

การออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง

REAL-TIME MULTIPLE OUTPUTS FIR FILTER DESIGN

สุกัญญา แพร่สมบูรณ์
SUKANYA PRAESOMBOON

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2308-4

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง

REAL-TIME MULTIPLE OUTPUTS FIR FILTER DESIGN



สุกัญญา แพรสมบูรณ์

SUKANYA PRAESOMBOON

เลขหมู่.....
เลขทะเบียน..... 61669
วัน,เดือน,ปี..... 19 ก.ค. 2549

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ISBN 974-15-2308-4

REAL-TIME MULTIPLE OUTPUTS FIR FILTER DESIGN

SUKANYA PRAESOMBOON

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATIONS ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2006

ISBN 974-15-2308-4

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง
นักศึกษา	นางสาวสุกัญญา แพรสมบูรณ์
รหัสนักศึกษา	47061033
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริงโดยใช้ FPGA ในการออกแบบจะนำโครงสร้างของตัวกรองเชิงเลข FIR แบบ Direct Form มาปรับปรุงให้มี 4 เอาต์พุต แต่จำนวน Delay Element เท่าเดิม ซึ่งทำให้เกิด โครงสร้างของตัวกรองเชิงเลข FIR ขึ้นใหม่ ที่สามารถให้ผลตอบสนองทางความถี่ 4 รูปแบบในเวลาเดียวกัน คือ Low-pass High-pass Band-pass และ Band-stop นอกจากนี้ยังมีผลตอบสนองทางเฟสเป็นแบบเชิงเส้นตามคุณสมบัติของตัวกรองเชิงเลข FIR การออกแบบและจำลองการทำงานใช้โปรแกรม MATLAB ส่วนการสร้างจริงทำบนบอร์ดทดลอง APEX DSP FPGA ผลการทดลองจากการสร้างจริงด้วยบอร์ด APEX DSP FPGA พบว่าผลตอบสนองทางความถี่ของตัวกรองเชิงเลข FIR ที่ออกแบบโดยใช้โครงสร้างใหม่ทั้ง 4 เอาต์พุตมีผลการทดลองที่ถูกต้องตามหลักทฤษฎี

Thesis Title	Real-Time Multiple Outputs FIR Filter Design
Student	Miss Sukanya Praesomboon
Student ID.	47061033
Degree	Master of Engineering
Programme	Telecommunications Engineering
Year	2006
Thesis Advisor	Assoc.Prof.Dr.Kobchai Dejhan

ABSTRACT

This paper proposes a new design of real time multiple outputs FIR filter on FPGA. This FIR filter improves direct form of FIR filter to create a filter with 4 outputs with the same number of delays as the single output version. It creates the new FIR filter structure that generates 4 forms of frequency response at the real time such as low-pass, high-pass, band-pass and band-stop with linear phase response. The filter design and simulation are executed based on MATLAB and implemented on the APEX DSP FPGA board. The experiments on the APEX DSP FPGA board show that frequency responses of this FIR filter are quite accuracy compared to the theoretical results.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดีด้วยคำแนะนำและคำปรึกษาจาก รศ.ดร. กอบชัย เดชหาญ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ศศ.สุรพันธ์ ยิ้มมั่น และ อาจารย์พวง เดชอยู่ ข้าพเจ้ารู้สึกทราบบ้างในความอนุเคราะห์จากท่านอาจารย์ทั้งสามท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือในเรื่องต่างๆ

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนในสาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือและคอยเป็นกำลังใจ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

สุกัญญา แพรสมบูรณ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	2
1.6 ขอบเขตการวิจัย.....	2
1.7 ขั้นตอนการศึกษา.....	3
บทที่ 2 ทฤษฎีตัวกรองเชิงเลข.....	4
2.1 สัญญาณที่ไม่ต่อเนื่องทางเวลา.....	5
2.1.1 การกระทำของสัญญาณที่ไม่ต่อเนื่องทางเวลา.....	10
2.2 ระบบแบบไม่ต่อเนื่องทางเวลา.....	12
2.2.1 ไดอะแกรมที่ใช้แสดงในระบบไม่ต่อเนื่อง.....	12
2.2.2 ชนิดของระบบไม่ต่อเนื่อง.....	13
2.2.3 การเชื่อมต่อระบบไม่ต่อเนื่อง.....	16
2.2.4 การวิเคราะห์ระบบไม่ต่อเนื่องแบบเชิงเส้นและแปรตามเวลา.....	17
2.2.5 การสร้างระบบไม่ต่อเนื่องทางเวลา.....	20
2.3 z-Transform.....	23
2.3.1 นิยามของการแปลงแซด.....	23
2.3.2 คุณสมบัติของ z-Transform.....	24
2.3.3 Rational z-Transform.....	26

สารบัญ (ต่อ)

	หน้า
2.3.4	ฟังก์ชันของระบบเชิงเส้นที่แปรตามเวลา.....28
2.3.5	การแปลง z ผกผัน.....29
2.3.5.1	การแปลง z ผกผันด้วยวิธีอนุกรมอนันต์.....29
2.3.5.2	การแปลง z ผกผันด้วยวิธีแยกเศษส่วนย่อย.....30
2.3.5.3	การแปลง z ผกผัน โดยวิธี Residue31
2.4	ผลตอบสนองความถี่.....31
2.4.1	ผลตอบสนองทางขนาด.....32
2.4.2	ผลตอบสนองทางเฟส.....32
2.5	ตัวกรองดิจิทัลแบบ FIR.....35
2.5.1	การออกแบบตัวกรองเชิงเลขแบบ FIR.....39
2.5.2	การออกแบบตัวกรองเชิงเลขแบบ FIR ด้วยฟังก์ชันวินโดว์.....43
2.5.3	การออกแบบตัวกรองความถี่สูงผ่าน ช่วงความถี่ผ่าน ช่วงความถี่หยุดแบบ FIR47
2.6	ระบบตัวเลขที่ใช้ในการประมวลผล.....54
2.7	FPGA55
บทที่ 3	การออกแบบตัวกรองเชิงเลข FIR.....56
3.1	การออกแบบโครงสร้างของตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง.....56
3.2	การออกแบบตัวกรองเชิงเลข FIRหลายเอาต์พุตแบบเวลาจริง.....59
3.3	การสร้างตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริงด้วย FPGA72
3.4	การประยุกต์ใช้งาน.....76
บทที่ 4	ผลการทดลอง.....77
4.1	การออกแบบหาค่าสัมประสิทธิ์ของตัวกรองเชิงเลข FIR แบบ Low-pass และจำลองการทำงานตัวกรองเชิงเลข FIR หลายเอาต์พุต ด้วยโปรแกรม MATLAB.....77

สารบัญ (ต่อ)

	หน้า
4.2 การออกแบบและสร้างตัวกรองเชิงเลข FIR หลายเข้าพุตโดยการสร้างจริง ด้วย FPGA ตามโครงสร้างในรูปที่ 3.2.....	77
4.3 ทำการวัดผลตอบสนองทางขนาดด้วยเครื่องวิเคราะห์ สัญญาณแบบไดนามิกส์ (Dynamic Signal Analyzer).....	77
4.4 การเปรียบเทียบผลการออกแบบระหว่างการจำลองด้วยโปรแกรม MATLAB.....	77
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	91
เอกสารอ้างอิง.....	92
ภาคผนวก ก ภาษาวีเอชดีแอล.....	93
ภาคผนวก ข การเผยแพร่งานวิจัย.....	109
ประวัติผู้เขียน.....	115

สารบัญตาราง

ตารางที่	หน้า
2.1 z-Transform ของสัญญาณแบบไม่ต่อเนื่อง	24
3.1 การเปรียบเทียบการใช้ Delay Adder และ Multiply จากหลักการเดิมกับหลักการที่ นำเสนอกำหนดจำนวนลำดับ M	59
3.2 การเปรียบเทียบการใช้ Delay Adder และ Multiply จากหลักการเดิมกับหลักการที่ นำเสนอกำหนดจำนวนลำดับ $M = 65$	59

สารบัญรูป

รูปที่	หน้า
2.1 ไคอะแกรมของระบบประมวลผลสัญญาณดิจิทัล.....	1
2.2 ไคอะแกรมของระบบประมวลผลสัญญาณดิจิทัลที่สมบูรณ์.....	1
2.3 สัญญาณต่อเนื่องและสัญญาณไม่ต่อเนื่อง.....	5
2.4 Unit Sample Sequence.....	6
2.5 Unit Step Sequence.....	6
2.6 Exponential Sequences.....	7
2.7 Sinusoidal Sequences.....	8
2.8 Periodic Signals Sequence.....	8
2.9 Symmetric (even) Sequence.....	9
2.10 Antisymmetric (odd) Sequence.....	9
2.11 Block Diagram ของระบบแบบไม่ต่อเนื่อง.....	11
2.12 ไคอะแกรมการบวกในระบบแบบไม่ต่อเนื่อง.....	12
2.13 ไคอะแกรมตัวคูณคงที่ในระบบแบบไม่ต่อเนื่อง.....	12
2.14 ไคอะแกรมตัวคูณสัญญาณในระบบแบบไม่ต่อเนื่อง.....	12
2.15 ไคอะแกรมตัวหั่นวงในระบบไม่ต่อเนื่อง.....	13
2.16 ไคอะแกรมตัวล้าหน้าในระบบไม่ต่อเนื่อง.....	13
2.17 ระบบไม่ต่อเนื่องที่แปรตามเวลาและไม่แปรตามเวลา.....	14
2.18 แสดงระบบ τ เป็นระบบที่เชิงเส้น.....	15
2.19 การเชื่อมต่อระบบไม่ต่อเนื่องแบบอนุกรม.....	16
2.20 การเชื่อมต่อระบบไม่ต่อเนื่องแบบขนาน.....	16
2.21 การแทนสัญญาณแบบไม่ต่อเนื่องในรูป Impulse.....	18
2.22 คุณสมบัติการสลับที่.....	19
2.23 คุณสมบัติการจัดกลุ่ม.....	19
2.24 คุณสมบัติการกระจาย.....	20
2.25 โครงสร้างของระบบเชิงเส้นแปรตามเวลา.....	21
2.26 โครงสร้างแบบตรง I ของระบบไม่ต่อเนื่องแบบ Recursive.....	22
2.27 โครงสร้างแบบตรง II ของระบบไม่ต่อเนื่องแบบ Recursive.....	22
2.28(a) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ.....	27
2.28(b) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ.....	27

สารบัญญรูป (ต่อ)

รูปที่	หน้า
2.28(c) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ.....	28
2.28(d) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ	28
2.29 ผลตอบสนองความถี่ของระบบ $H(z) = \frac{z+1}{z}$	34
2.30 ผลตอบสนองความถี่ของระบบ $y(n) = 0.9y(n-1) + x(n)$	36
2.31 แสดงโครงสร้างของตัวกรองเชิงเลข FIR	37
2.32 แสดงผลตอบสนองอิมพัลส์ของเฟสเชิงเส้นของตัวกรอง 4 ชนิด	39
2.33 แสดงผลตอบสนองความถี่และผลตอบสนองอิมพัลส์ในทางอุดมคติ.....	40
2.34 แสดงผลการตอบสนองความถี่	43
2.35 แสดงผลการตอบสนองความถี่ด้วยฟังก์ชันวินโดว์.....	44
2.36 แสดงกราฟตอบสนองความถี่ของอลพาสฟิลเตอร์ โลพาสฟิลเตอร์ ไฮพาสฟิลเตอร์.....	48
2.37 แสดงกราฟผลตอบสนองความถี่ของตัวกรองความถี่ต่ำผ่าน 2 ความถี่ และกราฟผลตอบสนองความถี่ของตัวกรองช่วงความถี่ผ่าน	50
2.38 แสดงผลตอบสนองของตัวกรองอลพาส ช่วงความถี่ผ่าน และ ช่วงความถี่หยุด.....	52
3.1 โครงสร้างแบบตรงของตัวกรองเชิงเลข FIR	56
3.2 โครงสร้างของตัวกรองเชิงเลข FIR ที่มีผลตอบสนองความถี่ หลายรูปแบบในเวลาเดียวกัน.....	58
3.3 โครงสร้างของ FPGA	71
3.4 Timing Diagram ของ FPGA	72
3.5 Crossover Filter	74
4.1ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	78
4.1ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	78
4.1 ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33	78
4.2ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	79

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	79
4.2ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33.....	79
4.3ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	80
4.3ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	80
4.3ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 33	80
4.4ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	81
4.4ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	81
4.4ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 33.....	81
4.5ก ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	82
4.5ข ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	82
4.5 ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$	82
4.6ก ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB	83
4.6ข ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	83
4.6ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$	83

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.7ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB.....	84
4.7ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	84
4.7ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 65	85
4.8ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB	85
4.8ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	85
4.8ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65	86
4.9ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB	86
4.9ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ที่ $f_{c1} = 5kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	86
4.9ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ ลำดับ 65	87
4.10ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB	87
4.10ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	87
4.10ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วย โปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c2} = 10kHz$ ลำดับ 65.....	88
4.11ก ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB	88
4.11ข ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	88

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่จุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65.....	89
4.12ก ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB	89
4.12ข ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Signal Analyzer.....	89
4.12ค ผลตอบสนองทางความถี่จากการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65	90
ก.1 การออกแบบระบบดิจิทัล	96
ก.2 แสดงขั้นตอนการออกแบบระบบดิจิทัล	97
ก.3 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี	101
ก.4 แสดงรูปแบบของมัลติเพลกซ์	102
ก.5 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย	103
ก.6 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก	103
ก.7 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม	104
ก.8 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน	105
ก.9 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์	105
ก.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภท โครงสร้าง	106
ก.11 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพหุคูณ	106
ก.12 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็คเกจ	107
ก.13 โครงสร้างของบอดีแพ็คเกจ	108
ก.14 โครงสร้างโดยทั่วไปของหน่วยการออกแบบ โครงแบบ.....	108

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ตัวกรองเชิงเลข FIR เป็นตัวกรองเชิงเลขรูปแบบหนึ่งที่ถูกนำมาประยุกต์ใช้งานกันอย่างกว้างขวาง ทั้งนี้เนื่องจากตัวกรองเชิงเลข FIR มีคุณสมบัติที่ดีหลายประการเมื่อเทียบกับตัวกรองเชิงเลข IIR เช่น มีผลตอบสนองทางเฟสเป็นเชิงเส้น มีเสถียรภาพดีเนื่องจากเป็นระบบแบบไม่ต่อเนื่อง (Discrete Time System) ที่มีโพล (Pole) อยู่ที่จุดศูนย์กลาง ของวงกลมรัศมีหนึ่งหน่วย นอกจากนี้ยังสามารถออกแบบให้มีผลตอบสนองทางความถี่ได้หลากหลายรูปแบบ ในการออกแบบตัวกรองเชิงเลข FIR สามารถกระทำได้หลายวิธีเช่น Fourier Series Frequency Sampling [1,2,3,4] เป็นต้น และสำหรับการสร้างตัวกรองเชิงเลข FIR นั้นกระทำได้โดยใช้โครงสร้างแบบ Direct Form หรือโครงสร้างแบบ Transpose Form โดยแต่ละโครงสร้างนั้นจะมีเพียงหนึ่งเอาต์พุตเท่านั้น ดังนั้นในการสร้างตัวกรองเชิงเลข FIR ด้วยโครงสร้างทั้งสองแบบนี้จะทำให้ได้ผลตอบสนองทางความถี่เพียงรูปแบบเดียวโดยผลตอบสนองทางความถี่ที่ได้จะเป็น Low-pass หรือ High-pass หรือ Band-pass หรือ Band-stop อย่างใดอย่างหนึ่ง ซึ่งจะเห็นได้ว่าการจัดโครงสร้างของตัวกรองเชิงเลข FIR ให้เป็นแบบ Direct Form หรือแบบ Transpose Form จึงมีข้อจำกัดที่จะทำให้เกิดผลตอบสนองทางความถี่หลายๆรูปแบบในเวลาเดียวกัน

ดังนั้นวิทยานิพนธ์นี้จึงได้นำเสนอการออกแบบ โดยใช้หลักการของตัวกรองเชิงเลข FIR ในการออกแบบเริ่มจากการออกแบบตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ (Cut-off Frequency) ต่างกัน 2 ความถี่ จากนั้นทำการจัดโครงสร้างของตัวกรองเชิงเลข FIR แบบ Low-pass ทั้งสองโดยใช้ Delay Element ร่วมกันแต่เพิ่ม Multiply Element และ Adder Element เพื่อให้ได้โครงสร้างใหม่ที่มี 4 เอาต์พุต และนอกจากนี้ยังสามารถเพิ่มจำนวน Order ของตัวกรองได้อย่างไม่จำกัด ทั้งนี้ขึ้นกับความสามารถของตัวประมวลผล

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้มุ่งหวังเพื่อออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง คือ ความถี่ต่ำผ่าน (Low-pass) ความถี่สูงผ่าน (High-pass) ความถี่ผ่าน (Band-pass) และความถี่หยุด (Band-stop) และนำมาสร้างจริงบนบอร์ด FPGA โดยใช้โปรแกรม QuatusII ของบริษัท Altera

1.3 สมมติฐานของการศึกษา

ตัวกรองเชิงเลข FIR มีคุณสมบัติที่คืออยู่หลายประการซึ่งทำให้สามารถนำไปประยุกต์กับงานหลายด้าน โดยเฉพาะงานที่เกี่ยวกับการกำจัดสัญญาณรบกวนและงานที่ใช้ปรับปรุงคุณภาพสัญญาณ[6] ทำให้ปัจจุบันได้มีการสร้างชิปสำเร็จรูปที่ทำหน้าที่เป็นตัวกรองเชิงเลข FIR ขึ้นแต่เนื่องจากชิปเหล่านั้นมีเพียงหนึ่งเอาต์พุตทำให้ไม่สามารถสร้างให้มีผลตอบสนองทางความถี่ได้หลายแบบในเวลาเดียวกัน โดยทำได้เพียงเป็น Low-pass หรือ High-pass หรือ Band-pass หรือ Band-stop อย่างใดอย่างหนึ่งเท่านั้น แต่สำหรับการประยุกต์ใช้งานบางประเภทจำเป็นต้องใช้ตัวกรองเชิงเลขที่มีผลตอบสนองความถี่หลายรูปแบบในเวลาเดียวกัน ดังนั้นจำเป็นต้องใช้ชิปที่ทำหน้าที่เป็นตัวกรองเชิงเลขเพิ่มขึ้นเพื่อให้สอดคล้องกับงานในปัจจุบัน

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

ออกแบบตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง โดยใช้ FPGA ในการออกแบบจะนำโครงสร้างของตัวกรองเชิงเลขแบบ Direct Form มาปรับปรุงให้มีหลายเอาต์พุตโดยใช้ Delay Element และ Adder Element ร่วมกันซึ่งทำให้เกิดโครงสร้างของตัวกรองเชิงเลข FIR ขึ้นใหม่ที่สามารให้ผลตอบสนองทางความถี่ (Frequency Response) 4 รูปแบบ ในเวลาเดียวกัน คือ Low-pass High-pass Band-pass และ Band-stop ในเวลาเดียวกัน

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

เนื่องจากการสร้างตัวกรองเชิงเลข FIR ด้วยวิธีพื้นฐานนั้นจะใช้โครงสร้างแบบตรง (Direct Form) มาออกแบบซึ่งจะเห็นได้ว่าเป็นระบบแบบไม่ต่อเนื่องที่มีเพียงเอาต์พุตเดียวทำให้การสร้างตัวกรองเชิงเลข FIR จะได้ผลตอบสนองเพียงรูปแบบเดียวดังนั้นจึงนำเสนอวิธีใหม่โดยการจัดโครงสร้างของตัวกรองเชิงเลข FIR ที่สามารให้ผลตอบสนองความถี่หลายรูปแบบในเวลาเดียวกัน

1.6 ขอบเขตการวิจัย

1.6.1 ออกแบบและสร้างตัวกรองเชิงเลข FIR หลายเอาต์พุตแบบเวลาจริง คือ Low-pass High-pass Band-pass และ Band-stop โดยใช้บอร์ด FPGA

1.6.2 เปรียบเทียบผลตอบสนองทางความถี่ที่สร้างจริงจาก FPGA กับผลการจำลองการทำงานด้วยโปรแกรม MATLAB

1.7 ขั้นตอนของการศึกษา

1.7.1 การวิจัยมีขั้นตอนดังนี้

- 1.7.1.1 ศึกษาหลักการของตัวกรองเชิงเลข FIR
- 1.7.1.2 ปรับปรุงโครงสร้างของตัวกรองเชิงเลข FIR ให้สามารถตอบสนองความถี่ 4 รูปแบบในเวลาเดียวกัน
- 1.7.1.3 ศึกษาโครงสร้างของชิพ FPGA (Field Programmable Gate Array)
- 1.7.1.4 ศึกษาโปรแกรม QuartusII
- 1.7.1.5 สร้างแบบจำลองของตัวกรองเชิงเลขแบบ FIR ที่ตอบสนองความถี่ 4 รูปแบบในเวลาเดียวกันและจำลองด้วยโปรแกรม MATLAB

1.7.2 เครื่องมือในการวิจัย

- 1.7.2.1 FPGA ของบริษัท Altera
- 1.7.2.2 โปรแกรม MATLAB
- 1.7.2.3 โปรแกรม Quartus II
- 1.7.2.4 เครื่อง Dynamic Signal Analyzer

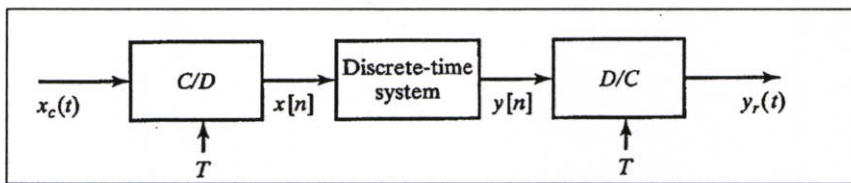
1.7.3 วิธีเก็บรวบรวมข้อมูล

ทดสอบการทำงานของตัวกรองเชิงเลขแบบ FIR ที่ออกแบบด้วยชิพ FPGA และเปรียบเทียบผลการจำลองการทำงานด้วยโปรแกรม MATLAB

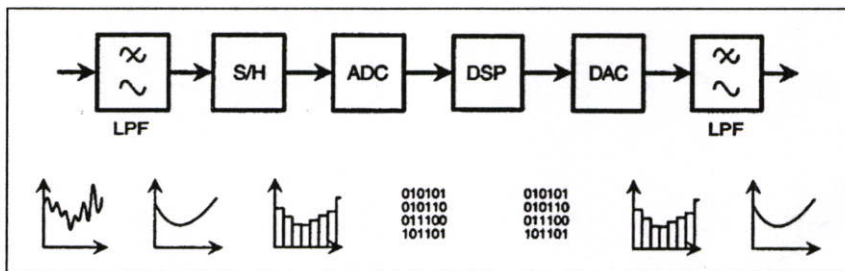
บทที่ 2

ทฤษฎีตัวกรองเชิงเลข

โดยทั่วไปสัญญาณไฟฟ้าที่เกิดขึ้นจากตัวตรวจจับ (Sensor) ส่วนใหญ่มักจะเป็นสัญญาณอนาล็อก ตัวอย่างเช่น สัญญาณเสียงที่ได้จากไมโครโฟน สัญญาณรูปที่ได้จากกล้องวิดีโอ แรงดันไฟฟ้าที่ได้จากตัวตรวจจับอุณหภูมิ สัญญาณทางการแพทย์ ซึ่งสัญญาณที่ได้กล่าวมาแล้วนี้เป็นสัญญาณอนาล็อกหรือสัญญาณที่มีความต่อเนื่องทางเวลา (Continuous Time Signal) ทั้งสิ้น ดังนั้นถ้าจะทำการวิเคราะห์สัญญาณเหล่านี้ด้วยระบบประมวลผลสัญญาณดิจิทัลจำเป็นต้องแปลงสัญญาณอนาล็อกหรือสัญญาณที่ต่อเนื่องทางเวลาให้เป็นสัญญาณดิจิทัลหรือสัญญาณที่ไม่ต่อเนื่องทางเวลา (Discrete Time Signal) โดยใช้วงจรแปลงอนาล็อกเป็นดิจิทัล (Analog to Digital Converter) จากนั้นข้อมูลจะถูกส่งเข้าไปยังระบบประมวลผลสัญญาณแบบไม่ต่อเนื่อง (Discrete Time System) เพื่อทำการคำนวณผลที่ได้จากการคำนวณก็จะถูกแปลงกลับให้เป็นสัญญาณอนาล็อกหรือสัญญาณที่ต่อเนื่องทางเวลา โดยวงจรแปลงดิจิทัลเป็นอนาลอกดังแสดงดังรูปที่ 2.1 แต่ในการใช้งานจริงการป้อนสัญญาณที่ต่อเนื่องเข้าไปยังวงจรแปลงอนาลอกเป็นดิจิทัลอาจเกิดข้อผิดพลาดเนื่องจากการสุ่มสัญญาณ โดยเฉพาะสัญญาณที่มีความถี่สูงความผิดพลาดนี้เรียกว่าการเกิด Aliasing การแก้ไขสามารถทำได้โดยการต่อ Anti-aliasing Filter (Low-pass Filter) เข้าทางด้าน Input และทางด้าน Output ก็เช่นเดียวกันสัญญาณที่ได้จากวงจรแปลงดิจิทัลเป็นอนาลอกจำเป็นต้องนำมาผ่าน Reconstruction Filter (Low-pass Filter) เพื่อปรับสัญญาณให้เรียบดังแสดงในรูปที่ 2.2



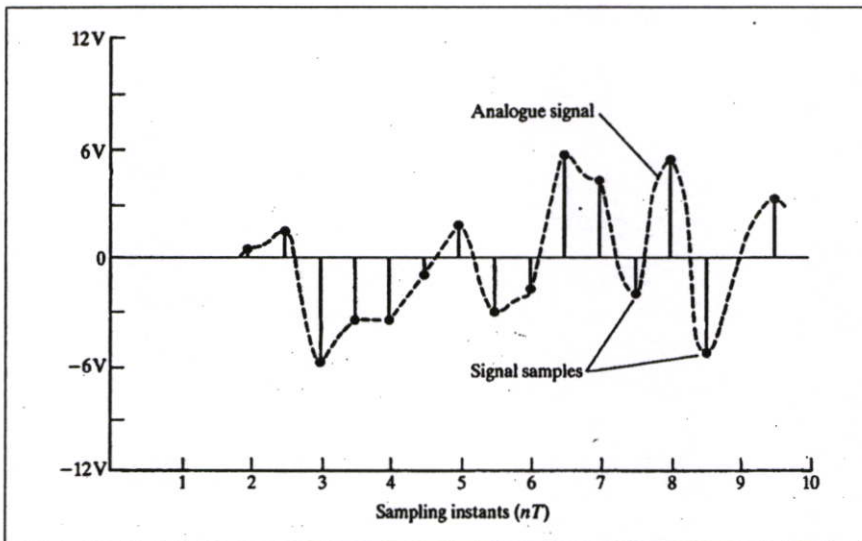
รูปที่ 2.1 ไคอะแกรมของระบบประมวลผลสัญญาณดิจิทัล



รูปที่ 2.2 ไคอะแกรมของระบบประมวลผลสัญญาณดิจิทัลที่สมบูรณ์

2.1 สัญญาณแบบไม่ต่อเนื่องทางเวลา [1,2]

ในระบบประมวลผลสัญญาณแบบไม่ต่อเนื่องสัญญาณที่ใช้ในระบบก็จะเป็นสัญญาณที่ไม่ต่อเนื่องเช่นเดียวกัน ดังที่ได้กล่าวมาแล้วซึ่งสัญญาณที่ไม่ต่อเนื่องนี้จะมีลักษณะดังรูปที่ 2.3 โดยจะเห็นว่าสัญญาณที่ไม่ต่อเนื่องก็คือสัญญาณค่าหนึ่ง ณ เวลาหนึ่งบนสัญญาณที่ต่อเนื่องแต่จะมีระยะเวลาห่างที่เท่ากันในแต่ละจุด เรียกระยะเวลาห่างที่เท่ากันนี้ว่า Sampling Time Sampling Rate หรือ Sampling Periods สัญญาณที่ไม่ต่อเนื่องสามารถเรียกเป็นลำดับสัญญาณ (Sequence Signal) ก็ได้ และสามารถเขียนแทนได้ด้วยสมการคณิตศาสตร์ได้หลายๆแบบดังต่อไปนี้



รูปที่ 2.3 สัญญาณต่อเนื่องและสัญญาณไม่ต่อเนื่อง

เขียนในรูปของฟังก์ชัน (Functional Representation)

$$x(n) = \begin{cases} 1, & n = 1, 3 \\ 4, & n = 2, 4 \\ 0, & n > 4 \end{cases} \quad (2.1)$$

เขียนในรูปการแจกแจง (Tabular Representation)

n	...	-2	-1	0	1	2	3	4	...
$x(n)$...	0	0	0	1	4	1	0	...

เขียนในรูปลำดับ (Sequence Representation)

$$x(n) = \{\dots, 0, 0, 1, 4, 1, 0, 0, \dots\} \quad (2.2)$$

$$x(n) = \{0, 1, 4, 1, 0, 0, \dots\} \quad (2.3)$$

$$x(n) = \{3, -1, -2, 5, 0, 4, -1\} \quad (2.4)$$

$$x(n) = \{0, 1, 4, 1\} \quad (2.5)$$

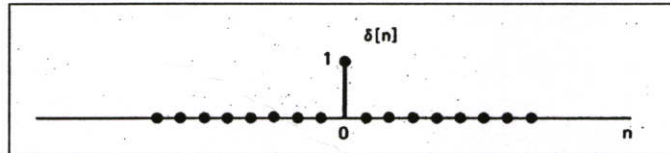
สมการที่ 2.2 และ 2.3 เป็นลำดับแบบไม่มีจุดจบ (Infinite - duration Signal or Sequence)

ส่วนสมการที่ 2.4 และ 2.5 เป็นลำดับแบบมีจุดจบ (Finite - duration Signal or Sequence)

- Unit Sample Sequence

มีนิยามดังสมการที่ 2.6 และมีรูปสัญลักษณ์ดังรูปที่ 2.4

$$\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases} \quad (2.6)$$

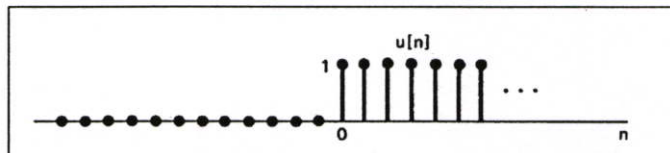


รูปที่ 2.4 Unit Sample Sequence

- Unit Step Sequence

มีนิยามดังสมการที่ 2.7 และมีรูปสัญลักษณ์ดังรูปที่ 2.5

$$u(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases} \quad (2.7)$$

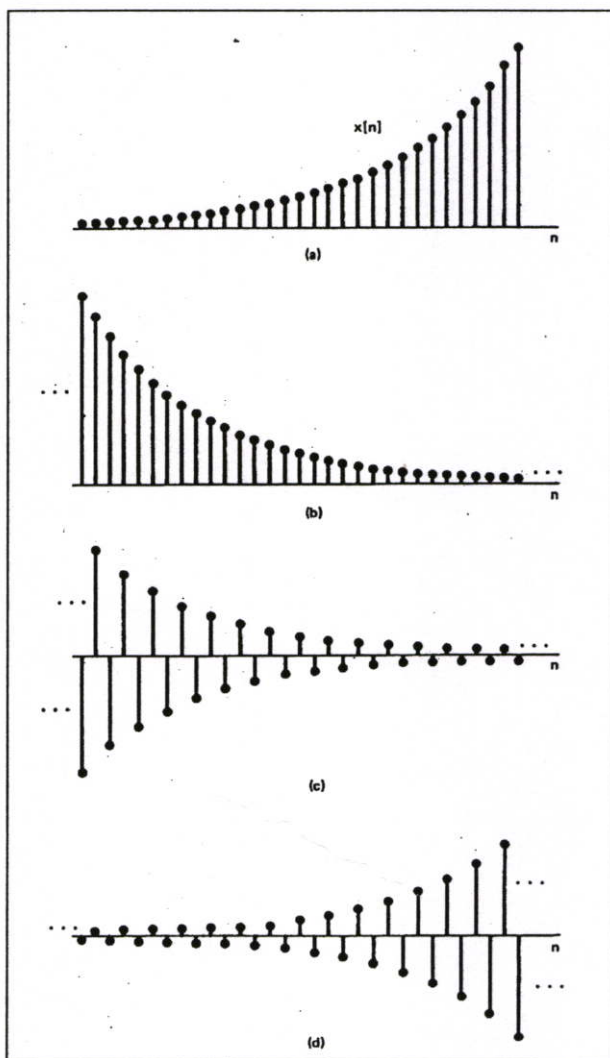


รูปที่ 2.5 Unit Step Sequence

- Exponential Sequences

มีนิยามดังสมการที่ 2.8 และมีรูปสัญญาณดังรูปที่ 2.6

$$x(n) = \alpha^n \quad (2.8)$$



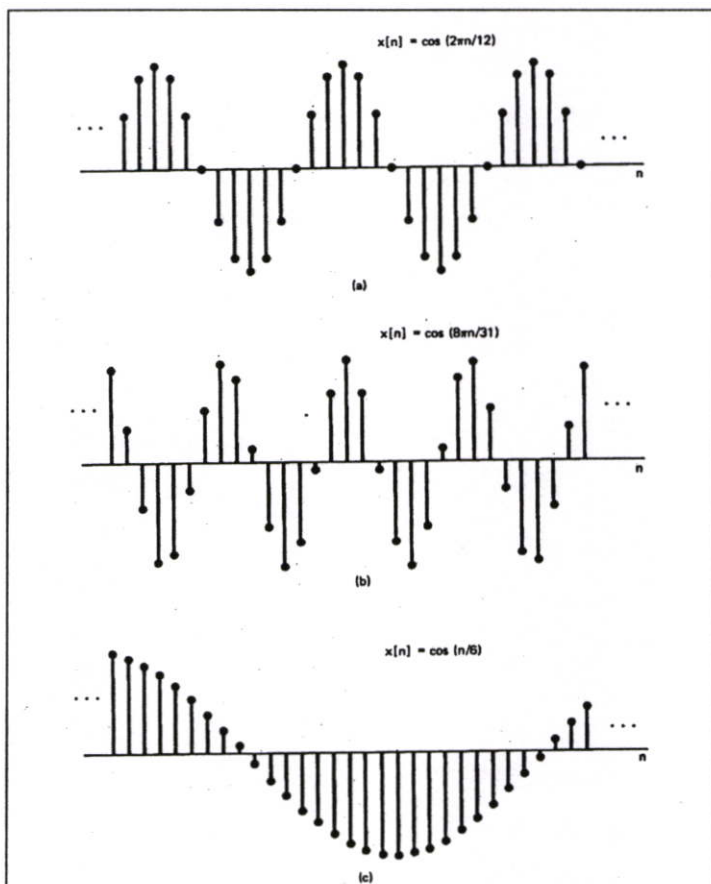
รูปที่ 2.6 Exponential Sequences $x(n) = A\alpha^n$

- (a) $\alpha > 1$
- (b) $0 < \alpha < 1$
- (c) $-1 < \alpha < 0$
- (d) $\alpha < -1$

- Sinusoidal Sequences

มีนิยามดังสมการที่ 2.9 และมีรูปสัญญาณดังรูปที่ 2.7

$$x(n) = A \cos(\omega_0 n + \phi) \quad (2.9)$$



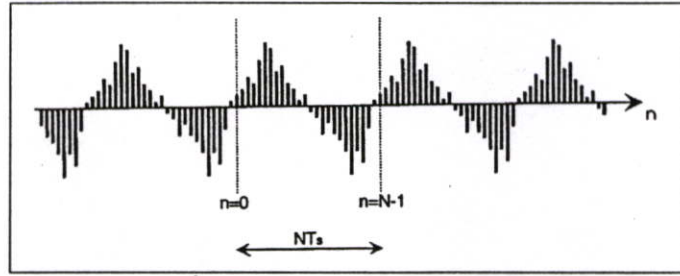
รูปที่ 2.7 Sinusoidal Sequences

- Periodic Signals and Aperiodic Signals

Periodic Signals Sequence เป็นสัญญาณที่มีคาบซ้ำมีนิยามดังสมการที่ 2.10

$$x(n + N) = x(n) \quad (2.10)$$

เมื่อ N คือ Fundamental Period และลำดับไม่เป็นดังสมการที่ 2.10 จะเป็นลำดับแบบ Aperiodic

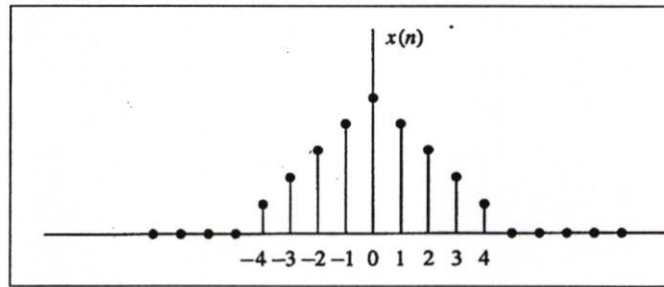


รูปที่ 2.8 Periodic Signals Sequence

- Symmetric (even) and Antisymmetric (odd) Signals

ลำดับสัญญาณที่สมมาตรกัน (Symmetric) มีนิยามดังสมการที่ 2.11 มีรูปสัญญาณดังรูปที่ 2.9

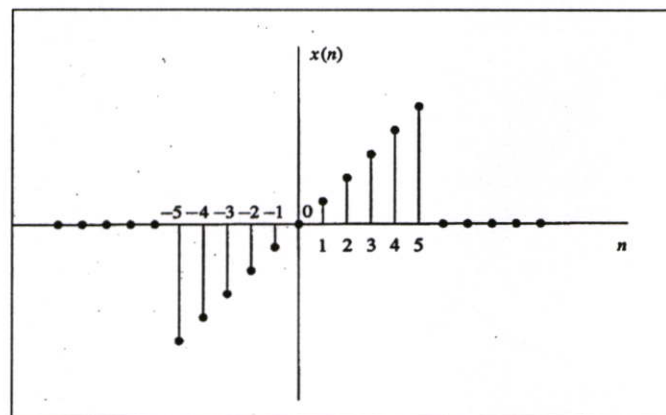
$$x(-n) = x(n) \quad (2.11)$$



รูปที่ 2.9 Symmetric (even) Sequence

ลำดับสัญญาณที่ไม่สมมาตรกัน (Antisymmetric) มีนิยามดังสมการที่ 2.12 มีรูปสัญญาณดังรูปที่ 2.10

$$x(-n) = -x(n) \quad (2.12)$$



รูปที่ 2.10 Antisymmetric (odd) Sequence

2.1.1 การกระทำของสัญญาณแบบไม่ต่อเนื่องทางเวลา (Simple Manipulations of Discrete-time Signals) [5]

- การเลื่อนสัญญาณ

อาจเรียกว่าการหน่วงสัญญาณก็ได้เขียนเป็นสมการได้ดังสมการที่ 2.13

$$y(n) = x(n - k) \quad (2.13)$$

เมื่อ k เป็นจำนวนเต็ม (Integer)

- การบวก การคูณ และการสเกลสัญญาณแบบไม่ต่อเนื่อง (Adding Multiplication and Scaling of Discrete Time Signal)

- การบวกสัญญาณ 2 สัญญาณ หรือ k สัญญาณเข้าด้วยกันได้ดังสมการที่ 2.14 และสมการที่ 2.15

$$y(n) = x_1(n) + x_2(n) \quad -\infty < n < \infty \quad (2.14)$$

$$y(n) = x_1(n) + x_2(n) + \dots + x_k(n) \quad -\infty < n < \infty \quad (2.15)$$

- การคูณสัญญาณ 2 สัญญาณ หรือ k สัญญาณเข้าด้วยกันดังสมการที่ 2.16 และสมการที่ 2.17

$$y(n) = x_1(n)x_2(n) \quad -\infty < n < \infty \quad (2.16)$$

$$y(n) = x_1(n)x_2(n)\dots x_k(n) \quad -\infty < n < \infty \quad (2.17)$$

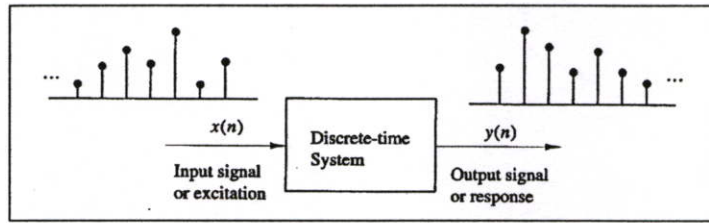
- การสเกลสัญญาณเป็นการปรับขนาดสัญญาณกระทำได้การคูณสัญญาณเข้ากับค่าคงที่ดังสมการที่ 2.18

$$y(n) = Ax(n) \quad -\infty < n < \infty \quad (2.18)$$

เมื่อ A เป็นค่าคงที่

2.2 ระบบแบบไม่ต่อเนื่องทางเวลา (Discrete Time System)[1,2]

ระบบแบบไม่ต่อเนื่องมีไคอะแกรม ดังรูปที่ 2.11



รูปที่ 2.11 Block Diagram ของระบบแบบไม่ต่อเนื่อง

เมื่อ $x(n)$ เป็น Input Signal หรือ Excitation ของระบบแบบไม่ต่อเนื่อง
 $y(n)$ เป็น Output Signal หรือ Response ของระบบแบบไม่ต่อเนื่อง
 จากรูปที่ 2.16 สามารถเขียนความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตได้ดังสมการที่ 2.19

$$y(n) = \tau[x(n)] \quad (2.19)$$

เมื่อ τ ตัวกระทำ (Operator)

นอกจากนี้ยังสามารถแยกระบบแบบไม่ต่อเนื่องได้ตามความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตได้
 ดังเช่นกรณีที่ 2.1

เช้นกรณีที่ 2.1 กำหนดให้ระบบแบบไม่ต่อเนื่องมีอินพุต $x(n)$ ดังสมการที่ 2.20 จงหา Output Response ของระบบ

$$x(n) = \begin{cases} |n|, & -3 \leq n \leq 3 \\ 0, & \text{otherwise} \end{cases} \quad (2.20)$$

$$(a) \ y(n) = x(n)$$

$$(b) \ y(n) = x(n-1)$$

$$(c) \ y(n) = x(n+1)$$

วิธีการของกรณี (a) $y(n) = x(n) = \{\dots, 0, 3, 2, 1, 0, 1, 2, 3, 0, \dots\}$ เรียกระบบที่มีเอาต์พุตเหมือนกับอินพุตว่า Identity System

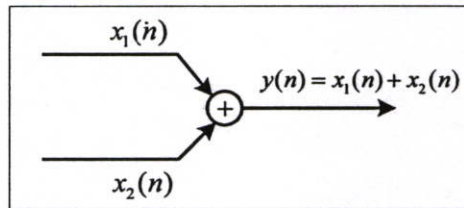
วิธีการของกรณี (b) $y(n) = x(n-1) = \{\dots, 0, 3, 2, 1, 0, 1, 2, 3, 0, \dots\}$ จะเห็นได้ว่าระบบนี้เอาต์พุตจะถูกหน่วงเป็นเวลา 1 Sample เรียกระบบนี้ว่า Delay System

วิธีการของกรณี (c) $y(n) = x(n+1) = \{\dots, 0, 3, 2, 1, 0, 1, 2, 3, 0, \dots\}$ จะเห็นได้ว่าระบบนี้เอาต์พุตจะเกิดก่อน Input เรียกระบบนี้ว่า Advances System และระบบนี้ไม่สามารถสร้างให้เป็นจริงได้

2.2.1 ไคอะแกรมที่ใช้แสดงในระบบไม่ต่อเนื่อง (Block Diagram Representation of Discrete-time Systems)

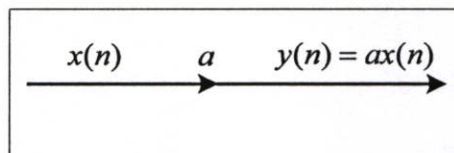
ในระบบแบบไม่ต่อเนื่องสามารถแทนได้ด้วยไคอะแกรมดังต่อไปนี้

- ตัวบวก (Adder) ใช้สำหรับบวกสัญญาณเข้าด้วยกันดังรูปที่ 2.12



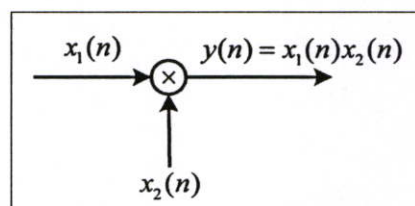
รูปที่ 2.12 ไคอะแกรมการบวกในระบบแบบไม่ต่อเนื่อง

- ตัวคูณคงที่ (Constant Multiplier) ใช้คูณค่าคงที่กับสัญญาณดังรูปที่ 2.13



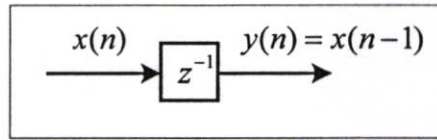
รูปที่ 2.13 ไคอะแกรมตัวคูณคงที่ในระบบแบบไม่ต่อเนื่อง

- ตัวคูณสัญญาณ (Signal Multiplier) ใช้สำหรับคูณสัญญาณเข้าด้วยกันดังรูปที่ 2.14



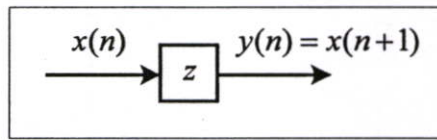
รูปที่ 2.14 ไคอะแกรมตัวคูณสัญญาณในระบบแบบไม่ต่อเนื่อง

- ตัวหน่วง (Delay Element) ใช้สำหรับหน่วงสัญญาณดังรูปที่ 2.15



รูปที่ 2.15 ไคอะแกรมตัวหน่วงในระบบไม่ต่อเนื่อง

- ตัวล้าหน้า (Advance Element) ใช้สำหรับกำหนดค่าล่วงหน้าให้กับระบบดังรูปที่ 2.16



รูปที่ 2.16 ไคอะแกรมตัวล้าหน้าในระบบไม่ต่อเนื่อง

2.2.2 ชนิดของระบบแบบไม่ต่อเนื่อง

- ระบบแบบ Static

เป็นระบบที่ไม่มีหน่วยความจำมีสมการของระบบดังสมการที่ 2.21 และสมการที่ 2.22

$$y(n) = ax(n) \quad (2.21)$$

$$y(n) = nx(n) + bx^3(n) \quad (2.22)$$

- ระบบแบบ Dynamic

เป็นระบบที่มีหน่วยความจำนั้นหมายถึงเป็นระบบที่สามารถบันทึกค่าอินพุตหรือเอาต์พุตที่เวลาผ่านไปแล้วได้มีสมการของระบบดังสมการที่ 2.23 - 2.25

$$y(n) = x(n) + 3x(n-1) \quad (2.23)$$

$$y(n) = \sum_{k=0}^n x(n-k) \quad (2.24)$$

$$y(n) = \sum_{k=0}^{\infty} x(n-k) \quad (2.25)$$

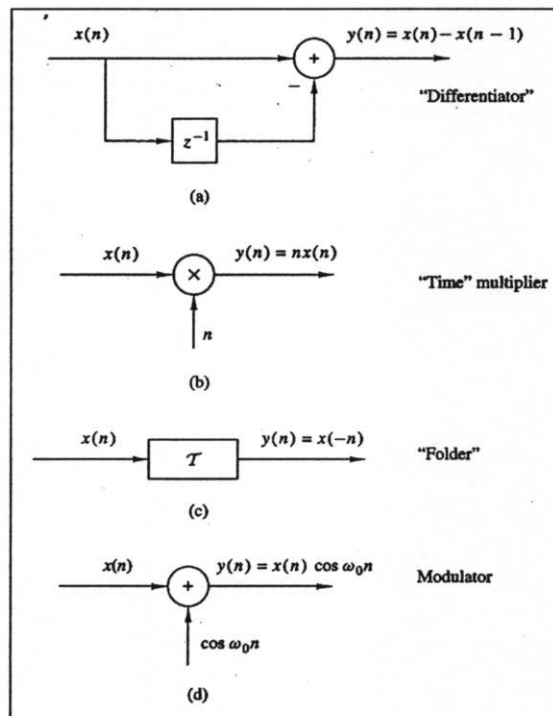
- ระบบแบบแปรตามเวลา (Time-invariant System)

กำหนดให้ \mathcal{T} เป็นระบบไม่ต่อเนื่อง $x(n)$ เป็นอินพุตของระบบ $y(n)$ เป็นเอาต์พุตของระบบ ดังสมการที่ 2.26 ถ้ามีการหน่วงอินพุตเป็น $x(n-k)$ ถ้าเอาต์พุตที่ได้ถูกหน่วงไปเป็น $y(n-k)$ เช่นเดียวกับอินพุตดังสมการที่ 2.27 สามารถสรุปได้ว่าระบบไม่ต่อเนื่อง \mathcal{T} เป็นระบบไม่ต่อเนื่องที่แปรตามเวลา

$$x(n) \xrightarrow{\mathcal{T}} y(n) \quad (2.26)$$

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k) \quad (2.27)$$

และระบบที่อินพุตถูกหน่วงไป $x(n-k)$ แต่ได้เอาต์พุตของระบบไม่เท่ากับ $y(n-k)$ ระบบนั้นเป็นระบบที่ไม่แปรตามเวลา (Time-variant System หรือ Time-varying System) รูปที่ 2.17 แสดงตัวอย่างระบบไม่ต่อเนื่องที่แปรตามเวลาและไม่แปรตามเวลา



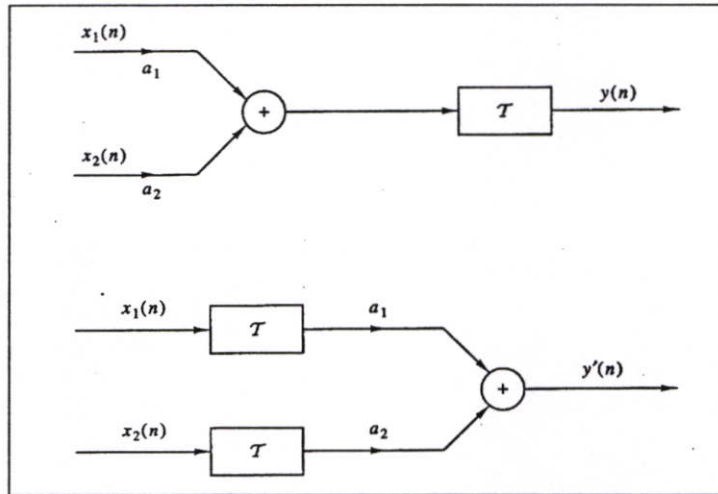
รูปที่ 2.17 ระบบไม่ต่อเนื่องที่แปรตามเวลาและไม่แปรตามเวลา

(a) ระบบแปรตามเวลา (b) - (d) ระบบที่ไม่แปรตามเวลา

- ระบบแบบเชิงเส้นและไม่เชิงเส้น (Linear and Nonlinear Systems)

ระบบไม่ต่อเนื่องแบบเชิงเส้นจะต้องมีคุณสมบัติสอดคล้องกับหลักการของ Superposition ดังสมการที่ 2.28 หรืออธิบายได้ดังรูปที่ 2.18 และระบบที่ไม่มีคุณสมบัติสอดคล้องกับหลักการของ Superposition จะเป็นระบบแบบไม่เชิงเส้น

$$\tau[a_1x_1(n) + a_2x_2(n)] = a_1\tau[x_1(n)] + a_2\tau[x_2(n)] \quad (2.28)$$



รูปที่ 2.18 แสดงระบบ τ เป็นระบบที่เชิงเส้น

และระบบที่ไม่มีคุณสมบัติสอดคล้องกับหลักการของ Superposition จะเป็นระบบแบบไม่เชิงเส้น

- ระบบแบบ Causal และแบบ Noncausal

ระบบแบบ Causal เป็นระบบที่สร้างได้จริงมีสมการของระบบดังสมการที่ 2.29

$$y(n) = F[x(n), x(n-1), x(n-2), \dots, x(n-k), \dots, y(n-1), y(n-2), \dots, y(n-k)]$$

$$y(n) = 2x(n) + 4.5x(n-2) - 3y(n-1) + 0.5y(n-2) \quad (2.29)$$

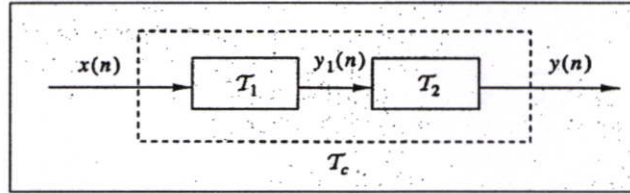
จะเห็นได้ว่าระบบแบบ Causal จะประกอบด้วยเทอมอินพุตปัจจุบันอินพุตในอดีต หรือเอาต์พุตในอดีตเท่านั้น ระบบที่ไม่เป็นดังสมการที่ 2.29 จะเป็นระบบแบบ Noncausal

- ระบบแบบ Stable และระบบแบบ Unstable

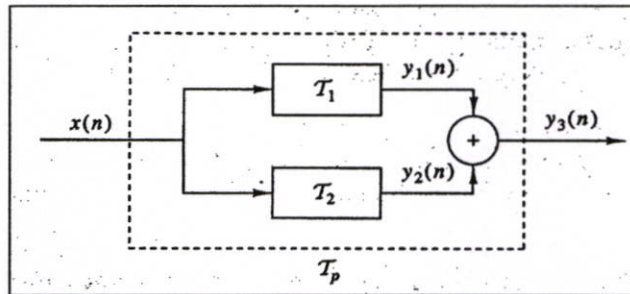
ระบบที่มีเอาต์พุตขึ้นอยู่กับอินพุตจะเป็นระบบแบบ Stable ส่วนระบบที่มีเอาต์พุตไม่ขึ้นอยู่กับอินพุตจะเป็นระบบแบบ Unstable

2.2.3 การเชื่อมต่อระบบไม่ต่อเนื่อง (Interconnection of Discrete-time Systems) [4,5]

กระทำได้ 2 ลักษณะคือ แบบอนุกรม (Cascade) ดังแสดงดังรูปที่ 2.19 และแบบขนาน (Parallel) ดังแสดงดังรูปที่ 2.20



รูปที่ 2.19 การเชื่อมต่อระบบไม่ต่อเนื่องแบบอนุกรม



รูปที่ 2.20 การเชื่อมต่อระบบไม่ต่อเนื่องแบบขนาน

ในการเชื่อมต่อระบบดังรูปที่ 2.19 เขียนเป็นสมการได้ดังสมการที่ 2.30 – 2.32

$$y_1(n) = \tau_1[x(n)] \quad (2.30)$$

$$y(n) = \tau_2[y_1(n)] \quad (2.31)$$

แทนสมการที่ 2.30 ลงในสมการที่ 2.31 จะได้สมการ $y(n)$ ดังสมการที่ 2.32

$$y(n) = \tau_2\{\tau_1[x(n)]\} \quad (2.32)$$

กำหนดให้ $\tau_c = \tau_1\tau_2$ จะได้ $y(n)$ ดังสมการที่ 2.33

$$y(n) = \tau_c[x(n)] \quad (2.33)$$

และถ้ามีการสลับที่ระบบ τ_1 กับ τ_2 $y(n)$ จะมีค่าเปลี่ยนแปลงไปดังนั้นจึงสรุปได้ว่า

$$\tau_1\tau_2 \neq \tau_2\tau_1 \quad (2.34)$$

นอกจากนี้ถ้าระบบ τ_1 และระบบ τ_2 เป็นระบบที่แปรตามเวลาก็จะได้ $\tau_1\tau_2$ หรือ τ_c เป็นระบบที่แปรตามเวลาด้วยดังสมการที่ 2.35

$$\begin{aligned}x(n-k) &\xrightarrow{\tau_1} y_1(n-k) \\y_1(n-k) &\xrightarrow{\tau_2} y(n-k) \\x(n-k) &\xrightarrow{\tau_c=\tau_2\tau_1} y(n-k)\end{aligned}\tag{2.35}$$

ส่วนการเชื่อมต่อแบบขนานเขียนเป็นสมการได้ดังสมการที่ 2.36

$$\begin{aligned}y_3(n) &= y_1(n) + y_2(n) \\&= \tau_1[x(n)] + \tau_2[x(n)] \\&= (\tau_1 + \tau_2)[x(n)] \\&= \tau_p[x(n)]\end{aligned}\tag{2.36}$$

เมื่อ $\tau_p = \tau_1 + \tau_2$

2.2.4 การวิเคราะห์ระบบไม่ต่อเนื่องแบบเชิงเส้นและแปรตามเวลา (Analysis of Discrete-time Linear Time-invariant) [4,5]

- Resolution of a Discrete-time Signal into Impulses

สัญญาณแบบไม่ต่อเนื่องใดๆสามารถเขียนแทนในรูปของ Impulse ได้เช่นดังรูปที่ 2.21 (b) เขียนเป็นสมการได้ดังสมการที่ 2.37

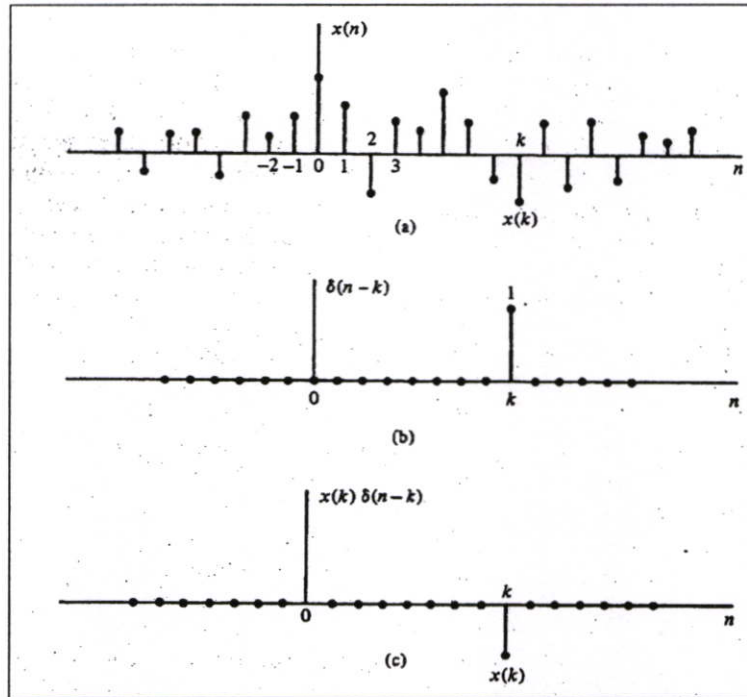
$$x(n) = \delta(n-k)\tag{2.37}$$

และรูปที่ 2.29 (c) เขียนได้ดังสมการที่ 2.38

$$x(n) = x(k)\delta(n-k)\tag{2.38}$$

ดังนั้นรูปที่ 2.29 (a) เขียนแทนได้ดังสมการที่ 2.39

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \quad (2.39)$$



รูปที่ 2.21 การแทนสัญญาณแบบไม่ต่อเนื่องในรูป Impulse

- Convolution Sum

เป็นการผสมผสานระหว่างสัญญาณแบบไม่ต่อเนื่อง $x(n]$ กับผลตอบสนองทางอิมพัลส์ของระบบไม่ต่อเนื่อง $h(n]$ เขียนได้ดังสมการที่ 2.40

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \quad (2.40)$$

เมื่อ $h(k]$ คือ ผลตอบสนองทางอิมพัลส์ของระบบไม่ต่อเนื่อง

$x(n]$ คือ สัญญาณแบบไม่ต่อเนื่อง

จะเห็นได้สมการสำหรับระบบแบบ Causal การทำ Convolution จะทำตามสมการที่ 2.41

$$y(n) = \sum_{k=0}^n h(k)x(n-k) \quad (2.41)$$

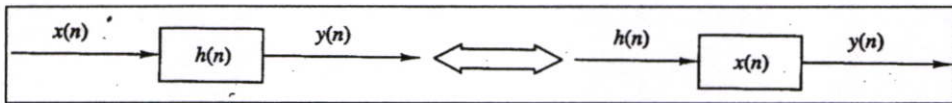
นอกจากนี้สมการ Convolution สามารถเขียนได้อีกแบบหนึ่งดังสมการที่ 2.42

$$y(n) = h(n) * x(n) \quad (2.42)$$

- Commutative law

$$x(n) * h(n) = h(n) * x(n) \quad (2.43)$$

เขียนเป็นไดอะแกรมได้ดังรูปที่ 2.22



รูปที่ 2.22 คุณสมบัติการสลับที่

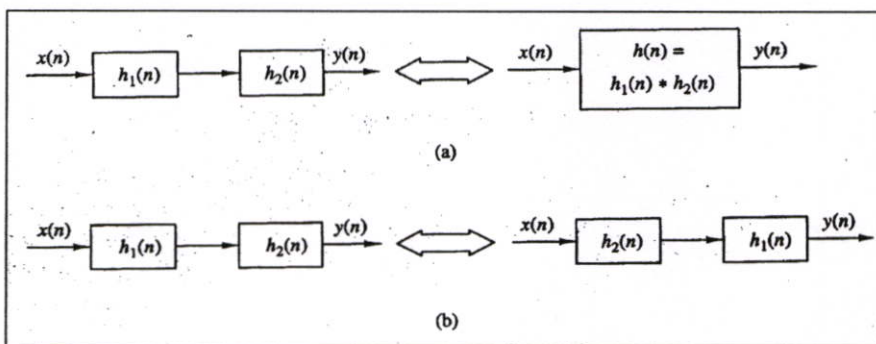
$$y(n) = x(n) * h(n) \quad (2.44)$$

$$y(n) = h(n) * x(n) \quad (2.45)$$

- Associative law

$$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)] \quad (2.46)$$

เขียนเป็นไดอะแกรมได้ดังรูปที่ 2.23

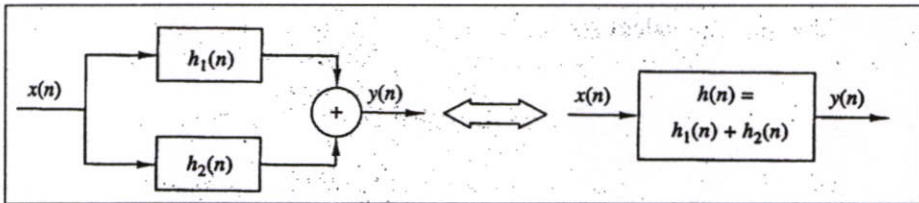


รูปที่ 2.23 คุณสมบัติการจัดกลุ่ม

- Distributive Law

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n) \quad (2.47)$$

เขียนเป็นไดอะแกรมได้ดังรูปที่ 2.24



รูปที่ 2.24 คุณสมบัติการกระจาย

2.2.5 การสร้างระบบไม่ต่อเนื่องทางเวลา (Implementation of Discrete Time System)[4,5]

- โครงสร้างของระบบเชิงเส้นที่แปรตามเวลา (Structures for the Realization of Linear Time-invariant Systems)

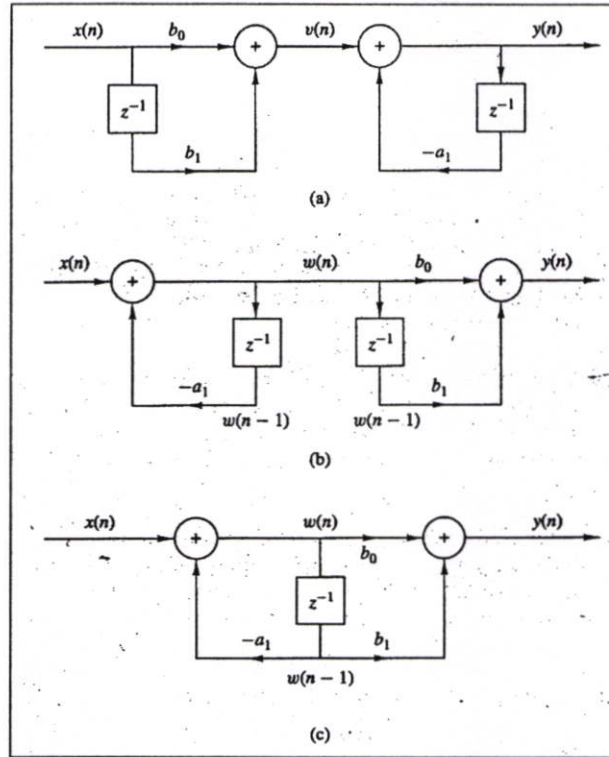
ระบบเชิงเส้นที่แปรตามเวลาจะมีสมการคังสมการที่ 2.48 และนำไปเขียนเป็นโครงสร้างได้ดังรูปที่ 2.34 (a)

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

$$v(n) = b_0 x(n) + b_1 x(n-1)$$

$$y(n) = -a_1 y(n-1) + v(n) \quad (2.48)$$

จากรูปที่ 2.25 (a) พบว่าระบบได้แยกออกเป็น 2 ระบบ โดย $v(n)$ เป็นจุดเชื่อมต่อ และจากคุณสมบัติการจับกลุ่ม ทำการสลับที่ระบบในรูปที่ 2.25 (a) ให้เป็นดังรูปที่ 2.25(b) เมื่อสลับที่แล้วพบว่าระบบทั้ง 2 ระบบสามารถใช้ตัวหน่วงร่วมกันได้ ทำให้เขียนเป็นโครงสร้างใหม่ได้ดังรูปที่ 2.25(c) และเขียนเป็นสมการได้ดังสมการที่ 2.49



รูปที่ 2.25 โครงสร้างของระบบเชิงเส้นแปรตามเวลา

$$w(n) = -a_1 w(n-1) + x(n)$$

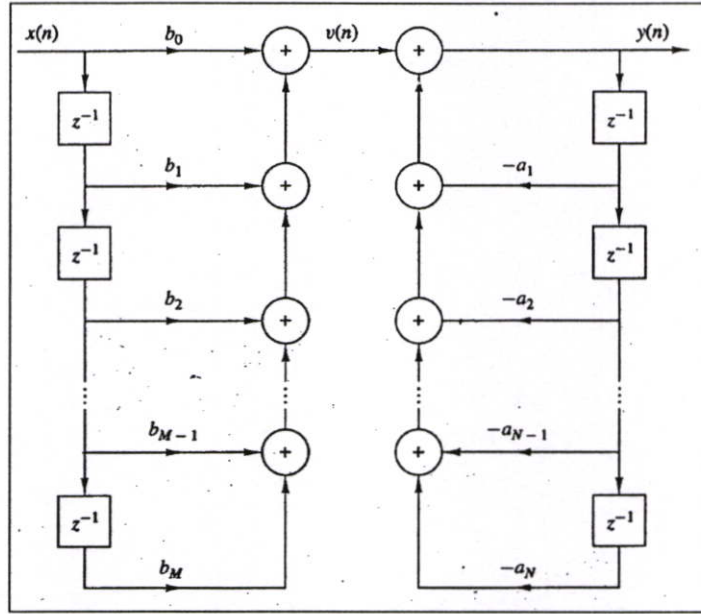
$$y(n) = b_0 w(n) + b_1 w(n-1) \quad (2.49)$$

เรียกโครงสร้างที่แสดงดังรูปที่ 2.25 (a) ว่าเป็นโครงสร้างแบบตรง I (Direct Form I) ส่วนรูป (c) นั้นเรียกว่าโครงสร้างแบบ (Direct Form II) และอาจเรียกระบบเชิงเส้นที่แปรตามเวลานี้ว่าระบบแบบ Recursive นอกจากนี้ระบบเชิงเส้นที่แปรตามเวลาสามารถเขียนในรูปสมการทั่วไปได้ดังสมการที่ 2.50

$$y(n) = -\sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (2.50)$$

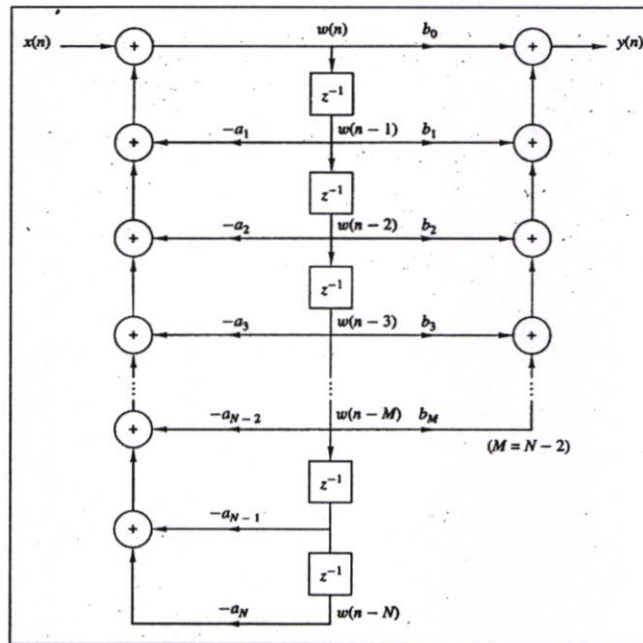
กำหนดให้ $v(n) = \sum_{k=0}^M b_k x(n-k)$ แทนค่าลงในสมการที่ 2.50 จะได้ดังสมการที่ 2.51 และนำไปเขียนโครงสร้างได้ดังรูปที่ 2.35

$$y(n) = -\sum_{k=1}^N a_k y(n-k) + v(n) \quad (2.51)$$



รูปที่ 2.26 โครงสร้างแบบตรง I ของระบบไม่ต่อเนื่องแบบ Recursive

และจากสมการที่ 2.50 และสมการที่ 2.51 สามารถแปลงโครงสร้างแบบตรง I ดังรูปที่ 2.26 เป็นโครงสร้างแบบตรง II ดังรูปที่ 2.27



รูปที่ 2.27 โครงสร้างแบบตรง II ของระบบไม่ต่อเนื่องแบบ Recursive

2.3 z-transform [1,2,3]

2.3.1 นิยามของการแปลงแซด[5,6]

การแปลงแซด (z-transform) ของสัญญาณที่ไม่ต่อเนื่องสามารถนิยามอยู่ในรูปของอนุกรมอนันต์ (Power Series) ได้ดังสมการที่ 2.48 เรียกสมการที่ 2.48 นี้ว่า การแปลงแซดแบบตรง (Direct z-transform)

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (2.52)$$

เมื่อ z เป็นตัวแปรเชิงซ้อน (Complex Variable)
 $x(n)$ เป็นลำดับสัญญาณ (Sequence Signal)

แต่สำหรับ Causal System การแปลงแซดของสัญญาณที่ไม่ต่อเนื่องสามารถเขียนได้ดังสมการที่ 2.53 เรียกสมการที่ 2.53 นี้ว่า One-side z-transform

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} \quad (2.53)$$

การแปลงแซดของสัญญาณแบบไม่ต่อเนื่องสามารถเขียนแทนได้อีกรูปแบบหนึ่งดังสมการที่ 2.54

$$X(z) = z\{x(n)\} \quad (2.54)$$

และจากสมการที่ 2.52 และ 2.53 จะเห็นได้ว่าผลการแปลงแซด ก็คืออนุกรมอนันต์ ดังนั้นจะมีค่า z ที่ทำให้ $X(z)$ หาคำตอบได้เรียกค่า z นี้ว่า Region of Convergence (ROC) โดยค่า z นี้จะมีค่าเป็นจำนวนเชิงซ้อน

ตารางที่ 2.1 z-transform ของสัญญาณแบบไม่ต่อเนื่อง

Entry Number	Discrete-time Sequence	z-Transform $X(z)$	Region of Convergence of $X(z)$
1	$\delta[n]$	1	All z
2	$u[n]$	$\frac{1}{1-z^{-1}}$	$ z > 1$
3	$-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$ z < 1$
4	$\delta[n-m]$	z^{-m}	All z except 0 (if $m > 0$) or ∞ (if $m < 0$)
5	$a^n u[n]$	$\frac{1}{1-az^{-1}}$	$ z > a $
6	$-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$ z < a $
7	$na^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z > a $
8	$-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z < a $
9	$[\cos \omega_0 n] u[n]$	$\frac{1 - [\cos \omega_0] z^{-1}}{1 - [2 \cos \omega_0] z^{-1} + z^{-2}}$	$ z > 1$
10	$[\sin \omega_0 n] u[n]$	$\frac{[\sin \omega_0] z^{-1}}{1 - [2 \cos \omega_0] z^{-1} + z^{-2}}$	$ z > 1$
11	$[r^n \cos \omega_0 n] u[n]$	$\frac{1 - [r \cos \omega_0] z^{-1}}{1 - [2r \cos \omega_0] z^{-1} + r^2 z^{-2}}$	$ z > r$
12	$[r^n \sin \omega_0 n] u[n]$	$\frac{[r \sin \omega_0] z^{-1}}{1 - [2r \cos \omega_0] z^{-1} + r^2 z^{-2}}$	$ z > r$
13	$\begin{cases} a^n, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases}$	$\frac{1 - a^N z^{-N}}{1 - az^{-1}}$	$ z > 0$

2.3.2 คุณสมบัติของ z-transform [5,6]

- คุณสมบัติความเป็นเชิงเส้น (Linearity Property)

แสดงได้ดังสมการที่ 2.55

$$Z[ax_1(n) + bx_2(n)] = aX_1(z) + bX_2(z) \quad (2.55)$$

- คุณสมบัติการเลื่อน (Shift Property)

กำหนดให้ $y(n) = x(n-m)$ $m \geq 0$ z-transform ของ $y(n) = Y(z)$ เขียนแทนได้ดังสมการที่ 2.56 และ 2.57

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n)z^{-n} \quad (2.56)$$

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n-m)z^{-n} \quad (2.57)$$

กำหนดให้ระบบที่กำลังพิจารณาเป็นระบบแบบ Causal ดังนั้นสมการที่ 2.57 สามารถเขียนได้ใหม่ดังสมการที่ 2.58 และ 2.59

$$Y(z) = \sum_{n=0}^{\infty} x(n-m)z^{-n} \quad (2.58)$$

$$Y(z) = x(-m) + x(1-m)z^{-1} + x(2-m)z^{-2} + \dots + x(0)z^{-m} + x(1)z^{-(m+1)} + \dots$$

$$Y(z) = x(0)z^{-m} + x(1)z^{-(m+1)} + x(2)z^{-(m+2)} + \dots + x(n)z^{-(m+n)} \quad (2.59)$$

จัดสมการที่ 2.59 ใหม่โดยดึงตัวร่วม z^{-m} ออกจะได้

$$Y(z) = z^{-m} [x(0) + x(1)z^{-1} + x(2)z^{-2} + \dots + x(n)z^{-n}] \quad (2.60)$$

$$Y(z) = z^{-m} \sum_{n=0}^{\infty} x(n)z^{-n}$$

$$Y(z) = z^{-m} X(z) \quad (2.61)$$

- คุณสมบัติการผสมผสาน (Convolution Property)

จากสมการ Convolution

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

$$y(n) = h(n) \otimes x(n)$$

ทำการแปลงเซต $x(n)$ หรือ $h(n)$ ให้เป็น $X(z)$ หรือ $H(z)$ จะได้การ Convolution ดังสมการที่ 2.62

$$Y(z) = H(z)X(z) \quad (2.62)$$

2.3.3 Rational z-transform

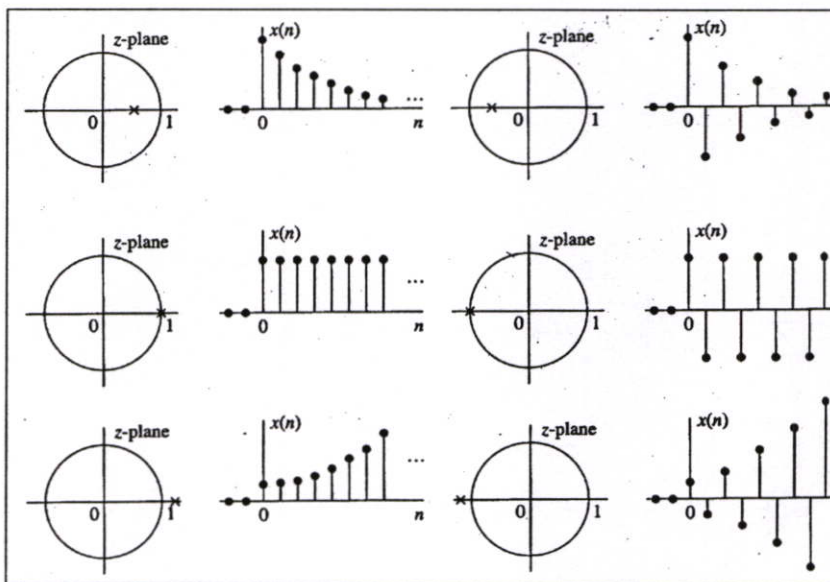
สัญญาณแบบไม่ต่อเนื่อง $x(n)$ หรือระบบแบบไม่ต่อเนื่อง $y(n)$ สามารถเขียนให้อยู่ในรูป $X(z)$ หรือ $Y(z)$ ได้โดยการแปลงเซต ค่า $X(z)$ หรือ $Y(z)$ นี้เมื่อจัดให้อยู่ในรูปเศษส่วน (Rational) สามารถบอกรายละเอียดคุณลักษณะของสัญญาณและระบบได้

- Pole and Zero

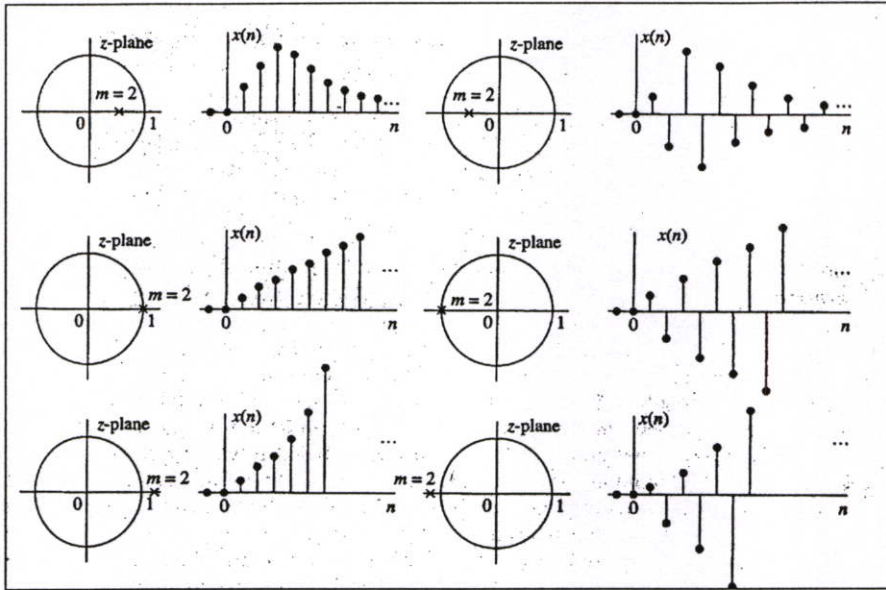
เป็นการนำค่า $X(z)$ หรือ $Y(z)$ ไปแก้สมการหาค่า z และนำคำตอบของ z ไป Plot ลงในระนาบ z เชิงซ้อน (Complex z plane) ค่า z ที่เป็นคำตอบของเศษเรียกว่า zero แทนด้วยสัญลักษณ์ \circ ค่า z ที่เป็นคำตอบของส่วนเรียกว่า Pole แทนด้วยสัญลักษณ์ \times

- ลักษณะสัญญาณกับการวางตำแหน่งของ Pole

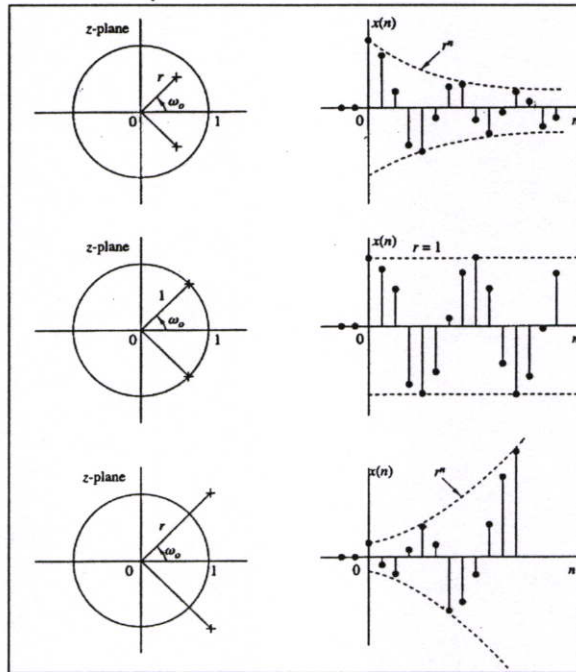
ตำแหน่งของ Pole ในระนาบ z สามารถบ่งบอกลักษณะของลำดับสัญญาณได้ซึ่งลักษณะของลำดับสัญญาณที่เกิดขึ้นสามารถคำนวณได้จากการทำ Inverse z -transform สามารถเขียนได้ดังรูปที่ 2.28



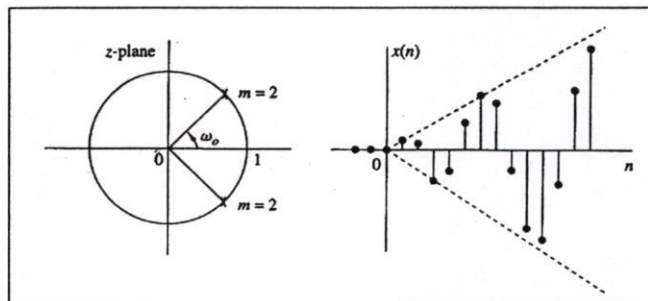
รูปที่ 2.28 (a) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ



รูปที่ 2.28 (b) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ



รูปที่ 2.28 (c) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ



รูปที่ 2.28 (d) ความสัมพันธ์ระหว่างตำแหน่งของ Pole กับลำดับสัญญาณ

2.3.4 ฟังก์ชันของระบบเชิงเส้นที่แปรตามเวลา

ในการหาผลตอบสนองทางเอาต์พุตของระบบที่แปรตามเวลาสามารถกระทำได้โดยใช้หลักการ Convolution ซึ่งการหาจะต้องรู้ค่าของ $x(n)$ และ $h(n)$ แต่จากคุณสมบัติของ z-transform ทำให้การหาผลตอบสนองทางเอาต์พุตหาได้จากสมการที่ 2.63

$$Y(z) = H(z)X(z) \quad (2.63)$$

เมื่อ $Y(z)$ คือ z-transform ของ Output Sequence $y(n)$
 $X(z)$ คือ z-transform ของ Input Sequence $x(n)$
 $H(z)$ คือ z-transform ของ Unit Sample Response $h(n)$

ดังนั้นถ้ารู้ค่า $x(n)$ และ $h(n)$ ก็สามารถหาค่า $X(z)$ และ $H(z)$ ได้โดยการแปลง z และเมื่อนำมาคูณกันดังสมการที่ 2.63 จะได้ $Y(z)$ และสามารถหาค่า $y(n)$ ได้โดยการแปลง z ผกผัน $Y(z)$ และในทางกลับกันถ้ารู้ค่า $x(n)$ และ $y(n)$ ก็สามารถหาค่า $H(z)$ ได้ดังสมการที่ 2.64

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.64)$$

เพื่อความชัดเจนจะเห็นได้ว่าคุณสมบัติของระบบนั้นอธิบายได้ 2 ลักษณะคือ $h(n)$ และ $H(z)$ โดย $h(n)$ นั้นจะอธิบายถึงคุณสมบัติของระบบใน Time-domain ส่วน $H(z)$ จะอธิบายถึงคุณสมบัติใน z-domain ซึ่งแท้ที่จริงแล้ว $H(z)$ และ $h(n)$ นั้นเหมือนกันเพียงแต่อยู่กันคนละ Domain และโดยทั่วไปจะเรียก $H(z)$ ในสมการที่ 2.64 ว่า ฟังก์ชันระบบ (System Function) หรือ ฟังก์ชันถ่ายโอน (Transfer Function)

สำหรับระบบที่เป็นแบบ Linear Constant-coefficient Difference Equation ดังสมการที่ 2.61 สามารถใช้คุณสมบัติของ z-transform เขียนเป็นฟังก์ชันถ่ายโอน $H(z)$ ได้ดังสมการที่ 2.66

$$y(n) = -\sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (2.65)$$

$$Y(z) = -\sum_{k=1}^N a_k Y(z)z^{-k} + \sum_{k=0}^M b_k X(z)z^{-k}$$

$$Y(z) \left(1 + \sum_{k=1}^N a_k z^{-k} \right) = X(z) \left(\sum_{k=0}^M b_k z^{-k} \right)$$

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

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.66)$$

$$y(n) = \sum_{k=0}^M h(k)x(n-k) \quad (2.67)$$

$$Y(z) = \sum_{k=0}^M h(k)X(z)z^{-k}$$

$$Y(z) = X(z) \left(\sum_{k=0}^M h(k)z^{-k} \right)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^M h(k)z^{-k} \quad (2.68)$$

2.3.5 การแปลง z ผกผัน (Inversion of z -transform)

เป็นการแปลง $X(z)$ กลับเป็น $x(n)$ ทำได้ 3 วิธีดังต่อไปนี้

- วิธีอนุกรมอนันต์ (Power Series Method)
- วิธีแยกเศษส่วนย่อย (Partial-Fraction Expansion Method)
- Residue Method

2.3.5.1 การแปลง z ผกผันด้วยวิธีอนุกรมอนันต์

กำหนดให้ $X(z)$ เป็น z -transform ของ Causal Sequence ดังสมการที่ 2.69 ซึ่งสมการนี้สามารถจัดให้เป็นอนุกรมอนันต์ในรูป z หรือ z^{-1} โดยใช้การหารยาว (Long Division)

$$X(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}} \quad (2.69)$$

เมื่อทำการหารยาวสมการที่ 2.69 จะได้ค่า $X(z)$ ดังสมการที่ 2.70

$$X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots \quad (2.70)$$

2.3.5.2 การแปลง z ผกผันด้วยวิธีแยกเศษส่วนย่อย

ทำโดยจัดสมการ $X(z)$ ให้อยู่ในรูปเศษส่วนย่อยจากนั้นนำสมการที่จัดรูปได้แปลงให้เป็น $x(n)$ โดยใช้ตาราง z -transform พิจารณา $X(z)$ ดังสมการที่ 2.69 การจัดรูปสมการจะเป็นไปตามตำแหน่งของ Pole ในระนาบ z ซึ่งแบ่งออกได้ดังต่อไปนี้

- กรณี Pole มีตำแหน่งไม่ซ้ำกัน สมการ $X(z)$ สมการที่ 2.69 สามารถจัดรูปได้ดังสมการที่ 2.71

$$X(z) = B_0 + \frac{C_1}{1-p_1z^{-1}} + \frac{C_2}{1-p_2z^{-1}} + \dots + \frac{C_3}{1-p_3z^{-1}} \quad (2.71)$$

$$X(z) = B_0 + \frac{C_1z}{z-p_1} + \frac{C_2z}{z-p_2} + \dots + \frac{C_Mz}{z-p_M}$$

$$X(z) = B_0 + \sum_{k=1}^M \frac{C_kz}{z-p_k} \quad (2.72)$$

- กรณี Pole อยู่มากกว่าหนึ่งตัวอยู่ตำแหน่งซ้ำกัน สมการ $X(z)$ สมการที่ 2.69 สามารถจัดรูปได้ดังสมการที่ 2.73

$$X(z) = \sum_{i=1}^m \frac{D_i}{(z-p_k)^i} \quad (2.73)$$

และค่า D_i หาได้จากสมการที่ 2.74

$$D_i = \frac{1}{(m-i)!} \frac{d^{m-i}}{dz^{m-i}} [(z-p_k)^m X(z)]_{z=p_k} \quad (2.74)$$

แต่ถ้าระบบหรือลำดับสัญญาณมี Pole มากกว่าหนึ่งตัวและตำแหน่งของ Pole อยู่ในตำแหน่งที่ซ้ำกันและไม่ซ้ำกันตัวอย่างเช่นระบบมี Pole ทั้งหมด 3 ตัว มี Pole 2 ตัวอยู่ตำแหน่งเดียวกันส่วนอีก 1 ตัวอยู่คนละตำแหน่งการจัดสมการในรูปเศษส่วนย่อยก็จะใช้สมการที่ 2.72 และ 2.73 รวมกัน

2.3.5.3 การแปลง z ผกผันโดยวิธี Residue

การแปลง z ผกผันด้วยวิธีนี้ค่า $x(n)$ จะถูกแทนด้วยการอินทิเกรตกรบรอบ (Contour Integral) ดังสมการที่ 2.85

$$x(n) = \frac{1}{2\pi j} \oint_C z^{n-1} X(z) dz \quad (2.75)$$

เมื่อ C คือเส้นทางการอินทิเกรตกรบรอบ Pole ของ $X(z)$ ที่อยู่ในรูปเศษส่วนและ โพลีโนเมียล (Rational and Polynomials) การอินทิเกรตกรบรอบดังสมการที่ 2.75 คำตอบจะถูกแทนอยู่ในรูปตัวแปรเชิงซ้อนด้วยทฤษฎีของ Cauchy's Residue [9] ดังนั้นอาจกล่าวได้ว่า $x(n)$ ก็คือผลรวมของการทำ Residue เฉพาะ Pole ในบริเวณ C ของฟังก์ชัน z^{n-1} ซึ่ง $X(z)$ เขียนได้ดังสมการที่ 2.76

$$\text{Res}[F(z), p_k] = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z-p_k)F(z)]_{z=p_k} \quad (2.76)$$

เมื่อ $F(z) = z^{n-1} X(z)$

2.4 ผลตอบสนองความถี่ (Frequency Response) [1,2,3]

พิจารณาฟังก์ชันถ่ายโอน $H(z)$ ดังสมการที่ 2.73

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots} \quad (2.77)$$

ในการหาผลตอบสนองทางความถี่ของระบบแบบไม่ต่อเนื่องนั้นจะแทน $z = e^{j\omega}$ ลงใน $H(z)$ ดัง สมการที่ 2.74

$$H(\omega) = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-2j\omega} + \dots}{1 + a_1 e^{-j\omega} + a_2 e^{-2j\omega} + \dots}$$

$$H(\omega) = H_R(\omega) + H_I(\omega) \quad (2.78)$$

จะเห็นได้ว่าสมการที่ 2.78 จะประกอบด้วยค่าที่เป็นจำนวนจริงและค่าที่เป็นจำนวนเชิงซ้อนซึ่งในการหาผลตอบสนองทางความถี่จะแบ่งการคำนวณออกเป็น 2 ลักษณะคือ

2.4.1 ผลตอบสนองทางขนาด (Magnitude Response)

คำนวณได้โดยการหาค่าสัมบูรณ์ของสมการที่ 2.78 เขียนได้ดังสมการที่ 2.79

$$|H(\omega)| = \sqrt{H_R^2(\omega) + H_I^2(\omega)} \quad (2.79)$$

2.4.2 ผลตอบสนองทางเฟส (Phase Response)

คำนวณได้จากสมการที่ 2.76 ซึ่งผลตอบสนองทางเฟสจะถูกแทนด้วย $\Theta(\omega)$ หรือ $\angle H(\omega)$

$$\Theta(\omega) = \angle H(\omega) = \tan^{-1} \frac{H_I(\omega)}{H_R(\omega)} \quad (2.80)$$

เช่นกรณีที่ 2.2 จงหาผลตอบสนองทางความถี่ของระบบไม่ต่อเนื่องดังสมการที่ 2.91

$$H(z) = \frac{1+z}{z} \quad (2.81)$$

วิธีการ กำหนดให้ $z = e^{j\omega}$

$$H(\omega) = \frac{1+e^{j\omega}}{e^{j\omega}} = \frac{1}{e^{j\omega}} + 1$$

$$H(\omega) = 1 + e^{-j\omega} \quad (2.82)$$

และจากสมการ $re^{j\omega} = r \cos(\omega) + jr \sin(\omega)$ ดังนั้นสมการที่ 2.82 สามารถเขียนใหม่ได้ดังสมการที่ 2.83

$$H(\omega) = 1 + [\cos(-\omega) + j \sin(-\omega)]$$

$$H(\omega) = 1 + \cos(\omega) - j \sin(\omega) \quad (2.83)$$

จากสมการที่ 2.83 จะพบว่าส่วนที่เป็นจำนวนจริงคือ $H_R(\omega) = 1 + \cos(\omega)$ และส่วนที่เป็นจำนวนเชิงซ้อนคือ $-\sin(\omega)$ ดังนั้นผลตอบสนองทางขนาดจะมีค่าดังสมการที่ 2.84

$$|H(\omega)| = \sqrt{[1 + \cos(\omega)]^2 + [-\sin(\omega)]^2}$$

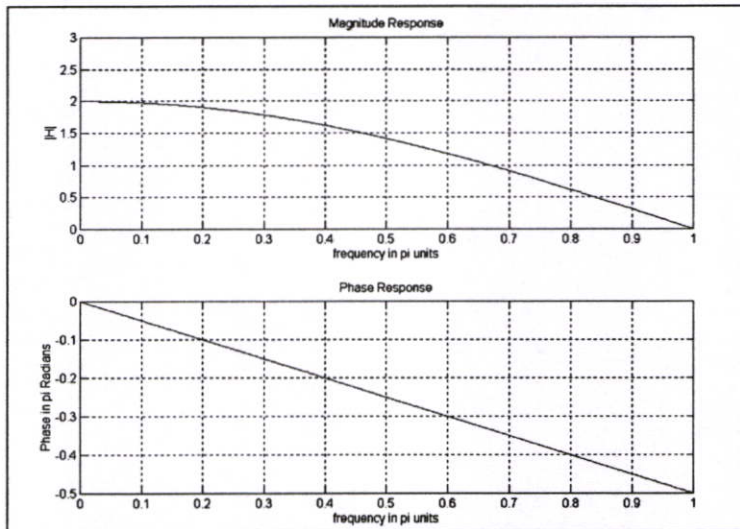
$$|H(\omega)| = \sqrt{1 + 2\cos(\omega) + \cos^2(\omega) + \sin^2(\omega)}$$

$$|H(\omega)| = \sqrt{2 + 2\cos(\omega)} \quad (2.84)$$

และผลตอบสนองทางเฟสเขียนได้จากสมการที่ 2.80 โดยนำค่าจำนวนจริงและจำนวนเชิงซ้อนจากสมการที่ 2.83 แสดงได้ดังสมการที่ 2.85

$$\Theta(\omega) = \angle H(\omega) = \tan^{-1} \frac{-\sin(\omega)}{1 + \cos(\omega)} \quad (2.85)$$

และจากสมการที่ 2.86 และ 2.85 เมื่อเปลี่ยนแปลงค่า ω ในช่วง $0 - \pi$ ทำให้เขียนกราฟผลตอบสนองความถี่ได้ดังรูปที่ 2.29



รูปที่ 2.29 ผลตอบสนองความถี่ของระบบ $H(z) = \frac{z+1}{z}$

เช่นกรณีที่ 2.3 จงหาผลตอบสนองความถี่จากสมการผลต่าง

$$y(n) = 0.9y(n-1) + x(n) \quad (2.86)$$

วิธีการ สมการที่ 2.86 สามารถแปลงให้เป็น $H(z)$ ได้ดังสมการที่ 2.87

$$Y(z) = 0.9Y(z)z^{-1} + X(z)$$

$$Y(z) - 0.9Y(z)z^{-1} = X(z)$$

$$Y(z)[1 - 0.9z^{-1}] = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 0.9z^{-1}} \quad (2.87)$$

แทนค่า $z = e^{j\omega}$ ลงในสมการที่ 2.87

$$H(\omega) = \frac{1}{1 - 0.9e^{-j\omega}}$$

พิจารณาเฉพาะส่วน

$$1 - 0.9e^{-j\omega} = (1 - 0.9\cos\omega) + j0.9\sin\omega$$

มีค่าสัมบูรณ์ดังสมการที่ 2.88 และมีมุมดังสมการที่ 2.89

$$|1 - 0.9e^{-j\omega}| = \sqrt{(1 - 0.9\cos\omega)^2 + (0.9\sin\omega)^2}$$

$$|1 - 0.9e^{-j\omega}| = \sqrt{1.81 - 1.8\cos\omega} \quad (2.88)$$

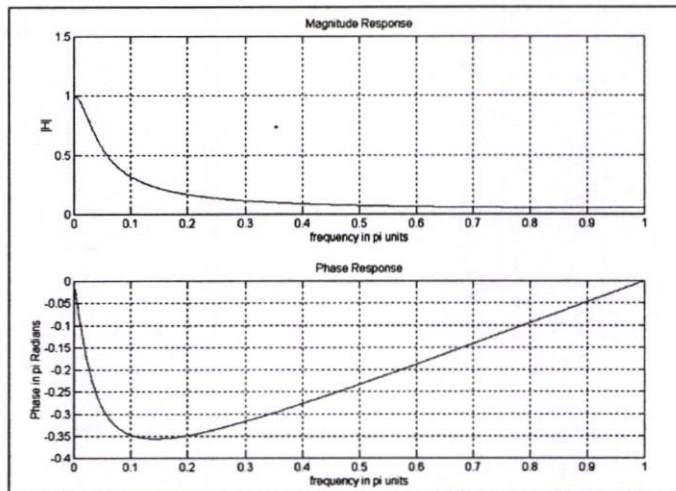
$$\angle(1 - 0.9e^{-j\omega}) = \tan^{-1} \frac{0.9\sin\omega}{1 - 0.9\cos\omega} \quad (2.89)$$

ดังนั้นสมการที่ 2.87 จะมีค่า Magnitude Response ดังสมการที่ 2.90 และมี Phase Response ดังสมการที่ 2.91

$$|H(\omega)| = \frac{1}{\sqrt{1.81 - 1.8 \cos \omega}} \quad (2.90)$$

$$\Theta(\omega) = -\tan^{-1} \frac{0.9 \sin \omega}{1.81 - 1.8 \cos \omega} \quad (2.91)$$

นำไปเขียนเป็นกราฟผลตอบสนองความถี่ได้ดังรูปที่ 2.30



รูปที่ 2.30 ผลตอบสนองความถี่ของระบบ $y(n) = 0.9y(n-1) + x(n)$

2.5 ตัวกรองดิจิทัลแบบ FIR [1,2,3,4]

ตัวกรองดิจิทัลแบ่งออกได้เป็น 2 ชนิด คือ

- ตัวกรองดิจิทัลแบบเอฟไออาร์ (FIR : Finite Impulse Response Filter)
- ตัวกรองดิจิทัลแบบไอไออาร์ (IIR : Infinite Impulse Response Filter)

ในวิทยานิพนธ์เล่มนี้กล่าวถึงเฉพาะตัวกรองดิจิทัลแบบ FIR ซึ่งตัวกรองดิจิทัลแบบ FIR มีฟังก์ชันการถ่ายโอน (Transfer Function) มีสมการคุณสมบัติดังสมการที่ 2.92

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (2.92)$$

เขียนให้อยู่ในแซด โดเมน (z-domain) ได้ดังสมการที่ 2.93

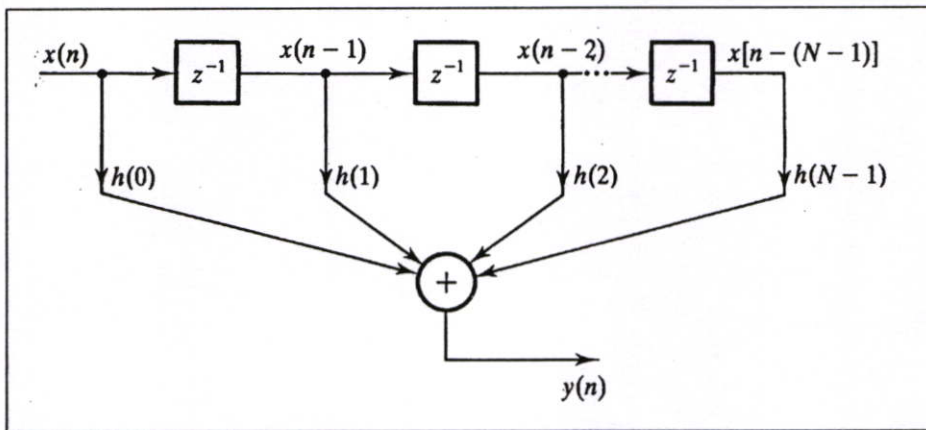
$$y(z) = \sum_{k=0}^{N-1} h(k)x(z)z^{-k} \quad (2.93)$$

หรือเขียนให้อยู่ในรูปของฟังก์ชันถ่ายโอนจะได้

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

$$Y(z) = h(0)x(z) + h(1)x(z)z^{-1} + h(2)x(z)z^{-2} + \dots + h(k)x(z)z^{-k} \quad (2.94)$$

จากสมการที่ 2.94 สามารถนำไปเขียนเป็นโครงสร้างได้ดังรูปที่ 2.31



รูปที่ 2.31 แสดงโครงสร้างของตัวกรองเชิงเลข FIR

ตัวกรองเชิงเลข FIR จัดเป็นตัวกรองนอนรีเคอร์ซีฟ [7] (Non Recursive) เนื่องจากไม่มีการป้อนกลับจากทางด้าน Output และนอกจากนี้ตัวกรองเชิงเลข FIR จะมีคุณสมบัติอื่นอีกคือ

1. สามารถสร้างได้ง่ายเมื่อเทียบกับตัวกรองแบบ IIR
2. มีคุณสมบัติการตอบสนองทางเฟสเป็นแบบเชิงเส้น (Linear phase)
3. สัมประสิทธิ์ที่เกิดจากการคำนวณจะมีค่าไม่เกินหนึ่งทำให้สามารถสร้างและทำงานได้ดิบบน ตัวประมวลผลแบบจุดทศนิยมคงที่ (Fix-point)
4. ตัวกรองเชิงเลข FIR จะเสถียรภาพ (Stable) แน่นอน เนื่องจากว่ามีโพลอยู่ที่จุดกำเนิด (Origin) บนระนาบแซด

พิจารณาสมการ

$$y(n) = x(n - k) \quad (2.95)$$

เมื่อทำการแปลงฟูเรียร์ในสมการที่ 2.95 จะได้

$$Y(j\omega) = e^{-j\omega kT} X(j\omega) \quad (2.96)$$

ย้ายข้างจะได้

$$\frac{Y(j\omega)}{X(j\omega)} = H(j\omega) = e^{-j\omega kT} \quad (2.97)$$

จากสมการจะมีค่าขนาด (Magnitude) เท่ากับ 1 และมีเฟสดังสมการที่ 2.98

$$\theta(\omega) = -\omega kT \quad (2.98)$$

และจากสมการที่ 2.98 ซึ่งเป็นสมการของเฟสสามารถนำไปหาค่าหน่วงกลุ่ม (Group Delay) ได้โดยทำการหาอนุพันธ์สมการที่ 2.98 เทียบกับ ω จะได้สมการของค่าหน่วงกลุ่มดังสมการที่ 2.99

$$d\theta = \frac{d(-\omega kT)}{d\omega} \quad (2.99)$$

$$\Theta = -kT \quad (2.100)$$

จากสมการที่ 2.100 จะเห็นได้ว่าตัวกรองเชิงเลข FIR มีผลตอบสนองทางเฟสเป็นแบบเชิงเส้น ดังนั้นตัวกรองเชิงเลข FIR จึงถูกนำไปใช้งานอย่างกว้างขวาง ดังตัวอย่างเช่น การประมวลผลสัญญาณเสียง การประมวลผลสัญญาณภาพ และการประมวลผลสัญญาณทางการแพทย์ เป็นต้น และถ้ากำหนดสมการผลตอบสนองทางเฟสใหม่โดยกำหนดให้

$$kT = \alpha \quad (2.101)$$

จะได้

$$\theta(\omega) = -\alpha\omega \quad (2.102)$$

หรือถ้าค่าของผลตอบสนองทางเฟสเท่ากับ

$$\theta(\omega) = \beta - \alpha\omega \quad (2.103)$$

β เป็นค่าคงที่

ถ้าตัวกรองมีทั้งผลการตอบสนองทางเฟสและค่าหน่วยกลุ่มเป็นแบบเชิงเส้นตามสมการที่ 2.102 จะให้ค่าผลตอบสนองอิมพัลส์ของตัวกรองเป็นแบบสมมาตรบวก (Positive Symmetry) ดังสมการที่ 2.104(a) และ 2.104(b) ซึ่งผลการตอบสนองทางเฟสจะเป็นฟังก์ชันของความยาวตัวกรอง (Filter Length)

$$h(n) = h(N - n - 1) \text{ ที่ } n = 0, 1, \dots, (N-1)/2 \text{ และ } n \text{ เป็นคี่} \quad (2.104 \text{ (a)})$$

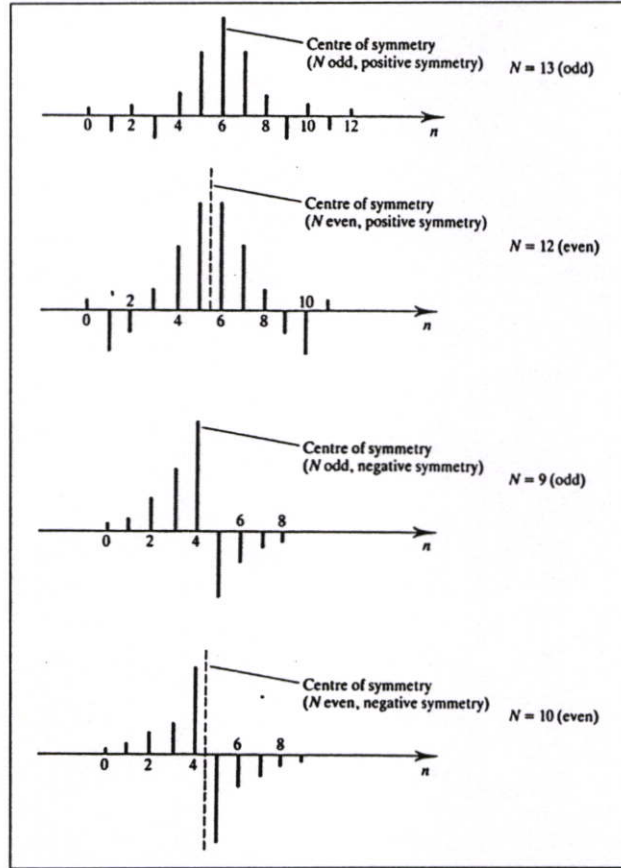
$$\alpha = \frac{(N-1)}{2} \text{ ที่ } n = 0, 1, \dots, (\frac{N}{2}-1) \text{ และ } n \text{ เป็นคู่} \quad (2.104 \text{ (b)})$$

และถ้าฟิลเตอร์มีผลการตอบสนองทางเฟสดังสมการที่ 2.103 จะได้ผลตอบสนองอิมพัลส์ของตัวกรองเป็นแบบสมมาตรลบ (Negative Symmetry) ดังสมการที่ 2.105(a) และ 2.105(b)

$$h(n) = -h(N - n - 1) \quad (2.105(a))$$

$$\alpha = \frac{(N-1)}{2} \quad (2.105(b))$$

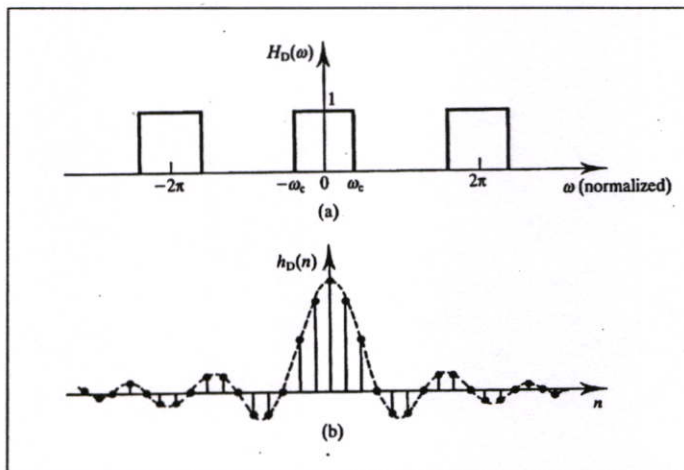
โดย N ในสมการที่ 2.104 และสมการที่ 2.105 เป็นจำนวนลำดับของตัวกรองและ $0 \leq n \leq N-1$ ดังนั้นถ้า $N = 7$ จะได้ $h(0) = h(6), h(1) = h(5), h(2) = h(4)$ และถ้า $N = 8$ จะได้ $h(0) = h(7), h(1) = h(6), h(2) = h(5), h(3) = h(4)$ และจากสมการที่ 2.105 และสมการที่ 2.106 สามารถนำมาเขียนเป็นกราฟได้ดังรูปที่ 2.32



รูปที่ 2.32 แสดงผลตอบสนองอิมพัลส์ของเฟสเชิงเส้นของตัวกรอง 4 ชนิด

2.5.1 การออกแบบตัวกรองดิจิทัลแบบ FIR [4,5]

จะใช้วิธีฟูรีเยร์ซีรีส์ (Fourier Series) มาทำการออกแบบพิจารณารูปที่ 2.33



รูปที่ 2.33 แสดงผลตอบสนองความถี่และผลตอบสนองอิมพัลส์ในทางอุดมคติ

(a) แสดงการตอบสนองความถี่ของตัวกรองแบบ Low-pass ในทางอุดมคติ

(b) แสดงผลตอบสนองอิมพัลส์ของตัวกรองแบบ Low-pass ในทางอุดมคติ

ตัวกรองที่มีผลตอบสนองทางความถี่ $H_D(\omega)$ สามารถหาค่าผลตอบสนองอิมพัลส์ $h_D(n)$ ได้จากความสัมพันธ์ของการแปลงกลับของฟูเรียร์ดังสมการที่ 2.106

$$h_D(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_D(\omega) e^{j\omega n} d\omega \quad (2.106)$$

พิจารณาจากรูปที่ 2.32 ซึ่งเป็นกราฟแสดงการตอบสนองความถี่ของตัวกรองแบบ Low-pass ซึ่งมีความถี่คัท-ออฟ คือ ω_c โดย ω_c นี้จะเป็นความถี่นอร์มอลไลซ์ (Normalized Frequency) ซึ่งมีค่าดังสมการที่ 2.107

$$f = \frac{f_c}{f_s} \quad (2.107)$$

โดย f คือ ความถี่คัท-ออฟนอร์มอลไลซ์ (Normalized Cut-off Frequency)
 f_c คือ ความถี่คัท-ออฟ (Cut-off Frequency (Hz))
 f_s คือ ความถี่สุ่ม (Sampling Frequency (Hz))

จากกราฟจะเห็นได้ว่าค่า $H_D(\omega)$ จะมีค่าเท่ากับ 1 ในช่วง ตั้งแต่ $-\omega_c$ จนถึง $+\omega_c$ ดังนั้นค่าของผลตอบสนองอิมพัลส์ของรูปที่ 2.33(a) จะหาได้จากสมการที่ 2.108

$$h_D(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} 1 \times e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega \quad (2.108)$$

$$h_D(n) = \frac{2 \times \sin(2\pi f n)}{2\pi f n} \quad (2.109)$$

$$h_D(n) = \frac{\sin(2\pi f n)}{n\pi} \quad (2.110)$$

เนื่องจากค่าผลตอบสนองอิมพัลส์นั้นมีความสมมาตรกันดังนั้นในการหาค่าจะทำการหาเพียงครึ่งหนึ่งก็พอดังตัวอย่าง ถ้าต้องการหาผลตอบสนองอิมพัลส์ $n = 53$ จะทำการหาค่า n ตั้งแต่ 0 จนถึง 26 ส่วนค่า -1 จนถึง -26 นั้นไม่จำเป็นต้องหาเนื่องจากความเป็นสมมาตรนั่นเอง

$$h(5) = h(47) = -0.00842$$

$$h(6) = h(46) = 0.01125$$

$$h(7) = h(45) = 0.01393$$

$$h(8) = h(44) = -0.00677$$

$$h(9) = h(43) = -0.01836$$

$$h(10) = h(42) = 0.0000$$

$$h(11) = h(41) = 0.02081$$

$$h(12) = h(40) = 0.00870$$

$$h(13) = h(39) = -0.02036$$

$$h(14) = h(38) = -0.01876$$

$$h(15) = h(37) = 0.01608$$

$$h(16) = h(36) = 0.02941$$

$$h(17) = h(35) = -0.00690$$

$$h(18) = h(34) = -0.03979$$

$$h(19) = h(33) = -0.00887$$

$$h(20) = h(32) = 0.04901$$

$$h(21) = h(31) = 0.03537$$

$$h(22) = h(30) = -0.05627$$

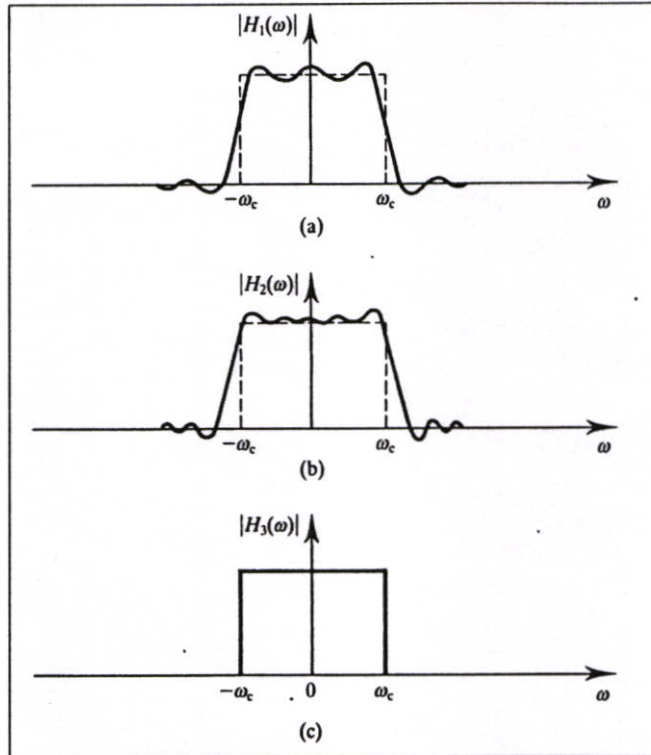
$$h(23) = h(29) = -0.08822$$

$$h(24) = h(28) = 0.06091$$

$$h(25) = h(27) = 0.31219$$

$$h(26) = h(26) = 0.43750$$

ตัวกรองเชิงเลขแบบ FIR จะมีผลตอบสนองทางความถี่เป็นแบบอุดมคติก็ต่อเมื่อมีการใช้ตัวสัมประสิทธิ์เป็นอนันต์ แต่ในทางปฏิบัติผลการตอบสนองทางความถี่ของตัวกรองเชิงเลขแบบ FIR จะมีการกระเพื่อม (Ripple) เกิดขึ้น ถ้าจำนวนสัมประสิทธิ์มีค่าน้อยก็จะทำให้เกิดกระเพื่อมมาก แต่ถ้าสัมประสิทธิ์มีจำนวนมากขึ้นก็จะทำให้ค่าการกระเพื่อมน้อยลงซึ่งจะมีผลใกล้เคียงกับผลการตอบสนองความถี่ทางอุดมคติมากขึ้น ดังแสดงในรูปที่ 2.34



รูปที่ 2.34 แสดงผลการตอบสนองความถี่

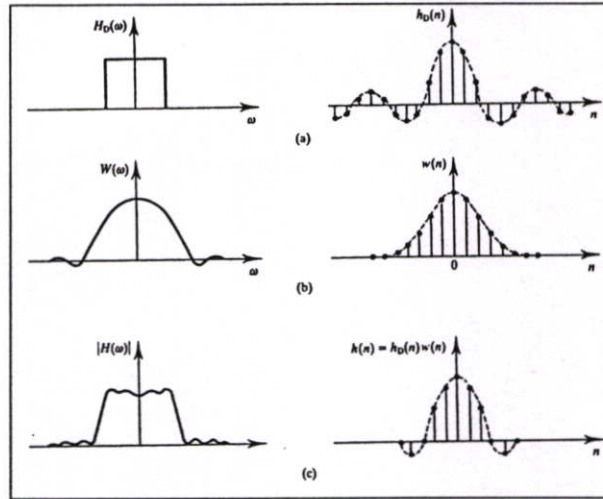
- (a) สัมประสิทธิ์ 13 ตัว
- (b) สัมประสิทธิ์ 25 ตัว
- (c) สัมประสิทธิ์เป็นอนันต์

2.5.2 การออกแบบตัวกรองดิจิทัลแบบ FIR ด้วยฟังก์ชันวินโดว์ [2,4]

เพื่อให้ผลการตอบสนองความถี่มีค่าใกล้เคียงกับผลตอบสนองความถี่ในทางอุดมคติ จึงได้มีการค้นคว้าวิธีการแก้ปัญหาโดยการนำสัมประสิทธิ์ $h_D(n)$ มาคูณกับสัมประสิทธิ์อีกตัวหนึ่ง สัมประสิทธิ์ตัวนี้ก็คือ $w(n)$ หรือ Window Function ผลที่ได้จากการคูณจะเป็นสัมประสิทธิ์ตัวใหม่หรือ $h(n)$ ซึ่งการหาสัมประสิทธิ์ $h(n)$ นี้เรียกว่า การหาโดยใช้ฟังก์ชันของวินโดว์ (Window) สมการของ $h(n)$ แสดงได้ดังสมการที่ 2.111

$$h(n) = h_D(n) \cdot w(n) \quad (2.111)$$

ค่าของ $w(n)$ สามารถหาได้หลายวิธีขึ้นอยู่กับความต้องการของผู้ออกแบบซึ่งแต่ละวิธีจะให้ผลการตอบสนองทางขนาดและผลตอบสนองทาง Impulse ที่แตกต่างกันดังแสดงในรูปที่ 2.35



รูปที่ 2.35 แสดงผลการตอบสนองความถี่ด้วยฟังก์ชันวินโดว์

นอกจากนี้สมการที่ใช้หาค่า $w(n)$ ของแต่ละวิธียังแตกต่างกันดังต่อไปนี้

- เรคแทนกูลาร์วินโดว์ (Rectangular Window)

$$w(n) = 1 \quad (2.112)$$

- ฮานนิงวินโดว์ (Hanning Window)

$$w(n) = 0.5 + 0.5 \cos\left(\frac{2\pi n}{N}\right) \quad (2.113)$$

- แฮมมิงวินโดว์ (Hamming Window)

$$w(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right) \quad (2.114)$$

- แบลคแมนวินโดว์ (Blackman Window)

$$w(n) = 0.42 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right) \quad (2.115)$$

เช่นกรณีที่ 2.5 การออกแบบตัวกรองความถี่ต่ำผ่านแบบเฟอไออาร์โดยใช้ฟังก์ชันของวินโดว์แบบแฮมมิ่ง กำหนดให้ f_c มีค่า 1.75 kHz f_s มีค่า 8 kHz จำนวนลำดับเท่ากับ 53 ดังนั้นค่าสัมประสิทธิ์ $h_D(n)$ อยู่ในช่วง -26 จนถึง 26

วิธีการ

$$N = 53 \quad f = \frac{f_c}{f_s} = \frac{1.75 \text{ kHz}}{8 \text{ kHz}} = 0.21875$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

แฮมมิ่งวินโดว์มีค่า $\omega(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right)$

$$n = 0; \quad h_D(0) = 2 \times f = 2 \times 0.21875 = 0.4375$$

$$\omega(0) = 0.54 + 0.46 \cos(0) = 1$$

$$h(0) = h_D(0)\omega(0)$$

$$h(0) = 0.4375 \times 1$$

$$h(0) = 0.4375 = h(52)$$

$$n = 1; \quad h_D(1) = \frac{\sin(2\pi \times 0.21875)}{\pi} = 0.31219$$

$$\omega(1) = 0.54 + 0.46 \cos\left(\frac{2\pi}{53}\right) = 0.98713$$

$$h(1) = h_D(1) \cdot \omega(1) = 0.31219 \times 0.98713 = 0.31119$$

$$h(1) = 0.31119 = h(-1)$$

$$n = 2; \quad h_D(2) = \frac{\sin(2 \times 2\pi \times 0.21875)}{2\pi}$$

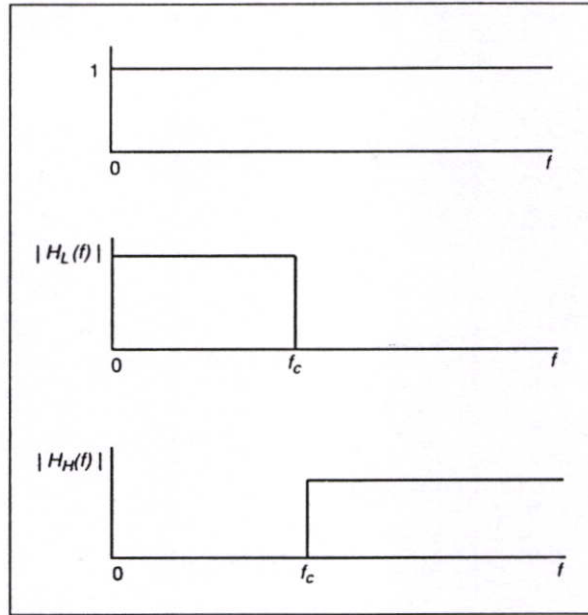
$$\begin{aligned}
h(6) &= h(46) = 2.3627 \times 10^{-3} \\
h(7) &= h(45) = 3.4846 \times 10^{-3} \\
h(8) &= h(44) = -1.9925 \times 10^{-3} \\
h(9) &= h(43) = -6.2837 \times 10^{-3} \\
h(10) &= h(42) = 4.5320 \times 10^{-9} \\
h(11) &= h(41) = 9.2669 \times 10^{-3} \\
h(12) &= h(40) = 4.3430 \times 10^{-3} \\
h(13) &= h(39) = -1.1271 \times 10^{-2} \\
h(14) &= h(38) = -1.1402 \times 10^{-2} \\
h(15) &= h(37) = 1.06307 \times 10^{-2} \\
h(16) &= h(36) = 2.0964 \times 10^{-2} \\
h(17) &= h(35) = -5.2583 \times 10^{-3} \\
h(18) &= h(34) = -3.2156 \times 10^{-2} \\
h(19) &= h(33) = -7.5449 \times 10^{-3} \\
h(20) &= h(32) = 4.3546 \times 10^{-2} \\
h(21) &= h(31) = 3.2593 \times 10^{-2} \\
h(22) &= h(30) = -5.3413 \times 10^{-2} \\
h(23) &= h(29) = -8.5682 \times 10^{-2} \\
h(24) &= h(28) = 6.0122 \times 10^{-2} \\
h(25) &= h(27) = 3.118 \times 10^{-1} \\
h(26) &= h(26) = 4.3750 \times 10^{-1}
\end{aligned}$$

2.5.3 การออกแบบตัวกรองความถี่สูงผ่าน ช่วงความถี่ผ่าน ช่วงความถี่หยุดแบบ FIR

[2,4]

ตัวกรองความถี่สูงผ่าน (High-pass Filter) ในการออกแบบ High-pass Filter จะทำโดยการนำผลการตอบสนองความถี่ของตัวกรองออลพาส (All-pass Filter) ลบด้วยผลตอบสนองความถี่ของตัวกรองเชิงเลข FIR ดังแสดงในรูปที่ 2.36 และในการหาค่าสัมประสิทธิ์ของตัวกรองแบบ High-pass สามารถหาได้จากสมการ

$$h_D(n)_{high} = \delta(n) - h_D(n)_{Low} \quad (2.116)$$



รูปที่ 2.36 แสดงกราฟตอบสนองความถี่ของตัวกรองแบบ All-pass Low-pass
High-pass

เช่นกรณีที่ 2.6 การออกแบบตัวกรองแบบ High-pass โดยใช้ฟังก์ชันของวินโดว์แบบเรคแทนกูลาร์ กำหนดให้ $f_c = 1.75\text{kHz}$ $f_s = 8\text{kHz}$ จำนวนลำดับเท่ากับ 53

$$f = \frac{f_c}{f_s} = \frac{1.75\text{kHz}}{8\text{kHz}} = 0.21875$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

$$n = 0; \quad h_D(n) = \delta(0) - 2 \times f = 1 - 2 \times 0.21875 = -0.5625$$

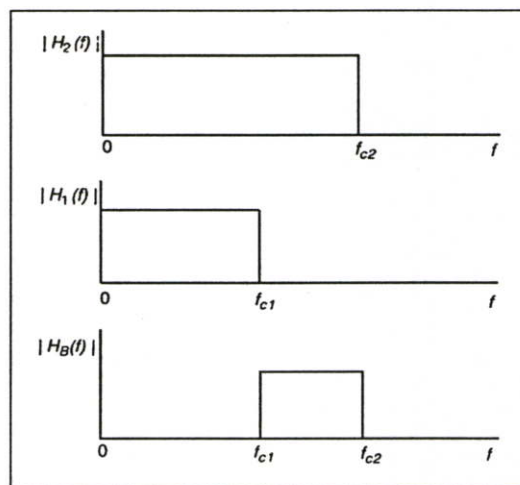
$$n = 1; \quad h_D(1) = \delta(1) - \frac{\sin(2\pi \times 0.21875)}{\pi} = 0 - 0.31219$$

$$h_D(1) = -0.31219 = h_D(-1)$$

$$n = 2; \quad h_D(2) = \delta(2) - \frac{\sin(2 \times 2\pi \times 0.21875)}{\pi} = -0.06012$$

$$\begin{aligned}
 h(19) &= h(33) = 0.00887 \\
 h(20) &= h(32) = -0.04901 \\
 h(21) &= h(31) = -0.03537 \\
 h(22) &= h(30) = 0.05627 \\
 h(23) &= h(29) = 0.08822 \\
 h(24) &= h(28) = -0.06091 \\
 h(25) &= h(27) = -0.31219 \\
 h(26) &= h(26) = 0.56250
 \end{aligned}$$

ตัวกรองช่วงความถี่ผ่าน (Band-pass Filter) ในการออกแบบตัวกรองช่วงความถี่ผ่านแบบ FIR จะทำโดยนำเอาผลการตอบสนองความถี่ของตัวกรองแบบ Low-pass 2 ความถี่มาลบกันดังแสดงในรูปที่ 3.4



รูปที่ 2.37 แสดงกราฟผลตอบสนองความถี่ของตัวกรองแบบ Low-pass 2 ความถี่ และกราฟผลตอบสนองความถี่ของตัวกรองแบบ Band-pass

และในการหาค่าสัมประสิทธิ์ของตัวกรองช่วงความถี่ผ่านสามารถหาได้จากสมการ

$$h_D(n)_{Band} = h_D(n)_{fc2} - h_D(n)_{fc1} \quad (2.117)$$

$$h_D(n)_{Band} = \frac{\sin(2\pi f_{c2}n)}{n\pi} - \frac{\sin(2\pi f_{c1}n)}{n\pi} \quad (2.118)$$

$$h(0) = h(10) = 0.0000$$

$$h(1) = h(9) = -0.12246$$

$$h(2) = h(8) = -0.16328$$

$$h(3) = h(7) = -0.05782$$

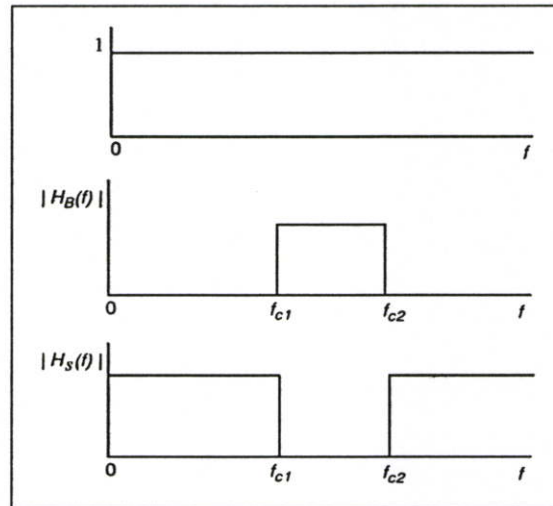
$$h(4) = h(6) = 0.11563$$

$$h(5) = h(5) = 0.2$$

ตัวกรองช่วงความถี่หยุด (Band-stop Filter) สามารถกระทำได้โดยนำผลตอบสนองความถี่ของตัวกรองออลพาสลบด้วยผลตอบสนองความถี่ของตัวกรองแบบ Low-pass ดังแสดงในรูปที่ 2.38

$$h_D(n) = \delta(n) - [h_D(n)_{f_{c2}} - h_D(n)_{f_{c1}}] \quad (2.119)$$

$$h_D(n) = \delta(n) - \left[\frac{\sin(n \times 2\pi f_2)}{n\pi} - \frac{\sin(n \times 2\pi f_1)}{n\pi} \right] \quad (2.120)$$



รูปที่ 2.38 แสดงผลตอบสนองของตัวกรองแบบ All-pass Band-pass และ Band-stop

เช่นกรณีที่ 2.8 การออกแบบตัวกรองเชิงเลข FIR แบบ Band-stop โดยใช้วินโดว์แบบฮานนิง กำหนดให้มีช่วง $f_{c1} = 1\text{kHz}$ $f_{c2} = 2\text{kHz}$ $f_s = 8\text{kHz}$ จำนวนลำดับเท่ากับ 11

$$f_1 = \frac{f_{c1}}{f_s} = \frac{1\text{kHz}}{8\text{kHz}} = 0.125$$

$$f_2 = \frac{f_{c_2}}{f_s} = \frac{2\text{kHz}}{10\text{kHz}} = 0.2$$

สัมประสิทธิ์มีค่า n อยู่ระหว่าง -5 ถึง 5 และ สมการ $\omega(n)$ ของฮามมิงวินโดว์ คือ

$$\omega(n)_{\text{hamming}} = 0.5 + 0.5 \cos\left(\frac{2\pi n}{N}\right)$$

$$n = 0;$$

$$h_D(0) = \delta(0) - (2f_{c_2} - 2f_{c_1})$$

$$h_D(0) = 1 - (2 \times 0.2 - 2 \times 0.1) = 0.8$$

$$\omega(0) = 0.5 + 0.5 \cos(0) = 1$$

$$h(0) = h_D(0)\omega(0) = 0.8$$

$$n = 1;$$

$$h_D(1) = \delta(1) - \left(\frac{\sin(2\pi \times 0.2)}{\pi} - \frac{\sin(2\pi \times 0.1)}{\pi} \right)$$

$$\omega(1) = 0.5 + 0.5 \cos\left(\frac{2\pi \times 1}{11}\right)$$

$$h(1) = h_D(1)\omega(1)$$

$$h(1) = -0.10789 = h(-1)$$

$$n = 2;$$

$$h_D(2) = \delta(2) - \left(\frac{\sin(2 \times 2\pi \times 0.2)}{2\pi} - \frac{\sin(2 \times 2\pi \times 0.1)}{2\pi} \right)$$

$$\omega(2) = 0.5 + 0.5 \cos\left(\frac{2\pi \times 2}{11}\right)$$

$$h(2) = h_D\omega(2)$$

เช่นกรณี 2.9 จากเช่นกรณี 2.7 ค่าสัมประสิทธิ์ที่หาค่าได้ถ้าตัวประมวลผลมีขนาด 16 บิตทำการแปลงเลขได้ดังนี้

วิธีการ

สัมประสิทธิ์ที่ได้จากการคำนวณ ในเช่นกรณี 2.7	$a_{fix} = round(a \times 2^{M-1})$	สัมประสิทธิ์ที่ใช้ในการ ประเมินผลในบอร์ด FPGA
0.0000	0	0
0.03061	1002	3EA
0.08164	2675	A73
0.04336	1420	58C
-0.10789	-3535	F231
0.8	26213	6665

2.7 FPGA (Field Programmable Gate Array) [8]

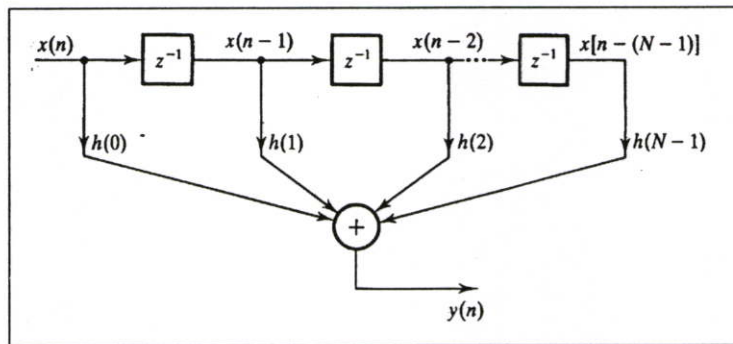
ในปัจจุบัน FPGA ได้มีการใช้งานกันอย่างกว้างขวางมากขึ้น โดย FPGA ได้ถูกนำมาแทนที่ Logic Gate ทั้งในระดับ SSI MSI หรือแม้แต่ในส่วนประมวลผลก็ตาม โดยจะเห็นได้จากอุปกรณ์ที่เป็นส่วนประกอบของระบบคอมพิวเตอร์ ทั้งนี้เนื่องจาก FPGA มีข้อดีอยู่หลายประการ ดังต่อไปนี้

- สามารถออกแบบวงจรดิจิทัลได้โดยมีจำนวนอุปกรณ์และขนาดของแผงวงจรลดลง เนื่องจากวงจรถูกออกแบบไว้ในชิพ
- วงจรที่ออกแบบสามารถทำงานที่ความถี่สูงกว่าไมโครคอนโทรลเลอร์ เพราะ FPGA มีลักษณะการทำงานเป็นแบบฮาร์ดแวร์
- สามารถแก้ไขการทำงานได้ง่าย เพราะเนื่องจากลักษณะการทำงานแบบฮาร์ดแวร์ จะถูกกำหนดและออกแบบโดยภาษาระดับสูง (High Level Design)

บทที่ 3

การออกแบบตัวกรองเชิงเลข FIR

การออกแบบการทดลองตัวกรองเชิงเลข FIR หลาย Output แบบเวลาจริง จะทำโดยการปรับปรุงโครงสร้างของตัวกรองเชิงเลข FIR จากโครงสร้างแบบตรงของตัวกรองเชิงเลข FIR แสดงดังรูปที่ 3.1 นั้น ได้ใช้วงจรรีจิสเตอร์แทนตัวหน่วง วงจรคูณ และวงจรวก ดังนั้นจึงทำการออกแบบโครงสร้างของตัวกรองเชิงเลข FIR ขึ้นใหม่โดยใช้ตัวแปรบางส่วนร่วมกันดังรูปที่ 3.2 ซึ่งจะทำให้เกิดผลตอบสนองทางความถี่ได้หลายรูปแบบในเวลาเดียวกัน



ภาพที่ 3.1 โครงสร้างแบบตรงของตัวกรองเชิงเลข FIR

3.1 การออกแบบโครงสร้างของตัวกรองเชิงเลข FIR หลาย Output แบบเวลาจริง

ในการออกแบบเริ่มจากการออกแบบตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ต่างกัน 2 ความถี่ดังสมการที่ 3.1 และ 3.2

$$y_{LP1}(n) = \sum_{k=0}^{M-1} h_1(k)x(n-k) \quad (3.1)$$

$$y_{LP2}(n) = \sum_{k=0}^{M-1} h_2(k)x(n-k) \quad (3.2)$$

- เมื่อ $h_1(k)$ คือผลตอบสนองทาง Impulse ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่จุดตัดความถี่ f_1
- $h_2(k)$ คือผลตอบสนองทาง Impulse ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่จุดตัดความถี่ f_2

ถ้ากำหนดให้ผลตอบสนองทาง Impulse ของตัวกรองเชิงเลข FIR เป็นแบบ Odd Positive Symmetry ดังนั้นสมการที่ 3.1 และสมการที่ 3.2 สามารถกระจายออกได้ดังสมการที่ 3.3 และ 3.4

$$y_{LP1}(n) = h_1(0)x(n) + h_1(1)x(n-1) + \dots + h_1\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) + \dots + h_1(k)x(n-k) \quad (3.3)$$

$$y_{LP2}(n) = h_2(0)x(n) + h_2(1)x(n-1) + \dots + h_2\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) + \dots + h_2(k)x(n-k) \quad (3.4)$$

และสมการ Output ของตัวกรองเชิงเลข FIR แบบ High-pass ที่มีจุดตัดความถี่ f_{c1} แสดงได้ดังสมการที่ 3.5

$$y_{HP1}(n) = -h_1(0)x(n) - h_1(1)x(n-1) - \dots - \left[-1 + h_1\left(\frac{k}{2}\right)\right]x\left(n-\frac{k}{2}\right) - \dots - h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - h_1(0)x(n) - h_1(1)x(n-1) - \dots - h_1\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) - \dots - h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - \sum_{k=0}^{M-1} h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - y_{LP1}(n) \quad (3.5)$$

ส่วนสมการ Output ของตัวกรองเชิงเลข FIR แบบ High-pass ที่มีจุดตัดความถี่ f_{c2} สามารถหาได้เหมือนกับสมการ Output ของตัวกรองเชิงเลข FIR แบบ High-pass ที่มีจุดตัดความถี่ f_{c1} ซึ่งแสดงได้ดังสมการที่ 3.6

$$y_{HP2}(n) = x\left(n-\frac{k}{2}\right) - y_{LP2}(n) \quad (3.6)$$

และสมการ Output ของตัวกรองเชิงเลข FIR แบบ Band-pass สามารถแสดงได้ดังสมการที่ 3.7

$$y_{BP}(n) = (h_2(0) - h_1(0))x(n) + (h_2(1) - h_1(1))x(n-1) + \dots + (h_2(k) - h_1(k))x(n-k)$$

$$y_{BP}(n) = \sum_{k=0}^{M-1} h_2(k)x(n-k) - \sum_{k=0}^{M-1} h_1(k)x(n-k)$$

$$y_{BP}(n) = y_{LP2}(n) - y_{LP1}(n) \quad (3.7)$$

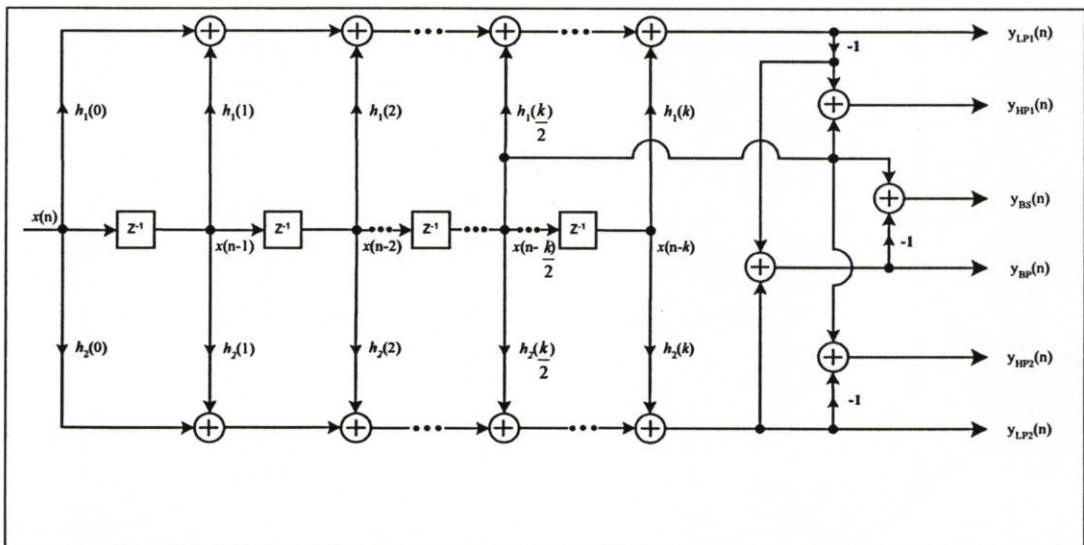
ส่วนสมการ Output ของตัวกรองเชิงเลข FIR แบบ Band-stop แสดงได้ดังสมการที่ 3.8

$$y_{BS}(n) = -(h_2(0) - h_1(0))x(n) - (h_2(1) - h_1(1))x(n-1) - \dots - \left(-1 + \left(h_2\left(\frac{k}{2}\right) - h_1\left(\frac{k}{2}\right)\right)\right)x\left(n - \frac{k}{2}\right) \dots - (h_2(k) - h_1(k))x(n-k)$$

$$y_{BS}(n) = x\left(n - \frac{k}{2}\right) - (h_2(0) - h_1(0))x(n) - (h_2(1) - h_1(1))x(n-1) - \dots - \left(h_2\left(\frac{k}{2}\right) - h_1\left(\frac{k}{2}\right)\right)x\left(n - \frac{k}{2}\right) \dots - (h_2(k) - h_1(k))x(n-k)$$

$$y_{BS}(n) = x\left(n - \frac{k}{2}\right) - y_{BP}(n) \quad (3.8)$$

เมื่อพิจารณาสมการที่ 3.1-3.8 พบว่าสมการทั้งหมดสามารถใช้ตัวแปร $x(n), x(n-1), \dots, x\left(n - \frac{k}{2}\right), \dots, x(n-k)$ ร่วมกันได้ดังนั้นจึงทำการออกแบบโครงสร้างของตัวกรองเชิงเลข FIR ขึ้นใหม่โดยใช้ตัวแปรบางส่วนร่วมกันดังรูปที่ 3.2 ซึ่งจะทำให้เกิดผลตอบสนองทางความถี่ได้หลายรูปแบบในเวลาเดียวกัน



รูปที่ 3.2 โครงสร้างของตัวกรองเชิงเลข FIR ที่มีผลตอบสนองความถี่หลายรูปแบบในเวลาเดียวกัน

จากโครงสร้างตามหลักการที่นำเสนอจะมีการใช้ Delay Adder และ Multiply น้อยลงเมื่อเทียบกับการสร้างตามหลักการเดิมซึ่งสามารถแสดงให้เห็นได้ดังตารางที่ 3.1 และ 3.2

ตารางที่ 3.1 การเปรียบเทียบการใช้ Delay Adder และ Multiply จากหลักการเดิมกับหลักการที่นำเสนอ กำหนดจำนวนลำดับ M

	หลักการเดิม			หลักการใหม่		
	Delay	Adder	Multiply	Delay	Adder	Multiply
Low-pass Filter (Low)	M	M	$M+1$	M	M	$M+1$
Low-pass Filter (High)	M	M	$M+1$		M	$M+1$
High-pass Filter (Low)	M	M	$M+1$		1	1
High-pass Filter (High)	M	M	$M+1$		1	1
Band-pass Filter	M	M	$M+1$		1	1
Band-stop Filter	M	M	$M+1$		1	1
รวม	$6M$	$6M$	$6M+1$	M	$M+4$	$2M+6$

ตารางที่ 3.2 การเปรียบเทียบการใช้ Delay Adder และ Multiply จากหลักการเดิมกับหลักการที่นำเสนอ กำหนดจำนวนลำดับ $M = 65$

	หลักการเดิม			หลักการใหม่		
	Delay	Adder	Multiply	Delay	Adder	Multiply
Low-pass Filter (Low)	65	65	66	65	65	66
Low-pass Filter (High)	65	65	66		65	66
High-pass Filter (Low)	65	65	66		1	1
High-pass Filter (High)	65	65	66		1	1
Band-pass Filter	65	65	66		1	1
Band-stop Filter	65	65	66		1	1
รวม	390	390	391	65	69	136

3.2 การออกแบบตัวกรองเชิงเลข FIR หลาย Output แบบเวลาจริง

จากโครงสร้างในรูปที่ 3.2 ตัวกรองเชิงเลข FIR หลาย Output แบบเวลาจริงจะใช้สัมประสิทธิ์ของตัวกรองความถี่ต่ำผ่าน (Low-pass Filter) ที่มีจุดตัดความถี่ (f_c) ต่างกัน 2 ความถี่ ซึ่งจะมีการใช้ Delay Element ร่วมกัน ทำให้ได้ตัวกรองความถี่สูงผ่าน (High-pass Filter) ตัวกรองความถี่ผ่าน (Band-pass Filter) และตัวกรองความถี่หยุด (Band-stop Filter) ในเวลาเดียวกัน โดยมีขั้นตอนการออกแบบดังนี้

การออกแบบที่ 1 เริ่มจากการออกแบบตัวกรองเชิงเลข FIR แบบ Low-pass ที่มี $f_{c1} = 5\text{kHz}$ และ $f_{c2} = 10\text{kHz}$ ความถี่สุ่ม (f_s) = 100kHz ลำดับ (order) 33 และใช้ฟังก์ชันของวินโดว์แบบเรคแทนกูลาร์วินโดว์ (Rectangular Window)

วิธีการ กำหนดให้ $f_{c1} = 5\text{kHz}$ ดังนั้นค่าสัมประสิทธิ์ $h_D(n)$ อยู่ในช่วง -16 จนถึง 16 จากสมการที่ 2.103

$$N = 33 \quad f = \frac{f_{c1}}{f_s} = \frac{5\text{kHz}}{100\text{kHz}} = 0.05$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

เรคแทนกูลาร์วินโดว์มีค่า $\omega(n) = 1$

$$n = 0; \quad h_D(0) = 2 \times f = 2 \times 0.05 = 0.1$$

$$\omega(0) = 1$$

$$h_1(0) = h_D(0)\omega(0)$$

$$h_1(0) = 0.1 \times 1$$

$$h_1(0) = 0.01 = h_1(32)$$

$$n = 1; \quad h_D(1) = \frac{\sin(2\pi \times 0.05)}{\pi} = 0.098363$$

$$\omega(1) = 1$$

$$h(1) = h_D(1)\omega(1) = 0.0984 \times 1 = 0.098363$$

$$h(1) = 0.098363 = h(-1)$$

$$h_1(5) = 6.3662 \times 10^{-2} = h_1(27)$$

$$h_1(6) = 5.0455 \times 10^{-2} = h_1(26)$$

$$h_1(7) = 3.6788 \times 10^{-2} = h_1(25)$$

$$h_1(8) = 2.3387 \times 10^{-2} = h_1(24)$$

$$h_1(9) = 1.0929 \times 10^{-2} = h_1(23)$$

$$h_1(10) = 0.0000 = h_1(22)$$

$$h_1(11) = -8.9421 \times 10^{-3} = h_1(21)$$

$$h_1(12) = 1.5591 \times 10^{-2} = h_1(20)$$

$$h_1(13) = -1.9809 \times 10^{-2} = h_1(19)$$

$$h_1(14) = -2.1624 \times 10^{-2} = h_1(18)$$

$$h_1(15) = -2.1221 \times 10^{-2} = h_1(17)$$

$$h_1(16) = -1.8921 \times 10^{-2} = h_1(16)$$

ดังนั้นสมการ Output ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ f_{c1} แสดง
ได้ดังสมการที่ 3.9

$$y_{LP1} = (1 \times 10^{-1})x(n) + (9.8363 \times 10^{-2})x(n-1) + \dots + (-1.8921 \times 10^{-2})x(n-16) \quad (3.9) \\ + \dots + (1 \times 10^{-1})x(n-32)$$

วิธีการ กำหนดให้ $f_{c2} = 10\text{kHz}$ จากสมการที่ 2.103

$$N = 33 \quad f = \frac{f_{c2}}{f_s} = \frac{10\text{kHz}}{100\text{kHz}} = 0.1$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

เรคแทนกูลาร์วินโดร์มีค่า $\omega(n) = 1$

$$n = 0; \quad h_D(0) = 2 \times f = 2 \times 0.1 = 0.2$$

$$\omega(0) = 1$$

$$h(0) = h_D(0) \cdot \omega(0)$$

$$h(0) = 0.2 \times 1$$

$$h(0) = 0.2 = h(32)$$

$$n = 1; \quad h_D(1) = \frac{\sin(2\pi \times 0.1)}{\pi} = 0.18710$$

$$\omega(1) = 1$$

$$h(1) = h_D(1)\omega(1) = 0.18710 \times 1 = 0.18710$$

$$h(1) = 0.18710 = h(-1)$$

$$n = 2; \quad h_D(2) = \frac{\sin(2 \times 2\pi \times 0.1)}{2\pi}$$

$$h_D(2) = 0.15137$$

$$\omega(2) = 1$$

$$h(2) = h_D(2)\omega(2) = 0.15137 \times 1 = 0.15137$$

$$h(2) = 0.15137 = h(-2)$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$n = 16; \quad h_D(16) = \frac{\sin(2\pi \times 16 \times 0.1)}{16\pi}$$

$$h_D(16) = -0.011694$$

$$\omega(16) = 1$$

$$h(16) = h_D(16)\omega(16) = -0.011694 \times 1 = -0.011694$$

$$h(16) = -0.011694 = h(-16)$$

ทำการเรียงค่าสัมประสิทธิ์เหมือนการออกแบบที่ $f_{c1} = 5\text{kHz}$ จะได้สัมประสิทธิ์ของตัวกรองเชิงเลข FIR แบบเรคแทนกูลาร์วิน โคร์ที่มี $f_{c2} = 10\text{kHz}$ $f_s = 100\text{kHz}$ ลำดับ 33 ดังต่อไปนี้

$$h_2(0) = 2.0000 \times 10^{-1} = h_2(32)$$

$$h_2(1) = 1.8710 \times 10^{-1} = h_2(31)$$

$$h_2(2) = 1.5137 \times 10^{-1} = h_2(30)$$

$$h_2(3) = 1.0091 \times 10^{-1} = h_2(29)$$

$$h_2(4) = 4.6774 \times 10^{-2} = h_2(28)$$

$$h_2(5) = 0.0000 = h_2(27)$$

$$h_2(6) = -3.1183 \times 10^{-2} = h_2(26)$$

$$h_2(7) = -4.3247 \times 10^{-2} = h_2(25)$$

$$h_2(8) = -3.7841 \times 10^{-2} = h_2(24)$$

$$h_2(9) = -2.0789 \times 10^{-2} = h_2(23)$$

$$h_2(10) = 0.0000 = h_2(22)$$

$$h_2(11) = 1.7009 \times 10^{-2} = h_2(21)$$

$$h_2(12) = 2.5227 \times 10^{-2} = h_2(20)$$

$$h_2(13) = 2.3287 \times 10^{-2} = h_2(19)$$

$$h_2(14) = 1.3364 \times 10^{-2} = h_2(18)$$

$$h_2(15) = 0.0000 = h_2(17)$$

$$h_2(16) = -1.1694 \times 10^{-2} = h_2(16)$$

ดังนั้นสมการ Output ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ f_{c2} แสดงได้ดังสมการที่ 3.10

$$y_{LP2} = (2 \times 10^{-1})x(n) + (1.8709 \times 10^{-2})x(n-1) + \dots + (-1.1694 \times 10^{-2})x(n-16) + \dots + (2 \times 10^{-1})x(n-32) \quad (3.10)$$

เมื่อทำการออกแบบตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ f_{c1} และ f_{c2} ก็นำมาเข้าตามโครงสร้างดังรูปที่ 3.2 แล้วทำให้ได้ตัวกรองเชิงเลข FIR แบบ High-pass ที่มีค่าความถี่ต่างกัน 2 ความถี่ ตัวกรองเชิงเลข FIR แบบ Band-pass และตัวกรองเชิงเลข FIR แบบ Band-stop ซึ่งแสดงได้ดังสมการที่ 3.11-3.14

$$y_{HP1} = (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - [-1 + (-1.8920 \times 10^{-2})]x(n-16) \\ - \dots - (1 \times 10^{-1})x(n-32)$$

$$y_{HP1} = (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - (-1.018920 \times 10^{-2})x(n-16) \\ - \dots - (1 \times 10^{-1})x(n-32) \quad (3.11)$$

$$y_{HP2} = (-2 \times 10^{-1})x(n) - (1.8709 \times 10^{-2})x(n-1) - \dots - [-1 + (-1.1169 \times 10^{-2})]x(n-16) \\ - \dots - (2 \times 10^{-1})x(n-32)$$

$$y_{HP2} = (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - (-1.011169 \times 10^{-2})x(n-16) \\ - \dots - (1 \times 10^{-1})x(n-32) \quad (3.12)$$

$$y_{BP} = (1 \times 10^{-1})x(n) + (-2.1005 \times 10^{-2})x(n-1) + \dots + (1 \times 10^{-1})x(n-32) \quad (3.13)$$

$$y_{BS} = -(1 \times 10^{-1})x(n) - (-2.1005 \times 10^{-2})x(n-1) - \dots - [1 - (1 \times 10^{-1})x(n-16)] \\ - \dots - (1 \times 10^{-1})x(n-32) \quad (3.14)$$

การออกแบบที่ 2 ออกแบบตัวกรองเชิงเลข FIR แบบ Low-pass ที่มี $f_{c1} = 5kHz$ และ $f_{c2} = 10kHz$ ความถี่สุ่ม (f_s) = 100kHz ลำดับ (Order) 65 และใช้ฟังก์ชันของวินโดว์แบบเรคแทนกูลาร์วินโดว์ (Rectangular Window)

วิธีการ กำหนดให้ $f_{c1} = 5kHz$ ดังนั้นค่าสัมประสิทธิ์ $h_D(n)$ อยู่ในช่วง -32 จนถึง 32 จากสมการที่ 2.103

$$N = 65 \quad f = \frac{f_{c1}}{f_s} = \frac{5KHz}{100KHz} = 0.05$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

เรคแทนกูลาร์วินโดร์มีค่า $\omega(n) = 1$

$$n = 0; \quad h_D(0) = 2 \times f = 2 \times 0.05 = 0.1$$

$$\omega(0) = 1$$

$$h_1(0) = h_D(0) \cdot \omega(0)$$

$$h_1(0) = 0.1 \times 1$$

$$h_1(0) = 0.01 = h_1(32)$$

$$n = 1; \quad h_D(1) = \frac{\sin(2\pi \times 0.05)}{\pi} = 0.098363$$

$$\omega(1) = 1$$

$$h(1) = h_D(1) \cdot \omega(1) = 0.0984 \times 1 = 0.098363$$

$$h(1) = 0.098363 = h(-1)$$

$$n = 2; \quad h_D(2) = \frac{\sin(2 \times 2\pi \times 0.05)}{2\pi}$$

$$h_D(2) = 0.093549$$

$$\omega(2) = 1$$

$$h(2) = h_D(2)\omega(2) = 0.093549 \times 1 = 0.093549$$

$$h(2) = 0.093549 = h(-2)$$

⋮

$$n = 32 \quad h_D(32) = \frac{\sin(2\pi \times 32 \times 0.05)}{32\pi}$$

$$h_D(32) = -0.0058468$$

$$\omega(32) = 1$$

$$h(32) = h_D(32)\omega(32) = -0.0058468 \times 1 = -0.0058468$$

$$h(32) = -0.018920 = h(-32)$$

ทำตัวกรองให้เป็นคอลซอล (Causal) ทำโดยบวก n ทุกตัวด้วย 32 ทำให้ n เปลี่ยนจากช่วง -32 ถึง 32 มาเป็นช่วง 0 ถึง 64 จะได้สัมประสิทธิ์ของตัวกรองเชิงเลข FIR แบบเรคแทนกูลาร์ วินโดร์ $f_{ci} = 5\text{kHz}$ ความถี่สุ่ม (f_s) = 100kHz ลำดับ (Order) 65 ดังต่อไปนี้

$$h_1(0) = 1.0000 \times 10^{-2} = h_1(64)$$

$$h_1(1) = 9.8363 \times 10^{-2} = h_1(63)$$

$$h_1(2) = 9.3549 \times 10^{-2} = h_1(62)$$

$$h_1(3) = 8.5839 \times 10^{-2} = h_1(61)$$

$$h_1(4) = 7.5683 \times 10^{-2} = h_1(60)$$

$$h_1(5) = 6.3662 \times 10^{-2} = h_1(59)$$

$$h_1(6) = 5.0455 \times 10^{-2} = h_1(58)$$

$$h_1(7) = 3.6788 \times 10^{-2} = h_1(57)$$

$$h_1(8) = 2.3387 \times 10^{-2} = h_1(56)$$

$$h_1(9) = 1.0929 \times 10^{-2} = h_1(55)$$

$$h_1(10) = 0.0000 = h_1(54)$$

$$h_1(11) = -8.9421 \times 10^{-3} = h_1(53)$$

$$h_1(12) = 1.5591 \times 10^{-2} = h_1(52)$$

$$\begin{aligned}
h_1(13) &= -1.9809 \times 10^{-2} = h_1(51) \\
h_1(14) &= -2.1624 \times 10^{-2} = h_1(50) \\
h_1(15) &= -2.1220 \times 10^{-2} = h_1(49) \\
h_1(16) &= -1.8920 \times 10^{-2} = h_1(48) \\
h_1(17) &= -1.5148 \times 10^{-2} = h_1(47) \\
h_1(18) &= -1.0394 \times 10^{-2} = h_1(46) \\
h_1(19) &= -5.1770 \times 10^{-3} = h_1(45) \\
h_1(20) &= 0.0000 = h_1(44) \\
h_1(21) &= 4.6840 \times 10^{-3} = h_1(43) \\
h_1(22) &= 8.5044 \times 10^{-3} = h_1(42) \\
h_1(23) &= 1.1196 \times 10^{-2} = h_1(41) \\
h_1(24) &= 1.2614 \times 10^{-2} = h_1(40) \\
h_1(25) &= 1.2732 \times 10^{-2} = h_1(39) \\
h_1(26) &= 1.1643 \times 10^{-2} = h_1(38) \\
h_1(27) &= 9.5377 \times 10^{-3} = h_1(37) \\
h_1(28) &= 6.682 \times 10^{-3} = h_1(36) \\
h_1(29) &= 3.3918 \times 10^{-3} = h_1(35) \\
h_1(30) &= 0.0000 = h_1(34) \\
h_1(31) &= -3.1730 \times 10^{-3} = h_1(33) \\
h_1(32) &= -5.8468 \times 10^{-3} = h_1(32)
\end{aligned}$$

ดังนั้นสมการ Output ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ f_{c1} แสดงได้ดังสมการที่ 3.15

$$\begin{aligned}
y_{LP1} &= (1 \times 10^{-1})x(n) + (9.8363 \times 10^{-2})x(n-1) + \dots + (-1.8920 \times 10^{-2})x(n-16) \\
&\quad + \dots + (1 \times 10^{-1})x(n-64)
\end{aligned} \tag{3.15}$$

วิธีการ กำหนดให้ $f_{c2} = 10\text{kHz}$ จากสมการที่ 2.103

$$N = 65 \qquad f = \frac{f_{c2}}{f_s} = \frac{10\text{kHz}}{100\text{kHz}} = 0.1$$

$$h_D(n) = \frac{\sin(2\pi fn)}{n\pi}$$

เรขาคณิตอาร์วินโคร์มีค่า $\omega(n) = 1$

$$n = 0; \quad h_D(0) = 2 \times f = 2 \times 0.1 = 0.2$$

$$\omega(0) = 1$$

$$h(0) = h_D(0) \cdot \omega(0)$$

$$h(0) = 0.2 \times 1$$

$$h(0) = 0.2 = h(32)$$

$$n = 1; \quad h_D(1) = \frac{\sin(2\pi \times 0.1)}{\pi} = 0.18710$$

$$\omega(1) = 1$$

$$h(1) = h_D(1) \cdot \omega(1) = 0.18710 \times 1 = 0.18710$$

$$h(1) = 0.18710 = h(-1)$$

$$n = 2; \quad h_D(2) = \frac{\sin(2 \times 2\pi \times 0.1)}{2\pi}$$

$$h_D(2) = 0.15137$$

$$\omega(2) = 1$$

$$h(2) = h_D(2)\omega(2) = 0.15137 \times 1 = 0.15137$$

$$\begin{aligned}
h_2(13) &= 2.3287 \times 10^{-2} = h_2(51) \\
h_2(14) &= 1.3364 \times 10^{-2} = h_2(50) \\
h_2(15) &= 0.0000 = h_2(49) \\
h_2(16) &= -1.1169 \times 10^{-2} = h_2(48) \\
h_2(17) &= -1.7808 \times 10^{-2} = h_2(47) \\
h_2(18) &= -1.6818 \times 10^{-2} = h_2(46) \\
h_2(19) &= -9.8473 \times 10^{-1} = h_2(45) \\
h_2(20) &= 0.0000 = h_2(44) \\
h_2(21) &= 8.9094 \times 10^{-3} = h_2(43) \\
h_2(22) &= 1.3760 \times 10^{-2} = h_2(42) \\
h_2(23) &= 1.3162 \times 10^{-2} = h_2(41) \\
h_2(24) &= 7.7957 \times 10^{-3} = h_2(40) \\
h_2(25) &= 0.0000 = h_2(39) \\
h_2(26) &= -7.1961 \times 10^{-3} = h_2(38) \\
h_2(27) &= -1.1212 \times 10^{-2} = h_2(37) \\
h_2(28) &= -1.0812 \times 10^{-2} = h_2(36) \\
h_2(29) &= -6.4517 \times 10^{-3} = h_2(35) \\
h_2(30) &= 0.0000 = h_2(34) \\
h_2(31) &= 6.0354 \times 10^{-3} = h_2(33) \\
h_2(32) &= 9.4603 \times 10^{-3} = h_2(32)
\end{aligned}$$

ดังนั้นสมการ Output ของตัวกรองเชิงเลข FIR แบบ Low-pass ที่มีจุดตัดความถี่ f_{c2} แสดงได้ดังสมการที่ 3.16

$$\begin{aligned}
y_{LP2} &= (2 \times 10^{-1})x(n) + (1.8709 \times 10^{-2})x(n-1) + \dots + (-1.1694 \times 10^{-2})x(n-16) \\
&\quad + \dots + (2 \times 10^{-1})x(n-64)
\end{aligned} \tag{3.16}$$

ดังนั้นสมการ Output ของตัวกรองเชิงเลข FIR แสดงได้ดังสมการที่ 3.17-3.20

$$\begin{aligned}
y_{HP1} &= (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - [-1 + (-1.8920 \times 10^{-2})]x(n-16) \\
&\quad - \dots - (1 \times 10^{-1})x(n-64)
\end{aligned}$$

$$y_{HP1} = (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - (-1.018920 \times 10^{-2})x(n-16) \\ - \dots - (1 \times 10^{-1})x(n-64) \quad (3.17)$$

$$y_{HP2} = (-2 \times 10^{-1})x(n) - (1.8709 \times 10^{-2})x(n-1) - \dots - [-1 + (-1.1694 \times 10^{-2})]x(n-16) \\ - \dots - (2 \times 10^{-1})x(n-64)$$

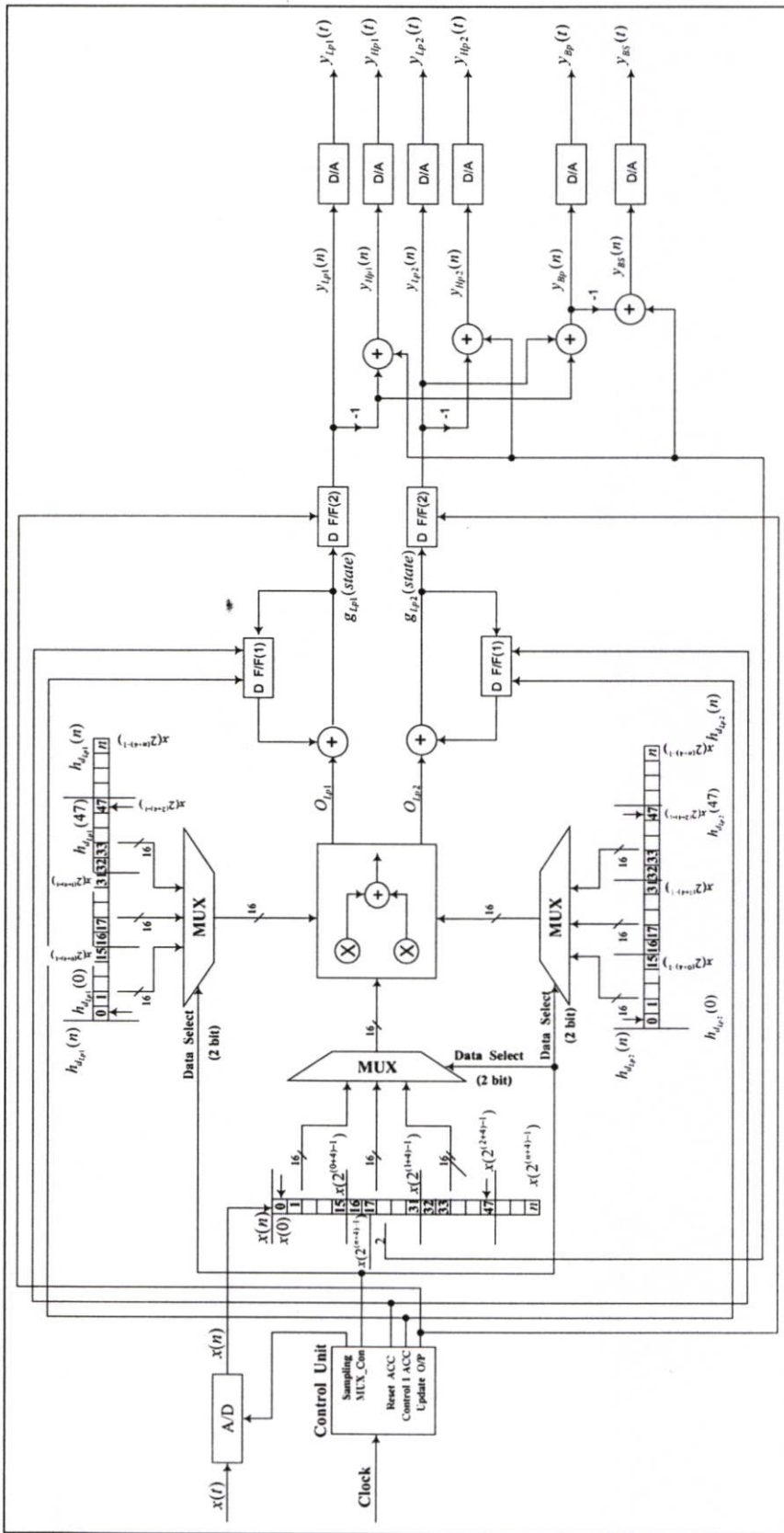
$$y_{HP2} = (-1 \times 10^{-1})x(n) - (9.8363 \times 10^{-2})x(n-1) - \dots - (-1.011694 \times 10^{-2})x(n-16) \\ - \dots - (1 \times 10^{-1})x(n-64) \quad (3.18)$$

$$y_{BP} = (1 \times 10^{-1})x(n) + (-2.1005 \times 10^{-2})x(n-1) + \dots + (1 \times 10^{-1})x(n-64) \quad (3.19)$$

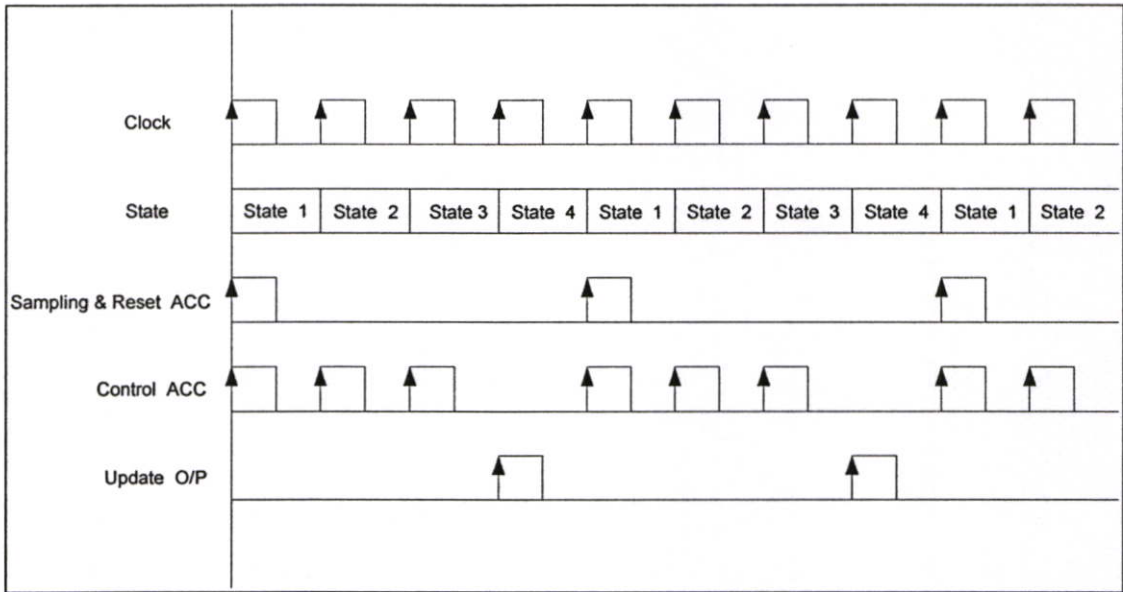
$$y_{BS} = -(1 \times 10^{-1})x(n) - (-2.1005 \times 10^{-2})x(n-1) - \dots - [1 - (1 \times 10^{-1})x(n-16)] \\ - \dots - (1 \times 10^{-1})x(n-64) \quad (3.20)$$

3.3 การสร้างตัวกรองเชิงเลข FIR หลาย Output แบบเวลาจริงด้วย FPGA

ในการออกแบบ Hardware สำหรับสร้างตัวกรองเชิงเลข FIR ที่ให้ผลตอบสนองความถี่ได้หลายรูปแบบในเวลาเดียวกันได้ใช้ APEX DSP Development Board with 8320 Logic [8] ส่วน Software ได้ใช้โปรแกรม Quartus II Limited Edition โดย Diagram ของ FPGA แสดงดังรูปที่ 3.3 ซึ่ง Diagram นี้จะถูกออกแบบให้สอดคล้องกับโครงสร้างของตัวกรองเชิงเลข FIR ในรูปที่ 3.2 และจาก Diagram ดังรูปที่ 3.3 นี้จะมีส่วนประกอบดังนี้คือ Control Unit Memory Array Multiply Adder Unit Adder Element Latch Element A/D และ D/A [11,12]



รูปที่ 3.3 โครงสร้างของ FPGA



รูปที่ 3.4 Timing Diagram ของ FPGA

ในการประมวลผลจะทำในลักษณะ 16 บิต Fix-point [7] แบบคิดเครื่องหมายซึ่งใน 1 รอบการทำงาน (Cycle) จะใช้เวลา 4 Clock Pulse ดังรูปที่ 3.4 ซึ่งในการอธิบายการทำงานนั้นสามารถอธิบายได้จากช่วงเวลาของแต่ละช่วง Clock Pulse โดยในสถานะที่ 1 2 และ 3 จะเป็นการคูณค่าสัมประสิทธิ์ของ Low-pass Filter ซึ่งแบ่งการคำนวณทีละ 16 บิต ส่วนสถานะที่ 4 นั้นเป็นการนำค่าทั้งหมดมา Sum และทำการเรียงค่า Input ใหม่

สถานะ 1: รีเซต D F/F(1) D F/F(2) รับค่า $x(0)$

$$O_{Lp1} = h_{d_{lp1}}(0)x(0) + h_{d_{lp1}}(1)x(1) + \dots + h_{d_{lp1}}(15)x(15)$$

$$g_{Lp1}(1) = 0 + O_{Lp1}$$

$$O_{Lp2} = h_{d_{lp2}}(0)x(0) + h_{d_{lp2}}(1)x(1) + \dots + h_{d_{lp2}}(15)x(15)$$

$$g_{Lp2}(1) = 0 + O_{Lp2}$$

สถานะ 2:

$$O_{Lp1} = h_{d_{lp1}}(16)x(16) + h_{d_{lp1}}(17)x(17) + \dots + h_{d_{lp1}}(31)x(31)$$

$$g_{LP1}(2) = O_{LP1} + g_{LP1}(1)$$

$$O_{LP2} = h_{d_{LP2}}(16)x(16) + h_{d_{LP2}}(17)x(17) + \dots + h_{d_{LP2}}(31)x(31)$$

$$g_{LP2}(1) = O_{LP2} + g_{LP2}(1)$$

สภาวะ 3 :

$$O_{LP1} = h_{d_{LP1}}(32)x(32) + h_{d_{LP1}}(33)x(33) + \dots + h_{d_{LP1}}(47)x(47)$$

$$g_{LP1}(3) = O_{LP1} + g_{LP1}(2)$$

$$O_{LP2} = h_{d_{LP2}}(32)x(32) + h_{d_{LP2}}(33)x(33) + \dots + h_{d_{LP2}}(47)x(47)$$

$$g_{LP2}(3) = O_{LP2} + g_{LP2}(2)$$

สภาวะ 4 :

$$y_{LP1}(n) = g_{LP1}(3)$$

$$y_{LP2}(n) = g_{LP2}(3)$$

$$x(n+1) = x(n)$$

$$\vdots$$

$$x(47) = x(46)$$

$$x(46) = x(45)$$

$$x(45) = x(44)$$

$$\vdots$$

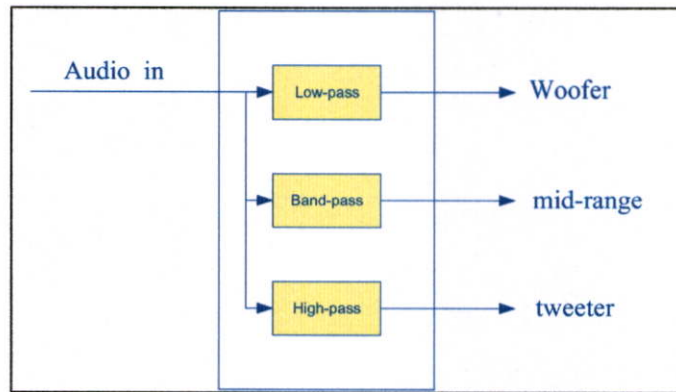
$$x(1) = x(0)$$

การออกแบบในทาง Software จะทำตามโครงสร้างดังรูปที่ 3.3 โดยทำการแบ่งโครงสร้างออกเป็นส่วนๆหรือมอดูล ซึ่งในแต่ละมอดูลจะถูกเขียนด้วยภาษา VHDL [13,14] และการใช้วงจร

สำเร็จรูปที่มีอยู่ใน โปรแกรม Quartus II จากนั้นนำมาวางลงในผังวงจร (Schematic) ในการออกแบบแต่ละมอดูลจะออกแบบให้มีจำนวน input/output เท่ากับ 2^N

3.4 การประยุกต์ใช้งาน

เนื่องจากตัวกรองเชิงเลข FIR หลายเข้าชุดแบบเวลาจริงโดยใช้ FPGA ให้ผลตอบสนองทางความถี่ (Frequency Response) 4 รูปแบบในเวลาเดียวกัน สามารถนำไปใช้เกี่ยวกับงานทางด้านการปรับปรุงคุณภาพเสียง เช่น Crossover Filter [6] ซึ่งใช้ในการแยกเสียงจากลำโพงแสดงดังรูปที่ 3.5 โดยสามารถใช้ตัวกรองความถี่ทั้ง 3 รูปแบบในเวลาเดียวกัน



รูปที่ 3.5 Crossover Filter

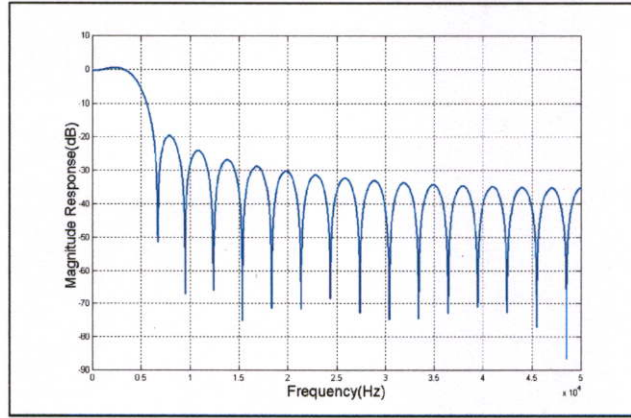
บทที่ 4

ผลการทดลอง

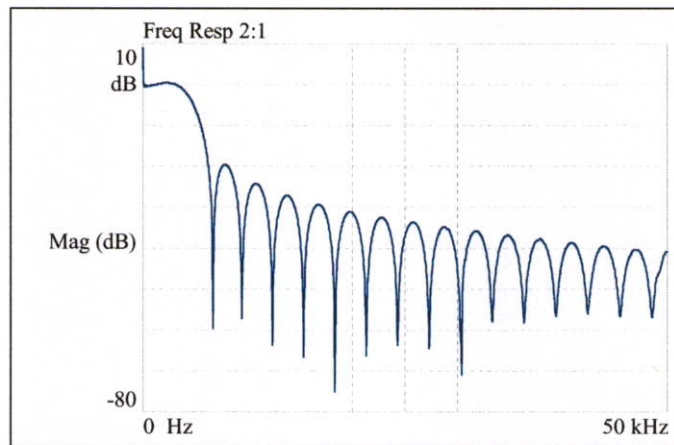
ในการทดลองได้ทำการออกแบบและสร้างตัวกรองเชิงเลข FIR หลาย Output ในเวลาเดียวกันด้วย FPGA ได้แบ่งการทดลองออกเป็นขั้นตอนต่างๆดังนี้

- 4.1 การออกแบบ หาค่าสัมประสิทธิ์ ของตัวกรองเชิงเลข FIR แบบ Low-pass และจำลองการทำงานของตัวกรองเชิงเลข FIR หลาย Output ด้วยโปรแกรม MATLAB
- 4.2 การออกแบบและสร้างตัวกรองเชิงเลข FIR หลาย Output โดยการสร้างจริงด้วย FPGA ตามโครงสร้างในรูปที่ 3.2
- 4.3 ทำการวัดผลตอบสนองทางขนาดด้วยเครื่องวิเคราะห์สัญญาณแบบไดนามิกส์ (Dynamic Signal Analyzer)
- 4.4 การเปรียบเทียบผลการออกแบบระหว่างการจำลองด้วย โปรแกรม MATLAB

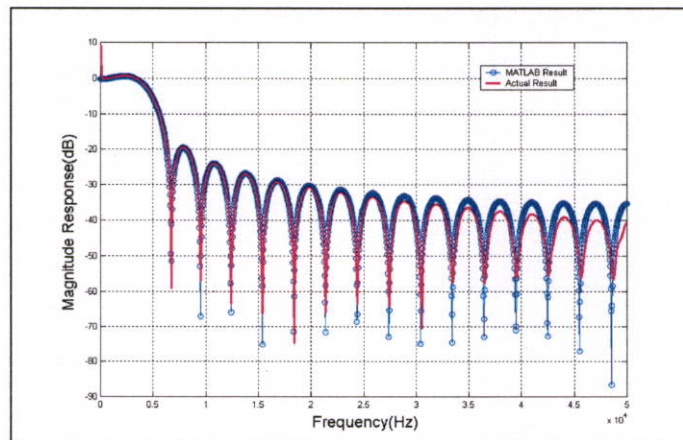
ผลการออกแบบที่ 1 จากการออกแบบในบทที่ 3 การออกแบบที่ 1 ที่มีจำนวนลำดับเท่ากับ 33 $f_s = 100\text{kHz}$ $f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ จะได้ผลตอบสนองทางขนาดดังรูปที่ 4.1ก-4.6ค



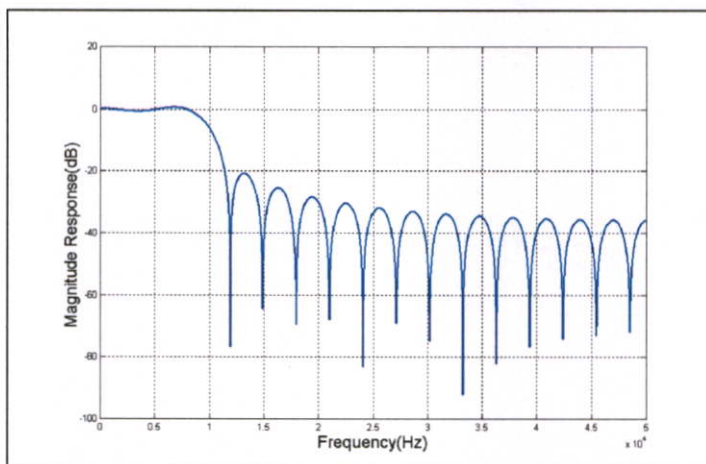
ภาพที่ 4.1ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$
ลำดับ 33 ด้วยโปรแกรม MATLAB



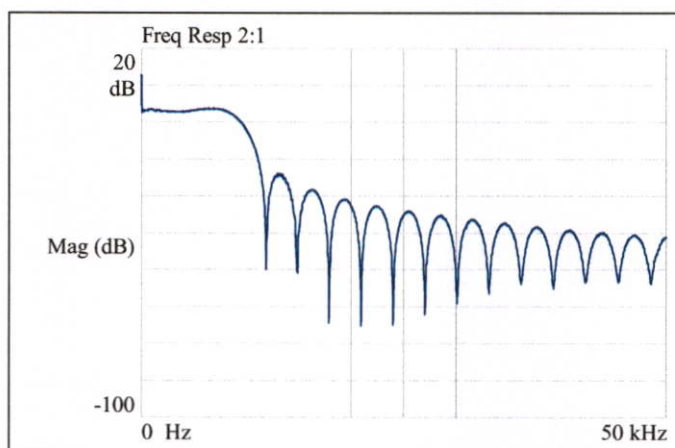
ภาพที่ 4.1ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$
ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



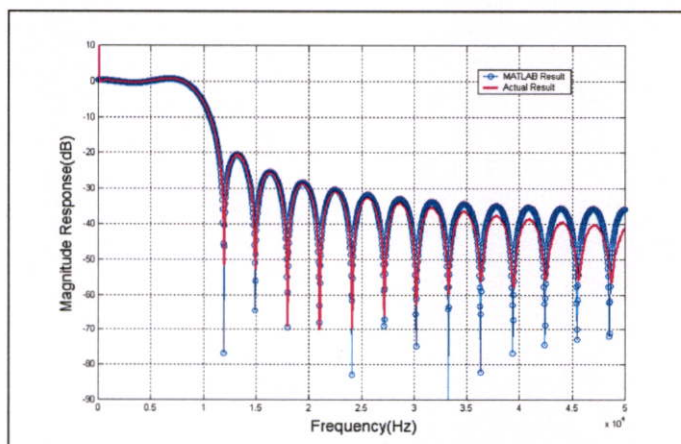
ภาพที่ 4.1ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วย
โปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5\text{kHz}$ ลำดับ 33



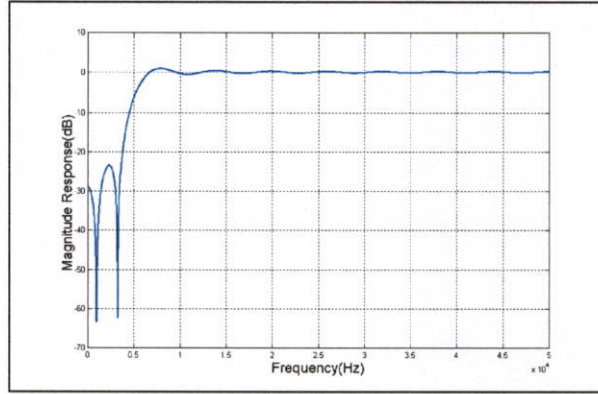
ภาพที่ 4.2ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$
ลำดับ 33 ด้วยโปรแกรม MATLAB



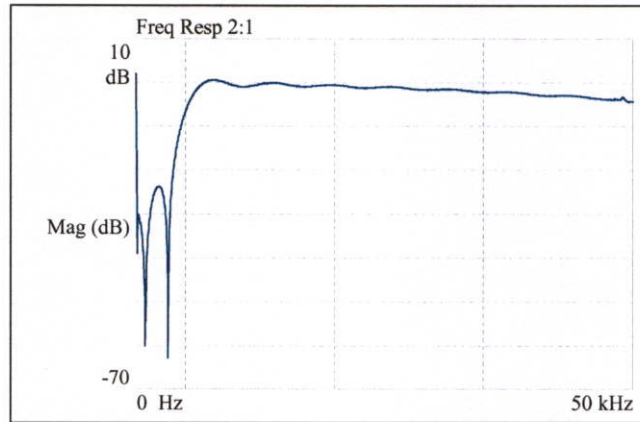
ภาพที่ 4.2ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$
ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



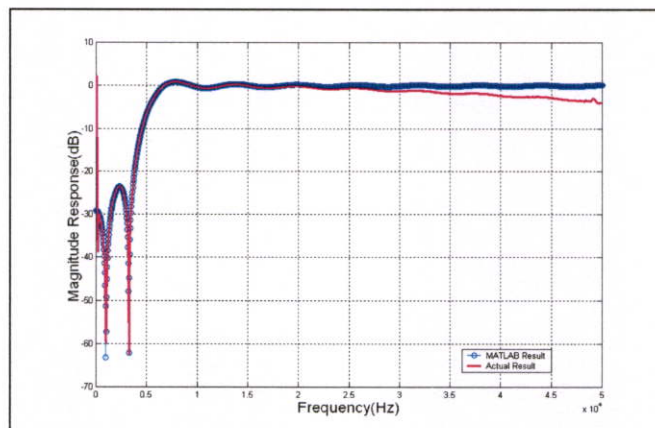
ภาพที่ 4.2ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วย
โปรแกรม MATLAB กับการสร้างจริงที่ $f_{c2} = 10\text{kHz}$ ลำดับ 33



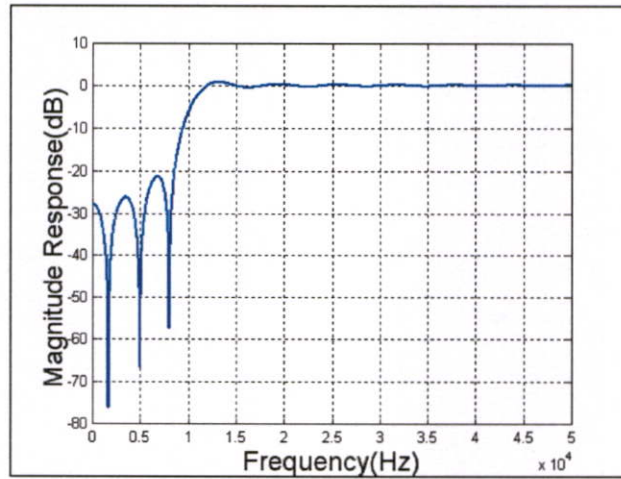
ภาพที่ 4.3ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$
ลำดับ 33 ด้วยโปรแกรม MATLAB



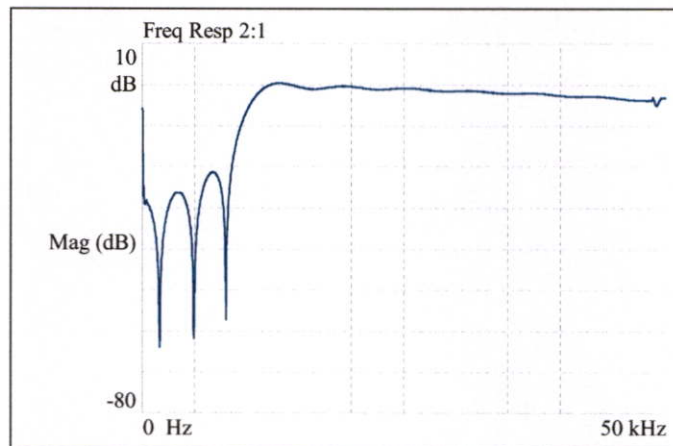
ภาพที่ 4.3ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$
ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



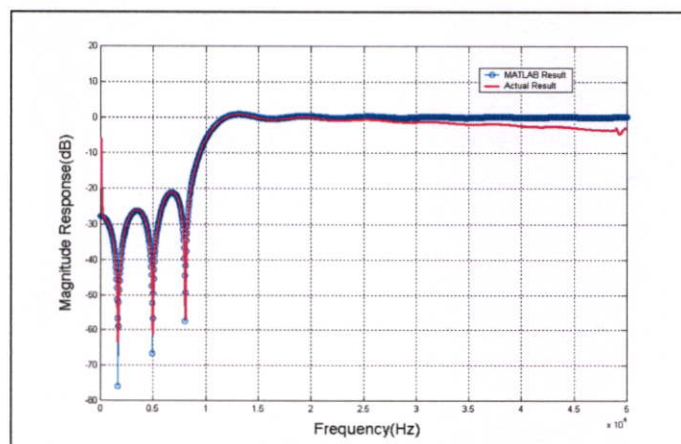
ภาพที่ 4.3ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วย
โปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5\text{kHz}$ ลำดับ 33



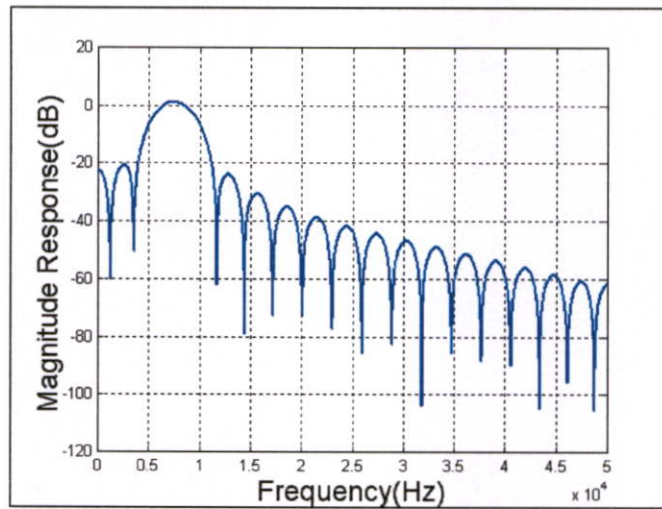
ภาพที่ 4.4ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 33 ด้วยโปรแกรม MATLAB



ภาพที่ 4.4ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 33จากการวัดจริงด้วยเครื่อง Dynamic Analyzer

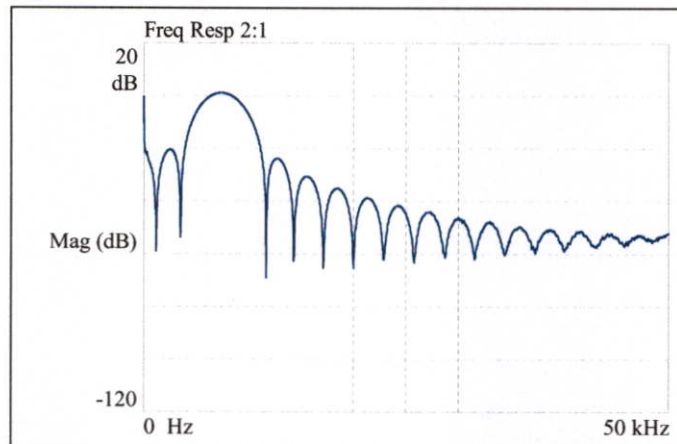


ภาพที่ 4.4ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่ $f_{c2} = 10\text{kHz}$ ลำดับ 33



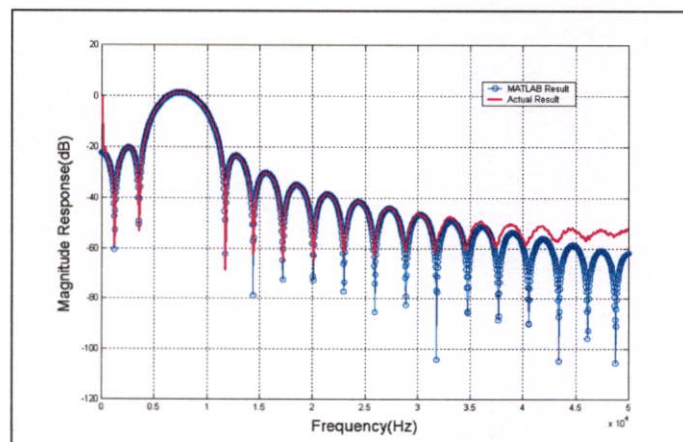
ภาพที่ 4.5ก ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่

$f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ ลำดับ 33 ด้วยโปรแกรม MATLAB



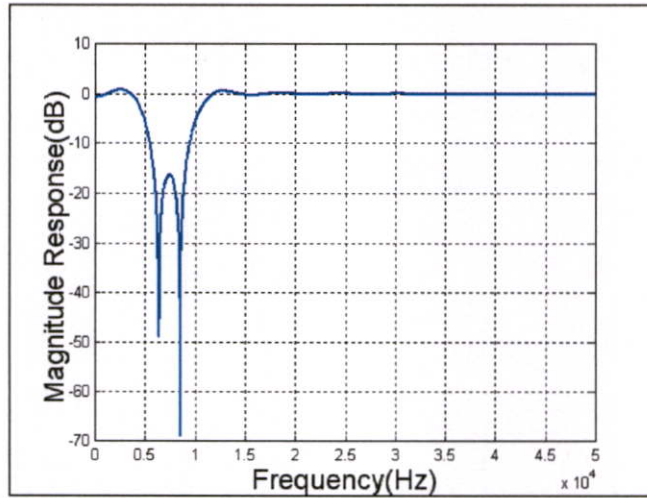
ภาพที่ 4.5ข ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$

$f_{c2} = 10\text{kHz}$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



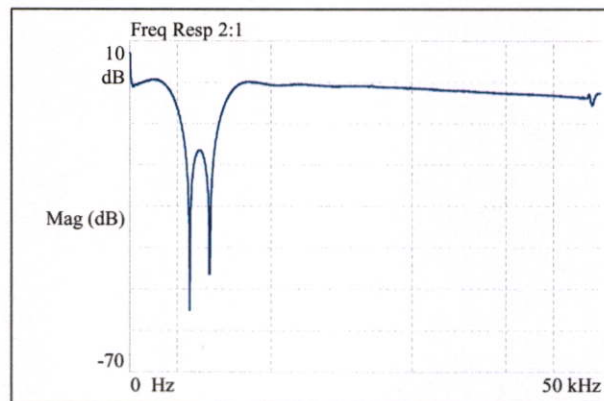
ภาพที่ 4.5ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วย

โปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ ลำดับ 33



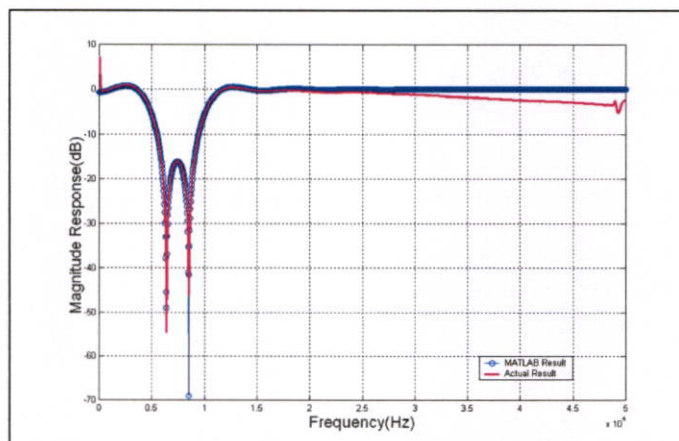
ภาพที่ 4.6ก ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่

$f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33 ด้วยโปรแกรม MATLAB



ภาพที่ 4.6ข ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$

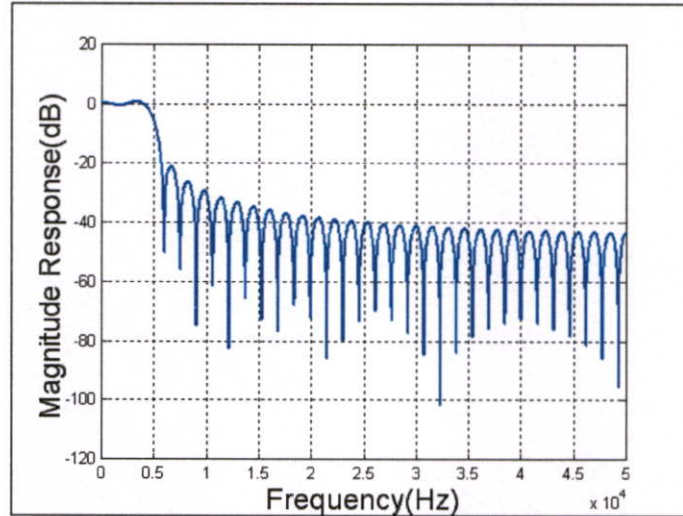
$f_{c2} = 10kHz$ ลำดับ 33 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



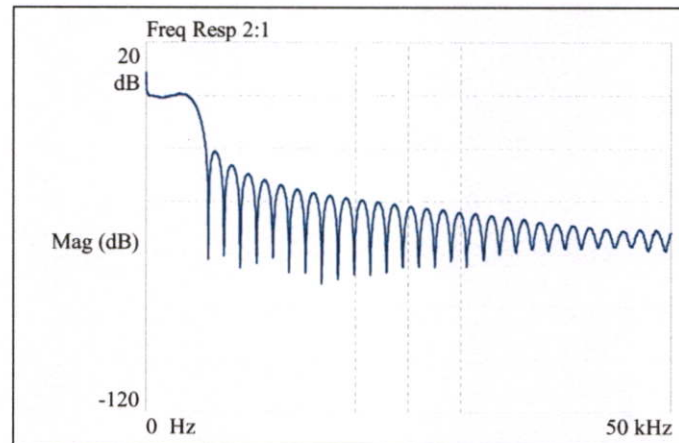
ภาพที่ 4.6ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วย

โปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 33

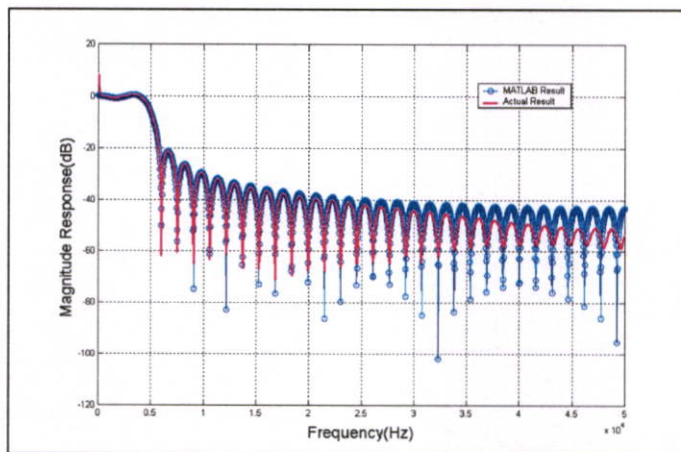
ผลการออกแบบที่ 2 จากการออกแบบในบทที่ 3 การออกแบบที่ 2 ที่มีจำนวนลำดับเท่ากับ 65 $f_s = 100\text{kHz}$ $f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ จะได้ผลตอบสนองทางขนาดดังรูปที่ 4.7ก-4.12ค



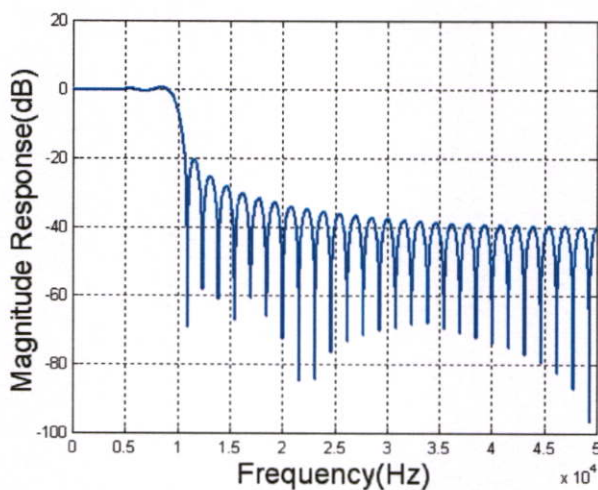
ภาพที่ 4.7ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$ ลำดับ 65 ด้วยโปรแกรม MATLAB



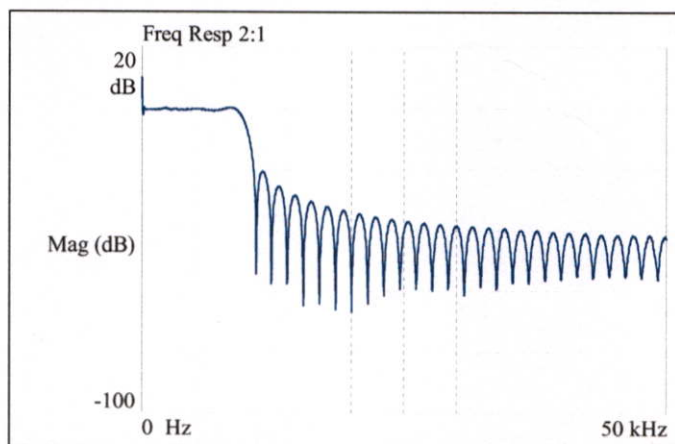
ภาพที่ 4.7ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



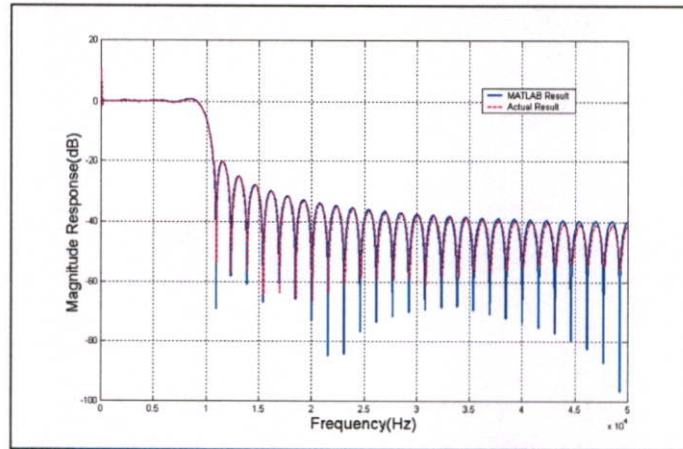
ภาพที่ 4.7ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5\text{kHz}$ ลำดับ 65



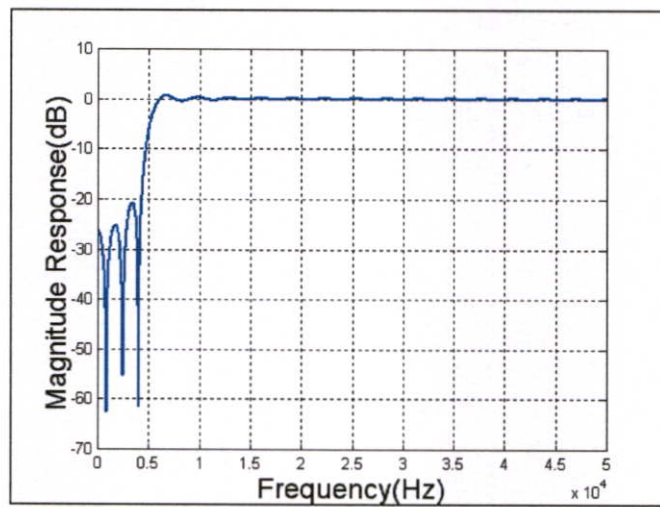
ภาพที่ 4.8ก ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 65 ด้วยโปรแกรม MATLAB



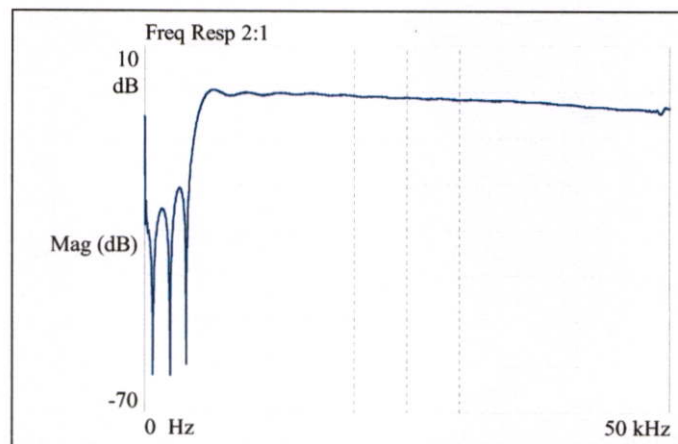
ภาพที่ 4.8ข ผลตอบสนองทางความถี่แบบ Low-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



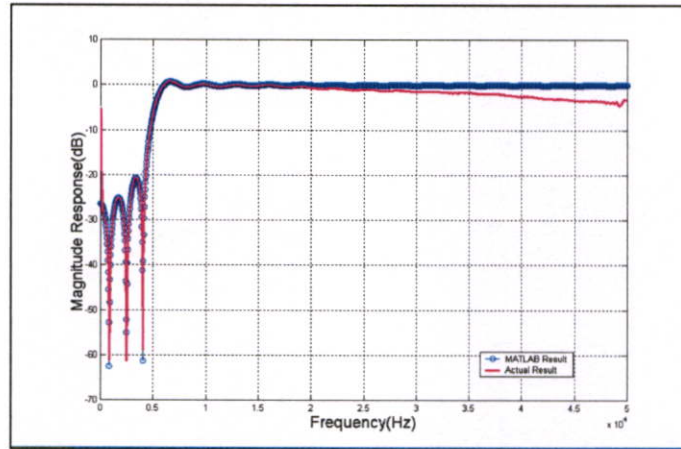
ภาพที่ 4.8ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริง $f_{c2} = 10\text{kHz}$ ลำดับ 65



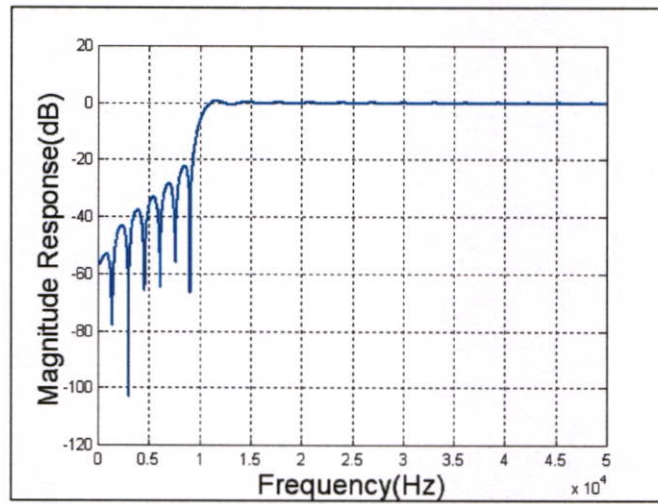
ภาพที่ 4.9ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$ ลำดับ 65 ด้วยโปรแกรม MATLAB



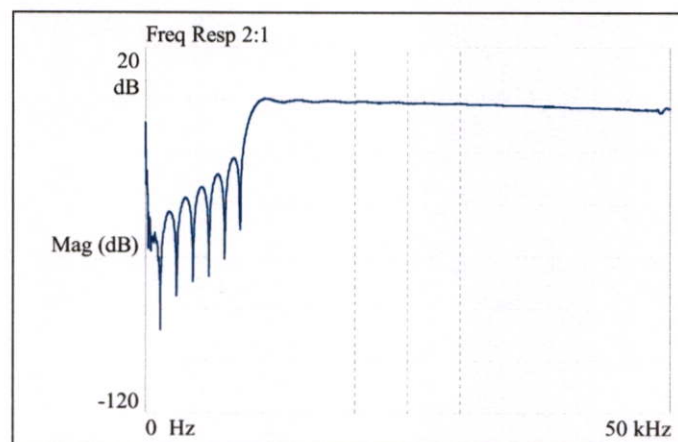
ภาพที่ 4.9ข ผลตอบสนองทางความถี่แบบ High-pass $f_{c1} = 5\text{kHz}$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



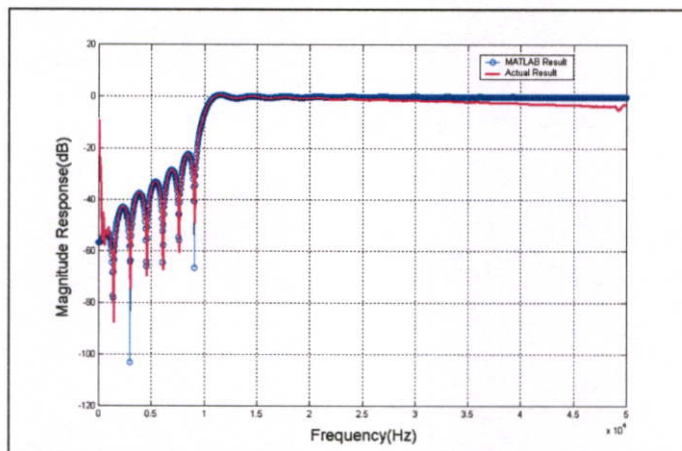
ภาพที่ 4.9ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่ $f_{c1} = 5\text{kHz}$ ลำดับ 65



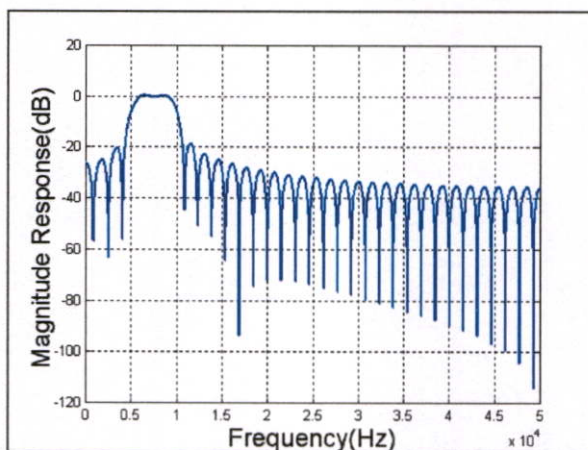
ภาพที่ 4.10ก ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 65 ด้วยโปรแกรม MATLAB



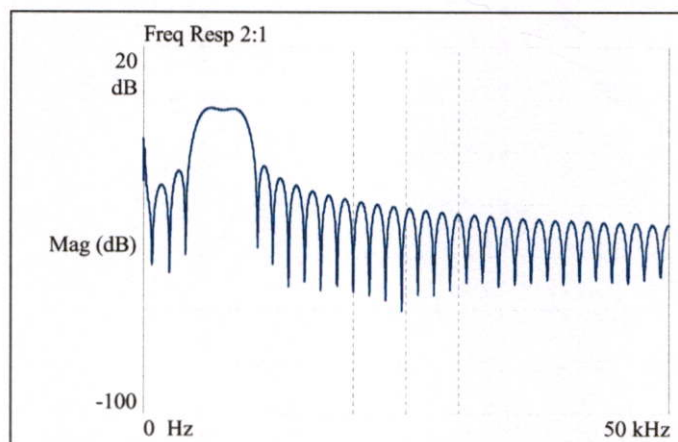
ภาพที่ 4.10ข ผลตอบสนองทางความถี่แบบ High-pass ที่มีจุดตัดความถี่ $f_{c2} = 10\text{kHz}$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



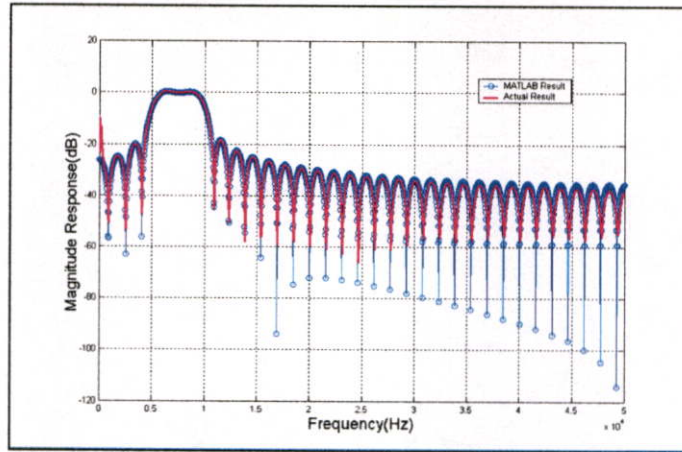
ภาพที่ 4.10ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริงที่ $f_{c2} = 10\text{kHz}$ ลำดับ 65



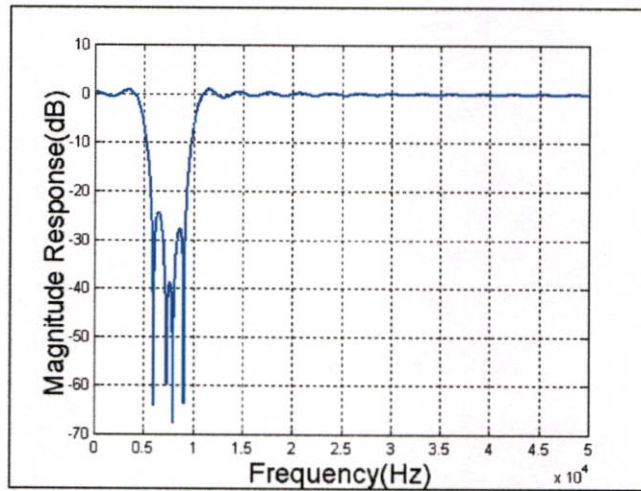
ภาพที่ 4.11ก ผลตอบสนองทางความถี่แบบ Band-pass ที่มีจุดตัดความถี่ $f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ ลำดับ 65 ด้วยโปรแกรม MATLAB



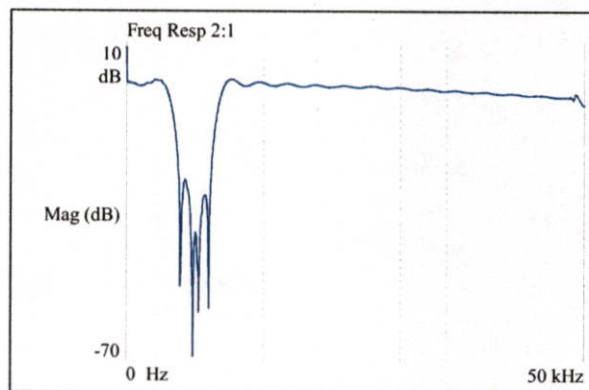
ภาพที่ 4.11ข ผลตอบสนองทางความถี่แบบ Band-pass $f_{c1} = 5\text{kHz}$ $f_{c2} = 10\text{kHz}$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



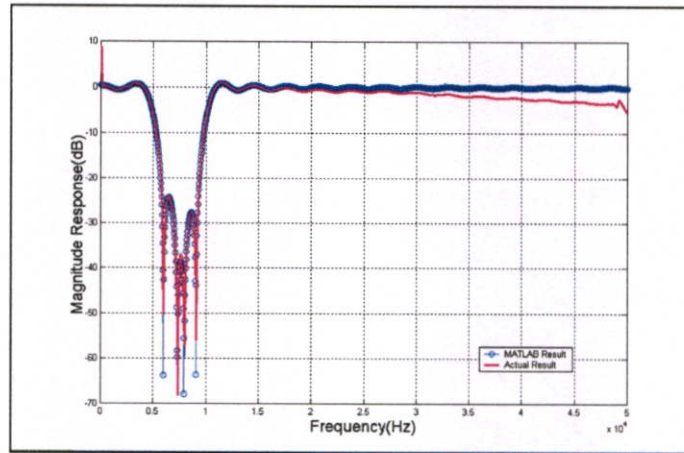
ภาพที่ 4.11ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริง $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65



ภาพที่ 4.12ก ผลตอบสนองทางความถี่แบบ Band-stop ที่มีจุดตัดความถี่ $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 ด้วยโปรแกรม MATLAB



ภาพที่ 4.12ข ผลตอบสนองทางความถี่แบบ Band-stop $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65 จากการวัดจริงด้วยเครื่อง Dynamic Analyzer



ภาพที่ 4.12ค ผลตอบสนองทางความถี่โดยการเปรียบเทียบระหว่างการจำลองการทำงานด้วยโปรแกรม MATLAB กับการสร้างจริง $f_{c1} = 5kHz$ $f_{c2} = 10kHz$ ลำดับ 65

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

จากการทดลองพบว่าตัวกรองเชิงเลข FIR หลายเข้าพุทแบบเวลาจริงที่สร้างขึ้นตามหลักการที่นำเสนอโดยใช้ FPGA สามารถให้เข้าพุทที่มีผลตอบสนองทางขนาดแตกต่างกันในเวลาเดียวกันดังการทดลองในบทที่ 4 โดยผลตอบสนองทางขนาด(Magnitude Response) ของตัวกรองเชิงเลข FIR ที่ความถี่ต่ำจะมีค่าใกล้เคียงกับผลตอบสนองทางขนาดที่จำลองการทำงานด้วยโปรแกรม MATLAB ส่วนที่ความถี่สูง จะมีความผิดพลาดเกิดขึ้น ซึ่งค่าผิดพลาดที่เกิดขึ้นนี้จะอยู่ในรูปของอัตราขยาย (Gain Error) ทั้งนี้สาเหตุเกิดจากวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอกโดย Chip ที่ใช้แปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอกทางด้านเข้าพุทจะมีอัตราการขยายลดลงเมื่อความถี่สูงขึ้น และในส่วนของ FPGA นั้นในการทดลองสร้างตัวกรองเชิงเลข FIR หลายเข้าพุทแบบเวลาจริงลำดับของตัวกรองเท่ากับ 33 จะใช้ทรัพยากรภายในไปประมาณ 32% ขณะที่ ลำดับของตัวกรองเท่ากับ 65 จะใช้ทรัพยากรภายในไปประมาณ 75% และการคำนวณเป็นแบบ Fix-point 16 บิตแบบคิเครื่องหมาย ซึ่งถ้าจะเพิ่มลำดับของตัวกรองก็สามารถกระทำได้อีกเล็กน้อยเท่านั้น

สำหรับการประยุกต์ใช้งาน ตัวกรองเชิงเลข FIR หลายเข้าพุทแบบเวลาจริงนั้น ลักษณะงานจะต้องมีการใช้เข้าพุทหลายๆเข้าพุทในเวลาเดียวกัน ตัวอย่างเช่น วงจร Digital Electronic Crossover [6] หรืออาจจะนำโครงสร้างของตัวกรองเชิงเลข FIR ตามหลักการที่นำเสนอไปสร้างเป็น FIR Filter Chip ทดแทน FIR Filter Chip ที่สามารถให้ผลตอบสนองทางขนาดได้เพียงรูปแบบเดียว

เอกสารอ้างอิง

- [1] J.G. Proakis, D.G. Manolakis. **Digital Signal Processing Principle, Algorithms, and Applications.** New JERSEY : Prentice Hall, 1996.
- [2] Alan V. Oppenheim and Ronald W. Schaffer. **Discrete-Time Signal Processing.** New JERSEY : Prentice Hall, 1999.
- [3] Sanjit K. Mitra. **Digital Signal Processing.** New York : McGraw-Hill, 2001.
- [4] T.W. Parks, C.S. Burrus, **Digital Filter Design,** New York : John Wiley & SONS, INC., 1987.
- [5] C.Sidney Burrus. 1995. "Multiple Least Squares FIR Filter Design.", **IEEE Transactions on Signal Processing.** Feb(vol.43, N0.2): pp. 412-421.
- [6] Sophocles J. Orfanidis. **Signal Processing.** New JERSEY : Prentice Hall . 1996.
- [7] Rulph Chassaing. **Digital Signal Processing Laboratory Experiments Using C and TMS320C31 DSK.** Canada : John Wiley & SONS, INC., 1999.
- [8] Geoff Bostock. **FPGA and Programmable LSI:A Designer's Handbook.** Enland : Clays Ltd,St lves plc., 1996.
- [9] Ruel V. Churchill, Jame Ward Brown. **Complex Variables and Applications.** New York Mcgraw-Hill International Edition, 1990.
- [10] Xinghuo Yu. "Periodic Behaviors in A Digital Filter With Two's Complement Arithmetic.", **ISSPA '99,** Brisbane, Australia, 22-25 Aug., pp.399-402, 1999.
- [11] Lars Wanhammar, **DSP Integrated Circuits.** ACADEMIC PRESS, 1999.
- [12] A.Dawood, Z.Asdani, B.Bravo. "FIR Filter Design and Implementation on Reconfigurable Computing Technology." **ISSPA '99.** 22-25 Aug. : pp.383-386.
- [13] Topdown Design Solutions, Inc. **VHDL.** Topdown Design Solutions, Inc.,1992,1993.
- [14] ชำนาญ ปัญญาใส, วัชรกร หนูทอง. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล. กรุงเทพฯ : บริษัท ซีเอ็ดดูเคชั่น จำกัด(มหาชน).

ภาคผนวก ก
ภาษาวีเอชดีแอล

วีเฮซีแอล คือ อะไร

ภาษาวีเฮซีแอล VHDL หรือ VHSIC (Very High Speed Integrated Circuit) Hardware Description Language ซึ่งเป็นภาษาที่พัฒนาขึ้นโดยกระทรวงกลาโหมของสหรัฐอเมริกาในช่วงปี ค.ศ. 1980 โดยเป้าหมายของโครงการนี้ก็เพื่อพัฒนา ชีตความสามารถในการออกแบบวงจรรวมให้สูงขึ้นและสามารถทำได้ง่ายมากยิ่งขึ้น เป้าหมายของหลักของพัฒนาภาษาวีเฮซีแอล มี 2 ประการ คือ

- เนื่องจากนักออกแบบวงจรรวมมีความต้องการภาษาที่สามารถรองรับการออกแบบวงจรที่มีความซับซ้อน

- นักออกแบบต้องการภาษาที่เป็นมาตรฐานหรือเป็นภาษากลางที่ทำให้สามารถเผยแพร่ผลงานการออกแบบกันภายใน กลุ่มนักออกแบบด้วยกันได้

ในปี 1986 ภาษาวีเฮซีแอล ได้เริ่มมีการปรับปรุงภาษาวีเฮซีแอล เพื่อให้สามารถกำหนดเป็นมาตรฐานของ IEEE โดยสามารถประกาศเป็นมาตรฐานได้ในเดือนธันวาคมปี 1987 โดยอยู่ในหมวด IEEE1076-1987 หลังจากนั้นก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องโดยได้มีการประกาศปรับปรุงอีกครั้งในปี 1993 ซึ่งเรียกว่า IEEE1076-1993 โดยได้มีการเพิ่มเติม Syntax พิเศษเพื่อให้ผู้ใช้สามารถใช้งานได้สะดวกมากยิ่งขึ้น

ข้อดีของภาษาวีเฮซีแอล

- 1 มาตรฐาน (Standard) ภาษาวีเฮซีแอล เป็นมาตรฐานของ IEEE ทำให้มี Tools และ บริษัทที่สนับสนุนการทำงานมากมาย นอกจากนี้วงจรที่ออกแบบโดยวีเฮซีแอล ก็จะใช้งานได้นาน เนื่องจากมีความเข้ากันได้ของภาษากับวงจรที่ได้รับการออกแบบใหม่

- 2 ได้รับการสนับสนุนจากรัฐบาล (Government Support) เนื่องจากวีเฮซีแอล ได้รับการพัฒนาโดยกระทรวงกลาโหมของสหรัฐอเมริกา ดังนั้นการออกแบบวงจรโดยใช้ภาษาวีเฮซีแอล จึงได้รับการสนับสนุนจากรัฐบาลสหรัฐอเมริกา

- 3 ได้รับการสนับสนุนจากอุตสาหกรรม (Industrial Support) เนื่องจากภาษา วีเฮซีแอล เป็นภาษาที่เป็นมาตรฐานของ IEEE จึงมีอุตสาหกรรมจำนวนมากที่รองรับการออกแบบที่ใช้ภาษา วีเฮซีแอล

- 4 มีความยืดหยุ่น การออกแบบโดยใช้ภาษาวีเฮซีแอล สามารถนำไปจำลองการทำงาน หรือสังเคราะห์ด้วยซอฟต์แวร์ตัวใดก็ได้ที่รองรับภาษาวีเฮซีแอล จึงทำให้การออกแบบด้วยภาษา วีเฮซีแอล จึงเป็นการออกแบบที่ไม่ยึดติดกับซอฟต์แวร์ที่ใช้ในการออกแบบ

5 สามารถใช้โมเดลในการออกแบบได้ ผู้ออกแบบวงจรสามารถออกแบบวงจรโดยใช้ภาษาวีเอชดีแอล ได้หลายระดับตั้งแต่ระดับบล็อกวงจรถึงระดับทรานซิสเตอร์ และสามารถออกแบบวงจรที่มีความซับซ้อนสูงและมีขนาดใหญ่มากได้

6 สามารถนำมาใช้ใหม่ได้ วงจรที่ออกแบบโดยภาษาวีเอชดีแอล สามารถนำกลับมาใช้ใหม่ได้ง่าย เนื่องจากสามารถเปลี่ยนแปลงแก้ไขวงจรได้ง่าย

7 สามารถอธิบายการทำงานได้ เป็นภาษาในรูปแบบบรรยายพฤติกรรม ทำให้เราสามารถอธิบายการทำงานของวงจรภายในการออกแบบได้ทันที

ลักษณะการใช้งานภาษาวีเอชดีแอล

การใช้งานภาษาวีเอชดีแอล อาจจำแนกออกเป็น 5 ประเภทดังนี้คือ

1 ภาษาอธิบาย (Document Language) ใช้สำหรับบรรยายรายละเอียดการทำงานของวงจรที่ออกแบบ

2 ภาษาออกแบบ (Design Language) ใช้สำหรับออกแบบวงจรที่มีความซับซ้อนสูงเพื่อใช้ในการจำลองการทำงาน

3 ภาษาตรวจสอบการทำงาน (Verification Language) ใช้ตรวจสอบการทำงานของวงจรที่ออกแบบว่ามีความถูกต้อง

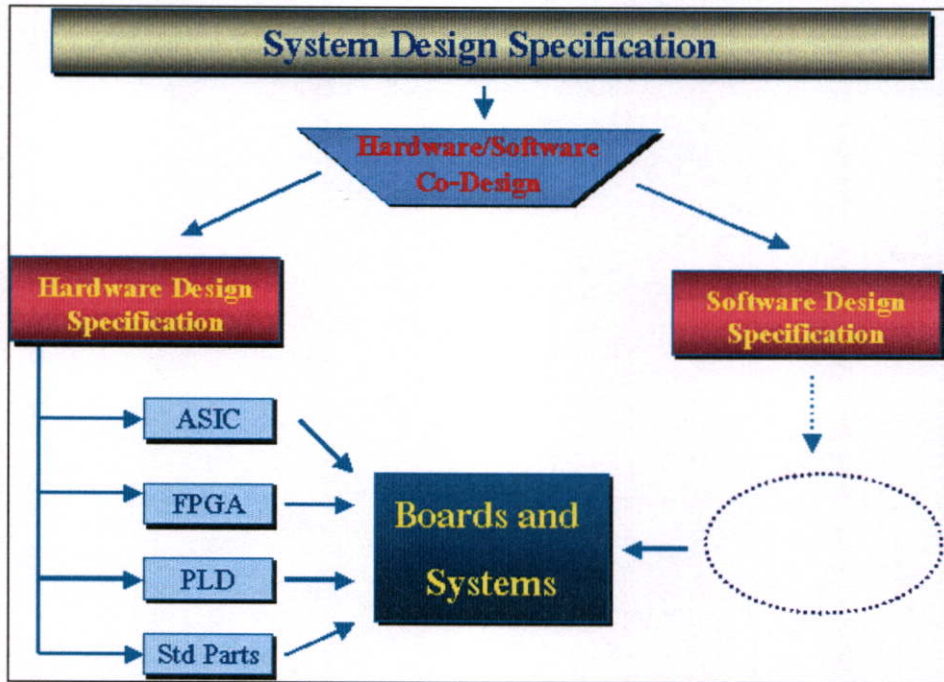
4 ภาษาทดสอบ (Test Language) ใช้สำหรับสร้าง (Test Vector) เพื่อใช้สำหรับเป็นข้อมูลที่ใช้สำหรับทดสอบการทำงานของวงจรที่ออกแบบ

5 ภาษาสังเคราะห์ (Synthesis Language) ใช้สำหรับสร้างวงจรจริงเพื่อนำไปสร้างเป็นชิพอุตสาหกรรมฮาร์ดแวร์ ต่อไป

วิธีการออกแบบระบบดิจิทัล

การออกแบบระบบดิจิทัลจะเริ่มต้นจากการกำหนดหน้าที่การทำงานของระบบอิเล็กทรอนิกส์ที่ต้องการซึ่งโดยส่วนใหญ่แล้ว ระบบอิเล็กทรอนิกส์มักจะประกอบด้วย 2 ส่วนคือ ส่วนที่เป็นฮาร์ดแวร์ และส่วนที่เป็นซอฟต์แวร์ โดยทั่วไปแล้วการออกแบบทั้ง 2 ส่วนดังกล่าวนี้จะต้องทำไปพร้อมๆ กัน หลังจากนั้นจะเป็นขั้นตอนการกำหนดหน้าที่การทำงานของ ฮาร์ดแวร์ และ ซอฟต์แวร์ โดยสำหรับการออกแบบทางด้าน ฮาร์ดแวร์สามารถเลือกได้ว่าต้องการออกแบบเป็น ASIC FPGA PLD หรือจาก อุปกรณ์มาตรฐานอื่น ๆ (Standard Component) ที่มีขายอยู่ทั่วไปตามท้องตลาด สำหรับการออกแบบทางซอฟต์แวร์ผู้ออกแบบสามารถเลือกใช้ภาษา การเขียน โปรแกรม ต่างๆ เช่น ภาษา C/C++ ภาษา Assembly เป็นต้น สำหรับการออกแบบเฟิร์มแวร์ (Firmware) โดยใน

ระหว่างการออกแบบนั้น ผู้ออกแบบทั้งทางด้าน ฮาร์ดแวร์ และ ซอฟต์แวร์ จะต้องมีการทดสอบการทำงานร่วมกันเพื่อทำให้สามารถแก้ไขความผิดพลาดในการออกแบบที่เกิดขึ้นได้



รูปภาคผนวกที่ ก.1 การออกแบบระบบดิจิทัล

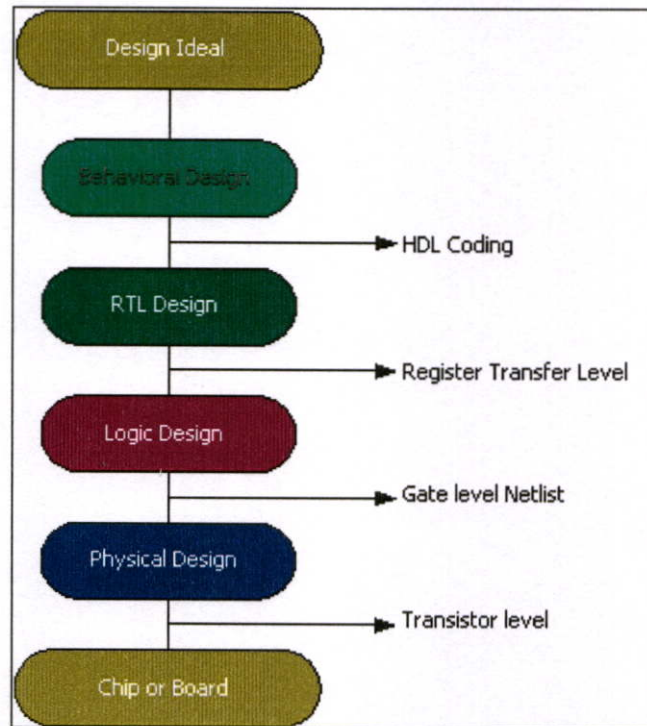
ภาษาวีเอชดีแอลเบื้องต้น

ความซับซ้อนในระบบดิจิทัลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะส่งผลให้มีการคอมพิวเตอร์เพื่อช่วยในการออกแบบมาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการ ออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษายอธิบายอุปกรณ์ฮาร์ดแวร์ (HDL :ฮาร์ดแวร์ Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่อง เพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิทัลเป็นไปอย่างมีประสิทธิภาพ

กระบวนการออกแบบระบบดิจิทัล

ในการออกแบบระบบดิจิทัล เริ่มตั้งแต่การกำหนดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ ที่ใช้งานได้จะต้องผ่านขั้นตอนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้น ก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป รูปภาคผนวกที่ 1 แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิทัลทั่วไป ขั้นแรกผู้ออกแบบจะกำหนดแนวความคิดในการออกแบบแล้วทำการพัฒนาให้สามารถนำมาใช้ได้อย่างสมบูรณ์ ซึ่งภาย

ในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบในเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะ
เป็นผังงานแสดงแบบหรือ รหัสคำสั่งเทียม (Pseudo code) ก็ได้



รูปภาพผนวกที่ ก.2 แสดงขั้นตอนการออกแบบระบบดิจิทัล

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูลผู้ออกแบบจะกำหนดส่วนประกอบ
ของรีจิสเตอร์และวงจรถลอจิก ที่จำเป็นทั้งหมดเพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละ
องค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสองทิศทาง(Unidirectional or Bidirectional Bus)
ส่วนกระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์และวงจรถลอจิกจะขึ้นอยู่กับ
พฤติกรรมของระบบที่ ขั้นตอนถัดมาเป็นการออกแบบวงจรถลอจิก ซึ่งจะเกี่ยวข้องกับการนำเกท
ดิจิทัลพื้นฐานและฟลิปฟลอป (Flip-Flop) มาประกอบเป็นอุปกรณ์ย่อยต่างๆ เช่นรีจิสเตอร์เก็บข้อมูล
บัสวงจรถลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการ
โยงใยระหว่างเกทและ ฟลิปฟลอปนั่นเองการออกแบบในขั้นตอนนี้คือการเปลี่ยนเครือข่าย
การโยงใยที่ได้จากขั้นตอนที่แล้วให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ Layout ซึ่ง
ขั้นตอนนี้จะเกี่ยวข้องโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์เพื่อ แทนเกทและฟ
ลิปฟลอปต่างๆและในขั้นตอนนี้สุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือสารที่โรงงาน
เพื่อผลิตออกมาเป็น หรือการไหลไปใส่ไว้ในอุปกรณ์ประเภท PLD

ประวัติภาษาวีเอชดีแอล

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิทัล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และสามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษาวีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปที่ละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกร ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น วีเอชดีแอล ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิทัลที่มีความซับซ้อนได้ทั้งหมด ดังนั้น วีเอชดีแอล จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง วิวัฒนาการของภาษา วีเอชดีแอล เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกา หรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้น ประกอบกับเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้ จากการนำวงจรดิจิทัลหลายๆ วงจรมาทำการผลิตอยู่บนแผ่นซิลิกอนที่มีพื้นที่เพียง 1 - 2 ตารางเซนติเมตรเท่านั้น ซึ่งเป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือ ในการทำงานและความคงทนต่อสภาพแวดล้อมสูง แต่เนื่องจากในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกร และเวลาในดำเนินการมาก ฉะนั้นทาง DoD จึงจัดตั้งโครงการขึ้นมาเพื่อศึกษาวิธีการที่ช่วยในการพัฒนา วงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรรบบดิจิทัล ให้สามารถนำไปผลิตได้เร็วขึ้น ซึ่งโครงการดังกล่าวมีชื่อว่า (Very High Speed Integrated Circuits) หรือ VHSIC โดยในระยะแรกนั้น โครงการนี้ถือเป็นความลับทาง ด้านความมั่นคงของประเทศ และอยู่ภายใต้ความควบคุมดูแลของ United States International Traffic and Arms Regulations (ITAR) สำหรับมาตรฐานของภาษาที่ใช้บรรยาย พฤติกรรมวงจรหรือฮาร์ดแวร์ของระบบ สำหรับโครงการ VHSIC ที่ DoD ได้ให้ไว้สามารถสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิทัล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่อง คอมพิวเตอร์โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก

- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้

- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรมภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า (Hardware Description Language: HDL)

ในตอนเริ่มแรกนั้น DoD ได้มอบหมายให้บริษัทไอบีเอ็ม เท็กซัสอินสตรูเมนต์ และอินเตอร์เนชันนัล เป็นผู้ศึกษาและพัฒนา โครงการ ซึ่งการดำเนินงานเป็นไปอย่างต่อเนื่อง จนกระทั่งในปี ค.ศ.1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอด เทคโนโลยีทางทหารออกจากโครงการนี้ ดังนั้นภาษา วิสเชดีแอล จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป และประมาณปี ค.ศ. 1987 IEEE ได้ทำการกำหนดมาตรฐานของภาษานี้เป็น IEEE 1076-1987 และมีชื่อเรียกว่า วิสเชดีแอล ซึ่งมาตรฐานนี้ได้รับการปรับปรุงจนเป็นมาตรฐาน IEEE 1076-1993 หรือ VHDL 1993 เนื่องจากในขณะนั้น DoD เป็นลูกค้ารายใหญ่ ของอุตสาหกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ ดังนั้นจึงมีผู้รับโครงการต่างๆ จาก DoD ไปดำเนินการวิจัยและพัฒนา เป็นจำนวนมาก และเพื่อให้ทุกโครงการอยู่ในมาตรฐานเดียวกันหมด ดังนั้นทาง DoD จึงได้กำหนดว่า ทุกๆ โครงการต้อง เขียนอยู่ในรูปของภาษา วิสเชดีแอลเท่านั้น ซึ่งทำให้ DoD สามารถนำโครงการเหล่านี้ไปจำลองกับเครื่องคอมพิวเตอร์ได้หลายระบบ

ข้อกำหนดของภาษาวิสเชดีแอล

DoD ได้ตั้งข้อกำหนดสำหรับภาษา วิสเชดีแอล ในเดือนมกราคมปี ค.ศ.1983 ไว้ดังนี้
ลักษณะทั่วไป

DoD ได้กำหนดให้ วิสเชดีแอล เป็นภาษาสำหรับการออกแบบและบรรยายของฮาร์ดแวร์ ซึ่งหมายถึงความสามารถ ในการอธิบายและออกแบบในระดับสูง การจำลอง (Simulation) การสังเคราะห์ (Synthesis) และการทดสอบ (Testing) นอกจากนี้ วิสเชดีแอล ยังถูกกำหนดไว้สำหรับการบรรยายฮาร์ดแวร์ตั้งแต่ระดับบนซึ่งก็คือระบบจนถึง ระดับเกทอีกด้วย เนื่องจากในการทำงานของระบบดิจิทัลนั้น ทุกๆ องค์ประกอบภายในระบบไม่ว่าเล็กหรือใหญ่ จะทำงานไปพร้อมๆ กัน ซึ่งในเรื่องของความพร้อมเพรียงในการทำงานนี้ก็ถือเป็นข้อกำหนดที่สำคัญอย่างหนึ่งของ วิสเชดีแอล ด้วยเช่นกัน (สำหรับในภาษาที่ใช้ในการบรรยายฮาร์ดแวร์นั้นความพร้อมเพรียงจะหมายถึง ทุกๆ คำสั่ง องค์ประกอบ เกทหรือวงจรถอดจิกจะถูกนำมาปฏิบัติทั้งหมด ดังนั้นในที่สุดแล้วก็จะดูเหมือนว่าได้มีการปฏิบัติไปพร้อมๆ กัน)

สนับสนุนการออกแบบแบบลำดับขั้น

การออกแบบแบบลำดับขั้นเป็นลักษณะที่สำคัญอย่างหนึ่ง สำหรับการออกแบบระบบที่มีหลาย ๆ ระดับ โดยในการ ออกแบบจะประกอบด้วยส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายหน้าที่การทำงาน ซึ่งหน้าที่การทำงาน ของระบบสามารถกำหนดได้ด้วยตัวเอง หรืออาจถูกกำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลง ไปได้เช่นกัน แต่ที่ระดับล่างสุด องค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวเอง และไม่สามารถกำหนด การทำงาน โดยลักษณะแบบโครงสร้างได้

ไลบรารี

วิเศษดีแอล ได้สนับสนุนการมีไลบรารีเพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของ อุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูก ต้องควรจัดเก็บไว้ในไลบรารีหลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้ว เพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไป ใช้ได้ด้วย

ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่งหรือกระบวนการ โดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ วิเศษดีแอล ก็ตาม ตัวภาษายังมีการจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่งไว้ให้ด้วย เมื่อผู้ ออกแบบได้กำหนดหน้าที่และองค์ประกอบ ที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออก แบบยังสามารถบรรยายหน้าที่การทำงาน ซึ่งเป็นรายละเอียดภายในของแต่ละองค์ประกอบได้ใน ลักษณะเดียวกับการเขียน โปรแกรมที่ประกอบด้วย โครงสร้างแบบ case, if - then - else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์กระทำ ได้ สะดวกและง่ายขึ้น อย่างไรก็ตาม โครงสร้างทั้งหมดของ วิเศษดีแอล ก็ยังคงเป็นการทำงานแบบ พร้อมเพรียงกันเช่นเดิม

การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุตและเอาต์พุตแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของ อุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อมและลักษณะทางกายภาพของอุปกรณ์ นั้นๆ ด้วย ซึ่งภาษาสำหรับการออกแบบที่ดีควร ให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลาไหล และเงื่อนไขทางสภาพแวดล้อม อื่นๆ ซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา วิเศษดีแอล ด้วยเช่น กัน

ชนิดของข้อมูล

วิเศษดีแอล สามารถกำหนดชนิดของข้อมูลไม่เพียงแต่ชนิดบิต (BIT) และบูลีน (BOOLEAN) เท่านั้น แต่ยังสามารถกำหนดชนิด ของข้อมูลเป็นจำนวนเต็ม จำนวนจริง จุดทศนิยม และชนิดลำดับ การนับ (Enumerate Type) หรือแม้แต่ชนิดของ ข้อมูลที่ผู้ออกแบบกำหนดขึ้นมาเองก็ได้

โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชันและ โพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน วิเศษดีแอล ซึ่งผู้ออกแบบ สามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การ กำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการ ได้เช่นเดียวกับการเขียนโปรแกรมทั่วไป

การควบคุมเวลา

วีเฮดดีแอล อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการ การตรวจสอบ การออกแบบเกทหรือการห้วงเวลาก็สามารถกระทำได้โดยการกำหนดช่วงเวลาที่แน่นอนหรือกำหนดให้มีการรอคอย เหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบ โดยการกำหนด โครงสร้างขององค์ประกอบรวมที่เกิดจากองค์ประกอบย่อยซึ่งแตกต่างกันหรือเหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ วีเฮดดีแอล เช่นกัน

ส่วนประกอบต่างๆ ของภาษาวีเฮดดีแอล

ในการเขียนรูปแบบบรรยายระบบดิจิทัลในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของ โครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเฮดดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

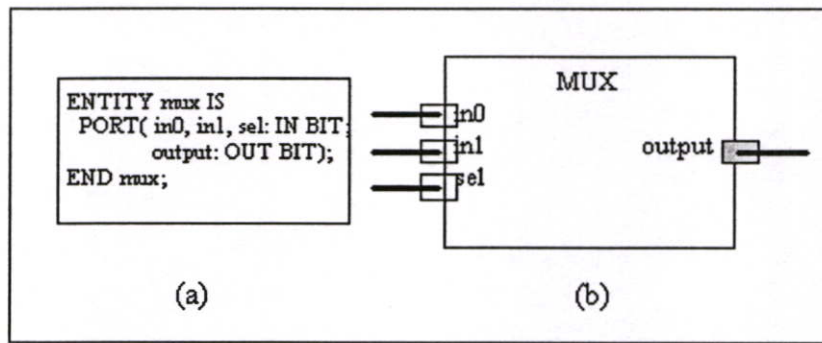
- หน่วยการออกแบบเอนทิตี (Entity Design Unit)
- หน่วยการออกแบบสถาปัตยกรรม (Architecture Design Unit)
- หน่วยการออกแบบแพ็คเกจ (Package Design Unit)
- หน่วยการออกแบบ โครงแบบ (Configuration Design Unit)

หน่วยการออกแบบเอนทิตี หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้น ที่เรียกว่า หน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น ภาพที่ ก-3 แสดงให้เห็น โครงสร้างอย่างง่าย ๆ ของ หน่วยการออกแบบเอนทิตี

```
ENTITY component_name IS
    Input and output ports
    Physical and other parameters
END [component_name];
```

รูปภาคผนวกที่ ก.3 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

ส่วนนี้จะขึ้นต้นด้วยคำ ENTITY และ IS ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component_name) สำหรับการตั้งชื่อนั้นต้องเป็นไปตามกฎเกณฑ์ของภาษาหลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (input-output) รวมทั้งพารามิเตอร์อื่นๆ ส่วนนี้เรียกว่าส่วนหัว (entity header) และที่สำคัญคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า END และเครื่องหมายอัฒภาคเสมอ (;)



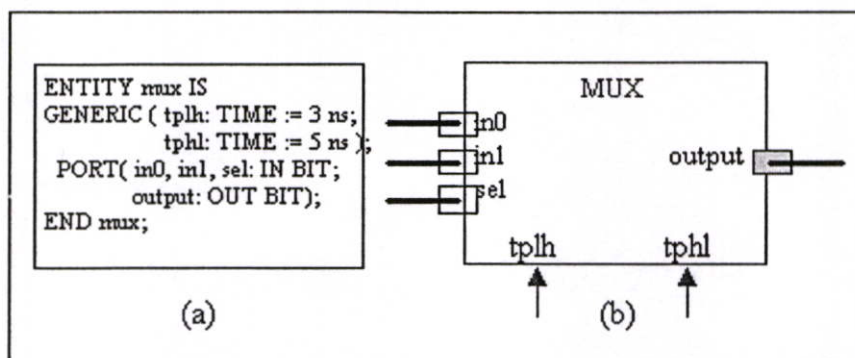
รูปภาคผนวกที่ ก.4 แสดงรูปแบบของมัลติเพลกซ์

(a) หน่วยการออกแบบเอนทิตีในรูปแบบของวีเอชดีแอล

(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

ในรูปรูปภาคผนวกที่ 4 เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพลกซ์ หรือ MUX ในส่วนหัวของเอนทิตี มีการกำหนดจุดต่อ 4 จุดภายใต้ชุดคำสั่ง PORT โดยที่ 3 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ in0, in1, sel ซึ่งกำหนดด้วยทิศทางารติดต่อกับภายนอกเป็นการไหลเข้าของข้อมูล (IN) ที่แสดงด้วยรูปสี่เหลี่ยมโปร่งในรูปรูปภาคผนวกที่ 4 ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ซึ่งกำหนดด้วยทิศทางารติดต่อกับภายนอกเป็นการไหลออก (OUT) ที่แสดงด้วยรูปสี่เหลี่ยมในภาพที่ ก-4 ส่วนประเภทของข้อมูลที่จะไหล เข้าและออก นั้นเป็นประเภท BIT ที่สามารถมีค่าได้เพียงสองค่าคือ '0' และ '1' เท่านั้น

นอกจากนั้นผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ ที่เป็นข้อมูลเพิ่มเติมอื่นๆ ลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์ อัน ได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างในรูปรูปภาคผนวกที่ ก.5

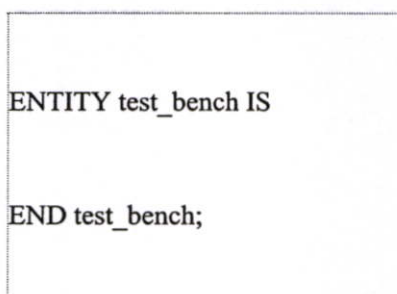


รูปภาคผนวกที่ ก.5 รูปแบบมัลติเพล็กซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย

(a) หน่วยการออกแบบเอนทิตีในรูปแบบของวีเอชดีแอล

(b) มุมมองของตัวเชื่อมประสาน

ในบางกรณีสามารถใช้ภาษาวีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหลเข้าและออกของข้อมูล ได้ ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบ สำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่ง คือ วีเอชดีแอลสำหรับการทดสอบเปรียบเทียบ (Test bench) รูปภาคผนวกที่ ก.6



รูปภาคผนวกที่ ก.6 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

หน่วยการออกแบบสถาปัตยกรรม คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงาน พฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดใน หน่วยการออกแบบเอนทิตี รูปภาคผนวกที่ 7 แสดงให้เห็นถึงโครงสร้างอย่างง่าย ๆ ของหน่วยการออกแบบสถาปัตยกรรม

```

ARCHITECTURE identifier OF component_name IS
    [declaration]
BEGIN
    specification of the functionality of the
    component in terms of its input lines and as
    influenced by physical and other parameters
END [identifier];

```

รูปภาคผนวกที่ ก.7 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (Identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OF <entity design unit> IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture Declarative Area) ที่เป็นเพียงส่วนเพื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่นประเภท (Type) ต่างๆ (ตัวอย่างเช่น Bit, bit_vector) สัญญาณ (Signal) ตัวคงที่ (Constant) โปรแกรมย่อย (ได้แก่ Function และ Procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า และไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง PORT) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า BEGIN กับ END ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (Concurrent Statement) เท่านั้น หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้นๆ ที่เป็นส่วนเพื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิทัลด้วยภาษาวีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

- ประเภทการไหลของข้อมูล (Dataflow description)
- ประเภทพฤติกรรม (Behavioral description)
- ประเภทโครงสร้าง (Structure description)
- ประเภทผสม (Mixed model description)

```

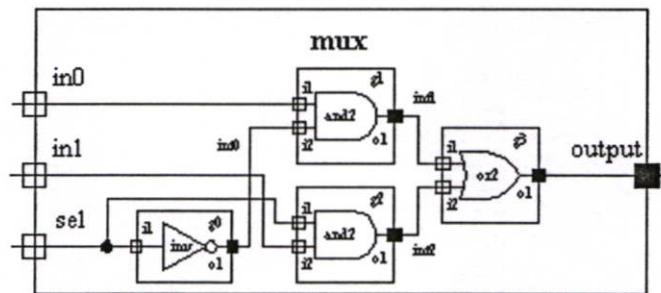
ARCHITECTURE data_flow OF mux IS
BEGIN
    output <= ((NOT sel) AND in0) OR (sel AND in1);
END data_flow;

```

รูปภาคผนวกที่ ก.8 แสดงหน่วยการออกแบบสถาปัตยกรรมของ
มัลติเพลกซ์ตามฟังก์ชันบูลีน

$$Output = (\overline{sel}.in0) + (sel.in1) \quad (ก-1)$$

จากรูปภาคผนวกที่ ก.8 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า ($in0, in1$) กับข้อมูลที่ไหลออก ($Output$) ประกอบด้วยชุดคำสั่งแบบแข่งขันานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพลกซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL: Register Transfer Level)



รูปภาคผนวกที่ ก.9 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์

จากรูปภาคผนวกที่ ก.9 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง โดยใช้ อินเวอร์เตอร์ (inv ที่ตำแหน่ง g0) แอนด์เกต 2 อินพุตจำนวน 2 ตัว (and2 ที่ตำแหน่ง g1 และ g2) และ ออร์เกต 2 อินพุต (or2 ที่ตำแหน่ง g3) มาสร้างตามฟังก์ชันบูลีนของรูปภาคผนวกที่ ก.10

```

ARCHITECTURE struc OF mux IS
    COMPONENT inv
    PORT ( i1 : IN BIT ; o1 : OUT BIT );
    COMPONENT and2
    PORT ( i1, i2 : IN BIT ; o3 : OUT BIT );
    COMPONENT or2
    PORT ( i1, i2 : IN BIT ; o1 : OUT BIT );
END COMPONENT;
    SIGNAL int0, int1, int2 : BIT;
BEGIN
    g0 : inv  PORT MAP (i1 => sel, o1 => int0);
    g1 : and2 PORT MAP (i1 => in0, i2 => int0, o1 => int1);
    g3 : or2  PORT MAP (i1 => int1, i2 => int2, o1 => ouput);
END struc;

```

รูปภาคผนวกที่ ก.10 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง

```

ARCHITECTURE behav OF mux IS
BEGIN
    PROCESS (in0, in1, sel)
    BEGIN
        IF (sel = '0') THEN output <= in0;
        ELSE output <= in1;
        END IF;
    END PROCESS;
END behav;

```

รูปภาคผนวกที่ ก.11 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพลกซ์ในลักษณะของ ประเภทพฤติกรรม ประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งนี่ก็เป็นข้อดีของภาษาวีเอชดีแอล

หน่วยการออกแบบแพ็คเกจ

ข้อมูลต่างๆตลอดจน โปรแกรมย่อยที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิทัล สามารถเก็บไว้ในส่วนของแพ็คเกจได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดย หน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือ จากหน่วยการออกแบบแพ็คเกจอื่นๆ นอกจากนั้นสิ่งที่น่าสนใจมากคือรูปแบบมาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล 74XX เป็นต้น) จะถูกเก็บไว้ในแพ็คเกจ ที่ทุกคนสามารถเข้าถึง โดยปกติแล้ว แพ็คเกจจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็คเกจ (Package Declaration) และ ส่วนของบอดีแพ็คเกจ (Package Body) เนื่องจาก แพ็คเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้น การที่นำแพ็คเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอล สามารถกระทำได้ด้วยชุดคำสั่ง USE

การประกาศแพ็คเกจ (Package Declaration)

ส่วนที่มีความสำคัญที่สุดของแพ็คเกจ ได้แก่ ส่วนการประกาศแพ็คเกจ เพราะจะเป็นส่วนที่กำหนดชื่อ ของสิ่งที่ประกาศอยู่ในแพ็คเกจ สำหรับนำไปใช้ภายนอกตัวของแพ็คเกจเอง สิ่งใดๆ ถูกประกาศในส่วนของ ส่วนบอดีแพ็คเกจ แต่ไม่ถูกประกาศในส่วนการประกาศแพ็คเกจ จะไม่สามารถถูกนำค่า และพฤติกรรม ไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือ จุดเชื่อมต่อ หรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้น โดยทั่วไปแล้ว แพ็คเกจ สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศ ชนิด (Type) หรือ สัญญาณ เช่นเดียวกันกับ ส่วนบอดีแพ็คเกจ ที่ไม่จำเป็นต้องมี ส่วนของการประกาศแพ็คเกจ แต่แพ็คเกจ นั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
PACKAGE package_name IS
    Package_declarative_part
END package_name;
```

รูปภาคผนวกที่ ก.12 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็คเกจ

โครงสร้างส่วนบอดี (Package Body) คือ โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็คเกจ แล้วจะถูกเก็บไว้ในส่วน

บอดีแพ็คเกจ ทั้งนี้รวมทั้ง การกำหนดค่าคงที่ต่างๆ อันได้แก่ตัวคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็คเกจ แต่ถูกกำหนดค่าในส่วนของบอดีแพ็คเกจ ฉะนั้นส่วนบอดีแพ็คเกจ จึงไม่จำเป็นต้องมี ถ้าในส่วนของ การประกาศแพ็คเกจ ไม่มีการประกาศชื่อ ที่เป็นโปรแกรมย่อย หรือค่าคงที่ การเขียนบอดีแพ็คเกจนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปภาคผนวกที่ ก.13

```
PACKAGE BODY package_name IS
    declarative part
END package_name;
```

รูปภาคผนวกที่ ก.13 โครงสร้างของบอดีแพ็คเกจ

หน่วยการออกแบบโครงแบบ

ระบบดิจิทัลจะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตี หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

```
CONFIGURATION identifier OF entity_name IS
    Configuration_declarative_part
END ;
```

รูปภาคผนวกที่ ก.14 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Sukanya Praesombool, Phayung Desyoo, Surapun Yimman, Anushit Thoopluang and Kobchai Dejhan, “**RealTime Multiple Output FIR Filter Design.**” 2005 RISP International Workshop on Nonlinear Circuit and Signal Processing (NCSP’05), pp. 45-48, Hawaii, USA, Mar. 4-6, 2005.



NCSP'05

Proceedings of the
2005 RISP International Workshop
on Nonlinear Circuits and Signal Processing

Waikiki Beach Marriott Resort
Honolulu, Hawaii, USA
March 4 - 6, 2005

Organizer:

The Research Institute of Signal Processing Japan (RISP)

Printed in Japan. © Research Institute of Signal Processing, Japan
office@risp.jp <http://www.risp.jp/>

Real Time Multiple Output FIR Filter Design

Sukanya Praesomboon^{**}, Phayung Desyoo^{*}, Surapun Yimman^{*},
Anushit Thooplung^{***} and Kobchai Dejhan^{**}

^{*} Department of Industrial Physic & Medical Instrumentation,
King Mongkut's Institute of Technology North Bangkok, Bangkok, 10800, Thailand.
Tel: +66-2-913-2500; E-mail: sym_imi@yahoo.com

^{**} Faculty of Engineering and Research Center for Communication and Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand.

^{***} TOT Corporation Public Company Limited, Bangkok, 10210, Thailand.

Abstract

This paper propose a new design of real time multiple output FIR filter based on FPGA. This FIR filter uses improved direct form of FIR filter to create a filter with four outputs with the same number of delay as the single output versions. This change creates the new FIR filter structure that generates four forms of magnitude response at the real time such as low-pass, high-pass, band-pass and band-stop with linear phase response. The filter design and simulation are executed in MATLAB and implemented on the APEX DSP FPGA board. The experiments on the APEX DSP FPGA board show that frequency responses of this FIR filter are quite accurate compared to the theoretical results.

1. Introduction

Digital signal processing, especially in the field of FPGA is one of the fastest developed area of electronic technology. Rather than fitting the algorithm to the architecture, FPGA fits the architecture to the algorithm and makes it the ideal platform for DSP implementation. It combines the re-programmability, architectural flexibility and customizability that allow the system designers to meet any processing purposes. Recently, hundreds of FPGA applications have been introduced in many engineering fields such as digital control, data compression, image processing, signal processing, biomedical signal processing and etc. FIR Filter is the high stability digital filter because the positions of its poles are on the origin. Moreover, the linear phase response of the FIR filter is suitable for wide-area applications. However, the direct form design of real time multiple output FIR filter is required so many gate elements (delay, adder and multiply elements) in the FPGA chip [1,2].

There are two different cutoff frequencies of the FIR low-pass filters at the initial step. Then, these two FIR low-pass filters will be restructured by using the shared elements (delay, adder, and multiply elements) to create

the new structures with four outputs. This four different frequency response outputs, with linear phase, are simultaneously generated in real time. The filter orders are unlimited by theory but are determined by the power of processors.

2. Theory

2.1. Previous design

FIR digital filter has a transfer function as shown in (1) and output equation as shown in (2). The researchers can derive the direct form of FIR filter from (2) as shown in Fig.1.

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

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad (2)$$

The calculation of the filter coefficients and the design of FIR filter will be done by applying Fourier transform [3-5].

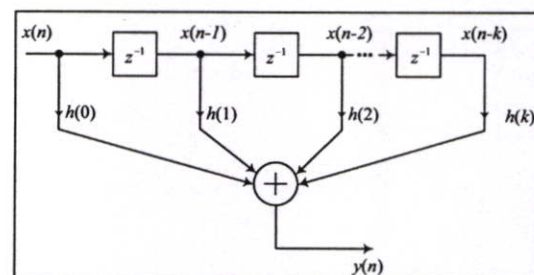


Fig. 1 Realizations of Direct form FIR Filter $h(k)$ is Impulse Response or FIR filter coefficients

After taking the Direct form of FIR filter structure as shown in Fig. 1, the result show that there will be a single form of output either lowpass, highpass, bandpass or bandstop filter, but the structure cannot come up with FIR filter with multiple forms at the same time.

2.2 Proposed design

The new design of multiple frequency response FIR filter will start from FIR Low-pass Filter with 2 different cutoff frequencies as shown in (3) and (4).

$$y_{LP1}(n) = \sum_{k=0}^{M-1} h_1(k)x(n-k) \quad (3)$$

$$y_{LP2}(n) = \sum_{k=0}^{M-1} h_2(k)x(n-k) \quad (4)$$

$h_1(k)$ = Filter coefficients of FIR Low-pass Filter the Cut-off frequency f_1

$h_2(k)$ = Filter coefficients of FIR Low-pass Filter at the Cut-off frequency f_2

If the FIR filter coefficients have been designated in the form of odd positive symmetry, expression (3) and (4) will be transformed into (5) and (6).

$$y_{LP1}(n) = h_1(0)x(n) + h_1(1)x(n-1) + \dots + h_1\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) + \dots + h_1(k)x(n-k) \quad (5)$$

$$y_{LP2}(n) = h_2(0)x(n) + h_2(1)x(n-1) + \dots + h_2\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) + \dots + h_2(k)x(n-k) \quad (6)$$

The output of FIR highpass filter at the cutoff frequency f_1 will be shown in (7).

$$y_{HP1}(n) = -h_1(0)x(n) - h_1(1)x(n-1) - \dots - \left[-1 + h_1\left(\frac{k}{2}\right)\right]x\left(n-\frac{k}{2}\right) - \dots - h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - h_1(0)x(n) - h_1(1)x(n-1) - \dots - h_1\left(\frac{k}{2}\right)x\left(n-\frac{k}{2}\right) - \dots - h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - \sum_{k=0}^{M-1} h_1(k)x(n-k)$$

$$y_{HP1}(n) = x\left(n-\frac{k}{2}\right) - y_{LP1}(n) \quad (7)$$

The output equation of FIR highpass filter with cutoff frequency f_2 can be found in the same way as the output equation for FIR highpass filter with cutoff frequency f_1 as shown in (8).

$$y_{HP2}(n) = x\left(n-\frac{k}{2}\right) - y_{LP2}(n) \quad (8)$$

The output of FIR bandpass filter will be shown in (9).

$$y_{BP}(n) = (h_2(0) - h_1(0))x(n) + (h_2(1) - h_1(1))x(n-1) + \dots + (h_2(k) - h_1(k))x(n-k)$$

$$y_{BP}(n) = \sum_{k=0}^{M-1} h_2(k)x(n-k) - \sum_{k=0}^{M-1} h_1(k)x(n-k)$$

$$y_{BP}(n) = y_{LP2}(n) - y_{LP1}(n) \quad (9)$$

The equation for the output of FIR Band-pass Filter and Band-stop Filter will be shown in (10)

Notice that the expression (3) to (10) which is the non-recursive discrete equations have shared the same variables $x(n)$, $x(n-1)$, ..., $x(n-k/2)$, ..., $x(n-k)$ without using the previous output $y(n-k)$ as a part of filter. After that, the new design of structure for FIR digital filters will be derived from (3) to (10) to come up with the multiple forms of frequency responses at the same time as shown in Fig. 2.

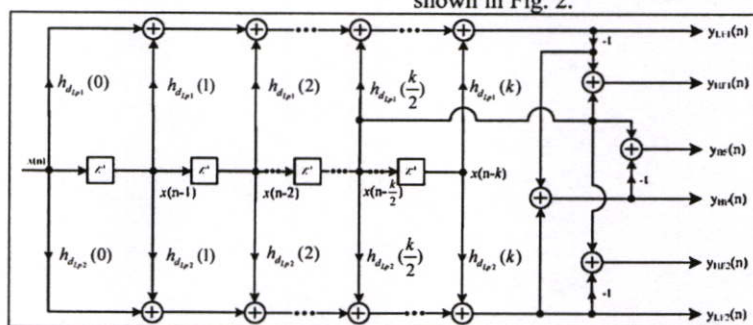


Fig. 2 FIR Digital filter structure with multiple from of frequency responses at the same time

3. Design and Implementation

The APEX DSP development board with 8320 logic is used for FIR filter which simultaneously produces multiple frequency responses. This APEX DSP board is implemented with program which generated from Quartus II limited edition. The diagram of FPGA as shown in Fig.4 and designed to be consistent with diagram in Fig.2. The diagram of FPGA system consisted with control unit,

memory array, multiplex, multiply and adder unit, latch element, adder element, A/D, D/A. The processing of FPGA system is performed in 16 bit sign fix point using 4 state or 4 clock pulse per cycle as shown in Fig.5 [6, 7]. The detail of each state is shown in Fig 3. The design examples have specified the two cut-off frequency of 41-order FIR lowpass filter at 1.0 kHz and 1.5 kHz with the sampling at 8 kHz.

4. Experiment Result

The experiments will simulate the filter response by using MATLAB commands while the actual implementation will be executed on APEX DSP FPGA board the measurements from Dynamic signal analyzer HP 35670A to shown magnitude response in Figs. 6-11.

5. Conclusions

The experiment results have shown that the FIR filter design according the introduced structure will provide the frequency response in 4 forms at the same time. Furthermore, the actual implementations on APEX DSP FPGA board have comparing with the simulated result from MATLAB. After investigating the filter structure in details, it was found that the 4 filter structures have shared the adder elements, multiplier elements and delay elements which cut down the number of elements comparing to the separated and independent versions with direct form. This design will be very useful for the construction FIR filter chip for the applications such as adaptive filter and electronic crossover [8, 9] and so on.

References

- [1] X. Yu, "Periodic behaviors in a digital filter with two's complement arithmetic," Proc. of ISSPA '99, pp. 399-402, Australia, August 1999.
- [2] A. Dawood, Z. Asdani and B. Bravo, "FIR Filter Design and Implementation on Reconfigurable Computing Technology," Proc. of ISSPA '99, pp. 383-386, Australia August 1999.
- [3] T. W. Park, C. S. Burrus, *Digital Filter Design*, John Wiley & Son, Inc., 1987.
- [4] J.G. Proakis, D.G. Manolakis, *Digital Signal Processing Principle, Algorithms, and Applications*, Prentice-Hall, 1996.
- [5] A. V. Oppenheim, R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall Inc., 1998.
- [6] MathWorks, *Fixed-Point Blockset for Use with SIMULINK*, The MathWorks, 2001.
- [7] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, 2000.
- [8] Sophocles J. Orfanidis, *Introduction to Signal Processing*, Prentice-Hall, 1996.
- [9] R. H. Yang, S. B. Chiah and W. Y. Chan, "Design and Implementation of a Digital Audio Tone Control Unit Using an Efficient FIR Filter Structure," Proc. of 1996 IEEE TENCON, pp.273-277, 1996.

<p>State 1 : Reset D F/F(1) , D F/F(2) get $x(0)$</p> $O_{LP1} = h_{d_{LP1}}(0)x(0) + h_{d_{LP1}}(1)x(1) + \dots + h_{d_{LP1}}(15)x(15)$ $g_{LP1}(1) = 0 + O_{LP1}$ $O_{LP2} = h_{d_{LP2}}(0)x(0) + h_{d_{LP2}}(1)x(1) + \dots + h_{d_{LP2}}(15)x(15)$ $g_{LP2}(1) = 0 + O_{LP2}$ <p>State 2 :</p> $O_{LP1} = h_{d_{LP1}}(16)x(16) + h_{d_{LP1}}(17)x(17) + \dots + h_{d_{LP1}}(31)x(31)$ $g_{LP1}(2) = O_{LP1} + g_{LP1}(1)$ $O_{LP2} = h_{d_{LP2}}(16)x(16) + h_{d_{LP2}}(17)x(17) + \dots + h_{d_{LP2}}(31)x(31)$ $g_{LP2}(2) = O_{LP2} + g_{LP2}(1)$	<p>State 3 :</p> $O_{LP1} = h_{d_{LP1}}(32)x(32) + h_{d_{LP1}}(33)x(33) + \dots + h_{d_{LP1}}(47)x(47)$ $g_{LP1}(3) = O_{LP1} + g_{LP1}(2)$ $O_{LP2} = h_{d_{LP2}}(32)x(32) + h_{d_{LP2}}(33)x(33) + \dots + h_{d_{LP2}}(47)x(47)$ $g_{LP2}(3) = O_{LP2} + g_{LP2}(2)$ <p>State 4 :</p> $y_{LP1}(n) = g_{LP1}(3)$ $y_{LP2}(n) = g_{LP2}(3)$ $x(47) = x(46)$ $x(46) = x(45)$ $x(45) = x(44)$ \vdots $x(1) = x(0)$
---	--

Fig.3 Process of Real Time Multiple Output FIR Filter

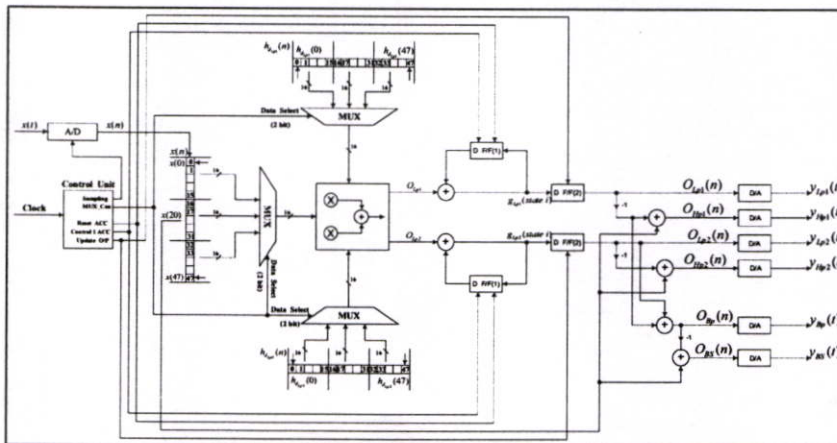


Fig.4 Diagram of FPGA

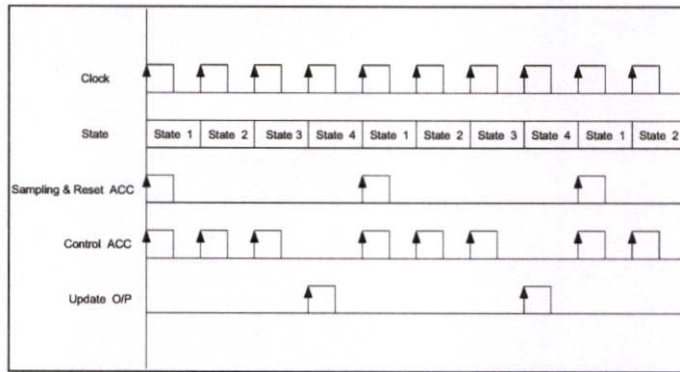


Fig.5 Timing Diagram of Real Time Multiple Output FIR Filter

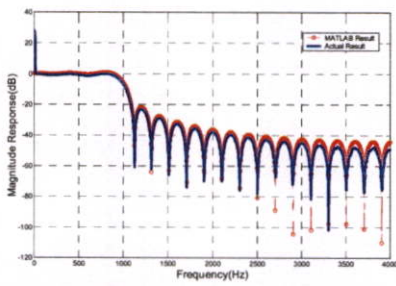


Fig. 6 Low-pass Filter $f_{C1} = 1kHz$

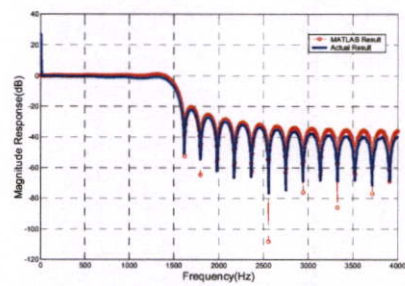


Fig. 7 Low-pass Filter $f_{C2} = 1.5kHz$

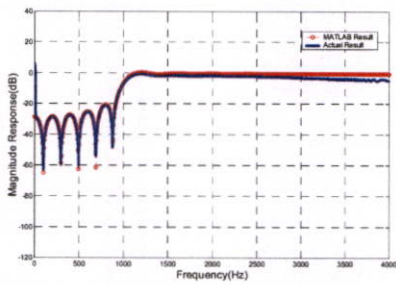


Fig. 8 High-pass Filter $f_{C1} = 1kHz$

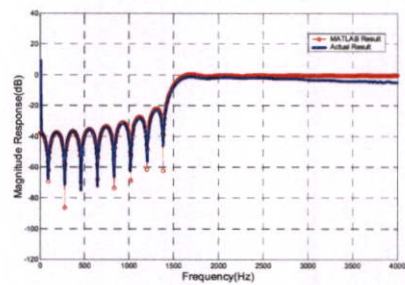


Fig. 9 High-pass Filter $f_{C2} = 1.5kHz$

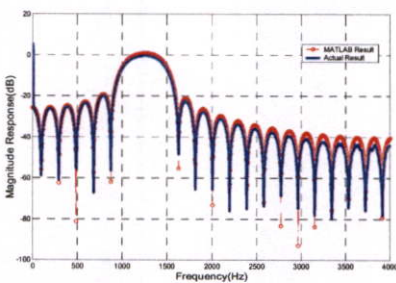


Fig. 10 Band-pass Filter $f_L = 1kHz$, $f_H = 1.5kHz$

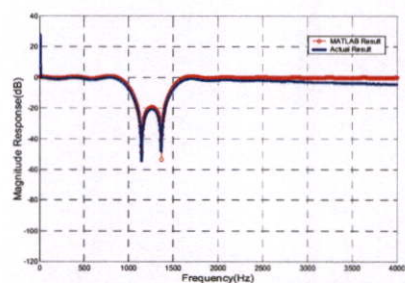


Fig. 11 Band-stop Filter $f_L = 1kHz$, $f_H = 1.5kHz$

ประวัติผู้เขียน

นางสาวสุกัญญา แพรสมบูรณ์ เกิดเมื่อวันที่ 27 กุมภาพันธ์ พ.ศ. 2524 ที่จังหวัดชุมพร สำเร็จการศึกษา วิทยาศาสตร์บัณฑิต สาขาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2545

ปี พ.ศ. 2548 ได้บรรจุเป็นพนักงานมหาวิทยาลัย ตำแหน่ง นักวิทยาศาสตร์ สังกัดภาควิชา ฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์