การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วน ชนิดลากรานจ์และการประยุกต์ใช้งาน

DESIGN OF LAGRANCE – TYPE VARIABLE FRACTIONAL DELAY FILTER AND ITS APPLICATION

โดย

นางสาวชลธิชา อิทธินราเดช

นายไชโย พร้อมพรชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วน ชนิดลากรานจ์และการประยุกต์ใช้งาน

DESIGN OF LAGRANGE - TYPE VARIABLE FRACTIONAL DELAY FILTER AND ITS APPLICATION

โดย

นางสาว ชลธิชา อิทธินราเดช

นาย ไชโย พร้อมพรชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานต์และการประยุกต์ใช้งาน

DESIGN OF LAGRANGE-TYPE VARIABLE FRACTIONAL DELAY FILTER AND ITS APPLICATION

ผู้จัดทำ

1. นางสาว ชฎธิชา อิทธินราเดช 53010320
2. นาย ไชโย พร้อมพรชัย 53010394

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาตราจารย์ ดร. ศรวัฒน์ ชิวปรีชา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาของศาสตราจารย์ ดร.ศรวัฒน์ ชิวปรีชา อาจารย์ที่ปรึกษา ผู้คอยดูแลเอาใจใส่และเมตตาแก่ศิษย์ด้วยดีเสมอมา คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาเป็นอย่างยิ่งและขอบพระคุณอย่างสูง

ขอบคุณห้องปฏิบัติการสำหรับสถานที่ เครื่องมือ และอุปกรณ์ในการทำโครงงานฉบับนี้ ขอบพระคุณอาจารย์ทุกๆ ท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆ ให้ทั้งทางตรงและทางอ้อม ตลอดจนกำลังใจ คำแนะนำ เกี่ยวกับปัญหาในการทำโครงงานต่างๆ และความช่วยเหลือในการทำโครงงานฉบับนี้

สุดท้ายนี้ขอกรอบขอบพระคุณบิดา มารดา ที่ให้ความสำคัญกับการศึกษาของบุตร และให้การสนับสนุนเอาใจใส่ดูแลด้วยดีเสมอมารวมทั้งกำลังใจที่หาเปรียบมิได้

นางสาว ชลธิชา อิทธินราเดช
นาย ไชโย พร้อมพรชัย
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานต์และการประยุกต์ใช้งาน
DESIGN OF LAGRANGE-TYPE VARIABLE FRACTIONAL DELAY FILTER AND ITS APPLICATION

โดย นางสาว ชลธิชา อิทธินราเดช 53010320
นาย ไชโย พร้อมพรชัย 53010394

อาจารย์ที่ปรึกษา ผู้ช่วยศาตราจารย์ ดร. ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

ปริญญานิพนธ์เรื่องนี้เป็นการศึกษาและออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ (Lagrange - Type Variable Fractional Delay Filter : VFD Filter) เพื่ออธิบายหลักการออกแบบระบบ และการนำมาประยุกต์ใช้งาน โดยการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์จะนำเสนอในโครงสร้างแบบแฟร์โรว์ และโครงสร้างแบบแฟร์โรว์ดัดแปลง จากนั้นจะทำการประยุกต์ใช้งานเกี่ยวการประมาณค่าในช่วงของสัญญาณต่างๆ ซึ่งจะทำให้องค์ประกอบของสัญญาณนั้นมีความละเอียดมากยิ่งขึ้น โดยจะใช้โปรแกรม MATLAB ในการออกแบบและจำลองการทำงานของระบบ และใช้ภาษา VHDL ในการจำลอง ออกแบบวงจรสังเคราะห์ และสร้างฮาร์ดแวร์ลงบนอุปกรณ์ FPGA

ABSTRACT

This project presents the study and design of Lagrange - Type Variable Fractional Delay (VFD) Filter. Design procedure of Lagrange – Type VFD Filter and its application will be explained. The Farrow structure and modified Farrow structure are used for VFD filter structure realization. Application of the proposed VFD filter is signal interpolation in order to increase the number of data points in discrete - time signal. MALAB is used for design of VFD filter and simulation. Finally, the hardware implementation of VFD filter on FPGA will be deployed using VHDL.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		VII
สารบัญตาราง		XII
บทที่ 1	**บทนำ**	1
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	3
	1.3 ขอบเขตของปริญญานิพนธ์	3
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	4
	2.1 วงจรกรองสัญญาณเชิงเลข (Digital Filter)	4
	2.2 วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์	8
	2.2.1 โครงสร้างแบบแฟร์โรว์สำหรับวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนลากรานจ์ (Farrow Structure for Lagrange - Type VFD Filter)	9
	2.2.2 โครงสร้างแบบแฟร์โรว์ดัดแปลงสำหรับวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนลากรานจ์ (Modified Farrow Structure for Lagrange - Type VFD Filter)	13
	2.3 การเขียนภาษา VHDL	17
	2.3.1 Terminology และ Convention	18
	2.3.2 การออกแบบจากบนลงล่าง	21
	2.3.3 ภาษา VHDL และส่วนประกอบต่างๆของภาษา	22
	2.3.3.1 หน่วยการออกแบบเอนทิตี้	23
	2.3.3.2 หน่วยการออกแบบสถาปัตยกรรม	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.3.3 หน่วยการออกแบบแพ็กเก็ต	25
2.3.3.4 หน่วยการออกแบบโครงแบบ	26
2.3.4 ชุดคำสั่งลำดับ (Sequential Statements)	27
2.3.4.1 Process Statement	28
2.3.4.2 Wait Statement	28
2.3.4.3 IF-THEN-ELSE statement	29
2.3.4.4 CASE statement	30
2.3.5 ชุดคำสั่งแบบแข่งขนาด (Concurrent Statement)	30
2.4 FPGA ((Field Programmable Gate Array)	30
2.4.1 การออกแบบวงจรด้วยอุปกรณ์ FPGA	31
2.4.2 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์	32
2.4.2.1 การจำลองการทำงานของวงจร (Simulation)	33
2.4.2.2 การสังเคราะห์วงจร	33
2.4.2.3 การแบ่งวงจร (Partitioning)	33
2.4.2.4 การวางอุปกรณ์ (Placement)	34
2.4.2.5 การเชื่อมต่อสัญญาณ (Routing)	34
2.4.2.6 การโปรแกรมอุปกรณ์ FPGA (Configuration)	34
2.4.3 FPGA Discovery-III XC3S200	34
2.5 การประยุกต์ใช้งาน	35
2.5.1 การเข้าจังหวะในโมเด็มสัญญาณเชิงเลข (Synchronization in digital modem)	35
2.5.2 วงจรแปลงอัตราการชักตัวอย่าง (Sampling rate conversion: SRC)	36
2.5.3 การทำ Oversampling D/A	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	39
	3.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์	39
	3.1.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 2 ในโครงสร้างแบบแฟร์โรว์	39
	3.1.2 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 2 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง	40
	3.1.3 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์	42
	3.1.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง	43
	3.2 การนำมาประยุกต์ใช้งาน Signal Interpolation	47
	3.2.1 ขั้นตอนการทำ Double Interpolation	48
	3.2.2 ขั้นตอนการทำ Triple Interpolation	49
	3.2.3 ขั้นตอนการทำ Four time Interpolation	51
	3.3 การทำ Oversampling D/A	52
	3.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ดัดแปลงบน Hardware	54
	3.4.1 ส่วนประกอบของวงจรย่อย	56
บทที่ 4	**ผลการทดลอง**	60
	4.1 คุณสมบัติของตัววงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์	60
	4.1.1 Software simulation (MATLAB) ของตัวกรองอันดับที่ 2	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	4.1.2 Software simulation (MATLAB) ของตัวกรองอันดับที่ 5	61
	4.1.3 Hardware simulation (ภาษา VHDL) ของตัวกรองอันดับ ที่ 2	62
	4.1.4 Hardware simulation (ภาษา VHDL) ของตัวกรองอันดับ ที่ 5	63
	4.2 การทำ Oversampling D/A	64
	4.3 ผลการจำลองการทำงานทางซอฟแวร์ในการทำ Oversampling D/A	66
	4.3.1 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz	67
	4.3.2 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz	68
	4.3.3 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz	70
	4.4 ผลการจำลองการทำงานทางฮาร์ดแวร์ในการทำ Oversampling D/A	71
	4.4.1 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 12 kHz	72
	4.4.2 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz	73
	4.4.3 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz	74
	4.5 ผลการทำงานจริงทางฮาร์ดแวร์ในการทำ Oversampling D/A	75
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	78
	5.1 สรุปผล	78
	5.2 ข้อเสนอแนะ	79
บรรณานุกรม		80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

1.1	การหน่วงสัญญาณ 1 หน่วย	1
1.2	การหน่วงสัญญาณ 0.5 หน่วย	1
1.3	ภาพรวมของระบบของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์	2
1.4	วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 ในโครงสร้างแบบแฟร์โรว์	2
2.1	ความแตกต่างระหว่างค่าความหน่วงเฟส และค่าความหน่วงกลุ่ม	7
2.2	โครงสร้างแบบ Direct Form ของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2	9
2.3	แสดงโครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ [7]	10
2.4	โครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณในอันดับที่ 2 (แสดง-สัมประสิทธิ์วงจรกรองย่อย)	11
2.5	แสดงโครงสร้างแบบแฟร์โรว์ดัดแปลงที่ได้จากเมตริกซ์การแปลง [7]	15
2.6	โครงสร้างแบบแฟร์โรว์ดัดแปลงของวงจรกรองสัญญาณที่ได้จากการแปลงที่อันดับตัวกรองที่ 2 (แสดงสัมประสิทธิ์วงจรกรองย่อย)	16
2.7	การออกแบบจากบนลงล่าง	21
2.8	โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้	23
2.9	รูปแบบของ RS – flip flop	23
2.10	โครงสร้างโดยทั่วไปของหน่วการออกแบบสถาปัตยกรรม	24
2.11	โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต	26
2.12	โครงสร้างโดยทั่วไปของส่วนของบอดี้แพ็กเก็ต	26
2.13	โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	27
2.14	ผังแสดงการแบ่งกลุ่มของวงจรรวม ASIC	31
2.15	ลักษณะของตัว FPGA และการนำไปใช้งาน	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

2.16 แสดงการควบคุมเวลาของสัญญาณการชักตัวอย่างของส่วนประมวลผลสัญญาณอนาลอก 35

2.17 แสดงการควบคุมเวลาของสัญญาณการชักตัวอย่างของส่วนประมวลผลสัญญาณเชิงเลข 36

2.18 แสดงสัญญาณอินพุตที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s1} แปลงไปสู่สัญญาณเอาต์พุตที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s2} 36

2.19 แสดงแผนผังของระบบวงจรแปลงอัตราการชักตัวอย่าง 37

2.20 ผลการเปรียบเทียบของสัญญาณคลื่นไซน์กับสัญญาณคลื่นไซน์ที่ผ่านการทำ Oversampling D/A 2 เท่า 38

3.1 Block diagram การทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ 39

3.2 โครงสร้างการทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิด ลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์-โรว์ดัดแปลง 45

3.3 แสดงความแตกต่างของการหน่วงสัญญาณ 0.5 หน่วย กับสัญญาณที่ถูกหน่วงไป 1 หน่วย 47

3.4 ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.5 หน่วยพร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 2 เท่า 48

3.5 ผลของการทำ Double Interpolation เทียบสัญญาณ Input เดิม 49

3.6 ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.67 หน่วย และ 0.33 หน่วย พร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 3 เท่า 49

3.7 ผลของการทำ Triple Interpolation เทียบสัญญาณ Input เดิม 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.8	ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.75 หน่วย 0.5 หน่วย และ 0.25 หน่วย พร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 4 เท่า	51
3.9	ผลของการทำ Four time Interpolation เทียบสัญญาณ Input เดิม	52
3.10	Block diagram การทำงานของการทำ Oversampling D/A 2 เท่า	53
3.11	Block diagram การทำงานของการทำ Oversampling D/A 3 เท่า	53
3.12	Block diagram การทำงานของการทำ Oversampling D/A 4 เท่า	54
3.13	การออกแบบของ VFD filter (order5) ในโครงสร้างแบบแฟร์โรว์ดัดแปลง	55
3.14	ผลของการสังเคราะห์วงจรบวก	56
3.15	ผลของการสังเคราะห์วงจรลบ	56
3.16	ผลของการสังเคราะห์วงจรคูณ	57
3.17	ผลของการสังเคราะห์วงจรหน่วงเวลา	57
3.18	ผลของการสังเคราะห์วงจรตัดบิต	57
3.19	ผลของการสังเคราะห์วงจร PIPO	58
3.20	ผลของการสังเคราะห์วงจร look up table	58
3.21	ผลของการสังเคราะห์วงจร Counter	58
3.22	ผลของการสังเคราะห์วงจร Multiplexer	59
4.1	Magnitude response ของตัวกรองอันดับที่ 2 ที่ได้จาก MATLAB	60
4.2	Phase delay response ของตัวกรองอันดับที่ 2 ที่ได้จาก MATLAB	60
4.3	Magnitude response ของตัวกรองอันดับที่ 5 ที่ได้จาก MATLAB	61
4.4	Phase delay response ของตัวกรองอันดับที่ 5 ที่ได้จาก MATLAB	61
4.5	Magnitude response ของตัวกรองอันดับที่ 2 ที่ได้จาก ภาษา VHDL	62
4.6	Phase delay response ของตัวกรองอันดับที่ 2 ที่ได้จาก ภาษา VHDL	62
4.7	Magnitude response ของตัวกรองอันดับที่ 5 ที่ได้จาก ภาษา VHDL	63
4.8	Phase delay response ของตัวกรองอันดับที่ 5 ที่ได้จาก ภาษา VHDL	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.9	Block diagram แสดงปัญหาของการทดลองที่เกิดขึ้น	64
4.10	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz ตามรูป Block diagram ที่ 4.9	64
4.11	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz ตามรูป Block diagram ที่ 4.9	65
4.12	ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz ตามรูป Block diagram ที่ 4.9	65
4.13	Block diagram แสดงการทำงานของ Over sampling D/A	66
4.14	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB	67
4.15	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB	67
4.16	ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB	68
4.17	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 3 kHz ด้วยโปรแกรม MATLAB	68
4.18	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 3 kHz ด้วยโปรแกรม MATLAB	69
4.19	ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 3 kHz ด้วยโปรแกรม MATLAB	69
4.20	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 4.8 kHz ด้วยโปรแกรม MATLAB	70
4.21	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 4.8 kHz ด้วยโปรแกรม MATLAB	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.22	ผลของการทำ Over Sampling D/A 4 เท่า ที่ความถี่ 4.8 kHz ด้วยโปรแกรม MATLAB	71
4.23	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL	72
4.24	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL	72
4.25	ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL	72
4.26	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL	73
4.27	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL	73
4.28	ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL	73
4.29	ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL	74
4.30	ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL	74
4.31	ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL	74
4.32	แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 1.2 kHz	75
4.33	แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 3 kHz	75
4.34	แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 4.8 kHz	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 จำนวนตัวคูณและตัวบวกที่ใช้ในวงจรที่อันดับตัวกรองต่างๆ 17

3.1 ค่าสัมประสิทธิ์ sub filter ในรูปที่ 3.15 46

3.2 ช่วงการหน่วงสัญญาณในโครงสร้างแบบแฟร์โรว์และแฟร์โรว์ดัดแปลง 46

3.3 ความสัมพันธ์ระหว่างการทำ Interpolation และ จำนวนค่าการหน่วงที่ใช้ 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

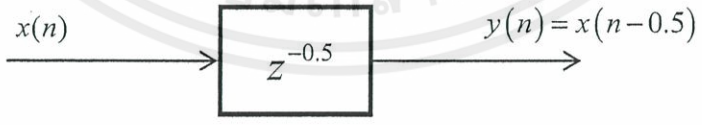
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

วงจรกรองสัญญาณเชิงเลข (Digital Filter) เป็นวงจรกรองที่สามารถสร้างขึ้นให้อยู่ในรูปแบบของ hardware หรือ software ได้ โดยวงจรกรองสัญญาณเชิงเลขที่ออกแบบและสร้างขึ้นมาจะเป็นวงจรกรองที่สังเคราะห์ขึ้นให้สามารถทำงานได้เช่นเดียวกับวงจรกรองสัญญาณเชิงอุปมาน ซึ่งในปัจจุบันวงจรกรองสัญญาณเชิงเลขเป็นส่วนประกอบที่สำคัญมากโดยเฉพาะงานที่เกี่ยวข้องกับการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing : DSP) เนื่องจากวงจรกรองสัญญาณเชิงเลขไม่ได้มีคุณสมบัติเพื่อกรองสัญญาณเชิงเลขเพียงอย่างเดียว แต่ยังสามารถประยุกต์ใช้ในงานการประมวลสัญญาณเชิงเลขอื่นๆ ได้อีกมากมาย

และเนื่องจากในระบบประมวลผลสัญญาณดิจิตอลสามารถทำการหน่วงสัญญาณได้อย่างน้อยที่สุดคือการหน่วงหนึ่งหน่วย และสามารถหน่วงได้เฉพาะเลขจำนวนเต็มเท่านั้น ซึ่งในลักษณะงานบางประเภทต้องการการหน่วงสัญญาณที่น้อยกว่าหนึ่งหน่วย หรือการหน่วงสัญญาณแบบเป็นเศษส่วน ซึ่งในระบบประมวลผลสัญญาณดิจิตอลลักษณะทั่วไปไม่สามารถทำได้

รูปที่ 1.1 การหน่วงสัญญาณ 1 หน่วย



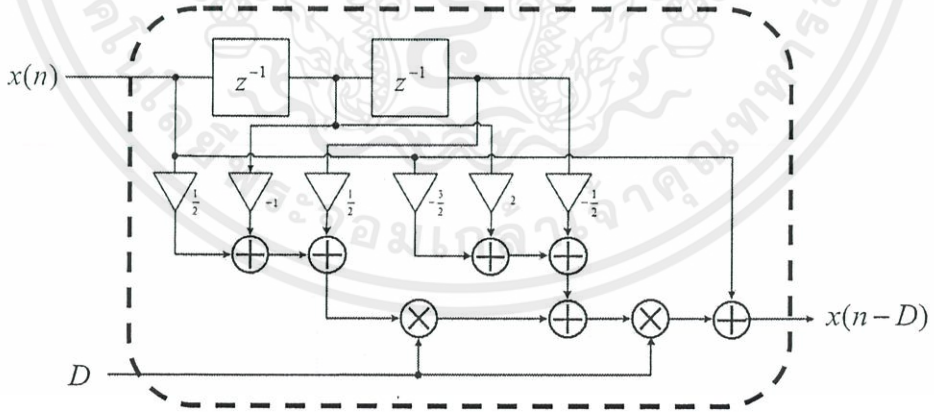
รูปที่ 1.2 การหน่วงสัญญาณ 0.5 หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราจึงนำวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์มาแก้ไขปัญหาการหน่วงสัญญาณในระบบประมวลผลสัญญาณดิจิตอลให้สามารถทำการหน่วงสัญญาณแบบเป็นเศษส่วนได้ โดยนำเสนอวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ด้วยโครงสร้างแบบแฟร์โรว์ และโครงสร้างแบบแฟร์โรว์ดัดแปลง ซึ่งจะสามารถปรับค่าความหน่วงได้อย่างต่อเนื่อง และตลอดช่วงเวลาตลอดเวลาโดยไม่ต้องคำนวณค่าสัมประสิทธิ์ภายในวงจรกรองใหม่ทุกครั้ง ดังรูปที่ 1.3

รูปที่ 1.3 ภาพรวมของระบบของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์

จากรูปที่ 1.3 การออกแบบภายในวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์นั้นจะนำเสนอในโครงสร้างแบบแฟร์โรว์ ซึ่งสามารถแสดงตัวอย่างการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 ในโครงสร้างแบบแฟร์โรว์ได้ดังรูปที่ 1.4 โดยที่ค่า D คือค่าความหน่วงแบบเป็นเศษส่วน



รูปที่ 1.4 วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 ในโครงสร้างแบบแฟร์โรว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ยังสามารถนำมาประยุกต์ใช้งานได้อย่างมากมาย ซึ่งในปริญญานิพนธ์ฉบับนี้จะแสดงการประยุกต์ใช้งานทางด้านการทำ Oversampling D/A โดยจะทำการแก้ไขปัญหาสัญญาณที่มีความละเอียดต่ำให้มีความละเอียดมากยิ่งขึ้น โดยจะใช้ภาษา VHDL ในการออกแบบและสังเคราะห์วงจรขึ้นมา และทำการ Implement ลงบนอุปกรณ์ FPGA เพื่อแสดงผลบนเครื่อง Oscilloscope จากนั้นจะทำการเปรียบเทียบผลที่ได้กับผลการจำลองการทำงานทางซอฟแวร์ในโปรแกรม MATLAB และผลการจำลองการทำงานทางฮาร์ดแวร์ใน MODEL SIM ด้วยภาษา VHDL

1.2 วัตถุประสงค์

1) เพื่อศึกษาหลักการออกแบบและการทำงานของ Lagrange - Type VFD (Variable Fractional Delay) Filter
2) เพื่อศึกษาโปรแกรม MATLAB และภาษา VHDL ในการออกแบบและจำลองการทำงานของ Lagrange - Type VFD Filter ที่ใช้โครงสร้างแบบแฟร์โรว์ (Farrow Structure) และ แฟร์โรว์แบบดัดแปลง (Modified Farrow Structure)
3) เพื่อศึกษาภาษา VHDL ในการออกแบบและสังเคราะห์วงจร Lagrange Type VFD Filter ในโครงสร้างแบบแฟร์โรว์ และแฟร์โรว์ดัดแปลง ให้เป็นฮาร์ดแวร์และทำการจัดโครงสร้าง (Configuration) ลงบนอุปกรณ์ FPGA เพื่อทดสอบการทำงาน

1.3 ขอบเขตของปริญญานิพนธ์

1) สามารถเข้าใจหลักการออกแบบและการทำงานของ Lagrange - Type VFD Filter
2) สามารถออกแบบและจำลองการทำงานของ Lagrange - Type VFD Filter ในโครงสร้างแบบแฟร์โรว์ดัดแปลง โดยใช้อันดับตัวกรองที่ 5 และสามารถนำมาประยุกต์ใช้ในการทำ Over sampling D/A ที่ 2, 3 และ 4 เท่าที่ความถี่ 1.2 kHz, 3 kHz และ 4.8 kHz ได้
3) สามารถใช้ภาษา VHDL ในออกแบบ สังเคราะห์วงจร และจำลองการทำงานของ Lagrange Type VFD Filter ที่ใช้โครงสร้างแบบแฟร์โรว์ดัดแปลงให้เป็นฮาร์ดแวร์ โดยใช้อันดับตัวกรองที่ 5 และสามารถนำมาประยุกต์ใช้ในการทำ Over sampling D/A ที่ 2, 3 และ 4 เท่าที่ความถี่ 1.2 kHz, 3 kHz และ 4.8 kHz ได้ โดยทำการจัดโครงสร้าง (Configuration) ลงบนอุปกรณ์ FPGA เพื่อทดสอบการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 วงจรกรองสัญญาณเชิงเลข (Digital Filter)

ในอดีตจนถึงปัจจุบันได้มีการออกแบบและพัฒนาวงจรกรองสัญญาณ (filter) ซึ่งวงจรกรองสัญญาณโดยทั่วไปแล้วจะนำมาใช้เพื่อให้กรองสัญญาณที่ไม่ต้องการใช้งานออกไปและคงเหลือสัญญาณที่ต้องการไว้ ถ้าจะแบ่งวงจรกรองสัญญาณตามผลตอบสนองทางความถี่ (frequency response) จะสามารถแบ่งวงจรกรองสัญญาณได้ 4 ประเภทได้แก่ วงจรกรองความถี่ต่ำผ่าน (low-pass filter), วงจรกรองความถี่สูงผ่าน (high-pass filter), วงจรกรองแถบความถี่ผ่าน (band-pass filter) และวงจรกรองแถบความถี่หยุด (band-stop filter) โดยทั่วไปวงจรกรองสัญญาณสามารถแบ่งได้ตามคุณลักษณะของสัญญาณที่นำมากรองคือสัญญาณเชิงอุปมาน (analog signal) และสัญญาณเชิงเลข (digital signal) ดังนั้นเราสามารถจำแนกวงจรกรองได้ 2 ชนิด นั่นคือวงจรกรองสัญญาณเชิงอุปมาน (analog signal) และวงจรกรองสัญญาณเชิงเลข (digital signal)

วงจรกรองสัญญาณเชิงอุปมาน (analog signal) อาจเรียกอีกชื่อหนึ่งว่าวงจรกรองอนาล็อก โดยวงจรกรองเชิงอุปมานสามารถที่จะจำแนกได้ตามอุปกรณ์ที่นำมาใช้คือ วงจรกรองเชิงอุปมานแบบแพสซิฟ (passive analog filter) และวงจรกรองเชิงอุปมานแบบแอ็กทิฟ (active analog filter) โดยวงจรกรองเชิงอุปมานแบบแพสซิฟเป็นวงจรกรองที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์และสามารถทำงานได้ได้โดยไม่ต้องอาศัยแหล่งจ่าย จะประกอบด้วยตัวต้านทาน (resister, R) ตัวเก็บประจุ (capacitor, C) ตัวเหนี่ยวนำ (inductor, L) เป็นส่วนประกอบหลักของวงจรกรอง ส่วนวงจรกรองเชิงอุปมานแบบแอ็กทิฟเป็นวงจรกรองที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ที่ต้องพึ่งพิงแหล่งจ่ายในการทำงานได้แก่ ทรานซิสเตอร์และออปแอมป์และอุปกรณ์แบบแอ็กทิฟจะทำงานโดยต่อร่วมกับอุปกรณ์อิเล็กทรอนิกส์แบบแพสซิฟ

วงจรกรองสัญญาณเชิงเลข (digital filter) เป็นวงจรกรองที่สามารถสร้างขึ้นให้อยู่ในรูปแบบของ hardware หรือ software ได้ โดยวงจรกรองสัญญาณเชิงเลขที่ออกแบบและสร้างขึ้นมาจะเป็นวงจรกรองที่สังเคราะห์ให้สามารถทำงานได้เช่นเดียวกับวงจรกรองสัญญาณเชิงอุปมาน แต่จะสามารถประมวลผลกับสัญญาณเชิงเลขเท่านั้น ในปัจจุบันวงจรกรองเชิงเลขเป็นส่วนประกอบที่สำคัญมากโดยเฉพาะงานที่เกี่ยวข้องกับการประมวลผลสัญญาณเชิงเลข (Digital signal processing: DSP) เนื่องจากวงจรกรองสัญญาณเชิงเลขไม่ได้มีคุณสมบัติเพื่อกรองสัญญาณเชิงเลขเพียงอย่างเดียว แต่ยังสามารถประยุกต์ใช้ในงานการประมวลสัญญาณเชิงเลขอื่นๆ ได้เช่น data compression, biomedical signal processing, speech processing, image processing, data transmission, digital audio, phone echo cancellation เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสามารถสรุปข้อเปรียบเทียบวงจรกรองสัญญาณเชิงเลข (Digital filter) กับวงจรกรองสัญญาณเชิงอุปมาน (Analog filter) ได้ดังนี้

1. การสร้างวงจรกรองสัญญาณให้มีคุณลักษณะ (characteristics) ตามที่ออกแบบไว้วงจรกรองสัญญาณเชิงเส้นสามารถสร้างให้วงจรกรองมีคุณลักษณะตรงตามที่ออกแบบได้แต่วงจรกรองเชิงอุปมานไม่สามารถสร้างได้

2. การเปลี่ยนแปลงของสภาพแวดล้อมเช่น อุณหภูมิ, ความชื้น, สัญญาณรบกวน วงจรกรองสัญญาณเชิงอุปมานจะมีการแปรเปลี่ยนตามสภาพแวดล้อมได้มากกว่าวงจรกรองสัญญาณเชิงเลข และผลของการแปรเปลี่ยนตามสภาพแวดล้อมของวงจรกรองสัญญาณเชิงอุปมานจึงต้องมีการแก้ไขวงจรกรองสัญญาณเชิงอุปมานตามสภาพแวดล้อมนั้นๆ เพื่อให้วงจรกรองมีคุณลักษณะตามที่ต้องการ

3. ในทางปฏิบัติค่าความละเอียดในการกรองสัญญาณที่ได้จากวงจรกรองสัญญาณเชิงอุปมานมีขีดจำกัด ยกตัวอย่างเช่นค่า maximum stopband attenuation ของวงจรกรองสัญญาณเชิงอุปมานแบบแอ็กทิฟ (active filters) จะมีค่าอยู่ในช่วง 60-70 dB ซึ่งการหาอุปกรณ์ให้ตรงตามต้องการนี้ไม่สามารถหาได้ทั่วไป ในขณะที่ขีดจำกัดค่าความละเอียดในการกรองสัญญาณของวงจรกรองสัญญาณเชิงเลขขึ้นอยู่กับความยาวของคำ (wordlength) ที่เลือกใช้

4. การออกแบบและสร้างวงจรกรองสัญญาณเชิงเลขที่สร้างโดยอาศัยเทคโนโลยีทางด้าน VLSI (Very Large Scale Integrate circuits) สามารถทำให้วงจรกรองสัญญาณเชิงเลขมีขนาดเล็ก กินกำลังงานไฟฟ้าต่ำ และมีคาคาถูกลง

5. ในทางปฏิบัติการนำวงจรกรองสัญญาณไปใช้งานในย่านความถี่ต่ำมากๆ วงจรกรองสัญญาณเชิงอุปมานไม่สามารถทำงานได้ดีเมื่อเทียบวงจรกรองสัญญาณเชิงเลข นอกจากนั้นวงจรกรองสัญญาณเชิงเลขยังสามารถทำงานได้ในย่านความถี่กว้างได้ โดยขึ้นอยู่กับการเปลี่ยนแปลงค่าของอัตราการชักความถี่ (sampling frequency)

วงจรกรองสัญญาณเชิงเลขสามารถแบ่งได้เป็น 2 ชนิดคือวงจรกรองสัญญาณผลตอบสนองอิมพัลส์จำกัด (finite impulse response filter: FIR filter) และวงจรกรองสัญญาณผลตอบสนองอิมพัลส์ไม่จำกัด (infinite impulse response filter: IIR filter) ซึ่งสามารถแสดงสมการฟังก์ชันถ่ายโอน (transfer function) ของวงจรกรองผลตอบสนองอิมพัลส์จำกัดได้ดังสมการที่ (2.1)

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

และสมการฟังก์ชันถ่ายโอนของวงจรกรองผลตอบสนองอิมพัลส์ไม่จำกัดได้ดังสมการที่ (2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

วงจรกรองสัญญาณเชิงเลขที่ปรับค่าได้ (Variable digital filter) คือวงจรกรองสัญญาณเชิงเลขที่สามารถเปลี่ยนแปลงคุณลักษณะของผลตอบสนองทางความถี่ได้ โดยในวงจรกรองสัญญาณเชิงเลขเรามักจะให้ความสนใจกับผลตอบสนองทางความถี่เนื่องจากผลตอบสนองทางความถี่จะเป็นสิ่งที่บอกคุณลักษณะของวงจรกรองสัญญาณเชิงเลขได้ว่าให้ผลตอบสนองกับความถี่เป็นอย่างไร โดยผลตอบสนองทางความถี่สามารถจำแนกได้ 2 ส่วนคือผลตอบสนองทางขนาด (magnitude response) และผลตอบสนองทางเฟส (phase response) ซึ่งสามารถแสดงได้จากสมการฟังก์ชันถ่ายโอนของวงจรกรองสัญญาณเชิงเลขดังนี้

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

สามารถหาผลตอบสนองทางขนาด (magnitude response) ได้ดังนี้

$$A(\omega) = \left|H(e^{i\omega})\right| \tag{2.4}$$

และผลตอบสนองทางเฟส (phase response)

$$\theta(\omega) = \angle H(e^{i\omega}) \tag{2.5}$$

ซึ่งการเปลี่ยนแปลงผลตอบสนองทางขนาดและผลตอบสนองทางเฟสของวงจรกรองสัญญาณเชิงเลข อาจทำได้ด้วยการคำนวณหาค่าสัมประสิทธิ์ที่สัมพันธ์กับพารามิเตอร์ใดๆ หรืออาจทำได้ด้วยการปรับค่าพารามิเตอร์ใดๆ เพื่อให้ผลตอบสนองของวงจรกรองสัญญาณเชิงเลขเปลี่ยนแปลงไปตามต้องการ

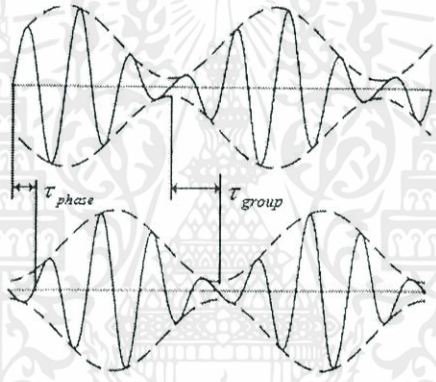
นอกจากการแสดงให้เห็นถึงผลตอบสนองทางขนาดที่เปลี่ยนแปลงไปตามค่าพารามิเตอร์ที่ปรับเปลี่ยน วงจรกรองสัญญาณเชิงเลขที่ปรับค่าได้ยังสามารถแสดงให้เห็นถึงผลตอบสนองทางเฟสที่เปลี่ยนแปลงไปด้วย จากผลตอบสนองทางเฟสสามารถแสดงได้ได้เป็น 2 ลักษณะคือ ผลตอบสนองความหน่วงเฟส (phase delay response) และผลตอบสนองความหน่วงกลุ่ม (group delay response) ดังสมการที่ (2.6) และสมการที่ (2.7) ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

Phase delay response ของวงจรกรองสัญญาณคือปริมาณของ time delay ของแต่ละองค์ประกอบทางความถี่ของสัญญาณที่เกิดขึ้นจากการผ่านวงจรกรองสัญญาณ Group Delay Response คือค่า Average Time Delay แต่ละองค์ประกอบทางความถี่ของสัญญาณที่ประกอบไปด้วยหลายองค์ประกอบทางความถี่



รูปที่ 2.1 ความแตกต่างระหว่างค่าความหน่วงเฟส และค่าความหน่วงกลุ่ม

วงจรกรองสัญญาณที่มีผลตอบสนองเฟสแบบไม่เป็นเชิงเส้นจะเป็นสาเหตุให้เกิดการผิดเพี้ยนทางเฟส (phase distortion) ของสัญญาณซึ่งอาจส่งผลอย่างรุนแรงถึงขนาดทำให้ waveform ของสัญญาณที่ผ่านวงจรกรองสัญญาณออกมาผิดเพี้ยนไปเลย ซึ่งการผิดเพี้ยนดังกล่าวเป็นสิ่งที่ไม่พึงปรารถนาให้เกิดขึ้นสำหรับหลายๆ applications โดยเฉพาะอย่างยิ่ง applications ที่ต้องใช้สายตาในการมองรูปคลื่นหรือรูปของสัญญาณ เช่นงานทางด้าน video และ Image processing งานทางด้าน biomedical signal processing ด้าน data transmission การผิดเพี้ยนทางเฟสดังกล่าวสามารถหลีกเลี่ยงได้โดยการใช้วงจรกรองสัญญาณที่มีผลตอบสนองเฟสแบบเป็นเชิงเส้นในช่วงแถบความถี่ที่เราสนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์

วิธีการที่เป็นต้นแบบได้ดี (classical) ของวงจรกรองสัญญาณผลตอบสนองอิมพัลส์จำกัด (FIR filter) ที่ปรับค่าความหน่วงแบบเป็นเศษส่วนได้คือวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ (Lagrange-type VFD filter) โดยหัวข้อย่อยนี้จะแสดงให้เห็นถึงวงจรกรองสัญญาณเชิงเลขที่สามารถทำให้เป็นวงจรกรองสัญญาณเชิงเลขที่ปรับค่าได้ (variable digital filter) ซึ่งวงจรกรองสัญญาณดังกล่าวได้มาจากการประมาณค่าในช่วงลากรานจ์ (Lagrange interpolation) โดยวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์สามารถหาเอาต์พุตได้ดังสมการที่ (2.8) [8]

$$y(n) = \sum_{i=0}^{k} h_i(D)\, x(n-i) \tag{2.8}$$

และสามารถหาค่าสัมประสิทธิ์วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ได้ดังสมการที่ (2.9) [8]

$$h_i(D) = \prod_{\substack{l=0 \\ l \neq i}}^{k} \frac{D-l}{i-l} \quad ; i = 0, 1, 2, \ldots, k \tag{2.9}$$

สำหรับ $l = 0, 1, 2, \ldots, k$ เมื่อ

$h_i(D)$ คือ ค่าสัมประสิทธิ์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์

D คือ พารามิเตอร์ความหน่วง (delay parameter)

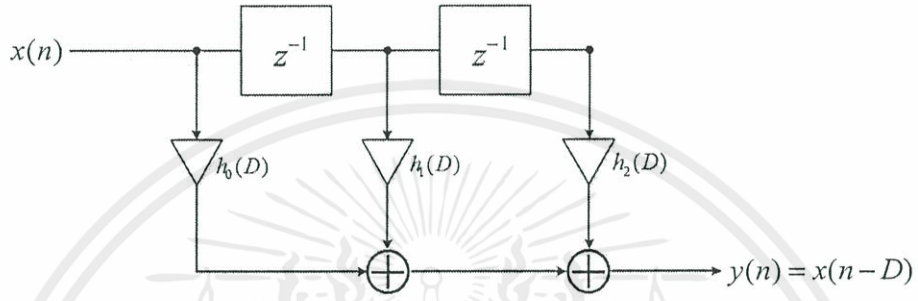
k คือ อันดับของวงจรกรองสัญญาณเชิงเลขชนิดลากรานจ์

จากสมการที่ (2.9) จะแสดงตัวอย่างวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่สอง ($k = 2$) ซึ่งสามารถแสดงฟังก์ชันถ่ายโอนของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่สองได้ดังนี้

$$H(z) = \frac{1}{2}(D^2 - 3D + 2) + (-D^2 - 2D)z^{-1} + \frac{1}{2}(D^2 - D)z^{-2} \tag{2.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $h_0(D)=\frac{1}{2}(D^2-3D+2)$, $h_1(D)=(-D^2-2D)$ และ $h_2(D)=\frac{1}{2}(D^2-D)$ โดยเราสามารถแสดงตัวอย่างโครงสร้างแบบ direct form ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ได้ดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างแบบ Direct Form ของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2

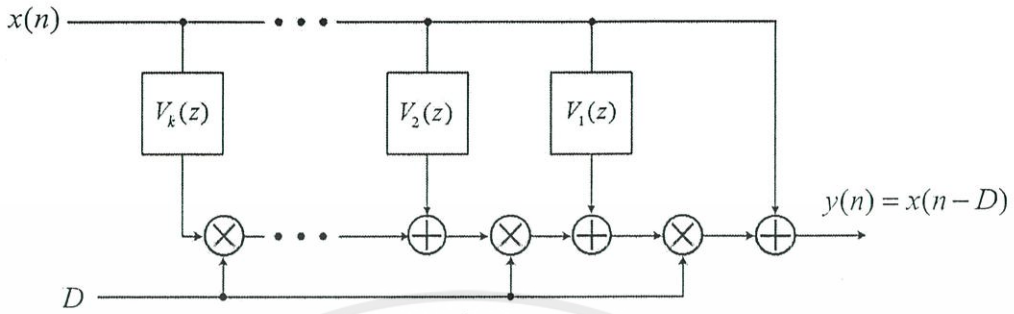
จากฟังก์ชันถ่ายโอนในสมการที่ (2.10) จะเห็นได้ว่าตัวอย่างของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่สองเป็นวงจรกรองที่สามารถเปลี่ยนแปลงคุณลักษณะของวงจรกรองได้ด้วยการเปลี่ยนแปลงพารามิเตอร์ความหน่วง แต่จะต้องมีการคำนวณค่าสัมประสิทธิ์ของตัวกรองใหม่ทุกครั้ง จึงได้มีการนำวงจรรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์เสนอในโครงสร้างแบบแฟร์โรว์และโครงสร้างแบบแฟร์โรว์ดัดแปลง

2.2.1 โครงสร้างแบบแฟร์โรว์สำหรับวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนลากรานจ์ (Farrow Structure For Lagrange - Type VFD Filter)

วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ได้มาจากการประมาณค่าในช่วงลากรานจ์ ซึ่งการประมาณค่าในช่วงลากรานจ์นั้นจะได้มาจากหาพหุนามลากรานจ์ที่อันดับ k^{th} โดยจะมีขนาดของข้อมูลเท่ากับ $k+1$ วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์สามารถเรียกได้ว่าเป็นวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนที่มีความราบเรียบที่สุด

ปกติวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์มักจะนิยมที่จะนำเสนอในโครงสร้างแบบแฟร์โรว์ ซึ่งสามารถแสดงได้ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงโครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ [7]

โดยปกติฟังก์ชันถ่าโอนของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์จะมีค่าสัมประสิทธิ์ของวงจรกรองย่อย $V_0(z)=1$ และค่าสัมประสิทธิ์ของวงจรกรองย่อยอื่น ๆ $V_i(z)$ สามารถคำนวณหาได้จากสมการที่ (2.9) ตัวอย่างเช่น กำหนดให้วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 $(k=2)$ จากสมการที่ (2.11) และ สมการที่ (2.12) สามารถหาค่าเอาต์พุต $y(n)$ และค่าสัมประสิทธิ์ของวงจรกรองได้ดังนี้

$$y(n)=h_0(D)\,x(n)+h_1(D)\,x(n-1)+h_2(D)\,x(n-2) \tag{2.11}$$

เมื่อ

$$\begin{aligned} h_0(D)&=\frac{1}{2}\left[D^2-3D+2\right]\\ h_1(D)&=-\left[D^2-2D\right]\\ h_2(D)&=\frac{1}{2}\left[D^2-D\right] \end{aligned} \tag{2.12}$$

ดังนั้นสมการเอาต์พุตได้ดังนี้

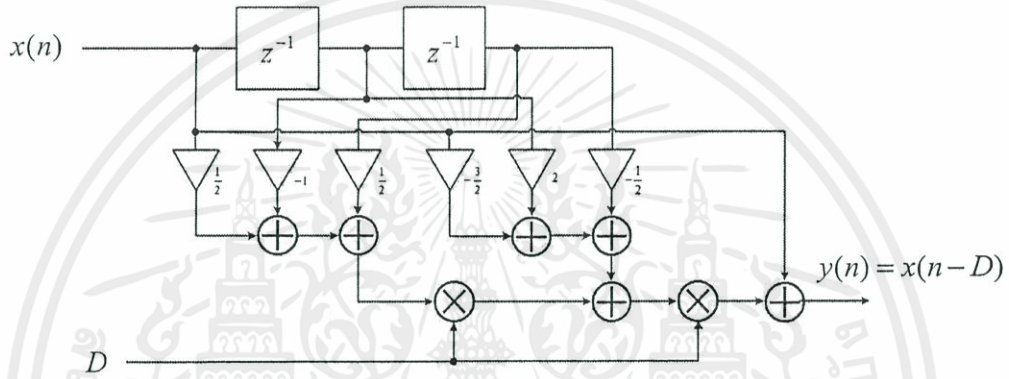
$$y(n)=\frac{1}{2}\left[D^2-3D+2\right]x(n)+\left[2D-D^2\right]x(n-1)+\frac{1}{2}\left[D^2-D\right]x(n-2) \tag{2.13}$$

และหาฟังก์ชันถ่ายโอนของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(z,D)=1+D\left[-\frac{3}{2}+2z^{-1}-\frac{1}{2}z^{-2}\right]+D^2\left[\frac{1}{2}-z^{-1}+\frac{1}{2}z^{-2}\right] \qquad (2.14)$$

เมื่อ $V_0(z)=1$, $V_1(z)=-\frac{3}{2}+2z^{-1}-\frac{1}{2}z^{-2}$ และ $V_2(z)=\frac{1}{2}-z^{-1}+\frac{1}{2}z^{-2}$ ดังนั้นเราสามารถเขียนโครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ได้ดังรูปที่ 2.4, 2.5



รูปที่ 2.4 โครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณในอันดับที่ 2
(แสดงสัมประสิทธิ์วงจรกรองย่อย)

จากรูปที่ (2.4) จะเห็นได้ว่าโครงสร้างที่แสดงในกรณีทั่วไปสามารถจะประกอบด้วยจำนวนของการคูณ 8 ตัวและจำนวนของการบวก 6 ตัว

อีกวิธีหนึ่งในการหาค่าสัมประสิทธิ์ย่อยของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์คือการหาค่าเมตริกซ์ $\mathbf{V}$ โดยที่เมตริกซ์ $\mathbf{V}$ คือ Vandermonde matrix ถูกกำหนดโดย [7]

$$\mathbf{V}=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 1 & 1 & 1 & \cdots & 1\\ 1 & 2 & 2^2 & \cdots & 2^k\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & k & k^2 & \cdots & k^k\end{bmatrix},\quad z=\begin{bmatrix}1\\ z^{-1}\\ z^{-2}\\ \vdots\\ z^{-k}\end{bmatrix}$$

โดยที่สมการ Transfer function ของ Lagrange – Type VFD Filter คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(z,D)=\sum_{i=0}^{k}V_i(z)D^i=\mathbf{D}^T\mathbf{V}(z) \tag{2.15}$$

เมื่อ

$$\mathbf{D}^T=\begin{bmatrix}1 & D & D^2 & \dots & D^k\end{bmatrix}$$

และ

$$\mathbf{V}(z)=\begin{bmatrix}V_0(z)\\V_1(z)\\V_2(z)\\\vdots\\V_k(z)\end{bmatrix}$$

ตามนิยามของตัวกรองหน่วงสัญญาณแบบจำนวนเต็มสามารถแสดงความสัมพันธ์ระหว่างเอาท์พุตกับอินพุตได้ดังนี้

$$Y(z)=z^{-D}X(z)\,,\ D=0,1,2,\dots,k \tag{2.16}$$

เมื่อพิจารณาสมการที่ (3.1) และ (3.2) จะได้

$$\sum_{i=0}^{k}V_i(z)D^i=z^{-D}\,,\ D=1,2,\dots,k \tag{2.17}$$

จากสมการที่ (2.17) สามารถแสดงให้อยู่ในรูปแบบ Matrix ได้ดังนี้

$$\mathbf{V}\mathbf{V}(z)=z \tag{2.18}$$

สัมประสิทธิ์ย่อยของ $\mathbf{V}(z)$ สามารถหาค่าได้จาก

$$\mathbf{V}(z)=\mathbf{V}^{-1}z \tag{2.19}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแถวของ Vandermonde Matrix ผกผัน $\mathbf{V}^{-1}$ จะแสดงค่าสัมประสิทธิ์ย่อยของตัวกรองที่ออกแบบไว้ขึ้นอยู่กับอันดับของตัวกรองที่ใช้ และสามารถแสดง Transfer function ได้ดังนี้

$$H(z,D) - \sum_{i=0}^{k} V_i(z) D^i - \mathbf{D}^{\mathrm{T}} \mathbf{V}^{-1} z \qquad (2.20)$$

ซึ่งในวิธีนี้จะสังเกตได้ว่าค่าสัมประสิทธิ์ย่อยของตัวกรองที่ได้มีลักษณะเหมือนกันกับวิธีแรกที่ทำการจัดรูปสัมการให้อยู่ในรูปของ D ยกกำลังในสมการที่ (2.6) แต่จะมีลักษณะการหาค่าสัมประสิทธิ์ย่อยของตัวกรองที่สะดวกกว่าแบบเดิม

2.2.2 โครงสร้างแบบแฟร์โรว์ดัดแปลงสำหรับวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนลากรานจ์ (Modified Farrow Structure For Lagrange - Type VFD Filter)

การลดความซับซ้อนของการคำนวณของโครงสร้างแบบแฟร์โรว์ต้นแบบสามารถทำได้ด้วยการใช้เมตริกซ์การแปลง (Transformation Matrix) โดยค่าสัมประสิทธิ์ตัวกรองจะมีคุณสมบัติสมมาตรและปฏิสมมาตรในวงจรกรองย่อยทั้งหมด และสามารถหาค่าสัมประสิทธิ์วงจรกรองย่อยใหม่ได้ด้วยการใช้เมตริกซ์แวนเดอร์มอนด์ผกผัน $\mathbf{V}^{-1}$ กับเมตริกซ์การแปลง $\mathbf{T}$ โดยฟังก์ชันถ่ายโอนใหม่ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ที่ได้จากการแปลงเมื่อเปรียบเทียบกับโครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ ส่งผลทำให้จำนวนของการคูณในโครงสร้างใหม่สามารถลดลงได้เกือบ 50 เปอร์เซ็นต์

โครงสร้างใหม่ซึ่งตัวกรองย่อยเป็นเฟสเชิงเส้น สามารถหาได้โดยใช้เมทริกซ์การแปลง $\mathbf{T}$ [7] ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mathbf{T}=\begin{bmatrix} 1 & \left(\frac{k}{2}\right) & \left(\frac{k}{2}\right)^2 & \left(\frac{k}{2}\right)^3 & \cdots & \left(\frac{k}{2}\right)^k \\ 0 & 1 & \binom{2}{1}\left(\frac{k}{2}\right) & \binom{3}{1}\left(\frac{k}{2}\right)^2 & \cdots & \binom{k}{1}\left(\frac{k}{2}\right)^{k-1} \\ 0 & 0 & 1 & \binom{3}{2}\left(\frac{k}{2}\right) & \cdots & \binom{k}{2}\left(\frac{k}{2}\right)^{k-2} \\ 0 & 0 & 0 & 1 & \cdots & \binom{k}{3}\left(\frac{k}{2}\right)^{k-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

เมื่อ $\binom{a}{b}=\frac{a!}{b!(a-b)!}$ หลังจากการแปลง จะได้เมตริกซ์สัมประสิทธิ์วงจรกรองย่อย $\hat{\mathbf{V}}$ ดังนี้

$$\hat{\mathbf{V}}=\mathbf{T}\mathbf{V}^{-1} \tag{2.21}$$

เมื่อ $\hat{\mathbf{V}}$ คือ เมตริกซ์สัมประสิทธิ์วงจรกรองย่อยใหม่
$\mathbf{T}$ คือ เมตริกซ์การแปลง
$\mathbf{V}^{-1}$ คือ เมตริกซ์แวนเดอร์มอนด์ผกผัน

วงจรกรองย่อยใหม่สามารถแสดงได้ดังนี้

$$\hat{\mathbf{V}}(z)=\hat{\mathbf{V}}\mathbf{z} \tag{2.22}$$

เมื่อ $\hat{\mathbf{V}}(z)$ คือ เวคเตอร์สัมประสิทธิ์วงจรกรองย่อยใหม่
$\hat{\mathbf{V}}$ คือ เมตริกซ์สัมประสิทธิ์วงจรกรองย่อยใหม่
$\mathbf{z}$ คือ เวคเตอร์ unit delay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

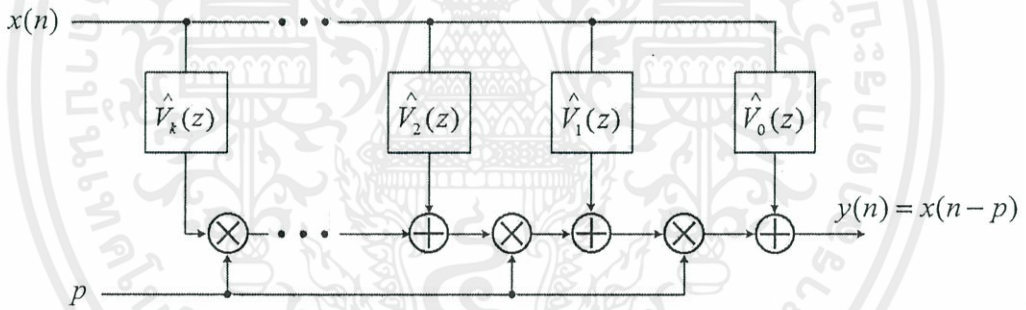
โดยกำหนดค่าพารามิเตอร์ความหน่วงของวงจรกรองย่อยใหม่ $\hat{\mathbf{V}}$ ทำการแยกค่าความหน่วง D ออกเป็นสองส่วนคือ

$$D = \frac{k}{2} + p \tag{2.23}$$

โดยเป็นตัวแปรค่าความหน่วงใหม่คือ p โดย $p \in \left[-\frac{k}{2}, \frac{k}{2}\right]$ ดังนั้นเราสามารถแสดงฟังก์ชันถ่ายโอนได้ใหม่ดังนี้

$$\hat{H}(z,p) = \sum_{i=0}^{k} \hat{V}_i(z) p^i = \mathbf{p}^{\mathrm{T}} \hat{\mathbf{V}}^{-1} z \tag{2.24}$$

และสามารถแสดงโครงสร้างแบบแฟร์โรว์ที่ได้จากเมตริกซ์การแปลงได้ดังรูปที่ (2.5)



รูปที่ 2.5 แสดงโครงสร้างแบบแฟร์โรว์ดัดแปลงที่ได้จากเมตริกซ์การแปลง [7]

ตัวอย่างกำหนดให้วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 ($k = 2$) จากสมการที่ (2.21) จะสามารถหาค่าได้ดังนี้

$$\hat{\mathbf{V}} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -3/2 & 2 & -1/2 \\ 1/2 & -1 & 1/2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1/2 & 0 & 1/2 \\ 1/2 & -1 & 1/2 \end{bmatrix}$$

จากสมการที่ (2.22) สามารถหาสัมประสิทธิ์วงจรกรองย่อยใหม่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} \hat{V}_0(z) \\ \hat{V}_1(z) \\ \hat{V}_2(z) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1/2 & 0 & 1/2 \\ 1/2 & -1 & 1/2 \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \end{bmatrix}$$

ดังนั้น

$$\hat{V}_0(z) = z^{-1}$$

$$\hat{V}_1(z) = \frac{1}{2} + \frac{1}{2} z^{-2}$$

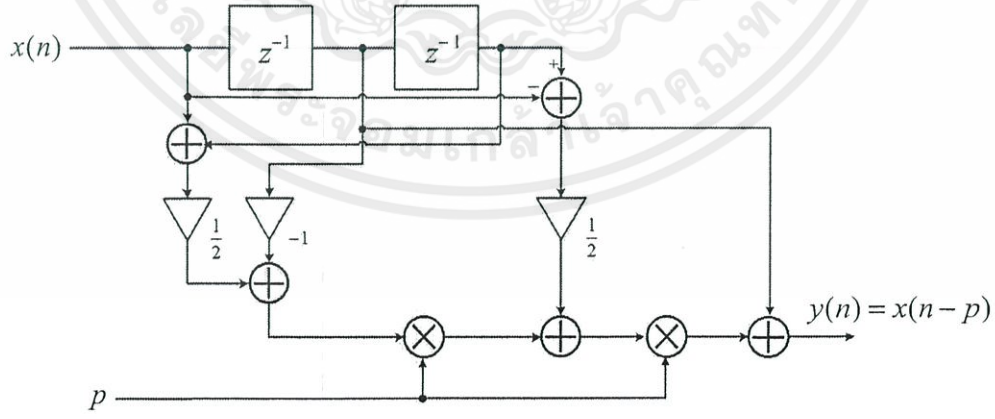
$$\hat{V}_2(z) = \frac{1}{2} - z^{-1} + \frac{1}{2} z^{-2}$$

จากสมการที่ (2.24) จะได้ฟังก์ชันถ่ายโอนวงจรกรองย่อยใหม่ได้ดังนี้

$$\hat{H}(z,p) = \hat{V}_0(z) + p\,\hat{V}_1(z) + p^2\,\hat{V}_2(z)$$

$$\hat{H}(z,p) = z^{-1} + p\left[-\frac{1}{2} + \frac{1}{2} z^{-2}\right] + p^2\left[\frac{1}{2} - z^{-1} + \frac{1}{2} z^{-2}\right] \qquad (2.25)$$

ดังนั้นเราสามารถเขียนโครงสร้างแบบแฟร์โรว์ของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ที่ได้จากการแปลงดังรูปที่ 2.6



รูปที่ 2.6 โครงสร้างแบบแฟร์โรว์ดัดแปลงของวงจรกรองสัญญาณที่ได้จากการแปลงที่อันดับตัวกรองที่ 2 (แสดงสัมประสิทธิ์วงจรกรองย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 แสดงโครงสร้างแบบแฟร์โรว์ดัดแปลงของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 ($k=2$) จะเห็นได้ว่าโครงสร้างแบบแฟร์โรว์ดัดแปลง สามารถลดความซับซ้อนของการคำนวณได้เกือบ 50 % โดยมีจำนวนของการคูณ 5 ตัวและจำนวนของการบวก 5 ตัว

ตารางที่ 2.1 จำนวนตัวคูณและตัวบวกที่ใช้ในวงจรที่อันดับตัวกรองต่างๆ

VFD Filter Structure	No. of Multiplications	No. of Additions
Farrow Structure	$k(k+1)+k$	k^2+k
Modify Farrow Structure	Odd-order $(\frac{k+1}{2})(k+1)+k$	$(k+1)+\frac{(k+1)(k-1)}{2}$
	Even-order $(\frac{k}{2}+1)\frac{k}{2}+(\frac{k}{2})^2$	$(\frac{k^2}{4}+\frac{k}{2})+(\frac{k^2-2k}{4}+\frac{k}{2})+k$

* k = อันดับของตัวกรองสัญญาณ

2.3 การเขียนภาษา VHDL

วีเอชดีแอล ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (high level language) ที่ใช้ในการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับขั้นได้ และสามารถที่จะเขียนได้หลายรูปแบบซึ่งจะกล่าวต่อไป จึงทำให้ภาษาวีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้นวีเอชดีแอลยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Terminology และ Convention

การเขียนรูปแบบของดิจิตอลด้วยภาษา VHDL นั้น จะมีศัพท์เทคนิคเฉพาะ ฉะนั้นในส่วนนี้จะเป็นการบรรยาย และอธิบายศัพท์บางคำที่จะต้องพบในรายงานชุดนี้

ลักษณะของรูปแบบ (modal styles) : ลักษณะการเขียนรูปแบบ (model) ด้วยภาษา VI IDL สามารถแบ่งได้เป็น

1. Behavioral Model : หรือที่เรียกอีกอย่างหนึ่งว่า algorithmic description เป็นรูปแบบที่บรรยายพฤติกรรมของระบบดิจิตอล ในส่วนที่บรรยายจะมีโครงสร้างคล้ายกับภาษาชั้นสูง (high level language) ทั่วไป เช่น PASCAL หรือ C เป็นต้น ในการจำลองการทำงาน (simulation) คำสั่งแต่ละคำสั่ง (statement) จะถูกประเมินผลเป็นไปตามลำดับ (sequential) จากบนลงล่าง ยกเว้นในกรณีของคำสั่ง LOOP หรือการเรียกใช้โปรแกรมย่อย รูปแบบลักษณะนี้จะไม่ให้รายละเอียดที่เกี่ยวกับการผลิต หรือโครงสร้างของ Hardware แต่ในทางตรงข้ามที่รายละเอียดเกี่ยวกับความสัมพันธ์ระหว่าง input กับ output ที่ดี

2. Dataflow Model: เรียกอีกอย่างหนึ่งได้ว่า "Register transfer level" (RTL) เป็นรูปแบบที่ถูกเขียนขึ้นเพื่อจุดประสงค์ที่จะใช้เป็นเครื่องมือสำหรับสังเคราะห์วงจรอัตโนมัติ รูปแบบลักษณะนี้ส่วนใหญ่จะเป็น procedural constructs และ functional operators

3. Structural Model: เป็นรูปแบบที่แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ต่างๆ ที่ประกอบขึ้นเป็นวงจรหรือระบบดิจิตอลและสามารถเรียกอีกอย่างได้ว่า "net list representation" เป็นการเขียนที่แสดงให้เห็นโครงสร้างของ hardware

4. Mixed – Level Model: จากคุณสมบัติที่อ่อนตัวของภาษา VHDL จึงสามารถที่จะเขียนรูปแบบโดยใช้ลักษณะต่างๆ บรรยายวงจรหรือระบบดิจิตอลเดียวกันได้ ฉะนั้นรูปแบบเช่นนี้จึงมีการเขียนแบบผสม

Concurrency ในภาษา VHDL นั้น ชุดคำสั่งจะทำงานในเวลาเดียวกันและอิสระต่อกัน ลักษณะเช่นนี้เป็นคุณสมบัติที่เป็นความจริงทางฟิสิกส์ของวงจรอิเล็กทรอนิกส์ ชุดคำสั่งนี้เรียกว่า "concurrent statement" และจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงค่าของสัญญาณ

1. Sequential: นอกจากความสามารถที่ชุดคำสั่งจะทำงานแบบ concurrent แล้วบางครั้งการเขียนรูปแบบในลักษณะที่บรรยายพฤติกรรมของวงจร มีความจำเป็นที่จะต้องให้ชุดคำสั่งทำงานเป็นลำดับขั้นเรียงกันจากบนลงล่าง อย่างเช่น การเขียนแบบ behavioral model เป็นต้น ชุดคำสั่งที่เป็น sequential นี้จะใช้ในโปรแกรมย่อย (subprogram) และ process statement

2. Driver: สัญญาณต่างๆ (signal) ใน VHDL นั้นจะถูกควบคุมด้วยตัวขับ หรือ "driver" สัญญาณเหล่านี้จะรับค่าใหม่ (ระดับของสัญญาณ) ได้ด้วยตัวขับนี้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transaction : การเกิด transaction กับ signal นั้นจะเกิดขึ้นเมื่อมีการกำหนดค่าๆ หนึ่งให้กับ signal นั้น ค่าใหม่ที่ signal ได้รับอาจจะมีผลหรือไม่มีผลทำให้เกิดการเปลี่ยนแปลงของระดับสัญญาณ (event) เช่นการเปลี่ยนค่าจาก logic "0" เป็นค่า logic "0" เป็นต้น

1. Event : คือการเปลี่ยนระดับค่าของ SIGNAL จากระดับหนึ่งไประดับอื่น อย่างเช่น ในระบบดิจิตอลการเปลี่ยนจาก logic "0" เป็นค่า logic "1" หรือในทางตรงกันข้ามถือว่า SIGNAL นั้นเกิด "event" ฉะนั้นจะเห็นได้ว่า การที่จะเกิด event ได้นั้นจะต้องเกิด transaction ไม่จำเป็นต้องเกิด event ทุกครั้ง

2. Sensitivity List: คือรายชื่อของ signal ต่างๆ ที่มีผลทำให้เกิดการทำงานของ concurrent statement เมื่อเกิด event ขึ้นกับ signal ตัวใดตัวหนึ่งหรือหลายตัวพร้อมกันในรายชื่อนั้น

3. Object: ในภาษานั้น คำว่า ใช้เขียนเพื่อบ่งบอกถึงองค์ประกอบส่วนหนึ่งของรูปแบบ ซึ่งเปรียบได้เหมือนกับภาชนะที่มีไว้สำหรับบรรจุค่าต่างๆ สามารถแบ่งออกได้เป็นสามชั้น (Class) ด้วยกันคือ

4. CONSTANT: ได้แก่ object ประเภทหนึ่งที่เมื่อกำหนดค่าเริ่มต้นให้แล้ว จะคงค่านั้นไว้ตลอด ไม่สามารถดัดแปลงหรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนประกาต่างๆ ของรูปแบบ (model)

5. SIGNAL: หมายถึง object ประเภทหนึ่งที่สามารถกำหนดค่าที่สัมพันธ์กับเวลาให้ได้นั้น หมายความว่า SIGNAL สามารถรับค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง SIGNAL จะรับค่าๆ หนึ่งได้จากขับสัญญาณ หรือ driver ซึ่งขับตัวนี้อาจจะเก็บค่าในอนาคตสำหรับ SIGNAL ไว้ด้วย SIGNAL สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้น ดังนั้น SIGNAL จึงสามารถถูกนำไปใช้ตลอดโครงสร้างของรูปหรือที่เรียกว่า global object

6. VARIABLE: หรือตัวแปรกได้แก่ object ที่สามารถกำหนดค่าใดๆ ให้ได้ และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดการจำลองการทำงาน แต่จะเก็บค่าเพียงค่าเดียวเท่านั้น ในขณะเวลาหนึ่ง เนื่องจาก VARIABLE สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้น อันได้แก่ส่วนประกอบของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น VARIABLE จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้เท่านั้น (local object)

ประเภทของ object ที่กำหนดไว้แล้ว (predefined type) : ได้แก่ TYPE ที่กำหนดไว้ใน package ชื่อ STANDARD โดย IEEE ว่าจะต้องมีในระบบที่ใช้พัฒนา VHDL ฉะนั้นจึงไม่จำเป็นต้องประกาศใช้ในทุกรูปแบบที่เขียนขึ้น TYPE ประเภทนี้ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. BOOLEAN คือกลุ่มของค่า FALSE และ TURE
2. BIT คือกลุ่มของค่า “0” และ “1”
3. INTEGER คือกลุ่มของค่า -214748347 ถึง 21474837
4. PEAK คือกลุ่มของค่า -1.0E38 ถึง 1.5E38
5. CHARACTER คือกลุ่มของค่าพยัญชนะ “A” - “Z”, “a” - “z” อักษรหรือเครื่องหมายพิเศษและตัวอักษรควบคุม
6. TIME ได้แก่หน่วยเวลาที่มีค่าพื้นฐานเป็นวินาที (second ย่อด้วย s)
7. SEVERITY LEVEL คือกลุ่มของค่า NOTE, WARING, ERROR, FAILURE ส่วนต่างในการเขียน VHD

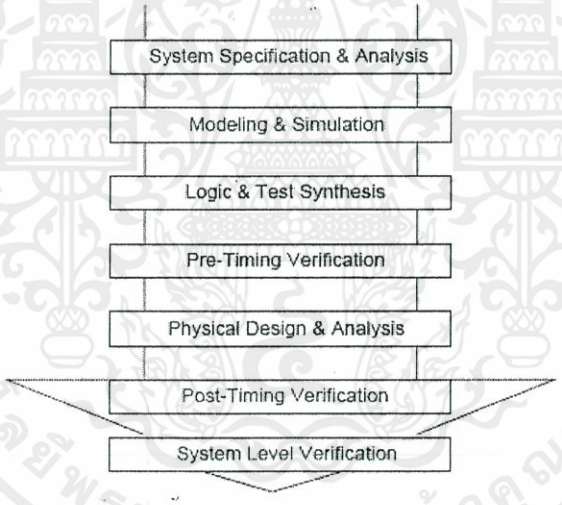
การเขียนรูปแบบหรือ modeling ด้วยภาษา VHDL มีความจำเป็นที่จะต้องแนะนำให้รู้จักกับส่วนต่างๆ ของแบบ (design units) ที่ใช้ภาษาเสียก่อน และนี่ก็เป็นขั้นตอนแรกที่สำคัญที่สุดของการศึกษา ศึกษาการเรียนรู้การใช้ภาษา VHDL เขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะ Top - Down Design นอกจากนั้นการที่จะเข้าใจในเรื่องของโครงสร้าง และส่วนต่างๆ ของรูปแบบ VHDL ให้ถูกต้องเสียก่อน

การออกแบบวงจรเชิงเลข (Digital Circuit) นั้น ในปัจจุบันก้าวหน้าไปอย่างมาก โดยการใช้ภาษาบรรยายการทำงานของระบบ (Hardware Description Language : HDL) ซึ่งเป็นภาษาที่ใช้สำหรับออกแบบฮาร์ดแวร์ โดยใช้ภาษาที่เป็นมาตรฐานสากล เช่น Verilog หรือ VHDL (VHSIC Hardware Description Language), (VHSIC : Very High Speed Integrated Circuit) หรือภาษาที่ไม่เป็นมาตรฐานเช่น AHDL (Altera Hardware Description Language) หรือ PHDL (Philips Hardware Description Language) เป็นต้น มาบรรยายการทำงานของวงจรที่ได้ออกแบบไว้ ซึ่งในปริญานิพนธ์ได้ใช้ภาษา VHDL มาทำการออกแบบวงจร Digital Oscillator ทำให้ลดความยุ่งยากในการนำเอาอุปกรณ์มาเชื่อมต่อให้เป็นวงจร รวมทั้งลดเวลาที่ใช้ในการออกแบบและทดสอบการทำงาน ซึ่งมีความแตกต่างเป็นอย่างมาก เมื่อเปรียบเทียบกับการออกแบบในอดีตที่ผ่านมา คือผู้ออกแบบจะต้องนำเอาอุปกรณ์แต่ละตัวที่ทำการออกแบบไว้ มาทำการต่อทดลองในแผงวงจรจริง และทำการทดสอบวงจรเพื่อหาข้อผิดพลาด ซึ่งต้องใช้เวลานานกับการแก้ปัญหาแต่ละอย่างที่เกิดขึ้น แต่ในการออกแบบด้วยภาษา VHDL ผู้ออกแบบเพียงแต่เขียนซอสโค๊ด (Source Code) บรรยายการทำงานของวงจร หลังจากนั้นก็ทำการคอมไพล์ (Compile) แล้วจำลองการทำงาน (Simulate) ดูว่าได้ฟังก์ชั่นการทำงานและไทม์มิ่ง (Timing) ตามที่ต้องการหรือไม่ จากนั้นก็นำซอสโค๊ดที่ได้ไปทำการสังเคราะห์ด้วยโปรแกรมสังเคราะห์ (Synthesis Tool) สุดท้ายนำวงจรที่ได้จากทำการแม็ป (Map) ลงไปยัง FPGA (Field Programmable Gate Array) เพื่อเป็นชิป (Chip) ต้นแบบสำหรับการนำไปทดสอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมเชิงเลขขนาดใหญ่ที่มีความซับซ้อน ผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อน จากนั้นจึงวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษา VHDL นั้น อนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์ เพื่อให้ได้การทำงานตามต้องการ โดยการออกแบบในลักษณะนี้เรียกว่า หลักการออกแบบจากบนลงล่าง (Top - Down Design) ซึ่งถ้าเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom - Up Design) จะเห็นได้ว่าการออกแบบมากกว่า เพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ออกแบบจำลองการทำงานตรวจสอบความถูกต้องซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากในการออกแบบลักษณะนี้ ดังนั้นการใช้ภาษา VHDL กับหลักการออกแบบจากบนลงล่าง จึงเป็นวิธีการที่เหมาะสมสำหรับการออกแบบและพัฒนาวงจรที่มีความซับซ้อนมากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ



รูปที่ 2.7 การออกแบบจากบนลงล่าง

จากรูปที่ 2.7 แสดงให้เห็นถึงขั้นตอนการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปบ้างเล็กน้อย โดยขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบเพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบ โดยใช้ภาษา VHDL สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3. ขั้นตอนการสังเคราะห์ซึ่งจะต้องทำการกำหนดเทคโนโลยีที่จะมารองรับวงจร ออกแบบและระบบช่วยออกแบบจะทำการสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท (Gate Level) และการเชื่อมต่อกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสต์ (Net list) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือเน็ทลิสต์แล้ว ข้อมูลที่ได้นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชั่นแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งจากความจริงที่ว่าอุปกรณ์อิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการเคลื่อนผ่าน (Propagation Delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาที แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชั่นต่างๆ จำนวน 10,000 เกท ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้นจนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาสูงๆ ได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้ จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของอุปกรณ์ FPGA

6. หลังจากที่ได้วงจรจริงมาแล้วยังต้องมีความจำเป็นที่จะต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จำนำวงจรไปรวมเข้ากับอุปกรณ์อื่นๆในระบบ เพราะในขั้นตอนนี้วงจรที่ออกแบบจะประกอบไปด้วยอินพุทและเอาต์พุตแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. หลังจากที่นำวงจรที่ออกแบบไปรวมเข้ากับอุปกรณ์อื่นๆให้เป็นระบบแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆอีกครั้ง ซึ่งเป็นการทดสอบการทำงานจริงขั้นสุดท้าย

2.3.3 ภาษา VHDL และส่วนประกอบต่างๆของภาษา

วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปีค.ศ. 1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ DOD (Department of Defense) ได้ทำการพัฒนาโครงการที่มีชื่อว่า VHSIC ซึ่งเป็นการพัฒนาโปรแกรมซึ่งจัดเป็นภาษาระดับสูงเช่นเดียวกับภาษา C หรือ Pascal แต่สามารถบรรยายพฤติกรรมการทำงานของวงจรเชิงเลขหรือโครงสร้างของวงจรได้ ทั้งนี้เพื่อให้สามารถออกแบบและสร้างวงจรรวมได้รวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนรูปแบบบรรยายระบบเชิงเลขในลักษณะของการออกแบบจากบนลงล่าง จะต้องทำความเข้าใจเรื่องของโครงสร้าง และส่วนประกอบต่างๆ ของรูปแบบภาษา VHDL เสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วย คือ

1. หน่วยการออกแบบเอนทิตี (Entity Design unit)
2. หน่วยการออกแบบสถาปัตยกรรม (Architecture Design unit)
3. หน่วยการออกแบบแพ็กเก็ต (Package Design unit)
4. หน่วยการออกแบบโครงแบบ (Configuration Design unit)

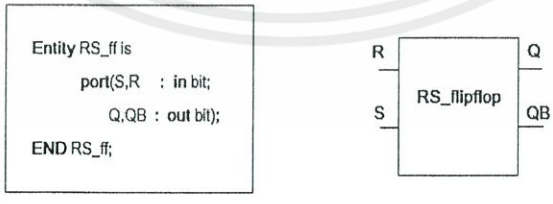
2.3.3.1 หน่วยการออกแบบเอนทิตี้

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างภายนอกกับรูปแบบที่เขียนขึ้นโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.8 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบเอนทิตี้

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

รูปที่ 2.8 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้

ส่วนนี้จะขึ้นต้นด้วยคำว่า Entity และ is ระหว่างคำทั้งสองคำเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component name) หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (input-output) รวมทั้งพารามิเตอร์อื่นๆและที่สำคัญคือหน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า End และเครื่องหมายอัฒภาคเสมอ (;)



รูปที่ 2.9 รูปแบบของ RS – flip flop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.9 เป็นหน่วยการอกแบบเอนทิตีที่บรรยายอุปกรณ์ชื่อ RS - flip flop ในส่วนหัวของเอนทิตีมีการกำหนดจุดต่อ 4 จุด ภายใต้ชุดคำสั่ง port โดยที่ 2 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ R, S ซึ่งกำหนดด้วยทิศทางการติดต่อกับโลกภายนอก เป็นการไหลเข้าของข้อมูล ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ได้แก่ Q , QB ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (out) ส่วนประเภทของข้อมูลที่จะไหลเข้าและออกนั้นเป็นประเภท bit ที่สามารถมีค่าได้เพียงสองค่าเท่านั้น คือ “0” และ “1” เท่านั้น

2.3.3.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรม คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงาน พฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดในส่วนการออกแบบเอนทิตี รูปที่ 2.10 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบสถาปัตยกรรม

```
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];
```

รูปที่ 2.10 โครงสร้างโดยทั่วไปของหน่วการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเพื่อเลือก (Option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น ประเภท (Type) ต่างๆ (ตัวอย่างเช่น bit, bit vector), สัญญาณ (signal), ค่าคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง port) นั้นจะถูกบรรยายในบริเวณเนื้อที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่าง Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้ จะเป็นชุดคำสั่งแบบแข่งขนาด (Concurrent statement) เท่านั้น คือทุกๆ statement จะทำงานพร้อมกัน ลำดับก่อนและหลังจะไม่มีผลต่อการทำงานของรูปแบบ หน่วยการออกแบบสถาปัตยกรรม จะต้องปิดท้ายด้วยคำสั่ง End และชื่อของสถาปัตยกรรมนั้นๆ โดยทั่วไป การเขียนรูปแบบระบบเชิงเลขด้วยภาษา VHDL สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

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

2.3.3.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆตลอดจนโปรแกรมย่อยที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบเชิงเลข สามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดยหน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือจากหน่วยการออกแบบแพ็กเก็ตอื่นๆ โดยปกติแล้วแพ็กเก็ตจะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็กเก็ต (Package Declaration) และส่วนของบอดี้แพ็กเก็ต (Package Body) เนื่องจากแพ็กเก็ตถูกสร้างขึ้นเพื่อเป็นส่วนแยกต่างหาก ออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

Package Declaration เป็นส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็กเก็ต สำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเอง สิ่งใดๆ ที่ถูกประกาศไว้ในส่วนของบอดี้แพ็กเก็ต แต่ไม่ได้ถูกประกาศไว้ในส่วนการประกาศแพ็กเก็ต จะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศในส่วนของการประกาศเอนทิตี คือจุดเชื่อมต่อหรือพอร์ทที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็กเก็ตสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศชนิด (Type) หรือสัญญาณเช่นเดียวกันกับส่วนบอดี้แพ็กเก็ต ที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.11 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

Package Body เป็นโครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ (Sequence) ที่ใช้บรรยายฟังก์ชั่นการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยๆ นั้น ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ต แล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ต ทั้งนี้รวมทั้งการกำหนดค่าคงที่ต่างๆ อันได้แก่ ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต ฉะนั้นส่วนบอดี้แพ็กเก็ตจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเก็ตไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.12

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

รูปที่ 2.12 โครงสร้างโดยทั่วไปของส่วนของบอดี้แพ็กเก็ต

2.3.3.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบแล้วว่ารูปแบบหนึ่งของระบบดิจิตอล ไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตี้ได้เพียงหนึ่งหน่วยเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตีหนึ่งหน่วยนี้ อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.3.4 ชุดคำสั่งลำดับ (Sequential Statements)

ภาษา VHDL สามารถใช้เขียนในรูปแบบ (modeling) บรรยายระบบดิจิตอล ในลักษณะของ behavioral description ที่โครงสร้างภายในประกอบด้วย sequential statement การศึกษาในรายละเอียดของโครงสร้างดังกล่าว สำหรับ software engineering ที่มีความคุ้นเคยกับการเขียนโปรแกรมด้วยภาษาชั้นสูง อาทิเช่น C หรือ PASCAL อยู่ก่อนแล้วจะสามารถเข้าใจโครงสร้างแบบ sequential ได้ง่าย เพียงแต่ต้องทำความเข้าใจเกี่ยวกับลักษณะการทำงานของ hardware เพิ่มเติม ในภาษา VHDL มีคำสั่งดังต่อไปนี้

1. WAIT statement
2. VARIABLE assignment
3. Signal assignment
4. IF - THEN-ELSE statement
5. CASE statement
6. Loops
7. NEXT statement
8. EXIT statement
9. RETURN statement
10. NULL statement
11. Procedure call
12. ASERTION statement

จะกล่าวเฉพาะ statement ที่สำคัญๆ เพื่อความเข้าใจในการทำงานแบบ sequential เท่านั้น ตามที่เคยกล่าวมาแล้วว่าภาษา VHDL เป็นภาษาที่มีคุณสมบัติแบบแข่งขันนานนั้น คือชุดคำสั่งภายในตัวโครงสร้างจะเป็นชุดคำสั่งแบบแข่งขนาน เช่นเดียวกับภาษา ADA ชุดคำสั่งลำดับ หรือ sequential statement ที่เรียกว่า อันได้แก่ process statement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.1 Process Statement

หัวใจสำคัญของ Concurrent shell ที่ทำให้สามารถเขียน VHDL model เพื่อบรรยายพฤติกรรมของระบบดิจิตอลอิเล็กทรอนิกส์ในลักษณะ behavioral description ได้แก่คำสั่ง process ที่โครงสร้างภายในจะประกอบด้วยชุดคำสั่งแบบลำดับเท่านั้น ชุดคำสั่งเหล่านี้จะทำงานเป็นลำดับจากบนลงล่าง เมื่อ PROCESS ถูกกระตุ้นให้ทำงาน

คำ PROCESS ในบรรทัดแรกของโครงสร้าง แสดงถึงจุดเริ่มต้นของชุดคำสั่ง process ในบางกรณีใน architecture หนึ่งอาจจะมีโครงสร้างของชุดคำสั่ง process หลายชุดได้ คำ END PROCESS บอกถึงจุดสิ้นุดของชุดคำสั่ง process เมื่อคำสั่ง END PROCESS ถูกปฏิบัติแล้วชุดคำสั่ง process จะหยุดการทำงานลงชั่วขณะ (แต่ยังคง active อยู่ตลอดเวลา) จนกว่าจะมีสัญญาณอย่างน้อยตัวหนึ่งใน sensitivity list เกิด event ขึ้นอีกชุดคำสั่ง process เป็นชุดคำสั่งแบบแข่งขนาน นั้นหมายความว่าโดยปกติแล้วชุดคำสั่ง process จะทำงานตลอดเวลา การที่ชุดคำสั่งทั้งหลายอยู่ภายในบล็อกของชุดคำสั่ง process เป็นชุดคำสั่งแบบลำดับนั้น ถ้าคำสั่ง process ใดที่ไม่มี sensitivity list เป็นตัวควบคุมการทำงาน จะทำให้เกิดการทำงานที่เปรียบเทียบเสมือนว่าเป็นวงรอบ (loop) ที่ไม่รู้จบขึ้น วิธีการป้องกันการเกิดเหตุการณ์เช่นนี้ คือการเติม wait statement ลงในส่วนของชุดคำสั่ง process

2.3.4.2 Wait Statement

ชุดคำสั่ง process สามารถมี sensitivity list ได้เพียงอันเดียว หมายความว่า ชุดคำสั่ง process จะถูกกระตุ้นได้จากการที่สัญญาณหนึ่งในรายชื่อที่เกิด event ขึ้นเท่านั้น หลังจากที่ถูกกระตุ้นแล้ว คำสั่งทั้งหลายที่อยู่ภายในจะทำงานแบบลำดับลงมาจนกระทั่งหมด และชุดคำสั่ง process จะหยุดการทำงานชั่วคราวจนกว่าจะมี event เกิดขึ้นอีกกับสัญญาณตัวใดตัวหนึ่งในรายชื่อนั้นอีก ถ้ากรณีที่ภายในรายชื่อประกอบด้วยสัญญาณหลายตัว และเกิด event ขึ้นในเวลาเดียวกัน จะมีสัญญาณเพียงตัวเดียวจากทั้งหมดเท่านั้น ที่กระตุ้นการทำงานของชุดคำสั่ง process ซึ่งไม่สามารถที่จะบอกได้ว่าเป็นตัวใด ดังนั้นการใช้ชุดคำสั่ง process ร่วมกับ sensitivity list จึงมีขีดจำกัดอยู่มากในภาษา VHDL มีวิธีการหลีกเลี่ยงปัญหาเช่นนี้ โดยใช้ชุดคำสั่ง wait statement

คำสั่ง wait statement นั้นให้ความคล่องตัวกว่า เพราะ wait statement มีข้อดีคือ ประการแรกสามารถที่จะกำหนดลงตรงตำแหน่งใดๆ ภายในโครงสร้างของชุดคำสั่งลำดับได้ เพื่อระงับการทำงานภายในชุดคำสั่ง process ตรงตำแหน่งที่ wait statement อยู่ ส่วนการใช้ sensitivity list เป็นตัวควบคุมการทำงานชุดคำสั่ง process จะหยุดตรงตำแหน่งสุดท้ายของชุดคำสั่ง process เท่านั้น ประการที่สอง สามารถที่จะกำหนด wait statement มีหลายรูปแบบ นั่นหมายความว่าสามารถที่จะใช้ควบคุม PROCESS ได้หลายลักษณะ แต่มีสิ่งที่ต้องจำไว้ว่า wait statement ไม่สามารถที่จะใช้ร่วมกับ sensitivity list ภายใน PROCESS เดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉะนั้นผู้เขียนรูปแบบจึงต้องตัดสินใจก่อนว่าจะใช้อะไร ทั้งนี้ขึ้นอยู่กับระบบดิจิตอลที่จะเขียนบรรยายและประสบการณ์ในการเขียน ในภาษา VHDL สามารถใช้ wait statement ได้ 4 แบบคือ

```
1. WAIT ON signal list        -- signal sensitivity
2. WAIT UNTIL condition;      -- condition
3. WAIT FOR time;             -- timeout
4. WAIT;                      -- forever
```

แต่ละอย่างมีความหมายและวิธีที่แตกต่างกันตามตัวอย่างต่อไปนี้

- WAIT ON clock, clear, preset, d;

คำสั่งจะหยุดการทำงานของชุดคำสั่งลำดับไว้ จนกว่าจะเกิด event ขึ้นที่สัญญาณ clock หรือ clear หรือ preset หรือ d

- WAIT UNTIL (clock = '1');

การทำงานของชุดคำสั่งลำดับจะหยุดที่ตำแหน่งนี้ และจะทำงานต่อไปเมื่อสัญญาณ clock เกิด event และ Boolean expression เป็น TRUE ทุกครั้งเมื่อลำดับการทำงานถูกหยุดตรงจุดนี้ และก่อนที่จะทำต่อไปได้นั้นต้องตรวจสอบว่ามี event เกิดขึ้นบนสัญญาณ clock หรือเปล่า และสัญญาณ มีค่าเป็น '1' ฉะนั้น Boolean expression จะต้องเป็นจริง

- WAIT FOR 10 NS;

คำสั่งนี้เป็นการรอหรือหยุดในขณะที่จำลองการทำงานจนกว่าเวลาการจำลองจะล่วงเลยไปแล้ว 10 ns. จะมีผลทำให้ PROCESS เริ่มทำงานต่อไป

- WAIT;

เป็นคำสั่งให้ PROCESS หยุดทำงานตลอดไป โดยไม่มีการเริ่มต้นการทำงานใหม่ บางครั้งการเขียนรูปแบบบรรยายการทำงาของระบบดิจิตอล มีความจำเป็นที่ต้องใช้คำสั่งนี้ เพื่อหยุดการทำงานอย่างถาวร นอกจากนั้นยังสามารถใช้ควบคุม มี sensitivity list หรือ wait statement อย่างใดอย่างหนึ่ง เพื่อป้องกันปัญหาอันอาจจะเกิดขึ้นได้

2.3.4.3 IF-THEN-ELSE statement

เป็นชุดคำสั่งพื้นฐานที่ใช้สอบถาม เพื่อการตัดสินใจกระทำอะไรบางสิ่งบางอย่าง

<u>IF statement</u>

```
Format      IF condition THEN
            Sequential_ statement
            END IF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.4 CASE statement

โครงสร้างอีกอันหนึ่งที่เป็นลำดับ และมีความคล้ายคลึงกับ IF-THEN - ELSE คือ CASE statement เพราะใช้เป็นคำสั่งเลือกหนทางปฏิบัติ ตามข้อแม้ (condition) ที่กำหนดให้คำสั่ง CASE และ END CASE กำหนดจุดเริ่มต้นและจุดสิ้นสุด คำสั่ง WHEN ใช้สำหรับกำหนดตัวเลือกที่จะนำมาเปรียบเทียบกับ expression ที่กำหนด PROCESS จะเริ่มต้นทำงานที่ชุดลำดับคำสั่งลำดับที่ตามมา จนกระทั่งคำสั่งสุดท้ายของตัวเลือกนั้นๆ และจะออกจาก CASE statement โดยไม่ทำหนทางเลือกอื่นๆ ที่ยังคงเหลืออยู่

2.3.5 ชุดคำสั่งแบบแข่งขนาด (Concurrent Statement)

ภาษา VHDL เป็นภาษาที่มีการทำงานในลักษณะแข่งขนาน concurrency หรือสามารถที่จะมองชุดคำสั่งแบบแข่งขนานแต่ละอันเป็น PROCESS ทำงานอิสระไม่ขึ้นต่อกัน ที่เรียกว่า asynchronous ดังนั้น concurrency statement ส่วนใหญ่จึงสามารถเขียนแทนได้ด้วย PROCESS statement

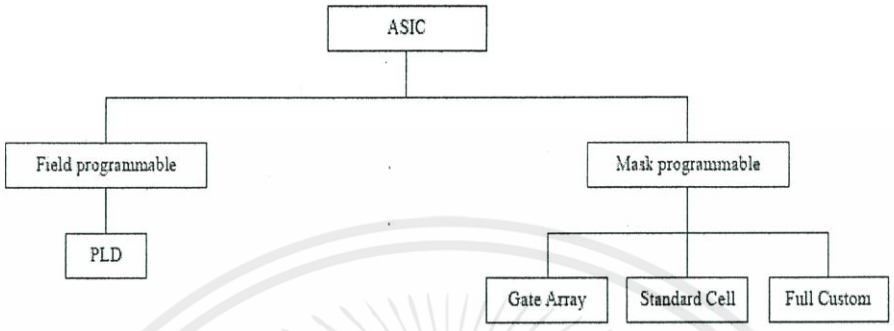
ชุดคำสั่งแบบ sequential ไม่สามารถที่จะใช้ในรูปของชุดคำสั่งแบบแข่งขนานได้ มีบางคำที่สามารถใช้ได้ใน VHDL ได้ทั้งสองรูปแบบ เช่น signal assignment เป็นต้น ชุดคำสั่งที่ใช้ในโครงสร้างแบบ concurrent ซึ่งมีทั้งหมดได้แก่

1. Signal assignment statement
2. Component instantiation statement
3. Assert statement
4. Generate statement
5. Process statement
6. Procedure statement
7. Block statement

2.4 FPGA ((Field Programmable Gate Array)

ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาขีดความสามารถของอุปกรณ์ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่ายการสิ้นเปลืองพลังงานและขนาดในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโพรเซสเซอร์และหน่วยความจำปัจจุบันทุกๆครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้นในการพัฒนาเพิ่มความหนาแน่นและจำนวนฟังก์ชั่นลอจิกที่เหมาะสมนักออกแบบอุปกรณ์ทางด้านดิจิตอลได้พิจารณาถึงการผลิตให้ขนาดมากๆและการผลิตวงจรรวม (ASIC : Application Specific Integrated Circuit) ซึ่งวงจรรวมจะแบ่งตามการสร้างออกเป็น 2 กลุ่มคือ Field programmable และ Mask programmable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.14 ผังแสดงการแบ่งกลุ่มของวงจรรวม ASIC

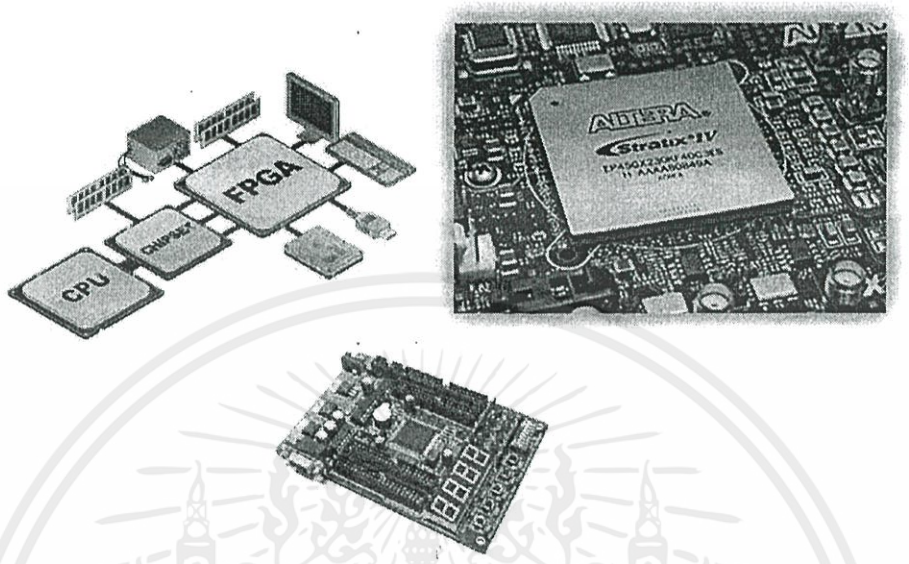
2.4.1 การออกแบบวงจรด้วยอุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เปนไอซีหรือชิพอเนกประสงคที่สามารถโปรแกรมใหเปนวงจรดิจิตอลอะไรก็ไดโดยวิธีการโปรแกรมแบบงายๆและสามารถแกไขวงจรไดอยางสะดวกดวยการโปรแกรมซ้ำ FPGA จะเหมาะสำหรับการออกแบบวงจรดิจตอลขนาดกลางจนถึงวงจรขนาดใหญมากๆ เชน ไมโครโพรเซสเซอร ดิจิตอลฟลเตอร สวนประกอบหลักของแผงวงจรดิจิตอลของอุปกรณทางดานการแพทย เครื่องมือวัดตางๆ อุปกรณสื่อสาร หรือ อุปกรณเครือขาย นักออกแบบดิจิตอลที่เคยออกแบบวงจรขนาดใหญและวงจรดิจิตอลที่ซับซอนมากๆ ก็จะทราบวาอาจมีความจำเปนตองใช FPGA ขนาดใหญที่มีความจุวงจรมากตั้งแตขนาดหลายหมื่นเกตจนไปถึงระดับหลักหลายลานเกต

ในการโปรแกรมวงจรที่ได้ออกแบบเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามแบบที่ออกแบบไว้ในการทำ FPGA ซึ่งเป็นวิธีการออกแบบ IC (Integrated Circuit) แบบ Semicustom อีกวิธีหนึ่ง เมื่อเทียบกับการทำ ASIC แล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือการทำFPGA จะมีข้อจำกัดในด้านขนาดของวงจร เพราะภายในอุปกรณ์ FPGA จะมีจำนวนเกท (Gate) ให้ใช้จำนวนจำกัดและการทำ FPGA ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดนั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

การทำ FPGA ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายใน หรือปรับปรุงโครงสร้างสถาปัตยกรรมภายใน และยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้อุปกรณ์นั้นๆ ด้วย ลักษณะของตัว FPGA และการนำไปใช้งานแสดงดังใน รูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ลักษณะของตัว FPGA และการนำไปใช้งาน

สำหรับตัวอุปกรณ์ FPGA นั้นก็มีโครงสร้างพื้นฐานเทคโนโลยีที่ใช้สร้างตลอดจนเทคโนโลยีวิธีการโปรแกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นอุปกรณ์FPGA ของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วนในการใช้งานอุปกรณ์ FPGA สามารถนำไปประยุกต์ใช้งานได้ เช่นการประมวลผลสัญญาณเชิงเลข (DSP : Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์เป็นต้น

2.4.2 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

ในการออกแบบวงจรเชิงเลขนั้น ทำได้โดยการวาดวงจรหรือใช้ภาษาอธิบายฮาร์ดแวร์ ในขั้นตอนนี้เป็นขั้นตอนที่ไม่แตกต่างกันระหว่างการออกแบบด้วย FPGA และ ASIC ในกรณีที่ใช้ภาษาอธิบายฮาร์ดแวร์ แต่ในกรณีที่ออกแบบโดยวิธีการวาดวงจรจะแตกต่างกัน โดยที่การทำวิธีนี้จะต้องคำนึงถึงเทคโนโลยีที่จะใช้ ซึ่งแต่ละเทคโนโลยีก็มีความแตกต่างกันไป จะเห็นได้ว่าการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ ทำได้สะดวกกว่าเพราะการทำด้วยวิธีนี้สามารถที่จะแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี

ในการเขียนโค้ดสิ่งที่ต้องคำนึงถึง คือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรได้ และให้คุณสมบัติของวงจรเป็นไปตามที่กำหนดเพราะลักษณะการเขียนโค๊ดจะส่งผลโดยตรงกับวงจรที่ได้ เนื่องจากในการสังเคราะห์วงจรนั้นซอฟต์แวร์สังเคราะห์วงจรจะทำการสังเคราะห์ตามโค๊ดที่เขียน ถ้าอธิบายการทำงานของวงจรเดียวกัน แต่เขียนโค๊ดในลักษณะที่ต่างกัน เมื่อสังเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วจะได้ใช้วงจรที่ต่างกันและจากวงจรที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันในด้านของขนาดหรือความเร็ว ส่วนการเขียนโค๊ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

2.4.2.1 การจำลองการทำงานของวงจร (Simulation)

ขั้นตอนนี้เป็นขั้นตอนสำคัญเพราะเป็นขั้นตอนที่ใช้ตรวจสอบฟังก์ชั่นการทำงานของวงจรว่าถูกต้องหรือไม่มีข้อผิดพลาดตรงไหน เพื่อที่จะได้ทำการแก้ไขได้ถูกต้องในขั้นตอนนี้จะใช้ซอฟต์แวร์สำหรับทำการจำลองการทำงานของวงจรเช่น V-System และ Model SIM ของบริษัท Model Technology

2.4.2.2 การสังเคราะห์วงจร

ในขั้นตอนนี้จะใช้ซอร์ฟแวร์สังเคราะห์วงจร (Synthesis tools) ทำการสังเคราะห์โค้ดเพื่อให้ได้เป็นวงจรขึ้นมา แต่ต้องตรวจสอบด้วยว่าซอฟต์แวร์นั้นๆสนับสนุนเทคโนโลยี FPGA Library ที่ต้องการใช้หรือไม่ โดย FPGA ที่นิยมใช้งาน เช่น ของบริษัท Xilinx ตระกูล XC4000 และบริษัท Altera ตระกูล Exemplar Logic ซึ่งในขั้นตอนนี้ซอฟต์แวร์สังเคราะห์วงจรจะแปลงโค้ด และทำการออปติไมซ์ (Optimization) เพื่อให้ได้วงจรตามเทคโนโลยีที่เลือกใช้ นอกจากนี้ยังสามารถกำหนดข้อบังคับสำหรับวงจรได้ เช่นข้อบังคับในเรื่องของเวลาหรือข้อบังคับในเรื่องของพื้นที่ ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอนการออปติไมซ์เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในเรื่องการการออปติไมซ์คือการเทียบ (Mapping) วงจรให้เข้ากับเทคโนโลยีที่เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างสถาปัตยกรรมภายในอุปกรณ์ FPGA ในกรณีของXilinxตระกูล XC4000 และบริษัท Altera ตระกูล FLEX 10 K จะเทียบโดยใช้วิธี LUT (Look Up Table) เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอฟต์แวร์สังเคราะห์วงจรก็จะมีการรายงานผลว่าวงจรที่ออกแบบไปนั้นเป็นอย่างไร เช่น มีความหน่วงเท่าไร ใช้ทรัพยากรต่างๆ ใน FPGA อะไรบ้าง เป็นต้น

2.4.2.3 การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ให้เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือ ให้แต่ละส่วนที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อช่วยลดความหนาแน่นในตอนทำการเชื่อมสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำ โดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจร เช่น เกท(Gate), ฟลิปฟลอป (Flip flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ภายในอุปกรณ์ FPGA(CLBs, IOBs, BUFT) และ(Edge Decoder) หลังจากทำขั้นตอนนี้เสร็จแล้วสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนซอฟต์แวร์ที่ใช้ในขั้นตอนนี้ขึ้นอยู่กับตัว FPGA ที่ใช้งาน เช่น FPGA ของบริษัท Xilinx จะใช้ Xilinx Foundation Series 2.1i ซึ่งซอฟต์แวร์ตัวนี้จะรวมเอา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ย่อยอื่นๆ อีก เพื่อให้ทำการ PPR (Partitioning Placement and Routing) เป็นไปอย่างต่อเนื่อง ส่วน FPGA ของบริษัทAltera จะใช้ Altera MAX + II

2.4.2.4 การวางอุปกรณ์ (Placement)

ขั้นตอนการเลือกที่ตั้งแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควรจะอยู่ในตำแหน่งใดในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญ เพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้ว จะทำให้ความหน่วงเพิ่มขึ้นหรือตัว Routerทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

2.4.2.5 การเชื่อมต่อสัญญาณ (Routing)

ขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดี ซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมด หรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับ โดยสามารถทำขั้นตอนนี้ได้ โดยใช้ซอฟต์แวร์เช่นกัน หรือทำการเชื่อมต่อสัญญาณด้วยตนเอง (Manual Layout) ก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่าโดยให้ทำการค้นหาเส้นทางหลายๆครั้ง เพื่อหาครั้งที่ดีที่สุด นอกจากนั้นการกำหนดข้อบังคับทางเวลาจะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

2.4.2.6 การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่วงจรผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR(Partitioning, Placement and Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลดลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้ให้เป็นข้อมูลวงจร (Configuration data) ซึ่งอยู่ในรูปของบิทสตรีม(Bit-Stream) ก่อน แล้วจึงดาวน์โหลดลงไป เพื่อให้อุปกรณ์ FPGA มีฟังก์ชั่นการทำงานตามวงจรที่ออกแบบไว้

จากที่อธิบายมาทั้งหมดนี้ เห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้น ทำได้สะดวกว่าการทำ ASIC มาก เพราะใช้เวลาน้อยกว่ามาก ส่วนสำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่การเขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงในอุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็นซอฟต์แวร์ที่ใช้ทำงานต่อเนื่องกัน

2.4.3 FPGA Discovery-III XC3S200

หลักการทำงานของบอรดอเนกประสงคชิพ FPGA ตระกูล Spatan3 เบอรนี้มีความโดดเดน คือ มีหนวยความจำแบบ RAM รวมกันมากถึง 216Kb และมีตัวคูณที่เปนฮารดแวรมากถึง 12 ชุด รวมทั้ง DCM อีก 4 ชุด ซึ่งเหมาะสำหรับงานที่ตองใชไมโครคอนโทรลเลอรแบบฝงตัวทำงานรวมกับวงจรดิจิตอลอื่นๆ จึงทำใหการออกแบบวงจรทำไดอยางสะดวกและมีประสิทธิภาพ

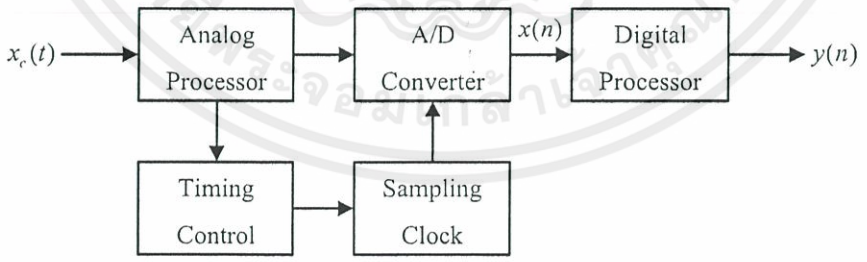
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การประยุกต์ใช้งาน

วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบราบเรียบที่สุดชนิดลากรานจ์ในโครงสร้างแฟร์โรว์และในโครงสร้างแบบบแฟร์โรว์ดัดแปลงสามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยจะยกตัวอย่างการนำมาประยุกต์งานได้ดังนี้

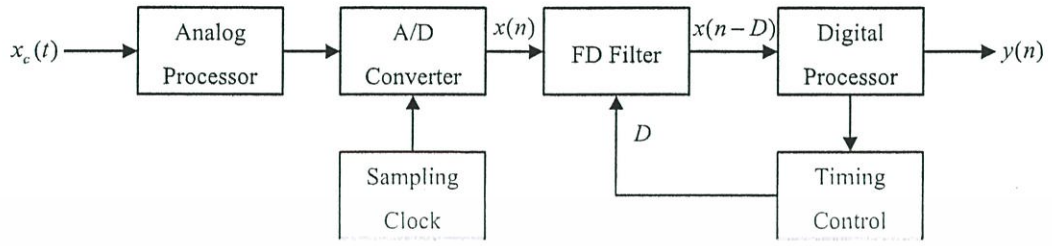
2.5.1 การเข้าจังหวะในโมเด็มสัญญาณเชิงเลข (Synchronization in digital modem)

ในปัจจุบันการสื่อสารจะประมวลสัญญาณที่เป็นเชิงเลข ซึ่งการสื่อสารข้อมูลที่เป็นสัญญาณเชิงเลขจะมีจุดเริ่มต้นมากจากสัญญาณอนาล็อกจากนั้นจะผ่านกระบวนการแปลงเพื่อเป็นสัญญาณเชิงเลข เนื่องจากการประมวลสัญญาณอนาลอกจะมีผลต่อปัจจัยภายนอกได้แก่ อุณหภูมิ การสูญเสียของสัญญาณ ตลอดจนการควบคุมให้ระบบมีเอาต์พุตที่ตรงตามที่ต้องการได้ยากกว่า ดังนั้นในหัวข้อนี้จะแสดงตัวอย่างของการเข้าจังหวะในโมเด็มสัญญาณเชิงเลขที่เป็นแบบเดิมดังรูปที่ 2.16 จะเห็นได้ว่าระบบมีการจัดการข้อมูลในส่วนที่เป็นสัญญาณอนาลอกอินพุต $x_c(t)$ และการควบคุมเวลา (timing control) ของสัญญาณการชักตัวอย่าง (sampling clock) จะอยู่ภายใต้การควบคุมของส่วนประมวลผลสัญญาณอนาลอก (analog processor) เมื่อเสร็จกระบวนการจึงค่อยแปลงเป็นสัญญาณเชิงเลข $x(n)$ หากเรามองระบบจากรูปที่ 2.16 จะพบว่าระบบมีการควบคุมเวลา (timing control) ในกระบวนการของสัญญาณอนาลอกซึ่งอาจจะเกิดค่าผิดพลาดได้จากปัจจัยที่กล่าวมาแล้วขั้นต้น ดังนั้นเราสามารถแก้ไขได้โดยให้ระบบมีการควบคุมเวลา (timing control) ในกระบวนการของสัญญาณเชิงเลข ดังรูปที่ 2.17 ซึ่งสามารถทำให้ระบบมีประสิทธิภาพได้โดยนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) มาช่วยในการปรับการควบคุมเวลา (timing control) ให้มีค่าแม่นยำตามที่ต้องการได้ตลอดด้วยการปรับเปลี่ยนค่าความหน่วงของวงจรกรอง (delay parameter)



รูปที่ 2.16 แสดงการควบคุมเวลาของสัญญาณการชักตัวอย่างของส่วนประมวลผลสัญญาณอนาลอก

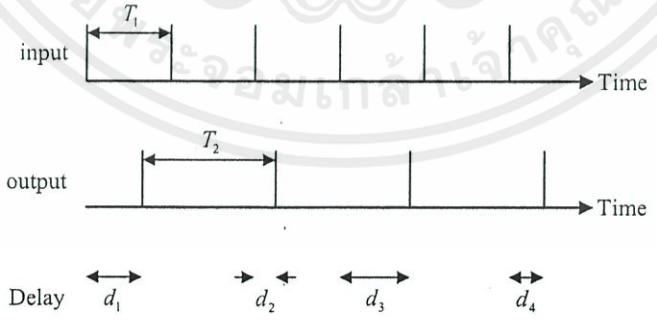
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงการควบคุมเวลาของสัญญาณการชักตัวอย่างของ
ส่วนประมวลผลสัญญาณเชิงเลข

2.5.2 วงจรแปลงอัตราการชักตัวอย่าง (Sampling rate conversion: SRC)

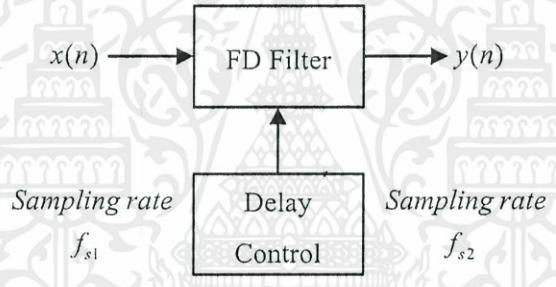
ในการประมวลสัญญาณเชิงเลขสิ่งที่สำคัญสิ่งหนึ่งคืออัตราการชักตัวอย่าง (sampling rate) ซึ่งในการปฏิบัติใช้ในงานแต่ละส่วนหรือแต่ละอย่าง อัตราการชักตัวอย่างของความถี่มีความเป็นไปได้ที่จะมีค่าไม่เท่ากัน ซึ่งในการประมวลสัญญาณเชิงเลขจะทำให้แต่ละส่วนสามารถเชื่อมโยงหรือทำงานร่วมกันได้โดยไม่เกิดความผิดพลาดในการประมวลสัญญาณเชิงเลขย่อมต้องประกอบด้วยส่วนที่สามารถปรับเปลี่ยนอัตราการชักตัวอย่างให้มีค่าที่ตรงตามต้องการ ซึ่งก็คือวงจรแปลงอัตราการชักตัวอย่าง (sampling rate conversion: SRC) ยิ่งในความเป็นจริงแล้วเมื่อนำวงจรแปลงอัตราการชักตัวอย่างไปใช้งานในทางปฏิบัติอัตราการชักตัวอย่างอาจมีค่าที่ไม่คงที่ตลอด ดังนั้นการนำวงจรแปลงอัตราการชักตัวอย่างมาใช้ต้องเป็นวงจรที่มีความยืดหยุ่นในการปรับเปลี่ยนอัตราการชักตัวอย่าง หรือเป็นวงจรที่มีคุณลักษณะที่ปรับเปลี่ยนได้แบบทันทีทันใดหรือตลอดเวลา (real time) นั่นคือเราสามารถประยุกต์ใช้งานของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) ซึ่งมีคุณสมบัติที่สามารถปรับเปลี่ยนได้แบบทันทีทันใดหรือตลอดเวลา (real time) ด้วยการปรับเปลี่ยนค่าความหน่วงของวงจรกรอง (delay parameter) ดังรูป



รูปที่ 2.18 แสดงสัญญาณอินพุตที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s1} แปลงไปสู่สัญญาณเอาต์พุต
ที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s2}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.18 แสดงให้เห็นถึงสัญญาณอินพุตที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s1} และต้องการให้ได้สัญญาณเอาต์พุตที่มีอัตราการชักตัวอย่างที่ความถี่ f_{s2} ซึ่งสามารถปรับเปลี่ยนอัตราการชักตัวอย่างได้โดยการปรับเปลี่ยนค่าความหน่วงของวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) โดยจะเห็นได้ว่าค่าความหน่วงในแต่ละช่วงเวลาจะมีค่าที่ไม่เท่ากัน เพราะว่าค่าความหน่วงจะต้องปรับเปลี่ยนเพื่อให้ได้สัญญาณเอาต์พุตที่มีอัตราการชักตัวอย่างที่ถูกต้อง ซึ่งถ้าหากนำวงจรกรองสัญญาณเชิงเลขแบบอื่นมาประยุกต์ใช้งาน วงจรกรองสัญญาณอื่นอาจจะไม่ให้ผลตอบสนองแบบทันทีทันใดเท่าวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วน (VFD filter) ดังนั้นเราสามารถแสดงแผนผังของระบบวงจรแปลงอัตราการชักตัวอย่าง (sampling rate conversion: SRC) ได้ดังรูปที่ 2.19 เมื่อสัญญาณอินพุตคือ $x(n)$ และสัญญาณเอาต์พุตคือ $y(n)$ และระบบสามารถแปลงอัตราการชักตัวอย่างจากความถี่ f_{s1} ไปสู่อัตราการชักตัวอย่างความถี่ f_{s2} ได้ด้วยการปรับค่าความหน่วง (D)



รูปที่ 2.19 แสดงแผนผังของระบบวงจรแปลงอัตราการชักตัวอย่าง

2.5.3 การทำ Oversampling D/A

ปัจจุบันการเล่นเครื่องเสียงผ่าน Digital To Analog Converter หรือ DAC กำลังได้รับความนิยมอย่างสูง ทั้ง นี้เพราะ DAC สามารถพัฒนาคุณภาพเสียงในระบบเสียงให้ดีขึ้นอย่างชัดเจนในงบประมาณที่ต่ำกว่าการใช้ เครื่องเล่นซีดีราคาแพงอย่างมาก ทั้งยังสะดวกเพราะสามารถใช้เล่นกับ CD Transport, CD Player, DVD Player และเล่นไฟล์เพลงจากคอมพิวเตอร์ DAC ทั่วไปในท้องตลาดเกือบ 100% นั้นทำงานในระบบ Oversampling ซึ่งจะทำการทวีคูณความถี่ มาตราฐานจากแผ่นซีดีที่ปกติจะอยู่ที่ 16 Bit/44.1 kHz ขึ้นไปเป็น 2 เท่า 4 เท่า 8 เท่า หรือมากกว่า โดย เชื่อว่าหากทำการ Oversampling ขึ้นไปมากเท่าไร คุณภาพเสียงจะยิ่งดีขึ้นมากเท่านั้น

Oversampling คือ การ quantized จริงๆ ทำเกินกว่า 1 ครั้ง แล้วเอาค่าเฉลี่ยของความสูงของ คลื่น ออกมาเป็น 1 ค่าแทน สมมติ จากรูป ว่า มีค่า oversampling 32

เอกสารนี้ ... ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่า เราจะได้ค่า Spec เป็น 1 channel - 4 bit - 1 Hz - oversampling 32 time แทน และ Data ที่ออกมา ควรจะมีค่า เป็น 00 (<- 4 Bit) ด้วยเพราะ ผลรวม ของพื้นที่ใน กราฟ = 0 พอดี ถ้าค่า oversampling เป็น 16,8,4 เท่า เราจะได้ค่า Spec เป็น

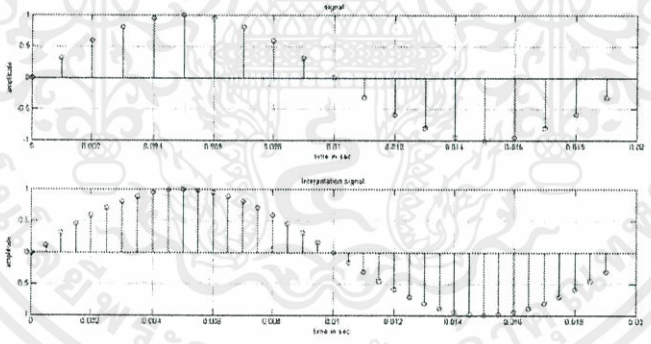
1 channel - 4 bit - 2 Hz - oversampling 16 time

1 channel - 4 bit - 4 Hz - oversampling 8 time

1 channel - 4 bit - 8 Hz - oversampling 4 time

จะเห็นได้ว่า Hz ของ ADC/DAC กับ Hz ของเสียงเป็นคนละตัวกันกับของ ADC/DAC จะบอกว่าใน 1 วิ มีจุดกี่จุดบนกราฟเท่านั้นเอง

สัญญาณที่มีลักษณะความละเอียดต่ำจะสามารถทำให้มีความละเอียดของสัญญาณที่สูงขึ้นได้โดยทำการทำ Interpolation ซึ่งการทำ Interpolation คือกระบวนการหนึ่งของการทำ Oversampling D/A ดังนั้นวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบราบเรียบที่สุดชนิดลากรานจ์ในโครงสร้างแฟร์โรว์สามารถประยุกต์ใช้งานในการทำ Over sampling D/A ได้โดยการประมาณค่าในช่วงจุดแซมเปิ้ลที่ติดกันโดยความแม่นยำในการประมาณค่านั้นจะขึ้นอยู่กับอันดับของตัวกรองที่ใช้ ยิ่งใช้อันดับของตัวกรองมากก็จะยิ่งมีความละเอียดมากในการประมาณค่าในช่วง ดังรูปที่ 2.20



รูปที่ 2.20 ผลการเปรียบเทียบของสัญญาณคลื่นไซน์กับสัญญาณคลื่นไซน์ที่ผ่านการทำ Oversampling D/A 2 เท่า

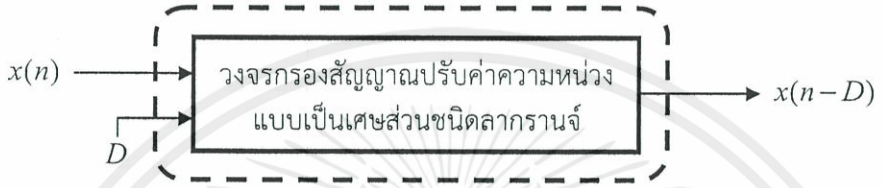
การนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนแบบราบเรียบที่สุดชนิดลากรานจ์ในโครงสร้างแฟร์โรว์มาประยุกต์ใช้เพื่อทำ Oversampling D/A เปรียบเสมือนเป็นการทวีคูณความถี่ดังที่ได้กล่าวมา ส่วนมากจะเน้นไปที่เครื่องเสียงเป็นส่วนใหญ่ซึ่งในยุคปัจจุบันนี้ความต้องการความละเอียดของเสียง และภาพเป็นจำนวนมาก ฟิลเตอร์ชนิดนี้จึงสามารถนำมาใช้งานได้จริงในอุตสาหกรรมทางด้านเทคโนโลยีเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์



รูปที่ 3.1 Block diagram การทำงานของวงจรกรองสัญญาณปรับค่าความหน่วง แบบเป็นเศษส่วนชนิดลากรานจ์

จากรูปที่ 3.1 จะเป็น Block diagram การทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ ซึ่งในการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์นี้จะใช้ตัวกรองอันดับที่ 2 และ 5 ในโครงสร้างแบบแฟรโรว์และโครงสร้างแบบแฟร์โรว์ดัดแปลง โดยจะแสดงการออกแบบในรูปของ Matrix ดังนี้ จาก Vandermonde Matrix จะได้

3.1.1 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 2 ในโครงสร้างแบบแฟร์โรว์

การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์นี้จะแสดงค่าสัมประสิทธิ์ย่อยในรูปของ Matrix ซึ่งจะเห็นได้ว่าวงจรกรองที่ออกแบบมานี้มีค่าสัมประสิทธิ์เท่าใด และต้องใช้ทรัพยากรณ์มากเพียงใด โดยการใช้ Vandermonde Matrix ดังนี้

$$\mathbf{V}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^k \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & k & k^2 & \cdots & k^k \end{bmatrix}, \quad z=\begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ \vdots \\ z^{-k} \end{bmatrix}$$

จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mathbf{V}=\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 2 & 4\end{bmatrix},\quad z=\begin{bmatrix}1\\ z^{-1}\\ z^{-2}\end{bmatrix} \tag{3.1}$$

สัมประสิทธิ์ย่อยของ $\mathbf{V}(z)$ สามารถหาค่าได้จาก

$$\mathbf{V}(z)=\mathbf{V}^{-1}z$$

จะได้

$$\mathbf{V}^{-1}=\begin{bmatrix}1 & 0 & 0\\ -1.5 & 2 & -0.5\\ 0.5 & -1 & 0.5\end{bmatrix},\quad z=\begin{bmatrix}1\\ z^{-1}\\ z^{-2}\end{bmatrix} \tag{3.2}$$

โดยแถวของ Vandermonde Matrix ผกผัน $\mathbf{V}^{-1}$ จะแสดงค่าสัมประสิทธิ์ย่อยของตัวกรองอันดับที่ 2 ในโครงสร้างแบบแฟร์โรว์ และสามารถแสดง Transfer function ได้ดังสมการที่ (2.14)

3.1.2 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 2 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง

โครงสร้างแบบแฟร์โรว์ดัดแปลงนี้จะมีประโยชน์ในการลดจำนวนทรัพยากรณ์ที่ใช้ในการออกแบบสังเคราะห์วงจรขึ้นบนฮาร์ดแวร์ โดยการใช้ Transformation Matrix ($\mathbf{T}$) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mathbf{T}-\begin{bmatrix} 1 & \left(\frac{k}{2}\right) & \left(\frac{k}{2}\right)^2 & \left(\frac{k}{2}\right)^3 & \cdots & \left(\frac{k}{2}\right)^k \\ 0 & 1 & \binom{2}{1}\left(\frac{k}{2}\right) & \binom{3}{1}\left(\frac{k}{2}\right)^2 & \cdots & \binom{k}{1}\left(\frac{k}{2}\right)^{k-1} \\ 0 & 0 & 1 & \binom{3}{2}\left(\frac{k}{2}\right) & \cdots & \binom{k}{2}\left(\frac{k}{2}\right)^{k-2} \\ 0 & 0 & 0 & 1 & \cdots & \binom{k}{3}\left(\frac{k}{2}\right)^{k-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

เมื่อ $\binom{a}{b}=\frac{a!}{b!(a-b)!}$ หลังจากการแปลง จะได้เมตริกซ์สัมประสิทธิ์วงจรกรองย่อยใหม่ $\hat{\mathbf{V}}$ ดังนี้

$$\hat{\mathbf{V}} = \mathbf{T}\mathbf{V}^{-1} \tag{3.3}$$

จาก Transformation Matrix จะได้

$$\mathbf{T}=\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

นำสมการที่ (3.2) และสมการที่ (3.4) มาแทนค่าในสมการที่ (3.3) จะได้เมตริกซ์สัมประสิทธิ์วงจรกรองย่อยใหม่ $\hat{\mathbf{V}}$ ดังนี้

$$\hat{\mathbf{V}}=\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ -1.5 & 2 & -0.5 \\ 0.5 & -1 & 0.5 \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 \\ -0.5 & 0 & 0.5 \\ 0.5 & -1 & 0.5 \end{bmatrix} \tag{3.5}$$

โดยแต่ละแถวของเมตริกซ์ $\hat{\mathbf{V}}$ จะแสดงค่าสัมประสิทธิ์ย่อยของตัวกรองอันดับที่ 2 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง จะสังเกตได้ว่าในแต่ละแถว ค่าสัมประสิทธิ์ของวงจรกรองย่อยใหม่นี้จะมีลักษณะสมมาตรกัน หรือมีตัวเลขที่เหมือนกันในแต่ละแถว ทำให้สามารถลดจำนวนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คูณลงได้เกือบ 50 % เลยทีเดียว ซึ่งจะเห็นผลได้ชัดในการออกแบบอันดับตัวกรองที่สูงขึ้น ซึ่งจำนวนสัมประสิทธิ์ย่อยจะมีจำนวนมากจนเกินไป ทำให้เกิดข้อจำกัดในการออกแบบและสังเคราะห์วงจรขึ้นมาใช้บนฮาร์ดแวร์ และสามารถแสดง Transfer function ได้ดังสมการที่ (2.25)

3.1.3 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์

เพื่อประสิทธิภาพในการหน่วงสัญญาณที่ดีขึ้นทั้งในเรื่องของการหน่วงสัญญาณได้แม่นยำขึ้น และลดปัญหาในการหน่วงสัญญาณที่ความถี่สูง จึงจำเป็นต้องออกแบบอันดับตัวกรองที่สูงขึ้นด้วย ดังนั้นจึงได้ทำการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์

จาก Vandermonde Matrix จะได้

$$\mathbf{V}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 \\ 1 & 3 & 9 & 27 & 81 & 243 \\ 1 & 4 & 16 & 64 & 256 & 1024 \\ 1 & 5 & 25 & 125 & 625 & 3125 \end{bmatrix}, \quad z=\begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ z^{-3} \\ z^{-4} \\ z^{-5} \end{bmatrix}$$

สัมประสิทธิ์ย่อยของ $\mathbf{V}(z)$ สามารถหาค่าได้จาก

$$\mathbf{V}(z)=\mathbf{V}^{-1}z$$

จะได้

$$\mathbf{V}^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -2.283 & 5.000 & -5.000 & 3.333 & -1.250 & 0.200 \\ 1.875 & -6.417 & 8.917 & -6.500 & 2.542 & -0.417 \\ -0.708 & 2.958 & -4.917 & 4.083 & -1.708 & 0.292 \\ 0.125 & -0.583 & 1.083 & -1.000 & 0.458 & -0.083 \\ 0.008 & 0.042 & -0.083 & 0.083 & -0.042 & 0.008 \end{bmatrix}, \quad z=\begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ z^{-3} \\ z^{-4} \\ z^{-5} \end{bmatrix} \qquad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถวของ Vandermonde Matrix ผกผัน $\mathbf{V}^{-1}$ จะแสดงค่าสัมประสิทธิ์ย่อยของตัวกรองอันดับที่ 5 ในโครงสร้างแบบแฟร์โรว์ จากสมการแสดงให้เห็นถึงจำนวนของสัมประสิทธิ์ย่อยของวงจรกรองที่จำนวนมากเกินไป ทำให้มีการใช้ทรัพยากรณ์ที่มากจนเกินไป ดังนั้นจึงได้มีการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วยแบบเป็นเศษส่วนชนิดลากรานจ์ในรูปของโครงสร้างแฟร์โรว์ดัดแปลง

3.1.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง

ในการออกแบบวงจรกรองสัญยาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 นั้นจะสังเกตได้ว่ามีการใช้ทรัพยากรณ์ที่มากเกินไปทำให้เกิดผลเสียกับระบบเช่น มีการทำงานที่ไม่เสถียร วงจรที่ออกแบบมีความซ้ำซ้อนมาก และบางครั้งทรัพยากรณ์อาจจะเกินขีดจำกัดของฮาร์ดแวร์ที่ใช้ ทำให้ไม่สามารถทำงานได้ โดยการออกแบบดังกล่าวจะมีลักษณะดังนี้

จาก Transformation Matrix จะได้

$$\mathbf{T}=\begin{bmatrix} 1 & 2.500 & 6.250 & 15.625 & 39.063 & 97.656 \\ 0 & 1 & 5 & 18.750 & 62.500 & 195.313 \\ 0 & 0 & 1 & 7.500 & 37.500 & 156.250 \\ 0 & 0 & 0 & 1 & 10 & 62.500 \\ 0 & 0 & 0 & 0 & 1 & 12.500 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

นำสมการที่ (3.6) และสมการที่ (3.7) มาแทนค่าในสมการที่ (3.3) จะได้เมตริกซ์สัมประสิทธิ์วงจรกรองย่อยใหม่ $\hat{\mathbf{V}}$ ดังนี้

$$\hat{\mathbf{V}}=\begin{bmatrix} 1 & 2.500 & 6.250 & 15.625 & 39.063 & 97.656 \\ 0 & 1 & 5 & 18.750 & 62.500 & 195.313 \\ 0 & 0 & 1 & 7.500 & 37.500 & 156.250 \\ 0 & 0 & 0 & 1 & 10 & 62.500 \\ 0 & 0 & 0 & 0 & 1 & 12.500 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -2.283 & 5.000 & -5.000 & 3.333 & -1.250 & 0.200 \\ 1.875 & -6.417 & 8.917 & -6.500 & 2.542 & -0.417 \\ -0.708 & 2.958 & -4.917 & 4.083 & -1.708 & 0.292 \\ 0.125 & -0.583 & 1.083 & -1.000 & 0.458 & -0.083 \\ 0.008 & 0.042 & -0.083 & 0.083 & -0.042 & 0.008 \end{bmatrix}$$

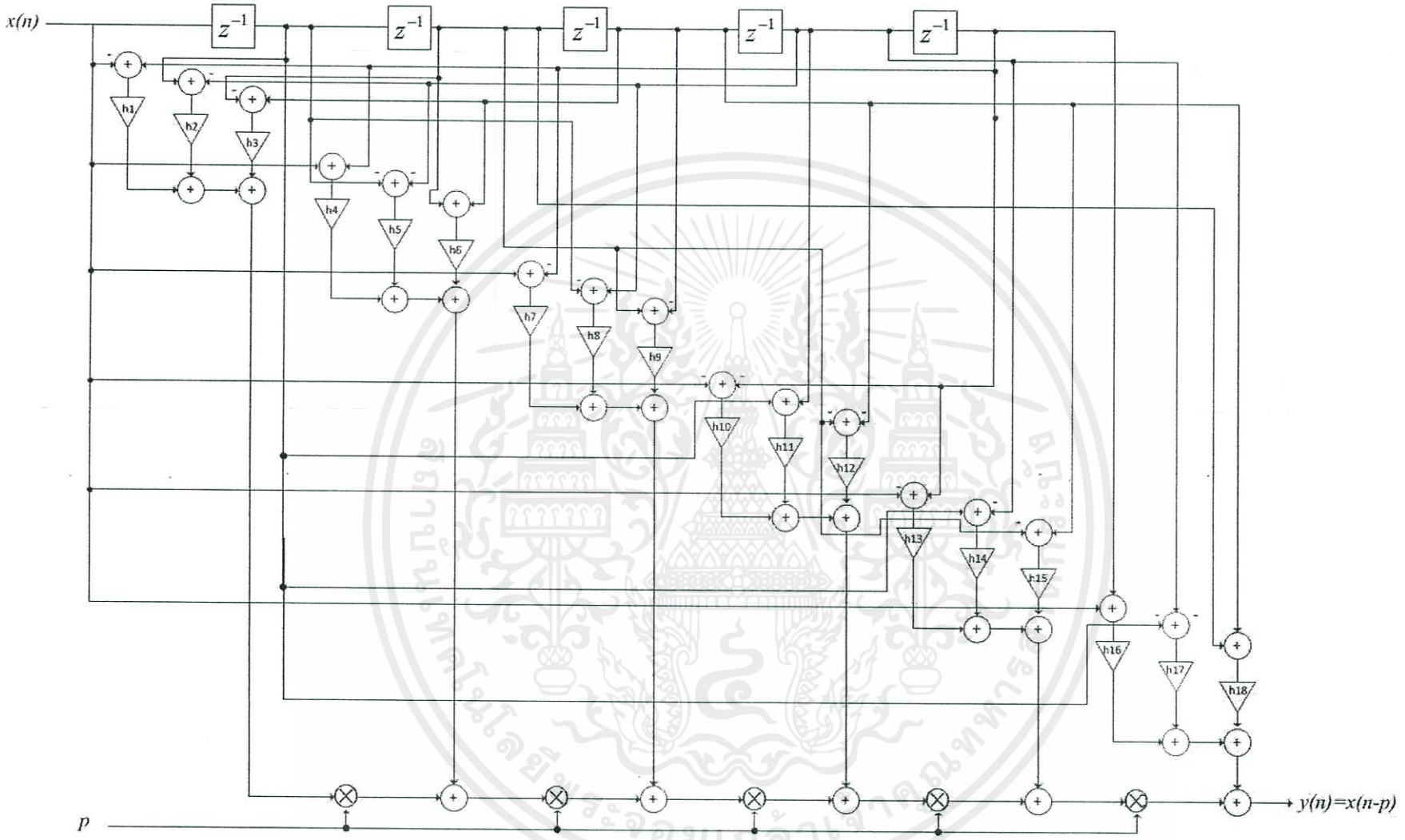
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\hat{\mathbf{V}} = \begin{bmatrix} 0.0117 & -0.0977 & 0.5859 & 0.5859 & -0.0977 & 0.0117 \\ -0.0047 & 0.0651 & -1.1719 & 1.1719 & -0.0651 & 0.0047 \\ -0.0521 & 0.4062 & -0.3542 & -0.3542 & 0.4063 & -0.0521 \\ 0.0208 & -0.2708 & 0.7083 & -0.7083 & 0.2708 & -0.0208 \\ 0.0208 & -0.0625 & 0.0417 & 0.0417 & -0.0625 & 0.0208 \\ -0.0083 & 0.0417 & -0.0833 & 0.0833 & -0.0417 & 0.0083 \end{bmatrix}$$

โดยแต่ละแถวของเมตริกซ์ $\hat{\mathbf{V}}$ จะแสดงค่าสัมประสิทธิ์ย่อยของตัวกรองอันดับที่ 5 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง จะสังเกตได้ว่าในแต่ละแถวของเมตริกซ์ $\hat{\mathbf{V}}$ สัมประสิทธิ์ของวงจรกรองย่อยจะมีความสมมาตรกันทำให้สามารถลดจำนวนตัวคูณได้เกือบ 50 % ทำให้วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์มีประสิทธิภาพ และมีการใช้ทรัพยากรณ์ที่ไม่มากจนเกินไป และสามารถนำมาจัดโครงสร้างแบบแฟร์โรว์ดัดแปลงได้ดังรูปที่ 3.2

การออกแบบที่ได้กล่าวมาทั้งหมดนี้จะนำวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์ดัดแปลงมาทำการทดสอบผลการทำงานทั้งหมด ในรูปของการนำมาประยุกต์ใช้งานในการทำ Oversampling D/A และจะนำผลที่ได้จากการทำงานจริงมาเปรียบเทียบผลจากการจำลองการทำงานว่ามีค่าตรงกันหรือมีค่าใกล้เคียงกันหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างการทำงานของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิด ลากรานจ์โดยใช้อันดับตัวกรองที่ 5 ในโครงสร้างแบบแฟร์โรว์ดัดแปลง

ตารางที่ 3.1 ค่าสัมประสิทธิ์ sub filter ในรูปที่ 3.15

h1	- 0.0083
h2	- 0.0417
h3	- 0.0833
h4	0.0208
h5	- 0.0625
h6	0.0417
h7	- 0.0208
h8	- 0.2708
h9	- 0.7083
h10	- 0.0521
h11	0.4062
h12	- 0.3542
h13	- 0.0047
h14	- 0.0651
h15	- 1.1719
h16	0.0117
h17	- 0.0977
h18	0.5859

ตารางที่ 3.2 ช่วงการหน่วงสัญญาณในโครงสร้างแบบแฟร์โรว์และแฟร์โรว์ดัดแปลง

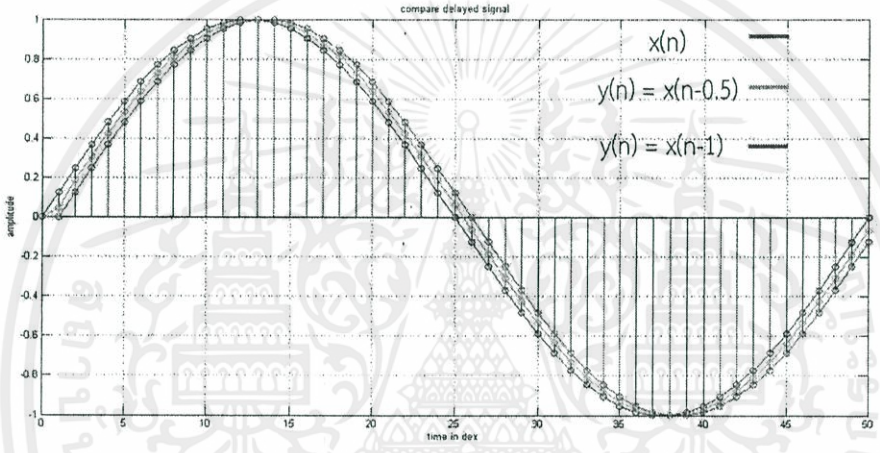
Structure	Relationship between Delay and order of filter
Farrow Structure for Lagrange - type VFD Filter	$D=[\frac{k}{2}-0.5,\frac{k}{2}+0.5]$
Modify Farrow Structure for Lagrange - type VFD Filter	$p=\left[-\frac{k}{2},\frac{k}{2}\right]$

จากตารางที่ 3.2 จะแสดงค่าสัมประสิทธิ์ย่อยภายในวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 2 และตารางที่ 3.3 คือช่วงการปรับค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหน่วงแบบเป็นเศษส่วน โดยที่ค่า D คือค่าความหน่วงที่ใช้ในโครงสร้างแบบแฟร์โรว์และค่า p คือค่าความหน่วงที่ใช้ในโครงสร้าแบบแฟร์โรว์ดัดแปลง

ตัวอย่างการหน่วงสัญญาณแบบเป็นเศษส่วนโดยใช้วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์จะสามารถแสดงได้ดังรูปที่ 3.3 เพื่อให้เห็นถึงความแตกต่างระหว่างการหน่วงสัญญาณ 1 หน่วย กับการหน่วงสัญญาณ 0.5 หน่วย ซึ่งสิ่งที่สังเกตได้ก็คือการหน่วงสัญญาณ 0.5 หน่วยนั้นจะมีค่าแอมพลิจูดอยู่ระหว่างสัญญาณอินพุตกับสัญญาณที่ถูกหน่วงไป 1 หน่วย โดยดูจากจุดแซมเปิ้ลของสัญญาณทั้งสาม



รูปที่ 3.3 แสดงความแตกต่างของการหน่วงสัญญาณ 0.5 หน่วย
กับสัญญาณที่ถูกหน่วงไป 1 หน่วย

3.2 การนำมาประยุกต์ใช้งาน Signal Interpolation

การนำมาประยุกต์ใช้งานเป็น Signal Interpolation นั้น ก่อนอื่นจะต้องทำการหน่วงสัญญาณแบบเป็นเศษส่วนชนิดลากรานจ์เสียก่อน ซึ่งค่าการหน่วงนั้นจะขึ้นอยู่กับว่าเราต้องการ Interpolation สัญญาณกี่เท่า โดยสามารถสรุปความสัมพันธ์ได้ดังตารางที่ 3.2

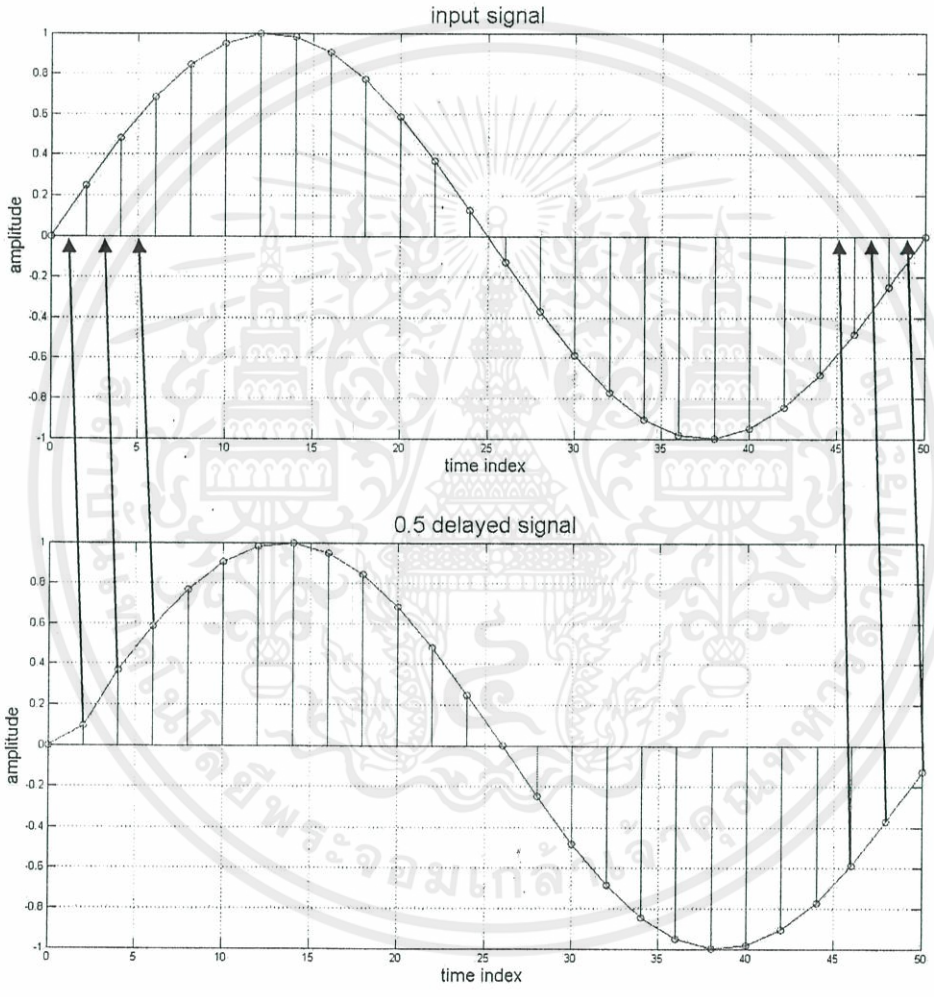
ตารางที่ 3.3 ความสัมพันธ์ระหว่างการทำ Interpolation และ จำนวนค่าการหน่วงที่ใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้เห็นภาพในการทำ Signal Interpolation จะขอแสดงภาพแต่ละขั้นตอนว่ามีกระบวนการทำงานอย่างไรดังนี้

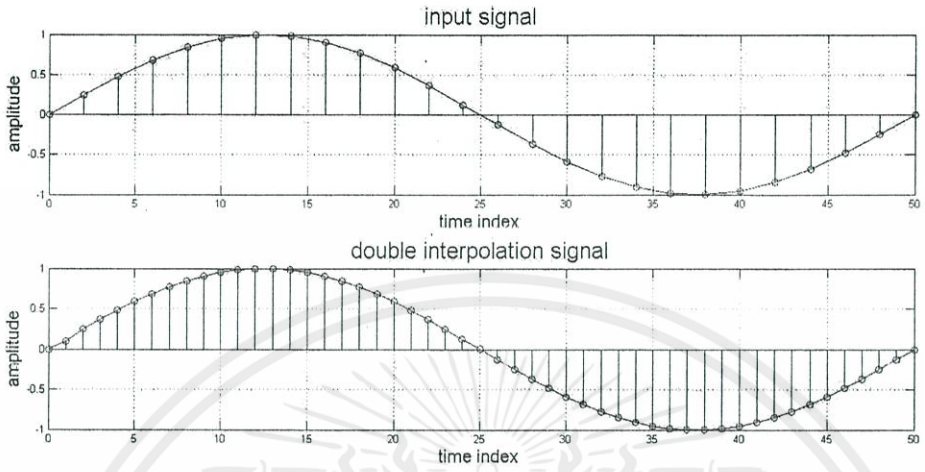
3.2.1 ขั้นตอนการทำ Double Interpolation



รูปที่ 3.4 ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.5 หน่วย พร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 2 เท่า

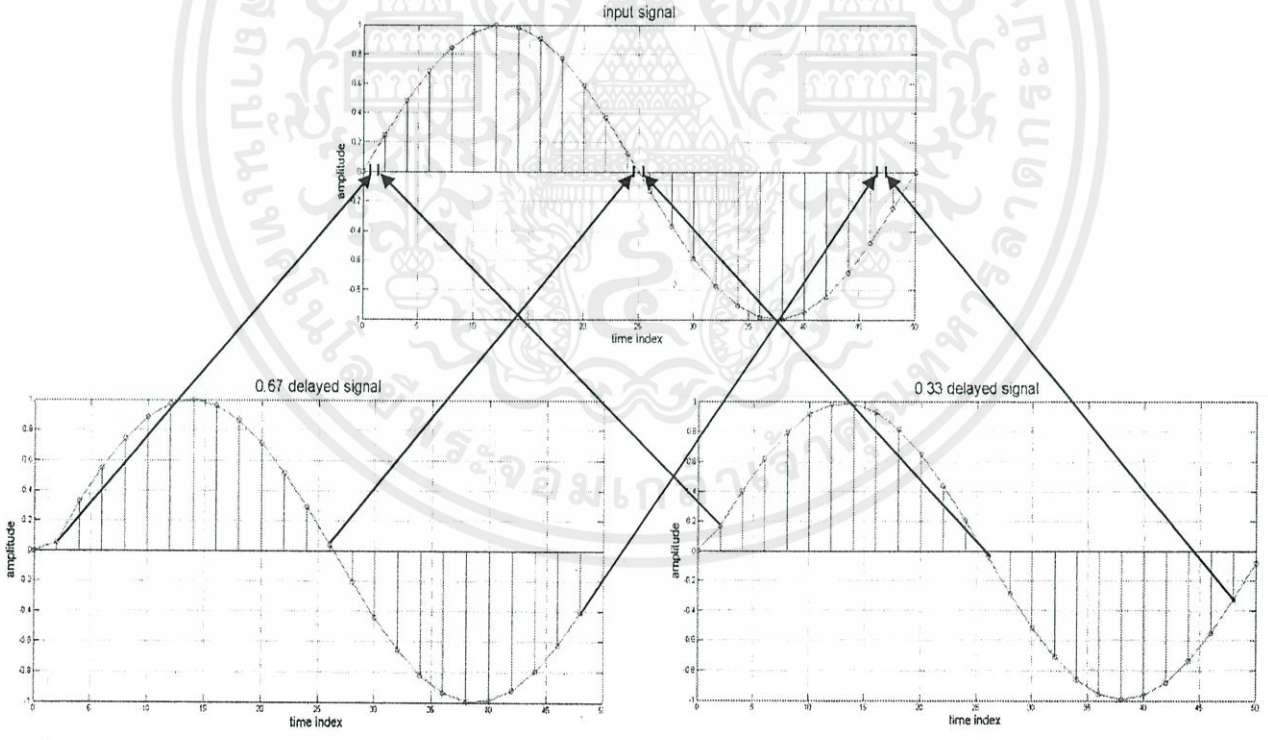
จากรูปที่ 3.4 จะเห็นสัญญาณที่ถูกหน่วงไป 0.5 หน่วย และแสดงการทำ Signal Interpolation 2 เท่า โดยการนำสัญญาณที่ถูกหน่วงไปทำการแทรกกับสัญญาณเดิม ซึ่งจะทำการแทรกสัญญาณไปเรื่อยๆ จนกระทั่งสิ้นสุดสัญญาณสัญญาณที่ได้ก็คือสัญญาณดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ผลของการทำ Double Interpolation เทียบสัญญาณ Input เดิม

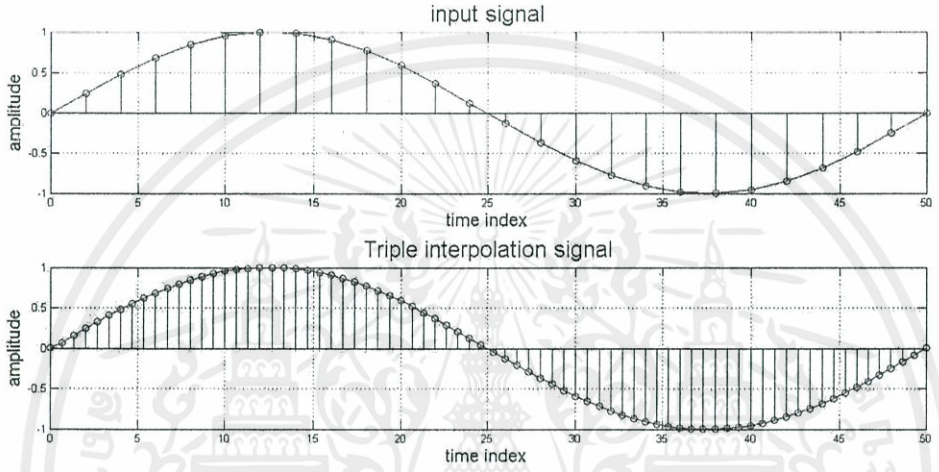
3.2.2 ขั้นตอนการทำ Triple Interpolation



รูปที่ 3.6 ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.67 หน่วย และ 0.33 หน่วย พร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 3 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.6 จะเห็นสัญญาณที่ถูกหน่วงไป 0.33 หน่วย และ 0.67 หน่วย พร้อมทั้งแสดงการทำ Signal Interpolation 3 เท่า โดยการนำสัญญาณที่ถูกหน่วงไปทำการแทรกกับสัญญาณเดิม ซึ่งจะทำการแทรกสัญญาณไปเรื่อยๆ จนกระทั่งสิ้นสุดสัญญาณ สัญญาณที่ได้จะสามารถแสดงได้ดังรูปที่ 3.7

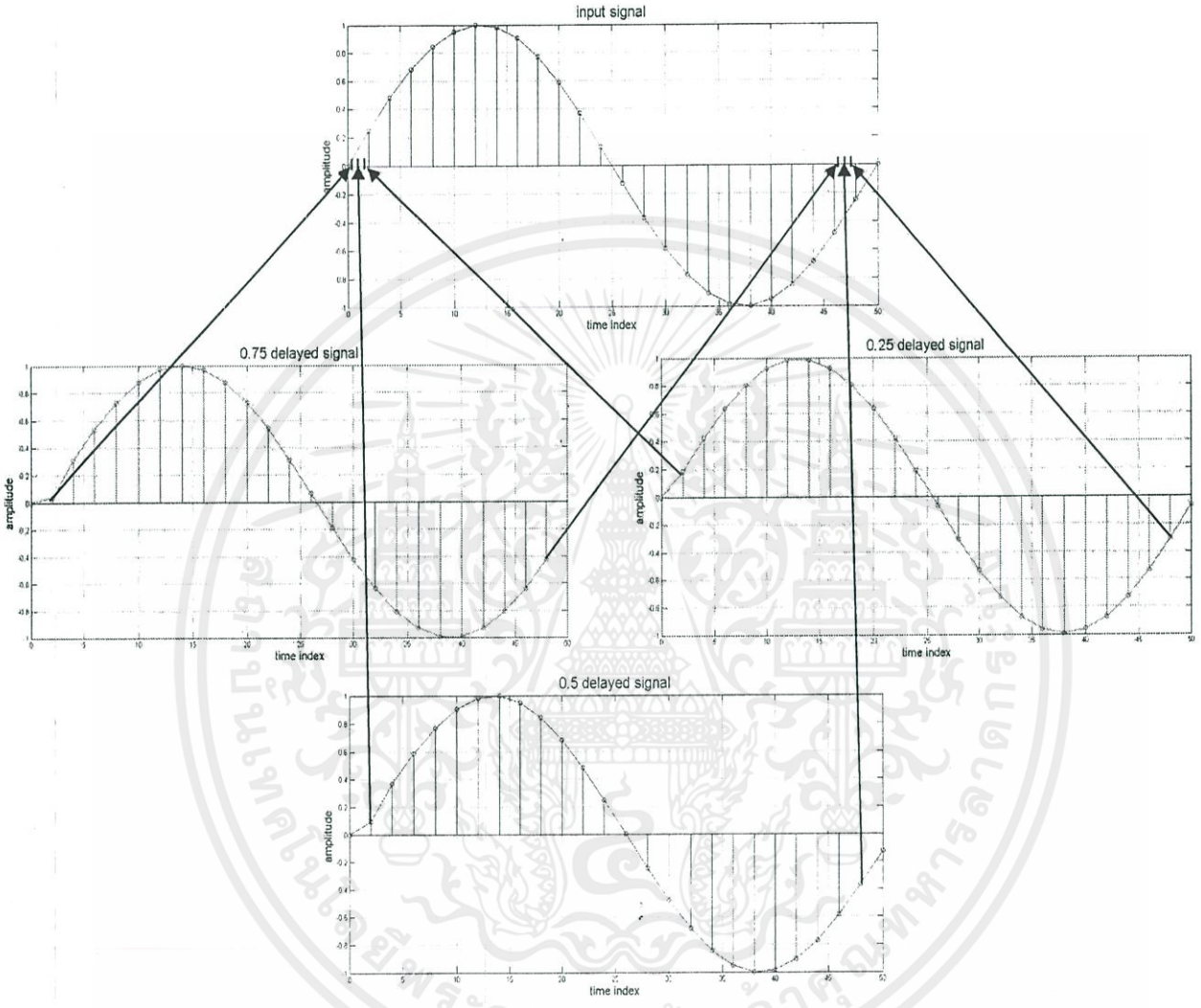


รูปที่ 3.7 ผลของการทำ Triple Interpolation เทียบสัญญาณ Input เดิม

ถ้าสัญญาณที่มีความละเอียดต่ำมากๆ ก็จะต้องเพิ่มจำนวนเท่าในการทำ Signal Interpolation ด้วย แต่อาจจะมีข้อจำกัดในการออกแบบและสังเคราะห์วงจรเนื่องจากต้องใช้จำนวนทรัพยากรณ์มาก ดังนั้นวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์จะสามารถนำมาประยุกต์ใช้งานเกี่ยวกับ Signal Interpolation ได้ด้วยจำนวนเท่าที่จำกัด ขึ้นอยู่กับรุ่นของอุปกรณ์ที่นำมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

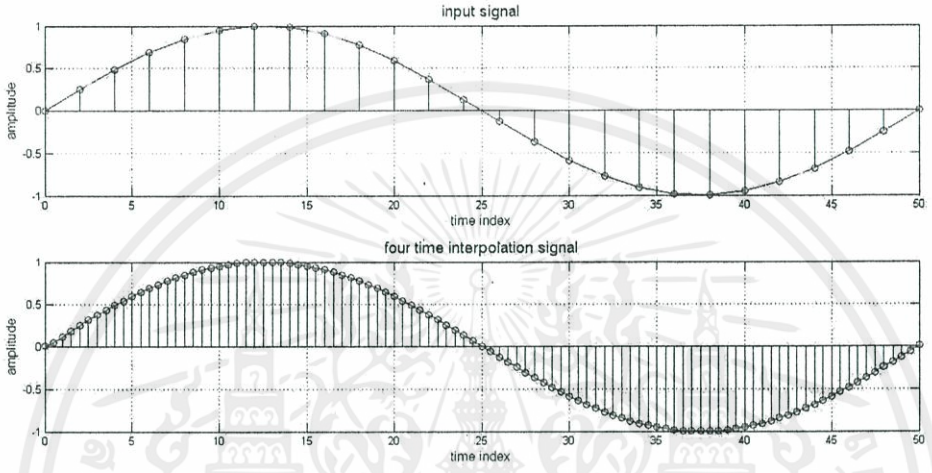
3.2.3 ขั้นตอนการทำ Four time Interpolation



รูปที่ 3.8 ผลการเปรียบเทียบระหว่างสัญญาณอินพุต กับสัญญาณที่ถูกหน่วง 0.75 หน่วย 0.5 หน่วย และ 0.25 หน่วย พร้อมทั้งแสดงตัวอย่างการทำ Signal Interpolation 4 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 จะเห็นสัญญาณที่ถูกหน่วงไป 0.25 หน่วย 0.5 หน่วย และ 0.75 หน่วย พร้อมทั้งแสดงการทำ Signal Interpolation 4 เท่า โดยการนำสัญญาณที่ถูกหน่วงไปทำการแทรกกับสัญญาณเดิม ซึ่งจะทำการแทรกสัญญาณไปเรื่อยๆ จนกระทั่งสิ้นสุดสัญญาณ สัญญาณที่ได้จะสามารถแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 ผลของการทำ Four time Interpolation เทียบสัญญาณ Input เดิม

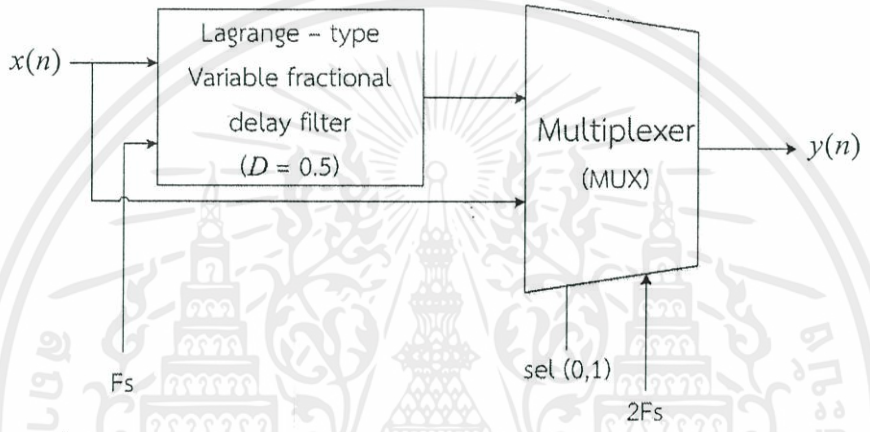
จะเห็นได้ว่าสัญญาณที่ทำ Signal Interpolation จะมีความละเอียดมากยิ่งขึ้น เนื่องจากนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์มาใช้ในการประมาณค่าในช่วง และนำการประมาณค่าในช่วงนั้นมาแทรกในสัญญาณเดิม อาจจะมีเฉพาะช่วงค่าแซมเปิ้ลแรกๆ เท่านั้นที่อาจจะมีประมาณค่าได้ไม่ตรงจึงทำให้ค่าแอมพลิจูดในช่วงนั้นเกิดความผิดเพี้ยนขึ้น

3.3 การทำ Oversampling D/A

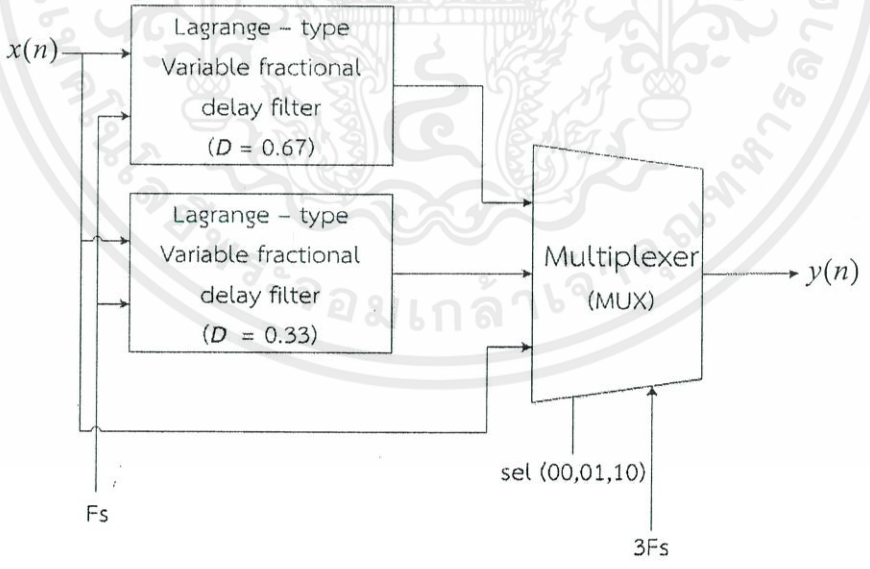
ภายใน FPGA นั้นจะทำการออกแบบและสังเคราะห์วงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ดังรูปที่ 3.1 เพื่อนำวงจรกรองนี้มาประยุกต์ใช้งานในการทำ Oversampling D/A สัญญาณ 2 เท่า, 3 เท่า และ 4 เท่า เพื่อให้สัญญาณที่มีความละเอียดเพิ่มขึ้น โดยจะการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ค่า D คือค่าความหน่วงสัญญาณที่ถูกใช้ในโครงสร้างแบบแฟร์โรว์ และค่า p คือค่าความหน่วงสัญญาณใหม่ที่ถูกใช้ในโครงสร้างแบบแฟร์โรว์ดัดแปลง ซึ่งในแต่ละอันดับของตัวกรอง (k) จะมีความสัมพันธ์กันอยู่ ซึ่งจะจำกัดค่าความหน่วงไว้ที่อันดับตัวกรองต่างๆ เมื่อทำการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์แล้ว จากนั้นจะทำการออกแบบวงจรเพื่อทำ Oversampling D/A 2 เท่า 3 เท่า และ 4 เท่า ดังนี้

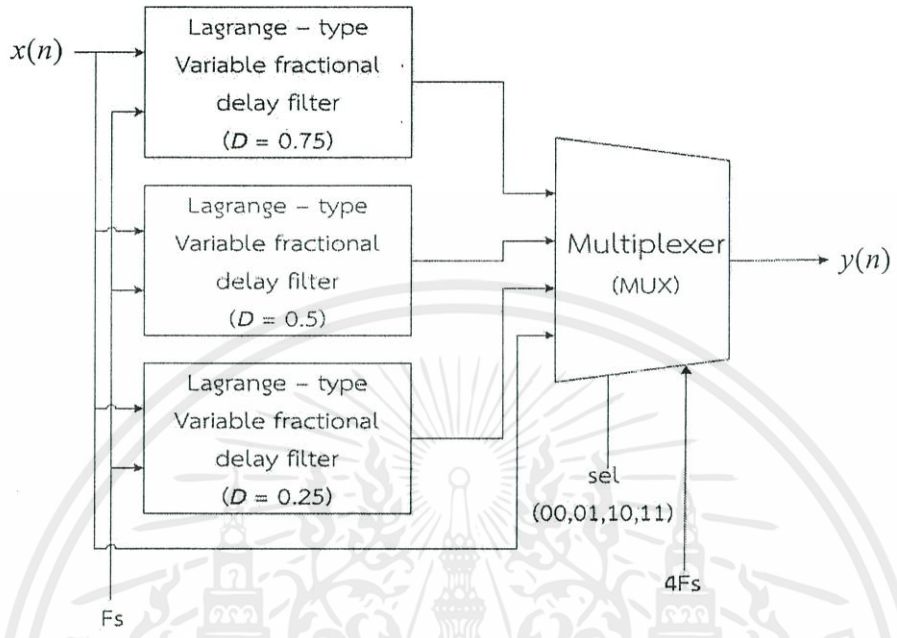


รูปที่ 3.10 Block diagram การทำงานของการทำ Oversampling D/A 2 เท่า



รูปที่ 3.11 Block diagram การทำงานของการทำ Oversampling D/A 3 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 Block diagram การทำงานของการทำ Oversampling D/A 4 เท่า

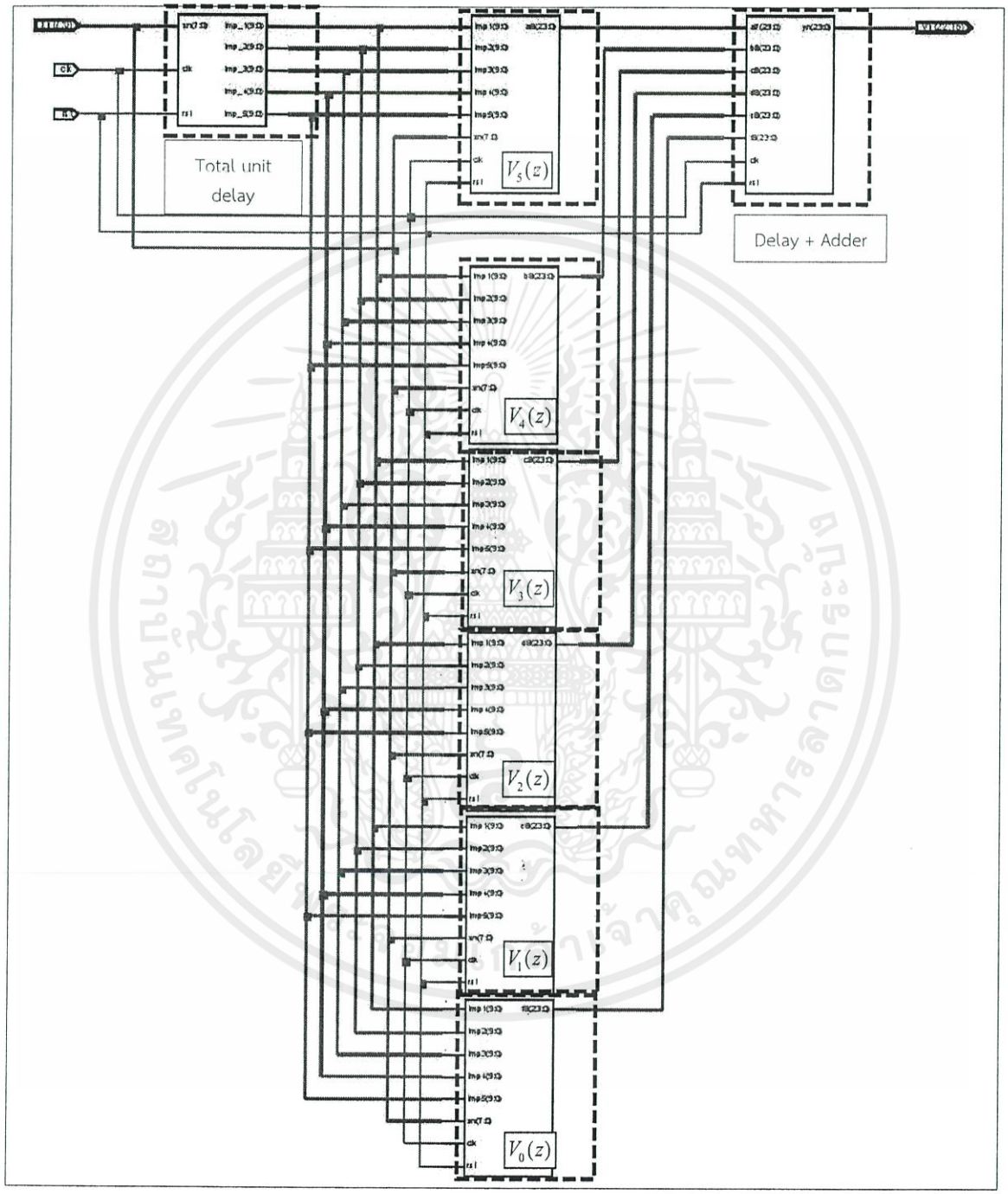
จากรูปที่ 3.11 – 3.13 จะแสดงการนำวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์มาประยุกต์ใช้งานในการทำ Oversampling D/A 2 เท่า 3 เท่า และ 4 เท่า ตามลำดับ โดยการใช้วงจร Multiplexer (Mux) ในการรวมสัญญาณ สัญญาณที่ได้นั้นจะมีลักษณะที่มีความละเอียดมากยิ่งขึ้นเมื่อเทียบกับสัญญาณอินพุตเดิม

3.4 การออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ดัดแปลงบน Hardware

การออกแบบ และการสังเคราะห์วงจรขึ้นมาใช้นั้น จะใช้ภาษา VHDL ในการออกแบบ และสังเคราะห์วงจร โดยใช้โปรแกรม Xilinx ซึ่งเป็นภาษาที่ถูกใช้เพื่อทำการออกแบบและสังเคราะห์วงจรทางดิจิตอล จากนั้นจึงทำการ Implement ลงบนอุปกรณ์ FPGA เพื่อให้สามารถนำวงจรที่ทำการออกแบบไว้นำมาใช้งานได้จริง ซึ่งในที่นี้เราจะทำการรออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ดัดแปลงโดยใช้อันดับตัวกรองที่ 5

เหตุผลที่ทำการออกแบบวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ดัดแปลงโดยใช้อันดับตัวกรองที่ 5 นั้น คือ เพื่อให้วงจรกรองนี้มีประสิทธิภาพในการหน่วงสัญญาณที่ดีขึ้น และสามารถหน่วงสัญญาณได้ที่ความถี่ที่สูงขึ้น เมื่อเทียบกับการใช้อันดับตัวกรองที่น้อยกว่า อีกทั้งในโครงสร้างแบบแฟร์โรว์ดัดแปลงยังช่วยลดจำนวนทรัพยากรณ์ที่ใช้ไปได้เกือบถึง 50 % เมื่อเทียบกับโครงสร้างแบบแฟร์โรว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



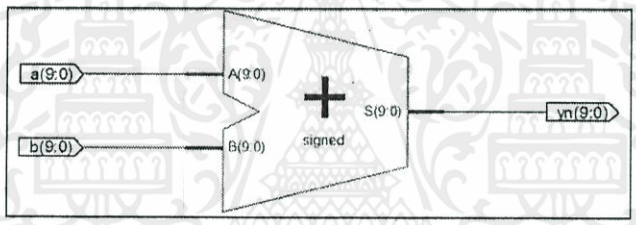
รูปที่ 3.13 การออกแบบของ VFD Filter (order5)
ในโครงสร้างแบบแฟร์โรว์ดัดแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.13 แสดงให้เห็นการออกแบบ และการสังเคราะห์วงจรทางฮาร์ดแวร์ของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ ดัดแปลงโดยใช้อันดับตัวกรองที่ 5 ซึ่ง Total unit delay คือวงจรการหน่วงสัญญาณ 1 หน่วยที่ต้องใช้ในวงจร มีตัวกัน 2 ตัว ค่า **V** คือค่าสัมประสิทธิ์ย่อยของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ และ Dealy + Adder คือวงจรที่ป้อนความหน่วงสัญญาณแบบเป็นเศษส่วน และวงจรบวก หลังจากการออกแบบ และสังเคราะห์วงจรแล้วนั้น จะทำการ Implement ลงบน FPGA ต่อไป เพื่อแสดงผลของการออกแบบว่าผลที่ได้ถูกต้องหรือไม่ โดยทำการเปรียบเทียบกับผลของการจำลองการทำงานในโปรแกรม MATLAB

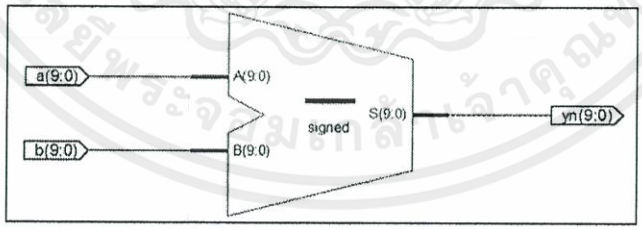
3.4.1 ส่วนประกอบของวงจรย่อย

วงจรบวก ประกอบไปด้วย ข้อมูลอินพุต 2 ขา และเอาต์พุตข้อมูล 1 ขา ทำหน้าที่คำนวณผลรวมของค่าข้อมูลอินพุต 2 จำนวน ซึ่งเป็นการบวกแบบคิดเครื่องหมาย



รูปที่ 3.14 ผลของการสังเคราะห์วงจรบวก

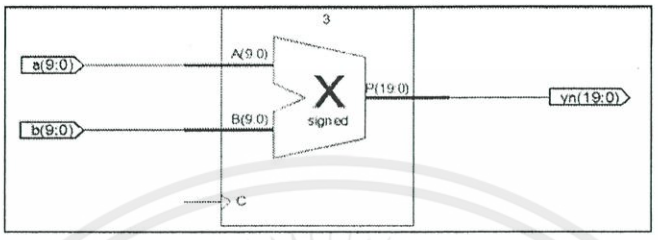
วงจรลบ ประกอบไปด้วย ข้อมูลอินพุต 2 ขา และเอาท์พุตข้อมูล 1 ขา ทำหน้าที่คำนวณผลต่างของค่าข้อมูลอินพุต 2 จำนวน ซึ่งเป็นการลบแบบคิดเครื่องหมาย



รูปที่ 3.15 ผลของการสังเคราะห์วงจรลบ

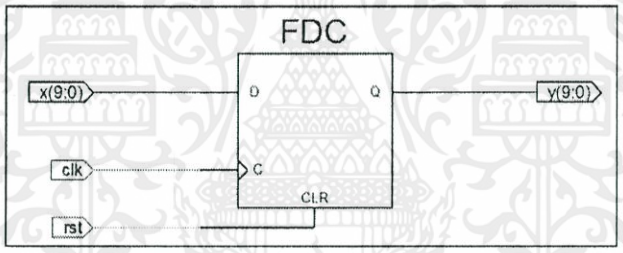
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรคูณ ประกอบไปด้วย ข้อมูลอินพุต 2 ขา และเอาท์พุตข้อมูล 1 ขา ทำหน้าที่คำนวณผลคูณของค่าข้อมูลอินพุต 2 จำนวน ซึ่งวงจรคูณแบบคิดเครื่องหมาย



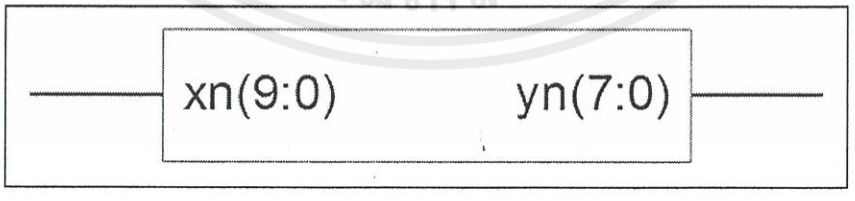
รูปที่ 3.16 ผลของการสังเคราะห์วงจรคูณ

วงจรหน่วงเวลา ประกอบไปด้วยข้อมูลอินพุต 3 ขาคือ ขาข้อมูลอินพุต x(n), ขา clk, ขา rst และเอาท์พุตข้อมูล 1 ขาคือ y(n) ทำหน้าที่หน่วงเวลาไปหนึ่งหน่วยเวลา



รูปที่ 3.17 ผลของการสังเคราะห์วงจรหน่วงเวลา

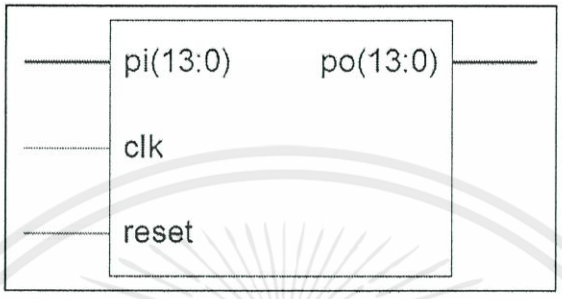
วงจรตัดบิต เป็นวงจรที่ใช้ตัดบิตให้มีขนาดและตำแหน่งตรงที่ต้องการ ตามรูปที่ 3.18 ซึ่งเป็นการตัดบิตจากบิตอินพุต 10 บิต ให้เหลือ 8 บิต



รูปที่ 3.18 ผลของการสังเคราะห์วงจรตัดบิต

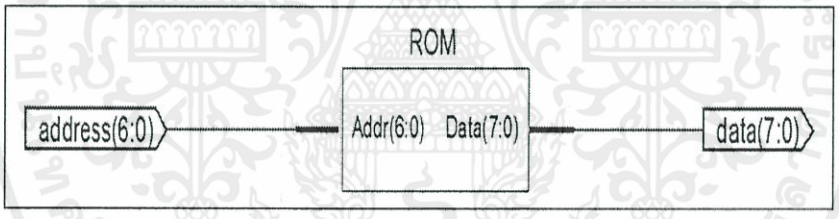
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร PIPO (Parallel In – Parallel Out) เป็นโมดูลของวงจรที่เสมือนนำ D Flip Flop มาต่อกัน และทำหน้าที่ Latch ค่าข้อมูลต่างๆ



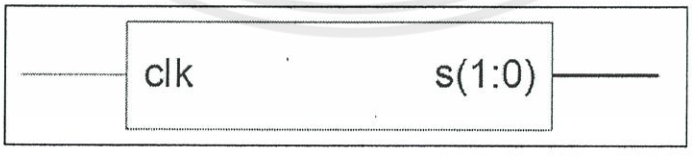
รูปที่ 3.19 ผลของการสังเคราะห์วงจร PIPO

วงจร look up table เป็นวงจรที่ทำหน้าที่เก็บค่าแอมพลิจูดของสัญญาณที่ตำแหน่งแอดเดรสต่างๆ



รูปที่ 3.20 ผลของการสังเคราะห์วงจร look up table

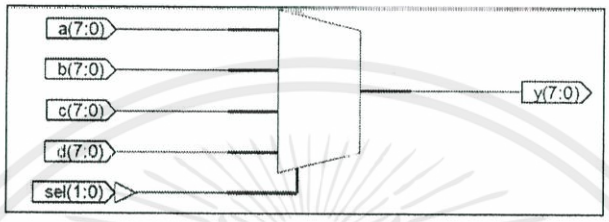
วงจร Counter เป็นวงจรที่ทำหน้าที่ นับวนจนซ้ำถึงค่าที่กำหนด โดยมี ขา clk เป็นขาที่ใช้ในการเข้าจังหวะการนับ



รูปที่ 3.21 ผลของการสังเคราะห์วงจร Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Multiplexer เป็นวงจรที่ทำหน้าที่รวมสัญญาณข้อมูลเข้าด้วยกัน ในที่นี้คือการนำสัญญาณที่ทำการหน่วงไว้มาทำการแทรกกับสัญญาณเดิมเพื่อให้ได้สัญญาณที่มีความละเอียดมากยิ่งขึ้น โดยมีขา s เป็นขา select ซึ่งจะเลือกว่าจะเอาสัญญาณข้อมูลที่ขาใดเข้าก่อน ขาใดเข้าตามมาเป็นต้น



รูปที่ 3.22 ผลของการสังเคราะห์วงจร Multiplexer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

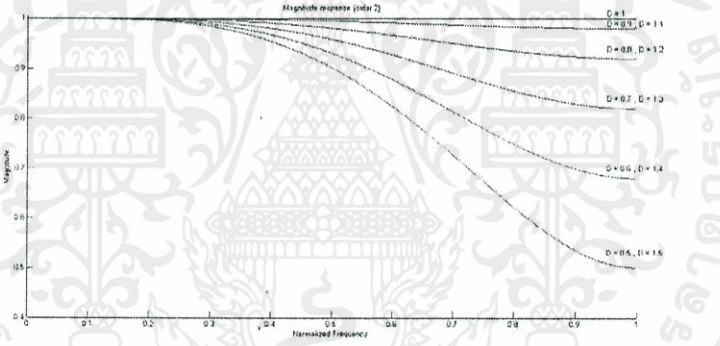
บทที่ 4

ผลการทดลอง

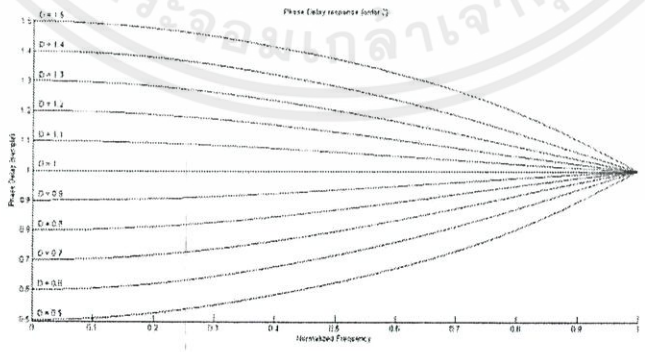
ผลการทดลองทั้งหมดนั้นจะแบ่งออกเป็น software simulation (MATLAB), Hardware simulation (ภาษา VHDL) และ ผลการทำงานจริงที่ได้จากเครื่อง Oscilloscope ซึ่งจะแบ่งผลการทดลองออกเป็นหัวข้อต่างๆ ดังนี้

4.1 คุณสมบัติของตัววงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์

4.1.1 Software simulation (MATLAB) ของตัวกรองอันดับที่ 2



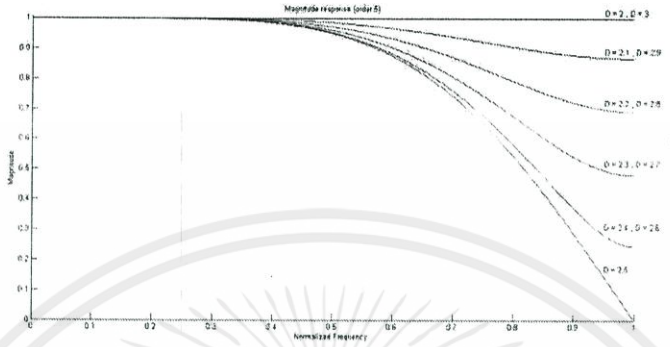
รูปที่ 4.1 Magnitude response ของตัวกรองอันดับที่ 2 ที่ได้จาก MATLAB



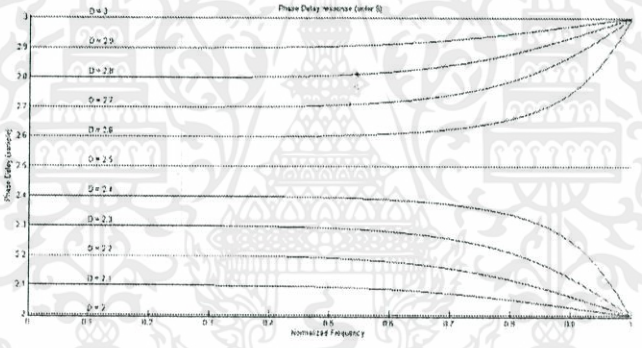
รูปที่ 4.2 Phase delay response ของตัวกรองอันดับที่ 2 ที่ได้จาก MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 Software simulation (MATLAB) ของฟิลเตอร์อันดับที่ 5



รูปที่ 4.3 Magnitude response ของตัวกรองอันดับที่ 5 ที่ได้จาก MATLAB

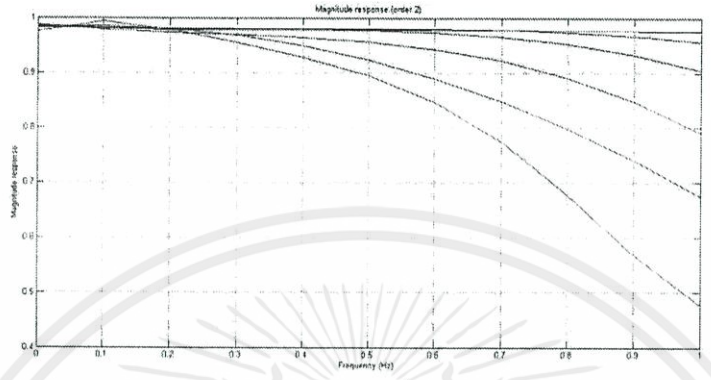


รูปที่ 4.4 Phase delay response ของตัวกรองอันดับที่ 5 ที่ได้จาก MATLAB

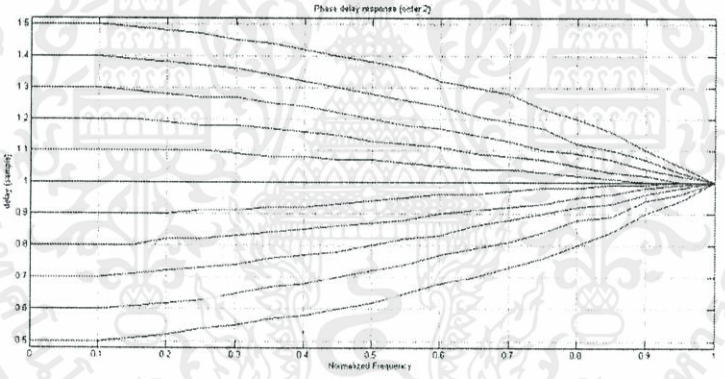
จากกราฟแสดงคุณสมบัติของตัวกรองจะสังเกตเห็นว่า ตัวกรองอันดับที่ 5 สามารถทำการหน่วงสัญญาณแบบเป็นเศษส่วนได้จริงในย่านความถี่ที่สูงกว่าตัวกรองอันดับที่ 2 สังเกตได้จากค่า Magnitude response และ Phase delay response ไม่เปลี่ยนแปลงในช่วงความถี่ ซึ่งถ้าสังเกตตัวกรองอันดับที่ 2 จะเห็นว่าไม่สามารถทำการหน่วงสัญญาณแบบเป็นเศษส่วนได้ในย่านความถี่สูงๆ เนื่องจากค่าการหน่วงสัญญาณเกิดความผิดเพี้ยน และ Magnitude response ถูกลดทอนลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 Hardware simulation (ภาษา VHDL) ของตัวกรองอันดับที่ 2



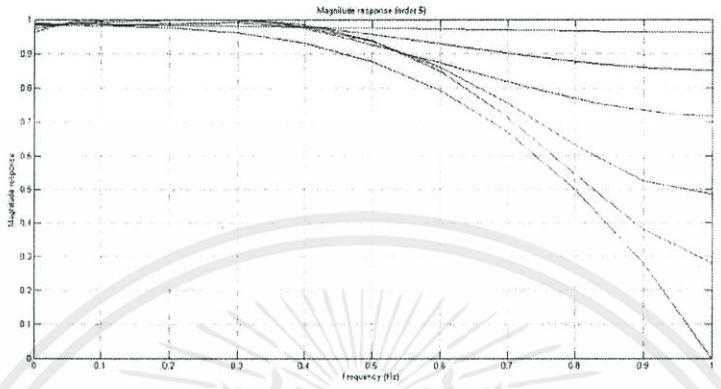
รูปที่ 4.5 Magnitude response ของตัวกรองอันดับที่ 2 ที่ได้จาก ภาษา VHDL



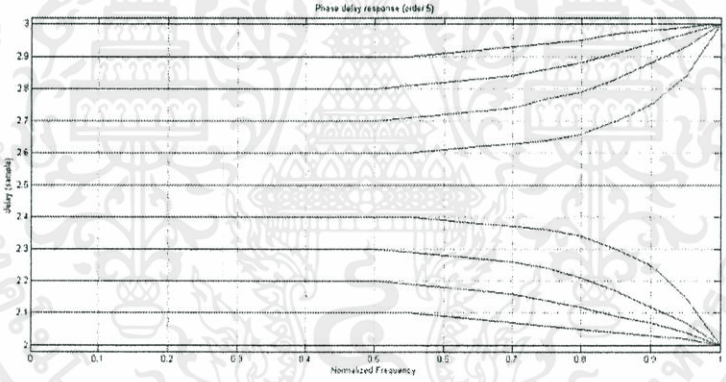
รูปที่ 4.6 Phase delay response ของตัวกรองอันดับที่ 2 ที่ได้จาก ภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 Hardware simulation (ภาษา VHDL) ของตัวกรองอันดับที่ 5



รูปที่ 4.7 Magnitude response ของตัวกรองอันดับที่ 5 ที่ได้จาก ภาษา VHDL



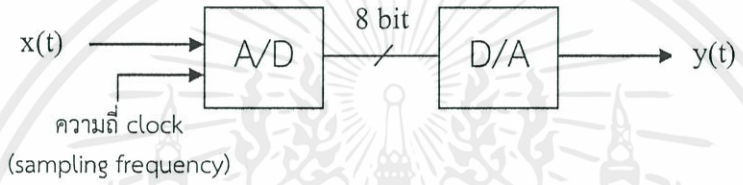
รูปที่ 4.8 Phase delay response ของตัวกรองอันดับที่ 5 ที่ได้จาก ภาษา VHDL

ผลของคุณสมบัติที่ได้จากการออกแบบจริงทางฮาร์ดแวร์ของวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์มีลักษณะใกล้เคียงกับผลการจำลองจากโปรแกรม MATLAB ซึ่งผลที่ได้อาจจะมีความคลาดเคลื่อนบ้าง เนื่องจากจำนวนบิตที่ใช้กับวงจรนั้นมีค่า 8 บิต ซึ่งอาจจะได้ผลลัพธ์ออกมาไม่ละเอียดเท่าที่ควร

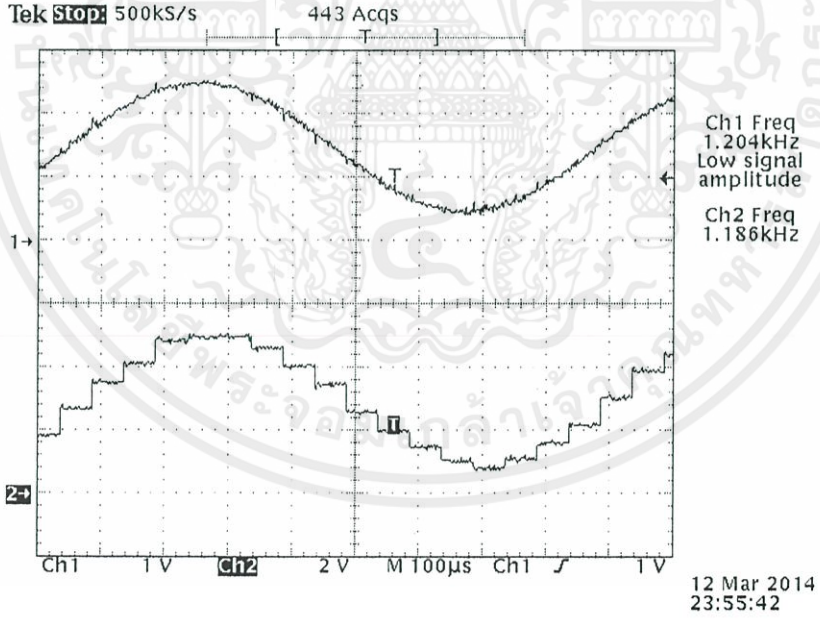
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำ Oversampling D/A

Oversampling D/A คือการนำสัญญาณที่มีความละเอียดน้อยไปทำการประมวลผลสัญญาณทางดิจิตอลเพื่อให้ได้เอาท์พุตออกมามีความละเอียดสูงขึ้น หรือการทวีคูณ sampling frequency ซึ่งวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแบบแฟร์โรว์ และโครงสร้างแบบแฟร์โรว์ดัดแปลงสามารถนำมาประยุกต์ใช้งานทางด้านนี้ได้ เพื่อให้เห็นสาเหตุ ปัญหาที่เกิดขึ้น และเหตุผลที่ต้องทำ Oversampling D/A สามารถแสดงการทำงานได้ดังรูปที่ 4.9 โดยจะใช้ sampling frequency เท่ากับ 20 kHz

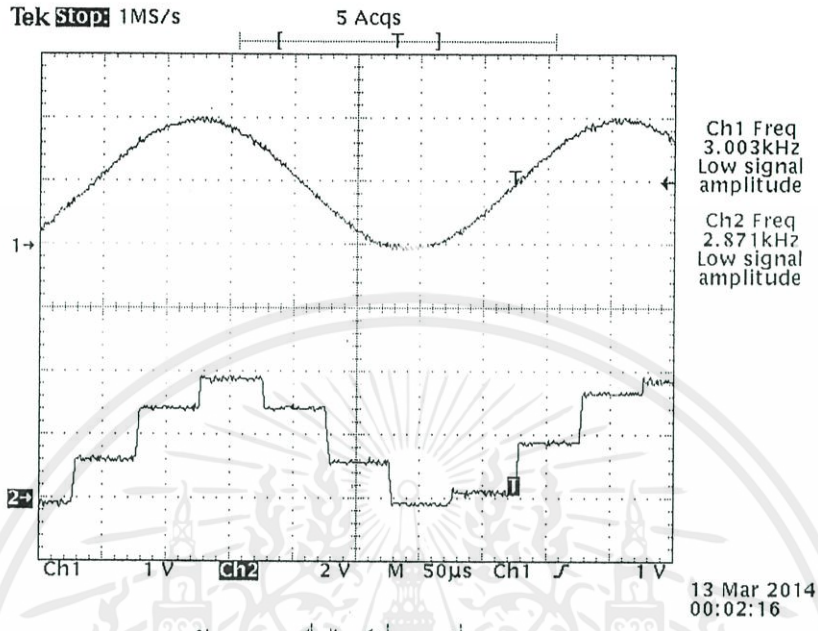


รูปที่ 4.9 Block diagram แสดงปัญหาของการทดลองที่เกิดขึ้น

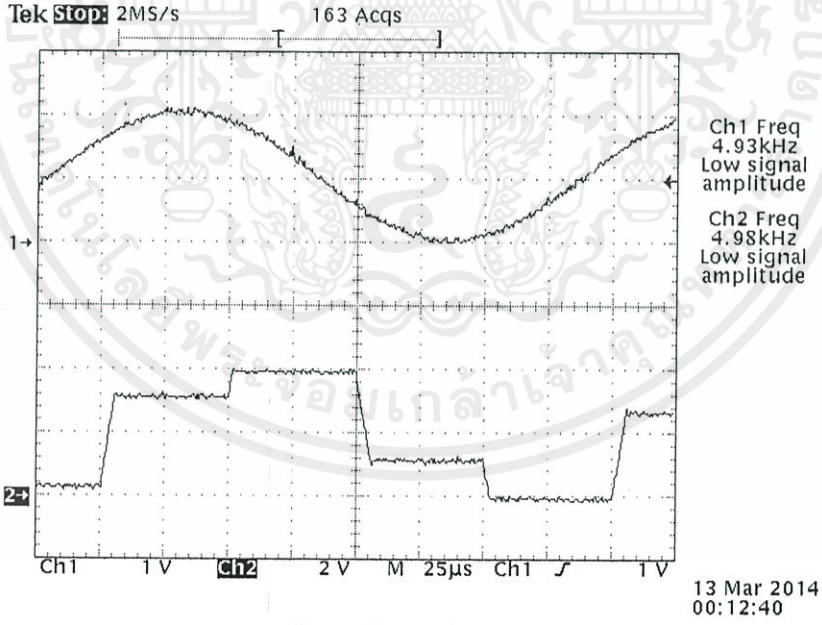


รูปที่ 4.10 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz ตามรูป Block diagram ที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



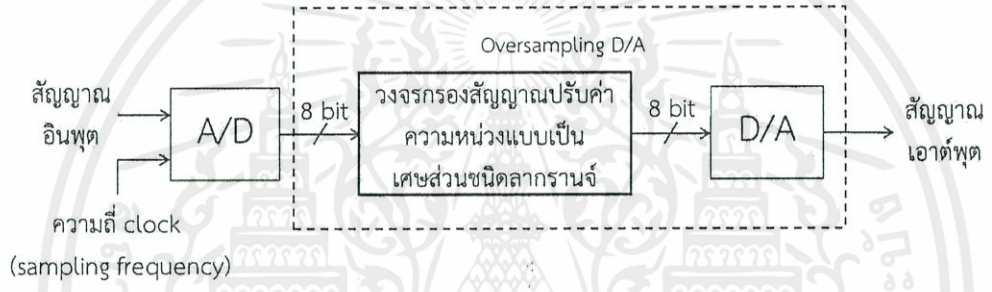
รูปที่ 4.11 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz ตามรูป Block diagram ที่ 4.9



รูปที่ 4.12 ผลการทดลองสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz ตามรูป Block diagram ที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองนี้ความถี่ sampling rate หรือ sampling frequency จะถูกกำหนดค่าไว้ตายตัวคือ 20 kHz และทำการปรับค่าความถี่ที่ 1.2 kHz 3 kHz และ 4.8 kHz เพื่อสังเกตผลที่เกิดขึ้นตามรูป block diagram ที่ 4.9 ซึ่งตามกฎของ Nyquist คือความถี่ sampling rate จะต้องมากกว่าหรือเท่ากับ 2 เท่าของความถี่สัญญาณอินพุต แต่ในความเป็นจริงแล้วจากผลการทดลองจะเห็นว่าเมื่อปรับความถี่ไปที่ 1200 Hz สัญญาณเอาต์พุตที่ออกมามีลักษณะของสัญญาณไซน์ที่เป็นแบบขั้นบันได ดังนั้นเราจึงสามารถนำวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแฟร์โรว์ และแฟร์โรว์ดัดแปลงมาประยุกต์ใช้งานในการแก้ไขปัญหาดังกล่าวได้ ซึ่งก็คือการทำ Oversampling D/A



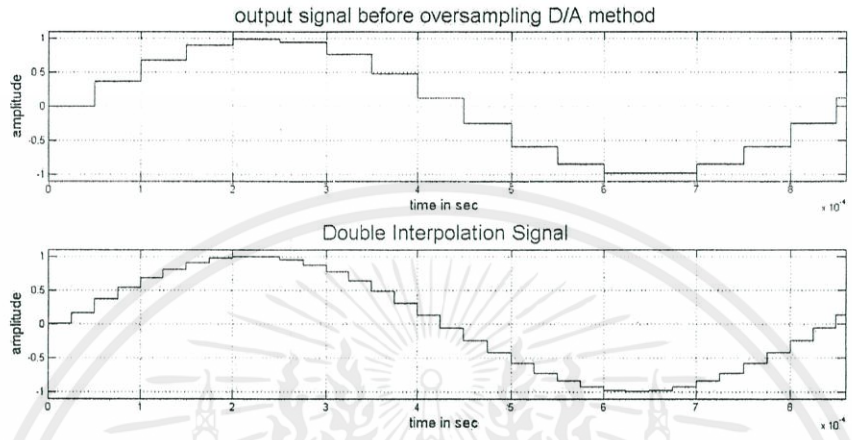
รูปที่ 4.13 Block diagram แสดงการทำงานของ Oversampling D/A

4.3 ผลการจำลองการทำงานทางซอฟแวร์ในการทำ Oversampling D/A

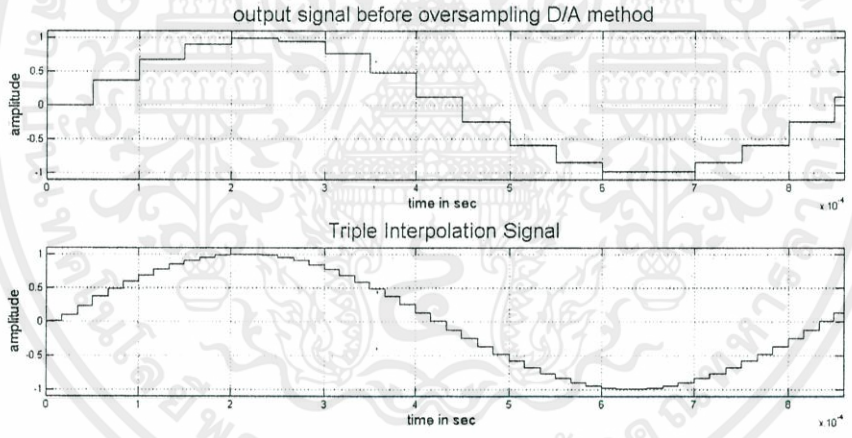
การจำลองการทำงานทางซอฟแวร์จะใช้โปรแกรม MATLAB ในการจำลองการทำงานโดยเริ่มจากการสร้างสัญญาณคลื่นไซน์เป็นสัญญาณอินพุตที่ความถี่ 1.2 kHz, 3 kHz, 4.8 kHz และ sampling frequency เท่ากับ 20 kHz เพื่อให้เห็นถึงผลของการจำลองการทำงานว่าสัญญาณนั้นมีความละเอียดขึ้นจริงหรือไม่ โดยจะใช้วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 5 และจะทำการ Oversampling D/A 2, 3 และ 4 เท่าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz

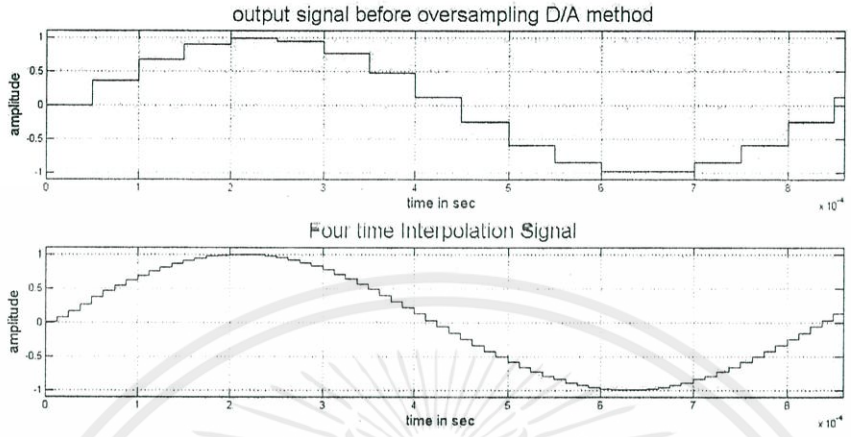


รูปที่ 4.14 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB



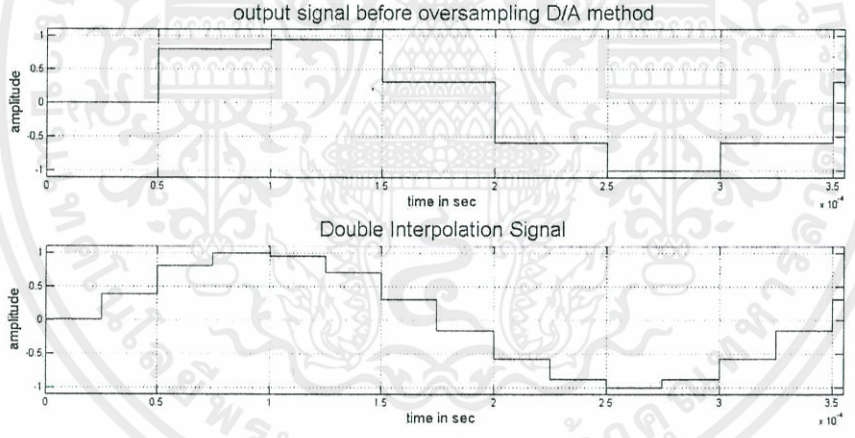
รูปที่ 4.15 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



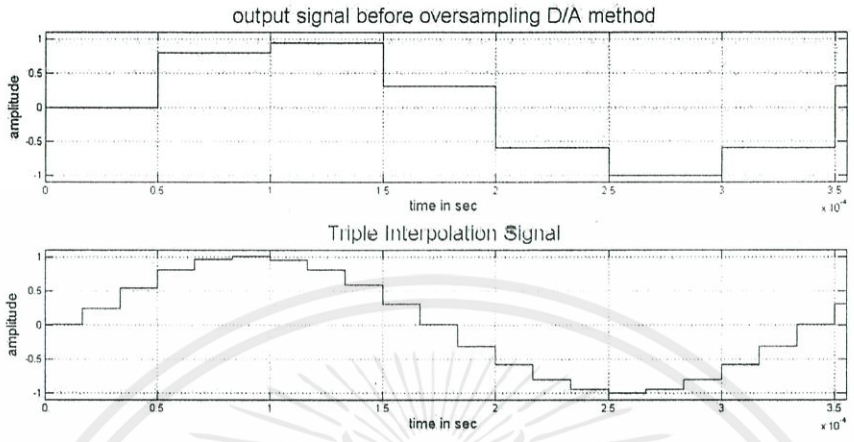
รูปที่ 4.16 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 1.2 kHz ด้วยโปรแกรม MATLAB

4.3.2 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz

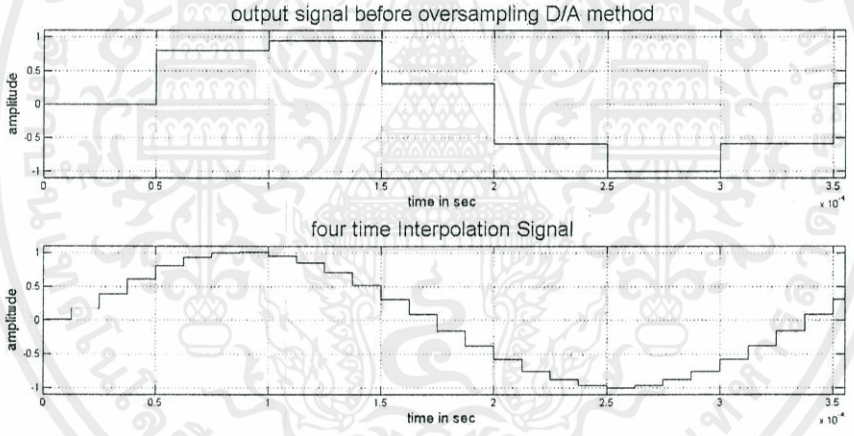


รูปที่ 4.17 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 3 kHz
ด้วยโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



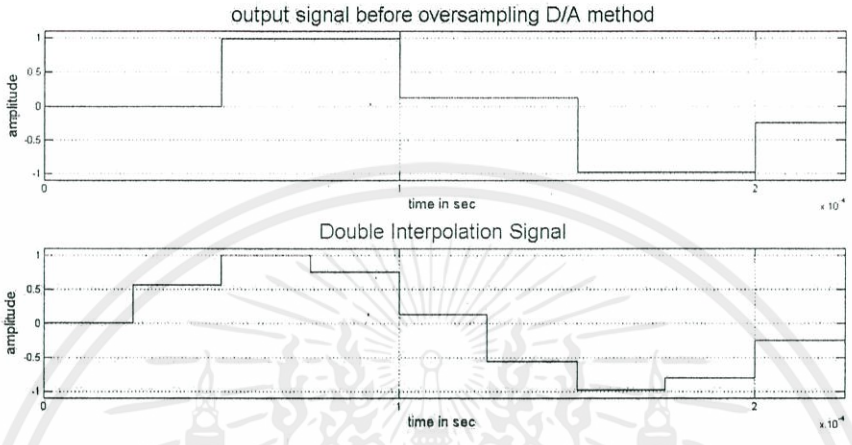
รูปที่ 4.18 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 3 kHz ด้วยโปรแกรม MATLAB



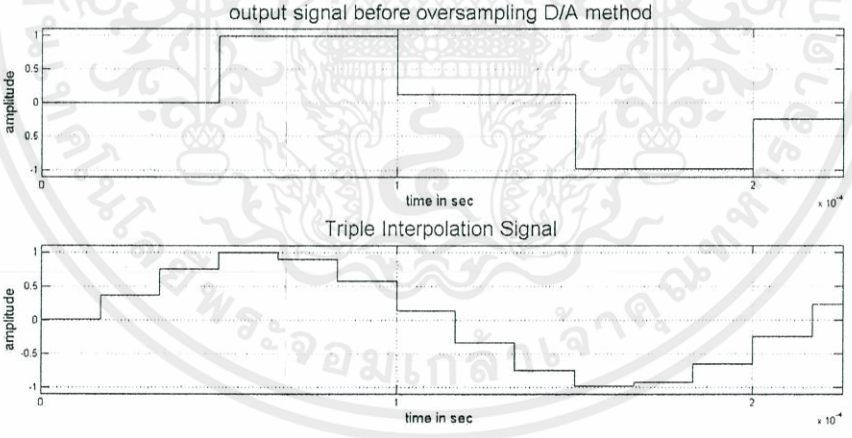
รูปที่ 4.19 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 3 kHz ด้วยโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลการจำลองการทำงานทางซอฟแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz

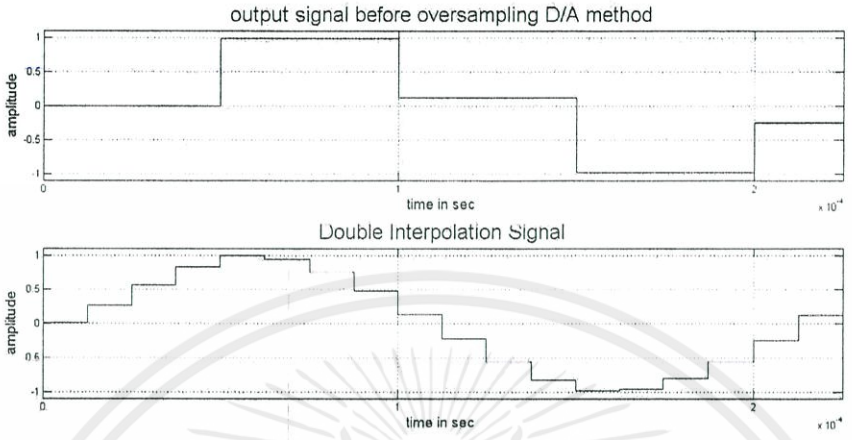


รูปที่ 4.20 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 4.8 kHz ด้วยโปรแกรม MATLAB



รูปที่ 4.21 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 4.8 kHz ด้วยโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 4.8 kHz
ด้วยโปรแกรม MATLAB

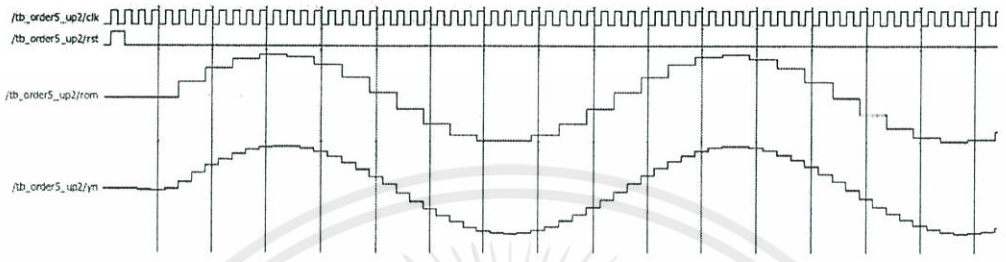
จากรูปที่ 4.14 - 4.22 โดยที่ด้านบนของแต่ละรูปจะเป็นสัญญาณอินพุตที่มีความละเอียดของสัญญาณค่อนข้างต่ำ แต่เมื่อทำ Oversampling D/A แล้วจะเห็นว่าสัญญาณมีค่าความละเอียดที่เพิ่มมากขึ้นตามจำนวนเท่าที่ทำ Oversampling D/A ผลการจำลองนี้ยืนยันได้ว่าวงจรกรองสัญญาณปรับค่าความหน่วงเป็นเศษส่วนชนิดลากรานจ์ในโครงสร้างแฟร์โรว์ดัดแปลงนั้นสามารถนำมาประยุกต์ใช้งานได้จริงในการทำ Oversampling D/A

4.4 ผลการจำลองการทำงานทางฮาร์ดแวร์ในการทำ Oversampling D/A

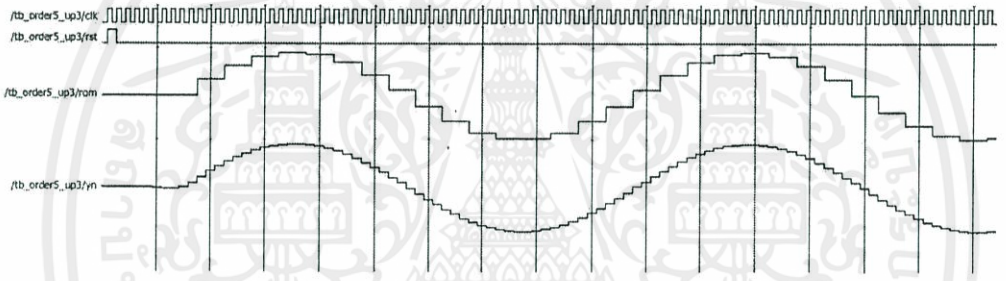
การจำลองการทำงานทางฮาร์ดแวร์จะใช้โปแกรม Xilinx ที่ใช้ภาษา VHDL ในการออกแบบ และสังเคราะห์วงจรทางฮาร์ดแวร์ขึ้นเพื่อทำการ Implement ลงบนอุปกรณ์ FPGA ให้สามารถทำงานได้ โดยเริ่มจากการสร้างสัญญาณคลื่นไซน์เป็นสัญญาณอินพุตที่ความถี่ 1.2 kHz, 3 kHz, 4.8 kHz และ sampling frequency เท่ากับ 20 kHz เพื่อให้เห็นถึงผลของการจำลองการทำงานว่าสัญญาณนั้นมีความละเอียดขึ้นจริงหรือไม่ โดยจะใช้วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ที่อันดับตัวกรองที่ 5 และจะทำการ Oversampling D/A 2, 3 และ 4 เท่า ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

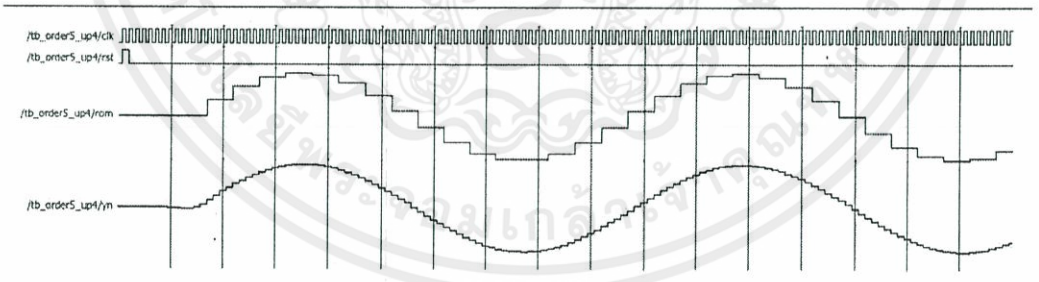
4.4.1 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz



รูป 4.23 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL



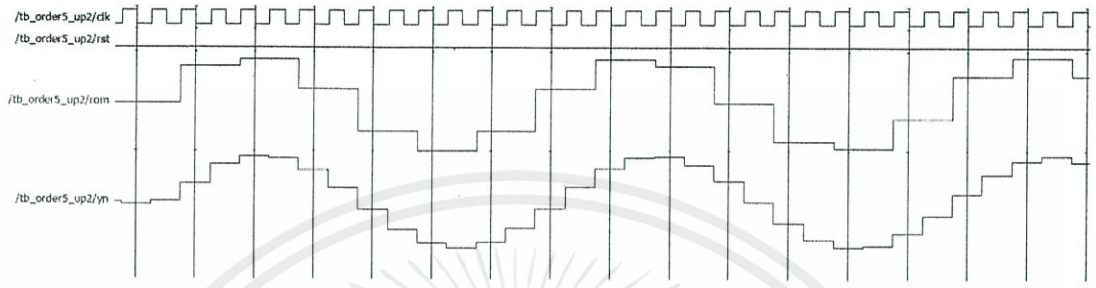
รูป 4.24 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL



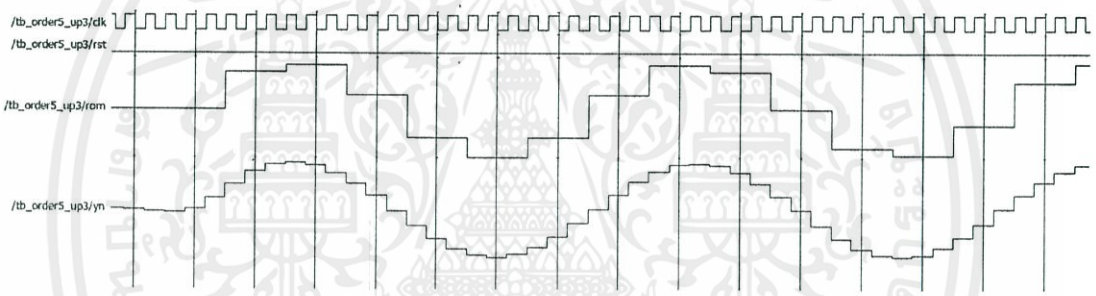
รูป 4.25 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 1.2 kHz ด้วยภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

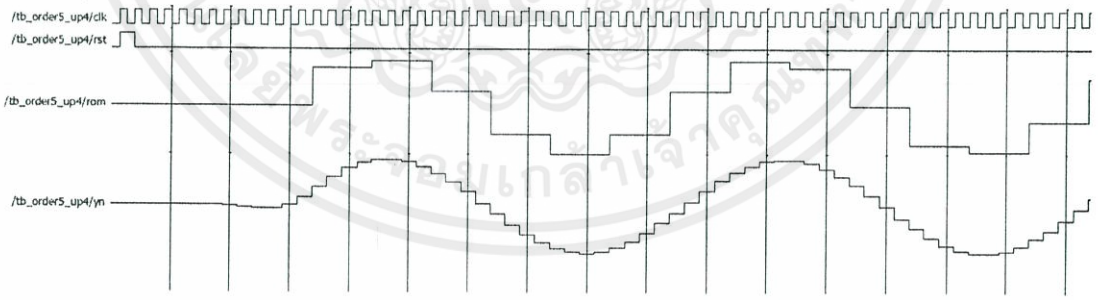
4.4.2 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 3 kHz



รูป 4.26 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL



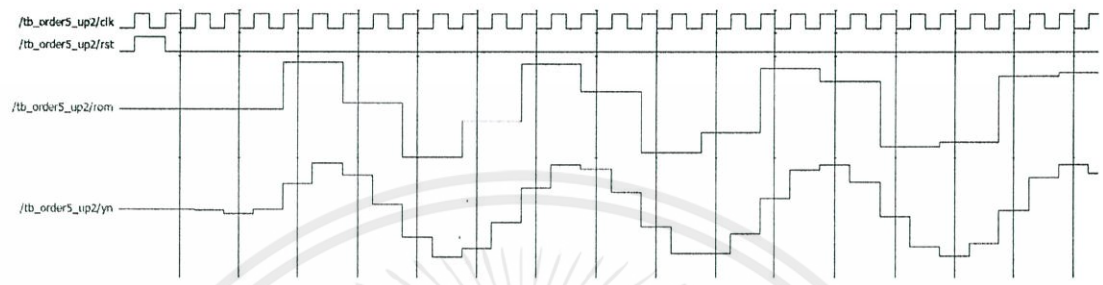
รูป 4.27 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL



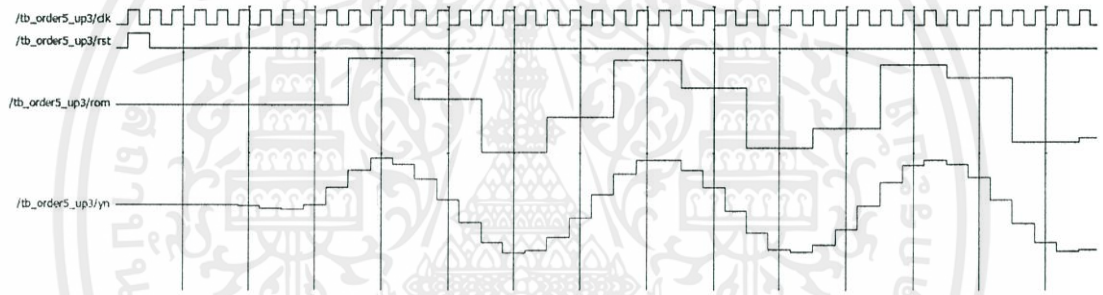
รูป 4.28 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 3 kHz ด้วยภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

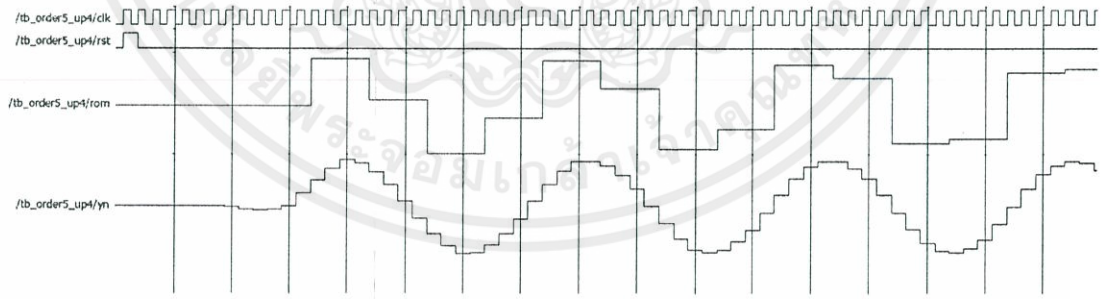
4.4.3 ผลการจำลองการทำงานทางฮาร์ดแวร์ของสัญญาณคลื่นไซน์ที่ความถี่ 4.8 kHz



รูป 4.29 ผลของการทำ Oversampling D/A 2 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL



รูป 4.30 ผลของการทำ Oversampling D/A 3 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL



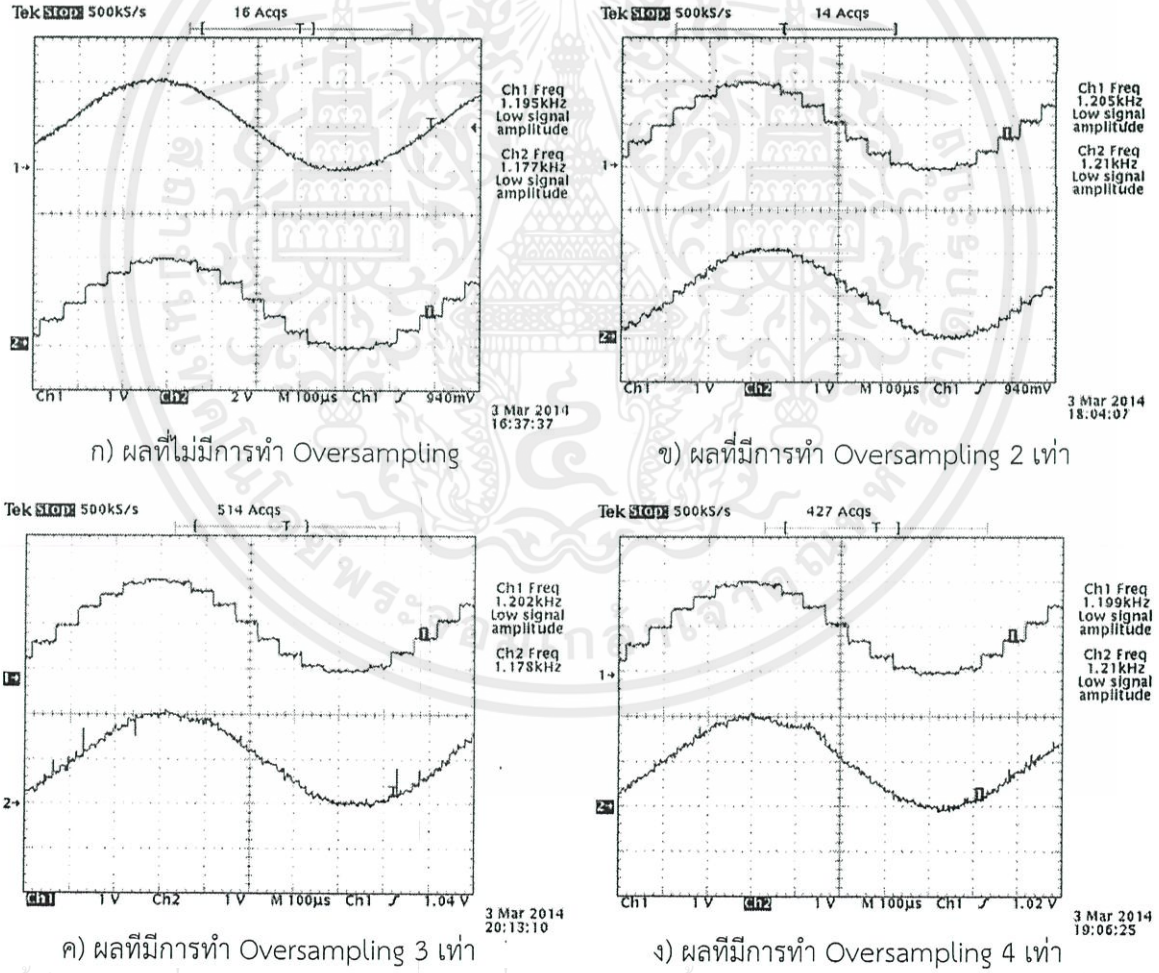
รูป 4.31 ผลของการทำ Oversampling D/A 4 เท่า ที่ความถี่ 4.8 kHz ด้วยภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จะมีลักษณะที่ใกล้เคียงกับการจำลองการทำงานทางซอฟแวร์ด้วยโปรแกรม MATLAB ซึ่งผลการจำลองการทำงานทางฮาร์ดแวร์นี้จะแสดงให้เห็นถึงการออกแบบและการสังเคราะห์วงจรที่ทำงานถูกต้อง และสามารถนำมาประยุกต์ใช้งานทางด้านการทำ Over Sampling D/A ได้จริง

4.5 ผลการทำงานจริงทางฮาร์ดแวร์ในการทำ Oversampling D/A

ผลการทำงานจริงคือผลที่ได้จาก Oscilloscope ซึ่งจะทำการป้อนสัญญาณคลื่นไซน์จากเครื่อง Generator เข้าไปยังวงจรแปลงสัญญาณจากอนาล็อกเป็นสัญญาณดิจิตอล จากนั้นจะเป็นการประมวลผลสัญญาณทางดิจิตอลซึ่งเป็นวงจรที่ออกแบบและสังเคราะห์ขึ้นแล้ว สุดท้ายนำสัญญาณที่ทำการประมวลผลทางดิจิตอลแล้วไปแปลงสัญญาณจากสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกเพื่อแสดงผลบน Oscilloscope ต่อไป โดยผลที่ได้จะสามารถแสดงได้ดังนี้



ก) ผลที่ไม่มีการทำ Oversampling

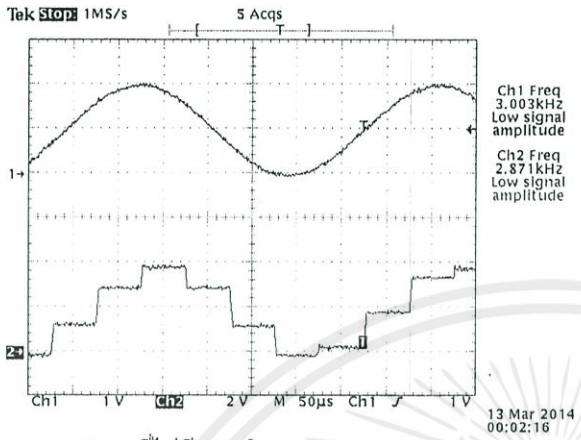
ข) ผลที่มีการทำ Oversampling 2 เท่า

ค) ผลทีมีการทำ Oversampling 3 เท่า

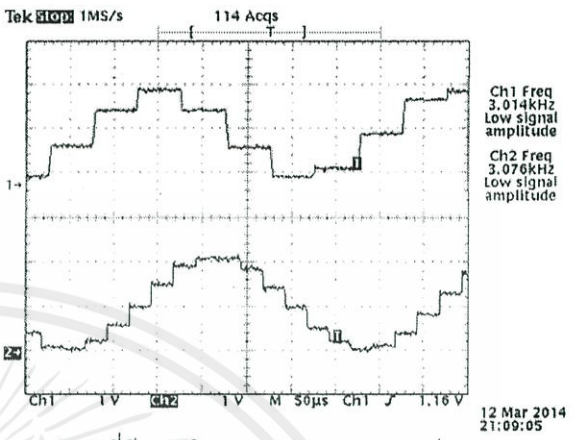
ง) ผลทีมีการทำ Oversampling 4 เท่า

รูปที่ 4.32 แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 1.2 kHz

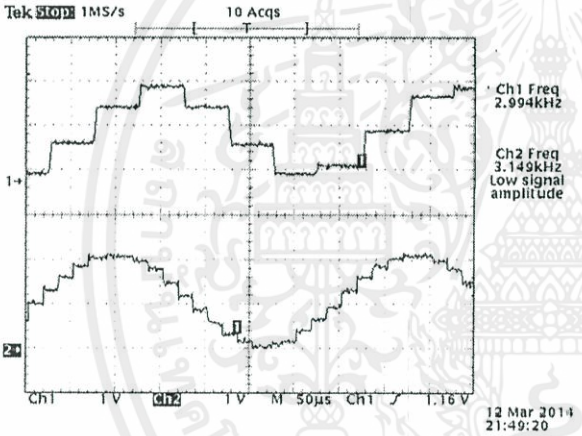
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



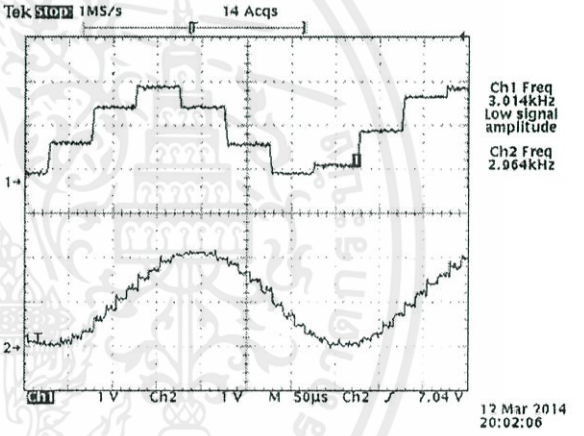
ก) ผลที่ไม่มีการทำ Oversampling



ข) ผลที่มีการทำ Oversampling 2 เท่า



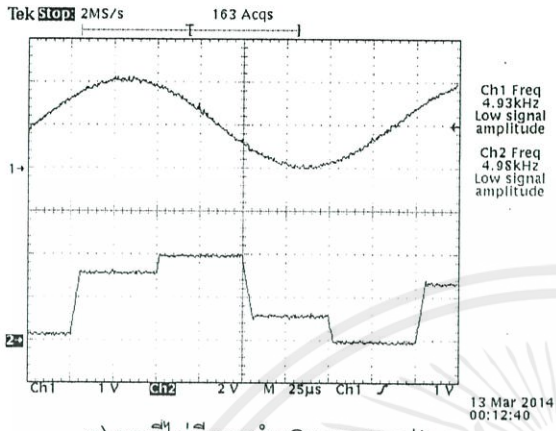
ค) ผลที่มีการทำ Oversampling 3 เท่า



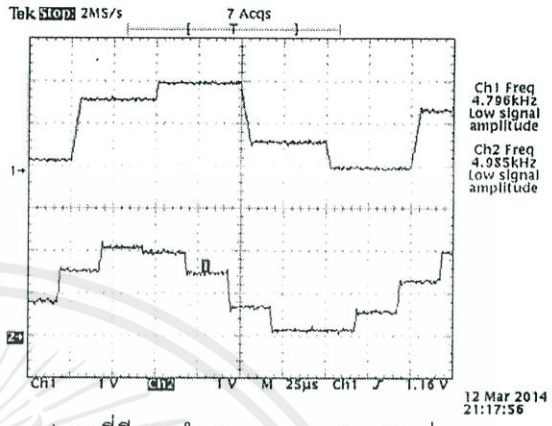
ง) ผลที่มีการทำ Oversampling 4 เท่า

รูปที่ 4.33 แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 3 kHz

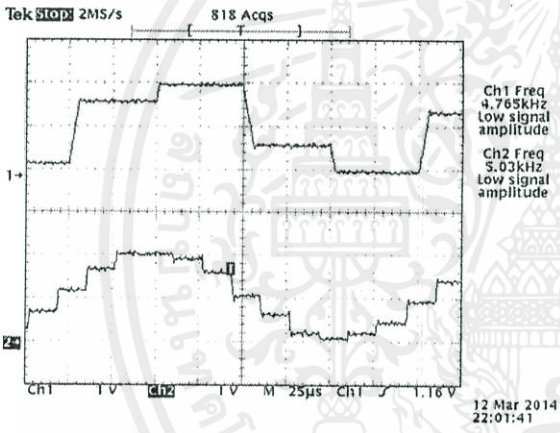
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



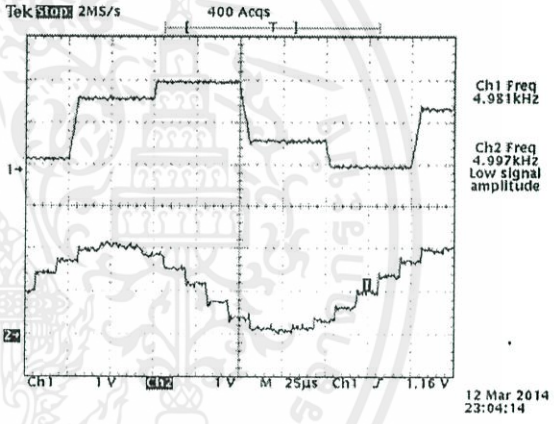
ก) ผลที่ไม่มีการทำ Oversampling



ข) ผลที่มีการทำ Oversampling 2 เท่า



ค) ผลที่มีการทำ Oversampling 3 เท่า



ง) ผลที่มีการทำ Oversampling 4 เท่า

รูปที่ 4.34 แสดงผลการเปรียบเทียบสัญญาณเอาต์พุตของสัญญาณที่ความถี่ 4.8 kHz

จากรูปที่ 4.32 – 4.34 จะแสดงผลการเปรียบเทียบในการทำ Oversampling D/A ที่ความถี่ 1.2 kHz 3 kHz และ 4.8 kHz จะเห็นว่าสัญญาณที่มีความถี่เข้าใกล้ครึ่งหนึ่งของความถี่ Nyquist สัญญาณที่ออกมานั้นจะสูญเสียรายละเอียดของสัญญาณไปดูได้จากรูปที่ 4.32 ก) – 4.34 ก) โดยเฉพาะการทดลองที่ความถี่ 4.8 kHz จะเห็นว่าสัญญาณที่ไม่ได้ทำการ Oversampling D/A มีลักษณะของสัญญาณเดิมน้อยมาก แต่เมื่อทำ Oversampling D/A แล้ว ในรูปที่ 4.32ข) – 4.32ง), 4.33ข) – 4.33ง) และ 4.34ข) – 4.34ง) สัญญาณจะค่อยๆ มีความละเอียดมากยิ่งขึ้นตามการทำ Oversampling D/A ซึ่งแสดงให้เห็นว่าวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิด ลากรานจ์สามารถนำมาประยุกต์ใช้ในการทำ Oversampling D/A ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์มิได้นำมาเพื่อทำการกรองสัญญาณทั่วๆไป แต่จะนำมาใช้ในการหน่วงสัญญาณแบบเป็นเศษส่วน โดยที่ตัวกรองชนิดนี้จะนำมาจัดโครงสร้างแบบแฟร์โรว์และโครงสร้างแบบแฟร์โรว์ดัดแปลง ข้อดีของโครงสร้างดังกล่าวคือสามารถปรับค่าความหน่วงแบบเป็นเศษส่วนได้ด้วยการปรับค่าดีเลย์พารามิเตอร์ โดยที่ไม่จำเป็นต้องคำนวณค่าสัมประสิทธิ์ใหม่ทุกครั้ง และยังสามารถนำมาประยุกต์ใช้งานทางด้าน Oversampling D/A ได้ ซึ่งจะเห็นได้จากผลการจำลองการทำงานทางซอฟแวร์ ผลการจำลองการทำงานทางฮาร์ดแวร์ และผลที่ได้จากการทดสอบจริง

ในส่วนของการจำลองการทำงานทางซอฟแวร์จะใช้โปรแกรม MATLAB ในการออกแบบ และจำลองการทำงาน ซึ่งแสดงให้เห็นถึงคุณสมบัติของตัวกรองที่ได้ทำการออกแบบไว้ จากนั้นจะทำการจำลองสัญญาณขึ้นมาเพื่อนำใช้ในการทดสอบคือ สัญญาณคลื่นไซน์ ที่ความถี่ 1.2 kHz, 3 kHz, 4.8 kHz เมื่อนำสัญญาณดังกล่าวผ่านวงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 5 จะเห็นว่าสัญญาณจะถูกหน่วงสัญญาณแบบไม่เต็มหน่วยไปจริงตามที่เราปรับค่า และนำสัญญาณที่ทำการหน่วงไว้มาประยุกต์ใช้ในการทำ Oversampling D/A 2 เท่า, 3 เท่า, 4 เท่า ซึ่งผลที่ได้ก็คือสัญญาณนั้นจะมีความละเอียดมากขึ้นเมื่อเทียบกับสัญญาณอินพุตเดิม

ในส่วนของการจำลองการทำงานทางฮาร์ดแวร์จะใช้ภาษา VHDL ในการออกแบบและสังเคราะห์วงจร ซึ่งจะทำการสร้างสัญญาณอินพุตจากโปรแกรม MATLAB และทำการแปลงสัญญาณเป็นไบนารี่แบบทูคอมพลีเม้นต์ ซึ่งจะได้สัญญาณอินพุตในการจำลองการทำงานทางฮาร์ดแวร์ที่ความถี่ 1.2 kHz, 3 kHz, 4.8 kHz จากนั้นจะทำการออกแบบ และสังเคราะห์วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์อันดับที่ 5 และทำการออกแบบวงจร Multiplexer เพื่อที่จะทำการรวมสัญญาณระหว่างสัญญาณอินพุตกับสัญญาณที่ทำการหน่วงสัญญาณแบบเป็นเศษส่วน ซึ่งก็คือการทำ Oversampling D/A ผลที่ได้ก็คือสัญญาณนั้นมีความละเอียดมากขึ้น 2 เท่า, 3เท่า และ 4 เท่าตามลำดับเมื่อเทียบกับสัญญาณอินพุตเดิม ซึ่งจากผลการจำลองสามารถยืนยันได้ว่ามีการออกแบบ และสังเคราะห์วงจรได้ถูกต้อง และสามารถนำมาประยุกต์ใช้งานทางด้าน Oversampling D/A ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายคือส่วนของการดำเนินการจริงลงบนอุปกรณ์ฮาร์ดแวร์ ซึ่งฮาร์ดแวร์ที่ใช้ประกอบไปด้วย A/D, FPGA และ D/A โดยใช้ภาษา VHDL ในการสร้าง Hardware ลงบนอุปกรณ์ FPGA และทำการแสดงผลที่ได้ใน Oscilloscope ซึ่งผลที่ได้นั้นจะทำการเปรียบเทียบผลกับสัญญาณที่ยังไม่ทำ Oversampling D/A โดยจะป้อนสัญญาณคลื่นไซน์ที่ความถี่ 1.2 kHz, 3 kHz และ 4.8 kHz ผลที่ได้คือสัญญาณจะมีความละเอียดมากขึ้น 2, 3 และ 4 เท่าจริงตามที่เราได้ออกแบบวงจรไว้ทั้งหมด

5.2 ข้อเสนอแนะ

การออกแบบสังเคราะห์วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ขึ้นเพื่อนำมาลงบนอุปกรณ์ FPGA ให้สามารถทำงานตามที่เราต้องการได้นั้น อาจจะมีข้อจำกัดในเรื่องของทรัพยากรณ์ที่ใช้ ถ้าต้องการให้ประสิทธิ์ภาพในการหน่วงสูงขึ้น จะต้องใช้อันดับตัวกรองที่สูงขึ้นด้วย ทรัพย์กรณ์ที่ใช้อาจจะมากเกินไป ทำให้ FPGA ไม่สามารถทำงานได้

วงจรกรองสัญญาณปรับค่าความหน่วงแบบเป็นเศษส่วนชนิดลากรานจ์ยังสามารถนำไปประยุกต์ใช้งานได้อย่างมากมาย เช่น งานทางด้าน Sampling Rate Conversion (SRC) งานทางด้าน timing recovery และงานทางด้านการวิเคราะห์ภาพ ซึ่งสามารถศึกษาเพิ่มได้จากงานวิจัยเรื่อง Variable Fractional delay filter โดยที่มีนักวิจัยให้ความสนใจเป็นจำนวนมากและได้เผยแพร่งานวิจัยนี้จนถึงปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Sorawat Chivapreecha and Tian - Bo Deng, "Very Low-Complexity Structure for Lagrange-Type Variable Fractional-Delay Filter," in Proc. International Conference on Green Circuits and Systems (ICGCS), pp. 137 – 141, June. 2010.

[2] TIMO I. LAAKSO, VESA VALIMAKI, MATTI KARJALAINEN, and UNTO K. LAINE, "Splitting the unit delay [FIR/all pass filters design]," IEEE Signal Processing Magazine Vol.13, Issue. 1, pp. 30 – 60, Jan. 1996.

[3] ผู้ช่วยศาสตราจารย์ดร.ศรวัฒน์ ชิวปรีชา. เอกสารประกอบการสอน *Digital Signal Processing AND Digital Filter Design*

[4] ผู้ช่วยศาสตราจารย์ดร.ศรวัฒน์ ชิวปรีชา. เอกสารประกอบการสอน Modern Digital System Design and Applications,

[5] ดร.ปัญญา สงวนวัตย์. *คู่มือ MATLAB ฉบับสมบูรณ์*, นนทบุรี : ไอดีซี, 2553

[6] Chaen- Cheng Tseng. *DESIGN OF VARIABLE FRACTIONAL DELAY FIR FILTER USING DIFFERENTIATOR BANK*, pp. 421 – 424, 2002.

[7] Tian-Bo Deng. *Farrow Structures Using Different Transformation Matrices*, pp. 33 – 36, 2010

[8] Vesa Välimäki. *Fractional Delay Filter Design Based on Truncated Lagrange Interpolation*, pp. 816 – 819, NOVEMBER 2007

[9] C.C. Tseng. *Digital Integrator Design Using Recursive Romberg Integration Rule and Fractional Sample Delay*, pp. 2726 – 2729, 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้