สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องรับและส่ง 16-QAM โดยใช้อุปกรณ์ FPGA

FPGA Implementation of 16-QAM Modulation and Demodulation

โดย

นางสาวพราวทัศน์ สมบุญ 45010513

นางสาวสุขุมาล ช่วงไชยยะ 45010842

เลขหมู่.....
เลขทะเบียน 62540
วัน,เดือน,ปี 19 ส.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรับและส่ง 16-QAM โดยใช้อุปกรณ์ FPGA

FPGA Implementation of 16-QAM Modulation and Demodulation

โดย

นางสาวพราวทัศน์ สมบุญ 45010513

นางสาวสุขุมาล ช่วงไชยยะ 45010842

อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

อาจารย์ สมปอง วิเศษพาณิชกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครื่องรับและส่ง 16-QAM โดยใช้อุปกรณ์ FPGA**

FPGA Implementation of 16-QAM Modulation and Demodulation

ผู้จัดทำ

1. นางสาว พราวทัศน์ สมบุญ 45010513

2. นางสาว สุชุมาล ช่วงไชยยะ 45010842

..... อาจารย์ที่ปรึกษา

(รศ.ดร. กอบชัย เดชหาญ)

..... อาจารย์ที่ปรึกษา

(อ. สมปอง วิเศษพาณิชกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรับและส่ง 16-QAM โดยใช้อุปกรณ์ FPGA

FPGA Implementation of 16-QAM Modulation and *Demodulation*

โดย นางสาวพราวทัศน์ สมบุญ 45010513

นางสาวสุขุมาล ช่วงไชยยะ 45010842

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ

อ. สมปอง วิเศษพาณิชกิจ

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อทำการศึกษาและทำการสร้างเครื่องรับและส่ง 16-QAM โดยทำการจำลองกระบวนการทำงานในโปรแกรม MATLAB และสร้างลงบนอุปกรณ์ FPGA โดยอธิบายวิธีการทำงานของวงจรด้วยภาษา VHDL

ABSTRACT

This project studies and implements 16-QAM modulation and demodulation, using MATLAB program to simulate processes of the work. Then, the processor will be implemented on FPGA using VHDL language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้จะสามารถไม่สำเร็จลุล่วงได้เลย ด้วยประการทั้งปวง ถ้าไม่มีพี่ที่ผู้คอยอุปถัมภ์ ให้คำปรึกษาและสนับสนุนอย่างเป็นทางการ ตั้งแต่เรื่องเล็กที่สุดจนถึงใหญ่ที่สุด และตั้งแต่ต้นจนจบ พวกเราซาบซึ้งใจมากจนไม่ทราบจะกล่าวเป็นภาษาคนได้อย่างไร จึงขอให้เกียรติท่านพี่ด้วยการกล่าวถึงเป็นอันดับแรก พร้อมกับการคารวะหนึ่งจอกมาด้วย ณ ที่นี้

บุพพการีของพวกเราที่ไม่ยอมให้เราลืมแน่นอน ขอบคุณพระคุณท่านพ่อผู้คอยส่งส่วยเช้า-ค่ำเมื่อยามลูกไม่ได้กลับบ้านเสียยาวนานเนื่องด้วยภารกิจชีวิตครั้งนี้ ขอบพระคุณท่านแม่ที่ช่วยแปลเปเปอร์กองพะเนินเมื่อยามลูกจวนตัวและจนใจ

ขอบพระคุณท่านอาจารย์ที่ปรึกษา และท่านอาจารย์กอบชัยใจดี ที่เอ็นดูและช่วยเหลือพวกเราตลอดมา พวกเราอาจประพฤติตัวไม่ฉลาดบ้างในบางครั้ง แต่ศิษย์ขออภัยและขอขอบพระคุณจากใจจริงสำหรับคำตักเตือนว่ากล่าว

สุดท้ายนี้ขอขอบพระคุณท่านเพื่อนทุกๆท่านที่ร่วมทุกข์ร่วมสุขกันตลอดระยะเวลาที่ผ่านมา ขอบคุณจริงๆสำหรับกำลังใจ

นางสาวพราวทัศน์ สมบุญ

นางสาวสุขุมาล ช่วงไชยยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	
2.1 โครงสร้างของระบบสื่อสาร	2
2.2 การมอดูเลตแบบดิจิทัลเบื้องต้น	3
2.3 การมอดูเลตเชิงแอมพลิจูดแบบควอเดรเจอร์	5
2.4 แมตช์ฟิลเตอร์	7
2.5 การแทรกสอดระหว่างสัญลักษณ์และการทำพัลส์เชปปิง	10
2.5.1 การแทรกสอดระหว่างสัญลักษณ์	10
2.5.2 การทำพัลส์เชปปิงแบบช่องสัญญาณไนควิสต์อุดมคติ	12
2.5.3 การทำพัลส์เชปปิงด้วยเรสโคไซน์พัลส์	13
2.6 การแปลงอัตราการสุ่ม	15
2.6.1 สัญญาณไม่ต่อเนื่อง	15
2.6.1.1 การสุ่มสัญญาณไม่ต่อเนื่อง	15
2.6.1.2 โพลีเฟส	17
2.6.2 การลดอัตราการสุ่ม	19
2.6.2.1 ดาวน์แซมปลิง	19
2.6.2.2 การกรองสัญญาณและสเปกตรัมการสุ่มค่าตัวอย่าง	21
2.6.2.3 เอกลักษณ์	25
2.6.3 การเพิ่มอัตราการสุ่ม	26
2.6.3.1 อัพแซมปลิง	26
2.6.3.2 การกรองสัญญาณและสเปกตรัมการสุ่มค่าตัวอย่าง	27
2.6.3.3 เอกลักษณ์	28
2.7 อินเตอร์โพเลชัน	29
2.7.1 อินเตอร์โพเลชันที่มีทรานส์เวอร์ซอลฟิลเตอร์	30
2.7.1.1 คอนโวลูชันด้วยการเพิ่มอัตราการสุ่มแบบลำดับ	30
2.7.1.2 ประสิทธิภาพของโครงสร้างทรานส์เวอร์ซอล	31
2.7.2 อินเตอร์โพเลชันที่มีโพลีเฟสฟิลเตอร์	32
2.7.2.1 โพลีเฟสอินเตอร์โพเลชันในไทม์โดเมน	32
2.7.2.2 เอาท์พุทคอมมิวเทชัน	34
2.7.2.3 โพลีเฟสอินเตอร์โพเลชันโดยใช้การแปลง Z	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ทฤษฎีดิจิทัลเฟสล็อกลูป 37
2.8.1 นิยามของหลักการทำงานเฟสล็อกลูปตามแบบฉบับ 37
2.8.2 นิยามดิจิทัลเฟสล็อกลูป 40
2.9 คอร์ดิกอัลกอริธึม 41
2.9.1 ทฤษฎี CORDIC 41
2.9.2 การสร้างลงใน FPGA 47
2.9.2.1 ตัวประมวลผลแบบวนรอบการทำงาน CORDIC 47
2.9.2.2 ตัวประมวลผลคอร์ดิกแบบออนไลน์ 48
2.10 คอสแทส ลูป (COSTAS LOOP) 49
2.10.1 คอสแทส ลูปด้วยวิธีการสร้างแบบคอร์ดิก 53
2.11 บูธ อัลกอริธึม (Booth Algorithm) 55
บทที่ 3 การคำนวณและการสร้าง
3.1 การมอดูเลชันแบบ QAM โดยใช้ CORDIC อัลกอริธึม 57
3.2 การมอดูเลตสัญญาณแบบ QAM 59
3.2.1 การออกแบบวงจรเรสโคไซน์ฟิลเตอร์โดยใช้โปรแกรมแมทแลป 60
3.2.2 คอนสเตลเลชัน(Constellation) 60
3.2.3 ผลการทดลองการออกแบบวงจรคอร์ดิกมอดูเลเตอร์โดยใช้โปรแกรมแมทแลป 62
3.3 การดีมอดูเลตสัญญาณแบบ QAM 63
บทที่ 4 การทดลองและผลการทดลอง
4.1 การออกแบบการทดลองสำหรับการวัดคุณลักษณะ 64
4.1.1 ส่วนของการสร้างวงจรหารความถี่ (DIV) 64
4.1.2 ส่วนของการรับบิทข้อมูลจากพอร์ตอนุกรม (SERIAL_RX) 65
4.1.3 ส่วนของเรสโคไซน์ฟิลเตอร์ (Raised Cosine filter) 66
4.1.3.1 ส่วนเก็บค่าและเรียกค่าข้อมูล (Rom) 66
4.1.3.2 ส่วนของตัวคูณ 67
4.1.4 ส่วนของคอร์ดิกอัลกอริธึม(Cordic) 68
4.1.5 ส่วนของคอนโทรลยูนิต (Control unit) 69
4.2 การนำโปรแกรมที่ออกแบบโดยภาษา VHDL ไปสร้างเป็นวงจรมอดูเลเตอร์ 71
4.2.1 การออกแบบการทดลองสำหรับการวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบ 72
4.2.2 ผลการทดลองวงจรคอร์ดิก ออสซิลเลเตอร์ (Cordic Oscillator) 73
4.2.2.1 เพิ่มความต่างเฟส 73
4.2.3 ผลการทดลองวงจรคอร์ดิกมอดูเลเตอร์ (Cordic modulator) 74
4.2.4 ผลการทดลองวงจรกรองความถี่ต่ำผ่านแบบผลตอบสนองอิมพัลส์จำกัด 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และบทสรุป

เอกสารอ้างอิง

กิตติกรรมประกาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1	แบบจำลองของระบบสื่อสาร	2
รูปที่ 2.2	รูปของสัญญาณที่ผ่านการมอดูเลตด้วยวิธีการที่ต่างกัน สามแบบ ASK, PSK และ FSK	4
รูปที่ 2.3	โครงสร้างวงจรภาคส่งแบบ QAM	5
รูปที่ 2.4	โครงสร้างวงจรภาครับแบบ QAM	6
รูปที่ 2.5	แบบจำลองวงจรภาครับสัญญาณที่ใช้แมตช์ฟิลเตอร์	7
รูปที่ 2.6	ตัวอย่างการรบกวนระหว่างสัญลักษณ์ของพัลส์รูปสี่เหลี่ยมสองรูป ขณะส่งผ่าน ช่องสัญญาณที่มีแบนด์วิดธ์จำกัด	10
รูปที่ 2.7	สัญญาณซิงก์	12
รูปที่ 2.8	สเปกตรัมสัญญาณซิงก์	12
รูปที่ 2.9	ตัวอย่างการใช้สัญญาณซิงก์สำหรับส่งข้อมูลไบนารี 101101	13
รูปที่ 2.10	สัญญาณทางเวลา	14
รูปที่ 2.11	สัญญาณทางสเปกตรัมความถี่	14
รูปที่ 2.12	การสุ่มแบบดิสครีท ที่มีเฟสออฟเซท λ =3	16
รูปที่ 2.13	Polyphase representation ของสัญญาณดิสครีท	18
รูปที่ 2.14	ดาวน์แซมเพลอร์ (Downsampler)	20
รูปที่ 2.15	ขั้นตอนการประมวลผลสัญญาณที่ใช้ในการดาวน์แซมปลิง	21
รูปที่ 2.16	ตัวสุ่มค่าตัวอย่าง (Down Sampler or Decimator)	21
รูปที่ 2.17	สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มสัญญาณด้วยความถี่ f_s	22
รูปที่ 2.18	สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มสัญญาณด้วยความถี่ $2f_s$	22
รูปที่ 2.19	สเปกตรัมของสัญญาณอินพุทก่อนการดาวน์แซมปลิง	23
รูปที่ 2.20	สเปกตรัมของสัญญาณเอาท์พุทหลังจากผ่านการ ดาวน์แซมปลิง	24
รูปที่ 2.21	แสดงเดซิเมเตอร์ที่ใช้ฟิลเตอร์ที่มีผลตอบสนองทางความถี่เป็น h(n) และดาวน์แซมเพลอร์ที่มีแฟคเตอร์คือ M	24
รูปที่ 2.22	เอกลักษณ์ที่ 1	25
รูปที่ 2.23	เอกลักษณ์ที่ 2	25
รูปที่ 2.24	เอกลักษณ์ที่ 3	25
รูปที่ 2.25	อัพแซมเพลอร์ (upsampler)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.26 การอัพแซมปลิงของสัญญาณดิสครีท 26
รูปที่ 2.27 สเปกตรัมของสัญญาณอินพุทก่อนการอัพแซมปลิง 27
รูปที่ 2.28 สเปกตรัมของสัญญาณเอาท์พุทเมื่อผ่านการอัพแซมปลิง 28
รูปที่ 2.29 อินเตอร์โพเลชันโดยใช้อัพแซมเพลอร์ L และแอนตีอิมเมจจิงฟิลเตอร์ g(n) 28
รูปที่ 2.30 เอกลักษณ์ที่ 4 29
รูปที่ 2.31 เอกลักษณ์ที่ 5 29
รูปที่ 2.32 เอกลักษณ์ที่ 6 29
รูปที่ 2.33 แสดงโครงสร้างโดยตรงของวงจรกรองความถี่แบบ FIR 30
รูปที่ 2.34 อินเตอร์โพเลชัน ประกอบด้วยอัพแซมเพลอร์ : L 1ตัว 30
และแอนตีอิมเมจจิงฟิลเตอร์: g(n) 1ตัว
รูปที่ 2.35 สัญญาณที่เกี่ยวข้องกับการคอนโวลูชันในรูปที่ 2.34 31
รูปที่ 2.36 อินเตอร์โพเลชัน 32
รูปที่ 2.37 การสร้างโพลีเฟสฟิลเตอร์ด้วย $L=3$ 33
รูปที่ 2.38 โครงสร้างทั่วไปของโพลีเฟสอินเตอร์โพเลชันที่มี L สาขา 34
และอัตราการสุ่มที่เพิ่มขึ้นด้วยแฟกเตอร์ L
รูปที่ 2.39 โพลีเฟสอินเตอร์โพเลชันที่มีการหน่วงที่เอาท์พุท 35
รูปที่ 2.40 โพลีเฟสอินเตอร์โพเลชันที่มีคอมมิวเตเตอร์ (Commutator) 35
แบบเอาท์พุททวนเข็มนาฬิกา
รูปที่ 2.41 อินเตอร์โพเลชันทั่วไปที่เป็นสัญญาณที่ได้จากการแปลง Z (Z-Transform) 36
รูปที่ 2.42 : (a) รูปแบบเดิมของโพลีเฟสอินเตอร์โพเลชัน 36
(b) รูปแบบที่มีประสิทธิภาพและประหยัดหน่วยความจำ
รูปที่ 2.43 : (a) บล็อกไดอะแกรมของระบบเฟสล็อกลูป (b) ฟังก์ชันถ่ายโอนของ VCO 37
(c) ฟังก์ชันถ่ายโอนของ PD
รูปที่ 2.44 โครงสร้างเสถียรภาพทางไดนามิค และสแตติก ของ PLL แบบเชิงเส้น 39
รูปที่ 2.45 โครงสร้างแบบวนรอบการทำงานคอร์ดิก 47
รูปที่ 2.46 ตัวประมวลผลคอร์ดิกแบบ Unrolled 48
รูปที่ 2.47 โครงสร้างวงจรดีมอดูเลตสัญญาณ DSB-SC แบบ Costas loop 49
รูปที่ 2.48 สแควริง ลูป (Squaring Loop) 51
รูปที่ 2.49 การสร้างคอสแทส ลูปแบบธรรมดา 51
รูปที่ 2.50 คอสแทส ลูปแบบดิจิตอลโดยการใช้ NCO เป็นตัวสังเคราะห์ความถี่ 52
รูปที่ 2.51 ส่วนคอร์ดิก, $X_{in}=x$, $Y_{in}=0$ 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.52 การสร้างแบบคอร์ดิกของคอสแทส ลูป 54
รูปที่ 2.53 แสดงตัวอย่างการคูณแบบพิจารณาเครื่องหมาย (8*8) 55
รูปที่ 2.54 ผลของการคูณแบบพิจารณาเครื่องหมาย (8*8) 56
รูปที่ 3.1 การหมุนเวกเตอร์ 57
รูปที่ 3.2 โครงสร้างของ CORDIC algorithm 58
รูปที่ 3.3 การมอดูเลชันแบบ QAM 59
รูปที่ 3.4 ผลตอบสนองอิมพัลส์ของเรสโคไซน์ฟิลเตอร์ 60
รูปที่ 3.5 ผลตอบสนองความถี่ของเรสโคไซน์ฟิลเตอร์ 60
รูปที่ 3.6 คอนสเตเลชันของสัญญาณ 16QAM 60
รูปที่ 3.7 การมอดูเลชันจากการออกด้วยโปรแกรมแมทแลป 62
กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1 และ จุดที่ 2 คือ I = 0.1, Q = -0.2
รูปที่ 3.8 การมอดูเลชันจากการออกด้วยโปรแกรมแมทแลป 62
กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1 และ จุดที่ 2 คือ I = - 0.1, Q = -0.1
รูปที่ 3.9 ส่วนของคอร์ดิก 63
รูปที่ 3.10 การดีมอดูเลชันแบบ QAM 63
รูปที่ 4.1 การเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะสัญญาณ 64
รูปที่ 4.2 สัญลักษณ์ของส่วนวงจรหารความถี่ 65
รูปที่ 4.3 ผลการจำลองการทำงานของส่วนวงจรหารความถี่ 65
รูปที่ 4.4 สัญลักษณ์ของวงจรการรับบิทข้อมูลจากพอร์ตอนุกรม 65
รูปที่ 4.5 ผลการจำลองการทำงานของวงจรการรับบิทข้อมูลจากพอร์ตอนุกรม 66
รูปที่ 4.6 สัญลักษณ์ของวงจรเก็บค่าและเรียกค่าข้อมูล 66
รูปที่ 4.7 สัญลักษณ์ของส่วนตัวคูณ 67
รูปที่ 4.8 ผลการจำลองการทำงานของวงจรเรสโคไซน์ฟิลเตอร์ 67
รูปที่ 4.9 สัญลักษณ์ของส่วนคอร์ดิก 68
รูปที่ 4.10 ผลการจำลองการทำงานของวงจรคอร์ดิก 68
รูปที่ 4.11 สัญลักษณ์ของส่วนคอนโทรลยูนิต 69
รูปที่ 4.12 ผลการจำลองการทำงานของวงจรคอนโทรลยูนิต 69
รูปที่ 4.13 ส่วนประกอบของวงจรมอดูเลเตอร์ 70
รูปที่ 4.14 ผลการวิเคราะห์โปรแกรม(View synthesis report) 71
รูปที่ 4.15 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณเชิงอนาล็อก 71
รูปที่ 4.16 การเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.17 วงจรรวมสำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น 72
รูปที่ 4.18 ผลการทดลองที่เกิดจาก คอร์ดิก ออสซิลเลเตอร์ โดยที่ $\Delta z = 0.1013$ เรเดียน 73
รูปที่ 4.19 ผลการทดลองที่เกิดจาก คอร์ดิก ออสซิลเลเตอร์ โดยที่ $\Delta z = 0.2094$ เรเดียน 73
รูปที่ 4.20 ผลการมอดูเลชันจากการออกแบบด้วยภาษา VHDL 74
กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1 และ จุดที่ 2 คือ I = 0.1, Q = -0.2
รูปที่ 4.21 ผลการมอดูเลชันจากการออกแบบด้วยภาษา VHDL 74
กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1 และ จุดที่ 2 คือ I = -0.1, Q = -0.1
รูปที่ 4.22 ผลการทดลองที่ได้จากวงจรกรองความถี่ต่ำผ่านอันดับที่ 6 75
รูปที่ 4.23 ผลการทดลองที่ได้จากวงจรกรองความถี่ต่ำผ่านอันดับที่ 6 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงซีเลคชันเทเบิลของบูธ อัลกอริธึม (selection table) 55

ตารางที่ 3.1 แสดงการเข้ารหัสข้อมูลดิจิตอล 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาหลักการการทำงานของระบบรับ - ส่งข้อมูลที่เข้ารหัสสัญญาณแบบ 16-QAM
2. เพื่อศึกษาและประยุกต์ใช้งานภาษาวีเอชดีแอล (VHDL) กับชิปเอฟพีจีเอ (FPGA Chip)
3. เพื่อศึกษาและประยุกต์ใช้งานทฤษฎีคอร์ดิก (CORDIC) สำหรับการมอดูเลตและดีมอดูเลตการเข้ารหัสสัญญาณแบบ 16-QAM
4. เพื่อนำหลักการและทฤษฎีมาใช้ในการสร้างแบบจำลองการทำงานของระบบรับ-ส่งข้อมูลที่เข้ารหัสสัญญาณแบบ 16-QAM
5. เพื่อจำลองกระบวนการทำงานด้วยโปรแกรมแมทแลป (Matlab) และเปรียบเทียบกับผลในทางปฏิบัติจากแบบจำลองที่สร้างขึ้น

1.2 ขอบเขตของปริญญานิพนธ์

โครงงานนี้นำเสนอการมอดูเลตและดีมอดูเลตสัญญาณที่เข้ารหัสแบบ 16-QAM โดยได้นำทฤษฎีคอร์ดิก(CORDIC), มัลติเรท(Multirate) และพัลส์เชปปิง(Pulse Shaping) มาประยุกต์ใช้ ในการสร้างนั้นจะทำการจำลองการทำงานด้วยโปรแกรมแมทแลป ก่อนนำไปเขียนอธิบายพฤติกรรมการทำงานด้วยภาษาวีเอชดีแอล และทำการโปรแกรม(Configure) ลงบนอุปกรณ์เอฟพีจีเอ (Field Programmable Gate Array)

1.3 เนื้อหาของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีและหลักการของระบบสื่อสาร รวมถึงทฤษฎีต่างๆที่นำมาใช้ในการออกแบบและการสร้าง

บทที่ 3 กล่าวถึงการออกแบบและการสร้างชุดจำลองการรับ-ส่งสัญญาณที่เข้ารหัสแบบ 16-QAM เพื่อทำการโปรแกรมลงบนอุปกรณ์เอฟพีจีเอ

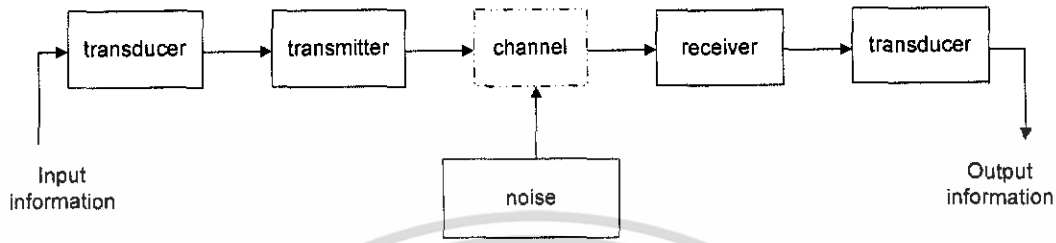
บทที่ 4 กล่าวถึงการทดลองและผลการทดลองจากการจำลองการทำงานโดยโปรแกรมแมทแลปและผลจากการทำงานจริงของชุดจำลองการรับ-ส่งสัญญาณและอุปกรณ์เอฟพีจีเอ

บทที่ 5 บทวิจารณ์และบทสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 โครงสร้างของระบบสื่อสาร



รูปที่ 2.1 แบบจำลองของระบบสื่อสาร

โดยทั่วไปเราสามารถแสดงแบบจำลองของระบบสื่อสารในรูปของแผนภาพบล็อคได้ดังรูปที่ 2.1 พิจารณาเริ่มจากทางด้านซ้ายมือ ข่าวสารด้านเข้าที่ป้อนเข้าสู่ระบบมีลักษณะได้หลากหลายรูปแบบ ทั้งที่เป็นสัญลักษณ์ล็อค(log) หรือ ดิจิตอล(digital) สัญญาณเหล่านี้จะถูกแปลงให้อยู่ในรูปสัญญาณที่เหมาะสมในการส่งผ่านระบบสื่อสาร เช่นสัญญาณเสียงจะถูกแปลงให้อยู่ในรูปของสัญญาณไฟฟ้า โดยใช้ไมโครโฟน ส่วนถัดมาของแบบจำลองระบบสื่อสารคือเครื่องส่ง (transmitter) อุปกรณ์ส่วนนี้มีหน้าที่ในการแปลงสัญญาณที่ออกจากทรานสดิวเซอร์ให้อยู่ในรูปที่เหมาะสมกับช่องสื่อสารที่จะส่งออก กระบวนการที่ใช้ในการแปลงสัญญาณนี้โดยทั่วไปเรียกว่า การมอดูเลต (modulation) การมอดูเลตสัญญาณมีได้หลายรูปแบบ เช่น การมอดูเลตเชิงแอมพลิจูด (amplitude modulation) การมองดูเลตเชิงความถี่ (frequency modulation) และการมอดูเลตเชิงเฟส (phase modulation) ส่วนถัดมาของแบบจำลอง คือ ช่องสื่อสาร(channel) ในส่วนนี้ก็คือ สื่อนำสัญญาณที่พาข่าวสารจากแหล่งกำเนิดไปสู่ปลายทางภาครับ สื่อที่ใช้มีตั้งแต่แบบที่มีสายนำสัญญาณ เช่น สายใยแก้วนำแสง สายคู่ตีเกลียว สายโคแอกซ์ (coaxial cord) ไปจนถึงส่งสัญญาณประเภทไม่มีสายนำสัญญาณ แต่อาศัยการแพร่ของคลื่นแม่เหล็กไฟฟ้าออกจากสายอากาศไปในบรรยากาศ ช่องสัญญาณส่วนใหญ่จะได้รับผลกระทบจากสัญญาณรบกวน และในสภาพที่สัญญาณรบกวนมีความรุนแรงมากอาจจะส่งผลให้ข่าวสารที่ส่งผ่านในช่องสัญญาณเกิดความผิดเพี้ยนไปจากเดิม เมื่อสัญญาณได้รับการส่งจนถึงจุดหมายปลายทาง สัญญาณเหล่านี้จะได้รับการแปลงกลับโดยเครื่องรับ (receiver) กระบวนการแปลงสัญญาณในขั้นตอนนี้เรียกว่า กระบวนการดีมอดูเลต (demodulation) จากนั้นสัญญาณที่ได้จะถูกแปลงกลับคืนให้กลายเป็นข่าวสารที่มีรูปแบบตามต้องการ โดยใช้อุปกรณ์ทรานส์ดิวเซอร์(transducer) ยกตัวอย่าง เช่น ลำโพงซึ่งทำหน้าที่แปลงสัญญาณไฟฟ้าให้กลายเป็นสัญญาณเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการมอดูเลตที่จัดว่าดีมีประสิทธิภาพควรจะมีคุณสมบัติดังต่อไปนี้

- สามารถให้อัตราความผิดพลาดบิตที่ต่ำ (bit error rate) แม้ว่าอัตราส่วนกำลังของสัญญาณต่อสัญญาณรบกวน (signal to noise ratio) ของสัญญาณที่รับได้มีค่าไม่สูงนัก

- มีความต้องการแบนด์วิดธ์ขนาดเล็ก และสามารถใช้ส่งข้อมูลด้วยอัตราบิตที่สูง นั่นคือ สามารถใช้งานสเปกตรัมความถี่ที่มีอยู่จำกัดได้อย่างมีประสิทธิภาพ

- วงจรภาครับและภาคส่งมีโครงสร้างเรียบง่าย ไม่ซับซ้อน สามารถสร้างจริงได้ด้วยต้นทุนที่ประหยัด

ในสภาพความเป็นจริงวิธีการมอดูเลตที่มีอยู่ในทางปฏิบัติมิได้มีคุณสมบัติที่ดีครบพร้อมกันทุกข้อการเลือกใช้วิธีการที่เหมาะสมจึงขึ้นอยู่กับความต้องการของระบบและทรัพยากรที่มีอยู่เป็นหลัก สำหรับในบทนี้จะอธิบายถึงหลักการพื้นฐานของการมอดูเลตแบบดิจิตอล

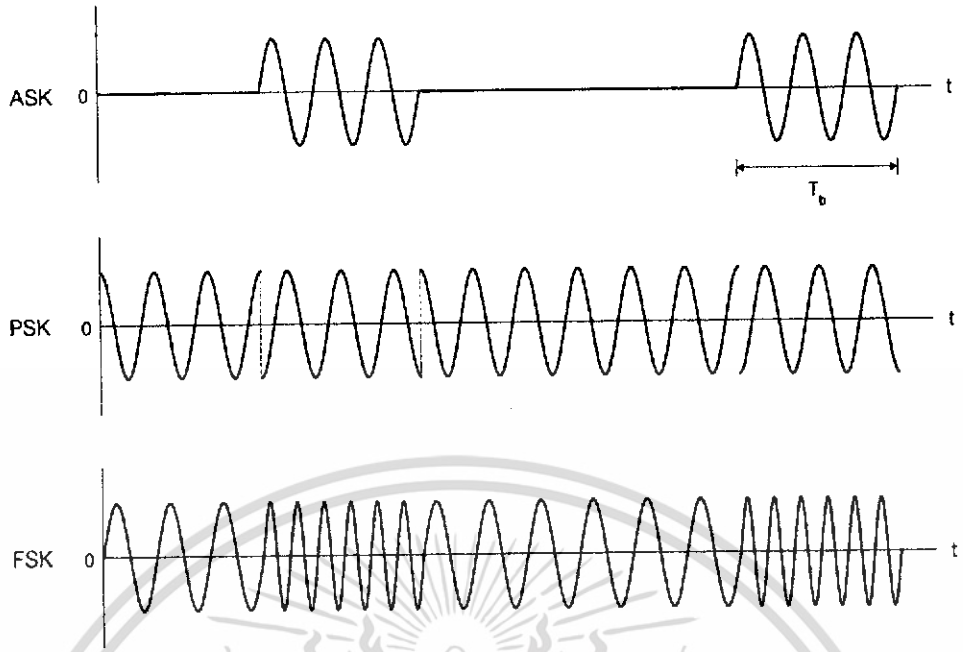
2.2 การมอดูเลตแบบดิจิตอลเบื้องต้น

ดิจิตอลมอดูเลชัน (Digital Modulation) คือ กระบวนการแปลงข้อมูลดิจิตอลที่อยู่ในรูป 0 และ 1 ให้ได้เป็นสัญญาณที่มีรูปลักษณ์เหมาะสมกับการส่งผ่านช่องสัญญาณโดยอาศัยคลื่นแม่เหล็กไฟฟ้า คลื่นสัญญาณที่ใช้กันทั่วไปจะเป็นสัญญาณไซนูซอยด์ (Sinusoidal signal) และเราเรียกสัญญาณที่เป็นเหมือนพาหนะในการนำข้อมูลข่าวสารดิจิตอลจากแหล่งกำเนิดไปถึงภาครับสัญญาณนี้ว่า คลื่นพาห์ (Carrier) การมอดูเลตสัญญาณข่าวสารดิจิตอลลงบนคลื่นพาห์สามารถกระทำได้หลายลักษณะ วิธีการพื้นฐานที่สำคัญและเป็นประโยชน์ในการนำมาใช้งานได้แก่ แอมพลิจูดชิฟต์คีย์อิง (ASK: Amplitude Shift Keying), ฟรีเควนซีชิฟต์คีย์อิง (FSK: Frequency Shift Keying) และเฟสชิฟต์คีย์อิง (PSK: Phase Shift Keying)

พิจารณาจากชื่อของแต่ละวิธีก็พอจะเห็นถึงความแตกต่างอย่างกว้าง ๆ ได้ดังนี้คือ การมอดูเลตแบบเอเอสเค(ASK) อาศัยการเปลี่ยนขนาดแอมพลิจูดของคลื่นพาห์ไปตามค่าของข้อมูลดิจิตอลที่ป้อนเข้ามา ดูตัวอย่างรูปสัญญาณเอเอสเคสำหรับข้อมูลดิจิตอลแบบไบนารี 2 ระดับ สังเกตว่าในตัวอย่างนี้ได้กำหนดให้แอมพลิจูดของคลื่นพาห์มีค่าเป็นศูนย์เพื่อแทนบิตข้อมูลที่เป็น 0 ซึ่งเทียบได้กับการไม่ส่งสัญญาณออกเลย และจะส่งสัญญาณไซนูซอยด์ที่มีแอมพลิจูดไม่เป็นศูนย์เพื่อแทนบิตข้อมูลที่มีค่าเป็น 1 ในกรณีเฉพาะแบบนี้จะเรียกกรรมวิธีการมอดูเลตแบบนี้ในอีกชื่อหนึ่งว่า ออนออฟคีย์อิง (OOK: On-Off Keying) เราสามารถเขียนสมการคณิตศาสตร์แทนสัญญาณไบนารีเอเอสเค (Binary ASK) ได้เป็น

$$s(t) = \begin{cases} A_c \cos(2\pi f_c t) & \text{symbol 1} \\ 0 & \text{symbol 0} \end{cases} \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 รูปของสัญญาณที่ผ่านการมอดูเลตด้วยวิธีการที่ต่างกันสามแบบ เอเอสเค (ASK), พีเอสเค (PSK) และ เอฟเอสเค (FSK)

ส่วนกรณีการมอดูเลตแบบเอฟเอสเค แอมพลิจูดของสัญญาณคลื่นพาห์ที่ใช้จะคงที่ไม่เปลี่ยนแปลง มีเพียงความถี่ของคลื่นพาห์เท่านั้นที่เปลี่ยนแปรไปตามข้อมูลดิจิตอล สัญญาณจากในตัวอย่างสัญญาณเอฟเอสเคจะมีความถี่ที่ต่างกัน 2 ค่า โดยความถี่สูงใช้แทนข้อมูลดิจิตอลที่มีค่าเป็น 1 และความถี่ต่ำใช้แทนข้อมูลที่เป็น 0 เราสามารถแทนเอฟเอสเคด้วยสมการข้างล่างนี้

$$s(t) = \begin{cases} A_c \cos(2\pi f_{c1} t) & \text{symbol 1} \\ A_c \cos(2\pi f_{c2} t) & \text{symbol 0} \end{cases} \tag{2.2}$$

สำหรับการมอดูเลตแบบพีเอสเค แอมพลิจูดและความถี่ของคลื่นพาห์จะกำหนดให้เป็นค่าคงที่ค่าหนึ่งโดยไม่มีการเปลี่ยนแปลงใด ๆ ส่วนที่เปลี่ยนไปตามข้อมูลดิจิตอล 0 หรือ 1 คือเฟส กล่าวคือถ้าบิตข้อมูลที่จะส่งมีค่าเป็น 0 ระบบจะส่งคลื่นพาห์ที่มีเฟสตามที่กำหนดค่าหนึ่ง และถ้าบิตข้อมูลมีค่าเป็น 1 ระบบจะส่งคลื่นพาห์ที่มีเฟสอีกค่าหนึ่งที่ต่างไปจากกรณีแรก โดยทั่วไปถ้าต้องการให้ได้ระบบที่มีสมรรถนะดี ก็มักจะเลือกให้เฟสทั้งสองมีค่าต่างกัน 180 องศา ซึ่งสามารถแทนด้วยสมการข้างล่างนี้

$$s(t) = \begin{cases} A_c \cos(2\pi f_c t) & \text{symbol 1} \\ A_c \cos(2\pi f_c t + \pi) & \text{symbol 0} \end{cases} \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวอย่างที่ได้ยกมานี้เป็นกรณีที่เรียบง่าย กล่าวคือรูปสัญญาณที่ใช้ในการส่งออกมีเพียง 2 รูปแบบ โดยใช้ในการแทนข้อมูลดิจิตอล 0 และ 1 แต่ในระบบสื่อสารที่ต้องการประสิทธิภาพที่ดีขึ้นมักจะมีรูปสัญญาณมากกว่า 2 แบบให้ใช้ และจะพิจารณาบิตข้อมูลครั้งละมากกว่าหนึ่งบิต ยกตัวอย่างเช่น ถ้าเราพิจารณาบิตข้อมูลในการส่งคราวละ 3 บิต ระบบจะต้องมีรูปสัญญาณที่แตกต่างกันถึง $2^3 = 8$ รูปแบบ เพื่อแทนสัญลักษณ์ที่แตกต่างกันดังนี้ 000, 001, 010, 011, 100, 101, 110 และ 111

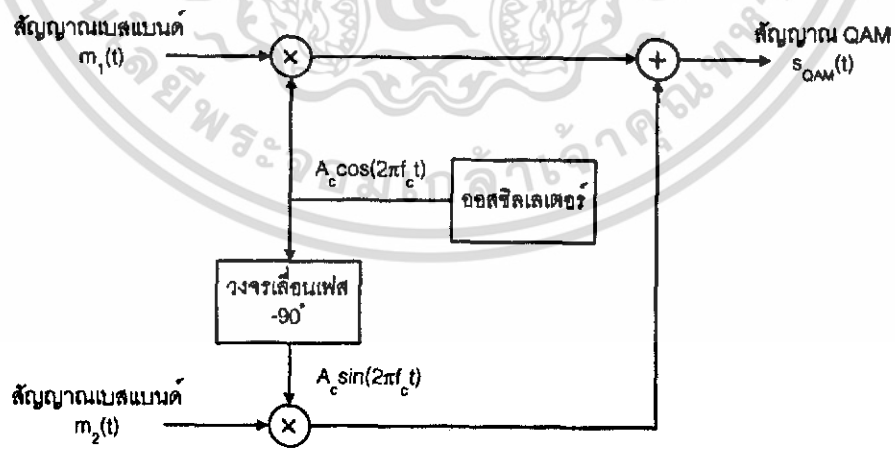
2.3 การมอดูเลตเชิงแอมพลิจูดแบบควอเดรเจอร์ (Quadrature Amplitude Modulation)

เราสามารถนำสัญญาณดีเอสบี-เอสซี (DSB-SC) 2 สัญญาณที่ใช้ส่งเบสแบนด์ 2 ชุดที่แตกต่างกันมาส่งลงในช่องสัญญาณสื่อสารในสเปกตรัมช่วงเดียวกันได้ นั่นคือระบบสามารถรับส่งข้อมูลได้มากขึ้นเท่าตัวโดยอาศัยแบนด์วิดธ์ขนาดเท่าเดิม การมอดูเลตสัญญาณในลักษณะนี้มีชื่อเรียกว่า การมัลติเพล็กซ์คลื่นพาห์แบบควอเดรเจอร์ (Quadrature carrier multiplexing) หรือ การมอดูเลตเชิงแอมพลิจูดแบบควอเดรเจอร์ (Quadrature Amplitude Modulation) ซึ่งเรียกย่อๆว่า คิวเอเอ็ม (QAM)

พิจารณาโครงสร้างของวงจรกำเนิดสัญญาณคิวเอเอ็ม(QAM) ในรูป 2.3 สัญญาณเบสแบนด์สองชุดคือ $m_1(t)$ และ $m_2(t)$ ถูกป้อนเข้าวงจรคูณกับสัญญาณคลื่นพาห์ 2 คลื่นที่มีความถี่เดียวกันแต่มีเฟสต่างกัน –90 องศา จากนั้นนำสัญญาณที่ได้มารวมกันผลลัพธ์ที่ได้คือสัญญาณที่มีค่าเท่ากับ

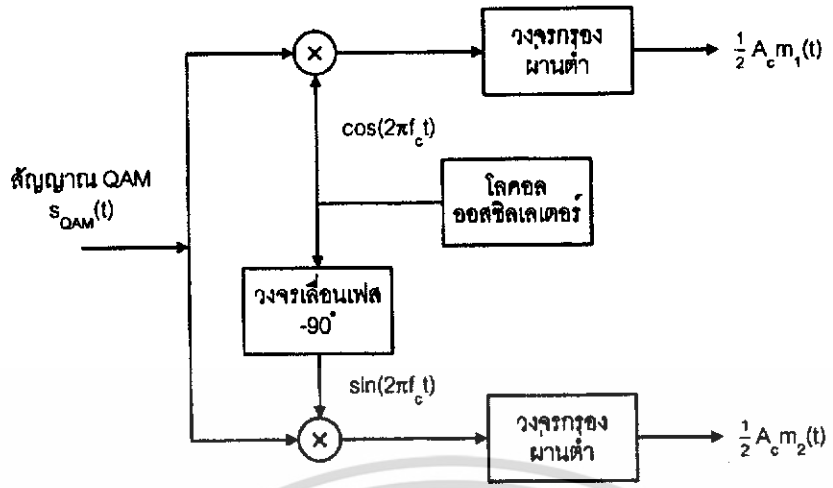
$$S_{QAM}(t) = A_c\, m_1(t)\cos(2\pi f_c t) + A_c\, m_2(t)\sin(2\pi f_c t) \qquad (2.4)$$

จากสมการสัญญาณคิวเอเอ็มนี้ เราจะเรียก $A_c\, m_1(t)$ ว่าเป็นองค์ประกอบร่วมเฟส (in-phase component) และเรียก $-A_c\, m_1(t)$ ว่าเป็นองค์ประกอบควอเดรเจอร์ (quadrature component)



รูปที่ 2.3 โครงสร้างวงจรภาคส่งแบบ คิวเอเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 โครงสร้างวงจรภาครับแบบ คิวเอเอ็ม

ส่วนวงจรภาครับสัญญาณมีโครงสร้างดังที่แสดงในรูปที่ 2.4 สัญญาณคิวเอเอ็มที่ได้ถูกแยกออกเป็นสองส่วน ส่วนแรกถูกนำไปคูณกับคลื่นสัญญาณ $\cos(2\pi f_c t)$ และนำไปผ่านวงจร กรองผ่านต่ำ ก็จะได้สัญญาณเบสแบนด์ $\frac{1}{2}A_c\, m_1(t)$ สำหรับส่วนที่สองนำไปคูณกับคลื่นสัญญาณ และนำไปผ่านวงจรกรองผ่านต่ำ ก็จะได้สัญญาณเบสแบนด์ $\frac{1}{2}A_c\, m_2(t)$ ปัญหาหลักของการดีมอดูเลตสัญญาณคิวเอเอ็ม คือ การซิงโครไนซ์สัญญาณคลื่นพาห์ทั้งเชิงความถี่และเฟสระหว่างสัญญาณคิวเอเอ็มกับสัญญาณที่เกิดจากโลคอลออสซิเลเตอร์ (local oscillator)ให้ตรงกันตลอดเวลา มิฉะนั้นจะเกิดการรบกวนกันระหว่างสัญญาณที่ดีมอดูเลตทั้งสองได้ ยกตัวอย่างเช่น หากมีความผิดพลาดเชิงเฟสของโลคอลออสซิลเลเตอร์ ทั้งสองตัวไปด้วยค่าเท่ากับ Φ_1 และ Φ_2 ตามลำดับ สัญญาณที่ได้จากวงจรภาครับ หลังจากผ่านวงจรกรองผ่านต่ำจะมีค่าเป็น

$$\frac{1}{2}A_c\, m_1(t)\cos(\Phi_1)+\frac{1}{2}A_c\, m_2(t)\sin(\Phi_1)$$

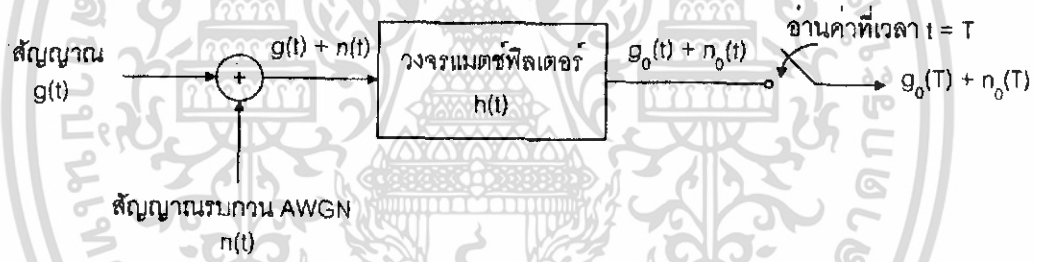
และ

$$\frac{1}{2}A_c\, m_1(t)\sin(\Phi_2)+\frac{1}{2}A_c\, m_2(t)\cos(\Phi_2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบสื่อสารดิจิตอล ข้อมูลจากแหล่งกำเนิดข่าวสารที่ต้องการส่งผ่านช่องสัญญาณ จะอยู่ในรูปของชุดสัญญาณที่มีรูปแบบหรือจำนวนสัญลักษณ์จำกัด และโดยทั่วไปรูปแบบข้อมูลที่ใช้จะอยู่ในรูปของสัญญาณไบนารี นั่นคือจะมีสัญลักษณ์เพียงสองลักษณะได้แก่ 0 และ 1 หลังจากคลื่นสัญญาณ ได้รับการส่งผ่านช่องสัญญาณและถึงภาครับปลายทางแล้ว ภาครับจะทำการตัดสินใจว่ารูปสัญญาณที่ได้รับจากต้นทางเป็นข่าวสารข้อมูลสัญลักษณ์ใดการทำงานในส่วนนี้จะต้องมีการศึกษาถึงหลักการทำงานของวงจรที่มีชื่อว่า แมตช์ฟิลเตอร์ (match filter) ซึ่งเป็นวงจรภาครับที่ให้อัตราความผิดพลาดบิตต่ำสุดสำหรับช่องสัญญาณที่มีสัญญาณรบกวนแบบเกาส์ เนื่องจากสัญญาณพัลส์รูปสี่เหลี่ยมที่ใช้ในการส่งผ่านสัญญาณเบสแบนด์มีแบนด์วิดธ์ที่กว้างมาก ฉะนั้นเมื่อสัญญาณดังกล่าวได้รับการส่งผ่านช่องสื่อสารที่มักจะมีแบนด์วิดธ์จำกัด จึงก่อให้เกิดความผิดเพี้ยนขึ้นกับรูปสัญญาณและผลกระทบที่เกิดขึ้นนี้มีชื่อเรียกว่า การแทรกสอดระหว่างสัญลักษณ์ (ISI : Intersymbol Interference) ปัญหา ISI ที่กล่าวถึงนี้สามารถแก้ไขได้โดยอาศัยกระบวนการที่เรียกว่า การพัลส์เชปปิง (pulse shaping) จากที่กล่าวมาทั้งหมดจะเห็นว่าวิธีการเหล่านี้ได้รับการพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อให้ระบบสามารถส่งข่าวผ่านช่องสัญญาณที่มีแบนด์วิดธ์จำกัด ได้ด้วยอัตราที่สูงสุดและมีความผิดพลาดน้อยที่สุด

2.4 แมตช์ฟิลเตอร์



รูปที่ 2.5 แบบจำลองวงจรภาครับสัญญาณที่ใช้แมตช์ฟิลเตอร์

ในลำดับต่อไปนี้จะอธิบายถึงหลักการทำงานของวงจรแมตช์ฟิลเตอร์ (Match Filter) ซึ่งเป็นวงจรกรองที่มีหน้าที่ในการดึงข่าวสารของสัญญาณที่ได้รับผลกระทบจากช่องสัญญาณที่มีการรบกวนเกาส์สีขาวแบบบวก (AWGN: Additive White Gaussian Noise) ตามแบบจำลองในรูปที่ 2.5 เพื่อให้ได้ค่าอัตราส่วนระหว่างขนาดของสัญญาณและค่าเฉลี่ยกำลังของสัญญาณ รบกวน (signal-to-noise ratio) มีค่าสูงสุด พิจารณาจากแบบจำลองในรูปที่ 2.5 สัญญาณ g(t) ที่บรรจุข่าวสารของข้อมูลดิจิตอลได้รับผลกระทบจากสัญญาณรบกวน n(t) ผลรวมของสัญญาณทั้ง 2 ส่วนได้รับการป้อนเข้าสู่วงจรแมตช์ฟิลเตอร์ที่มีฟังก์ชันตอบสนองอิมพัลส์เท่ากับ h(t) สัญญาณที่จากทางด้านออกของวงจรจะเขียนได้เป็น $g_0(t) + n_0(t)$ จากนั้นภาครับจะมีการอ่านค่าของสัญญาณที่ได้ ณ เวลาที่ $t = T$ เพื่อนำไปใช้ในการตัดสินใจบิตซึ่งเป็นขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวในข้างต้นแล้วว่าวัตถุประสงค์ของวงจรแมตช์ฟิลเตอร์คือการหาวงจรกรอง h(t) ที่ทำให้ขนาดของสัญญาณ $g_0(t)$ มีค่าสูงที่สุดเมื่อเทียบกับค่ากำลังเฉลี่ยของสัญญาณรบกวน $n_0(t)$ ณ ที่เวลา t = T ซึ่งเทียบได้กับการหาค่าสูงสุดของอัตราส่วน

$$\eta = \frac{|g_0(T)|^2}{E\left[n_0^2(t)\right]} \tag{2.5}$$

โดย $|g_0(T)|^2$ คือกำลังชั่วขณะของสัญญาณ $g_0(t)$ ที่ด้านออก และ $E[n_0^2(t)]$ คือค่ากำลังเฉลี่ยของสัญญาณรบกวน จากรูปที่ 2.5 เราสามารถเขียนความสัมพันธ์ระหว่าง $g_0(t)$ และ g(t) ดังนี้

$$g_0(t) = h(t) * g(t) \tag{2.6}$$

อาศัยคุณสมบัติของการแปลงฟูเรียร์ผกผันสมการที่ (2.6) สามารถเขียนใหม่ได้เป็น

$$g_0(t) = \int_{-\infty}^{\infty} H(f)G(f)\exp(j\pi ft)df \tag{2.7}$$

เพราะฉะนั้น

$$|g_0(T)|^2 = \left|\int_{-\infty}^{\infty} H(f)G(f)\exp(j\pi fT)df\right|^2 \tag{2.8}$$

สำหรับสัญญาณรบกวน $n_0(t)$ นั้นได้จากการนำสัญญาณรบกวนเกาส์สีขาวแบบบวก n(t) ไปผ่านวงจรแมตช์ฟิลเตอร์ที่มีฟังก์ชันถ่ายโอนเท่ากับ H(f) สมมติให้สัญญาณรบกวน n(t) มีความหนาแน่นสเปกตรัมกำลังเท่ากับ $N_0/2$ ดังนั้นสัญญาณรบกวน $n_0(t)$ จึงมีความหนาแน่นสเปกตรัมกำลังเท่ากับ

$$S_N(f) = \frac{N_0}{2}|H(f)|^2 \tag{2.9}$$

และมีกำลังเฉลี่ยเท่ากับ

$$E\left[n_0^2(t)\right] = \frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2 df \tag{2.10}$$

จากสมการที่ (2.8) และ (2.10) เราสามารถเขียนสมการที่ (2.5) ใหม่ได้เป็น

$$\eta = \frac{\left|\int_{-\infty}^{\infty} H(f)G(f)\ \exp\ (j\pi fT)df\right|^2}{\frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2 df} \tag{2.11}$$

พิจารณาจากสมการที่ได้สังเกตว่าเมื่อมีการกำหนดรูปสัญญาณ g(t) และสเปกตรัม G(f) ของสัญญาณที่รับได้มาให้ จะเหลือเพียงฟังก์ชันถ่ายโอน H(f) เท่านั้นที่เราสามารถเปลี่ยนแปลงได้และหากเลือกอย่างเหมาะสมจะให้ค่าอัตราส่วน η ที่มีค่าสูงสุดตามต้องการได้ ในการฟังก์ชัน H(f) ที่เหมาะสมสามารถกระทำได้โดยอาศัยหลักความไม่เท่ากันของชวาร์ซ (Schwarz's inequality) ซึ่งมีใจความว่า

สำหรับฟังก์ชันเชิงซ้อน X(f) และ Y(f) ใด ๆ ความสัมพันธ์ต่อไปนี้จะเป็นจริงเสมอ

$$\left|\int_{-\infty}^{\infty} X(f)Y^*(f)df\right|^2 \le \int_{-\infty}^{\infty}|X(f)|^2 df \int_{-\infty}^{\infty}|Y(f)|^2 df \tag{2.12}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และความสัมพันธ์จะเปลี่ยนเป็นเครื่องหมายเท่ากับได้สำหรับกรณีที่ X(f) = kY(f) เมื่อ k เป็นค่าคงที่ค่าหนึ่ง

ในการหาค่าสูงสุดของ η เราจะกำหนดให้ G(f) = X(f) และ $H(f)\exp(j\pi fT) = Y^*(f)$ จากนั้นแทนค่า $\left|\int_{-\infty}^{\infty} H(f)G(f)\exp(j\pi fT)df\right|^2$ ในสมการ(2.11)ด้วย $\int_{-\infty}^{\infty}|H(f)|^2df\int_{-\infty}^{\infty}|G(f)|^2df$ โดยอาศัยคุณสมบัติของหลักความไม่เท่ากันของชวาร์ซ ดังนั้นสมการที่ (2.11) จึงเขียนใหม่ได้เป็น

$$\eta \le \frac{\int_{-\infty}^{\infty}|H(f)|^2df\int_{-\infty}^{\infty}|G(f)|^2(df)}{\frac{N_0}{2}\int_{-\infty}^{\infty}|H(f)|^2df} \tag{2.13}$$

นั่นคือ

$$\eta \le \frac{2}{N_0}\int_{-\infty}^{\infty}|G(f)|^2df \tag{2.14}$$

จากคุณสมบัติเฉพาะของหลักความไม่เท่ากันของชวาร์ซ ที่ระบุว่าเครื่องหมายมากกว่าหรือเท่ากับในความสัมพันธ์จะเปลี่ยนเป็นเครื่องหมายเท่ากับได้ถ้ากำหนดให้

$$H(f) = kG^*(f)\exp(-j\pi fT) \tag{2.15}$$

โดย G*(f) แทนค่าสังยุคเชิงซ้อนของ G(f) ฉะนั้น n จะมีค่าสูงสุดเมื่อเงื่อนไขตามสมการที่ (2.15) เป็นจริง ดังนั้น

$$\eta_{\max} = \frac{2}{N_0}\int_{-\infty}^{\infty}|G(f)|^2df \tag{2.16}$$

สมการความสัมพันธ์ที่ได้บ่งว่าค่าของ η ขึ้นอยู่กับพลังงานของสัญญาณหรือ $\int_{-\infty}^{\infty}|G(f)|^2df$ และความหนาแน่นสเปกตรัมกำลังของสัญญาณรบกวนเท่านั้น มิได้ขึ้นกับฟังก์ชันถ่ายโอนของวงจรกรอง H(f) แต่อย่างใด อย่างไรก็ตาม η จะมีค่าสูงสุดได้เมื่อเลือกฟังก์ชันถ่ายโอน H(f) ให้มีค่าตรงตามสมการที่ (2.15) และจะแทนฟังก์ชันถ่ายโอนดังกล่าวด้วย $H_{opt}(f)$ สำหรับค่าฟังก์ชันตอบสนองอิมพัลส์ของวงจรแมตช์ฟิลเตอร์หาได้จากการแปลงฟูเรียร์ผกผันกับ $H_{opt}(f)$ และผลลัพธ์ที่ได้คือ

$$\begin{aligned} h_{opt}(t) &= \int_{-\infty}^{\infty} H_{opt}(f)\exp(j\pi ft)df \\ &= \int_{-\infty}^{\infty} kG^*(f)\exp(-j\pi fT)\exp(j\pi ft)df \end{aligned} \tag{2.17}$$

ในกรณีของสัญญาณค่าจริงจะได้ว่า $G^*(f) = G(-f)$ ดังนั้นสมการที่ (2.17) สามารถเขียนใหม่ได้เป็น

$$\begin{aligned} h_{opt}(t) &= \int_{-\infty}^{\infty} kG(-f)\exp\left[-j\pi f(T-t)\right]df \\ &= kg(T-t) \end{aligned} \tag{2.18}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้แสดงให้เห็นว่าฟังก์ชันตอบสนองอิมพัลส์ของวงจรกรองที่ใช้มีรูปลักษณ์ที่เหมือนกับสัญญาณ g(t) ที่ผ่านการพลิกกลับทางเวลาพร้อมกับการประวิงเวลาไปเท่ากับ T จากความเหมือนกันของฟังก์ชันตอบสนองอิมพัลส์ของวงจรภาครับกับสัญญาณ g(t) วงจรดังกล่าวจึงมีชื่อว่า วงจรแมตช์ฟิลเตอร์ ข้อสังเกตที่เกี่ยวกับคุณลักษณะของวงจรแมตช์ฟิลเตอร์ คือ สัญญาณ g(t) บางรูปแบบอาจส่งผลให้ฟังก์ชันตอบสนองอิมพัลส์ $h_{opt}(t)$ มีค่าไม่เป็นศูนย์สำหรับเวลา $t < 0$ ซึ่งในกรณีดังกล่าวเราจะมาสามารถสร้างวงจรกรองขึ้นจริงในทางปฏิบัติได้เพราะฟังก์ชันตอบสนองอิมพัลส์ $h_{opt}(t)$ ที่ต้องการขาดคุณสมบัติคอซอล อย่างไรก็ตามเราสามารถเลือกค่าเวลาประวิงออกไปให้นานพอที่จะกะประมาณว่าค่าของฟังก์ชัน $h_{opt}(t)$ ที่ $t < 0$ มีค่าน้อยมากและสามารถละทิ้งได้

พิจารณาดูความสัมพันธ์ตามสมการที่ (2.16) อีกครั้ง สังเกตว่าค่าสูงสุด η_{max} ขึ้นอยู่กับพจน์ $\int_{-\infty}^{\infty}|G(f)|^2\,df$ ซึ่งแท้จริงแล้วพจน์นี้เป็นค่าพลังงาน E_g ของสัญญาณ g(t) นั่นเอง ดังนั้นเราจะเขียนสมการ(2.16) ใหม่ในรูปดังต่อไปนี้

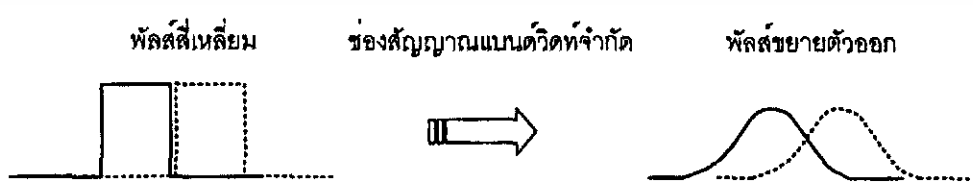
$$\eta_{\max} = \frac{2E_g}{N_0} \tag{2.19}$$

ความสัมพันธ์นี้ระบุว่าการใช้วงจรแมตช์ฟิลเตอร์จะให้ค่าอัตราส่วน η สูงสุดตามความต้องการได้ โดยค่าที่คำนวณได้จะขึ้นอยู่กับพลังงานของสัญญาณ g(t) ที่เลือกใช้(E_g)และความหนาแน่นสเปกตรัม กำลังของสัญญาณรบกวน $(N_0/2)$ เท่านั้น มิได้ขึ้นกับรูปลักษณ์หรือรายละเอียดของสัญญาณ g(t) แต่อย่างใด

2.5 การแทรกสอดระหว่างสัญลักษณ์และการทำพัลส์เชปปิง

2.5.1 การแทรกสอดระหว่างสัญลักษณ์

ในการส่งสัญญาณดิจิตอลโดยใช้พัลส์รูปสี่เหลี่ยมจัดว่าเป็นวิธีที่ตรงไปตรงมา ข้อมูลแต่ละบิทหรือแต่ละสัญลักษณ์จะได้รับการส่งในช่วงเวลาของมันเองที่แยกจากข้อมูลบิตหรือสัญลักษณ์อื่นอย่างเด็ดขาด ดังนั้นรูปสัญญาณที่พิจารณาทางเวลาของการส่งแต่ละบิตหรือสัญลักษณ์ที่ติดกันจะไม่มีการซ้อนทับหรือเกยกันเลย ในสภาวะเช่นนี้มีความหมายว่าการส่งข้อมูลแต่ละบิตหรือแต่ละสัญลักษณ์จะไม่มีการรบกวนหรือแทรกแซงระหว่างกันแต่อย่างใด อย่างไรก็ตาม พัลส์รูปสี่เหลี่ยมมีสเปกตรัมความถี่ที่กว้างมาก ฉะนั้นหากส่งพัลส์สี่เหลี่ยมผ่านช่องสัญญาณที่มีแบนด์วิดธ์จำกัด รูปสัญญาณที่รับได้ ณ ที่ภาครับปลายทางจะมีความผิดเพี้ยนไปจากเดิม ดังตัวอย่างที่แสดงในรูป



รูปที่ 2.6 ตัวอย่างการรบกวนระหว่างสัญลักษณ์ของพัลส์รูปสี่เหลี่ยมสองขณะส่งผ่าน ช่องสัญญาณที่มีแบนด์วิดธ์จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตว่าพัลส์ที่ส่งออกมีการแผ่ออกทางเวลาและเริ่มมีการเกยกันกับบิตข้างเคียง ผลที่เกิดขึ้นนี้มีชื่อเรียกว่า การแทรกสอดระหว่างสัญลักษณ์ (ISI: Intersymbol Interference) หมายเหตุ เพื่อความสะดวกในการอธิบายและง่ายต่อการทำความเข้าใจ เนื้อหาที่จะกล่าวถึงต่อไปจะขอใช้คำว่าบิตแทนการใช้คำว่าสัญลักษณ์ ถึงแม้คำว่าสัญลักษณ์จะให้ความหมายที่ครอบคลุมได้กว้างกว่า

จากการที่ช่องสัญญาณสื่อสารในทางปฏิบัติมีขนาดแบนด์วิดธ์จำกัด การส่งบิตข้อมูลโดยใช้พัลส์สี่เหลี่ยมจึงก่อปัญหาของการแทรกสอดระหว่างสัญลักษณ์อย่างรุนแรง ซึ่งส่งผลให้ให้การตัดสินใจบิตที่ภาครับมีความผิดพลาดได้โดยง่าย ดังนั้นจึงทำให้เกิดแนวคิดในการพัฒนารูปแบบของพัลส์ที่มีคุณสมบัติดีและสอดคล้องกับช่องสัญญาณที่มีแบนด์วิดธ์จำกัด ทั้งนี้โดยทฤษฎีรูปพัลส์สัญญาณที่ดีจะต้องมีคุณสมบัติตรงตามข้อกำหนด 3 ส่วน หลักดังต่อไปนี้ คือ

1.รูปพัลส์สัญญาณที่ใช้สำหรับบิตข้อมูลหนึ่งจะต้องมีค่าเป็นศูนย์ ณ ตำแหน่งที่เป็นจุดการสุ่มสัญญาณสำหรับการตัดสินใจบิตของข้อมูลอื่น ๆ ทั้งหมด เว้นแต่ข้อมูลของบิตที่พิจารณาเอง

2.รูปพัลส์สัญญาณจะต้องมีสเปกตรัมแบนด์วิดธ์ที่จำกัดอยู่ไม่เกินค่าหนึ่ง ซึ่งขึ้นกับแบนด์วิดธ์ของช่องสัญญาณ

3.รูปพัลส์สัญญาณควรจะมีขนาดลดลงอย่างรวดเร็วนอกช่วงเวลาการส่งของบิตที่พิจารณา เพื่อลดผลกระทบที่มีต่อบิตข้างเคียงให้มากที่สุด

ข้อกำหนดที่ 1 มีความหมายว่ารูปพัลส์สัญญาณที่ใช้อาจจะแผ่ตัวทางเวลาจะไปเกยกับพัลส์สัญญาณของบิตข้างเคียงได้ หากแต่ในจังหวะเวลาที่สำคัญคือจังหวะเวลาการสุ่มค่าของบิตข้างเคียงเพื่อใช้ในการตัดสินใจ ขนาดพัลส์สัญญาณของบิตที่พิจารณา ณ เวลาดังกล่าวจะต้องมีค่าเป็นศูนย์พอดี หากเราสามารถหารูปพัลส์สัญญาณที่ทำให้เงื่อนไขดังกล่าวเป็นจริงแล้ว การแทรกสอดระหว่างสัญลักษณ์ จะไม่ส่งผลต่อการตัดสินใจข้างเคียงอื่นอีกต่อไป สังเกตว่าพัลส์รูปสี่เหลี่ยมมีคุณสมบัติตรงตามข้อกำหนดที่ 1 ทุกประการ อย่างไรก็ตามพัลส์รูปสี่เหลี่ยมขาดคุณสมบัติข้อที่ 2 เพราะมีสเปกตรัมความถี่ที่กว้างมาก ทำให้เมื่อพัลส์เดินทางผ่านช่องสัญญาณที่มีแบนด์วิดธ์ของพัลส์สัญญาณเป็นปัจจัยสำคัญที่ต้องมีการพิจารณาอย่างชัดเจนและเป็นระบบ สำหรับข้อกำหนดข้อที่ 3 ระบุว่าขนาดของพัลส์สัญญาณนอกช่วงเวลาการส่งของมันควรจะมีค่าลดลงอย่างรวดเร็ว เพื่อลดผลกระทบของการแทรกสอดระหว่างสัญลักษณ์ที่เกิดขึ้นเนื่องจากการคลาดเคลื่อนของจังหวะเวลาสุ่มค่าสัญญาณ ถึงแม้ว่าพัลส์สัญญาณที่เลือกใช้มีคุณสมบัติตรงตามข้อกำหนดที่ 1 และ 2 แล้วก็ตาม การทำงานของวงจรภาครับสัญญาณอาจจะไม่สมบูรณ์ถูกต้องตลอดเวลา เช่น สัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะเวลาการสุ่มสัญญาณมีความคลาดเคลื่อนไปเล็กน้อย ดังนั้นหากจังหวะเวลาการสุ่มสัญญาณผิดไปจากค่าที่ควรจะเป็น ค่าของสัญญาณ ณ จังหวะการสุ่มค่าของบิตข้างเคียงอาจจะมิได้เป็นศูนย์อีกต่อไปและย่อมส่งผลกระทบต่อการตัดสินของบิตข้างเคียงในรูปของการแทรกสอดระหว่างสัญลักษณ์อย่างหลีกเลี่ยงไม่ได้ฉะนั้นในทางปฏิบัติจำเป็นต้องมีการพิจารณาถึงปัญหาที่อาจเกิดขึ้นจากความคลาดเคลื่อนของจังหวะในการสุ่มด้วย และมีการออกแบบเผื่อไว้รองรับกับปัญหาดังกล่าวด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การทำพัลส์เชปปิงแบบช่องสัญญาณในควิสต์อุดมคติ

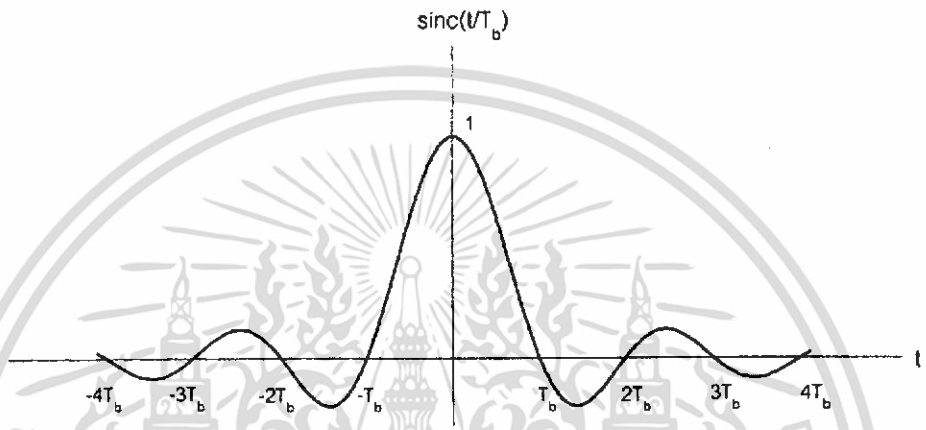
แนวทางในการค้นหารูปพัลส์สัญญาณที่เหมาะสมและมีคุณสมบัติตรงตามข้อกำหนดข้างต้นจะขอเริ่มจากการพิจารณาสัญญาณซิงก์

$$p(t) = \sin c(t/T_b) \tag{2.20}$$

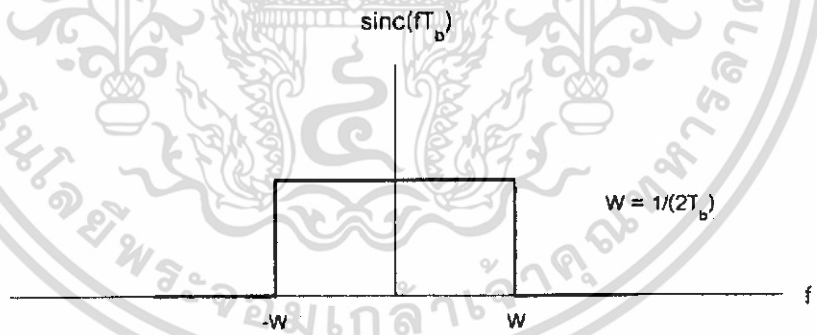
สัญญาณนี้มีสเปกตรัมทางความถี่เป็นฟังก์ชันพัลส์รูปสี่เหลี่ยม

$$P(f) = \prod(fT_b) \tag{2.21}$$

พิจารณาจากรูป



รูปที่ 2.7 สัญญาณซิงก์



รูปที่ 2.8 สเปกตรัมสัญญาณซิงก์

สังเกตว่าสัญญาณ p(t) มีค่าเป็นศูนย์ทุกจังหวะเวลา $t = nT_b$ โดย n เป็นจำนวนเต็มใด ๆ ที่ไม่ใช่ศูนย์ ซึ่งหมายความว่าหากระบบส่งข้อมูลด้วยอัตราบิตเท่ากับ $R_b = 1/T_b$ แล้ว การส่งด้วยสัญญาณซิงก์ดังกล่าวจะให้ผลตรงตามข้อกำหนดที่ 1 กล่าวคือ การแทรกสอดระหว่างสัญลักษณ์มีค่าเป็นศูนย์ (zero intersymbol interference) และเมื่อพิจารณาในทางความถี่พบว่าสัญญาณนี้มีแบนด์วิดธ์จำกัดเท่ากับ $W = R_b/2$ (ดูรูปที่ 2.8 ประกอบ) ฉะนั้นถ้าช่องสัญญาณที่ใช้ส่งมีขนาดแบนด์วิดธ์กว้างกว่า $B > W$ แล้วสัญญาณ

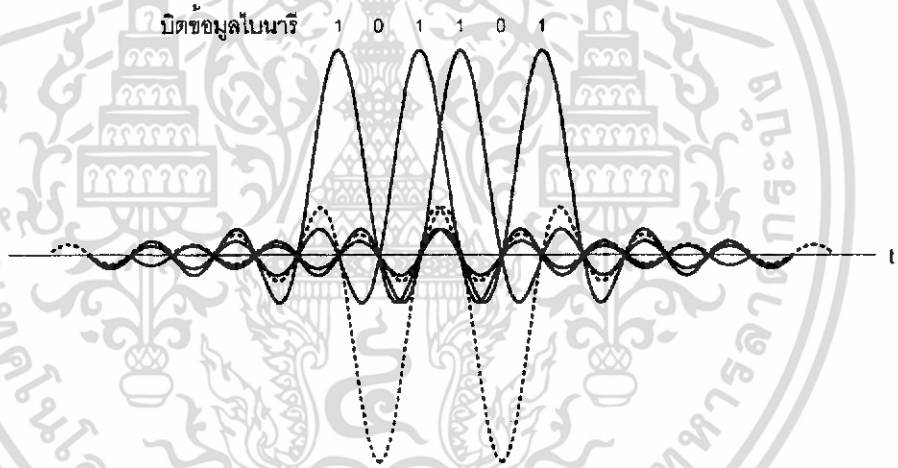
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ส่งผ่านจะไม่มีความผิดเพี้ยนเกิดขึ้นเลยจะเห็นว่าสัญญาณซิงก์สามารถตอบสนองตามข้อกำหนดข้อที่ 2 ด้วยเช่นกัน

จากที่กล่าวมาจะเห็นว่ารูปสัญลักษณ์ p(t) มีคุณสมบัติที่ดีตรงตามข้อกำหนดที่ต้องการ กล่าวคือ สามารถแก้ปัญหาการรบกวนกันระหว่างสัญลักษณ์โดยใช้แบนด์วิดธ์ในการส่งสัญญาณแคบที่สุดเท่าที่จะเป็นไปได้ และโดยทั่วไปจะเรียกรูปสัญญาณแบบนี้ว่าช่องสัญญาณในควิสต์อุดมคติ (Ideal Nyquist Channel) อย่างไรก็ตาม การนำรูปสัญญาณซิงก์ p(t) ดังกล่าวนี้มาใช้งานเชิงปฏิบัติต้องเผชิญกับปัญหาหรือข้อจำกัดที่สำคัญสองประการ

ประการแรกรูปสัญญาณซิงก์ขาดคุณสมบัติที่ดีตามข้อกำหนดข้อที่ 3 เพราะค่าของพัลส์สัญญาณมีขนาดลดลงด้วยอัตราเท่ากับ 1/t ซึ่งนับว่ายังค่อนข้างช้าเกินไป หากมีความคลาดเคลื่อนของจังหวะการสุ่มค่าของบิตเพียงเล็กน้อย การแทรกสอดระหว่างสัญลักษณ์จะมีค่าไม่เป็นศูนย์และมีแนวโน้มเพิ่มขึ้นอย่างรวดเร็วตามขนาดความคลาดเคลื่อนทางเวลาในการสุ่มค่าบิต

ประการที่สองในทางปฏิบัติการสร้างพัลส์สัญญาณซิงค์ซึ่งมีค่าตลอดทุกช่วงเวลาเป็นเรื่องที่เป็นไปได้ยาก ดูตัวอย่างรูปสัญญาณในกรณีที่ใช้สัญญาณซิงก์สำหรับส่งข้อมูลไบนารี 101101 ในรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างการใช้สัญญาณซิงก์สำหรับส่งข้อมูลไบนารี 101101

2.5.3 การทำพัลส์เชปปิ่งด้วยเรสโคไซน์พัลส์

จากการที่สัญญาณซิงก์มีคุณสมบัติตรงตามข้อกำหนดที่ต้องการถึง 2 ใน 3 ข้อคือ สามารถแก้ปัญหาการรบกวนกันระหว่างสัญลักษณ์ ได้โดยที่ใช้แบนด์วิดธ์ในการส่งสัญญาณแคบที่สุด ก่อให้เกิดแนวคิดในการปรับปรุงรูปพัลส์สัญญาณซิงก์เพิ่มเติมเพื่อให้พัลส์สัญญาณมีขนาดลดลงในอัตราที่รวดเร็วขึ้น ในขณะเดียวกันก็ต้องไม่ส่งผลกระทบต่อคุณสมบัติสองข้อแรกแนวทางที่ได้รับความสนใจมากเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

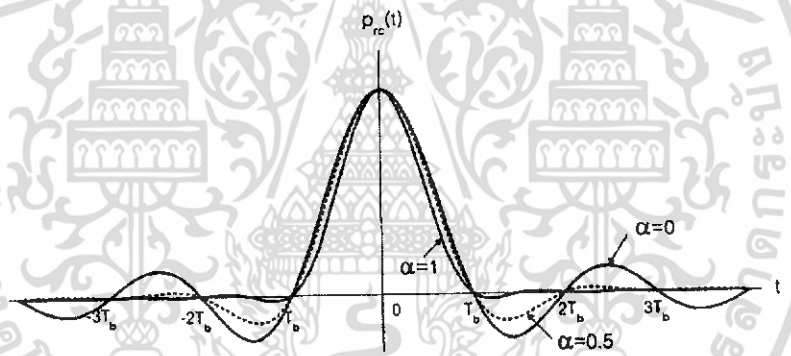
พิเศษคือ การทำพัลส์เชปปิงในรูปของเรสโคไซน์พัลส์ (Raised-cosine pulse) รูปลักษณ์ของฟังก์ชันเรสโคไซน์พัลส์ในทางเวลาเป็นดังนี้คือ

$$p_{rc}(t) = \frac{\cos(\pi\alpha t/T_b)}{1-(2\alpha t/T_b)^2}\sin c\left(\frac{t}{T_b}\right) \tag{2.22}$$

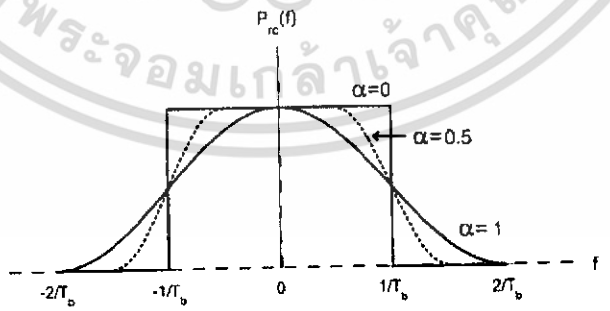
ส่วนสเปกตรัมความถี่มีลักษณะเป็น

$$p_{rc}(t) = \begin{cases} T_b, & |f| \le \dfrac{1-\alpha}{2T_b} \\ \dfrac{T_b}{2}\left(1+\cos\left[\dfrac{\pi T_b}{\alpha}\left(|f|-\dfrac{1-\alpha}{2T_b}\right)\right]\right), & \dfrac{1-\alpha}{2T_b} < |f| \le \dfrac{1+\alpha}{2T_b} \\ 0, & |f| > \dfrac{1+\alpha}{2T_b} \end{cases} \tag{2.23}$$

โดย α มีชื่อเฉพาะเรียกว่าตัวประกอบโรลล์ออฟ (roll-off factor) หากนำสมการทั้งสองมาวาดเป็นรูปกราฟสำหรับค่า α = 0,0.5 และ 1 จะได้ผลดังแสดงในรูป



รูปที่ 2.10 สัญญาณทางเวลา



รูปที่ 2.11 สัญญาณทางสเปกตรัมความถี่

รูปที่ 2.10 และ 2.11 ตัวอย่างรูปสัญญาณทางเวลาและสเปกตรัมความถี่ของฟังก์ชันเรสโคไซน์พัลส์สำหรับค่า α = 0,0.5 และ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาจากรูปจะเห็นว่าฟังก์ชันเรสโคไซน์มีขนาดแบนด์วิดธ์เพิ่มขึ้นตามค่าตัวประกอบโรลล์ออฟ เริ่มต้นที่ $\alpha = 0$ ฟังก์ชันเรสโคไซน์จะมีลักษณะเหมือนสัญญาณซิงก์ทุกประการคือมีขนาดแบนด์วิดธ์ที่แคบมาก เมื่อเพิ่มค่า α ขึ้นเป็น 0.5 แบนด์วิดธ์ของสัญญาณขยายขนาดขึ้น แต่การแกว่งตัวขึ้นลงของพัลส์อยู่ในช่วงที่แคบลงเมื่อเทียบกับฟังก์ชันซิงก์ และเมื่อเพิ่มค่าตัวประกอบโรลล์ออฟขึ้นจนเท่ากับ $\alpha = 1$ ฟังก์ชันเรสโคไซน์จะกลายเป็นสัญญาณที่มีแบนด์วิดธ์กว้างขึ้นเป็นสองเท่าเมื่อเทียบกับฟังก์ชันซิงก์ แต่รูปสัญญาณทางเวลาจะมีการแกว่งตัวน้อยลงมากเมื่อเทียบกับสัญญาณซิงก์ ข้อสังเกตที่เห็นนี้สามารถกล่าวเป็นข้อสรุปที่น่าสนใจดังนี้

สัญญาณเรสโคไซน์ $p_{rc}(t)$ มีค่าเป็นศูนย์ทุกจังหวะเวลา $t = nT_b$ โดย n เป็นจำนวนเต็มใด ๆ ที่ไม่ใช่ศูนย์ ซึ่งหมายความว่าหากระบบส่งข้อมูลด้วยอัตราบิตเท่ากับ $R_b = 1/T_b$ แล้วการส่งด้วยสัญญาณเรสโคไซน์จะให้ผลตรงตามข้อกำหนดที่ 1 กล่าวคือ การแทรกสอดระหว่างสัญลักษณ์มีค่าเป็นศูนย์ (zero intersymbol interference) และเมื่อพิจารณาในทางความถี่พบว่าสัญญาณนี้มีแบนด์วิดธ์ W จำกัดอยู่ในช่วง $R_b/2$ ถึง R_b ฉะนั้นถ้าช่องสัญญาณที่ใช้ส่งมีขนาดแบนด์วิดธ์กว้างกว่า $B > W$ แล้วสัญญาณที่ส่งผ่านจะไม่มีความผิดเพี้ยนเกิดขึ้นเลย จะเห็นว่าสัญญาณเรสโคไซน์สามารถตอบสนองตามข้อกำหนดข้อที่ 2 ด้วยเช่นกัน นอกจากนี้สัญญาณเรสโคไซน์ยังมีคุณสมบัติที่ดีตรงตามข้อกำหนดข้อที่ 3 ด้วย เพราะค่าของพัลส์สัญญาณมีขนาดลดลงอย่างรวดเร็วเมื่ออยู่นอกช่วงเวลาของบิตที่พิจารณา ดังนั้นแม้จะมีความคลาดเคลื่อนของจังหวะการสุ่มค่าของบิตเพียงเล็กน้อย ณ วงจรภาครับ การแทรกสอดระหว่างสัญลักษณ์จะมียังค่าที่ไม่สูงนัก จากการที่สัญญาณเรสโคไซน์มีคุณสมบัติที่ดีตรงทั้งสามข้อนั่นเองทำให้รูปสัญญาณประเภทนี้ได้รับความสนใจมากเป็นพิเศษ สำหรับการทำพัลส์เชปปิงในระบบการส่งผ่านสัญญาณสื่อสาร

2.6 การแปลงอัตราการสุ่ม (Sampling rate Conversion)

กระบวนการมัลติเรท (Multirate) ของสัญญาณดิจิตอล (digital) ระบบนี้สามารถลดหรือเพิ่มระยะห่างระหว่างแต่ละแซมเปิล (sample) ของแต่ละสัญญาณ ก่อนการประมวลผลของสัญญาณหรือขณะการประมวลผลของสัญญาณ จากนั้นสัญญาณเหล่านี้จะมีอัตราการสุ่มที่ต่างกัน

2.6.1 สัญญาณไม่ต่อเนื่อง (Representation of Discrete signal)

2.6.1.1 การสุ่มสัญญาณไม่ต่อเนื่อง (Discrete Sampling)

สัญญาณไม่ต่อเนื่องถูกอธิบายโดยใช้จำนวนเชิงซ้อน

$$W_M = \exp\left(-j2\pi / M\right) = \sqrt[M]{1} \tag{2.24}$$

เพื่ออธิบายการสุ่มของสัญญาณไม่ต่อเนื่อง เราสามารถใช้ดิสครีทแซมปลิงฟังก์ชัน (discrete sampling fuction) ดังสมการ (2.25)

$$w_M\left(n\right) = \frac{1}{M}\sum_{v=0}^{M-1} W_M^{vn} = \begin{cases} 1 & , for\ n = mM,\ m\ \text{integer} \\ 0 & , otherwise \end{cases} \tag{2.25}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า $n = mM$ ซึ่ง m เป็นจำนวนเต็มและ $W_M^{vn} = 1$ เมื่อบวก W_M^{vn} เข้าด้วยกัน M ซึ่งก็ได้ค่าเท่ากับ M สำหรับค่า n อื่น ๆ W_M^{vn} จะถูกแจกจ่ายเท่า ๆ กัน บนวงกลมหนึ่งหน่วยและผลรวมในสมการ (2.25) เท่ากับศูนย์

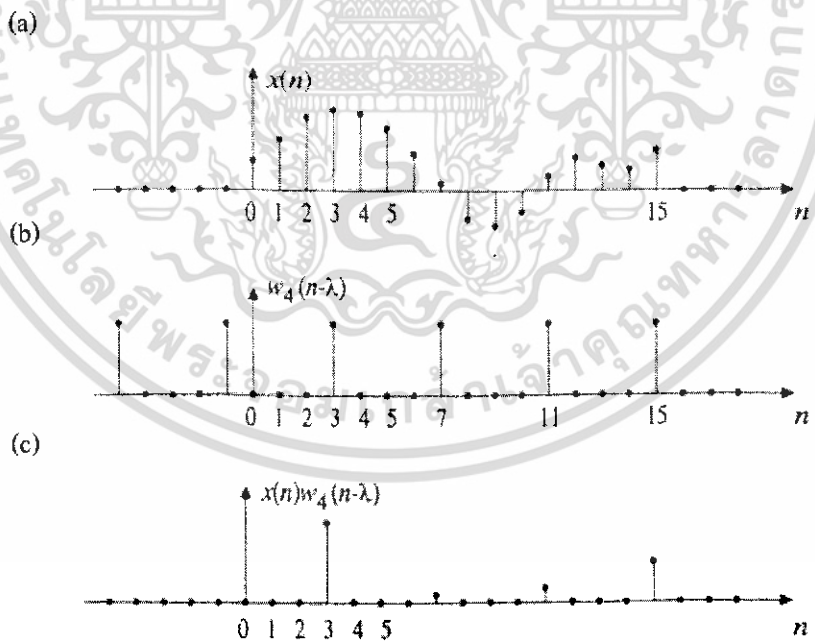
เนื่องจากค่าของดิสครีทแซมปลิงฟังก์ชันเท่ากับ 1 ถ้า n คือจำนวนเท่าของ M แต่ถ้าไม่ใช่ก็เท่ากับ 0 จะได้สมการที่เทียบเท่ากันคือ

$$w_M(n) = w_M(-n) = \frac{1}{M}\sum_{v=0}^{M-1} W_M^{-vn} \tag{2.26}$$

ปกติแล้วดิสครีทแซมปลิงจะทำโดยไม่ใช้เฟสออฟเซท (Phase offset) เช่นค่าสัญญาณที่ถูกสุ่มจะมีแต่ค่า $x(n)$ ซึ่งค่า $n = mM$ แต่ก็เป็นไปได้ที่จะใช้เฟส λ ในกรณีนี้ค่า $x(n)$ ที่ยังคงเหลือหลังจากการแซมปลิงจะมี $n = mM + \lambda$ เพื่อจะให้ได้ผลดังกล่าวสัญญาณเดิมจะต้องถูกคูณด้วยแซมปลิงฟังก์ชันซึ่งมีเฟสออฟเซท λ ดังสมการ (2.27)

$$w_M(n-\lambda) = \frac{1}{M}\sum_{v=0}^{M-1} W_M^{v(n-\lambda)} = \begin{cases} 1 & ,for\ n = \lambda + mM\ ,\ m\ \text{integer} \\ 0 & ,otherwise \end{cases} \tag{2.27}$$

รูป 2.12(a) แสดงสัญญาณดิสครีท $x(n)$, รูปที่ 2.12(b) เป็นดิสครีทแซมปลิงฟังก์ชัน $w_M(n-\lambda)$ และรูปที่ 2.12(c) คือสัญญาณที่สุ่มแบบดิสครีท $x(n)$ ที่มีเฟสออฟเซท $=\lambda$



รูปที่ 2.12 การสุ่มแบบดิสครีทที่มีเฟสออฟเซท λ =3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1.2 โพลีเฟส (Polyphase Representation)

เมื่อกำหนดค่าคงที่ M มา เราสามารถหาสัญญาณที่ถูกสุ่มแบบดิสครีทที่แตกต่างกัน M สัญญาณ ซึ่งแต่ละสัญญาณจะมีเฟสออฟเซทที่แตกต่างกันออกไป รูป 2.13 แสดงสัญญาณที่มีความยาวจำกัด $x(n)$ และสัญญาณที่ถูกสุ่มแบบดิสครีทที่แตกต่างกัน 4 สัญญาณ $x_\lambda^{(p)}, \lambda = 0,1,2,3$ ซึ่งแต่ละสัญญาณที่ได้มาจากการสุ่มทุก 4 ค่าของสัญญาณ $x(n)$ จากรูป 2.13 พบว่า $x(n)$ สามารถแสดงได้ว่าเป็นผลรวมของแต่ละสัญญาณที่แตกต่างกันทั้ง 4 ซึ่งแต่ละสัญญาณนั้นก็สามารถอธิบายได้โดยใช้ดิสครีทแซมปลิงฟังก์ชันดังสมการที่ (2.27)

$$\begin{aligned} x(n) &= x_0^{(p)}(n) + x_1^{(p)}(n) + x_2^{(p)}(n) + x_3^{(p)}(n) \\ &= x(n)w_4(n) + x(n)w_4(n-1) + x(n)w_4(n-2) + x(n)w_4(n-3) \end{aligned} \quad (2.28)$$

โดยทั่วไป

$$x(n) = \sum_{\lambda=0}^{M-1} x_\lambda^{(p)}(n) = \sum_{\lambda=0}^{M-1} x(n) \cdot w_M(n-\lambda) \quad (2.29)$$

สมการนี้จะเรียกว่าเป็นการแสดงรูปแบบโพลีเฟส(polyphase representation) ของสัญญาณ $x(n)$ ในไทม์โดเมน แต่ละสัญญาณ $x_\lambda^{(p)}(n)$ จะถูกเรียกว่าเป็นองค์ประกอบของ $x(n)$ การแสดงรูปแบบโพลีเฟสยังขึ้นอยู่กับค่า M ด้วย

การแปลง Z $$X(z) = \sum_{n=\infty}^{\infty} x(n) \cdot z^{-n} \quad (2.30)$$

สามารถแบ่งเป็นสับซิกแนล (sub-signal) M สัญญาณยกตัวอย่างเช่น สำหรับสัญญาณที่จำกัด $x(n)$ ในรูป 2.13

เรามี

$$\begin{aligned} X(z) &= x(0)z^{-0} + x(4)z^{-4} + x(8)z^{-8} + x(12)z^{-12} \\ &= x(1)z^{-1} + x(5)z^{-5} + x(9)z^{-9} + x(13)z^{-13} \\ &= x(2)z^{-2} + x(6)z^{-6} + x(10)z^{-10} + x(14)z^{-14} \\ &= x(3)z^{-3} + x(7)z^{-7} + x(11)z^{-11} + x(15)z^{-15} \end{aligned} \quad (2.31)$$

ดึงแฟกเตอร์ $z^{-\lambda}$ เมื่อ $\lambda = 0...3$ ออกมาจะได้

$$\begin{aligned} X(z) &= z^{-0}\left[x(0)z^{-0} + x(4)z^{-4} + x(8)z^{-8} + x(12)z^{-12}\right] \\ &= z^{-1}\left[x(1)z^{-0} + x(5)z^{-4} + x(9)z^{-8} + x(13)z^{-12}\right] \end{aligned}$$

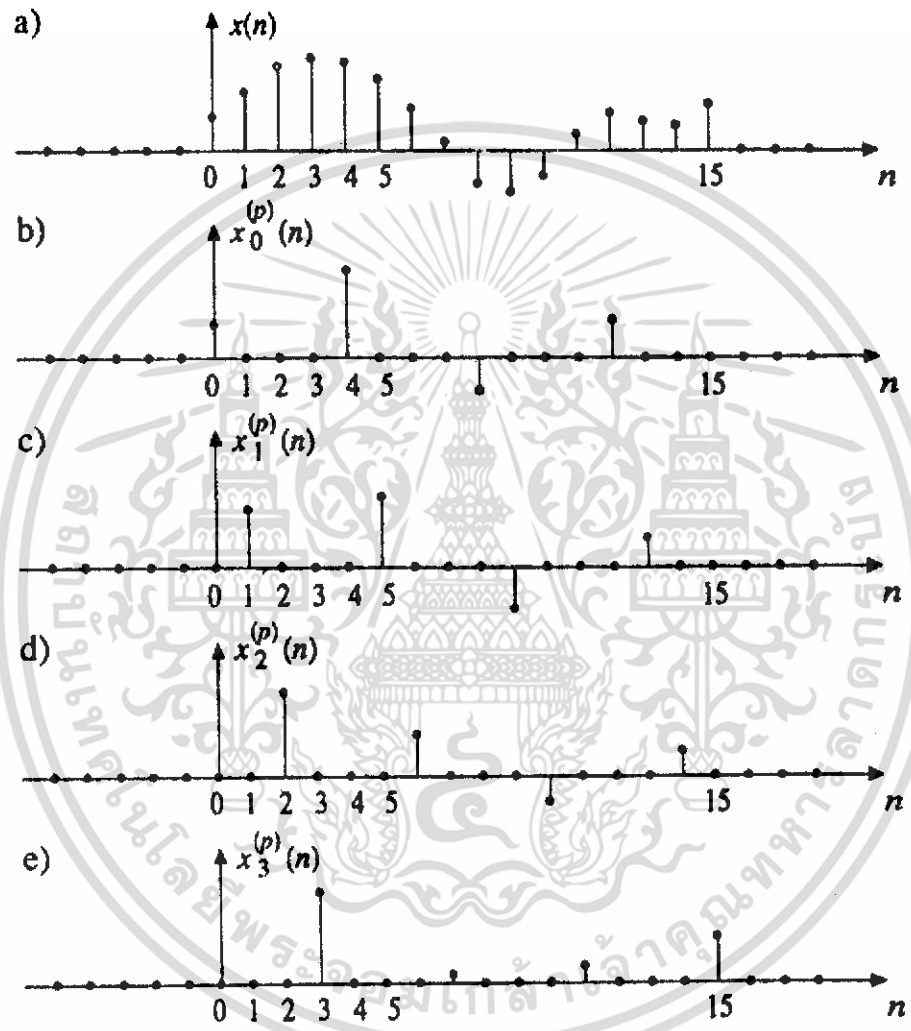
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

62540

$$= z^{-2}\left[x(2)z^{-0} + x(6)z^{-4} + x(10)z^{-8} + x(14)z^{-12}\right]$$
$$= z^{-3}\left[x(3)z^{-0} + x(7)z^{-4} + x(11)z^{-8} + x(15)z^{-12}\right]$$

(2.32)

การดึงแฟกเตอร์นี้ออกมาทำให้องค์ประกอบในวงเล็บลดลงสู่พหุนามใน z^{-4}



รูปที่ 2.13 การแสดงโพลีเฟสของสัญญาณดิสครีท

โดยทั่วไปสำหรับการแปลง Z ที่สอดคล้องกับสมการ (2.30) และ $n = mM + \lambda$ สามารถกล่าวได้ว่า

$$X(z) = \sum_{\lambda=0}^{M-1}\sum_{m=-\infty}^{\infty} x(mM+\lambda)\cdot z^{-(mM+\lambda)} = \sum_{\lambda=0}^{M-1} z^{-\lambda} X_{\lambda}^{(p)}\left(z^{M}\right) \qquad (2.33)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง

$$X_{\lambda}^{(p)}\left(z^{M}\right)=\sum_{m=-\infty}^{\infty}x\left(mM+\lambda\right)\cdot z^{-mM} \tag{2.34}$$

สมการ(2.33) เรียกว่าโพลีเฟสของการแปลง Z หรือ $X\left(z\right)$ พหุนาม $x_{\lambda}^{(p)}\left(z\right)$ จะให้องค์ประกอบในวงเล็บในสมการ(2.32) ยกตัวอย่างเช่นพหุนามตัวแรก

$$X_{\lambda}^{(p)}\left(z\right)=\left[x\left(0\right)z^{-0}+x\left(4\right)z^{-1}+x\left(8\right)z^{-2}+x\left(12\right)z^{-3}\right] \tag{2.35}$$

การเปรียบเทียบ สมการ(2.29) หรือรูป 2.13 กับ สมการ(2.33) แสดงให้เห็นถึงการแมปปิง(Mapping) แบบ 1 ต่อ 1 ระหว่างองค์ประกอบของโพลีเฟสในไทม์โดเมนและระหว่างองค์ประกอบของโพลีเฟสที่มีดัชนีเหมือนกันใน ฟรีเควนซีโดเมน

เนื่องจาก Z ทรานส์ฟอร์มมีคุณสมบัติการเป็นเชิงเส้นจึงกล่าวโดยทั่วไปได้ว่า

$$z^{-\lambda}X_{\lambda}^{(p)}\left(z^{M}\right)\longleftrightarrow x_{\lambda}^{(p)}\left(n\right),\ \lambda=0,1,2...M-1 \tag{2.36}$$

$\longleftrightarrow$ หมายถึง ความสอดคล้องกันระหว่างสัญญาณในโดเมนทรานส์ฟอร์ม (z- transform, Fourier transform) กับไทม์โดเมน

จากนั้นนำองค์ประกอบโพลีเฟสมารวมกันในรูปแบบของคอลัมน์เวกเตอร์ (Column vector)

$$x^{(p)}\left(z\right)=\left[X_{0}^{(p)}\left(z\right)\quad z^{-1}X_{1}^{(p)}\left(z\right)\ \ldots\ z^{-(M-1)}X_{(M-1)}^{(p)}\left(z\right)\right] \tag{2.37}$$

2.6.2 การลดอัตราการสุ่ม (Reducing Sampling rate)

การลดอัตราการสุ่มจะมีประโยชน์ถ้าอัตราการสุ่มการสุ่มนั้นมากกว่า 2 เท่าของแบนวิดธ์ของสัญญาณมาก การลดอัตราการสุ่มเรียกว่า แซมปลิงเรทเดซิเมชัน (Sampling rate decimation) หรือ เดซิเมชัน (decimation) ซึ่งประกอบด้วย 2 ขั้นตอนคือ แอนตีเอเลียสซิงฟิลเตอร์ (anti-aliasing filter) และ ดาวน์แซมปลิง (down sampling)

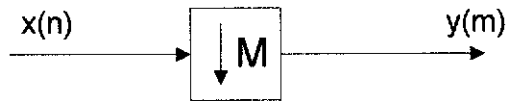
2.6.2.1 ดาวน์แซมปลิง (Downsampling)

อัตราการสุ่มสัญญาณดิสครีท $x\left(n\right)$ จะถูกลดด้วยแฟคเตอร์ M ความสัมพันธ์ระหว่างสัญญาณที่ได้ $y\left(m\right)$ และสัญญาณเดิม $x\left(n\right)$

$$y\left(m\right)=x\left(m\cdot M\right) \tag{2.38}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.14 แสดงการไหลของสัญญาณในกระบวนการนี้



รูปที่ 2.14 ดาวน์แซมเพลอร์ (Downsampler)

สัญลักษณ์สี่เหลี่ยมในรูปที่ 2.14 และลูกศรชี้ลงแสดงการถึง ดาวน์แซมเพลอร์, สัญญาณเอาท์พุท $y(m)$ เรียกสัญญาณที่ถูกดาวน์แซมเพลอร์ การทำดาวน์แซมปลิงของสัญญาณยังสามารถอธิบายโดยใช้การแสดงรูปแบบโพลีเฟสซึ่งกระบวนนี้สามารถทำได้ 2 ขั้นตอน รูปที่ 2.15(a) แสดงสัญญาณดิสครีท $x(n)$ ส่วนองค์ประกอบโพลีเฟส $x_0^{(p)}(n)$ ดังแสดงในรูปที่ 2.15(b) นั้นได้มาจากการแซมปลิงแบบดิสครีท เช่น การคูณโดยดิสครีทแซมปลิงฟังก์ชัน $w_M(n)$ ที่แสดงในสมการ(2.25) สัญญาณที่ถูกดาวน์แซมปลิง $y(m)$ ในรูป 2.1(c) จึงสามารถได้มาโดยง่ายโดยการเหลือศูนย์จำนวน M-1 ตัว ระหว่างแต่ละแซมเปิล (sample)

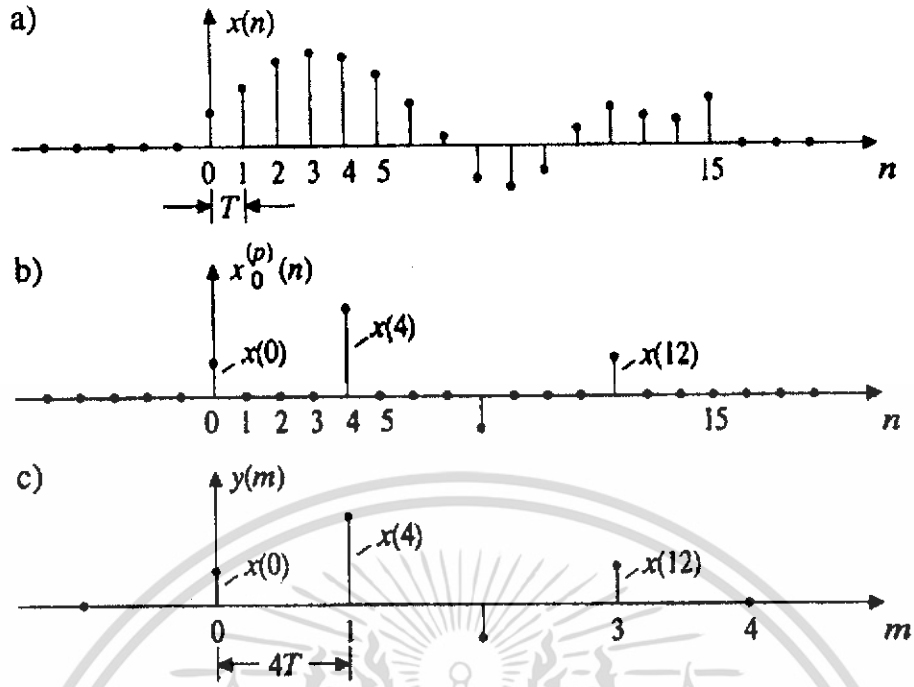
ใน Z โดเมน เราสามารถใช้การแปลง Z ของสัญญาณเดิม

$$X(z)=\sum_{n=-\infty}^{\infty} x(n)\cdot z^{-n} \tag{2.39}$$

และ สมการ(2.34) โดย $\lambda = 0$ เพื่อให้ได้มาซึ่งการแปลง Z ของสัญญาณที่ถูกสุ่มแบบดิสครีท ดังแสดงในรูปที่ 2.15

$$X_0^{(p)}(z^M)=\sum_{m=-\infty}^{\infty} x(m\cdot M)\cdot z^{-mM} \tag{2.40}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ขั้นตอนการประมวลผลสัญญาณที่ใช้ในการดาวน์แซมปลิง

2.6.2.2 การกรองสัญญาณและสเปกตรัมการสุ่มค่าตัวอย่าง (Filtering and Spectrum of the Downsampled signal)

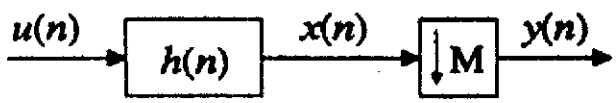
จากหลักการของการประมวลผลสัญญาณดิจิตอล การกรองสัญญาณสามารถทำได้โดยการทำคอนโวลูชันระหว่างสัญญาณอินพุทกับสัมประสิทธิ์ของวงจรกรองความถี่ (Filter coefficients or impulse response) โดยถ้าสมมติให้อินพุทคือ $u(n)$ และสัมประสิทธิ์ของวงจรกรองความถี่คือ $h(n)$ จะได้เอาท์พุทคือ $x(n)$ ดังสมการ

$$x(n) = \sum_{k=0}^{N-1} u(k)h(n-k) \tag{2.41}$$

จากรูปที่ 2.16 และ สมการ(2.38)

$$y(m) = x(m \cdot M) \tag{2.42}$$

การสุ่มค่าตัวอย่างจะเป็นการนำอินพุต $x(n)$ สร้างออกมาเป็นเอาท์พุต $y(m) = x(2m)$ ดังรูป



รูปที่ 2.16 ตัวสุ่มค่าตัวอย่าง (Down Sampler or Decimator)

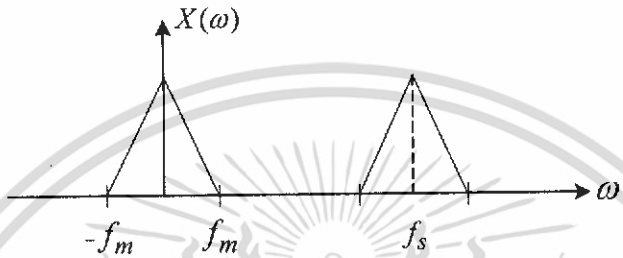
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำนอร์มอลไลซ์ (normalize) : $2f_c = 2\pi$ ∴ $f_c = \frac{\pi f_m}{\quad}$ (2.46)

จาก $$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (2.43)$$

และ y(m) = x(2m) ; $$Y(\omega) = \sum_{m=-\infty}^{\infty} x(2m)e^{-j\omega m} \quad (2.44)$$

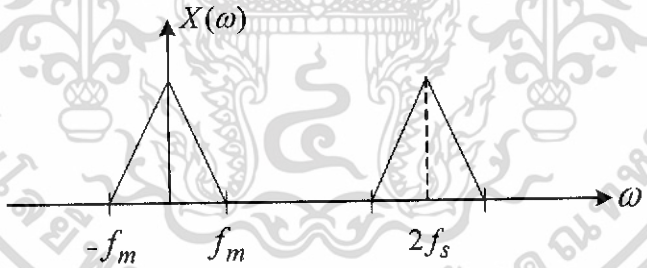
ถ้าความถี่สูงสุดของสัญญาณเป็น f_m และสุ่มสัญญาณด้วยความถี่ f_s จะไม่ทำให้เกิดเอเลียสซิง (aliasing) : $f_s \geq 2f_m$ ดังนั้นเมื่อทำการแซมปลิงด้วย f_s จะได้สเปกตรัมของสัญญาณเป็น



รูปที่ 2.17 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มสัญญาณด้วยความถี่ f_s

ถ้านอร์มอลไลซ์ (normalize) โดย $f_s = 2\pi$; $\therefore \quad f_m = \dfrac{2\pi f_m}{f_s}$ (2.45)

และถ้าแซมปลิงด้วย $2f_s$ จะได้สเปกตรัมของสัญญาณเป็น



รูปที่ 2.18 สเปกตรัมของสัญญาณ $X(\omega)$ เมื่อสุ่มสัญญาณด้วยความถี่ $2f_s$

ถ้านอร์มอลไลซ์ (normalize) : $2f_s = 2\pi$; $\therefore \quad f_m = \dfrac{\pi f_m}{f_s}$ (2.46)

จะเห็นว่าสเปกตรัมของการแซมปลิงด้วย $2f_s$ จะได้ $f_m = \dfrac{2\pi f_m}{f_s}$

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

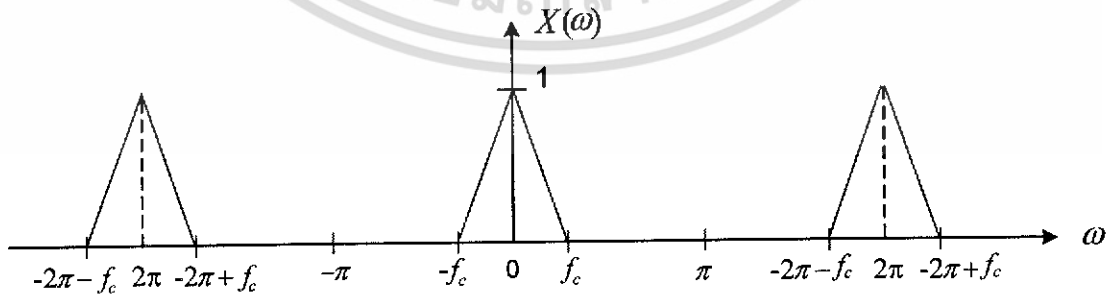
จาก
$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$
$$= \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega 2n} - \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j\omega(2n-1)} \quad (2.47)$$

และ
$$X(\omega-\pi) = \sum_{n=-\infty}^{\infty} x(2n)e^{-j(\omega-\pi)2n} - \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j(\omega-\pi)(2n-1)}$$
$$= \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega 2n} + \sum_{n=-\infty}^{\infty} x(2n-1)e^{-j\omega(2n-1)} \quad (2.48)$$

นำสมการ $\frac{(2.47)+(2.48)}{2}$ จะได้
$$\frac{X(\omega)+X(\omega-\pi)}{2} = \sum_{n=-\infty}^{\infty} x(2n)e^{-j\omega(2n)} \quad (2.49)$$

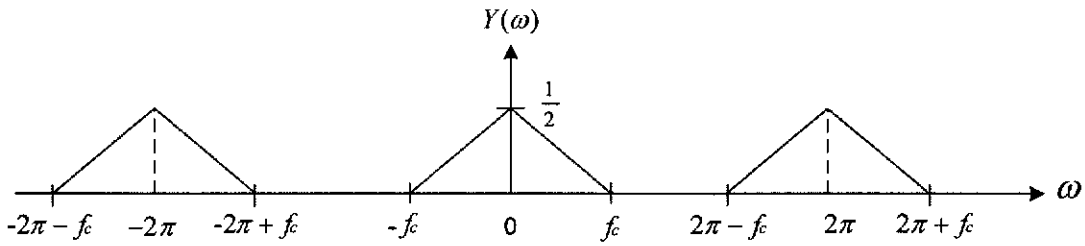
และถ้ากำหนดให้ y(m) = x(2m) จะได้
$$Y(\omega) = \sum_{m=-\infty}^{\infty} y(m)e^{-j\omega m}$$
$$= \sum_{m=-\infty}^{\infty} x(2m)e^{-j\frac{\omega}{2}(2m)}$$
$$= \frac{X(\frac{\omega}{2})+X(\frac{\omega}{2}-\pi)}{2} \quad (2.50)$$

เมื่อนำไปพล็อตกราฟเราได้



รูปที่ 2.19 สเปกตรัมของสัญญาณอินพุทก่อนการดาวน์แซมปลิง

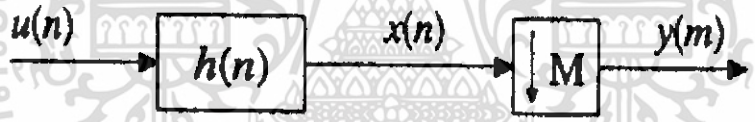
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 สเปกตรัมของสัญญาณเอาท์พุทหลังจากผ่านการดาวน์แซมปลิง

จากกราฟเราจะพบว่าการดาวน์แซมปลิง จะทำให้สเปกตรัมของสัญญาณอินพุทถูกขยายออกเป็น 2 เท่า และทำให้ขนาดลดลงครึ่งหนึ่ง

ถ้าต้องการไม่ให้เกิดการเอเลียสซิงจะต้องมีฟิลเตอร์กรองก่อนที่จะดาวน์แซมปลิง (เป็น LPF ที่มีคัทออฟ ที่ $\frac{\pi}{M}$) ซึ่งเรียกว่า เดซิเมชันฟิลเตอร์ (decimation filter) หรือ แอนตีเอเลียสซิง (anti-aliasing filter) เพื่อป้องกันไม่ให้เกิดการซ้อนทับเนื่องจากการขยายของความถี่ขึ้น M เท่า $\left(f_c < \frac{\pi}{M} \right)$ แสดงได้ดังรูปที่ 2.21



รูปที่ 2.21 แสดงเดซิเมเตอร์ที่ใช้ฟิลเตอร์ที่มีผลตอบสนองทางความถี่เป็น h(n) และดาวน์แซมเพลอร์ที่มีแฟคเตอร์คือ M

ดังนั้นจากข้างต้นสัญญาณที่ผ่านวงจรกรองความถี่คือ

$$\begin{aligned} y(n) &= u(n) * h(n) \\ &= \sum_{k=-\infty}^{\infty} u(k)h(n-k) \end{aligned} \tag{2.51}$$

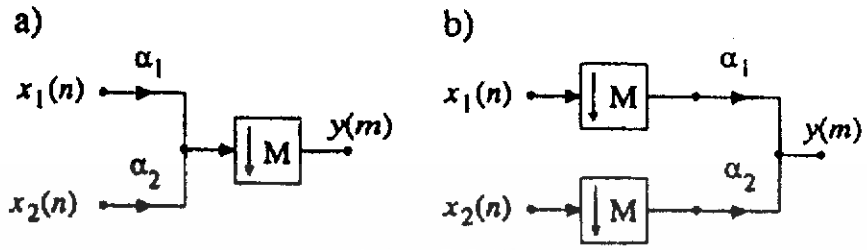
และเมื่อผ่านการดาวน์แซมปลิง ตามสมการ(2.42) กระบวนการทั้งหมดของการเดซิเมชันจึงเท่ากับ

$$y(n) = \sum_{k=-\infty}^{\infty} u(k) \cdot h(mM-k) \tag{2.52}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

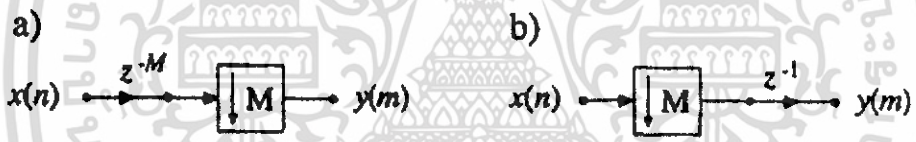
2.6.2.3 เอกลักษณ์ (Identities)

โครงสร้างการไหลของสัญญาณ (signal flow structure) และการรวมกันที่โหนด(node) ต่างๆ นั้น ไม่ขึ้นกับอัตราการสุ่ม รูปที่ 2.22 (a) และ 2.22 (b) จึงเทียบเท่ากันได้ ซึ่งเป็นเอกลักษณ์ที่สำคัญมาก



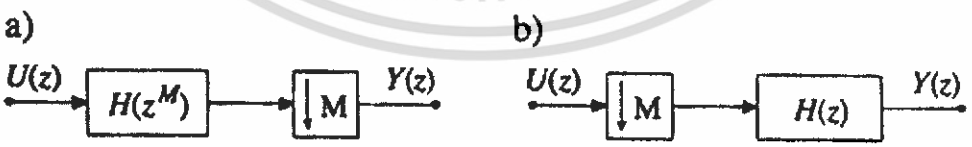
รูปที่ 2.22 เอกลักษณ์ที่ 1

เนื่องจากการหน่วงของคาบของการแซมเปิล M คาบก่อนดาวน์แซมปลิงนั้นเท่ากับการหน่วง ของ 1 คาบการแซมเปิลหลังการดาวน์แซมปลิง ได้เอกลักษณ์ที่ 2 ดังรูปที่ 2.23



รูปที่ 2.23 เอกลักษณ์ที่ 2

รูปที่ 2.24 แสดงรูปแบบทั่วไปของเอกลักษณ์ที่ 2



รูปที่ 2.24 เอกลักษณ์ที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

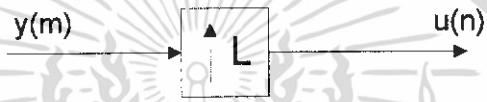
2.6.3 การเพิ่มอัตราการสุ่ม (Increasing Sampling rate)

2.6.3.1 อัพแซมปลิง (Upsampling)

อัตราการสุ่มของสัญญาณดิสครีท $y(m)$ จะถูกเพิ่มขึ้นด้วยแฟกเตอร์ L โดยการวางเลขศูนย์จำนวน $L-1$ ตัวโดยมีระยะห่างที่เท่ากันระหว่างแต่ละคู่ของแซมเปิลสัญญาณที่ได้คือ $u(n)$

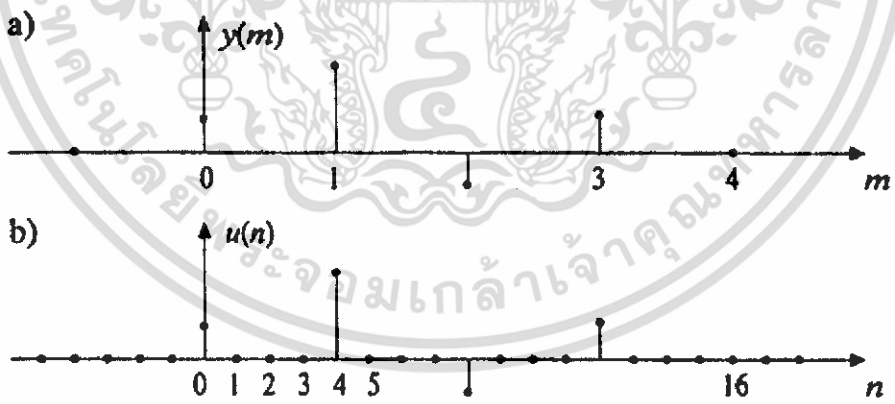
$$u(n) = \begin{cases} y(\frac{n}{2}) & \text{for n = mL, m integer} \\ 0 & \text{otherwise} \end{cases} \tag{2.53}$$

และมีสัญลักษณ์ที่เกี่ยวข้องแสดงดังรูปที่ 2.25



รูปที่ 2.25 อัพแซมเพลอร์ (upsampler)

รูปที่ 2.26 (a) แสดงสัญญาณ $y(m)$ และ รูปที่ 2.26 (b) แสดงสัญญาณที่ได้มาจากหลังการอัพแซมปลิงด้วยแฟคเตอร์ $L=4$ เปรียบเทียบรูปนี้กับรูปที่ 2.15 จะเห็นว่าสำหรับการสุ่มนั้นขั้นตอนมาจากรูปที่ 2.15 (b) และ รูปที่ 2.15(c) กลับกัน



รูปที่ 2.26 การอัพแซมปลิงของสัญญาณดิสครีท

ดังนั้นถ้าการแปลง Z

$$Y(z) = \sum_{m=-\infty}^{\infty} y(m)z^{-(m)} \tag{2.54}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสัญญาณ $y(m)$ ก่อนการอัพแซมปลิง ถูกระบุเป็น $Y(z^{'})$ จากสมการ (2.44) และการแปลง Z

$$U(z) = \sum_{m=-\infty}^{\infty} u(n)z^{-(n)} \tag{2.55}$$

ของสัญญาณอัพแซมปลิง $u(n)$ ด้วย $x_0^{(p)}(z^M)$ จากสมการ (2.44) เราสามารถใช้สมการ (2.44) เพื่อสรุปดังนี้

$$U(z) = Y(z^L) \tag{2.56}$$

2.6.3.2 การกรองสัญญาณและสเปกตรัมการสุ่มค่าตัวอย่าง (Filtering and Spectrum of the Upsampled signal)

จากเงื่อนไขการอัพแซมปลิง

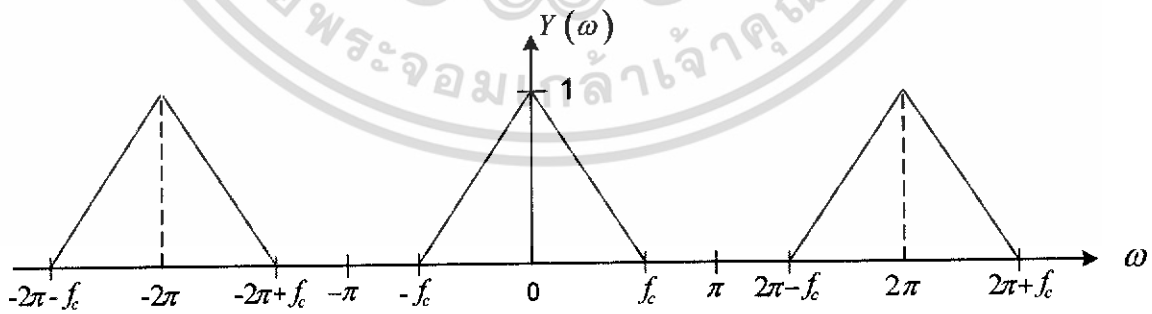
$$u(n) = \begin{cases} y(\frac{n}{2}) & n : even \\ 0 & n : odd \end{cases} \tag{2.57}$$

$$U(\omega) = \sum_{n=-\infty}^{\infty} u(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} u(2n)e^{-j\omega(2n)} \tag{2.58}$$

จาก u(n) = y(n /2) ;

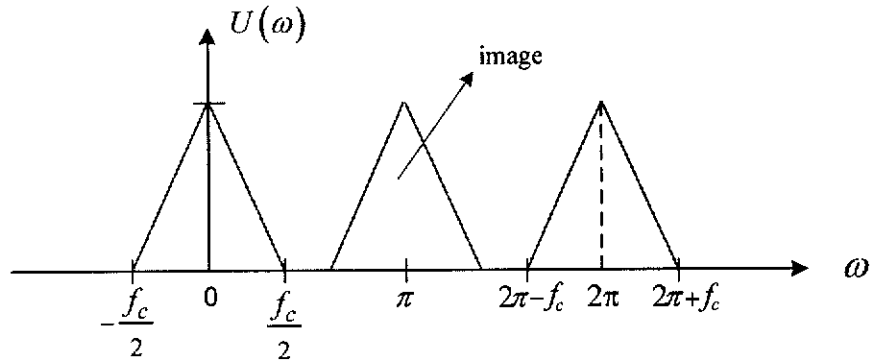
$$U(\omega) = \sum_{n=-\infty}^{\infty} y(n)e^{-j2\omega n} = Y(2\omega) \tag{2.59}$$

เมื่อนำไปพล็อตกราฟจะได้



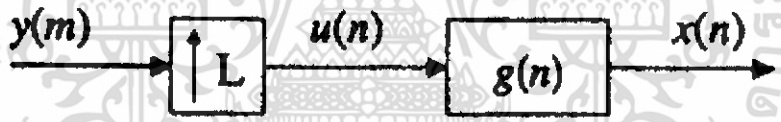
รูปที่ 2.27 สเปกตรัมของสัญญาณอินพุทก่อนการอัพแซมปลิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 สเปกตรัมของสัญญาณเอาท์พุทเมื่อผ่านการอัพแซมปลิง

จากกราฟเราจะพบว่าการอัพแซมปลิง จะทำให้ความกว้างสเปกตรัมของสัญญาณลดลงเหลือครึ่งหนึ่งของสัญญาณเดิม (เมื่อทำการอัพแซมปลิงที่มีแฟคเตอร์ L = 2 แต่ถ้าทำการอัพแซมปลิงที่ L ความกว้างของสเปกตรัมจะลดลง L เท่า ของสัญญาณเดิม) และจะเกิดสเปกตรัมของเงา (image) ขึ้น (เกิดจากการเติมศูนย์ในตำแหน่งคี่) ซึ่งเราต้องกรองสัญญาณเงาออกโดยใช้วงจรกรองความถี่ต่ำ เพื่อทำให้สเปกตรัมเหมือนเดิม เรียกตัวกรองความถี่ต่ำนี้ว่า อินเตอร์โพเลชันฟิลเตอร์ (interpolation filter) หรือ แอนตีอิมเมจจิงฟิลเตอร์ (anti-imaging filter) ดังแสดงดังรูป



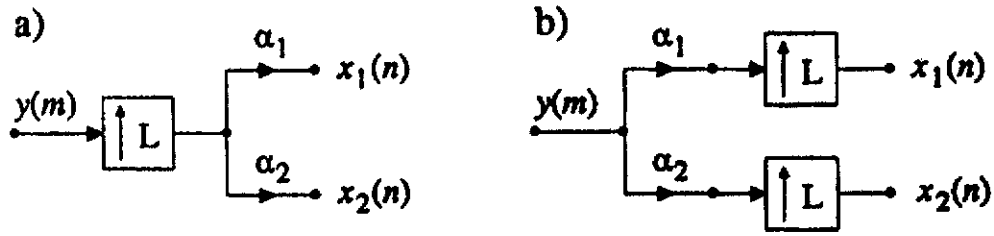
รูปที่ 2.29 อินเตอร์โพเลชันโดยใช้อัพแซมเพลอร์ L และแอนตีอิมเมจจิงฟิลเตอร์ g(n)

การอินเตอร์โพเลชัน ซึ่งประกอบด้วยการอัพแซมปลิง และ การใช้แอนตีอิมเมจจิงฟิลเตอร์ จะตรงข้ามกับการเดซิเมชัน ซึ่งประกอบด้วยการดาวน์แซมปลิง และ การใช้แอนตีเอเลียสซิงฟิลเตอร์นั่นเอง

2.6.3.3 เอกลักษณ์ (Identities)

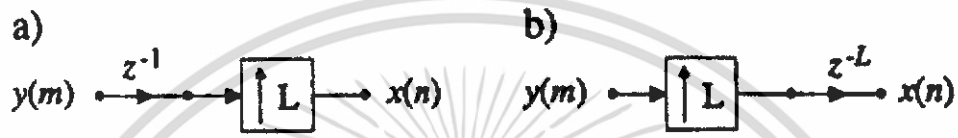
ในส่วนนี้แสดงเอกลักษณ์ที่สอดคล้องกันสำหรับการอัพแซมปลิง โครงสร้างการไหลของสัญญาณ (signal flow structure) และโครงสร้างสาขาของแต่ละโหนดนั้นไม่ขึ้นกับอัตราการสุ่มดังนั้น การไหลของสัญญาณ ทั้ง 2 ในรูปที่ 2.30 จึงเทียบเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



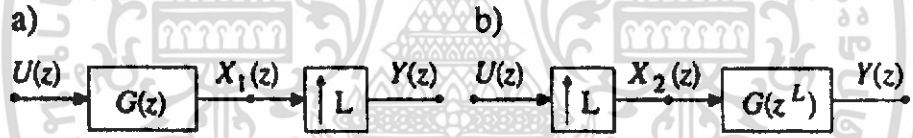
รูปที่ 2.30 เอกลักษณ์ที่ 4

เนื่องจากการหน่วงของ 1 คาบการสุ่มก่อนการอัพแซมปลิง นำไปสู่การหน่วงของ L คาบหลังการอัพแซมปลิง เราได้เอกลักษณ์ที่ 5 ในรูปที่ 2.31



รูปที่ 2.31 เอกลักษณ์ที่ 5

รูปทั่วไปของเอกลักษณ์ที่ 5 แสดงดังรูปที่ 2.32 ซึ่งพิสูจน์ได้โดยใช้สมการ (2.56)

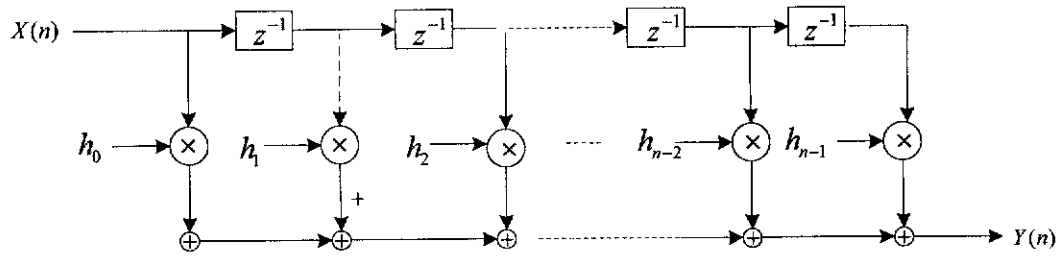


รูปที่ 2.32 เอกลักษณ์ที่ 6

2.7 อินเตอร์โพเลชัน (Interpolation)

สำหรับวงจรกรองสัญญาณดิจิตอลแบบเอฟไอาร์ (FIR : Finite impulse Response) จะใช้ผลตอบสนองอิมพัลส์ (Impulse Response) หรือ h(n) เพื่อสร้างวงจร ซึ่งกระบวนการทำงานของวงจรในที่นี้ คือ การทำคอนโวลูชั่น (Convolution) ระหว่าง h(n) กับสัญญาณอินพุท x(n) โดยจัดเป็นโครงสร้างแบบโดยตรง (Direct form)หรือ ทรานส์เวอร์ซอลเตอร์ฟิลเตอร์ (Transversal Filter) ได้ดังรูปที่ 2.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.33 แสดงโครงสร้างโดยตรงของวงจรกรองความถี่แบบเอฟไออาร์

การทำคอนโวลูชั่นระหว่างผลตอบสนองอิมพัลส์หรือ h(n) กับสัญญาณอินพุท x(n) จะแสดงได้ดังสมการ

$$y(n) = h(n) * u(n) = \sum_{k=0}^{N-1} h(n-k)u(k) \qquad (2.60)$$

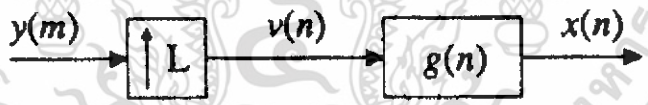
และสามารถแสดงสมการผลต่างสืบเนื่อง (Difference Equation) ได้ดังนี้

$$y(n) = h(0)u(n)+h(1)u(n-1)+h(2)u(n-2)+\ldots+h(N)u(n-N) \qquad (2.61)$$

2.7.1 อินเตอร์โพเลชันด้วยทรานส์เวอร์ซอลฟิลเตอร์ (Interpolation with Transversal Filter)

2.7.1.1 คอนโวลูชันด้วยการเพิ่มอัตราการสุ่มแบบลำดับ (Convolution with Subsequent upsampling)

ในข้างต้นได้แสดงว่าโดยทั่วไปนั้นอินเตอร์โพเลชัน จะประกอบด้วยอัพแซมเพลอร์ (Upsampler :L) 1 ตัว ตามด้วยแอนตีอิมเมจจิง (anti-imaging filter : g(n)) 1ตัว ดังแสดงในรูปที่ 2.34



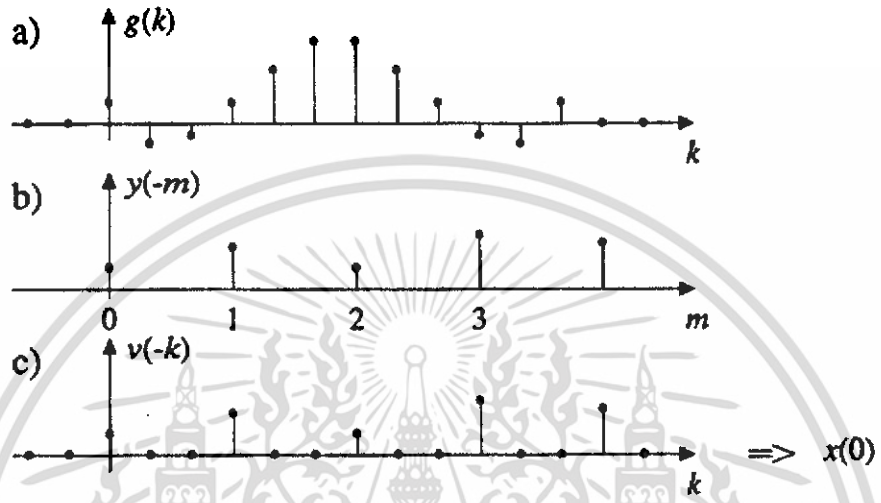
รูปที่ 2.34 อินเตอร์โพเลชัน ประกอบด้วยอัพแซมเพลอร์ : L 1ตัว
และแอนตีอิมเมจจิงฟิลเตอร์: g(n) 1ตัว

การเพิ่มอัตราการสุ่มที่ถูกอธิบายในสมการ (2.53) จะทำให้เกิดสัญญาณชั่วคราว $v(n)$ สัญญาณอินพุท $y(m)$ สัญญาณเอาท์พุท $x(n)$ จะได้มาด้วยการ คอนโวลูชันค่านี้กับค่าผลตอบสนองอิมพัลส์ (Impulse response: $g(n)$) สมมติให้เป็นค่าจำกัดซึ่งมีค่าสัมประสิทธิจำนวน N ค่า

$$x(n) = \sum_{k=0}^{N-1} v(n-k)\cdot g(k) \qquad (2.62)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.35(a) แสดงตัวอย่างผลตอบสนองอิมพัลส์ $g(k)$ รูปที่ 2.35(b) สัญญาณอินพุทไทม์รีเวิร์ส (time reversed: $y(-m)$) และรูปที่ 2.35(c) สัญญาณที่ถูกเพิ่มอัตราการสุ่ม $v(-k)$ เมื่อคูณ $g(k)$ กับ $v(-k)$ และบวกกันอย่างเช่นสมการ(2.62) ก็จะได้ค่า $x(0)$ ในขั้นตอนนี้ การคูณที่ไม่จำเป็นจะเกิดขึ้นเนื่องจากค่าศูนย์ที่แทรกเข้ามาระหว่างการเพิ่มอัตราการสุ่ม ซึ่งในการคำนวณค่าอินพุทที่ยังเหลือ $x(n)$ ก็เช่นกัน

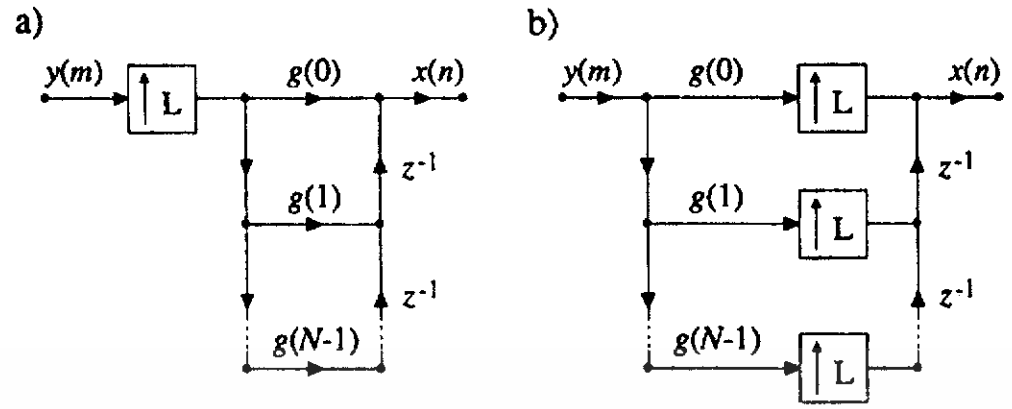


รูปที่ 2.35 สัญญาณที่เกี่ยวข้องกับการคอนโวลูชันในรูปที่ 2.34 :
(a) ค่าผลตอบสนองอิมพัลส์ $g(k)$
(b) สัญญาณอินพุทที่ถูกคอนโวลูชัน $y(-m)$
(c) สัญญาณที่ถูกเพิ่มอัตราการสุ่ม

2.7.1.2 ประสิทธิภาพของโครงสร้างทรานส์เวอร์ซอล (Efficient Transversal Structure)

การคูณที่ไม่ว่าเป็นด้วยค่าศูนย์ระหว่างการคอนโวลูชันในสมการ(2.62) สามารถหลีกเลี่ยงได้โดยการใช้เอกลักษณ์ที่ 4 ในรูปที่ 2.30 เปรียบเทียบโครงสร้างอินเตอร์โพเลชันแบบเดิมในรูปที่ 2.36(a) กับรูปที่ 2.30(a) กราฟที่เปลี่ยนรูปแบบ ในรูปที่ 2.30(b) นำไปสู่โครงสร้างที่มีประสิทธิภาพมากขึ้นในรูปที่ 2.36(b) ในที่นี้ประสิทธิภาพที่คำนวณสูงกว่าโครงสร้างที่เทียบเท่ากัน ในรูปที่ 2.36(a) L เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 อินเตอร์โพเลชัน : (a) โครงสร้างต้นแบบที่มีอัพแซมเพลอร์และทรานส์เวอร์ซอลฟิลเตอร์ (b) โครงสร้างที่มีประสิทธิภาพ

2.7.2 อินเตอร์โพเลชันด้วยโพลีเฟสฟิลเตอร์ (Interpolation with Polyphase Filter)

2.7.2.1 โพลีเฟสอินเตอร์โพเลชันในไทม์โดเมน (Representation of Polyphase Interpolation in Time Domain)

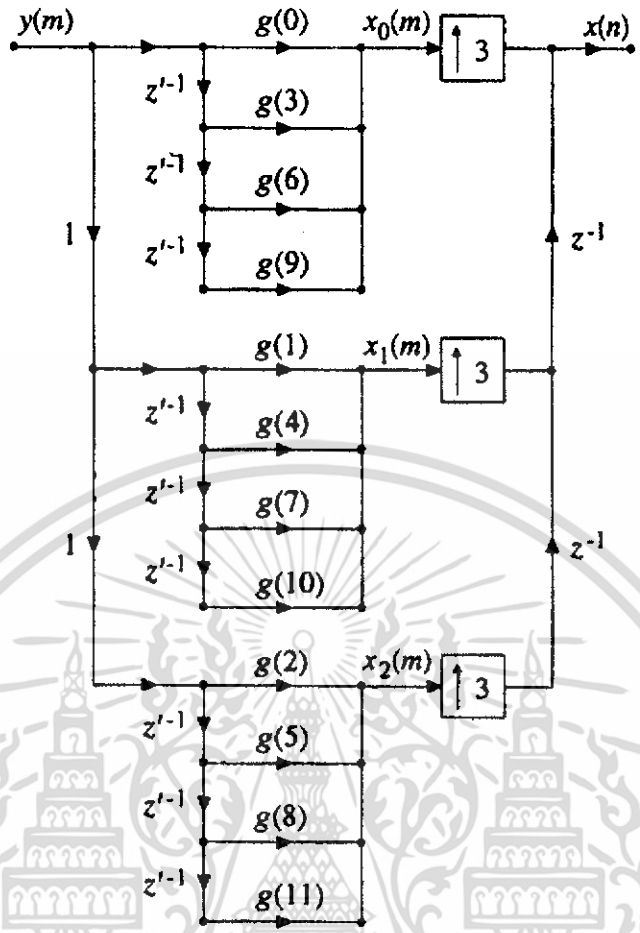
พิจารณาความสัมพันธ์ของการคอนโวลูชันในสมการ(2.62)

$$k = r \cdot L + \lambda \tag{2.63}$$

จะได้สมการสำหรับสัญญาณที่ได้จากการอินเตอร์โพเลชันดังนี้

$$x(n) = \sum_{k=0}^{N-1} v(n-k) \cdot g(k) = \sum_{\lambda=0}^{L-1} \sum_{r=0}^{R-1} v(n-rL-\lambda) \cdot g(rL+\lambda) \tag{2.64}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.37 การสร้างโพลีเฟสฟิลเตอร์ด้วย $L=3$

ดังนั้นจะถือได้ว่าจำนวน N ของค่าสัมประสิทธิของเอฟไอาร์ฟิลเตอร์ $g(n)$ คือผลคูณของอินเตอร์โพเลชันแฟกเตอร์ L (Interpolation factor : L) ซึ่ง $N=R\cdot L$ จากสมการ (2.53) จะได้สมการ (2.65)

$$v(n-rL-\lambda) = \begin{cases} y(m-r) & for\ \ n = mL+\lambda \\ 0 & otherwise \end{cases} \tag{2.65}$$

สำหรับแต่ละดัชนี (index) n ของเอาท์พุทจะมีดัชนีคือ $\lambda = \text{n mod } L$ เมื่อใช้สมการ(2.65) ในเงื่อนไขนี้จะทำให้ได้สมการ(2.66)

$$\begin{aligned} x_\lambda(m) &= \sum_{r=0}^{R-1} v(n-rL-\lambda)\cdot g(rL+\lambda)\Big|_{\lambda=n \bmod L} \\ &= \sum_{r=0}^{R-1} y(m-r)\cdot g_\lambda(r) \\ &= y(m)\cdot g_\lambda(m) \end{aligned} \tag{2.66}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้องค์ประกอบของผลตอบสนองอิมพัลส์จากก่อนหน้านี้จะได้มาจากสมการ(2.67)

$$g_\lambda(m) = g(mL + \lambda) \tag{2.67}$$

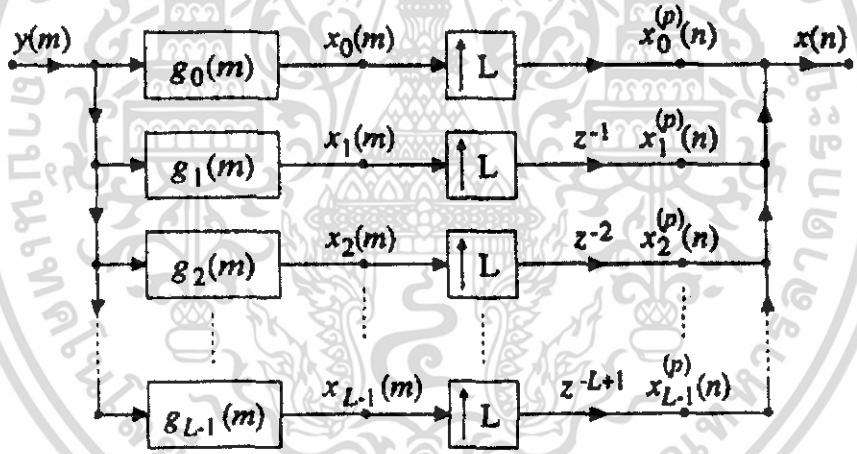
โดยการเพิ่มอัตราการสุ่มด้วยแฟกเตอร์ L และใช้การหน่วงของ λ ของคาบนาฬิกา (ที่อัตราการสุ่มของเอาท์พุทสูงขึ้น) จะได้องค์ประกอบของโพลีเฟส

$$x_\lambda^{(p)}(n) = \begin{cases} x_\lambda(m) & for\ n = mL + \lambda \\ 0 & otherwise \end{cases} \tag{2.68}$$

ซึ่งได้มาจากองค์ประกอบ $x_\lambda(m)$ จากนั้นพวกมันจะถูกรวมเพื่อให้ได้สัญญาณกระบวนการนี้แสดงในรูปที่ 2.37

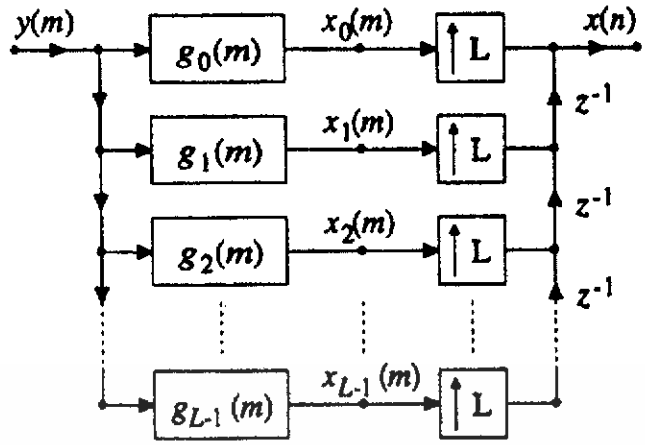
2.7.2.2 เอาท์พุทคอมมิวเทชัน (Representation with out put Commutator)

รูปที่ 2.38 แสดงโพลีเฟสอินเตอร์โพเลชันทั่วไปซึ่งประกอบด้วย L สาขา โดยที่ L คืออินเตอร์โพเลชันแฟกเตอร์ องค์ประกอบของอินเตอร์โพเลชันฟิลเตอร์ $g_0(m), g_1(m) \ldots g_{L-1}(m)$ จะสร้างแอนตีอิมเมจจิงฟิลเตอร์แบบเดิม $g(n)$

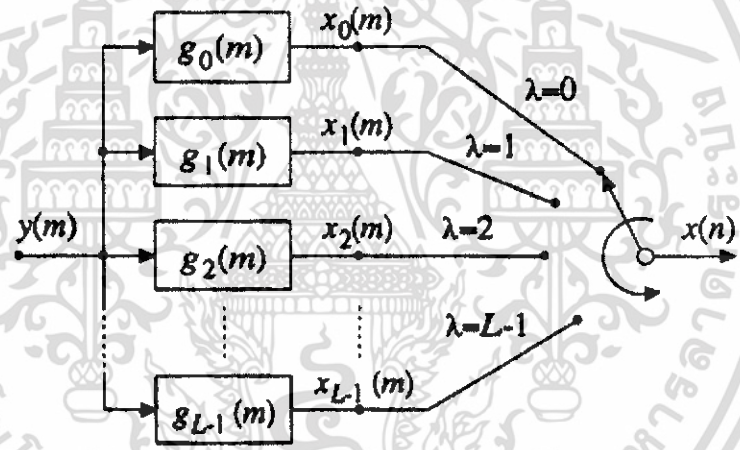


รูปที่ 2.38 โครงสร้างทั่วไปของโพลีเฟสอินเตอร์โพเลชันที่มี L สาขา
และอัตราการสุ่มที่เพิ่มขึ้นด้วยแฟกเตอร์ L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 โพลีเฟสอินเตอร์โพเลชันที่มีการหน่วงที่เอาท์พุท



รูปที่ 2.40 โพลีเฟสอินเตอร์โพเลชันที่มีคอมมิวเตเตอร์ (Commutator)

เอาท์พุทแบบทวนเข็มนาฬิกา

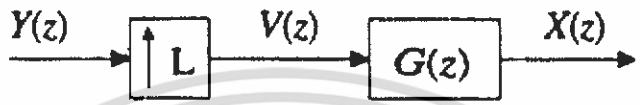
กระบวนการรวม $x_0(m), x_1(m) \ldots x_{L-1}(m)$ เพื่อสร้างสัญญาณเอาท์พุทสุดท้าย $x(n)$ ทำได้โดยเริ่มจากการสร้างองค์ประกอบโพลีเฟสของอินพุท องค์ประกอบโพลีเฟส $x_\lambda^{(p)}(n)$ เหล่านี้ ได้มาจากองค์ประกอบของสัญญาณ $x_\lambda(m)$ โดยการเพิ่มอัตราการสุ่มด้วยแฟกเตอร์ L และการรวมการหน่วง $z^{-\lambda}$ (ที่อัตราการสุ่มสูงขึ้น) โดยสอดคล้องกับสมการ (2.68) สำหรับ $\lambda = 0,1,2,\ldots,L-1$ ดังแสดงในรูปที่ 2.38 ดังนั้นองค์ประกอบของโพลีเฟสทั้งหมดก็จะรวมกันเพื่อสร้างสัญญาณเอาท์พุท $x(n)$ การหน่วง $z^{-\lambda}$ ในสาขาต่างๆของโพลีเฟส ถูกจัดให้อยู่ในรูปแบบที่ง่ายขึ้น เช่น ลำดับของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

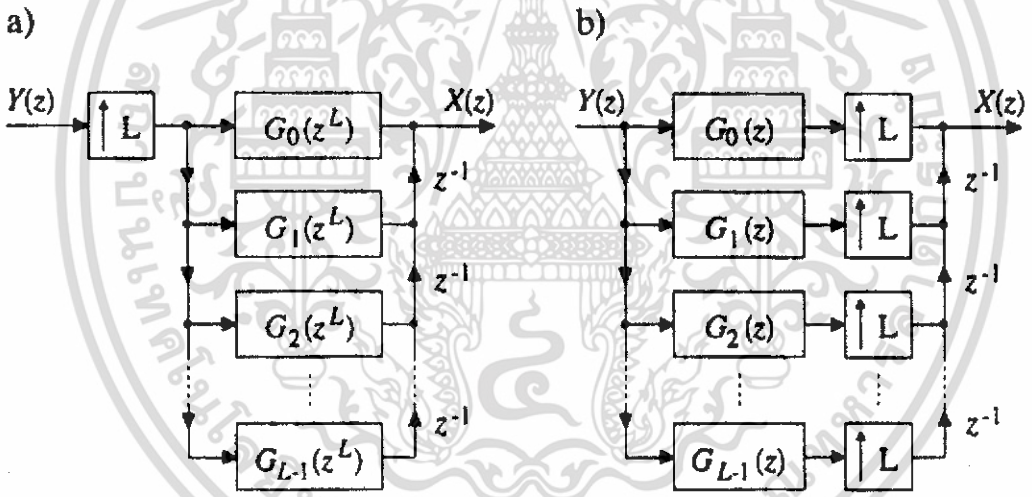
หน่วงในเอาท์พุท ดังแสดงในรูปที่ 2.39 ผู้เขียนบางท่านอาจใช้คอมมิวเตเตอร์ในเอาท์พุท เพื่อแสดงการรวมของสัญญาณ $x_\lambda(m)$ สู่สัญญาณเอาท์พุท $x(n)$ ดังแสดงในรูปที่ 2.40 ซึ่งทั้งสองแบบนั้นเท่ากัน

2.7.2.3 โพลีเฟสอินเตอร์โพเลชันโดยใช้การแปลง Z (Representation of Polyphase Interpolation Using the Z-Transform)

Z ทรานส์ฟอร์ม (Z-Transforms) จะถูกใช้เพื่อหาโพลีเฟส เริ่มจากโครงสร้างทั่วไปของโพลีเฟสอินเตอร์โพเลชันในรูปที่ 2.41



รูปที่ 2.41 อินเตอร์โพเลชันทั่วไปที่เป็นสัญญาณที่ได้จากการแปลง Z (Z-Transform)



รูปที่ 2.42 : (a) รูปแบบเดิมของโพลีเฟสอินเตอร์โพเลชัน
(b) รูปแบบที่มีประสิทธิภาพและประหยัดหน่วยความจำ

เราสามารถใช้สมการ (2.33) และ (2.34) เพื่อเขียนฟังก์ชันถ่ายโอน $G(z)$ ของอินเตอร์โพเลชันฟิลเตอร์ ซึ่งได้ว่า

$$G(z)=\sum_{\lambda=0}^{L-1} z^{-\lambda} G_\lambda\left(z^L\right) \tag{2.69}$$

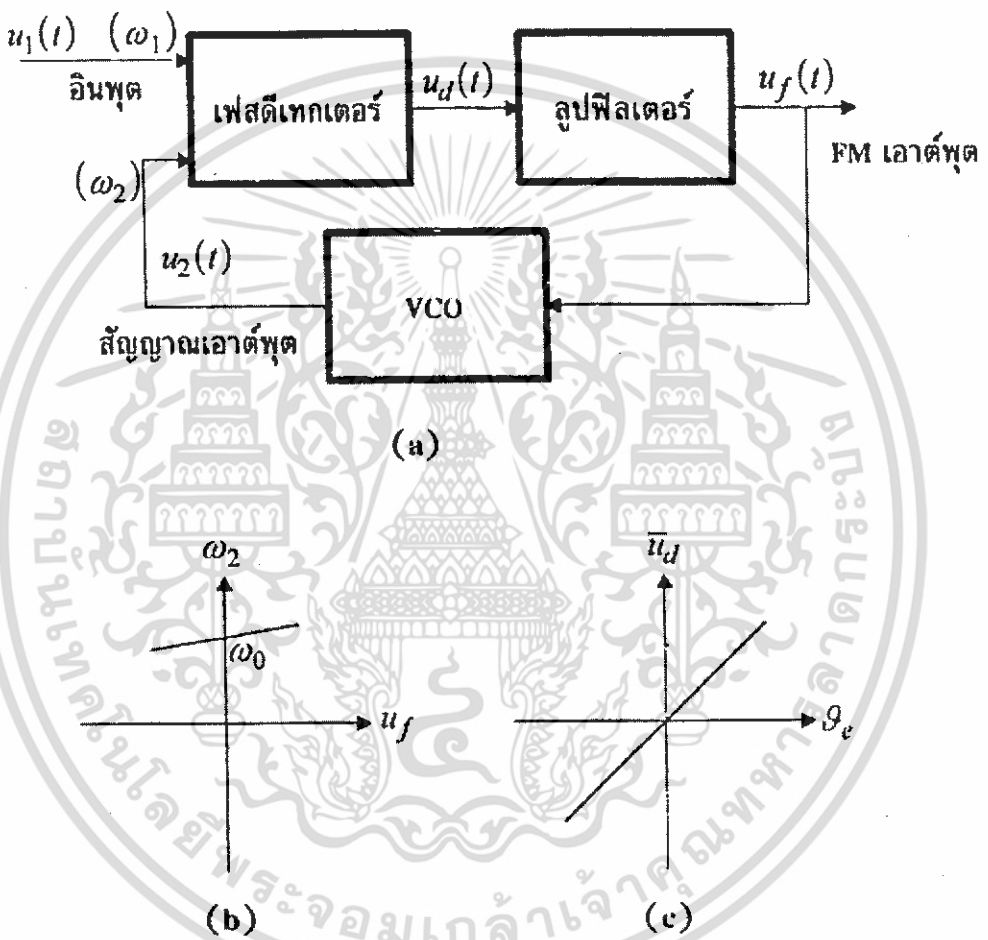
และจากสมการ(2.69) เราสามารถแทนที่โครงสร้างทั่วไปในรูปที่ 2.41 ด้วยโครงสร้างโพลีเฟสในรูปที่ 2.42(a) โดยการใช้เอกลักษณ์ที่ 4 และ 6 ดังแสดงในรูปที่ 2.42(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ทฤษฎีดิจิทัลเฟสล็อกลูป (Digital Phase-Locked Loop : DPLL)

2.8.1 นิยามของหลักการทำงานเฟสล็อกลูปตามแบบฉบับ

เฟสล็อกลูป คือ วงจรพิเศษที่ซึ่งทำให้ระบบหนึ่งปฏิบัติตามรอยอีกระบบหนึ่งได้ หรืออธิบายให้ชัดเจนคือ เฟสล็อกลูป (PLL) เป็นวงจรที่ทำซิงโครไนซ์สัญญาณเอาต์พุตกับสัญญาณอ้างอิงหรือสัญญาณอินพุตในความถี่และเฟสเดียวกัน ซึ่งการซิงโครไนซ์นี้ถูกเรียกบ่อย ๆ ว่า "สภาวะล็อก (locked state)" โดยที่เฟสเออเรอร์ระหว่างออสซิลเลเตอร์เอาต์พุตกับสัญญาณอ้างอิงนั้นเป็นศูนย์หรือน้อยมาก



รูปที่ 2. 43 (a) บล็อกไดอะแกรมของระบบเฟสล็อกลูป
(b) ฟังก์ชันถ่ายโอนของออสซิลเลเตอร์ควบคุมด้วยแรงดันไฟฟ้า (VCO)
(c) ฟังก์ชันถ่ายโอนของเฟสดีเทกเตอร์ (PD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นเป็นหลักการโดยย่อของเฟสล็อกลูป บล็อกไดอะแกรมแสดงตามรูปที่ 2.43 โดยวางเฟสล็อกลูป จะประกอบด้วยบล็อกล็อกฟังก์ชัน 3 ส่วนดังนี้

1. ออสซิลเลเตอร์ควบคุมด้วยแรงดันไฟฟ้า (VCO)
2. เฟสดีเทกเตอร์ (PD)
3. ลูปฟิลเตอร์ (LF)

สัญญาณภายในวงจรเฟสล็อกลูปที่น่าสนใจกำหนดดังนี้

- สัญญาณอ้างอิง หรือ อินพุต $u_1(t)$
- ความถี่เชิงมุม (angular frequency) ω_1 ของสัญญาณอ้างอิง
- สัญญาณเอาต์พุต $u_2(t)$ ของ วีซีโอ (VCO)
- ความถี่เชิงมุม ω_2 ของสัญญาณเอาต์พุต
- สัญญาณเอาต์พุต $u_d(t)$ ของเฟสดีเทกเตอร์
- สัญญาณเอาต์พุต $u_f(t)$ ของลูปฟิลเตอร์
- เฟสเออเรอร์ θ_e กำหนดเป็นความต่างเฟสของสัญญาณ $u_1(t)$ และ $u_2(t)$

เรามาสนใจกับการทำงานทั้งสามบล็อกตามรูป 2.43 ตัววีซีโอออสซิลเลตที่ความถี่เชิงมุม ω_2 ซึ่งถูกกำหนดจากสัญญาณเอาท์พุท u_f ของลูปฟิลเตอร์จะออสซิลเลตที่ความถี่เชิงมุม ω_2 กำหนดเป็นสมการดังนี้

$$\omega_2(t) = \omega_0 + K_0 u_f(t) \tag{2.70}$$

ที่ซึ่ง ω_0 คือความถี่กลางของวีซีโอและ K_0 คือเกนของวีซีโอในหน่วย $s^{-1}V^{-1}$

ตัวเฟสดีเทกเตอร์หรืออ้างเป็นเฟสคอมพาราเตอร์ได้ด้วย จะทำหน้าที่เป็นตัวเปรียบเทียบเฟสระหว่างสัญญาณเอาต์พุตกับเฟสของสัญญาณอ้างอิง ซึ่งจะถ่ายสัญญาณเอาต์พุต $u_d(t)$ โดยประมาณเป็นสัดส่วนโดยตรงของเฟสเออเรอร์ θ_e ภายในช่วงจำกัดช่วงหนึ่ง

$$u_d(t) = K_d \theta_e \tag{2.71}$$

ที่ซึ่ง K_d แทนด้วยเกนของเฟสดีเทกเตอร์มีหน่วยเป็นโวลต์ หรือ V / rad

ต่อไป มาดูการทำงานทั้งสามฟังก์ชันบล็อก โดยเริ่มแรกสมมติให้ความถี่เชิงมุมของสัญญาณอินพุต $u_1(t)$ มีค่าเท่ากับ ความถี่กลาง (center frequency) ซึ่งแทนด้วย ω_0 แล้วให้วีซีโอทำงานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่กลาง ดังนั้นจะเห็นว่าเฟสเออเรอร์ θ_e เป็นศูนย์ ซึ่งถ้า θ_e เป็นศูนย์นั้น u_d ของเฟสดีเทกเตอร์ต้องเป็นศูนย์ด้วย เพราะฉะนั้นสัญญาณเอาต์พุตของลูปฟิลเตอร์ u_f จะเป็นศูนย์ด้วย ซึ่งเงื่อนไขนี้เป็นการให้วีซีโอทำงานที่ความถี่กลาง

ถ้าเฟสเออเรอร์ θ_e เริ่มต้นไม่เป็นศูนย์แล้วเฟสดีเทกเตอร์จะสร้างสัญญาณเอาต์พุต u_d ไม่เป็นศูนย์หลังจากผ่านลูปฟิลเตอร์เป็นสัญญาณ u_f ด้วยเหตุนี้จะทำให้วีซีโอเปลี่ยนความถี่ทำงานในทิศทางซึ่งจะทำให้เฟสเออเรอร์หายไป

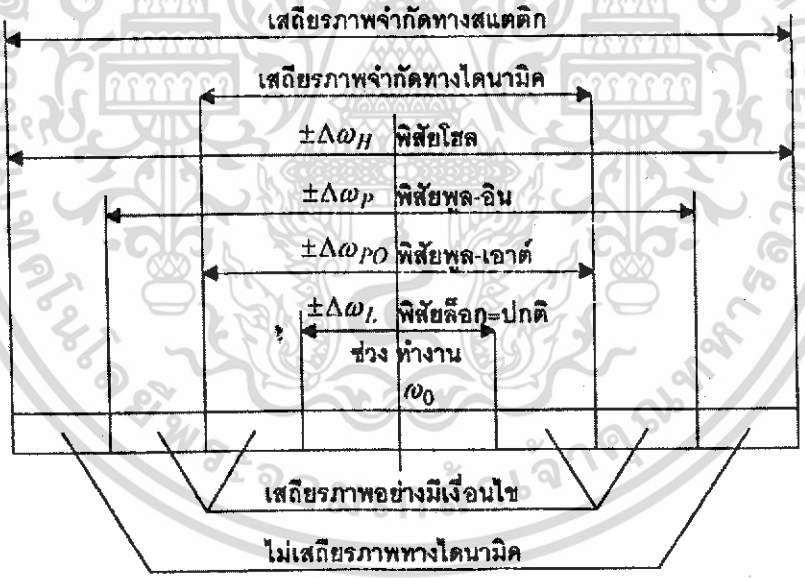
มีสามเงื่อนไขที่จำเป็นให้ระบบเฟสล็อกลูปแบบเชิงเส้นยังคงรักษาการแทรคกิ้ง (tracking)

1. ความถี่เชิงมุมของสัญญาณอ้างอิงต้องอยู่ภายในพิสัยโฮล (hold range)

2. สเต็ปความถี่สูงสุดจ่ายสู่อินพุตอ้างอิงของเฟสล็อกลูปต้องน้อยกว่าพิสัยพูลเอาต์ (pull-out range)

3. อัตราการเปลี่ยนของความถี่อ้างอิง $\Delta\omega$ ต้องน้อยกว่า ω_n^2

มีการจัดแบ่งพารามิเตอร์ระบุพิสัยความถี่ในการทำงานของระบบเฟสล็อกลูปซึ่งสามารถสรุปเนื้อหาที่สำคัญได้ดังนี้



รูปที่ 2.44 โครงสร้างเสถียรภาพทางไดนามิค(dynamic) และสแตติก(static) ของเฟสล็อกลูปแบบเชิงเส้น

1. พิสัยโฮล (hold range) $\Delta\omega_H$ ช่วงความถี่นี้เฟสล็อกลูปสามารถรักษาการติดตามอยู่กับเฟสได้ ซึ่งเฟสล็อกลูปจะมีเสถียรภาพอย่างมีเงื่อนไขในช่วงนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. พิสัยพูล-เอาต์ (pull-out range) $\Delta\omega_{PO}$ เป็นช่วงที่จำกัดไดนามิคสำหรับการทำงานเฟสล็อกลูปอย่างเสถียรภาพ ถ้าการแทรคกิ้งสูญเสียไปภายในช่วงนี้ ระบบเฟสล็อกลูปจะล็อกอีกครั้ง แต่กระบวนการจะช้าลง ซึ่งเป็นลักษณะของกระบวนการดึงเข้า (pull in process)

3. พิสัยพูล-อิน (pull-in range) $\Delta\omega_P$ คือช่วงที่ระบบเฟสล็อกลูปล็อกเสมอ แต่กระบวนการค่อนข้างช้า

4. พิสัยล็อก (lock range) $\Delta\omega_L$ คือช่วงความถี่ที่ระบบเฟสล็อกลูปจะทำการล็อกภายในหนึ่งบีต (beat) ระหว่างความถี่อ้างอิงกับความถี่เอาต์พุต ซึ่งโดยปกติเป็นช่วงความถี่การทำงานของระบบเฟสล็อกลูปถูกจำกัดภายในช่วงนี้

ดังนั้น ในการออกแบบโดยส่วนมาก เราต้องจัดขนาดพิสัยต่าง ๆ ไม่เท่ากันดังนี้

$$\Delta\omega_L < \Delta\omega_{PO} < \Delta\omega_P < \Delta\omega_H$$

<u>หมายเหตุ</u> เงื่อนไขประโยชน์ที่ได้รับจะเกี่ยวข้องเฉพาะระบบเฟสล็อกลูปแบบเชิงเส้นเท่านั้น กรณีที่เป็นระบบดิจิทัลเฟสล็อกลูปสภาวะเงื่อนไขนี้จะแตกต่างกันออกไป

ในหนังสืออ้างอิงหลายฉบับจะกล่าวถึง เทอมของ "พิสัยแคปเจอร์ (capture range)" ในบางครั้งสามารถใช้ในความหมายเป็น พิสัยพูล-อิน (pull-in range) ซึ่งไม่อาจแยกความแตกต่างกันได้ชัดเจน

2.8.2 นิยามดิจิทัลเฟสล็อกลูป (Digital Phase-Locked Loop Definition)

ดิจิทัลเฟสล็อกลูปนิยามความหมายได้ยาก เพราะว่าคำว่าดิจิทัลนั้นสามารถประยุกต์ใช้ได้หลายแบบอย่าง ในการกำหนดนิยามของดิจิทัลเฟสล็อกลูปตามแบบดั้งเดิม (classical) จะต้องมีเทอม ซึ่งมีความหมายของสัญญาณอ้างอิง u_1 และสัญญษณเอาต์พุต u_2 เป็นสัญญาณไบนารีโดยเป็นสัญญาณรูปคลื่นสี่เหลี่ยม อย่างไรก็ดี ภายในดิจิทัลเฟสล็อกลูปแบบดั้งเดิม บล็อกแอนะล็อกยังคงถูกใช้อยู่ ดังเช่น แอนะล็อกฟิลเตอร์ และ วีซีโอ ซึ่งเราสามารถกล่าวว่า ระบบดิจิทัลเฟสล็อกลูปแบบดั้งเดิมมีคุณลักษณะของสัญญาณแอนะล็อกเป็นตัวกลางอยู่

หลายปีมาแล้วระบบได้มีการพัฒนาสิ่งต่าง ๆ โดยสร้างจากอุปกรณ์ดิจิทัลอย่างแพร่หลาย ซึ่งจะไม่พบการใช้สัญญาณแอนะล็อก อย่างเช่น ลูปฟิลเตอร์ (loop filter) ซึ่งไม่ได้เป็นแอนะล็อกฟิลเตอร์อย่างที่เป็นมานาน แต่ใช้เป็นดิจิทัลฟิลเตอร์แบบออร์เดอร์หนึ่งหรือออร์เดอร์สองแทน ดังนั้น ดิจิทัลเฟสล็อกลูปแบบใหม่จึงต่างจากแบบดิจิทัลแบบดั้งเดิมโดยสิ้นเชิง โดยเรียกรวมทั้งหมดว่า "ระบบเฟสล็อกลูปแบบดิจิทัลล้วน (ADPLL)" ระบบดิจิทัลเฟสล็อกลูปยังไม่ได้เพิ่มเติมโดยใช้ซอฟต์แวรบนไมโครโปรเซสเซอร์ ในขั้นนี้จึงเป็นการง่ายที่จะแยกดิจิทัลเฟสล็อกลูปแบบดั้งเดิม และเฟสล็อกลูปแบบดิจิทัลล้วนออกจากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 คอร์ดิก อัลกอริธึม (CORDIC Algorithm)

การใช้ตัวคำนวณทางตรรกะที่สามารถทำการโปรแกรม ได้หลายครั้ง (Reconfigurable logic computers) ทำให้ความเร็วของฮาร์ดแวร์เพิ่มขึ้น ในขณะที่ราคาก็สามารถแข่งขันกับการใช้ซอฟท์แวร์แบบเดิมได้ แต่อย่างไรก็ตามอัลกอริธึมที่ได้ถูกใช้ให้เกิดประโยชน์สูงสุดสำหรับระบบที่ใช้ไมโครโพรเซสเซอร์นี้มักจะแมป (map) ลงฮาร์ดแวร์ได้ไม่ดี แม้จะมีทางแก้ที่มีประสิทธิภาพในทางฮาร์ดแวร์แต่ความเด่นของระบบซอฟท์แวร์ ก็ทำให้ทางแก้เหล่านั้นอยู่นอกสายตาไป

ในกลุ่มของอัลกอริธึมที่มีประสิทธิภาพในทางฮาร์ดแวร์เหล่านี้ ก็มีวิธีการแบบวนรอบการทำงาน (iterative solutions) สำหรับการแก้ฟังก์ชันตรีโกณมิติและฟังก์ชันที่ซับซ้อนอื่นๆ (transcendental function) ซึ่งใช้เพียงแค่การชิฟท์ (shift) และ แอด (add) ในการทำงาน ฟังก์ชันตรีโกณนั้นจะอ้างอิงกับการหมุนของเวคเตอร์ ในขณะที่ฟังก์ชันอื่นๆ เช่น สแควรูท(square root) นั้น จะสร้างจากการใช้สมการที่เพิ่มขึ้นมา (incremental expression)ของฟังก์ชันที่ต้องการ อัลกอริธึมแบบตรีโกณมิตินี้เรียกว่า CORDIC ซึ่งย่อมาจาก Coordinate Rotation Digital Computer ซึ่งฟังก์ชันที่เพิ่มขึ้นมานี้จะสร้างได้โดยโครงสร้างง่ายๆที่เพิ่มขึ้นมาจากโครงสร้างของฮาร์ดแวร์

อัลกอริธึมคอร์ดิกนั้น โดยทั่วไปจะสร้างบิทเพื่อความถูกต้องเพิ่มขึ้นมาอีกหนึ่งบิทสำหรับแต่ละรอบการทำงาน

2.9.1 ทฤษฎี CORDIC: อัลกอริธึมในการหมุนเวคเตอร์ (vector rotation)

ฟังก์ชันตรีโกณทั้งหมดสามารถคำนวณหรือได้มาจากฟังก์ชันที่ใช้การหมุนเวคเตอร์ซึ่งจะได้อธิบายต่อไปในภายหลัง การหมุนเวคเตอร์ยังสามารถใช้สำหรับการเปลี่ยนจากโพลาร์ (Polar) เป็น เรคแทงกูลาร์ (Rectangular) และ เรคแทงกูลาร์ เป็น โพลาร์ , การหาขนาดของเวคเตอร์ และการแปลง เช่น การแปลงฟูเรียร์สัญญาณไม่ต่อเนื่อง (DFT : Discrete Fourier Transform) และ การแปลงโคไซน์สัญญาณไม่ต่อเนื่อง (DCT : Discrete Cosine Transform)

คอร์ดิกอัลกอริธึมจะมีวิธีวนรอบการทำงานของการหมุนเวคเตอร์ด้วยมุมใดๆ โดยการใช้เพียงแค่การชิฟท์และแอด ซึ่งอัลกอริธึมนี้ได้มาจากการแปลงการหมุนทั่วไป (the general (givens) rotation transform)

$$\begin{aligned} x' &= x\cos\phi - y\sin\phi \\ y' &= y\cos\phi + x\sin\phi \end{aligned} \tag{2.72}$$

ซึ่งจะหมุนเวคเตอร์ในระนาบคาร์ทีเซียน ด้วยมุม ϕ และเขียนสมการใหม่นี้ได้เป็น

$$\begin{aligned} x' &= \cos\phi \cdot \left[x - y\tan\phi\right] \\ y' &= \cos\phi \cdot \left[y + x\tan\phi\right] \end{aligned} \tag{2.73}$$

จนถึงตอนนี้สมการยังไม่รับการทำให้อยู่ในรูปที่ง่าย อย่างไรก็ตาม ถ้ามุมที่หมุนถูกจำกัด ซึ่งทำให้ $\tan(\phi) = \pm 2$ การคูณด้วยเทอมแทนเจนท์ (tangent) จะถูกลดลงเหลือแค่การชิฟท์ธรรมดา มุมใดๆสามารถหาได้จากการหมุนด้วยมุมย่อยๆต่อเนื่องกัน ถ้าการตัดสินใจในแต่ละรอบการทำงานคือ “จะหมุนไปในทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใด" ไม่ใช่ "จะหมุนหรือไม่" เทอม $\cos(\delta_i)$ จะกลายเป็นค่าคงที่ (เนื่องจาก $\cos(\delta_i) = \cos(-\delta_i)$) การหมุนรอบการทำงานจะสามารถแสดงได้เป็น

$$\begin{aligned} x_{i+1} &= K_i \left[x_i - y_i \cdot d_i \cdot 2^{-i} \right] \\ y_{i+1} &= K_i \left[y_i + x_i \cdot d_i \cdot 2^{-i} \right] \end{aligned} \tag{2.74}$$

ซึ่ง

$$\begin{aligned} K_i &= \cos\left(\tan^{-1} 2^{-i} \right) = 1/\sqrt{1+2^{-2i}} \\ d_i &= \pm 1 \end{aligned} \tag{2.75}$$

การกำจัดค่าคงที่ที่มีผลต่อสเกล (scale constant) ออกจากสมการของการวนรอบการทำงานจะทำให้ได้ ชิฟท์-แอด อัลกอริธึม(shift-add algorithm) สำหรับการหมุนเวคเตอร์โปรดักต์ ของ K_i สามารถนำไปใช้ที่อื่นในระบบหรืออาจเป็นส่วนของเกน (gain) การทำงานของระบบ (system processing gain) โปรดักต์นี้จะเข้าใกล้ 0.6073 เมื่อรอบของการทำงานเข้าใกล้อินฟินิตี้ (infinity) ดังนั้นอัลกอริธึมการหมุนจะมีเกน, A_n, ประมาณ 1.647 ค่าเกน ที่แน่นอนขึ้นอยู่กับจำนวนรอบของการทำงานและจะเป็นไปตามความสัมพันธ์

$$A_n = \prod_n \sqrt{1+2^{-2i}} \tag{2.76}$$

มุมของการหมุนโดยรวมจะเป็นลำดับของทิศทางการหมุนย่อยๆ ซึ่งลำดับนี้จะแสดงเป็นเวคเตอร์ตัดสิน(Decision vector) เซตของเวคเตอร์ตัดสินที่จะเป็นไปได้ทั้งหมดคือ ระบบการวัดมุม ซึ่งอ้างอิงกับไบนารีอาร์คแทนเจนท์ (binary arctangent)

การแปลงระหว่างระบบมุมนี้สู่ระบบอื่นๆสามารถทำได้โดยใช้ตารางคู่มือ (A look-up) แต่วิธีการแปลงที่ดีกว่านั้นจะใช้ แอดเดอร์-ซับแทรคเตอร์ (adder-subtractor) เพิ่มขึ้นมาอีกหนึ่งตัว ซึ่งจะสะสมค่าของมุมการหมุนย่อยๆในแต่ละรอบการทำงาน มุมย่อยๆเหล่านี้อาจมาจากตารางคู่มือ (1 entry ต่อ 1 รอบการทำงาน) หรือจากการควบคุมของฮาร์ดแวร์ (hardwired) ขึ้นอยู่กับการสร้าง ตัวสะสมค่าของมุม (angle accumulator) จะเพิ่มสมการตัวที่ 3 สู่คอร์ดิกอัลกอริธึม

$$z_{i+1} = z_i - d_i \cdot \tan^{-1}\left(2^{-i}\right) \tag{2.77}$$

ซึ่งแน่นอนว่าในกรณีที่มุมสามารถใช้ได้ในอาร์คแทนเจนท์เบส (Arctangent base) สมการนี้ก็จะไม่จำเป็น

คอร์ดิก อัลกอริธึม นั้นโดยปกติจะทำงาน 1 หรือ 2 โหมด โหมดแรกเรียกการ "โรเทชัน (Rotation)" ซึ่งจะหมุนเวคเตอร์อินพุทด้วยมุมที่ระบุ (ซึ่งให้มาในรูปของอาร์กูเมนต์:argument) โหมดที่ 2 เรียก "เวคเตอร์ริง (Vectoring)" ซึ่งจะหมุนเวคเตอร์อินพุทไปที่แกน X และบันทึกมุมที่หมุนไปนั้นด้วย ในโหมดโรเทชัน ตัวสะสมค่ามุมจะถูกเริ่มที่มุมที่ต้องการให้หมุนไป การตัดสินใจในการหมุนของแต่ละรอบการทำงานมีขึ้นเพื่อลดขนาดมุมที่เหลืออยู่ในตัวสะสมค่ามุม ดังนั้นการตัดสินใจของแต่ละรอบการทำงานจะอ้างอิงกับเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของมุมที่เหลืออยู่หลังจากแต่ละขั้นตอน โดยธรรมชาตินั้น ถ้ามุมอินพุทนั้นถูกแสดงในรูปของไบนารีอาร์คแทนเจนท์เบส อยู่แล้ว ก็อาจจะไม่ต้องใช้ตัวสะสมค่ามุม

สำหรับ โรเทชันโหมด สมการคอร์ดิก จะเป็นดังนี้

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \tan^{-1}\left(2^{-i}\right) \end{aligned} \tag{2.78}$$

ซึ่ง

$$d_i = -1 \text{ if } z_i < 0,\ +1 \text{ otherwise} \tag{2.79}$$

และจะให้ผลลัพธ์ดังต่อไปนี้

$$\begin{aligned} x_n &= A_n\left[x_0 \cos z_0 - y_0 \sin z_0\right] \\ y_n &= A_n\left[y_0 \cos z_0 + x_0 \sin z_0\right] \\ z_n &= 0 \\ A_n &= \prod_n \sqrt{1+2^{-2i}} \end{aligned} \tag{2.80}$$

ในโหมด เวคเตอร์ริง นั้น ตัวหมุนของคอร์ดิก (CORDIC Rotator) จะหมุนเวคเตอร์อินพุทไปด้วยมุมเท่าใดก็ได้ที่จำเป็น เพื่อจะทำให้เวคเตอร์ลัพธ์นั้นขนานกับแกน X ผลลัพธ์ของกระบวนการเวคเตอร์ริง คือมุมที่ต้องหมุนไป (Rotation angle) และขนาดที่ได้รับการสเกล (the scaled magnitude) ของเวคเตอร์เดิม (x คอมโพเนนท์ของผลลัพธ์)

ฟังก์ชันของเวคเตอร์ริงทำงานโดยการค้นหา เพื่อลด y คอมโพเนนท์ (component) ของเวคเตอร์ที่เหลืออยู่ในแต่ละครั้งของการหมุน เครื่องหมายของ y คอมโพเนนท์ที่เหลือจะถูกใช้เพื่อตัดสินว่าครั้งต่อไปจะหมุนไปในทิศทางใด ถ้าตัวสะสมค่ามุมถูกเริ่มด้วยศูนย์ (0) มันก็จะเก็บมุมทรานวิร์สท์ (traversed) ไว้เมื่อแต่ละรอบการทำงานจบลง

เวคเตอร์ริงโหมด มีสมการคอร์ดิก ดังนี้

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \tan^{-1}\left(2^{-i}\right) \end{aligned} \tag{2.81}$$

ซึ่ง

$$d_i = +1 \text{ if } y_i < 0,\ -1 \text{ otherwise} \tag{2.82}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

$$\begin{aligned} x_n &= A_n\sqrt{x_0^2 + y_0^2} \\ y_n &= 0 \\ z_n &= z_0 + \tan^{-1}\left(y_0 / x_0\right) \\ A_n &= \prod_n \sqrt{1 + 2^{-2i}} \end{aligned} \tag{2.83}$$

อัลกอริธึมคอร์ดิกทั้งแบบโรเทชันและเวคเตอร์ริง ดังที่กล่าวมาจะถูกจำกัดมุมในการหมุนที่ระหว่าง $-\pi/2$ และ $\pi/2$ ซึ่งการจำกัดนี้มีเหตุมาจากการใช้ 2^0 สำหรับแทนเจนท์ในรอบการทำงานแรก สำหรับมุมรวมในการหมุน (composite rotation angle) ที่มากกว่า $\pi/2$ จะต้องมีการหมุนเพิ่มอีก 1 ครั้ง การหมุนเริ่มต้น $\pm\pi/2$ ให้รอบการทำงานดังนี้ (the correction iteration)

$$\begin{aligned} x' &= -d \cdot y \\ y' &= d \cdot x \\ z' &= z + d \cdot \pi/2 \end{aligned} \tag{2.84}$$

โดยที่

$$d = +1 \text{ if } y<0, -1 \text{ otherwise} \tag{2.85}$$

ซึ่งการหมุนเริ่มต้นนี้จะไม่มีการเติบโต แต่การหมุนเริ่มต้นของ π หรือ 0 (ศูนย์) ก็สามารถทำได้โดยการเลี่ยงการกำหนดค่าใหม่ของ x และ y คอมโพเนนท์สู่อุปกรณ์ของตัวหมุน (rotator elements) ย้ำว่าจะไม่มีการเติบโต เนื่องจากการหมุนเริ่มต้น

$$\begin{aligned} x' &= d \cdot x \\ y' &= d \cdot y \\ z' &= z \text{ if } d = 1, \text{ or } z - \pi \text{ if } d = -1 \\ d &= -1 \text{ if } x < 0, +1 \text{ otherwise} \end{aligned} \tag{2.86}$$

รูปแบบที่ลดลงทั้ง 2 (both reduction forms) จะใช้ มอดูโล(Modulo) 2π เพื่อแสดงมุมอินพุท วิธีของรูปแบบแรกจะสม่ำเสมอกับการหมุนที่ตามมามากกว่า ในขณะที่รูปแบบที่ 2 จะสะดวกกว่า

เมื่อการไวร์ริง (Wiring) นั้นถูกจำกัด ซึ่งมักจะเป็นในกรณีที่ใช้ เอฟพีจีเอ ตัวหมุนที่อธิบายไปนี้สามารถใช้เพื่อการคำนวณฟังก์ชันตรีโกณได้โดยตรงและสามารถคำนวณฟังก์ชันอื่นได้โดยอ้อม การเลือกค่าเริ่มต้นและโหมดอย่างระมัดระวังและมีเหตุผลจะทำให้สามารถคำนวณ ไซน์, โคไซน์, อาร์คแทนเจนท์, ขนาดของเวคเตอร์ และการแปลงระหว่าง โพลาร์ และ คาร์ทีเซียนโคออร์ดิเนต (Cartesian coordinate) ได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไซน์และโคไซน์

กระบวนการโรเทชันโหมดของคอร์ดิกสามารถคำนวณไซน์และโคไซน์ของมุมอินพุทได้พร้อมกัน การตั้งค่าของ y คอมโพเนนท์ของเวคเตอร์อินพุทจะลดผลลัพธ์ของ โรเทชันโหมดลงสู่

$$x_n = A_n \cdot x_0 \cos z_0$$
$$y_n = A_n \cdot x_0 \sin z_0 \qquad (2.87)$$

การตั้งค่า x_0 ให้เท่ากับ $1/A_n$ จะทำให้การหมุนนั้นสร้างไซน์และโคไซน์ที่ยังไม่ได้รับการสเกลของอาร์กูเมนต์มุม, z_0 บ่อยครั้งที่ค่าไซน์และโคไซน์จะเป็นตัวมอดูเลตค่าของขนาด การใช้เทคนิคอื่นๆ เช่น ตารางคู่มือ จะต้องใช้ตัวคูณ(multipliers) 1 คู่ เพื่อการมอดูเลชัน เทคนิคของคอร์ดิกนั้นจะใช้การคูณเป็นส่วนหนึ่งของกระบวนการการหมุน ดังนั้นจึงไม่ต้องใช้ตัวคูณคู่(a pair of explicit multipliers)ดังกล่าว เอาท์พุทของตัวหมุนคอร์ดิกจะได้รับการสเกลโดยเกนของตัวหมุน แต่ถ้าเกนนั้นไม่สามารถยอมรับได้ การคูณด้วยส่วนกลับของค่าคงที่ของเกนที่ข้างหน้าของตัวหมุนคอร์ดิกจะทำให้ได้ผลลัพธ์ที่ยังไม่ได้รับการสเกล และความซับซ้อนของฮาร์ดแวร์ของตัวหมุนคอร์ดิกนั้นก็เทียบเท่ากับฮาร์ดแวร์ของตัวคูณที่มีขนาดของเวิร์ด (word) เท่ากันเพียงตัวเดียวเท่านั้น

การแปลงโพลาร์สู่เรคแทงกูลาร์

ส่วนขยายในทางตรรกะของตัวคำนวณไซน์และโคไซน์ คือตัวแปลงคู่ลำดับโพลาร์สู่คาร์ทีเซียน การแปลงจากสเปซ (space) ของโพลาร์เป็นคาร์ทีเซียนนิยามโดย

$$x = r\cos\theta$$
$$y = r\sin\theta \qquad (2.88)$$

จะเห็นได้ว่า การคูณด้วยขนาดสามารถได้มาจากการใช้ตัวหมุนคอร์ดิกและการแปลงนั้นสามารถทำได้โดยการเลือกโหมดโรเทชัน โดยที่ x_0 = ขนาดของโพลาร์, z_0 = เฟสของโพลาร์, และ $y_0 = 0$ ผลลัพธ์ทางเวคเตอร์จะแสดงว่าอินพุทของโพลาร์ได้รับการแปลงสู่สเปซของคาร์ทีเซียน การแปลงนี้จะมีเกนเท่ากับเกนของตัวหมุน ถ้าเกนนั้นไม่สามารถยอมรับได้ อาจคูณขนาดของโพลาร์ด้วยส่วนกลับของเกนของตัวหมุนก่อนที่จะเข้าสู่ตัวหมุนคอร์ดิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนเวคเตอร์ทั่วไป (General vector rotation)

ตัวหมุนคอร์ดิกในโหมดโรเทชันยังมีประโยชน์สำหรับการทำการหมุนเวคเตอร์ทั่วๆไปอีกด้วย ดังที่มักจะพบเห็นในโมชันคอร์เรคชัน (motion correction) และระบบควบคุม สำหรับการหมุนทั่วๆไป เวคเตอร์อินพุท 2 มิติ จะถูกนำเข้าสู่อินพุทของตัวหมุน ตัวหมุนจะหมุนเวคเตอร์ด้วยมุมที่เราต้องการ เอาท์พุทจะได้รับการสเกลด้วยเกนของตัวหมุนคอร์ดิก ถ้าการสเกลไม่สามารถยอมรับได้ จะต้องใช้ตัวคูณค่าคงที่ 1 คู่ เพื่อชดเชยค่าเกน ตัวหมุนคอร์ดิกอาจต่อคาสเคด(cascade) ในรูปของโครงสร้างต้นไม้สำหรับการหมุนทั่วไปใน n มิติ เราสามารถเขียนโปรแกรมการหมุนในหลายมิติได้เพื่อลดการคำนวณในกรณีการหมุน n มิติทั่วไป

อาร์คแทนเจนท์

Arctangent, $\theta = A\tan(y/x)$ จะถูกคำนวณโดยตรงโดยใช้ตัวหมุนคอร์ดิกโหมดเวคเตอร์ริง ถ้าตัวสะสมค่ามุมถูกเริ่มด้วยศูนย์ (0) ตัวอาร์กูเมนต์จะต้องแสดงด้วยอัตราส่วนที่อยู่ในรูปของเวคเตอร์(x, y) การแสดงอาร์กูเมนต์ในรูปของอัตราส่วนนั้นมีข้อดีที่สามารถแสดงอินฟินิตี้ได้(โดยการให้ x = 0) และเนื่องจากผลลัพธ์ของอาร์คแทนเจนท์นั้นนำมาจากตัวสะสมค่ามุม การเติบโตของตัวหมุนคอร์ดิกจึงไม่มีผลต่อผลลัพธ์

$$z_n = z_0 + \tan^{-1}\left(y_0/x_0\right) \tag{2.89}$$

ขนาดของเวกเตอร์

ตัวหมุนคอร์ดิกโหมดเวคเตอร์ริงจะสร้างขนาดของเวคเตอร์อินพุทในรูปผลพลอยได้ของการคำนวณอาร์คแทนเจนท์หลังจากการหมุนในโหมดเวคเตอร์ริง เวคเตอร์จะขนานกับแกน X ขนาดของเวคเตอร์จึงยังเท่าเดิมเนื่องจาก x คอมโพเนนท์ของเวคเตอร์ที่ถูกหมุน ผลที่ได้นั้นชัดเจนในสมการผลลัพธ์สำหรับตัวหมุนในโหมดเวคเตอร์ริง

$$x_n = A_n\sqrt{x_0^2 + y_0^2} \tag{2.90}$$

ผลลัพธ์ของขนาดจะได้รับการสเกลโดยเกนของตัวประมวลผล การสร้างขนาดของเวคเตอร์จะมีความซับซ้อนของฮาร์ดแวร์ประมาณหนึ่งตัวคูณที่มีความกว้างเท่ากัน การสร้างของคอร์ดิกนั้นแสดงถึงการประหยัดฮาร์ดแวร์ได้มากกว่าตัวประมวลผลแบบพีธากอเรียนที่เทียบเท่ากัน ความถูกต้องของผลลัพธ์ของขนาดจะเพิ่มขึ้น 2 บิทสำหรับแต่ละรอบการทำงานที่ผ่านไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

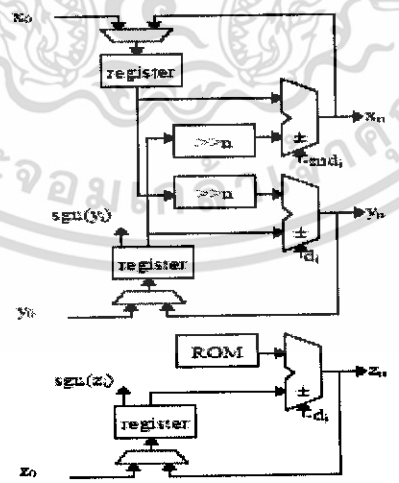
2.9.2 การสร้างลงใน เอฟพีจีเอ

ตัวประมวลผลแบบคอร์ดิกนั้นมีการสร้างหลายวิธี ซึ่งโครงสร้างในอุดมคติจะขึ้นอยู่กับความเร็วต่อพื้นที่ เราจะมาดูโครงสร้างของการวนรอบการทำงานซึ่งแปลมาโดยตรงจากสมการคอร์ดิก จากนั้นจะเป็นวิธีเพื่อให้ได้มาซึ่งการใช้ฮาร์ดแวร์ที่น้อยที่สุด และการทำงานที่มีประสิทธิภาพสูงที่สุด

2.9.2.1 ตัวประมวลผลแบบวนรอบการทำงานคอร์ดิก

โครงสร้างแบบวนรอบการทำงานคอร์ดิกนั้นสามารถได้มาโดยง่ายจากการทำซ้ำแต่ละสมการที่แตกต่างกัน 3 สมการในฮาร์ดแวร์ ดังแสดงในรูปที่ 2.45 ฟังก์ชันการตัดสินใจ d_i จะแปรตามเครื่องหมายของ y หรือ z รีจิสเตอร์โดยขึ้นอยู่กับว่ากำลังทำงานในโหมดโรเทชันหรือเวคเตอร์ริง ในการทำงาน ค่าเริ่มต้นจะถูกโหลดผ่านมัลติเพล็กเซอร์เข้าสู่ x, y, และ z รีจิสเตอร์ จากนั้นแต่ละรอบต่อไปของคล็อกไซเคิล (clock cycle) ค่าจากรีจิสเตอร์จะถูกส่งผ่านชิฟท์เตอร์(shifter)และแอดเดอร์-ซับแทรคเตอร์ และผลลัพธ์จะถูกส่งกลับไปที่รีจิสเตอร์ ชิฟท์เตอร์จะถูกเปลี่ยนแปลงในแต่ละรอบของการทำงานเพื่อให้ได้มาซึ่งการชิฟท์ที่ต้องการสำหรับแต่ละรอบการทำงาน เช่นเดียวกับแอดเดรสของรอม(ROM) ที่จะเพิ่มขึ้นในแต่ละรอบการทำงานเพื่อที่ z แอดเดอร์-ซับแทรคเตอร์ จะได้รับค่ามุมย่อยๆที่เหมาะสม ในรอบสุดท้ายของการทำงานนั้นผลลัพธ์จะถูกอ่านโดยตรงจากแอดเดอร์-ซับแทรคเตอร์ และแน่นอนว่าสเตทแมชีน(state machine) อย่างง่ายจะต้องถูกนำมาใช้เพื่อการติดตามบันทึกรอบการทำงานปัจจุบันและเพื่อเลือกมุม(degree) ของการชิฟท์และแอดเดรสของรอมสำหรับแต่ละรอบการทำงาน

การออกแบบที่แสดงในรูปที่ 2.45 นั้นใช้เส้นทางข้อมูลแบบเวิร์ดไวด์ (word-wide) (เรียกการออกแบบแบบ bit-parallel) ตัวชิฟท์เตอร์แบบบิทพาราลเลลแวริเอเบิลชิฟท์ (bit-parallel variable shift) จะแมปลงโครงสร้างของเอฟพีจีเอได้ไม่ดีนักเนื่องจากจะต้องใช้ไฮแฟนอิน(high fan-in) ถ้าทำการสร้างลงไป ชิฟท์เตอร์เหล่านั้นจะต้องใช้โลจิกหลายชั้น (เช่นสัญญาณจะต้องผ่านเอฟพีจีเอหลายเซลล์) ผลลัพธ์คืองานออกแบบที่ช้าและใช้โลจิกเซลล์จำนวนมาก

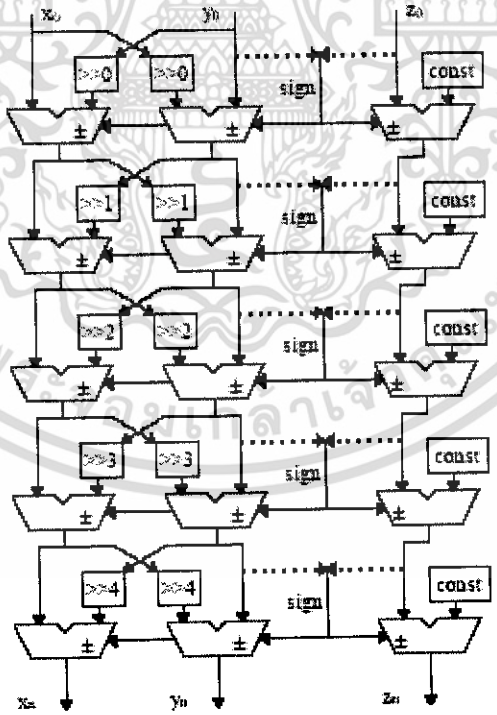


รูปที่ 2.45 โครงสร้างแบบวนรอบการทำงานคอร์ดิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2.2 ตัวประมวลผลคอร์ดิกแบบออน-ไลน์ (On-line CORDIC Processors)

ตัวประมวลผลคอร์ดิกที่ได้รับการกล่าวจนถึงตอนนี้เป็นแบบวนรอบการทำงานทั้งหมด ซึ่งหมายความว่าตัวประมวลผลจะต้องทำการวนรอบการทำงานด้วยความเร็ว n เท่าของความเร็วข้อมูลการวนรอบการทำงานจะสามารถอันโรล(unroll)ได้ เพื่อที่ว่าแต่ละส่วน(element)(ซึ่งมีอยู่ n ตัว) ของการประมวลผล จะทำการ วนรอบการทำงานเดียวกันเสมอ ตัวประมวลผลคอร์ดิกแบบอันโรลแสดงดังรูปที่ 2.46 การอันโรลตัวประมวลผลนี้จะให้ผลที่สำคัญ 2 อย่าง อย่างแรก คือ ชิฟท์เตอร์แต่ละตัวจะเป็นแบบฟิกซ์ท์ชิฟท์ (fixed shift) ซึ่งหมายความว่าสามารถสร้างพวกมันได้โดยการไวร์ริง(การเชื่อมสาย) อย่างที่ 2 คือ ค่าที่ได้จากการเปิดตารางสำหรับตัวสะสมค่ามุมจะถูกแจกจ่ายสู่แอดเดอร์แต่ละตัวในลูกโซ่ของตัวสะสมค่ามุมในฐานะค่าคงที่ ค่าคงที่เหล่านี้สามารถควบคุมได้ด้วยฮาร์ดแวร์(hardwired) แทนที่จะต้องใช้พื้นที่ในการเก็บทั้งหมดของตัวประมวลผลคอร์ดิกจะลดลงเหลือเพียงอาร์เรย์ของแอดเดอร์-ซับแทรคเตอร์ที่เชื่อมต่อกัน และก็ไม่มีความจำเป็นที่จะต้อง ใช้ตัวรีจิสเตอร์อีกต่อไป การดีเลย์ผ่านวงจรที่ได้นั้นจะมาก แต่ก็จะทำให้เวลาในการประมวลผลลดลงเนื่องจากวงจรการวนรอบการทำงาน(Iterative circuit)(ถ้าไม่ใช่เพราะเหตุผลอื่นนอกเหนือไปจากเวลาในการเซตอัพแอนด์โฮลด์ของตัวรีจิสเตอร์) โดยส่วนใหญ่แล้วโดยเฉพาะอย่างยิ่งในเอฟพีจีเอ ไม่มีเหตุผลที่จะต้องใช้วงจรคอมบิเนทอเรียล(Combinatorial)ที่ใหญ่ขนาดนั้น การไปป์ไลน์(pipeline)ตัวประมวลผลแบบอันโรลสามารถทำได้โดยง่ายโดยการแทรกตัวรีจิสเตอร์ไปที่ระหว่างแอดเดอร์-ซับแทรคเตอร์ ในกรณีของโครงสร้างเอฟพีจีเอส่วนใหญ่ รีจิสเตอร์จะอยู่ในแต่ละโลจิกเซลล์อยู่แล้ว ดังนั้นจึงไม่มีค่าใช้จ่ายของฮาร์ดแวร์รีจิสเตอร์แบบไปป์ไลน์ที่เพิ่มขึ้นมา

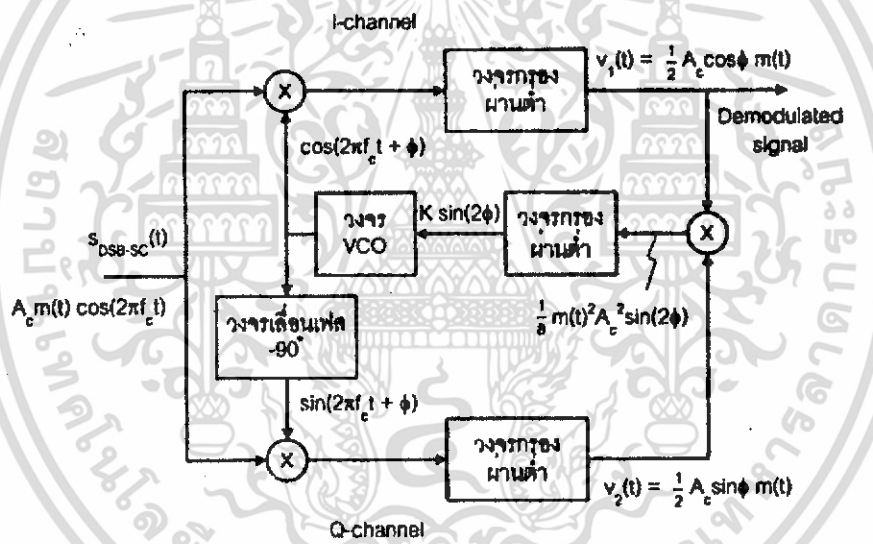


รูปที่ 2.46 ตัวประมวลผลคอร์ดิกแบบอันโรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 คอสแทส ลูป (Costas Loop)

ส่วนต่อไปนี้กล่าวถึงวงจรภาครับแบบซิงโครนัสชนิดหนึ่งที่สามารถนำมาใช้ได้งานดีในทางปฏิบัติ วงจรนี้มีชื่อเรียกว่า วงจรภาครับแบบคอสแทส ลูป ภายในวงจรประกอบด้วยวงจรโคฮีเรนต์ดีเทกเตอร์ 2 ชุด ดังที่แสดงในรูปที่ 2.47 สัญญาณที่ป้อนเข้าที่วงจรทั้งสองเหมือนกันคือสัญญาณ ดีเอสบี-เอสซี แต่ส่วนที่ต่างกันวงจรคือสัญญาณไซนูซอยด์ที่กำเนิดขึ้นที่วงจรภาครับมีเฟสต่างกัน 90 องศา หรือจะกล่าวว่าสัญญาณไซนูซอยด์ทั้งสองมีเฟสที่ควอเดรเจอร์ซึ่งกันและกัน (phase quadrature) วงจรโคฮีเรนต์ดีเทกเตอร์ส่วนบนเรียกว่า อินเฟสโคฮีเรนต์ดีเทกเตอร์(in-phase coherent detector) หรือ ไอแชนแนล(I-channel) ส่วนวงจรโคฮีเรนต์ดีเทกเตอร์ส่วนล่างเรียกว่า ควอเดรเจอร์เฟสโคฮีเรนต์ดีเทกเตอร์ (quadrature-phase coherent detector) หรือ คิวแชนแนล(Q-channel) วงจรโคฮีเรนต์ดีเทกเตอร์ถูกนำมาต่อรวมกันเป็นระบบป้อนกลับแบบลบ (negative feedback system) เพื่อให้สัญญาณไซนูซอยด์ที่สร้างขึ้นที่วงจรภาครับซิงโครไนซ์ กับคลื่นพาห์ของสัญญาณ ตลอดเวลา



รูปที่ 2.47 โครงสร้างวงจรดีมอดูเลตสัญญาณดีเอสบี-เอสซีแบบคอสแทส ลูป

พิจารณาของวงจรรับแบบคอสแทส ลูป จะเห็นว่าสัญญาณเบสแบนด์ที่ต้องการจะได้จากส่วนของวงจรโคฮีเรนต์ดีทกเตอร์ด้านบนไอแชนแนลในสภาพที่ไม่มีการคลาดเคลื่อนของเฟสเลยก็คือ $\phi = 0$ สัญญาณที่ออกจากวงจรโคฮีเรนต์ดีเทกเตอร์ด้านล่างคิวแชนแนลจะมีค่าเป็นศูนย์ แต่หากมีการคลาดเคลื่อนของเฟส ϕ เกิดขึ้นเล็กน้อย สัญญาณที่ออกจากวงจรไอแชนแนลก็คือสัญญาณเบสแบนด์ที่คูณอยู่กับ $\cos(\phi)$ ส่วนสัญญาณที่ออกจากวงจรคิวแชนแนลเป็นสัญญาณที่คูณอยู่กับ $\sin(\phi)$ โดยสัญญาณที่ออกจากคิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แชนแนลจะมีขั้วเดียวกับสัญญาณที่ออกจากไอแชนแนลเมื่อเกิดการคลาดเคลื่อนของเฟสในทิศทางหนึ่ง และจะมีขั้วสำหรับการคลาดเคลื่อนในอีกทิศทางตรงกันข้าม ดังนั้นเมื่อเรานำสัญญาณทั้งสองมาเข้าวงจรเฟสดิสคริมิเนเตอร์ (phase discriminator) ซึ่งประกอบด้วยวงจรคูณ และตามด้วยวงจรฟิลเตอร์ผ่านความถี่ต่ำที่ดึงเฉพาะองค์ประกอบความถี่ต่ำมากๆ จะได้สัญญาณกระแสตรงออกมาเพื่อป้อนเข้าที่วงจรวีซีโอ(VCO : voltage-controlled oscillator) สัญญาณนี้จะช่วยปรับแก้ไขสัญญาณไซนูซอยด์ที่กำเนิดขึ้นภายในวงจรภาครับซิงโครไนซ์ กับคลื่นพาห์ของสัญญาณดีเอสบี-เอสซีอยู่ตลอดเวลา

ข้อจำกัดสำคัญอย่างหนึ่งของวงจรภาครับแบบคอสแทส ลูป คือไม่สามารถแยกแยะว่าสัญญาณที่ถูกดีมอดูเลตมีเฟสตรงกับสัญญาณเบสแบนด์ $m(t)$ หรือมีเฟสต่างไป 180 องศา นั่นคือไม่ว่าจะป้อนสัญญาณดีเอสบี-เอสซีในรูปของ $m(t)A_c\cos(2\pi f_c t)$ หรือ $-m(t)A_c\cos(2\pi f_c t)$ เข้าไปในวงจร วงจรทั้งสองรูปแบบก็จะสามารถให้สัญญาณ $-m(t)$ และ $m(t)$ ออกมาได้ทั้งคู่ ข้อจำกัดนี้อาจจะไม่ส่งผลกระทบ หรือเป็นปัญหาต่อระบบ ที่ใช้รับส่งสัญญาณเสียง เพราะหูของมนุษย์จะรับรู้ตอบสนองต่อสัญญาณเสียง $m(t)$ และ $-m(t)$ เหมือนกัน แต่ถ้านำระบบนี้ไปส่งสัญญาณไบนารีเช่น 0 และ 1 จำเป็นจะต้องทราบค่าเฟสที่แน่นอนของสัญญาณ แนวทางหรือเทคนิคที่สามารถใช้ในการบอกวงจรภาครับถึงค่าเฟสมีอยู่ 2 วิธีการ คือ ส่งสัญญาณที่ทราบค่าแน่นอนออกไปให้ภาครับช่วงหนึ่งก่อนการส่งข้อมูลจริงเพื่อให้ภาครับสามารถระบุขั้วของสัญญาณให้ถูกต้องหรือใช้วิธีการเข้ารหัสหาค่าความแตกต่าง (differential coding)

เฟสล็อคลูป (Phase-Locked Loop: PLL) ได้ถูกนำมาใช้เป็นโครงสร้างมาตรฐานสำหรับการกู้สัญญาณพาหะมาได้ระยะหนึ่งแล้ว แต่ก่อนนั้น นักออกแบบจะใช้การมอดูเลตคลื่นพาห์ที่เหลือ (residual carrier modulation) ในการทำให้เฟสล็อคลูปทำการตาม (track) องค์ประกอบของคลื่นพาห์ แต่การส่งองค์ประกอบของคลื่นพาห์ที่เหลือไปกับสัญญาณที่ถูกมอดูเลตนั้นเป็นการสูญเสียพลังงาน เนื่องจาก องค์ประกอบนี้จะไม่บรรจุข้อมูลใดๆ ดังนั้นนักออกแบบทั้งหลายจึงเริ่มแนะนำวิธีในการกู้สัญญาณพาหะจากการมอดูเลตแบบกดคลื่นพาห์ (suppressed carrier modulation) และเนื่องจากสัญญาณจะไม่บรรจุองค์ประกอบคลื่นพาห์เฉลี่ยใดๆ เลย (any average carrier component) ผลของการมอดูเลตก็ควรที่จะถูกกำจัดทิ้งไปทางใดทางหนึ่ง เพื่อที่ว่าลูป (loop)จะสามารถทำการตามคลื่นพาห์ได้ กระบวนการทั้งหมดนี้สามารถทำได้ง่ายสำหรับการมอดูเลตบีพีเอสเคแบบกดคลื่นพาห์ (BPSK suppressed carrier modulation)โดยการใช้อุปกรณ์ในการทำการยกกำลัง 2 (squaring device) สมมติให้สัญญาณที่ถูกมอดูเลตแล้วมีสมการดังนี้

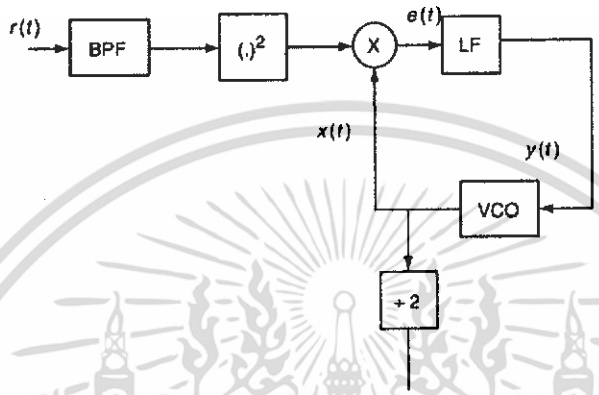
$$s(t) = m(t)\cdot\cos(\omega t) \tag{2.91}$$

โดยที่ m(t) คือแฟคเตอร์ของการมอดูเลต (ซึ่งมีค่าเป็น +1 หรือ –1 ด้วยความเป็นไปได้ที่เท่ากัน) จากนั้นสัญญาณที่ถูกทำการยกกำลัง 2 แล้ว (squared signal) สามารถแสดงได้ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

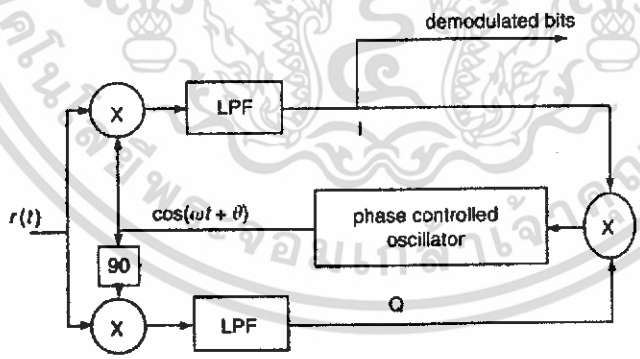
$$s^2(t) = m^2(t)\cdot\cos^2(\omega t)$$
$$= 1\cdot\cos^2(\omega t) = \frac{1}{2}+\frac{1}{2}\cos(2\omega t) \qquad (2.92)$$

ถ้าเฟสล็อคลูปถูกป้อนด้วยสัญญาณยกกำลัง 2 ที่ได้รับการกรองมาแล้ว (filtered squared signal) สัญญาณพาหะก็จะได้รับการกู้คืน บล็อคไดอะแกรม (Block Diagram) สำหรับลูปดังกล่าวแสดงดังรูป



รูปที่ 2.48 สแควริง ลูป (Squaring Loop)

อีกวิธีหนึ่งที่จะกำจัดการมอดูเลตจากสัญญาณบีพีเอสเคแบบกดคลื่นพาห์ และตามสัญญาณพาหะ คือ การใช้คอสแทส ลูป ไดอะแกรมวิธีการของลูปนี้แสดงดังรูป



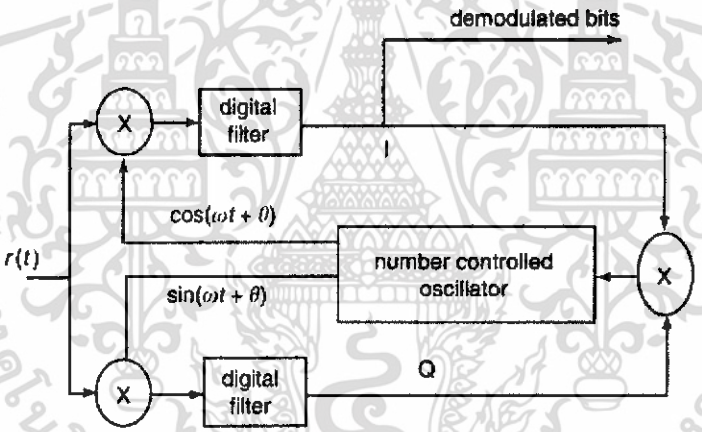
รูปที่ 2.49 การสร้างคอสแทส ลูปแบบธรรมดา

แม้ว่าลูปทั้ง 2 จะดูค่อนข้างแตกต่างกัน แต่ก็สามารถแสดงได้ว่าการทำงานมันนั้นนำไปสู่ผลลัพธ์เดียวกัน จากมุมมองของการสร้าง เรามีข้อที่จะต้องพิจารณาสำหรับแต่ละวิธีการ การสร้างอุปกรณ์ยกกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 นั้นเป็นข้อเสียหลักของสแควริง ลูป (squaring loop) เหตุผลคือปัญหาเรื่องความถี่ที่สูง สำหรับคอสแทสลูปนั้นก็มีประเด็นอื่นที่สำคัญอีก 2 ประเด็น หนึ่งคือ โคไซน์และไซน์ออสซิเลชันที่สร้างขึ้น(the generated cosine and sine oscillation) ควรจะมีเฟสต่างกัน 90 องศาพอดี และสอง คือ ในการที่จะให้ได้มาซึ่งประสิทธิภาพการทำงานสูงสุด ตัวกรองความถี่ต่ำผ่าน (Low-pass filter: LPF) ทั้ง 2 ในสาขาอินเฟส และควอเดรเจอร์ (in-phase and quadrature branch) ของโมเดลควรจะแมตช์ (match) กันพอดี

เนื่องจากรูปร่างที่ถูกต้องของรูปสัญญาณโคไซน์และไซน์ นั้นไม่สำคัญสำหรับลูปนี้ การสร้างในทางปฏิบัติจึงมักใช้สแควว-ออฟ เวอร์ชัน (squared-off version) ของโคไซน์และไซน์ วงจรดิจิตอลเช่น ฟลิป-ฟลอป จะถูกใช้สำหรับปรับแต่งความแตกต่างของเฟสให้ได้ 90 องศาพอดี และก็เป็นไปได้เช่นกันที่จะให้ได้มาซึ่งการแมตช์ที่พอดีระหว่างตัวกรอง โดยการใช้ตัวกรองแบบดิจิตอลซึ่งประกอบด้วยการคูณและการรวมในทางตรรก (logical) ของอินพุทตัวอย่าง (input samples) รูปนี้แสดงการสร้างแบบดิจิตอลโดยทั่วไปของคอสแทสลูป ซึ่งสามารถสร้างได้โดยง่ายด้วยชิปดีเอสพี (DSP) และหน่วยความจำรอม (ROM)



รูปที่ 2.50 คอสแทส ลูปแบบดิจิตอลโดยการใช้เอ็นซีโอ (NCO) เป็นตัวสังเคราะห์ความถี่

ในวิธีการนี้ การคูณและการกรองแบบดิจิตอลจะทำโดยใช้ชิปดีเอสพีส่วนฟังก์ชันโคไซน์และไซน์นั้น จะได้มาจากการเปิดดูตาราง ปัญหาหลักของงานออกแบบนี้คือความถี่ที่ต่ำ ซึ่งดีเอสพีส่วนใหญ่ใช้ในการประมวลผลข้อมูลและนั่นหมายถึงความถี่คลื่นพาห์ที่สามารถยอมรับได้นั้นต่ำ เราสามารถนำเอฟพีจีเอมาใช้เพื่อให้ได้ความถี่คลื่นพาห์ที่สูงขึ้นได้

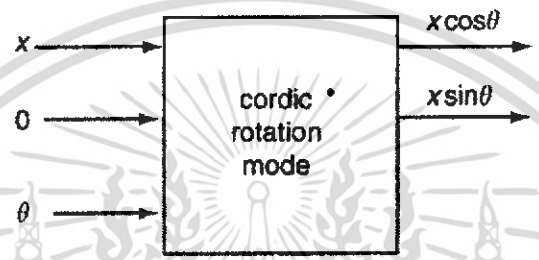
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.1 คอสแทส ลูปด้วยวิธีการสร้างแบบคอร์ดิก

ในส่วนของคอร์ดิก เอาท์พุทของส่วนนี้คือ อินพุทที่ถูกหมุนไป (the rotated version of the input) และสามารถแสดงได้ดังสมการ

$$\begin{aligned} x_{out} &= \frac{1}{\mathrm{A_n}} \cdot \left(x_{in} \cos\phi - y_{in} \sin\phi \right) \\ y_{out} &= \frac{1}{\mathrm{A_n}} \cdot \left(y_{in} \cos\phi + x_{in} \sin\phi \right) \end{aligned} \tag{2.93}$$

จากนั้นพิจารณากรณีที่แสดงในรูป



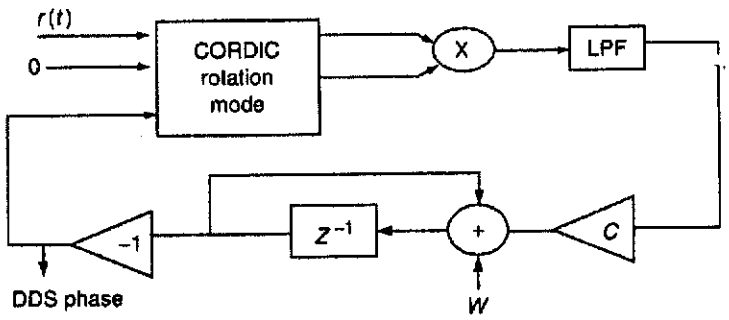
รูปที่ 2.51 ส่วนคอร์ดิก, $X_{in} = x$, $Y_{in} = 0$

ซึ่ง $y_{in} = 0$ เอาท์พุทจะเป็นดังสมการ

$$\begin{aligned} x_{out} &= \frac{1}{\mathrm{A_n}} \cdot \left(x_{in} \cos\phi \right) \\ y_{out} &= \frac{1}{\mathrm{A_n}} \cdot \left(x_{in} \sin\phi \right) \end{aligned} \tag{2.94}$$

ดังนั้นถ้าเราป้อน x_{in} ด้วยสัญญาณที่ได้รับมา r(t) เอาท์พุทของส่วนคอร์ดิกก็จะเป็นสัญญาณที่ถูกคูณด้วยไซน์และโคไซน์ ของ ϕ เมื่อเปรียบเทียบสมการที่ผ่านมากับสมการของคอสแทส ลูปธรรมดาจะเห็นว่าถ้า ϕ ถูกแทนที่ด้วยเฟสของออสซิเลเตอร์ เอาท์พุทของส่วนคอร์ดิกจะเป็นองค์ประกอบอินเฟสและควอเดรเจอร์ของสัญญาณนั้น อย่างไรก็ตาม เฟสของออสซิเลเตอร์ซึ่งเป็นฟังก์ชันเชิงเส้นของฟีดแบค (feedback) ก็สามารถสร้างได้ง่ายโดยการใช้รีจิสเตอร์และแอดเดอร์ (register and adder) ที่ไม่ซับซ้อน วิธีการนี้แสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.52 การสร้างแบบคอร์ดิกของคอสแทส ลูป

การสร้างแบบใหม่นี้มีข้อดีหลายอย่าง อย่างแรกคือไม่มีฟังก์ชันไซน์หรือโคไซน์ที่ชัดเจน (no explicit sine or cosine function) นั่นหมายถึงสามารถประหยัดหน่วยความจำได้มาก เนื่องจากตารางที่ใช้เปิดดูของคอร์ดิก (CORDIC look up table) จะเก็บแค่เพียงค่า n ของ $\tan^{-1}\left(2^{-i}\right)$ ในขณะที่ตารางของไซน์และโคไซน์จะใช้หน่วยความจำเท่ากับลำดับของขนาดของ 2^n (consume a memory of the order of magnitude of 2^n) ข้อดีอย่างที่สอง ก็คือ การสร้างนี้จะไม่รวมตัวคูณที่ต้องใช้งานหนักทั้งสอง (2 computationally intensive multipliers) นอกจากนี้ เรายังสามารถเปลี่ยนตัวคูณหน่วยสุดท้าย (last multiplier unit) ให้เป็นโปรดักต์ (product) ของบิทเครื่องหมาย เพื่อให้ได้การทำงานที่มีประสิทธิภาพขึ้นเช่นกัน แต่จะสามารถทำได้เพียงสำหรับช่อง(channels) ที่มีค่าเอสเอ็นอาร์ (SNR : signal-to-noise ratio) สูงเท่านั้น การสร้างคอสแทส ลูปนี้ได้ประโยชน์มาจากข้อดีทั้งหมดของการสร้างแบบดิจิตอล เช่น การแมตช์กันของตัวกรองความถี่ต่ำผ่าน ดังที่เห็นในรูปนี้ เอาท์พุทจากการดีมอดูเลต (demodulate) ก็มาสามารถได้มาจากสาขาอิน-เฟสเช่นกัน

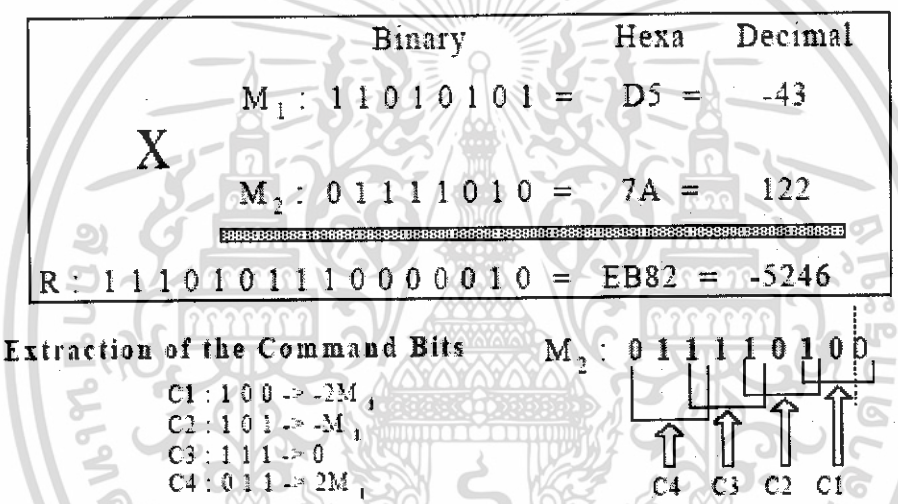
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 บูธ อัลกอริธึม (BOOTH Algorithm)

เนื่องจากการคูณเป็นกระบวนการที่ช้า มันจึงอยู่ในส่วนที่สำคัญของไมโครโปรเซสเซอร์ (Microprocessor) โดยเฉพาะอย่างยิ่งของ ดีเอสพี, เอเอสไอซี (ASIC) และ เอฟพีจีเอ ซึ่งคำนวณกระบวนการของจำนวนเต็มและทศนิยมในแบบขนานภายใน 1 รอบ (1 cycle)

การคูณนั้นสามารถแบ่งออกได้เป็น 3 ส่วน คือ การสร้างผลคูณย่อย (Partial products generation), ซัมเมชันเนทเวิร์ค (Summation network) และ ตัวบวกไฟนอลแคร์รีพร็อพพาเกชัน (final carry propation adder) เนื่องจากการดีเลย์ (delay) ของตัวคูณจะขึ้นอยู่กับจำนวนของผลคูณย่อยที่จะถูกบวกเข้ามา จึงเป็นการน่าสนใจที่จะใช้วิธีบูธ อัลกอริธึม เนื่องจากมันจะลดดีเลย์ด้วยแฟคเตอร์ของ 2 แต่ก็สร้างบิทเพิ่มเติมสำหรับการเพิ่มเข้ามาของเครื่องหมายและ 2's คอมพลีเมนเทชัน (2's complementation)

รูปที่ 2.53 แสดงตัวอย่างของกระบวนการบูธ สำหรับการคูณที่พิจารณาเครื่องหมายซึ่งแสดงได้โดย $M_1(7:0) \times M_2(7:0) = R(15:0)$

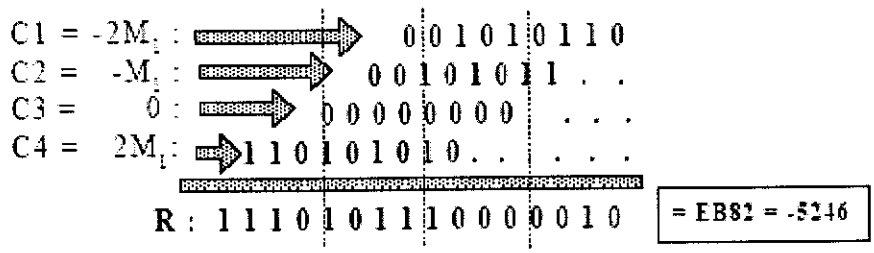


รูปที่ 2.53 แสดงตัวอย่างของการคูณแบบพิจารณาเครื่องหมาย (8x8)

SELECTION TABLE OF PARTIAL PRODUCT	
Bits of Operand M_2	**Selection**
000	+0
001	+ Multiplicand
010	+ Multiplicand
011	+2 x Multiplicand
100	-2 x Multiplicand
101	- Multiplicand
110	- Multiplicand
111	-0

ตารางที่ 2.1 แสดงซีเลคชันเทเบิล (selection table) ของบูธ อัลกอริธึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.54 ผลของการคูณแบบพิจารณาเครื่องหมาย (8x8)

ดังแสดงในรูปที่ 2.53 แต่ละ 3 บิทของ M_2 คือ คำสั่งบูธที่เลือกมาจากตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 การมอดูเลชันแบบ QAMโดยใช้คอร์ดิกอัลกอริธึม

จากสมการมอดูเลชันแบบ QAM

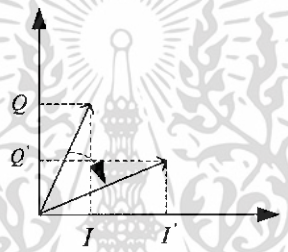
$$S_{QAM}(n) = I(n)\cos(\omega t(n)) + Q(n)\sin(\omega t(n)) \tag{3.1}$$

และจากหลักการของคอร์ดิกอัลกอริธึมมีรูปแบบสมการพื้นฐานคือ

$$x' = x\cos\phi - y\sin\phi \tag{3.2}$$

$$y' = y\cos\phi + x\sin\phi \tag{3.3}$$

ซึ่งโครงสร้างของสมการมอดูเลชันแบบ QAM สามารถสร้างได้จากการใช้คอร์ดิกอัลกอริธึม ดังรูปที่ 3.1 โดยคู่แกนเรคแทงกูลาร์นั้นหมุนตามเข็มนาฬิกา โดยมุม Ang จากนั้นพิกัดของเวกเตอร์ (I,Q) จะย้ายไปที่ (I',Q')



รูปที่ 3.1 การหมุนเวกเตอร์

จากการหมุนเวกเตอร์ได้สมการดังนี้

$$I' = I\cos(Ang) + Q\sin(Ang) \tag{3.4}$$

$$Q' = Q\cos(Ang) - I\sin(Ang) \tag{3.5}$$

โดยที่ $\tan(Ang) = \pm 2^{-i}$ (3.6)

จากที่กล่าวมาแล้วในบทที่ 2 สามารถเขียนสมการใหม่ได้ดังนี้

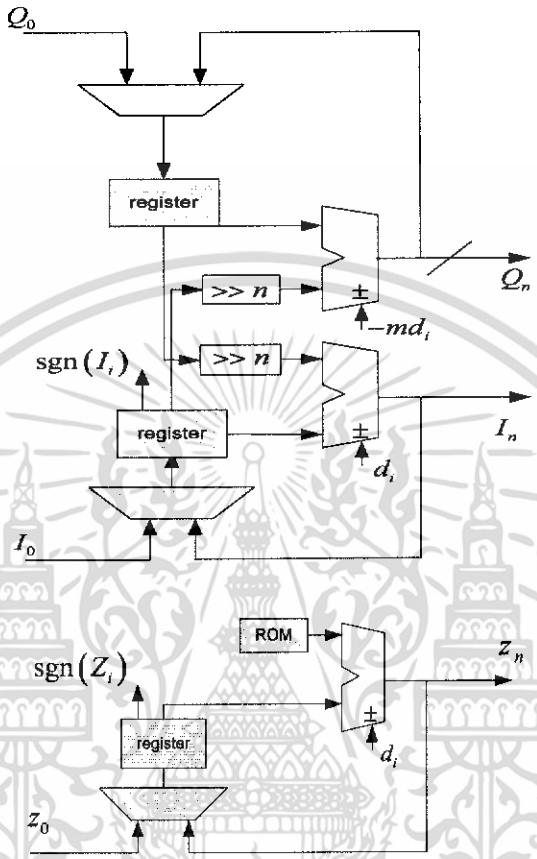
$$I_{i+1} = \left[I_i + Q_i d_i 2^{-i}\right] \tag{3.7}$$

$$Q_{i+1} = \left[Q_i - I_i d_i 2^{-i}\right] \tag{3.8}$$

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \tag{3.9}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการมอดูเลชันแบบ QAM ซึ่งตรงกับสมการที่ 3.4 ในที่นี้จึงใช้เฉพาะเทอมของ I' เท่านั้นสามารถทำได้โดยการใช้คอร์ดิกอัลกอริธึม แสดงดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างของคอร์ดิกอัลกอริธึม

โครงสร้างแบบวนรอบการทำงานคอร์ดิกนั้นสามารถได้มาโดยง่ายจากการทำซ้ำแต่ละสมการที่แตกต่างกัน 3 สมการในฮาร์ดแวร์ ดังแสดงในรูปที่ 3.2 ฟังก์ชันการตัดสินใจ d_i จะแปรตามเครื่องหมายของ I หรือ z รีจิสเตอร์ ในการทำงานค่าเริ่มต้นจะถูกโหลดผ่านมัลติเพล็กเซอร์เข้าสู่ I, Q, และ z รีจิสเตอร์ จากนั้นแต่ละรอบต่อไปของคล็อกไซเคิล(clock cycle) ค่าจากรีจิสเตอร์จะถูกส่งผ่านชิฟท์เตอร์และแอดเดอร์-ซับแทรคเตอร์ และผลลัพธ์จะถูกส่งกลับไปที่รีจิสเตอร์ ชิฟท์เตอร์จะถูกเปลี่ยนแปลงในแต่ละรอบของการทำงานเพื่อให้ได้มาซึ่งการชิฟท์ที่ต้องการสำหรับแต่ละรอบการทำงาน เช่นเดียวกับแอดเดรสของรอมที่จะเพิ่มขึ้นในแต่ละรอบการทำงานเพื่อที่ z แอดเดอร์-ซับแทรคเตอร์จะได้รับค่ามุมย่อยๆที่เหมาะสม ในรอบสุดท้ายของการทำงานนั้นผลลัพธ์จะถูกอ่านโดยตรงจากแอดเดอร์-ซับแทรคเตอร์ และสเตทแมชีนอย่างง่ายจะต้องถูกนำมาใช้เพื่อการ

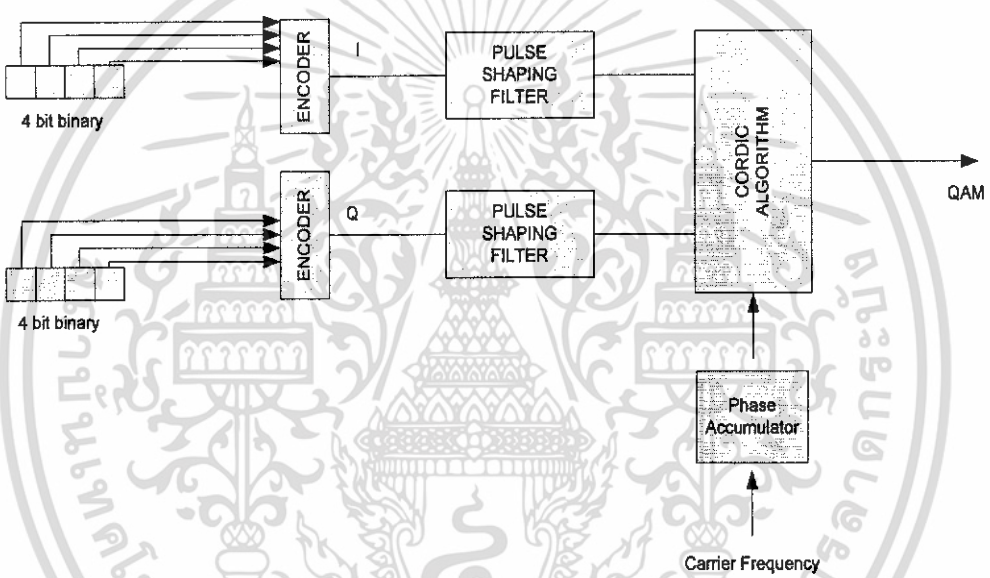
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดตามบันทึกรอบการทำงานปัจจุบันและเพื่อเลือกมุม (degree) ของการชิฟท์และแอดเดรสของรอมสำหรับแต่ละรอบการทำงาน

3.2 การมอดูเลตสัญญาณแบบ QAM

การมอดูเลตสัญญาณแบบ QAM ที่ใช้ในการทดลอง จะใช้เลขฐานสองจำนวน 4 บิททำการเข้ารหัส ซึ่งจะทำการแบ่งเป็นช่องสัญญาณ I และ Q จากนั้นทำการผ่านพัลส์เชปปิงฟิลเตอร์เพื่อให้แบนด์วิธแคบลงเป็นการลดการแทรกสอดของสัญญาณ แล้วนำค่าที่ได้ไปผ่านส่วนของคอร์ดิกก็จะได้สัญญาณ QAM ออกมา

โครงงานนี้ใช้บิทเรท (Bit Rate) 9600 bps, บอดเรท (Baud Rate) 2400 baud per sec, Sampling frequency 14400 Hz และ ความถี่คลื่นพาห์ 1800 Hz



รูปที่ 3.3 การมอดูเลชันแบบ QAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

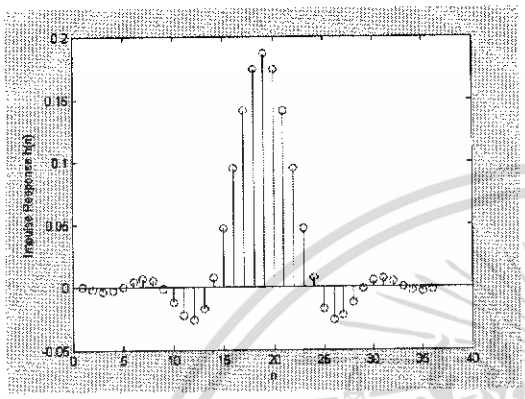
3.2.1 การออกแบบวงจรเรสโคไซน์ฟิลเตอร์ (Raised Cosine filter) โดยใช้โปรแกรมแมทแลป

การออกแบบวงจรเรสโคไซน์ฟิลเตอร์ที่ใช้ในการทดลอง

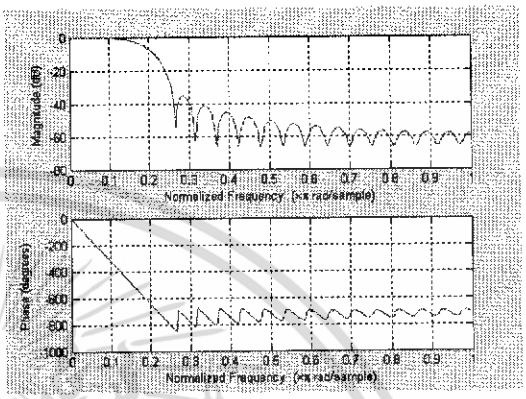
โรลล์ออฟแฟคเตอร์ (Roll-off factor) = 3

อัตราโอเวอร์แซมปลิ้ง (Over sampling rate) = 6

จำนวนค่าสัมประสิทธิ์ฟิลเตอร์ = 36 ค่า

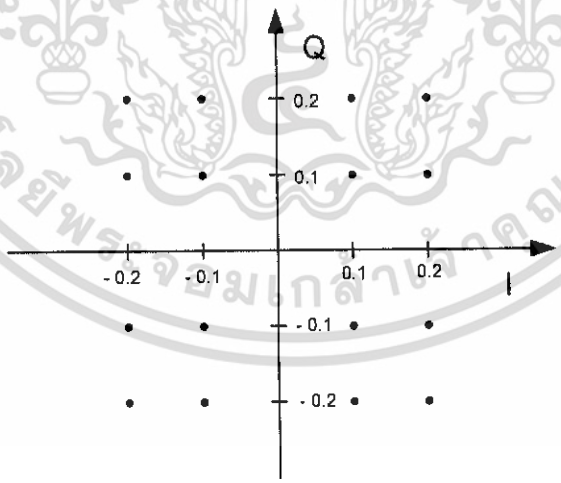


รูปที่ 3.4 ผลตอบสนองอิมพัลส์



รูปที่ 3.5 ผลตอบสนองทางความถี่

3.2.2 คอนสเตลเลชัน (Constellation) ของสัญญาณ 16 QAM ที่ใช้ในการทดลอง



รูปที่ 3.6 แสดงคอนสเตลเลชันของสัญญาณ 16 QAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

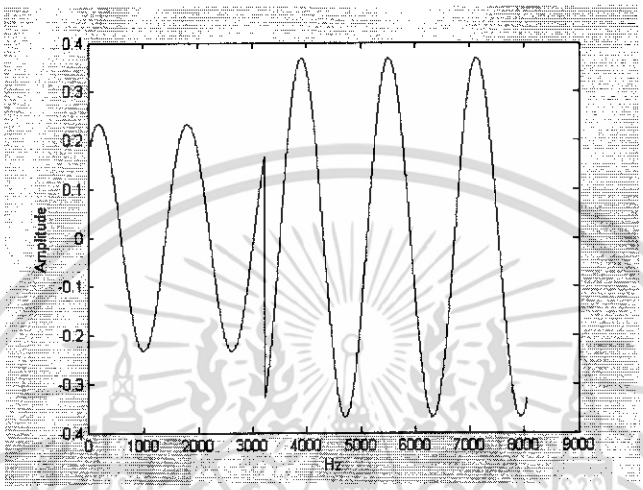
ข้อมูลรหัสดิจิตอล	แกน I	แกน Q
0000	-0.2	-0.2
0001	-0.2	-0.1
0010	-0.2	0.1
0011	-0.2	0.2
0100	-0.1	-0.2
0101	-0.1	-0.1
0110	-0.1	0.1
0111	-0.1	0.2
1000	0.1	-0.2
1001	0.1	-0.1
1010	0.1	0.1
1011	0.1	0.2
1100	0.2	-0.2
1101	0.2	-0.1
1110	0.2	0.1
1111	0.2	0.2

ตารางที่ 3.1 แสดงการเข้ารหัสข้อมูลดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

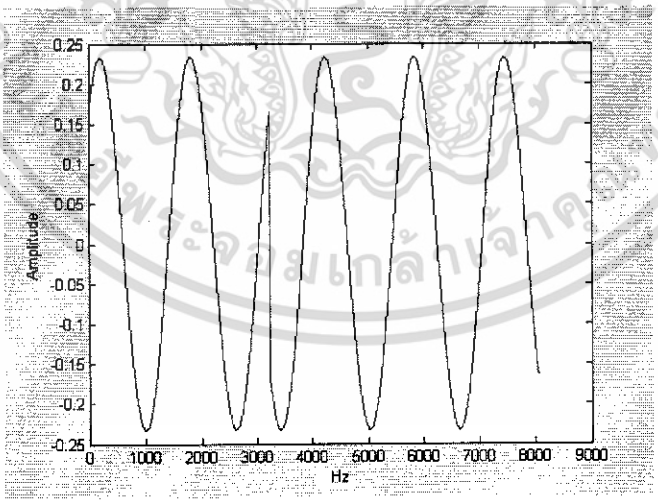
3.2.3 ผลการทดลองการออกแบบวงจรคอร์ดิกมอดูเลเตอร์โดยใช้โปรแกรมแมทแลป

1) กำหนดให้จุดที่ 1 คือ I=0.1, Q=0.1 และ จุดที่ 2 คือ I=0.1, Q=-0.2



รูปที่ 3.7 การมอดูเลชันจากการออกแบบด้วยโปรแกรมแมทแลป

2) กำหนดให้จุดที่ 1 คือ I=0.1, Q=0.1 และ จุดที่ 2 คือ I=-0.1, Q=-0.1



รูปที่ 3.8 การมอดูเลชันจากการออกแบบด้วยโปรแกรมแมทแลป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การดีมอดูเลตสัญญาณแบบ QAM .

การดีมอดูเลตสัญญาณแบบ QAM ที่ใช้ในการทดลองจะใช้วงจรภาครับแบบคอสแทส ลูป และคอร์ดิก อัลกอริธึม

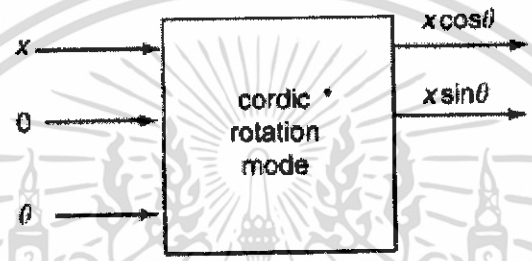
ในส่วนของคอร์ดิก เอาท์พุทของส่วนนี้คือ อินพุทที่ถูกหมุนไป และเอาท์พุทจะเป็นดังสมการ

ให้ $y_{in} = 0$, $X_{in} = x$, $Y_{in} = 0$

$$x_{out} = \frac{1}{A_n} \cdot (x_{in} \cos\phi) \tag{3.10}$$

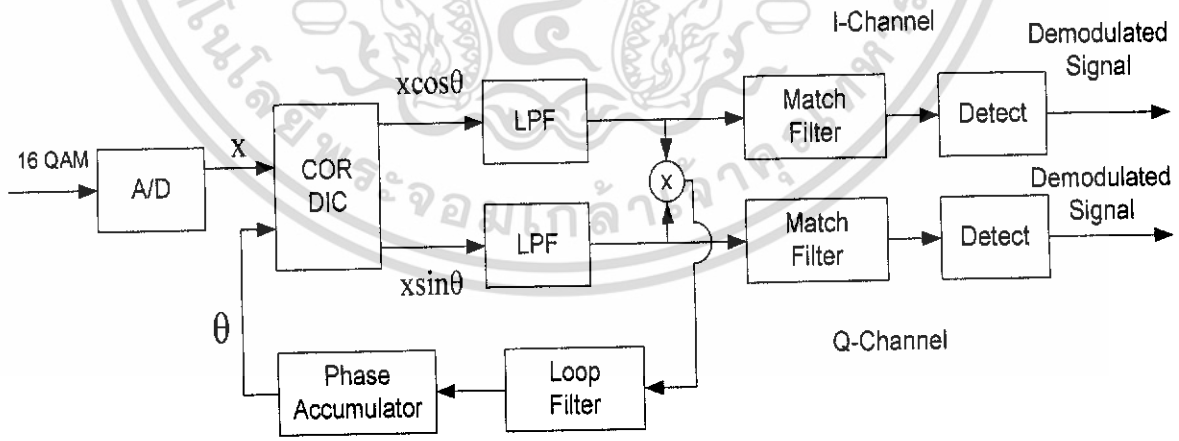
$$y_{out} = \frac{1}{A_n} \cdot (x_{in} \sin\phi) \tag{3.11}$$

จากนั้นพิจารณากรณีที่แสดงในรูป



รูปที่ 3.9 ส่วนคอร์ดิก

จากรูปที่ 3.10 เมื่อรับสัญญาณที่ถูกมอดูเลตเข้ามาผ่านวงจรดีมอดูเลตแล้วก็จะได้สัญญาณข่าวสารออกมาจาก I- channel จากนั้นนำสัญญาณที่ได้ผ่านแมตซ์ฟิลเตอร์ทางด้านรับ แล้วนำผลที่ได้ไปแสดงผลทางคอมพิวเตอร์



รูปที่ 3.10 การดีมอดูเลตแบบ QAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

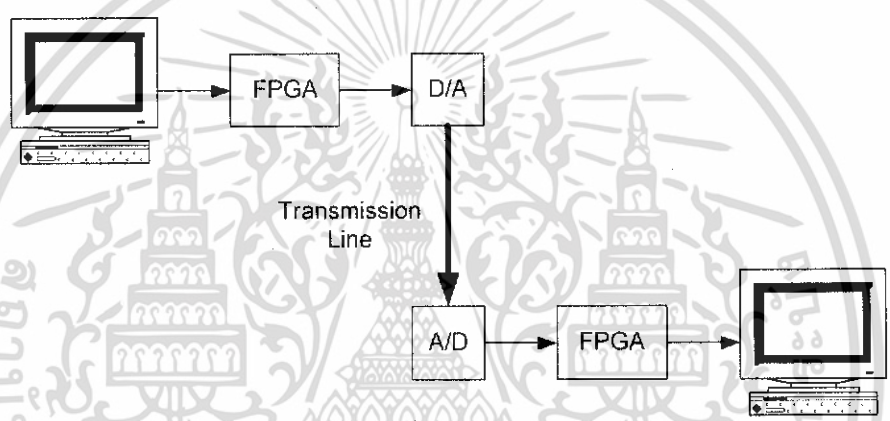
บทที่ 4

การทดลองและผลการทดลอง

ในการออกแบบส่วนต่างๆของการมอดูเลชันจะใช้คอร์ดิกอัลกอริธึม และใช้ตัวคูณแบบบูธอัลกอริธึมในส่วนของเรสโคไซน์ฟิลเตอร์ ทำการเขียนโปรแกรมโดยให้แต่ละส่วนทำงานตามที่ได้ออกแบบโดยใช้ภาษา VHDL ทำการ Compile และจำลองการทำงานของโปรแกรมแต่ละส่วนที่ได้เขียนขึ้นให้ได้ตามผลที่ออกแบบไว้ และวัดผลของส่วนต่างๆด้วยออสซิโลสโคป

4.1 การออกแบบการทดลองสำหรับการวัดคุณลักษณะ

บล็อกไดอะแกรมของการทดลองเป็นดังนี้

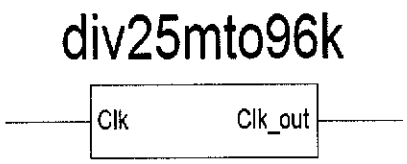


รูปที่ 4.1 การเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะสัญญาณ

4.1.1 ส่วนของการสร้างวงจรหารความถี่ (DIV)

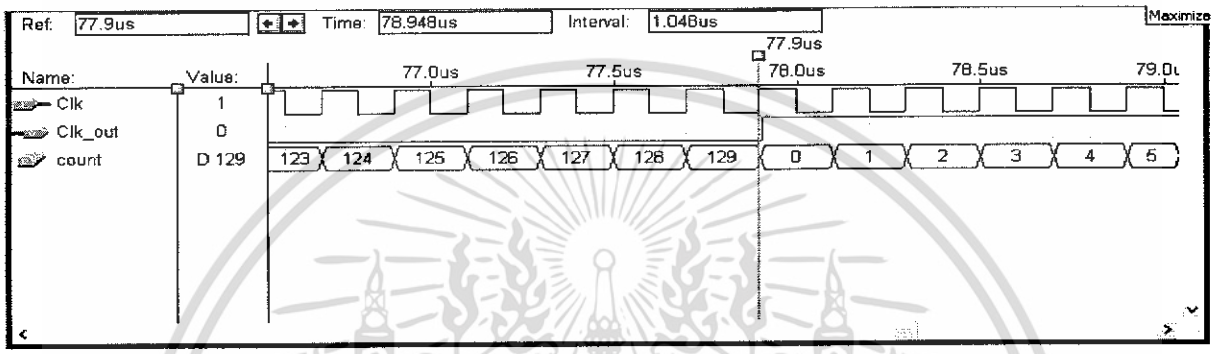
วงจรหารความถี่(DIV) ทำหน้าที่หารความถี่ที่ได้จากออสซิลเลเตอร์ที่กำเนิดสัญญาณนาฬิกาให้ได้อัตราการส่งข้อมูล (Baud rate) ตามที่เรากำหนดไว้ เขียนโปรแกรมที่สามารถสังเคราะห์ให้เป็นอุปกรณ์ที่มีสัญลักษณ์ (Symbol) ลักษณะดังรูป 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 สัญลักษณ์ของส่วนวงจรหารความถี่

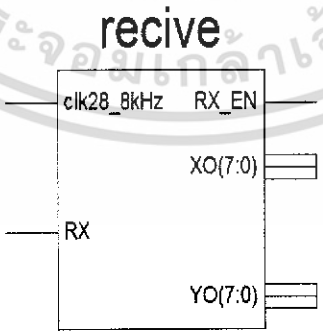
จากโปรแกรมที่เขียนขึ้นสามารถทำการจำลองการทำงาน (Simulation) ได้ดังนี้



รูปที่ 4.3 ผลการจำลองการทำงานของส่วนวงจรหารความถี่

4.1.2 ส่วนของการรับบิทข้อมูลจากพอร์ตอนุกรม (SERIAL_RX)

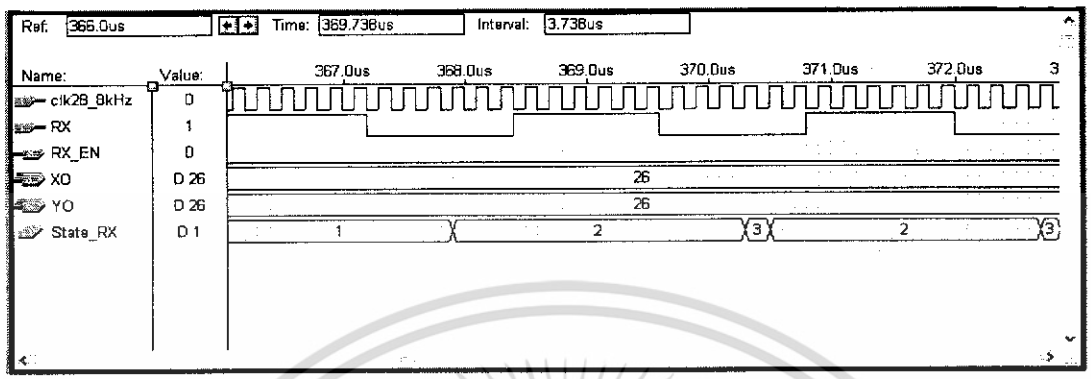
ส่วนของการรับข้อมูลจากพอร์ตอนุกรม ทำหน้าที่รับเฟรมบิทของข้อมูลจากพอร์ตอนุกรมมาทำการแปลงข้อมูลจากบิทอนุกรมเป็นบิทขนาน โดยจะทำการแซมปลิง (Sampling) ค่าอินพุทที่รับเข้ามาจากพอร์ตอนุกรมโดยบิทหนึ่งทำการแซมปลิง 12 ค่า จากนั้นนำค่าที่ได้จากการแซมปลิงมาเก็บไว้เพื่อหาค่าความเป็นไปได้ของแต่ละบิทอินพุท เพื่อป้องกันการรับข้อมูลจากพอร์ตอนุกรมผิดพลาด เมื่อรับข้อมูลครบ 8 ค่า ก็จะส่งต่อไป สามารถสังเคราะห์อุปกรณ์จากโปรแกรมของวงจรการรับบิทข้อมูลจากพอร์ตอนุกรมได้สัญลักษณ์ (Symbol) ดังรูปที่ 4.4



รูปที่ 4.4 แสดงสัญลักษณ์ของวงจรการรับบิทข้อมูลจากพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมที่เขียนขึ้นสามารถทำการจำลองการทำงาน (Simulation) ได้ดังนี้



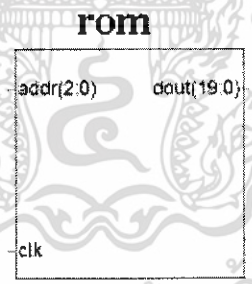
รูปที่ 4.5 แสดงผลการจำลองการทำงานของวงจรการรับบิทข้อมูลจากพอร์ตอนุกรม

4.1.3 ส่วนของเรสโคไซน์ฟิลเตอร์ (Raised Cosine filter)

ส่วนนี้จะใช้การออกแบบด้วยโพลีเฟสฟิลเตอร์ และใช้ตัวคูณแบบบูธอัลกอริธึม

4.1.3.1 ส่วนเก็บค่าและเรียกค่าข้อมูล (Rom)

เป็นส่วนที่ทำหน้าที่เก็บค่าสัมประสิทธิ์ของเรสโคไซน์ฟิลเตอร์โดยสัญญาณอินพุทจะไปชี้ค่าที่แอดเดรสเพื่อเรียกค่าในแต่ละแอดเดรสออกมาเป็นเอาท์พุท

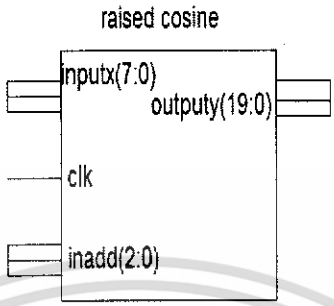


รูปที่ 4.6 แสดงสัญลักษณ์ของวงจรเก็บค่าและเรียกค่าข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

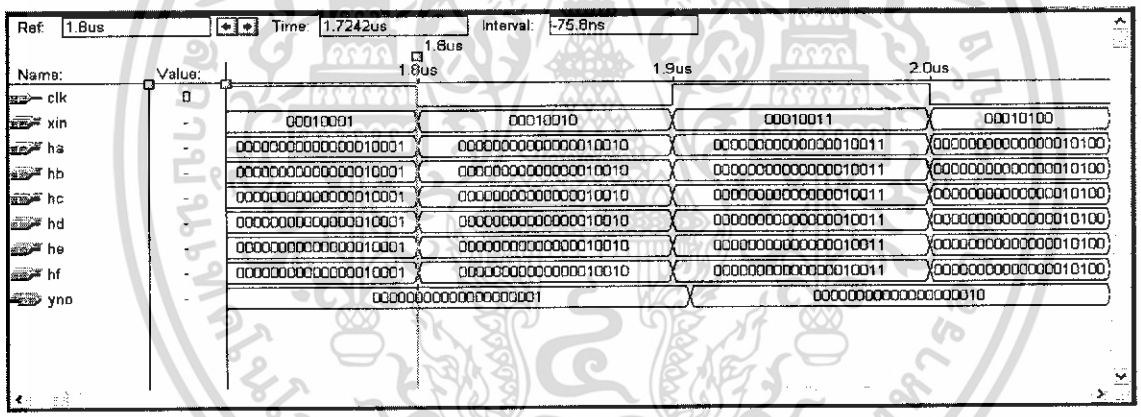
4.1.3.2 ส่วนของตัวคูณ

ส่วนนี้เป็นวงจรการคูณแบบบูธอัลกอริธึม



รูปที่ 4.7 แสดงสัญลักษณ์ของส่วนตัวคูณ

จากโปรแกรมที่เขียนขึ้นสามารถทำการจำลองการทำงาน (Simulation) ได้ดังนี้

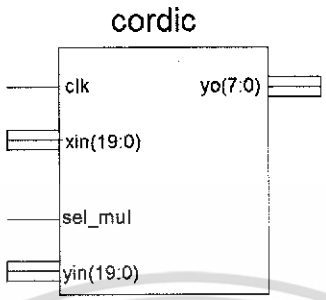


รูปที่ 4.8 แสดงผลการจำลองการทำงานของวงจรเรสโคไซน์ฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

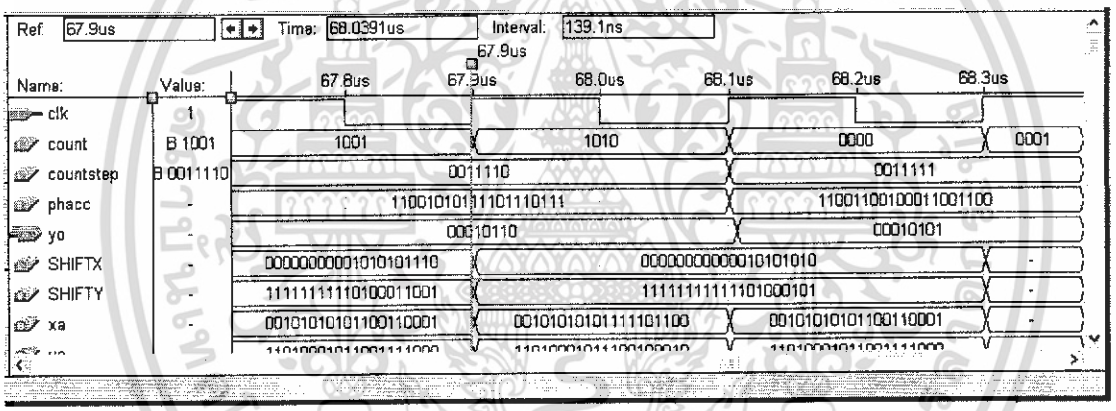
4.1.4 ส่วนของคอร์ดิกอัลกอริธึม (cordic)

ส่วนของคอร์ดิกอัลกอริธึมเป็นส่วนที่ใช้เป็นตัวมอดูเลตสัญญาณที่ออกมาจากส่วนของเรสโคไซน์ฟิลเตอร์



รูปที่ 4.9 แสดงสัญลักษณ์ของส่วนคอร์ดิก

จากโปรแกรมที่เขียนขึ้นสามารถทำการจำลองการทำงาน (Simulation) ได้ดังนี้

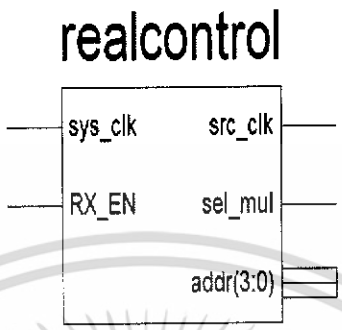


รูปที่ 4.10 แสดงผลการจำลองการทำงานของวงจรคอร์ดิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

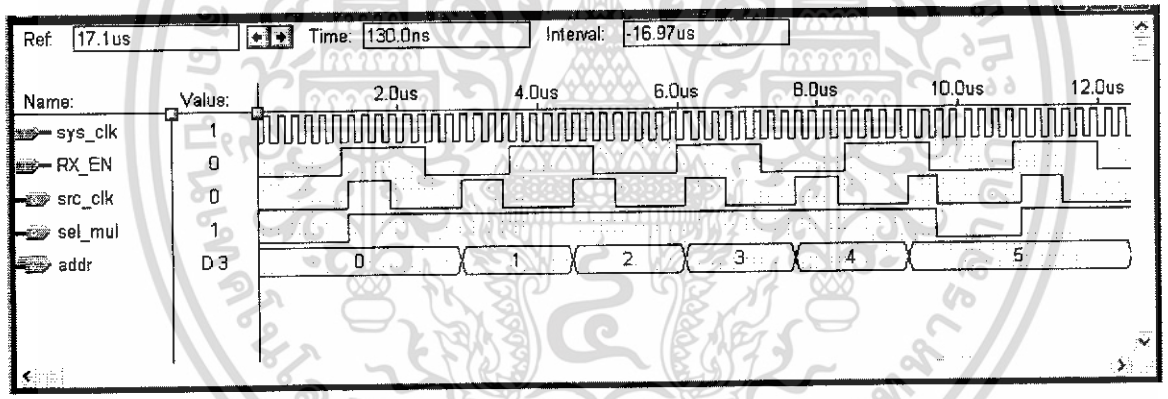
4.1.5 ส่วนของคอนโทรลยูนิต (Control unit)

เนื่องจากวงจรแต่ละส่วนทำงานแยกกันจึงจำเป็นต้องมีคอนโทรลยูนิตเพื่อควบคุมการทำงานของแต่ละส่วนให้สอดคล้องกันตามที่กำหนดไว้



รูปที่ 4.11 แสดงสัญลักษณ์ของส่วนคอนโทรลยูนิต

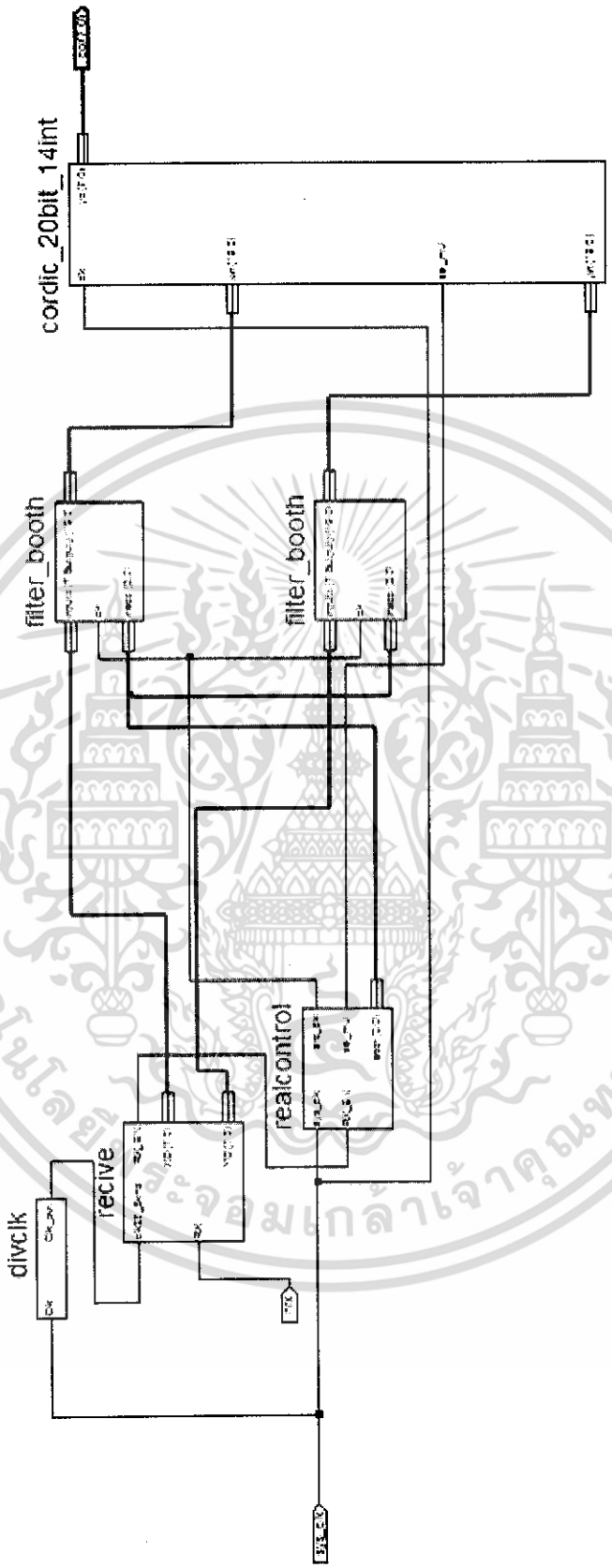
จากโปรแกรมที่เขียนขึ้นสามารถทำการจำลองการทำงาน (Simulation) ได้ดังนี้



รูปที่ 4.12 แสดงผลการจำลองการทำงานของวงจรคอนโทรลยูนิต

จากวงจรแต่ละส่วนที่กล่าวมาเมื่อนำมาทำการคอนฟิก(configure)โปรแกรมที่ออกแบบทั้งหมดลงบนอุปกรณ์ FPGA ไม่สามารถทำได้ แล้วเมื่อดูรายงานผลการวิเคราะห์โปรแกรม (view synthesis report) ได้ผลดังรูป

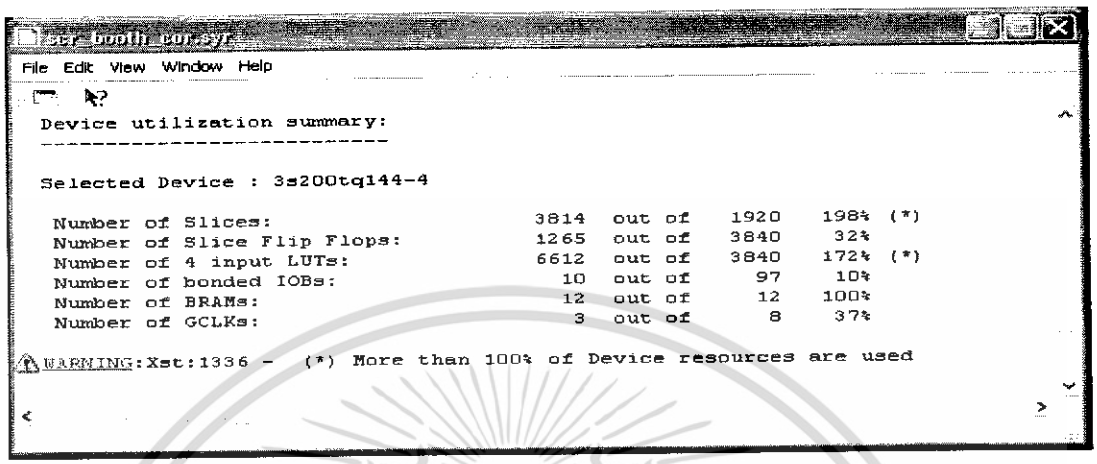
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4.13 ส่วนประกอบของวงจรมอดูเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรแต่ละส่วนที่กล่าวมาเมื่อนำมาทำการคอนฟิก(configure)โปรแกรมลงบนอุปกรณ์ FPGA ไม่สามารถทำได้ แล้วเมื่อดูรายงานผลการวิเคราะห์โปรแกรม (view synthesis report) พบว่าจำนวนเกทไม่พอ



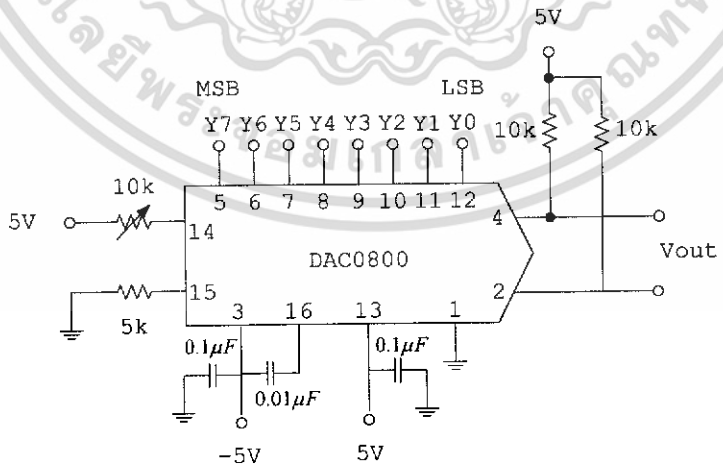
รูปที่ 4.14 ผลการวิเคราะห์โปรแกรม(view synthesis report)

4.2 การนำโปรแกรมที่ออกแบบโดยภาษา VHDL ไปสร้างเป็นวงจรมอดูเลเตอร์

เนื่องจากการทดลองติดปัญหาตรงที่ไม่สามารถคอนฟิก(configure)โปรแกรมที่ออกแบบทั้งหมดลงบนอุปกรณ์ FPGA ได้ จึงไม่สามารถแสดงผลของการออกแบบในรูปที่ 4.1 ได้ จึงได้ออกแบบการทดลองเพื่อทดสอบวงจรในแต่ละส่วน

การทดลองนี้จะมีอุปกรณ์ที่ใช้อยู่ 2 ส่วนดังนี้

1. วงจรแปลงสัญญาณดิจิตอล(Digital) เป็นสัญญาณเชิงอนาล็อก(Analog) ใช้อุปกรณ์ Digital To Analog Converter (DAC)
2. อุปกรณ์ FPGA ของบริษัท Xilinx ตระกูล Spatan-3 เบอร์ XC3S200

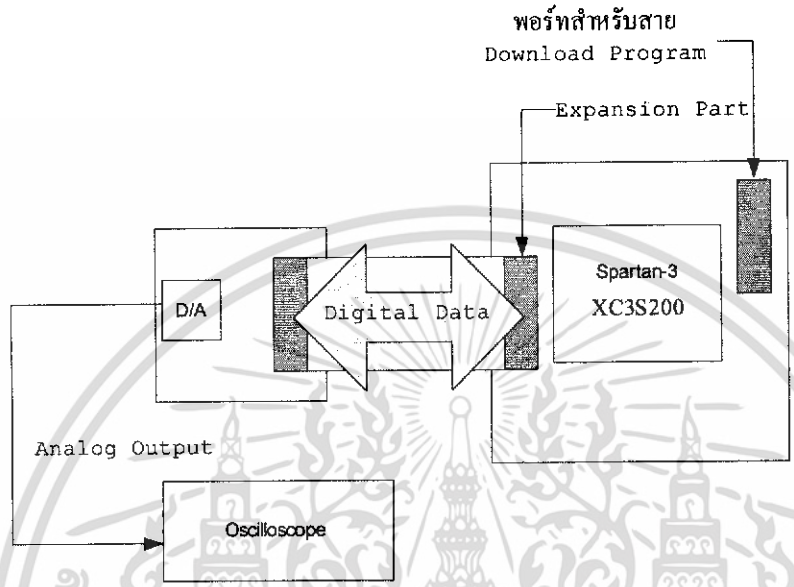


รูปที่ 4.15 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณเชิงอนาล็อก

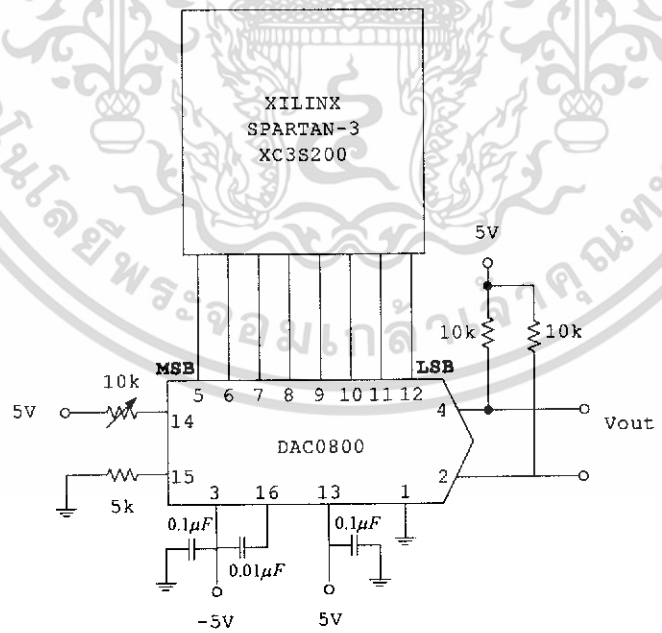
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การออกแบบการทดลองสำหรับการวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบ

ออกแบบวงจรรวมโดยใช้อุปกรณ์ทั้ง 2 ส่วนข้างต้นมาต่อรวมกันดังรูปที่ 4.15 เพื่อนำไปวัดคุณสมบัติของวงจรกรองสัญญาณที่ออกแบบไว้



รูปที่ 4.16 การเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น



รูปที่ 4.17 วงจรรวมสำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

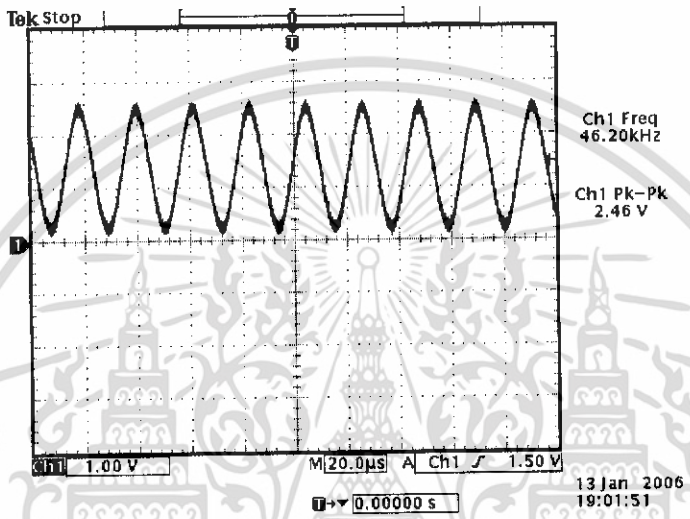
4.2.2 ผลการทดลองวงจรคอร์ดิก ออสซิลเลเตอร์ (Cordic Oscillator)

ในการทดลองนี้จะนำโปรแกรมที่ออกแบบไว้มาคอนฟิกลงบนอุปกรณ์ FPGA ได้สัญญาณเอาท์พุท ที่เกิดจากคอร์ดิก ออสซิลเลเตอร์ โดยมีการวนรอบการทำงาน (Iteration) 11 ครั้ง

4.2.2.1 เพิ่มความต่างเฟส

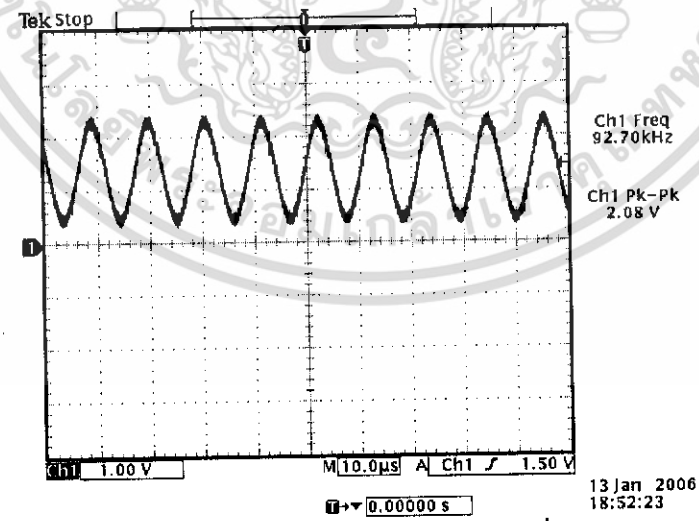
การเพิ่มความต่างเฟส เพื่อให้มีความถี่สูงขึ้นในสัญญาณนาฬิกา(Clock) เดียวกัน นั่นคือการเปลี่ยนค่าระยะห่างของเฟส (Δz_i)

1) กำหนดให้ $\Delta z = 0.1013$ เรเดียน



รูปที่ 4.18 ผลการทดลองที่เกิดจากคอร์ดิกออสซิลเลเตอร์ โดยที่ $\Delta z = 0.1013$ เรเดียน

2) กำหนดให้ $\Delta z = 0.2094$ เรเดียน



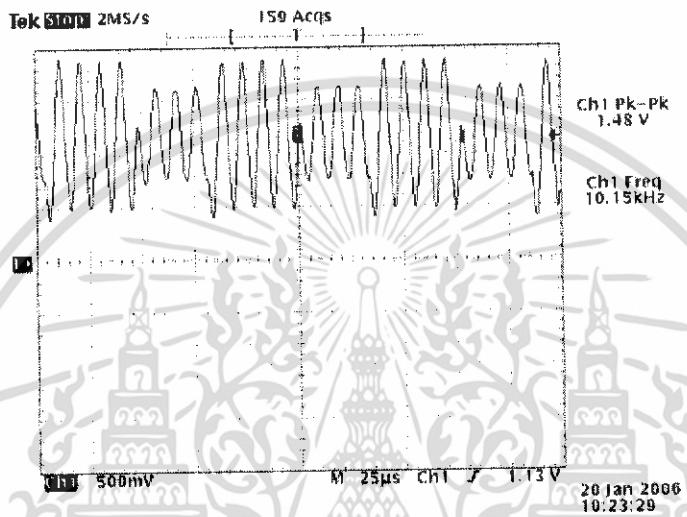
รูปที่ 4.19 ผลการทดลองที่เกิดจากคอร์ดิกออสซิลเลเตอร์ โดยที่ $\Delta z = 0.2094$ เรเดียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองวงจรคอร์ดิกมอดูเลเตอร์ (Cordic modulator)

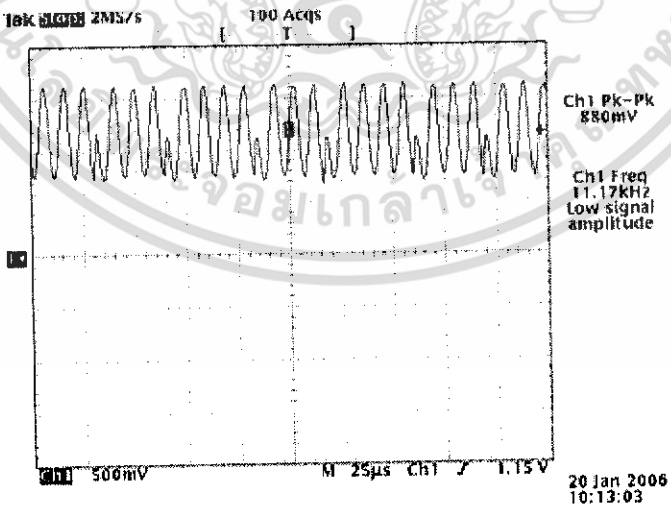
ในการทดลองนี้ทำการนำโปรแกรมที่ออกแบบไว้มาคอนฟิกลงบนอุปกรณ์ FPGA ได้สัญญาณเอาท์พุทที่เกิดจากคอดิก ออสซิลเลเตอร์ โดยมีการวนรอบการทำงาน (Iteration) 11 ครั้ง แล้วให้ค่าอินพุทเป็นคอน สเตลเลชันจำนวน 2 จุดเพื่อแสดงการเปลี่ยนเฟสและแอมปลิจูด

1) กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1 และ จุดที่ 2 คือ I = 0.1, Q = -0.2



รูปที่ 4.20 ผลการมอดูเลชันจากการออกแบบด้วยภาษา VHDL

2) กำหนดให้จุดที่ 1 คือ I = 0.1, Q = 0.1และ จุดที่ 2 คือ I = -0.1, Q = -0.1

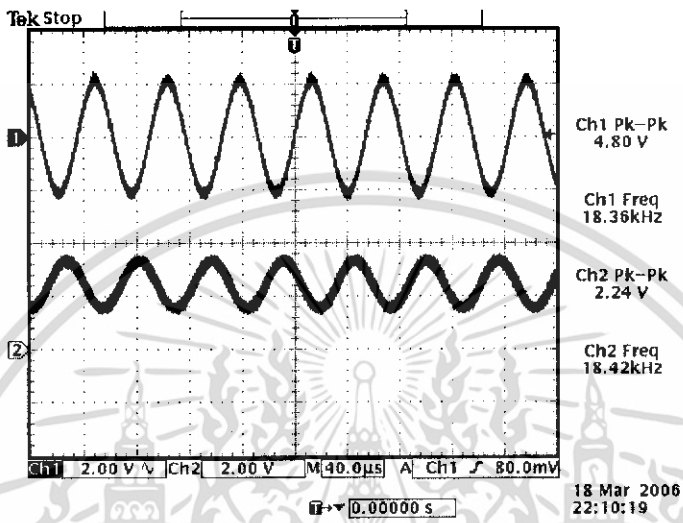


รูปที่ 4.21 ผลการมอดูเลชันจากการออกแบบด้วยภาษา VHDL

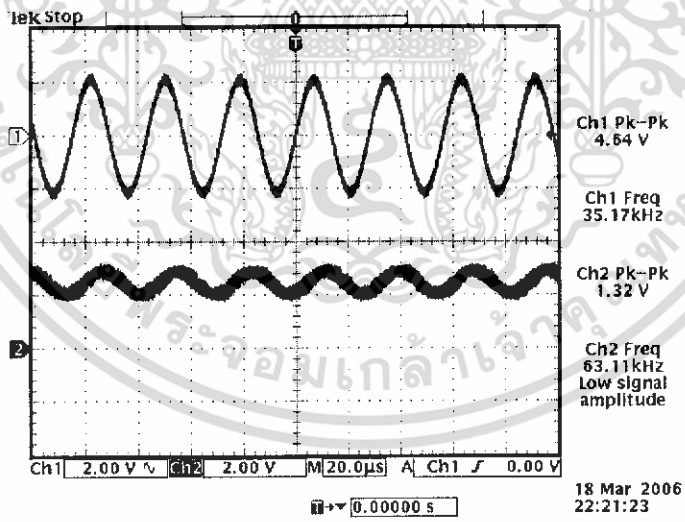
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ผลการทดลองวงจรกรองความถี่ต่ำผ่านแบบผลตอบสนองอิมพัลส์จำกัด

ในการทดลองนี้ทำการทดสอบวงจรคูณแบบบูธอัลกอริธึมในวงจรกรองความถี่ต่ำผ่านแบบผลตอบสนองอิมพัลส์จำกัดอันดับที่ 6 ทำการป้อนสัญญาณ Sinusoidal ขนาด 5 V ความถี่ 18 kHz และ 35 kHz



รูปที่ 4.22 ผลการทดลองที่ได้จากวงจรกรองความถี่ต่ำผ่านอันดับที่ 6



รูปที่ 4.23 ผลการทดลองที่ได้จากวงจรกรองความถี่ต่ำผ่านอันดับที่ 6

ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุทเป็นสัญญาณ Sinusoidal

ช่องสัญญาณที่ 2 แสดงสัญญาณเอาท์พุทจากวงจรกรองความถี่ต่ำผ่านอันดับที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

ทฤษฎีของ CORDIC Algorithm หลักการสำคัญคือการอาศัยความสัมพันธ์ของการเปลี่ยนแปลงตำแหน่งเวคเตอร์ที่มีขนาดเท่ากันหรือการเปลี่ยนตำแหน่งพิจารณาเส้นรัศมีของวงกลมใดๆ ซึ่งจากการเปลี่ยนตำแหน่งนี้จะทำให้ได้มุมเกิดขึ้น และเมื่อทำการโยงความสัมพันธ์ของมุมกับตำแหน่งเวคเตอร์เข้าด้วยกันแล้วก็จะทำให้ได้ความสัมพันธ์ที่เป็นสมการพื้นฐานของ CORDIC Algorithm ทั้งยังสามารถเพิ่มพารามิเตอร์เพื่อเปลี่ยนระบบในการพิจารณาหรือจัดรูปสมการเพื่อหาค่าที่ต้องการได้อีกเช่น ฟังก์ชันไฮเปอร์บอลิกและฟังก์ชันเอ็กซ์โปเนนเชียล เป็นต้น ส่วนการนำไปสร้างเป็นฮาร์ดแวร์ถูกทำให้ง่ายโดยการเลือกพจน์ที่มาคูณผลก็คือสามารถนำไปใช้ในฮาร์ดแวร์ที่เป็นระบบดิจิตอลหรือการคำนวณในระบบคอมพิวเตอร์ได้ง่ายและลดอุปกรณ์ในการสร้างลง อีกทั้งยังสามารถเลือกความละเอียดของการคำนวณจากจำนวนการวนรอบของการทำงานได้อีกด้วย

ในส่วนของโครงสร้างที่ได้ออกแบบไว้นั้น การมอดูเลตสัญญาณแบบ QAM ที่ใช้ในการทดลองจะใช้เลขฐานสองจำนวน 4 บิททำการเข้ารหัสซึ่งแบ่งเป็นช่องสัญญาณ I และ Q จากนั้นส่งผ่านไปยังพัลส์เชปปิงฟิลเตอร์เพื่อให้แบนด์วิธของสัญญาณแคบลงเป็นการลดการแทรกสอดของสัญญาณ แล้วนำค่าที่ได้ไปผ่านส่วนของ CORDIC ก็จะได้สัญญาณ QAM ออกมา ส่วนทางด้านรับนั้น เมื่อรับสัญญาณที่ถูกมอดูเลตเข้ามาและผ่านวงจรดีมอดูเลตแล้ว จะได้สัญญาณข่าวสารออกมาทางช่องสัญญาณ I และเมื่อนำสัญญาณที่ได้มาผ่านแมตช์ฟิลเตอร์แล้วทำการดีเทคท์ (detect) ก็จะได้สัญญาณที่ได้ทำการเข้ารหัสไว้ในตอนแรกไปแสดงผลทางคอมพิวเตอร์ทางด้านรับ

จากการทำการทดลองนั้นแต่ละส่วนสามารถทำงานได้และให้ผลการทดลองถูกต้องตามที่ต้องการ แต่เมื่อรวมทุกส่วนเข้าด้วยกันแล้วไม่สามารถทำการโปรแกรมลงบนชิป FPGA ได้ เนื่องจากจำนวนเกท (Gate) ของตัวชิปนั้นไม่เพียงพอ ไม่สามารถรองรับโปรแกรมที่เขียนขึ้นทั้งหมดได้ จึงได้ลองพยายามเปลี่ยนวิธีในการทดลอง เช่น เปลี่ยนตัวคูณ; จากที่ใช้ทฤษฎีเลขคณิตกระจาย (DA : Distributed Arithmetic) มาใช้ CSD (Canonic Signed Digit) และเปลี่ยนอีกครั้งมาเป็น บูธ อัลกอริธึม (Booth Algorithm) นอกจากนี้ยังได้ทำการเปลี่ยนเทคนิคการเขียนโปรแกรม เช่น จากโครงสร้างคอร์ดิกแบบอันโรลด์ (Unrolled cordic structure) มาเป็นแบบวนรอบการทำงาน (Iterative cordic structure) เพื่อที่ว่าจะสามารถทำการโปรแกรม โปรแกรมทั้งหมดลงบนชิปได้ แต่ก็ไม่เป็นผล ทำให้ไม่สามารถทำการทดลองตามวงจรที่ได้ออกแบบไว้แล้วได้

การที่การทดลองนี้ใช้เรสโคไซน์ฟิลเตอร์ซึ่งมีความยาว 36 แทป (tap) และใช้ทฤษฎีโพลีเฟสเข้ามาช่วยในการออกแบบ จะทำให้มีการคูณค่าสัมประสิทธิ์จำนวน 6 ตัวกับอินพุท และในส่วนของการสร้างได้ใช้หลักการทรานสโพสไดเรคท์ฟอร์ม(Transpose Direct Form) ทำให้ต้องใช้ตัวคูณ 6 ตัวพร้อมๆกัน เพื่อให้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาในการประมวลผลน้อยลง แต่ก็จะทำให้ต้องใช้เกทของอุปกรณ์ FPGA มาก และการทดลองนี้ซึ่งประสบปัญหาเรื่องเกทไม่พอก็สามารถแก้ได้โดยการใช้ตัวคูณตัวเดียว โดยทำการคูณค่าสัมประสิทธิ์กับอินพุทแล้วเก็บไว้ จากนั้นการคูณครั้งต่อไปนำผลคูณที่ได้ไปบวกกับค่าที่เก็บไว้ในตอนแรก ทำเช่นนี้ไปเรื่อยๆจนถึงการคูณของสัมประสิทธิ์สุดท้ายกับอินพุท การทำเช่นนี้จะทำให้ลดพื้นที่เกทได้แต่เวลาที่ใช้ในการประมวลผลก็จะมากขึ้น และความเร็ว (speed) ของวงจรจะลดลง จึงอาจต้องลดบิทเรทลง

ส่วน CORDIC Algorithm ที่ได้ออกแบบให้มีการวนรอบการทำงาน 11 รอบนั้น เป็นโปรแกรมที่ดีอยู่แล้ว เวลาทำงานจริงอาจจะเปลี่ยนตัวคริสตัลออสซิเลเตอร์ให้มีความถี่สูงขึ้น เพื่อให้รับกับค่าความถี่สูงสุด (maximum frequency) ของโปรแกรมที่ออกแบบไว้ ซึ่งค่าของความถี่สัญญาณนาฬิกาสามารถดูได้หลังจากที่ทำการ compile และ generate แล้วจะมีค่านี้บอกไว้ในโปรแกรม Xilinx ISE

และเมื่อทำแต่ละส่วนเสร็จแล้ว เวลาที่นำมาต่อทำงานร่วมกันต้องควบคุมให้ดี เพราะไม่เช่นนั้นการทำงานจะไม่สอดรับกัน ซึ่งเป็นอีกส่วนหนึ่งที่ทำยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ชำนาญ ปัญญาใส, วัชรากร หนูทอง, "ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล", ซีเอ็ดยูเคชั่น, กรุงเทพฯ, 2547

[2] พรชัย ภววงษ์ศักดิ์, "การประมวลผลสัญญาณดิจิตอลเบื้องต้น", มหาวิทยาลัยเทคโนโลยีมหานคร http://www.ee.mut.ac.th/home/pornchai, 2543

[3] มนัส สังวรศิลปี, วรรัตน์ ภัทรอมรกุล, "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์", สำนักพิมพ์อินโฟเพรส, กรุงเทพฯ, 2543

[4] วัลลภ สุระกำพลธร, "การประมวลผลสัญญาณเชิงเลข", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2533

[5] สมศักดิ์ ชุมช่วย, "การประมวลสัญญาณเชิงเลขเบื้องต้น", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2545

[6] Ifeachor E.C. and Jervis B.W., "Digital Signal Processing: A Practical Approach", Addison-Wesley Inc., 1993

[7] Oppenheim A.V. and Schafer R.W., "Discrete-time Signal Processing", Prentice Hall Inc., 1989

[8] Oppenheim A.V., Willsky A.S. and Nawab H.S., "Signal and System". 2nd edition, Prentice Hall Inc., 1997

[9] Sanjit K. Mitra., "Digital Signal Processing: A Computer Approach". 2nd edition, McGraw Hill Inc., 2002

[10] John Wiley, "Multirate Digital Signal Processing"

[11] Ray Andraka, "A Survey of CORDIC algorithms for FPGA based computers"

[12] A. Shoari, M. Kamarei and A. Radmand, "Implementation of Costas Loop using CORDIC algorithm for software radio applications"

[13] Jouko Vankka, Marko Kosunen, Jan Hubach and Kari Halonen, "A CORDIC-Based Multicarrier QAM Modulation"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้