สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบวงจรแบ่งความถี่แบบเฟสเป็นเชิงเส้นสมบูรณ์ด้วยวงจรกรองความถี่ดิจิตอลไอ ไอ อาร์ และสร้างบน เอฟ พี จี เอ

ON THE DESIGN OF PERFECT LINEAR PHASE LOUD SPEAKER CROSSOVER USING DIGITAL IIR FILTER AND IT REAL-TIME IMPLEMENTATION BASE ON FPGAs

T067440

เจตน์ ออสวัสดิ์

JEDT ORSAWAT

วท.
จ696ก
2549

เลขหมู่......
เลขทะเบียน 67440
วัน,เดือน,ปี 15 S.A. 2549

.b 11671051
.i

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2772-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ON THE DESIGN OF PERFECT LINEAR PHASE LOUD SPEAKER CROSSOVER USING DIGITAL IIR FILTER AND IT REAL-TIME IMPLEMENTATION BASE ON FPGAs

JEDT ORSAWAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2772-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การสร้างครอสโอเวอร์เน็ตเวอร์คแบบสองทางเฟสเป็นเชิงเส้นสมบูรณ์ด้วยวงจรกรองความถี่ดิจิตอลไอ ไอ อาร์ บน เอฟ พี จี เอ
นักศึกษา	นายเจตน์ ออสวัสดิ์
รหัสนักศึกษา	45061106
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.กนก เจนจิระพงศ์เวช

บทคัดย่อ

วิทยานิพนธ์นี้เป็นการเสนอการออกแบบและการสร้างวงจรดิจิตอลครอสโอเวอร์เน็ตเวอร์คแบบสองทางสำหรับย่านความถี่เสียงบนชิป FPGA (Field Programmable Gate Arrays) โดยในวิทยานิพนธ์นี้นำเสนอการสร้างโดยใช้โครงสร้างของวงจรกรองความถี่แบบ ไอ ไอ อาร์ ที่มีเฟสเป็นเชิงเส้นอย่างสมบูรณ์ ซึ่งโครงสร้างดังกล่าวจะมีข้อดีคือ มีผลตอบสนองทางเฟสต่อความถี่เป็นเชิงเส้นและผลตอบสนองกรุ๊ปดีทำให้มีลักษณะราบเรียบและมีความชันของสโลปในช่วงทรานซิชั่นแบนด์ที่สูงและยิ่งไปกว่านั้น ผลรวมทางขนาดต่อความถี่ของวงจรความถี่ที่นำเสนอมีค่าคงที่ตลอดย่านความถี่ซึ่งตรงกับคุณสมบัติของ all-pass function ของครอสโอเวอร์เน็ตเวอร์ค ซึ่งเป็นข้อพิจารณาหลักและเป็นประเด็นอันหนึ่งที่สำคัญมากในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์ค

ในการสร้างวงจรดังกล่าวบน FPGA นั้นได้นำเสนอการออกแบบวงจรกลับสัญญาณในเวลาจริง (Real-time time reversal) ขนาดความยาว L sample แบบ real-time (Last In First Out : LIFO) และภาคหน่วงเวลา (Delay) ซึ่งใช้หน่วยความจำมาออกแบบแทนชิปรีจีสเตอร์ซึ่งทำให้มีพื้นที่ในการออกแบบลดลง และในส่วนของวงจรกรองความถี่ IIR นั้น ใช้วิธีการประมวลผลแบบ DA (Distributed Arithmetic structure) ซึ่งการประมวลผลด้วยวิธีการดังกล่าวนี้จะเป็นวิธีที่หลีกเลี่ยงที่จะใช้การคูณหรือกระบวนการคูณโดยตรง แต่จะนำเอาวิธีการเปิดตารางแทน (Look-up table) ซึ่งจะมีข้อดีคือทำให้ได้ความเร็วในการประมวลผลที่สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	An implement 2 ways perfect linear phase crossover design using digital IIR filters base on FPGAs
Student	Mr. Jedt Orsawat
Student ID.	45061106
Degree	Master of Engineering
Program	Information Engineering
Year	2006
Thesis Advisor	Assoc.Prof.Dr. Kanok Janjitrapongvej

ABSTRACT

This thesis presents an implementation of two-ways digital crossover network for audio frequencies band on FPGAs using perfect linear phase IIR filters structure base on two-pass filtering of the real-time noncausal and causal IIR subfilters. This structure has a new characteristics such as linear phase, flat group delay, high transition band and flat magnitude response over audio frequencies, these are require and the important properties in designing a novel crossover network. In order to design and implemented the circuits on FPGAs, this thesis proposes the method of designing Last In First Out (LIFO) part and Delay part by using memory instead of a shift register which the designing area is minimize. For the filter circuits, the Distributed Arithmetic (DA) structure is used for signal processing. The structure use look-up table (LUT) instead of multiplication which can reduce the processing time of the system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....IX

สารบัญรูป.....X

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 ขอบเขตการวิจัย.....2

1.4 ขั้นตอนของการวิจัย.....2

1.5 รายละเอียดวิทยานิพนธ์.....3

บทที่ 2 การประมวลผลสัญญาณ.....4

2.1 บทนำ.....4

2.2 ข้อดีและข้อเสียของการประมวลผลเชิงเลข.....4

2.3 โครงสร้างของระบบประมวลสัญญาณเชิงเลข.....6

2.3.1 ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล.....6

2.3.2 ตัวประมวลผลสัญญาณเชิงเลข.....6

2.3.3 ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก.....6

2.4 โอโปรเรชั่นของลำดับสัญญาณ.....7

2.5 ทฤษฎีการสุ่มตัวอย่าง.....8

2.6 พื้นฐานวงจรกรองอนาลอกต้นแบบ.....11

2.6.1 ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอก.....12

2.6.2 วงจรกรองอนาลอกแบบต่าง ๆ.....12

2.6.2.1 วงจรกรองอนาลอกต้นแบบ บัทเตอร์เวิร์ธ.....13

2.6.2.2 วงจรกรองอนาลอกต้นแบบ เชบีเชฟ.....14

2.6.2.3 วงจรกรองอนาลอกต้นแบบ อิลิปติค.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.5 วงจรกลับสัญญาณในเวลาจริง.....73

5.5.1 การออกแบบวงจรกลับสัญญาณโดยใช้โครงสร้างของหน่วยความจำ....75

5.6 การออกแบบวงจรหน่วงสัญญาณ.....77

5.7 วงจรครอสโอเวอร์เน็ตเวอร์คจากโครงสร้างคณิตศาสตร์การกระจาย.....78

5.7.1 การสร้างไฟล์ตารางเปิดดู.....79

5.7.2 การออกแบบชิปรีจีสเตอร์แบบเข้าขนานออกอนุกรมขนาด 16 บิต.....80

5.7.3 การออกแบบชิปรีจีสเตอร์แบบเข้าอนุกรมออกอนุกรมขนาด 16 บิต.....81

5.7.4 การออกแบบแอกคิวมูเลเตอร์ขนาด 16 บิตแบบคิดเครื่องหมาย.....82

5.7.5 การออกแบบวงจรควบคุมจังหวะการทำงาน.....84

5.8 ผลการสังเคราะห์วงจร.....86

5.9 ผลการทดลอง.....87

5.9.1 การทดสอบโมดูลกลับสัญญาณในเวลาจริง.....87

5.9.2 การทดสอบโมดูลหน่วงสัญญาณ.....89

5.9.3 การทดสอบโมดูลสวิทช์มัลติเพล็กซ์ส่วนที่ 1.....90

5.9.4 การทดสอบโมดูลสวิทช์มัลติเพล็กซ์ส่วนที่ 2.....91

5.9.5 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์ค.....92

5.10 บทสรุป.....95

บทที่ 6 บทสรุปและข้อเสนอแนะ.....96

6.1 สรุปผลการวิจัย.....96

6.2 ปัญหาและอุปสรรคที่พบในงานวิจัย.....97

6.3 ข้อเสนอแนะในการพัฒนา.....98

บรรณานุกรม.....99

ภาคผนวก ก. โปรแกรมออกแบบวงจรครอสโอเวอร์เน็ตเวอร์ค.....102

ภาคผนวก ข. โปรแกรมสร้างไฟล์ตารางเปิดดู.....103

ภาคผนวก ค. โปรแกรมออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA.....105

ผลงานวิจัยที่ได้รับการตีพิมพ์.....109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

ประวัติผู้เขียน..135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ข้อดีข้อเสียของการประมวลผลเชิงเลข..5
2.2 การกระทำต่อลำดับสัญญาณในระบบการประมวลผลเชิงเลข..7
2.3 ข้อดีและข้อเสียของวงจรกรองความถี่แต่ละแบบ..17
2.4 ค่าของ $F_1\{.\}$ ภายใน ROM..24
3.1 จำนวนเกตและหน่วยความจำ ต่าง ๆ ภายในชิป Spartan III แต่ละเบอร์.............................36
3.2 จำนวนเกตและหน่วยความจำ ต่าง ๆ ภายในชิป Spartan III แต่ละเบอร์............................. 36
3.3 หน่วยความจำแบบพอร์ตเดี่ยวที่สร้างจากบล็อกหน่วยความจำภายใน FPGA.........................39
3.4 หมายเลขโค๊ดบนชิพ FPGA...42
5.1 ค่าสัมประสิทธิ์ของตารางเปิดดูจากฟังก์ชันถ่ายโอนจากสมการที่ (5.2)..................................67
5.2 ค่าสัมประสิทธิ์ของตารางเปิดดูจากฟังก์ชันถ่ายโอนจากสมการที่ (5.3)..................................67
5.3 พื้นที่ในการออกแบบตารางเปิดดูของวงจรกรองที่ออกแบบด้วยโครงสร้างแบบต่าง ๆ..........69
5.4 ผลการสังเคราะห์แต่ละโมดูลของวงจรครอสโอเวอร์เน็ตเวอร์ค...86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 การประมวลผลสัญญาณแบบอนาลอก และการประมวลผลเชิงเลข....4
2.2 บล็อกไดอะแกรมแสดงโครงสร้างระบบ DSP ที่กระทำกับสัญญาณอนาลอก....6
2.3 การสุ่มสัญญาณ (Sampling Signal)....8
2.4 สเปกตรัมของสัญญาณจากการสุ่มตัวอย่าง....10
2.5 ชนิดของวงจรกรองความถี่แบบต่างๆ....11
2.6 คุณลักษณะผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำ....13
2.7 โครงสร้างของวงจรกรองแบบ FIR....16
2.8 โครงสร้างของวงจรกรองความถี่แบบ IIR....17
2.9 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ขนาด 4 บิต....18
2.10 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ทั้ง 4 ชุด....19
2.11 ผลจากการรวมผลคูณย่อยจาก Serial Distributed Arithmetic แทน MAC ทั้ง 4 ชุด....20
2.12 โครงสร้างของ SDA-MAC ที่สร้างจากตารางดูค่า....20
2.13 ค่าของ LUT ที่เกิดจากค่าสัมประสิทธิ์ทั้ง 4 ค่า....21
2.14 โครงสร้างของ Distributed Arithmetic วงจรกรองความถี่แบบ IIR อันดับ 3....25
3.1 ขั้นตอนการออกแบบวงจรด้วยภาษา HDL....30
3.2 กระบวนการออกแบบวงจร (Design Methodology)....32
3.3 ไดอะแกรมโครงสร้างของ Register Transfer Level (RTL)....33
3.4 โครงสร้างของบล็อกอินพุตเอาต์พุต....37
3.5 โครงสร้างของ Slice....38
3.6 โครงสร้างหน่วยความจำพอร์ตเดี่ยวบน FPGA....39
3.7 โครงสร้างของวงจรคูณขนาด 18 บิต....40
3.8 โครงสร้างของ DCM....40
3.9 รหัสโค้ดบนชิป FPGA....41
3.10 บล็อกหน่วยความจำ 18 กิโลบิตและบล็อกวงจรคูณขนาด 18 x 18 บิต....35
4.1 โครงสร้างของวงจรครอสโอเวอร์เน็ตเวอร์ค....44
4.2 ตำแหน่งของโพลและซีโรของวงจรกรองความถี่ต่ำต้นแบบ....46
4.3 ผลการตอบสนองทางขนาดต่อความถี่....46
4.4 ตำแหน่งของโพลและซีโรของฟังก์ชันถ่ายโอน $A_0(Z)$....48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.5 การตอบสนองทางขนาดต่อความถี่และเฟสของของฟังก์ชันถ่ายโอน $A_0(Z)$ 48

4.6 ตำแหน่งของโพลและซีโรของฟังก์ชันถ่ายโอน $A_1(Z)$ 49

4.7 การตอบสนองทางขนาดต่อความถี่และเฟสของของฟังก์ชันถ่ายโอน $A_1(Z)$ 49

4.8 ตำแหน่งของโพลและซีโร ของ $H_{LP}(Z)$ และ $H_{HP}(Z)$ 50

4.9 ผลตอบสนองของขนาดต่อความถี่ของวงจรครอสโอเวอร์ที่นำเสนอ....................................51

4.10 รายละเอียดผลรวมการตอบสนองของขนาดต่อความถี่เมื่อทำการขยาย.................... 51

4.11 ผลการตอบสนองทางขนาดและเฟสต่อความถี่ของวงจรกรองความถี่ต่ำ 52

4.12 ผลตอบสนองของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ..52

4.13 ผลการตอบสนองทางขนาดและเฟสต่อความถี่ของวงจรกรองความถี่สูง............................53

4.14 ผลตอบสนองของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ..53

4.15 โครงสร้างวงจร Linear-phase IIR Filter..54

4.16 การจำกัดความยาวของวงจรกรองแบบ IIR โดยใช้วงจรกรองเรซิดิวซ์.............................56

4.17 บล็อกไดอะแกรมของวงจรจำกัดความยาวของตัวกรอง IIR...56

4.18 การนำเอาวงจรกรองเรซิดิวซ์มาเพิ่มให้กับวงจรครอสโอเวอร์เน็ตเวอร์ค.............................57

4.19 วงจรครอสโอเวอร์เน็ตเวอร์คที่ใช้โครงสร้างจากวงจร กรองทุกย่านความถี่ผ่าน....................57

4.20 ตำแหน่งของโพลและซีโรทั้งหมดของระบบ..58

4.21 ผลตอบสนองขนาดต่อความถี่ต้นแบบและระบบที่นำเสนอ 59

4.22 ผลรวมทางขนาดต่อความถี่ตลอดย่านความถี่..59

4.23 รายละเอียดเฉพาะริบเปิ้ลของผลตอบสนองของขนาดในช่วงความถี่ผ่าน..........................60

4.24 ผลรวมของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ..60

4.25 ผลรวมของกรุ๊ปดีเลย์ของวงจรกรองความถี่สูง..61

4.26 ผลรวมของเฟสต่อความถี่ของวงจรกรองความถี่ต่ำ..61

4.27 ผลรวมของเฟสต่อความถี่ของวงจรกรองความถี่สูง..62

4.28 ผลการตอบสนองทางขนาดต่อความถี่เมื่อเทียบกับงานวิจัยของ Linkwitz...........................62

4.29 ผลการตอบสนองทางขนาดต่อความถี่ เมื่อเทียบกับงานวิจัยของ Gacia..............................63

5.1 ขั้นตอนการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA 64

5.2 ผลตอบสนองความถี่จากสมการที่ (5.4) และ (5.5)..66

5.3 ตำแหน่งโพลและซีโรของฟังก์ชันถ่ายโอนจากสมการที่ (5.4) และ (5.5)...........................66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.4 โครงสร้างของวงจรครอสโอเวอร์ที่สร้างจากวงจรกรองทุกย่านความถี่ผ่านต่อขนานกัน.......67
5.5 วงจรครอสโอเวอร์เน็ตเวอร์คที่ให้เฟสเชิงเส้นสมบูรณ์..69
5.6 โครงสร้างวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางที่นำไปสร้างบน FPGA........................70
5.7 สัญลักษณ์และโครงสร้างของวงจรสวิทช์ส่วนที่ 1..71
5.8 ผลการจำลองการทำงานวงจรสวิทช์ส่วนที่ 1...71
5.9 สัญลักษณ์และโครงสร้างของวงจรสวิทช์ส่วนที่ 2...72
5.10 ผลการจำลองการทำงานวงจรสวิทช์ส่วนที่ 2...73
5.11 โครงสร้างวงจรกรองแบบไม่เป็นคอซอลที่สามารถสร้างได้จริง...73
5.12 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=0 ถึง 4..74
5.13 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=5 ถึง 9..74
5.14 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=10 ถึง 14..74
5.15 วงจรกลับสัญญาณในเวลาจริงที่สร้างจากวงจรชิปรีจีสเตอร์...75
5.16 วงจรกลับสัญญาณที่ออกแบบด้วยหน่วยความจำ...75
5.17 การอ่านการเขียนข้อมูลในบัฟเฟอร์...76
5.18 ผลการจำลองการทำงาน โมดูล LIFO...76
5.19 โครงสร้างของวงจรหน่วงสัญญาณ...77
5.20 โครงสร้างของวงจรหน่วงสัญญาณที่สร้างจากหน่วยความจำ..77
5.21 ผลการจำลองการทำงานของโมดูลหน่วงสัญญาณ..78
5.22 วงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางจากโครงสร้างของคณิตศาสตร์การกระจาย.........78
5.23 สัญลักษณ์ของโมดูลเปิดตารางจากตารางที่ 5.1 และ 5.2...80
5.24 โครงสร้างของโมดูลชิปรีจีสเตอร์แบบเข้าขนานออกอนุกรมขนาด 16 บิต...........................81
5.25 ผลการจำลองการทำงานของโมดูลชิปรีจีสเตอร์แบบเข้าขนานออกอนุกรม.........................81
5.26 โครงสร้างของโมดูลชิปรีจีสเตอร์แบบเข้าอนุกรมออกอนุกรมขนาด 16 บิต.........................81
5.27 ผลการจำลองการทำงานของโมดูลชิปรีจีสเตอร์แบบเข้าอนุกรมออกอนุกรม........................82
5.28 โครงสร้างของโมดูลแอกคิวมูเลเตอร์...83
5.29 ผลการจำลองการทำงานของโมดูลแอกคิวมูเลเตอร์...83
5.30 โครงสร้างของโมดูลควบคุมจังหวะการทำงาน...84
5.31 ผลการจำลองการทำงานของของโมดูลวงจรควบคุมจังหวะการทำงาน.................................85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.32 วงจรครอสโอเวอร์แบบสองทางจากวงจรกรองทุกความถี่ต่อขนานกันบน FPGA..................85

5.33 ผลการตอบสนองอิมพัลซ์ของวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางบน FPGA.............86

5.34 การเชื่อมต่อ FPGA กับอุปกรณ์ภายนอกเพื่อใช้ในการทดสอบ..87

5.35 การทดสอบการทำงานของโมดูลกลับสัญญาณในเวลาจริง..88

5.36 สัญญาณ saw tooth ความถี่ต่ำที่ผ่านวงจรกลับสัญญาณ..88

5.37 สัญญาณ saw tooth ความถี่สูงที่ผ่านวงจรกลับสัญญาณ..88

5.38 สัญญาณ sine wave ความถี่ต่ำที่ผ่านวงจรกลับสัญญาณ...89

5.39 สัญญาณ sine wave ความถี่สูงที่ผ่านวงจรกลับสัญญาณ...89

5.40 การทดสอบการทำงานของโมดูลหน่วงสัญญาณ...90

5.41 สัญญาณที่ผ่านโมดูลหน่วงสัญญาณ 300 แซมเปิ้ล..90

5.42 การทดสอบการทำงานของโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 1..91

5.43 สัญญาณที่ได้จากการการทดสอบโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 1......................................91

5.44 การทดสอบการทำงานของโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 2..92

5.45 ที่ได้จากการการทดสอบโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 2..92

5.46 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์ค...93

5.47 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 500 Hz..93

5.48 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 1 KHz...93

5.49 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 2 KHz...94

5.50 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 10 KHz...94

5.51 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 200 Hz รวมกับ 10 KHz.........................95

5.52 ต้นแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่ใช้ในวิทยานิพนธ์..95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1ความเป็นมาและความสำคัญของปัญหา

ในระบบเล่นสัญญาณเสียงกลับคืน (reproduction of audio signal) สำหรับระบบเสียงไฮไฟ จะเป็นการสร้างสัญญาณไฟฟ้าที่ถูกบันทึกเก็บไว้ในรูปแบบต่าง ๆ ทั้งในสื่อที่ใช้สัญญาณต่อเนื่อง เช่น tape deck, turn table การกระจายเสียงระบบ FM stereo และในสื่อที่ใช้สัญญาณดิจิตอลทั้งที่มีการบีบอัดและไม่บีบอัดสัญญาณ เช่น DAT (Digital audio tape) CD (compact disc) Laser disc และ SACD (super audio compact disc) เป็นต้น และเมื่อเราได้สัญญาณเสียงกลับคืนมาแล้วจะต้องมีการขยายสัญญาณดังกล่าวและส่งต่อไปยังระบบขับสัญญาณเสียง ซึ่งในการขยายสัญญาณย่านความถี่เสียง (Audio band) ในช่วงความถี่ 20 Hz - 20 KHz นั้นภาคขยายสัญญาณย่านความถี่เสียงสามารถปรับปรุงให้มีผลตอบสนองความถี่ให้มีขนาดของอัตราการขยายคงที่ตลอดย่านความถี่ได้ แต่ในระบบไฮไฟ (Hi-Fidelity) เมื่อสัญญาณสเตอริโอ (หรือระบบกระจายเสียงรอบทิศทาง) ส่งสัญญาณมายังตัวขับสัญญาณเสียง (speaker system) โดยพื้นฐานลำโพงหนึ่งตัวที่ทำหน้าที่แปลงสัญญาณไฟฟ้าเป็นสัญญาณเสียงนั้นมีผลตอบสนองความถี่ที่จำกัดเนื่องจากโครงสร้างจะไม่สามารถตอบสนองความถี่ได้ตลอดย่านความถี่เสียง ดังนั้นในการแก้ปัญหาดังกล่าวจึงจำเป็นต้องใช้ลำโพงมากกว่า 1 ตัว (หลาย ๆ ตัว) มาต่อรวมกัน เช่น ระบบขับสัญญาณเสียงแบบ 2 ทางจะประกอบด้วยตัวขับ สัญญาณเสียงความถี่ต่ำ (เสียงทุ้ม) เรียกว่าวูฟเฟอร์ และตัวขับสัญญาณเสียงแหลม (เสียงสูง) เรียกว่า ทวีทเตอร์ ซึ่งในการที่จะให้ลำโพงแต่ละตัวทำงานได้อย่างมีประสิทธิภาพดีนั้น จำเป็นต้องมีการแบ่งความถี่เพื่อแยกสัญญาณออกเป็นช่วง ๆ เพื่อป้อนเข้าลำโพงแต่ละตัว ซึ่งเราเรียกวงจรนี้ว่า วงจรครอสโอเวอร์เน็ตเวอร์ค

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้เป็นการเสนอการออกแบบและการสร้างวงจรดิจิตอลครอสโอเวอร์เน็ตเวอร์คแบบสองทางสำหรับย่านความถี่เสียงบนชิป FPGA (Field Programmable Gate Arrays) โดยทำการสร้างโดยใช้โครงสร้างของวงจรกรองความถี่แบบ IIR (Infinite Impulse Response) ที่มีเฟสเป็นเชิงเส้นอย่างสมบูรณ์ ซึ่งโครงสร้างดังกล่าวจะมีข้อดีคือ มีผลตอบสนองทางเฟสต่อความถี่เป็นเชิงเส้น ผลตอบสนองกรุ๊ปดีเลย์มีลักษณะราบเรียบและมีความชันในช่วงทรานซิชั่นแบนด์ที่สูง และยิ่งไปกว่านั้น ผลรวมทางขนาดต่อความถี่ของวงจรความถี่ที่นำเสนอมีค่าคงที่ตลอดย่านความถี่ซึ่งตรงกับคุณสมบัติของ all-pass function ของครอสโอเวอร์เน็ตเวอร์ค ซึ่งเป็นข้อพิจารณาหลักและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นประเด็นอันหนึ่งที่สำคัญมากในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์ค ซึ่งวัตถุประสงค์ของการวิจัยมีดังนี้

1. เพื่อศึกษาการทำงานของดิจิตอลครอสโอเวอร์เน็ตเวอร์ค
2. เพื่อพัฒนาสร้างดิจิตอลครอสโอเวอร์เน็ตเวอร์คให้มีโครงสร้างขนาดเล็ก ซึ่งทำให้ใช้พื้นที่ในการสร้างน้อย ทำให้การสร้างสามารถทำได้ง่ายและสามารถผลิตอุปกรณ์นี้ได้ในราคาถูก
3. พัฒนาสร้างดิจิตอลครอสโอเวอร์เน็ตเวอร์ค นี้ให้เป็นต้นแบบสำหรับงานที่เกี่ยวข้องกับระบบเสียง
4. สามารถนำวงจรดิจิตอลฟิลเตอร์นี้ไปประยุกต์ใช้งานในระบบงานประเภทอื่น ๆ ได้

1.3 ขอบเขตการวิจัย

วิทยานิพนธ์นี้เป็นการเสนอการออกแบบและการสร้างวงจรดิจิตอลครอสโอเวอร์เน็ตเวอร์คแบบสองทางสำหรับย่านความถี่เสียงบนชิป FPGA (Field Programmable Gate Arrays) โดยใช้โครงสร้างของวงจรกรองความถี่แบบ IIR ที่มีเฟสเป็นเชิงเส้นอย่างสมบูรณ์ ซึ่งในการสร้างวงจรดังกล่าวบน FPGA นั้นได้นำเสนอการออกแบบวงจรกลับสัญญาณในเวลาจริง (Real-time time reversal) ขนาดความยาว L sample และภาคหน่วงเวลา (Delay) ซึ่งใช้หน่วยความจำมาออกแบบแทนชิปรีจีสเตอร์ซึ่งทำให้มีพื้นที่ในการออกแบบลดลง และในส่วนของวงจรกรองความถี่ IIR นั้น ใช้วิธีการประมวลผลแบบ DA (Distributed Arithmetic structure) ซึ่งการประมวลผลด้วยวิธีการดังกล่าวนี้จะเป็นวิธีที่หลีกเลี่ยงวิธีการคูณโดยตรง แต่จะนำเอาวิธีการเปิดตารางแทน (Look-up table) ซึ่งจะมีข้อดีคือทำให้ได้ความเร็วในการประมวลผลที่สูงขึ้น

1.4 ขั้นตอนของการวิจัย

ในการวิจัยออกแบบสร้างวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางสำหรับย่านความถี่เสียงที่ให้ผลรวมทางขนาดต่อความถี่ราบเรียบตลอดย่านความถี่โดยมีขั้นตอนต่าง ๆ ของการวิจัยกำหนดไว้ตามลำดับดังนี้

1. ศึกษางานวิจัยที่มีลักษณะงานที่ใกล้เคียงกับวิทยานิพนธ์นี้ทั้งวิธีการออกแบบวงจรกรองและวิธีการที่จะสร้างจริง
2. ทดลองออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คด้วยโปรแกรม MATLAB เพื่อศึกษาถึงลักษณะโครงสร้างที่จะนำไปสร้างจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. นำผลที่ได้จากพัฒนามาออกแบบด้วยภาษาอธิบายฮาร์ดแวร์ และตรวจสอบความถูกต้อง โดยทำการจำลองการทำงานในระดับพฤติกรรม (Behavior Simulations)
4. สังเคราะห์ (Synthesis) เป็นวงจรระดับเกต และตรวจสอบความถูกต้องโดยทำการจำลองการทำงานในระดับฟังก์ชัน (Functional Simulation)
5. วางอุปกรณ์และการเชื่อมต่อสัญญาณ และตรวจสอบความถูกต้องโดยทำการจำลองการทำงานระดับฐานเวลา (Timing Simulation)
6. นำซอร์ฟแวร์ ที่ได้จากขั้นตอนที่แล้วโปรแกรมลงบน FPGA และตรวจสอบความถูกต้องในวงจร (Circuit testing) และทดสอบประสิทธิภาพ

1.5 รายละเอียดวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้แบ่งเนื้อหาออกเป็นทั้งหมด 6 บท โดยบทที่ 1 เป็นการกล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์การทำวิจัย ขอบเขตการวิจัย และขั้นตอนของการวิจัย

บทที่ 2 กล่าวถึง พื้นฐานการประมวลผลสัญญาณ โครงสร้างของตัวประมวลผลสัญญาณเชิงเลข ตัวแปรที่ใช้ในการแปลงระบบอนาลอกไปเป็นระบบเชิงเลข เงื่อนไขสำคัญต่าง ๆ ที่ใช้ในการพิจารณาคุณสมบัติของระบบเชิงเลข และหลักการที่ใช้ในการสร้างวงจรกรองเชิงเลขสำหรับงานวิจัยนี้

บทที่ 3 กล่าวถึงเทคโนโลยีการออกแบบวงจรรวมแบบดิจิตอล โครงสร้างพื้นฐานของอุปกรณ์เอฟพีจีเอ สถาปัตยกรรมเอฟพีจีเอ Xilinx Spatan III ตลอดจนเทคนิควิธีการการออกแบบและติดตั้งวงจรดิจิตอลบนเอฟพีจีเอ ประวัติความเป็นมาและโครงสร้างของภาษาเวริลอก

บทที่ 4 กล่าวถึงคุณสมบัติของวงจรครอสโอเวอร์เน็ตเวอร์ค โครงสร้างของวงจรกรองความถี่แบบ all-pass วงจรครอสโอเวอร์เน็ตเวอร์คจากวงจรกรองทุกความถี่ผ่านต่อขนาน 2 วงจร วงจรกรองความถี่แบบ IIR ที่ให้ผลการตอบสนองทางเฟสเป็นเชิงเส้น

บทที่ 5 กล่าวถึงวิธีการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คและทดสอบการทำงาน ผลการใช้อุปกรณ์ภายในเอฟพีจีเอ ความเร็วการทำงานและความถี่สูงสุดที่สามารถทำงานได้ และผลการทดสอบการจำลองการทำงาน ผลการทดสอบชุดถอดรหัสในวงจร (Circuit testing) ใช้งานจริงและการวิเคราะห์ผลการทดสอบ

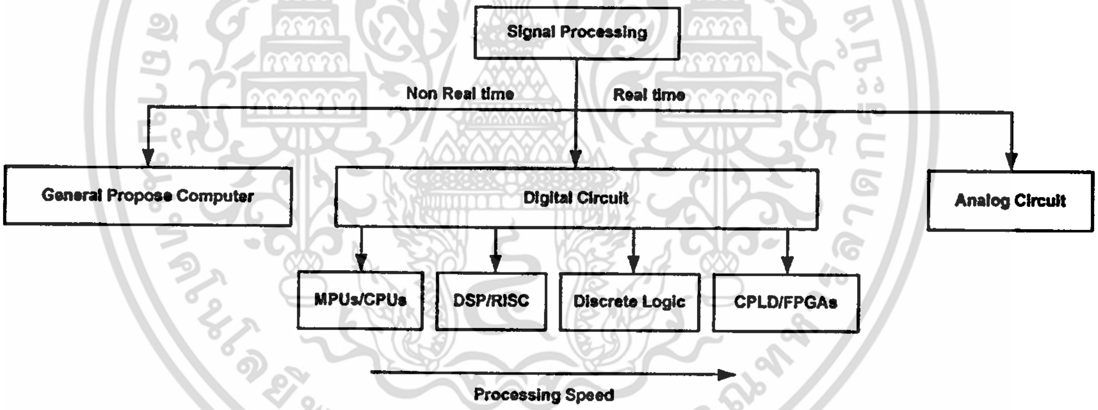
ส่วนบทสุดท้ายเป็นการสรุปผลการวิจัย ปัญหาและอุปสรรคในการวิจัย และข้อเสนอแนะในการพัฒนาสร้างดิจิตอลครอสโอเวอร์เน็ตเวอร์คต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การประมวลผลสัญญาณ (Signal Processing)

2.1 บทนำ

การประมวลผลสัญญาณ หมายถึง การกระทำบางอย่างกับสัญญาณที่มีผลให้เกิด การปรับปรุงหรือมีการเปลี่ยนแปลงรูปร่างไปด้วย กระบวนการขยาย (Amplification) การลดทอน (Attenuation) และการกรอง (Filtering) เป็นต้น โดยผ่านการควบคุมตัวแปรของระบบ (system) ที่อยู่ในรูปของฟังก์ชันถ่ายโอน (transfer function) โดยอาศัยพื้นฐานของการรวม การคูณและการหน่วงสัญญาณ โดยสามารถแบ่งการประมวลผลออกเป็น 2 ประเภทได้แก่ การประมวลผลสัญญาณอนาลอก (Analog Signal Processing) และการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing) ดังแสดงไว้ในรูปที่ 2.1



รูปที่ 2.1 การประมวลผลสัญญาณแบบอนาลอก และการประมวลผลเชิงเลข

2.2 ข้อดีและข้อเสียของการประมวลผลเชิงเลข

รูปแบบการประมวลผลสัญญาณอนาลอกและการประมวลผลสัญญาณเชิงเลขต่างก็มีคุณสมบัติเฉพาะที่แตกต่างกันออกไป ซึ่งสามารถเลือกพิจารณาการใช้งานระบบใด ๆ ได้ตามความเหมาะสม ซึ่งสามารถแบ่งข้อดี ข้อเสียของการประมวลผลเชิงเลขได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

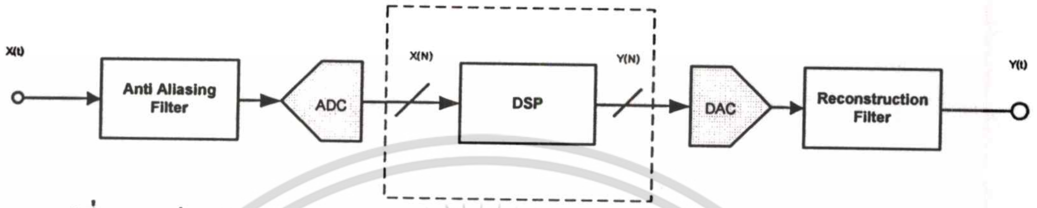
ตารางที่ 2.1 ข้อดีข้อเสียของการประมวลผลเชิงเลข

ข้อดีของการประมวลผลเชิงเลข	ข้อเสียของการประมวลผลเชิงเลข
1. การประมวลผลเหมาะสมกับข้อมูลที่อยู่ในรูปแบบเชิงเลขเหมือนกันเช่น ภาพดิจิตอล 2. อุปกรณ์ทางด้านดิจิตอลมีแนวโน้มราคาต่ำลง ขนาดเล็กลง ประสิทธิภาพและความเร็วที่สูงขึ้น ความเที่ยงตรงเพิ่มมากขึ้น 3. สามารถทำการรับส่งข้อมูลได้แม่นยำกว่าสัญญาณอนาลอก เนื่องจากสัญญาณเชิงเลขมีสถานะเพียงศูนย์ “0” และหนึ่ง “1” เท่านั้น 4. การประมวลผลเชิงเลขสามารถทำได้โดยง่ายเพราะอัลกอริทึม (Algorithm) มีเพียงการบวก การลบ การคูณ และการหน่วงเวลา 5. ในการประมวลผลเชิงเลขสามารถทำได้พร้อม ๆ กันหลายช่องสัญญาณในลักษณะการแบ่งช่วงเวลาทำงาน (Time Sharing) 6. มีเสถียรภาพที่ดีต่อการเปลี่ยนแปลงของอุณหภูมิ หรือแม้กระทั่งการเปลี่ยนแปลงพารามิเตอร์ของอุปกรณ์ 7. ความแม่นยำของการควบคุมสามารถกำหนดได้จากจำนวนบิตที่ใช้ (Word Length) 8. ตัวประมวลผลเชิงเลขสามารถทำงานซ้ำหน้าที่ (Function) เดิมได้โดยไม่จำกัดจำนวนครั้ง	1. ต้องมีสัญญาณในการซิงโครไนซ์ (Synchronize) การจัดเวลา (Timing) ซึ่งจะมีผลต่อระบบการทำงานอย่างมาก 2. การเชื่อมต่อ (Interface) กับระบบการประมวลผลสัญญาณอนาลอก ทำให้วงจรมีความซับซ้อนมากขึ้น 3. การออกแบบระบบประมวลผลเชิงเลขจะมีความซับซ้อนมากกว่า โดยเฉพาะกับระบบที่มีขนาดใหญ่ 4. แถบปฏิบัติงาน ของระบบประมวลผลเชิงเลขจะต่ำกว่าระบบประมวลผลแบบอนาลอกมากเนื่องจากข้อจำกัดของอุปกรณ์ที่ใช้ เช่น วงจรเกต วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล(analog to digital converter ADC) และวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอก (digital to analog converter DAC) เป็นต้น 5. ระบบประมวลผลเชิงเลขจำเป็นต้องมีไฟเลี้ยงอยู่ตลอดเวลา 6. ในการแปลงสัญญาณอนาลอกให้อยู่ในรูปแบบสัญญาณเชิงเลขแล้วจะทำให้ความถูกต้องของสัญญาณบางส่วนขาดหายไป และไม่สามารถทำคืนให้ถูกต้องเหมือนเดิมได้เนื่องจากผลของการกำหนดขนาดในการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โครงสร้างของระบบประมวลสัญญาณเชิงเลข (Structure of digital Signal Processing)

การพิจารณาถึงโครงสร้างของระบบประมวลผลสัญญาณเชิงเลขโดยแสดงองค์ประกอบทางฮาร์ดแวร์ (Hardware) เป็นหลักดังรูปที่ 2.2 จะแยกส่วนประกอบต่าง ๆ ภายในระบบได้ดังต่อไปนี้ [1]



รูปที่ 2.2 บล็อกไดอะแกรมแสดงโครงสร้างระบบ DSP ที่กระทำกับสัญญาณอนาลอก

2.3.1 ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter: ADC)

วงจรเชื่อมต่อที่ทำการแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอล ดังนั้นเพื่อให้สัญญาณอนาลอกที่จะถูกแปลงโดย ADC จะต้องสุ่มวางค่าขนาดของ แอมพลิจูด (Amplitude) ค้างไว้ในช่วงเวลาสั้น ๆ ช่วงหนึ่งจนกว่าการแปลงจะเสร็จสมบูรณ์ จึงจะทำให้ความผิดพลาดมีค่าน้อยที่สุดสามารถทำได้โดยวงจร sample & Hold (S&H) ซึ่งข้อมูลที่ได้จะถูกจัดระดับสัญญาณ (Quantized) ให้เป็นระดับอ้างอิงที่ใกล้เคียงกันที่สุด โดยระดับอ้างอิงนี้จะแบ่งเป็น 2^n-1 ระดับเมื่อ n เป็นจำนวนบิตของการเปลี่ยนแปลง หลังจากนั้นจึงถูกเข้ารหัสสัญญาณข้อมูลเลขฐานสองขนาด n บิต แล้วจึงทำการสุ่มสัญญาณที่เข้ามาใหม่อีกครั้ง เช่นนี้ไปเรื่อย ๆ

2.3.2 ตัวประมวลผลสัญญาณเชิงเลข (Digital Signal Processor: DSP)

ในส่วนนี้จะทำการประมวลผลสัญญาณเชิงเลขโดยมีรูปแบบอัลกอริทึมเฉพาะหน้าที่ ซึ่งขึ้นอยู่กับเงื่อนไขของสัญญาณทางออกว่าต้องการให้มีลักษณะเช่นไร เช่น การทำการกรองสัญญาณความถี่สูง การกรองสัญญาณรบกวน หรือการมอดูเลตสัญญาณเป็นต้น โดยรูปทรงแบบการประมวลผลเชิงเลขจะมีตัวดำเนินการ (Operator) คือ ตัวบวก (Adder) และตัวคูณ (Multiplier) ที่มักจะถูกใช้ดำเนินการกับสัญญาณเชิงเลขอยู่บ่อย ๆ

2.3.3 ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก (Digital to Analog Converter: DAC)

DAC ทำหน้าที่แปลงสัญญาณดิจิตอลกลับมาเป็นสัญญาณอนาลอกกรณีที่เอาต์พุตของ DAC ลักษณะเหมือนขั้นบันไดอาจจำต้องทำการปรับค่าจาก DAC ให้มีสัญญาณราบเรียบขึ้นสามารถทำได้โดยวงจร Reconstruction (Smoothing) Filter เช่น first-order LPF interpolation

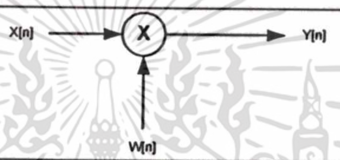
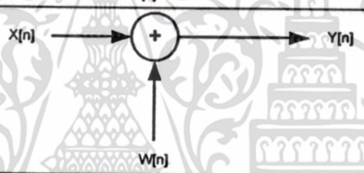
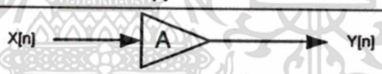
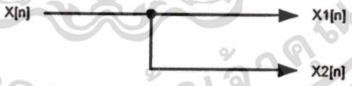
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในการใช้งานจริงนั้นการประมวลผลทั้งหมดจะถูกจำกัดแบนด์วิดธ์การประมวลผลด้วยความถี่การสุ่มสัญญาณของ ADC ดังนั้นจึงต้องมีการจำกัดแบนด์วิดธ์ที่เข้ามาเกินความถี่สูงสุดในการประมวลผลสัญญาณจำเป็นต้องใช้วงจรกรองอนาลอก (Anti-aliasing Filter) ต่อไว้ก่อนภาค ADC

2.4 โอโปรเรชั่นของลำดับสัญญาณ (Operation on Sequence)

ในระบบการประมวลผลเชิงเลขหนึ่ง ๆ จะมีโอเปอร์เรชันของสัญญาณ หรือการกระทำต่อลำดับสัญญาณ ซึ่งโอเปอร์เรชันจะมีระบบการประมวลผลเชิงเลขดังนี้ [2]

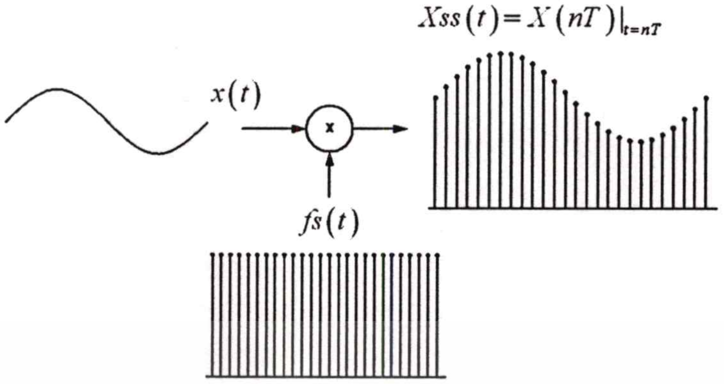
ตารางที่ 2.2 การกระทำต่อลำดับสัญญาณในระบบการประมวลผลเชิงเลข

การคูณกันของสัญญาณ		$Y[n]=X[n]*W[n]$
การบวกกันของสัญญาณ		$Y[n]=X[n]+W[n]$
การคูณสัญญาณด้วยค่าคงที่		$Y[n]=A*X[n]$
การหน่วงเวลา		$Y[n]=X[n-1]$
การแตกกิ่ง		$X1[n], X2[n]=X[n]$

2.5 ทฤษฏีการสุ่มตัวอย่าง (Sampling Theory)

ในการประมวลผลสัญญาณเชิงเลข สัญญาณอินพุตจะถูกเปลี่ยนให้สัญญาณเชิงเลขโดยมีการแทนสัญญาณต่อเนื่องช่วงซึ่งในแต่ละช่วงห่างกันที่เวลา Ts คงที่ โดยหากช่วงเวลาในการสุ่มตัวอย่าง Ts มีค่าที่เหมาะสมพอแล้วนั้นการแทนค่าดังกล่าวก็จะยังคงความถูกต้องไว้ได้ ดังนั้นผลการสุ่มตัวอย่างก็คือการคูณสัญญาณต่อเนื่องด้วยสัญญาณอิมพัลซ์ที่เลื่อนไปเรื่อย ๆ โดยมีเวลาห่างกันเป็นเวลา $T=Ts=1/fs$ ดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การสุ่มสัญญาณ (Sampling Signal)

จากรูปที่ 2.3 อิมพัลซ์แต่ละตัวกำหนดให้มีความห่างเท่ากับ T วินาทีโดยทั่วไปลำดับอิมพัลซ์หนึ่งหน่วยของ $fs(t)$ เขียนแทนด้วยสมการ

$$fs(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.1}$$

เนื่องจาก $fs(t)$ เป็นสัญญาณที่มีลักษณะที่มีสัญญาณเป็นคาบเพราะฉะนั้นจึงสามารถเขียนแทน $fs(t)$ ได้ด้วยอนุกรมฟูริเยอร์คือ

$$fs(t) = \sum_{n=-\infty}^{\infty} Cn \cdot e^{(j2n\pi/T)} = \sum_{n=-\infty}^{\infty} Cn \cdot e^{(jn\omega_s t)} \tag{2.2}$$

โดยที่ $\omega_s = 2\pi / T$ และ Cn คือค่าขนาดเชิงซ้อน (complex amplitude) ของสัญญาณความถี่เชิงซ้อน (complex frequency component) ω_o และ Cn หาค่าได้ตามลำดับคือ

$$Cn = \frac{1}{T} \int_{-\infty}^{\infty} fs(t) \cdot e^{(-jn\omega_s t)} dt$$

$$Cn = \frac{1}{T} \int_{-\infty}^{\infty} \delta(t) \cdot e^{(-jn\omega_s t)} dt$$

$$Cn = \frac{1}{T} \left\{ e^{(-jn\omega_s t)} \right\} \Big|_{-\infty}^{\infty}$$

$$Cn = \frac{1}{T} \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแทนค่า Cn จาก (2.3) กลับใน (2.2) จะได้

$$fs(t)=\frac{1}{T}\sum_{n=-\infty}^{\infty}e^{(jn\omega_s t)} \tag{2.4}$$

เมื่อทำการแปลงฟูริเยอร์เพื่อทำการหาค่าสเปกตรัมความถี่ของ $fs(t)$ จะได้

$$Fs(\omega)=\zeta\{fs(t)\}=\tfrac{1}{2\pi}\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_s) \tag{2.5}$$

ถ้าพิจารณาจากสมการ(2.4) และ (2.5) จะเห็นได้ว่า

$$\sum_{n=-\infty}^{\infty}\delta(t-nT)\rightarrow\omega_o\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_s) \tag{2.6}$$

จะเห็นได้ว่าเมื่อพิจารณาในโดเมนความถี่สเปกตรัมความถี่ของตัวสุ่มสัญญาณ $fs(t)$ เป็นอิมพัลซ์ที่วางห่างเท่า ๆ กันดังแสดงในรูปที่ 2.4 (b)

ถ้าให้ $Xss(t)$ เป็นสัญญาณที่ได้จากการสุ่มตัวอย่างดังนั้น

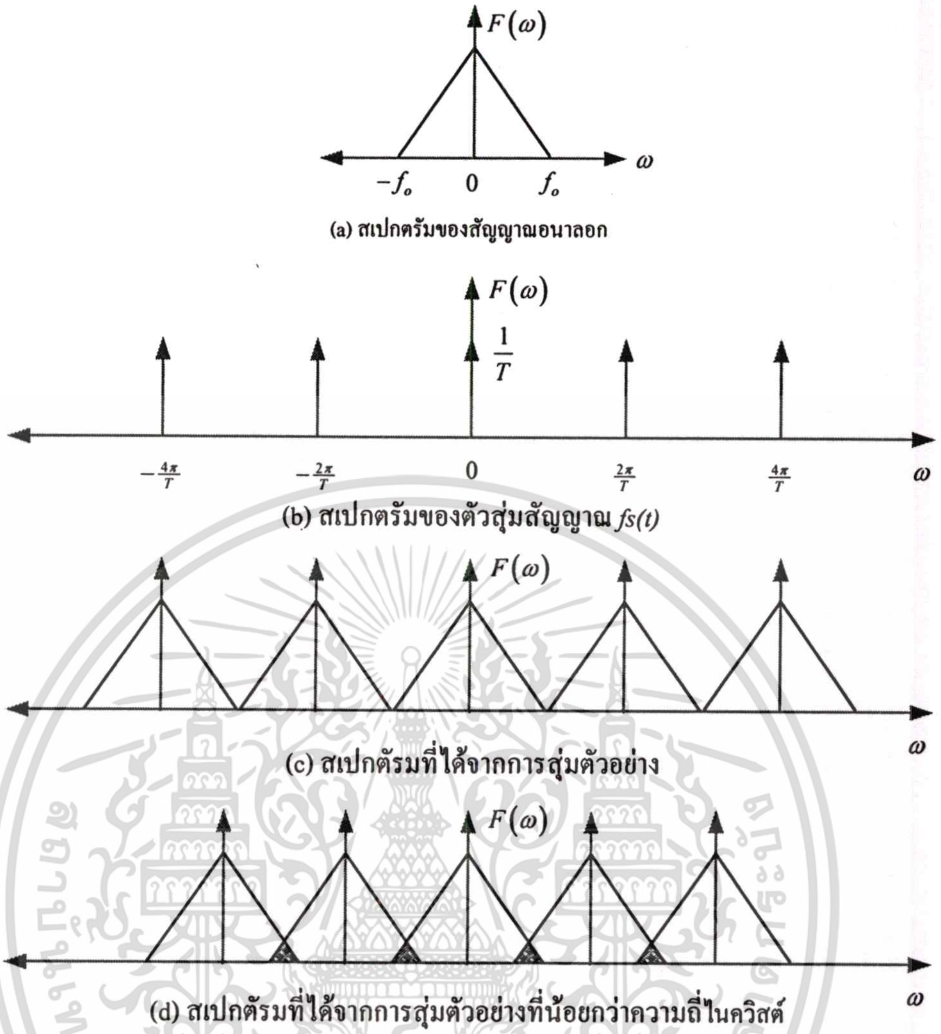
$$Xss(t)=fs(t)\cdot x(t) \tag{2.7}$$

และถ้าให้ $X(\omega)$ เป็นสเปกตรัมความถี่ของ $x(t)$ เนื่องจากในโดเมนเวลาสัญญาณที่ได้จากการสุ่มตัวอย่างเป็น การคูณกันของสองสัญญาณดังนั้นในโดเมนความถี่จึงเป็นการคอนโวลูชันกันระหว่างสเปกตรัมของ $fs(t)$ และ $x(t)$ หรือ

$$Xss(\omega)=Fs(\omega)*X(\omega) \tag{2.8}$$

โดยที่สัญลักษณ์ * แทนการคอนโวลูชันผลที่ได้แสดงดังรูป 2.4(c) ซึ่งจะเห็นได้ว่าจากกราฟสัญญาณอนาลอกที่มีแถบความถี่สมมุติให้มีค่าเท่ากับ $2\omega_o$ เมื่อสัญญาณนั้นถูกสุ่มตัวอย่าง $fs(t)$ ผลลัพธ์ที่ได้จะเห็นว่าสเปกตรัมของสัญญาณที่ได้จากการสุ่มตัวอย่าง $Xss(\omega)$ เป็นการนำเอาสเปกตรัมของสัญญาณ $X(\omega)$ มาวางเรียงห่างเท่ากันไปตลอดแกนความถี่ ω

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 สเปกตรัมของสัญญาณจากการสุ่มตัวอย่าง

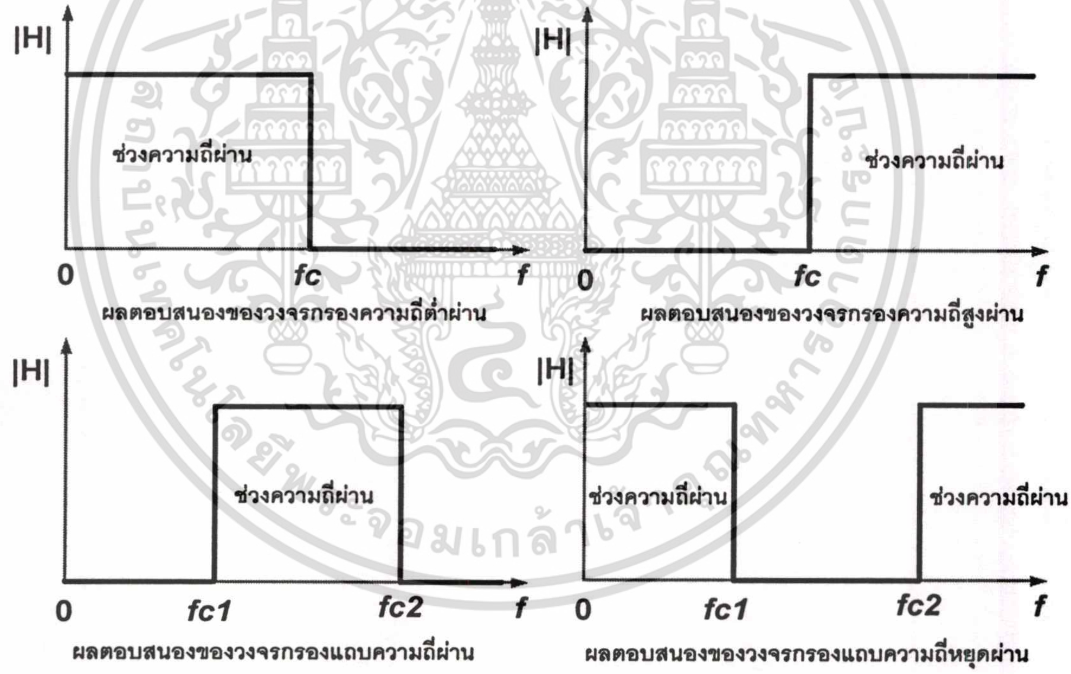
จากรูปที่ 2.4 (d) ถ้าความถี่ของการสุ่มสัญญาณ $fss(t)$ มีค่าต่ำหรือ ω_s มีค่าต่ำหรือมี T มาก จะทำให้ช่วงห่างของแต่ละกลุ่มสเปกตรัมของ $Xss(\omega)$ เข้ามาทับกันผลนี้ทำให้เกิดความผิดเพี้ยนไปของสเปกตรัมเดิมได้ผลนี้เรียกว่า เอเลสซิง (aliasing) หมายความว่าความถี่การสุ่มสัญญาณ (fs) ซึ่งไม่ทำให้สัญญาณสูญเสียข้อมูลที่สำคัญไป ทฤษฏีการสุ่มตัวอย่าง (Sampling Theory) ของแชนนอล (Shannon) กล่าวไว้ว่า "หากสัญญาณต่อเนื่อง $x(t)$ ที่มีความถี่ไม่เกิน $\omega_{\max} = 2\pi f_{\max}$ ข้อมูลของสัญญาณต่อเนื่องนั้นสามารถอธิบายด้วย $x(nT)$ ก็ต่อเมื่อความถี่ในการสุ่มตัวอย่าง fs มีค่ามากกว่าหรือเท่ากับสองเท่าของความถี่ใช้งานสูงสุด ($f_{\max}$) หรือ $fs \geq 2f_{\max}$ " [4] ซึ่งโดยทั่วไปจะทำการสุ่มตัวอย่างด้วยความถี่ $f_{sN} = 2f_{\max}$ พอดีค่าความถี่นี้มีชื่อเรียกว่าความถี่ไนควิสต์ (Nyquist frequency) และคาบเวลา $T_N = 1/(1f_{\max})$ นี้เรียกว่าช่วงเวลาสุ่มตัวอย่างไนควิสต์ (Nyquist interval)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 พื้นฐานวงจรกรองอนาลอกต้นแบบ

เนื่องจากในการออกแบบวงจรกรองดิจิตอลนั้นมีการอาศัยวงจรกรองอนาลอกต้นแบบ ดังนั้นในการเริ่มต้นออกแบบวงจรดิจิตอลจึงควรทราบการคำนวนฟังก์ชันถ่ายโอนของวงจรอนาลอกให้ได้ตามข้อกำหนดที่ต้องการเสียก่อน จากนั้นจึงทำการเขียนให้อยู่ในรูปแบบการชื่อมต่ออุปกรณ์อนาลอกชนิดต่าง ๆ หรืออาจแปลงสู่โดเมนดิจิตอลจากฟังก์ชันถ่ายโอนโดยตรงเพื่อสร้างวงจรกรองแบบต่าง ๆ ขึ้นมา

วงจรกรองความถี่ทำหน้าที่จำแนกความถี่ตามความต้องการของผู้ใช้ แบ่งตามคุณลักษณะของผลตอบสนองความถี่ (frequency response) ดังรูปที่ 2.5 ได้ 4 ชนิดด้วยกันคือ [5] วงจรกรองความถี่ต่ำผ่าน (low-pass filter : LPF) , วงจรกรองความถี่สูงผ่าน (high-pass filter : HPF) , วงจรกรองแถบความถี่ผ่าน (band-pass filter ; BPF) และวงจรกรองแถบความถี่หยุดผ่าน (band-stop filter : BSF)



รูปที่ 2.5 ชนิดของวงจรกรองความถี่แบบต่าง ๆ

จากผลตอบสนองความถี่ในรูปที่ 2.5 เมื่อให้ |H| คือขนาดของแรงดันทางด้านเอาต์พุต วงจรกรองความถี่ต่ำผ่าน จะยอมให้ความถี่ตั้งแต่ 0 Hz ถึงความถี่ f_c ผ่านไปยังเอาต์พุตของวงจรได้ ส่วนความถี่ที่สูงกว่า f_c ความถี่จะไม่ผ่านไปยังเอาต์พุตของวงจร สำหรับวงจรกรองความถี่สูงผ่านจะยอมให้ความถี่สูงกว่าความถี่ f_c ผ่านไปยังเอาต์พุตของวงจรได้ ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ f_c จะไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านไปยังเอาต์พุตของวงจร สำหรับวงจรกรองแถบความถี่ผ่าน จะยอมให้ความถี่ตั้งแต่ f_{c1} ถึงความถี่ f_{c2} ผ่านไปยังเอาต์พุตของวงจร ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ f_{c1} กับความถี่ที่สูงกว่า f_{c2} จะไม่ผ่านไปยังเอาต์พุตของวงจร และวงจรกรองแถบความถี่หยุดผ่าน จะไม่ยอมให้ช่วงความถี่ f_{c1} ถึงความถี่ f_{c2} ผ่านไปยังเอาต์พุตของวงจร ส่วนความถี่อื่น ๆ วงจรยอมให้ผ่านไปยังเอาต์พุตได้

2.6.1 ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอก

การสร้างฟังก์ชันถ่ายโอนต้นแบบอาศัยการประมาณพหุนามตามวิธีการแบบ บัตเตอร์เวิร์ธ เชบีเชฟ และอิลลิปติค ซึ่งอาจใช้วิธีการเปิดตาราง หรือใช้โปรแกรมช่วยในการคำนวณจากข้อกำหนดที่ต้องการ รูปแบบของฟังก์ชันถ่ายโอนต้นแบบในโดเมน s ดังสมการที่ (2.9)-(2.11)

$$H(s)=\frac{k(s-z_1)(s-z_2)\ldots}{(s-p_0)(s-p_1)(s-p_1^*)\ldots} \tag{2.9}$$

$$H(s)=\frac{N(s)}{(s+p_0)\prod(s+p_i)(s+p_i^*)}=\frac{N(s)}{D(s)} \tag{2.10}$$

$$H(s)=\frac{\sum_{i=0}^{m} b_i\cdot s^i}{\sum_{i=0}^{n} a_i\cdot s_i} \tag{2.11}$$

เมื่อ $z_1,z_2,\ldots,z_m$ ซีโรของวงจรกรอง $p_0,p_1,\ldots,p_n$ เป็นโพลของวงจรกรอง k อัตราขยาย และ a_i,b_i เป็นสัมประสิทธิ์ของพหุนาม s

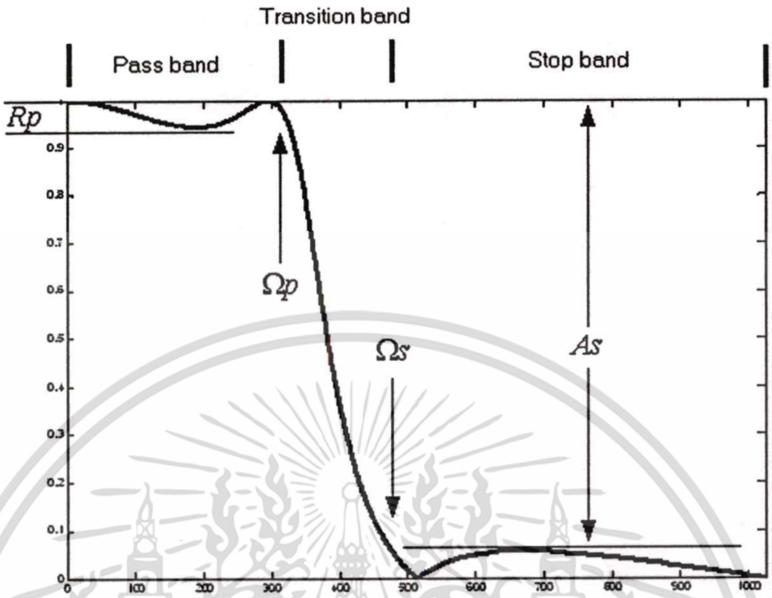
2.6.2 วงจรกรองอนาลอกแบบต่าง ๆ

ในการออกแบบวงจรกรองความถี่ [6] โดยทั่วไปจะต้องทำการกำหนดคุณสมบัติต่าง ๆ ซึ่งประกอบไปด้วย ชนิดและอันดับของวงจรกรอง (order: n) จุดตัดความถี่ (frequency cutoff : ω_c) การกระเพื่อมในแถบผ่าน (Passband ripple: R_p) และค่าลดทอนในช่วงความถี่หยุด (stopband ripple: R_s)

ในรูปที่ 2.6 แสดงผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำ Rp คือการกระเพื่อมในแถบผ่าน (dB) จะกำหนดให้มีระดับของขนาดจากยอดถึงยอดซึ่งโดยทั่วผู้ออกแบบจะกำหนดมีปริมาณเพียงเล็กน้อยช่วงความถี่ผ่านเท่านั้น (ในทางอุดมคติจะมีค่าเป็น 0) As คือการลดทอนในแถบหยุด (dB) Ωs คือความถี่ที่ขอบแถบหยุด (rad/s) Ωp เป็นความถี่ที่ขอบแถบผ่าน (rad/s) ความถี่ระหว่าง Ωp และ Ωs เรียกว่าความถี่ทรานซิชั่นแบนด์ซึ่งความชันนั้นแปรผันโดยตรงกับอันดับของวงจรกรองขึ้นอยู่กับผู้ออกแบบกำหนด อย่างไรก็ตาม ผลของริปเปิ้ลและทรานซิชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบนด์จะมีผลต่อกันโดยตรงซึ่งหากที่จะต้องการความชันที่มากก็จะทำให้มีค่าริปเปิ้ลมากตามไปด้วยเช่นเดียวกันซึ่งในทางอุดมคติค่า Ωp และ Ωs จะมีค่าเท่ากัน $(\Omega p = \Omega s)$



รูปที่ 2.6 คุณลักษณะผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำ

ซึ่งหากจะทำการออกแบบวงจรกรองความถี่แบบอื่น ๆ ไม่ว่าจะเป็นวงจรกรองแถบความถี่ผ่าน วงจรกรองแถบความถี่หยุด หรือวงจรกรองความถี่สูง สามารถนำเอาวงจรกรองความถี่ต่ำที่กล่าวไว้ในข้างต้นไปทำการออกแบบได้โดยวิธีการทรานฟอร์มทางความถี่ (frequency transform) ให้อยู่ในรูปแบบของวงจรกรองความถี่อื่น ๆ ซึ่งจะไม่ขอกล่าวถึง

ปัจจุบันการออกแบบวงจรกรองความถี่จะกำหนดค่าตัวแปรต่าง ๆ เหล่านี้ลงในซอฟท์แวร์ที่ช่วยในการออกแบบซึ่งจะได้พารามิเตอร์ต่าง ๆ ไปใช้งานต่อไปเช่นค่า R, L, C สำหรับการนำไปสร้างวงจรกรองแบบอนาลอก หรือค่าสัมประสิทธิ์ที่อยู่ในรูปของสมการผลต่างสำหรับนำไปสร้างไม่ว่าจะอยู่ในรูปแบบของ Hardware หรือ Software อื่น ๆ ก็ตาม โดยที่ชนิดของวงจรกรองความถี่จะเป็นไปตามวิธีการประมาณทางคณิตศาสตร์ซึ่งมีหลายแบบดังนี้

2.6.2.1 วงจรกรองอนาลอกแบบ บัทเตอร์เวิร์ธ (Butterworth filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกแบบ บัทเตอร์เวิร์ธ สร้างจาก อันดับ และ n ใดๆ ความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{\log\left\{\left(10^{\frac{Rp}{10}} - 1\right)\left(10^{\frac{As}{10}} - 1\right)\right\}}{2\log\left(\Omega_p / \Omega_s\right)} \tag{2.12}$$

$$\Omega c = \frac{\Omega p}{\sqrt[2N]{10^{\frac{Rp}{10}} - 1}} \tag{2.13}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.6.2.2 วงจรกรองอนาลอกแบบ เชบีเชฟ (Chebyshev filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกต้นแบบ เชบีเชฟ สร้างจาก อันดับ n ใดๆ และความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{\log\left(g+\sqrt{g^2-1}\right)}{\log\left(\Omega r+\sqrt{\Omega_r^2-1}\right)} \tag{2.14}$$

เมื่อ $g = \sqrt{\frac{10^{\frac{As}{10}}-1}{10^{\frac{Rp}{10}}-1}}$ และ $\Omega_r = \frac{\Omega s}{\Omega p}$ (2.15)

$$\Omega c = \Omega p$$

N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.6.2.3 วงจรกรองอนาลอกแบบ อิลิปติค (Elliptic filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกต้นแบบ อิลิปติค สร้างจาก อันดับ n ใดๆ และความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{K(k)K\left(\sqrt{1-k_1^2}\right)}{K(k_1)K\left(\sqrt{1-k^2}\right)} \tag{2.16}$$

เมื่อ $k = \frac{\Omega p}{\Omega s}, k_1 = \sqrt{\frac{10^{\frac{Rp}{10}}-1}{10^{\frac{As}{10}}-1}}$ และ $K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-x^2\sin^2\theta}}$ (2.17)

และมีความถี่ตัดเช่นเดียวกับวงจรของเชบีเชฟคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Omega c = \Omega p$$

N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.7 วงจรกรองความถี่แบบดิจิตอล

ในปัจจุบันเทคโนโลยีในการประมวลผลสัญญาณดิจิตอลมีความสามารถสูงขึ้นความละเอียดและความเร็วในการประมวลผลมากขึ้น จึงเป็นที่นิยมการประมวลผลในรูปแบบสัญญาณดิจิตอลมากขึ้นซึ่งระบบดิจิตอลนั้นมีความยืดหยุ่นมากกว่าเพราะอยู่ในรูปของซอฟท์แวร์เป็นส่วนใหญ่ซึ่งสามารถปรับปรุงและแก้ไขได้ง่ายกว่าโดยการเปลี่ยนค่าสัมประสิทธิ์โดยในขณะที่ในวงจรกรองความถี่ของอนาลอกสามารถทำได้ซับซ้อนกว่า

ในการออกแบบดิจิตอลฟิลเตอร์นั้นสามารถแบ่งออกได้สองประเภท [7] ได้แก่ การออกแบบโดยใช้วงจรกรองความถี่แบบ เอฟ ไอ อาร์ (Finite Impulse Response: FIR) และวงจรกรองความถี่แบบ ไอ ไอ อาร์ (Infinite Impulse Response: IIR)

2.7.1 วงจรกรองความถี่แบบ เอฟ ไอ อาร์ (FIR Filters)

วงจรกรองความถี่แบบ เอฟ ไอ อาร์ หรือ วงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวจำกัด ซึ่งเป็นการนำเอาเฉพาะสัญญาณอินพุตในแซมเปิ้ลปัจจุบันและแซมเปิ้ลที่ถูกหน่วงเวลามาทำการประมวลผลซึ่งสามารถเขียนสมการได้ดังนี้

$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{2.18}$$

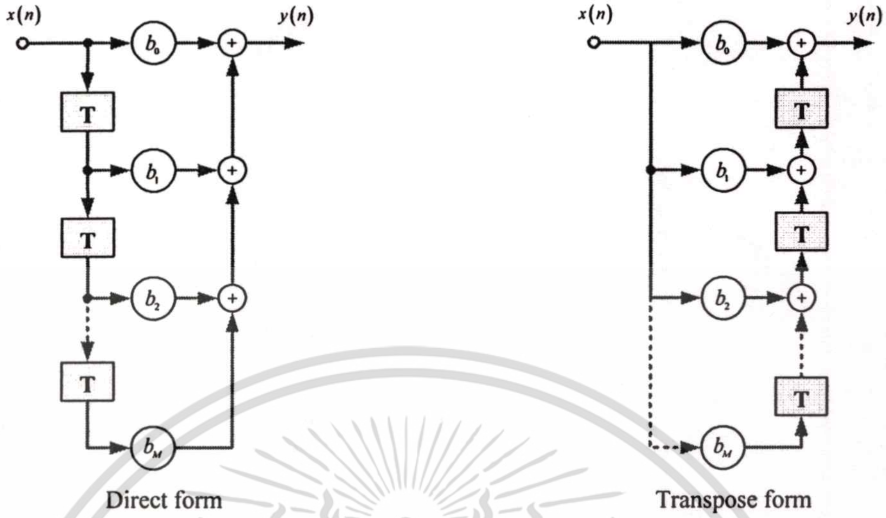
โดยที่ b_k คือค่าสัมประสิทธิ์ของวงจรกรองความถี่จะมีค่าจำนวนอยู่จำนวน M ค่าโดยที่จำนวนของค่าสัมประสิทธิ์นั้นจะขึ้นอยู่กับความต้องการของผู้ออกแบบซึ่งสามารถเขียนฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังนี้

$$H(z) = \sum_{k=0}^{M} b_k \cdot Z^{-k} \tag{2.19}$$

ซึ่งในการออกแบบจะพยายามลดค่าจำนวนสัมประสิทธิ์ให้มีจำนวนน้อยที่สุดในขณะที่ยังคงรักษาคุณสมบัติที่ต้องการไว้ได้อยู่ โดยปกติแล้วในการออกแบบใช้งานทั่วไปนั้นจะมีจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมประสิทธิ์ตั้งแต่ไม่กี่ตัวไปจนถึงร้อยตัวซึ่งทำให้มีโครงสร้างและวงจรที่มีขนาดใหญ่มาก แต่วงจรกรองดังกล่าวนั้นจะให้การตอบสนองทางเฟสที่เป็นเชิงเส้นโดยมีโครงสร้างดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างของวงจรกรองแบบ FIR

2.7.2 วงจรกรองความถี่แบบ ไอ ไอ อาร์ (IIR Filters)

วงจรกรองความถี่แบบ ไอ ไอ อาร์ หรือ วงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวไม่จำกัด ซึ่งเป็นการนำเอาสัญญาณอินพุตในแซมเปิ้ลปัจจุบันและแซมเปิ้ลที่ถูกหน่วงเวลาและเอาต์พุตแซมเปิ้ลที่ถูกหน่วงเวลามาทำการประมวลผลซึ่งสามารถเขียนสมการได้ดังนี้

$$\sum_{k=0}^{N} a_k \cdot y(n-k) = \sum_{k=0}^{M} b_k \cdot x(n-k) \quad (2.20)$$

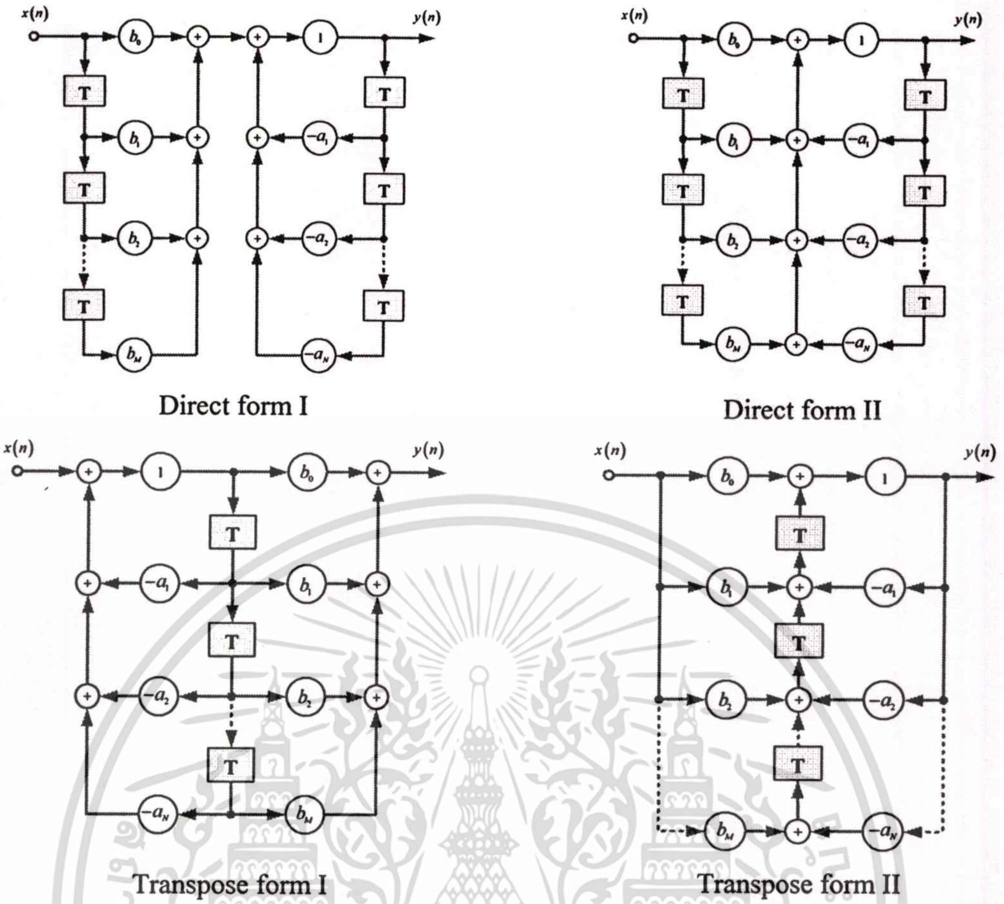
โดยที่ b_k และ a_k คือค่าสัมประสิทธิ์ของวงจรกรองความถี่จะมีค่าจำนวนอยู่จำนวน N และ M ค่าตามลำดับโดยที่จำนวนของค่าสัมประสิทธิ์นั้นจะมีจำนวนน้อยมากเมื่อเทียบกับวงจรกรองแบบ เอฟ ไอ อาร์ ภายใต้เงื่อนไขการออกแบบเดียวกัน ซึ่งสามารถเขียนฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังนี้

$$H(z) = \frac{\sum_{k=0}^{M} b_k \cdot Z^{-k}}{\sum_{k=0}^{N} a_k \cdot Z^{-k}} \quad (2.21)$$

วงจรกรองแบบ ไอ ไอ อาร์ นี้จะมีข้อดีก็คือมีจำนวนของสัมประสิทธิ์ที่น้อยกว่าวงจรกรองแบบ เอฟ ไอ อาร์ ทำให้ลดเวลาในการประมวลผลลงอย่างมากแต่จะมีข้อเสียเช่นกันก็คือในการออกแบบระบบนี้จะมีความไวในเรื่องของการปัดเศษค่าสัมประสิทธิ์ เพราะผลจากการปัดเศษค่าสัมประสิทธิ์อาจจะทำให้เกิดการไม่เสถียรภาพของวงจรได้และมีการตอบสนองทางเฟสที่ไม่เป็นเชิงเส้นซึ่งข้อดีข้อเสียของวงจรกรองความถี่ทั้งสองแบบแสดงดังตารางที่ 2.3

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.8 โครงสร้างของวงจรกรองความถี่แบบ IIR

โครงสร้างของวงจรกรองแบบ IIR จากสมการที่ (2.21) มีโครงสร้างแบ่งออกเป็น 4 แบบ ดังรูปที่ 2.8 ได้แก่โครงสร้างแบบตรง (Direct form) และแบบกลับ (Transpose form) ซึ่งจะมีความซับซ้อนของวงจรที่น้อยกว่า

ตารางที่ 2.3 ข้อดีและข้อเสียของวงจรกรองความถี่แต่ละแบบ

ไอ ไอ อาร์ (IIR)	เอฟ ไอ อาร์ (FIR)
1. วงจรอาจไม่เสถียรได้	1. วงจรเสถียรเสมอ
2. ให้ช่วงทรานซิชั่นแบนด์ที่แคบกว่า	2. ให้ช่วงทรานซิชั่นแบนด์ที่กว้างกว่า
3. มีการหน่วงสัญญาณที่เอาต์พุตน้อยกว่า	3. มีการหน่วงสัญญาณมากกว่า
4. มีความเพี้ยนของผลตอบสนองทางเฟสสูง	4. มีผลตอบสนองทางเฟสเป็นเชิงเส้น
5. มีสัญญาณรบกวนจากการปัดเศษมาก	5. มีสัญญาณรบกวนจากการปัดเศษน้อย
6. สามารถออกแบบได้จากวงจรอนาลอกต้นแบบ	6. การออกแบบทำได้โดยใช้วิธีการประมาณค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
67440

ในระยะเวลาหลายปีที่ผ่านมาได้มีนักวิจัยได้ทำการค้นคว้าหาทางแก้ข้อเสียต่าง ๆ ของ ไอ ไอ อาร์ ฟิลเตอร์ เช่น การมีความไวต่ำต่อการปัดเศษของค่าสัมประสิทธิ์ [8] โดยใช้โครงสร้างของ PAPF (Parallel all-pass Filter) โครงสร้างแบบอนุกรมคานอนิคัลที่มีค่าสัญญาณรบกวนต่ำ โครงสร้าง ไอ ไอ อาร์ ฟิลเตอร์ในรูปของวงจรกรองความถี่ผ่านตลอดย่าน โครงสร้างของ linear phase IIR filter ทำให้วงจรกรองแบบ IIR เป็นอีกทางเลือกหนึ่งในการนำไปใช้งาน

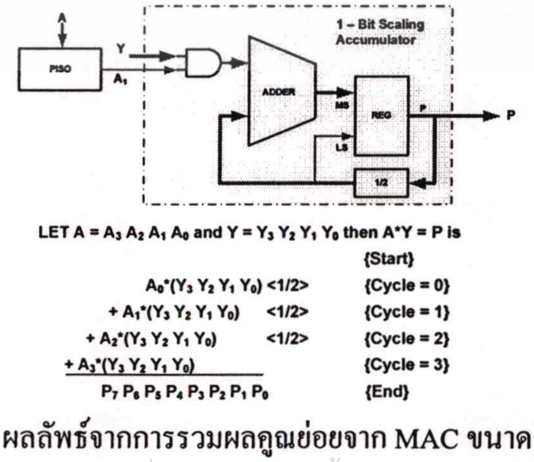
2.8 การประมวลผลสัญญาณเชิงเลขโดยวิธีการของคณิตศาสตร์แบบการกระจาย

คณิตศาสตร์แบบการกระจายหรือ (Distributed Arithmetic: DA) ได้ถูกนำเสนอขึ้นครั้งแรกในปี คศ. 1971 (US Patent 3,777,130) [9] โดย Croisier และ คณะ ซึ่งต่อมาได้ถูกนำไปประยุกต์ ใช้งานกับการประมวลผลสัญญาณดิจิตอลโดย Abrahan Peled และ Bede Liu [10] โดยรูปแบบการคำนวณของ DA เป็นรูปแบบที่สอดคล้องกับการประมวลผลสัญญาณดิจิตอลอีกทั้งยังสามารถนำไปสร้างในวงจรลอจิกได้อีกด้วย จึงทำให้โครงสร้างดังกล่าวถูกนำมาใช้งานอย่างแพร่หลาย โดยพื้นฐานของคณิตศาสตร์แบบการกระจายเป็นการเริ่มต้นที่การพิจารณาจากสมการ ผลบวกของผลคูณ (Sum of Product: SOP) หากกำหนดให้ Y เป็นข้อมูลเวกเตอร์ขนาดความยาว N ซึ่งทำการคูณอยู่กับค่าสัมประสิทธิ์เวกเตอร์ A ขนาดความยาว N เช่นเดียวกันจะได้ผลลัพธ์ P มีค่าดังนี้

$$P = \sum_{i=1}^{N} A_i \cdot Y_i \qquad (2.22)$$

2.8.1 การคูณค่าและการบวกสะสมค่าผลคูณ (MAC)

ถ้าพิจารณาจากสมการที่ (2.14) เฉพาะค่าของผลคูณย่อยของ $A_i \cdot Y_i$ โดยแทน A_i และ Y_i เป็นข้อมูลขนาด 4 บิตดังแสดงในรูปที่ 2.6 ผลคูณของเลขฐานสอง $A_i \cdot Y_i$ จะเป็นการคูณค่าแล้วทำการบวกสะสมค่าผลคูณที่ได้ทั้ง 4 ค่า (Multiply and Accumulate: MAC) ซึ่งจะถูกเรียกเป็นบล๊อก MAC [11]

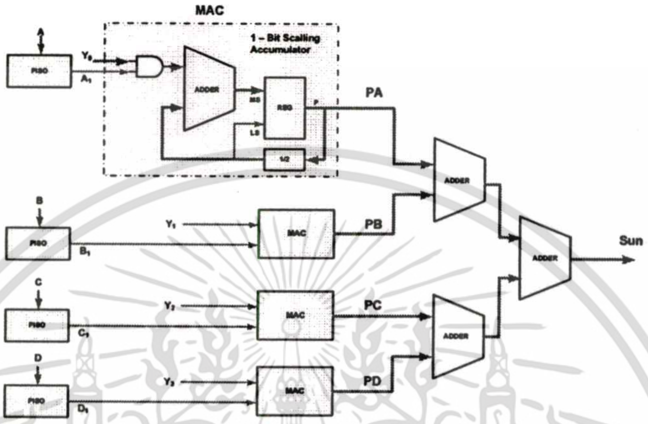


รูปที่ 2.9 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ขนาด 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAC จะทำการหาผลคูณย่อยในแต่ละครั้งระหว่างค่าสัมประสิทธิ์ Y และข้อมูล A ด้วยการ and กัน เทคนิคนี้จะทำการบวกค่าผลคูณย่อยเหล่านี้เข้าด้วยกันโดยแอกคิวมูเลเตอร์ซึ่งจะทำการเลื่อนบิตไปทางขวา 1 บิต ซึ่งเหมือนกับการหารข้อมูลนั้นด้วย 2 (แทนสัญลักษณ์ $\frac{1}{2}$) ในแต่ละรอบของสัญญาณนาฬิกาซึ่งเป็นการชดเชยสำหรับการถ่วงน้ำหนักบิตของผลคูณย่อย

เราสามารถหาค่าผลลัพธ์ในรูแบบของ SOP ได้ดังนี้ $SUM = A \cdot Y_0 + B \cdot Y_1 + C \cdot Y_2 + D \cdot Y_3$ ดังรูปที่ 2.10



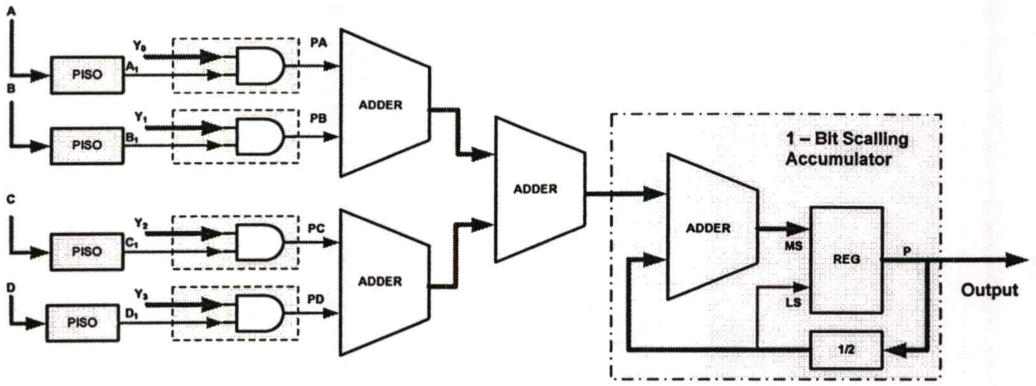
รูปที่ 2.10 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ทั้ง 4 ชุด

ตัวคูณทั้ง 4 ตัวจะถูกทำตามลำดับและผลลัพธ์จะถูกรวมเข้าด้วยกัน เมื่อทำการคูณเสร็จการทำงานจะใช้จำนวนรอบของสัญญาณนาฬิกา n รอบสำหรับข้อมูลขนาด n บิต ดังนั้นอัตราของสัญญาณนาฬิกาจึงมีค่าเท่ากับอัตราของข้อมูลหารด้วยจำนวนบิตโดยที่ระหว่างแต่ละรอบสัญญาณนาฬิกาของข้อมูลตัวคูณ 4 ตัวจะถูกรวมเข้าด้วยกันเป็นผลลัพธ์ออกมา (ในรูป PA, PB, PC และ PD) ทั้ง 4 ตัว จะถูกรวมกันเป็นสมการเอาต์พุต ดังนั้นจะเห็นได้ว่าถ้าจำนวนเทอมของผลคูณย่อยที่มีปริมาณมากขึ้นทำให้จำนวนของผลคูณย่อย และจำนวนของแอกคิวมูเลเตอร์เพิ่มมากขึ้นด้วยทำให้ขนาดของวงจรมีขนาดใหญ่ขึ้น

2.8.2 การลดขนาดของ MAC ด้วย DA [11]

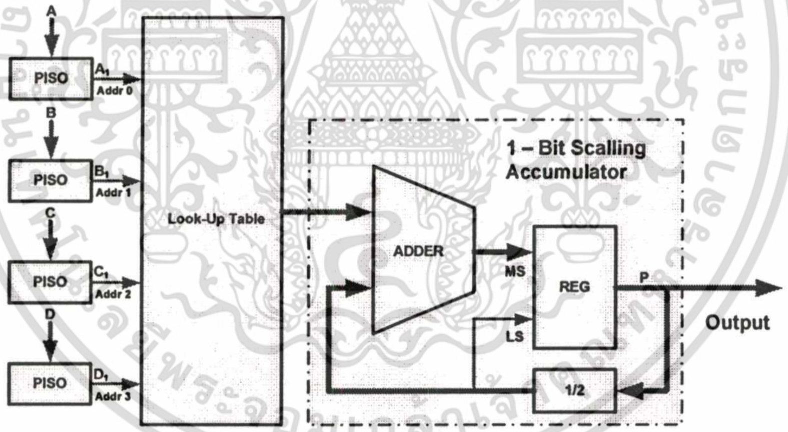
วิธีการประมวลผลแบบคณิตศาสตร์กระจายหรือ DA นั้น ต่างไปจากวิธีการทำงานของ MAC โดย DA จะทำการบวกค่าของผลคูณย่อยก่อน จากนั้นจะทำการชดเชยโดยการถ่วงน้ำหนักบิตของการบวกสะสมสามารถแสดงโครงสร้างการใช้ DA ได้ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ผลจากการรวมผลคูณย่อยจาก Serial Distributed Arithmetic แทน MAC ทั้ง 4 ชุด

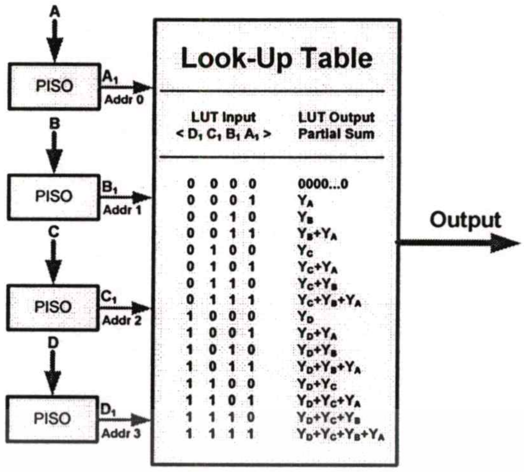
การทำงานของโครงสร้างนี้จะเป็นการเรียงลำดับของข้อมูลใหม่ โดยเทคนิคนี้จะช่วยลดจำนวนวงจรเลื่อนบิตและบวก (Shift and Add) รวมทั้งตัวบวกทั้ง 3 ตัว เป็นข้อมูล เราเรียกวิธีนี้ว่า ตารางดูค่า (Look up table: LUT) ขนาด 4 บิตให้ผลลัพธ์ ที่มีขนาดเล็กลง และเรียกรูปแบบนี้ว่า คณิตศาสตร์แบบการกระจายข้อมูลอนุกรม (Bit Serial Distributed Arithmetic MAC: SDA-MAC) ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของ SDA-MAC ที่สร้างจากตารางดูค่า

ข้อมูลของตารางดูค่า (LUT) ตามรูปที่ 2.13 ประกอบด้วยผลคูณย่อยของสัมประสิทธิ์ (Y_0 Y_1 Y_2 และ Y_3) LSB (output จากแต่ละ serial shift register) ของข้อมูลทั้ง 4 จะอ้างอิงตำแหน่งใน ตาราง LUT ถ้าทั้ง 4 บิตเป็นระดับสัญญาณ "1" เอาต์พุตของสัญญาณจากตารางดูค่าก็จะสอดคล้องกับสัมประสิทธิ์ทั้ง 4 ค่า ซึ่งจะมีค่าทั้งหมด 16 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ค่าของ LUT ที่เกิดจากค่าสัมประสิทธิ์ทั้ง 4 ค่า

2.9 การสร้างดิจิตอลฟิลเตอร์แบบเลขคณิตแจกแจง [12]

ในวงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวจำกัด (FIR) ในสมการ (2.23) สามารถเขียนเป็นสมการผลต่างได้เป็น

$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{2.23}$$

ในวงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวไม่จำกัด (IIR) ในสมการ (2.24) สามารถเขียนเป็นสมการผลต่างสืบเนื่องได้เป็น

$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) - \sum_{k=1}^{M} b_k \cdot y(n-k) \tag{2.24}$$

ซึ่งจะเห็นได้ว่าในการคำนวณ $y(n)$ นั้นจะต้องทำการคูณและการบวกโดยที่ทราบกันดีแล้วว่าการคูณในคอมพิวเตอร์นั้นจะใช้เวลาที่นานมากกว่าคำสั่งอื่น ๆ ทำให้ในการทำงานที่ความเร็วสูง ๆ อาจทำงานเป็นแบบเวลาจริงไม่ได้ (Real-time) ซึ่งโครงสร้างแบบเลขพีชคณิตแจกแจง (Distributed Arithmetic Structure) หรือที่เรียกว่าโครงสร้างแบบ Rom/Acc (Rom-Accumulator) สามารถเปลี่ยนการคูณให้อยู่ในรูปแบบของการบวก (Adding) และการเลื่อน (Shifting) โดยสามารถนำหน่วยความจำประเภท ROM มาประยุกต์ใช้งานร่วมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 หลักการโครงสร้างของเลขคณิตแจกแจง

เมื่อพิจารณาวงจรกรอง IIR ที่มีฟังก์ชันถ่ายโอนอันดับ 3 (n=3) มีรูปทั่วไปคือ

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} \tag{2.25}$$

สามารถเขียนให้อยู่ในรูปแบบของสมการผลต่าง (Difference equation) ได้คือ

$$\begin{aligned} y(n) = {} & a_0 \cdot x(n) + a_1 \cdot x(n-1) + a_2 \cdot x(n-2) + a3 \cdot x(n-3) \\ & - b_1 \cdot y(n-1) - b_2 \cdot y(n-2) - b_3 \cdot y(n-3) \end{aligned} \tag{2.26}$$

โดยให้ $x(n), x(n-1), x(n-2)$ และ $x(n-3)$ เป็นลำดับของสัญญาณที่เข้า

$y(n), y(n-1), y(n-2)$ และ $y(n-3)$ เป็นลำดับของสัญญาณที่ออก

$a_0, a_1, a_2, a_3, b_1, b_2$ และ b_3 เป็นค่าสัมประสิทธิ์ของตัวกรอง

วิธีการของโครงสร้างเลขคณิตแจกแจงทำโดยการเขียนแทนลำดับสัญญาณเข้าและลำดับสัญญาณออกด้วยตัวเลขแบบเติมเต็มสอง (2's Complement) ที่มีจำนวนบิตรวมทั้งเครื่องหมายด้วยเป็น B บิตหรือเขียนกระจายเป็นเลขฐานสองคือ

$$x(n) = x_0(n) \cdot x_1(n) x_2(n) \cdots x_{B-1}(n) \tag{2.27}$$

$$y(n) = y_0(n) \cdot y_1(n) y_2(n) \cdots y_{B-1}(n) \tag{2.28}$$

โดยที่ $x_0(n)$ และ $y_0(n)$ เป็นบิตที่แสดงเครื่องหมายของตัวเลข ส่วน $x_1(n)x_2(n) \cdots x_{B-1}(n)$ และ $y_1(n)y_2(n) \cdots y_{B-1}(n)$ เป็นบิตที่ i ของลำดับสัญญาณมีค่าเป็น "0" หรือ "1" เท่านั้น สามารถเขียนในรูปแบบของ 2's Complement ได้ดังนี้

$$x(n) = -x_0(n) + \sum_{i=1}^{B} x_i(n) \cdot 2^{-i} \tag{2.29}$$

$$y(n) = -y_0(n) + \sum_{i=1}^{B} y_i(n) \cdot 2^{-i} \tag{2.30}$$

เมื่อแทนสมการ (2.29) และ (2.30) ลงในสมการผลต่างสืบเนื่องจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(n) = a_0 \left\{\sum_{i=1}^{B} x_i(n)\cdot 2^{-i} - x_0(n)\right\} + a_1 \left\{\sum_{i=1}^{B} x_i(n-1)\cdot 2^{-i} - x_0(n-1)\right\}$$
$$+ a_2 \left\{\sum_{i=1}^{B} x_i(n-2)\cdot 2^{-i} - x_0(n-2)\right\} + a_3 \left\{\sum_{i=1}^{B} x_i(n-3)\cdot 2^{-i} - x_0(n-3)\right\}$$
$$- b_1 \left\{\sum_{i=1}^{B} y_i(n-1)\cdot 2^{-i} - y_0(n-1)\right\} - b_2 \left\{\sum_{i=1}^{B} y_i(n-2)\cdot 2^{-i} - y_0(n-2)\right\} \quad (2.31)$$
$$- b_3 \left\{\sum_{i=1}^{B} y_i(n-3)\cdot 2^{-i} - y_0(n-3)\right\}$$

เมื่อทำการจัดพจน์ใหม่โดยนำบิตที่สมนัยมาเขียนรวมกันจะได้

$$y(n) = \sum_{i=1}^{B} 2^{-i} \cdot \left\{\begin{matrix} a_0\cdot x_i(n) + a_1\cdot x_i(n-1) + a_2\cdot x_i(n-2) + a_3\cdot x_i(n-3) \\ -b_1\cdot y_1(n-1) - b_2\cdot y_2(n-2) - b_3\cdot y_3(n-3) \end{matrix}\right\}$$
$$- \left\{\begin{matrix} a_0\cdot x_0(n) + a_1\cdot x_0(n-1) + a_2\cdot x_0(n-2) + a_3\cdot x_0(n-2) - b_1 y_0(n-1) \\ -b_2 y_0(n-2) - b_3\cdot y_0(n-3) \end{matrix}\right\} \quad (2.32)$$

ให้

$$F_i\{\} = F_i\{x_i(n), x_i(n-1), x_i(n-2), x_i(n-3), y_i(n-1), y_i(n-2), y_i(n-3)\} \quad (2.33)$$

และ

$$F_i\{\cdot\} = a_0\cdot x_i(n) + a_1\cdot x_i(n-1) + a_2\cdot x_i(n-2) + a_3\cdot x_i(n-3)$$
$$-b_1\cdot y_i(n-1) - b_2\cdot y_i(n-2) - b_3\cdot y_i(n-3) \quad (2.34)$$

จะได้

$$y(n) = \sum_{i=1}^{B-1} 2^{-i}\cdot F_i\{\cdot\} - F_0\{\cdot\} \quad (2.35)$$

ผลจากสมการ (2.33) และ (2.34) จะได้โครงสร้างแบบเลขคณิตแจกแจง โดยนำค่าฟังก์ชันทั้งหมดของ $F_i\{\cdot\}$ มาทำเป็นตารางเปิดดู (Look up table) โดยค่าในตารางเปิดดูสามารถคำนวณจากสมการ (3.34) และเนื่องจาก $F_i\{\cdot\}$ มีตัวแปรอยู่ 7 ตัว คือ $x_i(n)$, $x_i(n-1)$, $x_i(n-2)$, $x_i(n-3)$, $y_i(n-1)$, $y_i(n-2)$ และ $y_i(n-3)$ ทำให้ตารางเปิดดูมีค่าฟังก์ชันของ $F_i\{\cdot\}$ 2^7 ค่า โดยค่าของฟังก์ชัน $F_i\{\cdot\}$ ทั้ง 128 ค่านี้จะถูกคำนวณแล้วทำการปัดเศษให้เหลือ B บิต แล้วเก็บไว้ใน ROM เพื่อนำไปสร้างตัวกรองต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 การสร้างตารางเปิดดูจากฟังก์ชัน $F_i\{.\}$ สำหรับวงจรกรองอันดับ 3

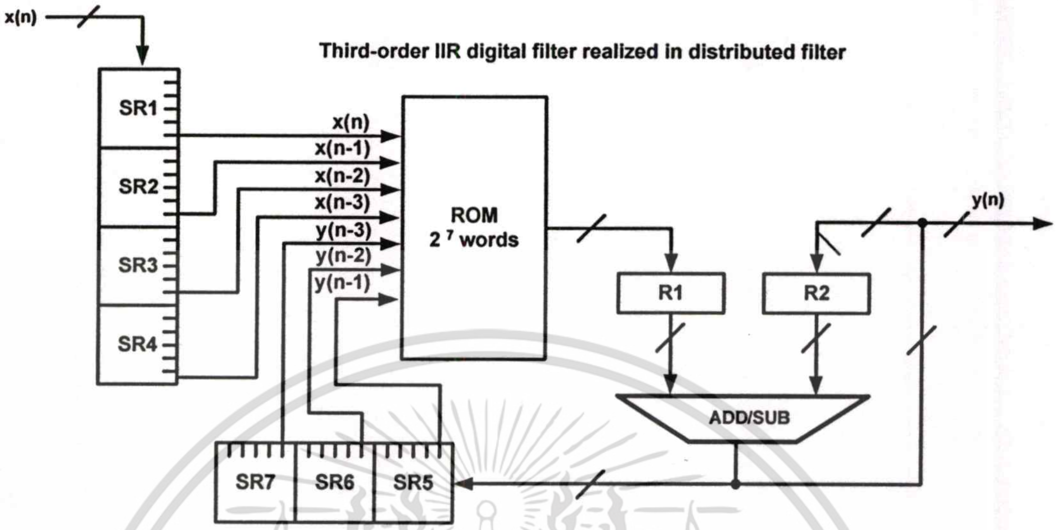
เมื่อได้สมการผลต่างสืบเนื่องอันดับ 3 แล้วนำค่าสัมประสิทธิ์คูณกับบิตแรกของแต่ละตัว แล้วบวกกันเก็บไว้ที่แอดเดรสที่ชี้โดยค่า $x_i(n), x_i(n-1), x_i(n-2), x_i(n-3), y_i(n-1), y_i(n-2)$ และ $y_i(n-3)$ ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 ค่าของ $F_1\{.\}$ ภายใน ROM

ตำแหน่งที่	แอดเดรสของรอม							ค่าของ $F_1\{.\}$ ภายใน ROM
	$x_i(n)$	$x_i(n-1)$	$x_i(n-2)$	$x_i(n-3)$	$y_i(n-1)$	$y_i(n-2)$	$y_i(n-1)$	
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	$-b_3$
2	0	0	0	0	0	1	0	$-b_2$
3	0	0	0	0	0	1	1	$-b_2-b_3$
4	0	0	0	0	1	0	0	$-b_1$
5	0	0	0	0	1	0	1	$-b_1-b_3$
6	0	0	0	0	1	1	0	$-b_1-b_2$
7	0	0	0	0	1	1	1	$-b_1-b_2-b_3$
8	0	0	0	1	0	0	0	a_3
9	0	0	0	1	0	0	1	a_3-b_3
10	0	0	0	1	0	1	0	$a3-b_2$
11	0	0	0	1	0	1	1	$a_3-b_2-b_3$
123	1	1	1	1	0	1	1	$a_0+a_1+a_2+a_3-b_2-b_3$
124	1	1	1	1	1	0	0	$a_0+a_1+a_2+a_3-b_1$
125	1	1	1	1	1	0	1	$a_0+a_1+a_2+a_3-b_1-b_3$
126	1	1	1	1	1	1	0	$a_0+a_1+a_2+a_3-b_1-b_2$
127	1	1	1	1	1	1	1	$a_0+a_1+a_2+a_3-b_1-b_2-b_3$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการนำค่าของสัมประสิทธิ์ที่ได้จากการคำนวณมาหาค่าของแอดเดรสของรอมซึ่งมีทั้งหมด 128 ตำแหน่งโดยต่อร่วมกับรีจิสเตอร์และวงจรบวกตามรูปที่ 2.14



รูปที่ 2.14 โครงสร้างของ Distributed Arithmetic วงจรกรองความถี่แบบ IIR อันดับ 3

จากรูปที่ 1 SR1 และ SR5 เป็นรีจิสเตอร์แบบเข้าขนานออกอนุกรม (Serial In Parallel Out: SIPO) ใช้เก็บข้อมูลอินพุตในปัจจุบัน ส่วน SR2, SR3, SR4, SR6 และ SR7 เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลอินพุตในอดีต และเอาต์พุตในอดีต ซึ่งแบบเข้าอนุกรมออกอนุกรม (Serial In Serial Out: SISO) และ R1, R2, R3 เป็นรีจิสเตอร์ชั่วคราวใช้เก็บข้อมูลในการประมวลผลของวงจร ADD/SUB ซึ่งเป็นวงจรบวกลบเลข 2's complement โดยมีขั้นตอนการทำงานดังนี้

1. ทำการเคลียร์ข้อมูลในรีจิสเตอร์ R2 จากนั้นทำการโหลดข้อมูล x(n) เข้ามาเก็บไว้ในรีจิสเตอร์ SR1
2. ข้อมูลในบิตต่ำสุด (LSB) ของรีจิสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 เป็นลำดับสัญญาณของ x(n), x(n-1) , x(n-2) , x(n-3) , y(n-1) , y(n-2) และ y(n-3) ตามลำดับ โดยข้อมูลดังกล่าวจะใช้เป็นแอดเดรสให้กับ ROM เพื่อหาค่า F_i เมื่อ i เป็นข้อมูลบิตที่ L โดยข้อมูลที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการบวกค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้จะถูกเก็บไว้ใน R2 พร้อมกับเลื่อนข้อมูลไปทางขวา 1 บิต
3. ทำการเลื่อนข้อมูลในรีจิสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 ไป 1 บิต เพื่อกำหนดแอดเดรสของ ROM ใหม่ ได้ค่า F_i เมื่อ i เป็นบิตที่ L-1 จากนั้นนำค่าที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการบวกค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้จะถูกเก็บไว้ใน R2 พร้อมกับเลื่อนข้อมูลไปทางขวา 1 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำซ้ำตามขั้นตอนที่ 3 สำหรับ i เป็นบิตที่ L-2, L-3, . . . , 1 ตามลำดับ
5. ทำการเลื่อนข้อมูลในรีจิสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 ไป 1 บิต เพื่อกำหนดแอดเดรสของ ROM ใหม่ ได้ค่า F_i เมื่อ i เป็นบิตที่ L-1 จากนั้นนำค่าที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการลบค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้คือลำดับสัญญาณ y(n)
6. ทำซ้ำตามขั้นตอนที่ 1-5 ใหม่สำหรับค่า y(n) ในลำดับถัดไป

2.12 บทสรุป

ในบทนี้ได้กล่าวถึงการประมวลผลสัญญาณที่อยู่ในรูปแบบของสัญญาณอนาลอกและสัญญาณดิจิตอล ซึ่งจะมีคุณสมบัติเด่นที่แตกต่างกันออกไปโดยในกระบวนการประมวลผลสัญญาณจะต้องทำการจัดรูปแบบให้เหมาะสมกับระบบนั้นรวมไปถึงเงื่อนไขที่จะต้องพิจารณา ไปจนถึงโครงสร้างของวงจรกรองแบบต่าง ๆ การประมวลผลเชิงเลขด้วยโครงสร้างของคณิตศาสตร์แบบการกระจายซึ่งโครงสร้างดังกล่าวเหมาะสมกับการออกแบบนำไปสร้างในวงจรดิจิตอล โดยในบทนี้ได้กล่าวถึงประเด็นที่สำคัญ ๆ ที่ควรทราบ ซึ่งในปัจจุบันนั้นได้มีเครื่องมือที่ช่วยในการออกแบบวงจรกรองความถี่ทั้งในรูปแบบอนาลอกและดิจิตอลโดยตรงทำให้ลดความยุ่งยากในการหาค่าของสัมประสิทธิ์ของวงจรกรองต้นแบบลงได้อย่างมากและยังสามารถนำไปประยุกต์ใช้ในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คซึ่งจะได้กล่าวถึงในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

สถาปัตยกรรมและการออกแบบ FPGA

3.1 บทนำ

ในปัจจุบันงานอุตสาหกรรมทางด้านอิเล็กทรอนิกส์ได้รุดหน้าไปอย่างรวดเร็วมากและอีกทั้งประสิทธิภาพที่เพิ่มขึ้นของอุปกรณ์อิเล็กทรอนิกส์ เพื่อให้รองรับกับความต้องการของการนำมาใช้กับงานที่มีความซับซ้อนมากขึ้นเรื่อย ๆ นั้น สิ่งสำคัญที่สุดย่อมขึ้นอยู่กับประสิทธิภาพของชิ้นส่วนแต่ละชิ้นที่อยู่ภายในชิ้นส่วนอิเล็กทรอนิกส์ โดยส่วนที่สำคัญอย่างมากคือ วงจรรวมหรือที่เรียกว่า ไอซี (IC: Integrate Circuit) หรือ ไมโครชิพ ซึ่งมีการพัฒนาไปอย่างเร็วมากเช่น การพัฒนาทางด้านเทคโนโลยีของไมโครโปรเซสเซอร์ (MPU) หรือไมโครคอนโทรลเลอร์ (MCU) และอุปกรณ์ประเภทหน่วยความจำ (Memory) ซึ่งในท้องตลาดปัจจุบันมีไอซีหลากหลายชนิดให้เลือกนำมาใช้งานตามความต้องการ ไม่ว่าจะเป็นในเรื่องของราคาและประสิทธิภาพ

3.2 เอฟ พี จี เอ (FPGA: Field Programmable Gate Array)

FPGA เป็นอุปกรณ์ที่ถูกพัฒนาต่อมาจากอุปกรณ์ประเภท LCA(Logic Cell Array File) ของบริษัทไซลิงซ์ (Xilinx) ในปี 1985 [13] โดยมีจุดเด่นตรงที่มีประสิทธิภาพการทำงานและความหนาแน่นของจำนวนเกตสูง การใช้งานที่สะดวกสามารถกำหนดฟังก์ชันการทำงานได้ตามต้องการโดยผ่านทางซอฟท์แวร์เพื่อจะทำการโปรแกรมข้อมูลลงในชิพ FPGA ให้สามารถทำงานตามที่ออกแบบไว้โดยอัตโนมัติดังนั้นทำให้เวลาในการออกแบบและพัฒนาวงจรรวมด้วย FPGA จึงมีความสะดวกและรวดเร็วกว่าวิธีอื่นและยังมีต้นทุนค่าใช้จ่ายที่ต่ำเพราะได้ลดความเสี่ยงในการที่จะต้องแก้ไขดัดแปลงวงจรต้นแบบและการเลื่อนเวลาออกแบบผลิตภัณฑ์ลงไปด้วย

หากทำการเปรียบเทียบ FPGA กับไมโครคอนโทรลเลอร์ที่เป็นชิพที่ใช้สำหรับงานทั่วไป (General Propose Device) ซึ่งหมายถึงชิพที่ได้มีการออกแบบโครงสร้างและสถาปัตยกรรมตลอดไปจนถึงชุดคำสั่งไว้เรียบร้อยแล้ว ผู้ใช้งานเพียงทำหน้าที่เรียบเรียงชุดคำสั่งเพื่อให้ชิพทำงานตาม แต่ด้วยข้อจำกัดทางด้านสถาปัตยกรรมและชุดทำสั่ง ดังนั้นในไมโครคอนโทรลเลอร์จึงไม่ใช่คำตอบกับทุกงานเสมอไปยิ่งโดยเฉพาะการนำเอาไปใช้งานในงานประมวลผลสัญญาณที่มีความเร็วสูงมาก ๆ เช่น สัญญาณภาพ เป็นต้น ซึ่งแตกต่างจาก FPGA ที่สามารถกำหนดโครงสร้างและสถาปัตยกรรม อีกทั้งยังสามารถออกแบบให้ทำงานแบบขนานกันได้ โดยในปัจจุบันยังสามารถทำงานที่สัญญาณนาฬิกาที่สูงถึง 100 MHz เลยทีเดียว ดังนั้นประสิทธิภาพการทำงานของชิพจริง ๆ จึงขึ้นอยู่กับสถาปัตยกรรมและโครงสร้างที่ได้ออกแบบเป็นหลัก นอกจากนั้นผู้ออกแบบยังสามารถ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบชุดคำสั่งใหม่ ๆ หรือฟังก์ชันใหม่ ๆ ได้เองเพื่อความเหมาะสมกับการใช้งาน จะเห็นได้ว่า FPGA จึงเป็นอีกทางเลือกหนึ่งของการออกแบบวงจรรวมตลอดไปจนถึงการนำไปใช้ในการออกแบบวงจรต้นแบบได้เป็นอย่างดี

3.2.1 คุณลักษณะเด่นของ FPGA

จากปัญหาในการออกแบบและพัฒนาที่ได้กล่าวมาแล้วนั้นจะเห็นได้ว่า FPGA เป็นอีกหนึ่งทางเลือกหนึ่งที่มีความนิยมนำมาใช้ในงานออกแบบและพัฒนาต้นแบบผลิตภัณฑ์ที่มีข้อดีในด้านต้นทุนการออกแบบและความสะดวกในการพัฒนา ซึ่งในปัจจุบันมีบริษัทผู้ผลิตชิพหลายรายเช่น Xilinx, Altera [14] หรือ Actel [15] เป็นต้น ซึ่งหากแบ่งตามส่วนแบ่งทางการตลาดแล้ว Xilinx จะมีส่วนแบ่งทางการตลาดสูงที่สุด โดยที่แต่ละบริษัทก็จะผลิตชิพ FPGA ที่มีขนาดและราคาที่ต่างกันออกไปส่วนมากจะถูกเรียกชื่อตามโครงสร้างและขนาดความจุของชิพ FPGA ซึ่งถูกวัดด้วยค่าจำนวนเทียบเท่าเกต (Equivalent Gate) ซึ่งจะมีตั้งแต่น้อยกว่า 10,000 เกต ไปจนถึงมากกว่า 5 ล้านเกต ขึ้นอยู่กับโครงสร้างที่นำมาผลิตชิพ เช่นชิพของบริษัท Xilinx ตระกูล 95xx จะเป็นเทคโนโลยีที่มีโครงสร้างจาก EEPROM ชิพตระกูลนี้จะมีขนาดความจุไม่เกิน 10,000 เกต ในขณะที่ชิพตระกูล spatan จะมีความจุมากถึง 4 แสนเกต และตระกูล Virtex ที่มีโครงสร้างแบบ SRAM จะมีความจุมากถึง 4 ล้านเกตเลยทีเดียวสามารถนำมาออกแบบวงจรรวมที่มีขนาดใหญ่ได้อย่างสบาย ๆ

3.3 การออกแบบวงจรดิจิตอล

รูปแบบการออกแบบวงจรรวมสามารถแบ่งออกได้สองประเภทหลัก ๆ ได้ดังนี้

3.3.1 การออกแบบโดยใช้การวาดผังวงจร (Schematic Design)

ในอดีตการออกแบบวงจรรวมดิจิตอลจะเป็นการออกแบบวงจรในระดับลอจิกเกต (Logic level) โดยใช้โปรแกรมช่วยวาดผังวงจรซึ่งการออกแบบวงจรด้วยวิธีนี้จะค่อนข้างใช้เวลามากในการออกแบบ เนื่องจากการออกแบบทุก ๆ อย่างนั้นผู้ออกแบบจะต้องออกแบบเองทั้งหมดซึ่งจะไม่เหมาะกับงานออกแบบวงจรที่มีความซับซ้อนสูง จึงเป็นข้อจำกัดอย่างหนึ่งในการออกแบบในวิธีนี้

3.3.2 การออกแบบโดยใช้ภาษาระดับสูง (High Description Language)

ด้วยเทคโนโลยีการออกแบบวงจรดิจิตอลได้พัฒนาสูงขึ้นและการพัฒนาภาษาในการออกแบบวงจรดิจิตอลพร้อมกับกระบวนการออกแบบแนวใหม่ที่เป็นการเขียนด้วยภาษาอธิบายฮาร์ดแวร์ (HDL: Hardware Description Language) [16] มาใช้ในการออกแบบวงจรซึ่งจะมีอยู่สองภาษที่นิยมใช้คือภาษา Verilog และ VHDL โดยโครงสร้างของภาษา Verilog จะคล้ายกับภาษาซี ส่วนภาษา VHDL จะคล้ายกับภาษา Pascal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโดยใช้ภาษา VHDL จะเป็นกระบวนการออกแบบที่เราเรียกว่า Top Down Design โดยจะอาศัยซอฟต์แวร์ในการออกแบบซึ่งเรียกว่า EDA (Electronic Design Automation) ซึ่งจำลองการทำงานจากโค้ดที่เขียนขึ้น (HDL Simulations) และโค้ดที่ผ่านการจำลองการทำงานจะถูกนำไปสังเคราะห์ (Synthesis) ให้เป็นวงจรรวมเกต (Logic Level) ที่สามารถนำไปใช้งานได้ซึ่งอุปกรณ์ที่รองรับเทคโนโลยีดังกล่าวได้แก่ CPLD (Complex Programmable Logic Device) หรือ FPGA (Field Programmable Gate Array) หรือแม้กระทั่งนำไปออกแบบในระดับเอสิก (ASIC: Application Specific Integrate circuit) จะเห็นได้ว่าการออกแบบโดยใช้ภาษา VHDL จะทำให้ผู้ออกแบบสามารถออกแบบวงจรที่มีความซับซ้อนได้อย่างสะดวกรวดเร็วขึ้น และยังสามารถนำกลับมาใช้ใหม่ได้ หรือหากต้องการเปลี่ยนแปลงเทคโนโลยีของวงจรที่ออกแบบไว้ก็เพียงสังเคราะห์วงจรใหม่ก็จะได้วงจรในเทคโนโลยีใหม่ทันที

3.4 การเขียนภาษา HDL

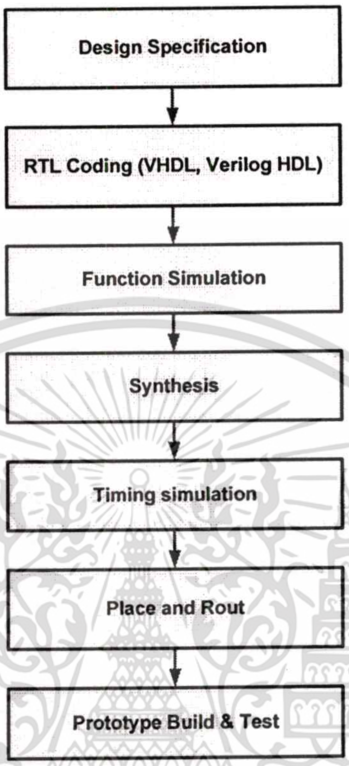
จากที่ได้กล่าวมาแล้วว่าแนวโน้มในปัจจุบันการออกแบบวงจรดิจิตอลขนาดใหญ่จะหันไปสู่การใช้ภาษาระดับสูงมากขึ้นเนื่องจากเป็นภาษาที่มีโครงสร้างและสามารถทำความเข้าใจได้ง่าย อีกทั้งยังสามารถนำกลับมาใช้งานได้อีก และยังสามารถรองรับได้กับหลายเทคโนโลยีอีกด้วย โดยโค้ดที่เขียนขึ้นมานั้นจะไม่ขึ้นอยู่กับเทคโนโลยีนั้น ๆ หรือมีผลเพียงส่วนน้อยเท่านั้น นอกจากนั้นยังมีซอฟต์แวร์ที่มีประสิทธิภาพอีกมากมายสำหรับใช้ในการสังเคราะห์วงจร ในปัจจุบันนี้ภาษาที่ใช้อธิบายฮาร์ดแวร์จะนิยมใช้ได้แก่ภาษา VHDL และ Verilog HDL ซึ่งเป็นมาตรฐานในทางอุตสาหกรรม ซึ่งขั้นตอนในการออกแบบและพัฒนาแสดงดังรูปที่ 3.1

3.4.1 การกำหนดลักษณะเฉพาะในการออกแบบ (Design Specifications)

จากรูปที่ 3.1 แสดงขั้นตอนการออกแบบโดยใช้ภาษา HDL จะเริ่มต้นที่การเขียนโค้ดหลังจากที่ได้มีการออกแบบโครงสร้างไว้เป็นที่เรียบร้อยแล้ว เช่น การแบ่งวงจรทั้งหมดออกเป็นบล็อกย่อย ๆ เพื่อให้ง่ายต่อการออกแบบตามวิธีการ Top-down Design Methodology ซึ่งในแต่ละบล็อกย่อยจะไม่มีความซับซ้อนมากจนเกินไป และง่ายต่อการสังเคราะห์วงจรในภายหลัง จากนั้นจึงทำการเขียนภาษาอธิบายพฤติกรรมของบล็อกเหล่านั้น (Behavioral Description) ด้วยภาษา HDL แล้วจึงนำมารวมเข้าด้วยกันโดยการเรียกใช้ส่วนประกอบ และทำการเชื่อมต่อสัญญาณระหว่างบล็อกย่อย ๆ เข้าด้วยกัน จะเห็นได้ว่าการอธิบายวงจรในระดับนี้จึงเป็นการอธิบายเชิงโครงสร้าง (Structural Description) การเขียนโค้ดภาษา HDL จะมีลักษณะคล้ายกับภาษาระดับสูงอื่น ๆ สามารถใช้โปรแกรมจำพวก Text Editor ทั่ว ๆ ไปได้แต่อย่างไรก็ตามก็มีซอฟต์แวร์หลาย ๆ ตัวที่สนับสนุนการเขียนโค้ดดังกล่าวอยู่หลาย ๆ ตัวได้แก่ ISE ของบริษัท Xilinx, Max plus II ของบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Altera หรือ Moselsim ของบริษัท Model Technology, Inc เป็นต้น ซึ่งซอฟต์แวร์ดังกล่าวจะสามารถสังเคราะห์และคอมไพล์ได้ขึ้นอยู่กับผู้ใช้งานว่าเลือกใช้ FPGA ของผู้ผลิตรายใด



รูปที่ 3.1 ขั้นตอนการออกแบบวงจรด้วยภาษา HDL

3.4.2 การจำลองการทำงานของฟังก์ชัน (Functional simulation)

ขั้นตอนที่สำคัญอีกขั้นตอนหนึ่งในการออกแบบวงจรรวมก็คือ การตรวจสอบว่าแบบจำลอง (Model) ในภาษา HDL ที่ได้สร้างขึ้นนั้นสามารถทำหน้าที่ตามที่เราได้กำหนดไว้หรือไม่ ซึ่งสามารถตรวจสอบได้โดยการจำลองการทำงาน เพื่อดูการทำงานของวงจรที่สร้างขึ้น (Behavioral Simulation) จึงต้องเขียนโมดูลพิเศษขึ้นมา เพื่อใช้ในการตรวจสอบพฤติกรรมการทำงานของโมดูลที่จะใช้งาน (Design Under Test) ซึ่งโมดูลที่ใช้ทดสอบนี้เรียกว่า HDL Test bench โดยจะเป็นการกำหนดรูปแบบของสัญญาณขาเข้า (Input Signal) เพื่อที่จะดูการทำงานของโมดูลที่สร้างขึ้นและนำไปเปรียบเทียบกับสัญญาณที่ป้อนให้กับโมดูลว่าการทำงานนั้นเป็นไปตามที่คาดหวังไว้หรือไม่

3.4.3 การสังเคราะห์วงจร (Synthesis circuit)

เมื่อทำการตรวจสอบโมดูลที่สร้างขึ้นเป็นที่แน่ใจแล้วว่า วงจรที่สร้างขึ้นมานั้นสามารถทำงานได้อย่างถูกต้องตามต้องการ ขั้นตอนต่อไปก็คือการนำเอาโค้ดต้นแบบดังกล่าวไปทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเคราะห์วงจร เพื่อให้ได้เน็ตลิสต์ในระดับเกตออกมา เครื่องมือที่ใช้ในการสังเคราะห์โค้ดภาษา HDL (HDL Synthesis Tool) เช่น ซอร์ฟต์แวร์ ISE ของบริษัท Xilinx ซึ่งสามารถสังเคราะห์วงจรสำหรับ FPGA ได้และสามารถเลือกไลบราลีให้เหมาะสมกับชิพที่เลือกใช้งานได้เช่น Vertec, spatan เป็นต้น

3.4.4 การจำลองทางเวลา (Timing simulation)

การจำลองการทำงานในระดับพฤติกรรม หรือหลังจากการสังเคราะห์วงจร แต่ยังไม่ได้ผ่านการเชื่อมวงจรเข้ากับโมดูลชุดอื่น ๆ จะเป็นการตรวจสอบความถูกต้องในระดับสัญญาณลอจิกเท่านั้น แต่ยังไม่ได้ให้รายระเอียดที่แน่นอนเกี่ยวกับเรื่องเวลา (Timing) มากนัก เช่น ความล่าช้าของสัญญาณตามเส้นทางต่าง ๆ ภายในวงจร ระยะคาบของสัญญาณเวลาที่น้อยที่สุดเท่าที่จะสามารถใช้กับวงจรได้ เป็นต้น เพราะจะต้องผ่านขั้นตอนการเชื่อมเส้นทางก่อน ดังนั้นการจำลองการทำงานทางเวลาจึงเป็นการตรวจสอบการทำงานของวงจรอีกครั้ง

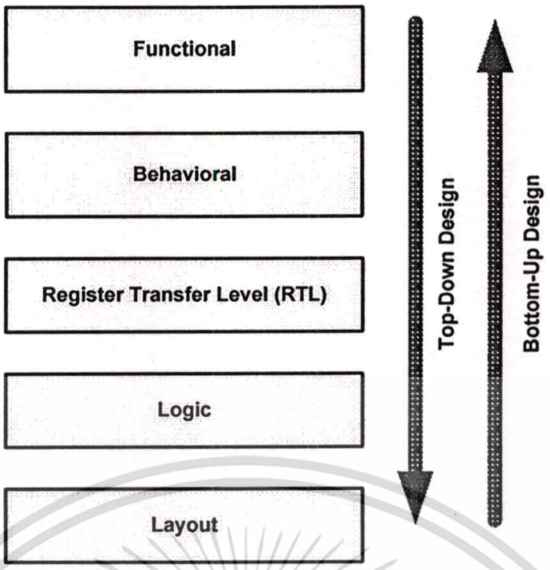
3.4.5 การวาง และการเชื่อมเส้นทาง (Place and Route)

ผลที่ได้จากการสังเคราะห์นั้นจะอยู่ในรูปแบบเน็ตลิสต์ในระดับเกต โดยสามารถนำไปผ่านขั้นตอนการแปลงให้เป็นลอจิกและรวมถึงการวางและการเชื่อมเส้นทางภายในอุปกรณ์ FPGA

3.5 การออกแบบจากบนลงล่างและการออกแบบจากล่างขึ้นบน

ในการออกแบบวงจรบน FPGA จะมีวิธีการหรือกระบวนการในการออกแบบอยู่ 2 วิธีคือ การออกแบบจากบนลงล่าง (Top-Down Design) และการออกแบบจากล่างขึ้นบน (Bottom-Up Design) ดังรูปที่ 3.2 ซึ่งการออกแบบทั้งสองวิธีต่างก็มีข้อดีข้อเสียต่างกัน ในกระบวนการออกแบบจากบนลงล่าง (Top-Down Design) จะมีวิธีการมองจากระดับสูงสุดก่อนว่าวงจรที่จะออกแบบนั้นมีลักษณะอย่างไร ซึ่งเรียกระดับนี้ว่าระดับของฟังก์ชัน (Functional level) จากนั้นจึงค่อยศึกษาลึกลงไปถึงรายละเอียดของแต่ละฟังก์ชันในระดับรีจีสเตอร์ (RTL) ของแต่ละฟังก์ชันและค่อยวิเคราะห์ลึกลงไปในรายระเอียดในระดับลอจิกและในระดับสุดท้าย ซึ่งเป็นระดับล่างสุดในการออกแบบก็คือระดับผังภูมิวงจร (Layout)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



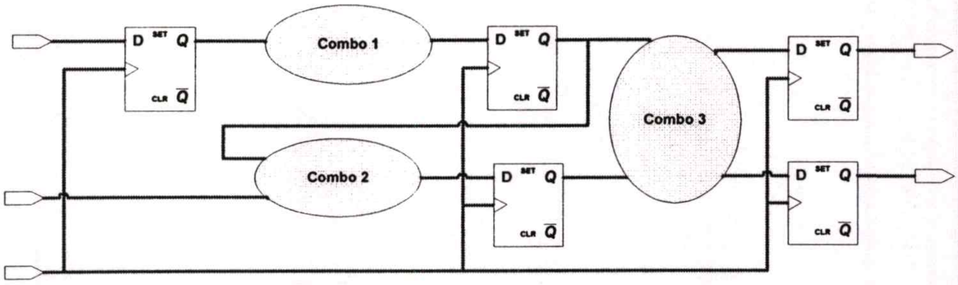
รูปที่ 3.2 กระบวนการออกแบบวงจร (Design Methodology)

ซึ่งผู้ออกแบบจะไม่ทราบรายละเอียดของวงจรที่ออกแบบ จนกว่าจะถึงระดับสุดท้ายในการออกแบบ ซึ่งจะตรงข้ามกับวิธีการออกแบบจากล่างขึ้นบน (Bottom-Up Design) ที่จะเริ่มต้นออกแบบจากระดับล่างสุดก่อนจากนั้นจึงนำแต่ละส่วนมารวมกันเป็นบล็อกในระดับบนเพื่อให้วงจรสามารถทำงานตามหน้าที่ตามฟังก์ชันนั้น ๆ ได้ กระบวนการการออกแบบลักษณะนี้ผู้ออกแบบจะทราบรายระเอียดของวงจรทั้งหมด แต่ข้อเสียของวิธีการนี้ก็คือ ผู้ที่ออกแบบจะต้องเลือกเทคโนโลยีของชิพ FPGA ก่อนที่จะเริ่มต้นออกแบบ ดังนั้นหากจำเป็นต้องเปลี่ยนชิพจำต้องทำการออกแบบใหม่ ซึ่งในกระบวนการออกแบบด้วยภาษาระดับสูงจะเป็นการออกแบบจากบนลงล่าง (Top Dow Design) โดยจะไม่ยึดติดกับเทคโนโลยีในการออกแบบวงจร เพราะการออกแบบด้วยรูปแบบภาษา HDL ซึ่งหากต้องการใช้เทคโนโลยีอะไร ก็เพียงนำไลบรารีของเทคโนโลยีมาสังเคราะห์เป็นโมเดลของวงจรในระดับล่างต่อไป

3.5.1 วงจรในระดับรีจีสเตอร์ (Register Transfer Level)

RTL เป็นการเขียนบรรยายเชิงพฤติกรรมของวงจรในระดับของรีจีสเตอร์ และวงจรคอมบิเนชั่น การออกแบบในระดับนี้ ผู้ออกแบบต้องคำนึงถึงรูปแบบการประมวลผลไทม์มิ่งการเชื่อมต่อและการเข้าจังหวะระหว่างกลุ่มวงจรคอมบิเนชันและรีจีสเตอร์ โดยจะไม่พิจารณาถึงพฤติกรรมวงจรในระดับลอจิกหรือระดับทรานซิสเตอร์โดยแสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ไดอะแกรมโครงสร้างของ Register Transfer Level (RTL)

ในการออกแบบวงจรดิจิตอลด้วย HDL ส่วนใหญ่ จะทำการเขียนบรรยายพฤติกรรมวงจรในระดับ RTL เนื่องจากมีความสะดวกในการออกแบบ และให้วงจรที่มีประสิทธิภาพมากกว่าการเขียนบรรยายในระดับลอจิก ซึ่งใช้เวลาในการออกแบบนานและหาข้อผิดพลาดลำบาก แต่วิธีการเขียนบรรยายพฤติกรรมนั้นก็ใช่ว่าจะให้ผลดีที่สุด แต่ด้วยความสะดวกในขั้นตอนการสังเคราะห์วงจร การวิเคราะห์เวลา (Timing Analysis) ซึ่งเป็นกระบวนการวิเคราะและแก้ปัญหาของค่าเวลาในวงจร รวมไปจนถึงการที่วงจรที่ถูกแยกออกเป็นบล็อกย่อย ๆ (Partitioning) เพื่อให้การสังเคราะห์วงจรได้ดีที่สุด และเข้าใจได้ง่าย ซึ่งการเขียนบรรยายพฤติกรรมจึงเป็นที่นิยมในงานที่มีความซับซ้อนสูง

3.6 ภาษาเวริล็อก (Verilog HDL)

ภาษาเวริล็อก (Verilog HDL) เป็นภาษาที่ใช้อธิบายฮาดร์แวร์ (Hardware Description Language: HDL) [17] ซึ่งใช้ในการออกแบบวงจรระบบดิจิตอล ภาษาเวริล็อกนั้นเป็นหนึ่งในสองภาษาอธิบายฮาดร์แวร์ที่นิยมใช้กันมากในการออกแบบวงจรระบบดิจิตอลในปัจจุบัน ซึ่งได้แก่ ภาษา วีเอชดีแอล (Very High Speed Integrated Circuit Hardware Description Language: VHDL) และภาษาเวริล็อก เนื่องจากภาษาเวริล็อกมีลักษณะคล้ายภาษาซี จึงทำให้วิศวกรซึ่งส่วนใหญ่นั้นมีความรู้ด้านภาษาซี อยู่แล้วสามารถเรียนรู้และเข้าใจได้ง่าย และโครงสร้างของภาษายังมีความซับซ้อนน้อยกว่าภาษาวีเอชดีแอลอีกด้วย

3.6.1 ประวัติความเป็นมา

ภาษาเวริล็อกนั้นถูกคิดค้นโดย บริษัท Gateway Design Automation ในปี 1981 ซึ่งเหตุการณ์ที่สำคัญและความเป็นมากล่าวเรียงตามลำดับได้ดังต่อไปนี้ [18]

ปี 1983 บริษัท Gateway Design Automation ได้เปิดตัวภาษาเวริล็อกนี้ซึ่งเรียกว่า “Verilog HDL” หรือเรียกง่าย ๆ ว่า “Verilog” พร้อมทั้งส่วนการจำลองการทำงานของภาษาเวริล็อก (Verilog Simulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปี 1985 ภาษาเวริล็อกได้มีการปรับปรุงเพิ่มเติมและได้ออกเวอร์ชันใหม่ของส่วนการจำลองการทำงาน ซึ่งเรียกว่า "Verilog XL"

ปี 1987 ภาษาเวริล็อกเป็นที่นิยมใช้ ขนาดของวงจรที่ออกแบบนั้นเริ่มเกินความสามารถของชุดจำลองการทำงานของ Gateway Design System และบริษัทอื่น ๆ

ปี 1989 บริษัท Gateway Design System ถูกซื้อโดยบริษัท Cadence ซึ่งเป็นบริษัทผู้ผลิตซอร์ฟแวร์ในการออกแบบวงจรรวมชั้นนำของโลกในปัจจุบัน

ปี 1990 บริษัท Cadence ได้แยกภาษาเวริล็อกและส่วนการจำลองการทำงานออกจากกัน โดยได้นำภาษาเวริล็อกเปิดเผยต่อสาธารณะ และในปีนี้ผู้ใช้ภาษาเวริล็อกและบริษัทต่าง ๆ ได้ตั้งสมาคม Open Verilog International (OVI) เป็นผู้กำหนดควบคุมรายละเอียด (specification) ของภาษาเวริล็อก และได้จัดทำคู่มืออ้างอิงสำหรับภาษาเวริล็อกเวอร์ชั่น 1.0 (Verilog 1.0 Reference Manual) ขึ้นมา

ปี 1993 สมาคม OVI ได้ปรับปรุงเพิ่มเติมความสามารถของภาษาเวริล็อก เช่น ความสามารถของอาร์เรย์ และได้ออกคู่มืออ้างอิงสำหรับภาษาเวริล็อกใหม่ เป็นเวอร์ชั่น 2.0 (Verilog 2.0 Reference Manual)

ปี 1995 สมาคม OVI ได้ส่งภาษาเวริล็อกเวอร์ชั่น 2.0 ให้สมาคม IEEE (Institute of Electrical and Electronics Engineers) ซึ่งสมาคม IEEE ได้ปรับปรุงและได้กำหนดเป็นมาตรฐานใหม่ที่เรียกว่า มาตรฐาน IEEE 1394-1995

ในปี 2001 มีปรับปรุงให้เพิ่มความสามารถของภาษาเวริล็อกมาตรฐาน IEEE 1394 ทั้งด้านการสังเคราะห์วงจรและการจำลองการทำงาน อีกทั้งยังปรับปรุงให้ง่ายต่อการใช้งาน ซึ่งสมาคม IEEE กำหนดเป็นมาตรฐานใหม่ที่เรียกว่า มาตรฐาน IEEE 1394-2001

3.6.2 โครงสร้างของภาษาเวริล็อก

ภาษาเวริล็อกนั้นสามารถอธิบายพฤติกรรมของวงจรดิจิตอลได้หลายระดับ เช่น สามารถอธิบายวงจรในระดับการวาด (Layout) รีซิสเตอร์ ทรานซิสเตอร์และการเชื่อมต่อ (Wire) บนชิพไอซี (Integrated Circuit chip) ซึ่งเรียกว่า ระดับการสวิทช์ (Switch level) สามารถอธิบายการเชื่อมต่อของลอจิกเกตและฟลิบฟลอบในวงจรดิจิตอล เรียกว่า ระดับเกต (Gate level) และสามารถอธิบายการส่งข้อมูลระหว่างรีจิสเตอร์ เรียกว่า ระดับ RTL (Register Transfer Level) [16] ซึ่งการออกแบบวงจรดิจิตอลด้วยภาษาภาษาเวริล็อกส่วนใหญ่จะใช้อธิบายพฤติกรรมของวงจรดิจิตอลในระดับ RTL นี้

ภาษาเวริล็อกนั้นมีไวยกรณ์คล้ายภาษาซี เช่น จบประโยคด้วย ; หรือใช้ // เพื่อไม่ให้นำไปใช้ในการคอมไพล์ เป็นต้น โครงสร้างของภาษาเวริล็อกนั้นประกอบด้วยโมดูล (Module) ซึ่งแต่ละโมดูลนั้นสามารถรวมเข้าด้วยกันเป็นโมดูลขนาดใหญ่หรือเรียกว่า ท็อปโมดูล (Top module)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแต่ละโมดูลย่อยนั้นสามารถทำงานพร้อมกัน (Concurrent) ซึ่งโครงสร้างของภาษาเวริล็อก ประกอบด้วยส่วนต่าง ๆ ดังแสดงในตัวอย่างการอธิบายพฤติกรรมของ D-flip flop ได้ดังนี้

```
module dff(clk,rst,din,dout);          // <ชื่อโมดูล> (<ชื่อพอร์ต>)
input din, clk, rst;                   //<ชื่ออินพุต>
output dout;                           //<ชื่อเอาท์พุต>
reg  dout;                                        //<ชื่อสัญญาณ>
always @(posedge clk or posedge rst)              //<รายละเอียด
โปรแกรม>
        if (rst)
                dout <= 1'b0;
        else
                dout <= din;
endmodule                              //<จบโมดูล>
```

3.7 รายละเอียดสถาปัตยกรรมของ FPGA ตระกูล Spartan III

FPGA ตระกูล Spartan III เป็น FPGA ของบริษัท xilinx ที่ใช้เทคโนโลยีในกระบวนการผลิต 90 ไมครอน [19] ที่ถูกออกแบบมาสำหรับผู้ที่ต้องการใช้งาน FPGA ที่มีความจุสูงในราคาที่ถูก โดยทำงานภายใต้สัญญาณนาฬิกาสูงสุดถึง 326 เมกกะเฮิร์ต เหมาะสำหรับงานอุตสาหกรรมโดย Spartan III จะมีความจุเกตเริ่มต้นตั้นแต่ 50,000 เกตไปจนถึง 5,000,000 เกตดังแสดงในตารางที่ 3.1 ซึ่งโครงสร้างภายในของ Spartan III ถูกพัฒนาแบบมาจาก Spartan-IIE โดยทำการเพิ่มความจุของลอจิกเกต และจำนวนอินพุต เอาต์พุตเพิ่มขึ้น ส่งผลให้ประสิทธิภาพการทำงานดีขึ้น อีกทั้งได้มีการรวมเอาโมดูลที่ใช้ในการจัดการสัญญาณนาฬิกา (DCMs) และ โมดูลในการคูณที่มีอยู่ในชิพ FPGA ตระกูล Virtex™ II จำนวนขาอินพุทเอาต์พุตใช้งานของ Spartan III ขึ้นอยู่กับแพคเก็ตของชิพซึ่งจะมีจำนวนขาอินพุตเอาต์พุตตั้งแต่ 63 จนถึง 784 ขาดังแสดงในตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 จำนวนเกตและหน่วยความจำ ต่าง ๆ ภายในชิพ Spartan III แต่ละเบอร์ [19]

Device	System Gates	Logic Cells	CLB Array (one CLB = Four Slices)			Distributed RAM (bits)	Block RAM (bits)	Dedicated Multipliers	DCMs	Maximum User I/O
			Rows	Columns	Total CLBs					
XC3S50	50K	1,728	16	12	192	12K	72K	4	2	124
XC3S200	200K	4,320	24	20	480	30K	216K	12	4	173
XC3S400	400K	8,064	32	28	896	56K	288K	16	4	264
XC3S1000	1M	17,280	48	40	1,920	120K	432K	24	4	391
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565
XC3S4000	4M	62,208	96	72	6,912	432K	1,738K	96	4	712
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	784

ตารางที่ 3.2 จำนวนเกตและหน่วยความจำ ต่าง ๆ ภายในชิพ Spartan III แต่ละเบอร์ [19]

	Available User I/O and Differential (Diff) I/O Pairs															
	XC3S50		XC3S200		XC3S400		XC3S1000		XC3S1500		XC3S2000		XC3S4000		XC3S5000	
	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff
VQ100	63	29	63	29	-	-	-	-	-	-	-	-	-	-	-	-
TQ144	97	46	97	46	97	46	-	-	-	-	-	-	-	-	-	-
PQ208	124	56	141	62	141	62	-	-	-	-	-	-	-	-	-	-
FT256	-	-	173	76	173	76	173	76	-	-	-	-	-	-	-	-
FG203	-	-	-	-	221	100	221	100	221	100	-	-	-	-	-	-
FG456	-	-	-	-	264	116	333	149	333	149	-	-	-	-	-	-
FG676	-	-	-	-	-	-	391	175	487	221	489	221	-	-	-	-
FG900	-	-	-	-	-	-	-	-	-	-	565	270	633	300	633	300
FG1156	-	-	-	-	-	-	-	-	-	-	-	-	712	312	784	344

ชิพ Spartan III ใช้แรงดันทั้งหมดสามชุดได้แก่ VccInt 1.2V VccAux 2.5V และ VccIO 1.2V-3.3V

3.7.1 บล็อกอินพุตเอาต์พุต (Input/Output Block: IOB)

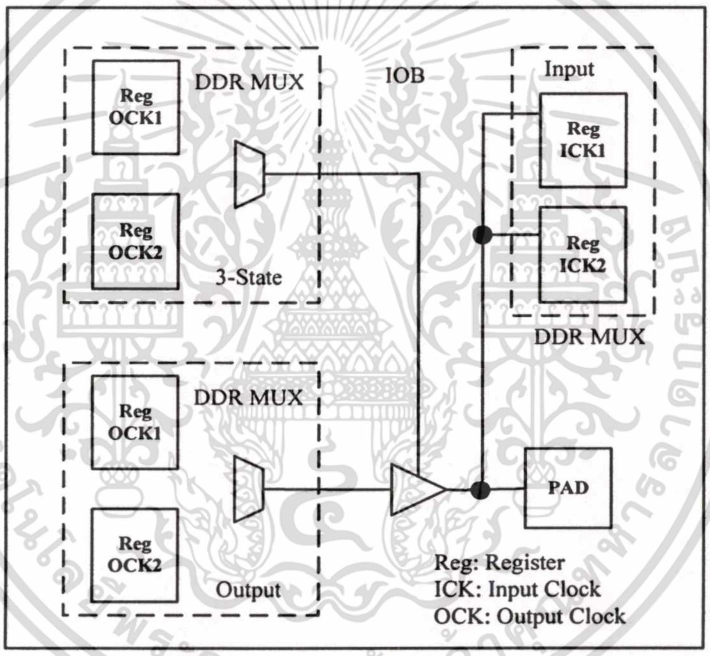
บล็อกอินพุตเอาต์พุต (IOB) เป็นส่วนที่ใช้ติดต่อกับวงจรภายนอก ซึ่งบล็อกอินพุตและเอาต์พุตสามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตแบบ Single Data Rate หรือแบบ Double Data Rate (DDR) ได้ ซึ่งบล็อกอินพุตเอาต์พุตนี้ประกอบด้วย 6 รีจิสเตอร์ (Register) และ 3 DDR Multiplexers ซึ่งแสดงดังรูปที่ 3.4 โดยรีจิสเตอร์นี้สามารถโปรแกรมเป็นดีฟลิปฟลอป (D Flip-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flop) หรือแลตช์ (Latch) ได้ บล็อกอินพุตเอาต์พุตนี้ต่ออยู่กับส่วน Switch Matrix เพื่อเชื่อมต่อกับอุปกรณ์ภายในอื่น ๆ ส่วน DDR Multiplexer ใช้ในการส่งข้อมูลแบบ DDR

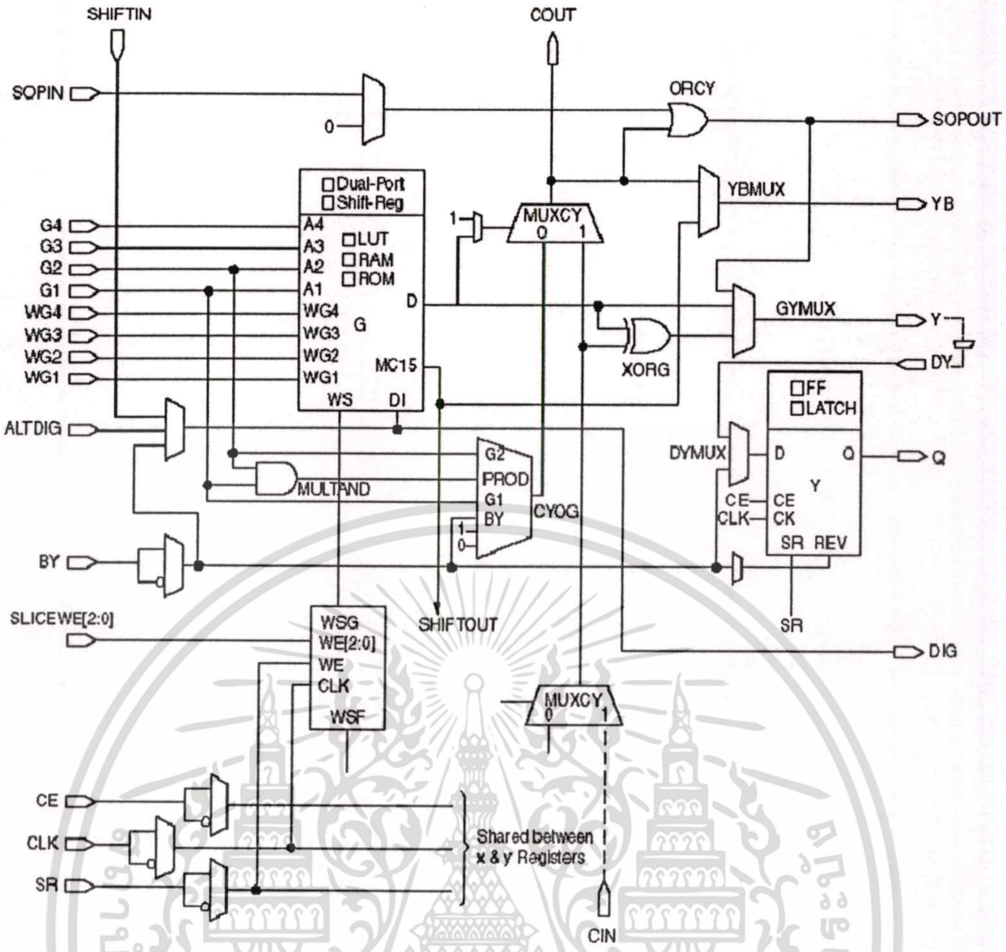
3.7.2 บล็อก Configurable Logic Block (CLB)

บล็อก Configurable Logic Block (CLB) ประกอบด้วย วงจรลอจิกแบบคอมไบเนชัน และแบบซิงโครนัส ซึ่ง CLB ต่ออยู่กับส่วน Switch Matrix เพื่อเชื่อมต่อกับอุปกรณ์ภายในอื่น ๆ โดยในแต่ละ CLB นั้นประกอบด้วย Slices จำนวน 4 Slices ซึ่งแต่ละ Slices มีลักษณะคล้ายกัน ซึ่ง Slice ประกอบด้วยอุปกรณ์ดังนี้ คือ ตัวสร้างฟังก์ชันสี่อินพุต (4-input Function Generators) จำนวน 2 ตัว วงจรโลจิกตัวทด (Carry Logic) วงจรบวก (Arithmetic Logic Gates) วงจรมัลติเพล็กขนาดใหญ่ (Wide Function Multiplexers) และหน่วยความจำ (Storage Elements) จำนวน 2 ตัว



รูปที่ 3.4 โครงสร้างของบล็อกอินพุตเอาต์พุต [20]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โครงสร้างของ Slice [20]

โดยตัวสร้างฟังก์ชันสี่อินพุตสามารถโปรแกรมให้เป็นตัวเปิดตารางสี่อินพุต (4-input Look-up Table) หรือโปรแกรมให้เป็นหน่วยความจำขนาด 16 บิต (16-bits of Distributed Select RAM Memory) หรือโปรแกรมให้ชิพเรจิสเตอร์ 16 บิตที่ปรับเปลี่ยนได้ (16-bit Variable Tap Shift Register) โดยเอาต์พุตของตัวสร้างฟังก์ชันนี้ต่อกับหน่วยความจำ (Storage Elements) ซึ่งเชื่อมต่อกับอุปกรณ์อื่น ๆ โดยหน่วยความจำนี้สามารถโปรแกรมเป็นดีฟลิปฟลอป (D Fip-Flop) หรือแลตช์ (Latch) ก็ได้ ซึ่งรูปที่ 3.5 นั้นแสดงโครงสร้างของ Slice

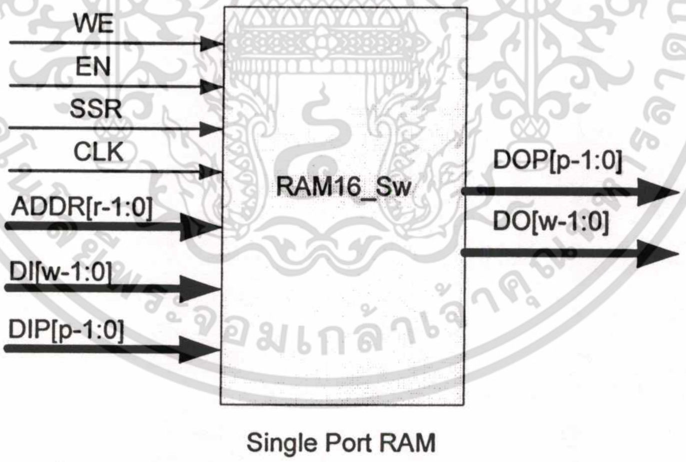
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3 หน่วยความจำขนาด 18 Kb block RAM

18 Kb block RAM เป็นหน่วยความจำที่มีความเร็วสูงมาก (โดยประมาณ) 200 Mhz ที่มีอยู่ในชิพจำนวน 12 สามารถฟอร์มให้เป็น RAM หรือ ROM ที่มีขนาดต่าง ๆ กันได้รวมทั้ง FIFO ด้วย โดยรูปที่ 3 แสดงขนาด RAM Single Port ส่วนในรูปที่ 3.6 แสดงตัวอย่าง RAM แบบ Single Port ขนาดต่าง ๆ ที่สร้างจาก Block RAM แต่ละชุด

ตารางที่ 3.3 หน่วยความจำแบบพอร์ตเดี่ยวที่สร้างจากบล็อกหน่วยความจำถายใน FPGA

Organizations	Memory Depth	Data Width	Parity Width
512 x 36	512	32	4
1K x 18	1024	16	2
2K x 9	2048	8	1
4K x 4	4096	4	-
8K x 2	8192	2	-
16K x 1	16384	1	-



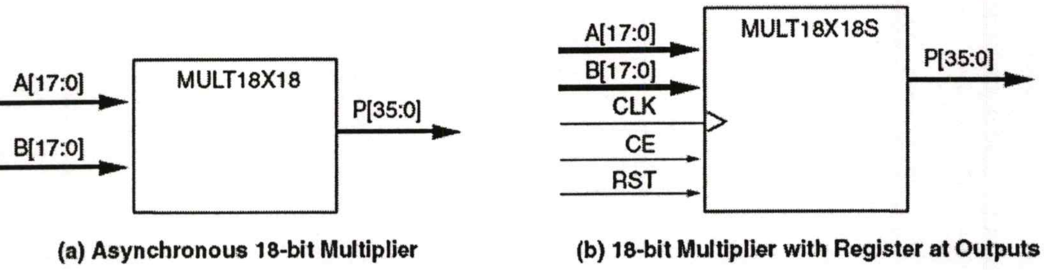
รูปที่ 3.6 โครสร้างหน่วยความจำพอร์ตเดี่ยวบน FPGA [20]

3.7.4 บล็อกวงจรคูณขนาด 18 บิต (18bits x 18bits Hardware multiplier)

บล็อกวงจรคูณขนาด 18 x 18 บิตนี้เป็นวงจรคูณที่สามารถปรับเปลี่ยนขนาดได้ โดยมีขนาดสูงสุดคือ 18 x 18 บิต และสามารถทำการคูณแบบคิดเครื่องหมาย (2's Complement Signed Multiplier)ได้ บล็อกวงจรคูณขนาด 18 x 18 บิตนี้สามารถทำงานอย่างอิสระหรือทำงานร่วมกับบล็อกหน่วยความจำ 18 กิโลบิตในการทำงานที่เรียกว่า "Multiply-Accumulator (MAC)" ซึ่งเป็นฟังก์ชันการทำงานของการประมวลผลสัญญาณดิจิตอล (Digital Signal Processing) โดยรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

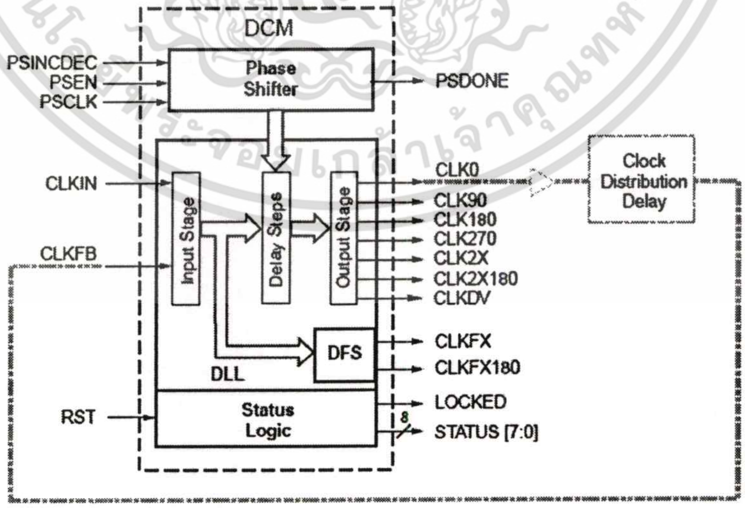
แสดงการเชื่อมต่อของบล็อกหน่วยความจำ 18 กิโลบิตและบล็อกวงจรคูณขนาด 18 x 18 บิต ซึ่งบล็อกทั้งสองจะถูกเชื่อมต่อกับ Switch Matrix



รูปที่ 3.7 โครงสร้างของวงจรคูณขนาด 18 บิต[20]

3.7.5 บล็อกจัดการสัญญาณนาฬิกา (Digital Clock Manager)

Digital Clock Manager (DCM) ในชิพตระกูล Spartan-III เป็นวงจรที่มีความสำคัญมากที่ช่วยจัดการเกี่ยวกับสัญญาณนาฬิกา ซึ่งมีอยู่ในชิพจำนวน 4 ชุด และถือได้ว่า DCM ช่วยทำให้การออกแบบวงจรง่ายขึ้นอย่างมากเนื่องจากสามารถสร้างความถี่ต่าง ๆ ได้อย่างมากมายจากออสซิลเตอร์จากภายนอกเพียงชุดเดียว จึงไม่มีความจำเป็นใด ๆ ที่ต้องหาสัญญาณนาฬิกาจากภายนอกหลาย ๆ แหล่ง อีกต่อไป ไม่เพียงเท่านั้นสัญญาณนาฬิกาดังกล่าวยังซิงค์โครไนซ์กับสัญญาณนาฬิกาจากออสซิลเลเตอร์เดิมอีกด้วย ทำให้สามารถนำไปใช้เป็นตัวกำเนิดสัญญาณนาฬิกาความถี่ต่าง ๆ ได้โดยไม่ต้องใช้ Variable Clock จากภายนอกแต่อย่างใด DCM มีสัญลักษณ์แสดง ดังรูปที่ 3.8 ซึ่ง DCMทำงานในหน้าที่ ดังต่อไปนี้



รูปที่ 3.8 โครงสร้างของ DCM [21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หารความถี่ (Clock Divider) เป็นวงจรหารซึ่งจะให้ความถี่เอาต์พุตเท่ากับความถี่อินพุตหารด้วยตัวเลขดังต่อไปนี้ คือ 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11,12, 13, 14, 15, หรือ 16 ตามลำดับ

- สร้างความถี่สองเท่า (Clock Doublers) เป็นวงจรซึ่งจะให้ความถี่ที่เอาต์พุตจะเป็น 2 เท่าของความถี่ อินพุต

- Digital Frequency Synthesizer (DFS) เป็นวงจรซึ่งสามารถกำหนดให้ความถี่เอาต์พุตเท่ากับผลคูณของความถี่อินพุตกับอัตราส่วนของ M/D โดยที่ M = 2 ถึง 32 และ D = 1 ถึง 32 วงจรนี้นำไปใช้งาน เช่น สร้างวงจรเปลี่ยนจากการส่งข้อมูลแบบขนานเป็นอนุกรม ซึ่งต้องสร้างสัญญาณนาฬิกาสูงกว่าของเดิม เช่น 10 – 11 เท่า เป็นต้น หรืองานอื่น ๆ ที่ต้องใช้วงจรฟรีเควนซีซินธิไซเซอร์

- Delay-Locked Loop (DLL) เป็นวงจรใช้แก้ปัญหาการเลื่อนเฟสในวงจรให้กลับมาตรงตามเฟสที่ต้องการ

- Quadrant Phase Shift เป็นวงจรเลื่อนเฟส 90, 180 และ 270 องศา ตามลำดับ

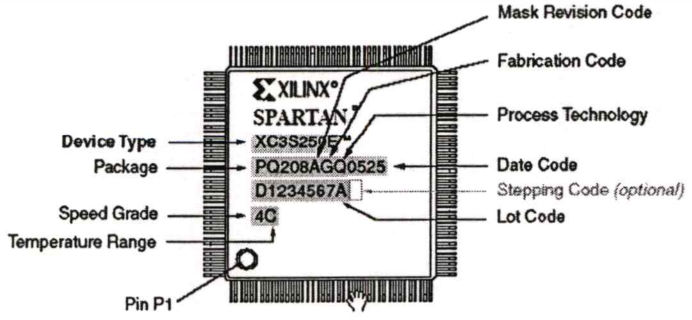
- Fine Phase Shift เป็นวงจรใช้ในการเลื่อนเฟสอย่างละเอียด มีความละเอียดอยู่ที่ 1/ 256 เท่าของคาบความถี่วงจรนี้มีความสำคัญมากเช่นกัน ที่ใช้ในการชดเชยการเลื่อนเฟสที่เกิดขึ้นในวงจร ทำให้การออกแบบง่ายขึ้นอย่างมาก

3.7.6 บล็อกควบคุมอิมพีแดนซ์ (Digitally Controlled Impedance: DCI)

Digitally Controlled Impedance (DCI) ใช้ป้องกันสัญญาณสะท้อนใน PCB โดยการควบคุมเอาต์พุต อิมพีแดนซ์ที่เหมาะสม

3.8 ข้อมูลเกี่ยวกับหมายเลขบนชิพ FPGA

ข้อมูลที่พิมพ์บน ตัวชิพ FPGA เป็นตัวอ้างถึงรายระเอียดต่าง ๆ เกี่ยวกับขบวนการผลิตซึ่งจะมีรายละเอียดเกี่ยวกับความจุ ตัวถัง ความเร็วเป็นต้น โดยมีรายระเอียดแสดงดังรูปที่ 3.9 ดังนี้



รูปที่ 3.9 รหัสโค้ดบนชิพ FPGA [21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 หมายเลขโค้ดบนชิพ FPGA [21]

อุปกรณ์	ความเร็ว		จำนวนขา และชนิดของแพกเกจ		ช่วงอุณหภูมิใช้งาน	
XC3S50	-4	มาตรฐานทั่วไป	VQ100	100-Pin Plastic Very Thin Quad Flat Pack (VQFP)	C	Commercial
XC3S200	-6	มาตรฐานสูง	TQ144	144-Pin Plastic Thin Quad Flat Pack (TQFP)		(0°C – 85°C)
XC3S400			PQ208	208-Pin Plastic Quad Flat Pack (PQFP)	I	Industrial
XC3S1000			FT256	256-Ball Fine-Pitch Thin Ball Grid Array (FTBGA)		(-40°C – 100°C)
XC3S1500			FG320	320-Ball Fine-Pitch Ball Grid Array (FBGA)		
XC3S2000			FG456	456-Ball Fine-Pitch Ball Grid Array (FBGA)		
XC3S4000			FG676	676-Ball Fine-Pitch Ball Grid Array (FBGA)		
XC3S5000			FG900	900-Ball Fine-Pitch Ball Grid Array (FBGA)		
			FG1156	1156-Ball Fine-Pitch Ball Grid Array (FBGA)		

3.9 บทสรุป

ในบทนี้ได้บรรยายถึงประวัติความเป็นมา สถาปัตยกรรม โครงสร้างองค์ประกอบต่าง ๆ ของ FPGA และรูปแบบชนิดต่าง ๆ ในการออกแบบ ขั้นตอนการออกแบบ และภาษาอธิบายพฤติกรรมทางฮาร์ดแวร์ โดยที่กล่าวมาทั้งหมดนั้นเป็นเพียงความรู้พื้นฐานในการที่จะออกแบบวงจรดิจิตอลซึ่งในการที่จะใช้งาน FPGA ให้ได้ประสิทธิภาพสูงสุดนั้นต้องมีประสบการณ์ในการออกแบบ ความเข้าใจในพื้นฐานวงจรดิจิตอล และเทคนิคในการแก้ไขปัญหาต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
วงจรครอสโอเวอร์เน็ตเวอร์ค

4.1 บทนำ

จากที่ได้กล่าวมาแล้วว่าในการแก้ไขปัญหาในการออกแบบระบบเสียงที่ตอบสนองความถี่ได้ไม่ตลอดย่านความถี่สามารถแก้ไขได้โดยการออกแบบระบบขับให้มีลำโพงเพิ่มขึ้น ซึ่งในการออกแบบครอสโอเวอร์เน็ตเวอร์คให้มีคุณภาพดี จะต้องคำนึงถึงคุณสมบัติที่สำคัญดังนี้ [22]

1.) ผลการตอบสนองทางขนาดต่อความถี่มีลักษณะราบเรียบตลอดย่าน

2.) มีผลตอบสนองทางเฟสที่สมมาตร

3.) มีสโลปของจุดตัดความถี่ที่ชันมาก

4.) มีการตอบสนองทางเฟสเป็นเชิงเส้นหรือมีผลตอบสนองกรุ๊ปดีเลย์ที่ราบเรียบ

ในระยะเวลาหลายปีที่ผ่านมาได้มีนักวิจัยหลายกลุ่มได้ทำการค้นคว้าหาทางออกแบบวงจรครอสโอเวอร์ให้มีคุณสมบัติตามข้อกำหนดดังกล่าวเช่น

ในปี 1984 Linkwitz ได้นำเสนอการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่มีผลการตอบสนองทางขนาดรวมตลอดย่านความถี่มีค่าคงที่ [23] โดยใช้วงจรกรองแบบบัตเตอร์เวิร์ทอันดับสองต่อคาสเคสกันซึ่งส่งผลให้ตำแหน่งในการฟังมีผลต่อการฟังลดลงและกำลังงานที่ใช้ในการขับลดลง

ในปี 1985 Saramaki ได้นำเสนอวิธีการออกแบบการสร้างวงจรกรองความถี่จากผลรวมและผลต่างของวงจรกรองทุกย่านความถี่ซึ่งสร้างจากวงจรกรองต้นแบบที่มีอันดับคี่ [24] ซึ่งผลรวมทางขนาดของวงจรกรองทั้งสองมีค่าคงที่ตลอดย่านความถี่

ในปี 1987 Mitra ได้กล่าวถึงการได้ยินของคนเราว่าจะมีความไวต่อการฟังในช่วงรอยต่อตรงจุดตัดความถี่ครอสโอเวอร์ และได้นำเสนอการออกแบบ Crossover network จากผลรวม และผลต่างจากวงจรกรองทุกย่านความถี่สองชุดซึ่งผลที่ได้ทำให้ผลรวมทางขนาดถี่ราบเรียบตลอดย่านความถี่ [25]

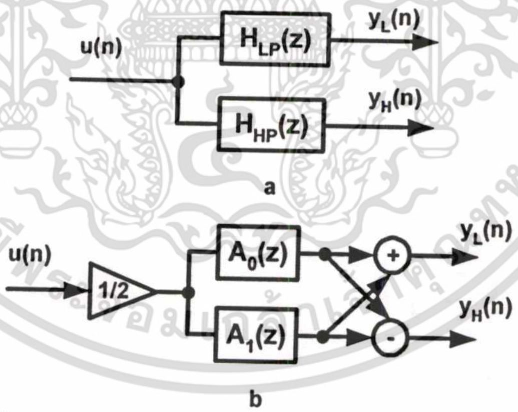
ในปี 1998 Gracia ได้กล่าวถึงสิ่งที่ต้องคำนึงในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คและนำเสนอการออกแบบวงจรกรองดิจิตอลแบบ IIR [22] อันดับ 8 และอันดับ 10 โดยผลที่ได้มีคุณสมบัติที่ดีกว่าที่ Linkwitz ได้นำเสนอไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามจากงานวิจัยที่ผ่านมานั้นก็สามารถออกแบบได้ไม่ครบทุกเงื่อนไขในการออกแบบทั้งหมดซึ่งในวิทยานิพนธ์นี้เป็นการนำเสนอการออกแบบดิจิตอลครอสโอเวอร์เน็ตเวอร์คโดยใช้โครงสร้างวงจรกรองทุกย่านความถี่ผ่านต่อขนาน 2 วงจร (Two parallel all-pass filters) [26, 27] ออกแบบบนวงจรกรองความถี่ต่ำผ่านที่มีเฟสเป็นเชิงเส้นที่นำเสนอโดยเพาเวลล์ และ ชุล [28] และใช้การจำกัดความยาวของผลตอบสนองต่ออิมพัลซ์ [29] ซึ่งผลที่ได้ทำให้เฟสเป็นเชิงเส้นและผลตอบสนองของ กรุ๊ปดีเลย์ราบเรียบตลอดช่วงย่านความถี่ของออดิโอ ผลรวมทางขนาดต่อความถี่มีค่าคงที่ตลอดย่านความถี่ และระบบดังกล่าวไม่มีผลต่อเวลา (Linear Time Invariant) ซึ่งมีวิธีการออกแบบดังนี้

4.2 วงจรครอสโอเวอร์เน็ตเวอร์คจากวงจรกรองทุกความถี่ผ่านต่อขนาน 2 วงจร

วงจรกรองความถี่จากโครงสร้างของวงจรกรองทุกความถี่ต่อขนานกันได้นำเสนอโดย Mitra [26, 27] ซึ่งออกแบบจากวงจรกรองความถี่ต้นแบบไม่ว่าจะเป็นแบบ elliptic cheavbechave หรือ butterwort สามารถนำมาแยกส่วนประกอบออกมาเป็น 2 ฟังก์ชันโดยใช้วิธี pole interacting prototype [26] ซึ่งฟังก์ชันที่ได้เป็น all-pass ฟังก์ชันต่อขนานกันโดยที่ผลรวมของทั้งสองฟังก์ชันยังคงมีคุณสมบัติเช่นเดียวกับวงจรกรองความถี่ต้นแบบทุกประการ



รูปที่ 4.1 โครงสร้างของวงจรครอสโอเวอร์เน็ตเวอร์ค

จากรูปที่ 4.1 ในการออกแบบดิจิตอลครอสโอเวอร์เน็ตเวอร์คแบบสองทางโดยทั่วไปนั้นจะใช้วงจรกรองความถี่สองชุด ได้แก่ วงจรกรองความถี่สูงและวงจรกรองความถี่ต่ำดังรูปที่ 4.1 (a) ซึ่งโครงสร้างดังกล่าวจะมีข้อด้อยในเรื่องของความซับซ้อนและในการคำนวณจุดตัดความถี่ของวงจรเป็นเหตุผลหนึ่งที่ต้องคำนึงในการออกแบบรวมไปจนถึงการสร้างจริง ซึ่งจากที่ได้กล่าวมาแล้วนั้นจะเป็นว่าทั้งวงจรกรองความถี่แบบ FIR และ IIR ต่างก็มีข้อเสียที่แตกต่างกัน เช่นจำนวนของสัมประสิทธิ์ หรือความไววงจรในการปัดเศษสัมประสิทธิ์ เป็นต้น ซึ่ง Mitra ได้กล่าวถึงโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของวงจรกรองความถี่ที่สร้างจากวงจรกรองความถี่ผ่านตลอดต่อขนาน 2 วงจร (Two Parallel All-Pass Filters) [27, 28, 29] ซึ่งโครงสร้างดังกล่าวนั้นจะมีผลต่อการปัดเศษของค่าสัมประสิทธิ์ที่ต่ำและมีโครงสร้างที่ไม่ซับซ้อนมากดังแสดงในรูปที่ 4.1 (b)

จากรูปที่4.1(b) จะเห็นได้ว่าสัญญาณที่ด้านเอาต์พุตของทั้งวงจรกรองความถี่ต่ำและวงจรกรองความถี่สูงจะเกิดจากผลรวมและผลต่างของวงจรกรอง $A_0(Z)$ และ $A_1(Z)$ ดังสมการที่ (4.1) และสมการที่ (4.2) ต่อไปนี้

$$H_{HPF} = \frac{1}{2}\left(A_0(Z) + A_1(Z)\right) \tag{4.1}$$

$$H_{HPF} = \frac{1}{2}\left(A_0(Z) - A_1(Z)\right) \tag{4.2}$$

ข้อดีอีกประการหนึ่งของโครงสร้างดังกล่าวคือมีคุณสมบัติของผลตอบสนองความถี่ตลอดย่าน มีค่าคงที่ตลอดช่วงความถี่ซึ่งเป็นคุณสมบัติประการแรกในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่ดี

4.3 การออกแบบวงจรกรองครอสโอเวอร์เน็ตเวอร์ค

ในหัวข้อที่ผ่านมาเราได้ทราบถึงเหตุผลและคุณสมบัติโครงสร้างของวงจรกรองทุกความถี่ต่อขนานกันแล้ว ต่อไปเป็นการออกแบบวงจรดังกล่าวโดยในการออกแบบจะกำหนดคุณลักษณะของวงจรกรองความถี่ต่ำเป็นต้นแบบขึ้นมาก่อนโดยมีพารามิเตอร์ต่าง ๆ ได้แก่ ชนิดของโพลิโนเมียล อันดับของวงจรกรองความถี่ (Orders) จุดตัดความถี่ (Cutoff Frequency) ขนาดของลูกคลื่นในช่วงความถี่ผ่าน (Pass band Ripple) และค่าลดทอนในช่วงความถี่หยุด (Stop band Attenuation) โดยในตัวอย่างการจำลองด้วยโปรแกรมได้กำหนดค่าต่าง ๆ ไว้ดังนี้คือ

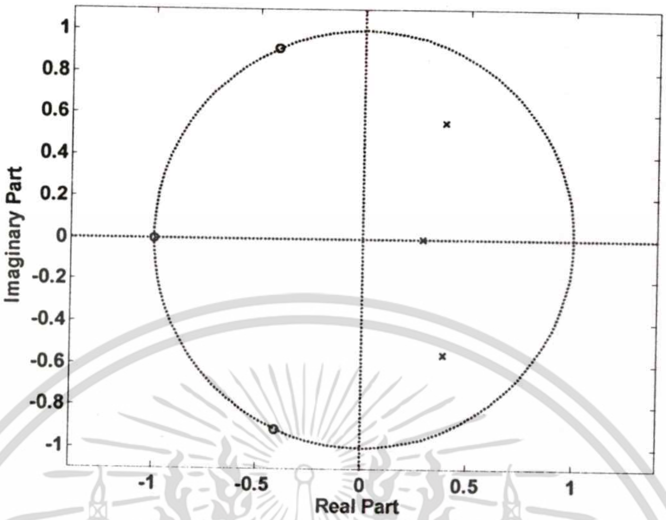
- วงจรกรองต้นแบบเป็น 3rd Elliptic Low Pass Filter
- จุดตัดความถี่ $\omega_n =$ 0.136 (3000 – 14000) เฮิรซ์ท
- ริบเปิ้ลในย่านความถี่ผ่านเท่ากับ 0.001 เดซิเบล
- ค่าลดทอนในช่วงความถี่หยุดเท่ากับ 35 เดซิเบล

และจากข้อกำหนดทางความถี่จากพารามิเตอร์ข้างบนเราสามารถเขียนสมการฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังนี้

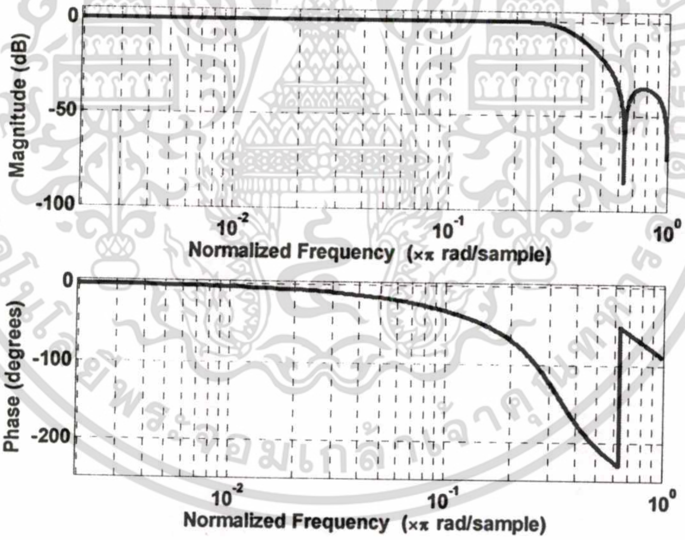
$$H(Z) = \frac{0.08651907 + 0.15752017Z^{-1} + 0.15752017Z^{-2} + 0.08651907Z^{-3}}{1 - 1.06030122Z^{-1} + 0.67884265Z^{-2} - 0.13046294Z^{-3}} \tag{4.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (4.3) เป็นฟังก์ชันถ่ายโอนของวงจรกรองความถี่ต่ำที่ได้จากการกำหนดคุณสมบัติข้างต้นเมื่อนำไปหาตำแหน่งโพล ซีโร ผลตอบสนองขนาดต่อความถี่ และผลตอบสนองเฟสต่อความถี่ดังแสดงในรูปที่ 4.2 และรูปที่ 4.3 ตามลำดับ



รูปที่ 4.2 ตำแหน่งของโพลและซีโรของวงจรกรองความถี่ต่ำต้นแบบ



รูปที่ 4.3 ผลการตอบสนองทางขนาดต่อความถี่ (บน) และผลตอบสนองเฟสต่อความถี่ (ล่าง) ของวงจรกรองความถี่ต่ำต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การออกแบบวงจรกรองทุกความถี่ผ่านด้วยวิธี Pole interacting prototype

การออกแบบวงจรครอสโอเวอร์จากวงจรกรองทุกความถี่ผ่านขนานกันด้วยวิธีการใช้ Pole interacting prototype [26] วิธีการแยกค่าจากฟังก์ชันถ่ายโอนต้นแบบให้อยู่ในรูปของผลรวมของสองฟังก์ชัน โดยสองฟังชันถ่ายโอนใหม่จะมีคุณสมบัติเป็น all-pass ฟังก์ชันโดยมีขั้นตอนการออกแบบดังนี้

เมื่อนำเอาเทอมส่วนจากสมการที่ (4.3) โดยนำเอาเทอมเศษมาทำการแยกหาค่ารากซึ่งจากสมการดังกล่าวมีอันดับ 3 จะได้รากที่เป็นค่าจริง 1 ค่าและรากที่เป็นคู่จำนวนเชิงซ้อน 1 คู่ ดังสมการที่ (4.4)

$$\begin{aligned}\delta_1 &= 0.3877030 + j0.5546346\\ \delta_2 &= 0.3877030 - j0.5546346\\ \delta_3 &= 0.2848951\end{aligned} \tag{4.4}$$

จากนั้นทำการนำค่ารากที่ได้ไปหาฟังก์ชันถ่ายโอน $A_0(Z)$ และ $A_1(Z)$ ได้จาสมการ

$$A_0(Z) = \frac{-\delta 3 + Z^{-1}}{1 - \delta 3 Z^{-1}} \tag{4.5}$$

$$A_1(Z) = \frac{\delta 1 \cdot \delta 2 - (\delta 1 + \delta 2)Z^{-1} + Z^{-2}}{1 - (\delta 1 + \delta 2)Z^{-1} + (\delta_1 \cdot \delta_2)Z^{-2}} \tag{4.6}$$

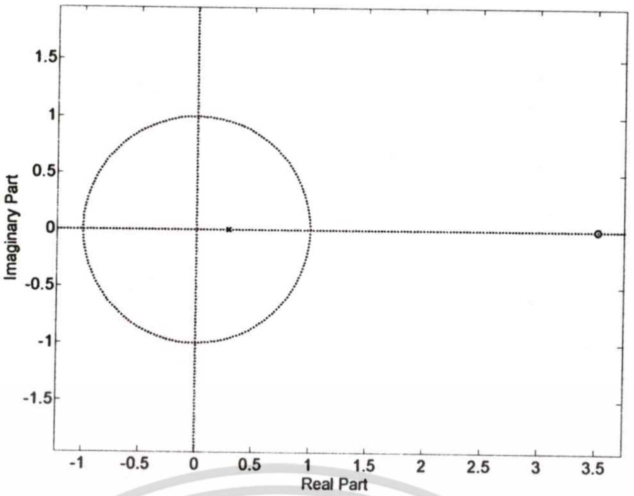
เมื่อนำสมการ (4.4) แทนลงในสมการ (4.5) และ (4.6) จะได้ฟังก์ชันถ่ายโอนของ $A_0(Z)$ และ $A_1(Z)$ ตามสมการที่ (4.7) และ (4.8) ตามลำดับ

$$A_0(Z) = \frac{-0.2848951 + Z^{-1}}{1 - 0.2848951Z^{-1}} \tag{4.7}$$

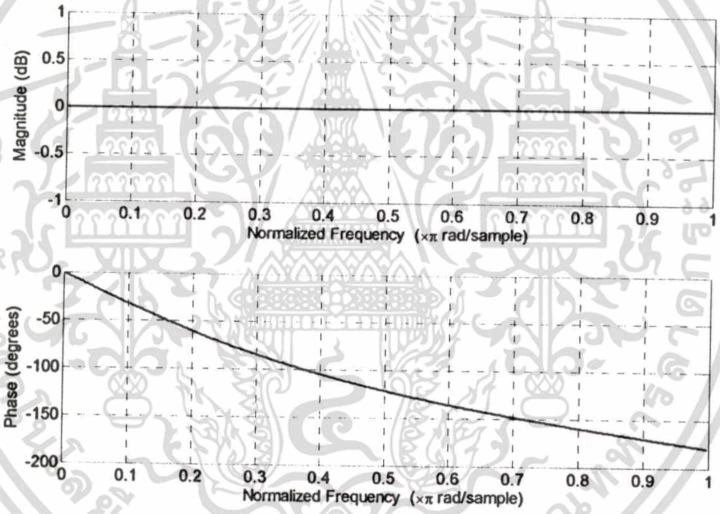
$$A_1(Z) = \frac{0.4579332 - 0.7754061Z^{-1} + Z^{-2}}{1 - 0.7754061Z^{-1} + 0.4579332Z^{-2}} \tag{4.8}$$

จากสมการที่ (4.7) เป็นฟังก์ชันถ่ายโอนของวงจรกรองทุกความถี่ผ่านอันดับหนึ่งโดยมีตำแหน่งของโพล ซีโรดังรูปที่ 4.4 และผลตอบสนองขนาดและเฟสดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



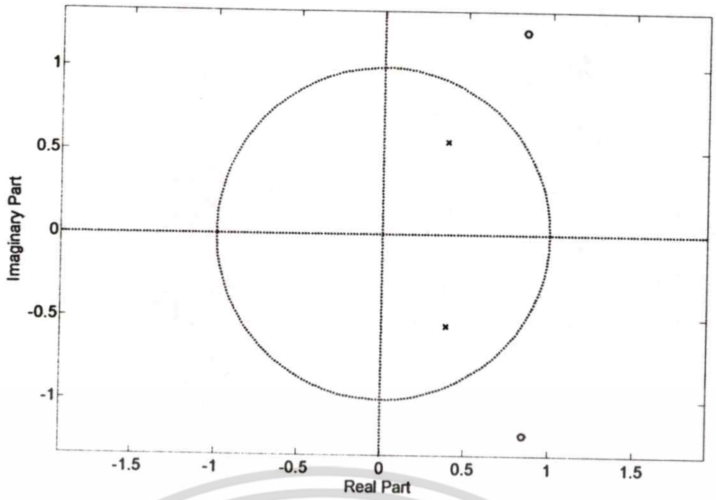
รูปที่ 4.4 ตำแหน่งของโพลและซีโรของฟังก์ชันถ่ายโอน $A_0(Z)$



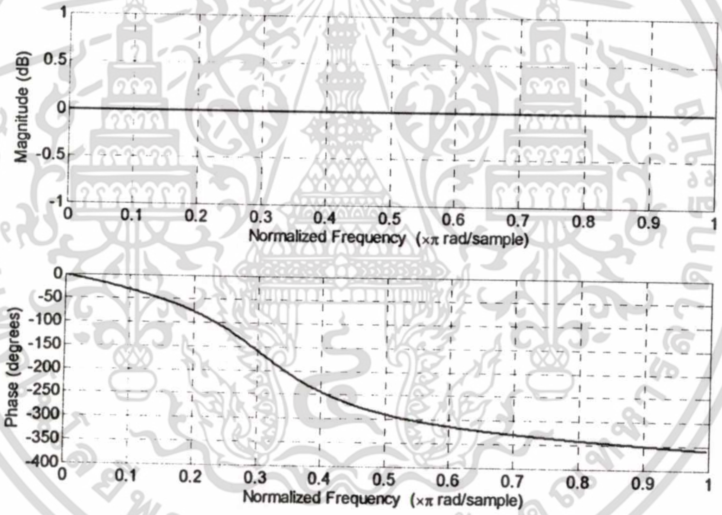
รูปที่ 4.5 การตอบสนองทางขนาดต่อความถี่ (บน) และผลตอบสนองเฟสต่อความถี่ (ล่าง) ของ ของฟังก์ชันถ่ายโอน $A_0(Z)$

จากสมการที่ (4.8) เป็นฟังก์ชันถ่ายโอนของวงจรกรองทุกความถี่ผ่านอันดับหนึ่งโดยมีตำแหน่งของโพล ซีโรดังรูปที่ 4.6 และผลตอบสนองขนาดและเฟสดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ตำแหน่งของโพลและซีโรของฟังก์ชันถ่ายโอน $A_1(Z)$



รูปที่ 4.7 การตอบสนองทางขนาดต่อความถี่ (บน) และผลตอบสนองเฟสต่อความถี่ (ล่าง) ของของฟังก์ชันถ่ายโอน $A_1(Z)$

เมื่อแทนสมการที่ 4.7 และ 4.8 ลงในสมการ 4.1 จะได้สมการฟังก์ชันถ่ายโอนในโดเมนของ Z ซึ่งตรงกับวงจรกรองความถี่ต่ำดังนี้

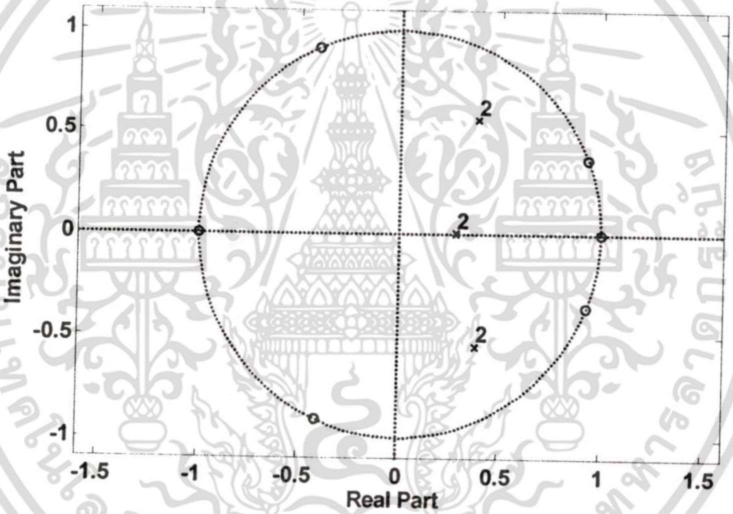
$$H_{LP}(Z)=\frac{0.0865190700+0.1575201758Z^{-1}+0.1575201758Z^{-2}+0.0865190700Z^{-3}}{1-1.0603012208Z^{-1}+0.6788426531Z^{-2}-0.1304629405Z^{-3}} \tag{4.9}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแทนสมการที่ 4.7 และ 4.8 ลงในสมการ 4.2 ก็จะได้สมการฟังก์ชันถ่ายโอนในโดเมนของ Z ซึ่งตรงกับวงจรกรองความถี่สูงดังนี้

$$H_{HP}(Z)=\frac{-0.3714141758+1.0633892314Z^{-1}-1.0633892314Z^{-2}+0.3714141758Z^{-3}}{1-1.0603012208Z^{-1}+0.6788426531Z^{-2}-0.1304629405Z^{-3}} \quad (4.10)$$

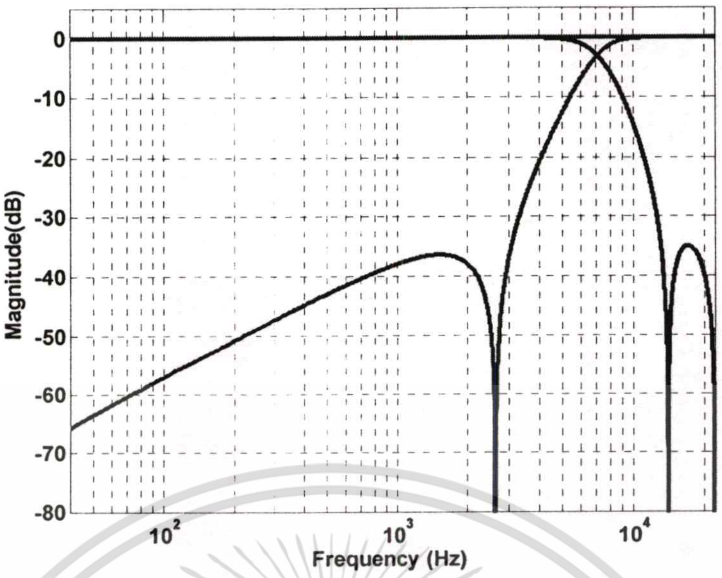
จากสมการที่ 4.9 และ 4.10 จะเห็นได้ว่าค่าของเทอมส่วนของทั้งสองสมการมีค่าเท่ากันซึ่งเมื่อนำไปพล๊อตตำแหน่งของโพลและซีโรของทั้งสองสมการจะเห็นว่าตำแหน่งของโพลนั้นมีตำแหน่งเดียวกัน (Double) และตำแหน่งของซีโรของทั้งสองสมการจะอยู่ตรงกันข้ามกันดังแสดงในรูปที่ 4.8 ซึ่งเรียกว่า mirror image polynomial โดยวงจรกรองความถี่สูงที่ได้จะมีคุณสมบัติตรงกันข้ามกับวงจรกรองความถี่ต่ำต้นแบบ



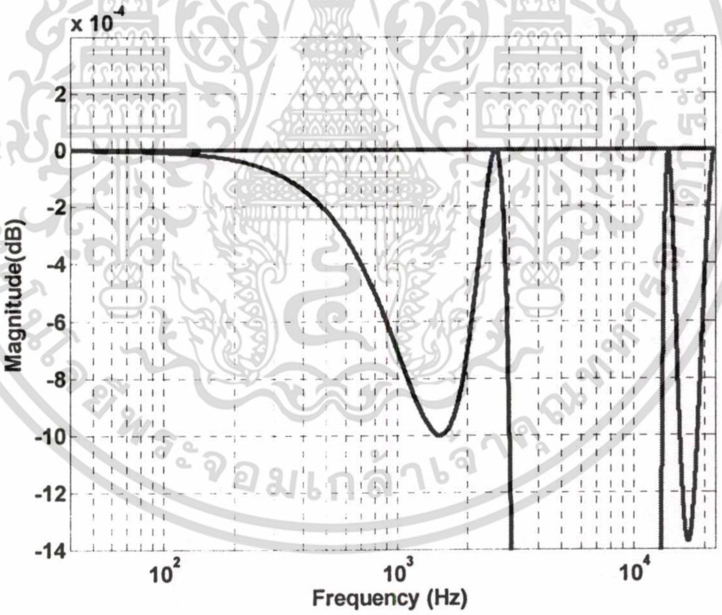
รูปที่ 4.8 ตำแหน่งของโพลและซีโร ของ $H_{LP}(Z)$ และ $H_{HP}(Z)$ ซึ่งตำแหน่งของโพลจะอยู่ตำแหน่งเดียวกัน (double)

เมื่อพิจารณาเฉพาะผลตอบสนองทางขนาดต่อความถี่จากสมการที่ (4.9) และ (4.10) ซึ่งเป็นวงจรกรองความถี่ต่ำและวงจรกรองความถี่สูงที่มีคุณสมบัติเช่นเดียวกับวงจรกรองความถี่ต้นแบบมีค่าลดทอนในช่วงความถี่หยุดเท่ากับ 35 dB และเมื่อหาผลตอบสนองทางขนาดรวมของทั้งสองวงจรนั้นจะมีค่าตลอดย่านความถี่เป็นไปตามคุณสมบัติของ All-pass function ดังแสดงในรูปที่ 4.9 และทำการขยายดูเฉพาะผลตอบสนองความถี่รวมแสดงดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



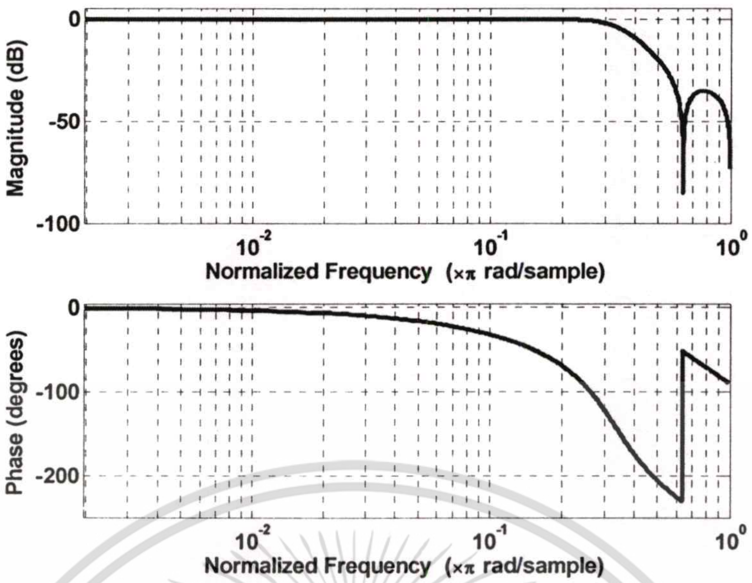
รูปที่ 3.9 ผลตอบสนองของขนาดต่อความถี่ของวงจรครอสโอเวอร์ที่นำเสนอ



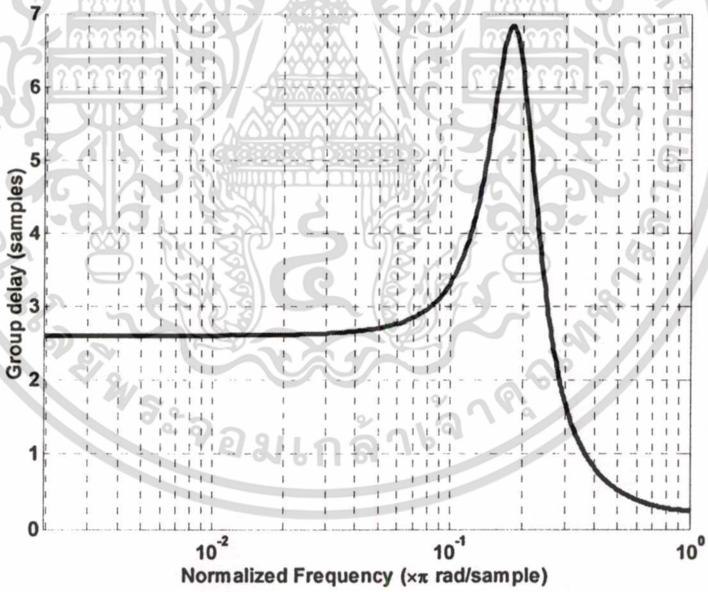
รูปที่ 3.10 รายละเอียดผลรวมการตอบสนองของขนาดต่อความถี่เมื่อทำการขยายจากรูปที่ 3.9

จากการออกแบบวงจรครอสโอเวอร์ที่ใช้วงจรกรองทุกความถี่ที่ต่อแบบขนานจะเห็นได้ว่ามีข้อดีในเรื่องของความซับซ้อนที่ลดลง และผลตอบสนองทางขนาดต่อความถี่ราบเรียบตลอดย่านความถี่ แต่เมื่อนำสมการที่ 4.9 ไปหาผลการตอบสนองทางขนาด ทางเฟสต่อความถี่และผลตอบสนองกรุ๊ปดีเลย์ดังแสดงในรูปที่ 4.11 และ 4.12 ซึ่งจะเห็นได้ว่าผลการตอบสนองทางเฟสไม่เป็นเชิงเส้นและกรุ๊ปดีเลย์ไม่ราบเรียบซึ่งเป็นไปตามลักษณะของวงจรกรองแบบ IIR

เอกสารนี้ ... ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



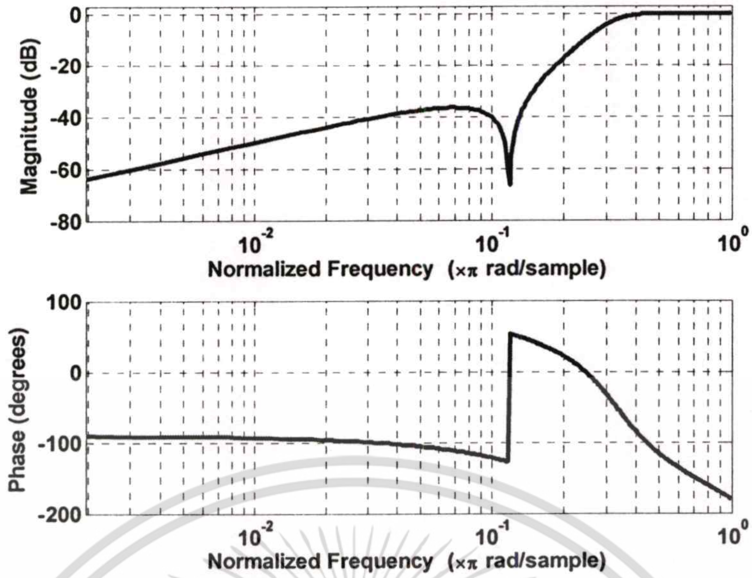
รูปที่ 4.11 ผลการตอบสนองทางขนาด (บน) และเฟสต่อความถี่ (ล่าง) ของวงจรกรองความถี่ต่ำ



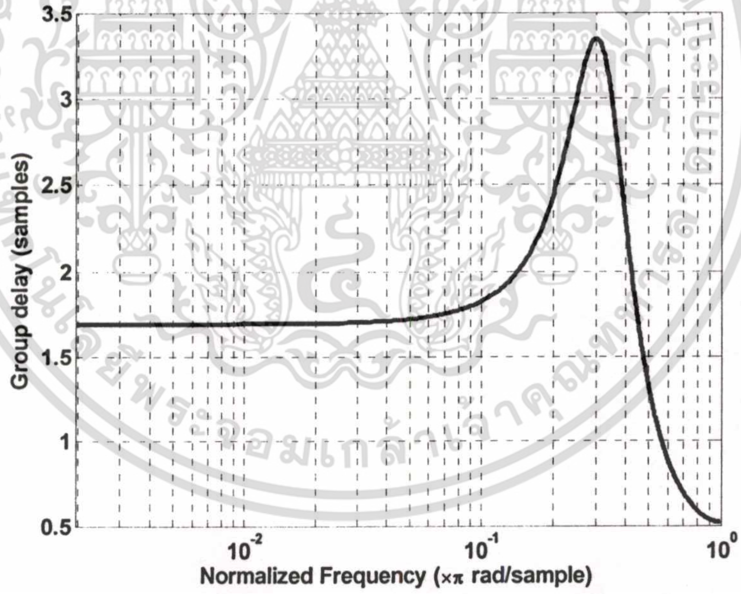
รูปที่ 4.12 ผลตอบสนองของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ

เมื่อนำสมการที่ 4.10 ไปหาผลการตอบสนองทางขนาด ทางเฟสต่อความถี่และผลตอบสนองกรุ๊ปดีเลย์ดังแสดงในรูปที่ 4.13 และ 4.14 ซึ่งจะเห็นได้ว่าผลการตอบสนองทางเฟสไม่เป็นเชิงเส้นและกรุ๊ปดีเลย์ไม่ราบเรียบเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผลการตอบสนองทางขนาด (บน) และเฟสต่อความถี่ (ล่าง) ของวงจรกรองความถี่สูง



รูปที่ 4.14 ผลตอบสนองของกรุ๊ปดีเลย์ของวงจรกรองความถี่สูง

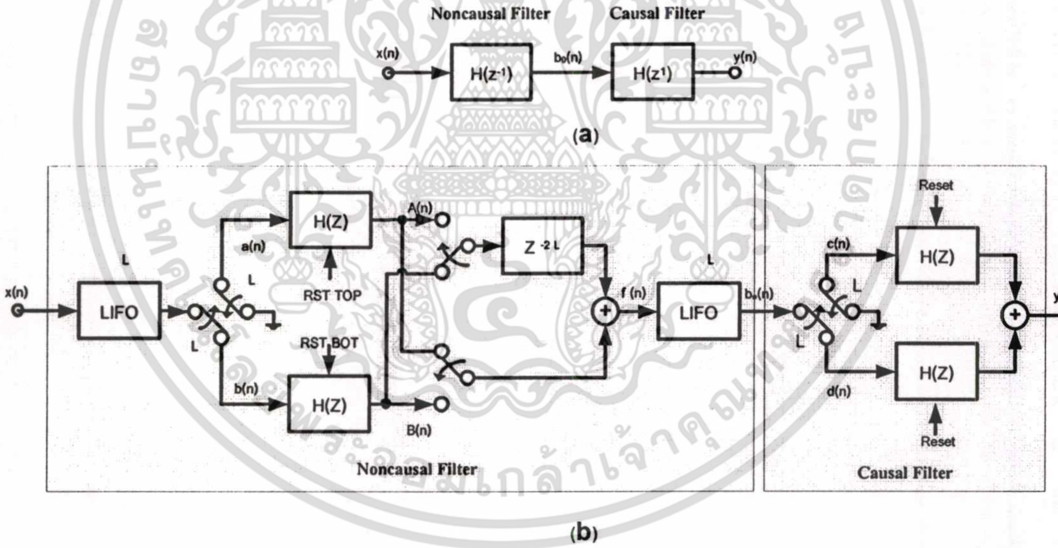
จะเห็นได้ว่าวงจรครอสโอเวอร์เน็ตเวอร์คที่สร้างจากวงจรกรองที่ความถี่ผ่านต่อขนานกันนั้นมีคุณสมบัติของวงจรครอสโอเวอร์เน็ตเวอร์คที่ดีแต่ไม่ครบทั้งหมดเนื่องจากผลการตอบสนองทางเฟสยังไม่เป็นเชิงเส้น และกรุ๊ปดีเลย์ไม่ราบเรียบตามเงื่อนไขในการออกแบบครอสโอเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เน็ตเวอร์คซึ่งเป็นคุณสมบัติเฉพาะของวงจรกรองความถี่แบบ IIR ซึ่งจะมีข้อดีในเรื่องของความชันในช่วงทรานซิชั่นแบนด์

4.5 วงจรกรองความถี่ IIR ที่ให้ผลการตอบสนองทางเฟสเป็นเชิงเส้น

ในการออกแบบ ซึ่งการสร้างตัวกรองความถี่ IIR ที่มีผลตอบสนองทางเฟสเป็นเชิงเส้นและสามารถประมวลผลได้ในเวลาจริงได้ถูกนำเสนอโดย เพาเวลล์และชุน [30] ดังแสดงดังรูปที่ 4.15 (a) เป็นโครงสร้างของวงจรกรองความถี่ต่อคาสเคดกัน ซึ่งระบบนี้ Rabiner [7] ได้อธิบายถึงวิธีที่ให้ประสิทธิภาพสำหรับการสร้างที่สามารถประมวลผลในเวลาจริงโดยใช้การเชื่อมต่อระบบย่อยแบบอนุกรมระหว่างตัวกรองที่ไม่เป็นคอลซอล (non causal filter) $H(z^{-1})$ และตัวกรองที่เป็นคอลซอล (causal filter) $H(z)$ และรูปที่ 4.15(b) เป็นโครงสร้างวงจรกรอง IIR ที่มีผลตอบสนองเฟสเชิงเส้นที่ประมวลผลในเวลาจริงที่ใช้วิธีกรองแบบบวกค่าของผลคูณประสานของแต่ละบล๊อคที่วางเหลื่อมกัน โดยวงจรกรองทั้งสองนี้จะเป็นตัวกรองแบบ IIR $H(z)$ ที่มีฟังก์ชันตามที่ต้องการเช่น เอลลิปติคฟังก์ชัน เป็นต้น



รูปที่ 4.15 โครงสร้างวงจร Linear-phase IIR Filter

$$A(Z) = X(Z^{-1}) \tag{4.11}$$

$$F(Z) = H(Z) \cdot A(Z) = H(Z) \cdot X(z^{-1}) \tag{4.12}$$

$$B(Z) = F(Z^{-1}) = H(Z^{-1}) \cdot X(Z) \tag{4.13}$$

$$Y(Z) = H(Z) \cdot B(Z) = H(Z) \cdot H(Z^{-1}) \cdot X(Z) \tag{4.14}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.15 (b) สัญญาณที่เข้ามาที่ด้านอินพุต $x(n)$ จะถูกแบ่งออกเป็น บล๊อคที่มีการจำกัดความยาวขนาด L แซมเปิ้ลและแต่ละบล็อกจะถูกพลิกกลับในเชิงเวลา (time-reversed) ด้วยวงจร LIFO (Last-in first-out) ตามสมการที่ (4.11) และถูกป้อนผ่านวงจรกรองความถี่ IIR $H(z)$ ทั้งด้านบน (top filter) และด้านล่าง (bottom filter) และผลของการคูณแบบประสาน (convolution) และนำสัญญาณเอาต์พุตของแต่ละบล็อกที่มีการวางเหลื่อมกันนำมารวมกันได้เป็นสัญญาณที่ด้านเอาต์พุทจากวงจรกรองทั้งสอง $A(n)$ และ $B(n)$ และถูกจำกัดความยาวโดยการรีเซ็ทให้มีความยาวเท่ากับ 2L จากนั้นสัญญาณที่ได้จะถูกป้อนผ่านวงจรสวิทช์ซึ่งมีเวลาในการสวิทช์เท่ากับ L วงจรหน่วงเวลาขนาดความยาวของการหน่วงเท่ากับ 2L และนำสัญญาณทั้งสองมาบวกกันตามสมการที่ (4.12) และส่งผ่าน วงจร LIFO อีกครั้งได้สัญญาณเป็น $b_0(n)$ ตามสมการที่ (4.13) ซึ่งสัญญาณที่ได้นี้มาจากวงจรกรองแบบ non causal $H(z^{-1})$ และถูกป้อนเข้าวงจรกรองแบบ causal โดยใช้การแบ่งสัญญาณออกเป็นบล็อกมีความยาว L แซมเปิ้ลและส่งผ่านวงจรกรองความถี่ IIR โดยมีการจำกัดความยาวของสัญญาณเอาต์พุตเท่ากับ 2L เหมือนในวงจรกรองแบบ non causal และนำสัญญาณที่ได้จากวงจรกรองทั้งสองมาบวกกันได้เป็นสัญญาณ $y(n)$ ตามสมการที่ (4.14) โดยระบบทั้งหมดจะมีการหน่วงสัญญาณไป 3L+1 แซมเปิ้ล ซึ่งเป็นข้อเสียที่สำคัญอันหนึ่งที่ได้จากระบบของเพาเวลล์และชุนและนอกจากนี้ข้อเสียที่สำคัญอีกข้อหนึ่งคือระบบนี้เป็นระบบเชิงเส้นแต่แปรเปลี่ยนตามเวลาและเป็นสาเหตุที่ทำให้เกิดการแปรผันของผลตอบสนองของกรุ๊ปดีเลย์และผลตอบสนองทางเฟสก็จะไม่เป็นเชิงเส้น

4.6 การจำกัดความยาวของผลตอบสนองต่ออิมพัลซ์ด้วยตัวกรองค่าเรซิดิวซ์

ตัวกรองค่าเรซิดิวซ์ [31] คือวงจรกรองความถี่ IIR ที่ใช้โครงสร้างแบบ direct form II ซึ่งใช้ประมาณผลตอบสนองอนันต์ต่อสัญญาณอิมพัลซ์ที่มีความยาวหลังจาก L แซมเปิ้ลเป็นต้นไป โดยมีอันดับใด ๆ ที่มากกว่าอันดับ 1 (n=1) และมีอันดับเป็นเลขคู่โดยสามารถสร้างได้จากการนำค่ารากของโพลของวงจรกรองความถี่ IIR ($H(z)$) ซึ่งอยู่ในลักษณะเป็นคู่เชิงซ้อน (complex pole) โดยเลือกจากคู่ที่มีตำแหน่งการวางใกล้เคียงกับวงกลมหนึ่งหน่วยที่มากที่สุดและผลของสัญญาณที่ได้จากตัวกรองค่าเรซิดิวซ์จะถูกหน่วงเวลาออกไป L แซมเปิ้ลจากนั้นนำสัญญาณเอาต์พุตที่ได้จากตัวกรองความถี่ IIR มาลบกับสัญญาณที่ได้จากตัวกรองค่าเรซิดิวซ์จะถูกหน่วงเวลาออกไป L แซมเปิ้ล และได้เป็นสัญญาณเอาต์พุตที่มีการจำกัดความยาวหรือเป็นการใช้การตัดความยาวของผลตอบสนองอนันต์ต่อสัญญาณอิมพัลซ์ให้มีค่าจำกัดแทนการรีเซ็ทสัญญาณในระบบของเพาเวลและชุน และการแทนการรีเซ็ทด้วยวิธีการจำกัดความยาวจะเป็นการเปลี่ยนจากการรีเซ็ทให้เป็นศูนย์ที่ค่าแซมเปิ้ลสุดท้ายของทุก ๆ บล็อกจากเอาต์พุตของตัวกรอง IIR ทั้งตัวบนและตัวล่างมาเป็นการทำให้สัญญาณทุกแซมเปิ้ลเป็นศูนย์ที่จุดต่อระหว่างบล็อก และสัญญาณที่เอาต์พุตของตัวกรอง

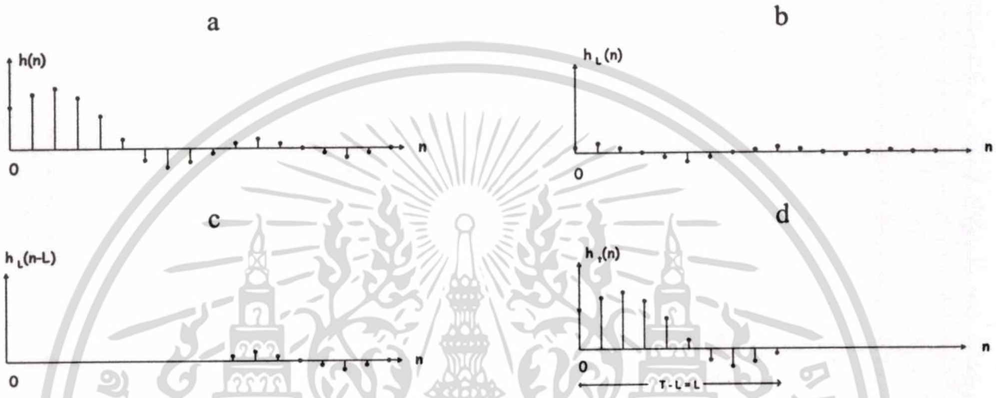
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเรซิดิวซ์ สามารถเขียนแสดงความสัมพันธ์ระหว่างผลตอบสนองอนันต์ต่อสัญญาณอิมพัลซ์ของตัวกรอง IIR กับตัวกรองค่าเรซิดิวซ์เป็นไปตามสมการที่ (4.15) ถึง (4.17)

$$h_L(n-L) = \begin{cases} h(n) & ; \ n > L \\ 0 & ; \ n < L \end{cases} \tag{4.15}$$

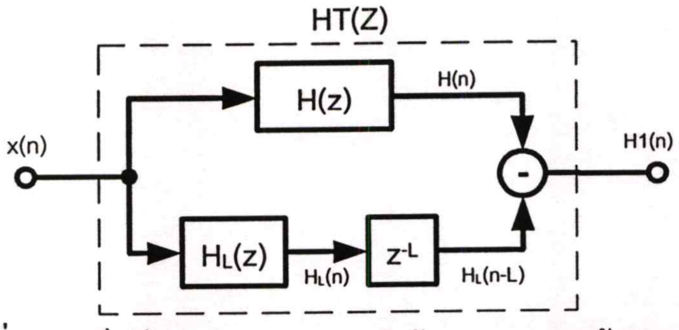
$$H_T(Z) = H(Z) - H_L(Z) \cdot Z^{-L} \tag{4.16}$$

$$H_T(Z) = \sum_{n=0}^{\infty} h(n) \cdot Z^{-n} - \sum_{n=L}^{\infty} h(n) \cdot Z^{-n} = \sum_{n=0}^{L-1} h(n) \cdot Z^{-n} \tag{4.17}$$



รูปที่ 4.16 การจำกัดความยาวของวงจรกรองแบบ IIR โดยใช้วงจรกรองเรซิดิวซ์

ในรูป 4.16(a) เป็นผลตอบสนองต่ออิมพัลซ์ของวงจรกรองความถี่ IIR หรือได้จาก $H(z)$ และในรูป 4.16(b) เป็นผลตอบสนองต่อ อิมพัลซ์ของวงจรกรองเรซิดิวซ์ *HL*(z) ซึ่งจะใช้การประมาณในการสร้างผลตอบสนองต่ออิมพัลซ์ของ $H(z)$ ที่มีความยาวมากกว่า L แซมเปิ้ลขึ้นไป แต่เราต้องการการจำกัดความยาวของผลตอบสนองต่ออิมพัลซ์ไว้ที่ค่า L แซมเปิ้ล ดังนั้นจึงต้องทำการดีเลย์สัญญาณที่ได้จาก *HL*(z) ดังแสดงในรูป 4.16(c) และนำผลที่ได้ไปลบออกจากผลตอบสนองต่ออิมพัลซ์อนันต์ของ *HL*(z) เราจึงได้การจำกัดความยาวของ IIR ที่มีความยาว L ดังแสดงในรูป 4.16(d) และวงจรทั้งหมดที่ใช้ในการจำกัดความยาวของ IIR ดังแสดงไว้ในรูปที่ 4.17

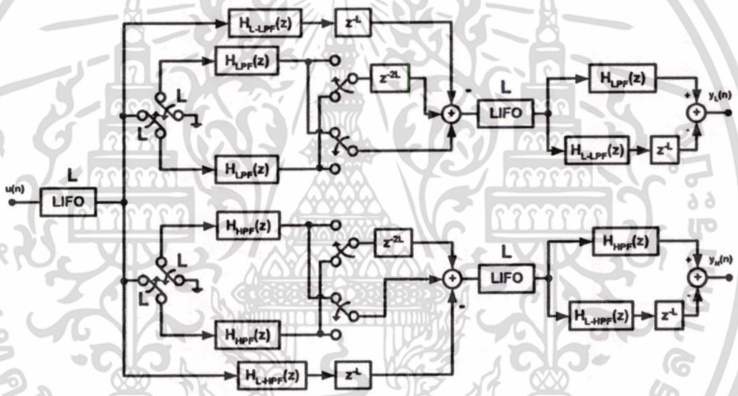


รูปที่ 4.17 บล็อกไดอะแกรมของวงจรจำกัดความยาวของตัวกรอง IIR

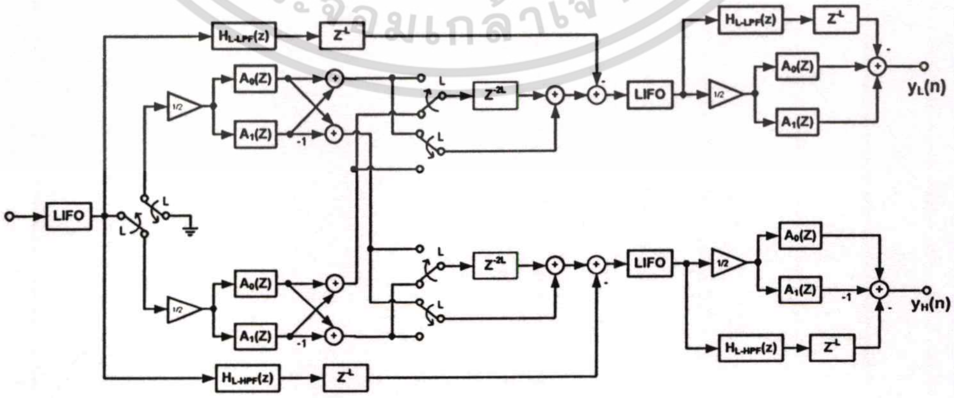
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 วงจรครอสโอเวอร์เน็ตเวอร์คที่ให้ผลการตอบสนองทางเฟสเป็นเชิงเส้น

จากหัวข้อที่ผ่านมาได้กล่าวถึงการออกแบบวงจรที่ผลตอบสนองทางเฟสเป็นเชิงเส้นและเทคนิคการใช้ตัวกรองเรซิดิวส์ ที่ออกแบบมาประมาณผลตอบสนองอิมพัลซ์ที่ความยาว L เป็นต้นไปแทนวิธีการรีเซ็ตซึ่งส่งผลให้ระบบรวมทั้งหมด เปลี่ยนจากระบบที่เป็นเชิงเส้นที่แปรเปลี่ยนตามเวลามาเป็นระบบเชิงเส้นที่ไม่แปรเปลี่ยนตามเวลา (Linear time invariant) ดังนั้นเมื่อนำโครงสร้างดังกล่าวมาออกแบบสร้างเป็นวงจรครอสโอเวอร์ซึ่งจะประกอบด้วยส่วนของวงจรกรองความถี่สูงและวงจรกรองความถี่ต่ำดังแสดงในรูปที่ 4.18 ซึ่งวิธีการดังกล่าวเรียกว่าเป็นการสร้างโดยตรงซึ่งเห็นว่าระบบโดยรวมนั้นยังมีความซับซ้อนอยู่มาก ในรูปที่ 4.19 แสดงโครงสร้างของวงจรครอสโอเวอร์ที่ได้ทำการลดความซับซ้อนโดยนำเอาโครงสร้างของวงจรครอสโอเวอร์จากวงจรทุกย่านความถี่ต่อขนานกันแทนวงจรกรอง $H_{LPF}(z)$ และ $H_{HPF}(z)$ ในส่วนของวงจรที่ non causal



รูปที่ 4.18 การนำเอาวงจรกรองเรซิดิวซ์มาเพิ่มให้กับวงจรครอสโอเวอร์เน็ตเวอร์ค



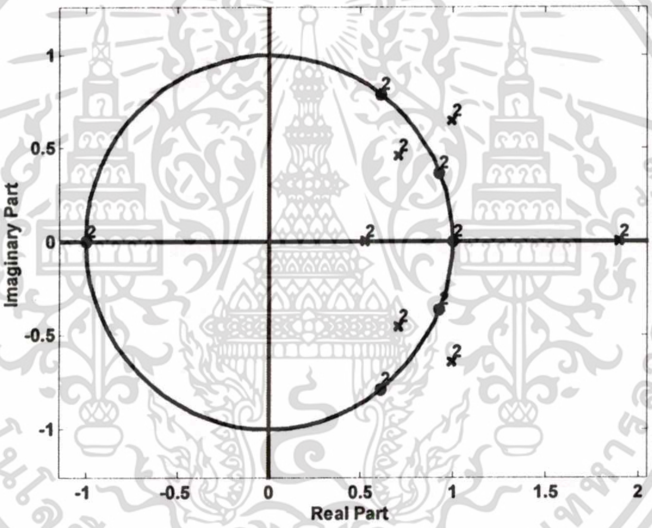
รูปที่ 4.19 วงจรครอสโอเวอร์เน็ตเวอร์คที่ใช้โครงสร้างจากวงจร กรองทุกย่านความถี่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนการจำลองด้วยโปรแกรมตามรูปที่ 4.19 จะกำหนดคุณลักษณะต่าง ๆ ไว้ดังนี้คือ

- วงจรกรองต้นแบบเป็น 3[rd] Elliptic Low Pass Filter
- จุดตัดความถี่ ω_n = 0.136 (3000 – 7000) เฮิรซ์ท
- ริบเปิ้ลในย่านความถี่ผ่านเท่ากับ 0.04 เดซิเบล
- ค่าลดทอนในช่วงความถี่หยุดเท่ากับ 20 เดซิเบล

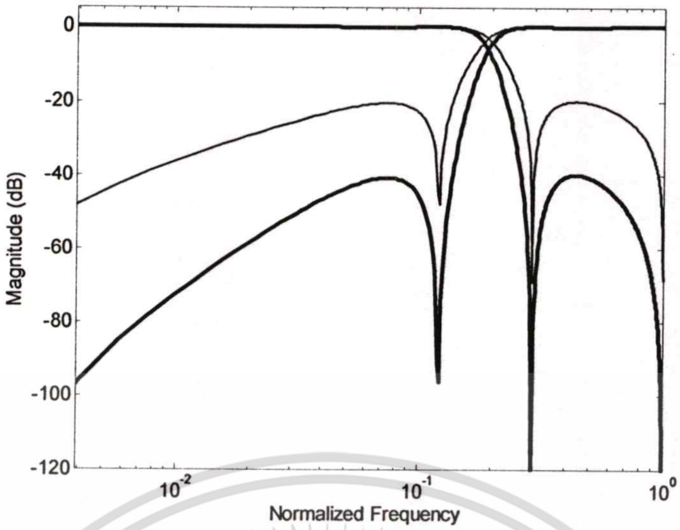
ซึ่งผลที่ได้จากการจำลองด้วยโปรแกรมในรูปที่ 4.20 แสดงตำแหน่งของโพลและซีโรทั้งหมดของระบบซึ่งเมื่อเทียบกับต้นแบบที่ออกแบบไว้ในหัวข้อที่ผ่านมาจะเห็นได้ว่าระบบทั้งหมดประกอบด้วยวงจรกรองแบบ causal (ตำแหน่งของโพลและซีโรที่อยู่ใน Unit circle) และวงจรกรองแบบ non causal (ตำแหน่งของโพลที่อยู่ใน unit circle และซีโรที่อยู่ใน unit circle) ต่ออนุกรมกันอยู่



รูปที่ 4.20 ตำแหน่งของโพลและซีโรทั้งหมดของระบบ

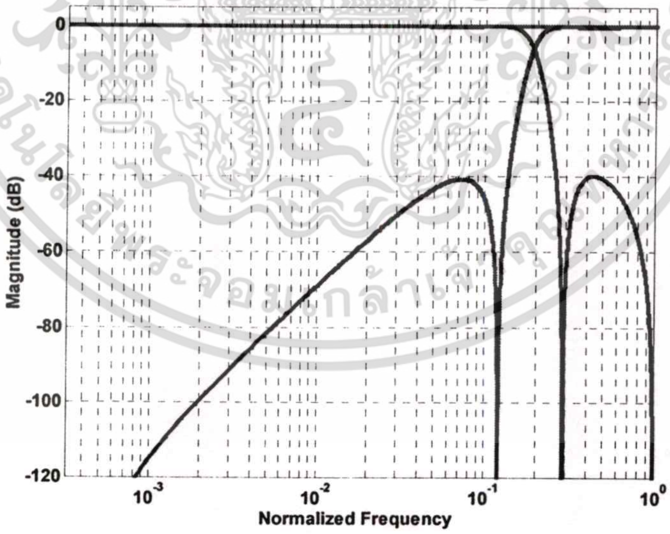
เมื่อพิจารณาถึงผลตอบสนองของขนาดต่อความถี่จะเห็นได้ว่าระบบที่นำเสนอนั้นมีผลการตอบสนองของขนาดต่อความถี่ในช่วงแถบความถี่หยุดดีกว่าต้นแบบไว้ถึงสองเท่าและมีสโลปที่ชันมากกว่าอันเนื่องมาจากโครงสร้างที่ได้ออกแบบไว้ซึ่งเป็นคุณลักษณะที่ต้องการในการออกแบบครอสโอเวอร์เน็ตเวอร์คดังแสดงไว้ในรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



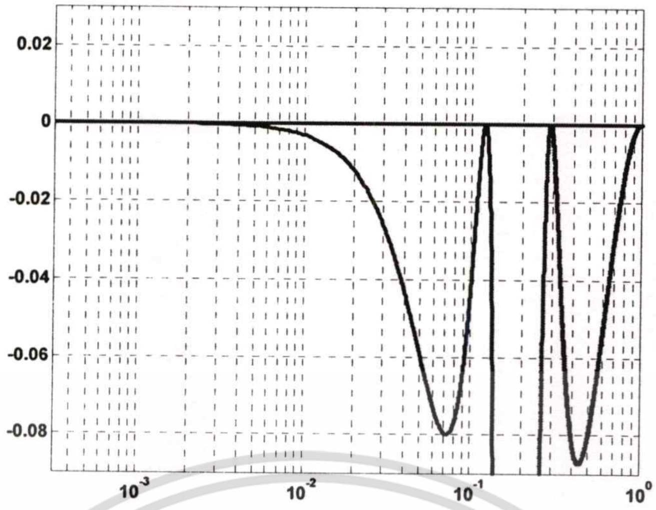
รูปที่ 4.21 ผลตอบสนองขนาดต่อความถี่ต้นแบบ (เส้นบาง) และระบบที่นำเสนอ (เส้นทึบ)

จากรูปที่ 4.22 แสดงถึงผลรวมทางขนาดต่อความถี่ตลอดย่านจะเห็นได้ว่ามีค่าคงที่ตลอดย่านความถี่ (Flat) ซึ่งได้จากคุณสมบัติจากโครงสร้างของ Two parallel All-pass functions และรูปที่ 4.23 แสดงถึงรายละเอียดเฉพาะผลตอบสนองของขนาดซึ่งจะเห็นว่าริบเปิ้ลในช่วงความถี่ผ่านนั้นจะเพิ่มขึ้นจากเดิมสองเท่าเนื่องจากผลของระบบที่นำเสนอ



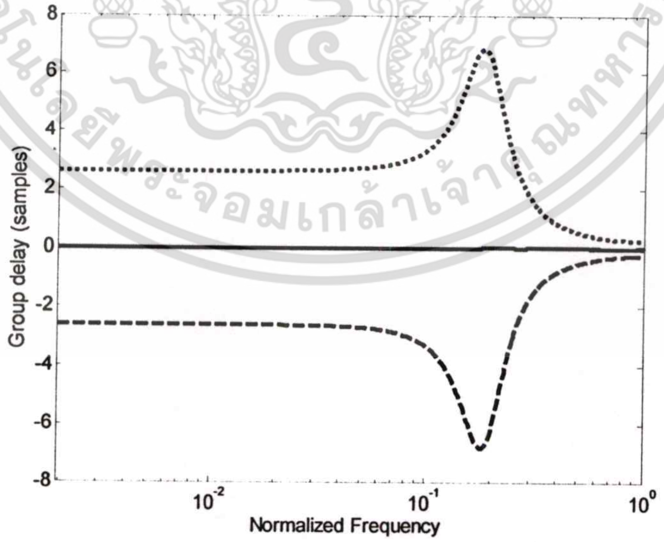
รูปที่ 4.22 ผลรวมทางขนาดต่อความถี่ตลอดย่านความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



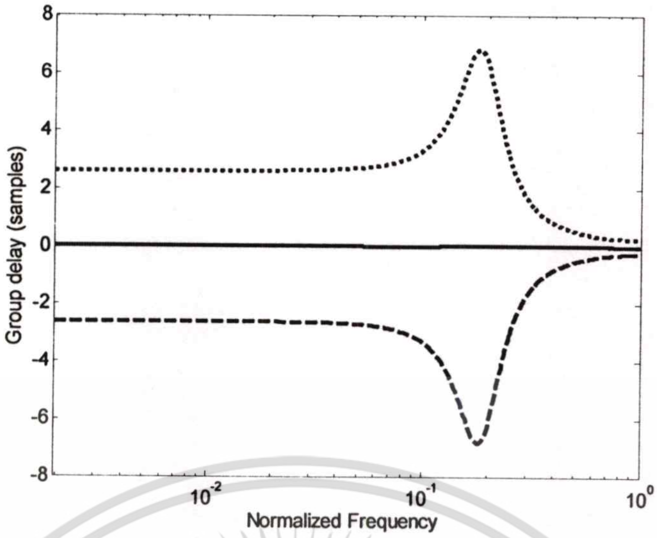
รูปที่ 4.23 รายละเอียดเฉพาะริบเปิ้ลของผลตอบสนองของขนาดในช่วงความถี่ผ่านจากรูป 4.22

จากรูปที่ 4.24 และ รูปที่ 4.25 แสดงถึงผลรวมของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ (เส้นทึบ) ซึ่งเกิดจากผลรวมของกรุ๊ปดีเลย์ของวงจรกรองแบบ causal (เส้นจุด) และวงจรกรองแบบ non causal (เส้นขีด) ที่ต่อคาสเคดกันอยู่ทั้งวงจรกรองความถี่ต่ำและความถี่สูง ซึ่งจะเห็นได้ว่าผลรวมของกรุ๊ปดีเลย์รวมทั้งหมดนั้นมีลักษณะราบเรียบ (flat) เช่นเดียวกับผลตอบสนองทางเฟสต่อความถี่ ดังแสดงในรูปที่ 4.26 และ รูปที่ 4.27

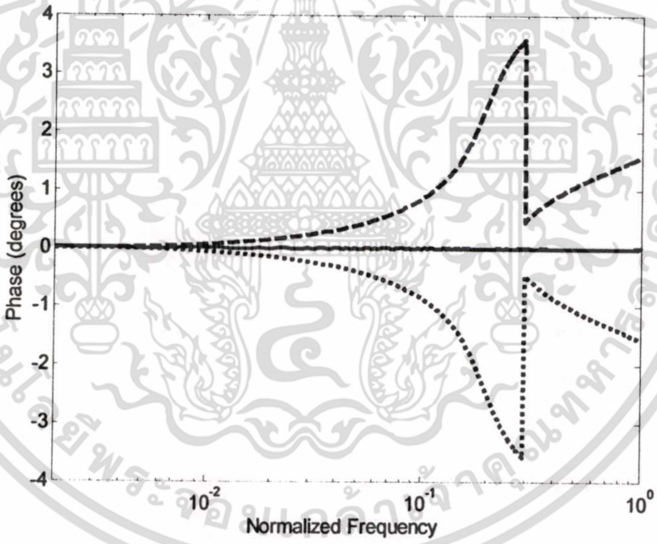


รูปที่3.24 ผลรวมของกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ (เส้นทึบ) กรุ๊ปดีเลย์ของวงจรกรองแบบ causal (เส้นจุด) และวงจรกรองแบบ non causal (เส้นขีด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

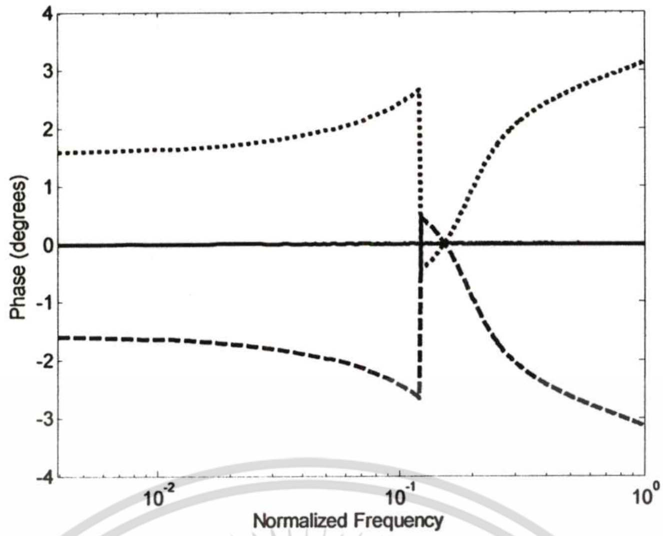


รูปที่ 4.25 ผลรวมของกรุ๊ปดีเลย์ของวงจรกรองความถี่สูง (เส้นทึบ) กรุ๊ปดีเลย์ของวงจรกรองแบบ causal (เส้นจุด) และวงจรกรองแบบ non causal (เส้นขีด)



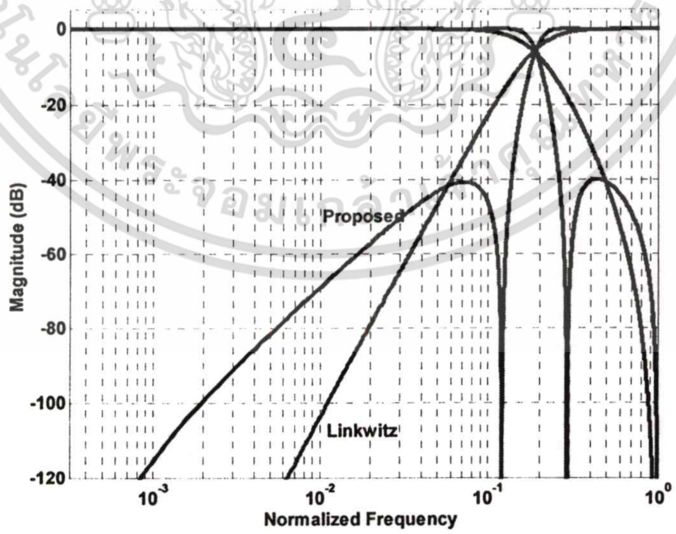
รูปที่ 4.26 ผลรวมของเฟสต่อความถี่ของวงจรกรองความถี่ต่ำ (เส้นทึบ) ซึ่งเกิดจากผลตอบสนองเฟสต่อความถี่ของวงจรกรองแบบ causal (เส้นจุด) และวงจรกรองแบบnon causal (เส้นขีด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



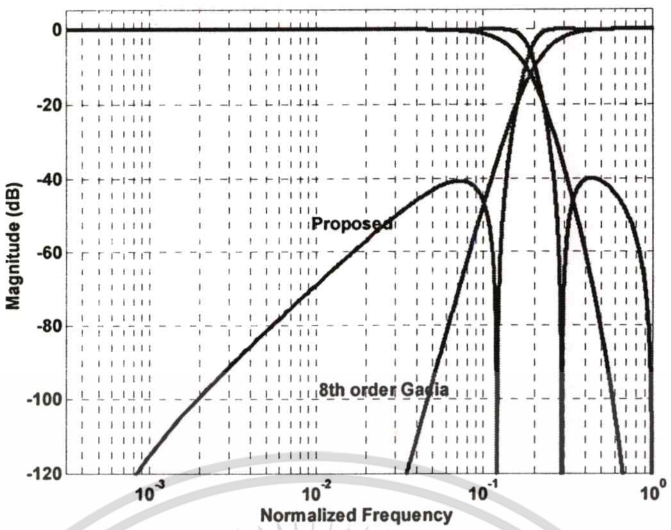
รูปที่ 4.27 ผลรวมของเฟสต่อความถี่ของวงจรกรองความถี่สูง (เส้นทึบ) ซึ่งเกิดจากผลตอบสนองเฟสต่อความถี่ของวงจรกรองแบบcausal (เส้นจุด) และวงจรกรองแบบnon causal (เส้นขีด)

เมื่อนำผลการทดลองดังกล่าวเทียบกับงานวิจัยของ Linkwitz [4.2] ดังรูปที่ 4.28 ซึ่งจะเห็นว่าผลตอบสนองทางขนาดต่อความถี่ที่ออกแบบนั้นมีผลรวมทางขนาดราบเรียบเช่นกันแต่งานวิจัยของ Linkwitz จะมีสโลปความชันที่น้อยกว่า และเมื่อเทียบกับงานวิจัยของ Gacia [4.1] ซึ่งใช้วงจรกรองอันดับ 8 ซึ่งผลที่ได้ก็จะมีสโลปความชันที่น้อยกว่าเช่นกันดังแสดงในรูปที่ 4.29



รูปที่ 4.28 ผลการตอบสนองทางขนาดต่อความถี่เมื่อเทียบกับงานวิจัยของ Linkwitz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 ผลการตอบสนองทางขนาดต่อความถี่ เมื่อเทียบกับงานวิจัยของ Gacia

4.8 บทสรุป

ในบทนี้ได้กล่าวถึงวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางที่สร้างจากวงจรกรองทุกย่านความถี่ผ่านตลอดซึ่งโครงสร้างดังกล่าวมีผลต่อการปัดค่าสัมประสิทธิ์น้อยมากซึ่งเหมาะกับการนำไปสร้างจริงรวมไปจนถึงการออกแบบครอสโอเวอร์ที่ให้เฟสเชิงเส้นด้วยวิธีการนำเอาวงจรกรองความถี่แบบ non causal และcausal ต่อขนานกัน มีผลให้ผลตอบสนองเฟสเป็นเชิงเส้น สโลปจุดตัดความถี่มีความชันเพิ่มขึ้นสองเท่า ซึ่งเป็นคุณสมบัติในการออกแบบครอสโอเวอร์เน็ตเวอร์คที่ดี

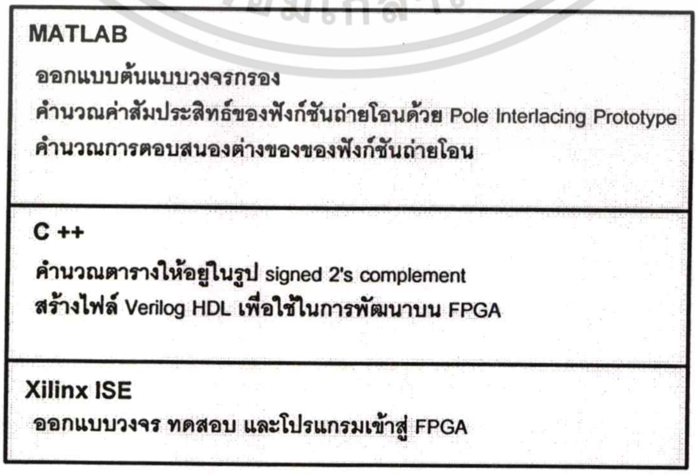
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบ

5.1 บทนำ

ในบทนี้จะเป็นการนำเสนอรายละเอียดขั้นตอนการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คแบบรีเคอร์ซีฟ ต้นแบบเป็นวงจรกรองทุกย่านความถี่ผ่านต่อขนานกัน ในขั้นตอนการออกแบบวงจรนี้จะเริ่มจากการหาฟังก์ชันถ่ายโอนของวงจรกรองต้นแบบที่ออกแบบไว้ก่อน โดยจะเริ่มตั้งแต่การกำหนดอันดับ (Order) ความถี่ตัด (cutoff frequency) ความถี่ในการสุ่ม (sampling frequency) และในรายละเอียดอื่นๆ เช่น ความถี่ที่ขอบแถบผ่าน (pass band edge frequency) ความถี่ที่ขอบแถบหยุด (stop band edge frequency) การกระเพื่อมในแถบผ่าน (pass band ripple) และการลดทอนในแถบหยุด (stop band attenuation) โดยใช้การประมาณค่าด้วยฟังก์ชันอิลลิปติก เพื่อคำนวณหาค่าสัมประสิทธิ์ของฟังก์ชันถ่ายโอนออกมาโดยอยู่ในโดเมนดิจิตอล ซึ่งเหมาะสมสำหรับการคำนวณและการนำไปสร้างบน FPGA

ค่าสัมประสิทธิ์ที่ได้จากการคำนวณจะถูกทำการควอนไตซ์ (Quantization) เพื่อจัดข้อมูลของค่าสัมประสิทธิ์ให้มีการแทนตัวเลขโดยตรงได้ (fix-point) โดยกำหนดให้มีขนาดของข้อมูลเท่ากับ 16 บิต ในขั้นตอนการทำงานต่างๆจะใช้โปรแกรม MATLAB ในการคำนวณหาค่าสัมประสิทธิ์ออกมาจากนั้นจึงนำค่าสัมประสิทธิ์ที่ได้ไปสร้างไฟล์ตารางเปิดดูในรูปแบบของภาษา Verilog HDL ด้วยภาษาซี ซึ่งลดความยุ่งยากในการคำนวณ จากนั้นจึงทำการออกแบบตามโครงสร้างฮาร์ดแวร์ของวงจรครอสโอเวอร์บน FPGA ซึ่งออกแบบและพัฒนาด้วยโปรแกรม Xilinx ISE ซึ่งขั้นตอนในการออกแบบแสดงดังรูปที่ 5.1



รูปที่ 5.1 ขั้นตอนการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA

เอกสารนี้เป็นเอกสาร ... ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การออกแบบวงจรครอสโอเวอร์ด้วยวงจรกรองทุกย่านความถี่ผ่านต่อขนานกัน

ในขั้นตอนการสร้างวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางจะเริ่มต้นออกแบบด้วยการกำหนดค่าคุณสมบัติเริ่มต้นของวงจรขึ้นมาก่อนซึ่งเป็นวงจรกรองความถี่ต่ำโดยมีคุณสมบัติของวงจรต้นแบบเป็นดังนี้

วงจรกรองต้นแบบ	อิลลิปติก อันดับ 3		
ความถี่ในการสุ่ม		48	KHz
แถบความถี่ขอบผ่าน ω_p		3	KHz
แถบความถี่ขอบหยุด ω_s		13	KHz
การกระเพื่อมในแถบผ่าน δ_p		0.002	dB
การลดทอนในแถบหยุด δ_s		35	dB

จากข้อกำหนดดังกล่าวสามารถหาฟังก์ชันถ่ายโอนได้เป็น

$$H(Z)=\frac{0.0650659+0.0951390\cdot Z^{-1}+0.0951390\cdot Z^{-2}+0.0650659\cdot Z^{-3}}{1-1.3724767\cdot Z^{-1}+0.8860779\cdot Z^{-2}-0.1931912\cdot Z^{-3}} \tag{5.1}$$

หลังจากได้ฟังก์ชันถ่ายโอนต้นแบบจากสมการที่ (5.1) สามารถแยกให้อยู่ในรูปแบบของผลรวมและผลต่างของสองฟังก์ชันถ่ายโอนด้วยวิธี Pole interacting prototype จะได้วงจรกรองทุกย่านความถี่ผ่านสองสมการเป็น

$$A_0(Z)=\frac{-0.3792592+Z^{-1}}{1--0.3792592\cdot Z^{-1}} \tag{5.2}$$

และ

$$A_1(Z)=\frac{0.5093910-0.9932175\cdot Z^{-1}+Z^{-2}}{1-0.9932175\cdot Z^{-1}+0.5093910\cdot Z^{-2}} \tag{5.3}$$

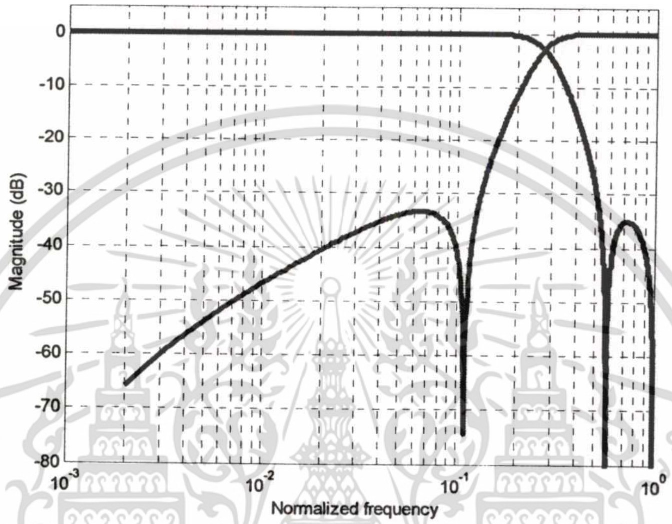
ซึ่งฟังก์ชันถ่ายโอน $A_0(Z)$ และ $A_1(Z)$ เป็นฟังก์ชันถ่ายโอนที่เรียกว่า All-Pass function ที่สามารถนำกลับมาสร้างเป็นวงจรกรองความถี่ต่ำต้นแบบและวงจรกรองความถี่สูงได้ตามสมการที่ (5.4) และ (5.5) ตามลำดับ

$$H_{LPF}(Z)=\frac{A_0(Z)+A_1(Z)}{2} \tag{5.4}$$

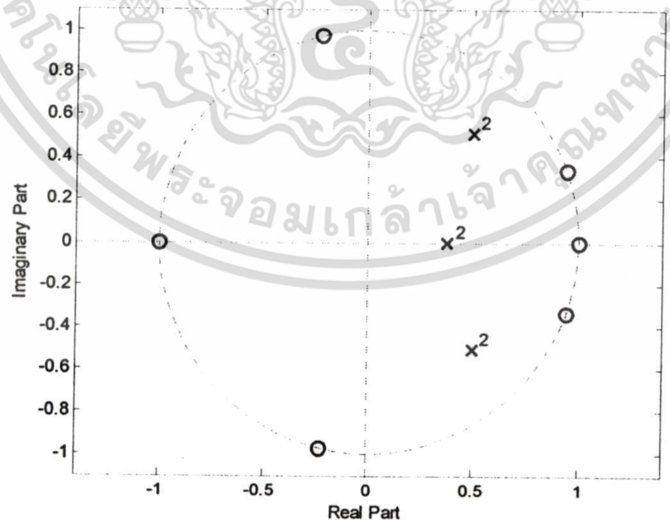
$$H_{HPF}(Z)=\frac{A_0(Z)-A_1(Z)}{2} \tag{5.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (5.4) และ (5.5) สามารถพล๊อตผลตอบสนองความถี่ เส้นทึบเป็นผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำจากสมการที่ (5.4) และเส้นบางเป็นผลตอบสนองความถี่สูงจากสมการที่ (5.5) ซึ่งเรียกว่า Mirror filter [26] โดยมีตำแหน่งของโพลของทั้งสองวงจรอยู่ที่จุดเดียวกัน ซึ่งจะมีคุณลักษณะตรงกันข้ามกับวงจรกรองต้นแบบ และเมื่อนำผลตอบสนองความถี่ทั้งสองมารวมกันจะได้ผลรวมมีค่าคงที่ดังรูปที่ 5.2 และ 5.3



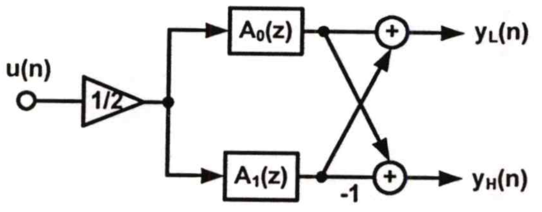
รูปที่ 5.2 ผลตอบสนองความถี่จากสมการที่ (5.4) และ (5.5)



รูปที่ 5.3 ตำแหน่งโพลและซีโร ของฟังก์ชันถ่ายโอนจากสมการที่ (5.4) และ (5.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ค่าฟังก์ชันถ่ายโอน $A_0(Z)$ และ $A_1(Z)$ เป็นฟังก์ชันถ่ายโอนที่เป็นวงจรกรองทุกย่านความถี่ผ่านมาทำการสร้างซึ่งจะมีความซับซ้อนน้อยกว่าวิธีการที่สร้างโดยตรงโดยมีโครงสร้างดังรูปที่ 5.4



รูปที่ 5.4 โครงสร้างของวงจรครอสโอเวอร์ที่สร้างจากวงจรกรองทุกย่านความถี่ผ่านต่อขนานกัน

จากรูปที่ 5.4 สามารถสร้างวงจรกรอง $A_0(Z)$ และ $A_1(Z)$ ด้วยโครงสร้างคณิตศาสตร์การกระจายโดยนำค่าสัมประสิทธิ์จากสมการ 5.2 และ 5.3 มาสร้างตารางเปิดดูจะได้เป็น

ตารางที่ 5.1 ค่าสัมประสิทธิ์ของตารางเปิดดูจากฟังก์ชันถ่ายโอนจากสมการที่ (5.2)

X(n)	X(n-1)	Y(n-1)	Output
0	0	0	0
0	0	1	$-b_0$
0	1	0	1
0	1	1	$1 - b_0$
1	0	0	b_0
1	0	1	0
1	1	0	$1 + b_0$
1	1	1	1

ตารางที่ 5.2 ค่าสัมประสิทธิ์ของตารางเปิดดูจากฟังก์ชันถ่ายโอนจากสมการที่ (5.3)

X(n)	X(n-1)	X(n-2)	Y(n-2)	Y(n-1)	Output
0	0	0	0	0	0
0	0	0	0	1	$-b_1$
0	0	0	1	0	$-b_0$
0	0	0	1	1	$-(b_1 + b_0)$
0	0	1	0	0	1
0	0	1	0	1	$1 - b_1$
0	0	1	1	0	$1 - b_0$
0	0	1	1	1	$1 - (b_1 + b_0)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 ต่อ

0	1	0	0	0	b_1
0	1	0	0	1	0
0	1	0	1	0	$b_1 - b_0$
0	1	0	1	1	$- b_0$
0	1	1	0	0	$1 + b_1$
0	1	1	0	1	1
0	1	1	1	0	$1 + b_1 - b_0$
0	1	1	1	1	$1 - b_0$
1	0	0	0	0	B_0
1	0	0	0	1	$b_0 - b_1$
1	0	0	1	0	0
1	0	0	1	1	$- b_1$
1	0	1	0	0	$1 + b_0$
1	0	1	0	1	$1 + b_0 - b_1$
1	0	1	1	0	1
1	0	1	1	1	$1 - b_1$
1	1	0	0	0	$b_0 + b_1$
1	1	0	0	1	b_0
1	1	0	1	0	b_1
1	1	0	1	1	0
1	1	1	0	0	$1 + b_0 + b_1$
1	1	1	0	1	$1 + b_0$
1	1	1	1	0	$1 + b_1$
1	1	1	1	1	1

จากตารางที่ 5.1 และ 5.2 สามารถนำไปสร้างไฟล์ตารางเปิดดูที่ใช้งานบน FPGA โดยใช้ขนาดในการเก็บข้อมูล 16 บิต ใช้พื้นที่หน่วยความจำ 8 ตำแหน่งและ 32 ตำแหน่ง ซึ่งหากออกแบบด้วยโครงสร้างแบบตรง (Direct form II) จะใช้พื้นหน่วยความจำมากถึง 128 ตำแหน่ง ซึ่งมากกว่าหลายเท่าตัว ดังแสดงในตารางที่ 5.3 เป็นการเปรียบเทียบการใช้พื้นที่ในการออกแบบตารางเปิดดูของวงจรกรองที่ออกแบบด้วยโครงสร้างแบบต่างๆจะเห็นว่าหากออกแบบครอสโอเวอร์แบบสองทางซึ่งเป็นวงจรกรอง 2 ชุด โครงสร้างของ Direct form II และแบบ FIR จะใช้พื้นที่ของตารางเปิดดูเพิ่มขึ้นเป็นเท่าตัวในขณะที่โครงสร้างแบบ Two parallel All-pass ใช้พื้นที่เท่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

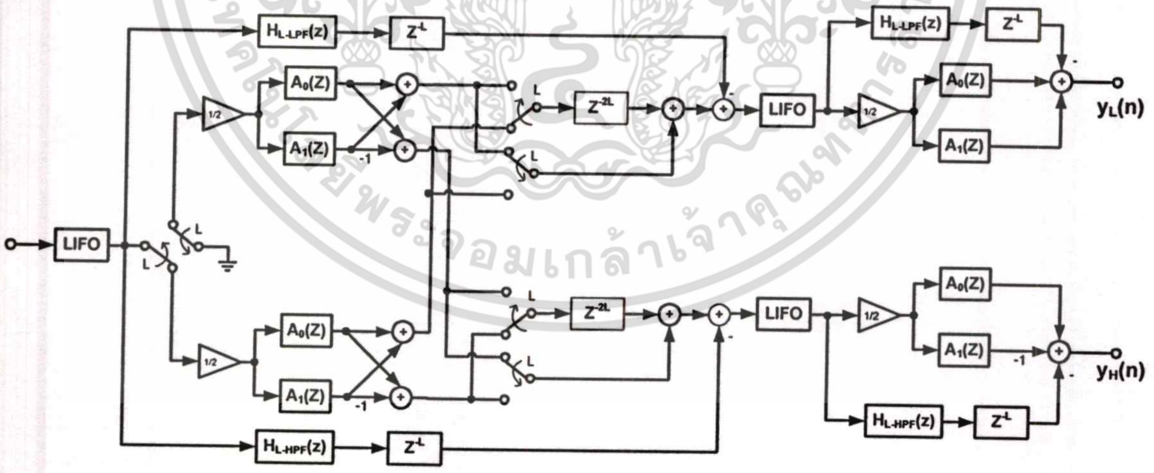
ตารางที่ 5.3 พื้นที่ในการออกแบบตารางเปิดดูของวงจรกรองที่ออกแบบด้วยโครงสร้างแบบต่างๆ

Type	Filter	2 Band Crossover
3^{rd} Order Direct form II	128	256
16 Tab FIR (REMEZ algorithm)*	64K	128K
Two parallel All-pass filter	40	40

* เป็นการประมาณคุณสมบัติของวงจรกรองต้นแบบที่เป็น IIR และมีการออกแบบโดยตรงไม่ได้ใช้เทคนิคลดพื้นที่หน่วยความจำ

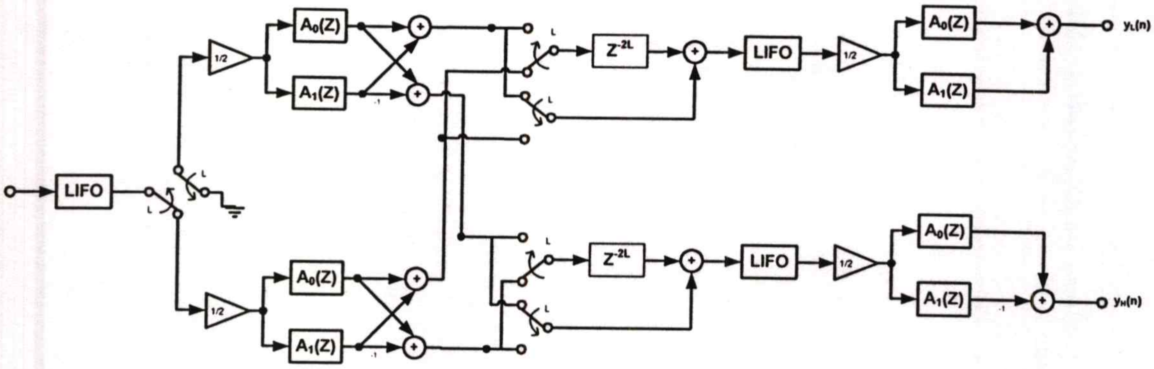
5.3 โครงสร้างวงจรครอสโอเวอร์แบบสองทางที่ให้เฟสเชิงเส้นสมบูรณ์บน FPGA

ในบทที่ผ่านมาได้กล่าวถึงการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่ให้เฟสเชิงเส้นสมบูรณ์ แต่ในการสร้างจริงบน FPGA นั้นได้ทำการตัดวงจรกรองเรซิดิวส์ ออกไปเนื่องจากมีผลกระทบต่อการทำงานน้อยมากเนื่องจากผลการตอบสนองของสัญญาณอิมพัลซ์ มีความยาวน้อยกว่าความยาวในการพลิกกลับสัญญาณของวงจรกลับสัญญาณจึงมีผลต่อระบบ โดยรวมน้อยมาก โดยแสดงดังรูปที่ 5.5 เป็นระบบที่สมบูรณ์ของวงจรครอสโอเวอร์เน็ตเวอร์ที่ให้เฟสเชิงเส้นสมบูรณ์ และรูปที่ 5.6 เป็นวงจรครอสโอเวอร์เน็ตเวอร์คที่นำไปสร้างบน FPGA



รูปที่ 5.5 วงจรครอสโอเวอร์เน็ตเวอร์คที่ให้เฟสเชิงเส้นสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 โครงสร้างวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางที่นำไปสร้างบน FPGA

จากโครงสร้างของวงจรครอสโอเวอร์เน็ตเวอร์คดังรูปที่ 5.6 ซึ่งเป็นโครงสร้างที่ให้ผลตอบสนองทางเฟสเชิงเส้นดังได้กล่าวในบทที่ผ่านมาแล้วนั้น สามารถแบ่งออกเป็นภาคต่างๆ เพื่อง่ายต่อการออกแบบบน FPGA อันได้แก่

โมดูลสวิทช์มัลติเพล็กซ์ (Multiplex switch)

โมดูลกลับสัญญาณในเวลาจริง (Real-time time reversal)

โมดูลหน่วงเวลา (Delay)

โมดูลกรองความถี่จากโครงสร้างแบบคณิตศาสตร์การกระจาย

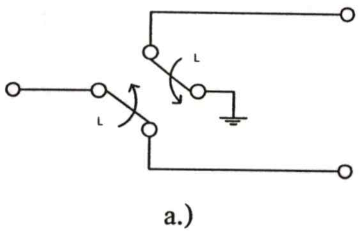
5.4 การออกแบบวงจรสวิทช์มัลติเพล็กซ์

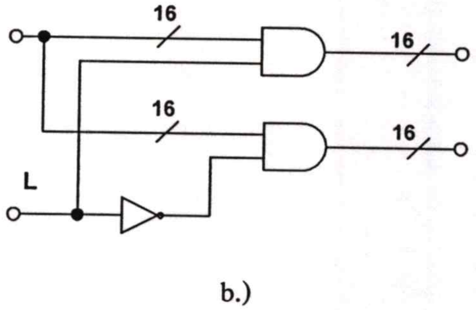
จากรูปที่ 5.6 วงจรสวิทช์จะมีอยู่สองส่วนได้แก่ ส่วนแรกหลังจากผ่านวงจรกลับสัญญาณในเวลาจริง (LIFO) และส่วนที่รวมสัญญาณจากวงจรกรองตัวบนและตัวล่าง

5.4.1 การออกแบบวงจรสวิทช์มัลติเพล็กซ์ส่วนที่ 1 (Multiplex switch)

วงจรสวิทช์ส่วนแรกจะเป็นการเลือกสลับสัญญาณให้กับวงจรกรองความถี่ชุดบนหรือชุดล่าง โดยมีจังหวะสลับการทำงานกันทุกๆ L แซมเปิ้ล โดยวงจรกรองที่ไม่ได้สัญญาณจากวงจรกลับสัญญาณก็จะถูกต่อเข้ากับข้อมูล “0” ซึ่งเสมือนต่อลงกราวน์ใน ซึ่งในวิทยานิพนธ์นี้ออกแบบให้มีการประมวลผลขนาด 16 บิต ดังนั้นบัสของวงจรสวิทช์ชุดแรกนี้จะเท่ากับ 16 บิต มีสัญลักษณ์และวงจรดังรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





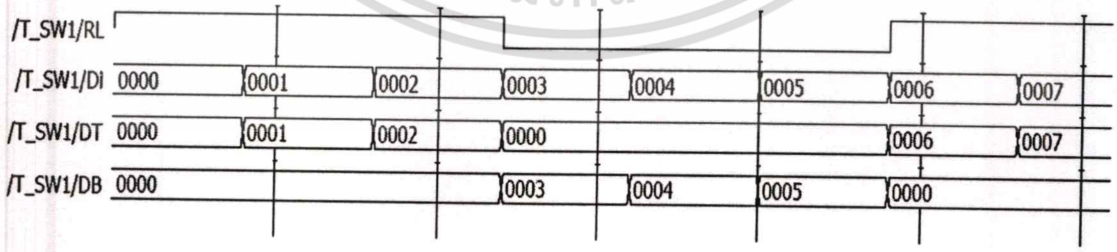
รูปที่ 5.7 สัญลักษณ์ (รูป a) และโครงสร้าง (รูป b) ของวงจรสวิทช์ส่วนที่ 1

จากวงจรในรูปที่ 5.7(b) สามารถนำไปเขียนด้วยภาษาอธิบายฮาร์ดแวร์โดยกำหนดอินพุตและเอาต์พุตขนาด 16 บิต โดยมี RL เป็นสัญญาณควบคุมการมัลติเพล็กซ์ดังนี้

```
module SW1(RL,Di,DT,DB);
  input RL;              // Multiplex control
  input [15:0] Di;       // Data input
  output [15:0] DT;      // Data output to top Filter
  output [15:0] DB;      // Data output to Bottom Filter

  assign DB[15:0] = Di[15:0] & {16{!RL}};
  assign DT[15:0] = Di[15:0] & {16{RL}};
endmodule
```

การจำลองการทำงานโดยกำหนดให้ข้อมูลอินพุต Di ที่ถูกควบคุมโดยสัญญาณ RL ซึ่งจะได้ผลจำลองการทำงานดังรูปที่ 5.8 โดยจะเห็นได้ว่าข้อมูลจะถูกส่งไปยังเอาต์พุต DT และ DB สลับกัน

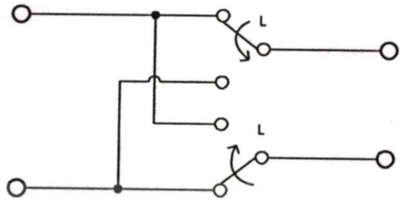


รูปที่ 5.8 ผลการจำลองการทำงานวงจรสวิทช์ส่วนที่ 1

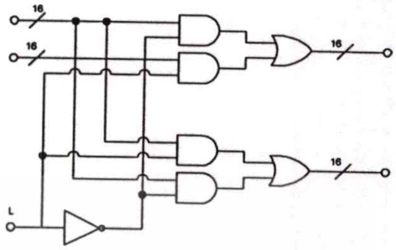
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 การออกแบบวงจรสวิทช์มัลติเพล็กซ์ส่วนที่ 2 (Multiplex switch)

วงจรสวิทช์ส่วนที่ 2 ทำหน้าที่สลับสัญญาณที่ได้จากวงจรกรองความถี่ด้านบนและด้านล่างโดยที่มีจังหวะสลับการทำงานทุกๆ L แซมเปิ้ลโดยมีขนาดข้อมูลเท่ากับ 16 บิตซึ่งมีสัญลักษณ์และวงจรดังรูปที่ 5.9



a



b

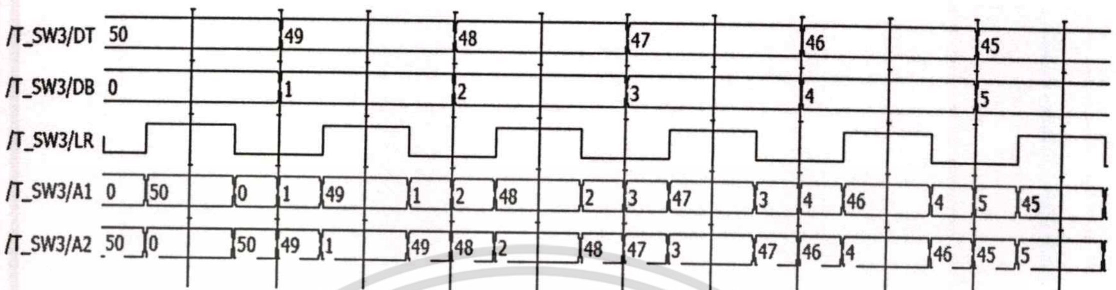
รูปที่ 5.9 สัญลักษณ์ (รูป a) และโครงสร้าง (รูป b) ของวงจรสวิทช์ส่วนที่ 2

จากวงจรในรูปที่ 5.9(b) สามารถนำไปเขียนด้วยภาษาอธิบายฮาร์ดแวร์โดยกำหนดอินพุตและเอาต์พุตขนาด 16 บิต โดยมี RL เป็นสัญญาณควบคุมการมัลติเพล็กซ์ดังนี้

```
module sw_3(DT,DB,LR,A1,A2);
    input [15:0] DT;          // Data input Bottom Filter
    input [15:0] DB;          // Data input Top Filter
    input LR;                 // Multiplex control
    output [15:0] A1;         // Data output to Delay
    output [15:0] A2;         // Data output to Adder
    wire [15:0]w1;
    wire [15:0]w2;
    wire [15:0]w3;
    wire [15:0]w4;
    assign w1[15:0] = DT[15:0] & {16{LR}};
    assign w2[15:0] = DB[15:0] & {16{! LR}};
    assign w3[15:0] = DT[15:0] & {16{! LR}};
    assign w4[15:0] = DB[15:0] & {16{LR}};
    assign A1 = w1 | w2 ;
    assign A2 = w3 | w4 ;
endmodule
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

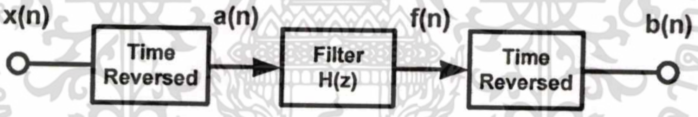
ในการจำลองการทำงานของโมดูลสวิทช์มัลติเพล็กซ์ตัวที่สองจะเห็นว่าเอาต์พุตทั้งสอง (A1, A2) จะเลือกข้อมูลจากอินพุต (DT, DB) ที่มาจากวงจรกรองความถี่โดยจะทำการสลับข้อมูลทุกครั้งที่ L มีการเปลี่ยนแปลงดังแสดงในรูปที่ 5.10



รูปที่ 5.10 ผลการจำลองการทำงานวงจรสวิทช์ส่วนที่ 2

5.5 วงจรกลับสัญญาณในเวลาจริง (Real –time Last In First Out)

การประมวลผลสัญญาณดิจิตอลของวงจรกรองความถี่แบบนอนคอซอลที่เพาเวลและชุลได้นำเสนอบทความ [32, 33] ซึ่งจะประกอบด้วยวงจรกลับสัญญาณในเวลาจริง วงจรกรองความถี่โดยแสดงดังรูปที่ 5.11



รูปที่ 5.11 โครงสร้างวงจรกรองแบบไม่เป็นคอซอลที่สามารถสร้างได้จริง

จากรูปที่ 5.11 จะเห็นว่าสัญญาณอินพุต x(n) จะถูกป้อนผ่านวงจรกลับสัญญาณครั้งละ L แซมเปิ้ลจากนั้นจะถูกส่งให้กับวงจรกรองความถี่และถูกป้อนเข้าสู่วงจรกลับสัญญาณอีกครั้งโดยสามารถเขียนในรูปสมการถ่ายโอนได้ดังนี้

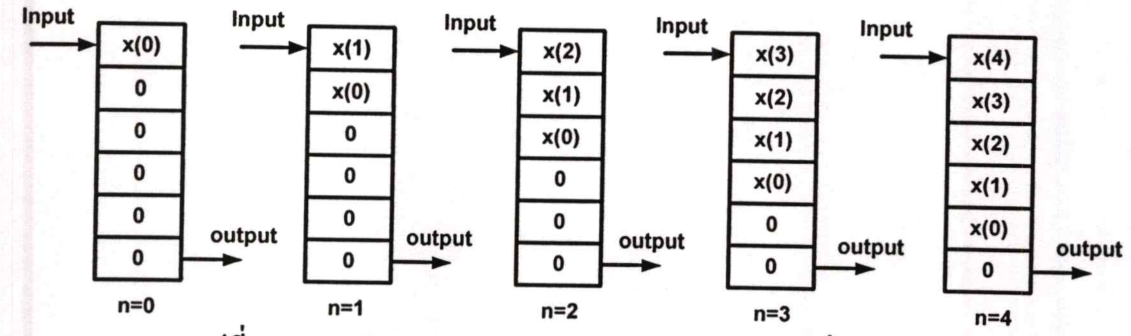
$$A(Z) = X(Z^{-1}) \quad (5.6)$$

$$F(Z) = H(Z) \cdot X(Z^{-1}) \quad (5.7)$$

$$B(Z) = F(Z^{-1}) = H(Z^{-1}) \cdot X(Z) \quad (5.8)$$

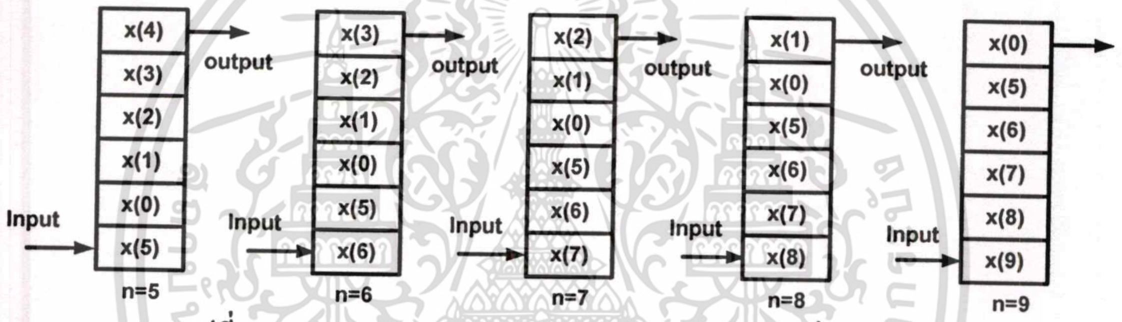
ในส่วนของวงจรกลับสัญญาณ เพาเวลและชุลได้นำเสนอวิธีการสร้างโดยใช้วงจรชิปรีจีสเตอร์ในการเก็บข้อมูลโดยเมื่อสัญญาณอินพุตในแต่ละแซมเปิ้ลเข้ามาจะถูกเก็บลงในชิปรีจีสเตอร์จนกว่าจะครบค่าความยาวที่จะกลับสัญญาณ (L)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



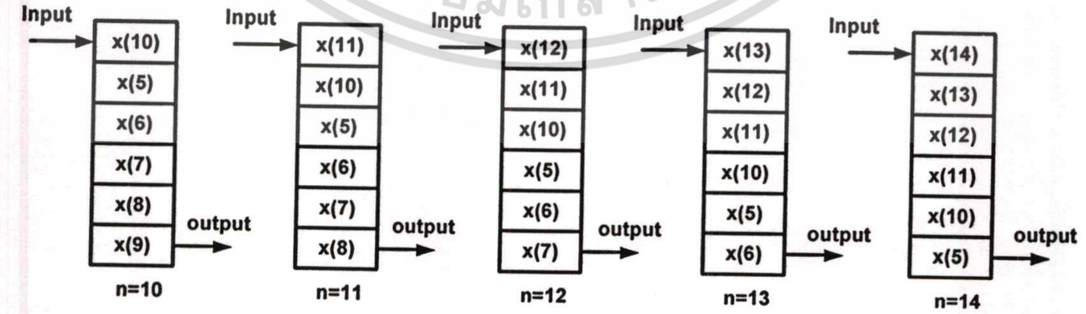
รูปที่ 5.12 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=0 ถึง 4

จากรูปที่ 5.12 จะเห็นว่าสัญญาณอินพุตที่ n=0 ถึง 4 จะถูกเก็บไว้ที่รีจิสเตอร์บนสุดจากนั้นก็จะถูกเลื่อนข้อมูลที่ถูกเก็บไว้ลงมาเพื่อรอรับข้อมูลในแซมเปิ้ลต่อไปโดยขนาดของชิปรีจิสเตอร์จะมีความยาวเท่ากับ L+1



รูปที่ 5.13 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=5 ถึง 9

เมื่อเก็บข้อมูลจนครบความยาว L แล้วจากนั้นสัญญาณอินพุตจะถูกเปลี่ยนมาเก็บไว้ที่ตำแหน่งล่างสุดของชิปรีจิสเตอร์แทนและเมื่อทำการเก็บข้อมูลแล้วก็จะทำการเลื่อนข้อมูลขึ้นไปซึ่งเอาต์พุตของวงจรกลับสัญญาณก็จะเปลี่ยนเป็นรีจิสเตอร์ตัวบนสุดแทนด้วยดังแสดงในรูปที่ 5.13

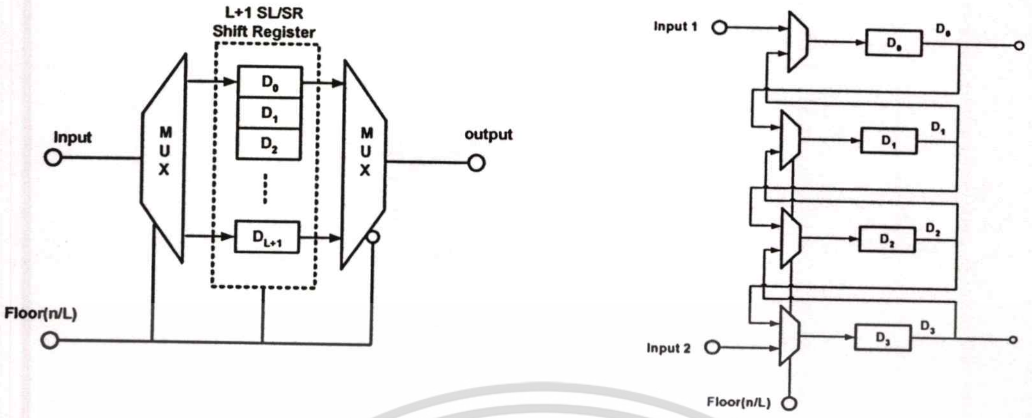


รูปที่ 5.14 การทำงานของวงจรกลับสัญญาณในเวลาจริงที่ n=10 ถึง 14

เมื่อข้อมูลแซมเปิ้ลที่ 10 อินพุตจะเปลี่ยนการเก็บข้อมูลจากด้านบนอีกครั้งดังแสดงในรูปที่ 5.14 ซึ่งจะสลับการทำงานแบบนี้ทุกๆ ความยาว L ซึ่งสามารถเขียนให้อยู่ในรูปแบบวงจรอย่างง่ายได้ดัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงในรูปที่ 5.15 (a) การทำงานของวงจรกลับสัญญาณในเวลาจริงที่สร้างจากวงจรชิปรีจีสเตอร์ (b) โครงสร้างในส่วนวงจรเก็บและเลื่อนข้อมูล



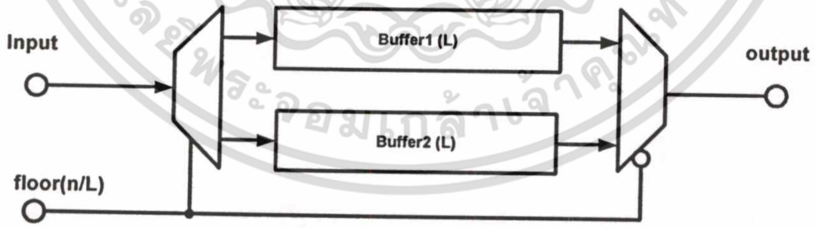
(a) (b)

รูปที่ 5.15 วงจรกลับสัญญาณในเวลาจริงที่สร้างจากวงจรชิปรีจีสเตอร์

จากรูปที่ 5.15(b) ความยาว L = 3 และขนาดข้อมูลเป็น 1 บิต ซึ่งในการใช้งานจริงนั้น L จะใช้งานราวๆ 150 ถึง 200 และขนาด 16 บิต ซึ่งนำไปสร้างจริงได้ยากเนื่องจากความซับซ้อนมากซึ่งเมื่อนำไปสร้างบนชิป FPGA จะใช้พื้นที่ในการออกแบบที่มากเกินที่จะสามารถโปรแกรมลงบนตัวชิปได้

5.5.1 การออกแบบวงจรกลับสัญญาณโดยใช้โครงสร้างของหน่วยความจำ

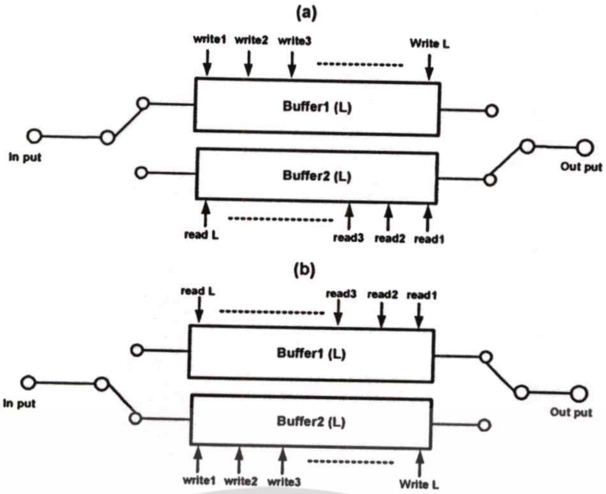
ในการออกแบบชิปรีจีสเตอร์บนโครงสร้างของ FPGA นั้นจะใช้พื้นที่ในการสร้างที่มากกว่าการออกแบบโดยใช้หน่วยความจำ ซึ่งในการออกแบบใหม่นี้ได้นำเสนอการออกแบบโดยใช้หน่วยความจำมาเก็บข้อมูลแทนชิปรีจีสเตอร์ซึ่งผลที่ได้นั้นมีการใช้พื้นที่น้อยกว่าอย่างมาก



รูปที่ 5.16 วงจรกลับสัญญาณที่ออกแบบด้วยหน่วยความจำ

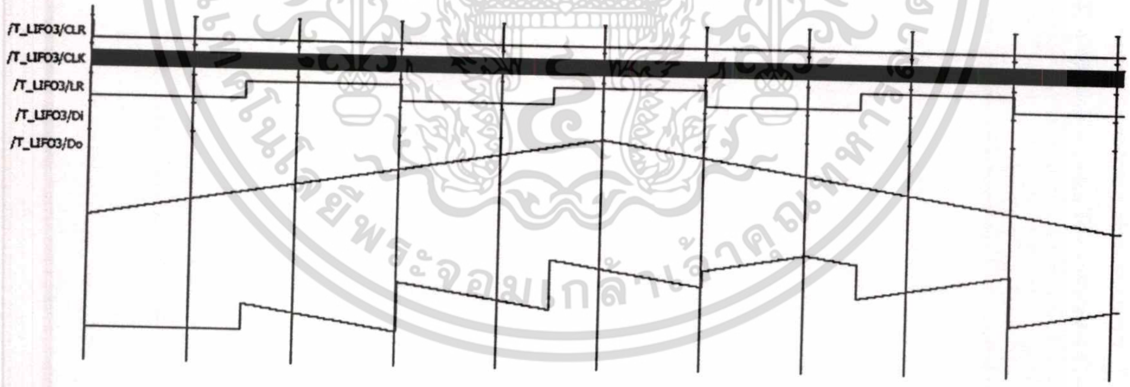
จากวงจรในรูปที่ 5.16 จะเห็นได้ว่าประกอบด้วยหน่วยความจำสองชุด (Buffer1, Buffer2) ซึ่งแต่ละตัวมีความยาวเท่ากับ L ซึ่งการเก็บข้อมูลและการอ่านข้อมูลออกจะใช้บัฟเฟอร์คนละชุด เช่น ในการเก็บข้อมูลลงในชุดบนข้อมูลเอาต์พุตก็จะใช้ชุดล่างซึ่งจะสลับกันทำงานทุกๆ ความยาว L ดังรูปที่ 5.17 ซึ่งในการสร้างบน FPGA ด้วยโครงสร้างที่นำเสนอนี้จะใช้พื้นที่น้อยกว่าการออกแบบด้วยโครงสร้างของชิปรีจีสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.17 การอ่านการเขียนข้อมูลในบัฟเฟอร์ (a) บัฟเฟอร์ 1 เขียนข้อมูล บัฟเฟอร์ 2 อ่านข้อมูล (b) บัฟเฟอร์ 2 เขียนข้อมูล บัฟเฟอร์ 1 อ่านข้อมูล

ผลการจำลองการทำงานของโมดูล LIFO ข้อมูล Di จะถูกเก็บข้อมูลเข้าสู่หน่วยความจำทุกๆ สัญญาณนาฬิกา (CLK) จนกระทั่งครบจำนวน L ขาควบคุม L จะทำการกลับสถานะสัญญาณและทำการนำข้อมูลโดยออกมาจากหน่วยความจำที่ทำการเก็บไว้ก่อนแล้วพร้อมกับเก็บข้อมูลชุดใหม่เข้าไปซึ่งจะทำให้สัญญาณที่เข้ามาถูกพลิกกลับเป็นบล๊อกด้วยความยาว L ต่อกัน ไปเรื่อยๆ ดังแสดงในรูปที่ 5.18

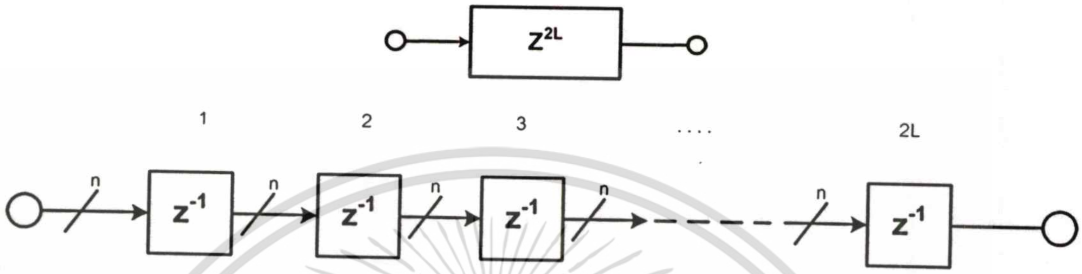


รูปที่ 5.18 ผลการจำลองการทำงานโมดูล LIFO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

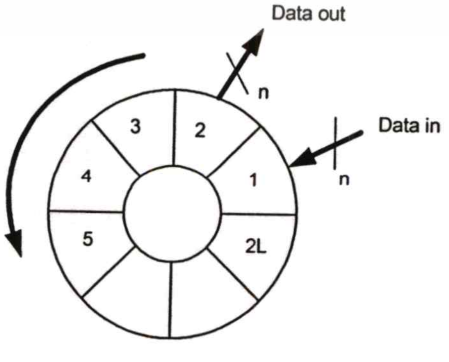
5.6 การออกแบบวงจรหน่วงสัญญาณ

จากที่ได้กล่าวมาแล้วว่าการประมวลผลสัญญาณดิจิตอลส่วนใหญ่จะประกอบไปด้วยวงจรบวก วงจรลบ วงจรคูณ และหน่วงสัญญาณ ซึ่ง ในการออกแบบวงจรหน่วงเวลานั้นสามารถออกแบบโดยการใช้โครงสร้างของชิปรีจีสเตอร์หรือโครงสร้างของหน่วยความจำได้ด้วยเช่นกันโดยโครงสร้างของวงจรหน่วงสัญญาณดังกล่าวแสดงดังรูปที่ 5.19



รูปที่ 5.19 โครงสร้างของวงจรหน่วงสัญญาณ

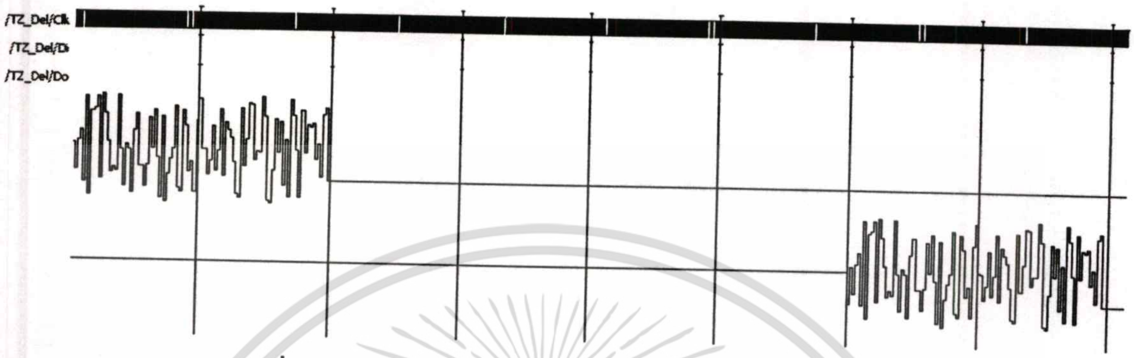
จากรูปที่ 5.19 โครงสร้างดังกล่าวสามารถสร้างได้จากวงจรดีฟลิบฟลอบ (D Flip Flop) โดย ฟลิบฟลอบจำนวนหนึ่งตัวจะหน่วงสัญญาณออกไปหนึ่งแซมเปิ้ล ดังนั้นในการสร้างวงจรหน่วงสัญญาณด้วยโครงสร้างที่กล่าวมานี้จะใช้พื้นที่ในการสร้างที่มากเช่นเดียวกับการออกแบบวงจรกลับสัญญาณในหัวข้อที่ได้กล่าวมาแล้ว จึงได้มีการออกแบบจากวงจรหน่วงสัญญาณด้วยการใช้โครงสร้างของชิปรีจีสเตอร์จะใช้พื้นที่ในการออกแบบที่มากมาเป็นการออกแบบโดยใช้โครงสร้างหน่วยความจำดังแสดงในรูปที่ 5.20 แทนซึ่งเรียกว่าหน่วยความจำแบบวงกลม (circular buffer) โดยในการทำงานนั้นจะทำการเก็บข้อมูล การอ่านข้อมูลด้วยโครงสร้างของ หน่วยความจำสองพอร์ต (dual port RAM) ซึ่งขนาดของหน่วยความจำนั้นจะเป็นตัวกำหนดจำนวนแซมเปิ้ลในการหน่วงสัญญาณ เช่น ในการออกแบบใช้หน่วยความจำขนาด 300 ตำแหน่ง หมายถึงข้อมูลที่เข้ามาในวงจรหน่วงสัญญาณนี้จะถูกหน่วงสัญญาณออกไป 300 แซมเปิ้ล



รูปที่ 5.20 โครงสร้างของวงจรหน่วงสัญญาณที่สร้างจากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

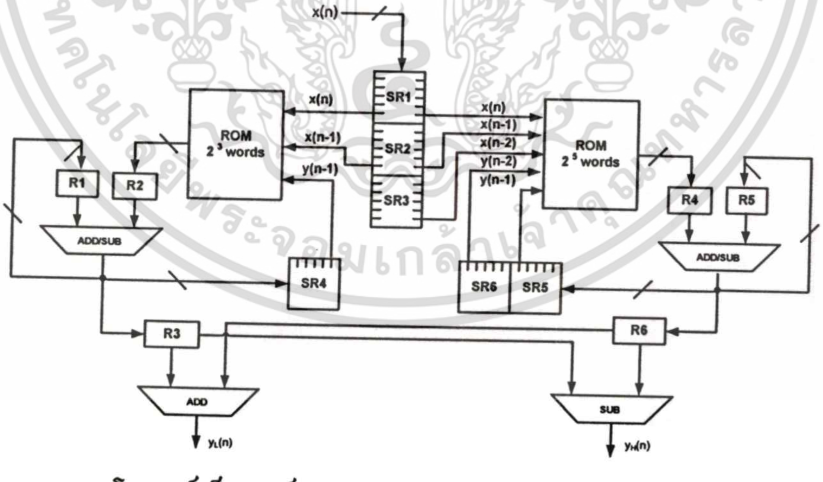
ในการจำลองการทำงานของโมดูลหน่วงเวลา L แซมเปิ้ลข้อมูล Di จะถูกเก็บเข้าไปในหน่วยความจำไปเรื่อยๆ จนครบเวลา L ครั้งจากนั้นข้อมูลที่ถูกเก็บไว้ในชุดแรกออกมา ดังแสดงในรูปที่ 5.21



รูปที่ 5.21 ผลการจำลองการทำงานของโมดูลหน่วงสัญญาณ

5.7 วงจรครอสโอเวอร์เน็ตเวอร์คจากโครงสร้างคณิตศาสตร์การกระจาย

การสร้างวงจรกรองบน เอฟ พี จี เอ ด้วยโครงสร้างของคณิตศาสตร์การกระจายด้วยภาษา Verilog HDL โดยมีโครงสร้างดังรูปที่ 5.22



รูปที่ 5.22 วงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางจากโครงสร้างของคณิตศาสตร์การกระจาย

จากรูปที่ 5.22 เป็นวงจรดิจิตอลครอสโอเวอร์เน็ตเวอร์คด้วยวงจรกรองความถี่ผ่านตลอดต่อขนาน 2 วงจร (two parallel all-pass filters) ที่สร้างด้วยวงจรคณิตศาสตร์การกระจายโดยจะรับข้อมูลมาจากวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลที่เข้ามาในลักษณะอนุกรม เพื่อให้ง่ายในการสร้างวงจรจะถูกแบ่งเป็นโมดูลย่อยๆตามความเหมาะสมซึ่งสามารถแบ่งออกได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โครงสร้างตารางเปิดดูค่า (LUT)
- ชิปรีจิสเตอร์แบบ เข้าขนานออกอนุกรมขนาด 16 บิต (16 Bit PISO)
- ชิปรีจีสเตอร์แบบ เข้าอนุกรมออกอนุกรมขนาด 16 บิต (16 Bit SISO)
- แอกคิวมูเลเตอร์ขนาด16 บิต แบบคิดเครื่องหมาย (16 Bit Accumulator)
- วงจรกำเนิดสัญญาณควบคุมจังหวะการทำงาน (System clock)

การออกแบบในแต่ละส่วนจะต้องทราบถึงหลักการทำงานของสัญญาณต่างๆที่เข้ามาแล้วทำการกำหนดฟังก์ชันการทำงานกับสัญญาณที่ป้อนเข้ามาว่าต้องการสัญญาณเอาต์พุตจากโมดูลนั้นเป็นอย่างไร การทำงานและผลการทำงานในแต่ละโมดูลนั้นสามารถออกแบบและจำลองการทำงานด้วยโปรแกรม Xilinx ISE และ ModelSim โดยขั้นตอนการออกแบบเป็นดังนี้

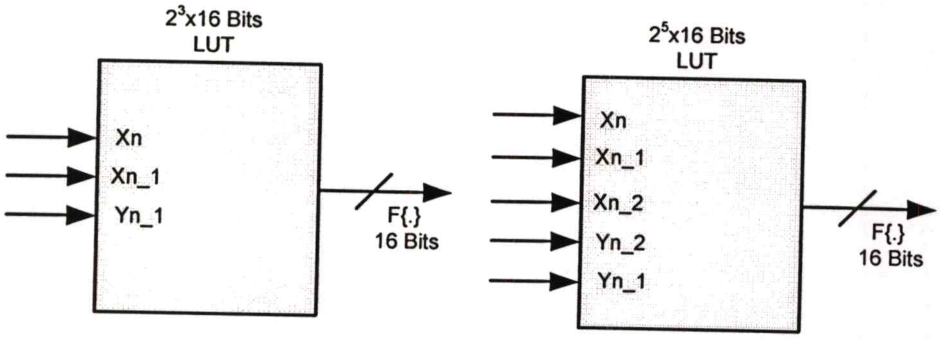
5.7.1 การสร้างไฟล์ตารางเปิดดู

จากผลรวมของค่าสัมประสิทธิ์ที่ได้จากตารางที่ 5.1 และ 5.2 สามารถนำไปสร้างตารางเปิดดูในรูปแบบของภาษา Verilog HDL ซึ่งจะมีโครงสร้างของภาษาดังนี้

```
module LUT(Di,Y);
  input [1:0] Di;          // Data input
  output [7:0] Y;          // Data output to top Filter
  always @(Di)
  begin
     case (Di)
           2 ' d0 : Y <= 8' d250;
             2 ' d1 : Y <= 8' d200;
             2 ' d2 : Y <= 8' d150;
             2 ' d3 : Y <= 8' d100;
          default : Y <= 16'b0;
     endcase
  end
endmodule
```

จากตัวอย่างข้างบนเป็นไฟล์ตารางเปิดดูขนาดอินพุต 2 บิตให้เอาต์พุตขนาด 8 บิตซึ่งหากค่าสัมประสิทธิ์มีจำนวนมากขึ้นก็ยิ่งทำให้มีจำนวนเงื่อนไขที่มากขึ้นตามไปด้วย ดังนั้นในการลดความยุ่งยากในการสร้างไฟล์ดังกล่าว จึงได้ออกแบบซอร์ฟแวร์ในการสร้างไฟล์ตารางเปิดดูขึ้นมาโดยสามารถกำหนดค่าของสัมประสิทธิ์ ขนาดของข้อมูลเอาต์พุต ซึ่งวิธีการนี้ทำให้สามารถลดระยะเวลาในการทำงานลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



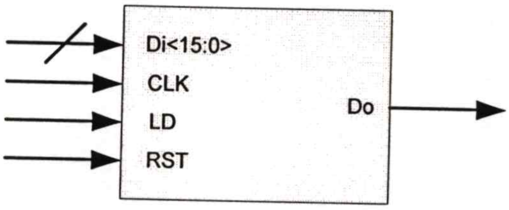
รูปที่ 5.23 สัญลักษณ์ของโมดูลเปิดตารางจากตารางที่ 5.1 และ 5.2

5.7.2 การออกแบบชิปรีจิสเตอร์แบบเข้าขนานออกอนุกรมขนาด 16 บิต (16 Bit PISO)

การออกแบบวงจรชิปรีจิสเตอร์แบบเข้าขนานออกอนุกรมขนาด 16 บิตแสดงดังรูปที่ 5.24 มี Di เป็นอินพุตขนาด 16 บิตมี Do สัญญาณเอาต์พุต ซึ่งมีขาที่ใช้ในการควบคุมได้แก่ รีเซ็ต (RST) โหลดข้อมูล (LD) และสัญญาณนาฬิกา (CLK) สามารถเขียนด้วยภาษาอธิบายฮาร์ดแวร์ได้ดังนี้

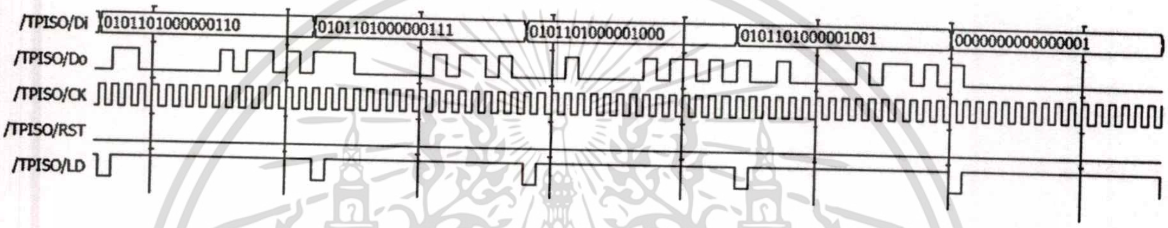
```
module PISO(Di,Do,CK,RST,LD);
input [width-1:0] Di CK,RST,LD;
output Do;
reg       [16:0]dat;
          assign Do = dat[0];
          always @ (posedge CK or posedge RST or negedge LD )
          begin
                    if (RST)
                              dat[16:0] = 0 ;
          else begin
                    if(!LD)
                              dat[16:0] = {0,Di};
                    else begin
                              dat[15:0]  = {dat[16:1],0};
                    end
          end
end
endmodule
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.24 โครงสร้างของโมดูลชิปรีจีสเตอร์แบบเข้าขนานออกอนุกรมขนาด 16 บิต

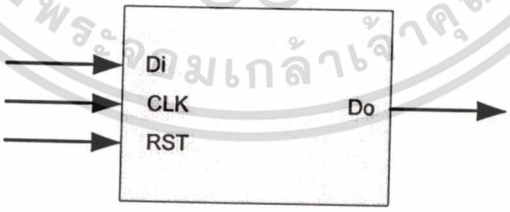
การจำลองการทำงานของโมดูล ข้อมูลจะถูกป้อนให้กับอินพุตก่อนที่จะทำการโหลดข้อมูลลงในชิปรีจีสเตอร์ทั้ง16 บิตก่อนที่จะทำการเลื่อนข้อมูลออกมาทีละบิตทางเอาต์พุตดังแสดงในรูปที่ 5.25



รูปที่ 5.25 ผลการจำลองการทำงานของโมดูลชิปรีจีสเตอร์แบบเข้าขนานออกอนุกรม

5.7.3 การออกแบบชิปรีจีสเตอร์แบบ เข้าอนุกรมออกอนุกรมขนาด 16 บิต (16 Bit SISO)

การออกแบบวงจรชิปรีจิสเตอร์แบบเข้าอนุกรมออกอนุกรมขนาด 16 บิตแสดงดังรูปที่ 5.26 โดยที่มี Di เป็นอินพุต Do เป็นสัญญาณเอาต์พุต ซึ่งมีขาที่ใช้ในการควบคุมได้แก่ รีเซ็ต (RST) และสัญญาณนาฬิกา (CLK) สามารถเขียนด้วยภาษาอธิบายฮาร์ดแวร์ได้ดังนี้

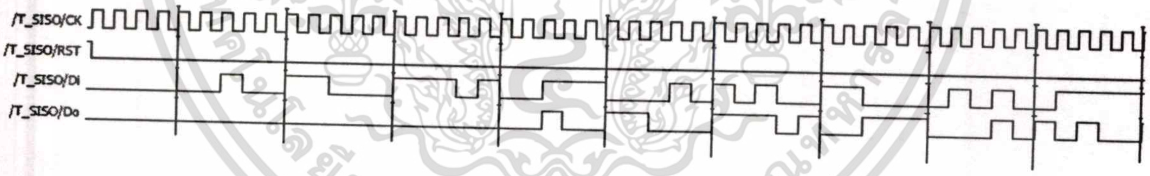


รูปที่ 5.26 โครงสร้างของโมดูลชิปรีจีสเตอร์แบบเข้าอนุกรมออกอนุกรมขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
module SiSo(Di,Do,CK,RST);
  input Di,RST,CK;
output Do;
  reg    [15:0]dat,Do;
  always @(posedge CK or posedge RST)
        if (RST)  begin
            dat[15:0]=0;
            Do=0;
        end
        else begin
            Do=dat[15];
            dat[15:0] = {dat[14:0],Di};
        end
endmodule
```

เมื่อทำการทดสอบการทำงานของโมดูลโดยการป้อนสัญญาณอินพุตแบบอนุกรมดังแสดงในรูปที่ 5.27 จะเห็นได้ว่าโมดูลดังกล่าวจะทำการหน่วงสัญญาณที่เข้ามาออกไปจำนวน 16 รอบของสัญญาณนาฬิกา โดยที่ข้อมูลที่ผ่านโมดูลดังกล่าวยังมีลักษณะเช่นเดิมแต่ถูกหน่วงไป 16 รอบสัญญาณนาฬิกา



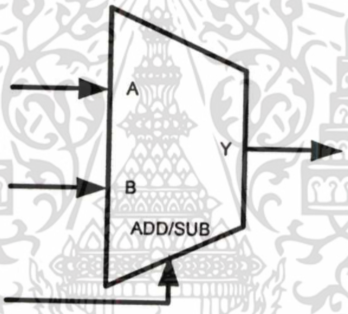
รูปที่ 5.27 ผลการจำลองการทำงานของโมดูลชิปรีจีสเตอร์แบบเข้าอนุกรมออกอนุกรม

5.7.4 การออกแบบแอกคิวมูเลเตอร์ขนาด16 บิต แบบคิดเครื่องหมาย (16 Bit Accumulator)

ส่วนประกอบภายในของแอกคิวมูเลเตอร์ที่ทำการออกแบบแสดงดังรูปที่ 5.28 ซึ่งเป็นวงจรบวกลบเลขขนาด 16 บิตแบบคิดเครื่องหมายโดยมี (A) และ (B) เป็นอินพุตส่วน (Y) เป็นเอาต์พุตขนาด 16 บิตสามารถควบคุมการบวกลบข้อมูลด้วยสัญญาณ (add/sub) สามารถเขียนด้วยภาษาอธิบายฮาร์ดแวร์ได้ดังนี้

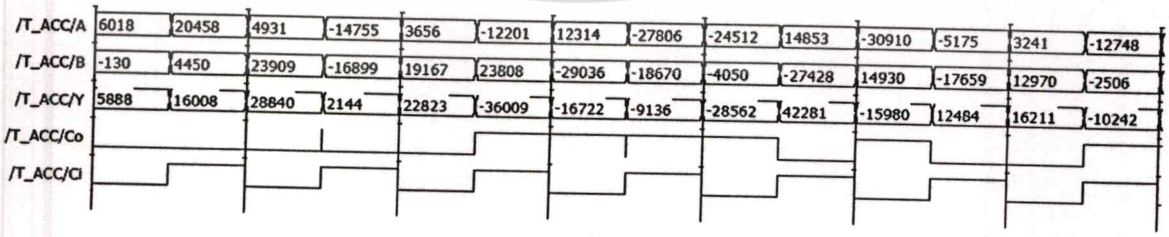
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
module acc(A,B,Y,Co,Ci);
 input [15:0] A, 15:0] B,Ci;
 output [16:0] Y,Co;
 wire [15:0]Do,t1;
 reg [16:0]result;
        assign Do=B^{16{Ci}};
        assign t1 = A[15]^Do[15];
        assign Co = result[16]^t1;
        assign Y[16]=Co;
        assign Y[15:0] =  result[15:0];
        always @(A or Do or Ci)
                result = A + Do + Ci;
endmodule
```



รูปที่ 5.28 โครงสร้างของโมดูลแอกคิวมูเลเตอร์

การจำลองการทำงานของ โมดูลแอกคิวมูเลเตอร์โดยที่เอาต์พุต (Y) จะเป็นผลบวกหรือผลต่างของข้อมูล A และ B ซึ่งถูกควบคุมด้วยสัญญาณ Ci แสดงดังรูปที่ 5.29



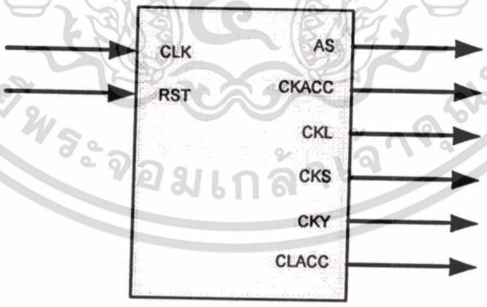
รูปที่ 5.29 ผลการจำลองการทำงานของของ โมดูลแอกคิวมูเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7.5 การออกแบบวงจรควบคุมจังหวะการทำงาน (System clock)

ส่วนประกอบภายในวงจรควบคุมจังหวะการทำงานทั้งหมดที่ทำการออกแบบแสดงดังรูปที่ 5.30 ประกอบไปด้วยสัญญาณนาฬิกาจากภายนอกและสัญญาณรีเซ็ตระบบทั้งหมด โดยการทำงานของโมดูลนี้จะนำสัญญาณนาฬิกาจากภายนอกมาทำการจัดใหม่ให้ระบบทั้งหมดทำงานเข้าจังหวะกันซึ่งในวงจรครอสโอเวอร์เน็ตเวอร์คจะประกอบไปด้วยสัญญาณต่างๆ ดังนี้

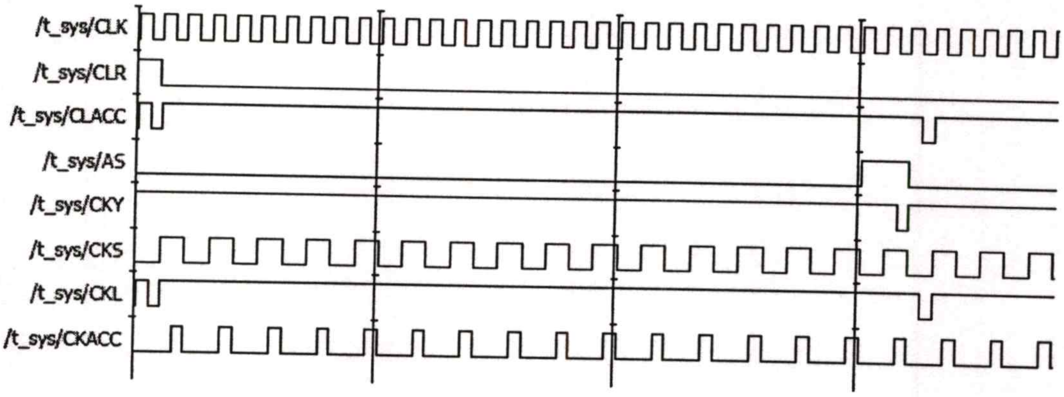
- CLK เป็นสัญญาณนาฬิกาจากวงจรออสซิลเลเตอร์ภายนอก
- RST เป็นสัญญาณที่ใช้ในการรีเซ็ตระบบทั้งหมด
- CKL สัญญาณที่ใช้ในการโหลดข้อมูลอินพุตซึ่งมีอัตราการสุ่มข้อมูลที่ 48KHz
- CLKS สัญญาณที่ใช้ในการชิปข้อมูลเข้ามาประมวลผลโดยในแต่ละรอบของการสุ่มข้อมูลมาประมวลผลสัญญาณ CLKS จะกำเนิดสัญญาณนาฬิกา 16 ครั้งเท่ากับขนาดของข้อมูลในการประมวลผลคือ 16 บิต
- CLKACC เป็นสัญญาณนาฬิกาที่ใช้ในการประมวลผลของแอกคิวมูเลเตอร์โดยสัญญาณนี้จะเกิดขึ้นภายหลังจากการโหลดข้อมูลเสร็จ
- AS สัญญาณที่ใช้ในการควบคุมการบวก ลบ ข้อมูลในแอกคิวมูเลเตอร์โดยจะเปลี่ยนสถานะเป็นลอจิก “1” ทุกๆ บิต MSB ของข้อมูลนั้นๆ
- CKY สัญญาณ ที่ใช้ในการโหลดข้อมูลออกจากแอกคิวมูเลเตอร์ภายหลังจากประมวลผลเสร็จ
- CLACC เป็นสัญญาณที่ใช้ในการเคลียร์ข้อมูลในแอกคิวมูเลเตอร์ก่อนที่จะทำการโหลดข้อมูลในชุดถัดไปมาประมวลผล



รูปที่ 5.30 โครงสร้างของโมดูลควบคุมจังหวะการทำงาน

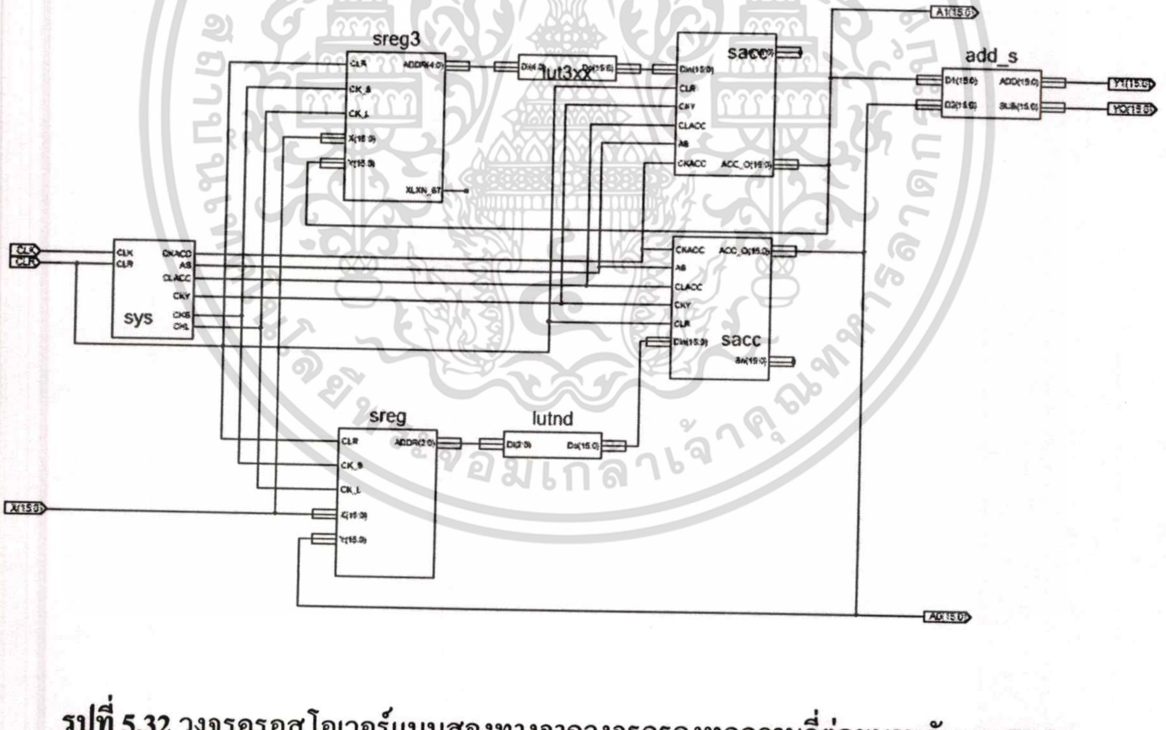
การจำลองการทำงานของโมดูลควบคุมจังหวะการทำงานจะทำการป้อนสัญญาณรีเซ็ตและสัญญาณนาฬิกาโมดูลจะสร้างสัญญาณควบคุมออกมาเพื่อนำไปควบคุมการทำงานของโมดูลอื่นๆต่อไปดังแสดงในรูปที่ 5.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.31 ผลการจำลองการทำงานของของโมดูลวงจรควบคุมจังหวะการทำงาน

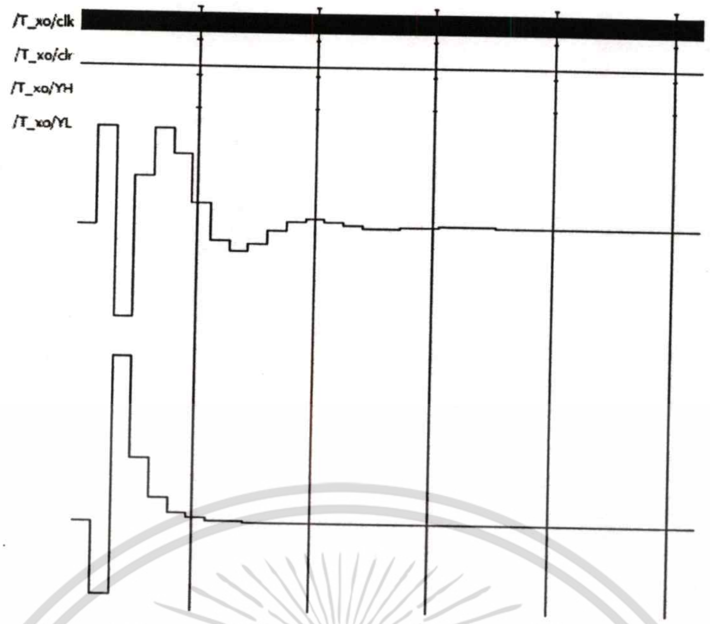
ซึ่งเมื่อนำโมดูลทั้งหมดที่ได้ออกแบบมาต่อกันเป็นวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA ตามโครงสร้างในรูปที่ 5.4 จะได้โมดูลของวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางเพื่อที่จะนำไปออกแบบรวมกับโมดูลอื่นต่อไปดังแสดงในรูปที่ 5.32



รูปที่ 5.32 วงจรครอสโอเวอร์แบบสองทางจากวงจรกรองทุกความถี่ต่อขนานกันบน FPGA

เมื่อนำโครงสร้างจากรูปที่ 5.32 ไปหาผลตอบสนองสัญญาณอิมพัลซ์ของระบบจะได้ผลจำลองการทำงานของ $A_0(Z)$ และ $A_1(Z)$ ดังรูปที่ 5.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.33 ผลการตอบสนองอิมพัลซ์ของวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางบน FPGA

5.8 การสังเคราะห์วงจร

หลังจากที่ได้กล่าวถึงการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คในแต่ละส่วนเป็นที่เรียบร้อยแล้วขั้นตอนสุดท้ายของการสร้างก็คือการสังเคราะห์วงจรที่ได้ออกแบบเพื่อทำการโปรแกรมลงสู่ FPGA จากซอร์ฟแวร์ที่ใช้ในการพัฒนาจะรายงานผลการสังเคราะห์วงจรรวมทั้งขีดความสามารถต่างๆ ที่วงจรสามารถทำงานได้รวมไปจนถึงการใช้งานทรัพยากรภายใน FPGA ที่ถูกใช้ไปดังแสดงในตาราง ที่ 5.3 ดังนี้

ตารางที่ 5.4 ผลการสังเคราะห์แต่ละโมดูลของวงจรครอสโอเวอร์เน็ตเวอร์ค

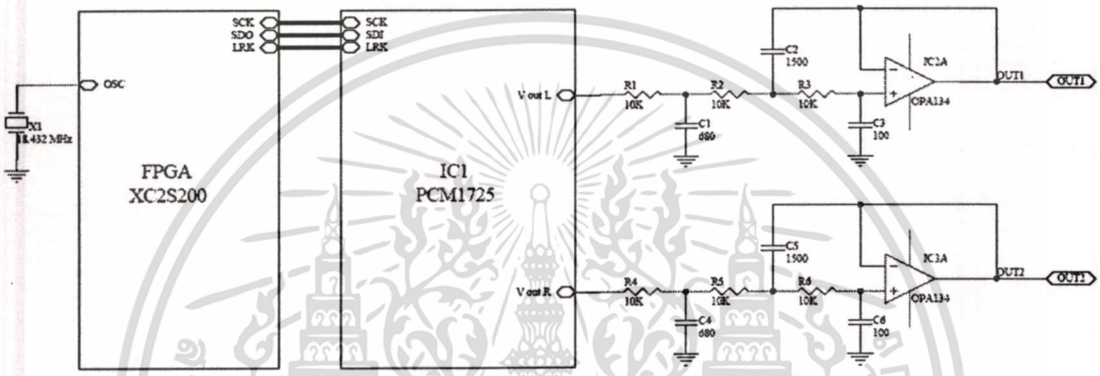
Device utilization summarizing **Selected device : XC2S2000 144 - 5**	**PAPF**	**DELAY (L=300)**	**LIFO (L=150)**	**SW1**	**SW2**
Number of Slices (1920 Max)	175 (9%)	19 (0.98%)	529 (27%)	18 (0.93%)	18 (0.93%)
Number of Slices Flip Flops (3840 Max)	240 (6%)	18 (0.46%)	42 (1%)	-	
Number of 4 input LUTs (3840 Max)	316 (8%)	34 (0.88%)	646 (16%)	32 (0.83%)	32 (0.83%)
Number of bonded IOBs (97 Max)	81 (83%)	33 (34%)	34 (35%)	49(50%)	65(67%)
Number of BRAMs (12 Max)	-	1 (8%)	-	-	-
Number of GCKs (8 Max)	1 (12%)	1 (12%)	1 (12%)	-	-
Maximum speed (MHz)	220.216	143.730	85.844	-	

* PAPF (Parallel All-pass Filter), LIFO (Last In First Out), sw1 (switch multiplex 1), sw2 (switch multiplex 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.9 ผลการทดสอบบน FPGA

เมื่อทำการออกแบบและจำลองผลการทำงานบนคอมพิวเตอร์เป็นที่เรียบร้อยแล้วจากนั้นจึงทำการรดาวน์โหลดโปรแกรมลงบนชิป FPGA แล้วทำการทดสอบการทำงานจริงอีกครั้งของแต่ละโมดูล ซึ่งในการทดสอบจะทำการสร้างสัญญาณอินพุตจากภายในตัว FPGA โดยใช้วิธีการเปิดตาราง และเอาต์พุตที่ได้จากแต่ละโมดูลจะต่อกับวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกเพื่อทำการวัดผลการทำงานโดยมีการเชื่อมต่อระหว่าง FPGA กับวงจรภายนอกดังรูปที่ 5.34



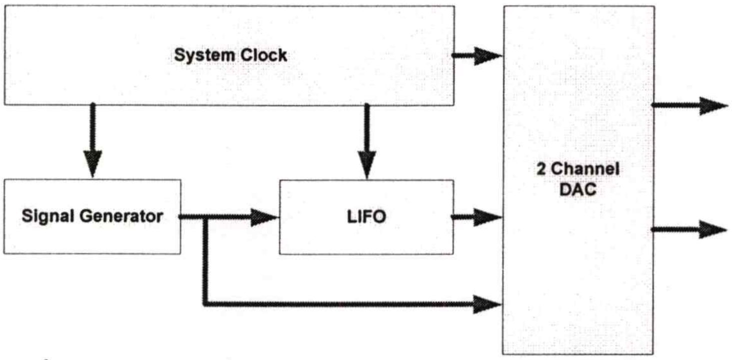
รูปที่ 5.34 การเชื่อมต่อ FPGA กับอุปกรณ์ภายนอกเพื่อใช้ในการทดสอบ

ในรูปที่ 5.34 เป็นการเชื่อมต่อระหว่างบอร์ด FPGA กับวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกที่มีความละเอียดขนาด 16 บิตโดยมีการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน I2S [34] โดยมีความถี่ในการสุ่มข้อมูล 48 KHz ซึ่งในการทดสอบการทำงานจะแบ่งทำสอบเป็นโมดูลย่อยๆดังต่อไปนี้

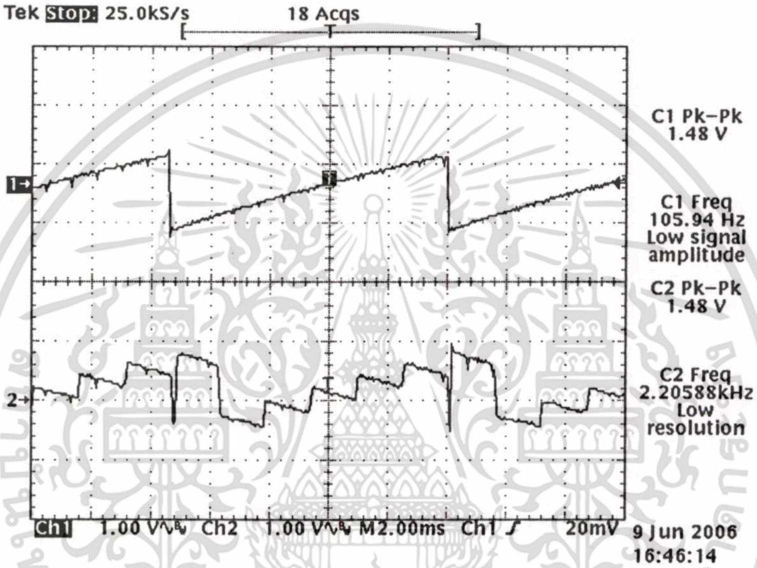
5.9.1 การทดลองโมดูลกลับสัญญาณในเวลาจริง

ในการทดสอบโมดูลกลับสัญญาณในเวลาจริงนั้นจะทำการเชื่อมต่อกับโมดูลตัวอื่นๆอีกเพื่อใช้ในการทดสอบได้แก่ โมดุลกำเนิดสัญญาณนาฬิกา โมดูลกำเนิดสัญญาณใช้ทดสอบ โมดูลเชื่อมต่อกับวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกโดยมีบล๊อกการเชื่อมต่อดังรูปที่ 5.35
เมื่อทำการดาวน์โหลดโปรแกรมลงบน FPGA โดยทำการเปลี่ยนลักษณะสัญญาณที่ใช้ทดสอบในรูปแบบต่างๆ ซึ่งผลการทดลองโมดูลกลับสัญญาณในเวลาจริงแสดงดังรูปที่ 5.36 - 5.39

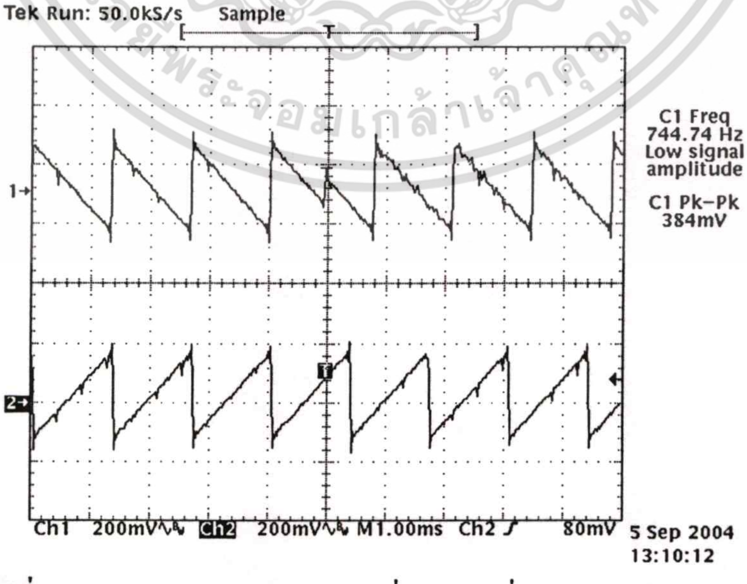
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.35 การทดสอบการทำงานของโมดูลกลับสัญญาณในเวลาจริง

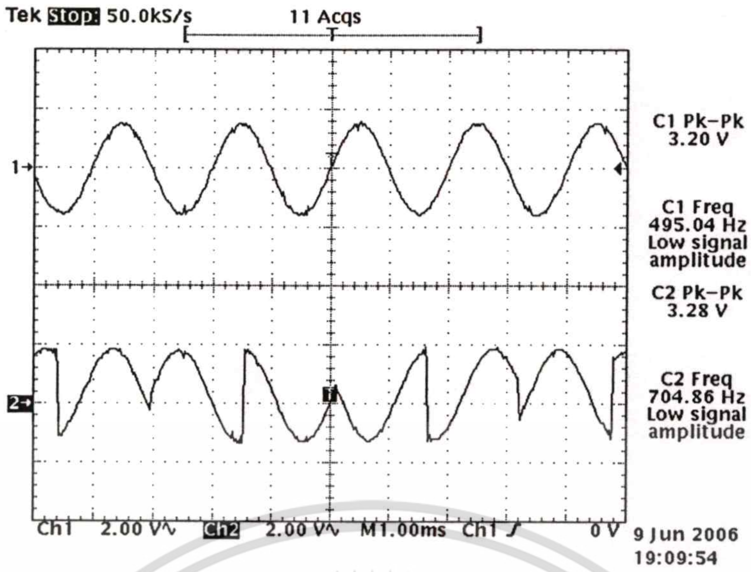


รูปที่ 5.36 สัญญาณ saw tooth ความถี่ 106 Hz ที่ผ่านวงจรกลับสัญญาณ

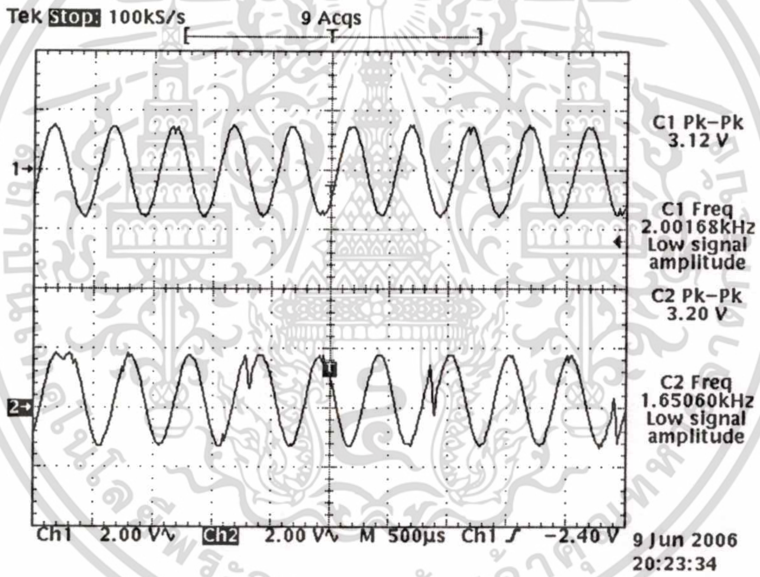


รูปที่ 5.37 สัญญาณ saw tooth ความถี่ 745 Hz ที่ผ่านวงจรกลับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.38 สัญญาณ sine wave ความถี่ 495 Hz ที่ผ่านวงจรกลับสัญญาณ



รูปที่ 5.39 สัญญาณ sine wave ความถี่ 2 KHz ที่ผ่านวงจรกลับสัญญาณ

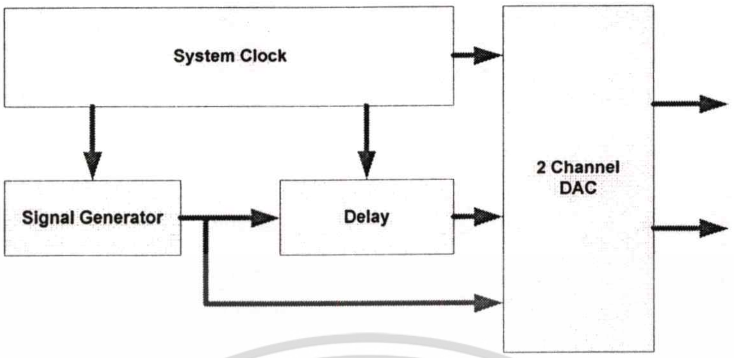
การทดสอบวงจรกลับสัญญาณจะให้สัญญาณ saw tooth และ sine wave ที่มีความถี่มากกว่าและน้อยกว่าขนาดหน่วยความจำของวงจรกลับสัญญาณ (L) โดยผลที่ได้จากรูปที่ 5.36 – 5.39 โดยรูปบนเป็นสัญญาณอินพุตก่อนเข้าวงจรกลับสัญญาณและรูปล่างเป็นสัญญาณที่ได้จากวงจรกลับสัญญาณซึ่งโมดูลดังกล่าวสามารถทำงานได้อย่างถูกต้อง

5.9.2 การทดลองโมดูลหน่วงสัญญาณ

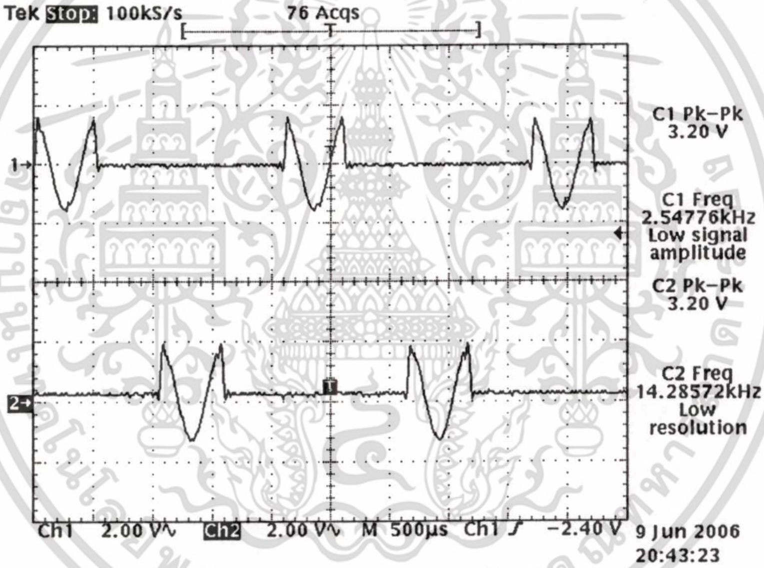
ในการทดสอบโมดูลหน่วงสัญญาณนั้นจะต้องทำการเชื่อมต่อกับโมดูลตัวอื่นๆ อีกเพื่อใช้ในการทดสอบได้แก่ โมดุลกำเนิดสัญญาณนาฬิกา โมดูลกำเนิดสัญญาณใช้ทดสอบ โมดูลเชื่อมต่อกับวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกโดยมีบล๊อกการเชื่อมต่อดังรูปที่ 5.40

เอกสารนี้เป็นของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการดาวน์โหลดโปรแกรมลงบน FPGA โดยสัญญาณที่ใช้ทดสอบจะใช้สัญญาณที่มีการเกิดแบบไม่ต่อเนื่อง (non continuous signal) ซึ่งผลการทดลองโมดูลหน่วงสัญญาณแสดงดังรูปที่ 5.41



รูปที่ 5.40 การทดสอบการทำงานของโมดูลหน่วงสัญญาณ



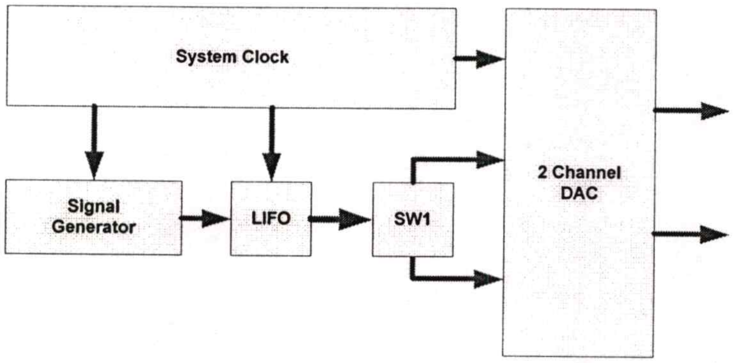
รูปที่ 5.41 สัญญาณที่ผ่านโมดูลหน่วงสัญญาณ 300 แซมเปิ้ล

จากรูปที่ 5.41 สัญญาณที่ป้อนให้กับโมดูลหน่วงสัญญาณเป็นสัญญาณโคซายน์ความถี่ 2 KHz จะถูกหน่วงออกไป 300 ซิมเบิ้ล (2L) ซึ่งแสดงดังรูปล่างเมื่อเทียบกับรูปบนซึ่งเป็นสัญญาณอินพุต

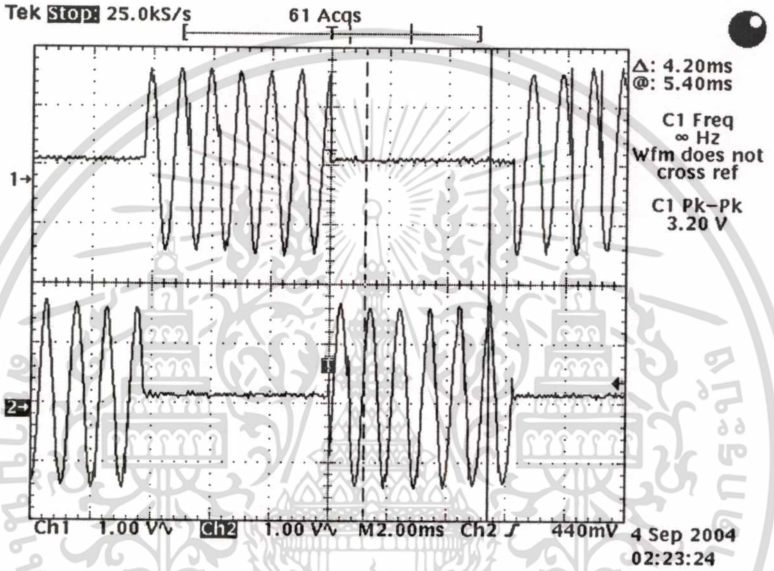
5.9.3 การทดสอบโมดูลสวิทช์มัลติเพล็กซ์ส่วนที่ 1

ในการทดสอบโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 1 ทดลองได้โดยการต่อโมดูลดังกล่าวเข้ากับโมดูลตัวอื่นๆดังรูปที่ 5.42 โดยสัญญาณที่ให้ในการทดสอบจะผ่านโมดูลกลับสัญญาณมาก่อนที่จะเข้าสูโมดูลสวิทช์ส่วนที่ 1 ซึ่งจะทำหน้าที่แยกสัญญาณออกเป็นสองทางให้กับวงจรกรองความถี่ต่อไปโดยสัญญาณที่ได้จากโมดูลนี้แสดงดังรูปที่ 5.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.42 การทดสอบการทำงานของโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 1



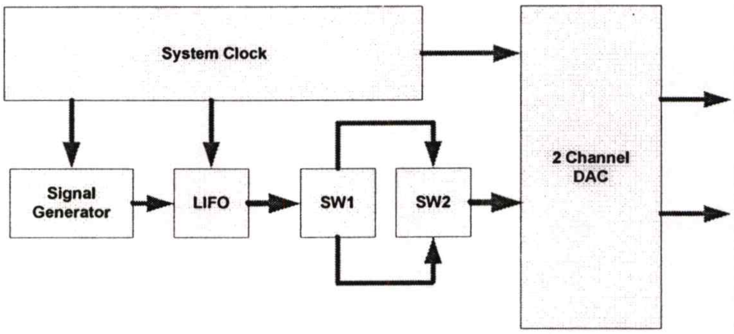
รูปที่ 5.43 สัญญาณที่ได้จากการการทดสอบ โมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 1

ผลการทดสอบการทำงานจากรูปที่ 5.43 บนเป็นสัญญาณที่ทำการมัลติเพล็กซ์ไปยังวงจรกรองความถี่ตัวบนและรูปล่างถูกส่งไปยังวงจรกรองความถี่ตัวล่างซึ่งสัญญาณที่นำผ่านจากวงจรกลับสัญญาณในเวลาจริง

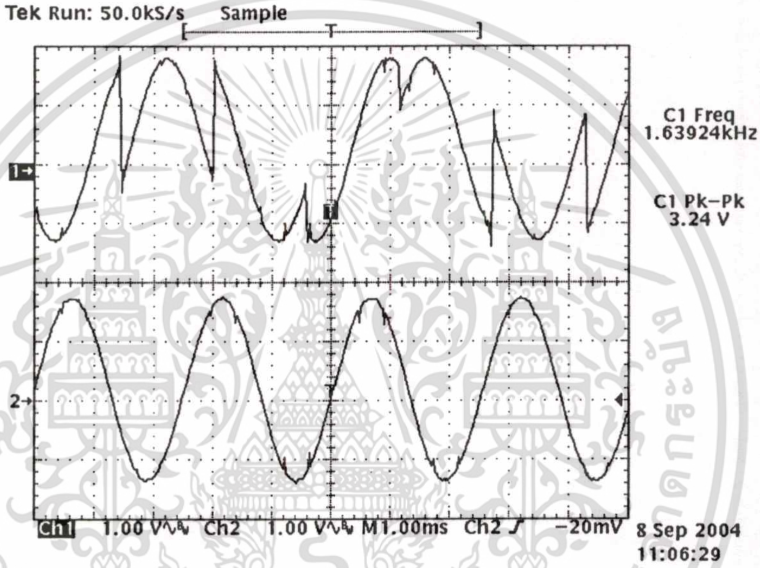
5.9.4 การทดสอบโมดูลสวิทช์มัลติเพล็กซ์ส่วนที่ 2

ในการทดสอบโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 2 ทดลองได้โดยการต่อโมดูลดังกล่าวเข้ากับโมดูลตัวอื่นๆ ดังรูปที่ 5.44 โดยสัญญาณที่ให้ในการทดสอบจะผ่านโมดูลกลับสัญญาณมาก่อนที่จะเข้าสู่โมดูลสวิทช์ส่วนที่ 1 ซึ่งจะทำหน้าที่แยกสัญญาณออกเป็นสองทางให้กับวงจรกรองความถี่ต่อไป จากนั้นจะถูกต่อกลับเข้ามายังโมดูลสวิทช์มัลติเพล็กซ์อีกครั้งโดยสัญญาณที่ได้จากโมดูลนี้แสดงดังรูปที่ 5.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.44 การทดสอบการทำงานของโมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 2

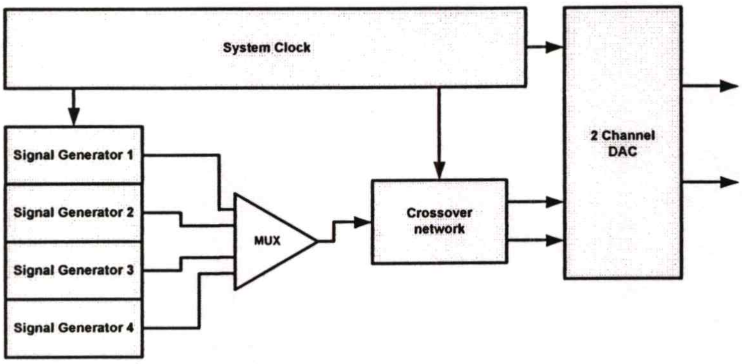


รูปที่ 5.45 สัญญาณที่ได้จากการการทดสอบ โมดูลมัลติเพล็กซ์สัญญาณส่วนที่ 2

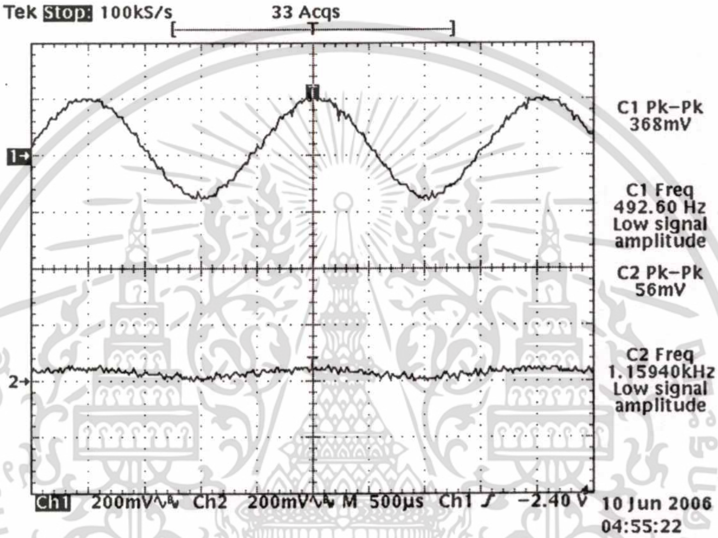
5.9.5 ผลการทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์ค

การทดสอบการทำงานของโมดูลครอสโอเวอร์เน็ตเวอร์คสามารถทดสอบโดยต่อโมดูลร่วมกับโมดูลอื่นดังรูปที่ 5.46 ซึ่งจะมีความแตกต่างจากการทดสอบโมดูลตัวอื่นโดยจะใช้สัญญาณความถี่สูงและความถี่ต่ำรวมกันและ สัญญาณ sine หลายๆ ความถี่เพื่อใช้ในการทดสอบการทำงานของวงจรกรองความถี่ซึ่งผลการทดสอบการทำงานแสดงดังรูปที่ 5.47-5.50

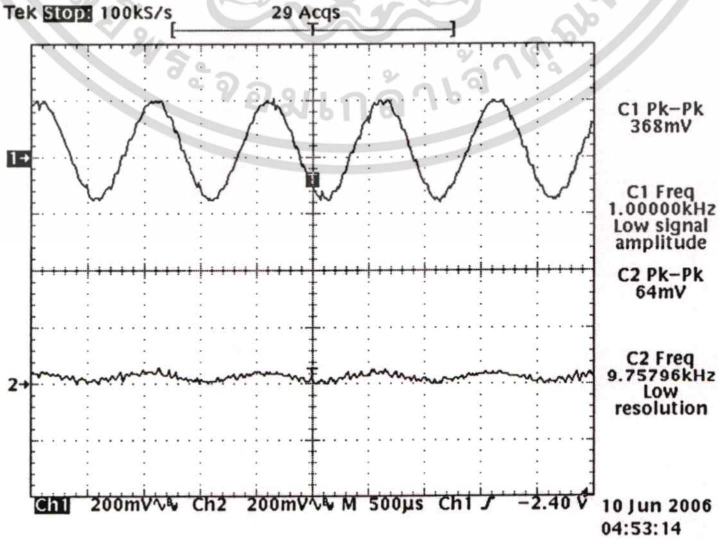
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.46 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์ค

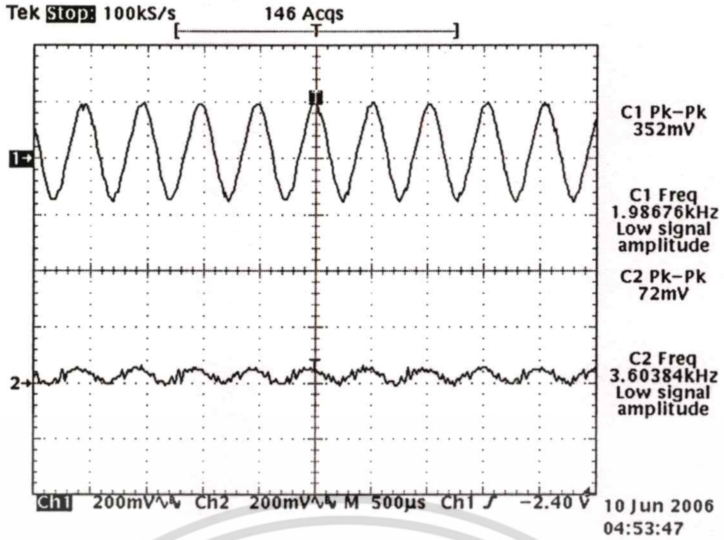


รูปที่ 5.47 การทดสอบ โมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 500 Hz (Ch1 วงจรกรองความถี่ต่ำผ่าน Ch2 วงจรกรองความถี่สูงผ่าน)

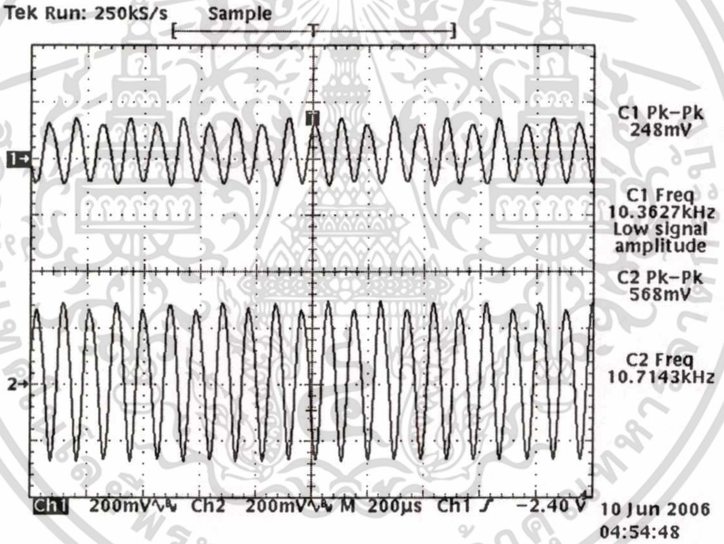


รูปที่ 5.48 การทดสอบ โมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 1 KHz (Ch1 วงจรกรองความถี่ต่ำผ่าน Ch2 วงจรกรองความถี่สูงผ่าน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



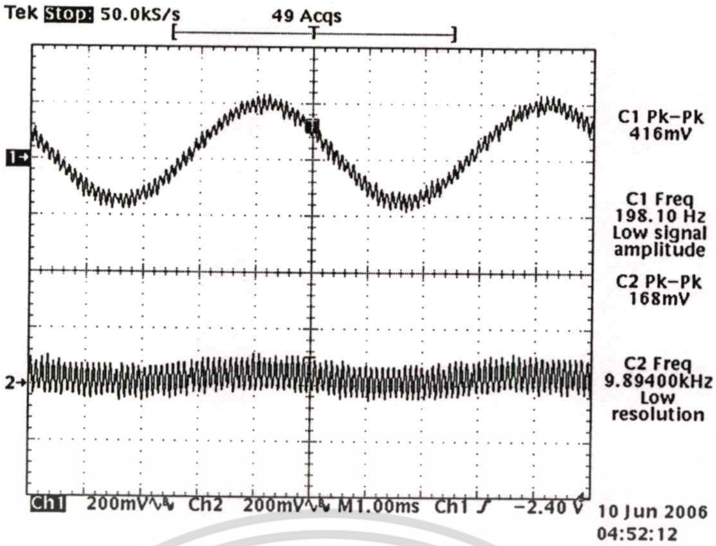
รูปที่ 5.49 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 2 KHz (Ch1 วงจรกรองความถี่ต่ำผ่าน Ch2 วงจรกรองความถี่สูงผ่าน)



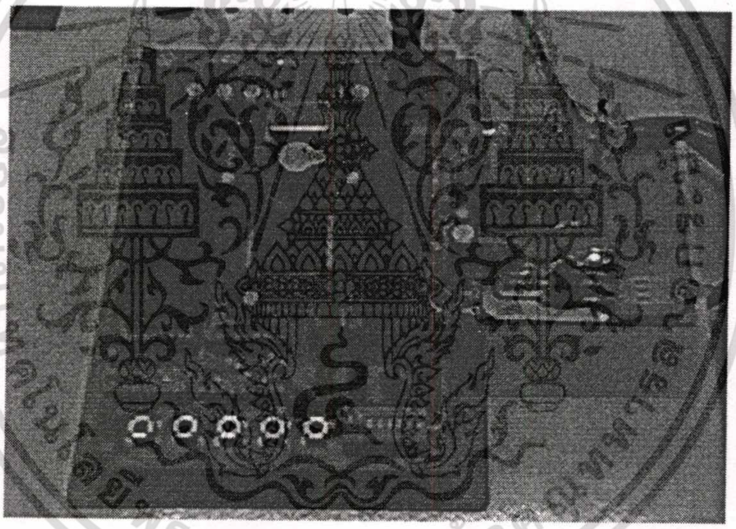
รูปที่ 5.50 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 10 KHz (Ch1 วงจรกรองความถี่ต่ำผ่าน Ch2 วงจรกรองความถี่สูงผ่าน)

จากรูปที่ 5.47 – 5.50 เป็นการทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คโดยป้อนสัญญาณ sine wave ความถี่ 500 Hz, 1 KHz. 2 KHz, 10 KHz โดยผลการทดสอบแต่รูปบนเป็นวงจรกรองความถี่ต่ำ และ รูปล่างเป็นวงจรกรองความถี่สูงจะเห็นว่าเมื่อทำการเพิ่มความถี่ในการทดสอบขนาดของเอาต์พุตความถี่ต่ำมีขนาดลดลงโดยที่เอาต์พุตความถี่สูงจะมีขนาดเพิ่มขึ้น เมื่อเปลี่ยนสัญญาณที่ใช้ในการทดสอบเป็นสัญญาณสองความถี่ ($0.5\cdot\sin(400\cdot\pi\cdot t)+0.1\cdot\sin(2000\cdot\pi\cdot t)$) ผลการทดลองเป็นไปตามรูปที่ 5.51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.50 การทดสอบโมดูลครอสโอเวอร์เน็ตเวอร์คที่ความถี่ 200 Hz รวมกับ 10 KHz (Ch1 วงจรกรองความถี่ต่ำผ่าน Ch2 วงจรกรองความถี่สูงผ่าน)



รูปที่ 5.52 ต้นแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่ใช้ในวิทยานิพนธ์

5.10 บทสรุป

ในบทนี้ได้อธิบายถึงขั้นตอนการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คแบบสองทางที่ให้เฟสเป็นเชิงเส้นโดยได้แบ่งวงจรออกเป็นโมดูลย่อยๆแล้วจึงทำการวิเคราะห์ ออกแบบ ก่อนที่จะนำและทำการทดสอบการทำงานในแต่ละโมดูลย่อย ซึ่งโมดูลของวงจรกรองความถี่นั้นจะมีความซับซ้อนในการสร้างมากที่สุดเนื่องจากเป็นตัวหลักในการประมวลผลซึ่งได้ทำการออกแบบด้วยโปรแกรม MATLAB ที่ได้นำเสนอไว้แล้วในบทที่ผ่านมาซึ่งหลังจากได้ค่าสัมประสิทธิ์แล้วจึงนำเอาไปสร้างวงจรครอสโอเวอร์เน็ตเวอร์ขึ้นมาก่อนที่จะนำโมดูลทั้งหมดไปรวมกันก่อนที่จะนำไปทดสอบบน FPGA อีกครั้ง ซึ่งผลการทดสอบในแต่ละโมดูลจะเห็นว่าเป็นไปตามที่ได้ทำการออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลงานวิจัยและข้อเสนอแนะ

6.1 สรุปผลการดำเนินงานวิจัย

การดำเนินงานวิจัยในวิทยานิพนธ์นี้เป็นการนำเสนอแนวทางเพื่อใช้ในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คบนโครงสร้างที่ให้เฟสเชิงเส้นสมบูรณ์โดยใช้โครงสร้างหลักการคณิตศาสตร์การกระจายในการประมวลผลสัญญาณบน เอฟ พี จี เอ ซึ่งได้มีการกำหนดลักษณะเฉพาะในการออกแบบโดยเลือกใช้โปรแกรม MATLAB, VC++, Xilinx ISE และ ModelSIM เพื่อทำการออกแบบและจำลองผลการทำงานของวงจรกรองความถี่ในระดับฟังก์ชันการทำงานตลอดไปจนถึงในระดับเกต ตามลำดับ โดยในการออกแบบวงจรกรองในระดับฟังก์ชันถ่ายโอนมาเป็นค่าตารางเปิดดู (LUT) ตามรูปแบบการคำนวณด้วยวิธีการของคณิตศาสตร์การกระจายซึ่งได้จากโปรแกรม MATLAB เพื่อทำการหาค่าสัมประสิทธิ์ของฟังก์ชันถ่ายโอนต้นแบบไม่ว่าจะเป็นวงจรกรองแบบบัทเตอร์เวิร์ธ วงจรกรองแบบอิลลิปติค วงจรกรองเชบีเชฟ หรือแม้แต่วงจรกรองอื่น ๆ เมื่อได้ฟังก์ชันของวงจรกรองต้นแบบแล้วจึงทำการแยกฟังก์ชันดังกล่าวให้อยู่ในรูปของฟังก์ชัน all-pass จากนั้นจึงนำค่าสัมประสิทธิ์ของฟังก์ชัน all-pass ไปสร้างตารางต่อไป

แต่อย่างไรก็ตามความยุ่งยากในขั้นตอนการสร้างตารางเปิดดูนั้นจะค่อนข้างมีการคำนวณที่มากและข้อมูลที่เยอะอาจทำให้เกิดความความผิดพลาดในขั้นตอนการเขียนข้อมูลที่อยู่ในรูปของเลขเติมเต็ม 2 (2's complement) ซึ่งจะมีผลต่อวงจรกรองความถี่อย่างมาก ด้วยเหตุผลนี้จึงได้พัฒนาซอฟท์แวร์เพื่อลดขั้นตอนการทำงานและแก้ปัญหาดังกล่าวโดยเพียงนำค่าสัมประสิทธิ์ของฟังก์ชันถ่ายโอนมาป้อนให้กับโปรแกรมซึ่งโปรแกรมจะสร้างไฟล์ตารางเปิดดูในรูปแบบของภาษา verilog ซึ่งทำให้สะดวกในการพัฒนาได้มากยิ่งขึ้น ซึ่งในขั้นตอนการสร้างวงจรครอสโอเวอร์เน็ตเวอร์คจะแบ่งระบบทั้งหมดออกเป็นโมดูลย่อย ๆ ตามความเหมาะสมเพื่อให้ง่ายต่อการพัฒนาและทำการทดสอบการทำงานของแต่ละโมดูลด้วยโปรแกรม XilinxISE และ ModelSIM จากนั้นจึงนำเอาวงจรทั้งหมดโปรแกรมลงบน เอฟ พี จี เอ และทดสอบความถูกต้องของการทำงานอีกครั้ง โดยในขั้นตอนการพัฒนาบน เอฟ พี จี เอ นั้นสามารถดัดแปลงแก้ไขได้โดยไม่จำเป็นต้องดัดแปลงหรือเพิ่มเติมในส่วนของฮาร์ดแวร์แต่อย่างใดทำให้การออกแบบและพัฒนาบน เอฟ พี จี เอ เหมาะสำหรับการพัฒนาและวิจัยอย่างยิ่งก่อนที่จะนำเอาต้นแบบดังกล่าวไปทำในลักษณะเชิงการค้าต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรครอสโอเวอร์เน็ตเวอร์คที่ได้ทำการออกแบบนั้นเมื่อพิจารณาจากผลการสังเคราะห์จะเห็นได้ว่าใช้พื้นที่ทั้งหมด 9 เปอร์เซ็นต์ซึ่งถือว่าน้อยมากและในด้านของประสิทธิภาพการทำงานนั้นโมดูลครอสโอเวอร์เน็ตเวอร์คสามารถทำงานได้ที่ความเร็วสูงสุดถึง 220.216 MHz ซึ่งมากพอสำหรับการนำไปใช้งานในย่านความถี่เสียงซึ่งในวิทยานิพนธ์นี้ใช้ความถี่ในการสุ่มข้อมูลที่ 48 KHz เท่านั้น แต่อย่างไรก็ตามจากผลการทดลองเมื่อใช้สัญญาณอินพุตในช่วงความถี่ตั้งแต่ครึ่งหนึ่งของความถี่ในการสุ่มข้อมูลนั้นในขั้นตอนการแปลงสัญญาณจากสัญญาณอนาลอกไปเป็นดิจิตอลและข้อมูลดิจิตอลไปเป็นสัญญาณอนาลอกจะทำให้สัญญาณในช่วงความถี่บริเวณดังกล่าวมีความผิดเพี้ยนของสัญญาณที่มากขึ้นซึ่งเป็นเรื่องปกติในการขบวนการสุ่มสัญญาณนั่นหมายความว่าหากมีการประมวลผลสัญญาณความถี่ที่สูงขึ้นคุณภาพของสัญญาณที่ได้ก็จะยิ่งแย่ลงเมื่อระบบดังกล่าวยังคงมีอัตราการสุ่มสัญญาณคงที่

6.2 ปัญหาและอุปสรรคที่พบในงานวิจัย

จากที่ได้เสนอมาแล้วข้างต้นซึ่งเป็นการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คที่ให้เฟสเป็นเชิงเส้นสมบูรณ์นั้นมีความซับซ้อนที่สูงมากถึงแม้ว่าจะออกแบบโมดูลครอสโอเวอร์ที่สามารถทำงานที่ความถี่สูง ๆ แต่ว่าเมื่อนำไปต่อรวมกับโมดูลส่วนอื่น ๆ แล้วทำให้ความสามารถในการประมวณผลสัญญาณลดลงซึ่งส่วนใหญ่จะอยู่ที่การเข้าถึงหน่วยความจำ จะใช้เวลาที่มากรวมไปจนถึงการออกแบบระบบสัญญาณนาฬิกาที่ใช้ในการควบคุมทั้งหมดซึ่งจะมีความยุ่งยากในการออกแบบตามความซับซ้อนของวงจร ดังนั้นสิ่งหนึ่งที่จะต้องพิจารณาหลัก ๆ ก็คือความซับซ้อนของวงจรที่ค่อนข้างซับซ้อนมากซึ่งในงานบางประเภทที่ต้องการการประมวลผลที่เร็วและมีหน่วยความจำที่มีไม่มากนักจะไม่เหมาะกับโครงสร้างที่ได้นำเสนอไว้

นอกจากปัญหาที่เกิดขึ้นในขั้นตอนการออกแบบบน FPGA แล้ว ปัญหาอีกประการก็คือการทดสอบการทำงานจริงซึ่งทำให้ขั้นตอนการดำเนินงานวิจัยช้าเป็นอย่างมาก และการเลือกตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก โดยขั้นตอนที่จะนำข้อมูลเข้าไปประมวลผลและนำข้อมูลออกมาแสดงผลซึ่งค่อนข้างมีความซับซ้อนในเรื่องของการจัดการข้อมูล การซิงโครไนซ์ข้อมูล และอัตราในการส่งข้อมูลซึ่งต้องออกแบบให้ตรงกับมาตรฐานของการเชื่อมต่อแบบ I2S ซึ่งหากจังหวะการทำงานไม่ถูกต้องแล้วจะทำให้การแสดงผลการทำงานผิดพลาดขึ้นได้รวมไปจนถึงการเชื่อมต่อระหว่างชิป FPGA เข้ากับอุปกรณ์ภายนอกซึ่งขาอินพุตเอาต์พุตใช้งานได้ในช่วง 3.3V ซึ่งอุปกรณ์ภายนอกที่นำมาเชื่อมต่อด้วยนั้นยังเป็นระดับแรงดันที่สูงกว่าจึงจำเป็นต้องมีการจัดระดับแรงดันให้เหมาะสมเสียก่อนไม่เช่นนั้นจะทำให้เกิดความเสียหายกับ FPGA ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 แนวทางในการพัฒนา

จากวิทยานิพนธ์ที่ได้นำเสนอการสร้างวงจรครอสโอเวอร์เน็ตเวอร์คที่ให้เฟสเชิงเส้นสมบูรณ์บน FPGA มาทั้งหมดนี้สามารถที่จะนำไปพัฒนาให้มีประสิทธิภาพที่ดีขึ้นได้อีกโดยแบ่งออกได้เป็น 3 แนวทางได้แก่

การปรับปรุงประสิทธิภาพในการประมวลผล ซึ่งในวิทยานิพนธ์นี้ออกแบบให้มีการประมวลผลแบบคณิตศาสตร์การกระจายเพื่อเลี่ยงการคูณโดยตรงซึ่งทำให้มีการประมวลผลที่เร็วขึ้นแต่อย่างไรก็ตามด้วยเทคโนโลยีของ VLSI ที่มีการพัฒนาที่ก้าวหน้าไปเรื่อย ๆ ทั้งในเรื่องของขนาดและความเร็วในการทำงานทำให้สามารถนำเอาการประมวลผลด้วยวิธีอื่น ๆ ที่มีประสิทธิภาพดีกว่ามาใช้แทนได้ เช่น การประมวลผลแบบ floating point ที่มีความละเอียดสูงกว่า แต่อย่างไรก็ตามในการออกแบบการประมวลผลใหม่จะต้องทำการออกแบบระบบบัสข้อมูลใหม่ให้เหมาะสมกับตัวประมวลผลด้วย

แนวทางที่สองเป็นการออกแบบให้วงจรครอสโอเวอร์เน็ตเวอร์คนั้นสามารถปรับเปลี่ยนจุดตัดความถี่ได้เนื่องจากในการใช้งานร่วมกับลำโพงนั้นลำโพงแต่ละตัวมีคุณสมบัติที่แตกต่างกันซึ่งจะต้องมาออกแบบจุดตัดความถี่ใหม่ซึ่งทำให้เสียเวลา ดังนั้นหากทำการปรับปรุงให้มีความสามารถดังที่กล่าวในข้างต้นจะทำให้ไม่เสียเวลาในเรื่องการออกแบบใหม่

แนวทางทางสุดท้ายเป็นการลดความซับซ้อนของโครงสร้างที่ได้นำเสนอไว้ไม่ว่าจะเป็นการลดความยาวในวงจรพลิกสัญญาณให้มีความเหมาะสมหรือเปลี่ยนไปเป็นโครงสร้างที่มีกรุ๊ปดีเลย์ที่ใกล้เคียงเชิงเส้นแทนเพื่อความซับซ้อนที่ลดลงทั้งนี้ขึ้นอยู่กับความต้องการของผู้ที่นำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Dietrich Schlichtharle, **Digital Filter Basic and Design**, Springer-Verlag Berlin Heidelberg, New York, 2000.

[2] mmanuel C. Ifeachor, Barrie W. Jervis, **Digital Signal Processing A Practical Approach**, Addision Wesley Publishing Company, 1993.

[3] Fred J. Taylor, **Digital Filter Design Handbook**, Marcel Dekker Inc., 1983.

[4] Justin Romberg , **"Nyquist theorem"**[Online].Available: http://cnx.org/content /m10791/latest

[5] วัลลภ สุระกำพลธร, **การประมวลสัญญาณเชิงเลข**, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2533

[6] Keith Jones, **"Digital filter design with Matlab"** [Online].Available: http://www.physics.uq.edu.au/people/ jones/ph360/lectures/topic4/l4/node10.html

[7] L.R. Rabiner, Bernard Gold, Theory and application of digital signal processing, Prentice-Hall,1973

[8] Vaidyanathan, P. Regalia, P. and Mitra, S, **"Design of doubly-complementary IIR digital filters, using a single complex allpass filter"**, IEEE ICASSP, Vol. 11, pp.2547 – 2550, Apr 1986

[9] Croisier et al, **"Digital Filter for PCM Encoded Signals"**, US Patent 3777130, December 1973.

[10] Abraham Peled, Bede Liu, **"A New Hardware Realization of digital Filters"**, IEEE Trans. On A.S.S.P. December 1974, vol. 22 , No.6

[11] Bernie New, "A Distributed Arithmetic approach to designing Scalable DSP Chips", EDN access for design by design, August 17, 1995

[12] A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review Stanley", IEEE ASSP MAGAZINE, pp 4-19, JULY 1989.

[13] Xilinx Inc.**"Programmable Logic Design Quick Start Hand Book."** [Online]. Available: http://www.xilinx.com

[14] Altera Inc. **"Device Family Overview"** [Online]. Available: http://www.altera.com/products/devices/common/dev-family_overview.html

[15] Actel Inc. **"Products and services"** [Online]. Available: http://www.actel.com/ products/devices.aspx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[16] วัชรากร หนูทอง. **FPGA Design Workshop.** ศูนย์พัฒนาธุรกิจออกแบบวงจรรวม (TIDI) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC). 2545.

[17] Daniel C. Hyde. **Handbook on Verilog HDL**. Computer Science Department Bucknell University. 1997.

[18] Douglas J Smith. **HDL Chip Design.** Madison, AL, USA: Doone Publication. 1999.

[19] Xilinc Inc. **"Spartan 3 PGA"** [Online].Available: http://www.xilinx.com/products/

[20] silicon_solutions/fpgas/spartan_series/spartan3_fpgas/index.htm

[21] Xilinc Inc. **"Spartan III Complete datasheet"** [Online].Available: http://direct.xilinx.com/bvdocs/publications/ds312. pdf

[22] Xilinx Inc. **"Spartan III Functional Description"** [Online].Available: http://duteela.et.tudelft.nl /~elca/ XilinxTutor/documents/ds099-2.pdf

[23] P.Reviriego, Jose Parera and R. Garcia, **"Linear-Phase Crossover Design Using Digital IIR Filters"**, J. Audio Eng.Soc.,Vol. 46, No. 5, pp 406-410, May. 1998.

[24] Siegfried H. Linkkwitz, **"Active Crossover Networks for Nonconincident Drivers"**, IEE Journal of the Audio Engineering Society, Vol. 24 No. 1, pp.1-8, Jan-Feb, 1976.

[25] Tapio Saramaki, **"On the Design of Digital Filters as a Sum of Two All-Pass Filters"**, IEEE Trans., Vol. CAS-32, No. 11, pp. 1191-1193, Nov. 1985.

[26] Phillip A. Regalia and Sanjit K.Mitra, **"A Class of Magnitude Complementary Loundspeaker Crossovers"**, IEEE Trans. ASSP., Vol. 35, No.11, pp.1509-1515, Nov. 1987

[27] Sanjit K. Mitra, **"Digital Signal Processing A Computer-Based Approach"**,2nd Edition, McGraw-Hill Companies Inc, 1998.

[28] P. P. Vaidyanathan A. Sanjit K. Mitra Yrjo Neuvo, **"A New Approach to the Realization of Low-Sensivity IIR Digital Filters"**, IEEE Trans. ASSP., Vol. 34, No.2, pp.350-361, Apar. 1986.

[29] T. Saramaki, **"On the design of digital filters as a sum of two all-pass filters"**, IEEE Transactions on circuit and system,Vol.32, No.11, pp. 1191-1193, Nov. 1985.

[30] Regalia, P.A.; Mitra, S.K.; Vaidyanathan, P.P., **"The digital all-pass filter: a versatile signal processing building block"**, IEEE Proceedings, Volume 76, Issue 1, Jan. 1988 pp. 19 - 37

[31] Scott R. Powell and Paul M. Chaul , **"A Technique for Realizing Linear Phase IIR Filter"**, IEEE Trans, Signal Processing, Vol.39, No.11, pp. 2425-2435, Nov.1991.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[32] A. Kurosu, S. Miyase, S. Tomiyama, T. Takebe, **"A technique to truncate IIR filter impulse response and its application to real-time implementation of linear-phase IIR filters"**, IEEE Trans. ASSP., Vol. 51, PP.1284-2292, May 2003

[33] Powell, S.R.; Chau, P.M., **"AN IMPROVED NONCAUSAL FILTERING TECHNIQUE"**, Signals, Systems and Computers, 1990. 1990 Conference Record Twenty-Fourth Asilomar Conference on Vol. 1, 5-7 Nov 1990, pp.57-62

[34] Powell, S.R.; Chau, P.M., **"Time reversed filtering in real-time"**, Circuits and Systems, 1990., IEEE International Symposium on vol.2, 1-3 May 1990, pp.1239 - 1243 vol.2

[35] Philips Semiconductors, **"I2S bus specification"** [Online].Available: http://www.semiconductors.philips.com/acrobat_download/various/I2SBUS.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมออกแบบวงจรครอสโอเวอร์เน็ตเวอร์ค

```
% 2 ways crossover design
close all; clear all; clc; format long;
FS = 48000; Fn = FS/2;
Wp = 3000/Fn; Ws = 13000/Fn; Rs = 35; Rp = 0.002;
[n, Wn] = ELLIPORD(Wp, Ws, Rp, Rs);
[b,a]= ellip(n,Rp,Rs,Wn);
zr = roots(b);
 pl = roots(a) ;

B0=[pl(1)*pl(2) -(pl(1)+pl(2)) 1];
A0=[1 -(pl(1)+pl(2)) pl(1)*pl(2)];

B1=[-(pl(3)) 1];
A1=[1 -(pl(3))];

N1 = 0.5*(conv(B0,A1));
N2 = 0.5*(conv(B1,A0));

D = conv(A0,A1);
NL = (N1+N2);
NH = (N1-N2);

[HL,T] = freqz(NL,D,1000);
[HH,T] = freqz(NH,D,1000);

t = 1/1000:1/1000:1;
semilogx(t,20*log10(abs(HL)),t,20*log10(abs(HH)))
axis([0 1 -80 5 ])
grid;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปรแกรมสร้างไฟล์ตารางเปิดดู

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>

double a[50] ;
double su;
double Q,qq,b,c,ii,k;
int Bit,Do,Di,i,j,S;
int ind;
FILE *fp0,*pFile;
void main(void)
{
        int tab[20];
        char file1[10]="default.v";
        char file2[10]="tab.txt";
        char str[20] ;
        int K;
//***************************************************************************
        printf("Generate Distribute Arithmetic\n");
        printf("\n Enter coefficient file name (tab.txt):");
        gets(file2);
        pFile = fopen (file2 , "r");
        if (pFile==NULL) perror ("Error opening file");
 else
 {
         for(i=0;i<100;i++)
         {
                 if(fscanf (pFile, "%s", str)==1)
                  a[i]=atof(str);
                  else
                   break;
         }
         fclose (pFile);
 }
 K = i-1;
 printf ("Scalling factor = %f \n",a[K]);
 for(i=0;i<K;i++)
 {
        a[i] =(double)(a[i] * a[K]);
        printf ("Coeficient of a(%d) = %f \n",i,a[i]);
 }
       printf("\n Enter output file name (default.v) :");
       gets(file1);
       if (pFile==NULL) perror ("Error opening file");
       else
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            fp0=fopen(file1,"w+");
            printf("\n Input Data Bus : %d\n",Di);
            Do = 16;
            Bit=Do;
//************************************************************************
            Q  = pow((2),Bit);
            qq = floor(a[0]*(Q-1))/Q;
            b  = pow((2),Di);
            c  = pow((2),Do);
            printf("Q = %10.0f \nB = %f  \nC = %f \n",Q,b,c);
            Bit  = Bit-1;
            Di  = Di-1;
            Do = Do-1;
//************************************************************************
            fprintf(fp0,"module coef1(Di,Do);\n");
            for(ind=0;ind<Di+1;ind++)
            {
                    fprintf(fp0,"// coef(%d) = %f  ;\n",ind,a[ind]);
            }
            fprintf(fp0,"\tinput[%d:0]Di;\n",Di);
            fprintf(fp0,"\toutput[%d:0]Do;\n",Do);
            fprintf(fp0,"\treg[%d:0]Do;\n",Do);
            fprintf(fp0,"\talways @(Di)\n\tbegin\n\t   case(Di)\n");
//************************************************************************
            for(i=0;i<=9;i++)
                    tab[i]=0;
            for (i=0;i<=b-1;i++)
            {
                    ii=i;
                    k=0;
                    for(j=0;j<=Di;j++)
                    {
                            k  = pow((2),(Di-j));
                            if(k<=ii)
                            {         ii=ii-k;
                                      tab[j]=1;}
                            else
                            tab[j]=0;
                    }
                    su = 0;
                    for(ind=0;ind<Di+1;ind++)
                    {
                            printf("%d ",tab[ind]);
                            su = su+(a[ind]*tab[ind]);
                    }
                    printf("= %f\n ",su);
                    coef(su);
            }
//************************************************************************
            fprintf(fp0,"\t     default : Do <= %d'b0;\n",Do+1);
            fprintf(fp0,"\t  endcase\n\t end\nendmodule");
            fclose(fp0);
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

โปรแกรมวงจรครอสโอเวอร์เน็ตเวอร์คบน FPGA

```
module SW1(RL,Di,DT,DB);
   input RL;
   input [15:0] Di;
   output [15:0] DT;
   output [15:0] DB;

   assign DB[15:0]=Di[15:0] & {16{!RL}};
   assign DT[15:0]=Di[15:0] & {16{RL}};

endmodule

module sw_3(DT,DB,LR,A1,A2);
   input [15:0] DT;
   input [15:0] DB;
   input LR;
   output [15:0] A1;
   output [15:0] A2;
   wire [15:0]w1;
   wire [15:0]w2;
   wire [15:0]w3;
   wire [15:0]w4;

  assign w1[15:0]= DT[15:0] & {16{LR}};
  assign w2[15:0]= DB[15:0] & {16{! LR}};
  assign w3[15:0]= DT[15:0] & {16{! LR}};
  assign w4[15:0]= DB[15:0] & {16{LR}};
  assign A1 = w1 | w2 ;
  assign A2 = w3 | w4 ;
endmodule


module LIFO3(Di,Do,CLK,CLR,LR);
   parameter L=150;
   parameter width=16;
   input [width-1:0] Di;
   input  CLK;
   input  CLR;
   output  LR;
         output [width-1:0] Do;
         reg st;
         reg [width-1:0] ram0[L+3:0];
         reg [width-1:0] ram1[L+3:0];
         reg        [7:0]count;
         reg        [width-1:0] x;
         reg                [7:0] ci;          // counter  up to L
         reg                [7:0] co;          // counter up to L
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
integer i;
initial
begin
 for (i = 0; i < L+3; i = i+1)
        ram0[i] = 0;
        ram1[i] = 0;
  end
assign LR=st;
assign Do=x;

always @(posedge CLK)
begin
  if(CLR) begin
        co = L;
        ci = 1;
        st = 0;
        count = 255;
        x = 0;
  end
else  begin
    count = count+1;
    if(count == L) begin
          count = 0;
          st=!st;
          if(st) begin
             ci = ci - 1;
             co = co + 1;
          end
          else begin
             ci = ci + 1;
             co = co - 1;
          end
        end
        if(st) begin
              ram0[co] <= Di;
              x = ram1[ci] ;
              co = co + 1;
              ci = ci - 1;
        end
        else begin
              ram1[ci] <= Di;
              x = ram0[co] ;
              ci = ci + 1;
              co = co - 1;
        end
  end
end

endmodule
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
module Z_Del(Di,Do,Clk,CLR);
   parameter      L =300;
   input CLR;
   input  [15:0] Di;
   output [15:0] Do;
   input  Clk;
   reg    [15:0] ram[L+1:0];
   reg    [15:0] x;
   reg    [8:0] ci;
   reg    [8:0] co;
   integer i;

   initial
   begin
         for (i = 0; i < L+1; i = i+1)
          ram[i] = 0;
   end
          assign Do=x;
           always @(posedge Clk)
           if(CLR) begin
           x <= 0;  ci = 0;   co = 0;
           end
           else begin
                   ram[ci] <= Di;
                   x <=ram[co];
                   co = co+1; ci = ci+1;
                   if(ci == L-1)
                   co = 0;
                   else if(ci == L)
                   ci = 0;
         end
endmodule

module acc(A,B,Y,Co,Ci);
   parameter width = 16 ;
   input [width-1:0] A;
   input [width-1:0] B;
   output [width:0] Y;
   input Ci;
   output Co;
   wire [width-1:0]Do;
    reg   [width:0]result;
    wire t1;
         assign Do=B^{16{Ci}};
         assign t1 = A[15]^Do[15];
         assign Co = result[width]^t1;
         assign Y[16]=Co;
         assign Y[15:0] =  result[width-1:0];
         always @(A or Do or Ci)
                    result = A + Do + Ci;
endmodule
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
module PISO(Di,Do,CK,RST,LD);
    parameter width = 16;
    input [width-1:0] Di;
    output Do;
    input CK,RST;
    input LD;
          reg      [width:0]dat;
          assign Do = dat[0];
          always @ (posedge CK or posedge RST or negedge LD )
          begin
                    if (RST)
                              dat[16:0] = 0  ;
            else begin
                    if(!LD) begin
                              dat[15:0] = Di;
                              dat[16] = 0;
                   end
                   else begin
                              dat[15:0]  = dat[16:1];
                              dat[16] = 0;
                   end
          end
 end
endmodule
```

```
module PISO(Di,Do,CK,RST,LD);
  parameter width = 16;
   input [width-1:0] Di;
   output Do;
   input CK,RST;
   input LD;
          reg [width:0]dat;
          assign Do = dat[0];
          always @ (posedge CK or posedge RST or negedge LD )
          begin
                    if (RST)
                              dat[16:0] = 0  ;
            else begin
                    if(!LD) begin
                              dat[15:0] = Di;
                              dat[16] = 0;
                   end
                   else begin
                              dat[15:0]  = dat[16:1];
                              dat[16] = 0;
                   end
          end
 end
endmodule
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] J.Orsawat, D. Sookcharoenphol and K. Janjitrapongvej, “An Implementation of Digital Crossover Network by using Perfect Linear Phase IIR Filters”, The Third International Symposium on Communications and Information Technologies (ISCIT 2003)

[2] C. Kanna, J.Orsawat, D. Sookcharoenphol and K. Janjitrapongvej,“Finite Word Length Effect of Digital Crossover Network using Perfect Linear Phase IIR Filters” , The 4 th Information and Computer Engineering Postgraduate Workshop (ICEP 2004)

[3] เจตน์ ออสวัสดิ์, ดลชัย สุขเจริญผล และ กนก เจนจิระพงศ์เวช ,"การออกแบบการลดทอนในช่วงหยุดความถี่ผ่านอย่างสูงของวงจรกรองความถี่ IIR ที่มีเฟสเป็นเชิงเส้น " , วารสารพระจอมเกล้าลาดกระบัง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เดือน เมษายน 2547 ฉบับที่ 2 ปีที่ 16

[4] ชาญณรงค์ การนา, เจตน์ ออสวัสดิ์, ดลชัย สุขเจริญผล และ กนก เจนจิระพงศ์เวช, " วงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คแบบ ใหม่ที่ให้กรุ๊ปดีเลย์ ราบเรียบ", จัดการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 27 "EECON-27 (27th Electrical Engineering Conference)"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISSN 0858-5430

วารสารพระจอมเกล้าลาดกระบัง

Vol.11 No.2 August 200
ปีที่ 11 ฉบับที่ 2 สิงหาคม 254

- คณะวิศวกรรมศาสตร์
- คณะสถาปัตยกรรมศาสตร์
- คณะครุศาสตร์อุตสาหกรรม
- คณะวิทยาศาสตร์
- คณะเทคโนโลยีการเกษตร
- คณะเทคโนโลยีสารสนเทศ
- โครงการคณะอุตสาหกรรมเกษตร
- บัณฑิตวิทยาลัย
- วิทยาเขตชุมพร
- สำนักวิจัยและบริการคอมพิวเตอร์
- สำนักหอสมุดกลาง
- โครงการสำนักวิจัยการสื่อสาร
  และเทคโนโลยีสารสนเทศ

การออกแบบการลดทอนในช่วงหยุดความถี่ผ่านอย่างสูง ของวงจรกรองความถี่ IIR ที่มีเฟสเป็นเชิงเส้น

On the Design of High Stopband Attenuation of Linear Phase IIR Filters

เจตน์ ออสวัสดิ์ — นักศึกษาปริญญาโท
ดลชัย สุขเจริญผล — อาจารย์
กนก เจนจิระพงศ์เวช — รองศาสตราจารย์
ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

บทความนี้เสนอการสร้างวงจรกรองความถี่ที่ให้ผลตอบสนองของขนาดต่อความถี่มีการลดทอนสัญญาณในช่วงหยุดความถี่ผ่านมีขนาดสูงมาก โดยอาศัยวงจรกรองความถี่แบบผ่านตลอดที่ใช้การต่อแบบคาสเคดที่ใช้ฟังก์ชั่นถ่ายโอนที่มีอันดับเหมือนกัน โดยผลตอบสนองความถี่ทั้งวงจรกรองความถี่ต่ำผ่านและวงจรกรองความถี่สูงผ่านจะมีผลตอบสนองตรงกันข้าม โดยผลจากการจำลองด้วยโปรแกรมแสดงให้เห็นชัดเจนว่า ผลของการลดทอนในช่วงหยุดความถี่ของวงจรกรองทั้ง 2 จะมีขนาดเป็น 4 เท่าของวงจรกรองความถี่เดิมโดยทั่วไป โดยผลรวมตลอดช่วงความถี่ยังมีคุณสมบัติเป็นฟังก์ชั่นของ all-pass

คำสำคัญ : องค์ประกอบวงจรกรองความถี่ผ่านตลอด ความถี่ IIR ที่มีเฟสเป็นเชิงเส้น การลดทอนในช่วงหยุดความถี่ผ่าน

Abstract

This paper presents an implementation of allpass complementary using identical cascade linear phase IIR filters and both transfer functions have the same polynomial degree. The proposed circuits have a high attenuation at both stopband magnitude response complementary by using the allpass filter. From simulated results, it is shown that the stopband attenuation has 4 tune of steeper rate than does the conventional filter. Their magnitude behavior is still an all-pass property.

Keywords : Allpass complementary filter, linear phase IIR filters, stopband attenuation

1. บทนำ

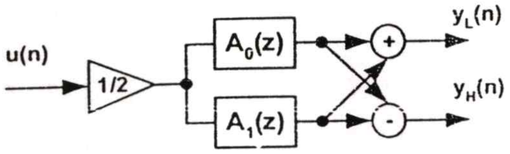
ในการสร้างวงจรกรองความถี่ดิจิตอลที่ต้องการค่าลดทอนของสัญญาณในช่วงที่ไม่ต้องการให้มีสัญญาณผ่านได้ (stopband) ให้มีค่าสูง (> -100 dB) สามารถออกแบบได้ยาก หากเราใช้โพลิโนเมียลแบบเอลลิปติกหรือเชฟบีเชบ แต่อย่างไรก็ตามการใช้โพลิโนเมียลดังกล่าวมีข้อด้อยหลายประการคือ การมีขนาดลูกคลื่นที่มากไปในช่วง passband, ผลตอบสนองของเฟสต่อความถี่ไม่เป็นเชิงเส้นหรือโดยประมาณให้เป็นเชิงเส้น และนอกจากนี้ยังไม่สามารถทำให้มีคุณสมบัติเป็น allpass complementary ได้ [1] ดังนั้นหากเราต้องการออกแบบให้วงจรกรองความถี่ดิจิตอลให้มีคุณลักษณะเป็น allpass complementary ดังเอกสาร [1], [2] และ [3] เราสามารถสร้างได้โดยการใช้โครงสร้างแบบวงจร all-pass filter (APF) ต่อขนานกันเพียง 2 วงจรดังรูปที่ 1 โดยสัญญาณด้านความถี่ต่ำผ่านและความถี่สูงผ่านคือ $Y_L(n)$ และ $Y_H(n)$ ตามลำดับ และเราสามารถเขียนสมการฟังก์ชั่นถ่ายโอน (transfer function) ได้ดังต่อไปนี้

$$H_L(z) = \frac{1}{2}\left[A_0(z)+A_1(z)\right] \qquad (1)$$

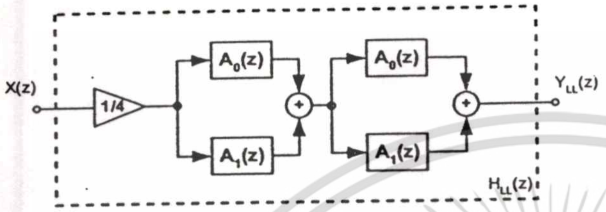
$$H_H(z) = \frac{1}{2}\left[A_0(z)+A_1(z)\right] \qquad (2)$$

เมื่อ $H_L(z)$ เป็นฟังก์ชั่นถ่ายโอนของวงจรกรองความถี่ต่ำผ่าน และ $H_H(z)$ เป็นฟังก์ชั่นถ่ายโอนของวงจรกรองความถี่สูงผ่าน $A_0(z)$ และ $A_1(z)$ เป็นวงจรกรองความถี่ผ่านตลอด (APF) ที่มีเสถียรภาพ

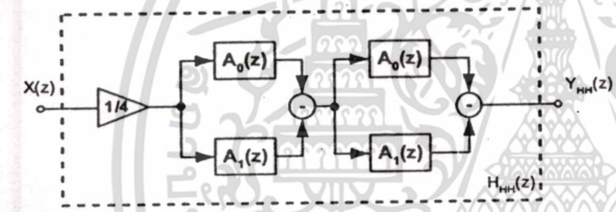
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 แสดงการสร้างวงจรกรองความถี่ต่ำผ่านและความถี่สูงผ่านโดยสร้าง จากวงจรความถี่ APF ต่อขนานกัน 2 วงจร



รูปที่ 2 แสดงการต่อแบบคาสเคดของวงจรความถี่ต่ำผ่าน (LPF) ที่สร้าง จากการต่อขนานของวงจรกรอง ความถี่แบบ APF 2 วงจร



รูปที่ 3 แสดงการต่อแบบคาสเคดของวงจรกรองความถี่สูงผ่าน (HPF) ที่สร้างจากการต่อขนานของวงจรกรอง ความถี่แบบ APF 2 วงจร

สำหรับวงจรกรองความถี่ต่ำผ่านต้นแบบเรากำหนดให้มีค่าต่าง ๆ ดังต่อไปนี้

Order	$n = 3$
Normalized frequency	$\omega_n = 0.3$
Ripple pass band	$\delta_p = 0.005$ dB
Ripple stop band	$\delta_s = 30$ dB

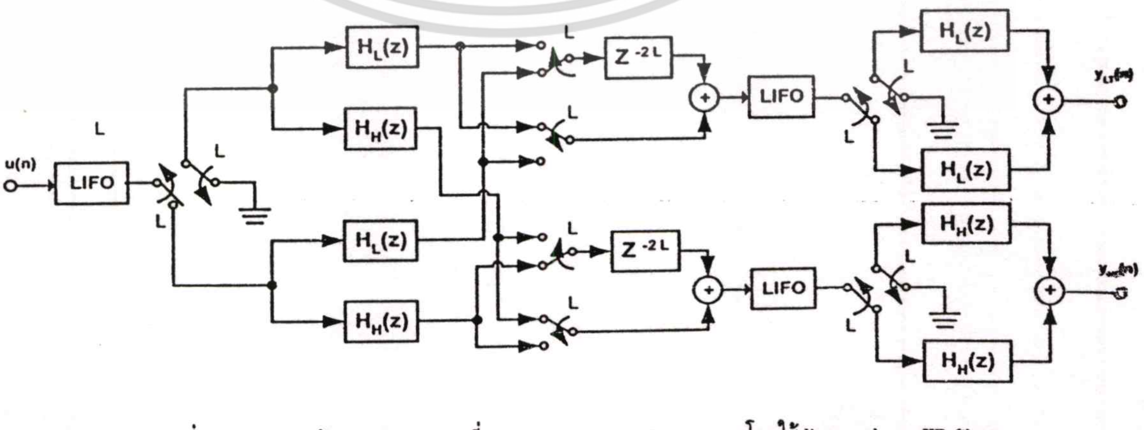
จากการกำหนดสเปคของการออกแบบวงจรกรองต่ำผ่าน ตามค่าตัวแปรที่ได้กำหนดไว้ เราสามารถเขียนสมการฟังก์ชั่นถ่ายโอนใน z-domain ได้ดังนี้

$$H_L(z) = \frac{0.2113729765 + 0.50047724171z^{-1} + 0.50047724171z^{-2} + 0.21137290765z^{-3}}{1 - 0.00006179502z^{-1} + 0.42451400628z^{-2} + 0.00075191252z^{-3}} \quad (3)$$

และจากเอกสารอ้างอิง [3] และสมการที่ (1) เราสามารถแยกสมการฟังก์ชั่นถ่ายโอนใน z-domain ของวงจรกรองความถี่ต่ำผ่าน (LPF) ตัวต้นแบบได้เป็นฟังก์ชั่นถ่ายโอนซึ่งอยู่ในรูปฟังก์ชั่น all-pass ได้ 2 ฟังก์ชั่นดังต่อไปนี้คือ

$$A_0 = \frac{-0.00177121874 + z^{-1}}{1 - 0.00177121874z^{-1}}$$

$$A_1 = \frac{0.42451703404 + 0.00170942371z^{-1} + z^{-2}}{1 + 0.00170942371z^{-1} \quad 0.42451703404z^{-2}} \quad (4)$$



รูปที่ 4 แสดงการสร้างวงจรกรองความถี่แบบ all-pass complementary โดยใช้ linear phase IIR filters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่าสัมประสิทธิ์ของฟังก์ชั่น all-pass ทั้ง 2 คือ $A_0(Z)$ และ $A_1(Z)$ จะเป็นค่าจริง (real) โดยโพลิโนเมียลเทอมเศษและโพลิโนเมียลเทอมส่วนจะมีลักษณะเป็น mirror-image polynomial และอันดับของ $A_0(Z)$ จะน้อยกว่า $A_1(Z)$ อยู่ 1 อันดับเสมอ และจากความสัมพันธ์ในสมการ (2) เราสามารถเขียนฟังก์ชั่นถ่ายโอนของวงจรกรองความถี่สูงผ่าน (HPF) ที่สร้างจาก $A_0(Z)$ และ $A_1(Z)$ ได้ดังนี้คือ

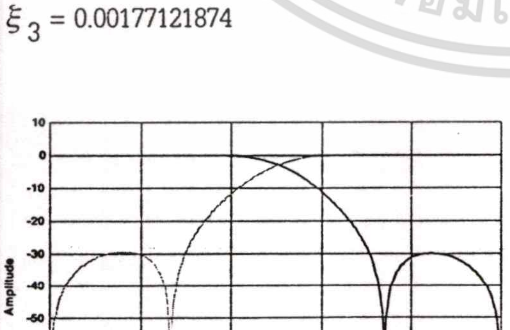
$$H_H(z) = \frac{-0.21314412639 + 0.49951973052z^{-1} + 0.49951973052z^{-2} + 0.21314412639z^{-3}}{1 - 0.00006179502z^{-1} + 0.42451400628z^{-2} - 0.00075191252z^{-3}} \quad (5)$$

โดยเราจะสังเกตได้ว่าฟังก์ชั่นถ่ายโอนทั้ง 2 จะมีข้อสังเกตดังต่อไปนี้

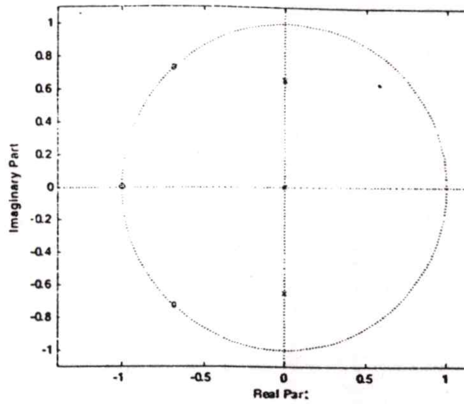
1. โพลิโนเมียลเทอมส่วน (denominator polynomial) ของทั้ง $H_L(z)$ และ $H_H(z)$ จะเหมือนกัน
2. โพลิโนเมียลเทอมเศษ (numerator polynomial) ของ $H_L(z)$ จะมีลักษณะเป็น symmetric polynomial
3. โพลิโนเมียลเทอมเศษ (numerator polynomial) ของ $H_H(z)$ จะมีลักษณะเป็น antisymmetric polynomial

เราสามารถหาค่ารากของโพลิโนเมียลเทอมส่วนได้ค่าราก 3 ค่า ซึ่งมีค่าดังต่อไปนี้

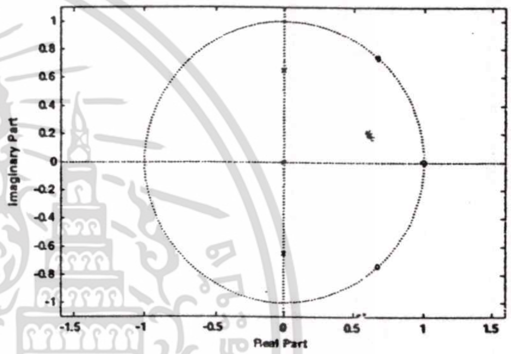
$$\xi_1 = -0.00085471185 + 0.65154915663i$$
$$\xi_2 = -0.00085471185 - 0.65154915663i$$
$$\xi_3 = 0.00177121874$$



รูปที่ 5 ผลตอบสนองของขนาดต่อความถี่ทั้งของวงจรกรองความถี่ต่ำและวงจรกรองความถี่สูง ซึ่งทั้ง 2 มีลักษณะเป็น allpass complementary



รูปที่ 6 แสดงการพล็อตตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่ต่ำตัวต้นแบบ $H_L(Z)$



รูปที่ 7 แสดงการพล็อตตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่สูง $H_H(Z)$

2. Identical Cascade Linear Phase IIR Filter

จากรูปที่ 5 จะสังเกตเห็นว่าผลตอบสนองของขนาดต่อความถี่ทั้งของวงจรกรองความถี่ต่ำผ่านและวงจรกรองความถี่สูงผ่านจะสมมาตรกันที่ความถี่ 0.5 rad/sec (normalized) และการลดทอนในช่วงหยุดความถี่ผ่านของวงจรกรองความถี่ทั้ง 2 จะมีขนาดเท่ากันและตรงกับที่ได้กำหนดไว้ในการออกแบบคือ -30 dB นอกจากนี้จะเห็นว่าผลรวมของขนาดต่อความถี่ตลอดทุกช่วงความถี่จะมีค่าคงที่ซึ่งตรงกับคุณสมบัติของฟังก์ชั่น all-pass และเมื่อเราต้องการการลดทอนในช่วงหยุดผ่านให้มีค่ามากกว่านี้โดยที่ผลตอบสนองของขนาดต่อความถี่ตลอดช่วงความถี่ยังมีค่าคงที่ ซึ่งตรงกับคุณสมบัติของ allpass complementary เราสามารถสร้างวงจรกรองความถี่ดังกล่าวขึ้นมาได้โดยการนำวงจรกรองความถี่ 2 วงจรที่มีฟังก์ชั่นถ่ายโอนเหมือนกันและอันดับเท่ากันมาต่อคาสเคดดังในรูปที่ 2 สำหรับวงจรกรองความถี่ต่ำผ่าน (LPF) ดังนั้นเราจะได้เป็นวงจรกรองความถี่ต่ำผ่านอันใหม่ซึ่งสามารถเขียนสมการ

เอกสารนี้เป็นเอกสารที่... ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นถ่ายโอนได้ดังนี้

$$H_{LL}(z) = \frac{0.0446785 + 0.2115746z^{-1} + 0.4620521z^{-2} + 0.5903119z^{-3}}{1 - 0.0001235z^{-1} + 0.8490280z^{-2} - 0.0015562z^{-3}}$$

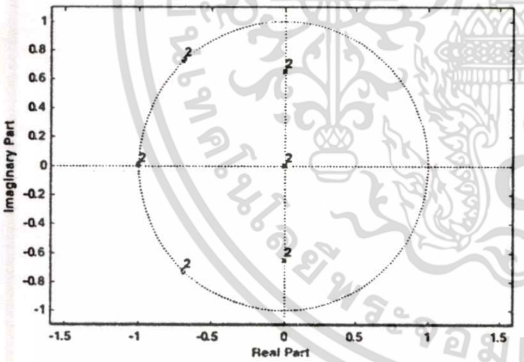
$$+ \frac{0.4620521z^{-4} + 0.2115746z^{-5} + 0.0446785z^{-6}}{+ 0.1802122z^{-4} - 0.0006383z^{-5} + 0.00000056z^{-6}} \qquad (6)$$

สำหรับวงจรกรองความถี่สูงผ่านเราสามารถสร้างได้ดังรูปที่ 3 ซึ่งเราสามารถเขียนสมการฟังก์ชั่นถ่ายโอนได้ดังนี้

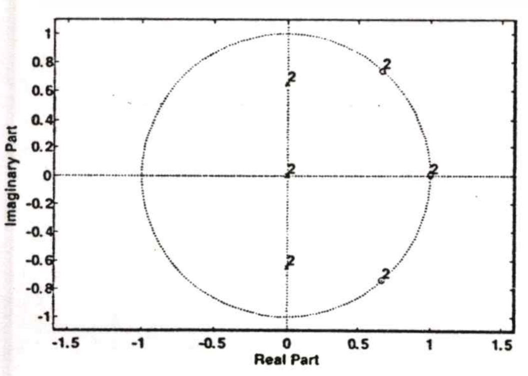
$$Y_{HH}(z) = \frac{0.04543041 - 0.21293939z^{-1}\ 0.4624593z^{-2} - 0.5899007z^{-3}}{1 - 0.0001235z^{-1} + 0.8490280z^{-2} - 0.0015562z^{-3}}$$

$$+ \frac{0.46245935z^{-4} - 0.21293939z^{-5} + 0.04543041z^{-6}}{+ 0.1802122z^{-4} - 0.0006383z^{-5} + 0.00000056z^{-6}} \qquad (7)$$

โดยฟังก์ชั่นถ่ายโอนใหม่ของวงจรกรองความถี่ทั้ง 2 วงจรยังคงมีลักษณะเหมือนกับลักษณะของฟังก์ชั่นการถ่ายโอน ดังที่กล่าวมาแล้วทั้ง 3 ข้อ และเมื่อเรานำสมการฟังก์ชั่นถ่ายโอนทั้ง 2 ไปพล็อตตำแหน่งของโพลและซีโร่จะแสดงดังรูปที่ 8 และรูปที่ 9 โดยที่เลข 2 แสดงถึงซีโร่และโพลทับซ้อน



รูปที่ 8 แสดงการพล็อตตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่ต่ำ $H_{LL}(z)$



รูปที่ 9 แสดงการพล็อตตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่สูง $H_{HH}(z)$

จากรูปที่ 8 และรูปที่ 9 จะสังเกตเห็นว่าตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่ทั้ง 2 ที่ใช้การต่อแบบคาสเคส 2 วงจรที่มีฟังก์ชั่นถ่ายโอนเหมือนกันจะมีตำแหน่งเหมือนกับตำแหน่งของโพลและซีโร่ของวงจรกรองความถี่ที่มีฟังก์ชั่นถ่ายโอนคือ $H_L(Z)$ และ $H_H(Z)$ ดังแสดงในรูปที่ 6 และรูปที่ 7 และผลตอบสนองของขนาดกำลัง 2 ต่อความถี่ (magnitude-square) สามารถหาได้จากสมการที่ 8 ดังนี้

$$|H_{eq}(z)|^2 = H_T(z) \cdot H_T(z^{-1})| \qquad (8)$$

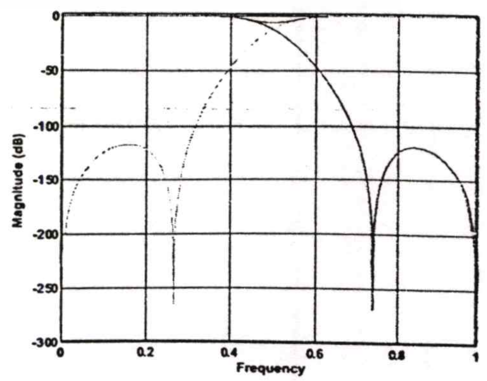
และสำหรับผลตอบสนองของขนาดกำลัง 2 ของวงจรกรองความถี่ต่ำผ่าน (LPF) เรากำหนดให้มีความสัมพันธ์ดังสมการต่อไปนี้

$$H_T(z) = H_{LL}(z) \cdot H_{LL}(z^{-1}) \qquad (9)$$

และสำหรับวงจรกรองความถี่สูง (HPF)

$$H_T(z) = H_{HH}(z) \cdot H_{HH}(z^{-1}) \qquad (10)$$

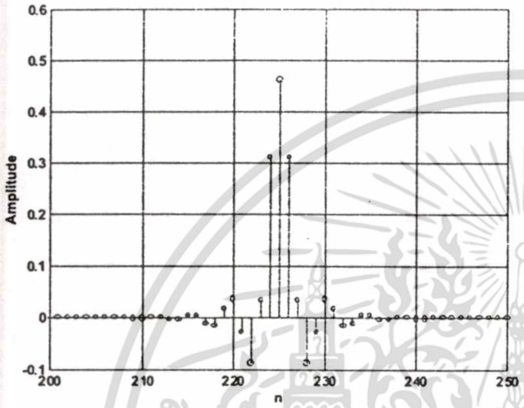
ดังนั้นเมื่อเรานำบล็อกของการต่อแบบคาสเคดที่มีฟังก์ชั่นถ่ายโอนเหมือนกันไปแทนในวงจรกรองความถี่ย่อย (sub filter H(z)) ในระบบที่นำเสนอโดยเพาเวลและชุน [4] ซึ่งเป็นการสร้างวงจรกรองความถี่ต่ำผ่านที่ให้ผลตอบสนองของเฟสเป็นเชิงเส้นและได้วิเคราะห์การทำงานของระบบด้วยสมการ state-space และเงื่อนไขในการทำให้ระบบเป็นระบบเชิงเส้นไม่แปรเปลี่ยนตามเวลา (linear- time invariance) ดังแสดงในเอกสาร [5] ดังนั้นในการหาผลตอบสนองของขนาดต่อความถี่ของวงจรกรองความถี่ ทั้งวงจรกรองความถี่ต่ำผ่าน (LPF) และวงจรกรองความถี่สูงผ่าน (HPF) ของระบบที่นำเสนอทั้งหมดดังปรากฏในรูปที่ 4 ได้แสดงไว้ดังรูปที่ 10



รูปที่ 10 แสดงผลตอบสนองของขนาดต่อความถี่ทั้ง LPF และ HPF ที่นำเสนอ

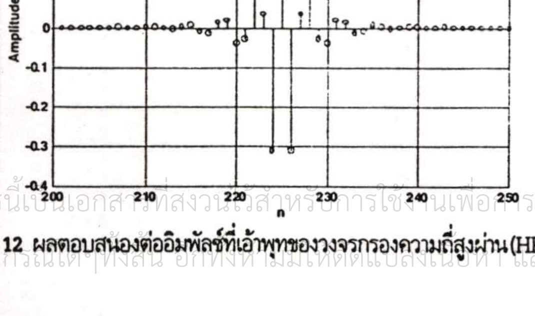
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 10 จะสังเกตเห็นว่าผลตอบสนองของขนาดต่อความถี่ (normalized) ของทั้งวงจรกรองความถี่ต่ำผ่าน (LPF) และวงจรกรองความถี่สูงผ่าน (HPF) ทั้งด้าน passband และ stopband จะมีลักษณะสมมาตรกันรอบจุดตัดความถี่ (ωc = 0.5 rad/sec.) และจะเห็นได้ชัดเจนว่าค่าการลดทอนในช่วง stopband ของวงจรกรองความถี่ทั้ง 2 มีค่าเท่ากันคือ -120 dB (เป็น 4 เท่าของวงจรกรองความถี่ต้นแบบ)



รูปที่ 11 ผลตอบสนองต่ออิมพัลซ์ที่เอ้าพุทของวงจรกรองความถี่ต่ำผ่าน (LPF)

จากรูปที่ 11 และรูปที่ 12 เป็นผลตอบสนองต่ออิมพัลซ์ โดยวัดที่ด้านเอ้าพุทของวงจรกรองความถี่ต่ำผ่าน $y_{LT}(n)$ และที่ด้านเอ้าพุทของวงจรกรองและผลรวมของขนาดต่อความถี่ตลอดช่วงความถี่ทั้งหมด จะมีค่าคงที่ซึ่งเป็นคุณสมบัติที่สำคัญของวงจรกรองแบบ APF และความถี่สูงผ่าน $y_{HT}(n)$ โดยทั้ง 2 รูปจะสังเกตเห็นว่าผลตอบสนองต่ออิมพัลซ์ที่อยู่รอบ ๆ จุดศูนย์กลาง (ค่าที่มีขนาดสูงสุด) จะมีลักษณะสมมาตรโดยประมาณ ซึ่งจะใกล้เคียงกับคุณสมบัติของวงจรกรองความถี่ที่มีลักษณะเฟสเป็นเชิงเส้นแบบที่ 1



รูปที่ 12 ผลตอบสนองต่ออิมพัลซ์ที่เอ้าพุทของวงจรกรองความถี่สูงผ่าน (HPF)

3. สรุป

บทความนี้เสนอการสร้างวงจรกรองความถี่ที่ให้ผลตอบสนองของขนาดต่อความถี่มีการลดทอนสัญญาณในช่วงหยุดความถี่ผ่านมีขนาดสูงมาก โดยอาศัยวงจรกรองความถี่แบบผ่านตลอดที่ใช้การต่อแบบคาสเคสที่ใช้ฟังก์ชั่นถ่ายโอนที่มีอันดับเหมือนกัน โดยผลตอบสนองความถี่ทั้งวงจรกรองความถี่ต่ำผ่านและวงจรกรองความถี่สูงผ่าน จะมีคุณสมบัติเป็น allpass complementary และผลจากการจำลองด้วยโปรแกรมแสดงให้เห็นชัดเจนว่าผลของการลดทอนในช่วงหยุดความถี่ของวงจรกรองทั้ง 2 จะมีขนาดเป็น 4 เท่าของวงจรกรองความถี่ต้นแบบ โดยผลรวมตลอดช่วงความถี่ยังมีคุณสมบัติเป็นฟังก์ชั่นของ all-pass และผลตอบสนองต่ออิมพัลซ์ได้แสดงให้เห็นว่าวงจรกรองที่นำเสนอมีความสมมาตรรอบ ๆ จุดศูนย์กลางของวงจรกรองทั้ง 2 แบบ

เอกสารอ้างอิง

[1] Tapio Saramaki, "On the Design of Digital Filters as a Sum of Two All-Pass Filters", **IEEE Trans.**, Vol. CAS- 32, No. 11, pp. 1191-1193, Nov. 1985

[2] Phillip A. Regalia, Sanjit K. Mitra and P. P. Vaidyanathan, "The Digital All-Pass Filter : A Versatile Signal Processing Building Block", **Proc. IEEE**, Vol. 76, No. 1, Jan. 1988.

[3] P. P. Vaidyanathan A. Sanjit K. Mitra Yrjo Neuvo, "A New Approach to the Realization of Low-Sensivity IIR Digital Filters", **IEEE Trans. ASSP.**, Vol. 34, No.2, pp. 350-361, Apar. 1986.

[4] Scott R. Powell and Paul M. Chaul, "A Technique for Realizing Linear Phase IIR Filter", **IEEE Trans**, Signal Processing, Vol.39, No.11, PP. 2425-2435, Nov.1991.

[5] D. Sookcharoenphol, K. Janjitrapongvej and S. Tomiyma, "An anaysis of linear phase IIR filters and a condition for perfect linear phase", **ICEP 2003**, pp. 144-147, Jan., 2003.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISCIT2003

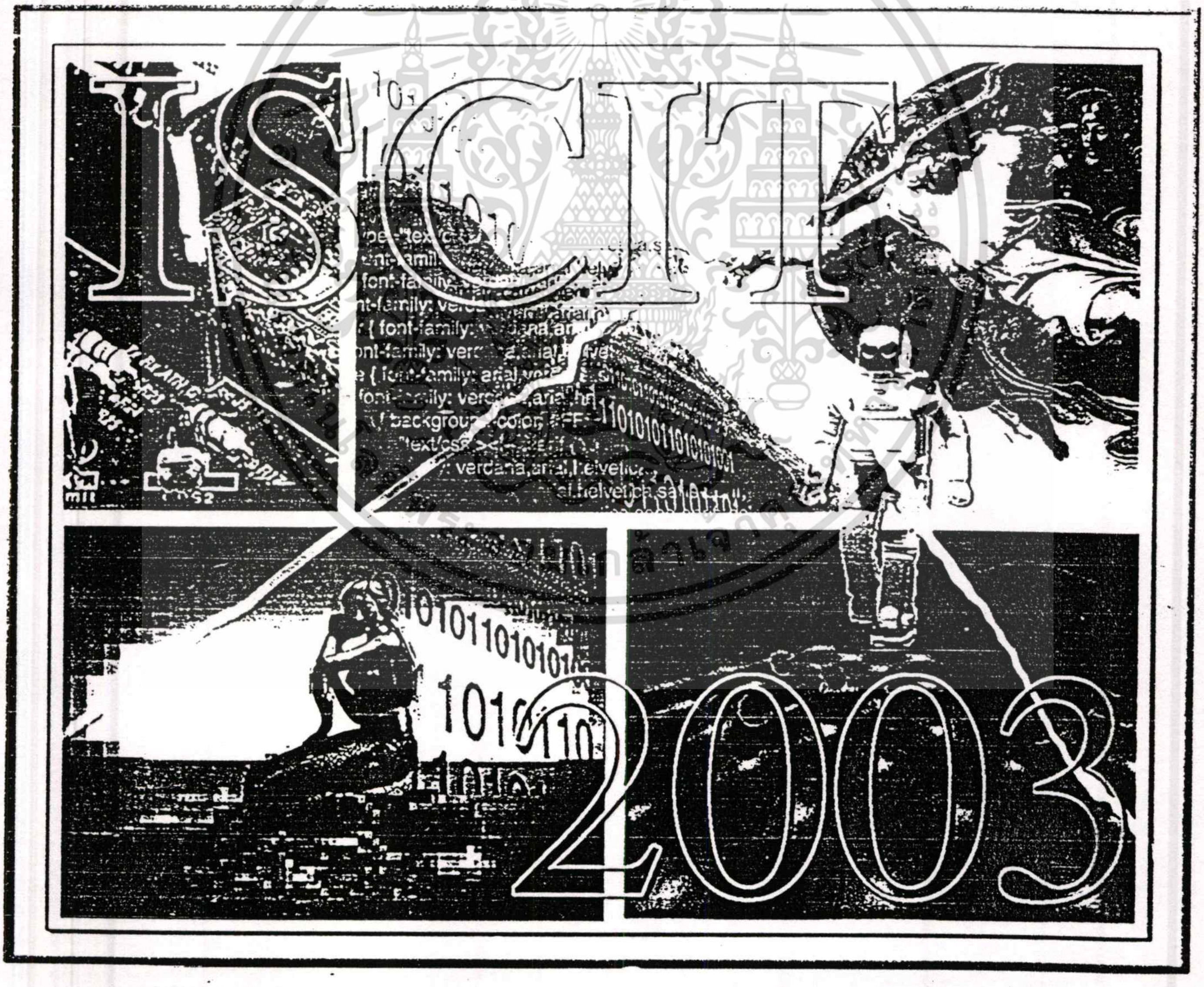
Volume II

Proceedings

The Third International Symposium on Communications and Information Technologie

September 3-5, 2003

BP Samila Beach Hotel and Resort, Songkhla, Thailand



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

CoE PSU NECTEC CS LOXINFO

An Implementation of Digital Crossover Network by using Perfect Linear Phase IIR Filters

J. Orsawat, D. Sookcharoenphol and K. Janjitrapongvej.
Information Dept., Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang Bangkok THAILAND
Email: ksdolcha@kmitl.ac.th

Abstract

In this paper, the implementation technique of digital crossover network using perfect linear phase IIR filters is presented. This system has various advantages which can not be obtained from analog crossover network such as linear phase response, flat group delay and sharp cut-off at low-order over audio frequency band. The simulation results show that the group delay response is maximally flat and twice more attenuation in stop-band than the prototype elliptic IIR filter at all desired frequency.

1. Introduction

Classical design of two-way digital crossover networks consist of low-pass filter and high-pass filter to split the audio signal into low-pass signal and high-pass signal, and each feeding to a separated loudspeaker. However, the amplitude, phase and group delay distortions can not be eliminated and it is difficult to achieve perfect reproduction [1] and [2]. We can derived the odd-order digital low-pass filter using the Bilinear transform and it can be implemented as a sum of two all-pass filters [3]. Two stable all-pass filters (APF) which transfers function pairs that is satisfied to all-pass complementary property was proposed by Regalia and et al [4]. It is well known that this two stable APF have less sensitivity and circuit complexity than the classical design. An implementation of the all-pass complementary filter pairs as the sum and difference of all pass functions is shown in fig1

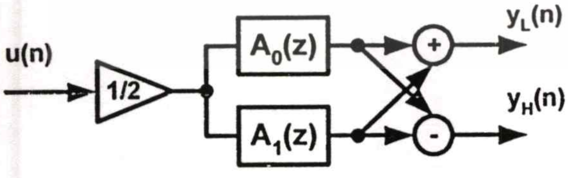


Fig. 1 Two-parallel APF is implementing of two-way crossover network.

The implementation of two-way digital crossover can be derived by using two-parallel IIR all-pass function as [5]. We can write in the expression of low-pass filter and high-pass filter. Transfer function is given as follows

$$H_{LPF}(z) = \frac{1}{2}\left(A_0(z) + A_1(z)\right) \qquad (1)$$

$$H_{HPF}(z) = \frac{1}{2}\left(A_0(z) - A_1(z)\right) \qquad (2)$$

where $A_0(z)$ and $A_1(z)$ are stable APF

2. Linear Phase IIR Filter

An implementation of real-time linear phase IIR low-pass filters was proposed by Powell and Chau [6]. Hence, that system is possible to implementation of two-band digital crossover networks. However Group delay distortion, linear-time variant system and computational complexity are complicated.

Fig. 2 shows an implementation of two-way digital crossover networks using Linear Phase IIR Filter with complementary filter pairs. In the system, input sequence is time reversed on block by block using LIFO (last-in first-out) and then sectioned-convoled with $H_{LPF}(z)$ and $H_{HPF}(z)$ to realize a real-time recursive implementation of the non-causal transfer function after time reversed by LIFO again, the sequence passes through causal transfer function H(z) to obtain a linear phase IIR filters

2.1. Perfect Linear Phase IIR Filters

We can derive almost perfect reproduction of audio signal from two-way digital crossover network by using perfect linear phase IIR filters [7]. This system can improve both group delay distortion and linear-time variance system to Linear-time invariant (LTI) of Powell and Chau system by truncating infinite impulse response method [8].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

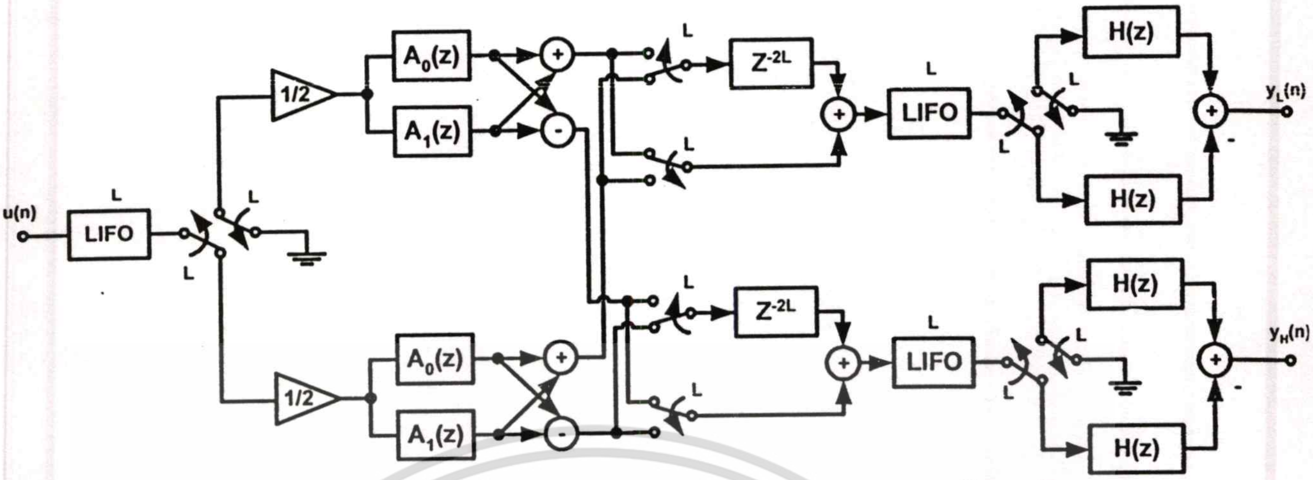


Fig. 2 Implementation of Linear Phase IIR Filter with complementary filter pairs

This method truncates the finite impulse response of linear phase IIR filter to finite length by using IIR filter direct form II to approximate a residue impulse response. Then subtract from the IIR filter with the transfer function H(z) and the output of the truncated filter is a finite impulse response.

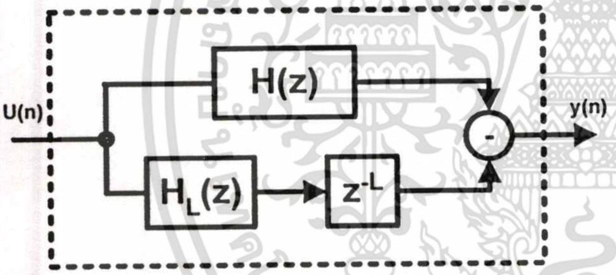


Fig. 3 Truncated IIR Filters

Impulse response of $H_L(z)$ with L sample delay can be defined as

$$h_L(n-L)=\begin{cases}h(n) & n\geq L\\ 0 & n\leq L\end{cases} \quad (3)$$

where h(n) is an impulse response of IIR filter H(z), L is sample length. We consider the direct form II IIR filter (K-order) in rational transfer function as follow

$$H(z)=\frac{a_0+a_1Z^{-1}+a_2Z^{-2}+...+a_KZ^{-K}}{1-b_1Z^{-1}-b_2Z^{-2}-...-b_KZ^{-K}} \quad (4)$$

where $a_1,a_2,...,a_K$ and $b_1,b_2,...,b_K$ are real number coefficients for efficient computational of $H_1(z)$, a minimum order in this case is second order. Therefore, we can rewritten Eq.(4) to the second order transfer function and applied to two-band of digital crossover networks is as follows

Truncated IIR filter for low-pass filter

$$H_{L-LPF}(z)=\frac{c_{L0}+c_{L1}Z^{-1}+c_{L2}Z^{-2}}{1-b_{L1}Z^{-1}-b_{L2}Z^{-2}}$$

(5)

Truncated IIR filter for high-pass filter

$$H_{L-HPF}(z)=\frac{c_{H0}+c_{H1}Z^{-1}+c_{H2}Z^{-2}}{1-b_{H1}Z^{-1}-b_{H2}Z^{-2}} \quad (6)$$

In Fig. 4 shows an implementation of two-way digital crossover networks by using perfect linear phase IIR filters and the output signal of low-pass filter is $y_L(n)$ and the output signal from high-pass filter is $y_h(n)$.

3. Simulations

An elliptic H(z) is designed according to the following specification

- Normalized frequency pass band $\omega_n = 0.5$
- Pass band ripple $\delta_p = 0.5dB$
- Stop band ripple $\delta_s = 20dB$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ อ้างอิงถึงเจ้าของเอกสารทุกครั้ง

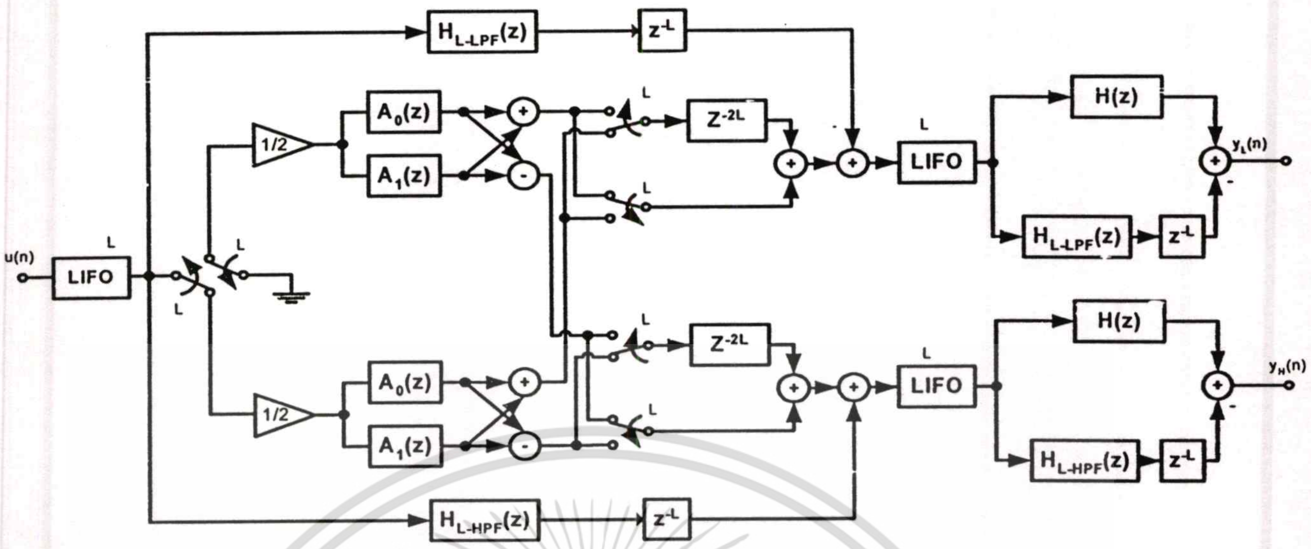


Fig. 4 Implementation of power complementary Perfect linear phase IIR filter by using over-lap add convolution.

Then, we can decompose the transfer function H(z) into two-stable APF as follows

$$A_0(z) = \frac{-0.7314 + Z^{-1}}{1 - 0.7314Z^{-1}} \quad (7)$$

$$A_1(z) = \frac{-0.6679 + 0.1151Z^{-1} + Z^{-2}}{1 + 0.1151Z^{-1} + 0.6679Z^{-2}} \quad (8)$$

Then, we can substitution $A_0(z)$ and $A_1(z)$ into Eq. (1), yields

$$H_{LPF}(z) = \frac{0.2759 + 0.5121Z^{-1} + 0.5121Z^{-2} + 0.2759Z^{-3}}{i - 0.0010Z^{-1} + 0.6546Z^{-2} - 0.0775Z^{-3}} \quad (9)$$

Similarly, we substitute $A_0(z)$ and $A_1(z)$ into Eq. (2), yields

$$H_{HPF}(z) = \frac{0.3920 - 0.4745Z^{-1} + 0.4745Z^{-2} - 0.3920Z^{-3}}{1 - 0.0010Z^{-1} + 0.6546Z^{-2} - 0.0775Z^{-3}} \quad (10)$$

Generally, the residual impulse response is governed domainly by the poles of the transfer function nearest to the unit circle in Z-plane. Therefore, H_{L-LPF} and H_{L-HPF} can be approximated by 2nd - order function $\tilde{H}_L(z)$ using pair of complex conjugate poles nearest to the unit circle. Denoting the conjugate pole as z_1 and $\bar{z}_1$, the numerator of $\tilde{H}_L(z)$ are determined by equating the first three impulse resonse samples of $\tilde{H}_L(z)$ with the residual response

Transfer function of $H_{L-LPF}(z)$ and $H_{L-HPF}(z)$ can be written in rational polynomial in Z-domain. It minimum order is second order(n=2) given as follows

$$H_{L-LPF}(z) = \frac{-0.00034986 - 0.00262908Z^{-1} - 0.00010842Z^{-2}}{1 + 0.11514478Z^{-1} + 0.66791907Z^{-2}} \quad (11)$$

$$H_{L-HPF}(z) = \frac{0.00034986 + 0.00262908Z^{-1} + 0.00030363Z^{-2}}{1 + 0.11514478Z^{-1} + 0.66791907Z^{-2}} \quad (12)$$

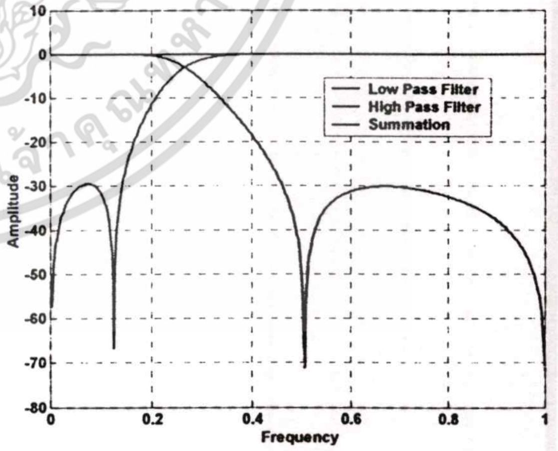


Fig. 5 Summation of complementary system magnitude response of prototype third-order elliptic IIR filter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

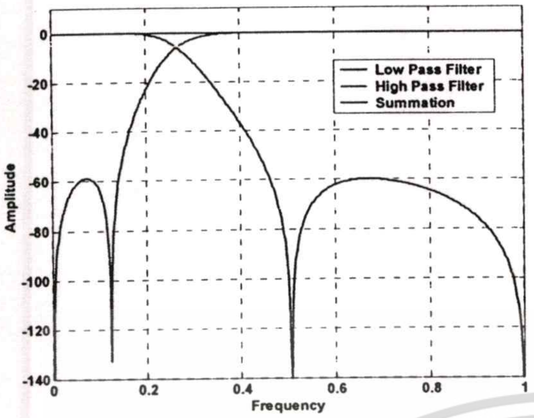


Fig. 6 Summation of complementary system magnitude response is flat response of proposed

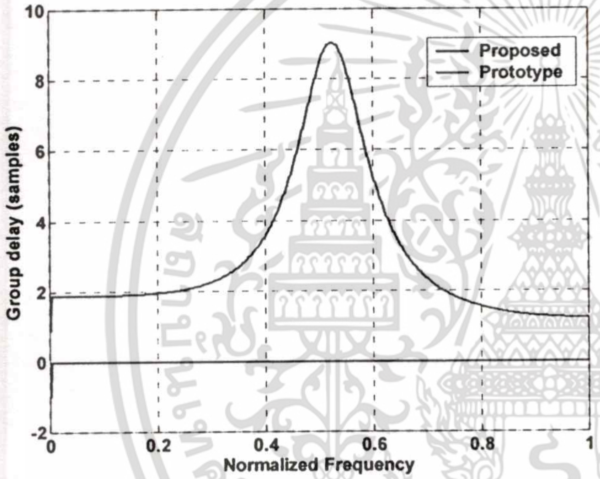


Fig. 7 Upper trace shown the group delay variation of prototype filter and lower trace shown the group delay of proposed system

4. Conclusion.

An implementation of two-band digital crossover networks by using perfect linear phase IIR filters is presented. The simulation results of proposed system show that group delay variation is flat as show in fig 7. magnitude response is twice attenuation in stop band in fig 6 than does the prototype filter in fig 5 their impulse response of both low-pass signal and high-pass signal are symmetry at the peak amplitude.

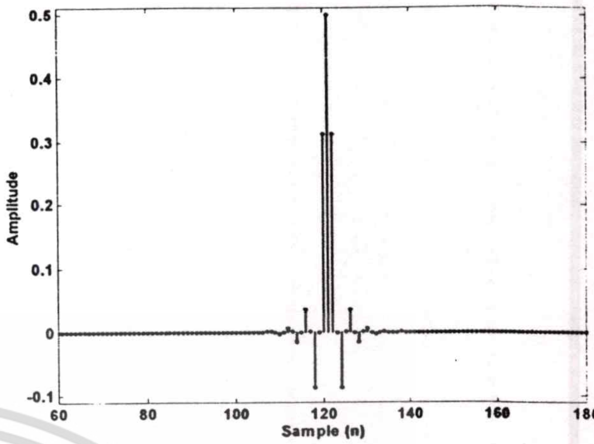


Fig. 8 Impulse response of propose low-pass filter $y_L(n)$

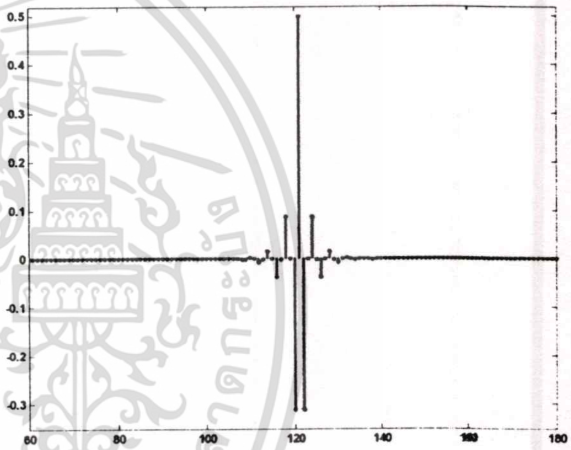


Fig. 9 Impulse response of propose high-pass filter $y_H(n)$

5. Reference.

[1] P.Reviriego, Jose Parera and R. Garcia, "Linear-Phase Crossover Design Using Digital IIR Filters", J. Audio Eng.Soc.,Vol. 46, No. 5, pp 406-410, May. 1998.

[2] Rhonda Wilson, Glyn Adams and Jonathan scott, "Applicacition of Digital Filter to Loundspeaker Crossover Networks", ", J. Audio Eng.Soc.,Vol. 37, No. 6, pp 455-463, June. 1998

[3] Tapio Saramaki, "On the Design of Digital Filters as a Sum of Two All-Pass Filters", IEEE Trans., Vol. CAS-32, No. 11, pp. 1191-1193, Nov. 1985.

[4] Phillip A. Regalia, Sanjit K. Mitra and P. P. Vaidyanathan, "The Digital All-Pass Filter : A V ersatile Signal Processing Building Block", Proc. IEEE, Vol. 76, No. 1, Jan. 1988.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่...ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ...อ้างอิงถึงเจ้าของเอกสารทุกค...

[5] P. P. Vaidyanathan A. Sanjit K. Mitra Yrjo Neuvo, "A New Approach to the Realization of Low-Sensivity IIR Digital Filters", IEEE Trans. ASSP., Vol. 34, No.2, pp.350-361, Apar. 1986.

[6] Scott R. Powell and Paul M. Chaul ,"A Technique for Realizing Linear Phase IIR Filter", IEEE Trans, Signal Processing, Vol.39, No.11, PP. 2425-2435, Nov.1991.

[7] D. Sookcharoenphol and et al. , "An anaysis of linear phase IIR filters and a condition for perfect linear phase", ICEP 2003, pp. 144-147, Jan., 2003.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าบางส่วนหรือทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

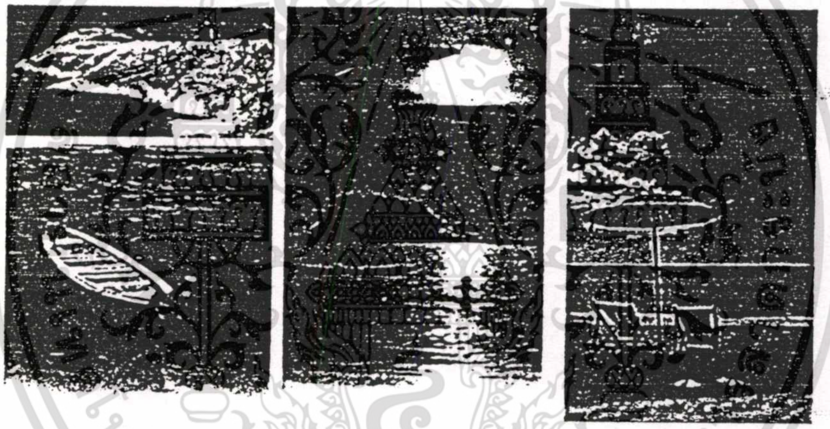
PROCEEDINGS

The 4th Information and Computer Engineering Postgraduate Workshop 2004

ICEP 2004

22nd - 23rd January 2004

Phuket, Thailand



Prof Jun Marai, KEIO University, Japan

Organised and Sponsored by
Department of Computer Engineering,
Faculty of Engineering,
Prince of Songkla University, Thailand
and Prof Jun Marai, KEIO University, Japan

In cooperation with
IEEE ComSoc. Thailand Chapter,
ECTI (Electrical Engineering/Electronics,
Computer, Telecommunications and Information
Technology Association of Thailand)

ISBN 974-644-518-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sri-Tang Room	Thursday 22nd January 2004
	Session 001:Wireless Communications
13.45-14.05	Neighbor Option Path mechanism for Ad Hoc network routing protocols .... 1 *Anan Phonphoem and Nuttapong Tongrungsri*
14.05-14.25	An Efficient Gain Clamped L-band EDFA by Incorporating a Broadband FBG in Double Pass Configuration .... 7 *Sulaiman Wadi Harun and Harith Ahmad*
14.25-14.45	Beam Scanning System of Quasi-Optical Antenna Mixer Array using Active Array Antenna .... 10 *S. Janin, P. Ngamjanyaporn, C. Phongcharoenpanich and M. Krairiksh*
14.45-15.05	An Adaptive Phased Array Antenna using CMA .... 15 *A. Boonpoongc, P. Ngamjanyaporn, P. Sirisuk, V. Pakkamarp, C. Phongcharoenpanich, M. Krairiksh*
15.05-15.25	On the Design of Aperture-Coupled Microstrip Antenna .... 19 *N. Anantrasirichai and K. Narkcharoen*

Fueng-Fa Room	Thursday 22nd January 2004
	Session 002:Speech and Audio Processing
13.45-14.05	A Study of Thai Tone Classification .... 24 *Li Tan, Montri Karnjanadecha, Thanate Khaorapapong and Pichaya Tandayya*
✓14.05-14.25	Finite Word Length Effect of Digital Crossover Network Using Perfect Linear-Phase IIR Filters .... 28 *C. Kanna, J. Orsawat, D. Sookcharoenphol and K. Janjitrapongvej.*
14.25-14.45	SYNTHESIS OF THAI MONOPHTHONGS BY ARTICULATORY MODELING .... 34 *K. Inthavisas, T. Khaorapapong and M. Karnjanadecha*
14.45-15.05	Harmonic Reduction with Binary Method for Three-phase System .... 39 *Pang Shen Yee, Hashim bin Saim, Pang Che Fong, Soon Chin Fhong*
✓15.05-15.25	Improvement of Group Delay and Reduction of Computational Complexity in Linear Phase IIR Filters .... 45 *S. Varasumanta, D. Sookcharoenphol, U. Sriteraviroj, K. Janjitrapongvej*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Finite Word Length Effect of Digital Crossover Network Using Perfect Linear-Phase IIR Filters

C. Kanna, J. Orsawat, D. Sookcharoenphol and K. Janjitrapongvej.
Information Dept., Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang Bangkok THAILAND

Email: s5061106@kmitl.ac.th

Abstract

In this paper, the implementation technique of digital crossover network using perfect linear phase IIR filters is presented. This system has various advantages which can not be obtained from analog crossover network such as linear phase response, flat group delay and sharp cut-off at low-order over audio frequency band. The simulation results show that the group delay response is maximally flat and twice more attenuation in stop-band than the prototype elliptic IIR filter at all desired frequency. Simulation results show that the finite word length effect to stopband attenuation of proposed two-band digital crossover is acceptable and all-pass function is also preserved.

1. Introduction

Classical design of two-way digital crossover networks consist of low-pass filter and high-pass filter to split the audio signal into low-pass signal and high-pass signal, and each feeding to a separated loudspeaker. However, the amplitude, phase and group delay distortions can not be eliminated and it is difficult to achieve perfect reproduction [1] and [2]. We can derived the odd-order digital low-pass filter using the Bilinear transform and it can be implemented as a sum of two all-pass filters [3]. Two stable all-pass filters (APF) which transfers function pairs that is satisfied to all-pass complementary property was proposed by Regalia and et a! [4]. It is well known that this two stable APF have less sensitivity and circuit complexity than the classical design. An implementation of the all-pass complementary filter pairs as the sum and difference of all pass functions is shown in Fig.1

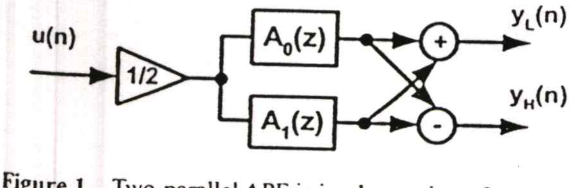


Figure 1. Two-parallel APF is implementing of two-way crossover network.

The implementation of two-way digital crossover can be derived by using two-parallel IIR all-pass function as [5]. We can write in the expression of low-pass filter and high-pass filter. Transfer function is given as follows

$$H_{LPF}(z) = \frac{1}{2}\left(A_0(z) + A_1(z)\right) \quad (1)$$

$$H_{HPF}(z) = \frac{1}{2}\left(A_0(z) - A_1(z)\right) \quad (2)$$

where $A_0(z)$ and $A_1(z)$ are stable APF

2. Linear Phase IIR Filter

An implementation of real-time linear phase IIR low-pass filters was proposed by Powell and Chau [6]. Hence, that system is possible to implementation of two-band digital crossover networks. However Group delay distortion, linear-time variant system and computational complexity are complicated.

Fig. 2 shows an implementation of two-way digital crossover networks using Linear Phase IIR Filter with complementary filter pairs. In the system, input sequence is time reversed on block by block using LIFO (last-in first-out) and then sectioned-convoled with $H_{LPF}(z)$ and $H_{HPF}(z)$ to realize a real-time recursive implementation of the non-causal transfer function after time reversed by LIFO again, the sequence passes through causal transfer function H(z) to obtain a linear phase IIR filters

2.1. Perfect Linear Phase IIR Filters

We can derive almost perfect reproduction of audio signal from two-way digital crossover network by using perfect linear phase IIR filters [7]. This system can improve both group delay distortion and linear-time variance system to Linear-time invariant (LTI) of Powell and Chau system by truncating infinite impulse response method [8].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

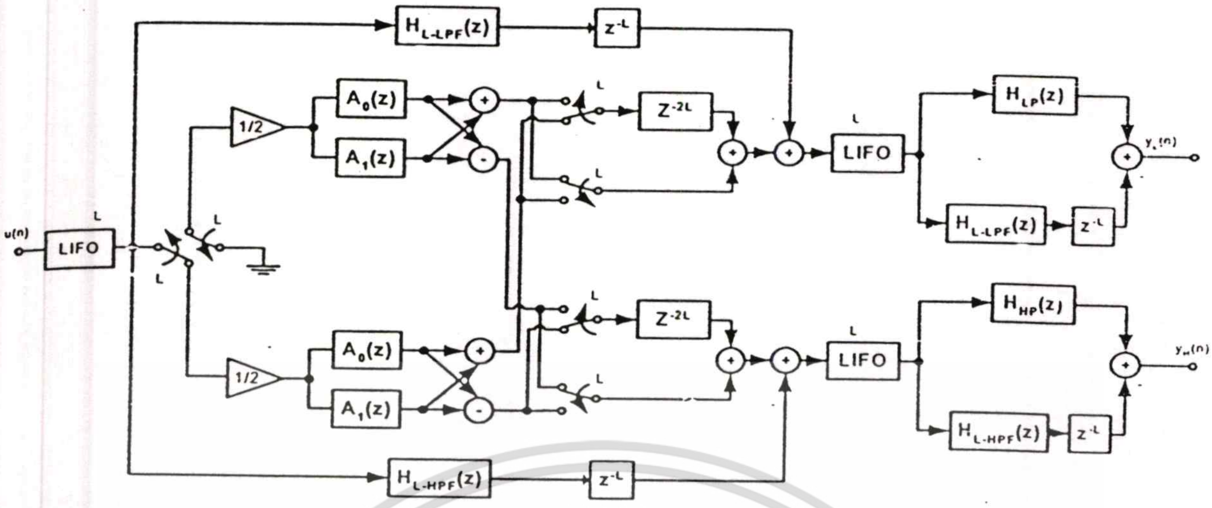


Figure 2. An implementation of all-pass complementary perfect linear-phase IIR filters by using overlap add convolution.

This method truncates the finite impulse response of linear phase IIR filter to finite length by using IIR filter direct form II to approximate a residue impulse response. Then subtract from the IIR filter with the transfer function H(z) and the output of the truncated filter is a finite impulse response.

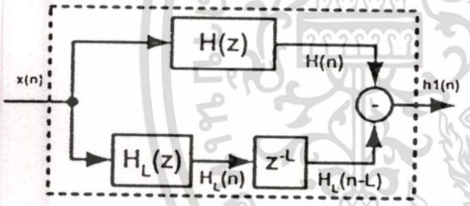


Figure 3. Truncated IIR Filters

Impulse response of $H_L(z)$ with L sample delay can be defined as

$$h_L(n-L) = \begin{cases} h(n) & n \geq L \\ 0 & n \leq L \end{cases} \quad (3)$$

where h(n) is an impulse response of IIR filter H(z), L is sample length. We consider the direct form II IIR filter (K-order) in rational transfer function as follow

$$H(z) = \frac{a_0 + a_1 Z^{-1} + a_2 Z^{-2} + \ldots + a_K Z^{-K}}{1 - b_1 Z^{-1} - b_2 Z^{-2} - \ldots - b_K Z^{-K}} \quad (4)$$

where $a_1, a_2, \ldots, a_K$ and $b_1, b_2, \ldots, b_K$ are real number coefficients for efficient computational of $H_l(z)$, a minimum order in this case is second order. Therefore, we can rewritten Eq.(4) to the second order transfer function and applied to two-band of digital crossover networks is as follows

Truncated IIR filter for low-pass filter

$$H_{L-LPF}(z) = \frac{c_{L0} + c_{L1} Z^{-1} + c_{L2} Z^{-2}}{1 - b_{L1} Z^{-1} - b_{L2} Z^{-2}} \quad (5)$$

Truncated IIR filter for high-pass filter

$$H_{L-HPF}(z) = \frac{c_{H0} + c_{H1} Z^{-1} + c_{H2} Z^{-2}}{1 - b_{H1} Z^{-1} - b_{H2} Z^{-2}} \quad (6)$$

In Fig. 2 shows an implementation of two-way digital crossover networks by using perfect linear phase IIR filters and the output signal of low-pass filter is $y_L(n)$ and the output signal from high-pass filter is $y_h(n)$..

3. Quantization of filter structures

All practical implementations of digital signal processing algorithm have to cope with finite register lengths. This lead to impairments of the filter performance that become visible in different ways.

The filter coefficients, which can be determined by means of presented design algorithm with arbitrary precision, can only be approximated by numbers with a given limited word length. This implies that the characteristic of the resulting filter deviate more or less form the original specification.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Future effect such as the generation of noise and unstable behavior are caused by truncation of bit which become necessary after multiplications if we multiply for instance a signal sample with a resolution of a bit by a coefficient with a word length of bits we obtain a result with *(a+b)* bits in recursive filters without that truncations. the number of digits would increase by b after each clock cycle.

3.1. IIR direct form II structure

For the implementations of digital IIR filter structure and quantized filter structure are shown as follows.

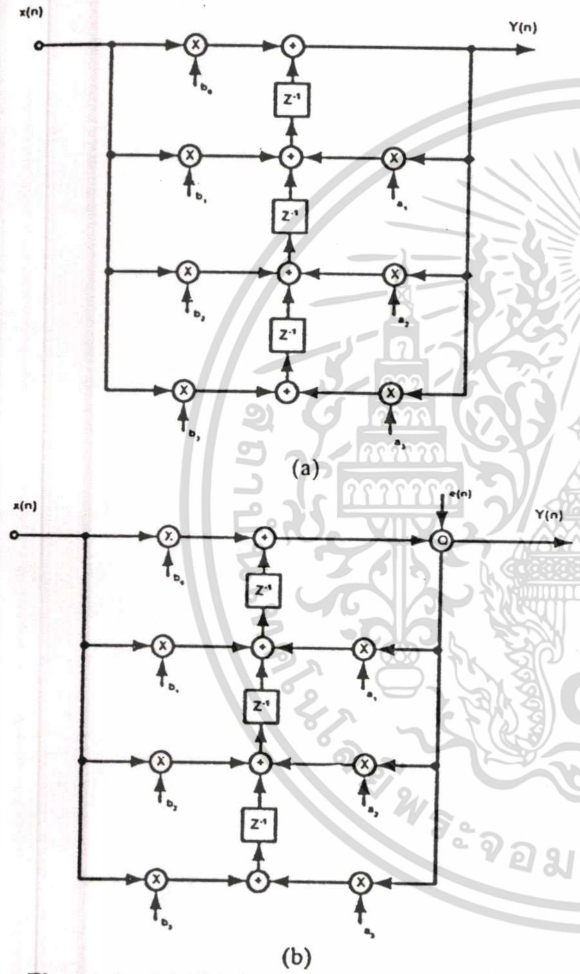


Figure 4. (a) Third-order IIR direct form II and (b) quantized of third-order IIR direct form II structure

3.2. All-pass filter structure

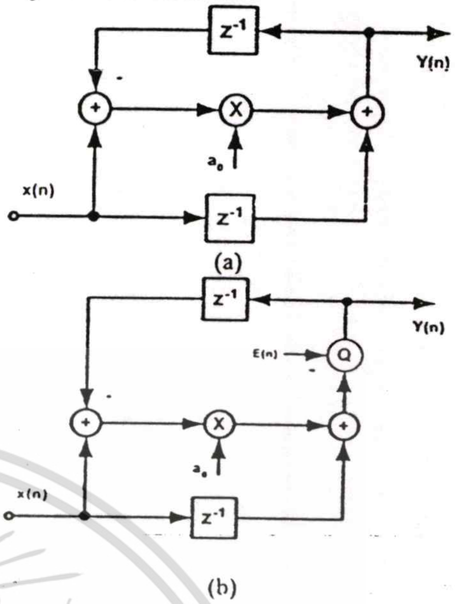


Figure 5. All-pass filters(APF) structure (a) First-order APF (b) Quantized of first-order APF

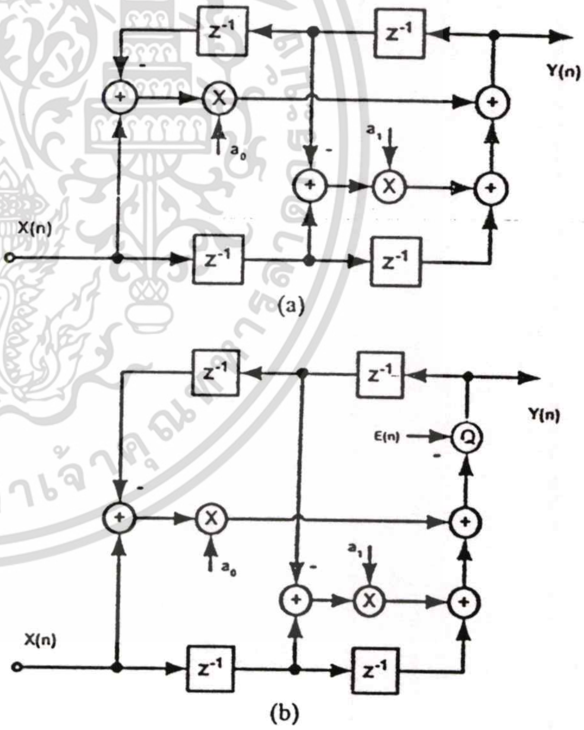


Figure 6. All-pass filters(APF) structure (a) Second-order All-pass filters(APF) structure and (b) Quantized of second-order APF structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Simulations Result

An elliptic low-pass IIR filter H(z) (prototype filter) is designed according to the following specification

3rd order Elliptic IIR Filter

Cutoff Frequency $(\omega_p) = 0.0136$

Ripple in passband $(\delta_p) = 0.005$ dB

Attenuation in stop band $(\delta_s) = 30$dB

All simulations results, we comparison between the third-order IIR filter direct form II structure as shown in Fig. 4 and all-pass IIR filter structure(based on the structure as shown in Fig. 1) as shown in Fig. 5 and Fig. 6.

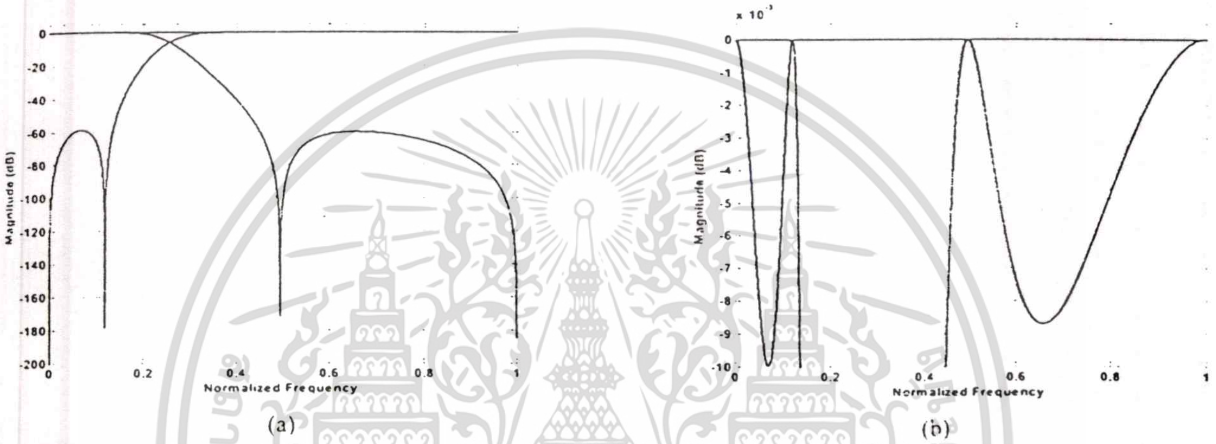


Figure 7. (a) Magnitude response of prototype elliptic IIR filter (b) show passband ripple magnitude response of LPF, HPF and summation.

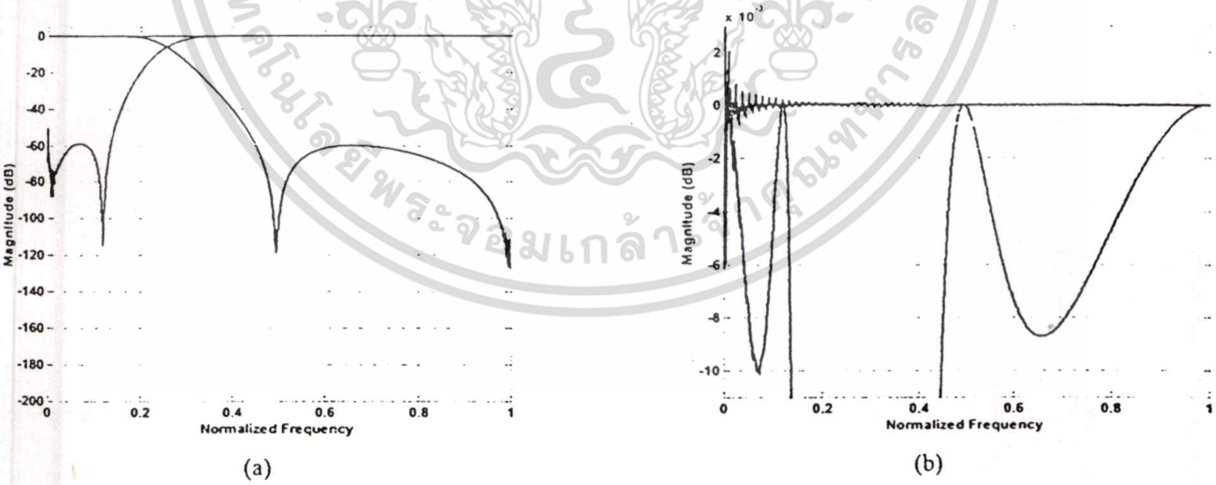
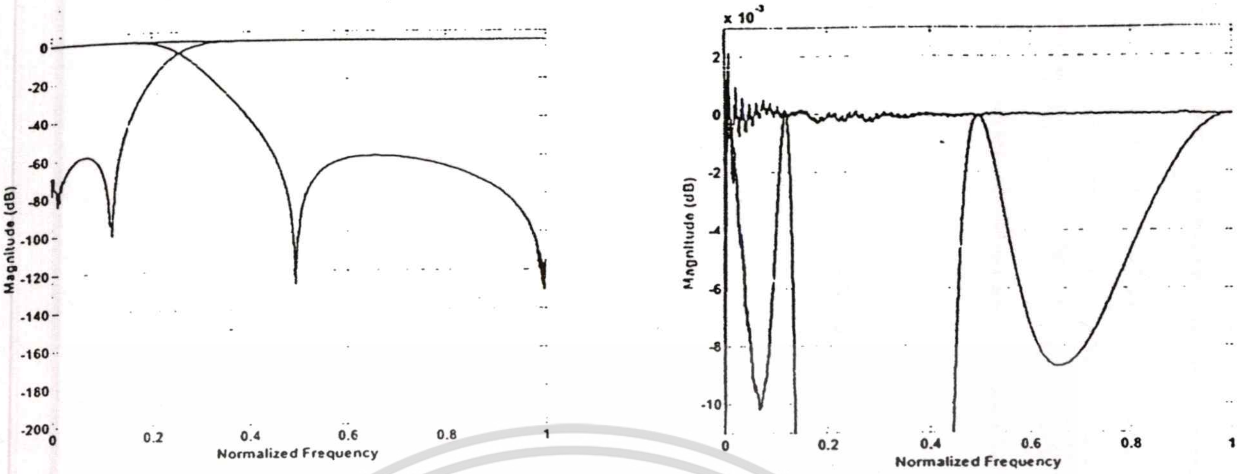


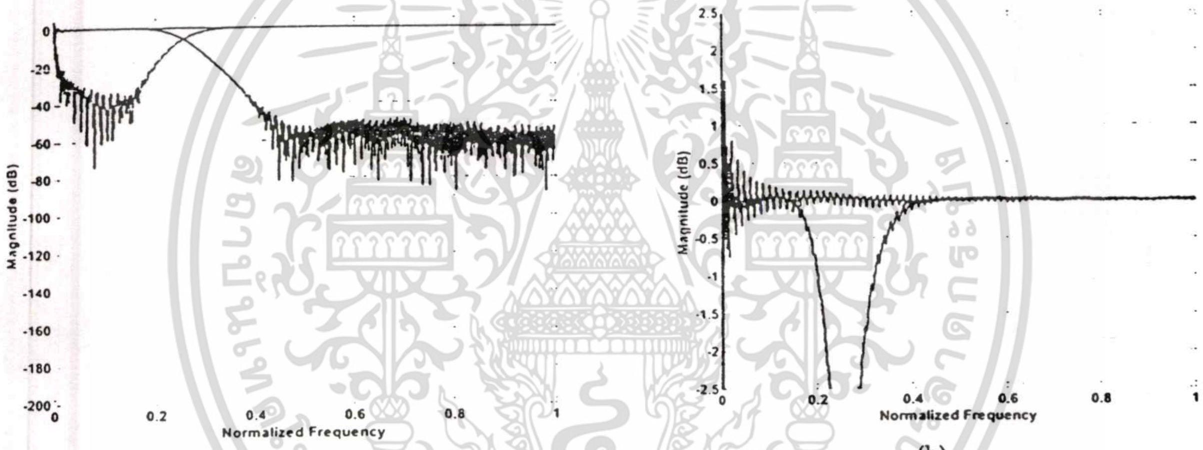
Figure 8. (a) Magnitude response elliptic IIR filter (b) show passband ripple magnitude response of LPF, HPF and summation in 20 bits quantized parallel all-pass filter structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



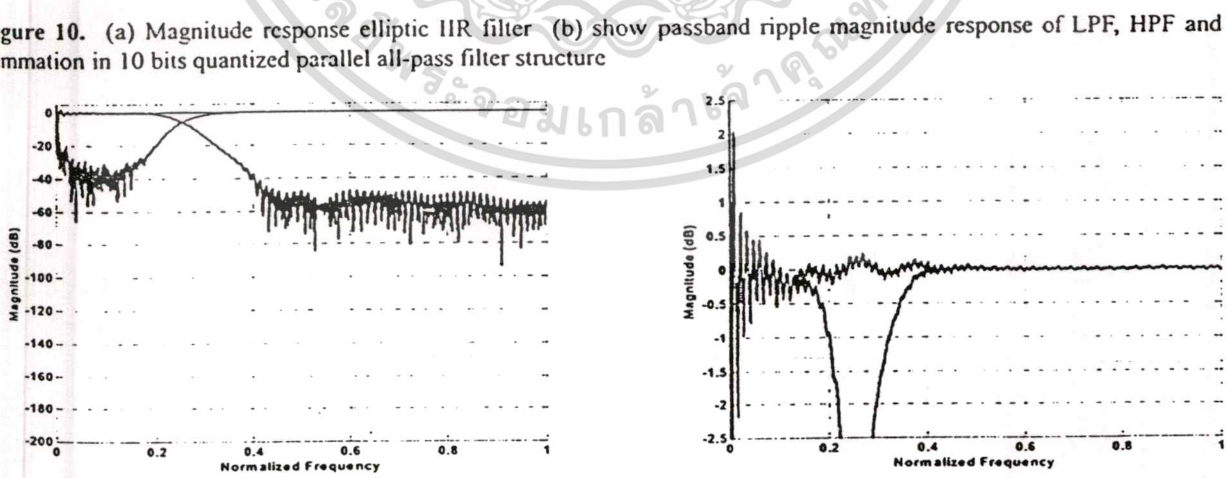
(a) (b)

Figure 9. (a) Magnitude response elliptic IIR filter (b) show passband ripple magnitude response of LPF, HPF and summation in 20 bits quantized direct form II filter structure



(a) (b)

Figure 10. (a) Magnitude response elliptic IIR filter (b) show passband ripple magnitude response of LPF, HPF and summation in 10 bits quantized parallel all-pass filter structure



(a) (b)

Figure 11. (a) Magnitude response elliptic IIR filter (b) show passband ripple magnitude response of LPF, HPF and summation in 10 bits quantized direct form II filter structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

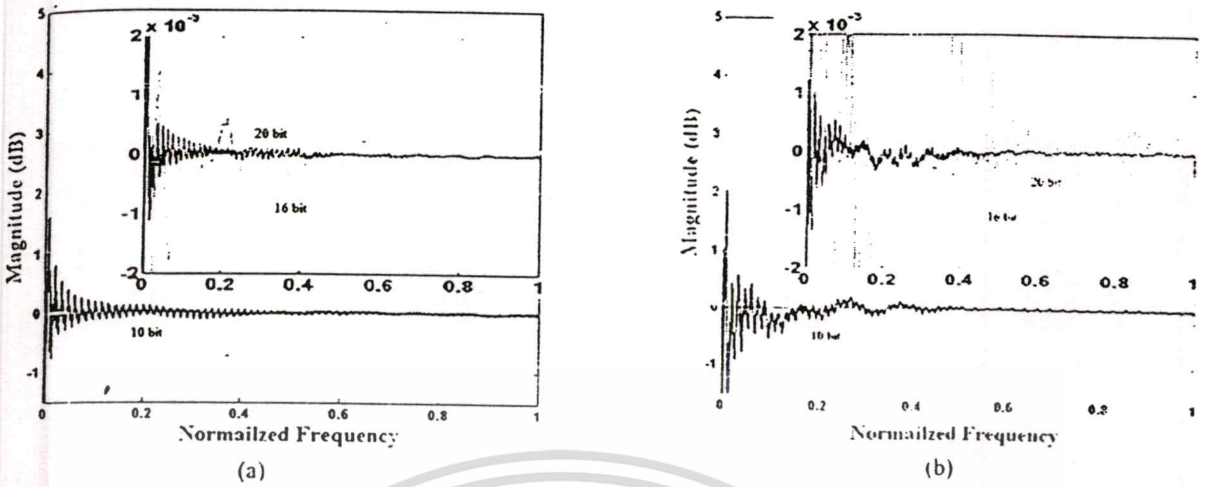


Figure 12. Show in details of magnitude response of elliptic IIR filter with quantized (a) PAPF structure (b) IIR direct form II

From Fig. 12 a) shown the finite word length effect comparison of summed magnitude response of PAPF structure when n = 20, 16 and 10 bit respectively. We seen that, when n= 20 and 16 bit the summed magnitude response are flat and n = 10 bit the summed magnitude response has small ripple on low frequency range. And Fig. b) shown the finite word length effect comparison of summed magnitude response of IIR direct form II structure when n = 20, 16 and 10 bit respectively. We seen that, when n = 20, 16 and 10 bit the summed magnitude response have more ripple cause an all-pass property of this filter structure is not obtained.

5. Conclusion

This paper present a finite word length effect of two-band digital crossover network using perfect linear-phase IIR filters. Simulation results shown that the magnitude response compared both IIR direct form II and parallel all-pass IIR filter, when a word length reduced(n= 20, 16 and 10 bits respectively) the proposed filter shown the magnitude response and their summed are better than IIR direct form II structure. It was shown that the stopband attenuation both low-pass filter and high-pass filter are acceptable. Therefore, the sensitivity of proposed filter is improved and the condition of all-pass function is preserved.

6. References

[1] P.Reviriego, Jose Parera and R. Garcia, "Linear-Phase Crossover Design Using Digital IIR Filters", J. Audio Eng.Soc., Vol. 46. No. 5, pp 406-410. May. 1998.

[2] Rhonda Wilson, Glyn Adams and Jonathan scott, "Applicacition of Digital Filter to Loundspeaker Crossover Networks". ". J. Audio Eng.Soc., Vol. 37. No. 6, pp 455-463, June. 1998.

[3] Tapio Saramaki, "On the Design of Digital Filters as a Sum of Two All-Pass Filters", IEEE Trans., Vol. CAS-32. No. 11, pp. 1191-1193, Nov. 1985.

[4] Phillip A. Regalia. Sanjit K. Mitra and P. P. Vaidyanathan. "The Digital All-Pass Filter : A V ersatile Signal Processing Building Block", Proc. IEEE, Vol. 76. No. 1. Jan. 1988.

[5] P. P. Vaidyanathan A. Sanjit K. Mitra Yrjo Neuvo, "A New Approach to the Realization of Low-Sensivity IIR Digital Filters", IEEE Trans. ASSP., Vol. 34, No.2, pp.350-361, Apar. 1986.

[6] Scott R. Powell and Paul M. Chaul ,"A Technique for Realizing Linear Phase IIR Filter", IEEE Trans, Signal Processing, Vol.39, No.11, PP. 2425-2435, Nov.1991.

[7] D. Sookcharoenphol and et al. , "An anaysis of linear phase IIR filters and a condition for perfect linear phase", ICEP 2003, pp. 144-147, Jan., 2003.

[8] D. Schlichtarle, "Digital Filters : Basics and Design," Springer, pp. 227-311, Berlin, 2000.

[9] J. Orsawat, D. Sookcharoenphol and K. Janjitrapongvej, "An Implementation of Digital Crossover Network by using Perfect Linear-Phase IIR Filters", ISCIT 2003, pp. 828-832, Sept., 2003.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 27

27th Electrical Engineering Conference

Volume II

สาขาบทความ

- ระบบควบคุม[illegible]
- คอมพิวเตอร์และเทคโนโลยีสารสนเทศ (CP)
- การประมวลผลสัญญาณดิจิตอล (DS)
- อิเล็กทรอนิกส์ (EL)
- ไฟฟ้าสื่อสาร (CM)

XXVII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์[illegible] ต้องอ้างอิง[illegible]

ภาควิชาวิศวกรรมไฟฟ้า [illegible]
มหาวิทยาลัยขอนแก่น

วงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คแบบใหม่ที่ให้กรุ๊ปดีเลย์ราบเรียบ

ชาญณรงค์ การนา, เจตน์ ออสวัสดิ์, ดลชัย ชุขเจริญผล และ กนก เจนจิระพงศ์เวช
ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
3 หมู่ 2 ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพมหานคร 10520
โทร. 0-2737-3000 E-mail: chan@enbit.co.th, s5061106@kmitl.ac.th, ksdolcha@kmitl.ac.th , kjknok@kmitl.ac.th,

บทคัดย่อ

ในบทความนี้เป็นการนำเสนอการออกแบบสร้างวงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คสำหรับย่านความถี่ออดิโอสองทางแบบใหม่โดยปรับปรุงจากโครงสร้างของวงจรกรองความถี่ทุกย่านความถี่ผ่านที่ต่อขนานกัน โดยโครงสร้างดังกล่าวจะมีข้อดีคือ มีผลตอบสนองทางเฟสต่อความถี่ใกล้เคียงเชิงเส้น, ผลตอบสนองกรุ๊ปดีเลย์เกือบราบเรียบ และผลรวมทางขนาดต่อความถี่มีค่าคงที่ตลอดย่านความถี่ ซึ่งเป็นสิ่งที่ต้องคำนึงถึงในการออกแบบวงจรครอสโอเวอร์เน็ตเวิร์คสำหรับระบบไฮไฟ ผลการจำลองของโครงสร้างที่นำเสนอนี้จะให้คุณลักษณะของกรุ๊ปดีเลย์ที่ราบเรียบกว่าโครงสร้างของวงจรกรองความถี่ทุกย่านความถี่ผ่านต่อขนานกันรวมทั้งแบบลิงค์วิทซ์ไอไออาร์ฟิลเตอร์ และผลรวมของผลตอบสนองขนาดต่อความถี่จะราบเรียบตลอดย่านความถี่ออดิโอ

คำสำคัญ: วงจรกรองความถี่ผ่านตลอด, วงจรกรองความถี่ IIR ที่ให้เฟสเป็นเชิงเส้น และวงจรแบ่งความถี่ 2 ย่าน

Abstract

This paper present an implementation of two-ways digital crossover network for audio frequency using parallel all-pass filter structure. The filter have advantages than those of analog filter and also FIR and IIR filter because it can improved various important characteristics such as linear phase, flat group delay, high transition band and flat magnitude response to over audio frequency range. Simulation results shown that our filter can be improved both flat summed magnitude response and flat group delay than oldest method.

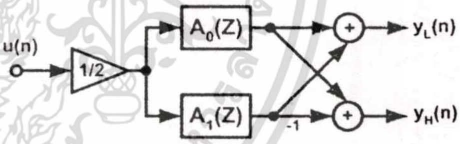
Keywords: All-Pass Filter, Linear-Phase IIR Filters and Two-way Digital Crossover Network

1. บทนำ

ในการออกแบบดิจิตอลครอสโอเวอร์เน็ตเวิร์คนั้นความซับซ้อนในการคำนวณของวงจรเป็นเหตุผลหนึ่งที่ต้องคำนึงในการออกแบบ รวมไปจนถึงการสร้างจริงซึ่งในการออกแบบครอสโอเวอร์เน็ตเวิร์คให้มีคุณภาพดี [1] จะต้องคำนึงถึงคุณสมบัติที่สำคัญดังนี้

1. ผลการตอบสนองทางขนาดต่อความถี่มีลักษณะราบเรียบตลอดย่านความถี่
2. มีผลตอบสนองทางเฟสที่สมมาตร
3. มีสโลปของจุดตัดความถี่ที่ชันมาก
4. มีการตอบสนองทางเฟสเป็นเชิงเส้นหรือมีผลตอบสนอง กรุ๊ปดีเลย์ที่ราบเรียบ ณ. บริเวณจุดตัดความถี่

ซึ่งจากที่ได้กล่าวมาแล้วนั้นจะเป็นว่าทั้งวงจรกรองความถี่แบบ เอฟ ไอ อาร์ และ ไอ ไอ อาร์ ต่างก็มีข้อเสียที่แตกต่างกันซึ่ง เช่นจำนวนของสัมประสิทธิ์ หรือความไววงจรในการปัดเศษสัมประสิทธิ์เป็นต้น ซึ่ง S.Mitra ได้กล่าวถึงโครงสร้างของวงจรกรองความถี่ที่สร้างจากวงจรกรองความถี่ผ่านตลอดต่อขนาน 2 วงจร (Two Parallel All-Pass Filters) [2-4] ซึ่งโครงสร้างดังกล่าวนี้มีข้อดีคือมีผลต่อการปัดเศษของค่าสัมประสิทธิ์ที่ต่ำ (Low Sensitivity) และมีโครงสร้างที่ไม่ซับซ้อนมาก (Low Complexity) ดังแสดงในรูปที่ 1



รูปที่ 1. โครงสร้างของวงจรครอสโอเวอร์เนตเวิร์คแบบ 2 ย่านความถี่ที่สร้างจากวงจรกรองความถี่ผ่านตลอด

สัญญาณที่ด้านเอาต์พุตของทั้งวงจรกรองความถี่ต่ำ($y_L(n)$) และวงจรกรองความถี่สูง ($y_H(n)$) สามารถสร้างได้จากสัญญาณเอ้าพุททั้งผลรวมและผลต่างของวงจรกรอง $A_0(z)$ และ $A_1(z)$ ดังสมการที่ 1 และสมการที่ 2 ต่อไปนี้

$$H_{LP}(z) = \frac{1}{2}\Big(A0(z) + A1(z)\Big) \qquad (1)$$

$$H_{HP}(z) = \frac{1}{2}\Big(A0(z) - A1(z)\Big) \qquad (2)$$

เมื่อ $A_0(z)$ และ $A_1(z)$ เป็นวงจรกรองความถี่ผ่านตลอดที่มีเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถสร้างวงจรดังกล่าวให้มีความซับซ้อนลดลงได้โดยการเลือกให้ตัวดำเนินการด้านบนเป็นวงจรหน่วงเวลาและด้านล่างเป็นวงจรกรองความถี่ผ่านตลอดสามารถเขียนสมการฟังชันถ่ายโอนได้ใหม่ดังนี้คือ

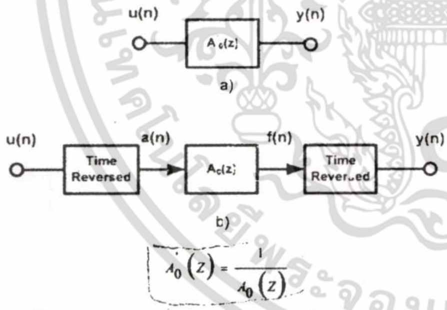
$$Y_L(Z) = \frac{1}{2}\left(Z^{-m} + A_0'(Z)\cdot A_1(Z)\right) \quad (3)$$

$$Y_H(Z) = \frac{1}{2}\left(Z^{-m} - A_0'(Z)\cdot A_1(Z)\right) \quad (4)$$

เมื่อกำหนดให้ $H(z) = \frac{Y(z)}{U(z)} = A_o(z^{-1}) = A_o^{-1}(z)$ แทนด้วย H(Z) และเราจะเรียก $A_0(Z)$ ว่า "Inverse All-Pass Filter"

2. การสร้างวงจร Inverse All-Pass Filter

การสร้างตัวกรองความถี่แบบนอนคอซอล(noncausal filter) ที่สามารถประมวลผลได้ในเวลาจริง(real-time processing) ได้ถูกนำเสนอโดย เพาเวลล์และฉู [5] เราสามารถที่จะนำวิธีดังกล่าวมาสร้างวงจร Inverse All-Pass Filter ได้ดังแสดงดังรูปที่ 3 ซึ่งวิธีนี้จะทำให้มีประสิทธิภาพสำหรับการสร้างที่สามารถประมวลผลในเวลาจริง และจากรูปดังกล่าวสัญญาณอินพุตที่เป็นลำดับอนันต์ u(n) จะถูกป้อนผ่านวงจรกลับสัญญาณ (time reverse) โดยมีลักษณะเป็นบล็อกที่มีความยาวในแต่ละบล็อกเท่ากันคือ บล็อกละ L แซมเปิล หลังจากนั้นจะถูกกรองสัญญาณด้วยวงจรกรองแบบคอซอล $A_0(Z)$ และลำดับที่ด้านเอ้าพุทจะถูกป้อนเข้าสู่วงจรกลับสัญญาณอีกครั้งโดยสามารถเขียนในรูปสมการถ่ายโอนได้ดังนี้



รูปที่ 3 การสร้างวงจร Inverse All-Pass Filter ด้วยวงจรกรองความถี่แบบนอนคอซอลที่ทำงานได้ในเวลาจริง

$$A(Z) = U(Z^{-1}) \quad (5)$$

$$F(Z) = H(Z)\cdot U(Z^{-1}) \quad (6)$$

$$Y(Z) = F(Z^{-1}) = H(Z^{-1})\cdot U(Z) \quad (7)$$

$$H(z) = \frac{Y(z)}{U(z)} = A_o(z^{-1}) = A_o^{-1}(z) \quad (8)$$

ดังนั้นเมื่อเรานำ $A_o'(z)$ แทนลงไปในสมการ (3) และสมการ (4) เราสามารถสร้างวงจรดิจิตอลครอสโอเวอร์ 2 ทางแบบใหม่ที่สามารถลดความซับซ้อนของวงจรลงได้ ดังแสดงในรูป 4

เฉพาะส่วนล่างของวงจรสัญญาณลำดับอนันต์ที่เข้ามาที่ด้านอินพุท u(n) จะถูกแบ่งออกเป็น บล็อคที่มีการจำกัดความยาวขนาด L แซมเปิลและแต่ละบล็อคจะถูกพลิกกลับในเชิงเวลา (time-reversed) ด้วยวงจร LIFO (Last-in first-out) หลังจากนั้นถูกป้อนผ่านวงจรกรองความถี่ไอ ไอ อาร์ $A_1(Z)$ ทั้งด้านบน (top filter) และด้านล่าง (bottom filter) และผลลัพท์ของการคูณประสาน (convolution) และนำสัญญาณเอ้าพุทของแต่ละบล็อคที่มีการวางเหลือมกันนำมารวมกันได้เป็นสัญญาณที่ด้านเอ้าพุท(overlap add convolution) โดยลำดับที่ได้จากวงจรกรองทั้งสอง $A(n)$ และ $B(n)$ จะถูกจำกัดความยาวโดยการรีเซ็ทให้มีความยาวเท่ากับ 2L จากนั้นสัญญาณที่ได้จะถูกป้อนผ่านวงจรสวิทช์ซึ่งมีเวลาในการสวิทช์เท่ากับ L วงจรหน่วงเวลาขนาดความยาวของการหน่วงเท่ากับ 2L และนำสัญญาณทั้งสองมารวมกันได้เป็นลำดับ $f(n)$ หลังจากนั้นส่งผ่านวงจร LIFO อีกครั้งได้สัญญาณเป็น $b_o(n)$ ซึ่งสัญญาณที่ได้นี้มาจากวงจรกรองแบบไม่เป็นคอลซอล โดยผลการวิเคราะห์แสดงไว้ในเอกสาร [6] โดยสัญญาณที่เอ้าพุทของวงจรดิจิตอลครอสโอเวอร์เป็นไปตามสมการที (3) และสมการ (4) ตามลำดับ

3. การออกแบบวงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คแบบสองทางที่มีกรุ๊ปดีเลย์ใกล้เคียงเชิงเส้น

ในการออกแบบจะกำหนดคุณลักษณะของวงจรกรองความถี่ต่ำเป็นต้นแบบขึ้นมาก่อนโดยมีพารามิเตอร์ต่างๆ โดยในตัวอย่างการจำลองด้วยโปรแกรมได้กำหนดค่าต่างๆ ไว้ดังนี้คือ

Frequency passband	3000	Hz
Frequency stopband	7000	Hz
Passband ripple	0.04	dB
Stopband ripple	20	dB

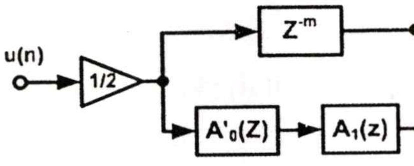
จากข้อกำหนดของสามารถหาฟังก์ชันถ่ายโอนของวงจรกรองความถี่ต่ำต้นแบบได้ดังสมการที่ 8

$$H_{LP}(Z) = \frac{a_0 + a_1\cdot Z^{-1} + a_2\cdot Z^{-2} + a_3\cdot Z^{-3}}{b_0 + b_1\cdot Z^{-1} + b_2\cdot Z^{-2} + b_3\cdot Z^{-3}} \quad (8)$$

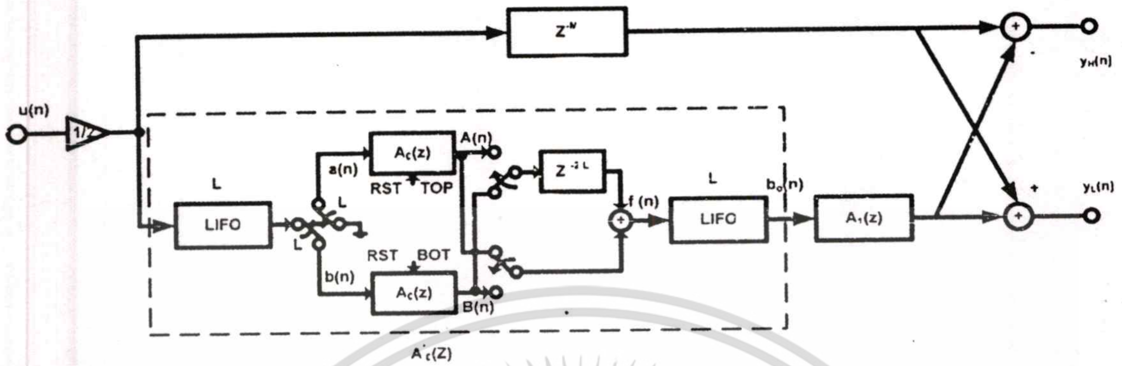
ตารางที่ 2 ค่าสัมประสิทธิ์ของวงจรกรองความถี่ต่ำต้นแบบ

	a	b
0	0.08998297633607	1.00000000000000
1	-0.01979741054655	-1.93618449317407
2	-0.01979741054655	1.44821651379210
3	0.08998297633607	-0.37166088903899

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น



a)



b)

รูปที่ 4 วงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คแบบสองทางที่มีกรุ๊ปดีเลย์ใกล้เคียงเชิงเส้นที่นำเสนอ

เมื่อนำเอาเทอมส่วนมาทำการแยกหาค่ารากเพื่อนำไปสร้างวงจรกรองทุกย่านความถี่ผ่านด้วยวิธีการ Pole interlacing prototype [7] ซึ่งจะได้สมการคุณลักษณะของ $A_0(z)$ และ $A_1(z)$ ตามสมการที่ 9 และ 10 ตามลำดับ

$$A_0(Z) = \frac{-0.52626196284744 + Z^{-1}}{1-0.52626196284744 \cdot Z^{-1}} \quad (9)$$

$$A_1(Z) = \frac{0.70622791551-1.40992253032 \cdot Z^{-1} + a_2 \cdot Z^{-2}}{1-1.40992253032 \cdot Z^{-1} + 0.70622791551 \cdot Z^{-2}} \quad (10)$$

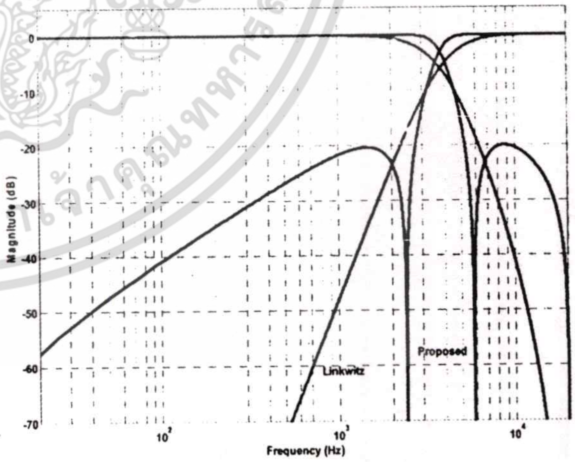
จากสมการที่ 9 และ 10 สามารถเขียนสมการ H(Z)

$$H(Z) = \left(\frac{a_1 + a_0 \cdot Z^{-1}}{a_0 + a_1 \cdot Z^{-1}}\right) \cdot \left(\frac{a_2 + a_3 \cdot Z^{-1} + a_4 \cdot Z^{-2}}{a_4 + a_3 \cdot Z^{-1} + a_2 Z^{-2}}\right) \quad (13)$$

a_0	-0.52626196284744	a_2	0.70622791551958
a_1, a_2	1.0	a_3	-1.40992253032663

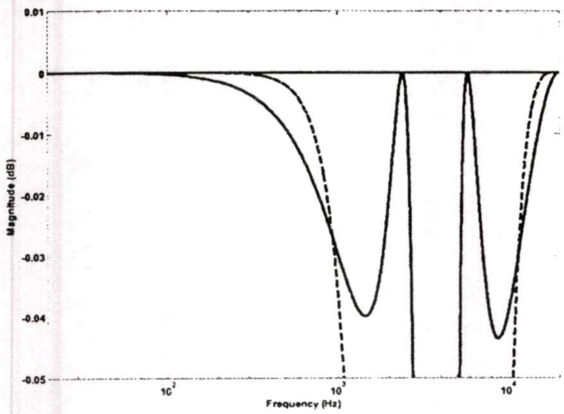
4. ผลการจำลองด้วยโปรแกรม

จากการจำลองด้วยโปรแกรมตามบล๊อกไดอะแกรมตามรูปที่ 4 b) จะได้ผลการตอบสนองทางขนาดต่อความถี่ตลอดย่านความถี่เสียงแสดงดังรูปที่ 5 โดยจะสังเกตเห็นได้ชัดว่าวงจรที่นำเสนอนี้จะมีช่วงแบนด์วิดธ์ระหว่างช่วงเปลี่ยนแปลง (transition band) แคบกว่าแบบวงจรครอสโอเวอร์ลิงค์วิทซ์ไอ ไอ อาร์ และผลรวมของขนาดตลอดช่วงความถี่ออดิโอมีริปเปิ้ลจากยอดถึงยอด 0.04dB โดยเฉพาะบริเวณช่วงเปลี่ยนแปลงจะมีลักษณะราบเรียบซึ่งตรงตามคุณสมบัติของฟังก์ชั่นผ่านตลอด (all-pass function) และเป็นคุณสมบัติที่สำคัญสำหรับวงจรครอสโอเวอร์เน็ทเวอค์



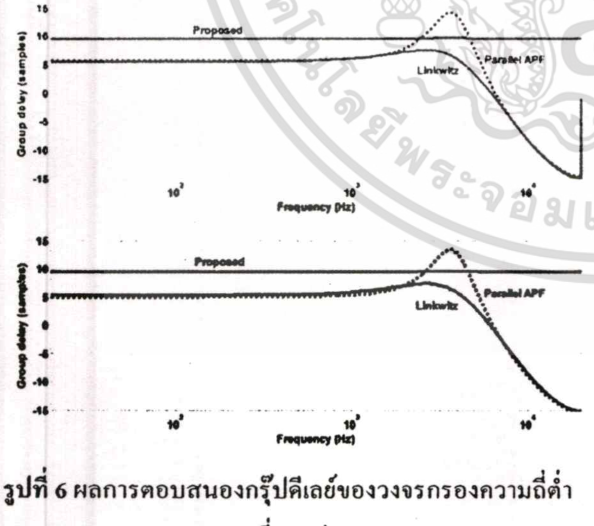
รูปที่ 5 ผลการตอบสนองความถี่ตลอดย่านความถี่ของระบบที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 ผลการตอบสนองความถี่รวมตลอดย่านความถี่ของระบบที่นำเสนอ

ผลการตอบสนองกรุ๊ปดีเลย์ของวงจรในรูป 4 แสดงไว้ในรูป 6 สำหรับวงจรกรองความถี่ต่ำแสดงในรูปบนและรูปล่างสำหรับวงจรกรองความถี่สูง และผลการตอบสนองกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ(รูปบน) เส้นทึบแสดงกรุ๊ปดีเลย์ ของระบบที่นำเสนอซึ่งจะเห็นว่าจะมีลักษณะที่ราบเรียบคงที่แต่ ส่วนเส้นประเป็นระบบที่สร้างจากวงจรกรองความถี่ตลอดย่านต่อขนานกัน 2 วงจรและวงจรของลิงค์วิทซ์ซึ่งจะให้ผลตอบสนองไม่ราบเรียบในบริเวณช่วง เปลี่ยนแปลง โดยในส่วนของวงจรครอสโอเวอร์สำหรับความถี่สูงผ่านก็ให้ผลตอบสนองในลักษณะที่คล้ายกัน



รูปที่ 6 ผลการตอบสนองกรุ๊ปดีเลย์ของวงจรกรองความถี่ต่ำ (บน) และวงจรกรองความถี่สูง (ล่าง)

5. สรุป

บทความนี้นำเสนอการออกแบบโครงสร้างวงจรดิจิตอลครอสโอเวอร์เน็ตเวิร์คสองทางแบบใหม่ ที่อาศัยหลักการของวงจรกรองความถี่แบบนอนคอซอลที่ประมวลผลในเวลาจริง โดยโครงสร้างที่นำเสนอนี้จะมีผลตอบสนองกรุ๊ปดีเลย์ใกล้เคียงราบเรียบ เมื่อเทียบกับวิธีการออกแบบวงจรที่ใช้โครงสร้างของวงจรกรองความถี่ตลอดย่านความถี่ต่อขนานกัน และวงจรกรองแบบลิงค์วิทซ์ไอ ไออาร์ และมีผลตอบสนองความถี่รวมตลอดย่านความถี่ออดิโอมีลักษณะราบเรียบ และมีความซับซ้อนที่ไม่มาก และสามารถสามารถนำไปสร้างได้บนวงจรรวมและโปรแกรมซอฟแวร์

เอกสารอ้างอิง

[1] P. Reviriego, J. Parera and R. Garcia, "Linear-Phase Crossover Design Using Digital IIR Filtes," Journal of the Audio Eng. Society, Vol. 46, No. 5, pp. 406-410, May, 1998

[2] Tapio Saramaki, "On the Design of Digital Filters as a Sum of Two All-Pass Filters", IEEE Trans., Vol. CAS-32, No. 11, pp. 1191-1193, Nov. 1985

[3] Phillip A. Regalia, Sanjit K. Mitra and P. P. Vaidyanathan, "The Digital All-Pass Filter : A Versatile Signal Processing Building Block", Proc. IEEE, Vol. 76, No. 1, Jan. 1988.

[4] P. P. Vaidyanathan A. Sanjit K. Mitra Yrjo Neuvo, "A New Approach to the Realization of Low-Sensivity IIR Digital Filters", IEEE Trans. ASSP., Vol. 34, No.2, pp.350-361, Apar. 1986.

[5] Scott R. Powell and Paul M. Chaul ,"A Technique for Realizing Linear Phase IIR Filter", IEEE Trans, Signal Processing, Vol.39, No.11, PP. 2425-2435, Nov.1991.

[6] D.Sookcharoenphol, K.Janjitrapongvej and S. Tomiyama, "An anaysis of linear phase IIR filters and a condition for perfect linear phase," ICEP 2003, pp. 144-147, Jan., 2003.

[7] S. M. Mitra, "DSP A Computer-Based Approach", McGraw-Hall, New-York, 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นายเจตน์ ออสวัสดิ์ เกิดเมื่อวันที่ 14 ธันวาคม 2521 ที่จังหวัดชัยภูมิ สำเร็จการศึกษาประกาศนียบัตรวิชาชีพชั้นสูงจาก สถาบันเทคโนโลยีราชมงคล วิทยาเขตขอนแก่น ในปีการศึกษา 2542 การศึกษาอุตสาหกรรมศาสตรบัณฑิต (อิเล็กทรอนิกส์) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2544

ในระหว่างปี 2545-2546 เคยทำงานในตำแหน่งผู้ช่วยวิจัยฝ่ายวิจัยและพัฒนาสาขาโทรคมนาคมและเครือข่ายคอมพิวเตอร์ (RDT2) ของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

มีความสนใจเกี่ยวกับ การประมวลผลสัญญาณ (Signal Processing) ระบบสมองกลฝังตัว (Embedded System) การเขียนโปรแกรมไมโครคอนโทรลเลอร์ MCS-51, AVR, PIC, Fujisu การออกแบบวงจรรวมบน CPLD, FPGAs การออกแบบลายวงจรด้วยโปรแกรม Protel 99SE และการเขียนโปรแกรมควบคุมอุปกรณ์แบบเน็ตเวอร์คผ่านระบบ SCADA โปรแกรมไมโครซอฟท์ วิชวลเบสิก เป็นต้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้