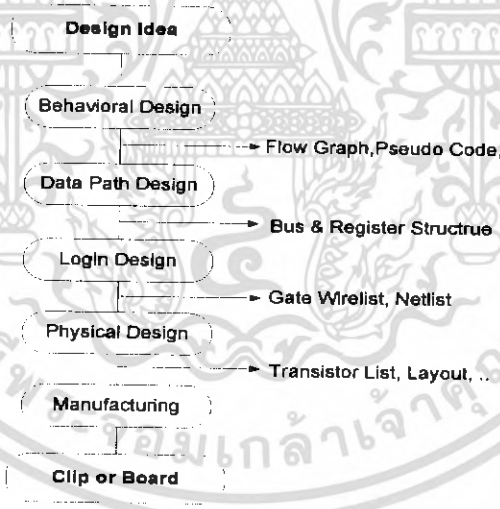
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่น เดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรมภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า ภาษาโปรแกรมระดับสูง

2.7.2 การออกแบบระบบดิจิตอล

ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำเนิดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ที่ใช้งานได้จะต้องผ่านขั้นตอนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้นก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป

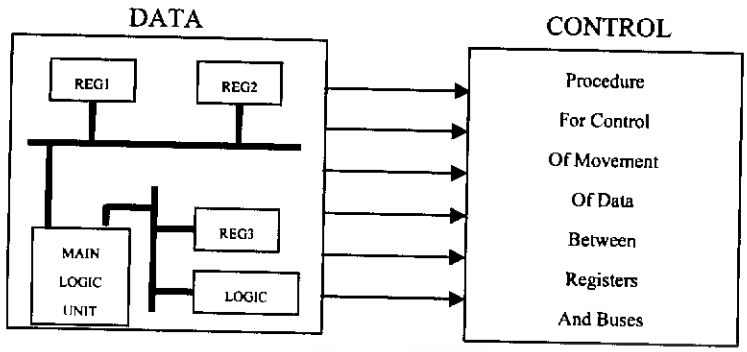
รูปที่ 2.37 แสดงขั้นตอนปกติที่ใช้ในการออกแบบ แล้วพัฒนาให้สามารถนำไปใช้ได้อย่างสมบูรณ์ ซึ่งภายในจะทำการสร้างรูปแบบเชิงพฤติกรรมขึ้นมาตรวจสอบ ซึ่งอาจจะเป็นผังงาน ขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบแสดงผล หรือรหัสคำสั่งเทียม (Pseudo code) ก็ได้



รูปที่ 2.37 ขั้นตอนการออกแบบระบบดิจิตอล

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิกที่จำเป็นทั้งหมด เพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสองทิศทาง (Unidirectional or Bidirectional Bus) กระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่กำหนดไว้ดังรูปที่ 2.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 การออกแบบระบบเส้นทางข้อมูล

ขั้นตอนถัดมาเป็นการออกแบบวงจรลอจิกซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐาน และฟลิปฟลอป (Flip – Flop) มาประกอบเป็นอุปกรณ์ย่อยต่าง ๆ เช่น รีจิสเตอร์เก็บข้อมูล บัสวงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการเชื่อมต่อระหว่างเกทและฟลิปฟลอป

การออกแบบในขั้นตอนถัดไปเป็นการเปลี่ยนเครือข่ายการเชื่อมต่อที่ได้จากขั้นตอนที่แล้ว ให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และโครงงาน (Layout) ซึ่งขั้นตอนนี้จะเกี่ยวข้องกันโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์เพื่อแทนเกทและฟลิปฟลอปต่าง ๆ

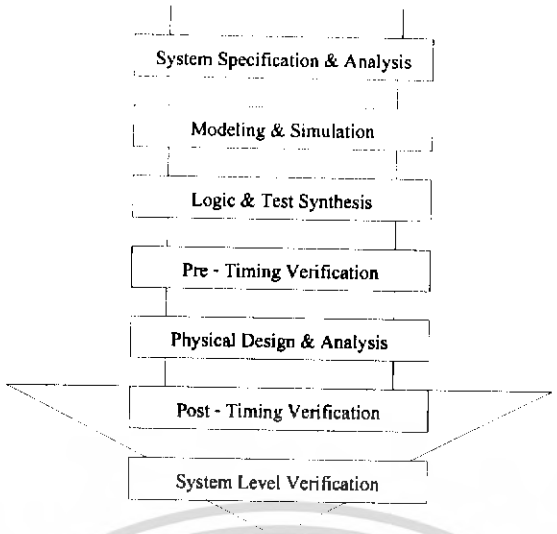
และในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือที่โรงงานเพื่อผลิตออกมาเป็นวงจรรวมในที่สุด

2.7.3 การออกแบบจากบนลงล่าง (Top-Down Design)

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อนที่วิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายและวิเคราะห์การทำงานของแต่ละบล็อก รวมถึงการปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามต้องการนอกจากนี้ยังสามารถเพิ่มเติมในรายละเอียดในแต่ละขั้นตอนได้ ซึ่งหลักการนี้สอดคล้องกับหลักการออกแบบจากบนลงล่างนั่นเอง ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เนื่องจากเป็นการวางวงจรด้วยอุปกรณ์ต่างๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบก่อน แล้วจึงทำการจำลองการทำงานและตรวจสอบความถูกต้อง

วีเอชดีแอลกับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรให้สามารถออกแบบและพัฒนาวงจรที่มีความซับซ้อนได้มากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.39 แสดงถึงขั้นตอนการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจมีข้อแตกต่างไปจากนี้บ้าง เล็กน้อยเนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายเทคโนโลยี สำหรับรายละเอียดของขั้นตอนการออกแบบจากบนลงล่างในแต่ละขั้นตอนมีดังนี้

1) ความต้องการของระบบและการวิเคราะห์ คือ การสร้างข้อกำหนดของความต้องการและวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2) รูปแบบและการจำลองการทำงาน คือ การเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษาวีเอชดีแอล หรือ ภาษาเอชดีแอล อื่นๆ สำหรับใช้ในกา รบรรยายพฤติกรรมการทำงาน พร้อมทั้งทำการจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3) ลอจิกและการทดสอบการสังเคราะห์ คือ หลังจากที่ได้หลักการขั้นต้นพร้อมแนวความคิดที่ผ่านการตรวจสอบแล้วหลักการนี้จะถูกเพิ่มเติมรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจรจริง หรือสังเคราะห์ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่จะเขียนขึ้นให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้นหรือไม่ก็อยู่ในรูปของโครงข่ายการเชื่อมต่อ ที่สามารถนำไปผลิตอุปกรณ์อื่นได้

4) การตรวจสอบเวลาก่อนการออกแบบ คือ หลังจากได้สังเคราะห์วงจรให้อยู่ในระดับเกทหรือโครงข่ายการเชื่อมต่อแล้ว ข้อมูลนี้จะถูกนำไปใช้สำหรับการจำลองการทำงานในเรื่องความถูกต้องของฟังก์ชันพร้อมกับนำข้อมูลที่เกี่ยวข้องกับเวลาเข้ามาใช้ในการประกอบในการพิจารณาด้วย ซึ่งตามปกติแล้วอุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะต้องมีเวลาหน่วงของการแพร่กระจาย (Propagation Delay Time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาทีก็ตาม แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชันต่างๆ จำนวน 10,000 เกท ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจทำให้การทำงานของวงจรรวมทั้งหมดผิดพลาดไป หรือไม่สามารถที่จะทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

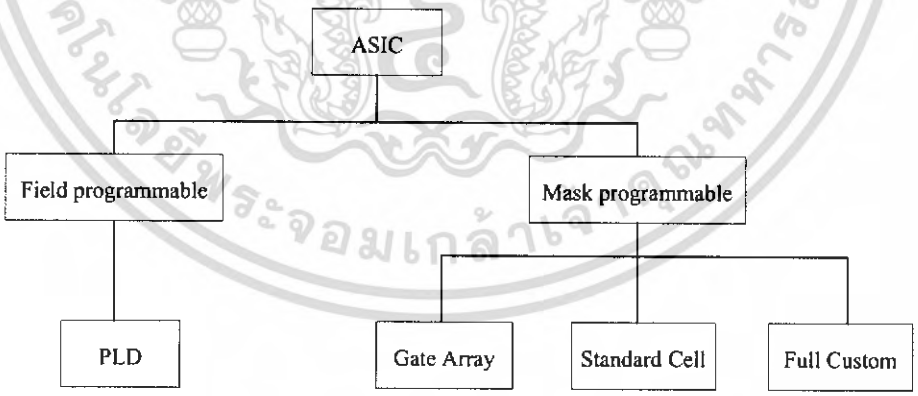
5) การออกแบบทางกายภาพและการวิเคราะห์ คือ ขั้นตอนในการผลิตเป็นวงจรจริง (Technology and device mapping) โดยจะนำข้อมูลที่ได้จากการสังเคราะห์มาใช้ในการผลิตเป็นวงจรรวม ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้นหรืออยู่ในรูปของวงจรรวมเอซิก (ASIC)

6) การตรวจสอบเวลาหลังการออกแบบ คือ การตรวจสอบการทำงานด้วยตัวแปรทางด้านเวลาทั้งหมด เพื่อความถูกต้องของวงจรเป็นครั้งสุดท้ายก่อนนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เนื่องจากในขั้นตอนนี้ วงจรที่ออกแบบจะประกอบด้วยจุดต่อทางอินพุตและเอาท์พุตซึ่งเป็นจุดต่อสำหรับการรับและส่งสัญญาณกับภายนอก

7) การตรวจสอบระบบ คือ การนำวงจรที่ออกแบบไว้ประกอบเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบที่สมบูรณ์ แล้วทำการทดสอบการทำงานทั้งระบบรวมกับอุปกรณ์อื่นๆ อีกครั้ง เพื่อควบคุมคุณภาพของผลิตภัณฑ์

2.8 เอฟพีจีเอ

เทคโนโลยีความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ในปัจจุบัน ทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่างๆ ซึ่งทำให้ลดค่าใช้จ่ายต่างๆได้มาก ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโปรเซสเซอร์ และหน่วยความจำปัจจุบัน ทุกๆครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างระหว่างวงจรรวมและไอซีมาตรฐานมากขึ้น นักออกแบบอุปกรณ์ทางด้านดิจิตอล ได้พิจารณาถึงการผลิตให้มีขนาดมากขึ้นและการผลิตวงจรรวมเอซิก (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวมจะแบ่งตามโครงสร้างออกเป็น 2 กลุ่ม คือ ฟิลด์โปรแกรมเมเบิล (Field programmable) และ แมสโปรแกรมเมเบิล (Mask programmable) ดังแสดงในรูปที่ 2.40



รูปที่ 2.40 ผังการแบ่งกลุ่มของวงจรรวมเอซิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 การออกแบบวงจรเชิงเลขด้วยชิพอุปกรณ์เอฟพีจีเอ

ชิพอุปกรณ์เอฟพีจีเอ เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์เอฟพีจีเอมีฟังก์ชั่นการทำงานตามที่ออกแบบไว้ในการทำชิพอุปกรณ์ ซึ่งเป็นวิธีการออกแบบไอซี (IC : Integrated Circuit) แบบ เซมิคัสตัม (Semi custom) อีกวิธีหนึ่ง เมื่อเทียบกับการทำเอซิกแล้วนั้น ก็มีทั้งข้อดีและข้อเสีย คือ การทำชิพอุปกรณ์เอฟพีจีเอ จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในชิพอุปกรณ์เอฟพีจีเอจะมีจำนวนเกท (gate) ให้ใช้จำนวนจำกัด และการทำชิพอุปกรณ์เอฟพีจีเอก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำชิพอุปกรณ์ก็คือ ระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (download) นั้นน้อยกว่าการทำเอซิกมากและการตรวจสอบหรือแก้ไขการออกแบบที่ทำได้สะดวก

การทำชิพอุปกรณ์เอฟพีจีเอ ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตชิพอุปกรณ์เอฟพีจีเอ ได้เพิ่มความสามารถของชิพอุปกรณ์เอฟพีจีเอ โดยเพิ่มจำนวนองค์ประกอบภายใน หรือปรับปรุงโครงสร้างสถาปัตยกรรมภายในและยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ พีพีอาร์ (PPR: Partitioning Placement and Routing) สำหรับอุปกรณ์นั้นๆด้วย

สำหรับตัวชิพอุปกรณ์เอฟพีจีเอนั้นมีโครงสร้างพื้นฐานเทคโนโลยีที่ใช้สร้าง ตลอดจนเทคนิควิธีการโปรแกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นชิพอุปกรณ์เอฟพีจีเอของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วน ในการใช้งานนั้นชิพอุปกรณ์เอฟพีจีเอสามารถนำไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP: Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้น

2.8.2 ปัจจัยที่ทำให้การออกแบบชิพอุปกรณ์เอฟพีจีเอง่ายและสะดวกรวดเร็ว

1. ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพเพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึงภาษาแอสแซมบลี (Assembly) ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2. มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจรหรือเอชดีแอล เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มันจากนั้นตัวซอฟต์แวร์จะสังเคราะห์ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของคอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 หลักการเบื้องต้นของโครงสร้างเลขคณิตกระจาย

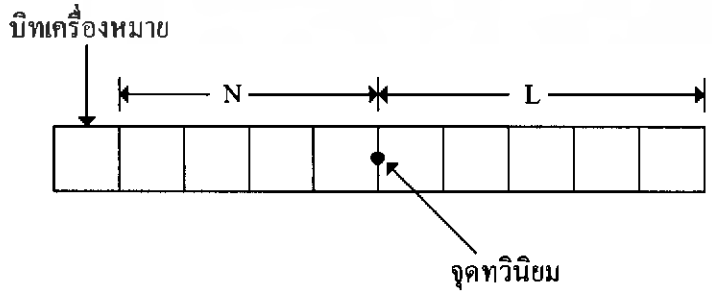
โครงสร้างเลขคณิตกระจาย (Distributed Arithmetic) หรือเรียกอย่างย่อๆว่า DA เป็นการจัดรูปแบบทางคณิตศาสตร์ของสมการที่อยู่ในรูปของผลบวกของผลคูณ (Sum of Product) ของเลขฐานสิบให้กระจายออกเป็นบิตหรือในรูปแบบของเลขฐานสอง เพื่อให้การคำนวณทางคณิตศาสตร์สามารถแปลงเป็นวงจรดิจิตอลอิเล็กทรอนิกส์ได้ โดยจะปรากฏอยู่ในงานทางด้านการประมวลผลสัญญาณเชิงเลข หลักการเบื้องต้นของโครงสร้างเลขคณิตกระจายที่นำมาใช้งานด้านการประมวลผลสัญญาณเชิงเลข คือการแปลงฟังก์ชันถ่ายโอน (Transfer Function) ซึ่งเป็นสมการที่อยู่ในรูปผลบวกของผลคูณระหว่างสัญญาณอินพุตกับค่าสัมประสิทธิ์ของฟังก์ชั่นถ่ายโอนของระบบ โดยในการคูณกันระหว่างค่าสัมประสิทธิ์ของฟังก์ชั่นถ่ายโอนกับสัญญาณอินพุต จะใช้การคูณเลขฐานสองแบบส่วนเติมเต็มสอง (2's complement) และการคูณจะใช้แบบเปิดตาราง (Look-up table) โดยค่าผลบวกของผลคูณระหว่างสัมประสิทธิ์และสัญญาณอินพุตจะถูกเก็บไว้ในหน่วยความจำ EPROM และจะใช้สัญญาณอินพุตเป็นแอดเดรสของ EPROMโดยตรง ทั้งค่าสัมประสิทธิ์ของวงจรกรองและสัญญาณอินพุตจะถูกแทนด้วยเลขส่วนเติมเต็มสอง ดังนั้นโครงสร้างเลขคณิตกระจายจึงมีทฤษฎีพื้นฐานอยู่บนการคูณแบบเลขส่วนเติมเต็มสอง (2's complement Multiplication)

2.9.1 ระบบตัวเลข

สำหรับระบบเชิงเลข ตัวเลขต่างๆจะถูกแทนด้วยเลขฐานสอง ซึ่งโดยทั่วไปมีรูปแบบที่นิยมใช้กันอยู่ 2 รูปแบบ คือ รูปแบบจำนวนโดยตรง (Fixed point format) และรูปแบบจำนวนอิงดรรชนี (Floating point format) ซึ่งรูปแบบจำนวนโดยตรงจะมีวงจรฮาร์ดแวร์ที่ใช้ในการคำนวณที่ง่ายกว่า แต่ให้ค่าจากการคูณค่อนข้างจำกัด ส่วนรูปแบบจำนวนอิงดรรชนีจะสามารถแทนค่าของสัญญาณ คือให้ย่านพลวัตร (Dynamic range) ได้มากกว่า แต่ต้องใช้วงจรฮาร์ดแวร์ที่สลับซับซ้อน แพงกว่า และให้ความเร็วในการประมวลผลที่ลดลง

2.9.1.1 รูปแบบจำนวนโดยตรง

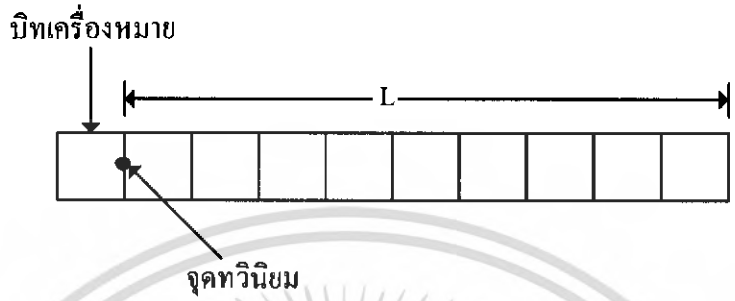
รูปแบบจำนวนโดยตรงปกติจะประกอบไปด้วย 3 ส่วน คือ บิตเครื่องหมาย (Sign bit) 1 บิต บิตจำนวนเต็ม (Integer bit) N บิต และบิตเศษส่วน (Fractional bit) L บิต โดยจะมีจุดทวินิยม (Binary point) อยู่ระหว่างบิตจำนวนเต็มและบิตเศษส่วนดังแสดงในรูปที่ 2.41



รูปที่ 2.41 การจัดรูปแบบจำนวนโดยตรงที่ประกอบด้วยบิตจำนวนเต็มและบิตเศษส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนบิต N เป็นตัวกำหนดย่านพลวัตรที่ต้องการ โดยถ้าเลือกให้มีจำนวนน้อยอาจทำให้เกิดการล้น (Overflow) จากการคำนวณได้ แต่ถ้าเลือกให้มีจำนวนมากความเที่ยงตรงก็จะน้อยลง ซึ่งในการสร้างวงจรกรองสัญญาณเชิงเลขโดยการแทนด้วยรูปแบบจำนวนโดยตรงนั้น นิยมที่จะทำมาตราส่วน (Scaling) เพื่อให้ขนาดของสัญญาณมีค่าอยู่ระหว่าง $-1 \le x < 1$ คือมีบิตเครื่องหมาย 1 บิต และบิตเศษส่วน L บิต ดังแสดงในรูปที่ 2.42



รูปที่ 2.42 การจัดรูปแบบจำนวนโดยตรงที่มีแต่บิตเศษส่วน

โดยทั่วไปเลขฐานสองแบบจำนวนโดยตรงแบ่งออกได้เป็น 3 รูปแบบด้วยกัน คือ (1) แบบขนาดและเครื่องหมาย (Sign magnitude) (2) แบบส่วนเติมเต็มหนึ่ง (1's complement) และ (3) แบบส่วนเติมเต็มสอง (2's complement) โดยคุณลักษณะที่สำคัญบางประการของการแทนตัวเลขด้วยเลขฐานสองแบบจำนวนโดยตรงทั้ง 3 รูปแบบสามารถสรุปได้ดังตารางที่ 2.1

Features	Sign and magnitude	2' complement	1'complement
Range	$-(1-2^{-L}) \le x \le (1-2^{-L})$	$-1 \le x \le (1-2^{-L})$	$-(1-2^{-L}) \le x \le (1-2^{-L})$
Representation of zero	0.000 and 1.000	0.000	0.000 and 1.111
Arithmetic rules	Simple must be kept track of, separately	Simple; negative numbers elegantly handled	Simple, but "end around carry" should be carefully handled
Suitability for serial arithmetic	Not so good	Excellent	Good

ตารางที่ 2.1 คุณสมบัติที่สำคัญของรูปแบบจำนวนโดยตรง

ใน 3 รูปแบบนี้ตัวเลขแบบส่วนเติมเต็มสองเป็นที่นิยมใช้กันมากในระบบการประมวลผลสัญญาณเชิงเลข ทั้งนี้เนื่องมาจาก

1. มีการแทนค่าเลขศูนย์ได้เพียงค่าเดียว

2. การสร้างวงจรฮาร์ดแวร์สำหรับการบวก ลบ และคูณ ของเลขส่วนเติมเต็มสองทำได้ง่ายโดยในการคูณสามารถใช้หลักการเลื่อนและบวก (Shift and add)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในระหว่างผลการบวกย่อย (Partial sum) ของการบวกเลขส่วนเติมเต็มสอง สามหรือสี่จำนวน ถึงแม้อาจจะเกิดการล้น (ตัวทดจากผลการบวกล้นข้ามไปทับบิตเครื่องหมาย) แต่ผลลัพธ์สุดท้ายมักให้ค่าถูกต้องเสมอ ถ้าผลบวกอยู่ในช่วง –1 ถึง $1-2^{-L}$ ดังตัวอย่าง

7/8	0.111	
+4/8	0.100	
11/8	1.011	ผลบวกย่อยที่ผิดเนื่องจากเกิดการล้น
6/8	1.010	
5/8	0.101	ผลบวกที่ถูกต้อง

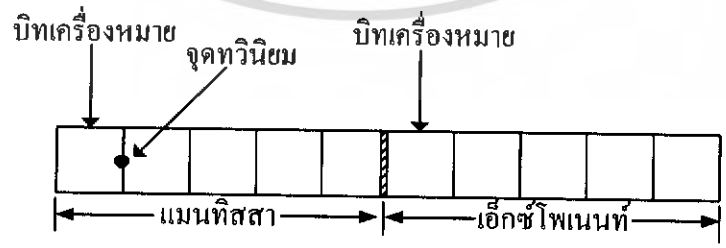
2.9.1.2 รูปแบบจำนวนอิงดรรชนี

รูปแบบจำนวนโดยตรงมีข้อเสียที่สำคัญ 2 ประการ คือ (1) ย่านพลวัตรของตัวเลขมีค่าน้อย เช่น การแทนด้วยเลขส่วนเติมเต็มสอง ค่าที่น้อยที่สุดคือ –1 และค่าที่มากที่สุด คือ $1-2^{-L}$ เปอร์เซ็นต์ความผิดพลาดที่เกิดจากการตัด (Truncation) หรือการปัด (Rounding) จะเพิ่มมากขึ้นเมื่อขนาดของตัวเลขมีค่าลดลง ตัวอย่างเช่น ถ้าจำนวน 0.11011010 และ 0.000110101 ถูกตัดให้จำนวนบิตเศษส่วนเหลือเพียง 4 บิต เปอร์เซ็นต์ความผิดพลาดจะเป็น 4.59 % และ 39.6 % ตามลำดับ โดยข้อเสียนี้สามารถแก้ไขได้โดยการใช้รูปแบบจำนวนอิงดรรชนี ซึ่งตัวเลข X แสดงได้โดย

$$X = M \times 2^{e} \tag{2.70}$$

โดย e เป็นจำนวนเต็ม และ $\frac{1}{2} \le |M| < 1$

M และ e เรียกว่า แมนทิสสา (Mantissa) และ เอ็กซ์โพเนนท์ (Exponent) ตามลำดับ ตัวอย่างเช่น จำนวน 0.00110101 และ 01001.11 สามารถแทนได้โดย 0.110101×2^{-2} และ 0.100111×2^{4} ตามลำดับ ส่วนจำนวนที่มีค่าเป็นลบก็ทำในลักษณะเดียวกัน รูปแบบจำนวนอิงดรรชนีสามารถแสดงได้ดังรูปที่ 2.43 โดยแบ่งเป็น 2 ส่วน คือส่วนหนึ่งสำหรับแมนทิสสา และอีกส่วนสำหรับเอ็กซ์โพเนนท์



รูปที่ 2.43 การจัดรูปแบบจำนวนอิงดรรชนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการใช้จำนวนอิงดรรชนี คือแทนค่าของสัญญาณได้ละเอียดกว่า และแม่นยำกว่าแบบจำนวนโดยตรง แต่การบวก ลบ หรือคูณจะยุ่งยากกว่ามาก วงจรจึงซับซ้อนและแพงกว่าแบบจำนวนโดยตรงมาก นอกจากนี้ความเร็วในการประมวลผลยังช้ากว่าด้วย ดังนั้นสำหรับการประมวลผลแบบเวลาจริง (Real time) จึงนิยมใช้ระบบตัวเลขแบบจำนวนโดยตรง

2.9.2 ทฤษฎีเลขคณิตกระจาย

จากที่ได้กล่าวมาแล้วว่า โครงสร้างเลขคณิตกระจายมีพื้นฐานอยู่บนการคูณแบบเลขส่วนเติมเต็มสอง ดังนั้นในหัวข้อนี้จะได้อธิบายถึงหลักการของการคูณเลขส่วนเติมเต็มสอง

ให้เลขส่วนเติมเต็มสองของ X ซึ่งแทนด้วย $\overline{X}$ และนิยามโดย

$$\overline{x} = \begin{cases} x & \text{ถ้า } X \geq 0 \\ 2 - |x| & \text{ถ้า } X < 0 \end{cases} \tag{2.71}$$

โดย X เป็นเลขที่เป็นเศษส่วน (Fractional number)

ในระบบเลขส่วนเติมเต็มสองจะใช้บิตที่มีนัยสำคัญสูงสุด (MSB) เป็นบิตแสดงเครื่องหมาย โดยถ้าเป็นบวกแทนด้วย "0" และถ้าเป็นลบแทนด้วย "1" ถ้าให้ X แทนด้วยเลขฐานสองขนาด L+1 บิต ดังนั้นรูปแบบของเลขส่วนเติมเต็มสองจะเขียนได้ดังนี้

$$\overline{X} = X_0.X_1.X_2...X_L \tag{2.72}$$

ค่าของ $\overline{X}$ ในรูปของเลขฐานสิบสามารถหาได้ดังนี้

$$X = -X_0 + \sum_{i=1}^{L} X_i 2^{-i} \tag{2.73}$$

จากนั้นพิจารณาผลคูณต่อไปนี้

$$Y = Xm \tag{2.74}$$

ให้ $\overline{Y}$, $\overline{X}$ และ $\overline{m}$ เป็นเลขส่วนเติมเต็มสองของ Y, X และ m ตามลำดับ จากนั้นพิจารณาจากสมการที่ (2.73) และ สมการที่ (2.74) จะได้

$$\begin{aligned} Y &= -Y_0 + \sum_{i=1}^{L} Y_i 2^{-i} \\ &= -X_0 m + \sum_{i=1}^{L} X_i m 2^{-i} \end{aligned} \tag{2.75}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

$$\overline{Y} = \text{ส่วนเติมเต็มสองของ}\left(-X_0 m + 2^{-1}X_1 m + 2^{-2}X_2 m + 2^{-3}X_3 m + \ldots + 2^{-L}X_L m\right)$$
$$= \text{ส่วนเติมเต็มสองของ}\left(-X_0 m + 2^{-1}(X_1 m + \ldots + 2^{-1}(X_{L-1} m + 2^{-1}(X_L m)))\right) \quad (2.76)$$

ต่อไปพิจารณาส่วนเติมเต็มสองของ $2^{-1}U$ โดย

$\overline{U} = U_0.U_1U_2\ldots U_M$; สำหรับ $U \geq 0$ (หรือ $U_0 = 0$)

ส่วนเติมเต็มสองของ $(2^{-1}U) = 2^{-1}\overline{U}$; และสำหรับ $U < 0$ (หรือ $U_0 = 1$)

ส่วนเติมเต็มสองของ $(2^{-1}U) = 2 - \left|2^{-1}U\right| = 1 + 2^{-1}\left(2 - |U|\right) = 1 + 2^{-1}\overline{U}$

ดังนั้นสรุปได้ว่า

$$\text{ส่วนเติมเต็มสองของ} \quad (2^{-1}U) = \begin{cases} 2^{-1}\overline{U} & ; U_0 = 0 \\ 1 + 2^{-1}\overline{U} & ; U_0 = 1 \end{cases} \quad (2.77)$$

สมการที่ (2.77) นี้แสดงให้เห็นได้ว่า ส่วนเติมเต็มสองของ $(2^{-1}U)$ เป็นการเลื่อนข้อมูลของ $\overline{U}$ ไปทางขวา 1 บิต

$$\therefore \text{ส่วนเติมเต็มสองของ} \quad (2^{-1}U) = 2_2^{-1}\overline{U} \quad (2.78)$$

โดย $2_2^{-1}\overline{U}$ แสดงถึงการเลื่อนข้อมูลของ $\overline{U}$ ไปทางขวา 1 บิต แบบเลขส่วนเติมเต็มสอง ซึ่งสัญลักษณ์ 2_2^{-1} (ซึ่งโดยทั่วไปนิยมเขียนเป็น 2^{-1}) เป็นการแสดงว่าในกรณีที่ $\overline{U}$ เป็นเลขบวก ซึ่งบิตเครื่องหมายจะเป็นเลขศูนย์ โดยหลังจากเลื่อนข้อมูลไปทางขวา 1 บิตแล้ว บิตที่มาแทนบิตเครื่องหมายก็ยังคงเป็นเลขศูนย์ ส่วนในกรณีที่ $\overline{U}$ เป็นเลขลบ หลังจากเลื่อนข้อมูลไปทางขวา 1 บิตแล้ว บิตที่มาแทนบิตเครื่องหมายจะเป็นเลขหนึ่ง (จาก $1 + 2^{-1}\overline{U}$) ซึ่งในการสร้างเพื่อใช้งานจริงจำเป็นจะต้องมีวงจรที่ใช้ในการตรวจสอบบิตเครื่องหมาย (Sign digit) ทุกครั้งที่มีการเลื่อนข้อมูล

จากนั้นพิจารณาสมการที่ (2.77) และสมการที่ (2.78) จะได้ว่า

$$\overline{Y} = -X_0\overline{m} + 2^{-1}X_1\overline{m} + 2^{-2}X_2\overline{m} + 2^{-3}X_3\overline{m} + \ldots + 2^{-L}X_L\overline{m}$$

$$= -X_0\overline{m} + 2^{-1}(X_1\overline{m} + \ldots + 2^{-1}(X_{L-1}\overline{m} + 2^{-1}(X_L\overline{m}\ ))) \quad (2.79)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากสมการที่ (2.79) จะเห็นได้ว่าผลคูณจากสมการที่ (2.74) สามารถหาได้โดยการใช้หลักการเลื่อนและบวก (Shift and add) โดยผลลัพธ์ที่ได้จากการคูณแบบเลขส่วนเติมเต็มสอง สามารถหาได้ตามขั้นตอนดังนี้

1. เคลียร์ค่าข้อมูลในแอคคิวมูเลเตอร์รีจิสเตอร์
2. บวก $X_L \overline{m}$ กับค่าที่อยู่ในแอคคิวมูเลเตอร์รีจิสเตอร์
3. เลื่อนค่าที่อยู่ในแอคคิวมูเลเตอร์รีจิสเตอร์ไปทางขวา 1 บิต
4. ทำซ้ำข้อ 2 และ 3 สำหรับค่า $X_{L-1},...,X_1$
5. ลบค่า $X_0 \overline{m}$ ออกจากค่าที่อยู่ในแอคคิวมูเลเตอร์รีจิสเตอร์ (ลบแบบเลขส่วนเติมเต็มสอง)

ตัวอย่างการทำงานตามอัลกอริธึมนี้

$Y = Xm = 0.8125(-0.390625)$ โดยสมมุติให้ใช้แอคคิวมูเลเตอร์รีจิสเตอร์ขนาด 12 บิต

$m = -0.390625$

$\overline{m} = 2 - |m|$ m เป็นเลขลบ

$= 2 - 0.390625$

$= 1.609375$

$\therefore \overline{m} = 1.100111$

$X = 0.8125 = \overline{X}$ $\because X$ เป็นเลขบวก

$\therefore \overline{X} = 0.1101 = X_0.X_1X_2X_3X_4$

โดยมีขั้นตอนการทำงาน ดังตารางต่อไปนี้

การดำเนินการ	ข้อมูลในแอคคิวมูเลเตอร์รีจิสเตอร์
เคลียร์ ACC	0.000 0000 0000
ACC+ $X_4\overline{m}$	1.100 1110 0000
เลื่อนข้อมูลใน ACC ไปทางขวา 1 บิต	1.110 0111 0000
ACC+ $X_3\overline{m}$	1.110 0111 0000
เลื่อนข้อมูลใน ACC ไปทางขวา 1 บิต	1.111 0011 1000
ACC+ $X_2\overline{m}$	1.100 0001 1000
เลื่อนข้อมูลใน ACC ไปทางขวา 1 บิต	1.110 0000 1100
ACC+ $X_1\overline{m}$	1.010 1110 1100
เลื่อนข้อมูลใน ACC ไปทางขวา 1 บิต	1.101 0111 0110
ACC- $X_0\overline{m}$	1.101 0111 0110

ตารางที่ 2.2 ขั้นตอนการคูณเลขส่วนเติมเต็มสอง

$\therefore \overline{Y} = 1.101 \quad 0111 \quad 0110 \quad = Y_0.Y_1Y_2...Y_{11}$

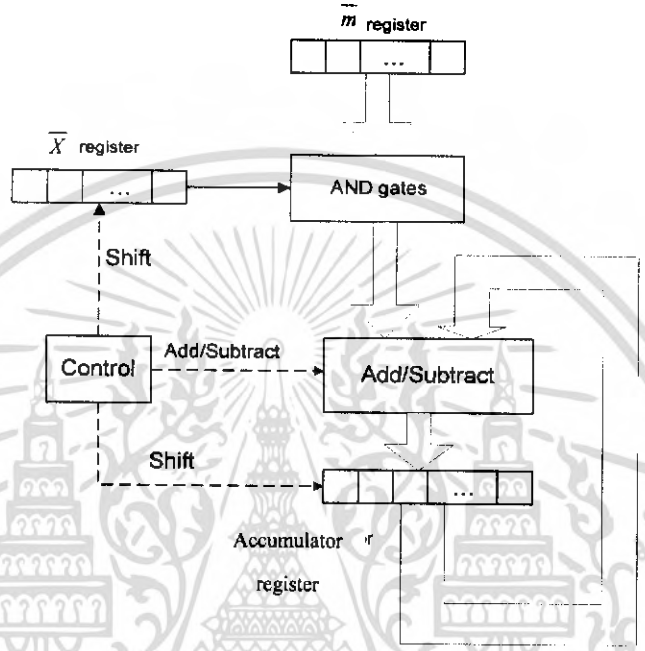
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ $$Y = -Y_0 + \sum_{i=1}^{11} Y_i\, 2^{-i}$$

$$= -1 + \left(2^{-1} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-9} + 2^{-10}\right)$$

$$= -0.3173828125$$

จากอัลกอริธึมดังกล่าวสามารถออกแบบการทำงานและสร้างวงจรแสดงได้ดังรูปที่ 2.44



รูปที่ 2.44 การคูณแบบเลขส่วนเติมเต็มสองโดยใช้เลขคณิตกระจาย

ที่ผ่านมาเป็นหลักการคูณแบบเลขส่วนเติมเต็มสอง ส่วนทฤษฎีเลขคณิตกระจายก็อาศัยหลักการดังกล่าวมาใช้ โดยทำการกระจายสมการที่อยู่ในรูปผลบวกของผลคูณให้แตกออกมาอยู่ในระดับบิต (Bit level) พิจารณาผลบวกของผลคูณต่อไปนี้

$$Y = \sum_{i=0}^{N} m_i\, X_i \qquad (2.80)$$

โดย m_i เป็นค่าสัมประสิทธิ์ที่ซึ่งเป็นค่าคงที่

X_i เป็นข้อมูลอินพุต

ถ้า X_i แต่ละค่าเป็นเลขส่วนเติมเต็มสอง โดย $|X_i| < 1$ สามารถแสดง X_i แต่ละค่าได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_i = -X_{i0} + \sum_{j=1}^{L} X_{ij} 2^{-j} \tag{2.81}$$

โดย X_{ij} = บิตต่างๆของข้อมูล X_i มีค่าเป็น 0 หรือ 1

X_{i0} = บิตแสดงเครื่องหมาย

X_{iL} = บิตที่มีนัยสำคัญต่ำสุด (LSB)

$L+1$ = จำนวนบิตที่แทนข้อมูลอินพุต

แทนค่า X_i ในสมการที่ (2.81) ลงในสมการที่ (2.80) จะได้

$$Y = \sum_{i=0}^{N} m_i \left[-X_{i0} + \sum_{j=1}^{L} X_{ij} 2^{-j} \right] \tag{2.82}$$

เมื่อจัดเทอมของผลบวกใหม่จะได้

$$Y = -X_{i0} \sum_{i=0}^{N} m_i + \sum_{j=1}^{L} X_{ij} 2^{-j} \sum_{i=0}^{N} m_i$$

$$Y = -\sum_{i=0}^{N} X_{i0} m_i + \sum_{i=0}^{N} \sum_{j=1}^{L} X_{ij} m_i 2^{-j} \tag{2.83}$$

จากนั้นกระจายออกให้เป็นระดับบิต ได้ดังนี้

$$\begin{aligned} Y = &-(X_{00}m_0 + X_{10}m_1 + X_{20}m_2 + \ldots + X_{N0}m_N) \\ &+ 2^{-1}(X_{01}m_0 + X_{11}m_1 + X_{21}m_2 + \ldots + X_{N1}m_N) \\ &+ 2^{-2}(X_{02}m_0 + X_{12}m_1 + X_{22}m_2 + \ldots + X_{N2}m_N) \\ &+ \ldots + 2^{-L}(X_{0L}m_0 + X_{1L}m_1 + X_{2L}m_2 + \ldots + X_{NL}m_N) \end{aligned} \tag{2.84}$$

สมการที่ (2.84) นี้ถูกกระจายออกให้อยู่ในรูปผลบวกของผลคูณระหว่างสัมประสิทธิ์กับข้อมูลอินพุตในระดับบิต ซึ่งเป็นนิยามของการคำนวณแบบเลขคณิตกระจาย และเมื่อเปรียบเทียบสมการที่ (2.84) กับสมการที่ (2.79) จะเห็นว่าการคำนวณหาค่า Y ก็ใช้เลขคณิตกระจายนั่นเอง เพียงแต่นำค่าผลคูณย่อย (Partial product) ที่คำนวณไว้ล่วงหน้าแล้วสำหรับแต่ละค่าที่สอดคล้องกับแต่ละบิตของข้อมูลอินพุตไปเก็บไว้ในตารางเปิดดู ซึ่งเป็นหน่วยความจำ EPROM และใช้ข้อมูลอินพุตเป็นแอดเดรสของหน่วยความจำ เพื่อนำค่าในตารางเปิดดูมาผ่านขั้นตอนการคำนวณตามบัฟอัลกอริธึม ซึ่งค่าในตารางเปิดดูสามารถแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit pattern ของข้อมูลอินพุต X_{Nj} ..... X_{2j} X_{1j} X_{0j}	ผลคูณย่อยที่เก็บไว้ในตารางเปิดดู
0 ..... 0 0 0	0
0 ..... 0 0 1	m_0
0 ..... 0 1 0	m_1
0 ..... 0 1 1	$m_1 + m_0$
0 ..... 1 0 0	m_2
0 ..... 1 0 1	$m_2 + m_0$
0 ..... 1 1 0	$m_2 + m_1$
0 ..... 1 1 1	$m_2 + m_1 + m_0$
N	N
1 ..... 1 1 1	$m_N + m_{N-1} + ... + m_2 + m_1 + m_0$

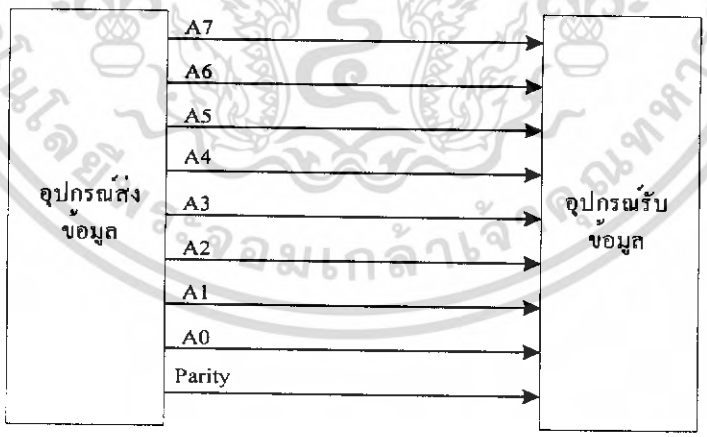
ตารางที่ 2.3 ค่าผลคูณย่อยที่เก็บไว้ในตารางเปิดดูที่กำหนดโดยข้อมูลอินพุต

2.10 พอร์ตอนุกรม

2.10.1 การสื่อสารข้อมูล

2.10.1.1 การสื่อสารข้อมูลแบบขนาน (Parallel Communication)

การสื่อสารข้อมูลแบบขนานคือการสื่อสารแบบที่ส่งข้อมูลพร้อมๆกัน n บิต ผ่านสายสัญญาณ n เส้น สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบขนานได้ดังรูปที่ 2.45



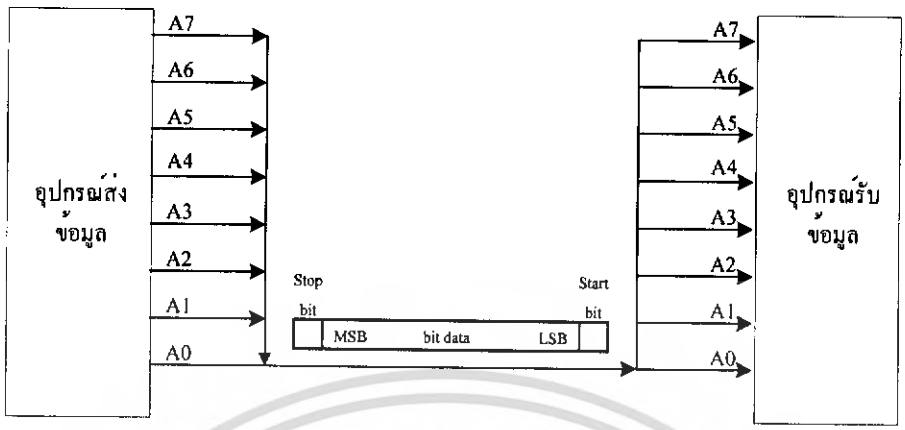
รูปที่ 2.45 บล็อกไดอะแกรมรูปแบบการสื่อสารข้อมูลแบบขนาน

2.10.1.2 การสื่อสารข้อมูลแบบอนุกรม (Serial Communication)

การสื่อสารข้อมูลแบบอนุกรม คือ การสื่อสารแบบที่ส่งข้อมูลทีละบิต ผ่านสายสัญญาณเส้นเดียว จนครบจำนวนข้อมูลที่ต้องการ โดยเฟรมของการสื่อสารข้อมูลแบบอนุกรมประกอบด้วย บิตสตาร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(start bit) บิตสต็อป (stop bit) บิตข้อมูล (data bit) สามารถแสดงรูปแบบการสื่อสารข้อมูลแบบอนุกรมได้ดังรูปที่ 2.46



รูปที่ 2.46 บล็อกไดอะแกรมรูปแบบการสื่อสารข้อมูลแบบอนุกรม

อัตราบอดเรท (Baud Rate) ในการสื่อสาร คือ ความเร็วในการรับ-ส่งข้อมูลแบบอนุกรมมีหน่วยเป็นบิตต่อวินาที (bit/sec) ซึ่งจะบอกถึงจำนวนบิตที่รับส่ง ในเวลา 1 วินาที เช่น ส่งข้อมูลด้วยอัตรา 9600 บิตต่อวินาที หมายถึง เวลา 1 วินาที รับส่งข้อมูลได้ 9600 บิต รวมทั้งบิตข้อมูล (data bit) บิตสตาร์ท (Start bit) บิตสต็อป (Stop bit) ด้วย

2.10.2 การอินเตอร์เฟสตามมาตรฐาน RS-232

มาตรฐาน RS-232 เป็นมาตรฐานที่ได้รับการพัฒนามานานและมีการใช้งานอย่างแพร่หลาย ทั้งการใช้ RS-232 เชื่อมต่อ DTE (Data Terminal Equipment) เช่น คอมพิวเตอร์หรือเทอร์มินัล (Terminal) เป็นต้น เข้ากับ DCE (Data Communication Equipment) เช่น โมเด็ม (Modem) ทีเอ อะแด็ปเตอร์ (TA adapter) พล็อตเตอร์ (Plotter) เป็นต้น ตัวอย่างการเชื่อมต่อ เช่น การต่อเทอร์มินัลเข้ากับโมเด็ม

มาตรฐาน RS-232 จะใช้สัญญาณเส้นเดียวในการส่งสัญญาณโดยจะสัญญาณจะส่งไปในทิศทางเดียวกัน สำหรับการแทนค่าแรงดันในการส่งสัญญาณเป็นดังนี้

- สัญญาณของลอจิก “1” แทนด้วยระดับแรงดันไฟฟ้าระหว่าง -3 ถึง -25 โวลต์
- สัญญาณของลอจิก “0” แทนด้วยระดับแรงดันไฟฟ้า ระหว่าง 3 ถึง 25 โวลต์
- ส่วนแรงดันไฟฟ้าระหว่าง 3 ถึง -3 โวลต์ ไม่มีการนิยาม

การเชื่อมต่อกับพอร์ตสื่อสารของคอมพิวเตอร์ส่วนบุคคลจะเลือกใช้พอร์ตสื่อสารแบบอนุกรม 9 ขา (DB-9) ซึ่งสามารถทำการส่งสัญญาณข้อมูลได้ตามมาตรฐาน RS-232 โดยลักษณะของคอนเน็คเตอร์แบบ DB-9 สามารถแสดงได้ดังรูปที่ 2.47 และการเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9 สามารถแสดงได้ดังตารางที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.47 แสดงลักษณะของคอนเน็กเตอร์แบบ DB-9

ตำแหน่งขาของ DB-9	สัญญาณ
1	Data Carrier Detect : DCD
2	Received Data : RxD
3	Transmitted Data : TxD
4	Data Terminal Ready : DTR
5	Signal Ground : GND
6	Data Set Ready : DSR
7	Request To Send : RST
8	Clear To Send : CTS
9	Ring Indicator : RI

ตารางที่ 2.4 การเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์แบบ DB-9

เปรียบเทียบข้อดีข้อเสียของการสื่อสารข้อมูลแบบอนุกรมและขนาน

- การสื่อสารข้อมูลแบบอนุกรมสามารถสื่อสารได้ระยะทางที่ไกลกว่า

- การสื่อสารข้อมูลแบบอนุกรมใช้สายสัญญาณที่ประหยัดกว่า

- การสื่อสารข้อมูลแบบขนานสามารถสื่อสารข้อมูลได้ครั้งละจำนวนมากและรวดเร็วกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

การสร้างตัวแปลงเวฟเลตจะต้องคำนวณและออกแบบส่วนประกอบต่าง ๆ ดังต่อไปนี้

3.1 การออกแบบตัวแปลงเวฟเลต

3.2 การออกแบบโครงสร้างเลขคณิตกระจาย

3.3 การออกแบบอุปกรณ์รับส่งข้อมูลผ่านพอร์ทอนุกรม

3.1 การออกแบบตัวแปลงเวฟเลต

3.1.1 สัมประสิทธิ์ที่ใช้สำหรับการแปลงเวฟเลตและการแปลงกลับเวฟเลต

เลือกใช้สัมประสิทธิ์ของตัวกรองสัญญาณแบบ 4 Tap Orthonormal Filter with Two Vanishing Moments (DB 4)

n	$H_0(n)$	$H_1(n)$
0	0.48296291314453	0.12940952255126
1	0.83651630373781	0.22414386804201
2	0.224143868042010	-0.83651630373781
3	-0.12940952255126	0.48296291314453

ตารางที่ 3.1 ค่าสัมประสิทธิ์ของดาร์บีชี 4 ค่า ของการแปลงเวฟเลต

เพื่อให้ได้ผลตรงตามหลักการของการสร้างกลับคืนอย่างสมบูรณ์ (Perfect Reconstruct) จะได้ค่าสัมประสิทธิ์ในการแปลงกลับเวฟเลตที่สามารถหาได้จากสมการที่ 2.44 แสดงดังตารางที่ 3.2

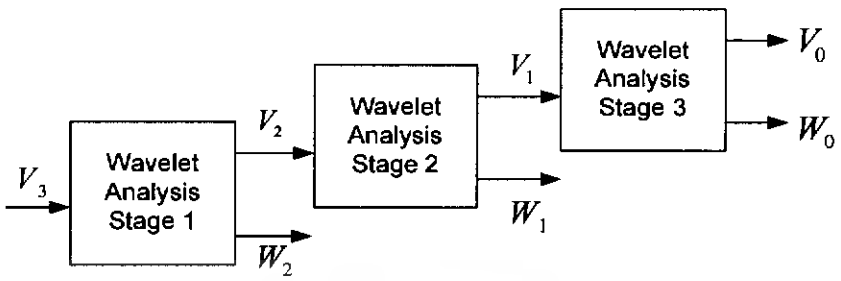
n	$F_0(n)$	$F_1(n)$
0	-0.12940952255126	0.48296291314453
1	0.22414386804201	-0.83651630373781
2	0.83651630373781	0.22414386804201
3	0.48296291314453	0.12940952255126

ตารางที่ 3.2 ค่าสัมประสิทธิ์ของดาร์บีชี 4 ค่า ของการแปลงกลับเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

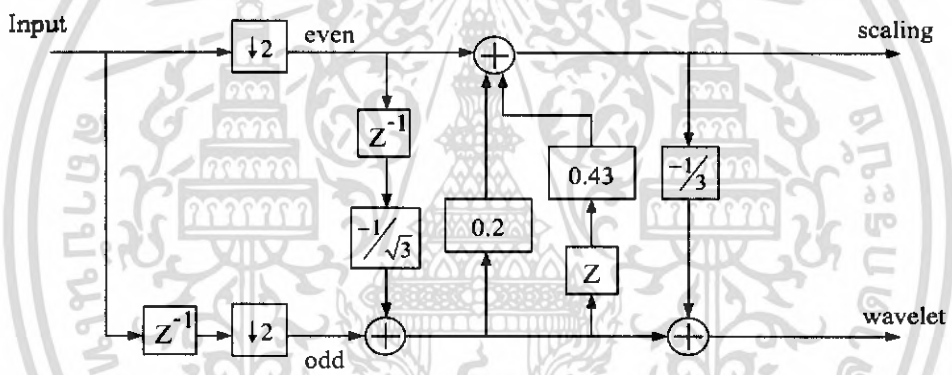
3.1.2 การแปลงเวฟเลต 3 ระดับ

จะประกอบด้วยการแปลงเวฟเลต 3 ระดับ ดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของการแปลงเวฟเลต 3 ระดับ

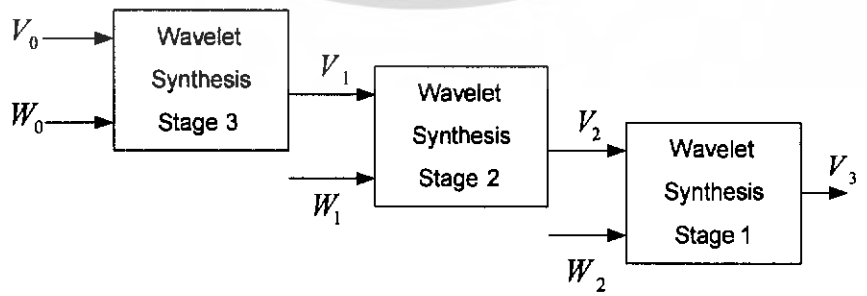
โดยที่ในแต่ละระดับจะมีโครงสร้างภายในเป็นโครงสร้างของการแปลงเวฟเลตแบบ lifting scheme เหมือนกันในทุกระดับ ซึ่งมีส่วนประกอบภายในดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างภายในของการแปลงเวฟเลตโดยใช้ lifting scheme

3.1.3 การแปลงกลับเวฟเลต 3 ระดับ

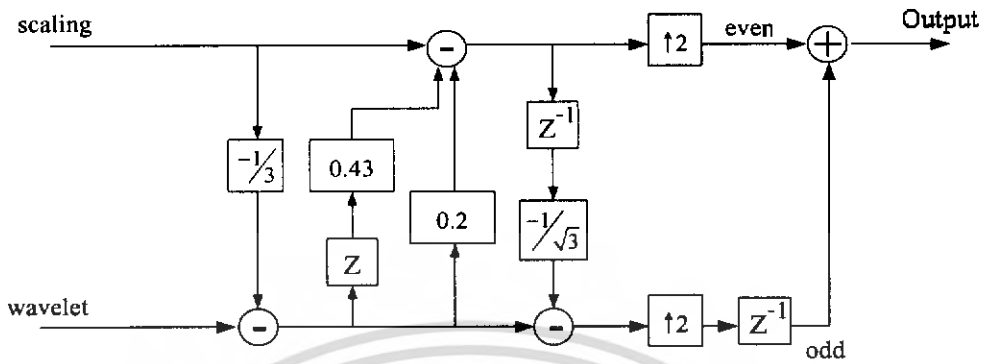
จะประกอบด้วยการแปลงกลับเวฟเลต 3 ระดับ ดังรูปที่ 3.3



รูปที่ 3.3 บล็อกไดอะแกรมของการแปลงกลับเวฟเลต 3 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ในแต่ละระดับจะมีโครงสร้างภายในเป็นโครงสร้างของฟิลเตอร์แบบ lifting scheme เหมือนกันในทุกระดับ พิจารณาโครงสร้างในระดับที่ 1 มีส่วนประกอบภายใน ดังรูปที่ 3.4

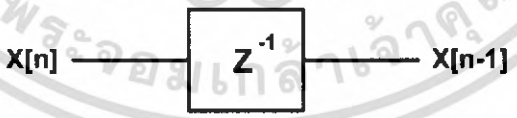


รูปที่ 3.4 โครงสร้างภายในของการแปลงกลับเวฟเลตโดยใช้ lifting scheme

ในการแปลงเวฟเลตและการแปลงกลับเวฟเลตในระดับที่ 2 และระดับที่ 3 จะใช้โครงสร้างภายในเหมือนในระดับที่ 1 แต่ต่างกันที่อินพุตที่เข้ามาในระดับที่ 3 จะมีคาบเวลาเป็นสองเท่าของระดับที่ 2 และอินพุตที่เข้ามาในระดับที่ 2 จะมีคาบเวลาเป็นสองเท่าของอินพุตที่เข้ามาในระดับที่ 1 ซึ่งการใช้ความถี่สัญญาณนาฬิกาที่แตกต่างนี้ทำให้ไม่ต้องออกแบบวงจรในระดับที่ 2 และระดับที่ 3 เพียงแต่ต้องเพิ่มการควบคุมสัญญาณนาฬิกาที่ป้อนให้กับระดับที่ 2 ให้เป็นสองเท่าของระดับที่ 1 และเพิ่มการควบคุมสัญญาณนาฬิกาในระดับที่ 3 ให้เป็นสองเท่าของระดับที่ 2

3.1.4 ส่วนประกอบของวงจรการแปลงเวฟเลตโดยใช้ Lifting scheme

3.1.4.1 วงจรหน่วงเวลา (Delay)

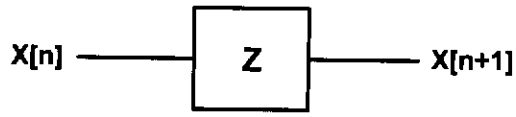


รูปที่ 3.5 บล็อกไดอะแกรมของวงจรหน่วงเวลา

วงจรหน่วงเวลามีหลักการทำงาน คือ จะเลื่อนเวลาของอินพุตที่เข้ามาให้มีเวลาที่ช้ากว่าเดิมอยู่ 1 หน่วย ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.2 วงจรฟอร์เวิร์ดเวลา (Forward)



รูปที่ 3.6 บล็อกไดอะแกรมของวงจรฟอร์เวิร์ดเวลา

วงจรฟอร์เวิร์ดเวลามีหลักการทำงาน คือ จะเลื่อนเวลาของอินพุตที่เข้ามาให้มีเวลาที่เร็วกว่าเดิมอยู่ 1 หน่วย ดังรูปที่ 3.6

3.1.4.3 วงจรสุ่มค่าตัวอย่าง (Down-sampling)

รูปที่ 3.7 บล็อกไดอะแกรมของวงจรสุ่มค่าตัวอย่าง

วงจรสุ่มค่าตัวอย่าง คือ วงจรที่ลดขนาดของอินพุตลงเหลือครึ่งหนึ่ง แสดงดังรูปที่ 3.7

3.1.4.4 วงจรเพิ่มค่าตัวอย่าง (Up-sampling)

รูปที่ 3.8 บล็อกไดอะแกรมของวงจรเพิ่มค่าตัวอย่าง

จากรูปที่ 3.8 เมื่อมีอินพุตเข้ามา 1 ค่า เอาต์พุตจะมีค่าของอินพุตตัวเดิมพร้อมด้วยการแทรกค่าศูนย์เข้าไประหว่างค่าเดิม ทำให้ขนาดของเอาต์พุตที่ได้มีขนาดเป็น 2 เท่าเมื่อเทียบกับขนาดของอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.5 วงจรบวก (Adder)

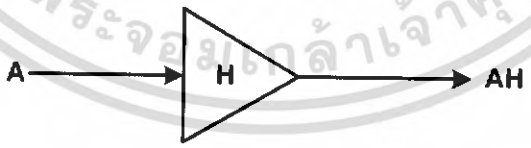
รูปที่ 3.9 บล็อกไดอะแกรมของวงจรบวก

ในการบวกข้อมูล 14 บิต จะได้ผลลัพธ์ของการบวกตามตารางความจริง ดังตารางที่ 3.3

Input 1	Input 2	Carry in	Output	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ตารางที่ 3.3 ตารางความจริงของวงจรบวก

3.1.4.6 วงจรคูณ (Multiplier)



รูปที่ 3.10 บล็อกไดอะแกรมของวงจรคูณ

การทำงานของวงจรคูณเมื่อมีอินพุตเข้ามาจะคูณค่าอินพุตนั้นกับค่าคงที่ที่ได้กำหนด ซึ่งจะได้ผลลัพธ์เป็นผลคูณระหว่างอินพุตกับค่าคงที่ ดังรูปที่ 3.10

ในแต่ละระดับการแปลงเวฟเลตและการแปลงกลับเวฟเลต จะใช้วงจรคูณระดับละ 4 วงจร สำหรับการคูณกับสัมประสิทธิ์ 4 ค่า และเนื่องจากการแปลงเวฟเลตสามารถทำได้หลายระดับ จึงจะต้องใช้

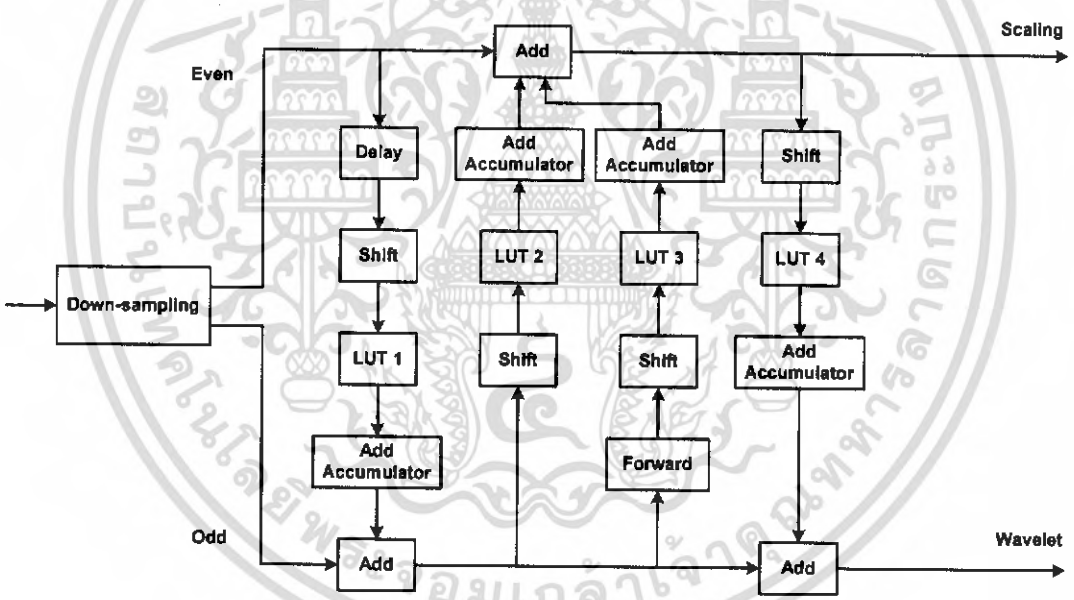
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรคูณจำนวนเป็นจำนวนมาก ในการจำลองการทำงานรวมถึงการสร้างจริงของวงจรคูณจะใช้ทรัพยากรลิจิกเกตจำนวนมาก ดังนั้นจึงนำหลักการเลขคณิตกระจายมาประยุกต์ใช้ ซึ่งหลักการนี้จะใช้จำนวนทรัพยากรลอจิกเกตที่น้อยกว่าวงจรคูณแบบโดยตรง

3.2 การออกแบบโครงสร้างเลขคณิตกระจาย

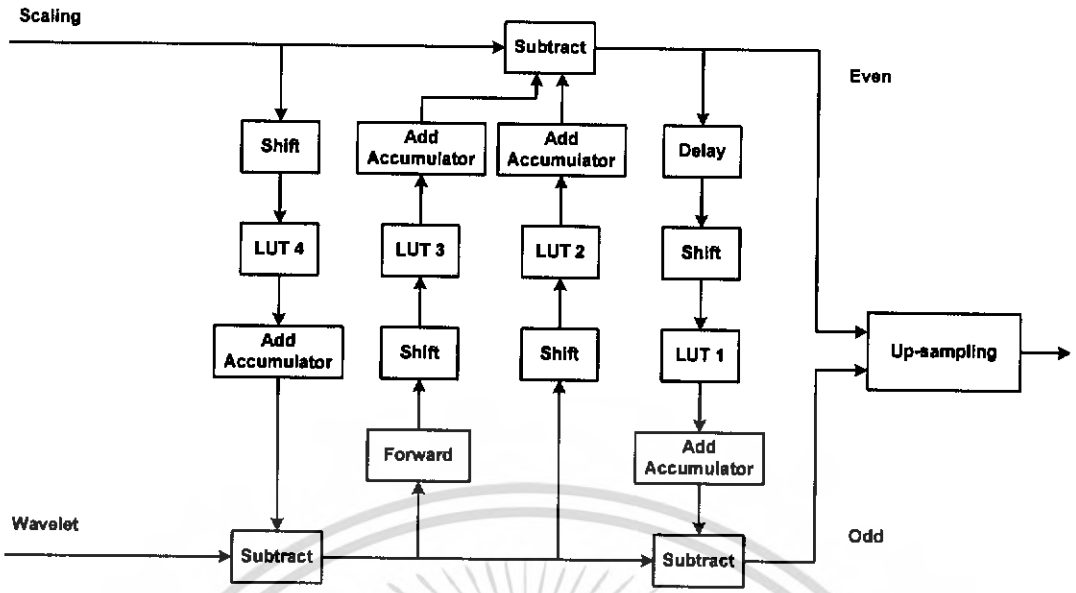
3.2.1 การประยุกต์ใช้โครงสร้างเลขคณิตกระจายกับการแปลงเวฟเลตที่มีโครงสร้างแบบ Lifting scheme

การแปลงเวฟเลตและการแปลงกลับโดยใช้โครงสร้างของฟิลเตอร์แบบ Lifting Scheme ดังรูปที่ 3.2 และ 3.4 มีจำนวนของวงจรคูณเป็นจำนวนมาก ทำให้เกิดการใช้ทรัพยากรลอจิกเกตสิ้นเปลือง จึงมีการนำโครงสร้างของเลขคณิตกระจายหรือ (Distributed Arithmetic : DA) มาประยุกต์ใช้ ซึ่งคุณสมบัติของเลขคณิตกระจายจะทำงานอยู่ในรูปแบบของผลบวกของผลคูณ (Sum of Products) โดยไม่ใช้การคูณโดยตรงในการประมวลผลของข้อมูล ซึ่งแนวความคิดนี้เหมาะสมกับรูปแบบของการคำนวณของการแปลงเวฟเลต โดยสามารถเปลี่ยนให้อยู่ในรูปโครงสร้างเลขคณิตกระจายได้ดังรูปที่ 3.11 และ 3.12



รูปที่ 3.11 การแปลงเวฟเลตโดยนำหลักการของ DA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

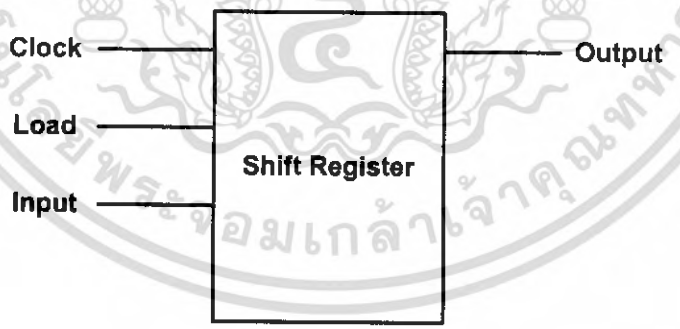


รูปที่ 3.12 การแปลงกลับเวฟเลตโดยนำหลักการของ DA มาใช้

3.2.2 โครงสร้างเลขคณิตกระจายแบบอนุกรม

โครงสร้างเลขคณิตกระจายแบบอนุกรมที่ใช้เป็นส่วนประกอบของการแปลงเวฟเลต และการแปลงกลับเวฟเลต ดังรูปที่ 3.11 และ 3.12 จะประกอบด้วยวงจรเลื่อนข้อมูลแบบเข้าขนานออกอนุกรม ตารางเปิดดู วงจรสะสมค่า วงจรบวกและลบ ดังต่อไปนี้

3.2.2.1 วงจรเลื่อนข้อมูลแบบเข้าขนานออกอนุกรม



รูปที่ 3.13 วงจรเลื่อนข้อมูลแบบเข้าขนานออกอนุกรม

การทำงานของวงจรเมื่อมีสัญญาณอินพุทเข้ามาและสัญญาณที่ขาโหลดมีค่า '1' จะโหลดข้อมูลเข้ามาเก็บในวงจรเลื่อนข้อมูล เมื่อขาโหลดมีค่า '0' และมีสัญญาณนาฬิกาจะทำให้วงจรเลื่อนข้อมูล เลื่อนข้อมูลออกเอาท์พุททีละ 1 บิต เริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ถึงบิตที่มีนัยสำคัญสูงสุด (MSB) ตามจังหวะของสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 ตารางเปิดดู

การแปลงเวฟเลตจะเก็บค่าที่ใช้ในการคูณไว้ในตารางเปิดดูหรือ LUT ซึ่งมีความยาว 14 บิต ในการแปลงเวฟเลตแบบ lifting scheme จะมีการคูณ 4 ครั้งต่อการคำนวณแต่ละระดับ ดังนั้นในการออกแบบวงจรแปลงเวฟเลตจะใช้ LUT 4 ตัว สำหรับเก็บค่าในการคูณทั้ง 4 ครั้ง ส่วนการแปลงกลับเวฟเลตจะใช้ LUT 4 ตัว สำหรับการคูณ 4 ครั้งเช่นเดียวกัน แต่เนื่องจากว่าในส่วนนี้จะมีวงจรลบซึ่งจะใช้วงจรบวกแทนวงจรลบ จึงต้องเปลี่ยนค่าที่เก็บใน LUT สำหรับการแปลงเวฟเลตให้เป็นค่าลบแทน ซึ่งสามารถออกแบบ LUT ของทั้ง 2 ส่วน ได้ดังตารางที่ 3.4 และ 3.5

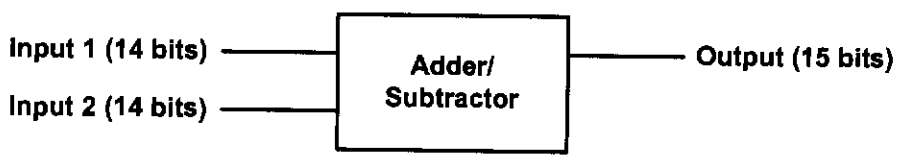
Input Bit	0	1
LUT 1	0	10110110000110
LUT 2	0	00011001101110
LUT 3	0	00110111011011
LUT 4	0	11010101010101

ตารางที่ 3.4 ผลลัพธ์ของตารางเปิดดูสำหรับการแปลงเวฟเลต

Input Bit	0	1
LUT 1	0	01001001111010
LUT 2	0	11100110010010
LUT 3	0	11001000100101
LUT 4	0	00101010101011

ตารางที่ 3.5 ผลลัพธ์ของตารางเปิดดูสำหรับการแปลงกลับเวฟเลต

3.2.2.3 วงจรบวกและลบ



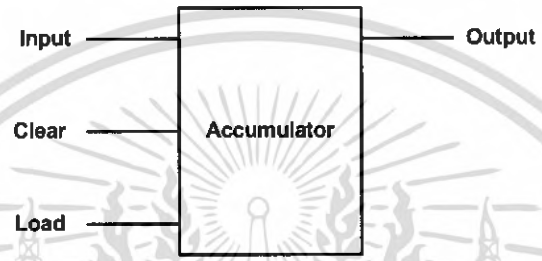
รูปที่ 3.14 วงจรบวกและลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรบวกและลบ (Adder/Subtractor) จะรับค่าที่อ่านจาก LUT เข้ามาทำการบวกหรือลบกับค่าในวงจรสะสมค่า ซึ่งสามารถเลือกโหมดว่าต้องการบวกหรือลบได้ ผลลัพธ์จะมี 15 บิต โดยบิตที่มีนัยสำคัญสูงสุดหรือบิตที่ 15 จะเป็นบิตเช็คเครื่องหมายเพื่อนำไปเลื่อนแทนที่ในการเลื่อนขวาของข้อมูลในวงจรสะสมค่า

3.2.2.4 วงจรสะสมค่า

วงจรสะสมค่าทำหน้าที่สะสมค่าที่ได้จากการบวกหรือลบ เข้ามาเก็บแล้วทำการเลื่อนขวา 1 บิต แล้วส่งกลับมาบวกหรือลบกับค่าต่อไปที่เข้ามา

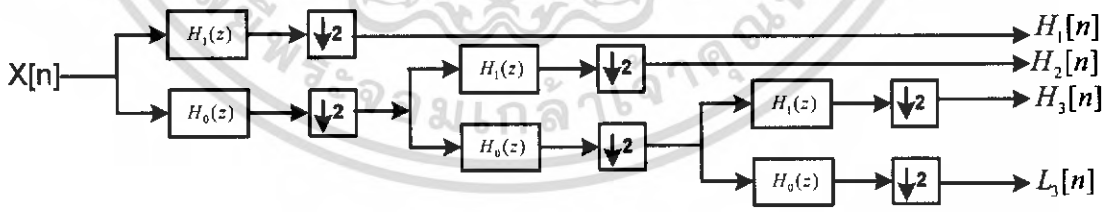


รูปที่ 3.15 วงจรสะสมค่า

3.2.3 โครงสร้างเลขคณิตกระจายแบบขนาน

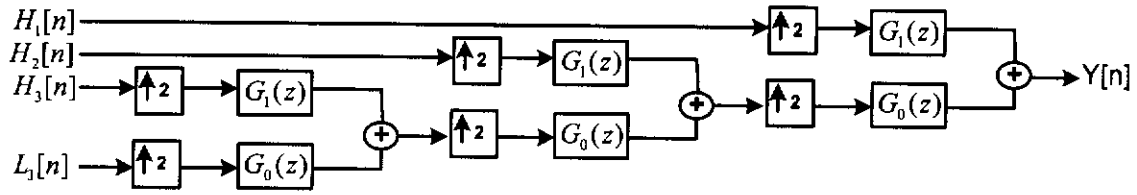
3.2.3.1 การสร้างโครงสร้างเลขคณิตกระจายแบบขนาน

สมการการแปลงเวฟเลตแบบไม่ต่อเนื่อง สามารถคำนวณให้มีประสิทธิภาพได้โดยการใช้ฟิลเตอร์แบงค์ ซึ่งแสดงในรูปที่ 3.16 โดยในที่นี้จะใช้โครงสร้างของเลขคณิตกระจายแบบขนาน (Parallel DA) หรือเรียกอย่างย่อๆว่า PDA ของวงจรกรองเอฟไอาร์



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

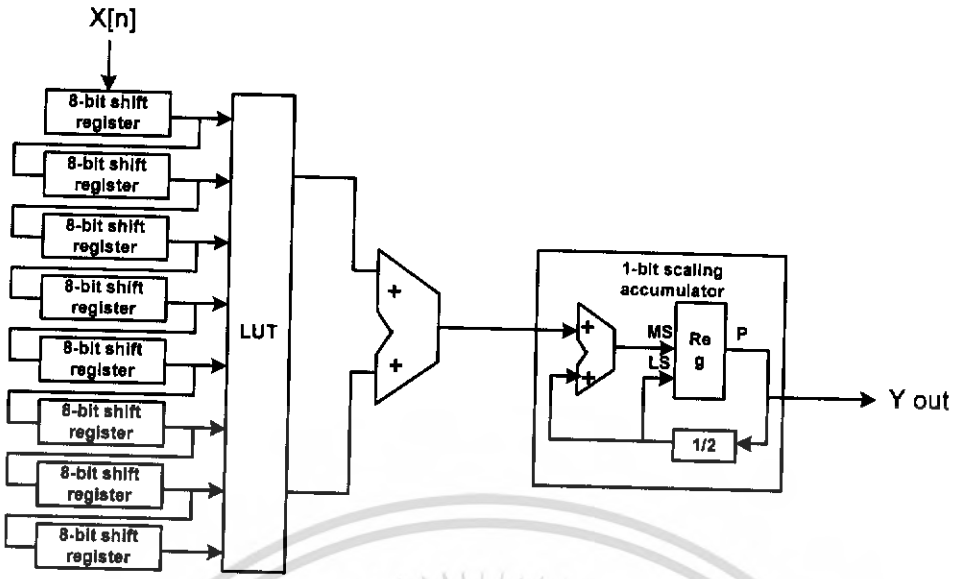
รูปที่ 3.16 โครงสร้างฟิลเตอร์แบงค์ (ก) การแปลงเวฟเลต (ข) การแปลงกลับเวฟเลต

3.2.3.2 โครงสร้างวงจรกรองเอฟไอาร์โดยการใช้เลขคณิตกระจายแบบขนาน

วงจรกรองทั้งหมดที่เป็นโครงสร้างที่แสดงในรูปที่ 3.16 สามารถสร้างได้โดยการใช้วงจรกรองเอฟไอาร์เนื่องจากมีความเสถียรภาพ ดังนั้นในการสร้างการแปลงเวฟเลตแบบไม่ต่อเนื่อง จึงมักจะใช้โครงสร้างเอฟไอาร์แบบโดยตรง ซึ่งในวงจรกรองจะประกอบไปด้วย วงจรหน่วงเวลา วงจรบวก และวงจรคูณ

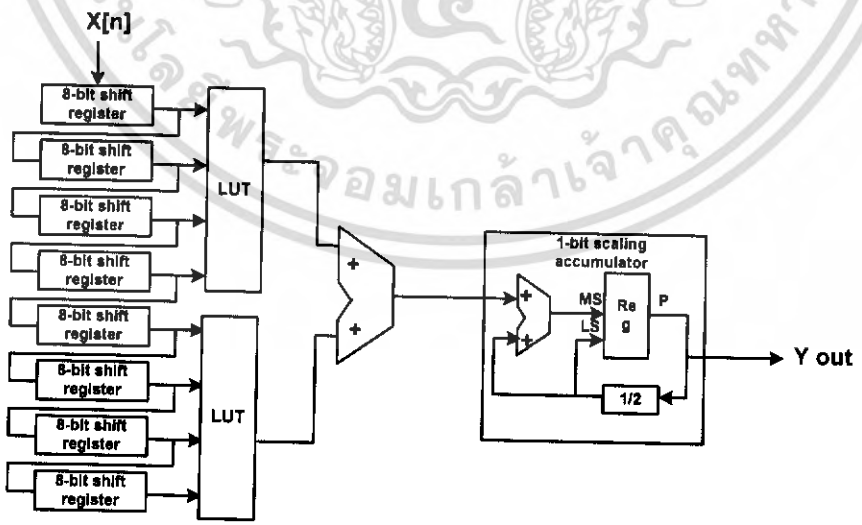
ในการออกแบบเบื้องต้นนี้จะพิจารณา DA ของเวฟเลตที่ใช้วงจรกรองเอฟไอาร์แบบดาร์บีชีอันดับ 8 (Daubechies 8-tap wavelet FIR filter) ซึ่งประกอบด้วยตารางเปิดดู (LUT) วงจรเลื่อนข้อมูล(shift register) และวงจรสะสมค่า (scaling accumulator) ซึ่งแสดงในรูปที่ 3.17 โดยที่ตารางเปิดดูจะเก็บผลที่เป็นไปได้ทั้งหมดของสัมประสิทธิ์เวฟเลตดาร์บีชี 8 ค่า ข้อมูลอินพุตจะนำไปประมวลผลโดยจะแยกบิตที่ขนานกันออกเป็นบิตโดยใช้วงจรเลื่อนบิตแบบอนุกรม (bit-serial shift register cascade) โดยจะปล่อย 1 บิตที่หนึ่งเวลา ชุดอนุกรมจะเก็บข้อมูลอินพุตทั้งหมดในรูปแบบของบิตอนุกรม (bit-serial) และจะเป็นตัวกำหนดผลคูณภายใน (inner product) เพื่อให้ได้บิตเอาต์พุตของวงจรเลื่อน โดยการใช้ข้อมูลอินพุตเป็นตัวชี้ตำแหน่งในตารางเปิดดู ผลลัพธ์ที่ได้จากตารางเปิดดูจะนำไปบวกโดยใช้วงจรสะสมค่า เมื่อบวกค่าจนครบชุดอนุกรมใน 1 ชุด ก็จะส่งค่าผลลัพธ์สุดท้ายออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 การสร้าง DA โดยใช้วงจรกรองเอฟไอาร์ดาร์บีชี

เนื่องจากขนาดของตารางเปิดดูของโครงสร้างเลขคณิตกระจายหนึ่ง ๆ จะเพิ่มแบบเอกซ์โพเนนเชียล (Exponentially) โดยขึ้นกับจำนวนสัมประสิทธิ์ ดังนั้นการเข้าถึงทางเวลาของตารางเปิดดู ซึ่งอาจทำให้เกิดการชนกันในการใส่อินพุทเข้าไปได้ โดยจะมีผลต่อความเร็วของระบบทั้งหมดเนื่องจากขนาดของตารางเปิดดูที่มีขนาดใหญ่ ดังนั้นเราจะทำการจัดแบ่งตารางเปิดดู 8 บิต ที่แสดงในรูปที่ 3.17 ไปเป็นตารางเปิดดู 4 บิต สองอัน และรวมค่าผลลัพธ์เหล่านั้นโดยการใช้วงจรสะสมค่า 2 อินพุต ตารางเปิดดูที่ทำการแบ่งแล้วแสดงไว้ในรูปที่ 3.18 ซึ่งจะทำให้ขนาดของความจุลดลง นอกจากนี้การแบ่งตารางเปิดดูที่ใหญ่กว่าไปยังตารางเปิดดูสองอันที่เล็กกว่าในแบบขนานก็จะลดเวลาในการเข้าถึงการทำงานด้วย

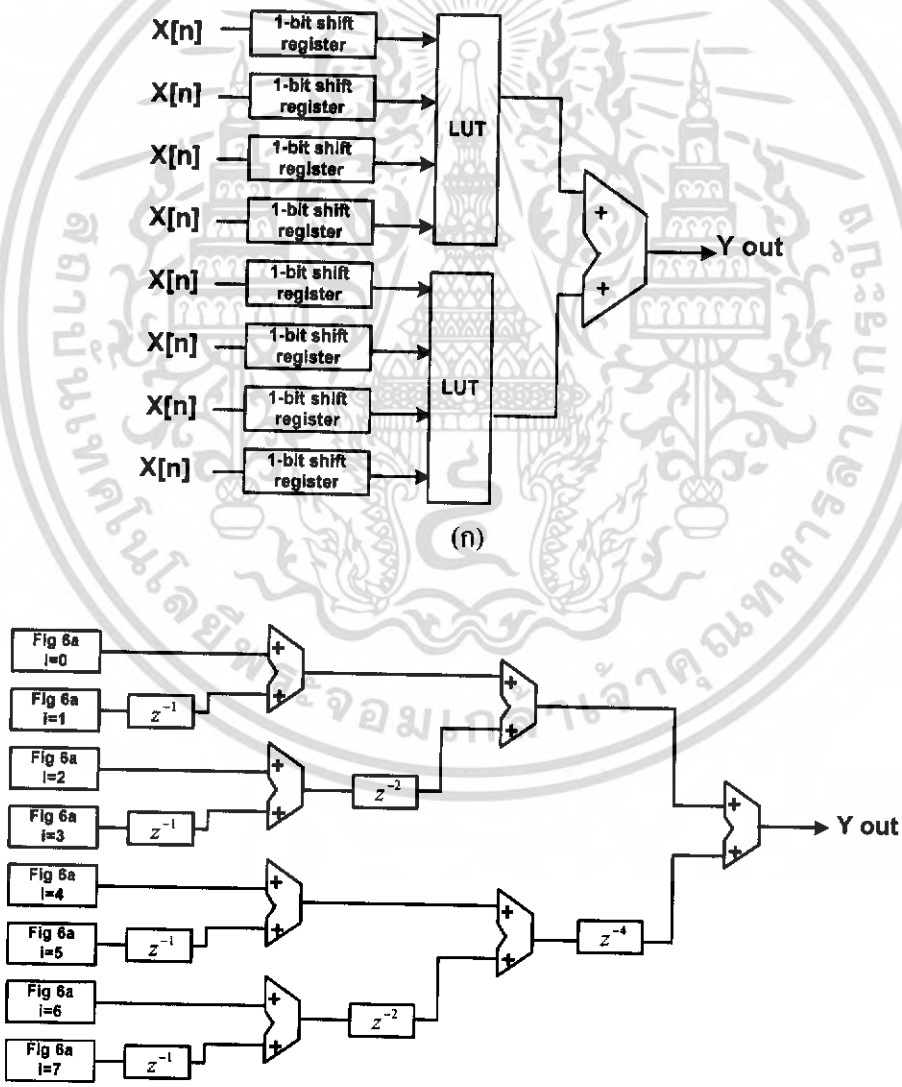


รูปที่ 3.18 การสร้าง DA ที่มีการแบ่งตารางเปิดดูของวงจรกรองเอฟไอาร์ดาร์บีชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำวงจรกรองเอฟไอาร์ที่เป็นโครงสร้างเลขคณิตกระจายแบบอนุกรม (Serial DA) หรือเรียกอย่างย่อว่า SDA ให้ทำงานเป็นแบบขนานดังแสดงในรูปที่ 3.18 เปรียบเสมือนกับการแบ่งค่าตัวอย่างอินพุต (Input Sample) ไปเป็นค่าตัวอย่างย่อยจำนวน m ตัว (m-Sub Sample) และประมวลผลค่าตัวอย่างย่อยเหล่านี้ในแบบขนาน ดังนั้นการทดลองแบบขนานต้องการจำนวน m ครั้งซึ่งต้องใช้ความจำของตารางเปิดดูจำนวนมากทำให้มีค่าใช้จ่ายที่เพิ่มจำนวนของลอจิก

ดังนั้นสำหรับการสร้างวงจรกรองเอฟไอาร์แบบ PDA 8 บิตแบบสมบูรณ์ (Fully Parallel 8-bit PDA FIR Filter) โดยอินพุท 8 บิต ถูกแบ่งออกเป็น 1 บิต ย่อยจำนวน 8 อัน ดังนั้นจึงทำให้ได้ความเร็วสูงสุด ดังรูปที่ 3.19 แสดงวงจรกรองเอฟไอาร์แบบ PDA 8 บิตแบบสมบูรณ์ ซึ่งอินพุต 8 บิตทั้งหมดถูกคำนวณแบบขนานและต่อจากนั้นจะรวมเข้าด้วยกันด้วยวงจรบวก อินพุทที่ต่ำกว่าในแต่ละวงจรบวกจะถูกเปลี่ยนขนาดสเกลด้วยแฟคเตอร์ 2 ดังนั้นจากการออกแบบด้วยวิธีนี้จึงไม่ต้องใช้วงจรสะสมค่า เนื่องจากเอาต์พุตเป็นผลบวกของผลลัพธ์ที่คำนวณเสร็จสมบูรณ์แล้ว



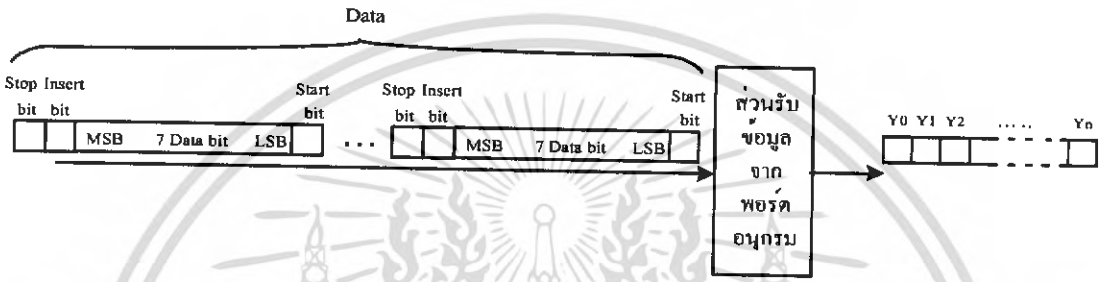
รูปที่ 3.19 วงจรกรองเอฟไอาร์ดาร์บีซี แบบ PDA (ก) บิตเดี่ยว (ข) 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบอุปกรณ์รับส่งข้อมูลผ่านพอร์ทอนุกรม

3.3.1 การรับข้อมูลของอุปกรณ์ FPGA ที่ส่งมาจากคอมพิวเตอร์ผ่านพอร์ทอนุกรม

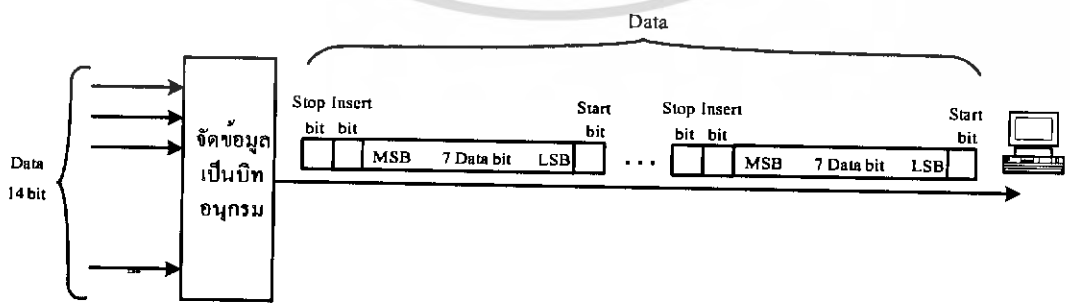
การรับข้อมูลแบบอนุกรมจากคอมพิวเตอร์จะเขียนโปรแกรมสำหรับส่งด้วยโปรแกรม MATLAB เพื่อทำหน้าที่ส่งเฟรมข้อมูลผ่านพอร์ตอนุกรมตามความถี่บอดเรท (Baud Rate) ออกไปทีละ 1 บิต เมื่ออุปกรณ์ FPGA รับเฟรมข้อมูลได้ครบ 1 ชุด จะตัดบิตสตาร์ท (Start bit) บิตสต็อป (Stop bit) และบิตแทรก (Insertion bit) ทิ้งไปให้เหลือเฉพาะบิตข้อมูล แล้วส่งต่อไปยังวงจรตัวแปลงเวฟเลตเพื่อประมวลผลต่อไป โดยบล็อกไดอะแกรมการรับข้อมูลผ่านทางพอร์ทอนุกรมของอุปกรณ์ FPGA จากคอมพิวเตอร์แสดงได้ดังรูปที่ 3.20



รูปที่ 3.20 บล็อกไดอะแกรมการรับข้อมูลของอุปกรณ์ FPGA ผ่านพอร์ทอนุกรม

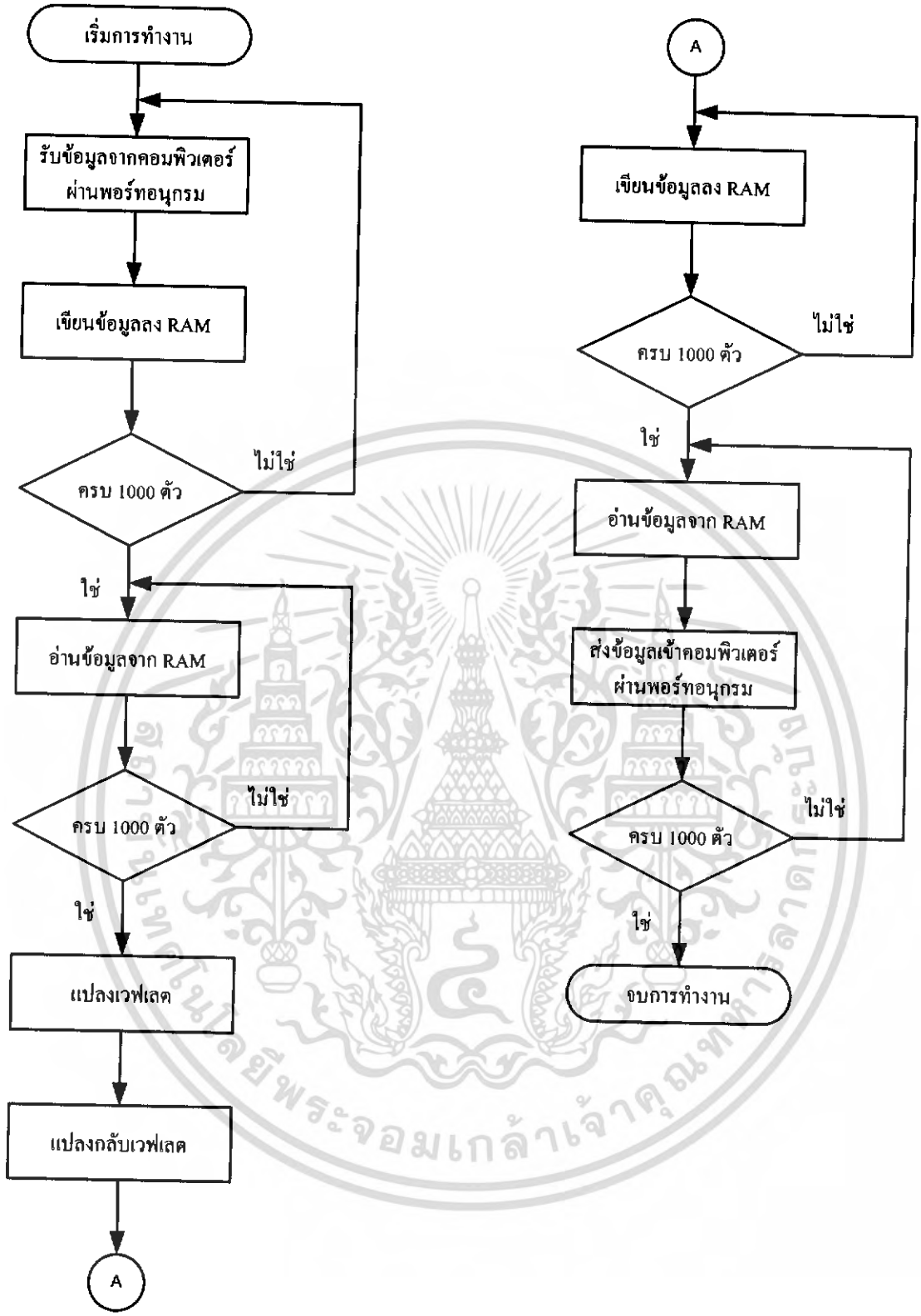
3.3.2 การส่งข้อมูลของอุปกรณ์ FPGA ไปยังคอมพิวเตอร์ผ่านพอร์ทอนุกรม

การส่งข้อมูลออกจากอุปกรณ์ FPGA จะมีอุปกรณ์ทำหน้าที่แปลงบิตข้อมูลแบบขนานขนาด 14 บิตที่ได้จากตัวแปลงเวฟเลต ให้เป็นบิตข้อมูลแบบอนุกรมโดยจะแบ่งออกเป็น 2 เฟรม แต่ละเฟรมข้อมูลประกอบด้วย บิตสตาร์ท (Start bit) 1 บิต บิตสต็อป (Stop bit) 1 บิต บิตข้อมูล (Data bit) 7 บิต และบิตแทรก (Insertion bit) 1 บิต เพื่อแก้ไขกรณีการส่งบิตบิตข้อมูลที่เป็น 0 ทั้งหมด จากนั้นจะส่งเฟรมข้อมูลดังกล่าวออกทางพอร์ตอนุกรมไปยังคอมพิวเตอร์ตามความถี่บอดเรท โดยบล็อกไดอะแกรมการส่งข้อมูลทางพอร์ทอนุกรมจากอุปกรณ์ FPGA ไปยังคอมพิวเตอร์แสดงดังรูปที่ 3.21



รูปที่ 3.21 บล็อกไดอะแกรมการส่งข้อมูลของอุปกรณ์ FPGA ผ่านพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของตัวแปลงเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

การจำลองการทำงานและการทดลองตัวแปลงเวฟเลตโดยใช้โครงสร้างลิฟท์ติ้ง และโครงสร้างเลขคณิตกระจายแบบขนาน สามารถแบ่งออกเป็น 4 ส่วน คือ

4.1 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ

4.2 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรม ModelSim XE III 6.0d

4.3 การจำลองการทำงานวงจรเลขคณิตกระจายโดยใช้โปรแกรม ModelSim XE III 6.0d

4.4 การเปรียบเทียบเพื่อหาค่าผิดพลาดของการแปลงเวฟเลตที่ได้จากการจำลองการทำงาน

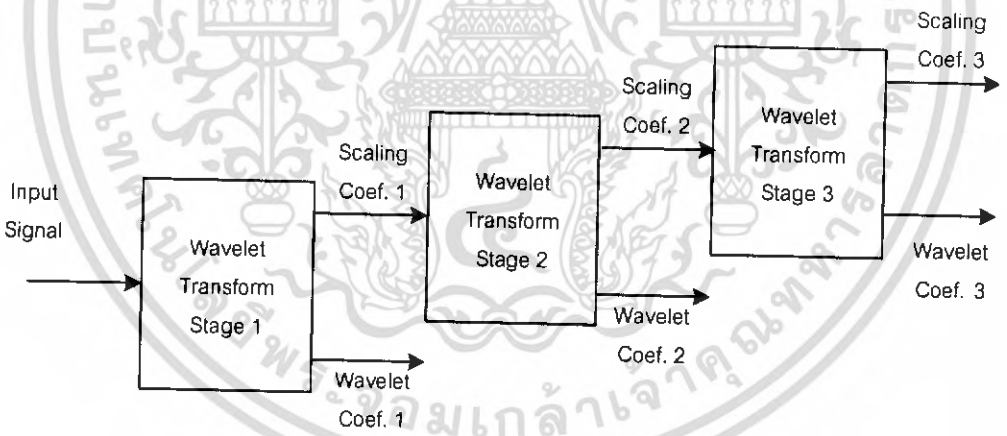
4.5 การจำลองการทำงานวงจรควบคุมการรับส่งข้อมูลใช้โปรแกรม ModelSim XE III 6.0d

4.6 การทดลองตัวแปลงเวฟเลตที่สร้างจริงลงอุปกรณ์ FPGA และแสดงผลผ่านจอคอมพิวเตอร์

4.1 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ

4.1.1 การแปลงเวฟเลต (Wavelet Transform)

การทดลองการแปลงเวฟเลตจะคำนวณหาค่าสัมประสิทธิ์สเกลลิ่งและสัมประสิทธิ์เวฟเลตทั้งหมด 3 ขั้น เพื่อเปรียบเทียบค่าที่ได้ในแต่ละระดับ โดยขั้นตอนการทดลองแสดงดังรูปที่ 4.1

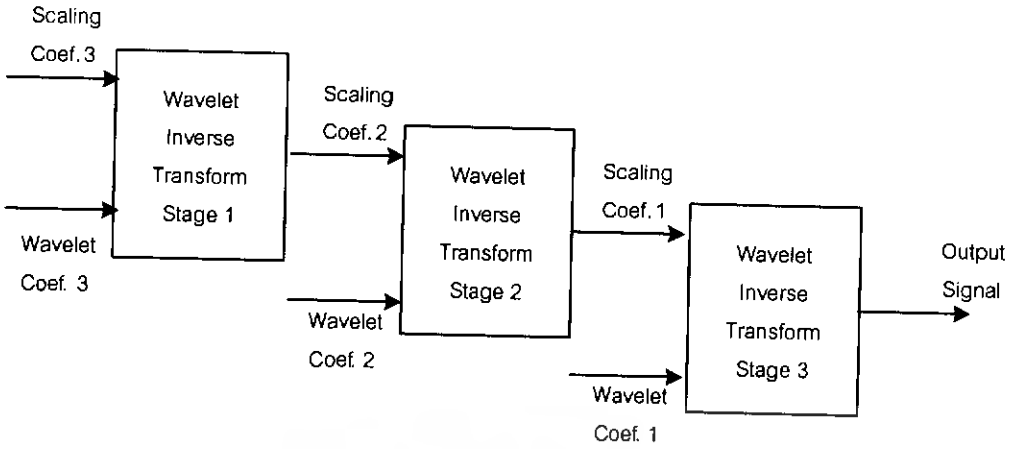


รูปที่ 4.1 บล็อกไดอะแกรมของการแปลงเวฟเลต

4.1.2 การแปลงกลับเวฟเลต (Wavelet Inverse Transform)

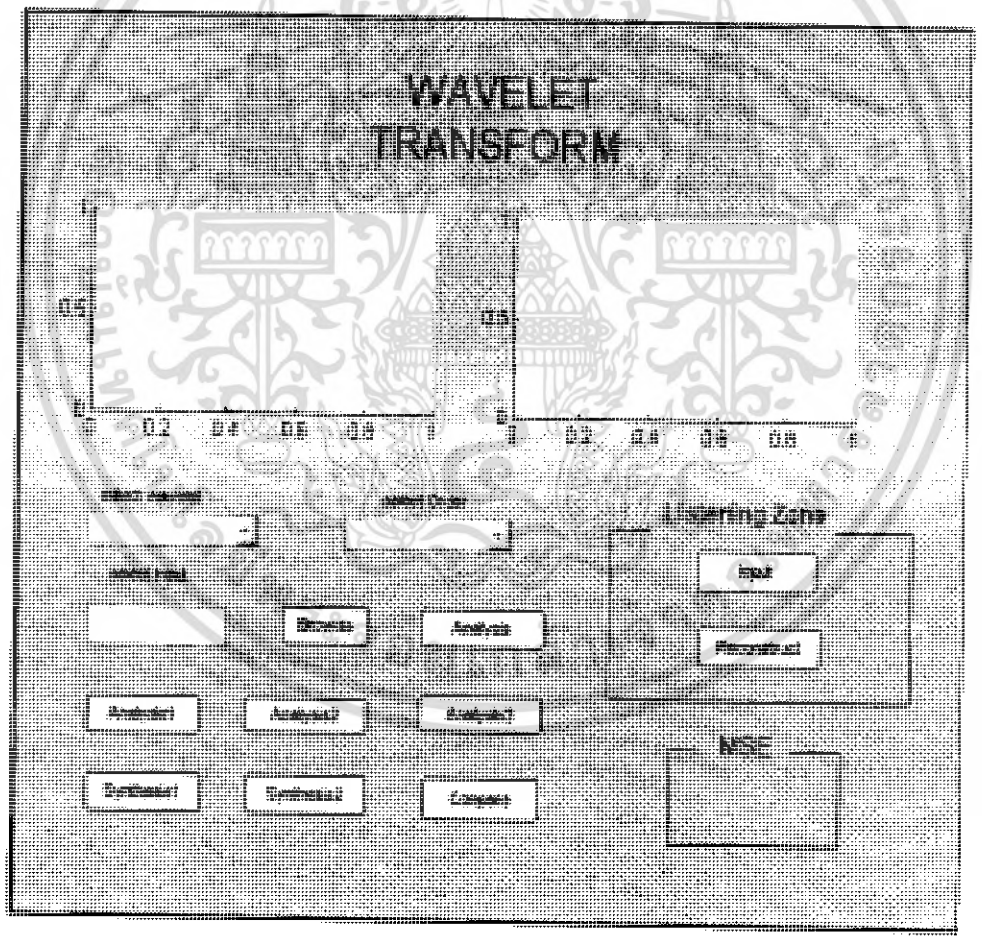
การทดลองการแปลงกลับเวฟเลตจะคำนวณหาสัมประสิทธิ์สเกลลิ่งของขั้นที่ต่ำกว่า โดยใช้สัมประสิทธิ์สเกลลิ่งและสัมประสิทธิ์เวฟเลตของขั้นที่สูงกว่ามาคำนวณ โดยขั้นตอนการทดลองแสดงดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 บล็อกไดอะแกรมของการแปลงกลับเวฟเลต

4.1.3 ขั้นตอนการทดลอง



รูปที่ 4.3 หน้าต่างของแมทแลบส่วนอินเตอร์เฟสของโปรแกรม Wavelet Transform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

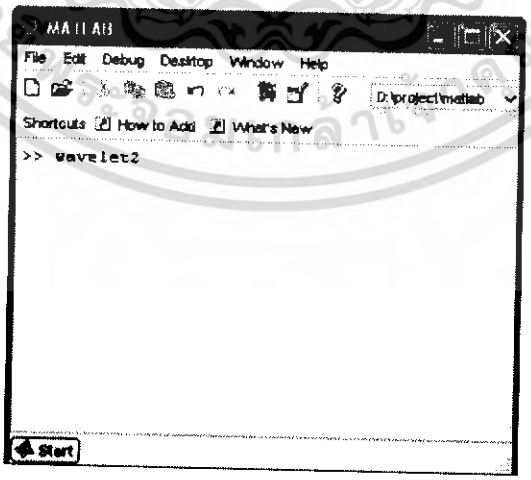
จากรูป 4.3 แสดงหน้าต่างของอินเตอร์เฟสของโปรแกรมคำนวณและแสดงผลการแปลงเวฟเลต โดยตัวโปรแกรมประกอบด้วย หน้าต่างที่ใช้แสดงผลกราฟ 2 กราฟ ซึ่งได้จากการคำนวณที่ได้จากการแปลงเวฟเลตในแต่ละขั้น ส่วนของการเลือกชนิดของเวฟเลตที่จะใช้ในการแปลงและอันดับ ส่วนของการเลือกสัญญาณที่จะใช้ในการแปลงเวฟเลต และส่วนแสดงค่าความผิดพลาดของสัญญาณที่กู้กลับมาได้เมื่อเทียบกับสัญญาณอินพุตหรือค่า MSE

ขั้นตอนการใช้งานโปรแกรมจำลองการทำงาน

1. เริ่มจากเปิดโปรแกรมแมทแลบขึ้นมาดังรูป 4.4

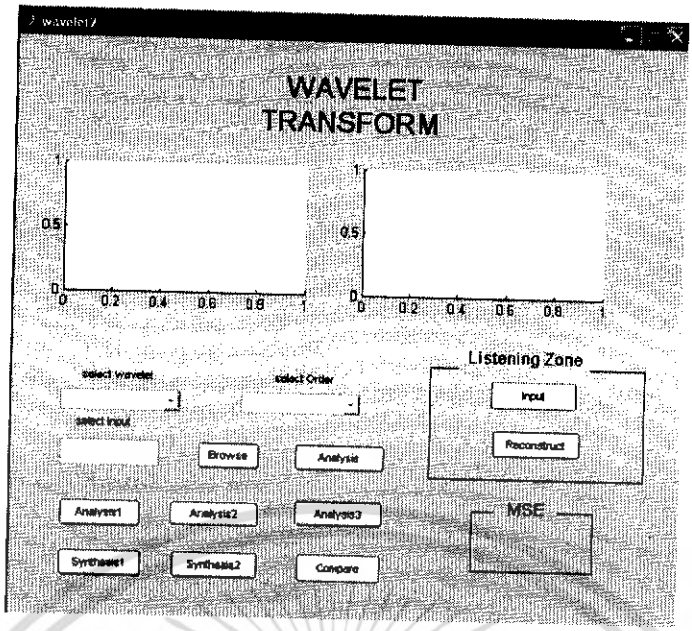
รูปที่ 4.4 หน้าต่าง command ของโปรแกรมแมทแลบ

2. พิมพ์คำสั่ง wavelet2 ลงบนหน้าต่าง command ดังรูป 4.5 จะปรากฏหน้าต่างของส่วนอินเตอร์เฟส ดังรูปที่ 4.6



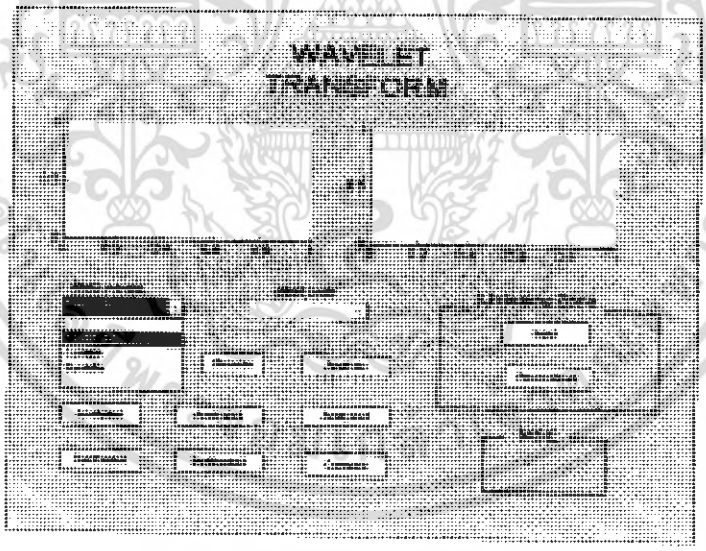
รูปที่ 4.5 คำสั่งที่ใช้เรียกส่วนของ GUI ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 หน้าต่างของ GUI หลังพิมพ์คำสั่ง wavelet 2

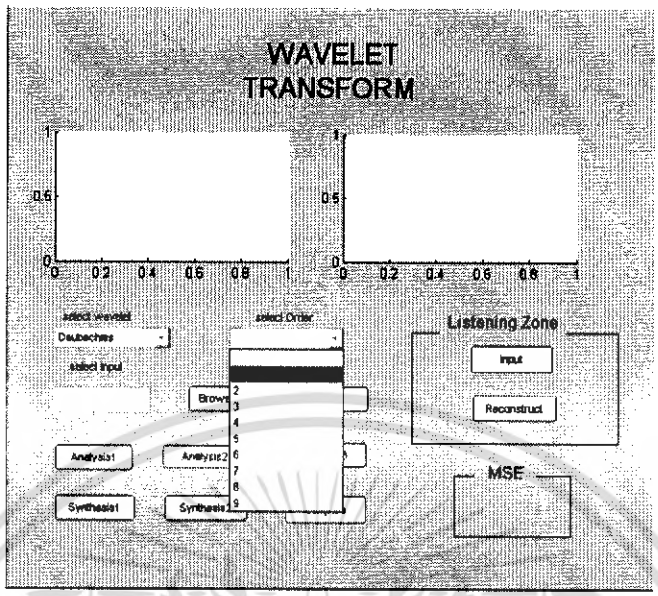
3. เลือกชนิดของ mother wavelet ที่จะใช้ในการแปลงเวฟเลต ดังรูป 4.7 โดยจะมีชนิดของ mother wavelet ให้เลือก 3 ชนิด คือ Daubechies , Coiflets และ Symlets



รูปที่ 4.7 การเลือกชนิดของ mother wavelet ที่จะใช้ในการแปลงเวฟเลต

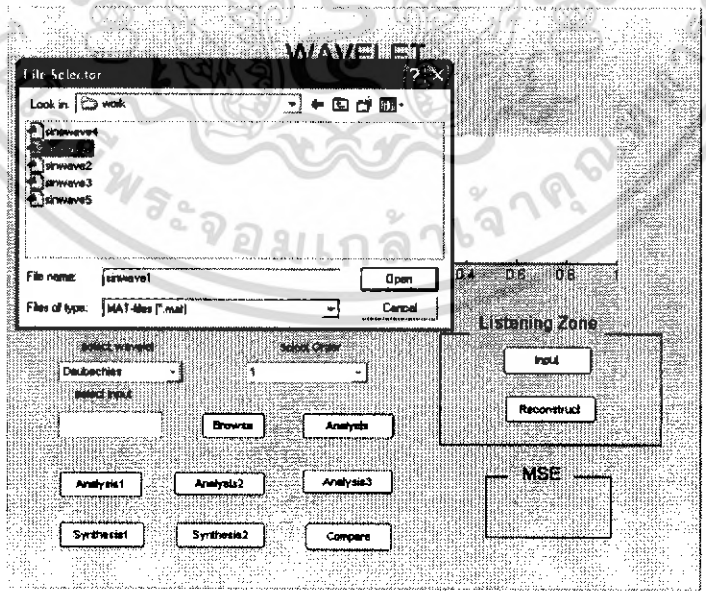
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกอันดับของ mother wavelet โดยมีให้เลือกตั้งแต่ 1- 9 ดังรูปที่ 4.8



รูปที่ 4.8 การเลือกอันดับของ mother wavelet

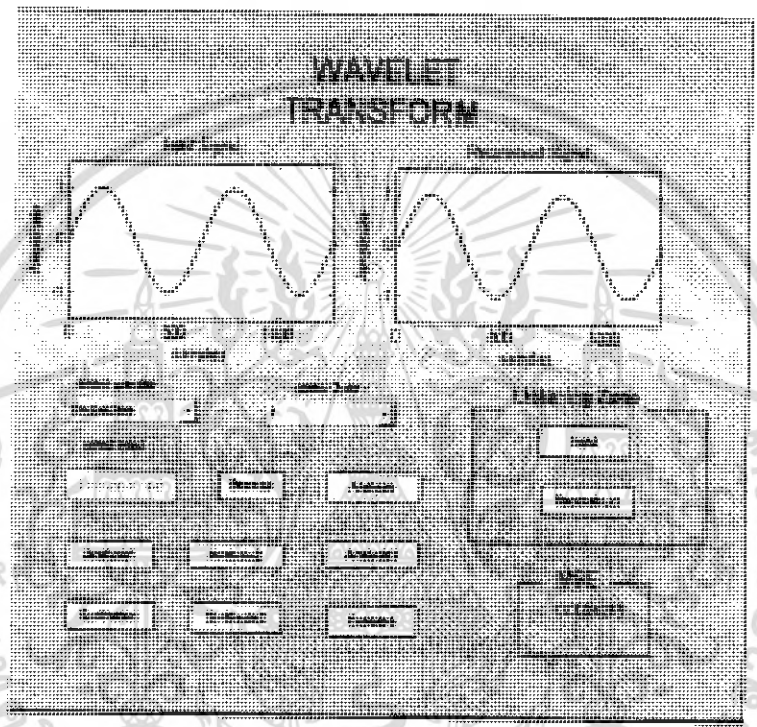
5. เลือกสัญญาณอินพุตที่จะนำมาแปลงเวฟเลต โดยกดปุ่ม Browse จะมีหน้าต่างชื่อ File Selector ขึ้นมา จากนั้นเลือกไฟล์สัญญาณอินพุต ซึ่งสามารถรับสัญญาณได้สองแบบคือ สัญญาณ 1 มิติที่สร้างจากโปรแกรมแมทแลบ (ไฟล์นามสกุล . mat) และสัญญาณเสียง (ไฟล์นามสกุล .wav) ดังรูปที่ 4.9



รูปที่ 4.9 การเลือกสัญญาณอินพุตที่ใช้ในการแปลงเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จากนั้นกดปุ่ม Analysis เพื่อแปลงเวฟเลตทั้งด้านการแปลงและการแปลงกลับ ซึ่งจะใช้สัมประสิทธิ์ของตัวกรองสัญญาณตามชนิดและอันดับของ mother wavelet ที่เลือกไว้ เมื่อประมวลผลเสร็จจะแสดงผลเป็นกราฟเปรียบเทียบกันระหว่างสัญญาณอินพุตและสัญญาณที่สร้างกลับคืนมาได้ดังรูปที่ 4.10
7. กดปุ่ม Analysis1 เพื่อแสดงผลระหว่างสัมประสิทธิ์สเกลลิ่งระดับที่ 1 ที่ออกมาจากตัวกรองความถี่ต่ำผ่านในการแปลงขั้นที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1 ที่ออกมาจากตัวกรองความถี่สูงผ่านในการแปลงขั้นที่ 1 ดังรูปที่ 4.11



รูปที่ 4.10 การเปรียบเทียบระหว่างสัญญาณอินพุตกับสัญญาณที่สร้างกลับคืน

8. กดปุ่ม Analysis2 เพื่อแสดงผลระหว่างสัมประสิทธิ์สเกลลิ่งระดับที่ 2 ที่ออกมาจากตัวกรองความถี่ต่ำผ่านในการแปลงขั้นที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2 ที่ออกมาจากตัวกรองความถี่สูงผ่านในการแปลงขั้นที่ 2 ดังรูปที่ 4.12
9. กดปุ่ม Analysis3 เพื่อแสดงผลระหว่างสัมประสิทธิ์สเกลลิ่งระดับที่ 3 ที่ออกมาจากตัวกรองความถี่ต่ำผ่านในการแปลงขั้นที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3 ที่ออกมาจากตัวกรองความถี่สูงผ่านในการแปลงขั้นที่ 3 ดังรูปที่ 4.13
10. กดปุ่ม Synthesis1 เพื่อแสดงผลระหว่างสัญญาณอินพุตกับสัญญาณที่สร้างกลับคืนมาได้เมื่อผ่านการแปลงกลับขั้นที่ 1 ดังรูปที่ 4.14
11. กดปุ่ม Synthesis2 เพื่อแสดงผลระหว่างสัญญาณอินพุตกับสัญญาณที่สร้างกลับคืนมาได้เมื่อผ่านการแปลงกลับขั้นที่ 2 ดังรูปที่ 4.15

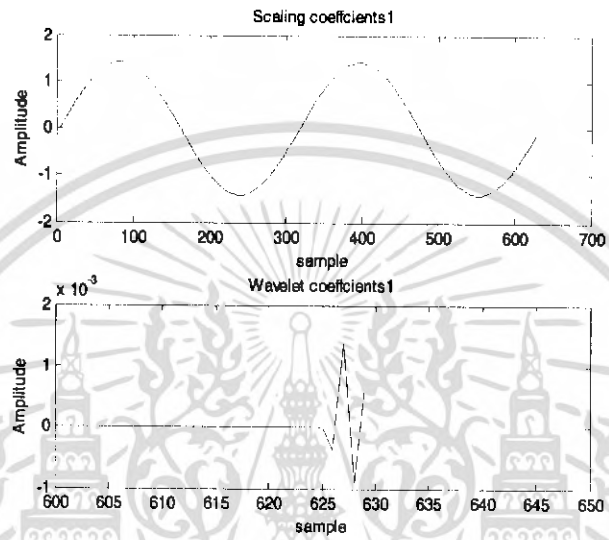
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. กดปุ่ม Synthesis3 เพื่อแสดงผลระหว่างสัญญาณอินพุตกับสัญญาณที่สร้างกลับคืนมาได้เมื่อผ่านการแปลงกลับขั้นที่ 3 ดังรูปที่ 4.16

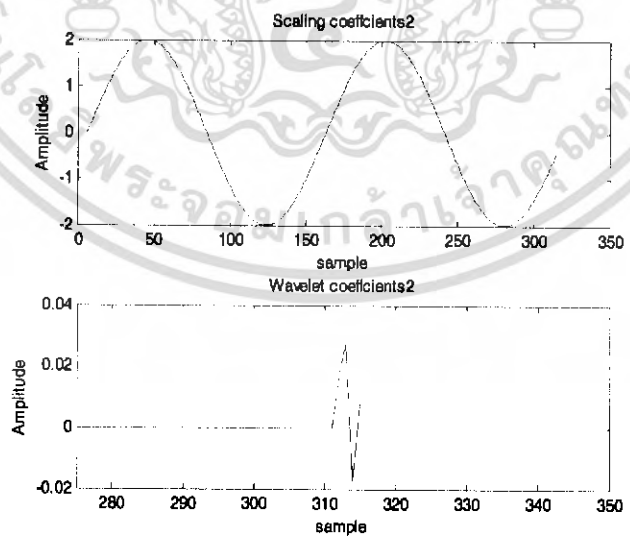
ผลการทดลอง

ผลการจำลองการทำงานโดยใช้โปรแกรมแมทแลบจะแสดงเป็นกราฟ โดยสัญญาณอินพุทที่ใช้ทดลอง มี 3 แบบ คือ สัญญาณไซน์ สัญญาณสี่เหลี่ยม และสัญญาณเสียง

4.1.3.1 ผลการแปลงเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณไซน์

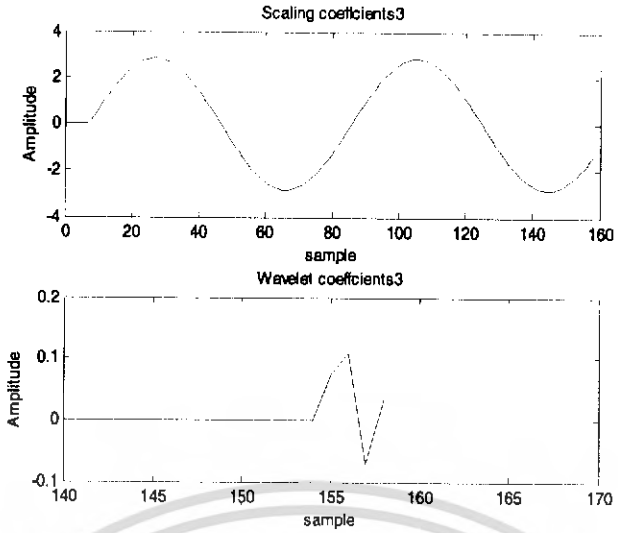


รูปที่ 4.11 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1



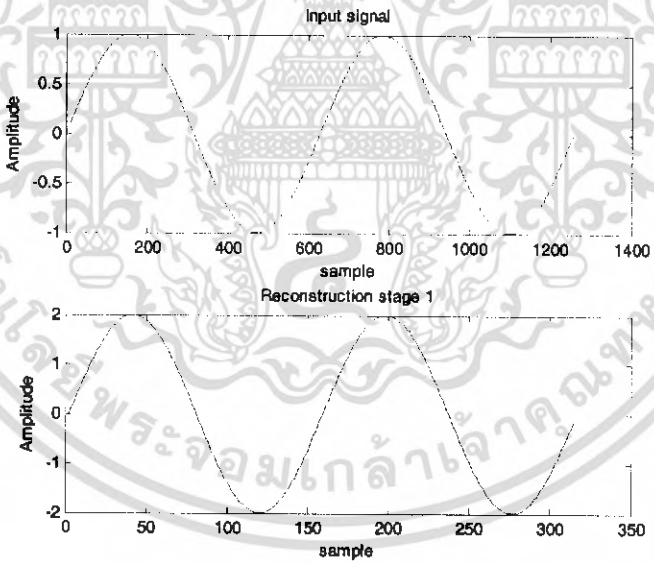
รูปที่ 4.12 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



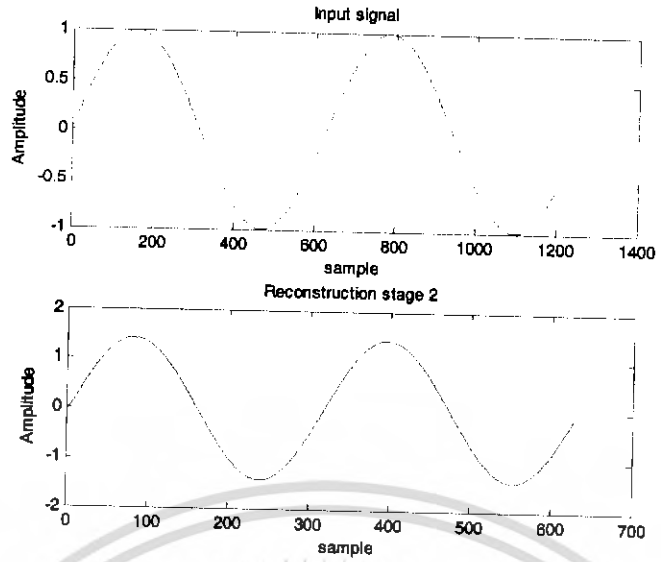
รูปที่ 4.13 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3

4.1.3.2 ผลการแปลงกลับเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณไซน์

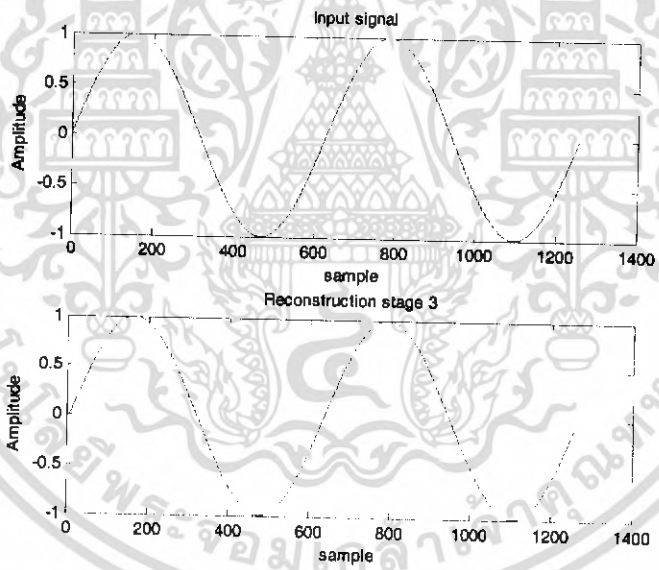


รูปที่ 4.14 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1

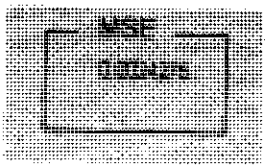
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต
และสัญญาณที่สร้างกลับคืนระดับที่ 2



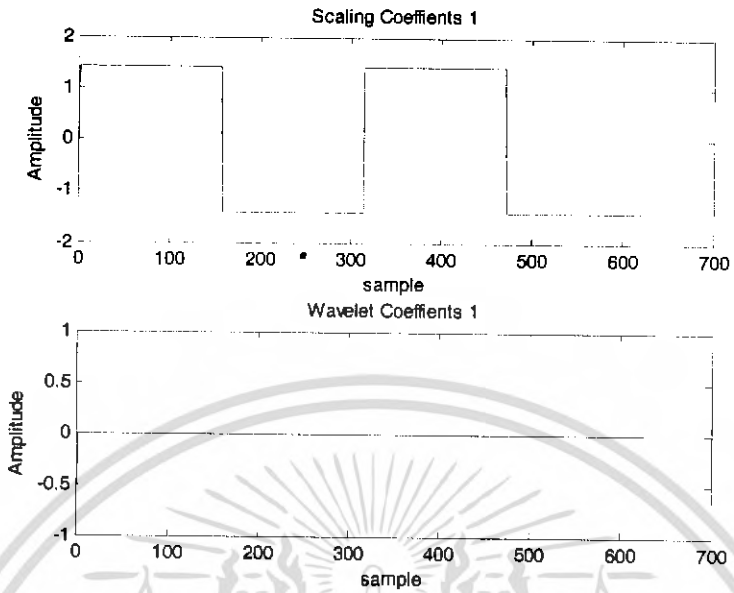
รูปที่ 4.16 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต
และสัญญาณที่สร้างกลับคืนระดับที่ 3



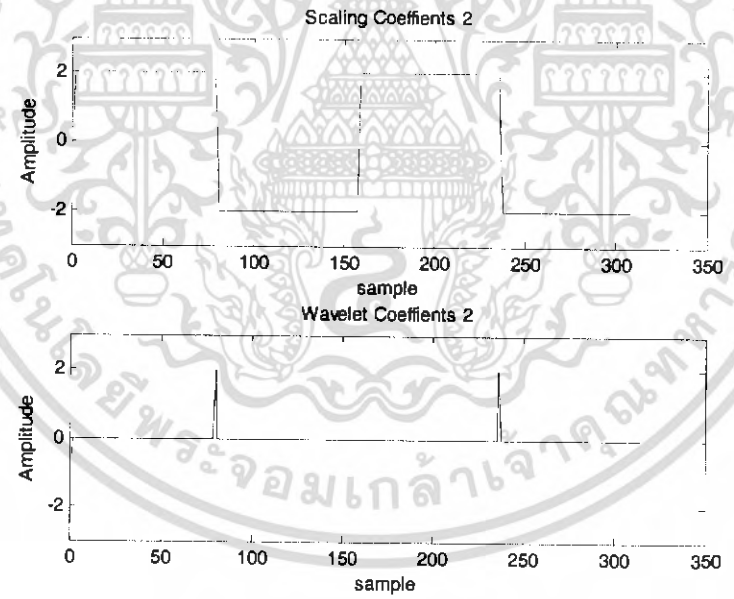
รูปที่ 4.17 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3.3 ผลการแปลงเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม

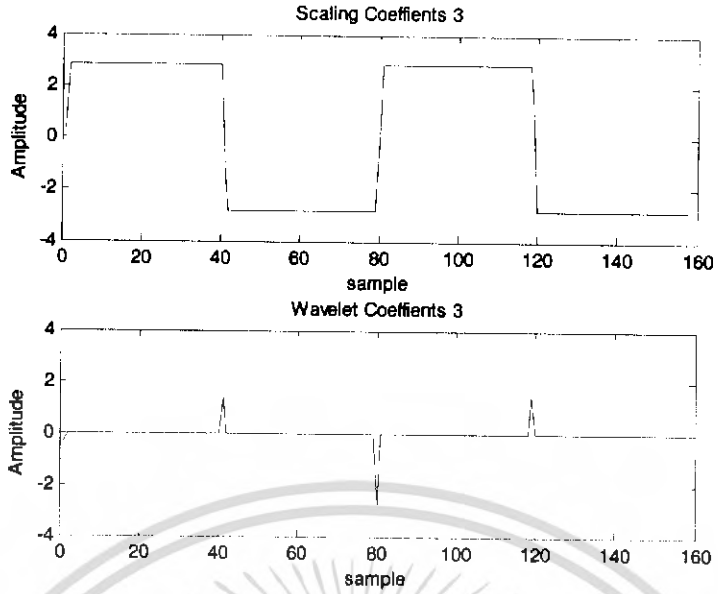


รูปที่ 4.18 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1



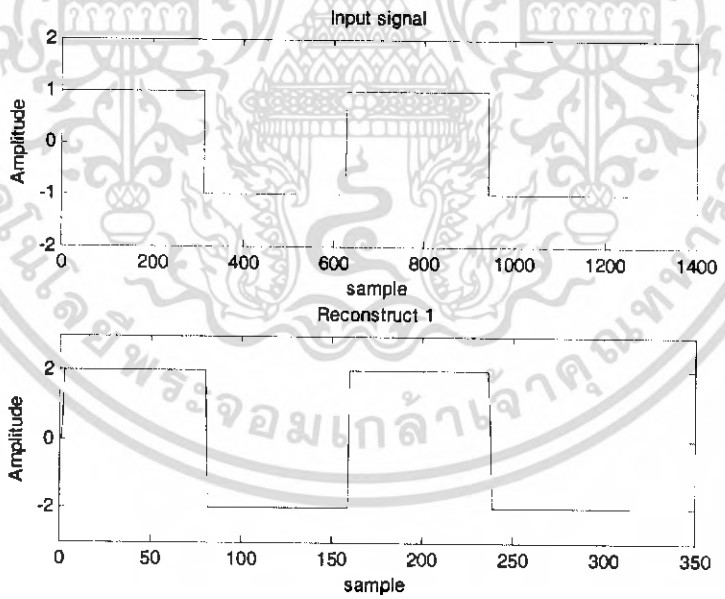
รูปที่ 4.19 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



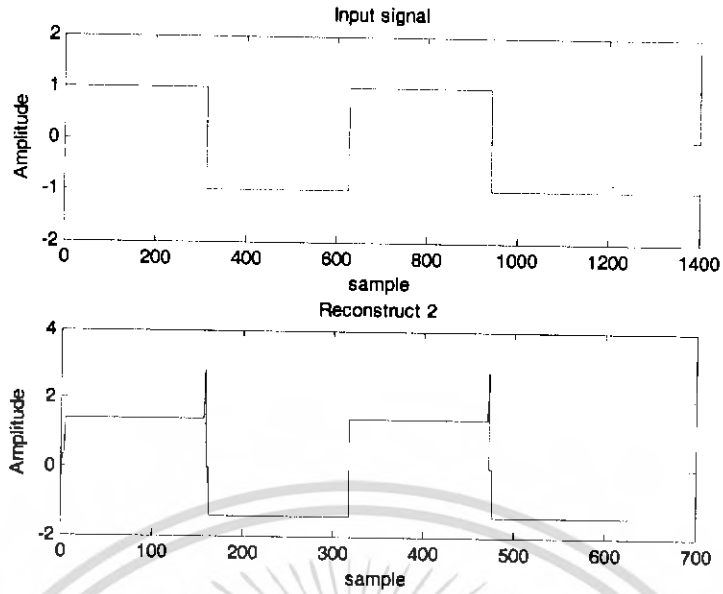
รูปที่ 4.20 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3

4.1.3.4 ผลการแปลงกลับเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม

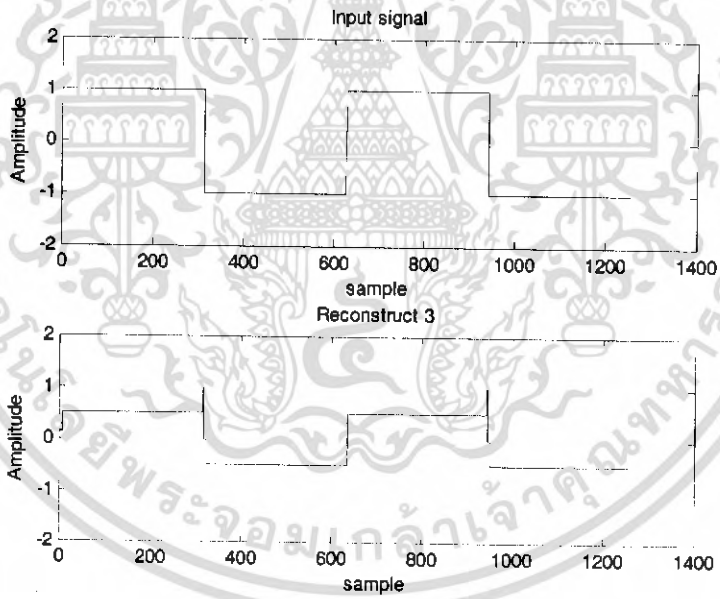


รูปที่ 4.21 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1

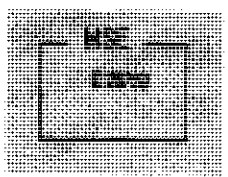
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 2



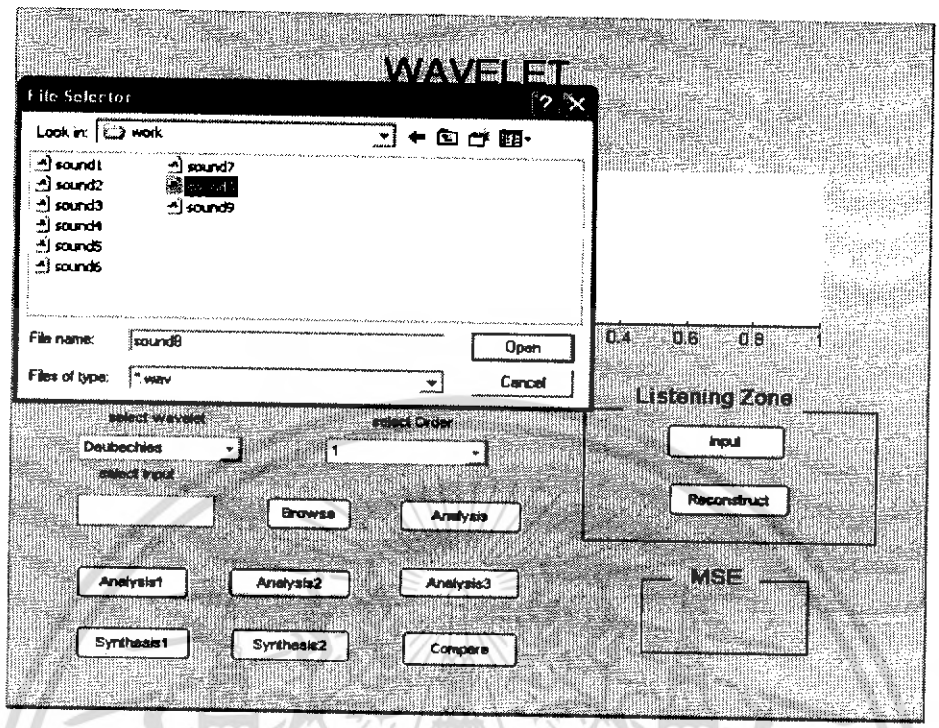
รูปที่ 4.23 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 3



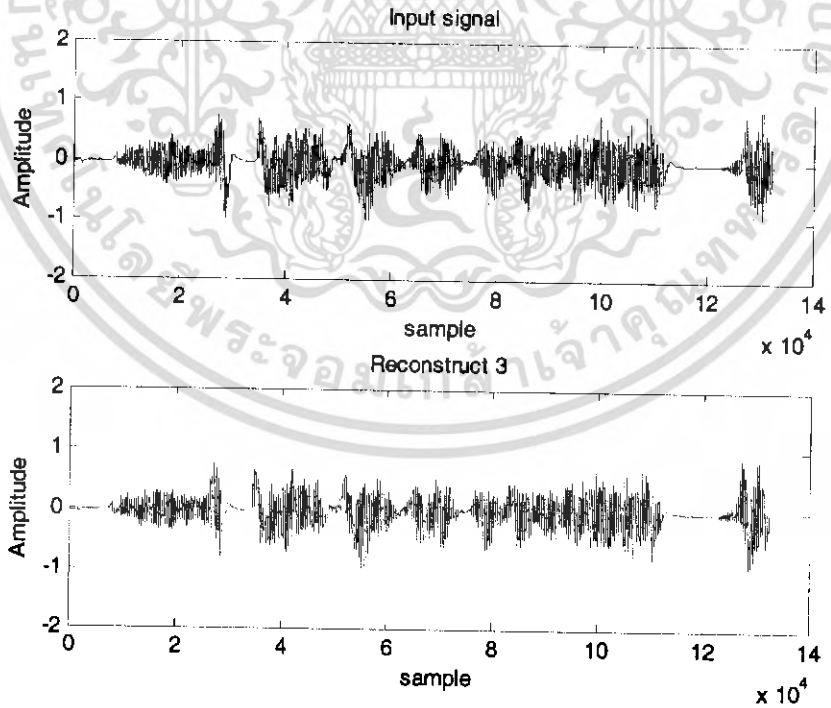
รูปที่ 4.24 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3.5 ผลการแปลงและการแปลงกลับเวฟเลตเมื่อสัญญาณอินพุตเป็นสัญญาณเสียง

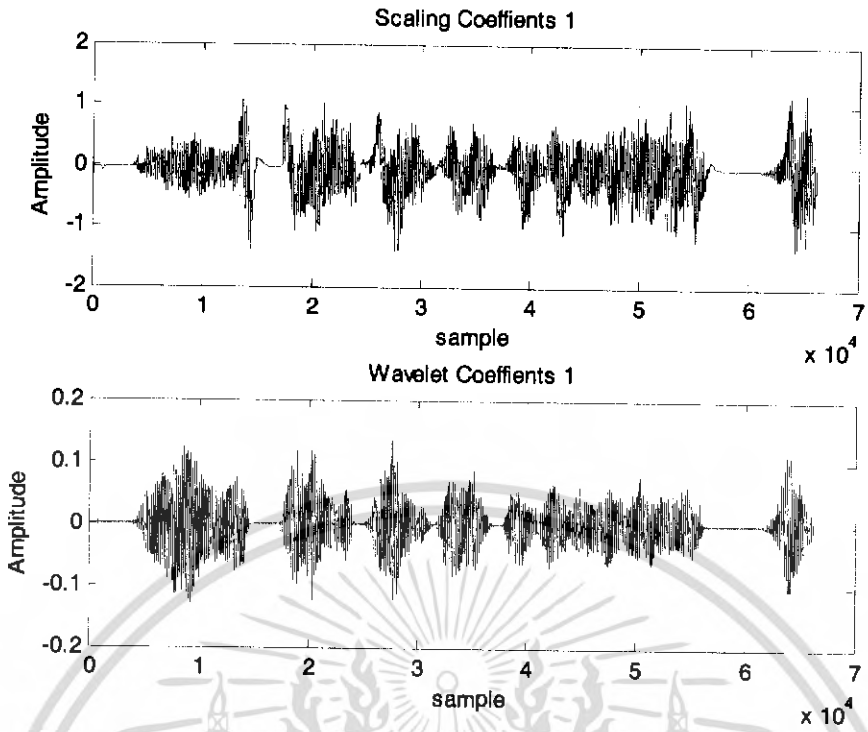


รูปที่ 4.25 เมื่อสัญญาณอินพุตเป็นสัญญาณเสียง

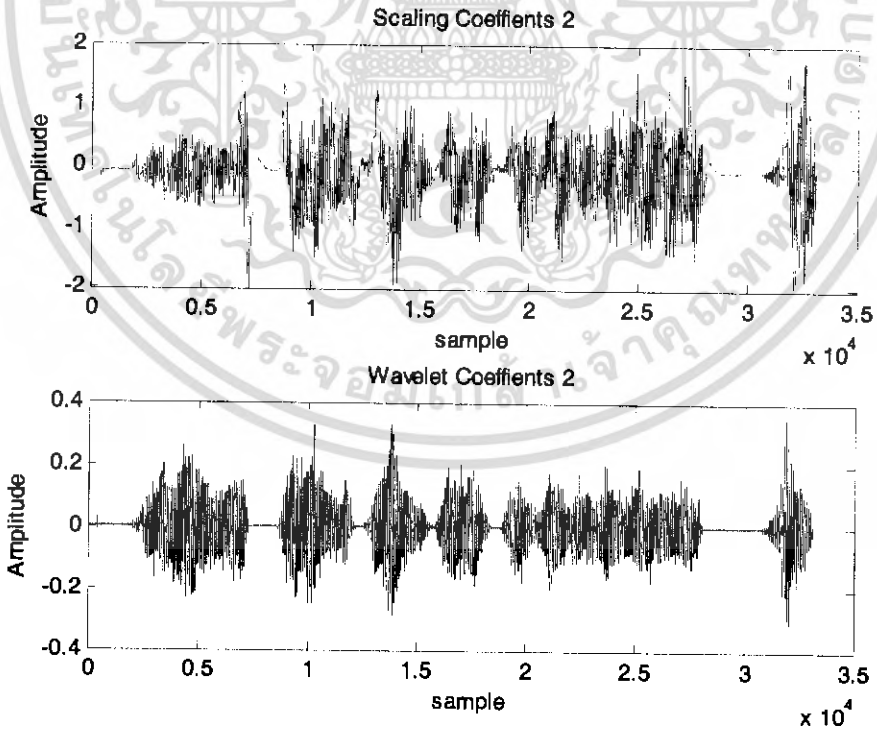


รูปที่ 4.26 การเปรียบเทียบระหว่างสัญญาณอินพุต
กับสัญญาณที่สร้างกลับคืนเมื่อกดปุ่ม Analysis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

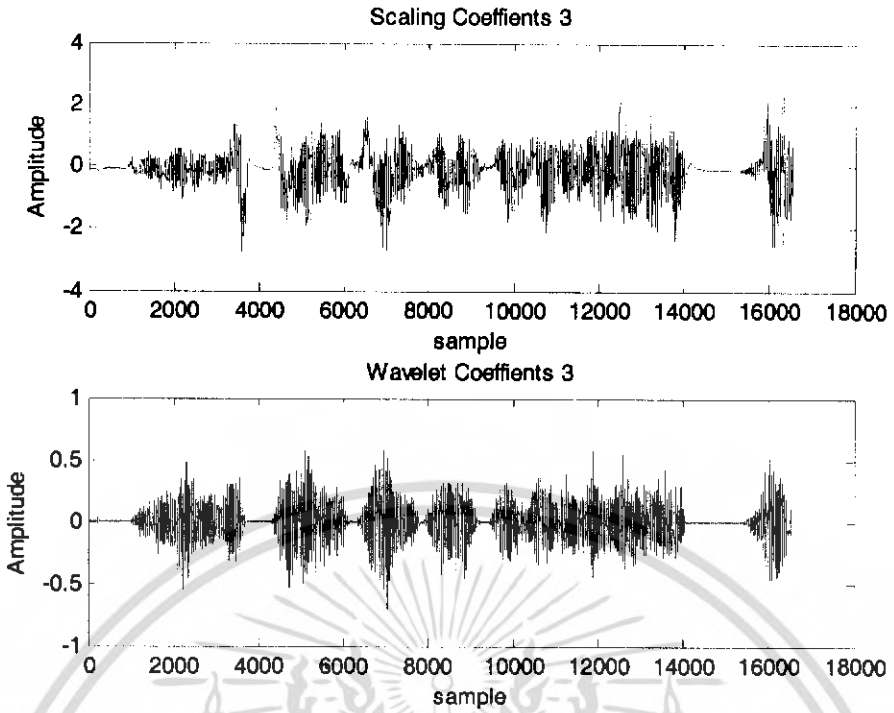


รูปที่ 4.27 ผลการแปลงเวฟเลตขั้นที่ 1 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 1 และสัมประสิทธิ์เวฟเลตระดับที่ 1

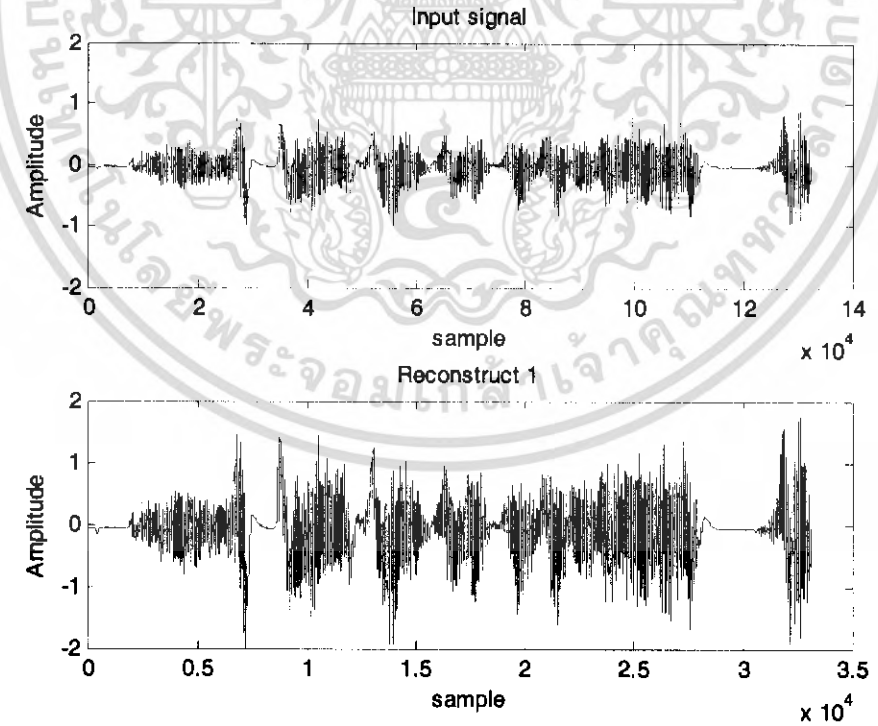


รูปที่ 4.28 ผลการแปลงเวฟเลตขั้นที่ 2 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 2 และสัมประสิทธิ์เวฟเลตระดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

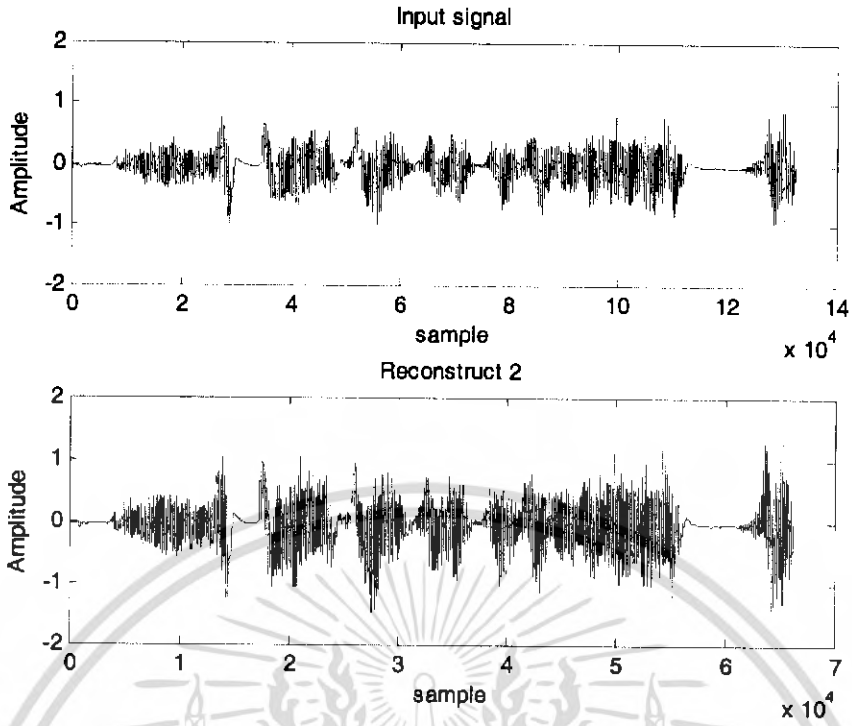


รูปที่ 4.29 ผลการแปลงเวฟเลตขั้นที่ 3 ประกอบด้วยสัมประสิทธิ์สเกลลิ่งระดับที่ 3 และสัมประสิทธิ์เวฟเลตระดับที่ 3

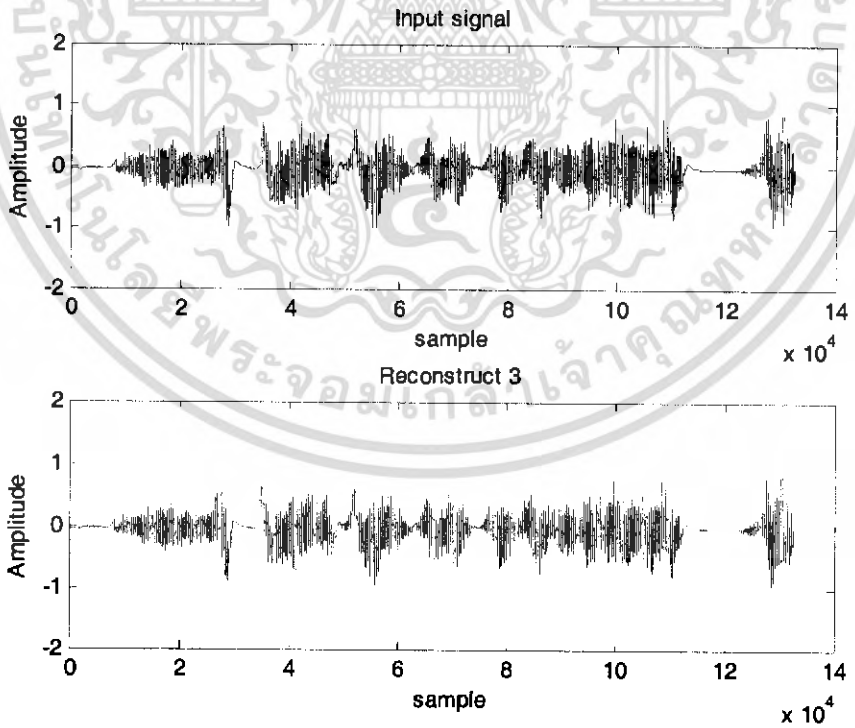


รูปที่ 4.30 ผลการแปลงกลับเวฟเลตขั้นที่ 1 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



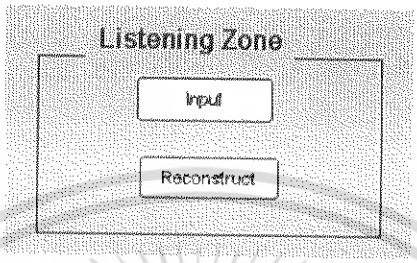
รูปที่ 4.31 ผลการแปลงกลับเวฟเลตขั้นที่ 2 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 2



รูปที่ 4.32 ผลการแปลงกลับเวฟเลตขั้นที่ 3 ประกอบด้วยสัญญาณอินพุต และสัญญาณที่สร้างกลับคืนระดับที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.33 ค่า MSE ที่คำนวณได้เมื่อกดปุ่ม Compare



รูปที่ 4.34 ปุ่มที่ใช้ในการฟังเสียงของสัญญาณเสียงอินพุต
และสัญญาณเสียงที่สร้างกลับคืน

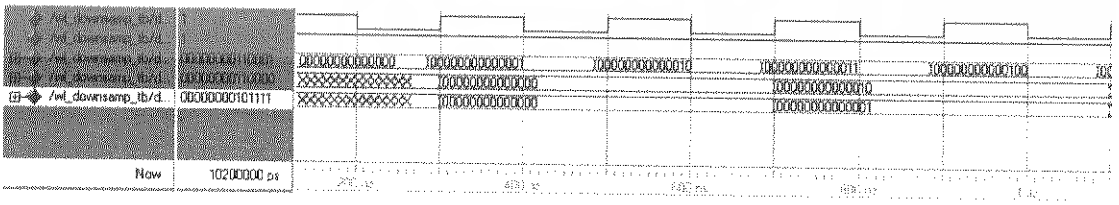
4.2 การจำลองการทำงานตัวแปลงเวฟเลตโดยใช้โปรแกรม ModelSim XE III 6.0d

การทดลองนี้จะจำลองการทำงานของอุปกรณ์ที่นำมาประกอบเป็นตัวแปลงเวฟเลต ทั้งแบบที่ใช้ DA อนุกรมและขนาน โดยที่อุปกรณ์แต่ละส่วนมีผลการจำลองการทำงานดังต่อไปนี้

4.2.1 วงจรสุ่มค่าตัวอย่าง (Down-sampler)

รูปที่ 4.35 วงจรสุ่มค่าตัวอย่าง

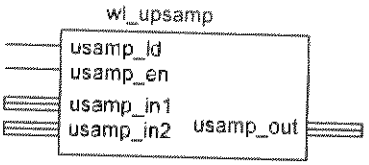
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.36



รูปที่ 4.36 ผลการจำลองการทำงานของวงจรสุ่มค่าตัวอย่าง

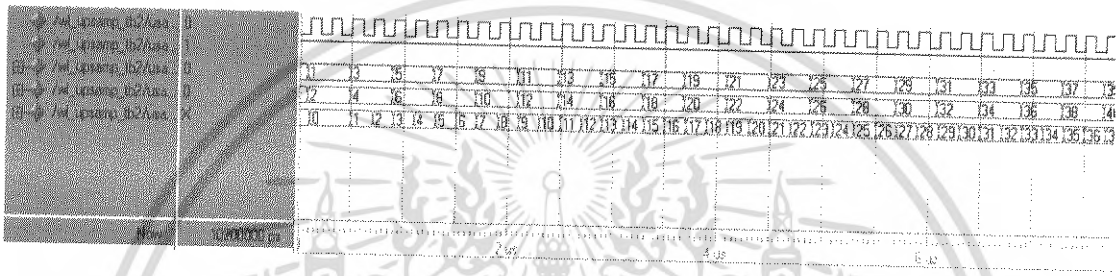
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 วงจรเพิ่มค่าตัวอย่าง (Up-sampler)



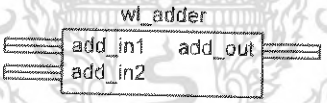
รูปที่ 4.37 วงจรเพิ่มค่าตัวอย่าง

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.38



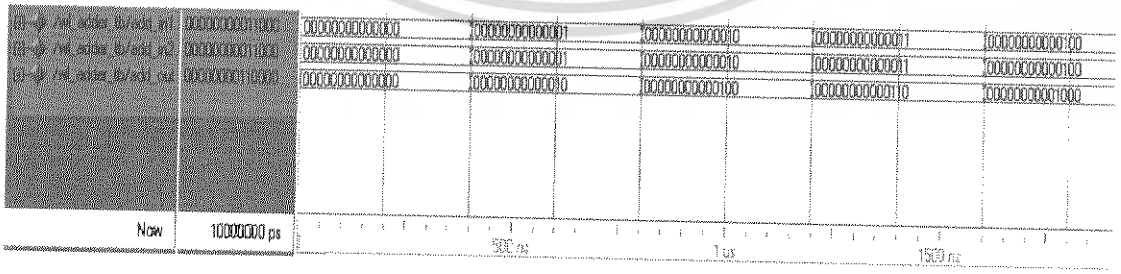
รูปที่ 4.38 ผลการจำลองการทำงานของวงจรเพิ่มค่าตัวอย่าง

4.2.3 วงจรบวก (Adder)



รูปที่ 4.39 วงจรบวก

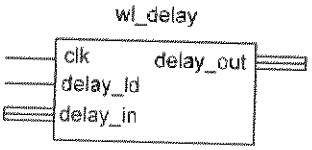
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.40



รูปที่ 4.40 ผลการจำลองการทำงานของวงจรบวก

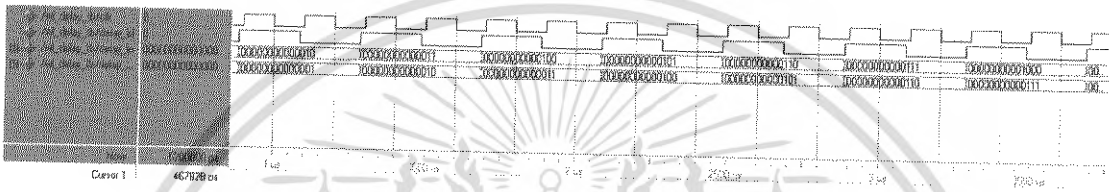
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 วงจรหน่วงเวลา 1 เท่า



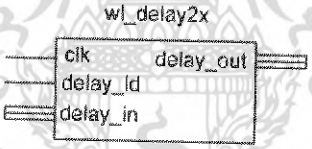
รูปที่ 4.41 วงจรหน่วงเวลา 1 เท่า

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.42



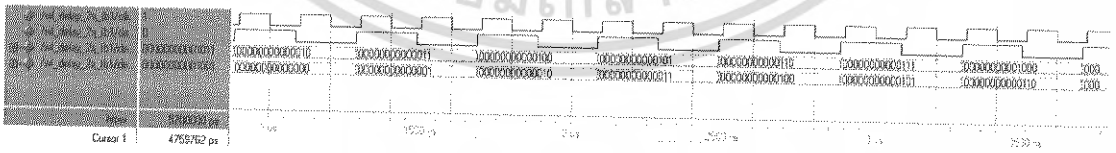
รูปที่ 4.42 ผลการจำลองการทำงานของวงจรหน่วงเวลา 1 เท่า

4.2.5 วงจรหน่วงเวลา 2 เท่า



รูปที่ 4.43 วงจรหน่วงเวลา 2 เท่า

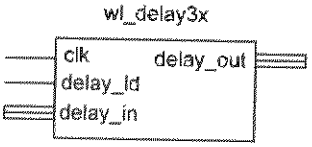
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.44



รูปที่ 4.44 ผลการจำลองการทำงานของวงจรหน่วงเวลา 2 เท่า

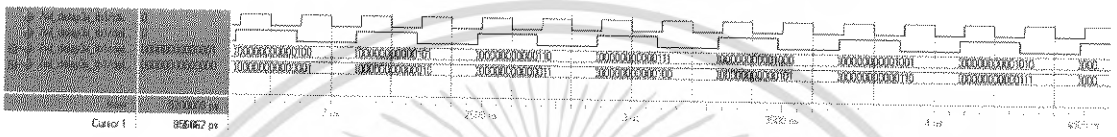
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 วงจรหน่วงเวลา 3 เท่า



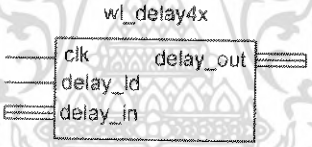
รูปที่ 4.45 วงจรหน่วงเวลา 3 เท่า

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.46



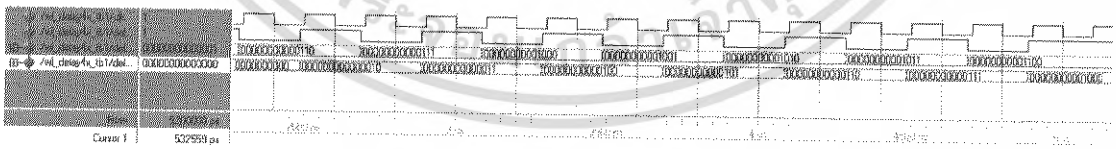
รูปที่ 4.46 ผลการจำลองการทำงานของวงจรหน่วงเวลา 3 เท่า

4.2.7 วงจรหน่วงเวลา 4 เท่า



รูปที่ 4.47 วงจรหน่วงเวลา 4 เท่า

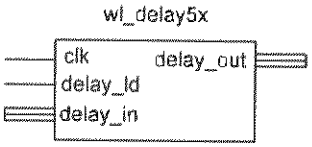
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.48



รูปที่ 4.48 ผลการจำลองการทำงานของวงจรหน่วงเวลา 4 เท่า

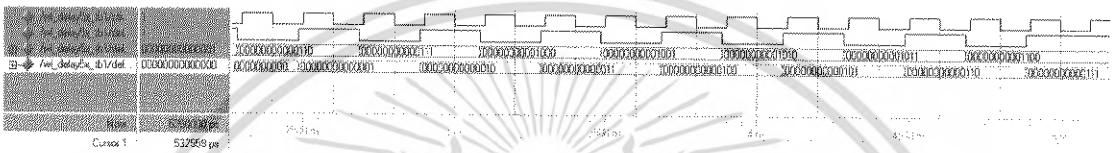
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 วงจรหน่วงเวลา 5 เท่า



รูปที่ 4.49 วงจรหน่วงเวลา 5 เท่า

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.50

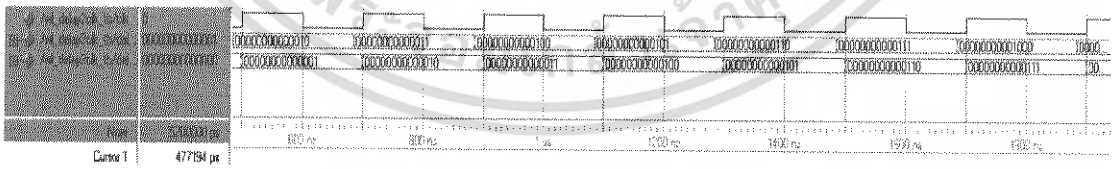


รูปที่ 4.50 ผลการจำลองการทำงานของวงจรหน่วงเวลา 5 เท่า

4.2.9 วงจรหน่วงเวลา 2 สัญญาณนาฬิกา

รูปที่ 4.51 วงจรหน่วงเวลา 2 สัญญาณนาฬิกา

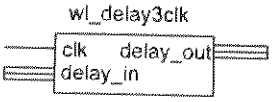
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.52



รูปที่ 4.52 ผลการจำลองการทำงานของวงจรหน่วงเวลา 2 สัญญาณนาฬิกา

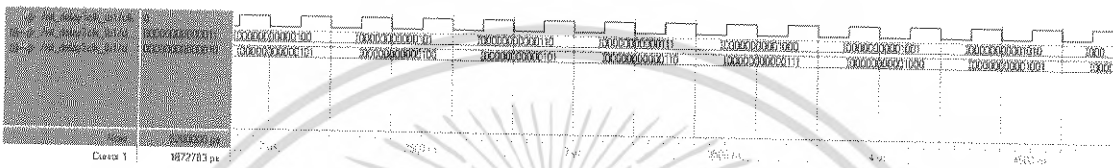
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.10 วงจรหน่วงเวลา 3 สัญญาณนาฬิกา



รูปที่ 4.53 วงจรหน่วงเวลา 3 สัญญาณนาฬิกา

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.54

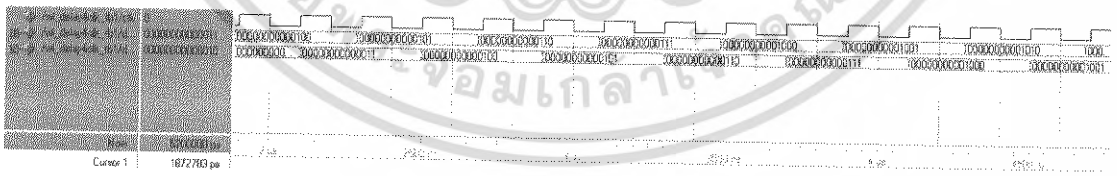


รูปที่ 4.54 ผลการจำลองการทำงานของวงจรหน่วงเวลา 3 สัญญาณนาฬิกา

4.2.11 วงจรหน่วงเวลา 4 สัญญาณนาฬิกา

รูปที่ 4.55 วงจรหน่วงเวลา 4 สัญญาณนาฬิกา

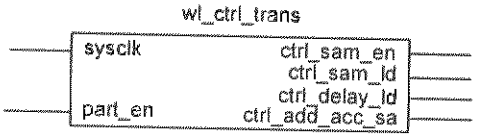
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.56



รูปที่ 4.56 ผลการจำลองการทำงานของวงจรหน่วงเวลา 4 สัญญาณนาฬิกา

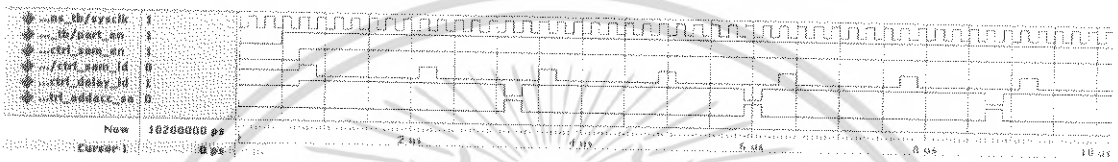
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.12 วงจรควบคุมการแปลงเวฟเลต



รูปที่ 4.57 วงจรควบคุมการแปลงเวฟเลต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.58

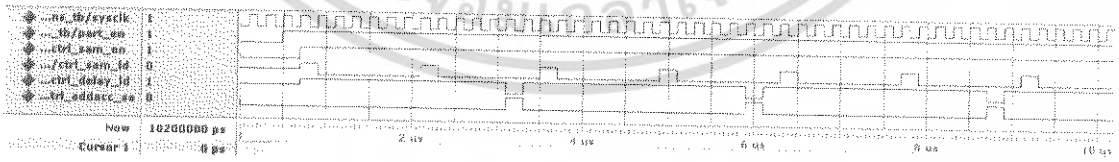


รูปที่ 4.58 ผลการจำลองการทำงานของวงจรควบคุมการแปลงเวฟเลต

4.2.13 วงจรควบคุมการแปลงกลับเวฟเลต

รูปที่ 4.59 วงจรควบคุมการแปลงกลับเวฟเลต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.60



รูปที่ 4.60 ผลการจำลองการทำงานของวงจรควบคุมการแปลงกลับเวฟเลต

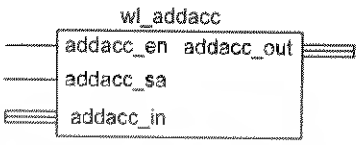
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การจำลองการทำงานวงจรเลขคณิตกระจายด้วยโปรแกรม ModelSim XE III 6.0d

4.3.1 การจำลองการทำงานของวงจรเลขคณิตกระจาย

การทดลองนี้จะจำลองการทำงานของอุปกรณ์ย่อยที่นำมาประกอบเป็นวงจรเลขคณิตกระจาย (DA) แบบอนุกรมและขนาน

4.3.1.1 วงจรบวกสะสมค่า

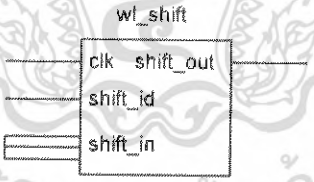


รูปที่ 4.61 วงจรบวกสะสมค่า

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.62

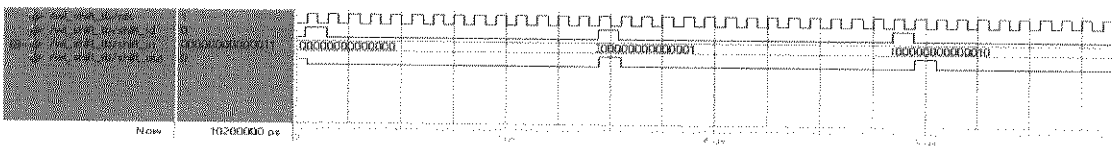
รูปที่ 4.62 ผลการจำลองการทำงานของวงจรวงจรบวกสะสมค่า

4.3.1.2 วงจรเลื่อนข้อมูล



รูปที่ 4.63 วงจรเลื่อนข้อมูล

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.64



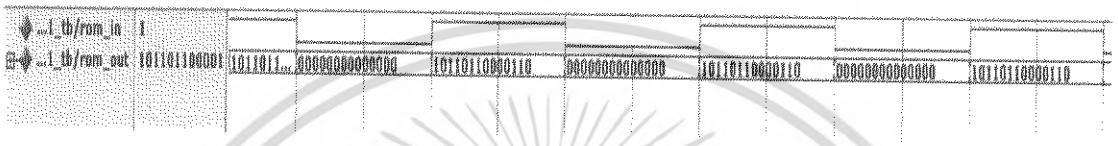
รูปที่ 4.64 ผลการจำลองการทำงานของวงจรเลื่อนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.3 หน่วยความจำการแปลงเวฟเลต 1

รูปที่ 4.65 หน่วยความจำการแปลงเวฟเลต 1

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.66



รูปที่ 4.66 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 1

4.3.1.4 หน่วยความจำการแปลงเวฟเลต 2

รูปที่ 4.67 หน่วยความจำการแปลงเวฟเลต 2

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.68

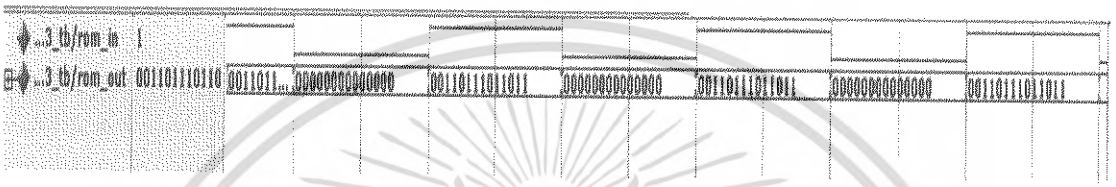
รูปที่ 4.68 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.5 หน่วยความจำการแปลงเวฟเลต 3

รูปที่ 4.69 หน่วยความจำการแปลงเวฟเลต 3

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.70

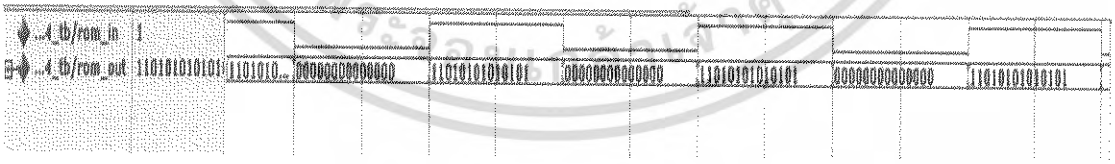


รูปที่ 4.70 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 3

4.3.1.6 หน่วยความจำการแปลงเวฟเลต 4

รูปที่ 4.71 หน่วยความจำการแปลงเวฟเลต 4

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.72



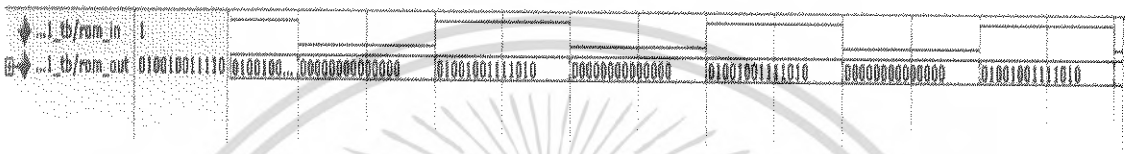
รูปที่ 4.72 ผลการจำลองการทำงานของหน่วยความจำการแปลงเวฟเลต 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.7 หน่วยความจำการแปลงกลับเวฟเลต 1

รูปที่ 4.73 หน่วยความจำการแปลงกลับเวฟเลต 1

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.74

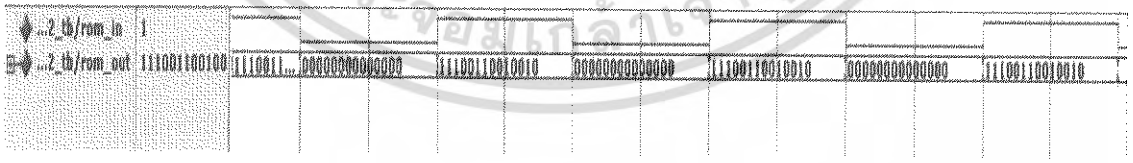


รูปที่ 4.74 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 2

4.3.1.8 หน่วยความจำการแปลงกลับเวฟเลต 2

รูปที่ 4.75 หน่วยความจำการแปลงกลับเวฟเลต 2

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.76



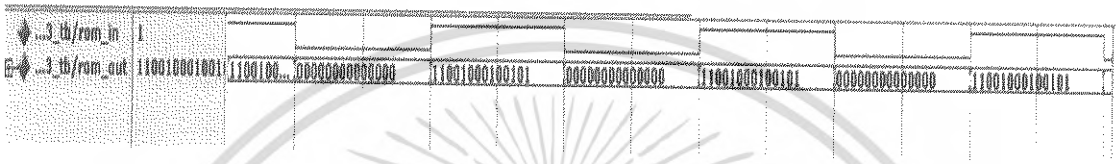
รูปที่ 4.76 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.9 หน่วยความจำการแปลงกลับเวฟเลต 3

รูปที่ 4.77 หน่วยความจำการแปลงกลับเวฟเลต 3

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.78

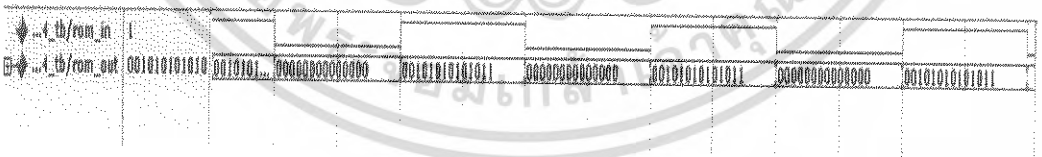


รูปที่ 4.78 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 3

4.3.1.10 หน่วยความจำการแปลงกลับเวฟเลต 4

รูปที่ 4.79 หน่วยความจำการแปลงกลับเวฟเลต 4

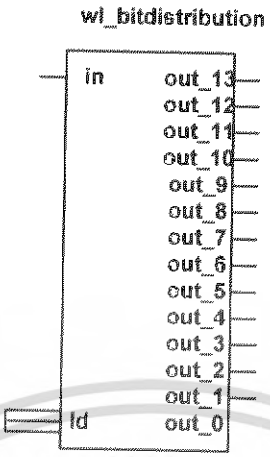
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.80



รูปที่ 4.80 ผลการจำลองการทำงานของหน่วยความจำการแปลงกลับเวฟเลต 4

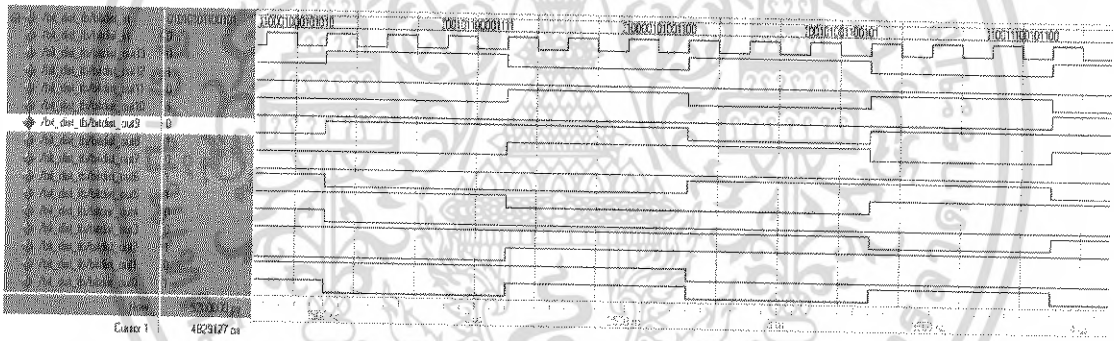
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.11 วงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต



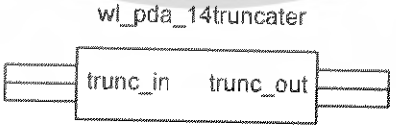
รูปที่ 4.81 วงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.82



รูปที่ 4.82 ผลการจำลองการทำงานของวงจรเลื่อนข้อมูลเข้าขนานออกขนานแยกบิต

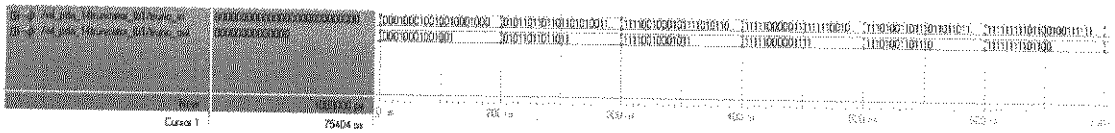
4.3.1.12 วงจรตัดบิตจาก 26 บิตเป็น 14 บิต



รูปที่ 4.83 วงจรตัดบิตจาก 26 บิตเป็น 14 บิต

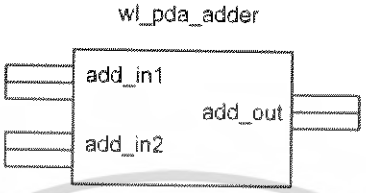
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.84 ผลการจำลองการทำงานของวงจรตัดบิตจาก 26 บิตเป็น 14 บิต

4.3.1.13 วงจรบวก 26 บิต

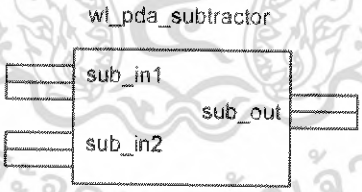


รูปที่ 4.85 วงจรบวก 26 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.86

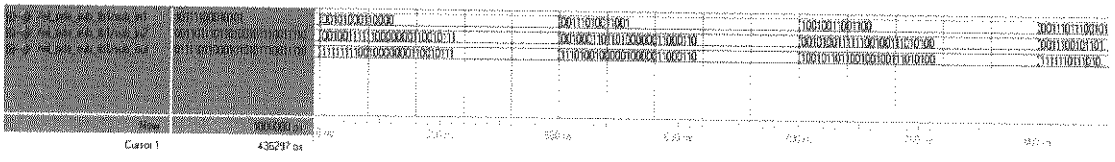
รูปที่ 4.86 ผลการจำลองการทำงานของวงจรบวก 26 บิต

4.3.1.14 วงจรลบ 26 บิต



รูปที่ 4.87 วงจรลบ 26 บิต

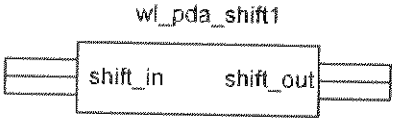
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.88



รูปที่ 4.88 ผลการจำลองการทำงานของวงจรลบ 26 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.15 วงจรเลื่อนบิต 1 บิต



รูปที่ 4.89 วงจรเลื่อนบิต 1 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.90

รูปที่ 4.90 ผลการจำลองการทำงานของวงจรเลื่อนบิต 1 บิต

4.3.1.16 วงจรเลื่อนบิต 2 บิต

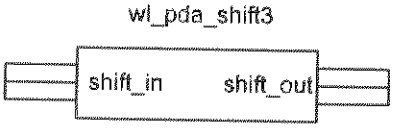
รูปที่ 4.91 วงจรเลื่อนบิต 2 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.92

รูปที่ 4.92 ผลการจำลองการทำงานของวงจรเลื่อนบิต 2 บิต

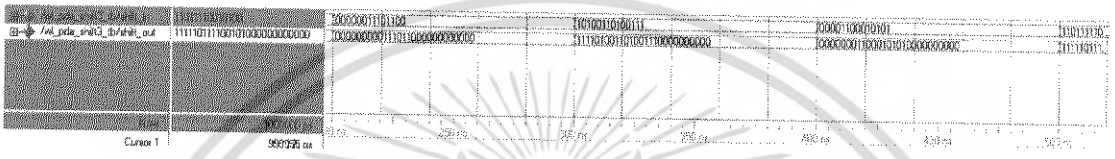
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.17 วงจรเลื่อนบิต 3 บิต



รูปที่ 4.93 วงจรเลื่อนบิต 3 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.94



รูปที่ 4.94 ผลการจำลองการทำงานของวงจรเลื่อนบิต 3 บิต

4.3.1.18 วงจรเลื่อนบิต 4 บิต

รูปที่ 4.95 วงจรเลื่อนบิต 4 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.96

รูปที่ 4.96 ผลการจำลองการทำงานของวงจรเลื่อนบิต 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

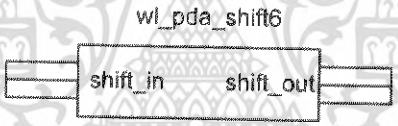
4.3.1.19 วงจรเลื่อนบิต 5 บิต

รูปที่ 4.97 วงจรเลื่อนบิต 5 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.98

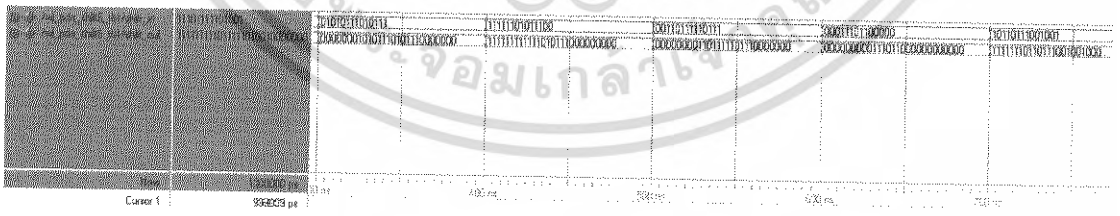
รูปที่ 4.98 ผลการจำลองการทำงานของวงจรเลื่อนบิต 5 บิต

4.3.1.20 วงจรเลื่อนบิต 6 บิต



รูปที่ 4.99 วงจรเลื่อนบิต 6 บิต

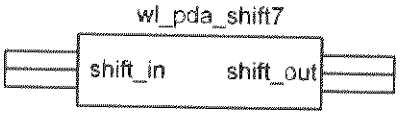
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.100



รูปที่ 4.100 ผลการจำลองการทำงานของวงจรเลื่อนบิต 6 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.21 วงจรเลื่อนบิต 7 บิต



รูปที่ 4.101 วงจรเลื่อนบิต 7 บิต

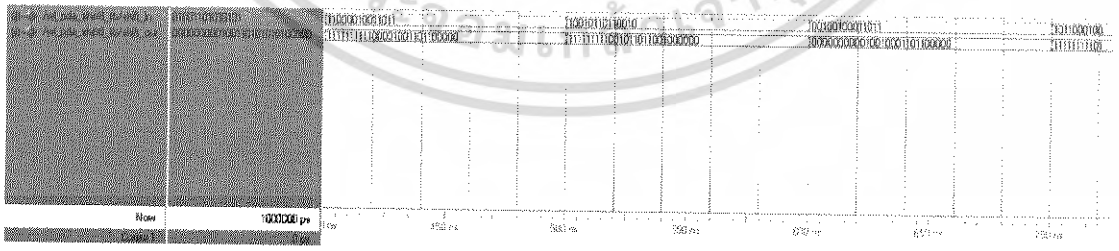
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.102

รูปที่ 4.102 ผลการจำลองการทำงานของวงจรเลื่อนบิต 7 บิต

4.3.1.22 วงจรเลื่อนบิต 8 บิต

รูปที่ 4.103 วงจรเลื่อนบิต 8 บิต

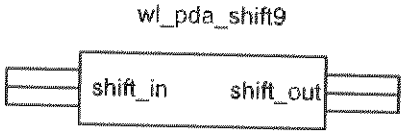
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.104



รูปที่ 4.104 ผลการจำลองการทำงานของวงจรเลื่อนบิต 8 บิต

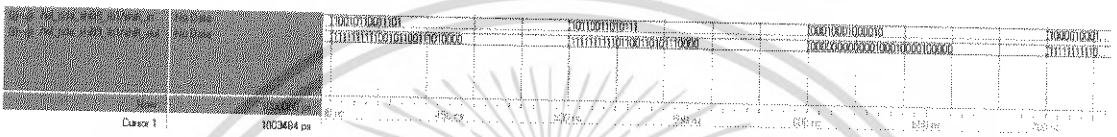
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.23 วงจรเลื่อนบิต 9 บิต



รูปที่ 4.105 วงจรเลื่อนบิต 9 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.106



รูปที่ 4.106 ผลการจำลองการทำงานของวงจรเลื่อนบิต 9 บิต

4.3.1.24 วงจรเลื่อนบิต 10 บิต

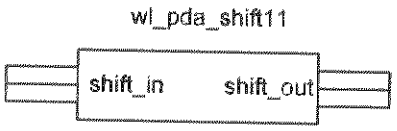
รูปที่ 4.107 วงจรเลื่อนบิต 10 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.108

รูปที่ 4.108 ผลการจำลองการทำงานของวงจรเลื่อนบิต 10 บิต

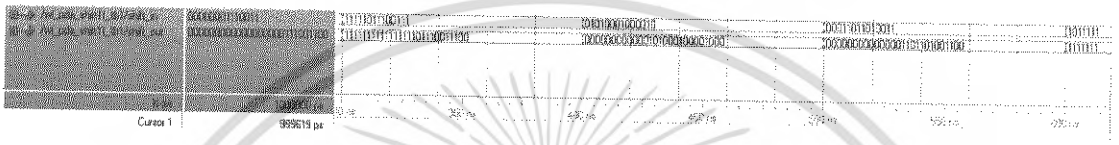
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.25 วงจรเลื่อนบิต 11 บิต



รูปที่ 4.109 วงจรเลื่อนบิต 11 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.110

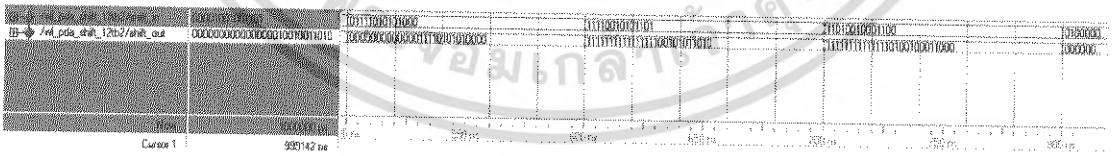


รูปที่ 4.110 ผลการจำลองการทำงานของวงจรเลื่อนบิต 11 บิต

4.3.1.26 วงจรเลื่อนบิต 12 บิต

รูปที่ 4.110 วงจรเลื่อนบิต 12 บิต

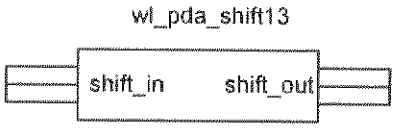
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.112



รูปที่ 4.112 ผลการจำลองการทำงานของวงจรเลื่อนบิต 12 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.27 วงจรเลื่อนบิต 13 บิต



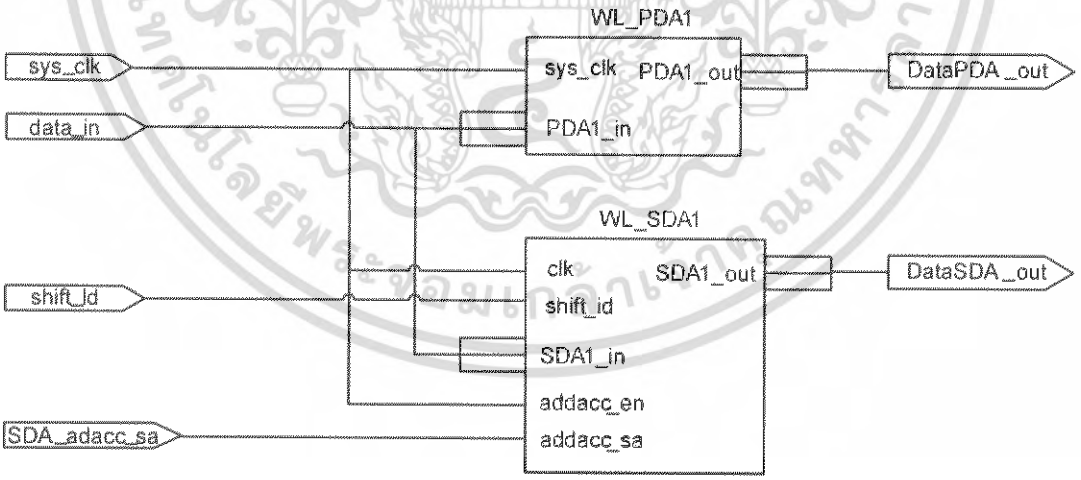
รูปที่ 4.113 วงจรเลื่อนบิต 13 บิต

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.114

รูปที่ 4.114 ผลการจำลองการทำงานของวงจรเลื่อนบิต 13 บิต

4.3.2 การเปรียบเทียบการทำงานของโครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน

การทดลองนี้จะป้อนข้อมูลอินพุต 000000011111101 ให้กับ DA แบบอนุกรมและขนาน ที่เก็บค่าคงที่ 1011011000011 0 เพื่อหาจำนวนสัญญาณนาฬิกาที่แต่ละแบบใช้ในการคำนวณผลลัพธ์ที่ถูกต้อง ซึ่งเท่ากับ 1111111101101 11

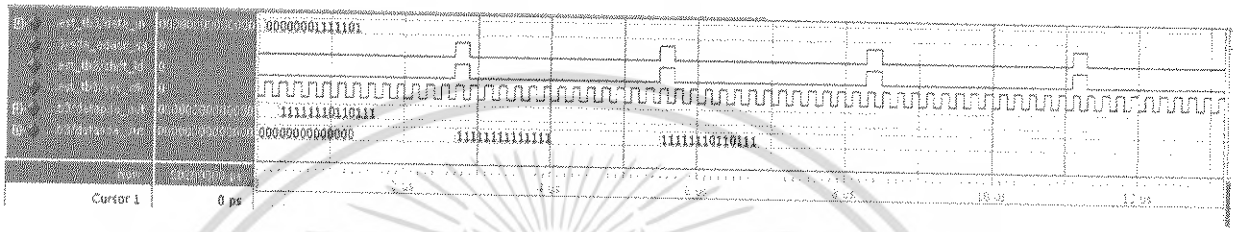


รูปที่ 4.115 วงจรการทดลองเปรียบเทียบการทำงานของ DA ทั้งสองแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

จากการทดลองพบว่า DA แบบอนุกรมจะทำการโหลดข้อมูลอินพุตเข้ามาเมื่อสัญญาณ Shift Id มีสถานะ 1 จากนั้นจะเลื่อนข้อมูลออกทีละบิตเพื่อไปชี้ค่าคงที่และบวกสะสมไปเรื่อย ๆ เมื่อข้อมูลถึงบิตที่ 14 จะนำไปลบกับค่าที่บวกสะสมไว้ จึงต้องใช้จำนวนสัญญาณนาฬิกาในการคำนวณผลลัพธ์แต่ละข้อมูลเท่ากับ 14 ลูก สำหรับ DA แบบขนานจะทำการโหลดข้อมูลเข้ามาเมื่อสัญญาณนาฬิกามีสถานะ 1 จากนั้นจะส่งข้อมูลทั้ง 14 บิตไปชี้ค่าคงที่พร้อมกัน นำผลที่ได้มาเลื่อนบิตตามที่กำหนด และบวกค่าของผลที่ได้จากการเลื่อนบิต จำนวนสัญญาณนาฬิกาที่ใช้ในการคำนวณผลลัพธ์แต่ละข้อมูลเท่ากับ 1 ลูก



รูปที่ 4.116 ผลการทดลองเปรียบเทียบการทำงานของ DA ทั้งสองแบบ

4.3.3 การเปรียบเทียบประสิทธิภาพสำหรับการสร้างจริงระหว่างโครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน

การเปรียบเทียบประสิทธิภาพสำหรับการสร้างจริงจะใช้เกณฑ์ในการเปรียบเทียบ 2 แบบ ได้แก่

1. ความเร็วสูงสุดในการประมวลผล (Throughput)
2. จำนวนทรัพยากรทางลอจิกที่ใช้ในการออกแบบ (Utilization)

หัวข้อเปรียบเทียบ	SDA	PDA
ความเร็วสูงสุดในการประมวลผล (MHz)	170.736	273.523
จำนวนทรัพยากรทางลอจิกที่ใช้ในการออกแบบ (Slice)	68	110

ตารางที่ 4.1 การเปรียบเทียบประสิทธิภาพของ SDA และ PDA

จากตารางที่ 4.1 จะพบว่า DA แบบขนานจะให้ความเร็วในการประมวลผลที่สูงกว่า DA แบบอนุกรม แต่ก็จะใช้จำนวนทรัพยากรที่สูงขึ้นตามไปด้วย ซึ่งในปริญญานิพนธ์นี้มุ่งเน้นในด้านความเร็วของการประมวลผลเป็นสำคัญ จึงเลือกใช้ DA แบบขนานและแก้ไขปัญหาการสิ้นเปลืองทรัพยากรโดยการเลือกใช้อุปกรณ์ FPGA ที่มีจำนวนลอจิกสูงมาก

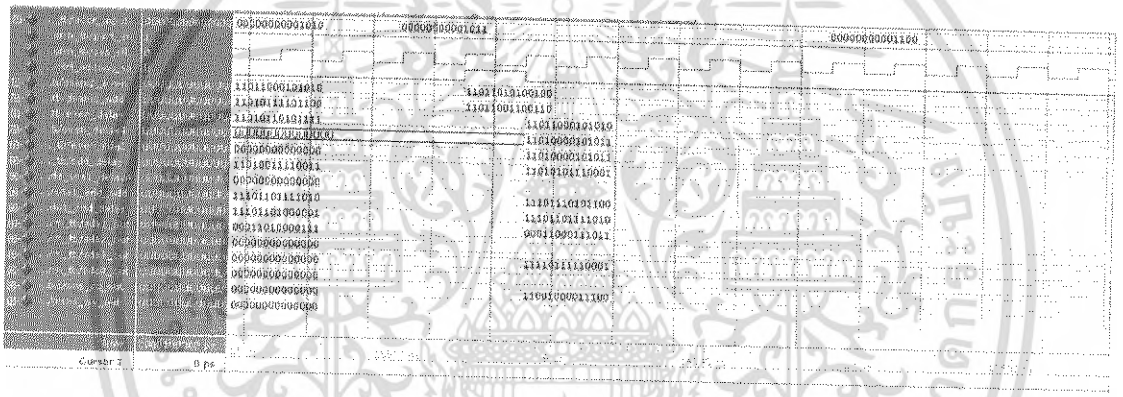
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การเปรียบเทียบการทำงานของตัวแปลงและตัวแปลงกลับเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรมและขนาน

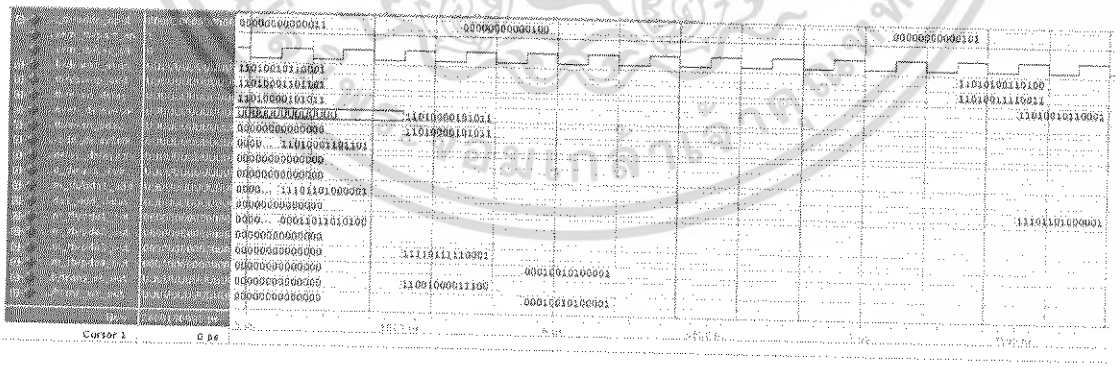
การทดลองนี้จะป้อนข้อมูลที่เป็นค่าตัวอย่างสัญญาณเสียงให้กับตัวแปลงเวฟเลต ซึ่งจะได้ค่าสัมประสิทธิ์สเกลลิ่งและสัมประสิทธิ์เวฟเลตออกมา โดยจะเปรียบเทียบเวลาที่ใช้ในการคำนวณผลจนได้สัมประสิทธิ์สเกลลิ่งค่าเท่ากับ 1100100001100 จากนั้นจะใช้สัมประสิทธิ์ทั้งสองแบบที่ได้ทั้งหมดป้อนเข้าตัวแปลงกลับเวฟเลต โดยจะเปรียบเทียบเวลาที่ใช้ในการคำนวณผลจนได้ข้อมูลสัญญาณเสียงที่สร้างกลับคืนค่าเท่ากับ 1101000010101O

ผลการทดลอง

จากการทดลองพบว่าเวลาที่ตัวแปลงเวฟเลตที่ใช้ DA แบบอนุกรมใช้คำนวณจนได้สัมประสิทธิ์สเกลลิ่งที่กำหนดเท่ากับ 15900 ns ตัวแปลงเวฟเลตที่ใช้ DA แบบขนานใช้เวลาเท่ากับ 5500 ns สำหรับตัวแปลงกลับเวฟเลตที่ใช้ DA แบบอนุกรมใช้เวลาในการคำนวณผลจนได้ข้อมูลสัญญาณเสียงที่สร้างกลับคืนค่าที่กำหนดเท่ากับ 33900 ns ตัวแปลงกลับเวฟเลตที่ใช้ DA แบบขนานใช้เวลาเท่ากับ 14300 ns

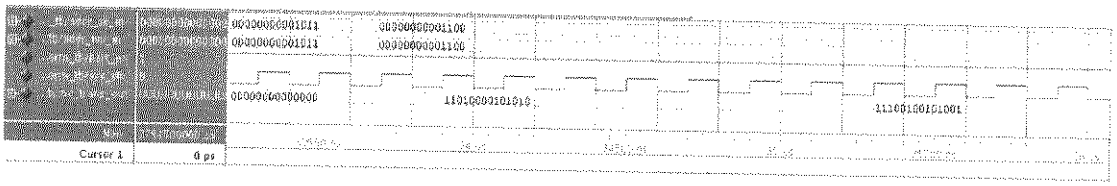


รูปที่ 4.117 ผลการทดลองของตัวแปลงเวฟเลทที่ใช้ DA แบบอนุกรม

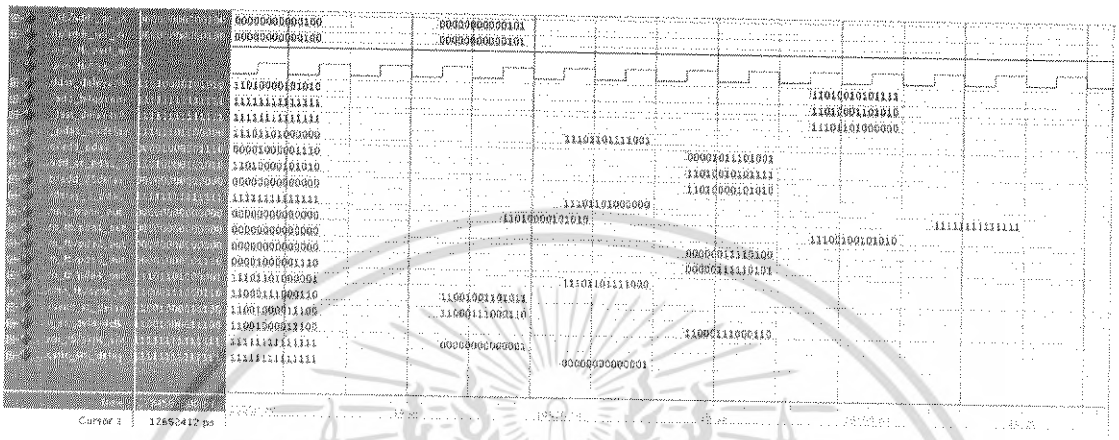


รูปที่ 4.118 ผลการทดลองของตัวแปลงเวฟเลทที่ใช้ DA แบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.119 ผลการทดลองของตัวแปลงกลับเวฟเลทที่ใช้ DA แบบอนุกรม



รูปที่ 4.120 ผลการทดลองของตัวแปลงกลับเวฟเลทที่ใช้ DA แบบขนาน

4.4 การเปรียบเทียบเพื่อหาค่าผิดพลาดของการแปลงเวฟเลตที่ได้จากการจำลองการทำงาน

4.4.1 การเปรียบเทียบการแปลงเวฟเลตเมื่ออินพุตเป็นสัญญาณไซน์

การทดลองนี้จะป้อนข้อมูลค่าตัวอย่างของสัญญาณรูปไซน์จำนวน 2000 ตัวอย่าง ให้กับตัวแปลงเวฟเลตที่ใช้ DA ทั้งสองแบบ เพื่อเปรียบเทียบสัมประสิทธิ์สเกลลิ่งและสัมประสิทธิ์เวฟเลตที่ได้กับค่าที่ได้จากการจำลองการทำงานด้วยโปรแกรมแมทแลบ จากนั้นนำค่าสัมประสิทธิ์ทั้งสองแบบป้อนให้กับตัวแปลงกลับเวฟเลต เพื่อเปรียบเทียบค่าสัญญาณที่สร้างกลับคืนกับสัญญาณต้นฉบับและสัญญาณที่สร้างกลับคืนด้วยโปรแกรมแมทแลบ

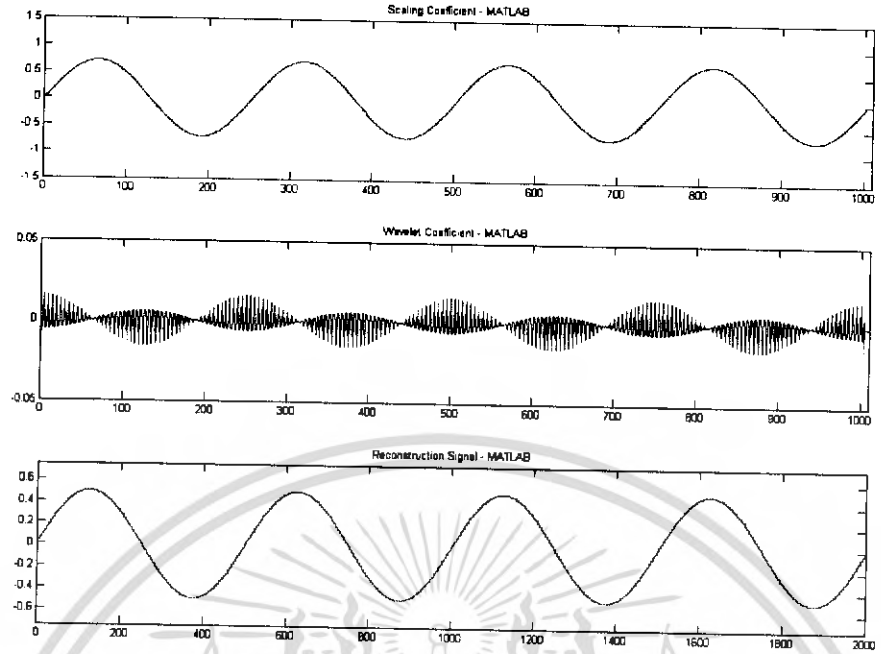
ผลการทดลอง

จากการทดลองพบว่าค่าความผิดพลาด (MSE) ระหว่างค่าจากโปรแกรมแมทแลบและตัวแปลงเวฟเลตแสดงได้ตามตารางที่ 4.2

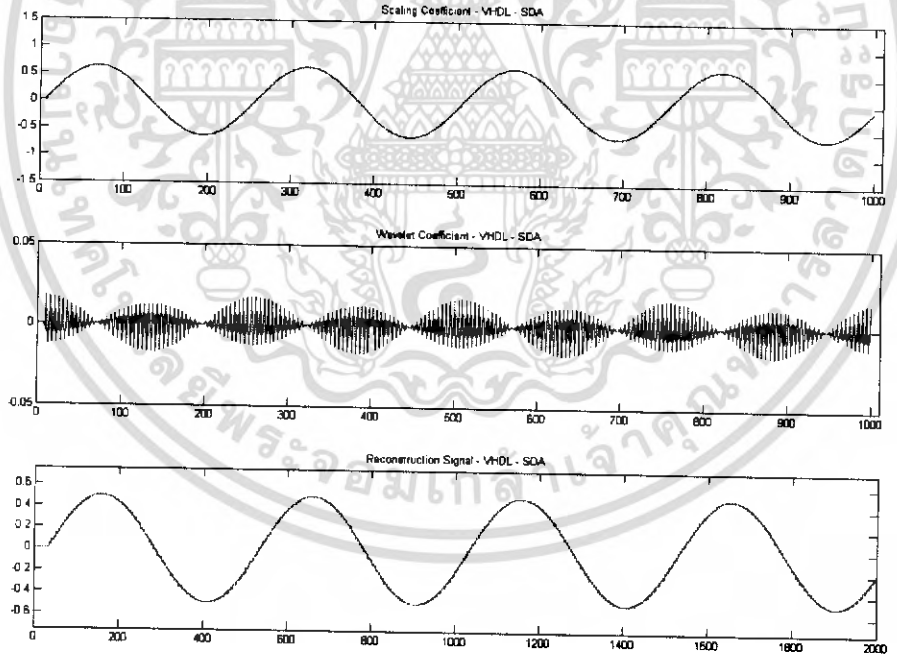
ชนิดของตัวแปลง	สัมประสิทธิ์สเกลลิ่ง	สัมประสิทธิ์เวฟเลต	สัญญาณที่สร้างคืน
ตัวแปลงเวฟเลตที่ใช้ DA อนุกรม	0.0038	5.692×10^{-5}	-
ตัวแปลงเวฟเลตที่ใช้ DA ขนาน	0.0028	1.212×10^{-5}	-
ตัวแปลงกลับเวฟเลตที่ใช้ DA อนุกรม	-	-	0.0034
ตัวแปลงกลับเวฟเลตที่ใช้ DA ขนาน	-	-	2.615×10^{-4}

ตารางที่ 4.2 ค่าความผิดพลาดระหว่างค่าจากโปรแกรมแมทแลบและตัวแปลงเวฟเลตแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

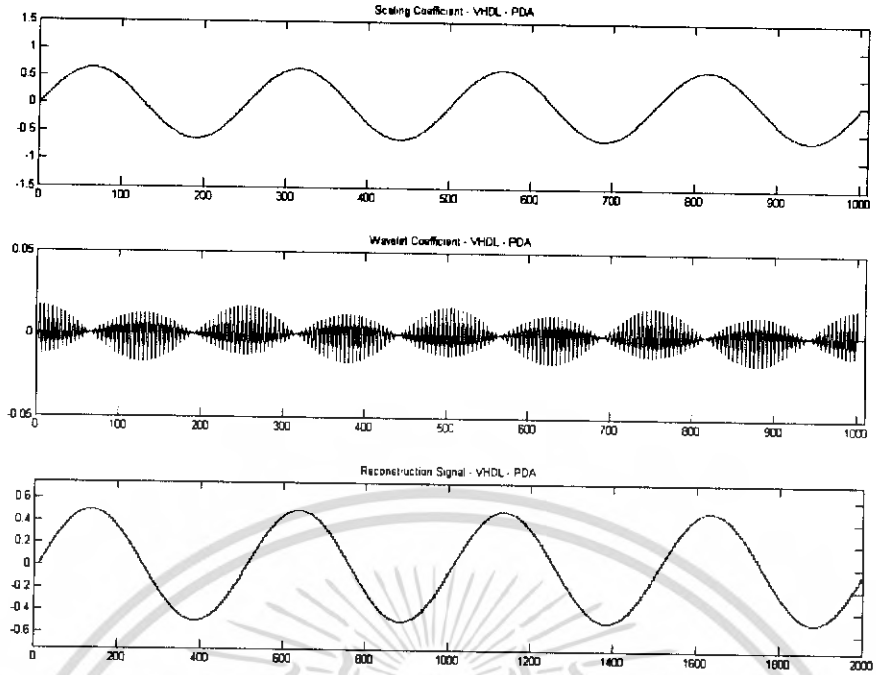


รูปที่ 4.121 ผลการแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ

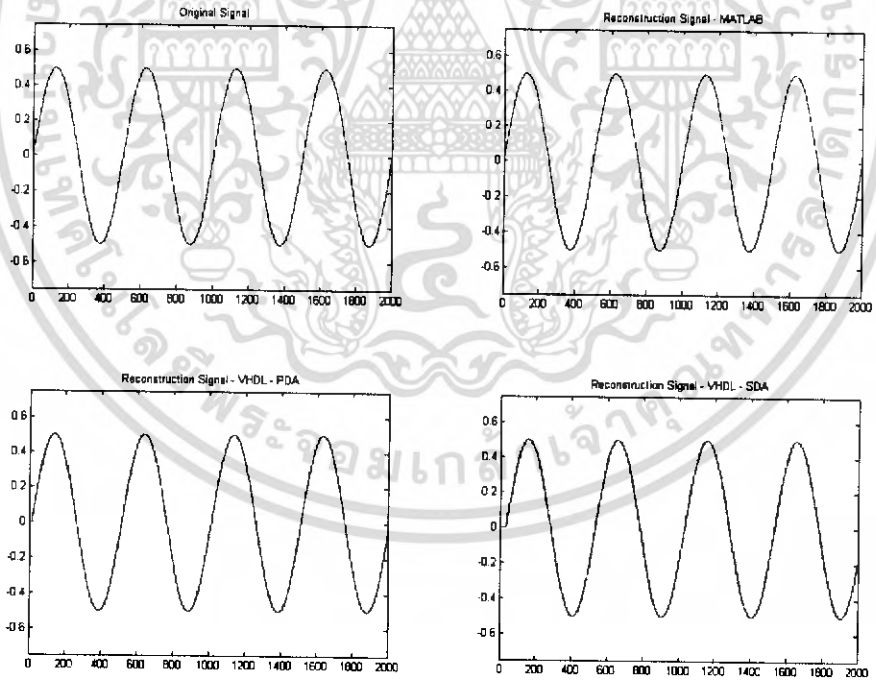


รูปที่ 4.122 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (SDA) โดยใช้โปรแกรม ModelSim XE III 6.0d

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.123 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (PDA) โดยใช้โปรแกรม ModelSim XE III 6.0d



รูปที่ 4.124 การเปรียบเทียบสัญญาณที่สร้างคืนจากการแปลงเวฟเลตทั้ง 3 วิธีกับสัญญาณต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การเปรียบเทียบการแปลงเวฟเลตเมื่ออินพุตเป็นสัญญาณเสียง

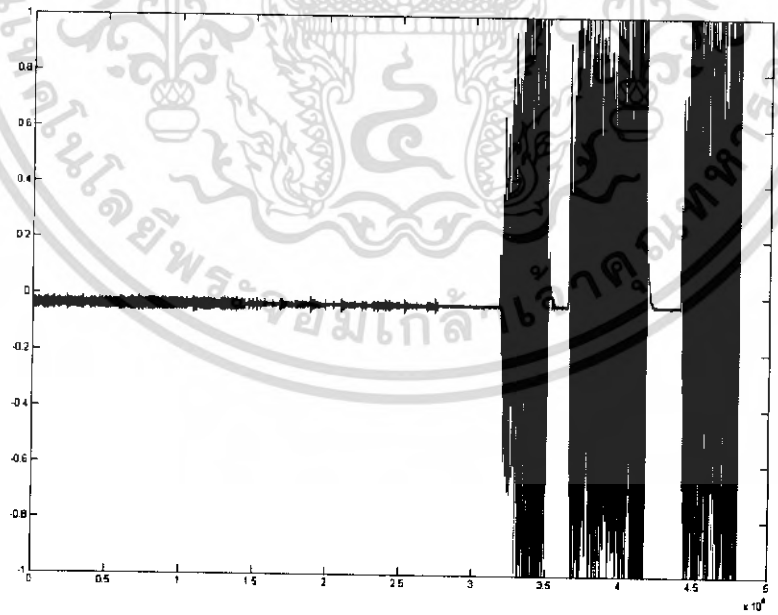
การทดลองนี้จะป้อนข้อมูลค่าตัวอย่างของสัญญาณเสียงจำนวน 2000 ตัวอย่าง ให้กับตัวแปลงเวฟเลตที่ใช้ DA ทั้งสองแบบ เพื่อเปรียบเทียบสัมประสิทธิ์สเกลลิ่งและสัมประสิทธิ์เวฟเลตที่ได้กับค่าที่ได้จากการจำลองการทำงานด้วยโปรแกรมแมทแลบ จากนั้นนำค่าสัมประสิทธิ์ทั้งสองแบบป้อนให้กับตัวแปลงกลับเวฟเลต เพื่อเปรียบเทียบค่าสัญญาณที่สร้างกลับคืนกับสัญญาณต้นฉบับและสัญญาณที่สร้างกลับคืนด้วยโปรแกรมแมทแลบ

ผลการทดลอง

จากการทดลองพบว่าค่าความผิดพลาด (MSE) ระหว่างค่าจากโปรแกรมแมทแลบและตัวแปลงเวฟเลตทั้งสองแบบแสดงได้ตามตารางที่ 4.3

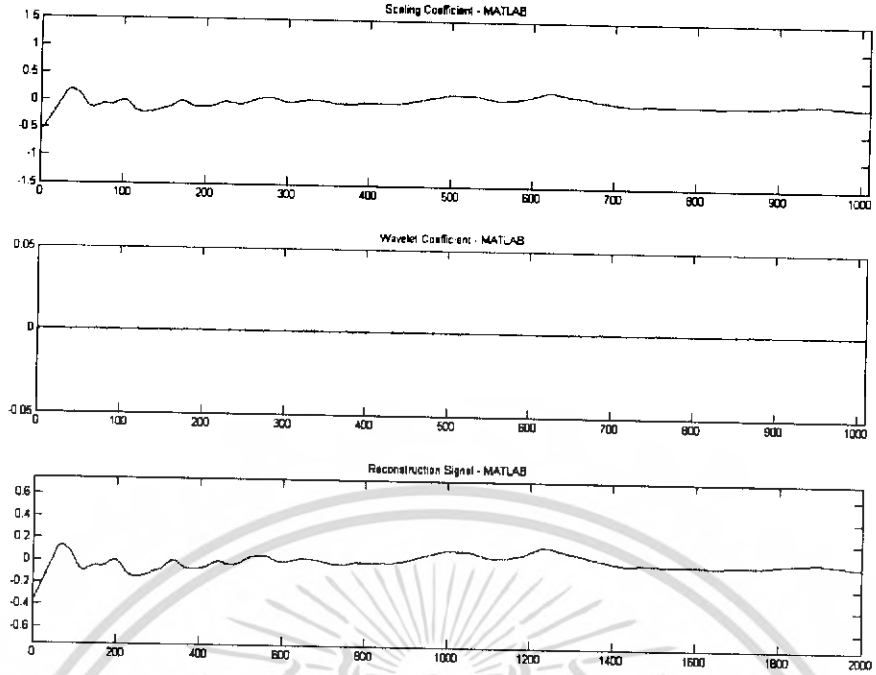
ชนิดของตัวแปลง	สัมประสิทธิ์สเกลลิ่ง	สัมประสิทธิ์เวฟเลต	สัญญาณที่สร้างคืน
ตัวแปลงเวฟเลตที่ใช้ DA อนุกรม	0.0018	2.114×10^{-5}	-
ตัวแปลงเวฟเลตที่ใช้ DA ขนาน	6.267×10^{-4}	2.119×10^{-5}	-
ตัวแปลงกลับเวฟเลตที่ใช้ DA อนุกรม	-	-	0.0038
ตัวแปลงกลับเวฟเลตที่ใช้ DA ขนาน	-	-	0.0010

ตารางที่ 4.3 ค่าความผิดพลาดระหว่างค่าจากโปรแกรมแมทแลบและตัวแปลงเวฟเลตแต่ละแบบ

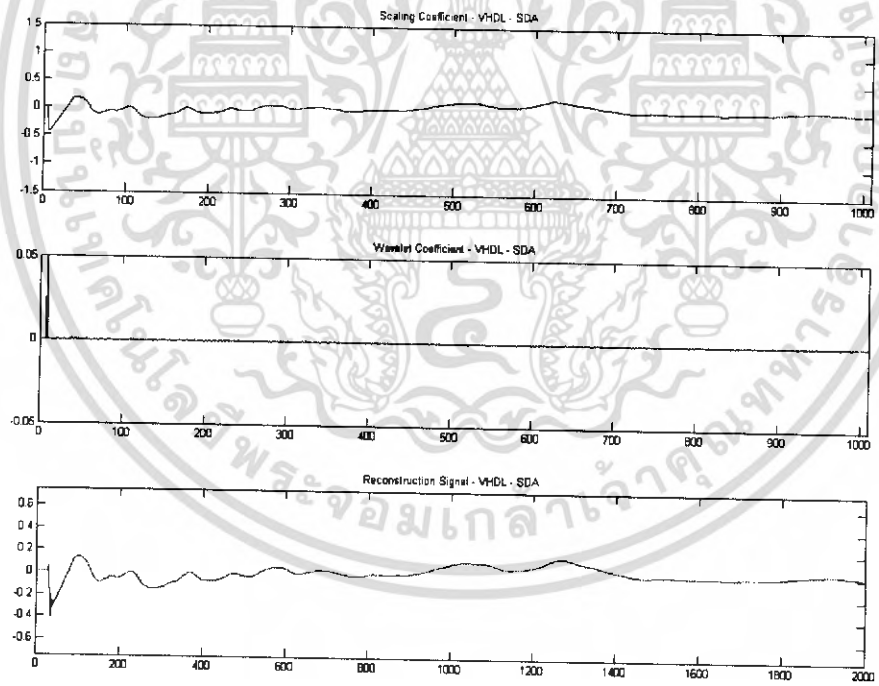


รูปที่ 4.125 สัญญาณเสียงที่ใช้ทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

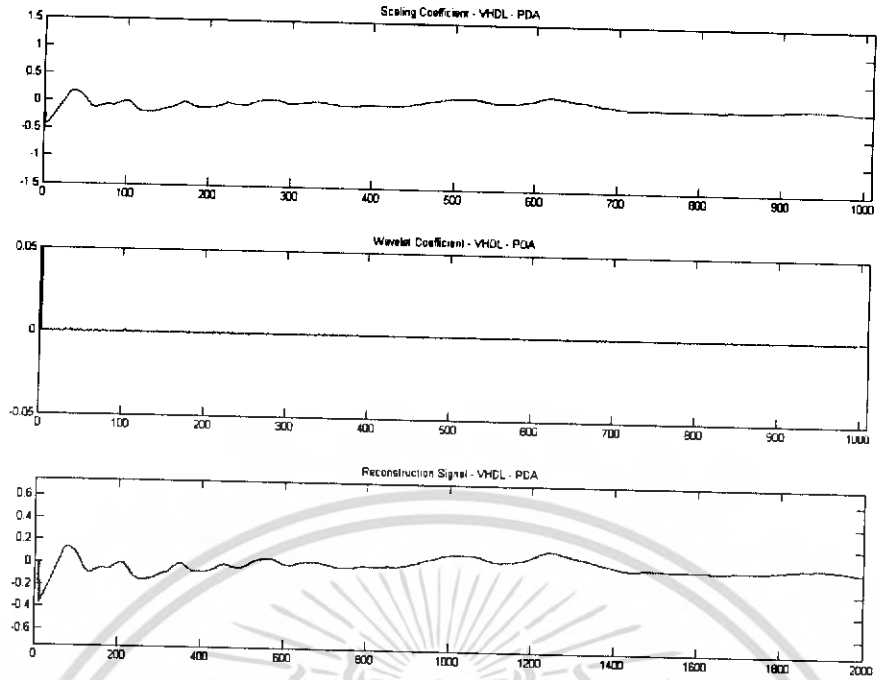


รูปที่ 4.126 ผลการแปลงเวฟเลตโดยใช้โปรแกรมแมทแลบ

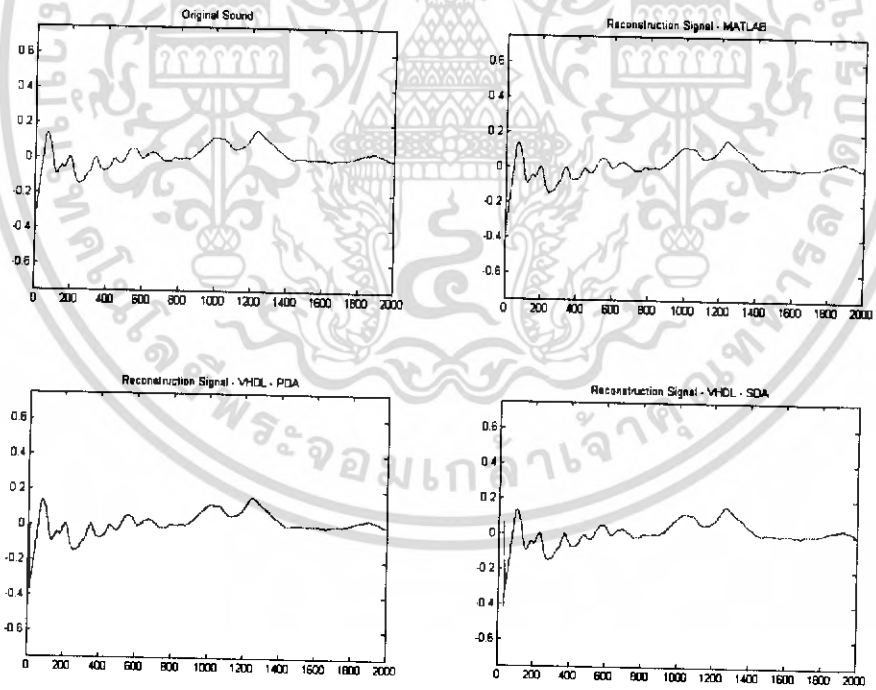


รูปที่ 4.127 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (SDA) โดยใช้โปรแกรม ModelSim XE III 6.0d

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.128 ผลการแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (PDA) โดยใช้โปรแกรม ModelSim XE III 6.0d



รูปที่ 4.129 การเปรียบเทียบสัญญาณที่สร้างคืนจากการแปลงเวฟเลตทั้ง 3 วิธีกับสัญญาณต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การจำลองการทำงานวงจรควบคุมการรับส่งข้อมูลใช้โปรแกรม ModelSim XE III 6.0d

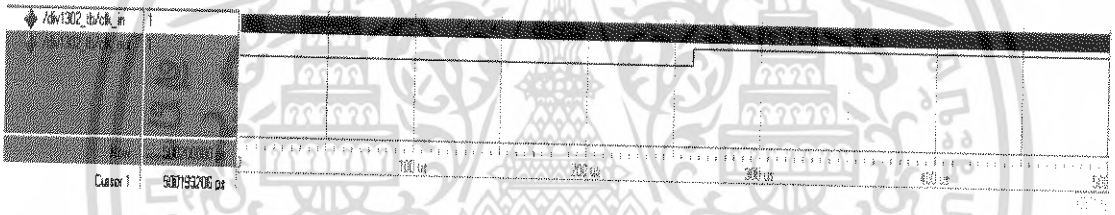
การทดลองนี้จะจำลองการทำงานของอุปกรณ์สำหรับการรับส่งข้อมูลผ่านพอร์ทอนุกรม

4.5.1 วงจรหารความถี่

วงจรหารความถี่ทำหน้าที่หารความถี่ของระบบให้ได้เป็นความถี่ที่เท่ากับบอดเรท (Baud Rate) ที่ใช้ในการรับส่งบิตข้อมูลทางพอร์ตอนุกรม อุปกรณ์ FPGA ที่ใช้มีแหล่งกำเนิดความถี่ 25 MHz สำหรับใช้เป็นความถี่ของระบบ ดังนั้นจึงต้องสร้างวงจรหารความถี่ลงประมาณ 2604 เท่า เพื่อให้ได้ความถี่สำหรับการรับส่งเท่ากับบอดเรท 9600 บิตต่อวินาที

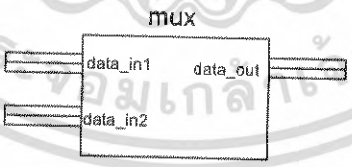
รูปที่ 4.130 วงจรหารความถี่

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.131



รูปที่ 4.131 ผลการจำลองการทำงานของวงจรหารความถี่

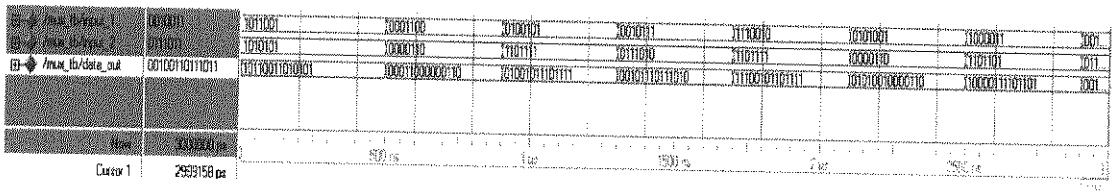
4.5.2 วงจรมัลติเพล็กซ์



รูปที่ 4.132 วงจรมัลติเพล็กซ์

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.133

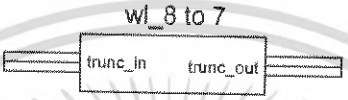
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.133 ผลการจำลองการทำงานของวงจรมัลติเพล็กซ์

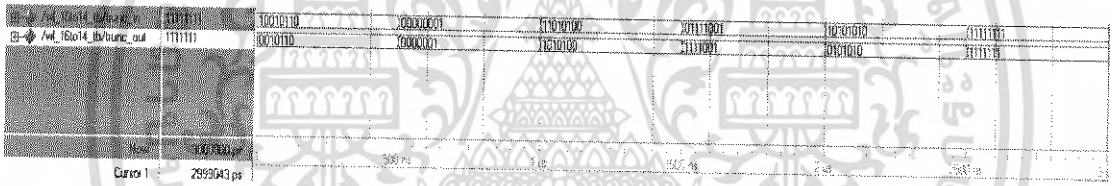
4.5.3 วงจรตัดความยาวบิตข้อมูล

เป็นวงจรที่ใช้ตัดบิตแรกสุดของข้อมูลออก 1 บิต เพื่อให้ได้ความยาว 7 บิต



รูปที่ 4.134 วงจรตัดความยาวบิตข้อมูล

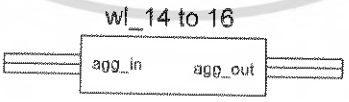
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.135



รูปที่ 4.135 ผลการจำลองการทำงานของวงจรวงจรตัดความยาวบิตข้อมูล

4.5.4 วงจรเพิ่มความยาวบิตข้อมูล

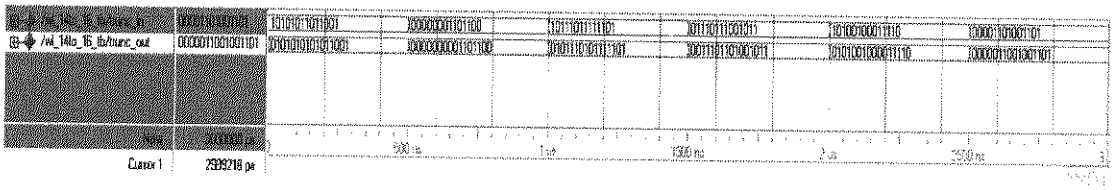
เป็นวงจรที่ใช้ในการเพิ่มบิตที่ตำแหน่งแรกสุด (หน้าบิตที่ 14) และระหว่างบิตที่ 6 และ 7 ของข้อมูลที่จะเขียนเข้าหน่วยความจำ ทำให้ความยาวข้อมูลจาก 14 บิตเพิ่มเป็น 16 บิต



รูปที่ 4.136 วงจรเพิ่มความยาวบิตข้อมูล

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.137

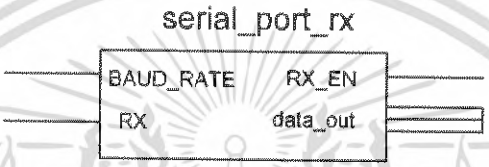
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.137 ผลการจำลองการทำงานของวงจรเพิ่มความยาวบิตข้อมูล

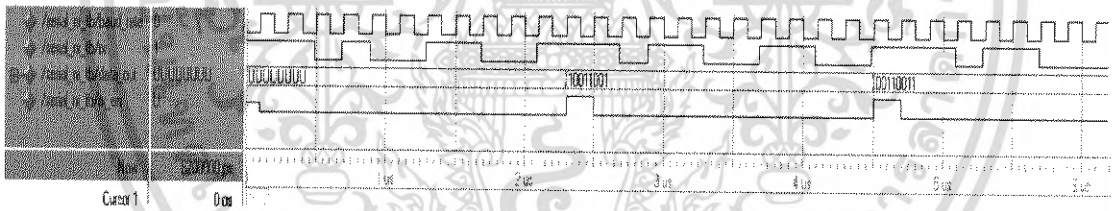
4.5.5 วงจรรับข้อมูลจากพอร์ทอนุกรม

เป็นอุปกรณ์ที่รับข้อมูลจากพอร์ทอนุกรมทีละบิต แล้วตัดบิตสตาร์ทและบิตสต็อปออกเพื่อส่งเฉพาะข้อมูลออกมาเป็นข้อมูลขนาด 8 บิต



รูปที่ 4.138 วงจรรับข้อมูลจากพอร์ทอนุกรม

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.139

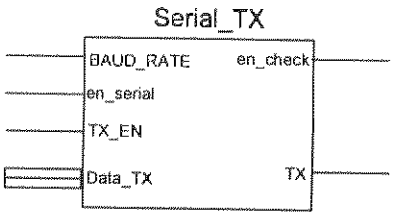


รูปที่ 4.139 ผลการจำลองการทำงานของวงจรรับข้อมูลจากพอร์ทอนุกรม

4.5.6 วงจรส่งข้อมูลเข้าพอร์ทอนุกรม

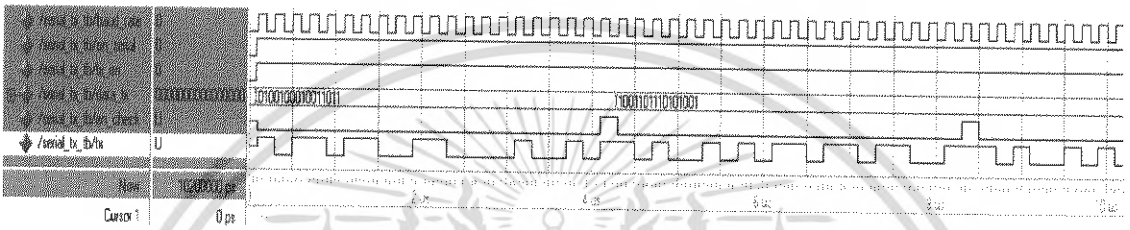
เป็นอุปกรณ์ที่นำข้อมูลความยาว 14 บิต จากหน่วยความจำ มาแทรกบิตสตาร์ทและบิตสต็อปออกเพื่อส่งผ่านพอร์ทอนุกรมไปยังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



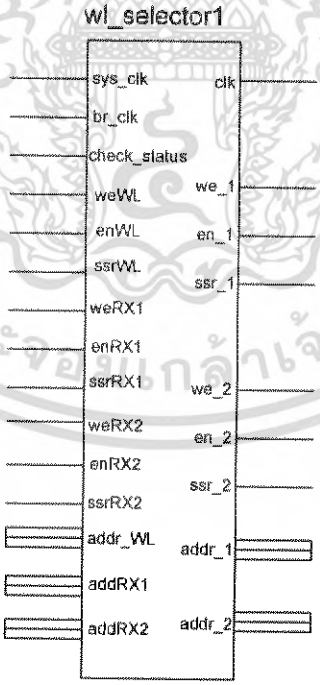
รูปที่ 4.140 วงจรส่งข้อมูลเข้าพอร์ทอนุกรม

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.141



รูปที่ 4.141 ผลการจำลองการทำงานของวงจรส่งข้อมูลเข้าพอร์ทอนุกรม

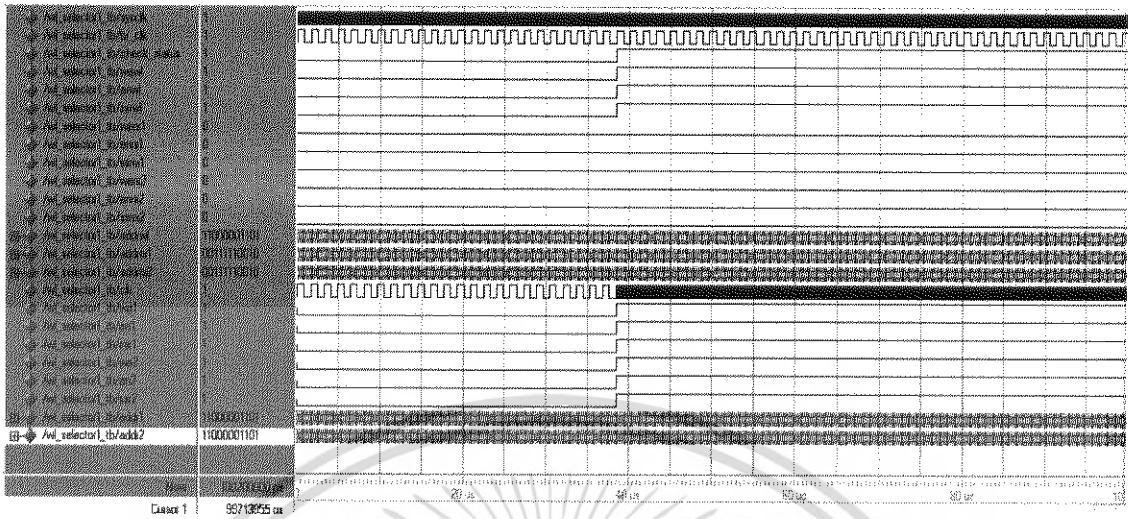
4.5.7 วงจรเลือกการอ่านเขียนหน่วยความจำ 1



รูปที่ 4.142 วงจรเลือกการอ่านเขียนหน่วยความจำ 1

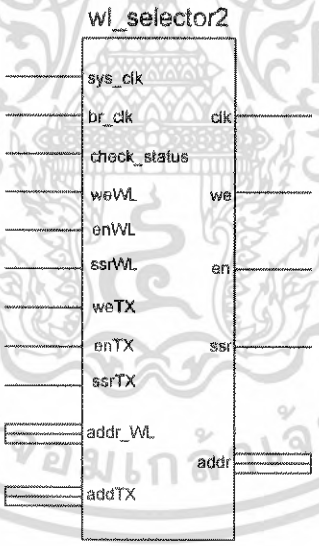
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.143



รูปที่ 4.143 ผลการจำลองการทำงานของวงจรเลือกการอ่านเขียนหน่วยความจำ 1

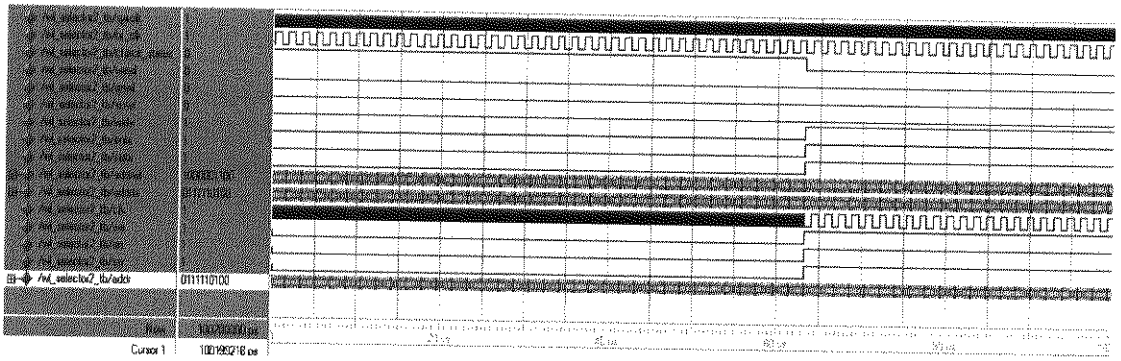
4.5.8 วงจรเลือกการอ่านเขียนหน่วยความจำ 2



รูปที่ 4.144 วงจรเลือกการอ่านเขียนหน่วยความจำ 2

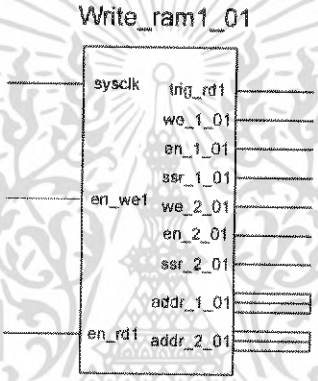
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.145

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



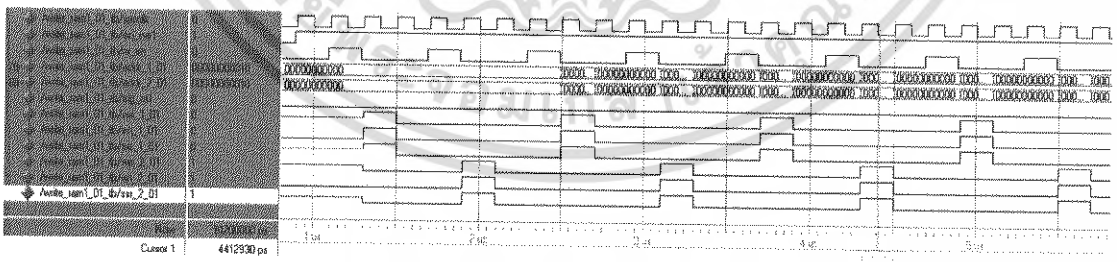
รูปที่ 4.145 ผลการจำลองการทำงานของวงจรเลือกการอ่านเขียนหน่วยความจำ 2

4.5.9 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 1 (วงจรควบคุม 1)



รูปที่ 4.146 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 1

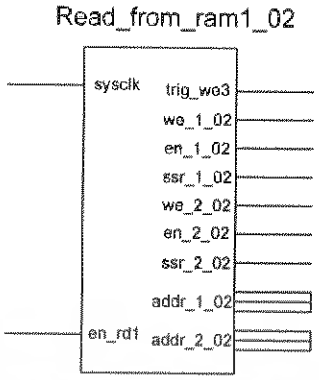
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.147



รูปที่ 4.147 ผลการจำลองการทำงานของวงจรควบคุมการเขียนหน่วยความจำชุดที่ 1

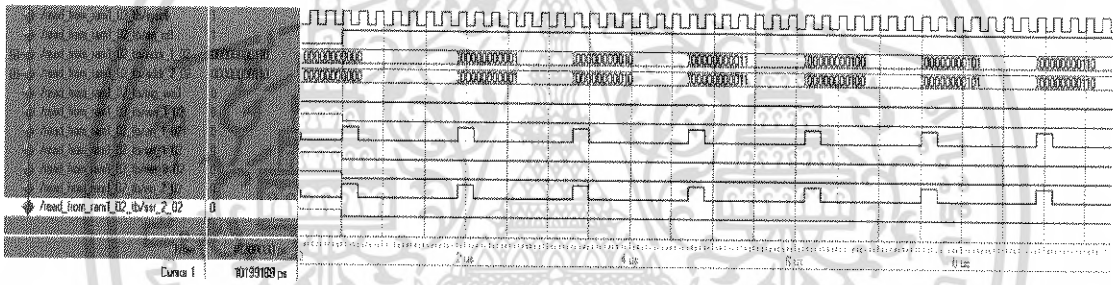
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.10 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 1 (วงจรควบคุม 2)



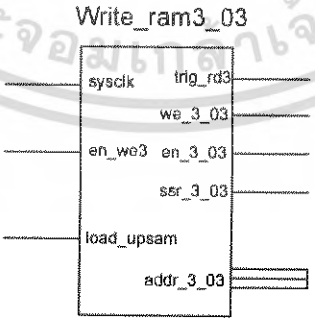
รูปที่ 4.148 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 1

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.149



รูปที่ 4.149 ผลการจำลองการทำงานของวงจรควบคุมการอ่านหน่วยความจำชุดที่ 1

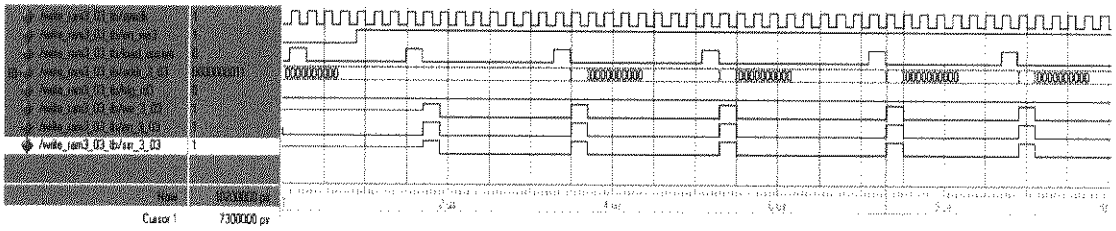
4.5.11 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 2 (วงจรควบคุม 3)



รูปที่ 4.150 วงจรควบคุมการเขียนหน่วยความจำชุดที่ 2

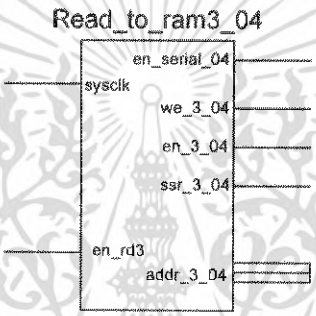
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.151



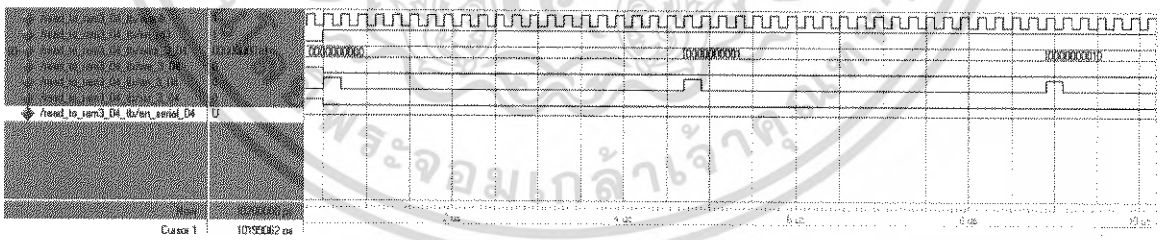
รูปที่ 4.151 ผลการจำลองการทำงานของวงจรควบคุมการเขียนหน่วยความจำชุดที่ 2

4.5.12 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 2 (วงจรควบคุม 4)



รูปที่ 4.152 วงจรควบคุมการอ่านหน่วยความจำชุดที่ 2

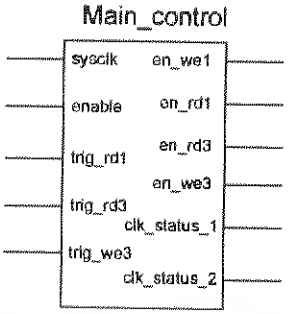
เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.153



รูปที่ 4.153 ผลการจำลองการทำงานของวงจรควบคุมการอ่านหน่วยความจำชุดที่ 2

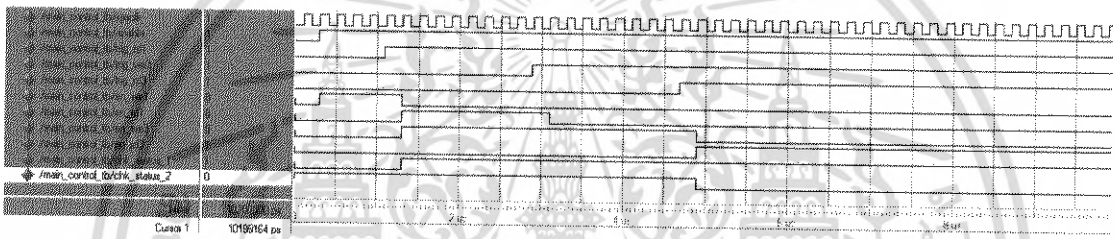
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.13 วงจรควบคุมหลัก



รูปที่ 4.154 วงจรควบคุมหลัก

เมื่อจำลองการทำงานด้วยโปรแกรม ModelSim XE III 6.0d ได้ผลการทดลอง ดังรูป 4.155



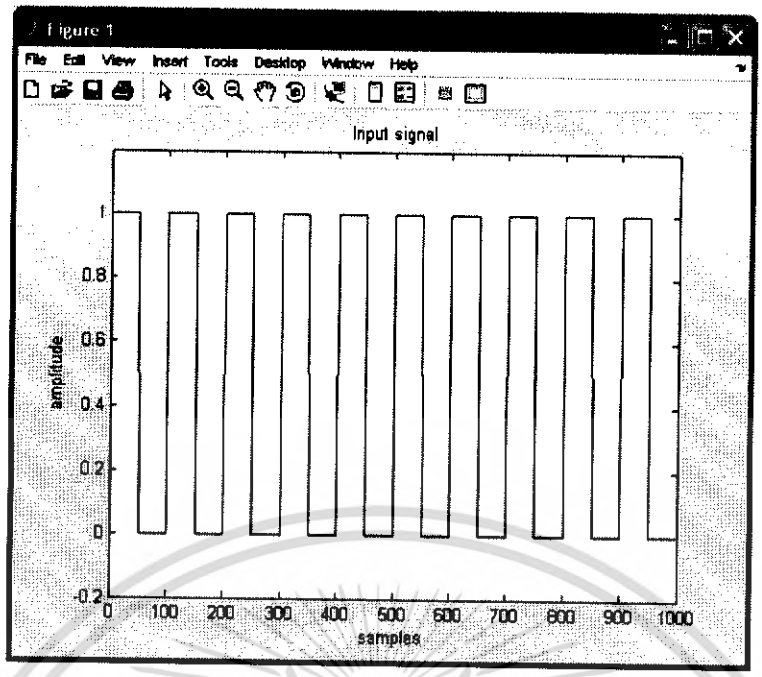
รูปที่ 4.155 ผลการจำลองการทำงานของวงจรควบคุมหลัก

4.6 การทดลองตัวแปลงเวฟเลตที่สร้างจริงลงอุปกรณ์ FPGA และแสดงผลผ่านจอคอมพิวเตอร์

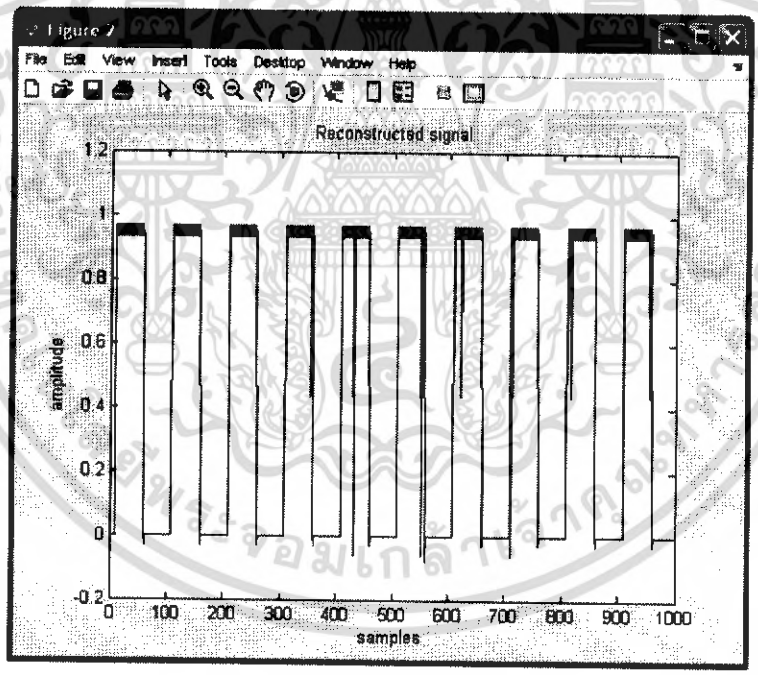
การทดลองนี้เป็นการทดสอบการทำงานจริงของตัวแปลงเวฟเลตที่ออกแบบและสร้างขึ้นมา โดยโครงสร้างของตัวแปลงเวฟเลตเป็นแบบลิฟท์ติ้งและใช้โครงสร้างเลขคณิตกระจายแบบขนานแทนการคูณโดยตรง

ในการทดลองจะส่งสัญญาณอินพุตจากพอร์ทอนุกรมของคอมพิวเตอร์ส่งไปยังอุปกรณ์ FPGA ที่มีวงจรตัวแปลงเวฟเลต ซึ่งสัญญาณที่ใช้มี 3 แบบ คือ สัญญาณรูปสี่เหลี่ยม (Square wave) สัญญาณรูปไซน์ (Sinusoidal wave) และสัญญาณเสียง (Sound wave) เมื่อตัวแปลงเวฟเลตได้รับสัญญาณจะคำนวณผลของการแปลงเวฟเลตและการแปลงกลับเวฟเลต แล้วส่งค่าที่ได้จากการคำนวณกลับคืนมายังคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมให้กับโปรแกรมแมทแลบ เพื่อแสดงผลเป็นกราฟของสัญญาณอินพุตและสัญญาณเอาท์พุตที่สร้างกลับคืนมาได้ โดยจะมีหน้าต่างแสดงผลการทดลองดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

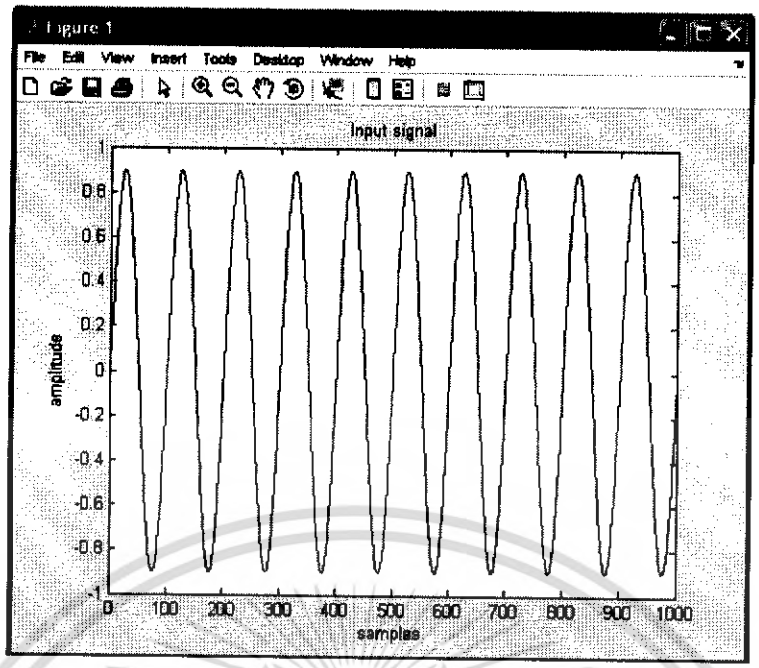


รูปที่ 4.156 สัญญาณอินพุตเป็นสัญญาณสี่เหลี่ยม

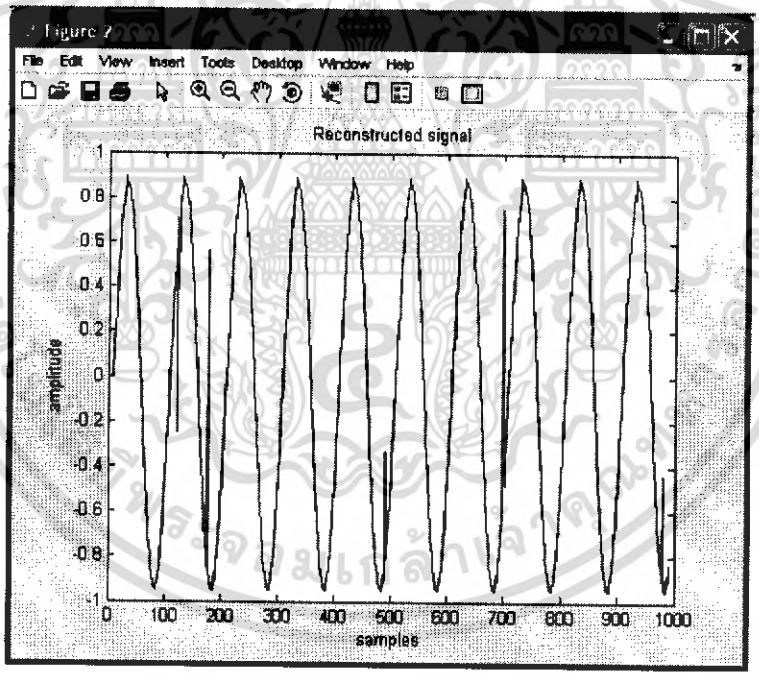


รูปที่ 4.157 สัญญาณเอาท์พุตของสัญญาณสี่เหลี่ยมที่สร้างกลับคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

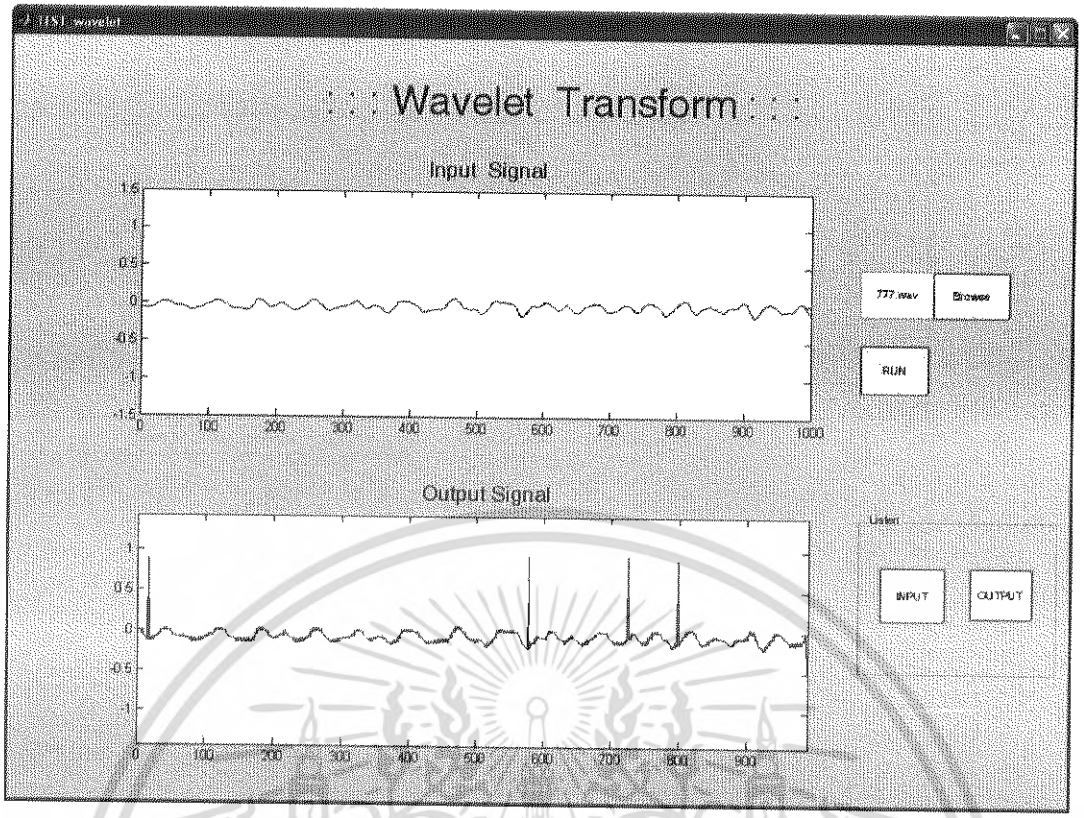


รูปที่ 4.158 สัญญาณอินพุตเป็นสัญญาณรูปไซน์

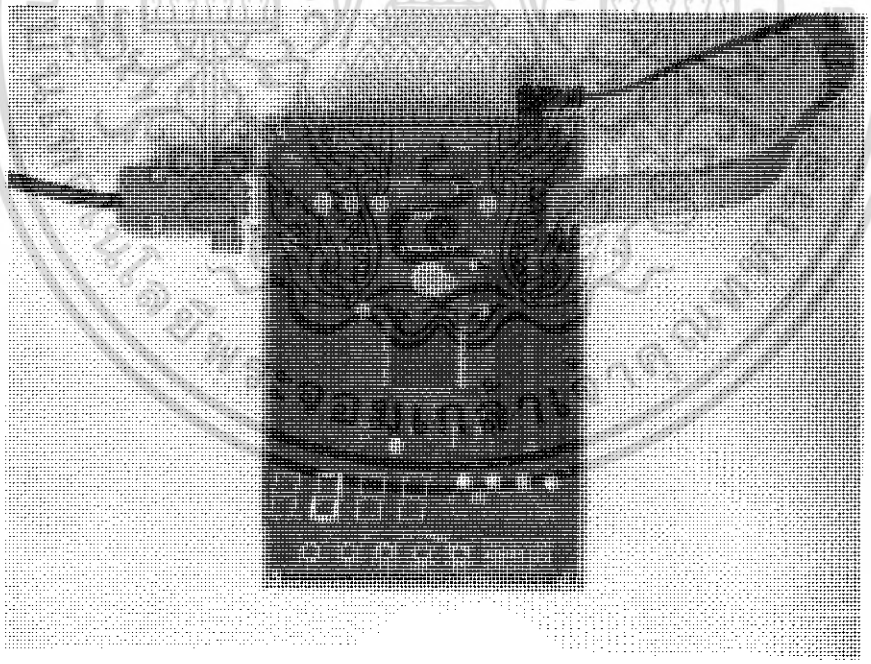


รูปที่ 4.159 สัญญาณเอาท์พุตของสัญญาณรูปไซน์ที่สร้างกลับคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.160 การเปรียบเทียบระหว่างสัญญาณเสียงอินพุตกับสัญญาณเสียงที่สร้างกลับคืน



รูปที่ 4.161 อุปกรณ์ FPGA ขณะทดสอบการทำงานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

ตัวแปลงเวฟเลตที่นำเสนอในปริญญานิพนธ์นี้ประกอบด้วยวงจรแปลงเวฟเลตและวงจรแปลงกลับเวฟเลต โดยการออกแบบจะใช้วิธีการเขียนโปรแกรมภาษาวีเอชดีแอลเพื่อสร้างอุปกรณ์ที่ทำงานได้ตามที่กำหนด และทดสอบอุปกรณ์ต่างๆ โดยการจำลองการทำงานจนได้อุปกรณ์ที่ทำงานได้อย่างถูกต้อง จากนั้นนำอุปกรณ์ย่อยมาประกอบรวมเป็นวงจรแปลงเวฟเลตและวงจรแปลงกลับเวฟเลต โดยใช้โครงสร้างแบบลิฟต์ติ้ง และนำโครงสร้างของเลขคณิตกระจายแบบขนานมาประยุกต์แทนการคูณโดยตรง และจำลองการทำงานของวงจรทั้งหมดอีกครั้งจนได้ผลที่ถูกต้อง จากนั้นจะโปรแกรมลงอุปกรณ์ FPGA

ปัญหาที่พบระหว่างการออกแบบ คือ จังหวะเวลาการทำงานของอุปกรณ์ย่อย ซึ่งแต่ละส่วนจะต้องทำงานกันได้อย่างสัมพันธ์กันถึงจะให้ผลลัพธ์ที่ถูกต้อง เมื่อวงจรซับซ้อนขึ้นจะต้องเพิ่มความละเอียดในการตรวจสอบตามไปด้วย แต่เนื่องจากการจำลองการทำงานใช้โปรแกรม ModelSim XE III 6.0d ที่สามารถแสดงผลเป็นกราฟเวลาที่บอกค่าสถานะของสัญญาณและจังหวะเวลา ซึ่งช่วยลดความยุ่งยากในการตรวจสอบลงได้มาก นอกจากนี้การออกแบบวงจรประเภทนี้ยังเป็นการออกแบบสำหรับเฉพาะงานจึงไม่สามารถหาวงจรมาตรฐานเพื่อนำมาใช้งานได้

สำหรับผลการจำลองการทำงานของโครงสร้างเลขคณิตกระจาย พบว่าแบบขนานจะใช้เวลาที่น้อยกว่าแบบอนุกรม เมื่อนำมาใช้ในตัวแปลงเวฟเลตที่มีการคูณหลายครั้งจะทำให้ความแตกต่างของเวลานี้สะสมไปเรื่อย ๆ จนเห็นความแตกต่างได้อย่างชัดเจนที่ผลลัพธ์สุดท้าย แต่เมื่อเปรียบเทียบผลของการแปลงและการแปลงกลับเวฟเลตที่ใช้โครงสร้างทั้ง 2 แบบกับค่าที่ได้จากโปรแกรมแมทแลบ พบว่าทั้ง 2 แบบให้ผลที่ใกล้เคียงกับค่าจากโปรแกรมแมทแลบ แตกต่างกันที่เวลาที่ประมวลจนได้ค่าแรกเท่านั้น ซึ่งก็เป็นไปตามผลของเลขคณิตกระจายที่เลือกใช้ ดังนั้นจากการจำลองการทำงานจึงแสดงให้เห็นว่าตัวแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนานจะมีความเร็วในการประมวลผลที่สูงกว่าแบบอนุกรม

เมื่อนำมาสร้างจริงปัญหาที่เกิดขึ้น คือ ลอจิกเกตที่ใช้มีจำนวนมาก เนื่องจากการใช้โครงสร้างเลขคณิตกระจายแบบขนาน จะมีการทำงานที่ซับซ้อนกว่าแบบอนุกรม จึงต้องใช้ลอจิกเกตมากตามไปด้วย จากข้อจำกัดนี้ทำให้สามารถสร้างตัวแปลงเวฟเลตได้เพียงระดับเดียวเท่านั้น โดยเป็นการแปลงเวฟเลต 1 ระดับและการแปลงกลับเวฟเลต 1 ระดับ ต่างจากที่ออกแบบไว้ให้ทำงาน 3 ระดับ

การทดสอบการทำงานของตัวแปลงเวฟเลต ซึ่งใช้สัญญาณอินพุต 3 แบบ คือ สัญญาณรูปไซน์ สัญญาณรูปสี่เหลี่ยม และสัญญาณเสียง พบว่าสัญญาณเอาท์พุตมีความผิดพลาดของบิตข้อมูลจากการรับส่ง เนื่องจากว่าอุปกรณ์ที่ใช้แปลงความถี่ของอุปกรณ์ FPGA เป็นบอดเรทไม่สามารถแปลงได้จังหวะพอดี เมื่อเปรียบเทียบค่าที่ผ่านตัวแปลงเวฟเลตกับค่าจากการรับส่งข้อมูลอย่างเดียวพบว่า มีค่าความผิดพลาดของบิตที่ใกล้เคียงกัน จึงสรุปได้ว่าค่าความผิดพลาดที่เกิดจากตัวแปลงเวฟเลตนั้นมีน้อยมาก เป็นไปตามผลที่ได้จากการจำลองการทำงาน ส่วนค่าความผิดพลาดที่เกิดขึ้นมาจากการรับส่งข้อมูลผ่านพอร์ทอนุกรม ซึ่งจะแก้ไขได้โดยการเลือกใช้อุปกรณ์ FPGA ที่มีความถี่ที่สามารถหารได้บอดเรท 9600 บิตต่อวินาทีได้ลงตัวพอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ชำนาญ ปัญญาใส และ วัชรากร หนูทอง. **ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล.** กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2547.

[2] Al-Haj, A. **Fast Discrete Wavelet Transformation Using FPGAs and Distributed Arithmetic.** International Journal of Applied Science and Engineering, 1, 2: 241-247, 2004.

[3] Al-Haj, A. **An FPGA-Based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform.** Informatica, 29, 2: 160-171, 2005.

[4] Daubechies, I. and Sweldens, W. **Factoring Wavelet Transforms Into Lifting Steps.** Journal of Fourier Analysis and Applications, 4, 3: 247-269, 1998.

[5] Misiti, M. et al. **Wavelets Toolbox User's Guide.** Natrick MA: The Math Works, 1996.

[6] Mitra, S. **Digital Signal Processing: AComputer Based Approach.** 3rd . n.p.: McGraw Hill, 2005.

[7] Sidney, C., Gopinath, R. and Guo, H. **Introduction to Wavelets and Wavelet Transforms.** New Hersey: Prentice Hall, 1998.

[8] Strang, G. and Nguyen, T. **Wavelets and Filter Banks,** Wellesley MA: Cambridge, 1996.

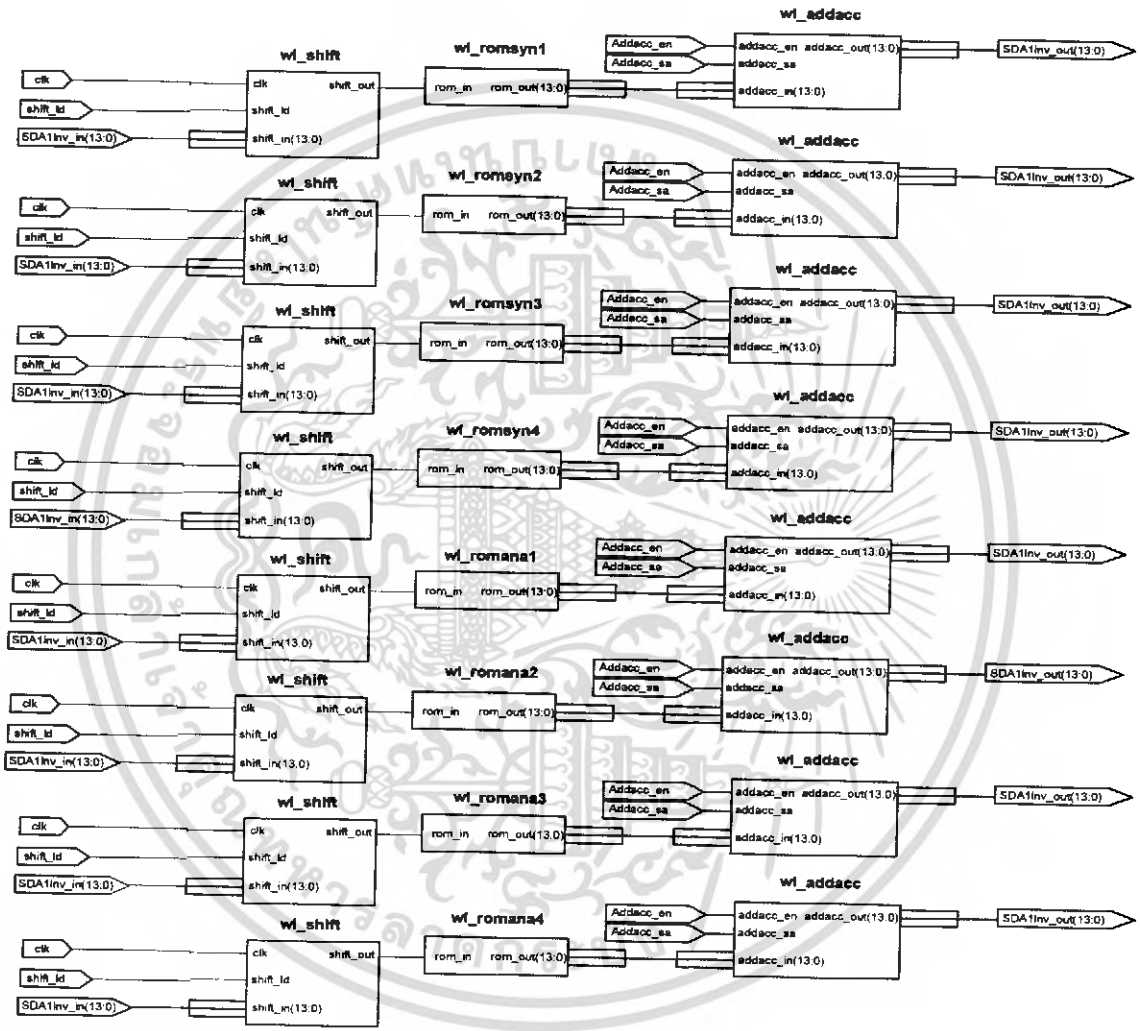
[9] White, S. **Application of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review.** IEEE ASSP Magazine, 6: 4-19, 1989.

[10] Young, R. **Wavelet Theory and Its Application Vol. 2.** Kluwer: Academics, 1992.

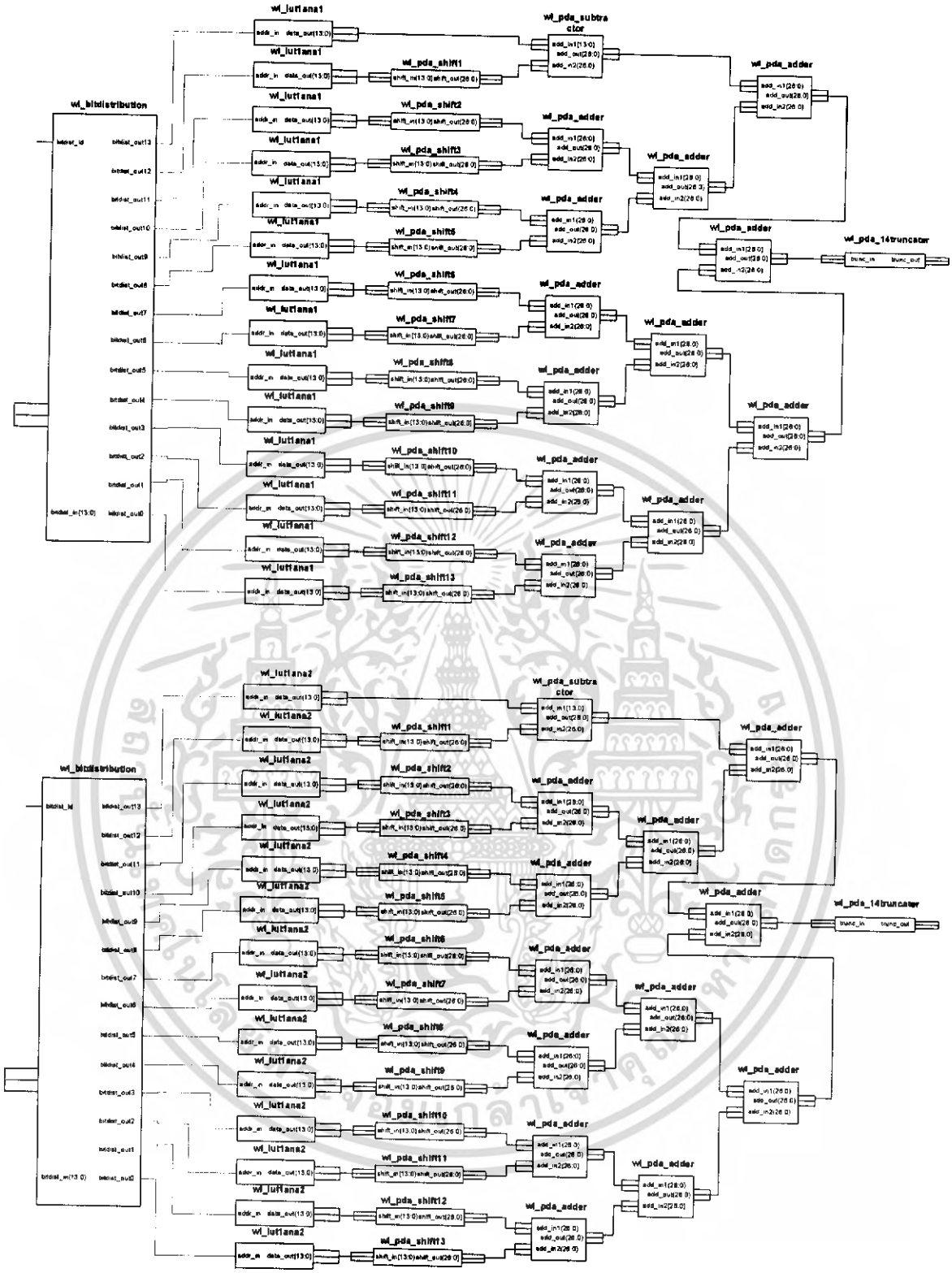
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

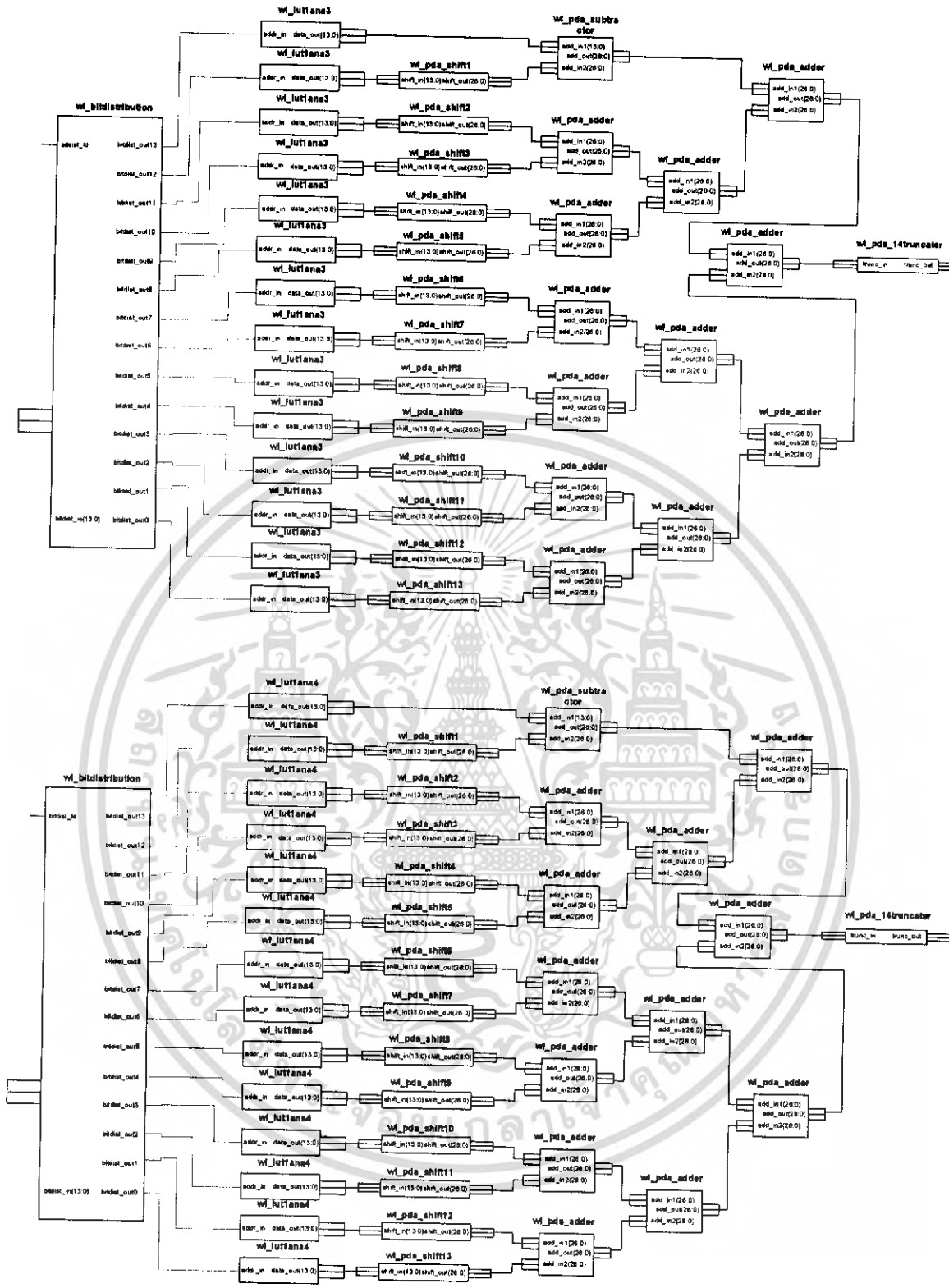


ก. โครงสร้างเลขคณิตกระจายแบบอนุกรม



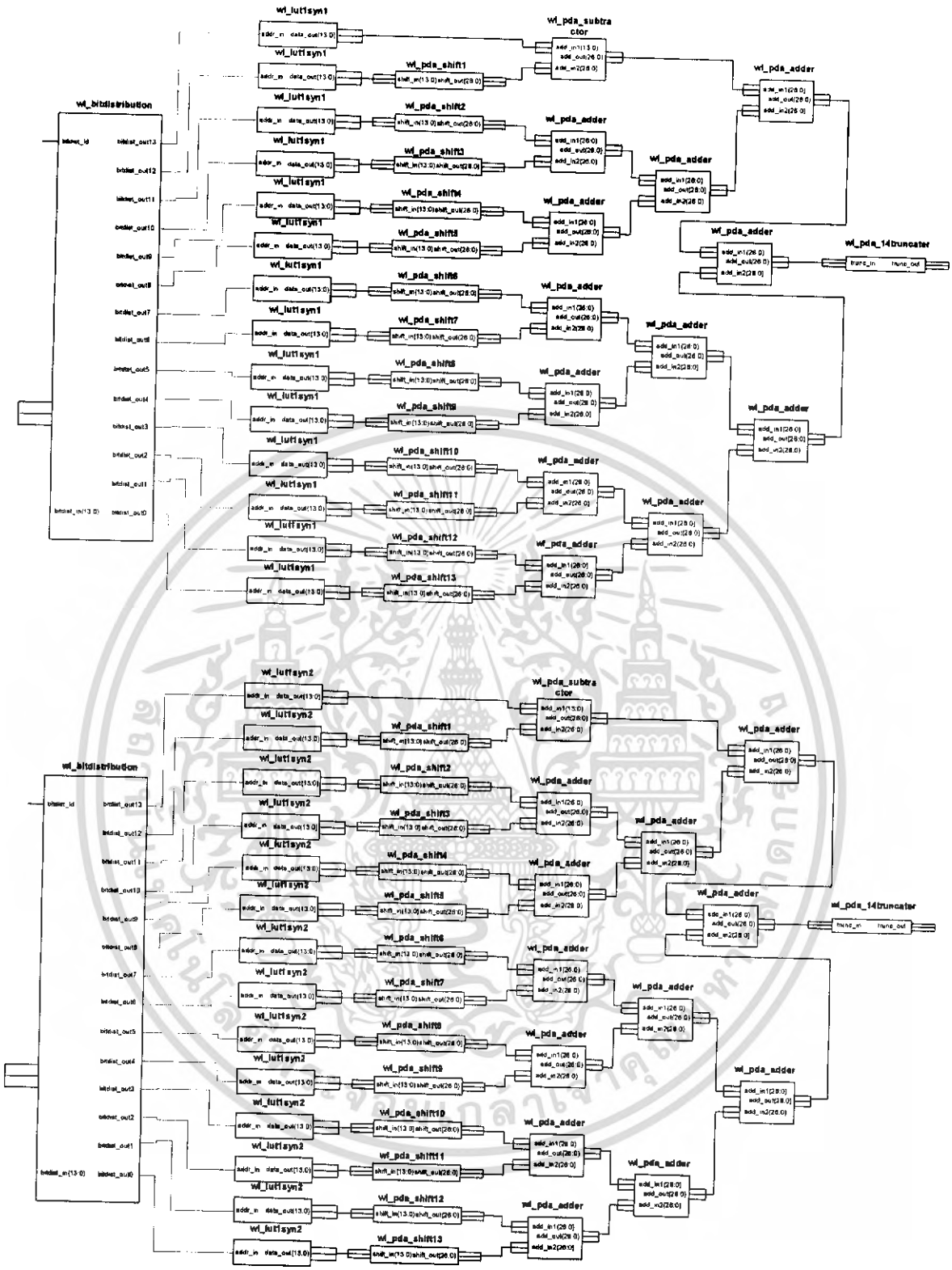
ข. โครงสร้างเลขคณิตกระจายแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



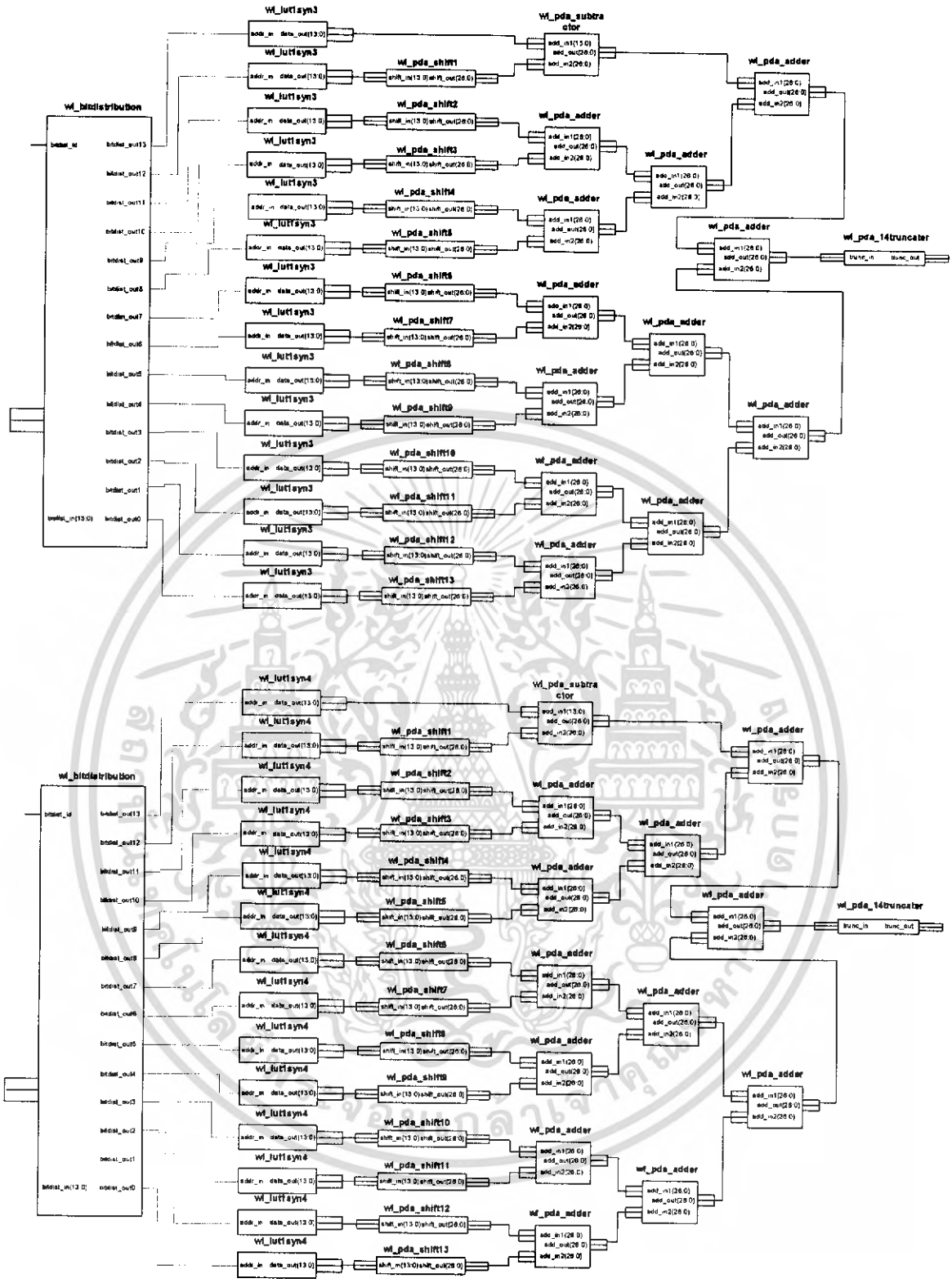
ข. โครงสร้างเลขคณิตกระจายแบบขนาน (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



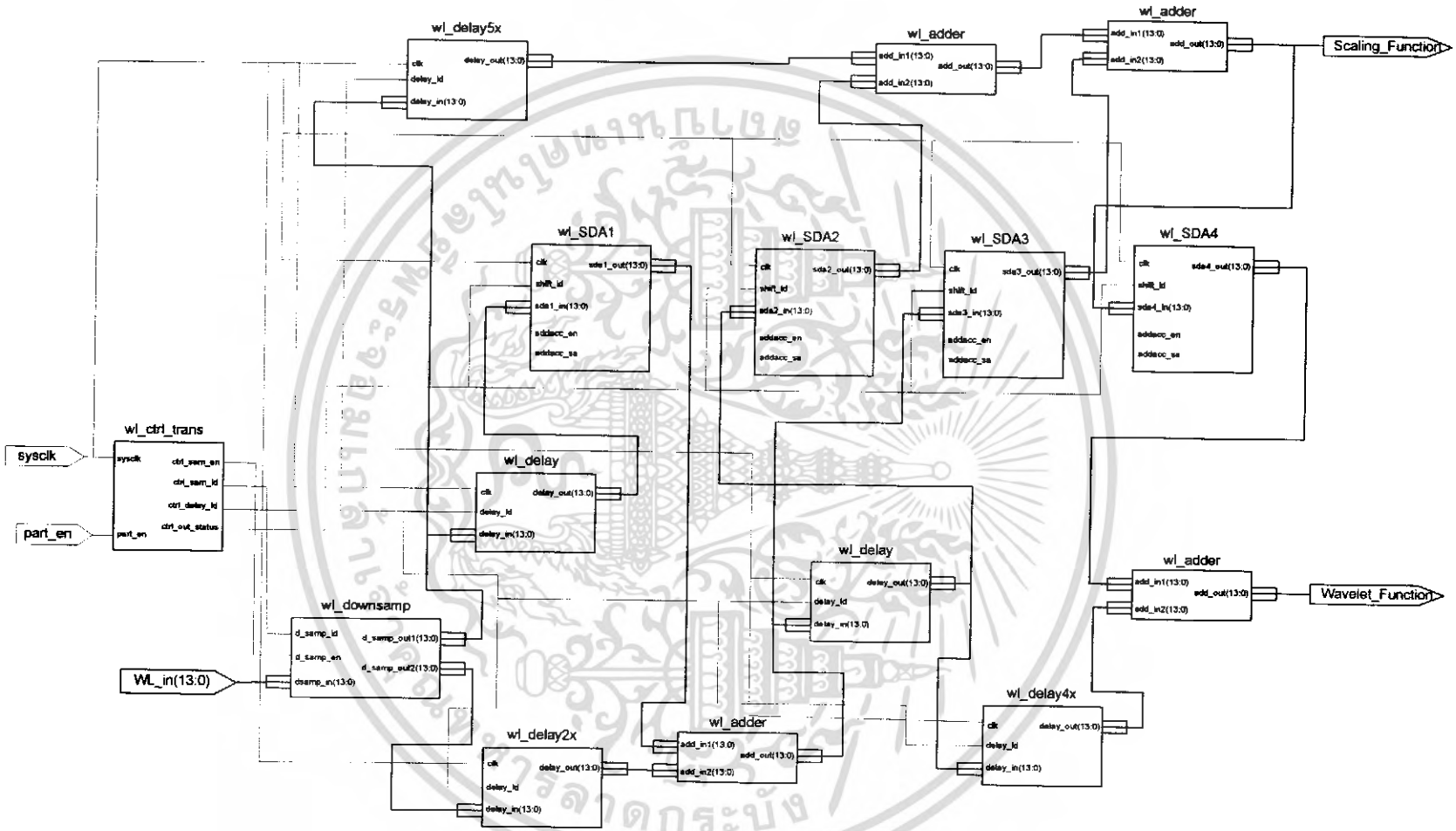
ข. โครงสร้างเลขคณิตกระจายแบบขนาน (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

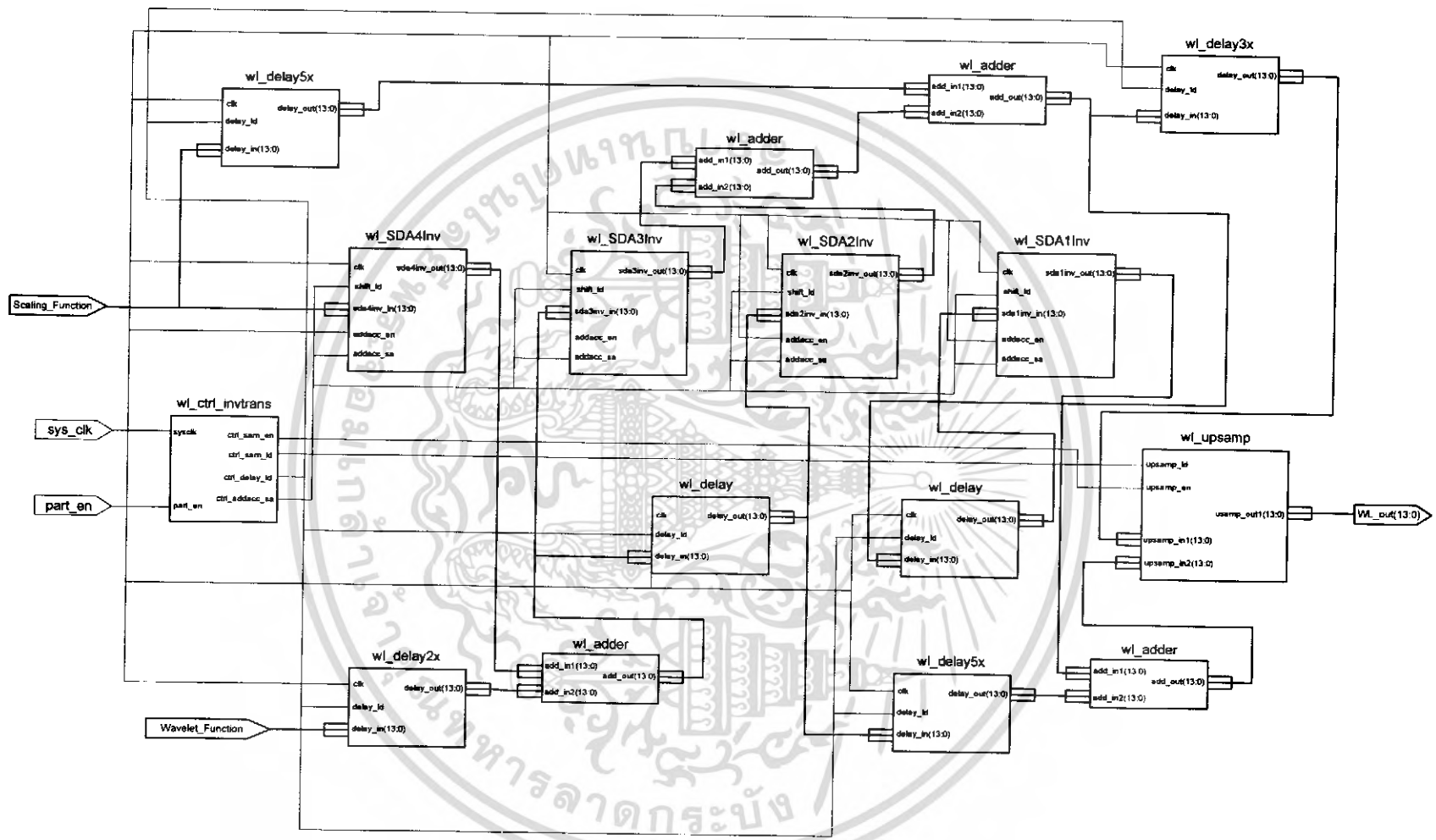


ข. โครงสร้างเลขคณิตกระจายแบบขนาน (ต่อ)

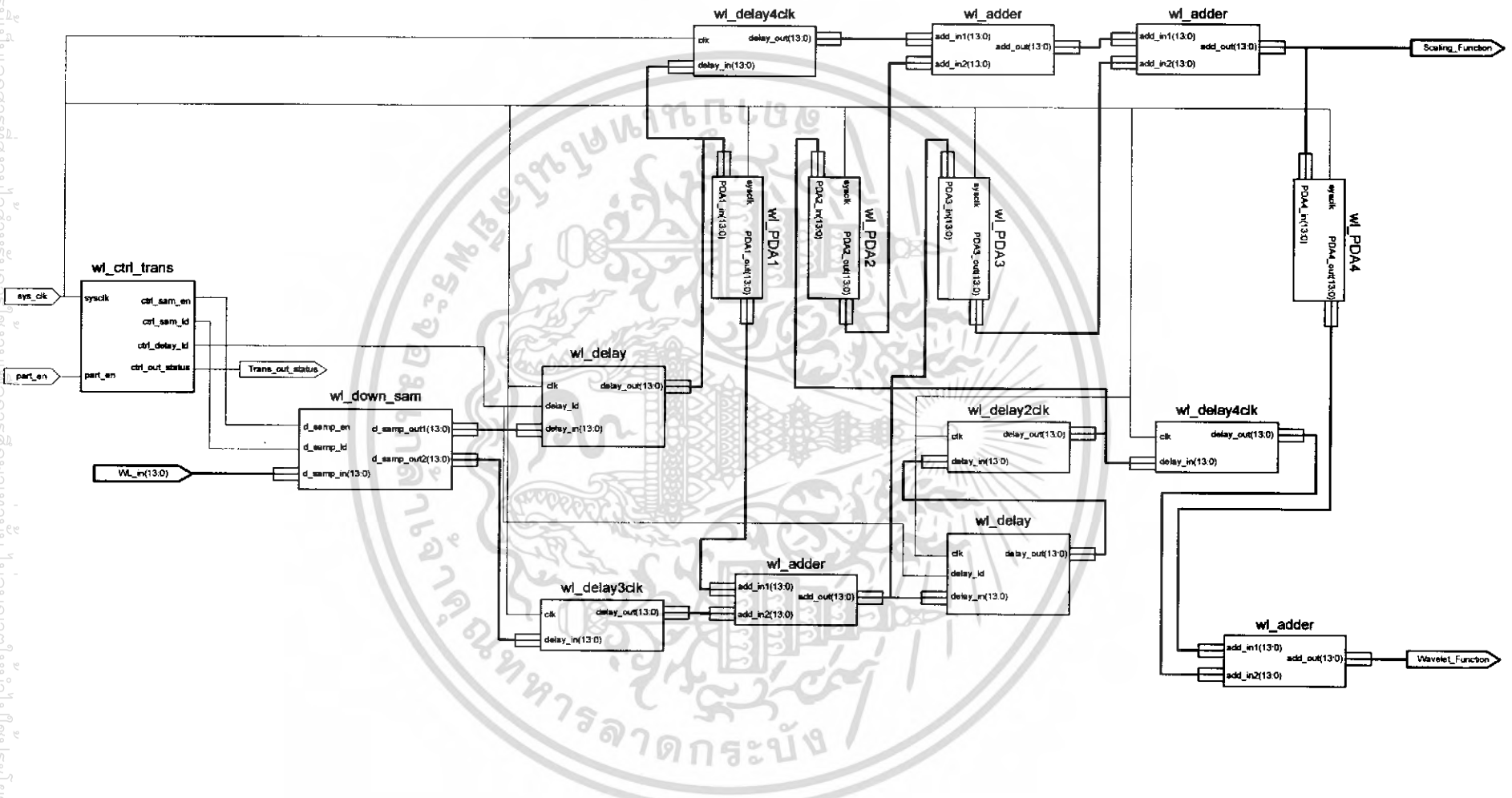
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



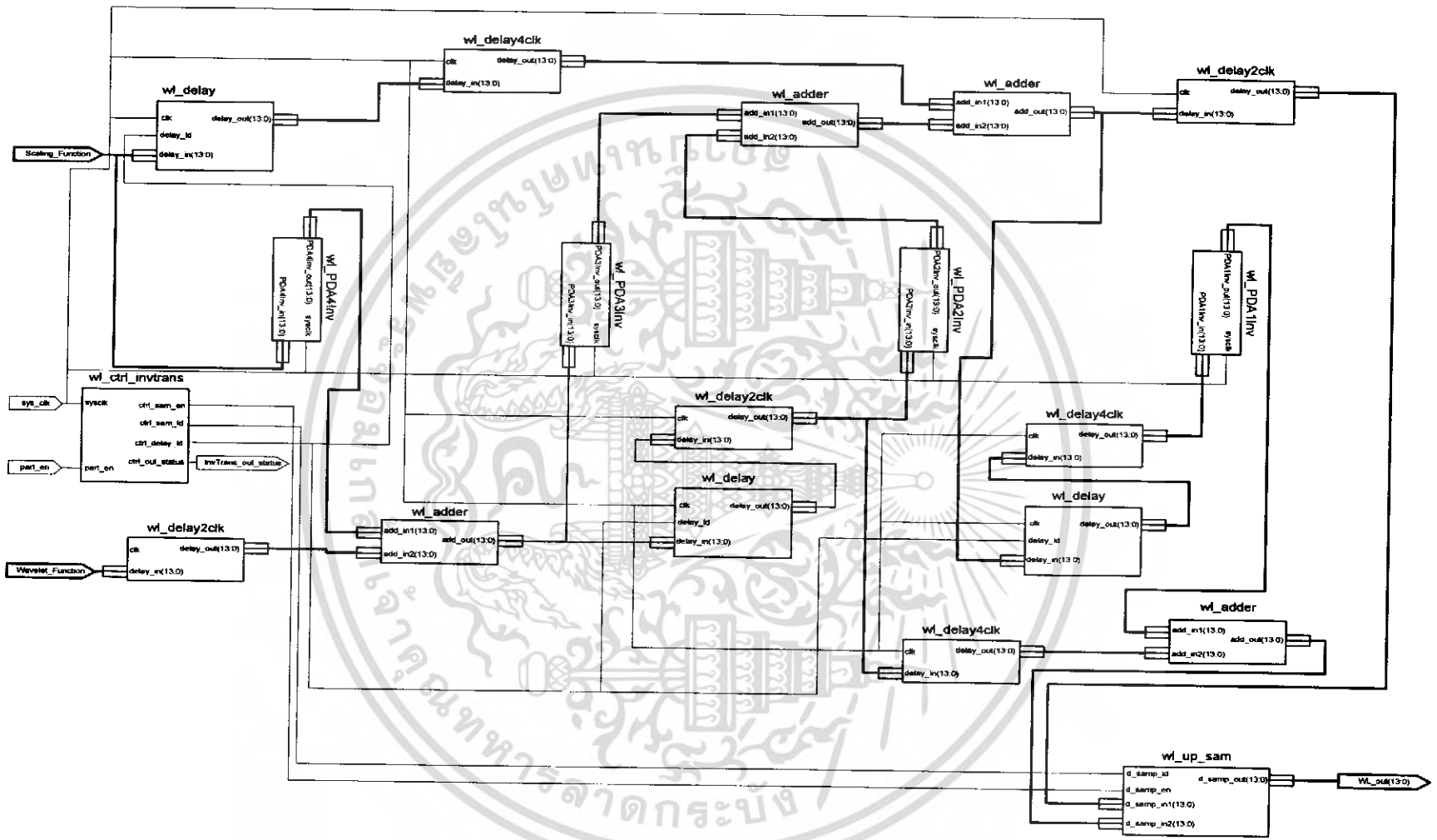
ค. วงจรแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (Wavelet Transform with Serial Distributed Arithmetic)



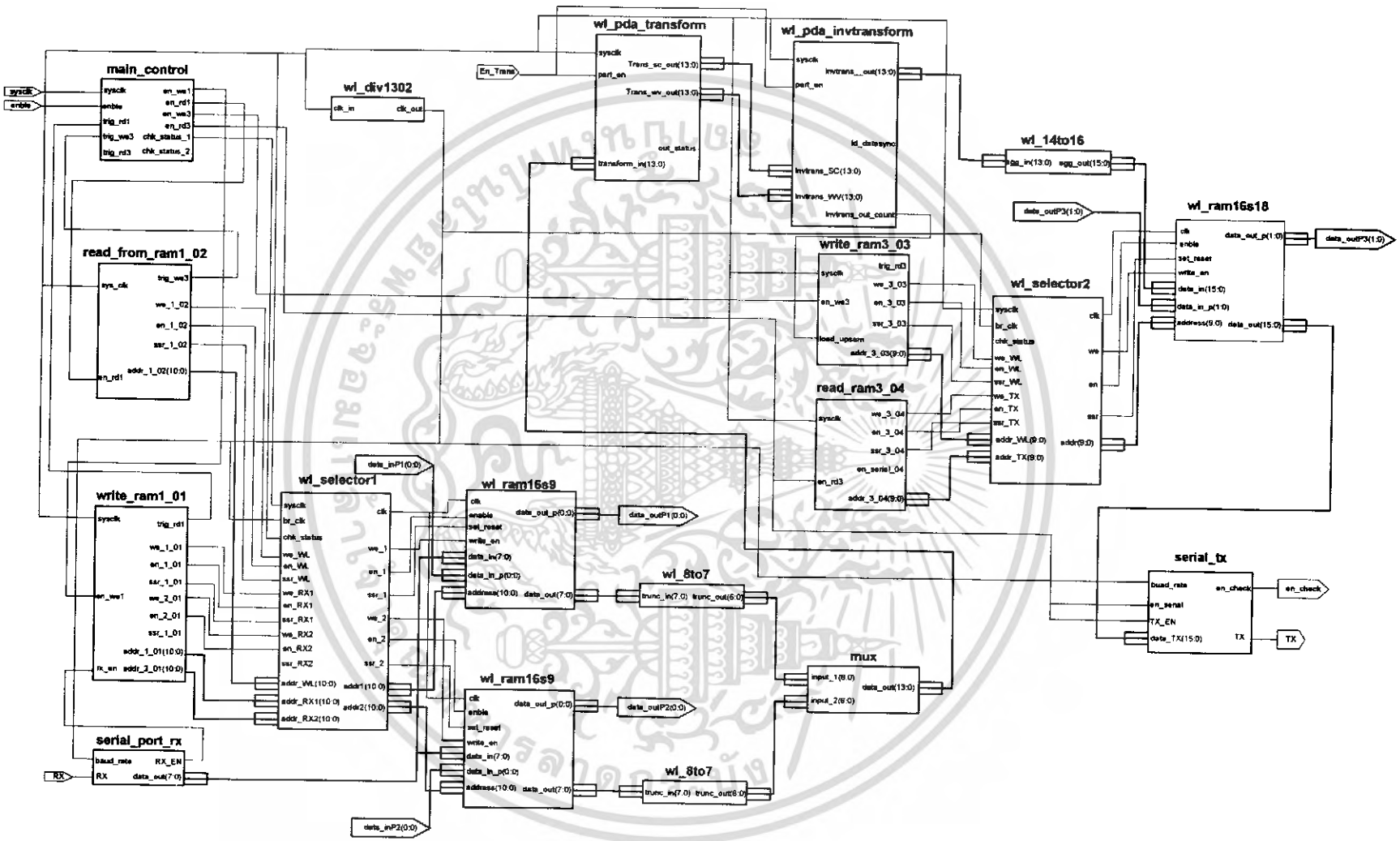
ฆ. วงจรแปลงกลับเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบอนุกรม (Inverse Wavelet Transform with Serial Distributed Arithmetic)



ง. วงจรแปลงเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (Wavelet Transform with Parallel Distributed Arithmetic)



จ. วงจรแปลงกลับเวฟเลตที่ใช้โครงสร้างเลขคณิตกระจายแบบขนาน (Inverse Wavelet Transform with Parallel Distributed Arithmetic)



ฉ. วงจรตัวแปลงเวฟเลต (Wavelet Transform)