

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน
แบบผ่านตลอด
DESIGN OF ALLPASS BASED VARIABLE FRACTIONAL DELAY
DIGITAL FILTER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน
แบบผ่านตลอด
DESIGN OF ALLPASS BASED VARIABLE FRACTIONAL DELAY
DIGITAL FILTER

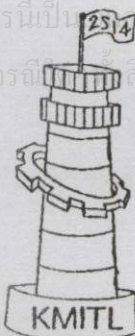


โดย

นายวรากร สุภารินทร์
นางสาววราภรณ์ ศรีสมใจ
นางสาวสุ เข้มวิชัย

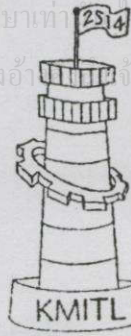
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ผู้ที่นำเอกสารนี้ไปใช้อื่นๆ นอกเหนือจากนี้ถือว่าผิดกฎหมายและต้องรับผิดชอบ



ผ่านการตรวจชิ้นงานแล้ว
(*[Signature]*)
กรรมการผู้ตรวจชิ้นงาน
14/5/58

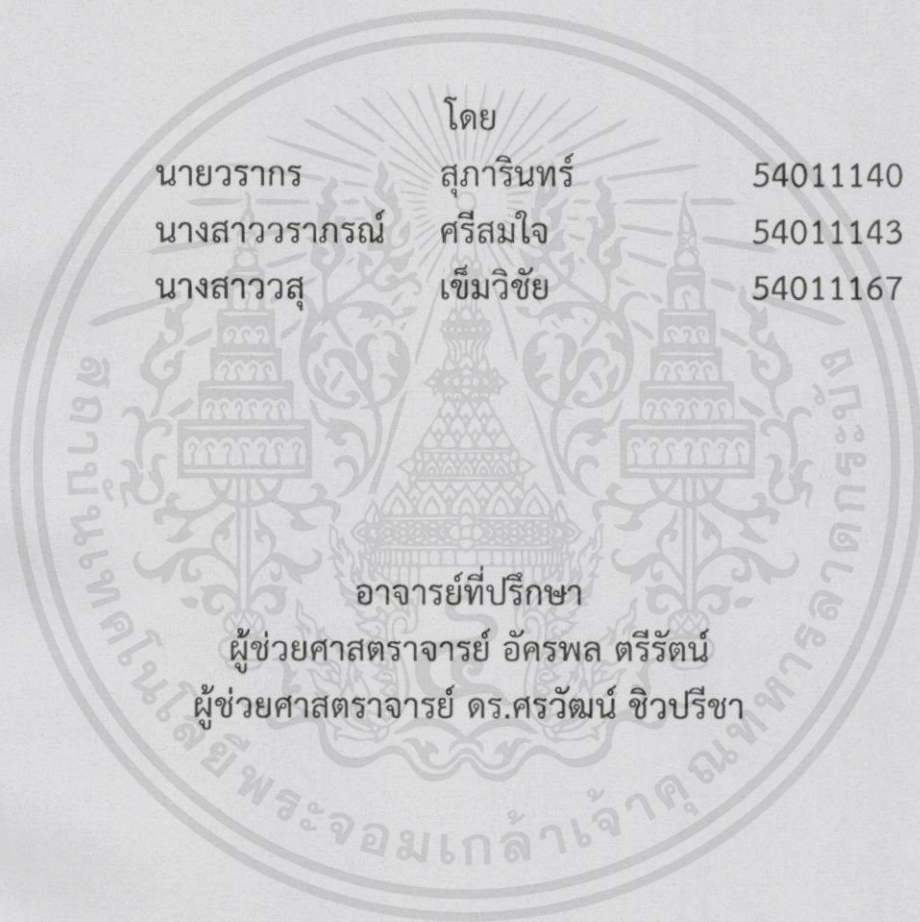
วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจรูปเล่มแล้ว
(*[Signature]*)
อาจารย์ที่ปรึกษา
15/5/58

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน
แบบผ่านตลอด
DESIGN OF ALLPASS BASED VARIABLE FRACTIONAL DELAY
DIGITAL FILTER



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก่อนนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอด

DESIGN OF ALLPASS BASED VARIABLE FRACTIONAL DELAY DIGITAL FILTER

ผู้จัดทำ

- | | | |
|-------------------|------------|----------|
| 1. นายวรารกร | สุภารินทร์ | 54011140 |
| 2. นางสาววรารภรณ์ | ศรีสมใจ | 54011143 |
| 3. นางสาวสุ | เข้มวิชัย | 54011167 |

.....
อธิการบดี ตรีรัตน์
..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ อัครพล ตรีรัตน์)

.....
..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.ศรวรัตน์ ชิวปรีชา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ อัครพล ตริรัตน์ และอาจารย์ที่ปรึกษาร่วม ผู้ช่วยศาสตราจารย์ ดร.ศรววัฒน์ ชิวปรีชา ที่กรุณาให้คำปรึกษา คำชี้แนะ และปรับปรุงแก้ไขจนปริญญาานิพนธ์เล่มนี้ประสบความสำเร็จด้วยความเอาใจใส่ยิ่งตลอดมา

ผู้จัดทำปริญญาานิพนธ์ขอกราบขอบพระคุณบิดามารดา อันเป็นที่รักและเคารพยิ่งที่ให้การสนับสนุนและให้กำลังใจตลอดมา รวมทั้งคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ ซึ่งเป็นประโยชน์ต่อการทำปริญญาานิพนธ์นี้

ในการจัดทำได้ประสบปัญหาหลายอย่าง นอกจากจะแก้ปัญหาด้วยตนเองและจากคำชี้แนะของอาจารย์ที่ปรึกษาทั้งสองท่านแล้ว ต้องขอขอบคุณเพิ่มเติมสำหรับความช่วยเหลือที่ได้รับจาก นายพีรเดช มีเจริญ ทั้งในเรื่องการเขียนโปรแกรมและการจัดทำเล่มปริญญาานิพนธ์

ประโยชน์และคุณค่าที่พึงมีจากปริญญาานิพนธ์นี้ ผู้จัดทำขอมอบให้แก่ผู้มีส่วนร่วมทุกท่านตลอดทั้งผู้ที่ต้องการศึกษาในเรื่องที่ทางผู้จัดทำได้ศึกษาต่อไป

วรากร สุภารินทร์
วรากรณ์ ศรีสมใจ
วสุ เข็มวิชัย
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็น
 เศษส่วนแบบผ่านตลอด

DESIGN OF ALLPASS BASED VARIABLE FRACTIONAL
 DELAY DIGITAL FILTER

โดย นายวรกร สุภารินทร์ 54011140
 นางสาวราภรณ์ ศรีสมใจ 54011143
 นางสาวสุ เข็มวิชัย 54011167

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ อัครพล ตรีรัตน์
 ผู้ช่วยศาสตราจารย์ ดร.ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

ปริญญานิพนธ์เล่มนี้เป็นการศึกษาและออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโดยใช้วงจรกรองสัญญาณดิจิทัลแบบผลตอบสนองอิมพัลส์ไม่จำกัด (IIR Filter) ตามทฤษฎีของไธราน (Thiran) เพื่อประยุกต์ใช้เป็นตัวแปลงอัตราการซีกตัวอย่าง (Sample-rate Converter) ให้ระบบที่มีอัตราการซีกตัวอย่างต่างกันสามารถทำงานร่วมกันได้อย่างเหมาะสม ทำการออกแบบและจำลองระบบผ่านโปรแกรม Modelsim SE 6.5 แล้วทดสอบการทำงานของระบบบนฮาร์ดแวร์ผ่านอุปกรณ์ FPGA (Field-Programmable Gate Array)

ABSTRACT

The purpose of this project is to design efficient structure for Sample-Rate Converter from 44.1 kHz 48 kHz using fractional delay all-pass filter. For efficient conversion, we use an infinite-impulse response (IIR) fractional delay based on Thiran-based IIR all-pass filter. The designed hardware is implemented via FPGA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น "ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้"

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 ทฤษฎีพื้นฐานของวงจรกรองสัญญาณดิจิทัล	3
2.2 วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอด (Allpass based fractional delay digital filter)	7
2.3 การออกแบบวงจรดิจิทัลด้วยภาษา VHDL	11
2.4 Field-Programmable Gate Array (FPGA)	21
2.5 การประยุกต์ใช้งาน	23
บทที่ 3 การออกแบบและการจัดทำปริณิงานพิมพ์	28
3.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบ ผ่านตลอดโรราน	28
3.2 การประยุกต์ใช้งาน Signal Interpolation	32
3.3 การทำ Oversampling D/A	35
3.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบ ผ่านตลอดโรรานบนฮาร์ดแวร์	37
บทที่ 4 ผลการทดลอง	42
4.1 คุณสมบัติของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบ ผ่านตลอดโรราน	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2 การทำ Oversampling D/A	44
4.3 การจำลองการทำงานทางซอฟต์แวร์ในการทำ Oversampling D/A	47
4.4 ผลการจำลองการทำงานทางฮาร์ดแวร์ในการทำ Oversampling D/A	52
4.5 ผลการทำงานจริงทางฮาร์ดแวร์ในการทำ Oversampling D/A	56
บทที่ 5 สรุปผลและข้อเสนอแนะ	64
5.1 สรุปผล	64
5.2 ข้อเสนอแนะ	65
บรรณานุกรม	66
ภาคผนวก ผลการทดลองปริญญาานิพนธ์ เทอมที่ 1	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	บล็อกไดอะแกรมของวงจรรองสัญญาณดิจิทัล	5
2.2	โครงสร้างของวงจรรองสัญญาณแบบผ่านตลอดชนิด IS	10
2.3	ชุดของตัวแบบปรับค่าได้หลังจากการประมาณค่าแบบ Taylor อันดับ 2	10
2.4	ชุดของตัวแบบปรับค่าได้หลังจากการประมาณค่าแบบ Taylor อันดับ 3	11
2.5	โครงสร้างของชุดตัวคูณ (โครงสร้าง IS จากวงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดอันดับ 2)	11
2.6	ขั้นตอนการออกแบบจากบนลงล่าง	12
2.7	โครงสร้างโดยทั่วไปของหน่วยการออกแบบแอนทีตี	14
2.8	โครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม	15
2.9	โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต	16
2.10	โครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต	17
2.11	โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	17
2.12	ลักษณะของตัว FPGA	18
2.13	แสดงการควบคุมเวลาของสัญญาณการซีกตัวอย่างของส่วนประมวลผลสัญญาณอนาล็อก	24
2.14	แสดงการควบคุมเวลาของสัญญาณการซีกตัวอย่างของส่วนประมวลผลสัญญาณเชิงเลข	24
2.15	สัญญาณอินพุตที่มีอัตราการซีกตัวอย่างที่ความถี่ f_{s1} แปลงไปสู่สัญญาณเอาต์พุตที่มีอัตราการซีกตัวอย่างที่ความถี่ f_{s2}	25
2.16	แผนผังของระบบวงจรแปลงอัตราการซีกตัวอย่าง	26
2.17	ผลการเปรียบเทียบของสัญญาณคลื่นไซน์กับสัญญาณคลื่นไซน์ที่ผ่านการทำ Oversampling D/A 2 เท่า	27
3.1	บล็อกไดอะแกรมการทำงานของวงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไรราน	28
3.2	ผลของการทำ Double Interpolation เทียบกับสัญญาณอินพุตเดิม	33
3.3	ผลของการทำ Triple Interpolation เทียบกับสัญญาณอินพุตเดิม	34
3.4	ผลของการทำ Four-time Interpolation เทียบกับสัญญาณอินพุตเดิม	34
3.5	บล็อกไดอะแกรมการทำงานของการทำงาน Oversampling D/A 2 เท่า	35
3.6	บล็อกไดอะแกรมการทำงานของการทำงาน Oversampling D/A 3 เท่า	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.7	บล็อกไดอะแกรมการทำงานของการทำงาน Oversampling D/A 4 เท่า	36
3.8	อุปกรณ์ในการจัดเก็บผลการทดลองบนฮาร์ดแวร์	37
3.9	ผลของการสังเคราะห์ห้วงจรบวก	38
3.10	ผลของการสังเคราะห์ห้วงจรลบ	38
3.11	ผลของการสังเคราะห์ห้วงจรคูณ	39
3.12	ผลของการสังเคราะห์ห้วงจรหน่วงเวลา	39
3.13	ผลของการสังเคราะห์ห้วงจร PIPO	40
3.14	ผลของการสังเคราะห์ห้วงจร look up table	40
3.15	ผลของการสังเคราะห์ห้วงจร Counter	41
3.16	ผลของการสังเคราะห์ห้วงจร Multiplexer	41
4.1	ผลตอบสนองทางเฟสของตัวกรองอันดับที่ 2 ที่ได้จากโปรแกรมแมตแล็บ	43
4.2	ผลตอบสนองทางเฟสของตัวกรองอันดับที่ 2 ที่ได้จากการจำลองฮาร์ดแวร์	44
4.3	บล็อกไดอะแกรมปัญหาของการทดลองที่เกิดขึ้น	45
4.4	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 1 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3	45
4.5	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3	46
4.6	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 5 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3	46
4.7	บล็อกไดอะแกรมการทำงานของ Oversampling D/A	47
4.8	ผลของการทำ Oversampling D/A 2 เท่า	48
4.9	ผลของการทำ Oversampling D/A 3 เท่า	48
4.10	ผลของการทำ Oversampling D/A 4 เท่า	49
4.11	ผลของการทำ Oversampling D/A 2 เท่า	49
4.12	ผลของการทำ Oversampling D/A 3 เท่า	50
4.13	ผลของการทำ Oversampling D/A 4 เท่า	50
4.14	ผลของการทำ Oversampling D/A 2 เท่า	51
4.15	ผลของการทำ Oversampling D/A 3 เท่า	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกร ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.16	ผลของการทำ Oversampling D/A 4 เท่า	52
4.17	ผลของการทำ Oversampling D/A 2 เท่า	53
4.18	ผลของการทำ Oversampling D/A 3 เท่า	53
4.19	ผลของการทำ Oversampling D/A 4 เท่า	53
4.20	ผลของการทำ Oversampling D/A 2 เท่า	54
4.21	ผลของการทำ Oversampling D/A 3 เท่า	54
4.22	ผลของการทำ Oversampling D/A 4 เท่า	54
4.23	ผลของการทำ Oversampling D/A 2 เท่า	55
4.24	ผลของการทำ Oversampling D/A 3 เท่า	55
4.25	ผลของการทำ Oversampling D/A 4 เท่า	55
4.26	ผลของการแปลงอัตราการซีกตัวอย่าง	56
4.27	ผลเปรียบเทียบสัญญาณเอาต์พุตที่ความถี่ 1 kHz	57
4.28	ผลเปรียบเทียบสัญญาณเอาต์พุตที่ความถี่ 3 kHz	58
4.29	ผลเปรียบเทียบสัญญาณเอาต์พุตที่ความถี่ 5 kHz	59
4.30	ผลการทำงานของแปลงอัตราการซีกตัวอย่างที่ความถี่ 1 kHz	60
4.31	ผลการทำงานของแปลงอัตราการซีกตัวอย่างที่ความถี่ 3 kHz	60
4.32	ผลการทำงานของแปลงอัตราการซีกตัวอย่างที่ความถี่ 5 kHz	61
4.33	ผลของการทำ Oversampling D/A 2 เท่า	61
4.34	ผลของการทำ Oversampling D/A 3 เท่า	62
4.35	ผลของการทำ Oversampling D/A 4 เท่า	62
4.36	(ก) อุปกรณ์การจำลองการทำงานจริงทางฮาร์ดแวร์	63
	(ข) บอร์ด FPGA , วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC) และวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (DAC)	
	(ค) วงจรไมโครโฟนและวงจรออฟเซต	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
3.1	ค่าสัมประสิทธิ์ a และ b ที่ใช้สำหรับแต่ละค่า D	29
3.2	ค่าความหน่วงที่ใช้สำหรับการแปลงอัตราการสุ่มตัวอย่าง	31
3.2	ค่าความหน่วงที่ใช้สำหรับการแปลงอัตราการสุ่มตัวอย่าง (ต่อ)	32
3.3	ความสัมพันธ์ระหว่างการทำ Interpolation และจำนวนค่าการหน่วงที่ใช้	33



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันวงจรกรองสัญญาณดิจิทัล (Digital Filter) ได้ถูกนำมาใช้งานเป็นจำนวนมาก โดยเฉพาะทางด้านที่เกี่ยวข้องกับการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing : DSP) วงจรกรองสัญญาณดิจิทัลนั้นไม่ได้มีคุณสมบัติเพื่อกรองสัญญาณเพียงอย่างเดียว แต่ยังสามารถประยุกต์ใช้ในการประมวลผลสัญญาณเชิงเลขอื่น ๆ ได้อีกมากมาย

และเนื่องจากในระบบประมวลผลสัญญาณดิจิทัลสามารถทำการห้วงสัญญาณได้อย่างน้อยที่สุดคือการห้วงหนึ่งหน่วย และสามารถห้วงได้เฉพาะค่าความห้วงที่เป็นเลขจำนวนเต็มเท่านั้น ซึ่งในลักษณะงานบางประเภทต้องการการห้วงสัญญาณที่น้อยกว่าหนึ่งหน่วย หรือการห้วงแบบเป็นเศษส่วน ซึ่งในระบบประมวลผลสัญญาณดิจิทัลลักษณะทั่วไปไม่สามารถทำได้

ดังนั้นจึงมีการนำวงจรกรองสัญญาณปรับค่าความห้วงเป็นเศษส่วนแบบผ่านตลอดไธราน (Thiran) มาแก้ไขปัญหาในระบบประมวลผลสัญญาณดิจิทัลให้สามารถทำการห้วงสัญญาณแบบเป็นเศษส่วนได้

นอกจากนี้วงจรกรองสัญญาณปรับค่าความห้วงเป็นเศษส่วนแบบผ่านตลอดไธรานยังสามารถนำมาประยุกต์ใช้งานได้อย่างมาก โดยในปริิณยานิพนธ์เล่มนี้จะแสดงการประยุกต์ใช้งานทางด้านการแปลงอัตราการซิกตัวอย่าง (Sampling Rate Converter) และการทำ Oversampling D/A เพื่อเพิ่มความละเอียดของสัญญาณให้สูงขึ้นพร้อมกับ อัตราการซิกตัวอย่างที่เพิ่มขึ้น

1.2 วัตถุประสงค์

- 1) สามารถออกแบบวงจรกรองสัญญาณที่มีความห้วงเป็นเศษส่วนโดยใช้วงจรกรองสัญญาณแบบผ่านตลอด (Allpass Filter) ได้
- 2) สามารถทำการสร้างวงจรกรองสัญญาณความห้วงเป็นเศษส่วนแบบผ่านตลอด (Fractional Delay Allpass Filter) ด้วยอุปกรณ์ FPGA ได้
- 3) สามารถนำวงจรกรองสัญญาณที่ออกแบบไปทดสอบและประยุกต์ใช้งานจริง

1.3 ขอบเขตของปริิณยานิพนธ์

- 1) ทำการศึกษาวิธีการออกแบบวงจรกรองสัญญาณปรับค่าความห้วงเป็นเศษส่วนแบบผ่านตลอด โดยที่มีคุณสมบัติของผลตอบสนองความห้วงเฟสแบบราบเรียบที่สุด (Maximally-flat Phase Delay Response) ในช่วงความถี่ที่ต้องการ
- 2) ทำการจำลองการทำงานและทดสอบคุณสมบัติที่ออกแบบด้วยโปรแกรมแมตแล็บ (MATLAB)

3) ทำการสร้างระบบที่ออกแบบโดยอาศัยอุปกรณ์ FPGA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ทฤษฎีพื้นฐานของวงจรกรองสัญญาณดิจิทัล

วงจรกรองสัญญาณดิจิทัล เป็นกระบวนการที่ใช้ในการดัดแปลงสเปกตรัมของสัญญาณให้มีลักษณะสเปกตรัมเป็นไปตามการกำหนดที่ต้องการ ซึ่งเป็นการเพิ่มหรือลดทอนขนาดของสเปกตรัมในช่วงความถี่ที่กำหนดอาจเป็นการตัดสัญญาณในช่วงความถี่ที่ไม่ต้องการออกหรือทำการแยกสัญญาณในช่วงความถี่ที่ต้องการออกมาใช้งาน ซึ่งในการวิเคราะห์และสังเคราะห์วงจรนั้นต้องใช้เครื่องมือพื้นฐานทางคณิตศาสตร์เข้าช่วย กระบวนการเหล่านี้มีจุดประสงค์หลักเพื่อ

- 1) ลดความผิดเพี้ยนของสัญญาณเอาต์พุต (Distortion)
- 2) กำจัดสัญญาณรบกวน (Noise signal)
- 3) ต้องการแยกสัญญาณออกจากสัญญาณผสมของสัญญาณที่ส่งมาพร้อมกันในสายส่งเดียวกัน
- 4) ใช้ในการแยกแยะสัญญาณในโดเมนเวลาออกเป็นองค์ประกอบของสัญญาณในโดเมนความถี่
- 5) ใช้ในการดีมอดูเลตสัญญาณ (Demodulation)

วงจรกรองสัญญาณดิจิทัลถือเป็นสิ่งที่สำคัญมากสำหรับการประมวลผลสัญญาณดิจิทัล (Digital signal processing: DSP) เนื่องจากการทำงานของวงจรกรองสัญญาณดิจิทัลสามารถประยุกต์ใช้งานในลักษณะอื่นได้ที่ไม่ได้จำกัดแค่ต้องใช้ในการกรองสัญญาณเพียงอย่างเดียว เช่น data compression, biomedical signal processing, speech processing, image processing, data transmission, digital audio, telephone echo cancellation

การที่วงจรกรองสัญญาณดิจิทัลมีการพัฒนาและนำมาประยุกต์ใช้งานกันมานอกจากเหตุผลเรื่องการพัฒนาาระบบดิจิทัลที่มากขึ้นในปัจจุบันแล้ว ยังมีข้อได้เปรียบดังต่อไปนี้

- 1) สามารถออกแบบวงจรกรองสัญญาณดิจิทัลให้มีผลตอบสนองเฟสเชิงเส้น (Linear phase response) ซึ่งเป็นไปไม่ได้ถ้าออกแบบด้วยวงจรกรองสัญญาณอนาล็อก (Analog filter)
- 2) คุณสมบัติของวงจรกรองสัญญาณดิจิทัลที่ออกแบบและสร้างแล้วจะไม่เปลี่ยนแปลงไปตามการเปลี่ยนแปลงสภาพแวดล้อม เช่น การเปลี่ยนแปลงของอุณหภูมิ ความชื้น ซึ่งทำให้ ไม่มีความจำเป็นต้องทำการปรับเทียบ (Calibrate)

- 3) การประยุกต์ใช้งานเป็นวงจรกรองสัญญาณดิจิทัลแบบปรับตัวได้ (Adaptive filter) เนื่องจากผลตอบสนองทางความถี่ (Frequency response) ของวงจรกรอง

สัญญาณดิจิทัลสามารถปรับค่าได้อย่างอัตโนมัติ ถ้าวางวงจรของสัญญาณดิจิทัลถูกสร้างโดยใช้ programmable processor

4) ความมีเสถียรภาพของวงจรรองสัญญาณดิจิทัล และความแม่นยำในการกรอง

5) ด้วยความก้าวหน้าอย่างมากของเทคโนโลยีทางด้าน VLSI (Very large scale integrated circuits) ทำให้การสร้างวงจรรองสัญญาณดิจิทัลมีขนาดเล็ก กินกำลังงานไฟเลี้ยงต่ำและมีราคาที่ถูกลง

6) สามารถออกแบบให้มีค่าความละเอียดในการกรองสัญญาณได้ตามความต้องการโดยขึ้นอยู่กับความยาวของคำ (Wordlength) ของตัวเลขฐานสองที่ต้องการใช้ ไม่เหมือนกับวงจรรองสัญญาณอนาล็อกที่ขึ้นอยู่กับอุปกรณ์ที่ขายตามท้องตลาด

นอกจากประโยชน์และข้อได้เปรียบของวงจรรองสัญญาณดิจิทัลที่ทำได้ดีกว่าวงจรรองสัญญาณอนาล็อกแล้ว ในการใช้งานจริงของวงจรรองสัญญาณดิจิทัลก็จำเป็นต้องคำนึงถึงข้อเสียหลักของตัวมันเมื่อเทียบกับวงจรรองสัญญาณอนาล็อกด้วย คือ

1) ข้อจำกัดเรื่องความเร็ว (Speed limitation) เนื่องจากค่าเวลาในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to digital converter: ADC) และการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to analog converter: DAC) เป็นตัวจำกัดค่าความถี่สูงสุดที่สามารถทำการประมวลผลได้ นอกจากนั้นยังขึ้นอยู่กับความเร็วของตัวประมวลผลที่ใช้งานและจำนวนตัวกระทำทางคณิตศาสตร์ (Arithmetic operations)

2) ผลของความยาวคำจำกัด (Finite wordlength effect) จะเป็นผลกระทบจาก ADC noise ซึ่งเป็นผลจากกระบวนการควอนไทซ์ (Quantization) ของสัญญาณต่อเนื่องและ roundoff noise ที่เกิดขึ้นระหว่างการคำนวณของวงจรรองสัญญาณดิจิทัล ยิ่งอันดับของวงจรรองสัญญาณดิจิทัลแบบป้อนกลับยิ่งสูง การสะสมของ roundoff noise ก็จะมีมากขึ้น และนำไปสู่ความไม่มีเสถียรภาพของวงจรรองสัญญาณดิจิทัล

3) ใช้เวลาในการออกแบบและพัฒนาที่นาน โดยเฉพาะอย่างยิ่งสำหรับการพัฒนาฮาร์ดแวร์ซึ่งใช้เวลานานกว่ากรณีวงจรรองสัญญาณอนาล็อกแต่ด้วยเทคโนโลยีการออกแบบโดยใช้คอมพิวเตอร์ช่วย (Computer-aided design: CAD) ช่วยทำให้ออกแบบและสร้างวงจรรองสัญญาณดิจิทัลได้อย่างสะดวกสบายและรวดเร็วมากขึ้น

วงจรรองสัญญาณดิจิทัลสามารถถูกเขียนอธิบายในรูปของบล็อกไดอะแกรมได้ ดังรูปที่ 2.1 โดยสัญญาณอินพุตเป็นสัญญาณอนาล็อกจะถูกสุ่ม (Sampling) ด้วยช่วงเวลาทีค่าคงที่ค่าหนึ่งและสัญญาณที่ถูกสุ่มนี้จะถูกเปลี่ยนให้อยู่ในรูปเลขฐานสอง ซึ่งเป็นการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บล็อกไดอะแกรมของวงจรกรองสัญญาณดิจิทัล

หลังจากนั้นเลขฐานสองที่เป็นเสมือนตัวแทนของสัญญาณอนาล็อกที่เข้ามาทางอินพุต จะถูกกรองโดยวงจรกรองสัญญาณดิจิทัล การกรองในระบบดิจิทัลเป็นการคำนวณทางคณิตศาสตร์ ซึ่งอาศัยวงจรที่ใช้ในระบบคอมพิวเตอร์ ได้แก่ ตัวบวก (Adder), ตัวคูณ (Multiplier), รีจิสเตอร์ (Shift register) และอุปกรณ์หน่วยความจำ (Memory devices) ต่อมาค่าเอาต์พุตที่ได้จากวงจรกรองสัญญาณดิจิทัลนี้จะถูกเปลี่ยนกลับเป็นสัญญาณอนาล็อก ซึ่งจะเป็นสัญญาณเอาต์พุตที่นำไปใช้งานต่อไป

วงจรกรองสัญญาณดิจิทัลสามารถแบ่งได้เป็น 2 ประเภท พิจารณาตามลักษณะของผลตอบสนองอิมพัลส์ ได้แก่ วงจรกรองสัญญาณดิจิทัลแบบผลตอบสนองอิมพัลส์ไม่จำกัด (Infinite impulse response filter: IIR filter) เป็นตัวกรองที่มีการป้อนกลับ และวงจรกรองสัญญาณดิจิทัลแบบผลตอบสนองอิมพัลส์จำกัด (Finite impulse response filter: FIR filter) เป็นตัวกรองที่ไม่มีการป้อนกลับ ซึ่งสามารถแสดงสมการฟังก์ชันถ่ายโอน (Transfer function) ของวงจรกรองแบบผลตอบสนองอิมพัลส์จำกัดได้ดังสมการที่ 2.1

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k} \quad (2.1)$$

สมการฟังก์ชันถ่ายโอนของวงจรกรองแบบผลตอบสนองอิมพัลส์ไม่จำกัดแสดงได้ดังสมการที่ 2.2

$$H(z) = \frac{\sum_{k=0}^N a_k z^{-k}}{1 + \sum_{k=1}^M b_k z^{-k}} \quad (2.2)$$

วงจรกรองสัญญาณดิจิทัลที่ปรับค่าได้ (Variable digital filter) คือวงจรกรองสัญญาณดิจิทัลที่สามารถเปลี่ยนแปลงคุณลักษณะของผลตอบสนองทางความถี่ได้ โดยทั่วไปเมื่อกล่าวถึงวงจรกรองสัญญาณดิจิทัลเรามักจะให้ความสนใจกับผลตอบสนองทางความถี่ เนื่องจากเป็นสิ่งที่บอกคุณลักษณะของวงจรกรองสัญญาณดิจิทัล สามารถจำแนกได้ 2 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองทางขนาด (Magnitude response) และผลตอบสนองทางเฟส (Phase response) ซึ่งสามารถแสดงได้จากสมการฟังก์ชันถ่ายโอนของวงจรกรองสัญญาณดิจิทัลได้ดังสมการที่ 2.3

$$H(z)|_{z=e^{j\omega}} = H(e^{j\omega}) \quad (2.3)$$

สามารถหาผลตอบสนองทางขนาดได้ดังสมการที่ 2.4

$$A(\omega) = |H(e^{j\omega})| \quad (2.4)$$

และผลตอบสนองทางเฟส ดังสมการที่ 2.5

$$\theta(\omega) = \angle H(e^{j\omega}) \quad (2.5)$$

ในการเปลี่ยนแปลงผลตอบสนองทางขนาดและผลตอบสนองทางเฟสของวงจรกรองสัญญาณดิจิทัล ทำได้ด้วยการคำนวณหาค่าสัมประสิทธิ์ที่สัมพันธ์กับพารามิเตอร์ หรือด้วยการปรับค่าพารามิเตอร์ เพื่อให้ผลตอบสนองของวงจรกรองสัญญาณดิจิทัลเปลี่ยนแปลงไปตามต้องการ

นอกจากนี้การแสดงให้เห็นถึงผลตอบสนองทางความขนาดที่เปลี่ยนแปลงไปตามค่าพารามิเตอร์ที่ปรับเปลี่ยน วงจรกรองสัญญาณดิจิทัลที่ปรับค่าได้ยังสามารถแสดงให้เห็นถึงผลตอบสนองทางเฟสที่เปลี่ยนแปลงไปด้วย จากผลตอบสนองทางเฟสสามารถแสดงได้เป็น 2 ลักษณะ คือ ผลตอบสนองความหน่วงเฟส (Phase delay response) และผลตอบสนองความหน่วงกลุ่ม (Group delay response) ดังสมการที่ 2.6 และสมการที่ 2.7 ตามลำดับ

$$T_p = -\frac{\theta(\omega)}{\omega} \quad (2.6)$$

$$T_g = -\frac{d\theta(\omega)}{d\omega} \quad (2.7)$$

ผลตอบสนองความหน่วงเฟสของวงจรกรองสัญญาณคือปริมาณของค่าความหน่วงเวลาในแต่ละองค์ประกอบทางความถี่ของสัญญาณที่ประกอบไปด้วยหลายองค์ประกอบทางความถี่

วงจรกรองสัญญาณที่มีผลตอบสนองเฟสแบบไม่เป็นเชิงเส้นจะเป็นสาเหตุที่ทำให้เกิดการผิดเพี้ยนทางเฟส (Phase distortion) ของสัญญาณซึ่งอาจส่งผลกระทบต่อรูปร่างของสัญญาณที่ผ่านวงจรกรองออกมาผิดเพี้ยนไป ซึ่งความผิดเพี้ยนดังกล่าวเป็นสิ่งที่ไม่พึงปรารถนาให้เกิดขึ้นในระหว่างการทำงาน โดยเฉพาอย่างยิ่งการไปประยุกต์ใช้ที่ต้องอาศัยสายตาในการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มองรูปคลื่นหรือรูปของสัญญาณ เช่นงานทางด้านวิดีโอและการประมวลผลสัญญาณด้านภาพ งานทางด้าน การประมวลผลสัญญาณชีวการแพทย์ การผิดเพี้ยนทางเฟสดังกล่าวสามารถหลีกเลี่ยงได้ โดยการใช้วงจรกรองสัญญาณที่มีผลตอบสนองเฟสแบบเป็นเชิงเส้นในช่วงแถบความถี่ที่เราสนใจ

2.2 วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอด (Allpass based fractional delay digital filter)

เมื่อพิจารณาวงจรกรองสัญญาณดิจิทัลแบบมีการป้อนกลับซึ่งมีฟังก์ชันการถ่ายโอน ดังสมการที่ 2.8

$$H(z) = \frac{z^{-N}D(z^{-1})}{D(z)} = \frac{a_N + a_{N-1}z^{-1} + \dots + a_1z^{-(N-1)} + z^{-N}}{1 + a_1z^{-1} + \dots + a_{N-1}z^{-(N-1)} + a_Nz^{-N}} \quad (2.8)$$

โดยที่ N คือ อันดับของวงจรกรองสัญญาณ

a_k คือ ค่าสัมประสิทธิ์ของวงจรกรองสัญญาณค่าความหน่วงเป็นเศษส่วน $k = 1, 2, \dots, N$ และเป็นจำนวนจริง

สำหรับฟังก์ชันถ่ายโอนนี้สัมประสิทธิ์ตัวเศษและตัวส่วนจะเหมือนกัน แต่อันดับเป็นส่วนกลับซึ่งกันและกัน ดังสมการที่ 2.9

$$H(z)H(z^{-1}) = \frac{z^{-N}D(z^{-1})}{D(z)} \frac{z^N D(z)}{D(z^{-1})} = 1 \quad (2.9)$$

รากของตัวส่วน (Poles) ของวงจรกรองสัญญาณแบบผ่านตลอด (All-pass filter) จะมีเสถียรภาพเมื่ออยู่ในวงกลมหนึ่งหน่วยบนระนาบเชิงซ้อน (z-plane) และรากของตัวเศษ (Zeros) จะอยู่ด้านนอกวงกลมหนึ่งหน่วย โดยทั้ง pole และ zero จะอยู่บนมุมเดียวกันแต่ความยาวของรัศมีจะเป็นส่วนกลับซึ่งกันและกัน ด้วยเหตุผลนี้ผลตอบสนองทางขนาดของวงจรกรองสัญญาณแบบผ่านตลอดจึงราบเรียบ ดังสมการที่ 2.10

$$|H(e^{j\omega})| = \left| \frac{e^{-j\omega N} D(e^{-j\omega})}{D(e^{j\omega})} \right| = 1 \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับนักศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความหน่วงของสัญญาณที่ไม่ต่อเนื่องทางเวลา (Discrete-time signal) ที่มีค่าเป็นจำนวนเต็มนั้นสามารถสร้างได้ง่าย อย่างไรก็ตามสำหรับการประยุกต์ใช้งานบางประเภทจำเป็นต้องใช้ค่าความหน่วงที่เป็นเศษส่วน

พิจารณาสัญญาณที่ไม่ต่อเนื่องทางเวลา $x(n]$ และ DTFT $X(\omega)$ ในสมการที่ 2.11

$$x(n] \leftrightarrow X(\omega) \quad (2.11)$$

เมื่อ n_1 คือ จำนวนเต็ม คุณสมบัติการเลื่อนของการแปลงฟูเรียร์ของสัญญาณที่ไม่ต่อเนื่องทางเวลาเป็นดังสมการที่ 2.12

$$x(n - n_1] \leftrightarrow X(\omega)e^{-jn_1\omega}, \quad n_1 \in \mathbb{Z} \quad (2.12)$$

จากคุณสมบัตินี้สามารถนำไปสู่การเลื่อนที่ไม่เป็นจำนวนเต็ม เพื่อให้ได้สัญญาณที่มีค่าความหน่วงแบบไม่เป็นจำนวนเต็ม ดังสมการที่ 2.13 เมื่อ D คือค่าความหน่วง

$$x(n - D] \leftrightarrow X(\omega)e^{-jD\omega}, \quad |\omega| < \pi, \quad D \in \mathbb{R} \quad (2.13)$$

จะได้ผลตอบสนองทางความถี่ของระบบที่มีความหน่วงเป็นเศษส่วนดังสมการที่ 2.14

$$H(\omega) = e^{-jD\omega}, \quad |\omega| < \pi \quad (2.14)$$

สังเกตได้ว่าเป็นระบบแบบผ่านตลอดเพราะ $|H(\omega)| = 1$ และมีผลตอบสนองทางเฟสดังสมการที่ 2.15

$$\angle H(\omega) = -D\omega, \quad |\omega| < \pi \quad (2.15)$$

และผลตอบสนองค่าความหน่วงกลุ่มคือ $H_{grpd}(\omega) = D$ โดยผลตอบสนองอิมพัลส์จะเป็นไปตามสมการที่ 2.16

$$h(n) = \text{sinc}(n - D) \quad (2.16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับระบบการหน่วงเวลาแบบเศษส่วนในอุดมคติ (Ideal fractional-delay) เป็นรูปแบบหนึ่งของวงจรกรองสัญญาณแบบผ่านตลอด ซึ่งได้มีการออกแบบระบบการกรองสัญญาณแบบผ่านตลอดที่ไม่ต่อเนื่องทางเวลา (Discrete-time all-pass system) เพื่อให้ค่าความหน่วงเป็นเศษส่วนไว้มากมาย หนึ่งในนั้นคือสมการของไธราน (Thiran) ที่ถูกนำมาประยุกต์ใช้สร้างวงจรกรองสัญญาณแบบผ่านตลอดที่มีความหน่วงเวลาที่ราบเรียบที่สุด (Maximally-flat delay all-pass filter) โดยที่ $\binom{N}{k}$ คือ สัมประสิทธิ์ทวินาม (Binomial coefficient) แสดงได้ดังสมการที่ 2.17

$$\binom{N}{k} = \frac{N!}{k!(N-k)!} \quad (2.17)$$

ค่าสัมประสิทธิ์ a_k จะสามารถคำนวณได้จากสมการที่ 2.18

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^N \frac{D-N+n}{D-N+k+n} \quad (2.18)$$

เมื่อ a_k คือ ค่าสัมประสิทธิ์ค่าสัมประสิทธิ์ของวงจรกรองสัญญาณค่าความหน่วงเป็นเศษส่วนของไธราน

D คือ ค่าพารามิเตอร์ความหน่วง (Delay parameter)

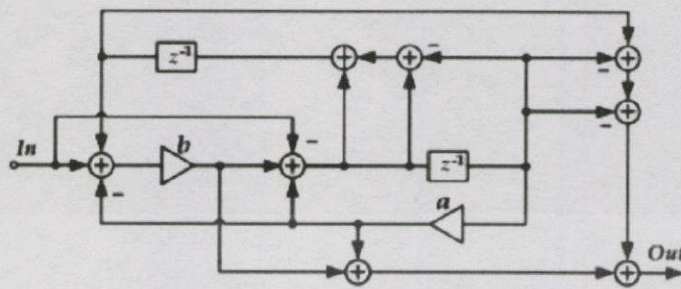
N คือ อันดับของวงจรกรองสัญญาณ

$k = 0, 1, 2, \dots, N$

2.2.1 โครงสร้างวงจรกรองแบบผ่านตลอดอันดับที่ 2

จากวิธีหาค่าสัมประสิทธิ์ของไธราน ได้มีการนำมาสร้างเป็นโครงสร้างวงจรกรองแบบผ่านตลอดที่สามารถใช้ค่าความหน่วงเป็นเศษส่วนได้ และยังได้รับการพัฒนาให้มีค่า sensitivity ต่ำ มีจำนวนตัวคูณและตัวหน่วงเวลาน้อย มีค่า round-off noise ที่ต่ำและคงที่ นอกจากนั้นแล้ว โครงสร้างยังมีการสูญเสียต่ำ โดยหนึ่งในนั้นคือโครงสร้าง IS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โครงสร้างของวงจรกรองสัญญาณแบบผ่านตลอดชนิด IS [5]

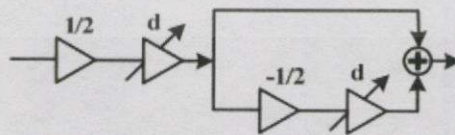
ซึ่งมีฟังก์ชันการถ่ายโอนดังสมการที่ 2.19

$$H_{is}(z) = \frac{b + (-a - 2b + ab)z^{-1} + z^{-2}}{1 + (-a - 2b + ab)z^{-1} + bz^{-2}} \quad (2.19)$$

โดยที่ a และ b คือ ค่าสัมประสิทธิ์ของตัวคูณในโครงสร้าง IS

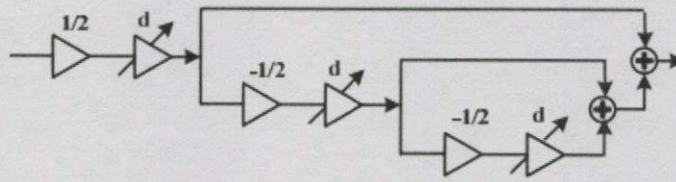
2.2.2 โครงสร้างวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดอันดับที่ 2 (Second-order variable FD allpass filter)

ค่าสัมประสิทธิ์ที่นำมาประกอบเป็นชุดตัวคูณ จะประกอบด้วยตัวคูณแบบปรับค่าได้ และแบบคงที่ ซึ่งชุดตัวคูณจากการประมาณเทย์เลอร์ (Taylor) อันดับ 2 แสดงดังรูปที่ 2.3 สำหรับชุดตัวคูณจากการประมาณเทย์เลอร์อันดับ 3 แสดงดังรูปที่ 2.4 และโครงสร้างของชุดตัวคูณแสดงดังรูปที่ 2.5

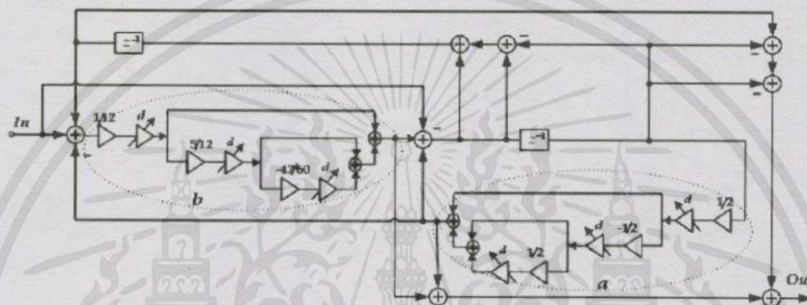


รูปที่ 2.3 ชุดของตัวแบบปรับค่าได้หลังจากการประมาณค่าแบบเทย์เลอร์อันดับ 2 [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายกเว้น ไม่อนุญาตให้เผยแพร่ไปบนเว็บไซต์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ชุดของตัวแบบปรับค่าได้หลังจากการประมาณค่าแบบเทย์เลอร์อันดับ 3 [5]



รูปที่ 2.5 โครงสร้างของชุดตัวคูณ (โครงสร้าง IS จากวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดอันดับ 2) [5]

2.3 การออกแบบวงจรดิจิทัลด้วยภาษา VHDL

ในการออกแบบวงจรดิจิทัล (Digital Circuit) นั้นในปัจจุบันก้าวหน้าไปอย่างมาก โดยการใช้ภาษาบรรยายการทำงานของวงจร (Hardware Description Language: HDL) ซึ่งเป็นภาษาที่ใช้สำหรับออกแบบฮาร์ดแวร์โดยภาษาที่เป็นมาตรฐานสากล เช่น Verilog หรือ VHDL (VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit)) หรือภาษาที่ไม่เป็นมาตรฐานเช่น AHDL (Altera Hardware Description Language) หรือ PHDL (Philips Hardware Description Language) เป็นต้น มาบรรยายการทำงานของวงจรที่ได้ ออกแบบไว้ ซึ่งในโครงการนี้ได้ใช้ภาษา VHDL มาทำการออกแบบวงจรกรองสัญญาณดิจิทัล ทำให้ลดความยุ่งยากในการนำเอาอุปกรณ์มาเชื่อมต่อให้เป็นวงจรรวมทั้งลดเวลาที่ใช้ในการออกแบบ และทดสอบการทำงาน ซึ่งมีความแตกต่างเป็นอย่างมากเมื่อเปรียบเทียบกับการทำงานในอดีตที่ผ่านมาคือผู้ออกแบบจะต้องนำเอาอุปกรณ์แต่ละตัวที่ทำการออกแบบไว้มาทำการต่อทดลองในแผงวงจรจริงและทำการทดสอบวงจรเพื่อหาข้อผิดพลาด ซึ่งต้องใช้เวลาอันยาวนานกับการแก้ปัญหาแต่ละอย่างที่เกิดขึ้น แต่ในการออกแบบด้วยภาษา VHDL ผู้ออกแบบเพียงแต่เขียนซอสโค้ด (Source

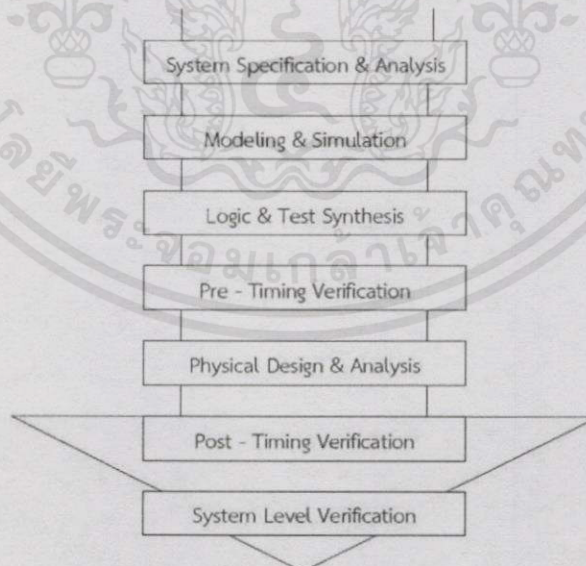
Code) บรรยายการทำงานของวงจรหลังจากนั้นก็ทำการคอมไพล์ (Compile) แล้วจำลองการทำงานการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน (Simulate) ดูว่าได้ฟังก์ชันการทำงานและไทม์มิ่ง (Timing) ตามที่ต้องการหรือไม่ จากนั้นก็นำซอสโค้ดที่ได้ไปทำการสังเคราะห์ด้วยโปรแกรมสังเคราะห์ (Synthesis Tool) สุดท้ายนำวงจรที่ได้จากการสังเคราะห์ไปทำการแมป (Map) ลงไปยัง FPGA (Field Programmable Gate Array) เพื่อเป็นชิป (Chip) ต้นแบบสำหรับการนำไปทดสอบการทำงาน

2.3.1 การออกแบบจากบนลงล่าง

ในการพัฒนางจรรวมดิจิทัลขนาดใหญ่ที่มีความซับซ้อน ผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อน จากนั้นจึงวิเคราะห์ที่ลึกถึงรายละเอียดต่อไป ซึ่งภาษา VHDL นั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อกและวิเคราะห์การทำงานแก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์ เพื่อให้ได้การทำงานตามที่ต้องการโดยการออกแบบในลักษณะนี้เรียกว่าหลักการออกแบบจากบนลงล่าง (Top-Down Design) ดังรูปที่ 2.6 ซึ่งถ้าเปรียบเทียบกับวิธีการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาในการออกแบบมากกว่า เพราะเป็นการวาดวงจรด้วยอุปกรณ์ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบจำลองการทำงานตรวจสอบความถูกต้อง ซึ่งใช้เวลามากและถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากในการออกแบบลักษณะนี้ ดังนั้นการใช้ภาษา VHDL กับหลักการออกแบบจากบนลงล่างจึงเป็นวิธีการที่เหมาะสมสำหรับการออกแบบและพัฒนางจรที่มีความซับซ้อนมากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเราซึ่งใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.6 ขั้นตอนการออกแบบจากบนลงล่าง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 แสดงให้เห็นถึงขั้นตอนการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติ อาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย โดยขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียด ดังนี้

- 1) ขั้นตอนการสร้างข้อกำหนดของความต้องการและวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา
- 2) ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษา VHDL สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงานเพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด
- 3) ขั้นตอนการสังเคราะห์ซึ่งจะต้องทำการกำหนดเทคโนโลยีที่จะมารองรับ วงจรที่ออกแบบ และระบบช่วยออกแบบจะทำการสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์หรือวงจรในระดับเกต (Gate Level) และการ เชื่อมต่อกันของอุปกรณ์เหล่านั้นหรือไม่ก็อยู่ในรูปของเน็ตลิสต์ (Net List) ที่สามารถนำไปผลิตลง บนอุปกรณ์อื่นได้
- 4) หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกตหรือเน็ตลิสต์แล้ว ข้อมูลที่ ได้นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชันแล้วยังมีข้อมูล ที่เกี่ยวกับเวลาด้วย ซึ่งจากความจริงที่ว่าอุปกรณ์อิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการเคลื่อน ผ่าน (Propagation Delay Time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาทีแต่ถ้า ภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันจำนวน 10,000 เกต ขึ้นไปเวลาดังกล่าวนี้จะสะสมกัน มากขึ้นจนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่ สัญญาณนาฬิกาสูงได้
- 5) ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของอุปกรณ์ FPGA หรือ วงจรรวม ASIC
- 6) หลังจากที่ได้วงจรจริงมาแล้ว ยังต้องมีความจำเป็นที่จะต้องตรวจสอบ การทำงานที่คำนึงถึงเวลาด้วยเพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับ อุปกรณ์อื่น ๆ ให้เป็นระบบ เพราะในขั้นตอนนี้วงจรที่ออกแบบจะประกอบด้วยอินพุตและเอาต์พุต แพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก
- 7) หลังจากที่น่าวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นให้เป็นระบบแล้ว นั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นอีกครั้ง ซึ่งเป็นการทดสอบการ ทำงานจริงขั้นสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ภาษา VHDL และส่วนประกอบของภาษา

วิวัฒนาการของภาษา VHDL นั้นเริ่มต้นประมาณปี ค.ศ.1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกาหรือ DOD (Department of Defense) ได้ทำการพัฒนาโครงการที่มีชื่อว่า VHSIC ซึ่งเป็นการพัฒนาโปรแกรมซึ่งจัดเป็นภาษาระดับสูง เช่นเดียวกับภาษา C หรือ Pascal แต่สามารถบรรยายพฤติกรรมการทำงานของวงจรดิจิทัลหรือโครงสร้างของวงจรได้ ทั้งนี้เพื่อให้สามารถออกแบบและสร้างวงจรรวมได้รวดเร็วขึ้น

ในการเขียนรูปแบบบรรยายระบบดิจิทัลในลักษณะของการออกแบบจากบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบของรูปแบบภาษา VHDL เสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

- 1) หน่วยการออกแบบเอนทิตี (Entity Design unit)
- 2) หน่วยการออกแบบสถาปัตยกรรม (Architecture Design unit)
- 3) หน่วยการออกแบบแพ็คเกจ (Package Design unit)
- 4) หน่วยการออกแบบโครงแบบ (Configuration Design unit)

2.3.2.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างภายนอกกับรูปแบบที่เขียนขึ้นโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น ซึ่งจะมีรูปแบบการเขียนดังรูปที่ 2.7

```
Entity component_name is
    Input and Output ports
    Physical and other parameter
End [component_name];
```

รูปที่ 2.7 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

จากรูปที่ 2.7 ส่วนนี้จะขึ้นต้นด้วยคำว่า Entity และ is ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อรูปแบบที่ต้องการจะเขียน (component_name) หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (Input-Output) รวมทั้งพารามิเตอร์อื่น ๆ และที่สำคัญคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า End และเครื่องหมายอัฒภาค (;) เสมอ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรมคือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงานพฤติกรรมที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ที่กำหนดในหน่วยการออกแบบเอนทิตี

ในหน่วยนี้มีรูปแบบการเขียนดังรูปที่ 2.8 ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่ง que แสดงให้เห็นว่า Architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตี (of <entity design unit> is) ส่วนที่อยู่ระหว่าง Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเพื่อเลือก (Option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น ประเภท (type) ต่าง ๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ค่าคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง port) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (Concurrent statement) เท่านั้นคือทุก statement จะทำงานพร้อมกันลำดับก่อนหลังจะไม่มีผลต่อการทำงานของรูปแบบ โดยทั่วไปการเขียนรูปแบบระบบดิจิทัลด้วยภาษา VHDL สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

- 1) ลักษณะการไหลของข้อมูล (Dataflow style)
- 2) ลักษณะพฤติกรรม (Behavioral style)
- 3) ลักษณะโครงสร้าง (Structural style)
- 4) ลักษณะผสม (Mixed Model style)

```

Architecture identifier of component_name is
    [declaration]
    Begin
        Specification of the functionality
        of the component in terms of its
        input lines and as influenced by
        physical and other parameters
    End [identifier];
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.8 โครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.3 หน่วยการออกแบบแพ็คเกจ

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อยที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิทัลสามารถเก็บไว้ในส่วนของแพ็คเกจได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดยหน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือจากหน่วยการออกแบบแพ็คเกจอื่น ๆ โดยปกติแล้วแพ็คเกจจะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็คเกจ (Package Body) เนื่องจากแพ็คเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็คเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อนซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

Package Declaration ส่วนที่มีความสำคัญที่สุดของแพ็คเกจ (ถ้ามองในแง่การนำไปใช้จากภายนอก) ได้แก่ส่วนการประกาศแพ็คเกจ เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ประกาศอยู่ในแพ็คเกจสำหรับนำไปใช้ภายนอกตัวของแพ็คเกจเอง สิ่งที่ถูกประกาศไว้ในส่วนของบอดีแพ็คเกจแต่ไม่ได้ถูกประกาศไว้ในส่วนการประกาศแพ็คเกจจะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือจุดเชื่อมต่อหรือพอร์ตที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็คเกจสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดีที่แสดงในรูปที่ 2.9 และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศชนิด (Type) หรือสัญญาณ เช่นเดียวกันกับส่วนบอดีแพ็คเกจที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็คเกจ แต่แพ็คเกจนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
Package package_name is
    Package_declaration_part
End package_name;
```

รูปที่ 2.9 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็คเกจ

Package body ดังรูปที่ 2.10 เป็นโครงสร้างที่ประกอบด้วยคำสั่งในรูปของคำสั่งลำดับ (Sequence) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลาย ที่ชื่อของโปรแกรมย่อยที่ถูกประกาศไปในส่วนของการประกาศแพ็คเกจแล้วจะถูกเก็บไว้ในส่วนบอดีแพ็คเกจ ทั้งนี้รวมทั้งการกำหนดค่าคงที่ อันได้แก่ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็คเกจ แต่ถูกกำหนดค่าในส่วนของบอดีแพ็คเกจ ฉะนั้นส่วนบอดีแพ็คเกจจึงไม่จำเป็นต้องมีถ้าในส่วนของการประกาศแพ็คเกจไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Package body package_name is
    declaration part
End package_name;
```

รูปที่ 2.10 โครงสร้างโดยทั่วไปของบอดีแพ็คเกจ

2.3.2.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิทัลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนิตีได้เพียงหนึ่งเดียวเท่านั้น แต่หน่วยการออกแบบเอนิตีหนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน มีรูปแบบการเขียนดังรูปที่ 2.11

```
Configuration identifier of entity_name is
    Configuration_declarative_part
End;
```

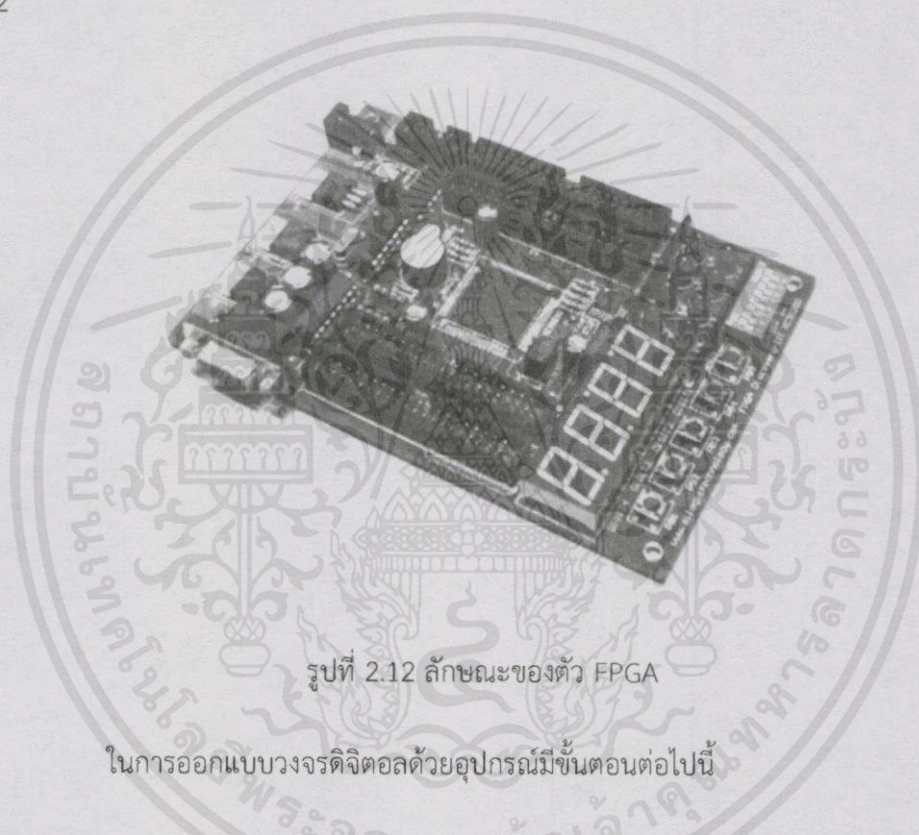
รูปที่ 2.11 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.3.3 การออกแบบวงจรถิจิตอลด้วยอุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรถิจิตอลที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ออกแบบไว้ในการทำ FPGA ซึ่งเป็นวิธีการออกแบบ IC (Integrated Circuit) แบบ Semicustom อีกวิธีหนึ่งเมื่อเทียบกับการทำ ASICs (Application Specific Integrated Circuit) แล้วนั้นก็ยังมีทั้งข้อดีและข้อเสียคือการทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรถะเพราะภายในอุปกรณ์ FPGA จะมีจำนวนเกต (Gate) ให้ใช้จำนวนจำกัด และการทำ FPGA ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิต ในปริมาณต่ำส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (Code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (Download) นั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ FPGA ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายในหรือปรับปรุงโครงสร้างสถาปัตยกรรมภายใน และยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ PPR (Partitioning, Placement and Routing) สำหรับอุปกรณ์ในการใช้งานนั้น อุปกรณ์ FPGA สามารถไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณดิจิทัล (DSP: Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้น ลักษณะของตัว FPGA แสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 ลักษณะของตัว FPGA

ในการออกแบบวงจรดิจิทัลด้วยอุปกรณ์มีขั้นตอนต่อไปนี

2.3.3.1 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

ในการออกแบบวงจรดิจิทัลนั้นทำได้โดยการวาดวงจรหรือใช้ภาษาอธิบายฮาร์ดแวร์ ในขั้นตอนนี้เป็นขั้นตอนที่ไม่แตกต่างกันระหว่างการออกแบบด้วย FPGA และ ASIC ในกรณีที่ใช้ภาษาอธิบายฮาร์ดแวร์ แต่ในกรณีที่ออกแบบโดยวิธีการวาดวงจรจะแตกต่างกัน โดยที่การทำวิธีนี้จะต้องคำนึงถึงเทคโนโลยีที่จะใช้ซึ่งแต่ละเทคโนโลยีก็มีความแตกต่างกันไป จะเห็นได้ว่าการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ทำได้สะดวกกว่า เพราะการทำด้วยวิธีนี้ไม่ต้องคำนึงถึงเทคโนโลยีที่จะใช้ (Technology independence) และที่สำคัญการออกแบบด้วยวิธีนี้สามารถที่จะแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่าเพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ یشنคันการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนโค้ดสิ่งที่จะต้องคำนึงถึงคือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรถูกและให้คุณสมบัติของวงจรถามที่กำหนดเพราะลักษณะการเขียนโค้ดจะมีผลโดยตรงกับวงจรถูกได้เนื่องจากในการสังเคราะห์วงจรมันซอฟต์แวร์สังเคราะห์วงจรถูก (Synthesis Tools) จะทำการสังเคราะห์ตามโค้ดที่เขียน ถ้าอธิบายการทำงานของวงจรถูกเดียวกันแต่เขียนโค้ดในลักษณะที่ต่างกัน เมื่อสังเคราะห์แล้วจะได้วงจรถูกที่ต่างกันและจากวงจรถูกที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันทั้งในด้านของขนาดหรือความเร็ว (Area and time) ส่วนการเขียนโค้ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

2.3.3.2 การจำลองการทำงานของวงจรถูก (Simulation)

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะเป็นขั้นตอนที่ใช้ตรวจสอบฟังก์ชันการทำงานของวงจรถูกว่าถูกต้องหรือไม่มีข้อผิดพลาดตรงไหน เพื่อที่จะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะใช้ซอฟต์แวร์สำหรับทำการจำลองการทำงานของวงจรถูก เช่น V-System และ ModelSim ของบริษัท Model Technology

2.3.3.3 การสังเคราะห์วงจรถูก

ในขั้นตอนนี้จะใช้ซอฟต์แวร์สังเคราะห์วงจรถูก (Synthesis tools) ทำการสังเคราะห์โค้ดเพื่อให้ได้เป็นวงจรถูกขึ้นมา แต่ต้องตรวจสอบด้วยว่าซอฟต์แวร์นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการใช้หรือไม่ โดย FPGA ที่นิยมใช้งานเช่นของบริษัท Xilinx และบริษัท Altera ซอฟต์แวร์สังเคราะห์วงจรถูกจะแปลงโค้ดและทำการออปติไมซ์ (Optimization) เพื่อให้ได้วงจรถูกตามเทคโนโลยีที่เลือกใช้นอกจากนี้ยังสามารถกำหนดข้อบังคับสำหรับวงจรถูกได้เช่น ข้อบังคับเรื่องของเวลา (Time constraints) หรือข้อบังคับในเรื่องของพื้นที่ ซึ่งข้อบังคับเหล่านี้จะถูกนำไป ใช้ในขั้นตอนออปติไมซ์เพื่อให้วงจรถูกที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการออปติไมซ์คือการเทียบ (Mapping) วงจรถูกให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรถูกที่เหมาะสมกับโครงสร้างสถาปัตยกรรมภายในอุปกรณ์ FPGA ในกรณีของ Xilinx จะเทียบโดยใช้วิธี LUT (Look Up Table) เมื่อทำการสังเคราะห์วงจรถูกเสร็จแล้วซอฟต์แวร์สังเคราะห์วงจรถูกก็จะมีรายงานผลว่าวงจรถูกที่ออกแบบไปนั้นเป็นอย่างไร เช่น มีความหน่วง (delay) เท่าไร ใช้ทรัพยากรใน FPGA อะไรบ้าง เป็นต้น

2.3.3.4 การแบ่งวงจรถูก (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรถูกที่ได้จากการสังเคราะห์ให้เป็นหน่วยย่อยสำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่นภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ การคำนวณนี้เป็นการคำนวณที่ค่อนข้างซับซ้อนและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ เพื่อช่วยลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำ โดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจร เช่น เกท (Gate), ฟลิปฟลอป (Flipflop) ลงในทรัพยากรที่มีอยู่ในอุปกรณ์ FPGA (CLBs, IOBs, BUFT และ Edge decoder) หลังจากทำขั้นตอนนี้เสร็จแล้วสามารถที่ทราบว่าจะวงจรถูกใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนซอฟต์แวร์ที่ใช้ในขั้นตอนนี้ขึ้นอยู่กับตัว FPGA ที่ใช้งานเช่น FPGA ของบริษัท Xilinx จะใช้ Xilinx Foundation Series 2.1i ซึ่งซอฟต์แวร์ตัวนี้จะรวมเอาซอฟต์แวร์ย่อยอื่น ๆ อีก เพื่อให้การทำ PPR (Partitioning, Placement and Routing) เป็นไปอย่างต่อเนื่อง ส่วน FPGA ของบริษัท Altera จะใช้ Altera MAX+II

2.3.3.5 การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าจะอยู่ในตำแหน่งใดในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่น วงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทาง (Route) ได้ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือตัว Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

2.3.3.6 การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่าง ๆ ภายในอุปกรณ์ FPGA เช่น ระหว่าง CLBs หรือระหว่าง CLBs กับ IOBs ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมดหรือเกิดความ หน่วงเกินค่าที่กำหนดในข้อบังคับโดยสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟต์แวร์ เช่นกัน หรือจะทำการเชื่อมต่อสัญญาณด้วยตัวเอง (Manual layout) ก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่าโดยให้ทำการค้นหาเส้นทางหลาย ๆ ครั้งเพื่อหาครั้งที่ดีที่สุด นอกจากนั้นการกำหนดข้อบังคับทางเวลา (Time constraints) จะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

2.3.3.7 การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่วงจรผ่านขั้นตอนจนกระทั่งผ่านการทำ PPR (Partitioning, Placement and Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (Download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้ให้เป็นข้อมูลวงจร (Configuration data) ซึ่งอยู่ในรูปบิตสตรีม (Bit-stream) ก่อนแล้วจึงดาวน์โหลดลงไป เพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามวงจรที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีนำไปใช้

จากที่อธิบายมาทั้งหมดจะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้น ทำได้สะดวกกว่าการทำ ASIC มากเพราะใช้เวลาน้อยกว่ามาก ส่วนสำคัญที่ใช้ในการทำ FPGA คือซอฟต์แวร์ที่ใช้ตั้งแต่การเขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงในอุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็นซอฟต์แวร์ที่ใช้งานต่อเนื่องกัน

2.4 Field-Programmable Gate Array (FPGA)

FPGA เป็นอุปกรณ์สารกึ่งตัวนำชนิดโปรแกรมได้ที่มีโครงข่ายการเชื่อมต่อภายในแบบเมตริกซ์ โครงสร้างภายในของ FPGA นั้นสามารถโปรแกรมให้มีหน้าที่การทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลายชนิด (Combinational logic gate) เพื่อให้ทำหน้าที่ที่มีความซับซ้อนเพิ่มขึ้น เช่น decoders หรือฟังก์ชันทางคณิตศาสตร์ใน FPGA ทั่วไป นอกจากจะประกอบด้วยส่วนของวงจรถอดจิกแบบโปรแกรมได้แล้ว ยังมีบล็อกของหน่วยความจำ ซึ่งอาจจะสร้างจากฟลิปฟล็อปอย่างง่าย หรือใช้พื้นที่ของสารกึ่งตัวนำสร้างเป็นหน่วยความจำจริง ๆ อยู่ภายในก็ได้

ในการออกแบบวงจรดิจิทัลอิเล็กทรอนิกส์ที่มี FPGA อยู่บนแผงวงจรด้วยนั้น จะช่วยให้ผู้ออกแบบสามารถลดขนาดของแผงวงจร รวมทั้งสามารถออกแบบได้รวดเร็ว ไม่ต้องทดสอบรายละเอียดภายในให้เสร็จสมบูรณ์ก่อน ก็สามารถออกแบบแผงวงจรได้ เมื่อได้รับแผงวงจรและประกอบอุปกรณ์ต่าง ๆ เสร็จแล้ว จึงค่อยกำหนดหน้าที่การทำงานของ FPGA ได้ในภายหลัง ต่างจากการออกแบบด้วยลอจิกเกตขนาดเล็กที่ต้องออกแบบทางเดินของสายทองแดงให้เสร็จสมบูรณ์ก่อน และไม่สามารถแก้ไขได้ในภายหลัง นอกจากนี้การใช้งาน FPGA สามารถโปรแกรมการทำงานได้ในทุกขณะแม้แต่ขณะที่จัดทำขึ้นงานแล้ว ก็ยังสามารถแก้ไขวงจรได้โดยง่าย จึงเป็นที่มาของคำว่า "Field Programmable" ซึ่งหมายถึงโปรแกรมได้ในภาคสนามหรือที่ทำงานนั่นเอง อย่างไรก็ตามรูปแบบของวงจรที่สังเคราะห์ (Configuration) ของ FPGA จะหายไปหลังจากปิดไฟเลี้ยง ดังนั้นจะต้องมีหน่วยความจำภายนอก (Flash) มาเก็บรูปแบบของวงจรที่สังเคราะห์บน FPGA ไว้ ซึ่ง FPGA จะมีกระบวนการอ่านรูปแบบของวงจรที่สังเคราะห์นั้นโดยอัตโนมัติหลังจากได้รับไฟเลี้ยง

2.4.1 ประเภทของ FPGA

ในปัจจุบันมี FPGA อยู่ 4 ชนิดที่วางขายอยู่ในท้องตลาดได้แก่ Symmetrical Array, Row-Based, Hierarchical PLD และ Sea-of-Gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการโปรแกรมที่แตกต่างไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรม ซึ่งมีอยู่ 2 แบบคือการโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรมโดยการใช้หน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

1) Fuse เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน

2) Anti Fuse เป็นวิธีการโปรแกรมที่คล้ายกันแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรมแล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

2.4.1.2 การโปรแกรมโดยใช้หน่วยความจำ

1) EEPROM Based FPGA

FPGA ที่ใช้การโปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกตต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกต แต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่ต้องมีไฟเลี้ยง และในโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิต ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

2) SRAM Based FPGA

FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถโปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกตในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 – 1,000,000 เกต) ซึ่งข้อดีของ SRAM Based FPGA คือใช้เวลาในการโปรแกรมน้อย (ระดับ nanosecond) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิปในขณะเริ่มต้นใช้งาน

2.4.2 ข้อได้เปรียบของการใช้ FPGA

1) ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิปเพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึงภาษาแอสเซมบลี (Assembly) ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2) การออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจรหรือเฮลดีแอล เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็วและไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มันจากนั้นตัวซอฟต์แวร์จะสังเคราะห์ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกันสามารถใช้ได้กับชิปทุกตัวและทุกบริษัท

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาว์นโหลดทางพอร์ตของคอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพ ขณะที่อยู่ในระบบได้โดยไม่ต้องจำเป็นต้องถอดมาโปรแกรมข้างนอกและที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มเติมแต่อย่างใด

2.4.3 เครื่องมือสำหรับการออกแบบ FPGA

การออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้เวลาน้อยกว่ามากด้วย ส่วนสำคัญที่ใช้ในการทำ FPGA คือซอฟต์แวร์ที่ใช้ตั้งแต่เขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาว์นโหลดลงในอุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็นซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้ สำหรับซอฟต์แวร์ที่ใช้ทำการจำลองการทำงานของวงจรมันต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบเพราะโมเดลที่ได้จากการทำขึ้นตอนต่าง ๆ ด้วยซอฟต์แวร์ต่าง ๆ ต้องเอามาจำลองการทำงานได้ และในการจำลองการทำงานของวงจรมันใช้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์คสเตชัน (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ที่ใช้บนพีซี (PC) มากขึ้นซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

2.5 การประยุกต์ใช้งาน

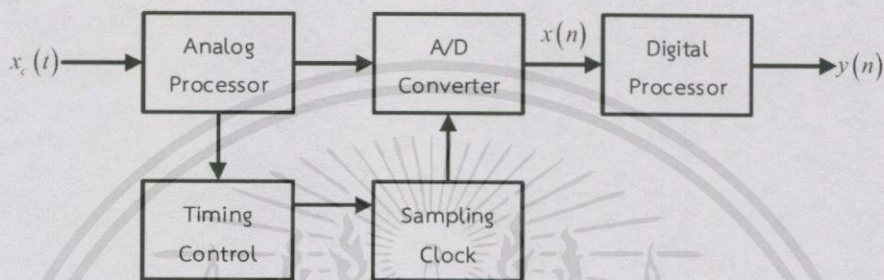
วงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดสามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยจะยกตัวอย่างการนำมาประยุกต์ใช้งานได้ดังนี้

2.5.1 การเข้าจังหวะในโมเด็มสัญญาณเชิงเลข (Synchronization in digital modem)

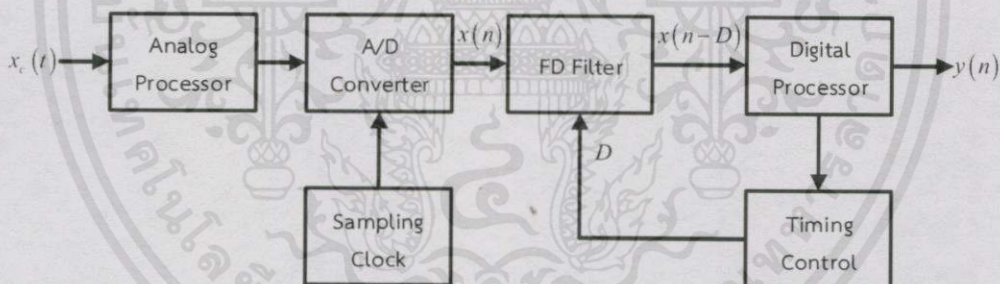
ในปัจจุบันการสื่อสารจะประมวลผลสัญญาณที่เป็นเชิงเลข ซึ่งการสื่อสารข้อมูลที่เป็นสัญญาณเชิงเลขจะมีจุดเริ่มต้นมาจากสัญญาณอนาล็อก จากนั้นจะผ่านกระบวนการแปลงเพื่อเป็นสัญญาณเชิงเลข เนื่องจากการประมวลผลสัญญาณอนาล็อกจะมีผลต่อปัจจัยภายนอกได้แก่ อุณหภูมิ การสูญเสียของสัญญาณ ตลอดจนควบคุมให้มีระบบเอาต์พุตที่ตรงตามที่ต้องการได้ยากกว่า ดังนั้นในหัวข้อนี้จะแสดงตัวอย่างของการเข้าจังหวะในโมเด็มสัญญาณเชิงเลขที่เป็นแบบดั้งเดิม ดังรูปที่ 2.13 จะเห็นได้ว่าระบบมีการจัดการข้อมูลในส่วนที่เป็นสัญญาณอนาล็อกอินพุต $x_c(t)$ และการควบคุมเวลา (Timing control) ของสัญญาณการซีกตัวอย่าง (Sampling clock) จะอยู่ภายใต้การควบคุมของส่วนประมวลผลสัญญาณอนาล็อก (Analog processor) เมื่อเสร็จจ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานในเชิงวิชาการเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบมีการควบคุมเวลาในกระบวนการของสัญญาณเชิงเลข ดังรูปที่ 2.14 ซึ่งสามารถทำให้ระบบมีประสิทธิภาพได้โดยนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) มาช่วยในการปรับการควบคุมเวลาให้มีค่าแม่นยำตามที่ต้องการได้ตลอดด้วยการปรับเปลี่ยนค่าความหน่วงของวงจรกรอง (Delay parameter)



รูปที่ 2.13 การควบคุมเวลาของสัญญาณการซีกตัวอย่างของส่วนประมวลผลสัญญาณอนาล็อก



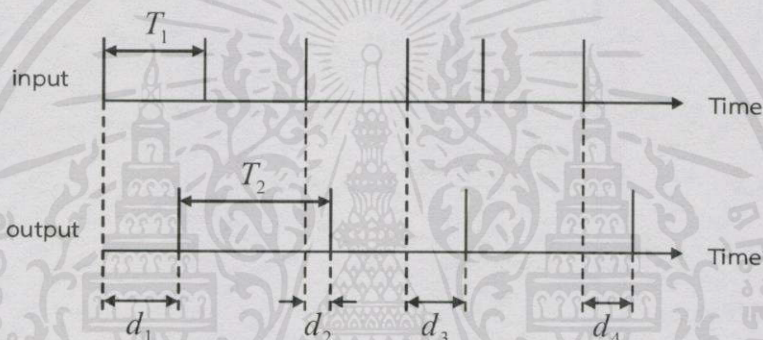
รูปที่ 2.14 การควบคุมเวลาของสัญญาณการซีกตัวอย่างของส่วนประมวลผลสัญญาณเชิงเลข

2.5.2 วงจรแปลงอัตราการซีกตัวอย่าง (Sampling rate conversion: SRC)

ในการประมวลผลสัญญาณเชิงเลขสิ่งที่สำคัญสิ่งหนึ่งคืออัตราการซีกตัวอย่าง (Sampling rate) ซึ่งในการนำไปใช้งานในแต่ละส่วนหรือแต่ละอย่าง อัตราการซีกตัวอย่างของความถี่มีความเป็นไปได้ที่จะมีค่าไม่เท่ากัน ซึ่งในการประมวลผลสัญญาณเชิงเลขจะทำให้แต่ละส่วนสามารถเชื่อมโยงหรือทำงานร่วมกันได้โดยไม่เกิดความผิดพลาด ในการประมวลผลสัญญาณเชิงเลข

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย่อมต้องประกอบด้วยส่วนที่สามารถปรับเปลี่ยนอัตราการซึกตัวอย่างให้มีค่าที่ตรงตามความต้องการ ซึ่งก็คือวงจรแปลงอัตราการซึกตัวอย่าง ในความเป็นจริงแล้วเมื่อนำวงจรแปลงอัตราการซึกตัวอย่างไปใช้งาน ในทางปฏิบัติอัตราการซึกตัวอย่างอาจมีค่าที่ไม่คงที่ตลอด ดังนั้นการนำวงจรแปลงอัตราการซึกตัวอย่างมาใช้ต้องเป็นวงจรที่มีความยืดหยุ่นในการปรับเปลี่ยนอัตราการซึกตัวอย่างหรือเป็นวงจรที่มีคุณลักษณะที่ปรับเปลี่ยนได้แบบทันทีทันใดหรือตลอดเวลา (Real time) นั่นคือเราสามารถประยุกต์ใช้งานของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) ซึ่งมีคุณสมบัติที่สามารถปรับเปลี่ยนได้แบบทันทีทันใดหรือตลอดเวลา ด้วยการปรับเปลี่ยนค่าความหน่วงของวงจรกรอง (Delay parameter) ดังรูปที่ 2.15

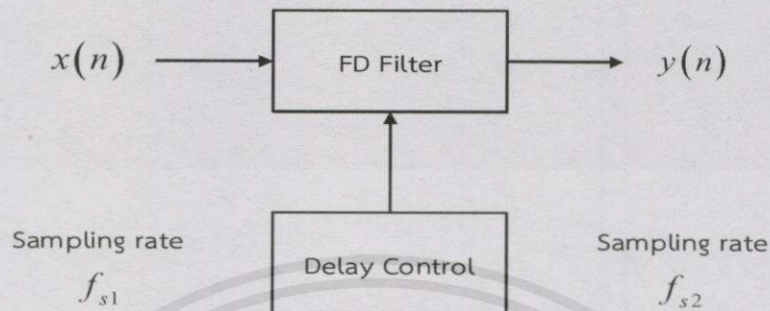


รูปที่ 2.15 สัญญาณอินพุตที่มีอัตราการซึกตัวอย่างที่ความถี่ f_{s1} แปลงไปสู่สัญญาณเอาต์พุตที่มีอัตราการซึกตัวอย่างที่ความถี่ f_{s2}

จากรูปที่ 2.15 แสดงให้เห็นถึงสัญญาณอินพุตที่มีอัตราการซึกตัวอย่างที่ความถี่ f_{s1} และต้องการให้ได้สัญญาณเอาต์พุตที่มีอัตราการซึกตัวอย่างที่ความถี่ f_{s2} ซึ่งสามารถปรับเปลี่ยนอัตราการซึกตัวอย่างได้โดยการปรับเปลี่ยนค่าความหน่วงของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน โดยจะเห็นได้ว่าค่าความหน่วงในแต่ละช่วงเวลาจะมีค่าที่ไม่เท่ากัน เพราะค่าความหน่วงจะต้องปรับเปลี่ยนเพื่อให้ได้สัญญาณเอาต์พุตที่มีอัตราการซึกตัวอย่างที่ต้องการ ซึ่งถ้าหากนำวงจรกรองสัญญาณเชิงเลขแบบอื่นมาประยุกต์ใช้งาน วงจรกรองสัญญาณอื่นอาจจะไม่แสดงผลตอบสนองแบบทันทีทันใดเท่าวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน ดังนั้นเราสามารถแสดงแผนผังของระบบวงจรแปลงอัตราการซึกตัวอย่างได้ดังรูปที่ 2.16 เมื่อสัญญาณอินพุตคือ $x(n)$ และสัญญาณเอาต์พุตคือ $y(n)$ และระบบสามารถแปลงอัตราการซึกตัวอย่างจาก

ความถี่ f_{s1} ไปสู่อัตราการซึกตัวอย่างความถี่ f_{s2} ได้ด้วยการปรับค่าความหน่วง (D)

ไม่ว่าการมีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แผนผังของระบบวงจรแปลงอัตราการซีกตัวอย่าง

2.5.3 การทำ Oversampling D/A

ปัจจุบันการเล่นเครื่องเสียงผ่าน Digital To Analog Converter หรือ DAC กำลังได้รับความนิยมอย่างสูง ทั้งนี้เพราะ DAT สามารถพัฒนาคุณภาพเสียงในระบบเสียงให้ดีขึ้นอย่างชัดเจนในงบประมาณที่ต่ำกว่าการใช้เครื่องเล่นซีดีราคาแพงอย่างมาก ทั้งยังสะดวกเพราะสามารถใช้เล่นกับ CD Transport, CD Player, DVD Player และเล่นไฟล์เพลงจากคอมพิวเตอร์ DAC ทั่วไป ในท้องตลาดเกือบทั้งหมดนั้นทำงานในระบบ Oversampling ซึ่งจะทำการทวีคูณความถี่มาตรฐานของแผ่นซีดี ที่ปกติจะอยู่ที่ 16 Bit/44.1 kHz ขึ้นไปเป็น 2 เท่า 4 เท่า 8 เท่า หรือมากกว่า โดยเชื่อว่าหากทำการ Oversampling ขึ้นไปมากเท่าไร คุณภาพเสียงจะยิ่งดีขึ้นเท่านั้น

Oversampling คือการควอนไทซ์ (Quantized) จริงๆ ทำเกินกว่า 1 ครั้ง แล้วเอาค่าเฉลี่ยของความสูงของคลื่นออกมาเป็น 1 ค่าแทน สมมติว่ามีค่า Oversampling 32 เท่า เราจะได้ค่า spec เป็น 1 channel - 4 bit - 1 Hz - Oversampling 32 time แทน และ Data ที่ออกมาควรจะมีค่าเป็น 00 (< 4 bit) ด้วย เพราะผลรวมของพื้นที่ในกราฟเท่ากับ 0 พอดี ถ้าค่า Oversampling เป็น 16,8,4 เท่า เราจะได้ค่า Spec เป็น

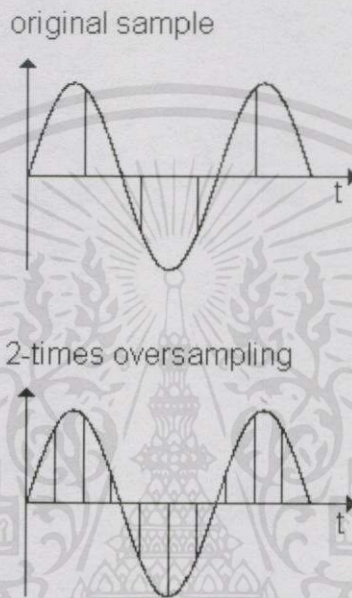
- 1 channel - 4 bit - 2 Hz - Oversampling 16 time
- 1 channel - 4 bit - 4 Hz - Oversampling 8 time
- 1 channel - 4 bit - 8 Hz - Oversampling 4 time

จะเห็นได้ว่า Hz ของ ADC/DAC กับ Hz ของเสียงเป็นคนละตัวกัน ซึ่งสามารถบอกได้ว่าใน 1 วินาที มีจุดที่จุดบนกราฟ

สัญญาณที่มีลักษณะความละเอียดต่ำจะสามารถทำให้มีความละเอียดของสัญญาณที่สูงขึ้นได้โดยการทำ Interpolation ซึ่งการทำ Interpolation คือกระบวนการหนึ่งของการทำ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากมีการนำไปใช้

Oversampling D/A ดังนั้นวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนสามารถประยุกต์ใช้ในการทำ Oversampling D/A ได้ โดยการประมาณในช่วงจุดแซมเปิ้ลที่ติดกันโดยความแม่นยำในการประมาณค่านั้นจะขึ้นอยู่กับอันดับของตัวกรองที่ใช้ ยิ่งใช้อันดับของตัวกรองมากก็จะยิ่งมีความละเอียดมากในการประมาณค่าในช่วง ดังรูปที่ 2.17



รูปที่ 2.17 ผลการเปรียบเทียบของสัญญาณคลื่นไซน์กับสัญญาณคลื่นไซน์ที่ผ่านการทำ Oversampling D/A 2 เท่า

การนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนมาประยุกต์ใช้เพื่อทำ Oversampling D/A เปรียบเสมือนเป็นการทวีคูณความถี่ดังที่ได้กล่าวมา ส่วนมากจะเน้นไปที่เครื่องเสียงเป็นส่วนใหญ่ ซึ่งในยุคปัจจุบันนี้ความต้องการความละเอียดของเสียงและภาพที่มาก ฟิลเตอร์ชนิดนี้จึงสามารถนำมาใช้งานได้จริงในอุตสาหกรรมทางด้านเทคโนโลยีเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปฏิญญาพันธ

3.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอด ไรราน



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไรราน

การทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไรรานสามารถแสดงเป็นบล็อกไดอะแกรมได้ดังรูปที่ 3.1 ซึ่งในการออกแบบวงจรนี้จะใช้ตัวกรองอันดับที่ 2 ทั้งแบบกำหนดค่าและปรับค่าได้ โดยจะแสดงการออกแบบและการคำนวณค่าสัมประสิทธิ์ในหัวข้อถัดไป

3.1.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไรรานโดยใช้ตัวกรองอันดับ 2 แบบกำหนดค่า

โครงสร้าง IS ถูกเลือกนำมาใช้เป็นวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดอันดับที่ 2 เนื่องจากค่าความไวที่ต่ำและความเสถียรของวงจร รวมถึง round-off noise ที่ต่ำและคงที่ โดยสามารถคำนวณค่าสัมประสิทธิ์ a และ b ได้จากสมการที่ 3.1 และ 3.2 โดยจะได้ค่าสัมประสิทธิ์ที่ใช้ในการออกแบบวงจรดังตารางที่ 3.1

$$a = \frac{D-2}{D} \tag{3.1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b = \frac{(D-1)(D-2)}{(D+1)(D+2)} \quad (3.2)$$

ตารางที่ 3.1 ค่าสัมประสิทธิ์ a และ b ที่ใช้สำหรับแต่ละค่า D

D	a	b
2.00	0	0
2.25	0.11	0.02
2.33	0.14	0.03
2.50	0.20	0.047
2.67	0.25	0.065
2.75	0.27	0.073

3.1.2 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรรานโดยใช้ตัวกรองอันดับ 2 แบบปรับค่าได้

สัญญาณดิจิทัลเป็นตัวแทนของสัญญาณอนาล็อกที่มีประโยชน์มาก แต่สามารถเกิดปัญหาขึ้นได้เมื่ออัตราการซีกตัวอย่างของสัญญาณที่ทำการบันทึกนั้น มีความแตกต่างจากอัตราการซีกตัวอย่างของกระบวนการที่จะกระทำต่อไป โดยมาตรฐานอัตราการซีกตัวอย่างในปัจจุบันมีอยู่เป็นจำนวนมาก ตัวอย่างเช่นการแปลงจาก CD (compact disc) ไปเป็น DAT (digital audio tape) เมื่อสัญญาณตัวอย่างมีค่าอัตราการซีกตัวอย่าง $F_{s,CD} = 44.1kSa/s$ เปลี่ยนไปเป็นสัญญาณที่มีอัตราการซีกตัวอย่าง $F_{s,DAT} = 48kSa/s$

ดังนั้นจะได้สัญญาณที่มีอัตราการซีกตัวอย่างเป็น $F_{s2} = L/MF_{s1}$ โดยกระบวนการนี้สามารถคำนวณได้โดยใช้สมการที่ 3.3 และ 3.4

$$L = F_{s2} / \gcd(F_{s1}, F_{s2}) \quad (3.3)$$

$$M = F_{s1} / \gcd(F_{s1}, F_{s2}) \quad (3.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการแปลงอัตราการชักตัวอย่างดังกล่าวเป็นวิธีการที่ง่ายต่อการดำเนินงานและการอธิบาย แต่การคำนวณยังไม่มีประสิทธิภาพที่เพียงพอ อย่างไรก็ตามการที่จะทำให้มีประสิทธิภาพขึ้นมาได้นั้นต้องใช้โครงสร้างหลายเฟส แล้วปัญหาที่มีต่อมาคือเมื่อ L และ M นั้นมีถึงหลักร้อยหรือมีขนาดใหญ่ เหมือนในตัวอย่างการแปลงจาก CD ไปเป็น DAT โดย $L=160$ และ $M=147$ นั้นจะทำให้เกิดปัญหาด้านการออกแบบตัวกรองการแทรกสอด โดยตัวกรองนี้ต้องการทรานซิสต์บนที่แคบและผลตอบสนองอิมพัลส์ที่ยาว สิ่งที่ทำหายที่สุดของขั้นตอนการแปลงอัตราการชักตัวอย่างคือการออกแบบตัวกรองการแทรกสอด นักออกแบบพยายามที่จะสร้างตัวกรองที่สั้นที่สุดที่สามารถเป็นไปได้ สำหรับผลตอบสนองอิมพัลส์ที่ยาวมาก วิธีแก้ปัญหานั้นที่เหมาะสมอาจจะไม่สามารถทำได้ และจะมีประสิทธิภาพที่น้อยกว่าการออกแบบตัวกรองโดยใช้หลักการ เช่นการออกแบบตัวกรองโดยใช้วิธีวินโดว์ ยิ่งไปกว่านั้น เมื่ออัตราส่วนของอัตราการการสุ่มตัวอย่างของอินพุตและเอาต์พุตเป็นจำนวนอตรรกยะ หรือเมื่อมีความแตกต่างทางเวลา ทำให้ตัวประกอบ L และ M ไม่สามารถกำหนดได้อย่างแน่นอนและตัวกรองการแทรกสอดไม่สามารถระบุได้

กระบวนการของตัวแปลงอัตราการชักตัวอย่างแท้จริงแล้วคือการคำนวณค่าของสัญญาณใหม่ที่อยู่ระหว่างสัญญาณเดิม ดังนั้นจึงสามารถหาเอาต์พุตแต่ละเอาต์พุตจากการหน่วงเวลาอินพุต

ค่าการหน่วงเวลาเป็นเศษส่วนระหว่างตัวอย่างกระแสเอาต์พุต $y[m]$ กับตัวอย่างอินพุตที่ใกล้ที่สุด $x[n]$ สามารถคำนวณได้จากสูตรรีเคอร์ซีฟ (Recursive) ดังสมการที่ 3.5

$$d[m] = d[m-1] - F_{s1} / F_{s2} + \Delta n[m] \quad \Delta n[m] \in [-0.5, 0.5] \quad (3.5)$$

โดยอัตราส่วนการชักตัวอย่าง $F_{s1} / F_{s2} = M / L$ และ $\Delta n[m]$ คือจำนวนค่าการชักตัวอย่างใหม่ที่ต้องการ แล้วใช้อินพุตเดิมเพื่อหาเอาต์พุตถัดไป ดังแสดงในสมการที่ 3.6

$$\Delta n[m] = \text{round}(F_{s1} / F_{s2} - d[m-1]) \quad (3.6)$$

ใช้ตัวแปรทั้งสองตัวนี้หาวิธีการแปลงอัตราการชักตัวอย่างได้

- 1). เริ่มต้นให้ $d[0]=0$ และ $n[0]=0$
- 2). รอ $\Delta n[m]$ ในอินพุตบัพเฟอร์
- 3). คำนวณค่าเอาต์พุต $y[m]$ จากการหน่วงเวลาของ $d[m]$
- 4). คำนวณค่า $\Delta n[m]$ และ $d[m]$ สำหรับตำแหน่ง m ถัดไป แล้ววน

กลับไปทำข้อ 2 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความห่วยเป็นเศษส่วนที่ปรับค่าได้นั้นคล้ายกับตัวแปลงอัตราการจัดตัวอย่าง การหาค่าเอาต์พุตแต่ละตัวต้องใช้วงจรกรองสัญญาณหน่วงเวลาเป็นเศษส่วนที่มีการประมาณหาผลตอบสนองอิมพัลส์สั้นกว่าวงจรกรองสัญญาณแทรกสอด (Interpolation filter) แบบดั้งเดิม โดยค่าความห่วยที่ใช้สามารถแสดงได้ดังตารางที่ 3.2

ตารางที่ 3.2 ค่าความห่วยที่ใช้สำหรับการแปลงอัตราการจัดตัวอย่าง

ลำดับ	D	ลำดับ (ต่อ)	D (ต่อ)	ลำดับ (ต่อ)	D (ต่อ)	ลำดับ (ต่อ)	D (ต่อ)
1	0.08125	21	-0.29375	41	0.33125	61	-0.04375
2	0.1625	22	-0.2125	42	0.4125	62	0.0375
3	0.24375	23	-0.13125	43	0.49375	63	0.11875
4	0.325	24	-0.05	44	-0.425	64	0.2
5	0.40625	25	0.03125	45	-0.34375	65	0.28125
6	0.4875	26	0.1125	46	-0.2625	66	0.3625
7	-0.43125	27	0.19375	47	-0.18125	67	0.44375
8	-0.35	28	0.275	48	-0.1	68	-0.475
9	-0.26875	29	0.35625	49	-0.01875	69	-0.39375
10	-0.1875	30	0.4375	50	0.0625	70	-0.3125
11	-0.10625	31	-0.48125	51	0.14375	71	-0.23125
12	-0.025	32	-0.4	52	0.225	72	-0.15
13	0.05625	33	-0.31875	53	0.30625	73	-0.06875
14	0.1375	34	-0.2375	54	0.3875	74	0.0125
15	0.21875	35	-0.15625	55	0.46875	75	0.09375
16	0.3	36	-0.075	56	-0.45	76	0.175
17	0.38125	37	0.00625	57	-0.36875	77	0.25625
18	0.4625	38	0.0875	58	-0.2875	78	0.3375
19	-0.45625	39	0.16875	59	-0.20625	79	0.41875
20	-0.375	40	0.25	60	-0.125	80	-0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ค่าความหน่วงที่ใช้สำหรับการแปลงอัตราสุ่มตัวอย่าง (ต่อ)

ลำดับ	D	ลำดับ (ต่อ)	D (ต่อ)	ลำดับ (ต่อ)	D (ต่อ)	ลำดับ (ต่อ)	D (ต่อ)
81	-0.41875	101	0.20625	121	-0.16875	141	0.45625
82	-0.3375	102	0.2875	122	-0.0875	142	-0.4625
83	-0.25625	103	0.36875	123	-0.00625	143	-0.38125
84	-0.175	104	0.45	124	0.075	144	-0.3
85	-0.09375	105	-0.46875	125	0.15625	145	-0.21875
86	-0.0125	106	-0.3875	126	0.2375	146	-0.1375
87	0.06875	107	-0.30625	127	0.31875	147	-0.05625
88	0.15	108	-0.225	128	0.4	148	0.025
89	0.23125	109	-0.14375	129	0.48125	149	0.10625
90	0.3125	110	-0.0625	130	-0.4375	150	0.1875
91	0.39375	111	0.01875	131	-0.35625	151	0.26875
92	0.475	112	0.1	132	-0.275	152	0.35
93	-0.44375	113	0.18125	133	-0.19375	153	0.43125
94	-0.3625	114	0.2625	134	-0.1125	154	-0.4875
95	-0.28125	115	0.34375	135	-0.03125	155	-0.40625
96	-0.2	116	0.425	136	0.05	156	-0.325
97	-0.11875	117	-0.49375	137	0.13125	157	-0.24375
98	-0.0375	118	-0.4125	138	0.2125	158	-0.1625
99	0.04375	119	-0.33125	139	0.29375	159	-0.08125
100	0.125	120	-0.25	140	0.375	160	0

3.2 การประยุกต์ใช้งาน Signal Interpolation

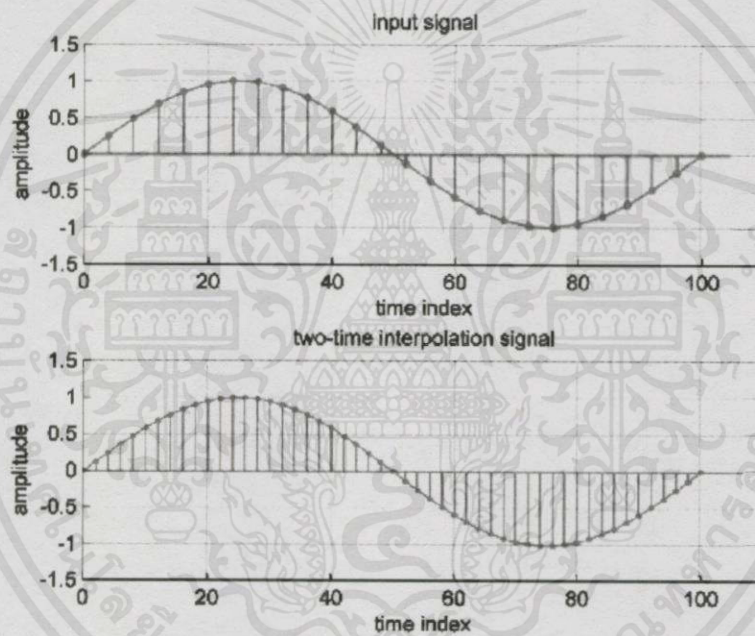
การนำมาประยุกต์ใช้งานเป็น Signal Interpolation นั้น ก่อนอื่นจะต้องทำการห้วงเวลาเป็นเศษส่วนแบบผ่านตลอดไครานเสียก่อน ซึ่งค่าการห้วงนั้นจะขึ้นอยู่กับว่าเราต้องการ Interpolation สัญญาณกี่เท่า แล้วนำสัญญาณที่ถูกห้วงไปทำการแทรกกับสัญญาณเดิม ซึ่งจะทำให้การแทรกสัญญาณไปเรื่อยๆ จนกระทั่งสิ้นสุดสัญญาณ โดยสามารถสรุปความสัมพันธ์ได้ดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ความสัมพันธ์ระหว่างการทำ Interpolation และจำนวนค่าการหน่วงที่ใช้

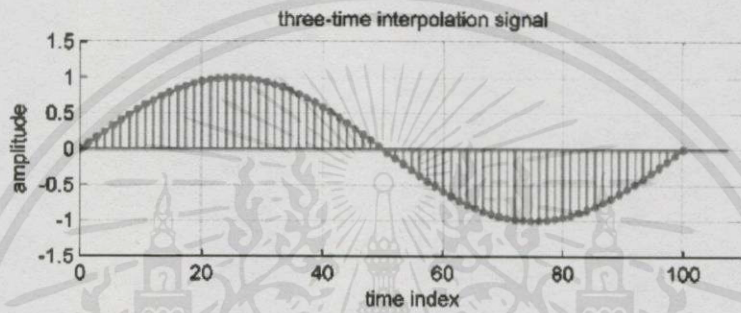
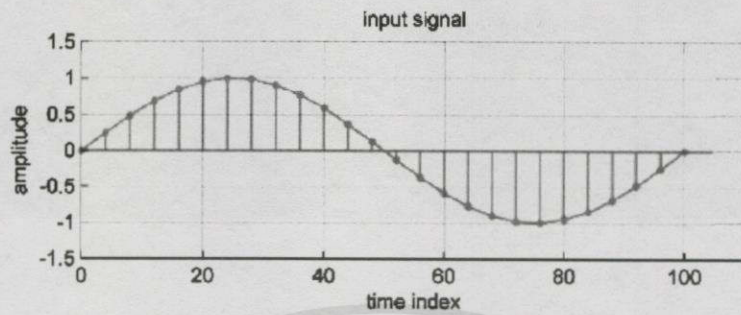
Interpolation	Delay Parameter ที่ใช้
Double Interpolation	0.50
Triple Interpolation	0.33,0.67
Four time Interpolation	0.25,0.50,0.75

การทำ Interpolation แต่ละแบบสามารถแสดงได้ดังรูปที่ 3.2, 3.3 และ 3.4

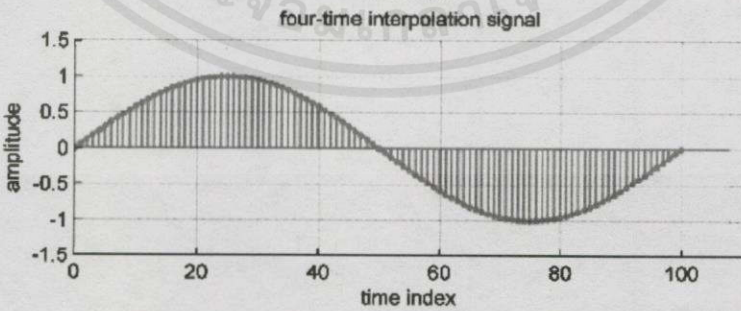
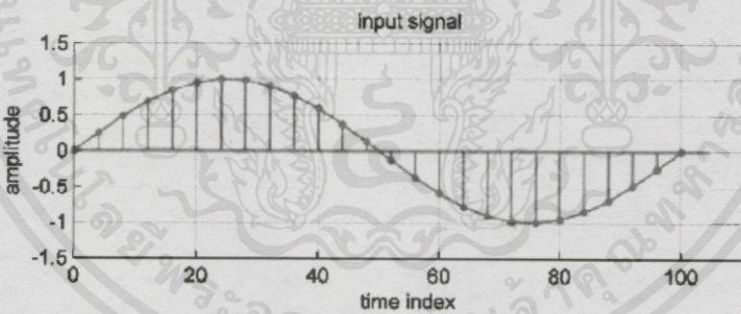


รูปที่ 3.2 ผลของการทำ Double Interpolation เทียบกับสัญญาณอินพุตเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ผลของการทำ Triple Interpolation เทียบกับสัญญาณอินพุตเดิม



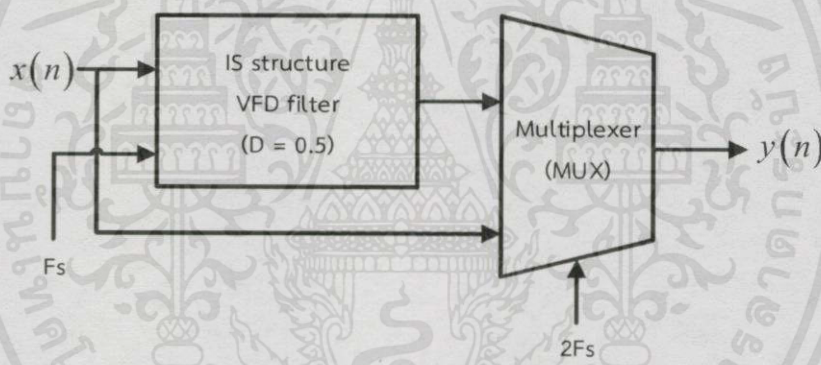
รูปที่ 3.4 ผลของการทำ Four time Interpolation เทียบกับสัญญาณอินพุตเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงนามเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากสถาบัน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าสัญญาณที่ทำ Signal Interpolation จะมีความละเอียดมากยิ่งขึ้น เนื่องจากนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรรานมาใช้ในการประมาณค่าในช่วง และนำการประมาณค่าในช่วงนั้นมาแทรกในสัญญาณเดิม

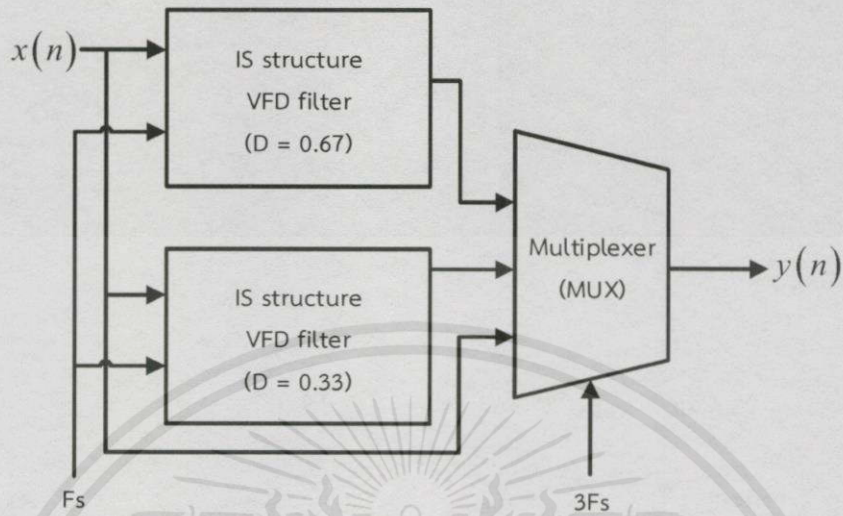
3.3 การทำ Oversampling D/A

ภายใน FPGA นั้นจะทำการออกแบบและสังเคราะห์วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนโรรานดังรูปที่ 3.1 เพื่อนำวงจรนี้มาประยุกต์ใช้งานในการทำ Oversampling D/A สัญญาณ 2 เท่า, 3 เท่า และ 4 เท่า เพื่อให้ได้สัญญาณที่มีความละเอียดเพิ่มขึ้น โดยทำการออกแบบวงจรได้ดังรูปที่ 3.5, 3.6 และ 3.7 โดยใช้วงจร Multiplexer (MUX) ในการรวมสัญญาณ สัญญาณที่ได้นั้นจะมีลักษณะที่มีความละเอียดมากยิ่งขึ้นเมื่อเทียบกับสัญญาณอินพุตเดิม

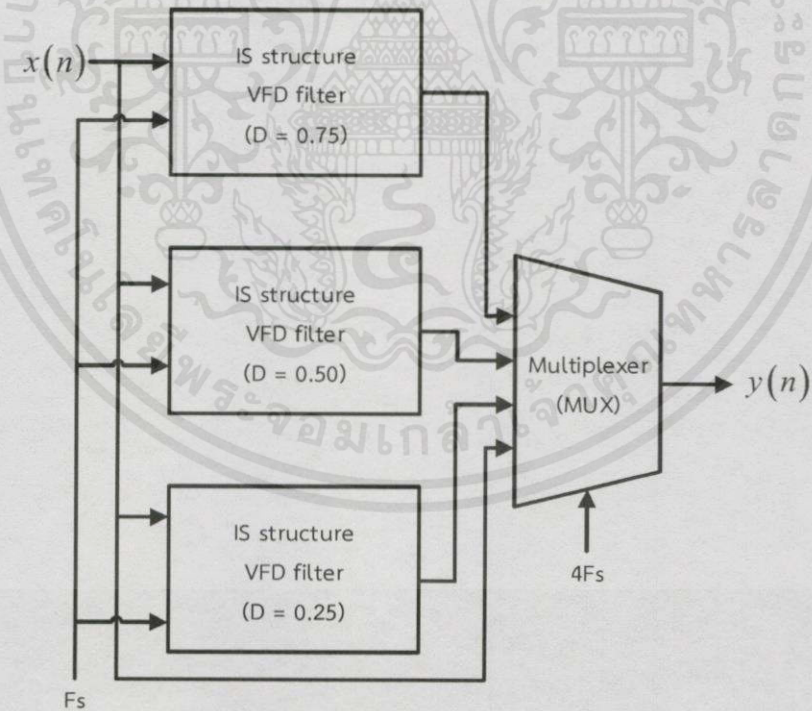


รูปที่ 3.5 บล็อกไดอะแกรมการทำงานของการทำงานของการทำ Oversampling D/A 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 บล็อกไดอะแกรมการทำงานของกรทำ Oversampling D/A 3 เท้า

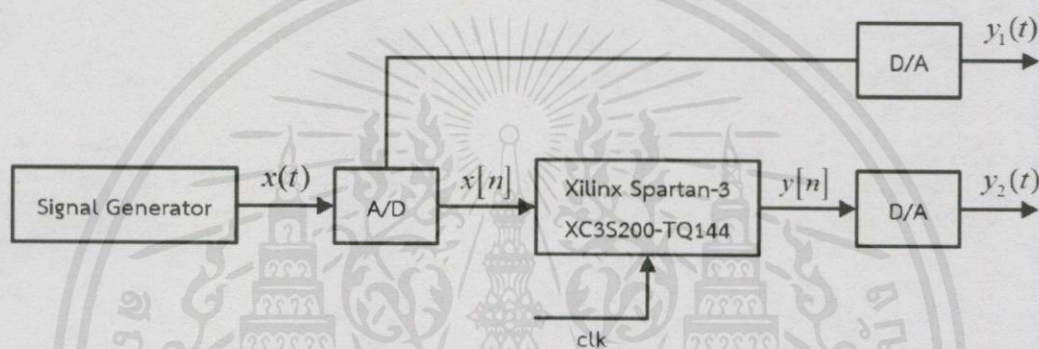


รูปที่ 3.7 บล็อกไดอะแกรมการทำงานของกรทำ Oversampling D/A 4 เท้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนที่ออกการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คิดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอด ไรโรนบนฮาร์ดแวร์

การออกแบบและการสังเคราะห์วงจรขึ้นมาใช้นั้น จะใช้ภาษา VHDL ในการออกแบบและสังเคราะห์วงจร โดยใช้โปรแกรม Xilinx ซึ่งเป็นภาษาที่ถูกใช้ในการออกแบบและสังเคราะห์วงจรทางดิจิทัล จากนั้นจึงทำการสังเคราะห์ลงบนอุปกรณ์ FPGA เพื่อให้สามารถนำวงจรที่ทำการออกแบบไว้มาใช้งานได้จริง โดยสามารถแสดงอุปกรณ์การจะจัดเก็บผลได้ดังรูปที่ 3.8



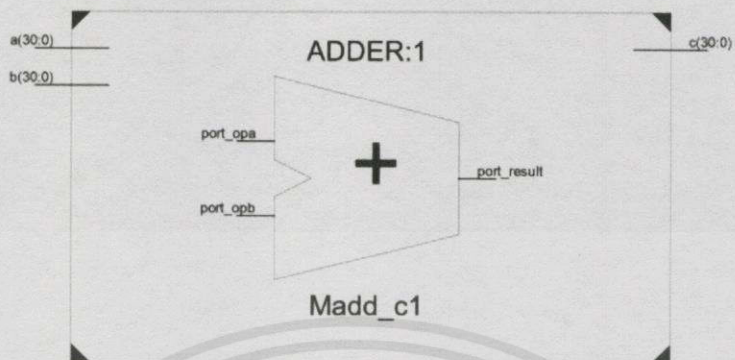
รูปที่ 3.8 อุปกรณ์ในการจัดเก็บผลการทดลองบนฮาร์ดแวร์

การจัดเก็บผลการทดลองถูกแบ่งออกเป็น 2 ส่วนตามการทดลอง คือการทดลองโดยใช้การจำลองการทำงานของวงจรกรองที่ออกแบบด้วยภาษา VHDL ผ่านโปรแกรม ModelSim SE 6.5 และผลการทดลองการทำงานของวงจรกรองที่สร้างเป็นฮาร์ดแวร์บน FPGA ผ่านดิจิทัลออสซิลโลสโคป

3.4.1 วงจรบวก

วงจรบวกเป็นวงจรที่ใช้ในการรวมค่าสัมประสิทธิ์ ประกอบไปด้วยข้อมูลทางด้านอินพุต 2 ขา และทางด้านเอาต์พุต 1 ขา สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.9

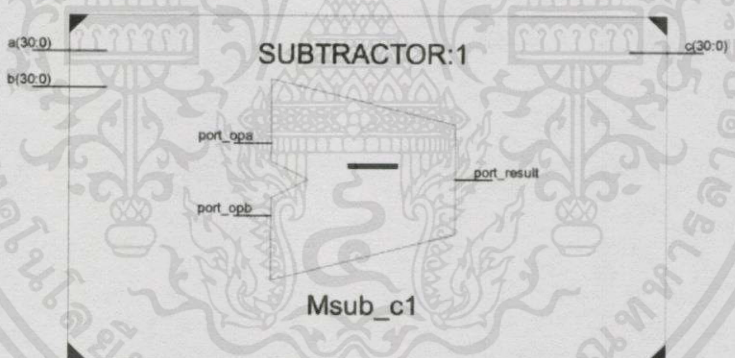
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ผลของการสังเคราะห์วงจรบวก

3.4.2 วงจรลบ

วงจรถลบเป็นวงจรที่ใช้ในการลบค่าสัมประสิทธิ์ ประกอบไปด้วยข้อมูลทางด้านอินพุต 2 ขา และทางด้านเอาต์พุต 1 ขา สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.10

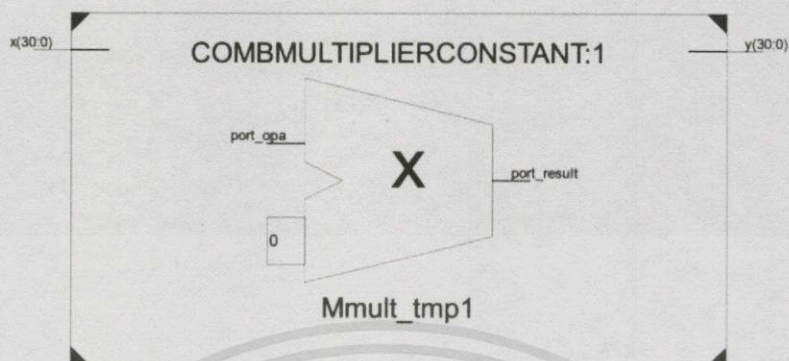


รูปที่ 3.10 ผลของการสังเคราะห์วงจรถลบ

3.4.3 วงจรคูณ

วงจรถคูณเป็นวงจรที่ใช้ในการคูณค่าสัมประสิทธิ์ ประกอบไปด้วยข้อมูลทางด้านอินพุต 1 ขา และทางด้านเอาต์พุต 1 ขา สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.11

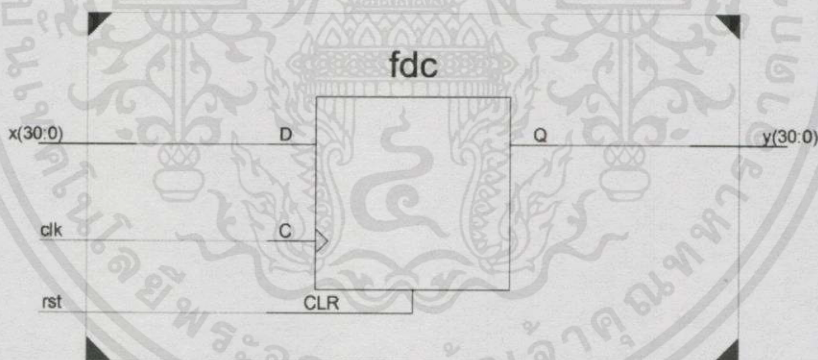
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ผลของการสังเคราะห์วงจรคูณ

3.2.4 วงจรหน่วงเวลา

วงจรหน่วงเวลาเป็นวงจรที่ใช้ในการหน่วงเวลาไปหนึ่งหน่วยเวลา ประกอบด้วยข้อมูลทางด้านอินพุต 1 ขา ทางด้านเอาต์พุต 1 ขา และมีสัญญาณควบคุม clk และ rst สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.12

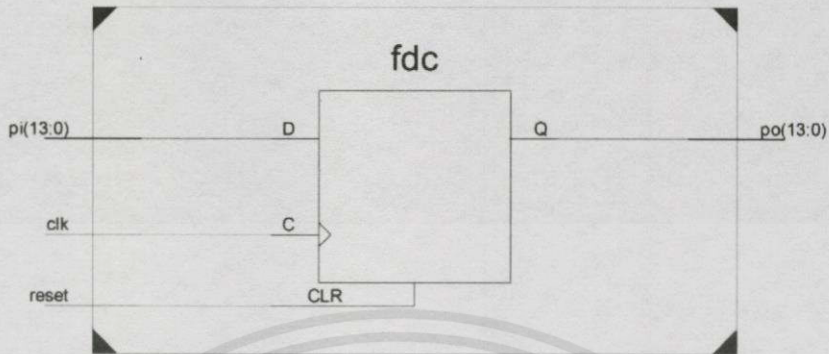


รูปที่ 3.12 ผลของการสังเคราะห์วงจรหน่วงเวลา

3.1.5 วงจร PIPO (Parallel In – Parallel Out)

วงจร PIPO เป็นโมดูลของวงจรที่เสมือนนำ D Flip Flop มาต่อกัน และทำหน้าที่ Latch ค่าข้อมูลต่างๆ สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.13

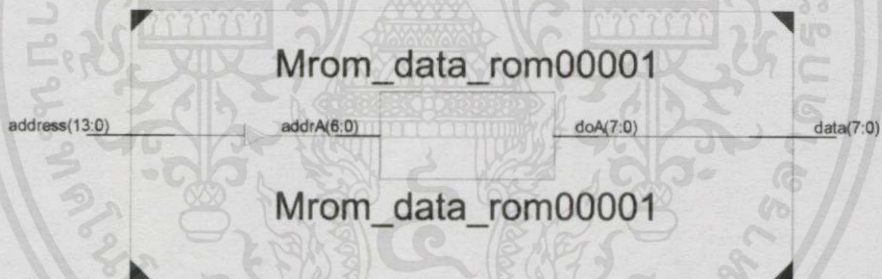
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ผลของการสังเคราะห์วงจร FIFO

3.1.6 วงจร look up table

วงจร look up table เป็นวงจรที่ทำหน้าที่เก็บค่าแอมพลิจูดของสัญญาณที่ตำแหน่งแอดเดรสต่างๆ สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.14



รูปที่ 3.14 ผลของการสังเคราะห์วงจร look up table

3.1.7 วงจรนับ (Counter)

วงจรถับเป็นวงจรที่ทำหน้าที่นับวนจนเข้าถึงค่าที่กำหนด โดยมีขา clk เป็นขาที่ใช้ในการเข้าจังหวะการนับ สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 ผลของการสังเคราะห์วงจร Counter

3.1.8 วงจรมัลติเพล็กซ์เซอร์ (Multiplexer)

วงจร Multiplexer เป็นวงจรที่ทำหน้าที่รวมสัญญาณข้อมูลเข้าด้วยกัน ในที่นี้คือการนำสัญญาณที่ทำการหน่วงไว้มาทำการแทรกกับสัญญาณเดิมเพื่อให้ได้สัญญาณที่มีความละเอียดมากยิ่งขึ้น โดยมีขา s เป็นขา select ซึ่งจะเลือกว่าจะเอาสัญญาณข้อมูลที่ขาใดเข้าก่อน ขาใดเข้าตามมา เป็นต้น สามารถเขียนแทนด้วยสัญลักษณ์ดังรูปที่ 3.16



รูปที่ 3.16 ผลของการสังเคราะห์วงจร Multiplexer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

สาเหตุในการเลือกใช้วงจรรอสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดอันดับที่ 2 คือผลของประสิทธิภาพในการทำงานของวงจรเมื่อเทียบกับจำนวนของอุปกรณ์ที่ใช้ ที่ทั้งทำให้ได้ผลตามที่ต้องการและไม่ใช้อุปกรณ์จำนวนมากเกินไป ซึ่งผลของการวัดประสิทธิภาพการทำงานของวงจรทั้งอันดับ 1 และ 2 โดยละเอียดจะแสดงในภาคผนวก โดยเป็นผลการทดลองในภาคเรียนที่ 1

อีกทั้งการเลือกค่าความหน่วงยังต้องสอดคล้องกับอันดับของวงจรรอส เพื่อให้ผลลัพธ์มีเสถียรภาพ และมีโพลอยู่บนวงกลมหนึ่งหน่วย

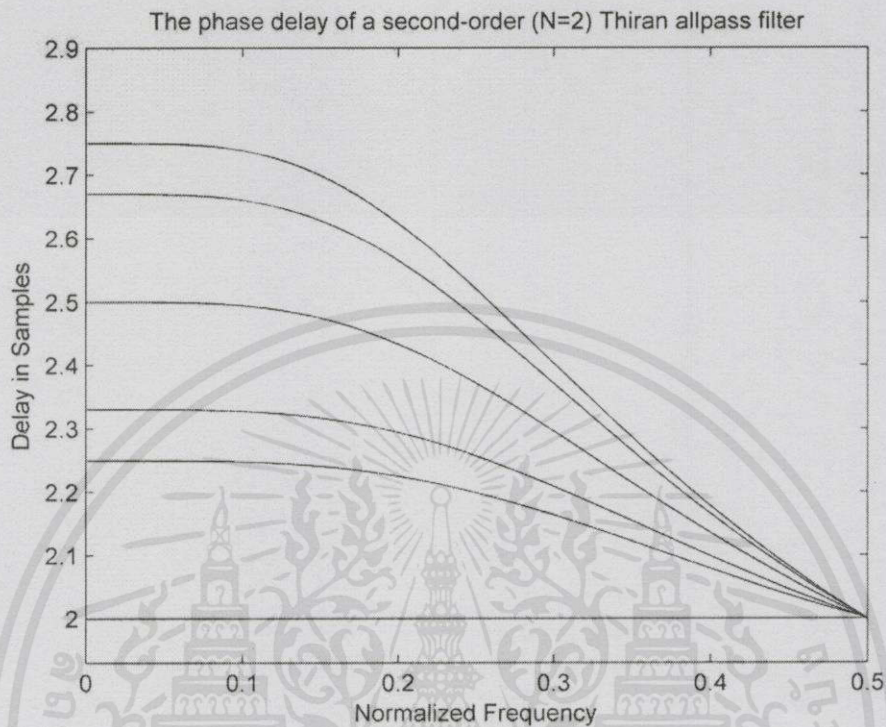
ในบทนี้จะแบ่งผลการทดลองทั้งหมดออกเป็นสามส่วนคือ ผลการจำลองซอฟต์แวร์ผ่านโปรแกรมแมตแล็บ ผลการจำลองฮาร์ดแวร์ด้วยภาษา VHDL และผลการทำงานจริงทางฮาร์ดแวร์ที่ได้จากเครื่อง Oscilloscope ซึ่งสามารถแบ่งผลการทดลองออกเป็นหัวข้อต่างๆ ได้ดังนี้

4.1 คุณสมบัติของวงจรรอสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไธรา

4.1.1 ผลตอบสนองทางเฟสจากการจำลองทางซอฟต์แวร์

กราฟผลตอบสนองทางเฟสของตัวกรองแสดงให้เห็นว่าตัวกรองอันดับที่ 2 ไม่สามารถทำการหน่วงสัญญาณแบบเป็นเศษส่วนได้ในย่านความถี่สูงๆ เนื่องจากค่าการหน่วงสัญญาณจะมีความผิดเพี้ยนเกิดขึ้น โดยผลตอบสนองทางเฟสแสดงได้ดังรูปที่ 4.1

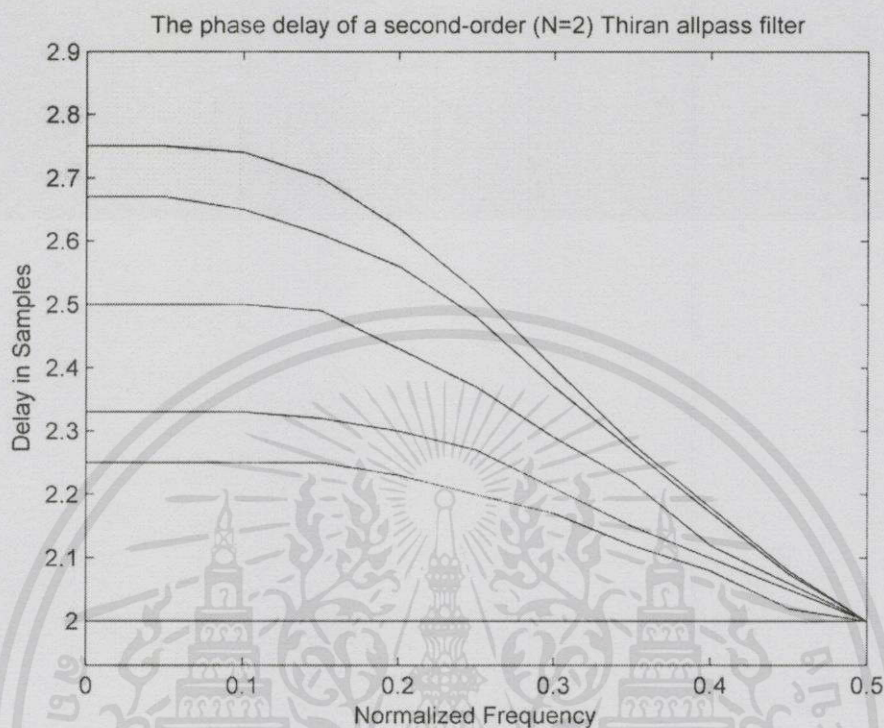
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ผลตอบสนองทางเฟสของตัวกรองอันดับที่ 2 ที่ได้จากโปรแกรมแมตแล็บ

4.1.2 ผลตอบสนองทางเฟสจากการจำลองทางฮาร์ดแวร์

ผลของคุณสมบัติที่ได้จากการออกแบบทางฮาร์ดแวร์ของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนไรรานมีลักษณะใกล้เคียงกับผลการจำลองจากโปรแกรมแมตแล็บ อย่างไรก็ตาม ผลที่ได้อาจจะมีความคลาดเคลื่อน เนื่องจากจำนวนบิตที่ใช้กับวงจรมีค่า 8 บิต ผลการคำนวณจึงไม่ละเอียดเท่าที่ควร ซึ่งสามารถแสดงผลตอบสนองทางเฟสได้ดังรูปที่ 4.2



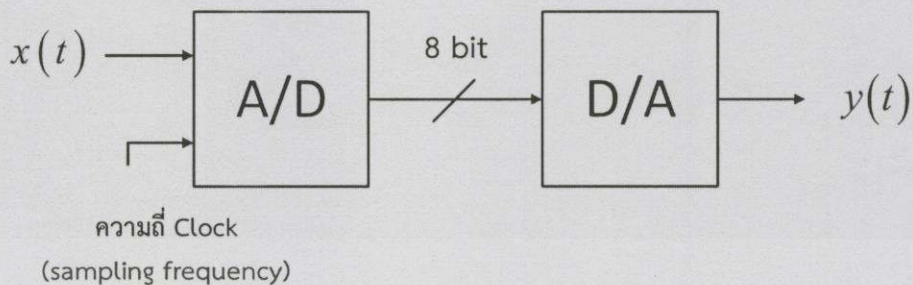
รูปที่ 4.2 ผลตอบสนองทางเฟสของตัวกรองอันดับที่ 2 ที่ได้จากการจำลองฮาร์ดแวร์

จากผลการวัดคุณสมบัติของวงจรกรองที่ได้ออกแบบขึ้น ทำให้ทราบว่าวงจรจะให้ผลการหน่วงสัญญาณตรงตามค่าตัวอย่างในการหน่วงที่ต้องการอยู่ในช่วงประมาณไม่เกิน 0.15 เท่าของความถี่การซีกตัวอย่าง Normalize Frequency ทำให้เกิดข้อจำกัดของการทำงานของวงจรกรองขึ้นแต่ถึงอย่างไร ข้อจำกัดที่เกิดขึ้นนี้ยังคงเป็นไปตามทฤษฎีของวงจรกรองแบบผ่านตลอดไครานที่ความถูกต้องของสัญญาณผลลัพธ์จะมีมากในช่วงความถี่ต่ำ

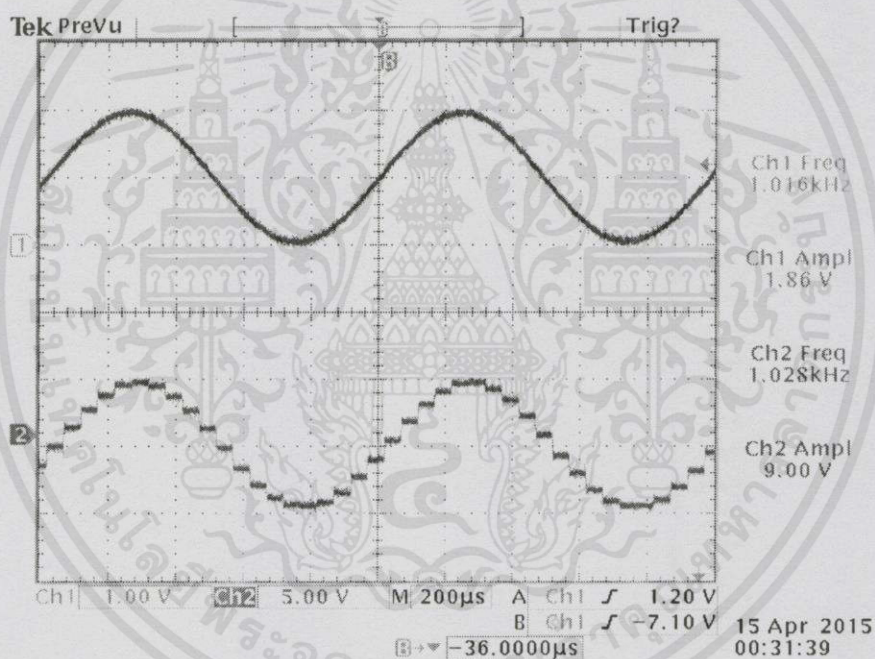
4.2 การทำ Oversampling D/A

Oversampling D/A คือการนำสัญญาณที่มีความละเอียดน้อยไปทำการประมวลผลสัญญาณทางดิจิทัลเพื่อให้ได้เอาต์พุตออกมามีความละเอียดสูงขึ้น หรือการทวิคูณ sampling frequency ซึ่งวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดไครานสามารถนำมาประยุกต์ใช้งานทางด้านนี้ได้ เพื่อให้เห็นสาเหตุ ปัญหาที่เกิดขึ้นและเหตุผลที่ต้องทำ Oversampling D/A สามารถแสดงการทำงานได้ดังรูปที่ 4.3 โดยใช้ sampling frequency เท่ากับ 20 kHz และผลการทดลองสัญญาณคลื่นไซน์ที่แต่ละความถี่ตามรูปบล็อกไดอะแกรมที่ 4.3

ไม่ว่ากรณีใดก็สามารถแสดงได้ดังรูปที่ 4.4-4.6 เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

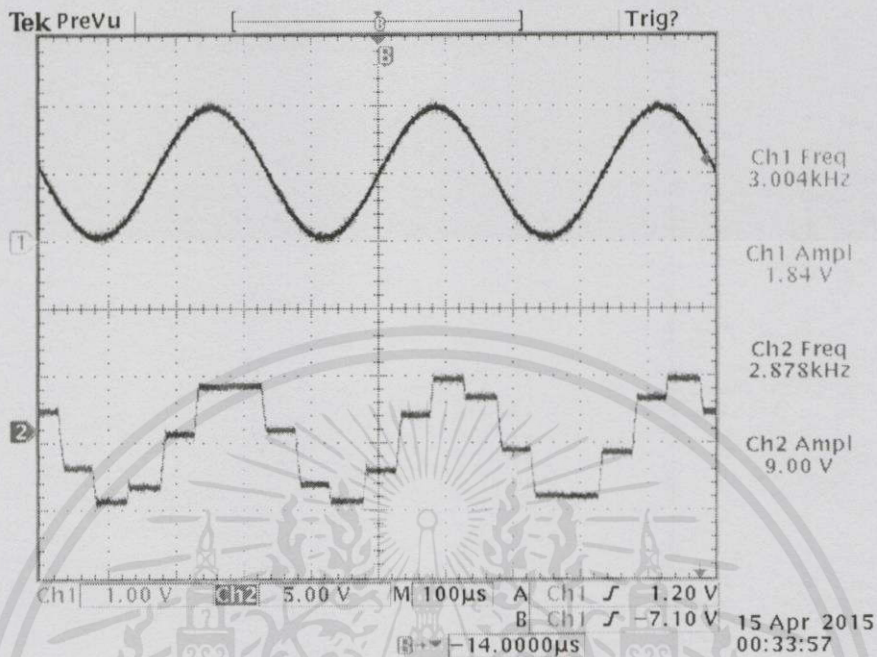


รูปที่ 4.3 บล็อกไดอะแกรมปัญหาของการทดลองที่เกิดขึ้น

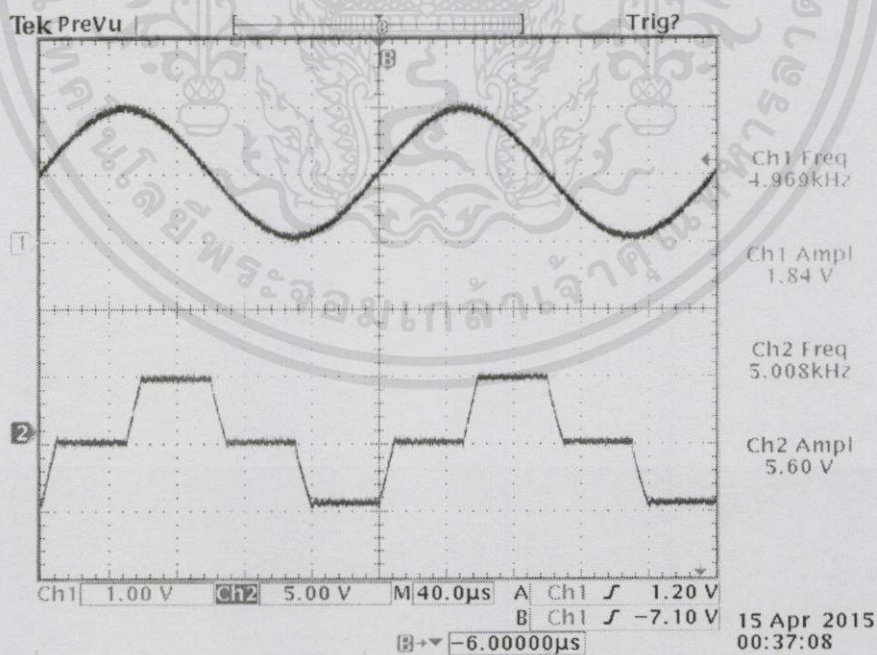


รูปที่ 4.4 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 1 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีนำไปใช้

รูปที่ 4.6 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 5 kHz ตามรูปบล็อกไดอะแกรมที่ 4.3

จากการทดลองนี้ความถี่ sampling rate หรือ sampling frequency จะถูกกำหนดค่าไว้ที่ 20 kHz และทำการปรับค่าความถี่ของสัญญาณอินพุตที่ 1 kHz 3 kHz และ 5 kHz เพื่อสังเกตผลที่เกิดขึ้นตามรูปบล็อกไดอะแกรมที่ 4.7 ซึ่งตามกฎของ Nyquist คือความถี่ sampling rate จะต้องมากกว่าหรือเท่ากับ 2 เท่าของความถี่สัญญาณอินพุต แต่ในความเป็นจริงแล้วจากผลการทดลองจะเห็นว่าเมื่อปรับความถี่ไปที่ 1 kHz สัญญาณเอาต์พุตที่ออกมามีลักษณะของสัญญาณไซน์ที่เป็นแบบขั้นบันได ดังนั้นเราจึงสามารถนำวงจรกรองสัญญาณปรับค่าความถี่เป็นเศษส่วนแบบผ่านตลอดโรรานมาประยุกต์ใช้งานในการแก้ไขปัญหาดังกล่าวได้ ซึ่งคือการทำ Oversampling D/A รวมทั้งการแปลงอัตราการซักรด้วยอย่างจาก 44.1 kHz เป็น 48 kHz



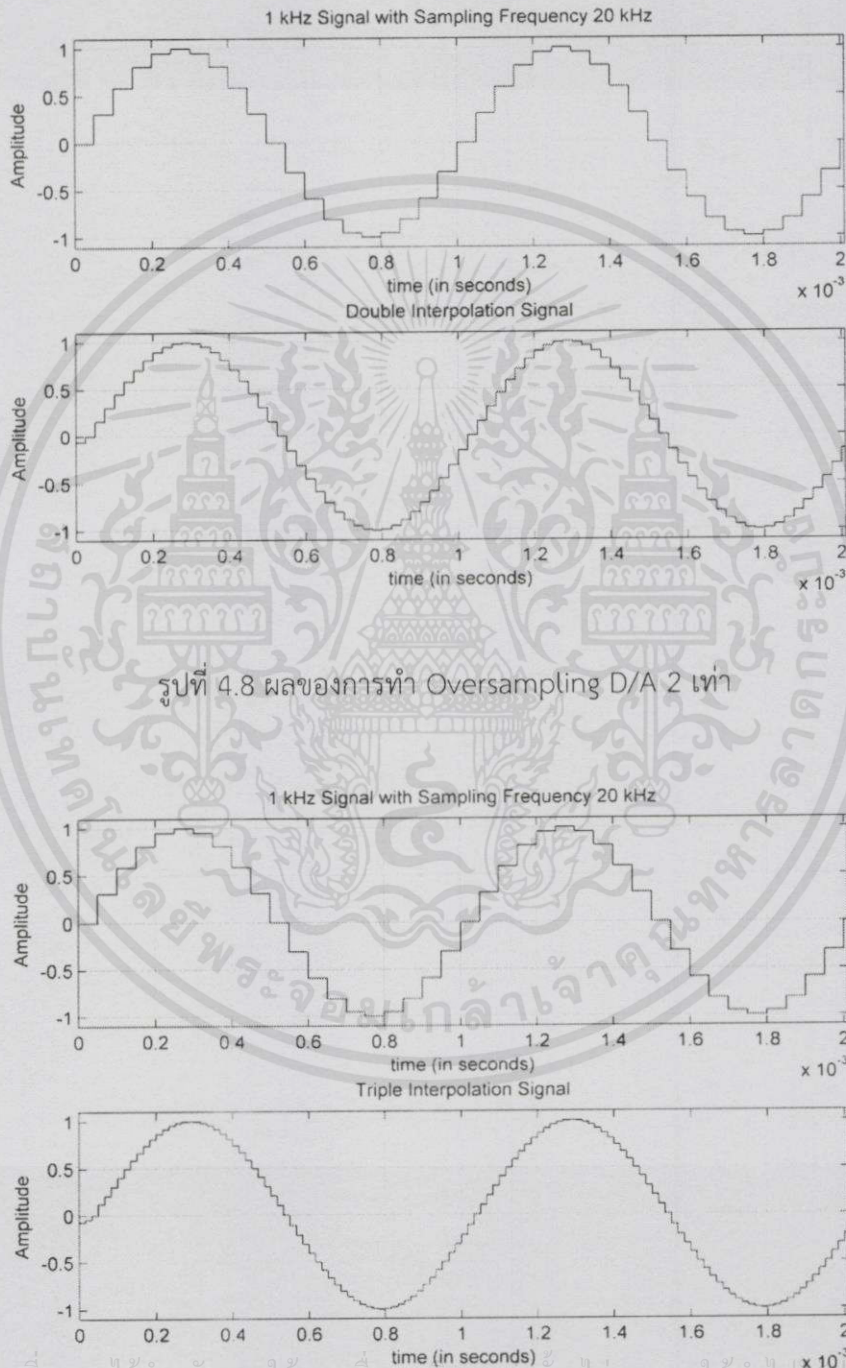
รูปที่ 4.7 บล็อกไดอะแกรมการทำงานของ Oversampling D/A

4.3 การจำลองการทำงานทางซอฟต์แวร์ในการทำ Oversampling D/A

การจำลองการทำงานทางฮาร์ดแวร์จะใช้โปรแกรม MATLAB ในการจำลองการทำงาน โดยเริ่มจากการสร้างสัญญาณคลื่นไซน์อินพุตที่ความถี่ 1 kHz, 3 kHz, 5 kHz, และ sampling frequency เท่ากับ 20 kHz โดยใช้วงจรกรองสัญญาณปรับค่าความถี่เป็นเศษส่วนแบบผ่านตลอดโรรานและทำการ Oversampling D/A 2 เท่า 3 เท่า และ 4 เท่า ดังรูปที่ 4.8 – 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 ผลจำลองการทำงานทางซอฟต์แวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 1 kHz

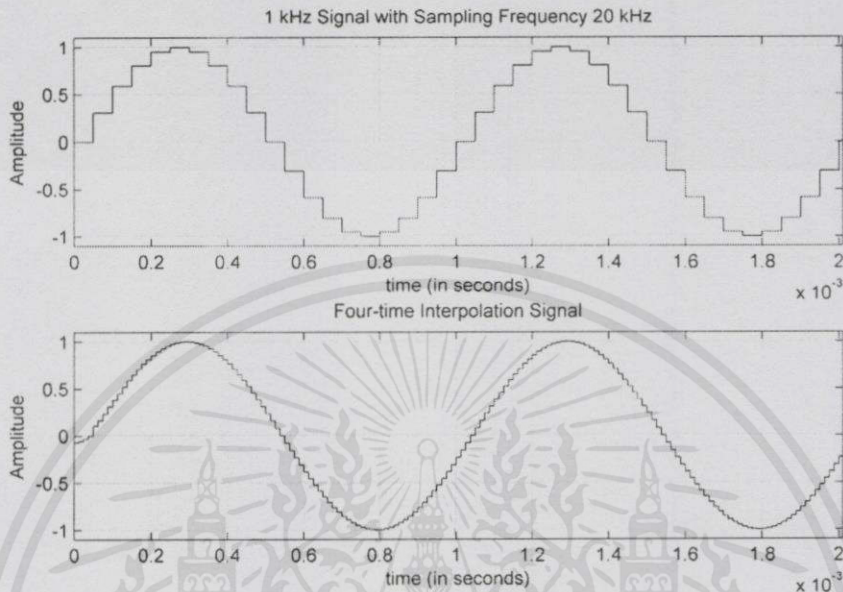


รูปที่ 4.8 ผลของการทำ Oversampling D/A 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

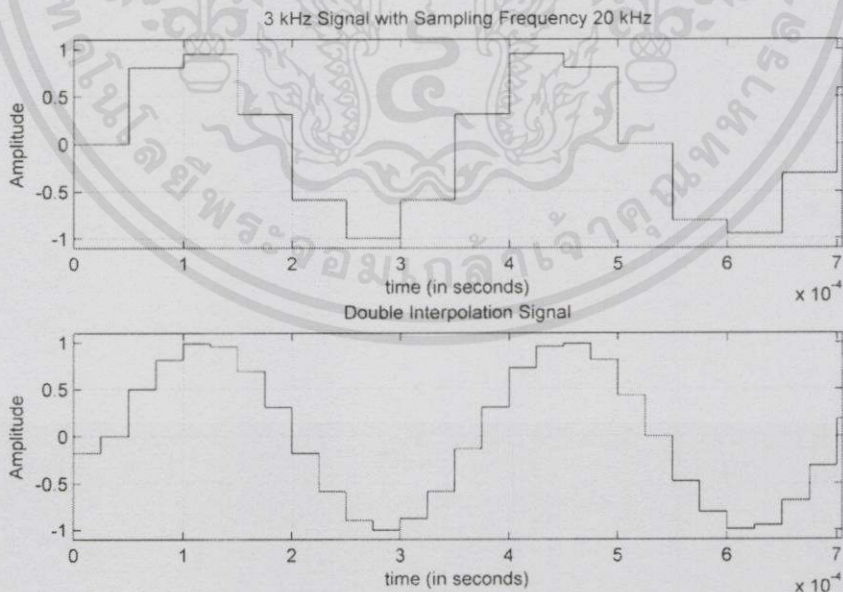
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 ผลของการทำ Oversampling D/A 3 เท่า

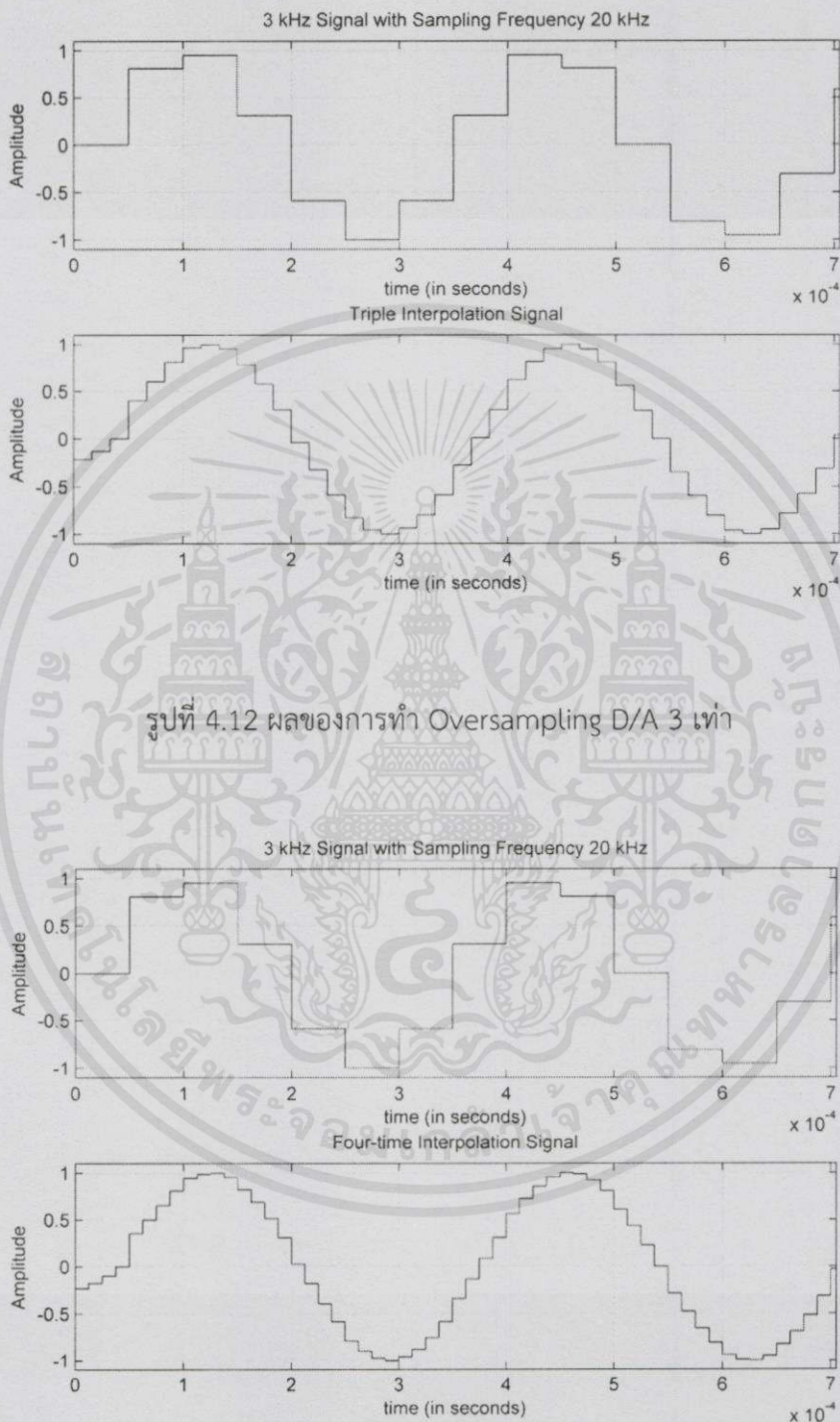


รูปที่ 4.10 ผลของการทำ Oversampling D/A 4 เท่า

4.3.2 ผลจำลองการทำงานทางซอฟต์แวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 3 kHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 4.11 ผลของการทำ Oversampling D/A 2 เท่า ทุกครั้งที่มีการนำไปใช้

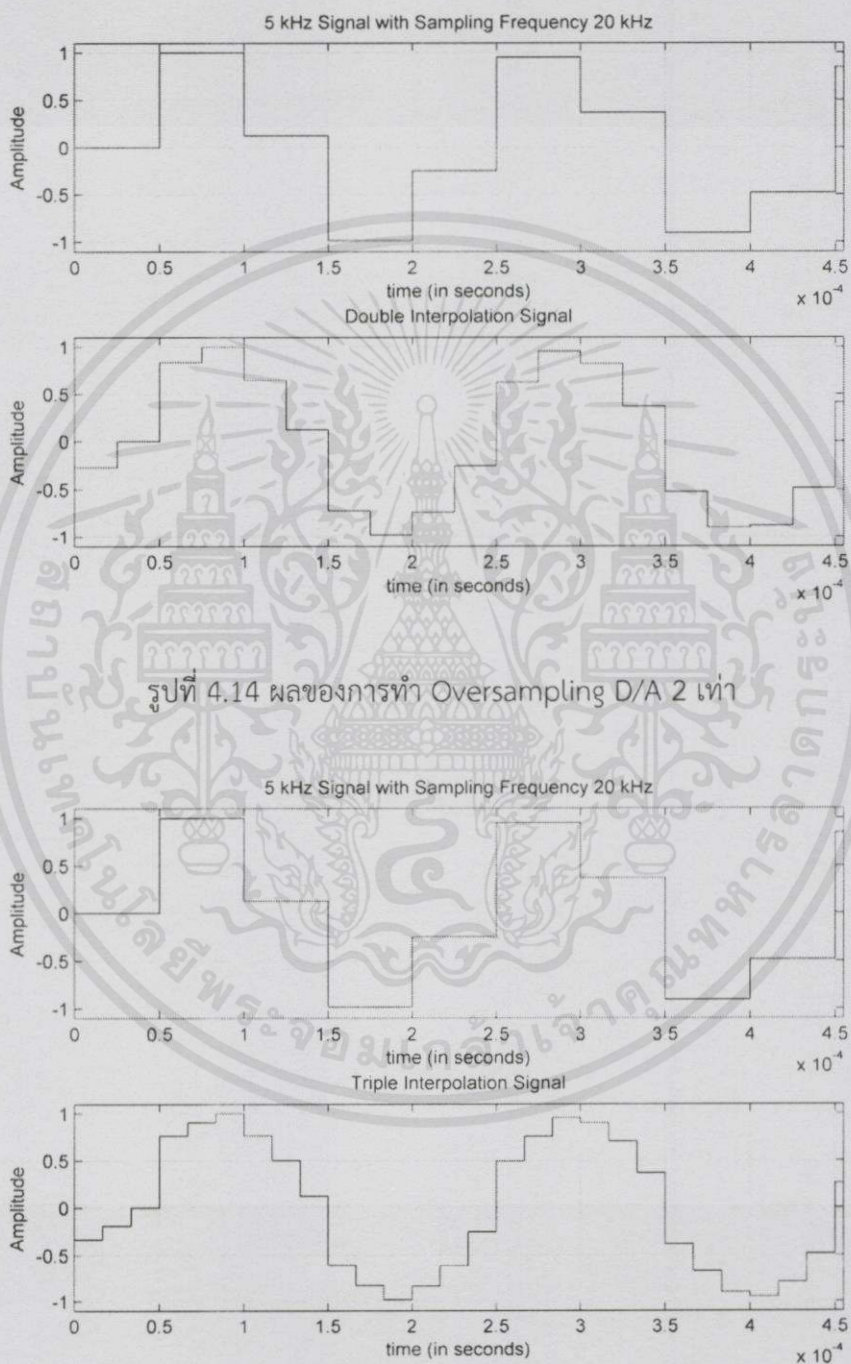


รูปที่ 4.12 ผลของการทำ Oversampling D/A 3 เท่า

รูปที่ 4.13 ผลของการทำ Oversampling D/A 4 เท่า

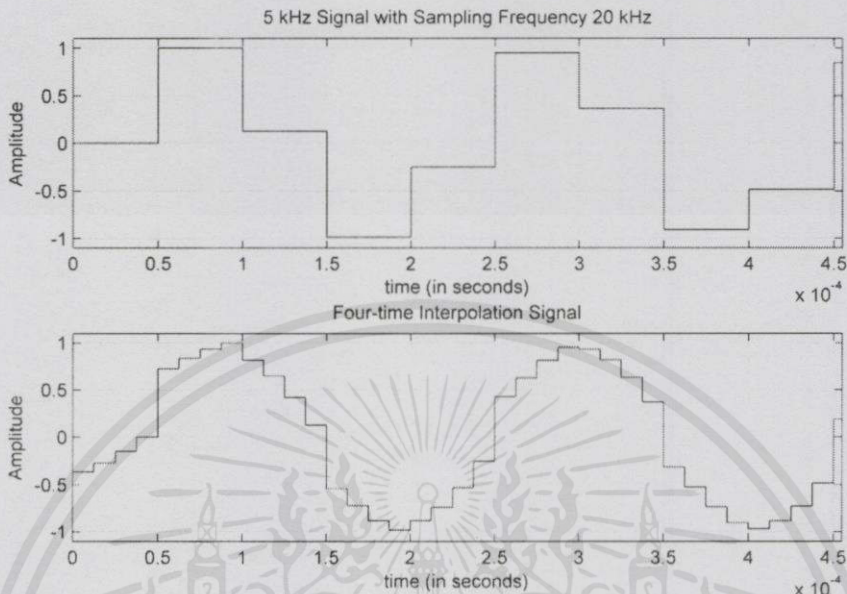
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงบนสื่ออิเล็กทรอนิกส์ใดๆทั้งปวงโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลจำลองการทำงานทางซอฟต์แวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 5 kHz



รูปที่ 4.14 ผลของการทำ Oversampling D/A 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.15 ผลของการทำ Oversampling D/A 3 เท่า
 ไม่ว่าจะพิมพ์ใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลของการทำ Oversampling D/A 4 เท่า

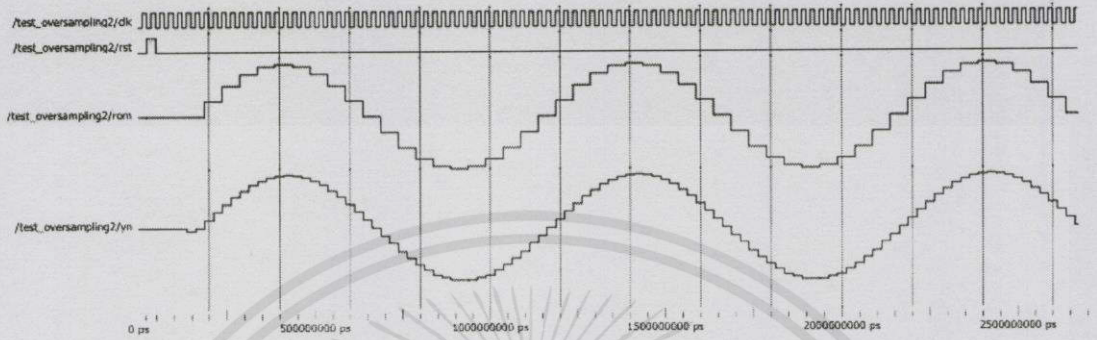
ด้านบนของแต่ละรูปคือสัญญาณอินพุตที่มีความละเอียดของสัญญาณค่อนข้างต่ำ ซึ่งเมื่อทำ Oversampling D/A แล้วจะเห็นว่าสัญญาณมีค่าความละเอียดที่เพิ่มมากขึ้นตามจำนวนเท่าที่ทำ Oversampling D/A ผลการจำลองนี้จึงยืนยันได้ว่าวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรรานสามารถนำมาประยุกต์ใช้งานได้จริงในการทำ Oversampling D/A

4.4 ผลการจำลองการทำงานทางฮาร์ดแวร์ในการทำ Oversampling D/A

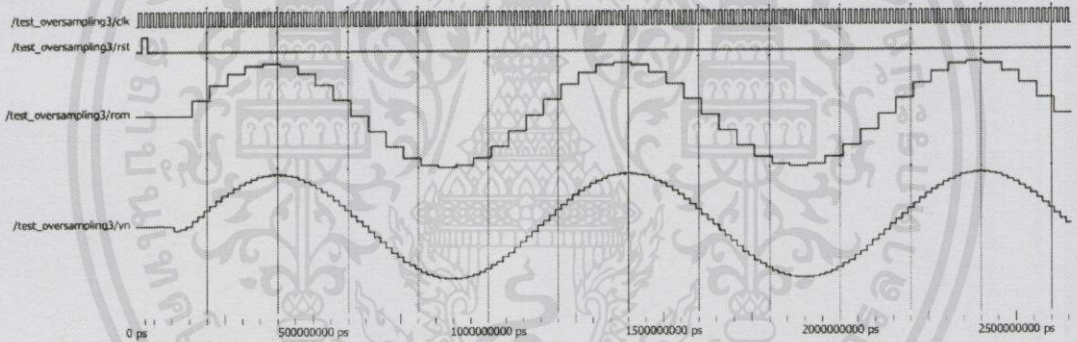
การจำลองการทำงานทางฮาร์ดแวร์จะใช้โปรแกรม ModelSim SE 6.5 ที่ใช้ภาษา VHDL ในการออกแบบอุปกรณ์และสังเคราะห์วงจรทางฮาร์ดแวร์ขึ้นเพื่อทำการนำไปใช้บนอุปกรณ์ FPGA ให้สามารถใช้งานวงจรได้จริง โดยเริ่มจากการสร้างสัญญาณรูปไซน์เป็นสัญญาณขาเข้าที่มีความถี่ 1 kHz, 3 kHz, 5 kHz และมี Sampling frequency เท่ากับ 20 kHz เพื่อให้เห็นถึงผลของการจำลองการทำงานว่าสัญญาณนั้นมีความละเอียดที่เพิ่มขึ้นจริงหรือไม่ โดยใช้วงจรกรอง

สัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรรานและทำการ Oversampling D/A 2 เท่า 3 เท่า 4 เท่า และการแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz ดังรูปที่ 4.17 - 4.26 ทั้งนี้ อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

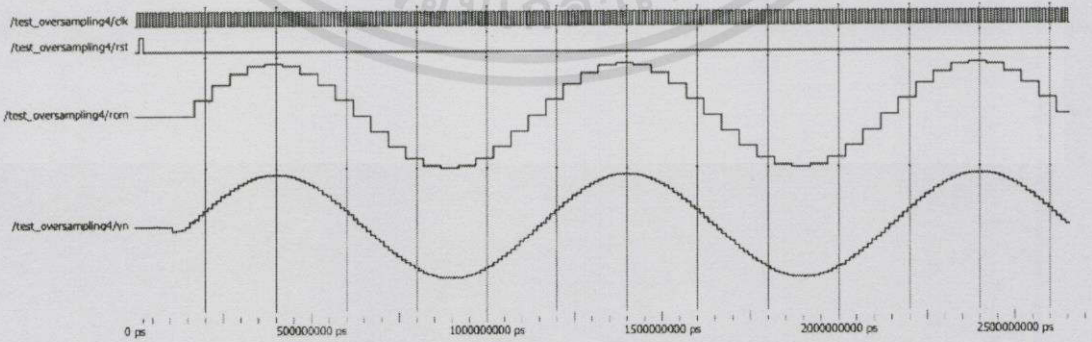
4.4.1 ผลจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 1 kHz



รูปที่ 4.17 ผลของการทำ Oversampling D/A 2 เท่า

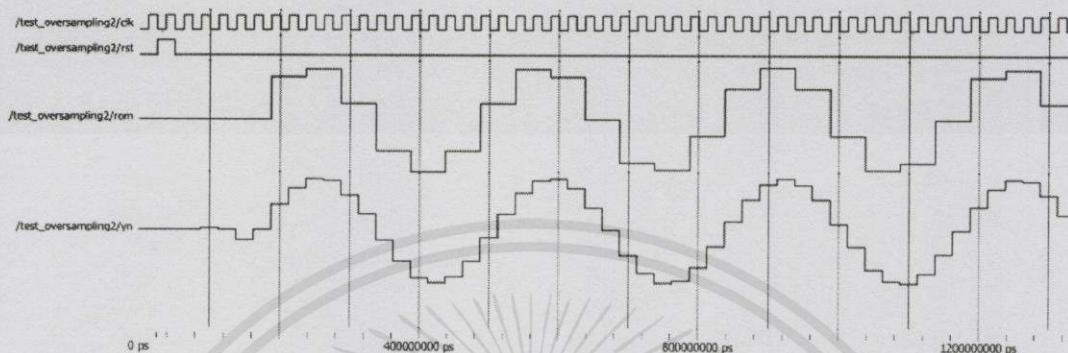


รูปที่ 4.18 ผลของการทำ Oversampling D/A 3 เท่า

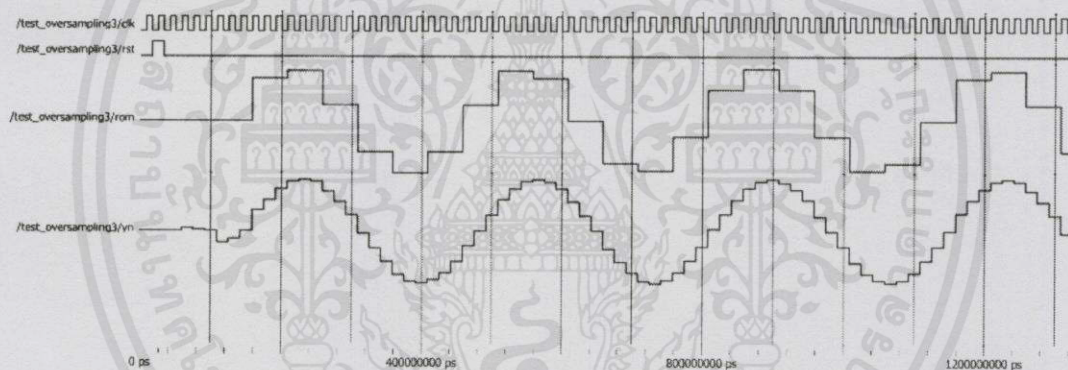


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 4.19 ผลของการทำ Oversampling D/A 4 เท่า ทุกครั้งที่มีการนำไปใช้

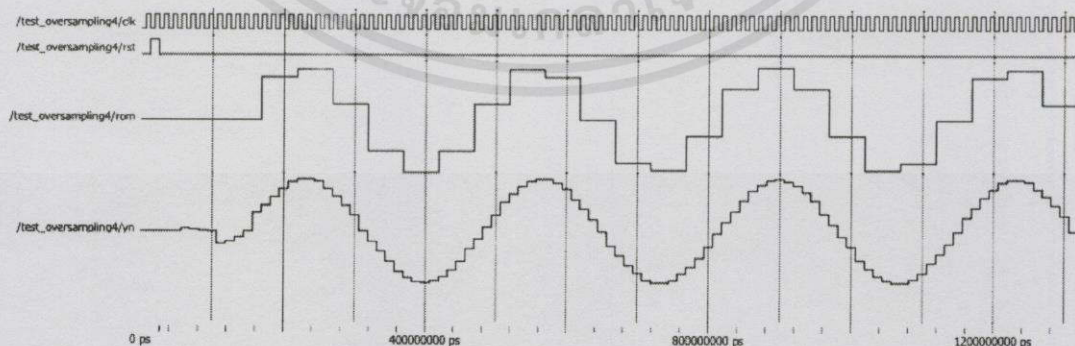
4.4.2 ผลจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz



รูปที่ 4.20 ผลของการทำ Oversampling D/A 2 เท่า

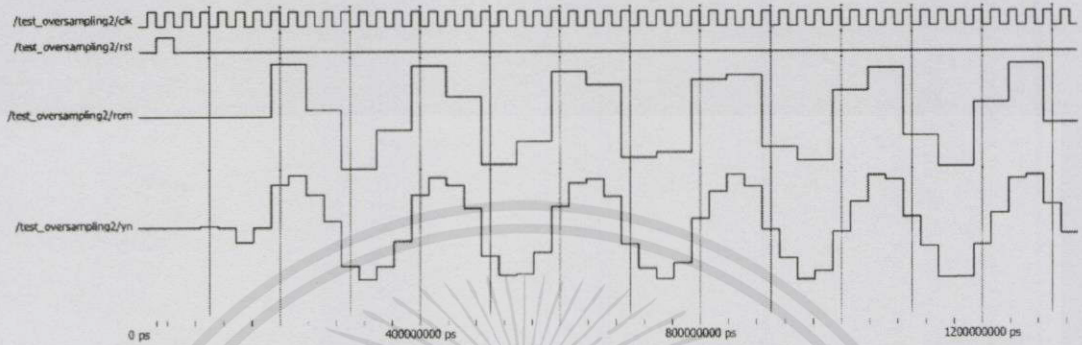


รูปที่ 4.21 ผลของการทำ Oversampling D/A 3 เท่า

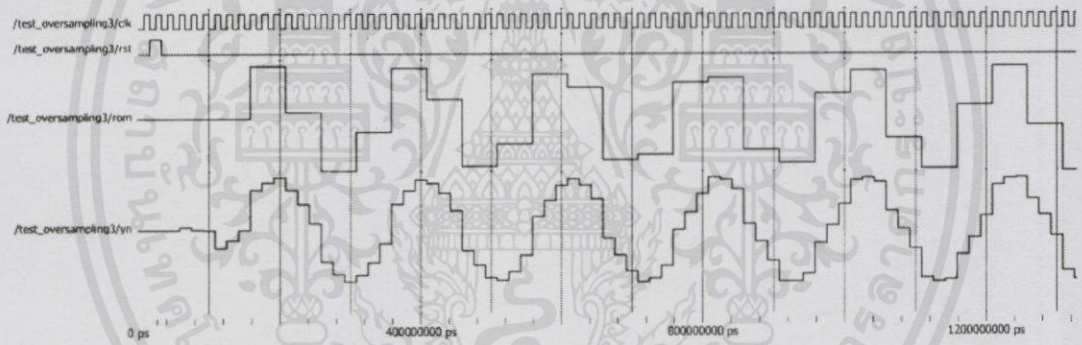


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 4.22 ผลของการทำ Oversampling D/A 4 เท่า ทุกครั้งที่มีการนำไปใช้

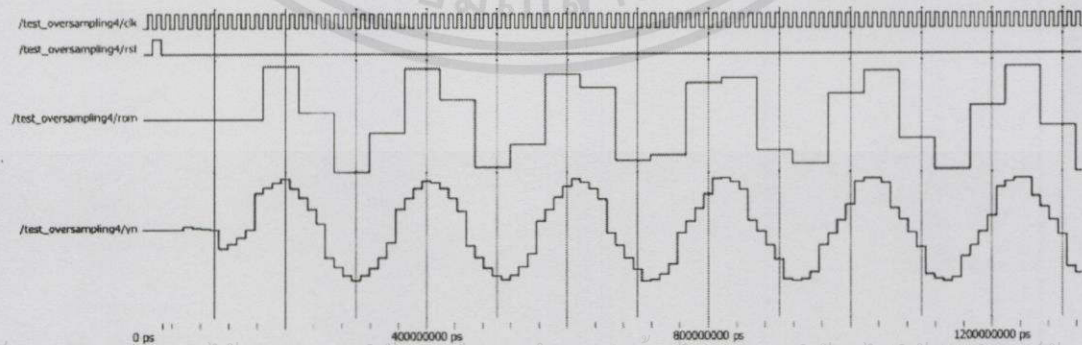
4.4.3 ผลจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 5 kHz



รูปที่ 4.23 ผลของการทำ Oversampling D/A 2 เท่า



รูปที่ 4.24 ผลของการทำ Oversampling D/A 3 เท่า

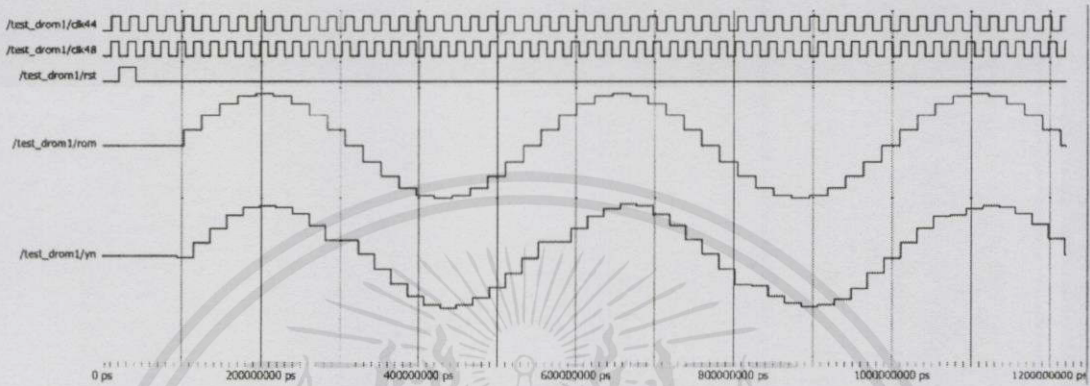


รูปที่ 4.25 ผลของการทำ Oversampling D/A 4 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำข้อมูลหรือเนื้อหาใดๆไปเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต

4.4.4 ผลจำลองการทำงานทางฮาร์ดแวร์ของการแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz



รูปที่ 4.26 ผลของการแปลงอัตราการซีกตัวอย่าง

ผลที่ได้จะมีลักษณะที่ใกล้เคียงกับการจำลองทางซอฟต์แวร์ด้วยโปรแกรม MATLAB ซึ่งผลการจำลองการทำงานทางฮาร์ดแวร์นี้จะแสดงให้เห็นถึงการออกแบบและการสังเคราะห์วงจรที่ทำงานถูกต้อง และสามารถนำมาประยุกต์ใช้งานทางด้านการทำ Oversampling D/A และการแปลงอัตราการซีกตัวอย่างได้จริง

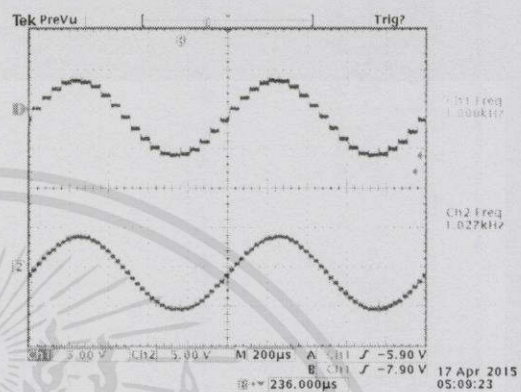
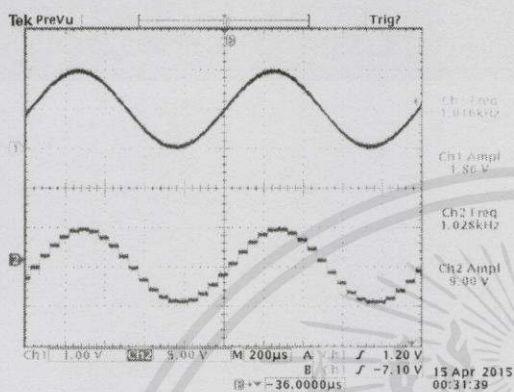
4.5 ผลการทำงานจริงทางฮาร์ดแวร์ในการทำ Oversampling D/A

ผลการทำงานจริงคือผลที่ได้จากออสซิลโลสโคป ซึ่งจะทำการป้อนสัญญาณรูปไซน์ที่มีความถี่ที่กำหนดจากอุปกรณ์กำเนิดสัญญาณ (Function generator) เข้าไปยังวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล จากนั้นเป็นการประมวลผลสัญญาณทางดิจิตอลด้วยอุปกรณ์ที่ออกแบบและสังเคราะห์ขึ้น สุดท้ายนำสัญญาณที่ทำการประมวลผลแล้วไปแปลงกลับเป็นสัญญาณอนาล็อกเพื่อแสดงผลบนออสซิลโลสโคป ดังรูปที่ 4.27 - 4.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

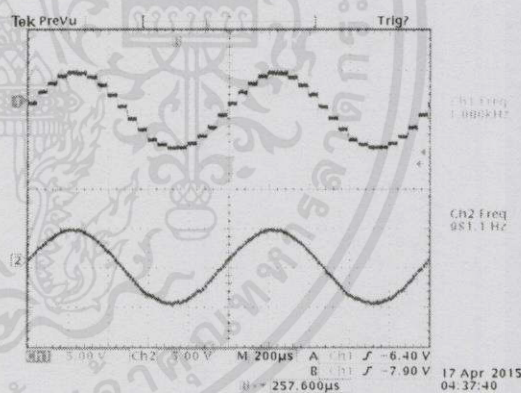
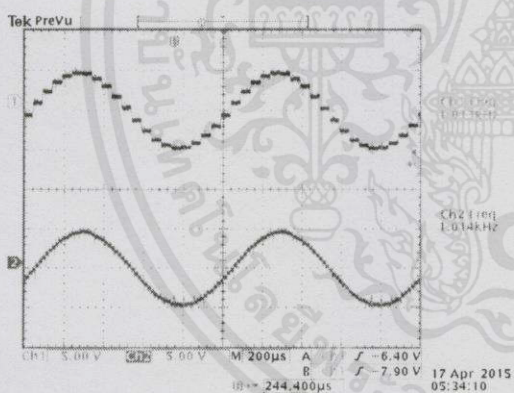
4.5.1 ผลจำลองการทำงานจริงทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 1

kHz



ก).ผลที่ไม่มีการทำ Oversampling D/A

ข).ผลของการทำ Oversampling D/A 2 เท่า



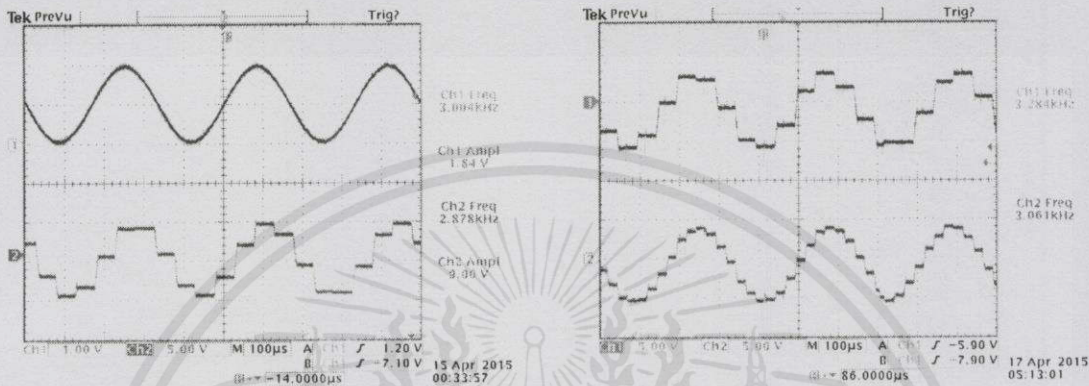
ค).ผลของการทำ Oversampling D/A 3 เท่า

ง).ผลของการทำ Oversampling D/A 4 เท่า

รูปที่ 4.27 ผลเปรียบเทียบสัญญาณเอาต์พุตที่มีความถี่ 1 kHz

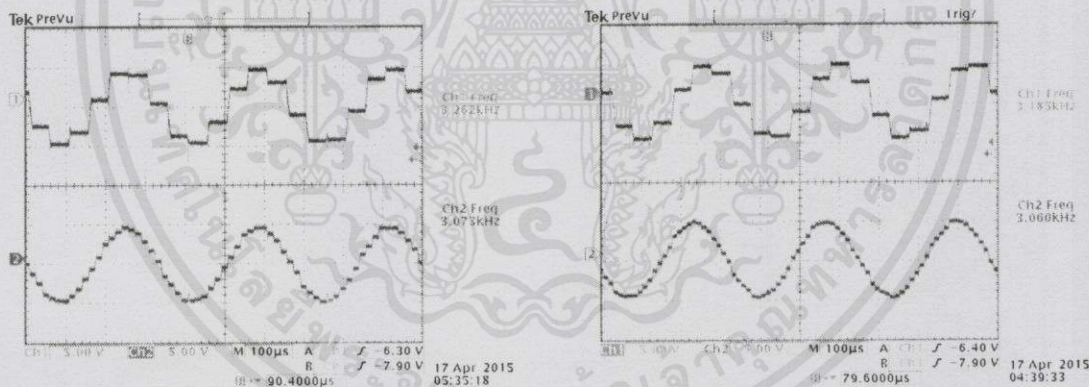
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 ผลจำลองการทำงานจริงทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่มีความถี่ 3 kHz



ก).ผลที่ไม่มีการทำ Oversampling D/A

ข).ผลของการทำ Oversampling D/A 2 เท่า



ค).ผลของการทำ Oversampling D/A 3 เท่า

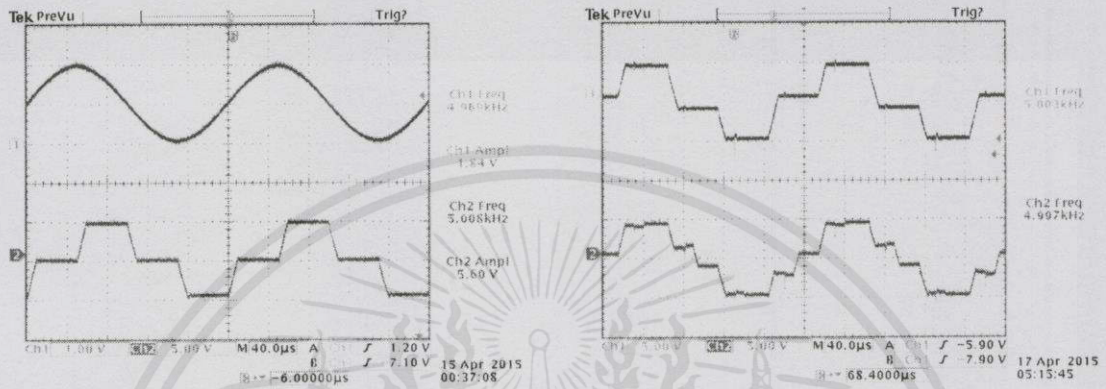
ง).ผลของการทำ Oversampling D/A 4 เท่า

รูปที่ 4.28 ผลเปรียบเทียบสัญญาณเอาต์พุตที่มีความถี่ 3 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

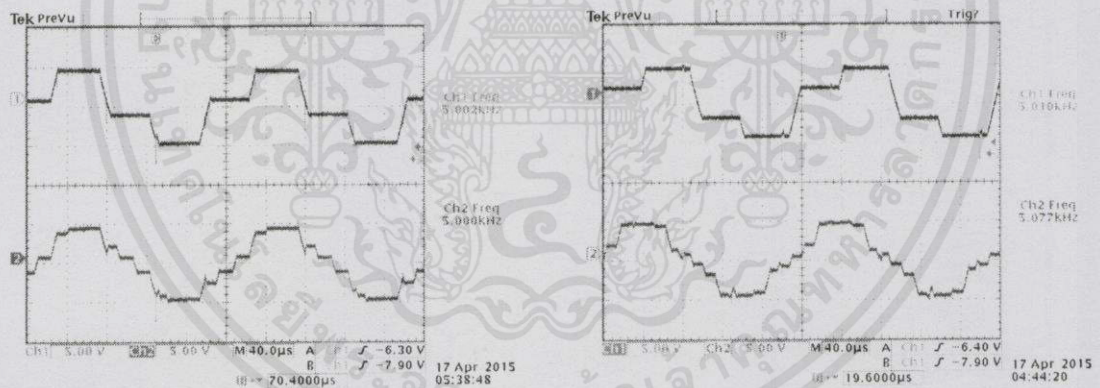
4.5.3 ผลจำลองการทำงานจริงทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 5

KHz



ก). ผลที่ไม่มีการทำ Oversampling D/A

ข). ผลของการทำ Oversampling D/A 2 เท่า



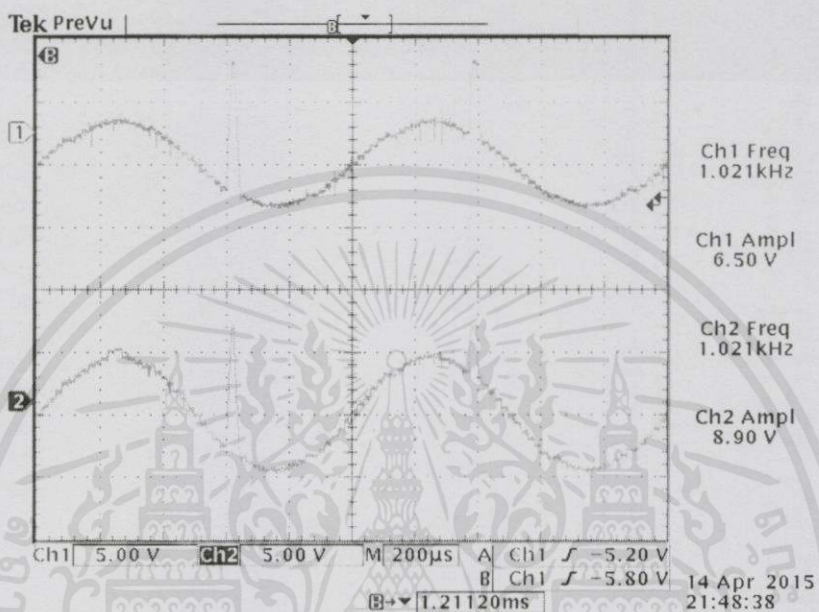
ค). ผลของการทำ Oversampling D/A 3 เท่า

ง). ผลของการทำ Oversampling D/A 4 เท่า

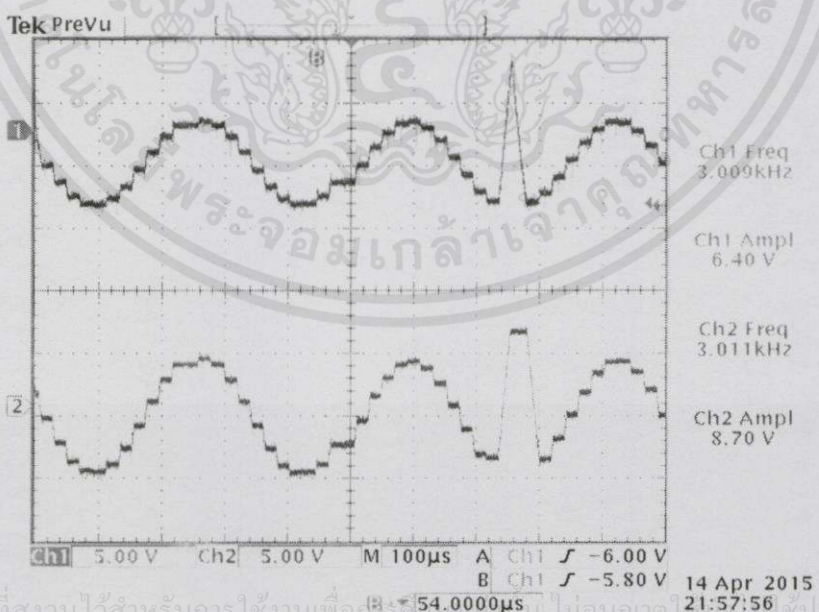
รูปที่ 4.29 ผลเปรียบเทียบสัญญาณเอาต์พุตที่ความถี่ 5 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.4 ผลจำลองการทำงานจริงทางฮาร์ดแวร์ของการแปลงอัตราการซีกตัวอย่าง จาก 44.1 kHz เป็น 48 kHz



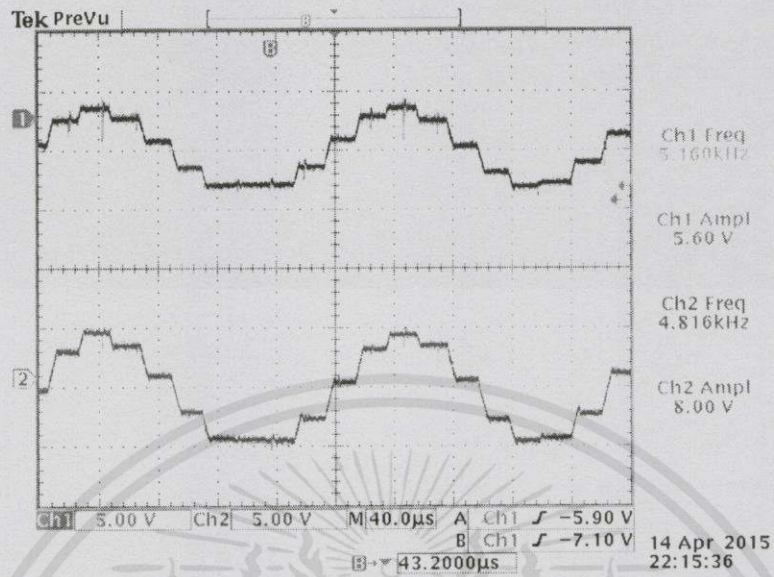
รูปที่ 4.30 ผลการทำงานของการทำงานการแปลงอัตราการซีกตัวอย่างที่มีความถี่ 1 kHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกผู้หนึ่งมีให้คำปรึกษาเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

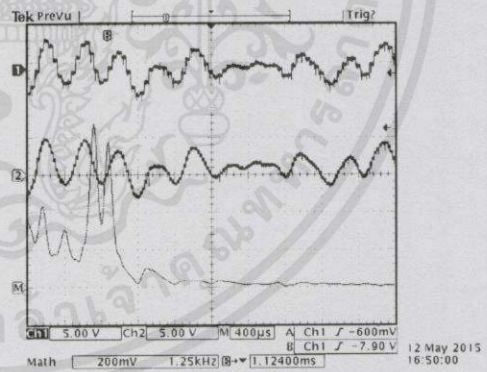
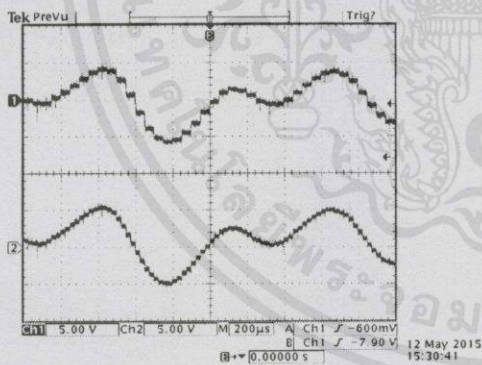
รูปที่ 4.31 ผลการทำงานของการทำงานการแปลงอัตราการซีกตัวอย่างที่มีความถี่ 3 kHz



รูปที่ 4.32 ผลการทำงานของ การแปลงอัตราการชักตัวอย่างที่ความถี่ 5 kHz

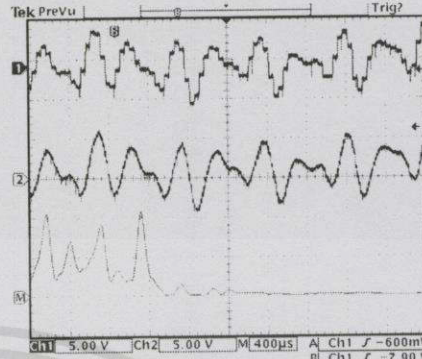
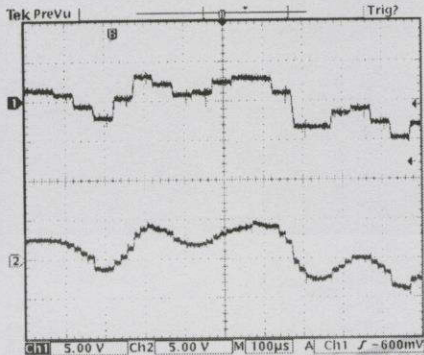
4.5.5 ผลจำลองการทำงานจริงทางฮาร์ดแวร์โดยใช้สัญญาณเสียงเป็นสัญญาณ

อินพุต

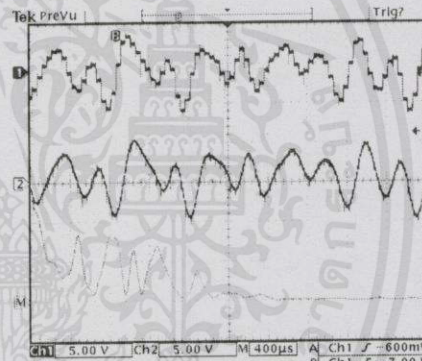
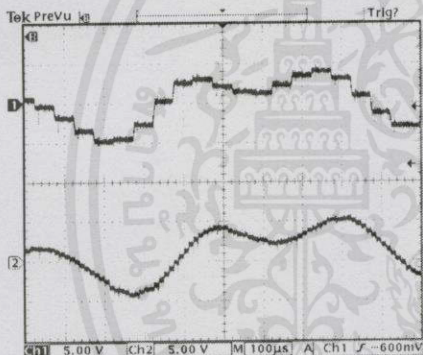


รูปที่ 4.33 ผลของการทำ Oversampling D/A 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 ผลของการทำ Oversampling D/A 3 เท่า

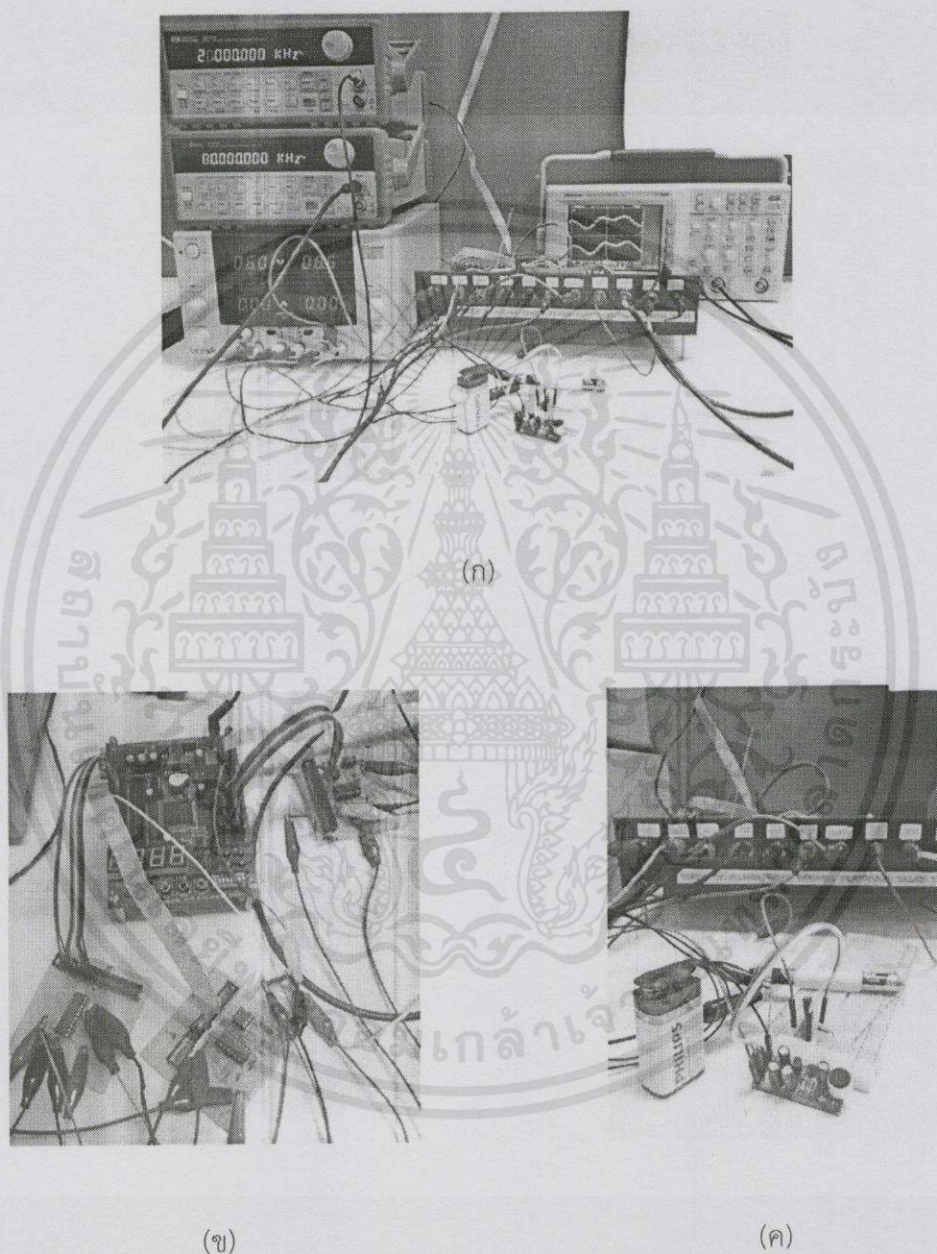


รูปที่ 4.35 ผลของการทำ Oversampling D/A 4 เท่า

จากการทำ Oversampling D/A ที่ความถี่ 1 kHz, 3 kHz และ 5 kHz จะเห็นว่าสัญญาณที่มีความถี่เข้าใกล้ครึ่งหนึ่งของความถี่ Nyquist สัญญาณที่ออกมา นั้นจะสูญเสียรายละเอียดของสัญญาณ ดังรูปที่ 4.26, 4.30 และ 4.34 แต่เมื่อทำ Oversampling D/A แล้วสัญญาณจะมีความละเอียดมากยิ่งขึ้นตามการทำ Oversampling D/A ซึ่งแสดงให้เห็นว่าวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโพรานสามารถนำมาประยุกต์ใช้ในการทำ Oversampling D/A และการแปลงอัตราการซึกตัวอย่างได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์การจำลองการทำงานจริงทางฮาร์ดแวร์สามารถแสดงได้ดังรูปที่ 4.36



รูปที่ 4.36 (ก) อุปกรณ์การจำลองการทำงานจริงทางฮาร์ดแวร์

(ข) บอร์ด FPGA , วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC)
 และวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (DAC)

(ค) วงจรไมโครโฟนและวงจรออปเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

วงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรรานสามารถนำมาปรับค่าความหน่วงได้ตามความต้องการ รวมทั้งค่าความหน่วงที่เป็นเศษส่วนโดยการปรับค่าดีเลย์พารามิเตอร์ แต่ค่าความหน่วงที่นำมาใช้จะต้องมีความสอดคล้องกับอันดับของตัวกรอง โดยตัวกรองที่ใช้เป็นวงจรรองสัญญาณดิจิทัลแบบผลตอบสนองอิมพัลส์ไม่จำกัด (Infinite impulse response filter: IIR filter) ทำให้ค่าสัมประสิทธิ์การคูณมีจำนวนไม่มากและมีความผิดพลาดเกิดขึ้นน้อย ในการนำมาประยุกต์ใช้งานสามารถนำมาประยุกต์ใช้ทางด้าน Oversampling D/A และการแปลงอัตราการซีกตัวอย่างได้ โดยผู้จัดทำเลือกใช้โครงสร้าง IS แบบปรับค่าได้อันดับ 2 สำหรับการแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz และใช้โครงสร้าง IS แบบกำหนดค่าอันดับ 2 สำหรับการทำ Oversampling D/A จึงอาจทำให้วงจรรองที่สร้างขึ้นเกิดข้อจำกัดในการใช้งานที่จะให้สัญญาณที่ถูกหน่วงค่าไปถูกต้องเฉพาะในช่วงความถี่ค่อนข้างต่ำเมื่อเทียบกับความถี่การซีกตัวอย่าง โดยจะเห็นได้จากผลการจำลองการทำงานทางซอฟต์แวร์ ผลการจำลองการทำงานทางฮาร์ดแวร์ และผลที่ได้จากการทดสอบจริง

ในส่วนของการจำลองการทำงานทางซอฟต์แวร์ได้ใช้โปรแกรม MATLAB ในการออกแบบและจำลองการทำงานของวงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรราน ซึ่งจะทำการหน่วงเวลาสัญญาณอินพุตแล้วนำสัญญาณที่ถูกหน่วงเวลารวมเข้ากับสัญญาณอินพุต ให้สัญญาณเอาต์พุตมีความละเอียดขึ้น นั่นคือการทำ Oversampling D/A เมื่อนำสัญญาณคลื่นไซน์ความถี่ 1 kHz 3 kHz และ 5 kHz อัตราการซีกตัวอย่าง 20 kHz ที่สังเคราะห์ขึ้นไปผ่านวงจรดังกล่าวพบว่าสัญญาณที่ได้มีความละเอียดมากขึ้น 2 เท่า, 3 เท่า และ 4 เท่า ของสัญญาณอินพุตเดิม แสดงว่าวงจรที่ออกแบบมีการทำงานได้อย่างถูกต้อง แต่มีความคลาดเคลื่อนเล็กน้อยที่ความถี่สูงใกล้เคียงกับครึ่งหนึ่งของอัตราการซีกตัวอย่างเนื่องจากความละเอียดในการซีกตัวอย่างอาจไม่เพียงพอ

ในส่วนของการจำลองการทำงานทางฮาร์ดแวร์ได้ใช้ภาษา VHDL ในการออกแบบและสังเคราะห์วงจรของวงจรรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโรราน โดยสังเคราะห์สัญญาณอินพุตเป็นสัญญาณคลื่นไซน์ความถี่ 1 kHz 3 kHz และ 5 kHz อัตราการซีกตัวอย่าง 20 kHz จากโปรแกรม MATLAB แล้วแปลงสัญญาณเป็นไบนารีแบบทวูคอมพลีเมนต์ เมื่อนำไปผ่านวงจรที่ได้ออกแบบขึ้นพบว่า สัญญาณมีความละเอียดมากขึ้น 2 เท่า, 3 เท่า และ 4 เท่า ของสัญญาณอินพุตเดิม และมีอัตราการซีกตัวอย่างเพิ่มขึ้นเป็นจำนวนเท่าตามความละเอียดที่เพิ่มขึ้นด้วย ส่วนการออกแบบการแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz ผลที่ได้มี

อัตราการซีกตัวอย่างตรงตามที่ต้องการ แสดงว่าได้ทำการออกแบบและสังเคราะห์วงจรได้ถูกต้อง และสามารถนำไปประยุกต์ใช้งานด้าน Oversampling D/A หรือ Sampling-rate conversion ได้

ในส่วนสุดท้ายคือการดำเนินการจริงลงบนอุปกรณ์ฮาร์ดแวร์ ซึ่งประกอบด้วย Function generator, A/D converter, FPGA board และ D/A converter แสดงผลที่ได้ใน Digital oscilloscope โดยป้อนสัญญาณคลื่นไซน์ความถี่ 1 kHz 3 kHz และ 5 kHz จาก Function generator ใช้ A/D converter มีอัตราการซีกตัวอย่าง 20 kHz พบว่าผลที่ได้สัญญาณมีความละเอียดมากขึ้น 2 เท่า, 3 เท่า และ 4 เท่า ของสัญญาณที่ไม่ได้ผ่านวงจร Sampling Rate Converter และเป็นไปตามที่ได้ออกแบบ ส่วนการใช้การแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz มีอัตราการซีกตัวอย่างที่เปลี่ยนไปจริง สังเกตได้จากความกว้างของคาบเวลาของแต่ละตัวอย่างที่ใช้เวลา 20.8333 us ซึ่งเป็นคาบเวลาของอัตราการซีกตัวอย่าง 48 kHz และมีความคลาดเคลื่อนในบางตัวอย่าง อาจเนื่องมาจากจำนวนบิตที่ใช้มีจำนวนไม่เพียงพอ

5.2 ข้อเสนอแนะ

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโธธานเพื่อนำมาสังเคราะห์บนอุปกรณ์ FPGA ให้สามารถทำงานได้นั้นมีข้อจำกัดในเรื่องทรัพยากรที่ใช้ทำให้เกิดความผิดพลาดในการคำนวณบิต สังเกตได้จากการแปลงอัตราการซีกตัวอย่างจาก 44.1 kHz เป็น 48 kHz จึงควรใช้อุปกรณ์ที่มีประสิทธิภาพที่สูงขึ้น เพื่อคุณภาพของสัญญาณที่ดีขึ้น พร้อมกันนั้นต้องคำนึงถึงความถี่การซีกตัวอย่างของระบบด้วย นั่นคือต้องมีความถี่มากกว่าสองเท่าของความถี่ของสัญญาณที่ต้องการประมวลผล อีกทั้งในการใช้งานวงจรกรองแบบผ่านตลอดของโธธานนี้ยังเกิดข้อจำกัดในการเลือกใช้สัญญาณอินพุตที่จำกัดในช่วงความถี่ไม่สูงมากเมื่อเทียบกับความถี่การซีกตัวอย่าง

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบผ่านตลอดโธธานยังสามารถนำมาประยุกต์ใช้งานได้อีกมากมาย เช่น Time adjustment, echo cancellation, phase-array antenna systems, trans-multiplexers และ software radio ซึ่งสามารถศึกษาเพิ่มเติมได้จากงานวิจัยที่ถูกเผยแพร่มากมาย เนื่องจากเป็นหัวข้อที่กำลังได้รับความนิยมอย่างมากในเรื่องวงจรกรองดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ชำนาญ ปัญญาใส, และวัชรกร หนูทอง. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2547.
- [2] Blok, M.. "Fractional delay filter design for sample rate conversion." 2012 *Federated Conference on Computer Science and Information Systems (FedCSIS)*. (9-12 Sept. 2012) : 701 - 706.
- [3] J.P. Thiran. Recursive digital filter with maximally flat group delay. *IEEE Trans. On Circuit Theory*, 18(6):659-664, November 1971.
- [4] Kyung-Ju Cho, Ji-Suk Park, Byeong-Kuk Kim, Jin-Gyun Chung, and Parhi, K.K.. "Design of a Sample-Rate Converter From CD to DAT Using Fractional Delay Allpass Filter." *IEEE Transactions on Circuits and Systems II: Express Briefs*, (Volume: 54, Issue: 1). (Jan. 2007) : 19-23.
- [5] Vesa Välimäki. "Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters." Dissertation for the degree of Doctor of Technology, Helsinki University of Technology (Espoo, Finland), (1995)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

ผลการทดลองปริญญานิพนธ์ เทอมที่ 1

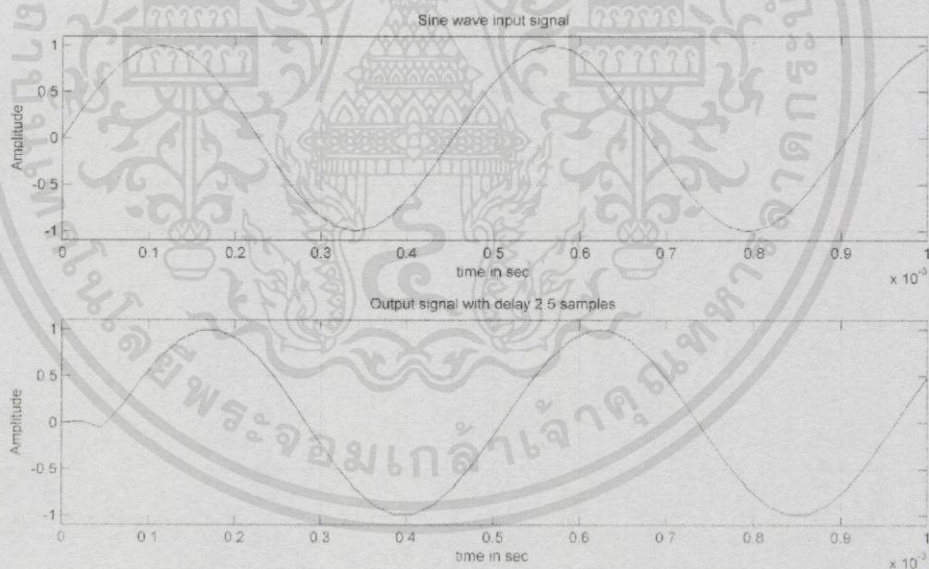
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองทั้งหมดจะแบ่งออกเป็นสามส่วนคือ ผลการจำลองซอฟต์แวร์ผ่านโปรแกรมแมตแล็บ ผลการจำลองฮาร์ดแวร์ด้วยภาษา VHDL และผลการทำงานจริงทางฮาร์ดแวร์ ซึ่งสามารถแบ่งผลการทดลองออกเป็นหัวข้อต่างๆ ได้ดังนี้

ผลการจำลองซอฟต์แวร์

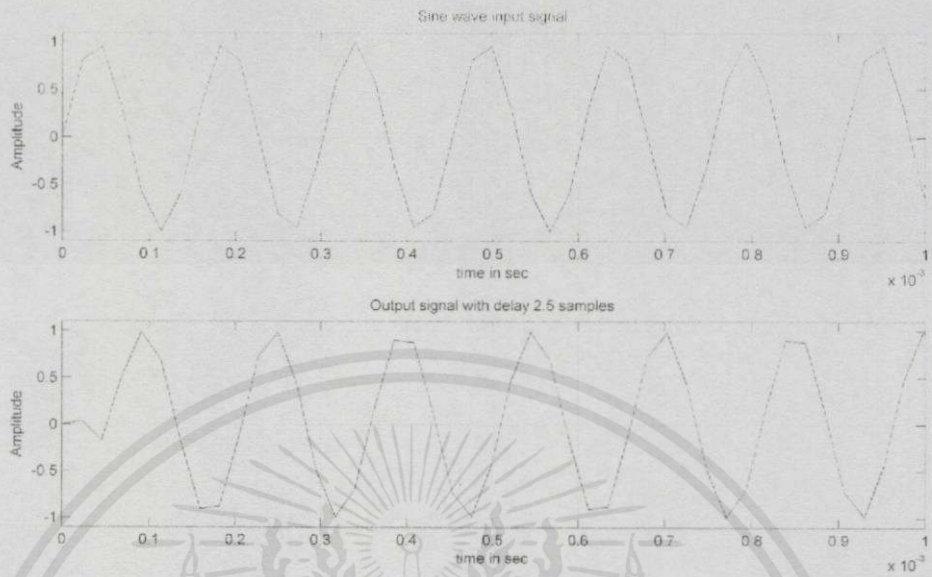
การจำลองการทำงานทางซอฟต์แวร์จะใช้โปรแกรมแมตแล็บในการจำลองการทำงาน โดยเริ่มจากการสร้างสัญญาณรูปไซน์เป็นสัญญาณขาเข้าที่มีความถี่ 2.205, 6.615 และ 11.025 kHz และมี Sampling frequency เท่ากับ 44.1 kHz เพื่อให้เห็นถึงผลของการจำลองการทำงานว่าสัญญาณนั้นมีการหน่วงเวลาเกิดขึ้นหรือไม่ จึงจะใช้ค่าความหน่วงเวลาในวงจรกรองอันดับ 1 เท่ากับ 1.5 แซมเปิ้ล หรือเทียบทางเวลาได้ประมาณ $34 \mu\text{s}$ ดังรูปที่ 1 ถึง 3 ส่วนวงจรกรองอันดับ 2 มีค่าความหน่วง 2.5 แซมเปิ้ล หรือประมาณ $56.7 \mu\text{s}$ ดังรูปที่ 4 ถึง 6

1. ผลจำลองการทำงานของวงจรกรองอันดับที่ 1

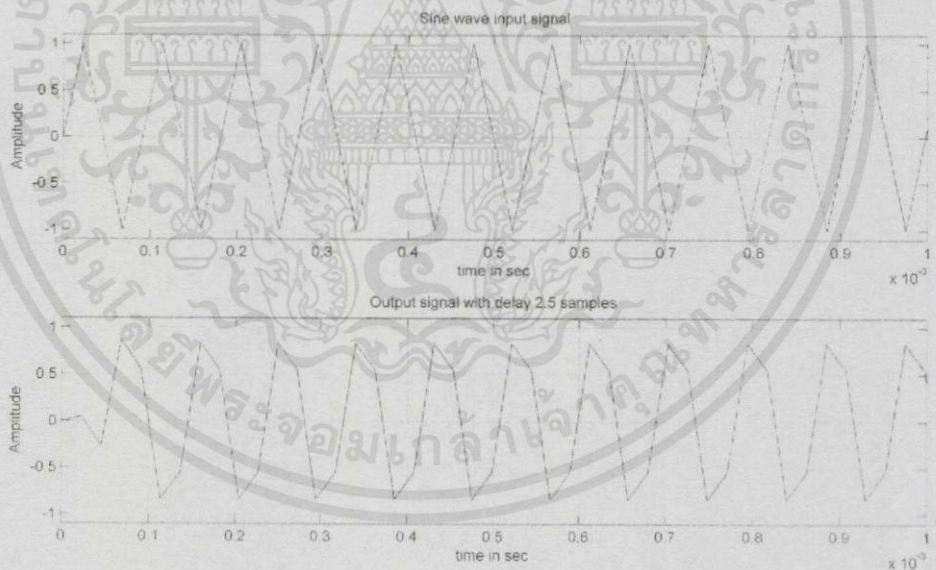


รูปที่ 1 ผลจากการจำลองวงจรกรองอันดับ 1 ด้วยโปรแกรมแมตแล็บที่ความถี่ 2.205 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



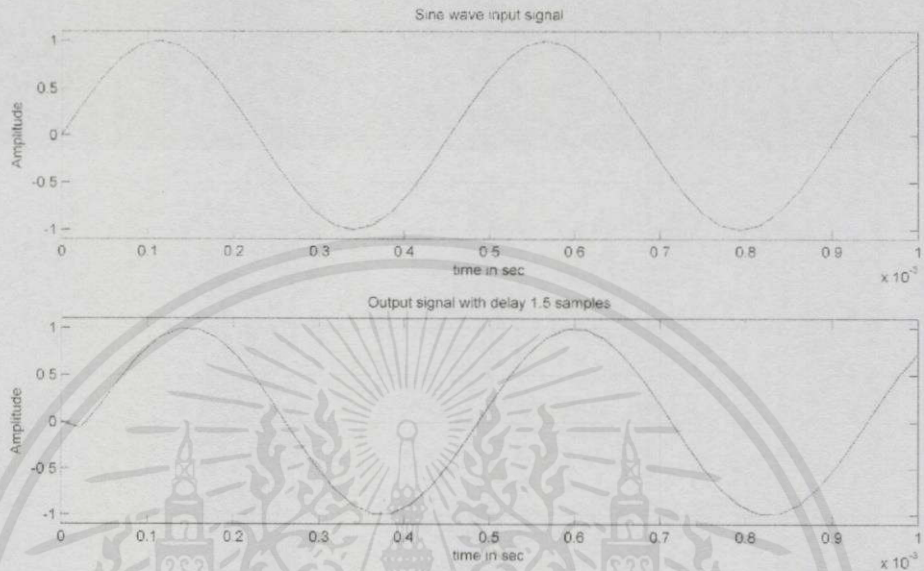
รูปที่ 2 ผลจากการจำลองวงจรกรองอันดับ 1 ด้วยโปรแกรมแมตแล็บที่ความถี่ 6.615 kHz



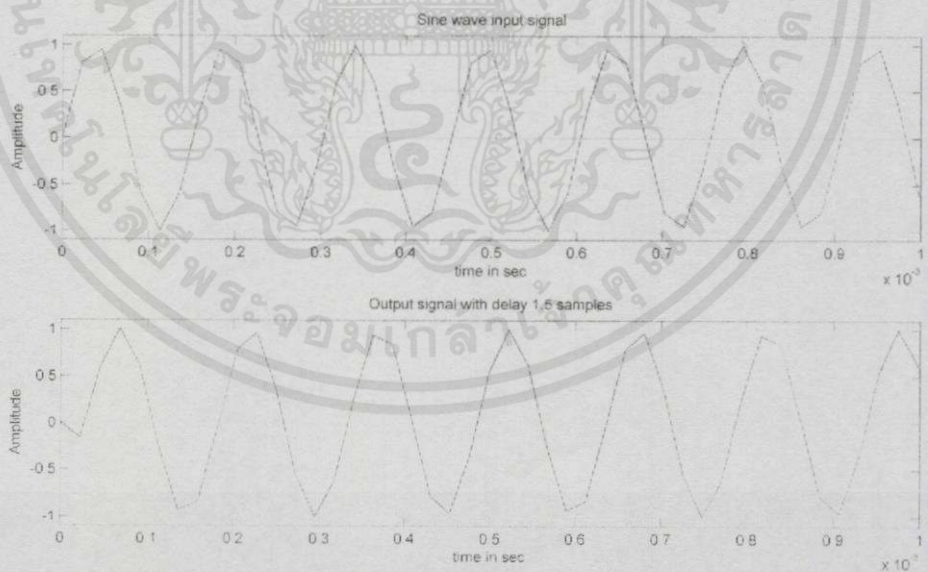
รูปที่ 3 ผลจากการจำลองวงจรกรองอันดับ 1 ด้วยโปรแกรมแมตแล็บที่ความถี่ 11.025 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ผลจำลองการทำงานของวงจรกรองอันดับที่ 2

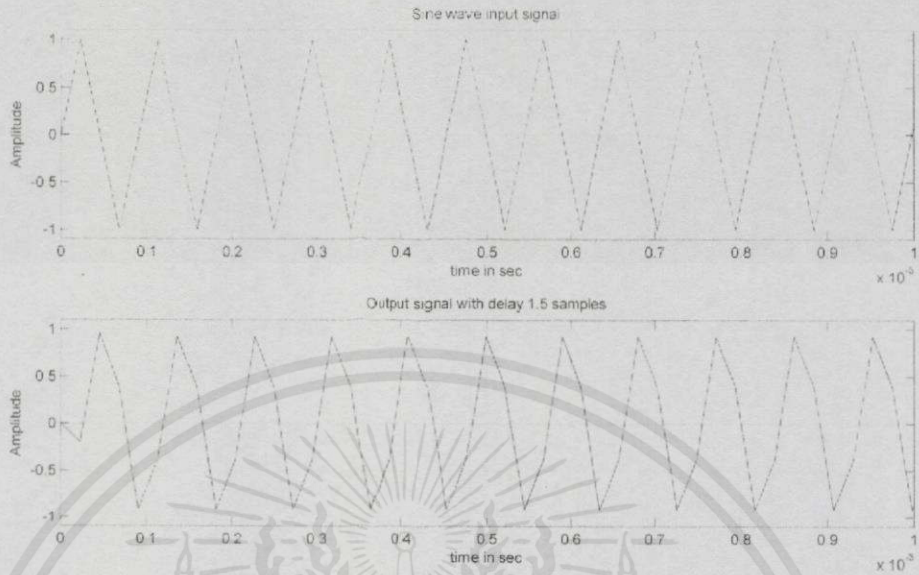


รูปที่ 4 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยโปรแกรมแมตแล็บที่ความถี่ 2.205 kHz



รูปที่ 5 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยโปรแกรมแมตแล็บที่ความถี่ 6.615 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยแมตแล็บที่ความถี่ 11.025 kHz

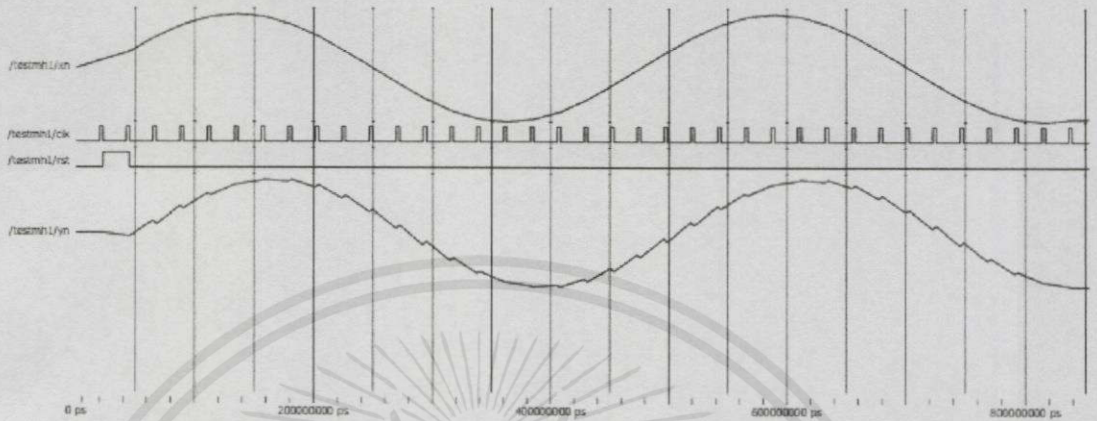
จากผลการทดลองที่แสดง รูปสัญญาณด้านบนของแต่ละรูปจะเป็นสัญญาณขาเข้า และรูปด้านล่างคือผลจากการนำสัญญาณขาเข้าผ่านวงจรหน่วงเวลา ซึ่งจะเห็นว่าสัญญาณได้ถูกหน่วงเวลาไปตามแซมเปิ้ลที่ได้กำหนดไว้ ผลการจำลองนี้ยืนยันได้ว่าโครงสร้างวงจรกรองสัญญาณหน่วงเวลาเป็นเศษส่วนแบบผ่านตลอด MH1 และ IS สอดคล้องกับทฤษฎี

ผลการจำลองฮาร์ดแวร์

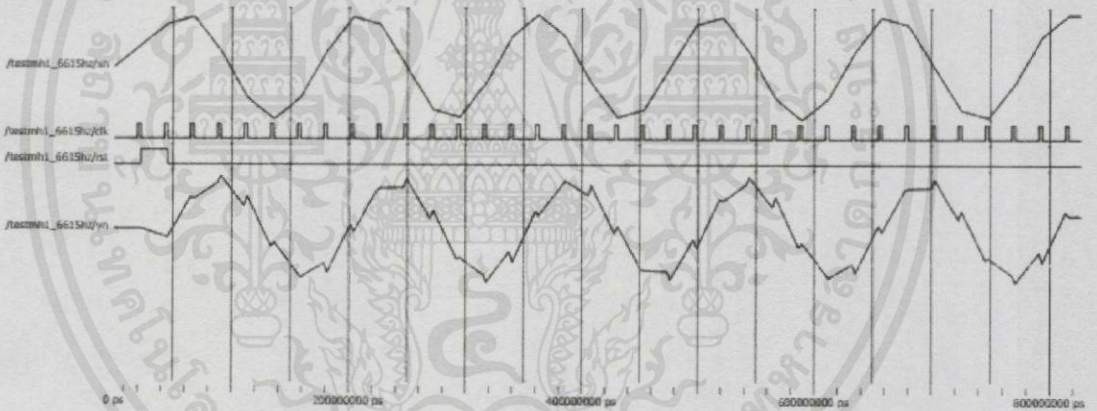
การจำลองการทำงานทางฮาร์ดแวร์จะใช้โปรแกรม ModelSim SE 6.5 ที่ใช้ภาษา VHDL ในการออกแบบอุปกรณ์และสังเคราะห์วงจรทางฮาร์ดแวร์ขึ้นเพื่อทำการนำไปใช้บนอุปกรณ์ FPGA ให้สามารถใช้งานวงจรได้จริง โดยเริ่มจากการสร้างสัญญาณรูปไซน์เป็นสัญญาณขาเข้าที่ความถี่ 2.205, 6.615 และ 11.025 kHz และมี Sampling frequency เท่ากับ 44.1 kHz เพื่อให้เห็นถึงผลของการจำลองการทำงานว่าสัญญาณนั้นมีการหน่วงเวลาเกิดขึ้นหรือไม่ จึงจะใช้ค่าความหน่วงเวลาในวงจรกรองอันดับ 1 เท่ากับ 1.5 แซมเปิ้ล หรือประมาณ $34 \mu\text{s}$ ดังรูปที่ 7 ถึง 9 และในวงจรกรองอันดับ 2 มีค่าความหน่วง 2.5 แซมเปิ้ล หรือประมาณ $56.7 \mu\text{s}$ ดังรูปที่ 10 ถึง 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

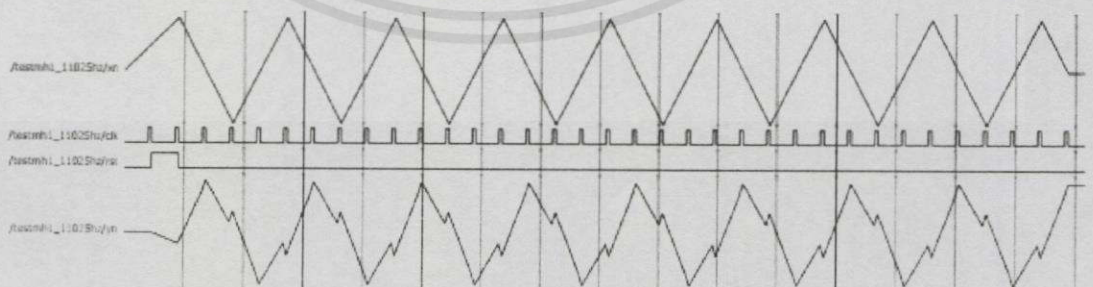
1. ผลจำลองการทำงานของวงจกรองอันดับที่ 1



รูปที่ 7 ผลจากการจำลองวงจกรองอันดับ 1 ด้วยภาษา VHDL ที่ความถี่ 2.205 kHz



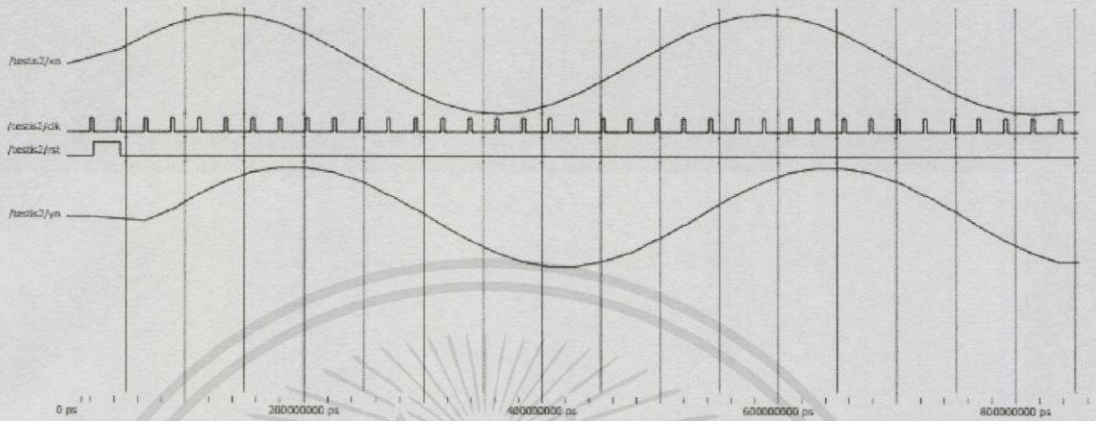
รูปที่ 8 ผลจากการจำลองวงจกรองอันดับ 1 ด้วยภาษา VHDL ความถี่ 6.615 kHz



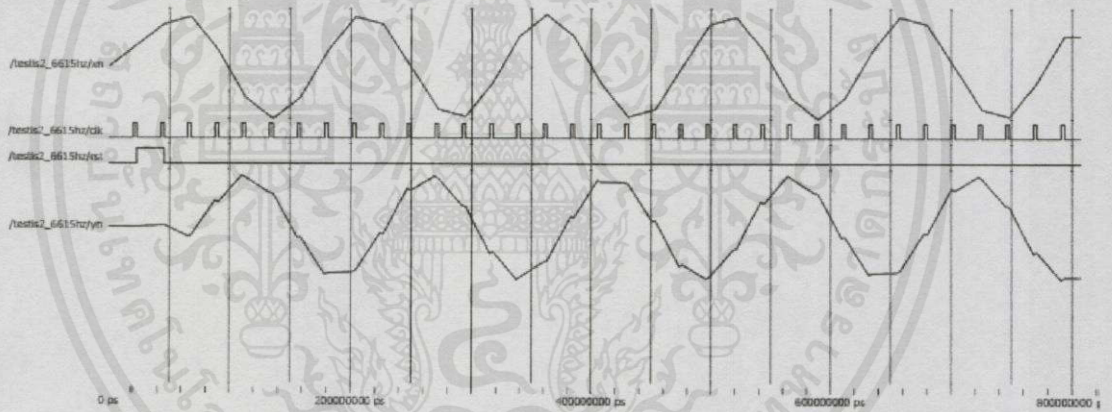
รูปที่ 9 ผลจากการจำลองวงจกรองอันดับ 1 ด้วยภาษา VHDL ที่ความถี่ 11.025 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกสิ่งที่มีให้สอดคล้องกับและต้องอ้างอิงถึงเจ้าของเอกสารฉบับนี้ที่มีการนำไปใช้

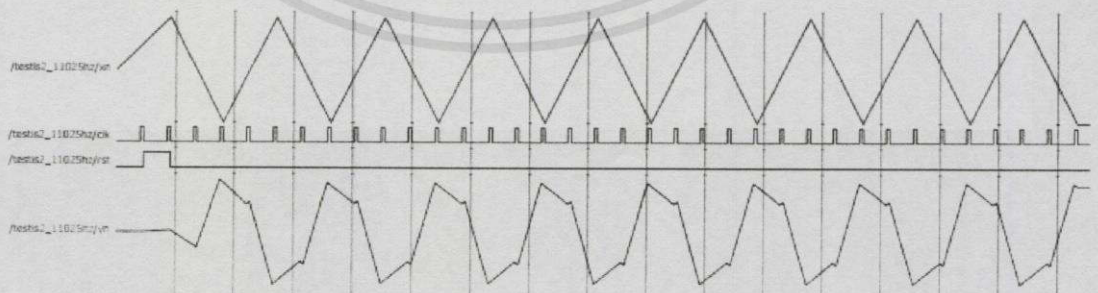
2. ผลจำลองการทำงานของวงจรกรองอันดับที่ 2



รูปที่ 10 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยภาษา VHDL ที่ความถี่ 2.205 kHz



รูปที่ 11 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยภาษา VHDL ที่ความถี่ 6.625 kHz



รูปที่ 12 ผลจากการจำลองวงจรกรองอันดับ 2 ด้วยภาษา VHDL ที่ความถี่ 11.025 kHz

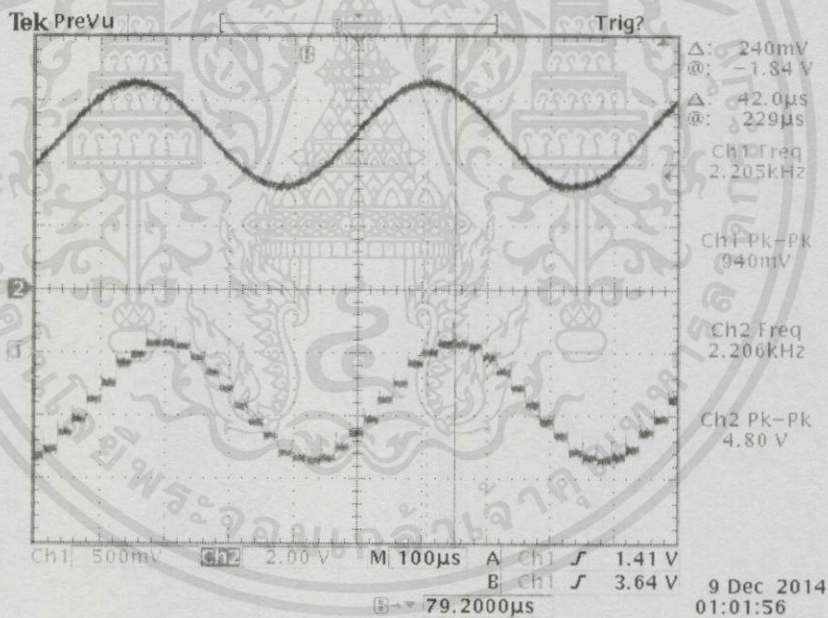
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร ซึ่งจะมีการนำไปใช้

ผลที่ได้จากการจำลองการทำงานทางฮาร์ดแวร์มีลักษณะใกล้เคียงกับการจำลองการทำงานทางซอฟต์แวร์ โดยที่ผลการจำลองการทำงานทางฮาร์ดแวร์นี้แสดงให้เห็นถึงการออกแบบและการสังเคราะห์วงจรที่ทำงานถูกต้อง และสามารถนำไปทำงานจริงบนอุปกรณ์ได้

ผลการจำลองการทำงานจริงทางฮาร์ดแวร์

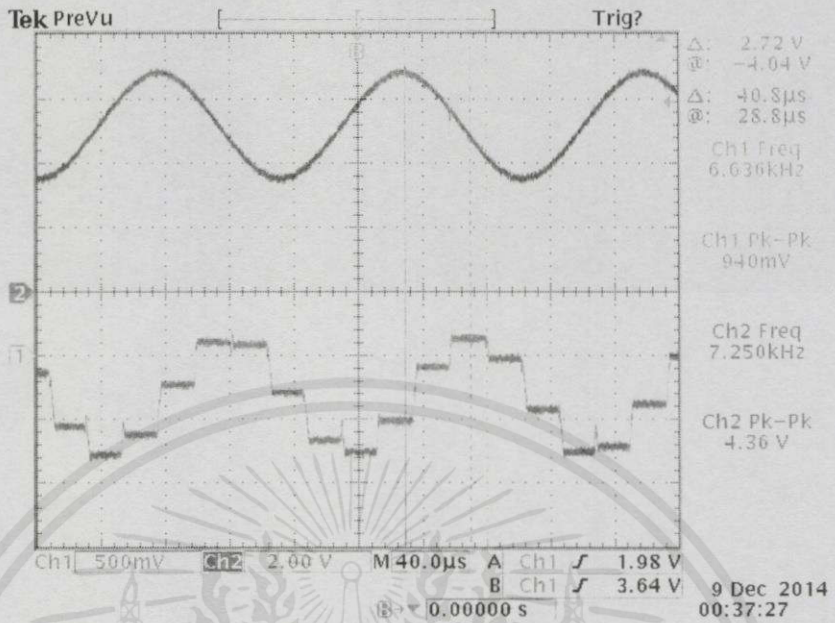
ผลการทำงานจริงคือผลที่ได้จากออสซิลโลสโคป ซึ่งจะทำการป้อนสัญญาณรูปไซน์ที่มีความถี่ที่กำหนดจากอุปกรณ์กำเนิดสัญญาณ (Function generator) เข้าไปยังวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล จากนั้นเป็นการประมวลผลสัญญาณทางดิจิทัลด้วยอุปกรณ์ที่ออกแบบและสังเคราะห์ขึ้น สุดท้ายนำสัญญาณที่ทำการประมวลผลแล้วไปแปลงกลับเป็นสัญญาณอนาล็อกเพื่อแสดงผลบนออสซิลโลสโคป ดังรูปที่ 13 ถึง 18

1. ผลจำลองการทำงานของวงจรกรองอันดับที่ 1

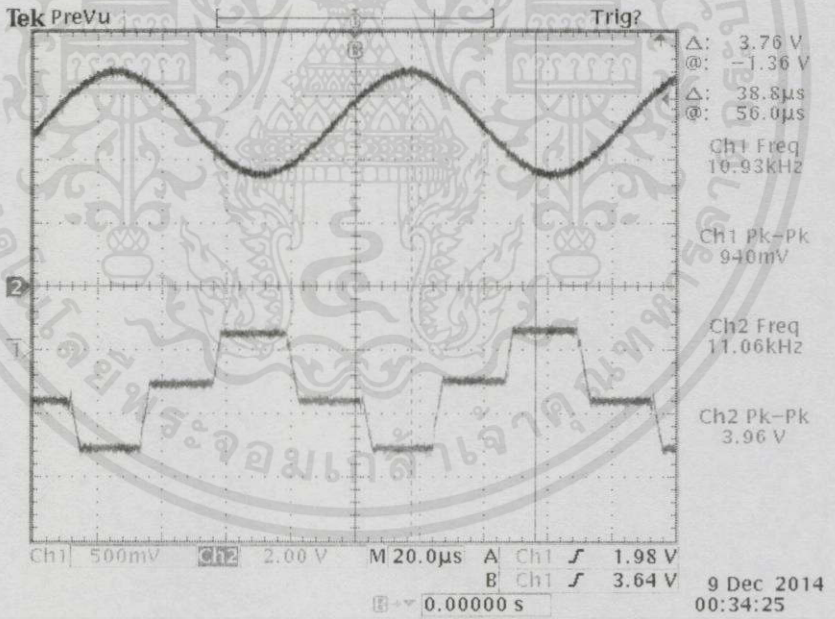


รูปที่ 13 ผลการทำงานจริงของวงจรกรองอันดับ 1 บนอุปกรณ์ที่ความถี่ 2.205 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



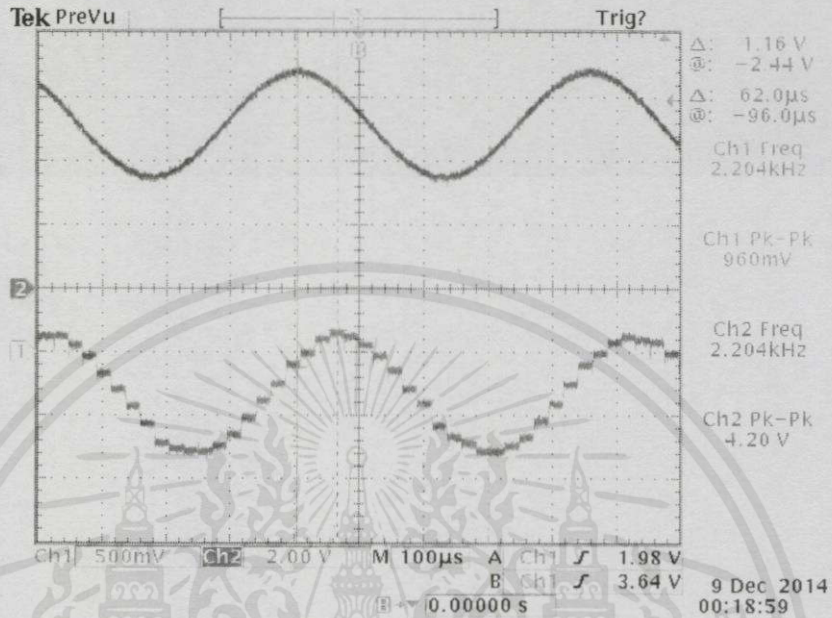
รูปที่ 14 ผลการทำงานจริงของวงจรกรองอันดับ 1 บนอุปกรณ์ที่มีความถี่ 6.625 kHz



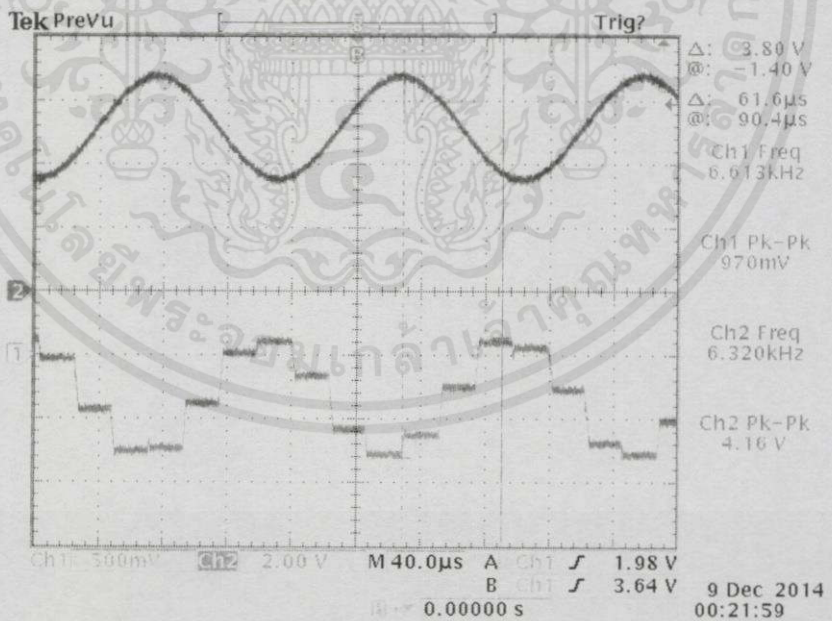
รูปที่ 15 ผลการทำงานจริงของวงจรกรองอันดับ 1 บนอุปกรณ์ที่มีความถี่ 11.025 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ผลจำลองการทำงานของวงจรกรองอันดับที่ 2

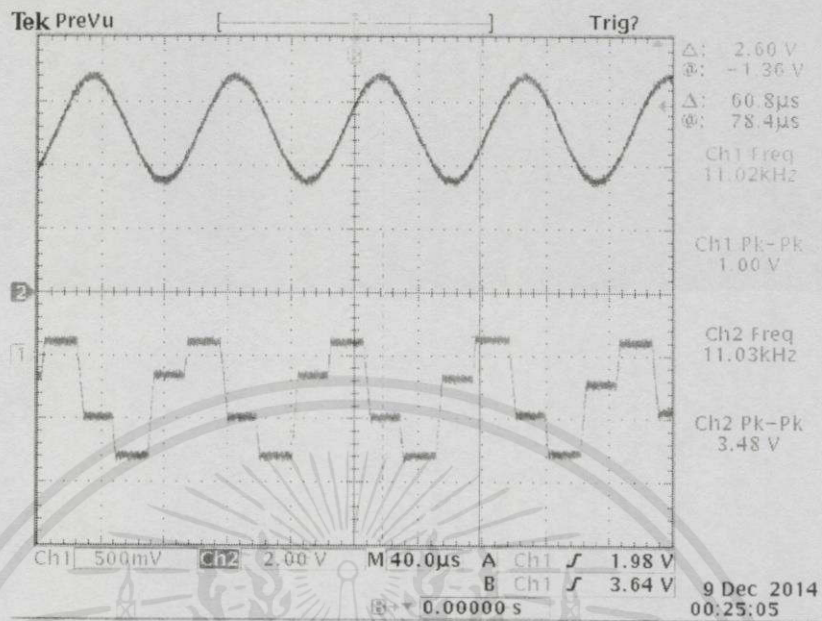


รูปที่ 16 ผลการทำงานจริงของวงจรกรองอันดับ 2 บนอุปกรณ์ที่ความถี่ 2.205 kHz



รูปที่ 17 ผลการทำงานจริงของวงจรกรองอันดับ 2 บนอุปกรณ์ที่ความถี่ 6.625 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 18 ผลการทำงานจริงของวงจรกรองอันดับ 2 บนอุปกรณ์ที่มีความถี่ 11.025 kHz

ผลจากออสซิลโลสโคปแสดงผลการเปรียบเทียบการหน่วงเวลาของสัญญาณที่มีความถี่ 2.205, 6.615 และ 11.025 kHz และมี Sampling frequency เท่ากับ 44.1 kHz ค่าความหน่วงเวลาที่ได้ใกล้เคียงกับค่าที่คำนวณได้ทางทฤษฎี แต่ลักษณะของสัญญาณต่างไปจากเดิมโดยเฉพาะเมื่อความถี่สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้