

การทำให้เป็นจริงของฮาร์ดแวร์อย่างมีประสิทธิภาพสำหรับการแปลงปาสกาล
เต็มหน่วยโดยใช้การแยกองค์ประกอบของเมตริกซ์และการประยุกต์

EFFICIENT HARDWARE REALIZATION FOR DISCRETE PASCAL
TRANSFORM USING MATRIX FACTORIZATION AND APPLICATION

นริศร รณรงค์ฤทธิ์

NARISON RONNARONGRIT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

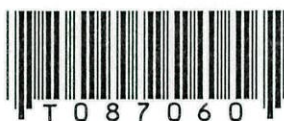
พ.ศ. 2551

KMITL-2008-EN-M-010-412

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การทำให้เป็นจริงของฮาร์ดแวร์อย่างมีประสิทธิภาพสำหรับการแปลงปาสคาล
เต็มหน่วยโดยใช้การแยกองค์ประกอบของเมตริกซ์และการประยุกต์

EFFICIENT HARDWARE REALIZATION FOR DISCRETE PASCAL
TRANSFORM USING MATRIX FACTORIZATION AND APPLICATION

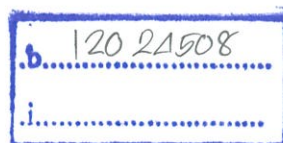


นริสร รณรงค์ฤทธิ์

NARISON RONNARONGRIT

กพ.
๒๖๖๒ก
๒๕๕๑

เลขหมู่.....
เลขทะเบียน..... 87060
วัน,เดือน,ปี..... 30 ส.ค. 2552



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2551

KMITL-2008-EN-M-010-412

**EFFICIENT HARDWARE REALIZATION FOR DISCRETE PASCAL
TRANSFORM USING MATRIX FACTORIZATION AND APPLICATION**

NARISON RONNARONGRIT

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATIONS ENGINEERING
SCHOOL OF GRADUATE STUDIES**

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

KMITL-2008-EN-M-010-412

COPYRIGHT 2008

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การทำให้เป็นจริงของฮาร์ดแวร์อย่างมีประสิทธิภาพสำหรับการแปลงปาสคาลเต็มหน่วยโดยใช้การแยกองค์ประกอบของเมตริกซ์และการประยุกต์
นักศึกษา	นาย นริศร ธรรมรงค์ฤทธิ์
รหัสนักศึกษา	47061005
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2551
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการแปลงปาสคาลเต็มหน่วยสำหรับการประมวลผลสัญญาณเชิงเลข โดยรูปแบบของการดำเนินการของการแปลงปาสคาลเต็มหน่วยจะอาศัยปาสคาลเมตริกซ์เป็นตัวดำเนินการ ดังนั้นทำให้การแปลงสัญญาณจำเป็นต้องใช้จำนวนตัวคูณและตัวบวกเป็นจำนวนมาก ซึ่งขึ้นอยู่กับมิติของเมตริกซ์ตัวดำเนินการที่นำมาใช้ การแยกองค์ประกอบของปาสคาลเมตริกซ์ให้อยู่ในรูปของเมตริกซ์เลขฐานสองจะทำให้การดำเนินการแปลงสัญญาณด้วยปาสคาลเมตริกซ์สามารถทำได้โดยปราศจากตัวคูณใช้เพียงตัวบวกสัญญาณเท่านั้น ทำให้โครงสร้างทางฮาร์ดแวร์ของวงจรที่ใช้ในการแปลงสัญญาณสามารถออกแบบได้อย่างมีประสิทธิภาพ นอกจากนี้จะนำเสนอการประยุกต์ใช้งานการแปลงปาสคาลเต็มหน่วยสำหรับประมวลผลสัญญาณทั้งในกรณีสัญญาณ 1 มิติ และสัญญาณ 2 มิติ

Thesis Title	Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization and Applications.
Student	Mr. Narison Ronnarongrit
Student ID.	47061005
Degree	Master of Engineering
Program	Telecommunications Engineering
Year	2008
Thesis Advisor	Assoc. Prof. Dr. Kobchai Dejhan

ABSTRACT

This thesis presents the discrete Pascal transform for digital signal processing, the operations of discrete Pascal transform is based on Pascal matrix perform as operator. Also, the transformation is necessary to have many multipliers and adders, which depends on the dimensions of used matrix operator. The factorization of Pascal matrix into binary matrices can give the transform by Pascal matrix operates without multipliers and only adders used. Therefore, the hardware realization for transformation circuits can be efficiency design. Moreover, the applications of discrete Pascal transform will be proposed for signal processing both of 1 and 2 dimensional cases.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจากอาจารย์ผู้ควบคุม วิทยานิพนธ์คือ รศ.ดร.กอบชัย เฉลยหาญ และ อาจารย์สรวิวัฒน์ ชิวปรีชา ข้าพเจ้ารู้สึกทราบบ้างในความอนุเคราะห์จากท่านอาจารย์ทั้งสองท่านและขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมาและขอขอบคุณบัณฑิตศึกษา บัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ ที่ให้ความช่วยเหลือในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่อบรมสั่งสอนและเป็นกำลังใจ พร้อมทั้งให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแต่ผู้มีพระคุณทุกท่าน

นริศร รณรงค์ฤทธิ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	3
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนของการศึกษา.....	3
บทที่ 2 ทฤษฎีและหลักการ.....	5
2.1 ทฤษฎีทวินาม (The binomial theorem).....	5
2.2 ทฤษฎีบททวินาม (Binomial Theorem) กับสามเหลี่ยมปาสคาล (Pascal's Triangle).....	7
2.2.1 ที่มาของชื่อสามเหลี่ยมปาสคาล.....	10
2.2.2 การแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform).....	11
2.3 ฟังก์ชันฐาน (Basis function) ของปาสคาลเมตริกซ์.....	11
2.3.1 ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาล ทรานสฟอร์มชนิดความถี่สูงผ่าน (High-pass Type DPT).....	12
2.3.2 ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาล ทรานสฟอร์มชนิดความถี่ต่ำผ่าน (Low-pass Type DPT).....	13
2.4 การแยกปาสคาลเมตริกซ์เป็นเมตริกซ์องค์ประกอบเลขฐานสอง.....	14
2.5 บทสรุป.....	17

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบโครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงปาสคาล เต็มหน่วย.....	18
3.1 การแปลงปาสคาลเต็มหน่วย.....	19
3.1.1 การสร้างเมตริกซ์การแปลงปาสคาลขนาดต่าง ๆ.....	21
3.2 คุณสมบัติของการแปลงปาสคาลเต็มหน่วย.....	22
3.3 โครงสร้างของวงจรแปลงปาสคาลเต็มหน่วยที่ได้จากการแยกองค์ประกอบ เมตริกซ์.....	23
3.3.1 การแปลงเมตริกซ์ของการแปลงปาสคาลให้อยู่ในรูปไบนารี เมตริกซ์.....	23
3.3.2 วิธีการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) โดยใช้เกาเซียน อิลิมีเนชัน (Gaussian elimination).....	25
3.3.3 ขั้นตอนการสร้างไบนารีเมตริกซ์ (1,0,-1).....	30
3.3.4 โครงสร้างของวงจรแปลงดีสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform).....	32
3.4 การแปลงปาสคาลเต็มหน่วยในสองมิติ (2-D Discrete Pascal Transform).....	34
3.5 บทสรุป.....	37
บทที่ 4 การประยุกต์ใช้งานการแปลงปาสคาลเต็มหน่วยสำหรับการประมวลผลสัญญาณ เชิงเลข.....	38
4.1 คุณสมบัติความถี่สูงผ่านของระบบการแปลงปาสคาลเต็มหน่วย.....	38
4.2 คุณสมบัติความถี่ต่ำผ่านของระบบการแปลงปาสคาลเต็มหน่วย.....	46
4.3 การใช้งานเป็นตัวกรองปาสคาล (Pascal Filter) สำหรับสัญญาณเต็มหน่วย.....	49
4.4 ผลตอบสนองอิมพัลส์ (Impulse response) ของวงจรกรองปาสคาล 2 มิติ (2-D Pascal Filter).....	51
4.5 บทสรุป.....	55
บทที่ 5 ผลการออกแบบและผลการจำลองการทำงาน.....	57
5.1 ผลการออกแบบ.....	57
5.1.1 ผลที่ได้จากการแปลงสามเหลี่ยมปาสคาลใน 1 มิติ.....	57

สารบัญ (ต่อ)

	หน้า
5.1.1.1 การจำลองการทำงานของข้อมูล	58
5.1.1.2 ผลการจำลองการทำงานชนิดความถี่สูงผ่าน	62
5.1.1.3 ผลการจำลองการทำงานชนิดความถี่ต่ำผ่าน	66
5.2 บทสรุป.....	69
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	71
6.1 สรุปผลการดำเนินการวิจัย.....	71
6.2 แนวทางในการพัฒนา.....	71
บรรณานุกรม.....	72
ภาคผนวก.....	74
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่.....	75
ประวัติผู้เขียน.....	88

สารบัญตาราง

ตารางที่	หน้า
2.1 การกระจายเลขชี้กำลัง.....	5
2.2 สามเหลี่ยมของปาสคาล.....	6
5.1 สรุปจำนวนการใช้ตัวบวกหรือตัวลบในการสร้างวงจรของการแปลงปาสคาลทราน สฟอร์มแบบ N จุด.....	70

สารบัญรูป

รูปที่	หน้า
2.1 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก.....	9
2.2 แสดงสามเหลี่ยมของ ยาง ฮุย.....	10
2.3 แสดงตำแหน่งองค์ประกอบ '0' ในส่วนของสามเหลี่ยมบนของแต่ละเมตริกซ์ $[q_{ij}]_k$	14
3.1 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วย.....	21
3.2 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วยชนิดความถี่สูงผ่าน (High-pass type).....	24
3.3 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วยชนิดความถี่ต่ำผ่าน (Low-pass type)	25
3.4 แสดงโครงสร้างการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) แบบ 4 จุด.....	33
3.5 แสดงโครงสร้างการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) แบบ 4 จุด....	34
3.6 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วยแบบ 2 มิติ เมื่อ $N=3$	36
3.7 แสดงโครงสร้างแบบวงจรรายในที่ใช้ในสัญลักษณ์ตามรูปที่ 3.6 เมื่อ $N=3$	36
4.1 โครงสร้างของวงจรรองปาสคาลจากสมการที่ (4.5).....	39
4.2 ผลตอบสนองทางแอมพลิจูดจากสมการที่ (4.5).....	39
4.3 โครงสร้างของวงจรรองปาสคาลจากสมการที่ (4.6).....	40
4.4 ผลตอบสนองทางความถี่จากสมการที่ (4.6).....	40
4.5 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{th} order.....	41
4.6 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{th} order และทำการนอร์มอลไลซ์แล้ว.....	41
4.7 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order.....	42
4.8 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order และทำการนอร์มอลไลซ์แล้ว.....	42
4.9 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order.....	43
4.10 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order และทำการนอร์มอลไลซ์แล้ว.....	43
4.11 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order.....	44

สารบัญญรูป(ต่อ)

รูปที่	หน้า
4.12 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order และทำการนอร์มอลไลซ์แล้ว.....	44
4.13 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order.....	45
4.14 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order เมื่อทำการนอร์มอลไลซ์แล้ว.....	45
4.15 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 99^{th} order.....	46
4.16 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order.....	48
4.17 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order เมื่อทำการนอร์มอลไลซ์แล้ว.....	48
4.18 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 99^{th} order.....	49
4.19 แสดงหน้ากากคอนโวลูชันของวงจรถองปาสคาล 2 มิติ ซึ่งมีขนาด 3×3	51
4.20 โครงสร้างของวงจรถองปาสคาล 2 มิติ ตามสมการที่ (2.40).....	52
4.21 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 3×3	52
4.22 หน้ากากคอนโวลูชันของวงจรถองปาสคาล 2 มิติ ซึ่งมีขนาด 3×3 ชนิดความถี่ต่ำผ่าน.....	53
4.23 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 3×3 ชนิดความถี่ต่ำผ่าน.....	53
4.24 วงจรสร้างลำดับของข้อมูลก่อนนำไปประมวลผลกับวงจรถองปาสคาล 2 มิติ.....	54
4.25 แสดงบัตรเตอร์ฟลายยูนิท เมื่อ $N=3$	54
4.26 ตัวอย่างรายละเอียดของวงจรบัตเตอร์ฟลายยูนิท เมื่อ $N=3$	55
5.1 แสดงผลตอบสนองทางขนาดของความเป็นตัวกรองความถี่สูงผ่านเมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 5^{th} order.....	57
5.2 ผลจากการจำลองการทำงานของโปรแกรมแปลงปาสคาลเต็มหน่วยดังใน [1] ขนาด 3×3 จาก MATLAB.....	58

สารบัญรูป(ต่อ)

รูปที่	หน้า
5.3 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ 2 มิติ ขนาด 4x4 จาก MATLAB.....	58
5.4 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ 2 มิติ ขนาด 5x5 จาก MATLAB.....	59
5.5 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ 2 มิติ ขนาด 6x6 จาก MATLAB.....	59
5.6 ผลจากการจำลองการทำงานของวงจรถ่ายฟูริเยร์ 1 มิติ อันดับที่ 2 จาก MATLAB....	60
5.7 ผลจากการจำลองการทำงานของวงจรถ่ายฟูริเยร์ 1 มิติ อันดับที่ 3 จาก MATLAB....	60
5.8 ผลจากการจำลองการทำงานของวงจรถ่ายฟูริเยร์ 1 มิติ อันดับที่ 4 จาก MATLAB....	61
5.9 ผลจากการจำลองการทำงานของวงจรถ่ายฟูริเยร์ 1 มิติ อันดับที่ 5 จาก MATLAB....	61
5.10 ผลจากการจำลองการทำงานของการแปลงคอสซีนฟูริเยร์ 2 มิติ ขนาด 1 มิติ ชนิดความถี่สูงผ่าน.....	62
5.11 ผลเปรียบเทียบการจำลองการทำงานของการแปลงคอสซีนฟูริเยร์ 2 มิติ ขนาด 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4 th order กับการแปลงฟูริเยร์ 2 มิติ ชนิดความถี่สูงผ่านโดยทดลองทำที่ 4 th order.....	63
5.12 ภาพมาตรฐานชื่อ Camera man ที่ถูกใช้เป็นสัญญาณอินพุตขนาด 256 x 256 จุด.....	64
5.13 เปรียบเทียบผลของการแปลงฟูริเยร์ 2 มิติ ชนิดความถี่สูงผ่านและการแปลงคอสซีนฟูริเยร์ 2 มิติ ชนิดความถี่สูงผ่าน.....	65
5.14 ผลจากการจำลองการทำงานของการแปลงคอสซีนฟูริเยร์ 2 มิติ ขนาด 1 มิติ ชนิดความถี่ต่ำผ่าน.....	66
5.15 ผลเปรียบเทียบการจำลองการทำงานของการแปลงคอสซีนฟูริเยร์ 2 มิติ ขนาด 1 มิติ ชนิดความถี่ต่ำผ่านแบบ 5 จุด หรือ 4 th order กับการแปลงฟูริเยร์ 2 มิติ ชนิดความถี่ต่ำผ่านโดยทดลองทำที่ 4 th order.....	67
5.16 เปรียบเทียบผลของการแปลงฟูริเยร์ 2 มิติ ชนิดความถี่ต่ำผ่านและการแปลงคอสซีนฟูริเยร์ 2 มิติ ชนิดความถี่ต่ำผ่าน.....	69

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัญหาที่กำลังศึกษาและวิจัยเป็นการศึกษาถึงที่มาและการใช้งานเมตริกซ์พิเศษตัวหนึ่ง ที่เรียกกันว่าปาสคาลเมตริกซ์ (Pascal Matrix) ซึ่งที่มาของเมตริกซ์ดังกล่าวนี้มาจากทฤษฎีบทที่เรียกว่าทฤษฎีบททวินาม (Binomial Theorem) และตัวทฤษฎีบทนี้เองก็เกี่ยวข้องกับการเกิดของสามเหลี่ยมปาสคาล (Pascal's Triangle) หลักการใช้งานปาสคาลเมตริกซ์ในปัจจุบันมีการนำไปประยุกต์ใช้งานกันอย่างแพร่หลายในงานทางด้านการประมวลผลสัญญาณเชิงเลข โดยใน [1-5] ได้มีการนำเอาหลักการของทฤษฎีบททวินามมาใช้งานร่วมกับการแปลงตัวแปรระหว่าง Analog Domain กับ Digital Domain เป็นผลให้ได้มาซึ่งปาสคาลเมตริกซ์ที่ใช้ในการแปลงตัวแปรระหว่างตัวแปร s และ ตัวแปร z สำหรับการแปลงระหว่าง Domain ทั้งสอง

สำหรับในงานวิจัยที่กำลังศึกษาวิจัยอยู่นั้นจะเป็นการศึกษาและวิจัยตัวปาสคาลเมตริกซ์ซึ่งแตกต่างจากปาสคาลเมตริกซ์ที่ใช้ในการแปลงตัวแปรดังในกรณี [1-5] แต่เป็นปาสคาลเมตริกซ์ที่สมาชิก (หรือจะเรียกว่าแต่ละ elements ก็ได้) ภายในเมตริกซ์นั้นได้มาจากการนำความสัมพันธ์ที่ใช้ในการสร้างสามเหลี่ยมปาสคาลมาสร้างให้เป็น elements ภายในปาสคาลเมตริกซ์นี้โดยแต่ละ column จะมีเครื่องหมาย บวก-ลบ สลับกันไป ซึ่งแนวความคิดเบื้องต้นมาจาก [6] จากนั้นจะให้ความสนใจไปที่ขั้นตอนการใช้งานปาสคาลเมตริกซ์ดังกล่าวในฐานะตัวดำเนินการในการแปลงสัญญาณ (Transform Signal : มิใช่ใช้สำหรับการแปลงตัวแปรดังในเช่น [1-5]) ยิ่งไปกว่านั้นจากการสังเกตผลหรือพฤติกรรมของการแปลงสัญญาณด้วยปาสคาลเมตริกซ์ดังกล่าว ซึ่งต่อไปนี้จะเรียกว่าการแปลงปาสคาลเต็มหน่วย (Discrete Pascal Transform) ให้ผลการทำงานเป็นไปในลักษณะของการผ่านความถี่สูงอีกด้วย (High-pass Filtering) ดังนั้นจึงทำการหาฟังก์ชันถ่ายโอน (Transfer function) ของระบบที่ทำหน้าที่แปลงปาสคาลเต็มหน่วยดังกล่าวอีกด้วย เพื่อนำมาวิเคราะห์ถึงคุณสมบัติของผลตอบสนองทางความถี่ของระบบดังกล่าว ซึ่งระบบดังกล่าวนี้เราอาจจะเรียกเป็นชื่อเล่นๆ ว่า Pascal Filter ณ.ที่นี้ ว่าผลตอบสนองของ Pascal Filter นี้มีความเป็น High-pass Filtering จริงหรือไม่ (ซึ่งหัวข้อนี้ยังมีใครเคยพิสูจน์มาก่อน)

อีกส่วนที่จะทำการศึกษาและวิจัยในวิทยานิพนธ์นี้คือกรรมวิธีในการแยกองค์ประกอบของปาสคาลเมตริกซ์ให้อยู่ในรูปของเมตริกซ์องค์ประกอบที่เป็นเมตริกซ์เลขฐานสอง (Pascal Matrix Factorization to Binary Matrices) เพื่อทำให้ขั้นตอนการแปลงซึ่งต้องทำการคูณระหว่างเวกเตอร์ของสัญญาณ (กรณีสัญญาณ 1-D) หรือเมตริกซ์ของสัญญาณ (กรณีสัญญาณ 2-D) ที่จะแปลงกับ

ปาสคาลเมตริกซ์สามารถหาผลการคูณได้โดยปราศจากตัวคูณ โดยอาศัย [7-9] ในการแยกองค์ประกอบของเมตริกซ์และจัดรูปกรรมวิธีการคำนวณในเชิงของฮาร์ดแวร์ ซึ่งการอาศัยกรรมวิธีในการแยกองค์ประกอบของเมตริกซ์ (Matrix Factorization) ใน [8] จะช่วยทำให้ได้โครงสร้างทางฮาร์ดแวร์ที่ประกอบไปด้วย Butterfly Unit ดังที่นำเสนอใน [9]

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ความน่าสนใจของปัญหาถือเป็นกระบวนการศึกษาและพัฒนา รูปแบบของการแปลงสัญญาณแบบใหม่ นอกเหนือไปจากการแปลงสัญญาณแบบที่รู้จักกันโดยทั่วไป เช่น การแปลงฟูเรียร์เต็มหน่วย (Discrete Fourier Transform: DFT), การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform: FFT), การแปลงโคไซน์เต็มหน่วย (Discrete Cosine Transform: DCT) หรือการแปลงเวฟเลตเต็มหน่วย (Discrete Wavelet Transform: DWT) เน้นอนว่ารูปแบบการแปลงสัญญาณโดยทั่วไปสามารถไม่ว่าจะเป็น การแปลงสัญญาณแบบใดสามารถจัดให้อยู่ในรูปของการดำเนินการเชิงเมตริกซ์ได้ทั้งสิ้น ในกรณีของการแปลงปาสคาลเต็มหน่วย (Discrete Pascal Transform: DPT) ก็เช่นกัน

ดังนั้นความน่าสนใจของปัญหานี้ก็อีกอย่างก็คือการหาหรือออกแบบ โครงสร้างการคำนวณเชิงฮาร์ดแวร์เพื่อใช้สำหรับการนำไปสร้างเป็นวงจรฮาร์ดแวร์จริงหรือจะช่วยให้การคำนวณเชิงซอฟต์แวร์ทำงานได้เร็วกว่าการคำนวณด้วยการคูณเมตริกซ์โดยตรงก็ได้ ซึ่งผลของการศึกษาปัญหาดังกล่าวนี้จะทำให้ได้โครงสร้างของ Butterfly Unit สำหรับการคำนวณของ DPT คล้ายๆ กับโครงสร้างของ Butterfly Unit ที่ใช้กันอยู่สำหรับการแปลง FFT ด้วยโครงสร้างของ Butterfly Unit สำหรับ DPT ทำให้สามารถทำการคำนวณหาผลของการแปลงสัญญาณสามารถทำได้ง่าย โดยเฉพาะอย่างยิ่งถ้านำไปสร้างเป็นฮาร์ดแวร์จริง โครงสร้างทางฮาร์ดแวร์ที่ได้ของการคำนวณ DPT จะไม่จำเป็นต้องใช้ตัวคูณสัญญาณเลย ในขณะที่ทำการคำนวณด้วยการคูณระหว่างปาสคาลเมตริกซ์กับสัญญาณโดยตรงจะต้องใช้จำนวนตัวคูณสัญญาณเป็นจำนวนมาก (ทั้งนี้ขึ้นอยู่กับ Dimension ของ Pascal Matrix ที่ใช้)

1.3 สมมติฐานของการศึกษา

กรรมวิธีในการแยกองค์ประกอบของปาสคาลเมตริกซ์ให้อยู่ในรูปของเมตริกซ์องค์ประกอบที่เป็นเมตริกซ์เลขฐานสอง (Pascal Matrix Factorization to Binary Matrices) เพื่อให้ขั้นตอนการแปลงซึ่งต้องทำการคูณระหว่างเวกเตอร์ของสัญญาณ (กรณีสัญญาณ 1-D) หรือเมตริกซ์ของสัญญาณ (กรณีสัญญาณ 2-D) ที่จะแปลงกับปาสคาลเมตริกซ์สามารถหาผลการคูณได้

โดยปราศจากตัวคูณ โดยอาศัย [7-9] ในการแยกองค์ประกอบของเมตริกซ์และจัดรูปกรรมวิธีการคำนวณในเชิงของฮาร์ดแวร์ ซึ่งการอาศัยกรรมวิธีในการแยกองค์ประกอบของเมตริกซ์ (Matrix Factorization) ใน [8] จะช่วยทำให้ได้โครงสร้างทางฮาร์ดแวร์ที่ประกอบไปด้วย Butterfly Unit ดังที่นำเสนอใน [9]

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

หลักการที่นำเสนอใหม่จะประกอบด้วยการศึกษา (Investigation) ถึงคุณสมบัติความเป็นวงจรรองสัญญาณความถี่สูง (High-pass filtering) ที่เกิดขึ้นจากการแปลงปาสคาลเต็มหน่วย และนำคุณสมบัติที่ได้นี้มาใช้ประโยชน์สำหรับการประมวลผลสัญญาณ จากนั้นจะทำการจัดรูปแบบ (Formulate) เพื่อหาเป็นฟังก์ชันถ่าย (transfer function) สำหรับระบบที่เกิดขึ้น

สรุปวิธีการแยกเมตริกซ์องค์ประกอบเป็นอัลกอริธึมที่ง่าย ๆ (โดยรูปแบบขั้นตอนสามารถเข้าใจได้ง่ายกว่าใน [8]) และทำการทวนสอบ (Verify) ถึงที่มาของวิธีการแยกองค์ประกอบ (Factorization) ของปาสคาลเมตริกซ์ให้เป็นเมตริกซ์เลขฐานสอง (Binary matrices) [7] พร้อมทั้งแสดงถึงที่มาของ butterfly unit สำหรับการแปลงปาสคาลเต็มหน่วย ซึ่งจุดนี้เองเป็นส่วนที่ทำให้โครงสร้างของการแปลงปาสคาลเต็มหน่วยสามารถจัดโครงสร้างทางฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ

สุดท้ายคือการประยุกต์ใช้การแปลงปาสคาลเต็มหน่วยสำหรับการกรองความถี่สูงทั้งสำหรับสัญญาณ 1 มิติ และสัญญาณ 2 มิติ

1.5 ขอบเขตการวิจัย

วิทยานิพนธ์นี้เป็นการศึกษาทำให้เป็นจริงของฮาร์ดแวร์อย่างมีประสิทธิภาพสำหรับการแปลงปาสคาลเต็มหน่วยโดยใช้การแยกองค์ประกอบของเมตริกซ์และการประยุกต์ ดำเนินการวิจัย คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เริ่มต้นดำเนินงานวิจัยในภาคเรียนที่ 1 ปีการศึกษา 2550 และสิ้นสุดในภาคเรียนที่ 2 ปีการศึกษา 2550 รวมระยะเวลาดำเนินงานวิจัย เป็นเวลา 1 ปี

1.6 ขั้นตอนของการศึกษา

โครงการฉบับนี้เป็นการทำให้เป็นจริงของฮาร์ดแวร์อย่างมีประสิทธิภาพสำหรับการแปลงปาสคาลเต็มหน่วยโดยใช้การแยกองค์ประกอบของเมตริกซ์ เพื่อทำให้ขั้นตอนการแปลงซึ่งต้องทำการคูณระหว่างเวกเตอร์ของสัญญาณ (กรณีสัญญาณ 1-D) หรือเมตริกซ์ของสัญญาณ (กรณีสัญญาณ 2-D) ที่จะแปลงกับปาสคาลเมตริกซ์สามารถหาผลการคูณได้โดยปราศจากตัวคูณ โดย

อาศัย [7-9] ในการแยกองค์ประกอบของเมตริกซ์และจัดรูปกรรมวิธีการคำนวณในเชิงของฮาร์ดแวร์ ซึ่งการอาศัยกรรมวิธีในการแยกองค์ประกอบของเมตริกซ์ (Matrix Factorization) ใน [8] จะช่วยทำให้ได้โครงสร้างทางฮาร์ดแวร์ที่ประกอบไปด้วย Butterfly Unit ดังที่นำเสนอใน [9] โดยทดสอบคุณสมบัติของวงจรกรองที่ออกแบบนี้ด้วยการจำลองเลขชี้ด้วยโปรแกรมแมทแลบ (MATLAB)

บทที่ 2

ทฤษฎีและหลักการ

2.1 ทฤษฎีทวินาม (The binomial theorem)

ทวินามหรือไบโนเมียล คือ สมการพีชคณิตที่มีเทอม 2 เทอม เช่น $(a+b)$ ในที่นี้จะกล่าวถึงสมการทั่วไปที่เรียกว่า ทฤษฎีทวินามสำหรับการยกกำลังทวินามเริ่มต้นจากการคูณ หรือการกระจายกำลัง $(a+b)^n$ เมื่อ n เป็นศูนย์หรือจำนวนเต็มบวกใดๆ เช่น $n = 0, 1, 2, 3, 4, 5$ จะได้ผลลัพธ์ในรูปผลรวมของพจน์ต่างๆดังแสดงในตารางที่ 2.1 ต่อไปนี้

ตารางที่ 2.1 การกระจายเลขชี้กำลัง

n	$(a+b)^n$	ผลที่ได้จากการกระจายกำลัง
0	$(a+b)^0$	1
1	$(a+b)^1$	$a+b$
2	$(a+b)^2$	$a^2 + 2ab + b^2$
3	$(a+b)^3$	$a^3 + 3a^2b + 3ab^2 + b^3$
4	$(a+b)^4$	$a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$
5	$(a+b)^5$	$a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$

จะเห็นว่าแถวต่างในตารางเริ่มต้นด้วยเทอม a^5 ซึ่งเราพิจารณาเป็น a^5b^0 เทอมที่ตามหลังมามีเลขชี้กำลังของ a ลดลงเทอมละ 1 และเลขชี้กำลังของ b เพิ่มขึ้นเทอมละ 1 จนมีเลขชี้กำลังเป็น 5 สำหรับในแถวอื่นๆก็มีลักษณะเช่นเดียวกัน ถ้าดับต่อไปเป็นการพิจารณาสัมประสิทธิ์และการหาค่าสัมประสิทธิ์

$$(a+b)^5 = (a+b)(a+b)(a+b)(a+b)(a+b) \quad (2.1)$$

สังเกตว่าสมการ (4.25) มีทวินาม 5 ตัว และเทอมที่อยู่ในรูปแบบที่ได้รับการขยายเป็นผลลัพธ์ของผลคูณของเทอมในทวินาม เช่นการหา a^3b^2 ในรูปการขยายของ $(a+b)^n$ นั้น เราจะต้องคูณเทอม a จากทวินาม 3 ตัว กับเทอม b อื่นๆ จากทวินาม 2 ตัว กรณีนี้หาแฟกเตอร์ a

ได้ $\binom{5}{3}$ วิธี ซึ่งก็คือ จำนวนวิธีที่เราเลือกทวินาม 3 ตัวจากทวินาม 5 ตัว ในทำนองเดียวกัน ใน
 การหาแฟกเตอร์ b ทำได้ $\binom{5}{2}$ วิธี ซึ่งก็คือ จำนวนวิธีที่เราเลือกทวินาม 2 ตัวจากทวินาม 5 ตัว กรณี
 เช่นนี้พิสูจน์ได้ว่า $\binom{5}{3} = \binom{5}{2} = 10$ และ จาก $(a+b)^n$ ทำให้สรุปได้ว่าสัมประสิทธิ์ของเทอม
 $a^{n-k} b^k$ คือ $\binom{n}{k}$ หรือ $\binom{n}{n-k}$ จากคำอธิบายเหล่านี้ ทำให้ทราบว่า ทฤษฎีทวินาม เป็น
 ทฤษฎีสำหรับการขยาย $(a+b)^n$ ให้อยู่รูปผลบวก อีกนั่นหนึ่งทฤษฎีทวินามโดยที่ n เป็นจำนวน
 เต็มบวก คือ

$$\begin{aligned} (a+b)^n &= \binom{n}{0} a^n b^0 + \binom{n}{1} a^{n-1} b^1 + \binom{n}{2} a^{n-2} b^2 + \binom{n}{3} a^3 b^3 + \dots + \binom{n}{n} a^0 b^n \\ &= a^n + \binom{n}{1} a^{n-1} b + \binom{n}{2} a^{n-2} b^2 + \binom{n}{3} a^3 b^3 + \dots + b^n \\ &= \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k \end{aligned} \quad (2.2)$$

นอกจากนี้ ยังเรียก $\binom{n}{k}$ ว่า สัมประสิทธิ์ทวินาม (Binomial coefficient) มีอีกวิธีหนึ่งที่ใช้
 คำนวณสัมประสิทธิ์ทวินามของ $(A+B)^n$ เมื่อ $n = 0, 1, 2, 3, 4, 5$ และจัดแถวสามเหลี่ยมใน
 ตารางที่ 2.2 ที่เรียกว่าสามเหลี่ยมของปาสคาลดังนี้

ตารางที่ 2.2 สามเหลี่ยมของปาสคาล

n	สัมประสิทธิ์ทวินาม					
0	1					
1	1	1				1
2	1	2	2		1	1
3	1	3	3	3	1	1
4	1	4	6	4	1	1
5	1	5	10	10	5	1

สิ่งที่ควรทราบในการกระจาย $(a+b)^n$ มีดังนี้

- ผลรวมของการกระจายมี $n+1$ พจน์เสมอ
- ผลรวมของกำลังของ a และ b ในแต่ละพจน์ต้องเท่ากับ n
- กำลังของ a ในพจน์แรกจะเริ่มจาก n แล้วลดลงไปที่ละ 1 ในแต่ละพจน์ถัดไปจนกระทั่งถึง 0
- กำลังของ b ในพจน์แรกจะเริ่มจาก 0 แล้วเพิ่มไปที่ละ 1 ในแต่ละพจน์ถัดไปจนกระทั่งถึง n

- สัมประสิทธิ์ของพจน์แรกเริ่มจาก $\binom{n}{0}$ และพจน์ถัดไปเป็น $\binom{n}{1}, \binom{n}{2}$ ไปเรื่อยๆ จนกระทั่งสุดท้ายเป็น $\binom{n}{n}$

- ถ้าแทน $a=1$ และ $b=1$ ไปในสมการที่ (2.2) เราจะได้ลักษณะที่สำคัญคือ

$$2^n = \binom{n}{1}a + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} \quad (2.3)$$

- พจน์ที่ $k = T_k = \binom{n}{k+1} a^{n-(k+1)} b^{k+1}$ แต่เพื่อความสะดวกเรานิยมใช้พจน์ที่ $k+1$ ดังสมการที่ (2.4)

$$k+1 = T_{k+1} = \binom{n}{k} a^{n-k} b^k \quad (2.4)$$

- ถ้า n เป็นจำนวนเต็มบวกและ $c < k < n$ แล้ว $C_{n,k} = C_{n-1,k-1} + C_{n-1,k}$
- $C_{k,k} = C_{k+1,k} + C_{n+2,k} + \dots + C_{n,k} = C_{n+1,k+1}$

2.2 ทฤษฎีทวินาม (Binomial Theorem) และสามเหลี่ยมปาสคาล (Pascal's Triangle)

ในวิทยานิพนธ์นี้จะทำการอธิบายถึงทฤษฎีบททวินามและความสัมพันธ์ของทฤษฎีดังกล่าวกับสามเหลี่ยมปาสคาลว่ามีความสัมพันธ์กับทฤษฎีบททวินามอย่างไร ยกตัวอย่างการอธิบายทฤษฎีบททวินาม เช่น ถ้าต้องการหาค่าหรือกระจายเทอม $(x+y)^2$ สามารถทำการกระจายได้ดังสมการ

$$(x+y)^2 = x^2 + 2xy + y^2 \quad (2.5)$$

ถ้าเราต้องการกระจายเทอม $(x+y)^n$ ใด ๆ จะหาได้โดยทฤษฎีบททวินาม ดังสมการ

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k \quad (2.6)$$

$$= x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \dots + y^n \quad (2.7)$$

$$= a_0 x^n + a_1 x^{n-1} y + a_2 x^{n-2} y^2 + \dots + a_{n-1} x y^{n-1} + a_n y^n \quad (2.8)$$

ซึ่งการกระจายเทอมในสมการ (2.6) - (2.8) บางครั้งเรียกได้ว่าเป็นการกระจายเทอมไบนอมิเยล (Binomial Expansion) โดย a_k เรียกว่าสัมประสิทธิ์ไบนอมิเยล (Binomial coefficient) ซึ่งจะมีค่าดังสมการ

$$a_k = \binom{n}{k} \quad (2.9)$$

โดยความหมายของสัมประสิทธิ์ไบนอมิเยลหรืออาจเรียกได้ว่าเป็นคอมไบเนทอเรียลนัมเบอร์ (Combinatorial number) คือ จำนวนวิธีการของการหยิบของ k สิ่งออกจากผลทั้งหมดที่เป็นไปได้ n สิ่ง หรือที่รู้จักกันว่า การจัดหมู่ (Combination number) ดังที่ได้พบเห็นในวิชาความน่าจะเป็นและสถิติ โดย

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!} = a_k \quad (2.10)$$

โดยที่ n, k เป็นจำนวนเต็มที่มีค่าเป็นบวก และ $n \geq k$ โดยที่ $\binom{n}{0} = \binom{n}{n} = 1$

สำหรับค่าของสัมประสิทธิ์ไบนอมิเยลนั้นสามารถนำมาจัดเรียงเป็นรูปฟอร์มที่เป็นสามเหลี่ยม ที่รู้จักกันในชื่อว่า สามเหลี่ยมปาสคาล ได้ดังนี้

1	→	n=0								
1	1	→ n=1								
1	2	1	→ n=2							
1	3	3	1							
1	4	6	4	1						
1	5	10	10	5	1					
1	6	15	20	15	6	1				
1	7	21	35	35	21	7	1			
1	8	28	56	70	56	28	8	1		
1	9	36	84	126	126	84	36	9	1	
1	10	45	120	210	252	210	120	45	10	1

(ก) สามเหลี่ยมปาสคาล

		k=0	k=1	k=2								
		↓	↓	↓								
n=0	→	1										
n=1	→	1	1									
n=2	→	1	2	1								
n=3	→	1	3	3	1							
n=4	→	1	4	6	4	1						
		1	5	10	10	5	1					
		1	6	15	20	15	6	1				
		1	7	21	35	35	21	7	1			
		1	8	28	56	70	56	28	8	1		
		1	9	36	84	126	126	84	36	9	1	
		1	10	45	120	210	252	210	120	45	10	1

(ข) สามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

รูปที่ 2.1 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

แต่ละแถว (n) ของสามเหลี่ยมปาสคาลสามารถสร้างได้ด้วยวิธีอื่นที่ไม่ใช่จากสมการที่ (2.8) โดยการบวกของ 2 องค์ประกอบที่อยู่ในแบบทแยงขึ้นไปในแถวข้างบน เช่น ในแถวที่ 4 (n=3) องค์ประกอบตัวที่ 3 (ที่ n=3, k=1) มาจาก 1+2 ในแถวบน (n=2) และ 3 ตัวถัดไป (ที่ n=3, k=2) มาจาก 2+1 เป็นต้น ซึ่งสามารถพิสูจน์คุณสมบัติ ดังกล่าว (การบวกเพื่อสร้างองค์ประกอบในแถวต่อไป) ได้ดังนี้

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)!}{k!(n-k)!} \quad (2.11)$$

$$= \frac{(n-k)(n-1)!}{k!(n-k)!} + \frac{k(n-1)!}{k!(n-k)!} \quad (2.12)$$

$$= \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} \quad (2.13)$$

$$= \binom{n-1}{k} + \binom{n-1}{k-1} \quad (2.14)$$

โดยสมาชิกในสามเหลี่ยมปาสคาลในแถวที่ n ใด ๆ สามารถคำนวณได้ด้วยการหาสัมประสิทธิ์ไบนอมิเยล ดังในสมการที่ (2.11) ถึง (2.14) สำหรับ k=1, 2, 3, ..., n ได้

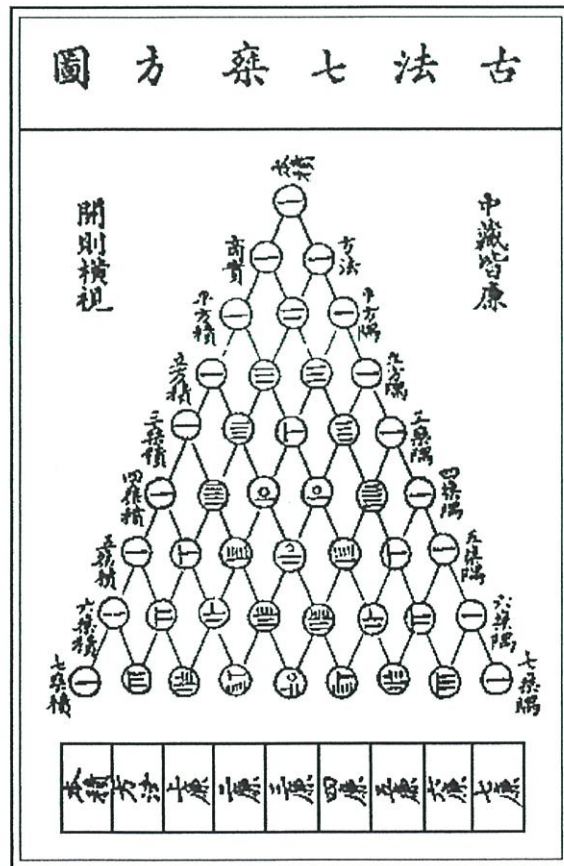
ดังนั้นเมื่อเราจะทำการกระจายเทอมของ $(x+y)^n$ ใด ๆ เราจะสามารถหาค่าของสัมประสิทธิ์ ไบโนเมียลจากสมการที่ (2.8) ได้โดยง่ายจากสามเหลี่ยมปาสคาลนั่นเอง เช่น $(x+y)^4$

$$(x+y)^4 = 1x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + 1y^4 \tag{2.15}$$

โดยสัมประสิทธิ์ในที่นี้มาจากสามเหลี่ยมปาสคาลในรูปที่ 2.1 ในแถวที่ 5 (n=4)

2.2.1 ที่มาของชื่อสามเหลี่ยมปาสคาล

ชื่อของสามเหลี่ยมปาสคาลนั้นมาจากชื่อของนักคณิตศาสตร์ชาวฝรั่งเศสชื่อ เบลส ปาสคาล (Blaise Pascal) แต่ถึงอย่างไรก็ตามรูปแบบของสามเหลี่ยมดังกล่าวเป็นที่รู้จักกันมาตั้งแต่ก่อนปี ค.ศ. 1100 ในประเทศจีนโดยนักคณิตศาสตร์ชาวจีนชื่อ ยาง ฮุย (Yang-Hui) หัวร้อยปีก่อนปาสคาล แต่ที่ เป็นผู้รู้จักกันอย่างแพร่หลายและนำมาใช้ ในการกระจายของทวินาม (expansion of the binomial) ซึ่งเรียกว่าเป็นทฤษฎีบททวินามนั้นถูกค้นพบและนำเสนอโดย ไอแซก นิวตัน (Isaac Newton) ในปี 1665



รูปที่ 2.2 แสดงสามเหลี่ยมของ ยาง ฮุย

2.2.2 การแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform)

การแปลงดิสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) ถูกนำเสนอแรก ๆ โดย โธมัส เจ กู๊ดแมน [1] ซึ่งมีรูปแบบสมการการแปลงดังนี้

$$X = P x \quad (2.16)$$

โดย X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุท

P = เมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix)

โดยการสร้างเมตริกซ์ของการแปลงปาสคาลนั้นสามารถสร้างได้ดังนี้

$$P = \begin{bmatrix} P_{ij} \end{bmatrix} = (-1)^j \binom{i}{j} = \frac{(-1)^j}{j!} i^{\underline{j}} ; i, j = 0, 1, 2, \dots, N-1 \quad (2.17)$$

โดย N คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส ส่วนฟังก์ชัน $i^{\underline{j}}$ จะเรียกว่าเป็นฟอลลิงแฟคทอเรียลพาวเวอร์ (falling factorial powers) ซึ่งมีความหมายคือ

$$i^{\underline{j}} = i(i-1)(i-2)\dots(i-j+1) ; j \geq 1 \text{ and } i^{\underline{0}} = 1 \quad (2.18)$$

และสมการที่ (2.17) นี้เองถือว่าเป็นฟังก์ชันฐาน (Basis function) ของการแปลงดิสครีตปาสคาลทรานสฟอร์มนี้ ซึ่งผลที่ได้จากการสร้างเมตริกซ์ของการแปลงปาสคาล ดังในสมการที่ (2.17) ซึ่งจะเหมือนกับการนำสมาชิกในสามเหลี่ยมปาสคาล ในรูปที่ 2.1 (ข) มาจัดวางเป็นเมตริกซ์และทำการสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับ

2.3 ฟังก์ชันฐาน (Basis function) ของปาสคาลเมตริกซ์

จากการแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform) ซึ่งมีรูปแบบสมการการแปลงดังนี้

$$X = P x \quad (2.19)$$

โดย X = เมตริกซ์ผลลัพธ์จากการแปลง
 x = เมตริกซ์ของข้อมูลอินพุท
 P = เมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix)

เราแบ่งชนิดของเมตริกซ์ของการแปลงปาสคาลออกเป็น 2 ชนิด ความถี่สูงผ่าน (High-pass) และความถี่ต่ำผ่าน (Low-pass) ซึ่งแต่ละชนิดจะขึ้นอยู่กับฟังก์ชันฐาน (Basis function) ของปาสคาลเมตริกซ์

2.3.1 ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาลทรานสฟอร์มชนิดความถี่สูงผ่าน (High-pass Type DPT)

จากสมการที่ 2.19 พิจารณาเมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix) ชนิดความถี่สูงผ่านจะได้ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาลทรานสฟอร์มชนิดความถี่สูงผ่าน (High-pass Type DPT) ดังนี้

$$P_k^{(HP)}(x) = P^{(HP)}(x, k) = \frac{(-1)^k x^{(k)}}{k!} = (-1)^k \binom{x}{k} \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (2.20)$$

โดย N คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส

$\binom{x}{k} = \frac{x!}{k!(x-k)!}$ เรียกว่า สัมประสิทธิ์ทวินาม (Binomial coefficient) ซึ่งมีความสัมพันธ์กันกับสามเหลี่ยมของปาสคาล

$P(x, k)$ คือ ตำแหน่งในแต่ละส่วนของเมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix)

x^{th} = ตำแหน่งของแถว (Row index)

k^{th} = ตำแหน่งของหลัก (Column index)

ส่วนฟังก์ชัน $x^{(k)}$ จะเรียกว่าเป็นพอลลิงแฟคทอเรียลพาวเวอร์ (falling factorial powers) ซึ่งมีความหมายคือ

$$x^{(k)} = x(x-1)(x-2)\dots(x-k+2)(x-k+1) = \frac{x!}{(x-k)!} \quad ; k \geq 1 \quad (2.21)$$

โดย $x^{(0)} = 1$, k = Order ของฟังก์ชันฐาน (Basis function)

2.3.2 ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาลทรานสฟอร์มชนิดความถี่ต่ำผ่าน (Low-pass Type DPT)

จากสมการที่ 2.19 พิจารณาเมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix) ชนิดความถี่ต่ำผ่านจะได้ฟังก์ชันฐาน (Basis function) ของการแปลงดีสครีตปาสคาลทรานสฟอร์มชนิดความถี่ต่ำผ่าน (Low-pass Type DPT) ดังนี้

$$P_k^{(LP)}(x) = P^{(LP)}(x, k) = \frac{x^{(k)}}{k!} = \binom{x}{k} \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (2.22)$$

โดย N คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส

$$\binom{x}{k} = \frac{x!}{k!(x-k)!} \text{ เรียกว่า สัมประสิทธิ์ทวินาม (Binomial coefficient) ซึ่งมี}$$

ความสัมพันธ์กับสามเหลี่ยมของปาสคาล

$$x^{th} = \text{ตำแหน่งของแถว (Row index)}$$

$$k^{th} = \text{ตำแหน่งของหลัก (Column index)}$$

ส่วนฟังก์ชัน $x^{(k)}$ จะเรียกว่าเป็นพอลลิงแฟกทอเรียลพาวเวอร์ (falling factorial powers) ซึ่งมีความหมายคือ

$$x^{(k)} = x(x-1)(x-2)\dots(x-k+2)(x-k+1) = \frac{x!}{(x-k)!} \quad ; k \geq 1 \quad (2.23)$$

$$\text{โดย } x^{(0)} = 1$$

$$k = \text{Order ของฟังก์ชันฐาน (Basis function)}$$

พิจารณาจากสมการที่ 2.20 และ 2.21 จะสังเกตเห็นว่าความแตกต่างของทั้งสองสมการมีเพียงอย่างเดียวคือในเทอมของ $(-1)^k$ ไม่มีปรากฏให้เห็นในสมการที่ 2.21 ซึ่งผลจากเทอมนี้จะทำให้แต่ละแถวของเมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix) ชนิดความถี่สูงผ่านมีเครื่องหมายบวกและลบสลับกัน

2.4 การแยกปาสคาลเมตริกซ์เป็นเมตริกซ์องค์ประกอบเลขฐานสอง

1. พิจารณาแนวเส้นทแยงมุมหลัก

$$P = \begin{bmatrix} q_{ij} \end{bmatrix}_{N-1} \begin{bmatrix} q_{ij} \end{bmatrix}_{N-2} \cdots \begin{bmatrix} q_{ij} \end{bmatrix}_1 \quad (2.24)$$

เรียก $[q_{ij}]_k$ ว่าเมตริกซ์ $[q_{ij}]$ แสดงที่ k^{th}
แสดงที่ $k = (N-1)$

$$\text{Diag}([q_{ij}]_{N-1}) = [1 \ 1 \ 1 \ \dots \ 1 \ -1] \quad (2.25)$$

โดยที่จะมี '1' เท่ากับจำนวน $N-1$

เช่น

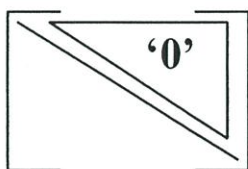
แสดงที่ $k=2$

$$\text{Diag}([q_{ij}]_2) = [1 \ 1 \ -1 \ -1 \ \dots \ -1] \quad (2.26)$$

แสดงที่ $k=1$

$$\text{Diag}([q_{ij}]_1) = [1 \ -1 \ -1 \ -1 \ \dots \ -1] \quad (2.27)$$

2. ทูกองค์ประกอบเหนือเส้นทแยงมุมหลัก (Upper Triangle) หรือส่วนบนของสามเหลี่ยมมีค่าเท่ากับ '0'



รูปที่ 2.3 แสดงตำแหน่งองค์ประกอบ '0' ในส่วนของสามเหลี่ยมบนของแต่ละเมตริกซ์ $[q_{ij}]_k$

3. พิจารณาสำหรับแต่ละแถว, ในแถวใดที่มีค่า '-1' เป็นองค์ประกอบอยู่ ให้องค์ประกอบในหลัก (column) ก่อนหน้าของ '-1' ตัวนั้นมีค่าเท่ากับ '1' หรืออาจกล่าวได้ว่า สำหรับแต่ละ i แล้ว ถ้า $q_{ij} = '-1'$ ดังนั้น $q_{i,j-1} = '1'$

4. นอกเหนือจากนี้ให้มีค่าเท่ากับ '0'

ตัวอย่าง $N = 3$ จะได้

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{array}{l} X_0 = x_0 \\ X_1 = x_0 - x_1 \\ X_2 = x_0 - 2x_1 + x_2 \end{array}$$

สามารถจัดรูปองค์ประกอบได้ดังนี้

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

Stage 2 *Stage 1*

ตัวอย่าง $N = 4$ จะได้

$$\mathbf{P} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

2.4 บทสรุป

เนื้อหาในบทนี้ได้บรรยายเกี่ยวกับความเป็นมาของสามเหลี่ยมปาสคาล (Pascal's Triangle) ทฤษฎีบททวินาม (Binomial Theorem) แสดงให้เห็นถึงการแปลงดิสครีตปาสคาลทรานสฟอร์ม (DPT : Discrete Pascal Transform) และการหาฟังก์ชันฐาน (Basis function) ของปาสคาลเมตริกซ์ทั้งสองแบบ และยังสรุปวิธีการแยกปาสคาลเมตริกซ์เป็นเมตริกซ์องค์ประกอบเลขฐานสอง ซึ่งเป็นขั้นตอนที่สำคัญสำหรับการออกแบบโครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพซึ่งจะอธิบายในบทถัดไป

บทที่ 3

การออกแบบโครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงปาสคาลเต็มหน่วย

การออกแบบโครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงปาสคาลเต็มหน่วยเมื่อเปรียบเทียบกับหลักการที่มีอยู่เดิม ใน [1-5] เป็นการศึกษาเกี่ยวกับปาสคาลเมตริกซ์เช่นกันแต่หน้าตาของปาสคาลเมตริกซ์ในกรณีเหล่านี้จะแตกต่างจากหน้าตาของปาสคาลเมตริกซ์ที่นำเสนอในวิทยานิพนธ์นี้ เป้าหมายของปาสคาลเมตริกซ์ที่นำมาใน [1-5] มีวัตถุประสงค์หลักในการแปลงตัวแปรระหว่าง s กับ z เพื่อแปลงความสัมพันธ์ระหว่างฟังก์ชันถ่ายโอนใน Analog Domain กับ Digital Domain โดยหน้าตาของปาสคาลเมตริกซ์แต่ละตัวนั้นก็จะมีหน้าตาแตกต่างกันไปตามแต่ความสัมพันธ์ที่จะใช้ในการแปลงตัวแปร เช่น Bilinear Transform, Backward-Difference Transform, Forward-Difference Transform เป็นต้น แต่ถึงกระนั้นใน [1-5] ก็ได้ให้ถึงนิยามและที่มา รวมทั้งความสัมพันธ์ในการสร้างปาสคาลเมตริกซ์แบบต่างๆ รวมถึงความเกี่ยวเนื่องไปยังทฤษฎีบททวินาม

สำหรับในวิทยานิพนธ์นี้ตัวปาสคาลเมตริกซ์จะมีรูปแบบการสร้างที่ง่ายและสมาชิกภายในเมตริกซ์เกี่ยวข้องกับสมาชิกในสามเหลี่ยมปาสคาลโดยตรงจะมีการปรับในส่วนเรื่องของเครื่องหมายเล็กน้อย ซึ่งหน้าตาของปาสคาลเมตริกซ์ที่ใช้ในการแปลงสัญญาณเหมือนกับที่นำเสนอใน [6] ใน [6] ดูเหมือนจะเป็นบทความแรกๆ ที่นำเสนอในเรื่องของการแปลงปาสคาลเต็มหน่วยอย่างจริงจังและเป็นแรงบันดาลใจสำหรับวิทยานิพนธ์นี้ เนื้อหาใน [6] เสมือนเป็น Overview สำหรับการแปลง DPT นี้แต่จะขาดไปในส่วนของ details ที่การแปลงชนิดนี้มีผลอย่างไรต่อสัญญาณจริงๆ ดังนั้นในส่วนของงานวิจัยนี้จึงทำการทดสอบแปลงสัญญาณจริงทั้งในกรณีสัญญาณ 1 มิติ และ สัญญาณ 2 มิติ แล้วสังเกตดูถึงผลที่ได้จากการแปลง ซึ่งนำมาถึงสมมติฐานการผ่านความถี่สูงของการแปลงชนิดนี้ และนำไปสู่การหาฟังก์ชันถ่ายโอนของระบบที่ทำหน้าที่แปลงสัญญาณด้วยกรรมวิธีดังกล่าว ซึ่งส่วนที่กล่าวมานี้ยังมีได้มีปรากฏเป็นหลักการมาก่อนหน้าใน [7] เป็นการนำพาไปสู่การ factorization ให้เมตริกซ์ที่ใช้ในการแปลงสัญญาณแตกออกมาอยู่ในรูปของเมตริกซ์เลขฐานสอง (Binary Matrices) โดยใน [7] เป็นการอธิบายถึง concept การแปลงพหุนามเต็มหน่วย (Discrete Polynomials Transform) หลายๆ ตัวประกอบกันและเน้นไปที่การ factorize สำหรับ Hermite Transform เป็นหลัก ส่วนใน [8] แสดงขั้นตอนทางคณิตศาสตร์ในการ factorize ให้ Pascal Matrix แยกออกมาเป็น Binary Matrices สำหรับการนำมาจัดฟอร์มเป็น Butterfly Unit ดังใน [9] ส่วนในงานวิจัยนี้จะผนวกรวมทั้งหมดเข้าด้วยกันและอธิบายวิธีการ

Factorize เป็น Algorithm ที่ง่ายแก่ความเข้าใจกว่าที่ผ่านมาและนำโครงสร้างที่ได้ทางฮาร์ดแวร์นั้นไปใช้เป็นโครงสร้างของสิ่งที่เรียกว่า Pascal Filter อีกด้วย

3.1 การแปลงปาสคาลเต็มหน่วย

การแปลงดิสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) ถูกนำเสนอแรก ๆ โดย โธมัส เจ กู๊ดแมน [1] ซึ่งมีรูปแบบสมการการแปลงดังนี้

$$X = P x \quad (3.1)$$

โดย X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุต

P = เมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix)

โดยการสร้างเมตริกซ์ของการแปลงปาสคาลนั้นสามารถสร้างได้ดังนี้

$$P = \left[P_{ij} \right] = (-1)^j \binom{i}{j} = \frac{(-1)^j}{j!} i^{\underline{j}} ; i, j = 0, 1, 2, \dots, N-1 \quad (3.2)$$

โดย N คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส ส่วนฟังก์ชัน $i^{\underline{j}}$ จะเรียกว่าเป็นฟอลลิงแฟกทอเรียลพาวเวอร์ (falling factorial powers) ซึ่งมีความหมายคือ

$$i^{\underline{j}} = i(i-1)(i-2)\dots(i-j+1) ; j \geq 1 \text{ and } i^{\underline{0}} = 1 \quad (3.3)$$

และสมการที่ (2.13) นี้เองถือว่าเป็นฟังก์ชันพื้นฐาน (Basis function) ของการแปลงดิสครีตปาสคาลทรานสฟอร์มนี้ ซึ่งผลที่ได้จากการสร้างเมตริกซ์ของการแปลงปาสคาล ดังในสมการที่ (2.13) ซึ่งจะเหมือนกับการนำสมาชิกในสามเหลี่ยมปาสคาล ในรูปที่ 2.20 (ข) มาจัดวางเป็นเมตริกซ์และทำการสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับดังที่ได้กล่าวมาแล้วแต่สำหรับในการออกแบบโครงสร้างนั้นเราแบ่งชนิดของเมตริกซ์ของการแปลงปาสคาลออกเป็น 2 ชนิด ความถี่สูงผ่าน (High-pass) และ ความถี่ต่ำผ่าน (Low-pass) ซึ่งแต่ละชนิดจะขึ้นอยู่กับฟังก์ชันพื้นฐาน (Basis function) ของปาสคาลเมตริกซ์

ตัวอย่าง เราจะแสดง 3rd order ฟังก์ชันฐาน (Basis function) ของปาสคาลเมตริกซ์

ในกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) จะมี 4 โพลีโนเมียล(polynomials) ดังนี้

$$\begin{aligned} P_0^{(HP)}(x) &= 1 & (3.4) \\ P_1^{(HP)}(x) &= -x \\ P_2^{(HP)}(x) &= \frac{1}{2}x(x-1) = -\frac{1}{2}(x-1)P_1^{(HP)}(x) \\ P_3^{(HP)}(x) &= -\frac{1}{6}x(x-1)(x-2) = -\frac{1}{3}(x-2)P_2^{(HP)}(x) \end{aligned}$$

จากสมการที่ 3.4 เราสามารถสรุปสูตรได้ดังนี้

$$P_{k+1}^{(HP)}(x) = -\frac{1}{k+1}(x-k)P_k^{(HP)}(x) \quad (3.5)$$

ดังนั้นเราสามารถสร้างเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) ได้ดังนี้

$$\mathbf{P}^{(HP)} = [P^{(HP)}(x, k)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

ในกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) จะมี 4 โพลีโนเมียล(polynomials) ดังนี้

$$\begin{aligned} P_0^{(LP)}(x) &= 1 & (3.6) \\ P_1^{(LP)}(x) &= x \\ P_2^{(LP)}(x) &= \frac{1}{2}x(x-1) = \frac{1}{2}(x-1)P_1^{(LP)}(x) \\ P_3^{(LP)}(x) &= \frac{1}{6}x(x-1)(x-2) = \frac{1}{3}(x-2)P_2^{(LP)}(x) \end{aligned}$$

จากสมการที่ 3.6 เราสามารถสรุปสูตรได้ดังนี้

$$P_{k+1}^{(LP)}(x) = \frac{1}{k+1}(x-k)P_k^{(LP)}(x) \quad (3.7)$$

ดังนั้นเราสามารถสร้างเมทริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) ได้ดังนี้

$$\mathbf{P}^{(LP)} = [P^{(LP)}(x, k)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{bmatrix}$$

3.1.1 การสร้างเมทริกซ์การแปลงปาสคาลขนาดต่าง ๆ

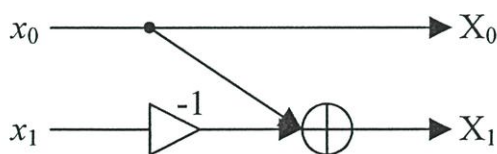
ในหัวข้อนี้เราจะเสนอตัวอย่างที่ $N=2$ และ $N=3$ ใน และพิจารณาในกรณีของเมทริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass) จะเห็นว่าองค์ประกอบในแต่ละตำแหน่งของเมทริกซ์จะสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับ

เมื่อ $N=2$ จะได้

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}$$

$$\text{ดังนั้น, } \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 - x_1 \end{aligned}$$

สามารถสร้างบัตเตอร์ฟลายยูนิท (butterfly unit) อย่างง่ายได้ดังนี้



รูปที่ 3.1 แสดงการสร้างบัตเตอร์ฟลายยูนิทสำหรับการแปลงปาสคาลเต็มหน่วย

เมื่อ $N=3$ จะได้

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

$$\text{ดังนั้น} \quad \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{array}{l} X_0 = x_0 \\ X_1 = x_0 - x_1 \\ X_2 = x_0 - 2x_1 + x_2 \end{array}$$

ในการสร้างบัตเตอร์ฟลายยูนิตสำหรับ $N = 3$ นี้แน่นอนว่าต้องมีส่วนประกอบของบัตเตอร์ฟลายยูนิต ดังในกรณีที่ $N = 2$ แต่ในทางปฏิบัติแล้ว เราใช้เมตริกซ์ที่มีองค์ประกอบในลักษณะเป็นไบนารีเมตริกซ์ $(1,0,-1)$ ดังที่เห็นในกรณี $N = 2$ ซึ่งเมตริกซ์ \mathbf{P} คือ เมตริกซ์ไบนารี $(1,0,-1)$ เราจะสามารถสร้าง บัตเตอร์ฟลายยูนิตจากไบนารีเมตริกซ์ได้โดยง่าย ทั้งนี้เราสามารถจัดปาสคาลเมตริกซ์ต่าง ๆ ให้อยู่ในรูปไบนารีเมตริกซ์ $(1,0,-1)$ ได้โดยใช้ผลคูณของไบนารีเมตริกซ์ ดังสมการต่อไปนี้

$$\mathbf{P} = \prod_{k=N-1}^1 [q_{ij}]_k \quad ; i, j = 0, 1, 2, \dots, N-1 \quad (3.8)$$

โดยเราเรียก $[q_{ij}]_k$ ว่าเมตริกซ์ $[q_{ij}]$ แสดงที่ k^{th}

3.2 คุณสมบัติของการแปลงปาสคาลเต็มหน่วย

คุณสมบัติของการแปลงปาสคาลเต็มหน่วยสามารถสรุปได้ดังนี้

1. ค่าในทศตำแหน่งของแถวแรกมีค่าเท่ากับ 1
2. ทุกเมตริกซ์จะเป็นเมตริกซ์แบบสามเหลี่ยมล่าง (lower triangular matrix)
3. สำหรับกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) ผลรวมของตำแหน่งในแต่ละหลัก(ยกเว้นหลักแรก)จะมีค่าเท่ากับศูนย์ สำหรับกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) ผลรวมของตำแหน่งในแต่ละหลักจะมีค่าเท่ากับ 2^x โดยที่ x^{th} คือตำแหน่งของหลัก
4. สำหรับกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type)

$$[\mathbf{P}^{(\text{HP})}]^{-1} = \mathbf{P}^{(\text{HP})}$$

สำหรับกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type)

$$[\mathbf{P}^{(LP)}]^{-1} = (-1)^{x+k} \mathbf{P}^{(LP)}$$

คุณสมบัติที่ตีประการหนึ่งของการแปลงปาสคาลเต็มหน่วยนี้คือ

$$P^{-1} = P \quad (3.9)$$

ดังนั้นทำให้การแปลงกลับคือสคริตปาสคาลทรานสฟอร์ม (Inverse DPT) สามารถทำได้เหมือนทำการแปลงฟอร์เวิร์ดคือสคริตปาสคาลทรานสฟอร์ม (Forward DPT) ทุกประการ ถ้ามองในลักษณะของการออกแบบฮาร์ดแวร์นั้นจะเห็นได้ว่าสามารถใช้วงจรเดิมที่ใช้ตอนแปลงมาสำหรับการแปลงกลับได้เช่นเดียวกัน ไม่ต้องทำการปรับเปลี่ยนวงจรแต่อย่างใด

$$x = P^{-1}X = PX \quad (3.10)$$

3.3 โครงสร้างของวงจรแปลงปาสคาลเต็มหน่วยที่ได้จากการแยกองค์ประกอบเมตริกซ์

การแปลงสคริตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) ถูกนำเสนอแรก ๆ โดย โทมัส เจ กู๊ดแมน [1] เสนอในรูปแบบพื้นฐานเท่านั้น ไม่ได้สนใจในการออกแบบเป็นฮาร์ดแวร์ จาก [2] ซึ่งนำเสนอในการออกแบบฮาร์ดแวร์แต่วิธีการในการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปของไบนารีเมตริกซ์ (1,0,-1) ยังไม่อยู่ในรูปแบบที่รัดกุม ซึ่งวิธีการที่รัดกุมถูกนำเสนอใน [4] ซึ่งทำให้โครงสร้างในการออกแบบเป็นฮาร์ดแวร์มีประสิทธิภาพมาก ในหัวข้อนี้จะแสดงวิธีการแยกองค์ประกอบเมตริกซ์ทั้งชนิดความถี่สูงผ่าน (High-pass type) และ ชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปของไบนารีเมตริกซ์ (1,0,-1) โดยใช้ อิลิมีเนชัน เมตริกซ์ (elimination matrix)ซึ่งรายละเอียดนี้ไม่ได้กล่าวถึงใน [4]

3.3.1 การแปลงเมตริกซ์ของการแปลงปาสคาลให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1)

พิจารณาเมตริกซ์ของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) $\mathbf{P}^{(HP)}$ ที่ขนาด $N=2$ แสดงด้วย เมตริกซ์ ดังนี้

$$\mathbf{P}_2^{(HP)} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \quad (3.11)$$

จากสมการการแปลงดีสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) 2 จุดสามารถแสดงได้ดังนี้

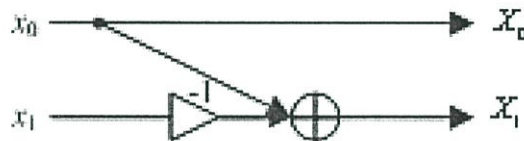
$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad (3.12)$$

จะได้

$$X_0 = x_0 \quad (3.13)$$

$$X_1 = x_0 - x_1 \quad (3.14)$$

จากสมการ 3.13 และ 3.15 สามารถเขียน โครงสร้างทางฮาร์ดแวร์ได้ดังนี้



รูปที่ 3.2 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วยชนิดความถี่สูงผ่าน (High-pass type)

จากรูปที่ 3.2 โครงสร้างที่เขียนได้จากสมการที่ 3.13 และ 3.14 จะสังเกตเห็นว่ามีการคูณค่าคงที่ -1 ซึ่งในการออกแบบฮาร์ดแวร์เราจะออกแบบตัวลบบมาแทนที่ ดังนั้นในโครงนี้จึงไม่มีการใช้ตัวคูณแต่อย่างใดในโครงสร้างนี้ ซึ่งเหตุที่เขียนรูปโครงสร้างดังกล่าวเพราะว่าต้องการแสดงให้เห็นถึงความแตกต่างของการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วยชนิดความถี่ต่ำผ่าน (Low-pass type) ซึ่งการสร้างบัตเตอร์ฟลายยูนิตทั้งแบบต่างกันเพียงว่าใช้ตัวบวก หรือ ตัวลบในโครงเท่านั้นเอง ในที่นี้ทั้งตัวบวกและตัวลบจัดอยู่ในประเภทเดียวกัน

พิจารณาเมตริกซ์ของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) $P^{(LP)}$ ที่ขนาด $N = 2$ แสดงด้วย เมตริกซ์ ดังนี้

$$P_2^{(LP)} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (3.15)$$

จากสมการการแปลงดีสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) 2 จุดสามารถแสดงได้ดังนี้

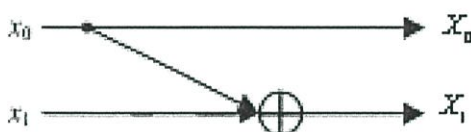
$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \quad (3.16)$$

จะได้

$$X_0 = x_0 \quad (3.17)$$

$$X_1 = x_0 + x_1 \quad (3.18)$$

จากสมการ 3.17 และ 3.18 สามารถเขียน โครงสร้างทางฮาร์ดแวร์ได้ดังนี้



รูปที่ 3.3 แสดงการสร้างบิตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วยชนิดความถี่ต่ำผ่าน (Low-pass type)

บิตเตอร์ฟลายยูนิตที่แสดงในรูป 3.2 และ 3.3 เป็นส่วนประกอบพื้นฐานในการเขียน โครงสร้างทางฮาร์ดแวร์ในการแปลงดีสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform) แบบ N จุด

3.3.2 วิธีการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) โดยใช้เกาเซียนอิลิมีเนชัน (Gaussian elimination)

การแปลงดีสครีตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform)แบบความถี่ต่ำผ่าน (Low-pass type)

อิลิมีเนชัน เมตริกซ์แบบความถี่ต่ำผ่าน $E^{(LP)}$ โดยที่ $E_{xx}^{(LP)} = 1$ และ $E_{x-1}^{(LP)} = -1$ จาก [5]

กำหนด $P_4^{(LP)}$ สำหรับพิจารณา

ขั้นตอนที่ 1 สำหรับ อิลิมีเนชัน เมตริกซ์

$$\mathbf{E}_4^{(LP)}\mathbf{P}_4^{(LP)} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_3^{(LP)} \end{bmatrix} \quad (3.19)$$

ขั้นตอนที่ 2

$$\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_3^{(LP)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{P}_2^{(LP)} \end{bmatrix} \quad (3.20)$$

ขั้นตอนที่ 3

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{P}_2^{(LP)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [\mathbf{I}] \quad (3.21)$$

จากสมการที่ 3.21 สามารถสรุปได้ดังนี้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix} \begin{bmatrix} \mathbf{E}_4^{(LP)} \\ \mathbf{P}_4^{(LP)} \end{bmatrix} = [\mathbf{I}] \quad (3.22)$$

เพราะฉะนั้น $\mathbf{P}_4^{(LP)}$ จะมีค่าดังนี้

$$\therefore \mathbf{P}_4^{(LP)} = [\mathbf{E}_4^{(LP)}]^{-1} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix}^{-1} \quad (3.23)$$

ดังนั้น เราสามารถแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) $\mathbf{P}_4^{(LP)}$ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้ดังนี้

$$\mathbf{P}_4^{(LP)} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & 1 & 1 & \\ & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (3.24)$$

จากวิธีการที่กล่าวมาแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) จะเหมือนกับ [2] ซึ่งยังไม่อยู่ในรูปแบบที่รัดกุมของไบนารีเมตริกซ์ (1,0,-1) เราจึงใช้คุณสมบัติของคุณสมบัติของการแปลงปาสคาลเต็มหน่วย สำหรับกรณีของเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) $[\mathbf{P}^{(LP)}]^{-1} = (-1)^{x+k} \mathbf{P}^{(LP)}$

จากสมการที่ 3.23 จะได้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix} [\mathbf{E}_4^{(LP)}] = (-1)^{x+k} \mathbf{P}_4^{(LP)} \quad (3.25)$$

จากสมการ 3.25 จัดรูปใหม่จะได้

$$\mathbf{P}_4^{(LP)} = (-1)^{x+k} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix} \times (-1)^{x+k} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix} \times (-1)^{x+k} [\mathbf{E}_4^{(LP)}] \quad (3.26)$$

สุดท้าย เราสามารถแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) $\mathbf{P}_4^{(LP)}$ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้ดังนี้ซึ่งเป็นรูปแบบที่รัดกุม

$$\mathbf{P}_4^{(LP)} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{matrix} \text{Stage3} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \end{matrix} \begin{matrix} \text{Stage2} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \end{matrix} \begin{matrix} \text{Stage1} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \right] \end{matrix} \quad (3.27)$$

จากสมการที่ 3.27 เป็นรูปแบบที่รัดกุมของการแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) $\mathbf{P}_4^{(LP)}$ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1)

การแปลงคีสคริตปาสคาลทรานสฟอร์ม (DPT :discrete Pascal Transform)แบบความถี่สูงผ่าน (High-pass type)

อิลิมิเนชัน เมตริกซ์แบบความถี่ต่ำผ่าน $\mathbf{E}^{(HP)}$ โดยที่ $E_{xx}^{(HP)} = -1$ ยกเว้น $E_{00}^{(HP)} = 1$ และ $E_{xx-1}^{(HP)} = 1$ กำหนด $\mathbf{P}_4^{(HP)}$ สำหรับพิจารณา

ขั้นตอนที่ 1 สำหรับ อิลิมิเนชัน เมตริกซ์

$$\mathbf{E}_4^{(HP)}\mathbf{P}_4^{(HP)} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} \quad (3.28)$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_3^{(LP)} \end{bmatrix}$$

ขั้นตอนที่ 2

$$\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_3^{(HP)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (3.29)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{P}_2^{(HP)} \end{bmatrix}$$

ขั้นตอนที่ 3

$$\begin{aligned} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{P}_2^{(HP)} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [\mathbf{I}] \end{aligned} \quad (3.30)$$

จากสมการที่ 3.30 สามารถสรุปได้ดังนี้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(HP)} \end{bmatrix} [\mathbf{E}_4^{(HP)}] [\mathbf{P}_4^{(HP)}] = [\mathbf{I}] \quad (3.31)$$

เพราะฉะนั้น $\mathbf{P}_4^{(HP)}$ จะมีค่าดังนี้

$$\therefore \mathbf{P}_4^{(LP)} = [\mathbf{E}_4^{(LP)}]^{-1} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(LP)} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(LP)} \end{bmatrix}^{-1} \quad (3.32)$$

ดังนั้น เราสามารถแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) $\mathbf{P}_4^{(HP)}$ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้ดังนี้

$$\mathbf{P}_4^{(HP)} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -1 & -1 & \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & 1 & -1 & \\ & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 & -1 \end{bmatrix} \quad (3.33)$$

จากวิธีการที่กล่าวมาแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) จะเหมือนกับ [2] ซึ่งยังไม่อยู่ในรูปแบบที่รัดกุมของไบ

นารีเมตริกซ์ (1,0,-1) เราจึงใช้คุณสมบัติของคุณสมบัติของการแปลงฟูรีเยอร์เต็มหน่วย สำหรับกรณีของเมตริกซ์ของการแปลงฟูรีเยอร์ชนิดความถี่สูงผ่าน (High-pass type) $[\mathbf{P}^{(HP)}]^{-1} = \mathbf{P}^{(HP)}$ จากสมการที่ 3.32 จะได้

$$\mathbf{P}_4^{(HP)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{E}_2^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{E}_3^{(HP)} \end{bmatrix} [\mathbf{E}_4^{(HP)}] \quad (3.34)$$

สุดท้าย เราสามารถแยกองค์ประกอบเมตริกซ์ของการแปลงฟูรีเยอร์ชนิดความถี่สูงผ่าน (High-pass type) $\mathbf{P}_4^{(HP)}$ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้ดังนี้ซึ่งเป็นรูปแบบที่รัดกุม

$$\mathbf{P}_4^{(HP)} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & -1 \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & -1 \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & -1 \\ & & & 1 & -1 \end{bmatrix} \quad (3.35)$$

จากวิธีการทำค่ามาสามารถใช้ในการแยกองค์ประกอบเมตริกซ์ของการแปลงฟูรีเยอร์ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้ทุกขนาดของเมตริกซ์ และได้ทั้งสองชนิด

3.3.3 ขั้นตอนการสร้างไบนารีเมตริกซ์ (1,0,-1)

จากวิธีการที่ได้กล่าวมาข้างต้น เราสามารถแยกองค์ประกอบเมตริกซ์ของการแปลงฟูรีเยอร์ชนิดความถี่สูงผ่าน (High-pass type) และชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1)

พิจารณาเมตริกซ์ขนาด $N=4$ หรือ \mathbf{P}_4 สามารถแยกองค์ประกอบเมตริกซ์ของการแปลงฟูรีเยอร์เป็นสามไบนารีเมตริกซ์ เราสามารถสรุปเป็นรูปแบบพื้นฐานได้ดังนี้

$$\mathbf{P} = \prod_{l=N-1}^1 [q_{xk}]_l \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (3.36)$$

$$= [q_{xk}]_{N-1} [q_{xk}]_{N-2} \cdots [q_{xk}]_1$$

โดยที่ $[q_{xk}]_l$ คือ ไบนารีเมตริกซ์ สเตจที่ l^{th}

จากสมการ 3.27 สามารถเขียนให้อยู่ในรูปแบบพื้นฐานได้ดังนี้

$$\mathbf{P}_4^{(LP)} = [q_{xk}^{(LP)}]_3 [q_{xk}^{(LP)}]_2 [q_{xk}^{(LP)}]_1 \quad (3.37)$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

จากสมการ 3.35 สามารถเขียนให้อยู่ในรูปแบบพื้นฐานได้ดังนี้

$$\mathbf{P}_4^{(HP)} = [q_{xk}^{(HP)}]_3 [q_{xk}^{(HP)}]_2 [q_{xk}^{(HP)}]_1 \quad (3.38)$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & -1 \\ & & 1 & -1 \\ & & & 1 & -1 \end{bmatrix}$$

จากวิธีการแยกองค์ประกอบเมตริกซ์ของการแปลงปาสคาลชนิดความถี่สูงผ่าน (High-pass type) และชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) เรานำเสนอขั้นตอนการสร้างไบนารีเมตริกซ์ (1,0,-1)

ขั้นตอนการสร้างไบนารีเมตริกซ์ (1,0,-1) มีดังต่อไปนี้

สำหรับชนิดความถี่ต่ำผ่าน (Low-pass type)

1. เส้นทแยงมุมของทุกสแตจมีค่าเท่ากับ 1
2. สามเหลี่ยมด้านบนของทุกสแตจมีค่าเท่ากับศูนย์



3. ที่สแตจ l^{th} พิจารณาตำแหน่งของหลัก (rowth) $x = N-1, N-2, \dots, l$ โดย

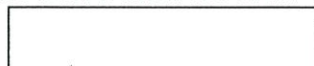
for each $x \Rightarrow$ if $q_{xx} = '1'$ then $q_{xx-1} = '1'$

4. ตำแหน่งที่เหลือที่ค่าเท่ากับศูนย์

สำหรับชนิดความถี่สูงผ่าน (High-pass type)

1. พิจารณาเส้นทแยงมุม

ที่สแตจ l^{th} พิจารณา ตำแหน่งของหลัก (rowth) $x = N-1, N-2, \dots, l$ โดย



$$q_{xx} = '-1', \text{ other are '1'}$$

ตัวอย่าง ที่สแดง $l = N - 1$

$$Diag([q_{xx}]_{N-1}) = [1 \ 1 \ 1 \ \dots \ 1 \ -1] \text{ จะมี 1 อยู่ } N-1 \text{ ตำแหน่ง ที่เหลือเป็น -1}$$

ที่สแดง $l = 2$

$$Diag([q_{xx}]_2) = [1 \ 1 \ -1 \ -1 \ \dots \ -1] \text{ จะมี 1 อยู่ 2 ตำแหน่ง ที่เหลือเป็น -1}$$

ที่สแดง $l = 2$

$$Diag([q_{xx}]_1) = [1 \ -1 \ -1 \ -1 \ \dots \ -1] \text{ จะมี 1 อยู่ 1 ตำแหน่ง ที่เหลือเป็น -1}$$

2. สามเหลี่ยมด้านบนของทุกสแดงมีค่าเท่ากับศูนย์



3. ที่สแดง l^{th} พิจารณาตำแหน่งของหลัก (rowth)

$$\boxed{\text{if } q_{xx} = '-1' \text{ then } q_{xx-1} = '1'}$$

4. ตำแหน่งที่ลือที่ค่าเท่ากับศูนย์

3.3.4 โครงสร้างของวงจรแปลงดิจิตอลพาสคาลทรานสฟอร์ม (DPT)

โครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงพาสคาลเต็มหน่วย สร้างจากแยกองค์ประกอบเมตริกซ์ของการแปลงพาสคาลชนิดความถี่สูงผ่าน (High-pass type) และชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ซึ่งในโครงสร้างประกอบด้วยบัตเตอร์ฟลายยูนิตจำนวนมาก โดยถือว่าสร้างบัตเตอร์ฟลายยูนิตเป็นส่วนประกอบพื้นฐานของการแปลงพาสคาลทรานสฟอร์ม

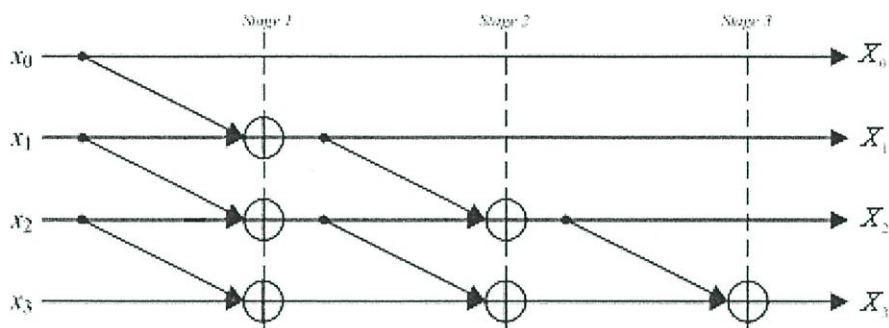
โครงสร้างการแปลงพาสคาลชนิดความถี่ต่ำผ่าน (Low-pass type) แบบ 4 จุดแสดงดังนี้

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 + x_1 \\ X_2 &= x_0 + 2x_1 + x_2 \\ X_3 &= x_0 + 3x_1 + 3x_2 + x_3 \end{aligned} \tag{3.39}$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.40}$$

Stage3 Stage2 Stage1

ดังนั้น สามารถเขียน โครงสร้างการแปลงฟูรีเยร์ความถี่ต่ำผ่าน(Low-pass type)แบบ 4 จุด ดังนี้



รูปที่ 3.4 แสดงโครงสร้างการแปลงฟูรีเยร์ความถี่ต่ำผ่าน (Low-pass type)แบบ 4 จุด

จากรูปที่ 3.4 เราใช้ 6 บิตเตอร์ฟลายยูนิตสำหรับการแปลงฟูรีเยร์ความถี่ต่ำผ่าน (Low-pass type) สำหรับโครงสร้างการแปลงฟูรีเยร์ความถี่ต่ำผ่าน (Low-pass type) แบบ 4 จุด จากโครงสร้างนี้เป็นโครงสร้างที่เหมาะสมและง่ายสำหรับสร้างเป็นฮาร์ดแวร์ โดยเฉพาะอย่างยิ่งในการออกแบบวงจรรวม (VLSI design) โครงสร้างนี้ เหมาะสำหรับการใช้เทคนิคที่เรียกว่า ไพป์ไลน์ (Pipeline technique) โดยการวางไฟไลน์ รีจิสเตอร์ (Pipeline register) ในแต่ละสเตจ เพื่อช่วยในเรื่องของความเร็วในการแปลงฟูรีเยร์ความถี่ต่ำผ่าน (DFT)

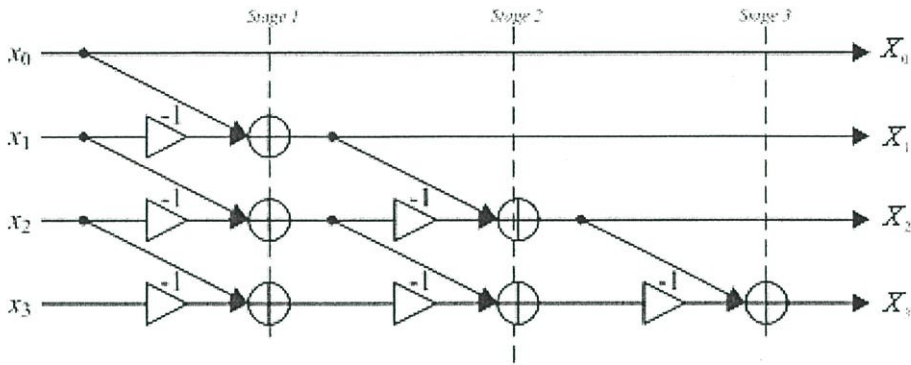
โครงสร้างการแปลงฟูรีเยร์ความถี่สูงผ่าน(High-pass type)แบบ 4 จุดแสดงดังนี้

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 - x_1 \\ X_2 &= x_0 - 2x_1 + x_2 \\ X_3 &= x_0 - 3x_1 + 3x_2 - x_3 \end{aligned} \quad (3.41)$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.42)$$

Stage3 Stage2 Stage1

ดังนั้น สามารถเขียน โครงสร้างการแปลงฟูรีเยร์ความถี่ต่ำผ่าน(Low-pass type)แบบ 4 จุด ดังนี้



รูปที่ 3.5 แสดงโครงสร้างการแปลงฟูรีเยร์ความถี่สูงผ่าน (highpass type)แบบ 4 จุด

จากรูปที่ 3.5 เราใช้ 6 บิตเตอร์ฟลายยูนิตสำหรับการแปลงฟูรีเยร์ความถี่สูงผ่าน (High-pass type) สำหรับโครงสร้างการแปลงฟูรีเยร์ความถี่สูงผ่าน (High-pass type) แบบ 4 จุด จากโครงสร้างนี้เป็นโครงสร้างที่เหมาะสมและง่ายสำหรับสร้างเป็นฮาร์ดแวร์ โดยเฉพาะอย่างยิ่งในการออกแบบวงจรรวม(VLSI design) โครงสร้างนี้ เหมาะสำหรับการใช้เทคนิคที่เรียกว่าไพพ์ไลน์ (Pipeline technique) โดยการวางไพพ์ไลน์ รีจิสเตอร์ (Pipeline register) ในแต่ละสเตจ เพื่อช่วยในเรื่องของความเร็วในการแปลงดีสครีตฟูรีเยร์ทรานสฟอร์ม (DFT)

3.4 การแปลงฟูรีเยร์ความถี่สูงผ่านในสองมิติ (2-D Discrete Pascal Transform)

สำหรับการแปลงฟูรีเยร์ความถี่สูงผ่านแบบ 2 มิติ (2-D DFT) สามารถทำการแปลงได้ดังนี้

$$X = PxP^T \quad (3.43)$$

เมื่อ X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุต

P = เมตริกซ์ของการแปลงฟูรีเยร์

และกรณีแปลงกลับ (Inverse Transforms) สามารถแสดงได้ดังสมการที่ (3.43)

$$x = PXP^T \quad (3.44)$$

เพื่อความง่ายและสะดวกในการอธิบายหลักการแปลงปาสคาลเต็มหน่วยในสองมิติ ให้พิจารณาตัวอย่างเมื่อ : $N = 3$

$$\begin{aligned}
 X &= PX P^T \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \\
 \therefore \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \\
 \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} &= \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{0,0} - x_{1,0} & x_{0,1} - x_{1,1} & x_{0,2} - x_{1,2} \\ x_{0,0} - 2x_{1,0} + x_{2,0} & x_{0,1} - 2x_{1,1} + x_{2,1} & x_{0,2} - 2x_{1,2} + x_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

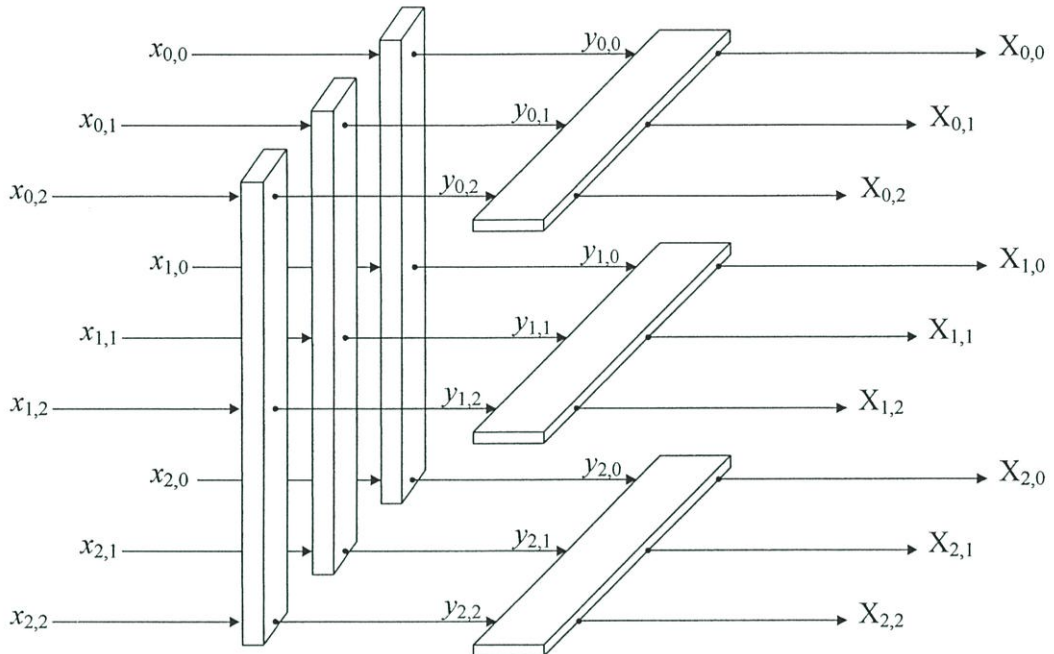
สมมติให้เมตริกซ์นี้ซึ่งผ่านการแปลงปาสคาลเต็มหน่วยในหนึ่งมิติ ในแนวหลัก (column operation) มาเป็น
 ดังนั้นจะได้ว่า

$$\begin{aligned}
 \therefore \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} &= \begin{bmatrix} y_{0,0} & y_{0,1} & y_{0,2} \\ y_{1,0} & y_{1,1} & y_{1,2} \\ y_{2,0} & y_{2,1} & y_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} y_{0,0} & y_{0,0} - y_{0,1} & y_{0,0} - 2y_{0,1} + y_{0,2} \\ y_{1,0} & y_{1,0} - y_{1,1} & y_{1,0} - 2y_{1,1} + y_{1,2} \\ y_{2,0} & y_{2,0} - y_{2,1} & y_{2,0} - 2y_{2,1} + y_{2,2} \end{bmatrix}
 \end{aligned}$$

เมตริกซ์นี้คือการดำเนินการในส่วนของแถว (row operation) สำหรับอินพุตที่ผ่านการดำเนินการในส่วนของคอลัมน์มาแล้ว ผลลัพธ์ทั้งหมดคือ การดำเนินการแปลงดีสครีตปาสคาลทรานสฟอร์มในสองมิติ ในที่สุดจะได้ว่า

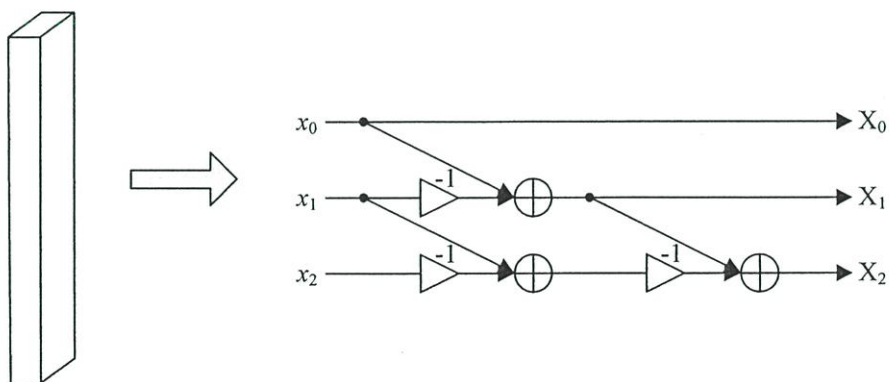
$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & x_{0,0} - x_{0,1} & x_{0,0} - 2x_{0,1} + x_{0,2} \\ [x_{0,0} - x_{1,0}] & [x_{0,0} - x_{1,0}] - [x_{0,1} - x_{1,1}] & [x_{0,0} - x_{1,0}] - 2[x_{0,1} - x_{1,1}] + [x_{0,2} - x_{1,2}] \\ [x_{0,0} - 2x_{1,0} + x_{2,0}] & [x_{0,0} - 2x_{1,0} + x_{2,0}] - [x_{0,1} - 2x_{1,1} + x_{2,1}] & [x_{0,0} - 2x_{1,0} + x_{2,0}] - 2[x_{0,1} - 2x_{1,1} + x_{2,1}] + [x_{0,2} - 2x_{1,2} + x_{2,2}] \end{bmatrix}$$

เราสามารถสรุปเป็นไดอะแกรม ได้ดังนี้



รูปที่ 3.6 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงฟูริเยร์แบบ 2 มิติ เมื่อ $N=3$

โดยสัญญาณที่ใช้ในรูปที่ 3.6 มีความหมายดังในรูปที่ 3.7



รูปที่ 3.7 แสดงโครงสร้างแบบวงจรรภายในที่ใช้ในสัญลักษณ์ตามรูปที่ 3.6 เมื่อ $N=3$

3.5 บทสรุป

เนื้อหาในบทนี้ได้นำกล่าวถึงหลักการการออกแบบ โครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงฟูรีเยอร์แบบหนึ่งมิติและสองมิติ โดยอาศัยการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์(1,0,-1) และยังคงแสดงให้เห็นถึงวิธีการแยกวิธีการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) โดยใช้ เกาสเซียน อีลิมีเนชัน (Gaussian elimination) สรุปขั้นตอนการสร้างไบนารีเมตริกซ์ (1,0,-1) และแสดง โครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงฟูรีเยอร์แบบหนึ่งมิติและสองมิติ ที่สร้างจากการแยกองค์ประกอบเมตริกซ์ของการแปลงฟูรีเยอร์ชนิดความถี่สูงผ่าน(High-pass type) และชนิดความถี่ต่ำผ่าน(Low-pass type)ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) จากโครงสร้างทางฮาร์ดแวร์ที่นำเสนอนี้เป็น โครงสร้างที่เหมาะสมและง่ายสำหรับสร้างเป็นฮาร์ดแวร์ โดยเฉพาะอย่างยิ่งในการออกแบบวงจรรวม (VLSI design) โครงสร้างนี้ เหมาะสำหรับการใช้เทคนิคที่เรียกว่าไพพ์ไลน์ (Pipeline technique) โดยการวางไพพ์ไลน์ รีจิสเตอร์ (Pipeline register) ในแต่ละสเตจ เพื่อช่วยในเรื่องของความเร็วในการแปลงฟูรีเยอร์แบบสองมิติ (DPT) และแสดงให้เห็นถึงคุณสมบัติของการแปลงฟูรีเยอร์แบบหนึ่งมิติทั้งสองชนิด

บทที่ 4

การประยุกต์ใช้งานการแปลงปาสคาลเต็มหน่วยสำหรับการ ประมวลผลสัญญาณเชิงเลข

การกรองความถี่สูงผ่านอาจจะเป็นลักษณะเฉพาะที่สำคัญของการแปลงปาสคาลเต็มหน่วย ซึ่งเป็นสมมติฐานที่เราตั้งขึ้น โดยต้องทดสอบและออกแบบการทดสอบนั้น ซึ่งแนวคิดนี้มาจาก [1] จากที่ได้กล่าวมาแล้ว ต่อไปจะเป็นการพิสูจน์การกรองความถี่สูงผ่านของปาสคาลเมตริกซ์

4.1 คุณสมบัติความถี่สูงผ่านของระบบการแปลงปาสคาลเต็มหน่วย

ปาสคาลทรานสเฟอร์ฟังก์ชันในหนึ่งมิติ

ตัวอย่างเช่น : เมตริกซ์ขนาด 3×3

$$\text{จาก } \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 - x_1 \\ X_2 &= x_0 - 2x_1 + x_2 \end{aligned}$$

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้โดยการเปลี่ยนตัวแปรเป็น

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{bmatrix}$$

ดังนั้น

$$x(n-2) = Y_0 \quad (4.1)$$

$$x(n-2) - x(n-1) = Y_1 \quad (4.2)$$

$$x(n-2) - 2x(n-1) + x(n) = Y_2 \quad (4.3)$$

จากสมการที่ (4.2) จะได้

$$Y_0 - x(n-1) = Y_1 \quad (4.4)$$

จากสมการที่ (4.3) และ (4.4) จะได้

$$Y_0 - 2[Y_0 - Y_1] + x(n) = Y_2$$

$$x(n-2) - 2[x(n-2) - (x(n-2) - x(n-1))] + x(n) = Y_2$$

$$x(n-2) - 2x(n-1) + x(n) = Y_2$$

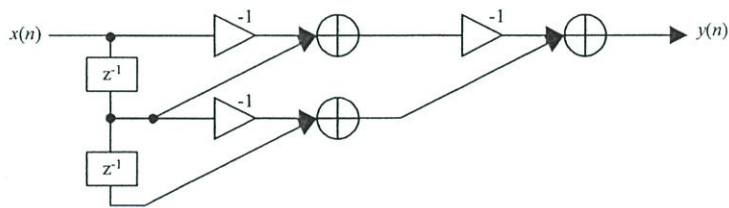
เราพิจารณา Y_2 สำหรับ $y(n)$ เพราะเราพิจารณา x_2 เป็น $x(n)$ ดังนั้น

$$\therefore y(n) = x(n) - 2x(n-1) + x(n-2)$$

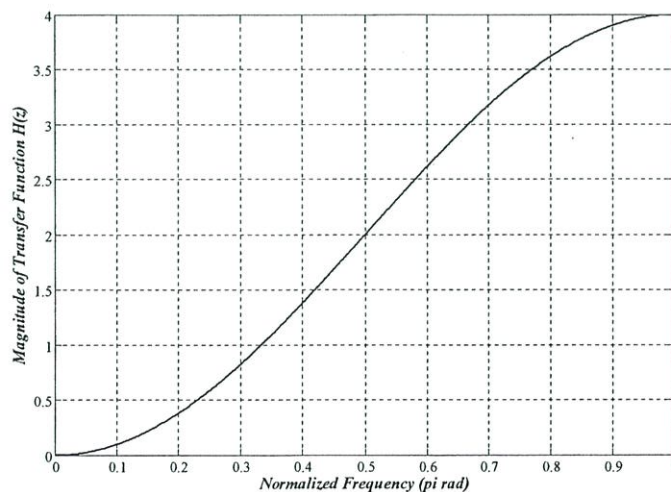
ดังนั้นจะได้ทรานสเฟอร์ฟังก์ชันคือ

$$\therefore H(z) = 1 - 2z^{-1} + z^{-2} \quad (4.5)$$

และจะได้โครงสร้างทางฮาร์ดแวร์ของวงจรกรองปาสคาล ดังในรูปที่ 4.5 และแสดงคุณลักษณะทางความถี่ (เชิงขนาด) ได้ดังรูปที่ 4.1 ซึ่งจะเห็นได้ว่ามีคุณลักษณะเป็นวงจรกรองความถี่สูงผ่าน



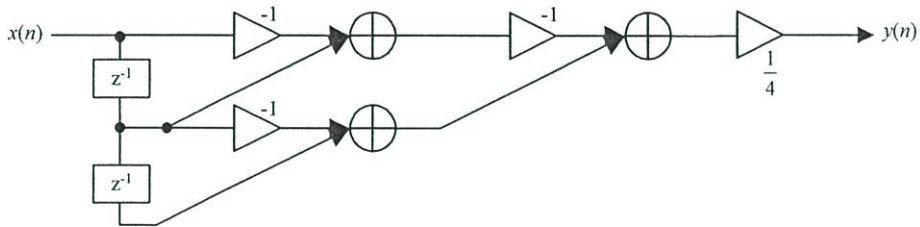
รูปที่ 4.1 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (4.5)



รูปที่ 4.2 ผลตอบสนองทางแอมพลิจูดจากสมการที่ (4.5)

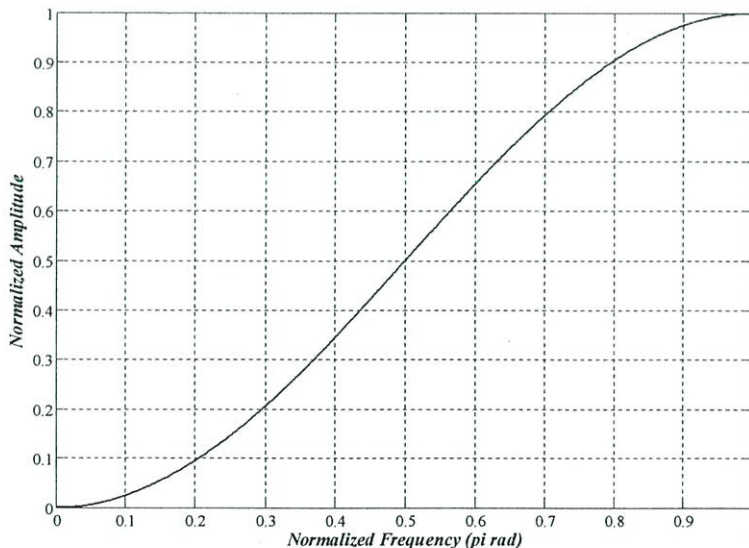
จากรูปที่ 4.2 ถ้าต้องการค่าสูงสุดของผลตอบสนองทางแอมพลิจูดเป็น 1 ต้องปรับค่าทรานสเฟอร์ฟังก์ชันก่อนหน้าโดยปรับลงด้วยผลรวมของค่าสมมูลของสัมประสิทธิ์ทั้งหมด จะได้ทรานสเฟอร์ฟังก์ชันใหม่คือ

$$\therefore H(z) = \frac{1}{4} (1 - 2z^{-1} + 1z^{-2}) \quad (4.6)$$



รูปที่ 4.3 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (4.6)

ในการปรับค่าสูงสุดของผลตอบสนองทางแอมพลิจูดนั้น ในส่วนของโครงสร้างฮาร์ดแวร์ จะทำได้โดยการเลื่อนบิตเอาต์พุตไปทางขวามือซึ่งให้ผลเสมือนกับการหารนั่นเอง จะได้ผลตอบสนองความถี่ใหม่ตามรูปที่ 4.3

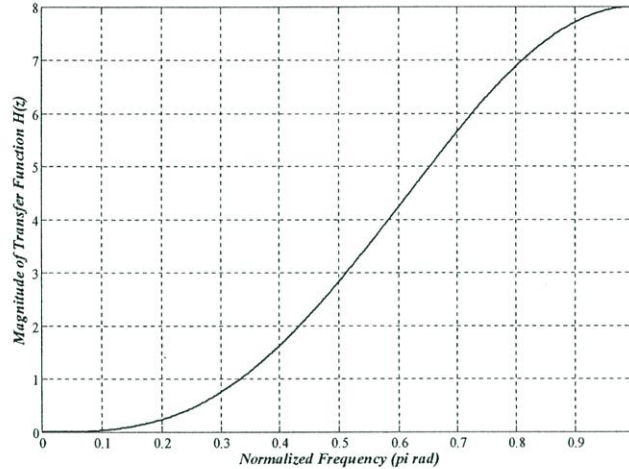


รูปที่ 4.4 ผลตอบสนองทางความถี่จากสมการที่ (4.6)

กรณีนี้ $N=4$, order ของ filter คือ $N-1 = 3^{\text{rd}}$ order ของวงจรกรองปาสคาล

ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน (Normalized transfer function) จะได้สมการดังนี้

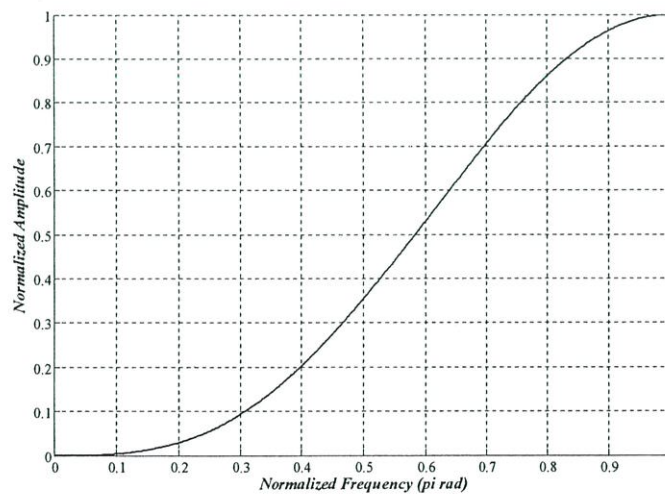
$$H(z) = 1 - 3z^{-1} + 3z^{-2} - 1z^{-3} \quad (4.7)$$



รูปที่ 4.5 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{rd} order

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

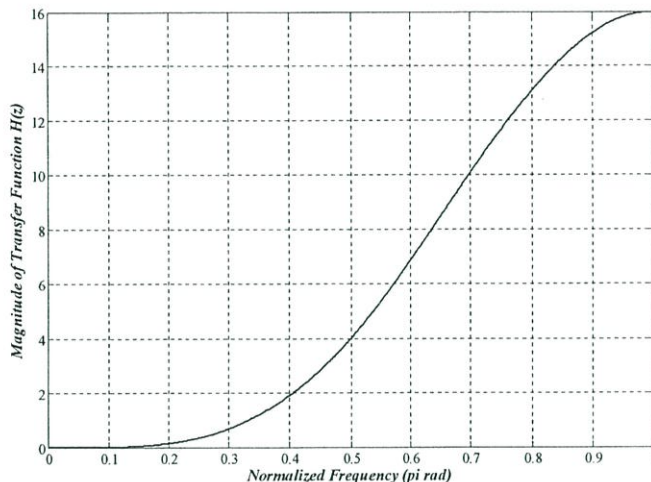
$$H(z) = \frac{1}{8} (1 - 3z^{-1} + 3z^{-2} - 1z^{-3}) \quad (4.8)$$



รูปที่ 4.6 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{rd} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่มี $N=5$, order ของฟิลเตอร์ คือ $N-1 = 5^{\text{th}}$ order ของวงจรรองปาสคาล
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

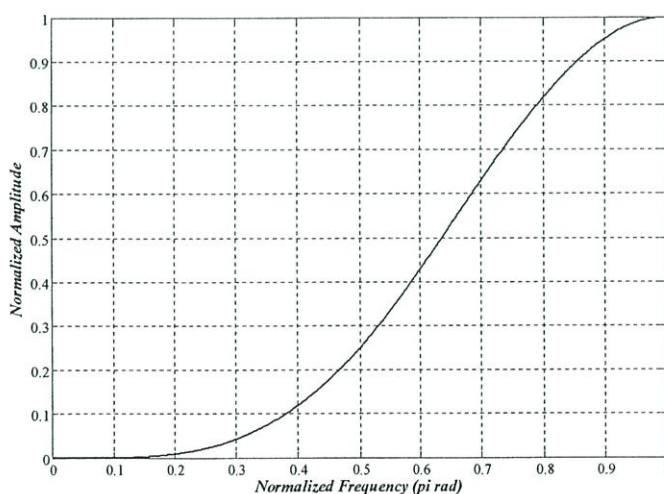
$$H(z) = 1 - 4z^{-1} + 6z^{-2} - 4z^{-3} + 1z^{-4} \quad (4.9)$$



รูปที่ 4.7 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

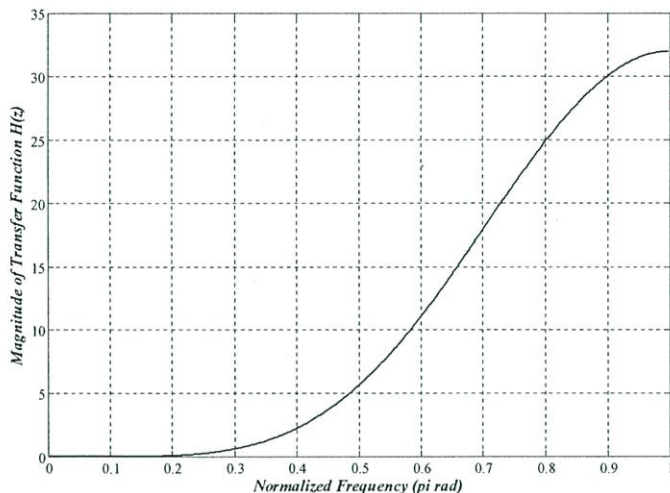
$$H(z) = \frac{1}{16} (1 - 4z^{-1} + 6z^{-2} - 4z^{-3} + 1z^{-4}) \quad (4.10)$$



รูปที่ 4.8 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่ $N=6$, order ของฟิลเตอร์ คือ $N-1 = 5^{\text{th}}$ order ของวงจรรองปาสคาต
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

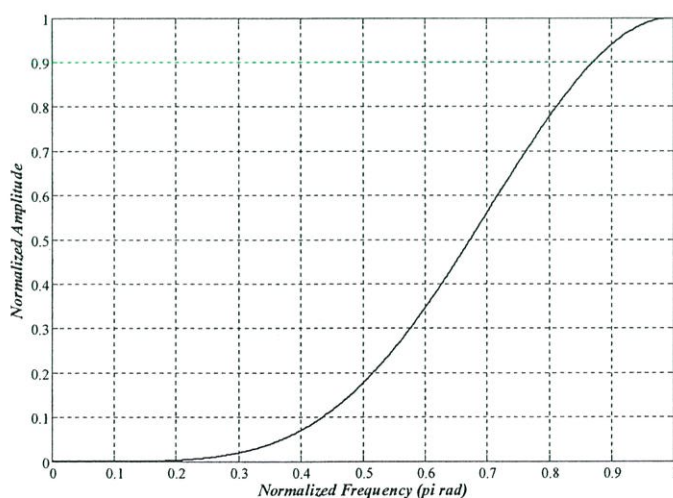
$$H(z) = 1 - 5z^{-1} + 10z^{-2} - 10z^{-3} + 5z^{-4} - 1z^{-5} \quad (4.11)$$



รูปที่ 4.9 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

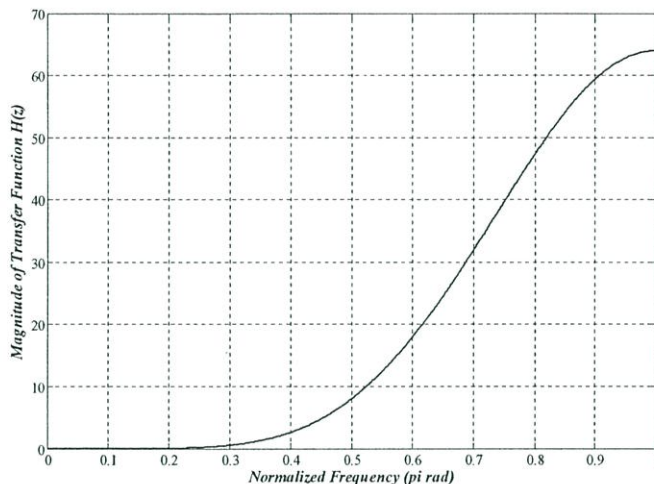
$$H(z) = \frac{1}{32} (1 - 5z^{-1} + 10z^{-2} - 10z^{-3} + 5z^{-4} - 1z^{-5}) \quad (4.12)$$



รูปที่ 4.10 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่ $N=7$, order ของฟิลเตอร์ คือ $N-1 = 6^{\text{th}}$ order ของวงจรรองปาสคาต
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

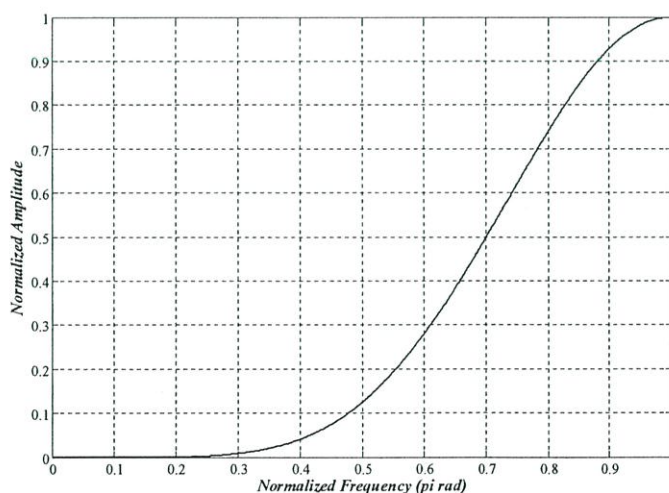
$$H(z) = 1 - 6z^{-1} + 15z^{-2} - 20z^{-3} + 15z^{-4} - 6z^{-5} + 1z^{-6} \quad (4.13)$$



รูปที่ 4.11 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order

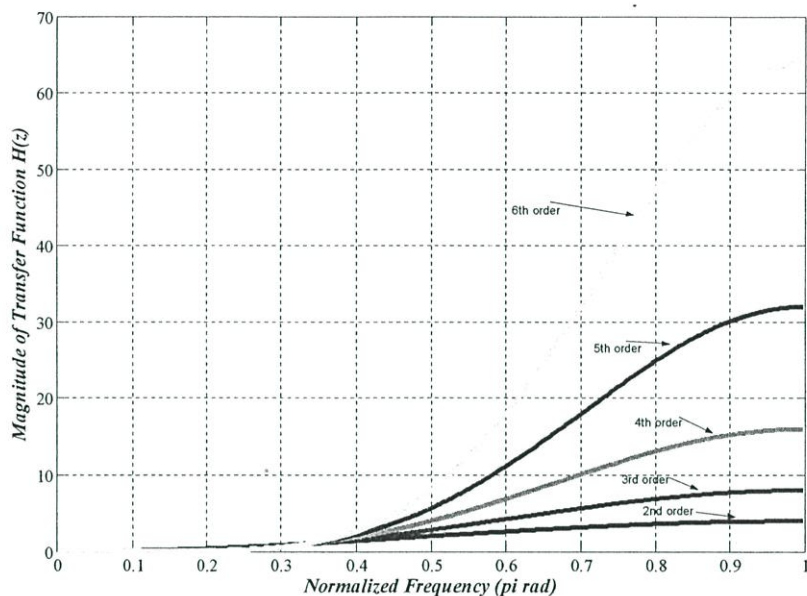
เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

$$H(z) = \frac{1}{64} (1 - 6z^{-1} + 15z^{-2} - 20z^{-3} + 15z^{-4} - 6z^{-5} + 1z^{-6}) \quad (4.14)$$

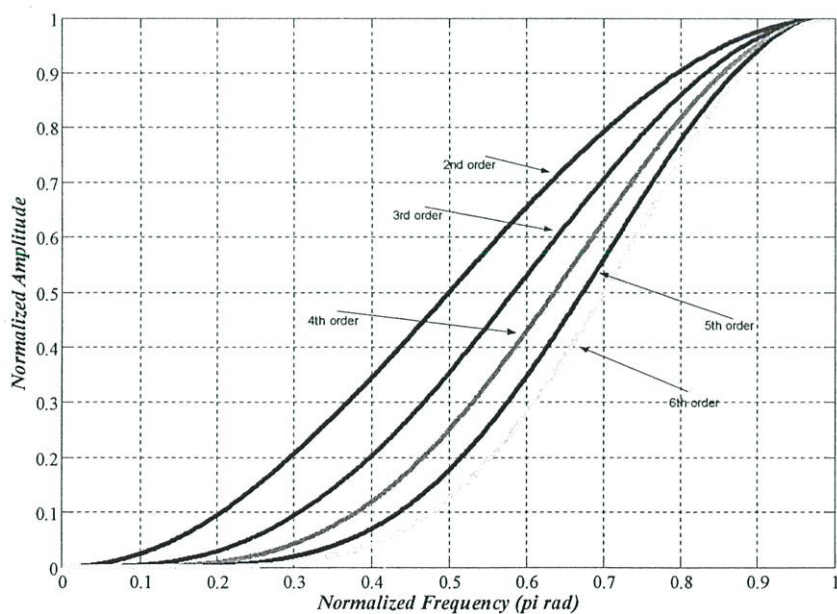


รูปที่ 4.12 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order และทำการนอร์มอลไลซ์แล้ว

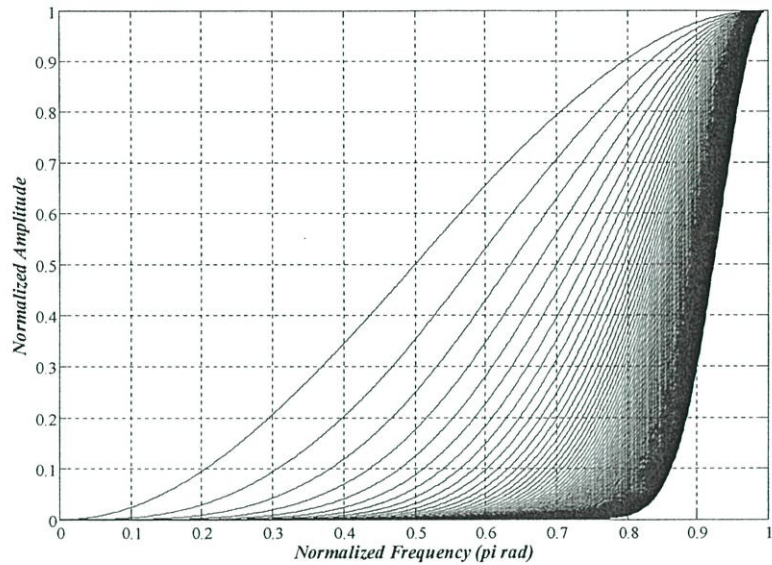
เมื่อทำการแสดงผลกราฟที่ N^{th} order มีค่าต่าง ๆ กัน



รูปที่ 4.13 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2nd order ถึง 6th order



รูปที่ 4.14 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2nd order ถึง 6th order
เมื่อทำการนอร์มอลไลซ์แล้ว



รูปที่ 4.15 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 99^{th} order

จากรูปที่ 4.14 และ 4.15 จะสังเกตเห็นได้ว่า เมื่ออันดับของวงจรกรองสูงขึ้นไป ค่าของความชันที่ได้จะมีค่าที่ค่อนข้างจะเปลี่ยนแปลงน้อยซึ่งผิดจากกรณีของวงจรกรองสัญญาณทั่ว ๆ ไป ที่เมื่ออันดับของวงจรกรองสูงขึ้นไป ความชันของผลตอบสนองทางขนาดจะเพิ่มขึ้นแต่สิ่งที่เปลี่ยนแปลงเมื่ออันดับของวงจรกรองที่ได้เปลี่ยนไปก็คือความกว้าง-แคบของย่านผ่านสัญญาณซึ่งคุณลักษณะนี้จะคล้ายกับวงจรกรองสัญญาณที่เรียกว่าวงจรกรองเกาส์เซียน (Gaussian Filter) เมื่ออันดับของวงจรเปลี่ยนไป จะมีผลทำให้ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของผลตอบสนองทางขนาดนี้เปลี่ยนไปด้วย

4.2 คุณสมบัติความถี่ต่ำผ่านของระบบการแปลงปาสคาลเต็มหน่วย

ปาสคาลทรานสเฟอร์ฟังก์ชันในหนึ่งมิติ

ตัวอย่างเช่น : เมตริกซ์ขนาด 3×3

$$\text{จาก } \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 + x_1 \\ X_2 &= x_0 + 2x_1 + x_2 \end{aligned}$$

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้ โดยการเปลี่ยนตัวแปรเป็น

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{bmatrix}$$

ดังนั้น

$$x(n-2) = Y_0 \quad (4.15)$$

$$x(n-2) + x(n-1) = Y_1 \quad (4.16)$$

$$x(n-2) + 2x(n-1) + x(n) = Y_2 \quad (4.17)$$

จากสมการที่ (4.16) จะได้

$$Y_0 + x(n-1) = Y_1 \quad (4.18)$$

จากสมการที่ (4.17) และ (4.18) จะได้

$$\begin{aligned} Y_0 + 2[Y_1 - Y_0] + x(n) &= Y_2 \\ x(n-2) + 2[(x(n-2) + x(n-1)) - x(n-2)] + x(n) &= Y_2 \\ x(n-2) + 2x(n-1) + x(n) &= Y_2 \end{aligned}$$

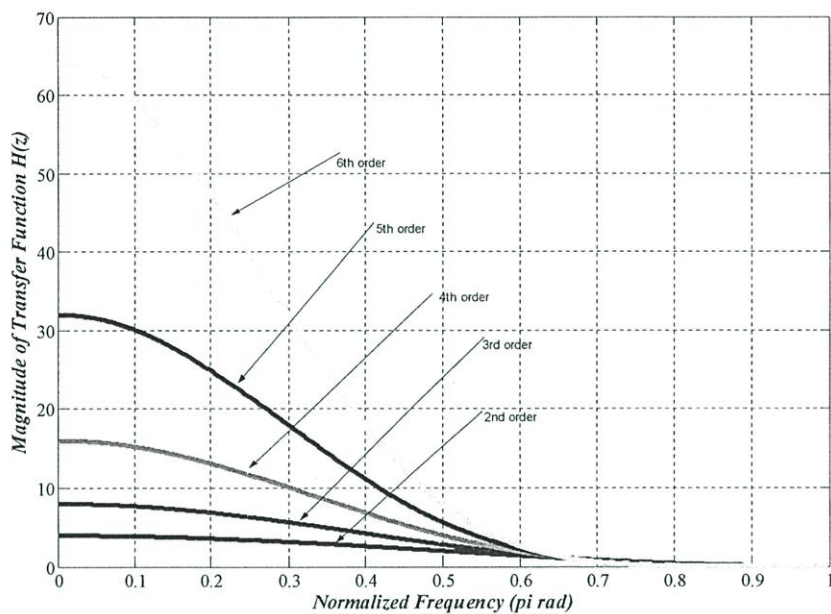
เราพิจารณา Y_2 สำหรับ $y(n)$ เพราะเราพิจารณา x_2 เป็น $x(n)$ ดังนั้น

$$\therefore y(n) = x(n) + 2x(n-1) + x(n-2)$$

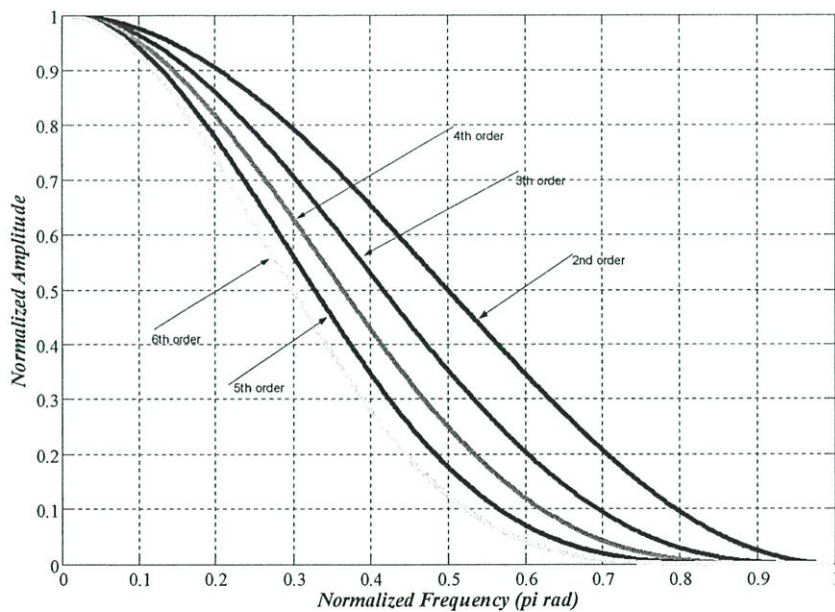
ดังนั้นจะได้ทรานสเฟอร์ฟังก์ชันคือ

$$\therefore H(z) = 1 + 2z^{-1} + 1z^{-2} \quad (4.19)$$

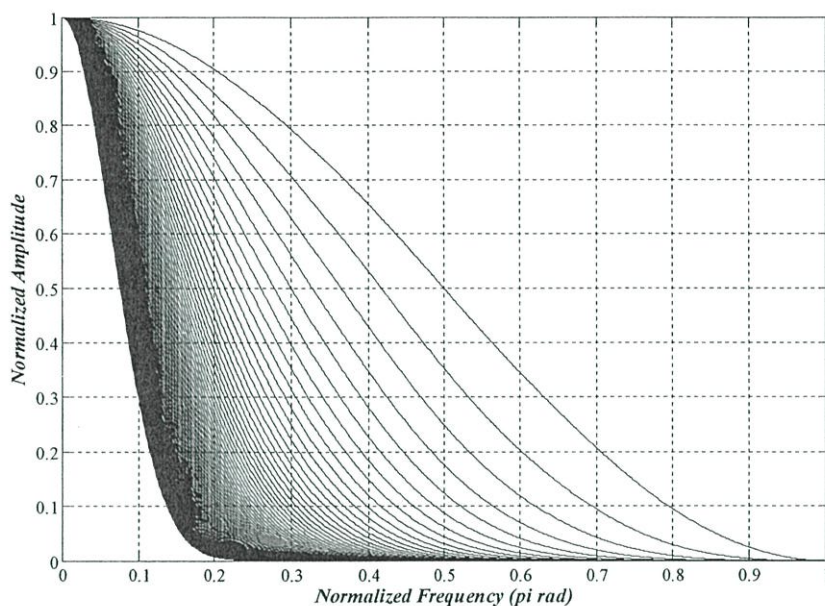
เมื่อทำการแสดงผลกราฟที่ N^{th} order มีค่าต่าง ๆ กัน



รูปที่ 4.16 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order



รูปที่ 4.17 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order
เมื่อทำการนอร์มอลไลซ์แล้ว



รูปที่ 4.18 กราฟแสดงผลตัวกรองความถี่ต่ำผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 99^{th} order

จากรูปที่ 4.17 และ 4.18 จะสังเกตเห็นได้ว่า เมื่ออันดับของวงจกรองสูงขึ้น ค่าของความชันที่ได้จะมีค่าที่ค่อนข้างจะเปลี่ยนแปลงน้อยซึ่งผิดจากกรณีของวงจกรองสัญญาณทั่ว ๆ ไป ที่เมื่ออันดับของวงจกรองสูงขึ้น ความชันของผลตอบสนองทางขนาดจะเพิ่มขึ้นแต่สิ่งที่เปลี่ยนแปลงเมื่ออันดับของวงจกรองที่ได้เปลี่ยนไปก็คือความกว้าง-แคบของย่านผ่านสัญญาณซึ่งคุณลักษณะนี้จะคล้ายกับวงจกรองสัญญาณที่เรียกว่าวงจกรองเกาส์เซียน (Gaussian Filter) เมื่ออันดับของวงจรถ่ายไป จะมีผลทำให้ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของผลตอบสนองทางขนาดนี้เปลี่ยนไปด้วย

4.3 การใช้งานเป็นตัวกรองปาสคาล (Pascal Filter) สำหรับสัญญาณเต็มหน่วย

การคุณสมบัติที่กล่าวมาข้างต้นคือคุณสมบัติความถี่สูงผ่านของระบบการแปลงปาสคาลเต็มหน่วยและคุณสมบัติความถี่ต่ำผ่านของระบบการแปลงปาสคาลเต็มหน่วยแสดงให้เห็นถึงคุณสมบัติการเป็นตัวกรองดังนั้นเราจะเรียกว่า "วงจกรองปาสคาล" ซึ่งสามารถประยุกต์ใช้กับงานทางด้านเสียงซึ่งมักพิจารณาว่าเสียงคือสัญญาณ 1 มิติที่ประกอบขึ้นจากสัญญาณความถี่ ต่าง ๆ ผสมกันอยู่ในสัดส่วนที่ต่างกัน การออกแบบตัวกรองจึงเป็นการกำหนดว่าเราต้องการกำจัดสัญญาณความถี่ใดออกไป หรือต้องการเลือกสัญญาณความถี่ใดบ้าง

ซึ่งตัวกรองแบ่งออกได้เป็น 4 ประเภทตามลักษณะการเลือกความถี่คือ

- ตัวกรองความถี่ต่ำผ่าน (Low-pass Filter)
- ตัวกรองความถี่สูงผ่าน (High-pass Filter)
- ตัวกรองแถบความถี่ผ่าน (Band-pass Filter)
- ตัวกรองหยุดแถบความถี่ (Band-stop Filter)

โดยที่ค่าพารามิเตอร์หลักในการกำหนดคุณสมบัติของตัวกรองคือ ค่าความถี่คัทออฟ (cut-off frequency) ความถี่คัทออฟคือ ความถี่ที่ระบุจุดตัดของสัญญาณว่าจะให้ผ่าน หรือไม่ผ่าน เราจะแสดงผลของวงจรกรองปาสคาล 1 มิติในบทที่ 5

ถ้าเราพูดถึง การกรองข้อมูลภาพ (Image Filtering) คือ การนำภาพไปผ่านตัวกรองสัญญาณ เพื่อให้ได้ภาพผลลัพธ์ออกมา ภาพผลลัพธ์ที่ได้จะมีคุณสมบัติแตกต่างจากภาพต้นแบบ วัตถุประสงค์หลักของการกรองข้อมูลภาพ คือการเน้น (enhance) หรือลดทอน (attenuate) คุณสมบัติบางประการของภาพ เพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ

การกรองข้อมูลภาพคือการประมวลผลภาพอย่างหนึ่งที่ทำเป็นประจำ เนื่องจากในการใช้งานจริง ภาพที่ได้มามักมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่น ๆ ปะปนอยู่ด้วย การกรองข้อมูลภาพสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้น เหมาะแก่การประมวลผลในขั้นต่อไป

ตัวกรองคือระบบ ๆ หนึ่งซึ่งรับสัญญาณเข้าประมวลผลสัญญาณและส่งสัญญาณออก โดยทั่วไปตัวกรองจะถูกสร้างให้เป็นระบบเชิงเส้น (linear system) เนื่องจากออกแบบได้ง่าย และมีประสิทธิภาพดี ซึ่งในการกรองข้อมูลภาพ เรามักพิจารณาว่าภาพคือสัญญาณ 2 มิติที่ประกอบขึ้นจากสัญญาณความถี่ ต่าง ๆ ผสมกันอยู่ในสัดส่วนที่ต่างกัน การออกแบบตัวกรองจึงเป็นการกำหนดว่าเราต้องการกำจัดสัญญาณความถี่ใดออกไป หรือต้องการเลือกสัญญาณความถี่ใดบ้าง

ซึ่งตัวกรองแบ่งออกได้เป็น 4 ประเภทตามลักษณะการเลือกความถี่คือ

- ตัวกรองความถี่ต่ำผ่าน (Low-pass Filter)
- ตัวกรองความถี่สูงผ่าน (High-pass Filter)
- ตัวกรองแถบความถี่ผ่าน (Band-pass Filter)
- ตัวกรองหยุดแถบความถี่ (Band-stop Filter)

โดยที่ค่าพารามิเตอร์หลักในการกำหนดคุณสมบัติของตัวกรองคือ ค่าความถี่คัทออฟ (cut-off frequency) ความถี่คัทออฟคือ ความถี่ที่ระบุจุดตัดของสัญญาณว่าจะให้ผ่าน หรือไม่ผ่าน เราจะแสดงผลของวงจรกรองปาสคาล 2 มิติในบทที่ 5 เช่นกัน

4.4 ผลตอบสนองอิมพัลส์ (Impulse response) ของวงจรกรองปาสคาล 2 มิติ (2-D Pascal Filter)

ตัวอย่างเช่น : เมตริกซ์ขนาด 3 x 3

$$\text{จาก } \begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \\ X_{20} & X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้ โดยการเปลี่ยนตัวแปรเป็น

$$\begin{bmatrix} Y_{00} & Y_{01} & Y_{02} \\ Y_{10} & Y_{11} & Y_{12} \\ Y_{20} & Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x(m-2, n-2) & x(m-2, n-1) & x(m-2, n) \\ x(m-1, n-2) & x(m-1, n-1) & x(m-1, n) \\ x(m, n-2) & x(m, n-1) & x(m, n) \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

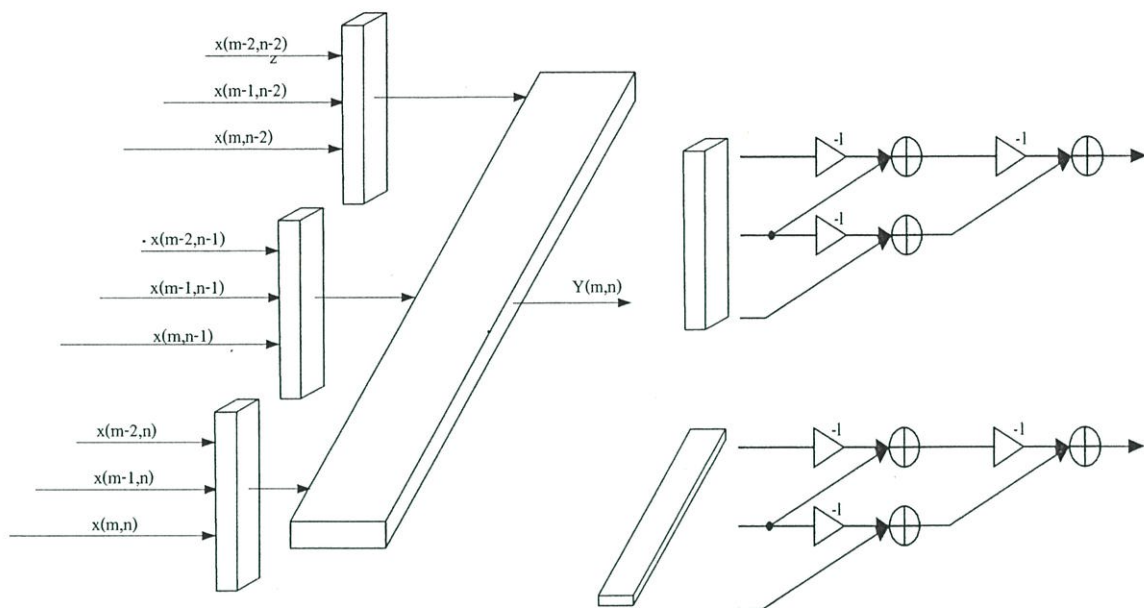
จากตัวอย่างในปาสคาลทรานสเฟอร์ฟังก์ชันในหนึ่งมิติ เมื่อเราพิจารณา Y_{22} สำหรับ $y(m, n)$ จะได้ดังนี้

$$\begin{aligned} \therefore y(m, n) &= x(m, n) - 2x(m, n-1) + x(m, n-2) \\ &\quad - 2x(m-1, n) + 4x(m-1, n-1) + 2x(m-1, n-2) \\ &\quad + x(m-2, n) - 2x(m-2, n-1) + x(m-2, n-2) \end{aligned} \quad (4.22)$$

ดังนั้นจะได้ผลตอบสนองอิมพัลส์ของระบบประมวลผลสัญญาณ 2 มิติ ในหน้ากากคอนโวลูชัน (Covolution Mask) และเราจะเรียกหน้ากากที่ได้นี้ว่าหน้ากากคอนโวลูชันปาสคาล (Pascal Convolution Mask) ดังแสดงในรูปที่ 4.19

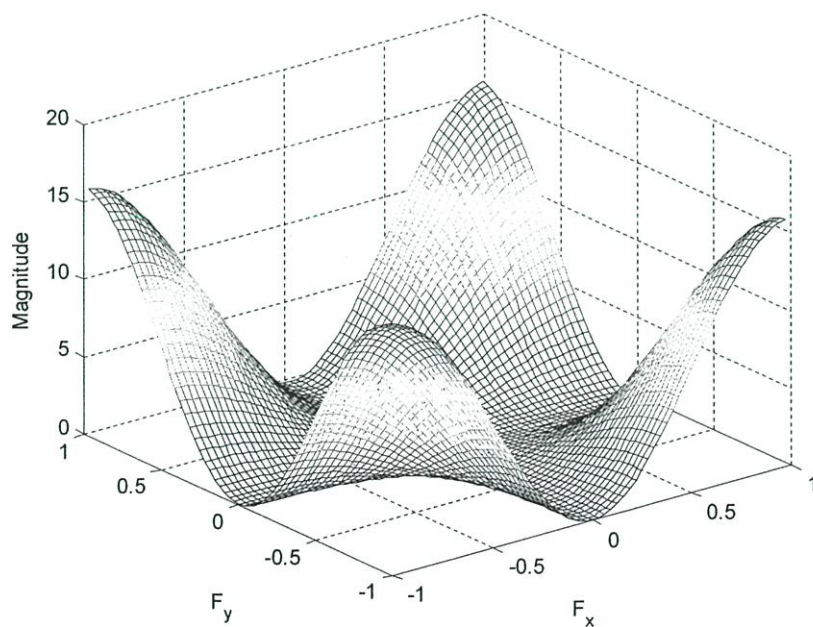
1	-2	1
-2	4	-2
1	-2	1

รูปที่ 4.19 แสดงหน้ากากคอนโวลูชันของวงจรกรองปาสคาล 2 มิติ ซึ่งมีขนาด 3x3



รูปที่ 4.20 โครงสร้างของวงจกรองปาสคาล 2 มิติ ตามสมการที่ (2.40)

ในส่วนของวงจกรองปาสคาล 2 มิตินี้ เราทำการแสดงผลตอบสนองทางความถี่ได้ดังรูป



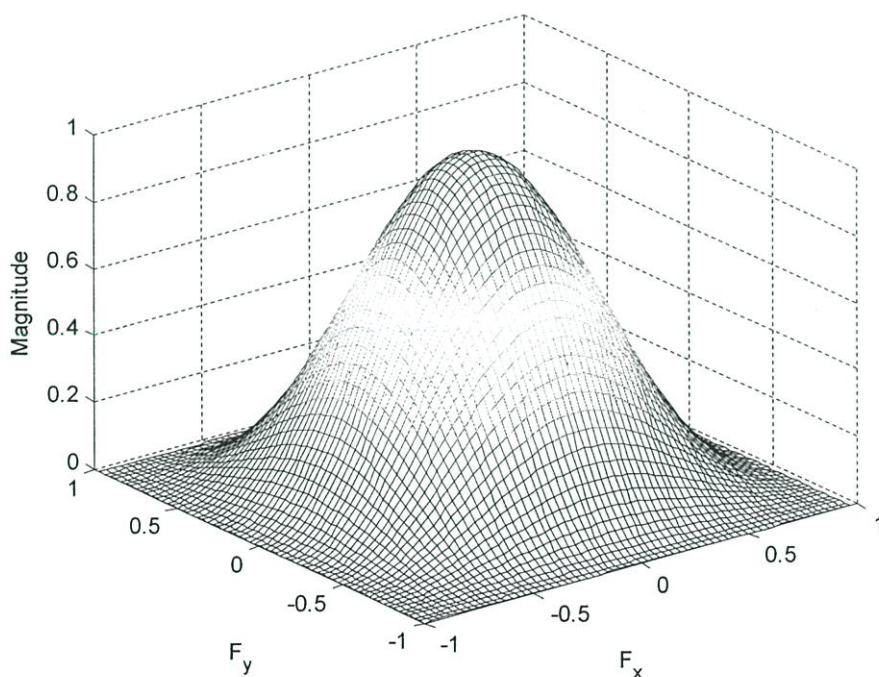
รูปที่ 4.21 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 3x3

จากรูปที่ 4.21 ผลตอบสนองทางความถี่ที่ได้เป็นแบบความถี่สูงผ่านดังนั้นถ้าพิจารณาแบบ
 ชนิดความถี่ต่ำผ่านจะได้ผลตอบสนองอิมพัลส์ของระบบประมวลผลสัญญาณ 2 มิติ ในหน้ากาคอน
 โวลูชัน (Covolution Mask) และเราจะเรียกหน้ากาคอนโวลูชันที่ได้นี้ว่าหน้ากาคอนโวลูชันปาสคาล
 (Pascal Convolution Mask) ชนิดความถี่ต่ำผ่าน ดังแสดงในรูปที่ 4.22

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

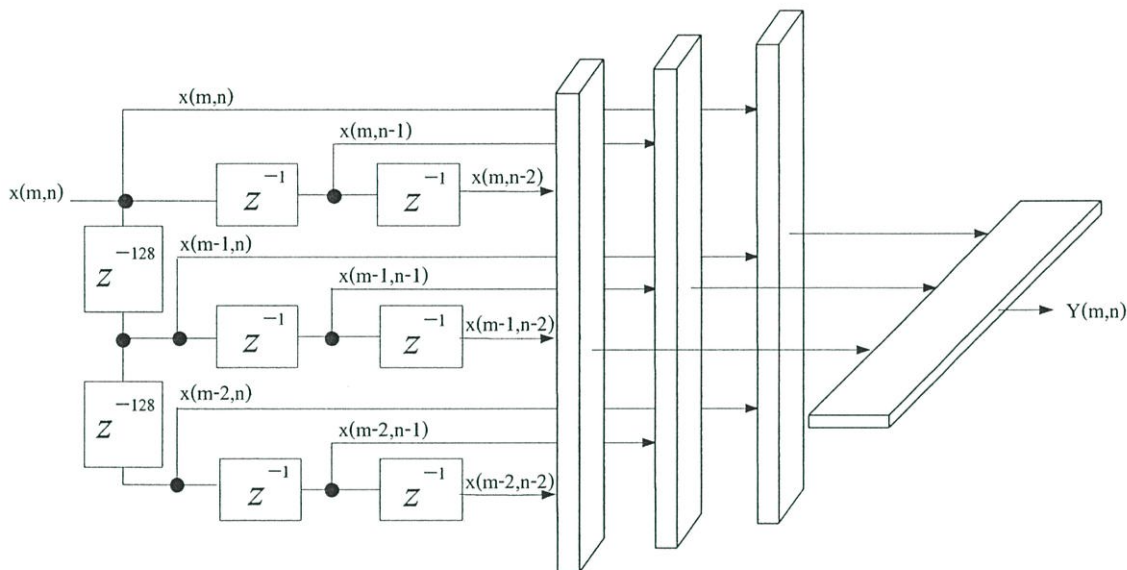
รูปที่ 4.22 หน้ากาคอนโวลูชันของวงจรรองปาสคาล 2 มิติ ซึ่งมีขนาด 3x3 ชนิดความถี่ต่ำผ่าน

ค่า $1/16$ มาจากผลรวมของทุกๆอิลิเมนต์ในหน้าการคอนโวลูชัน ใช้ในการทำการนอร์มอล
 ไลซ์ ในส่วนของวงจรรองปาสคาล 2 มิติชนิดความถี่ต่ำผ่านนี้ เราทำการแสดงผลตอบสนองทาง
 ความถี่ได้ดังรูป



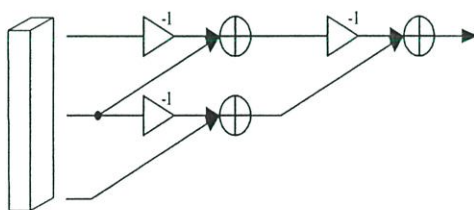
รูปที่ 4.23 ผลตอบสนองทางความถี่ที่ได้จากหน้ากาคอนโวลูชันปาสคาล ขนาด 3x3 ชนิดความถี่
 ต่ำผ่าน

จากโครงสร้างของวงจรกรองปาสคาล 2 มิติ เมื่อนำมาใช้ในการกรองข้อมูลภาพเราจะต้องคำนึงถึงลำดับของการแสดงผลข้อมูลดังนั้นจึงต้องมีการจัดลำดับของภาพก่อนที่จะทำการประมวลผลภาพ โดยวงจรนี้จะทำหน้าที่ ประมวลผลว่าจะให้ข้อมูลภาพนี้แสดงผลที่หน้าจอเมื่อใดเพื่อที่จะให้รูปภาพไม่เกิดการผิดเพี้ยนหรือบิดเบี้ยวไป ดังแสดงในรูป



รูปที่ 4.24 วงจรสร้างลำดับของข้อมูลก่อนนำไปประมวลผลกับวงจรกรองปาสคาล 2 มิติ

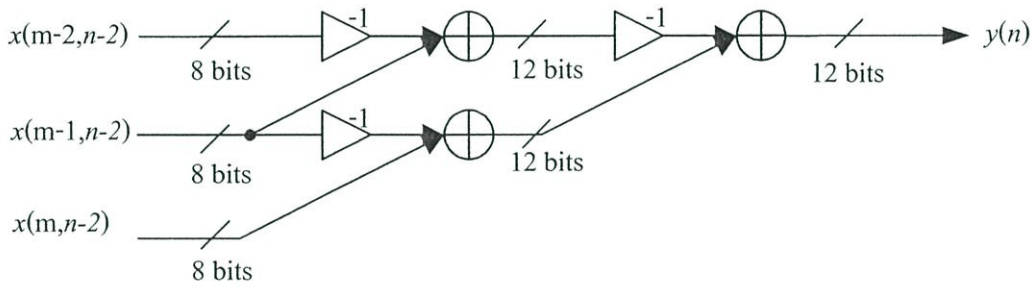
โดยสัญญาณที่ใช้ในรูปที่ 4.24 แสดงเป็นวงจรภายในดังรูปที่ 4.25



รูปที่ 4.25 แสดงบิตเตอร์ฟลายยูนิท เมื่อ $N=3$

โดยที่ข้อมูลนั้นจะมีลักษณะเป็นบิตจำนวน 8 บิต ซึ่งข้อมูลที่เข้ามาจะต้องจัดลำดับสัญญาณข้อมูลเข้าให้ถูกต้องก่อนที่จะนำไปประมวลผล นั่นคือต้องสร้างลำดับข้อมูล $X(m,n)$, $X(m,n-1)$, $X(m,n-2)$, $X(m-1,n)$, $X(m-1,n-1)$, $X(m-1,n-2)$, $X(m-2,n)$, $X(m-2,n-1)$, $X(m-2,n-2)$

ทั้งนี้กำหนดให้ m เป็นตัวหน่วยข้อมูลมีค่าเท่ากับ 128 ค่า จำนวน 2 ชุด โดยอาศัยตัวเลื่อนข้อมูล (shift register) ต่ออนุกรมกัน 256 ตัว ดังรูปที่ 4.25 จากนั้นนำข้อมูลที่จัดเรียงแล้วส่งไปคูณกับแผ่นข้อมูล (Mask) ที่มีเมตริกซ์ขนาด 3×3 เราสามารถออกแบบโครงสร้างของวงจรประมวลผลภาพได้จากสมการ ทรานสเฟอร์ฟังก์ชันตามสมการที่ 4.22 และจากรูปที่ 4.24 และ รูปที่ 4.25 สามารถออกแบบวงจรที่มีรายละเอียดดัง รูปที่ 4.26



รูปที่ 4.26 ตัวอย่างรายละเอียดของวงจรบิตเตอร์หลายยูนิต เมื่อ $N=3$

ซึ่งในการเลื่อนจุดแต่ละจุดของภาพ เราทำการเลื่อนในลักษณะของการทำคอนโวลูชัน (Convolution)

4.5 บทสรุป

เนื้อหาในบทนี้ได้กล่าวถึงหลักการการออกแบบ โครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงฟูริเยร์แบบเร็ว โดยอาศัยการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์ $(1,0,-1)$ และยังแสดงให้เห็นถึงวิธีการแยกวิธีการแยกองค์ประกอบเมตริกซ์ให้อยู่ในรูปไบนารีเมตริกซ์ $(1,0,-1)$ โดยใช้ เกาเซียน อิลิมีเนชัน (Gaussian elimination) สรุปขั้นตอนการสร้างไบนารีเมตริกซ์ $(1,0,-1)$ และแสดง โครงสร้างทางฮาร์ดแวร์อย่างมีประสิทธิภาพของการแปลงฟูริเยร์แบบเร็ว สร้างจากแยกองค์ประกอบเมตริกซ์ของการแปลงฟูริเยร์ชนิดความถี่สูงผ่าน (High-pass type) และชนิดความถี่ต่ำผ่าน (Low-pass type) ให้อยู่ในรูปไบนารีเมตริกซ์ $(1,0,-1)$ จากโครงสร้างทางฮาร์ดแวร์ที่นำเสนอนี้เป็น โครงสร้างที่เหมาะสมและง่ายสำหรับสร้างเป็นฮาร์ดแวร์ โดยเฉพาะอย่างยิ่งในการออกแบบวงจรรวม (VLSI design) โครงสร้างนี้ เหมาะสำหรับการใช้เทคนิคที่เรียกว่าไพพ์ไลน์ (Pipeline technique) โดยการวางไพพ์ไลน์ รีจิสเตอร์ (Pipeline register) ในแต่ละสเตจ เพื่อช่วยในเรื่องของความเร็วในการแปลงฟูริเยร์แบบเร็ว (DFT) และแสดงให้เห็นถึงคุณสมบัติของการแปลงฟูริเยร์แบบเร็วทั้งสองชนิด และการประยุกต์ใช้งาน

ทางด้านเสียงหรือทางด้านภาพ ซึ่งเราจะแสดงผลการออกแบบและผลการจำลองการทำงานในบทถัดไป

บทที่ 5

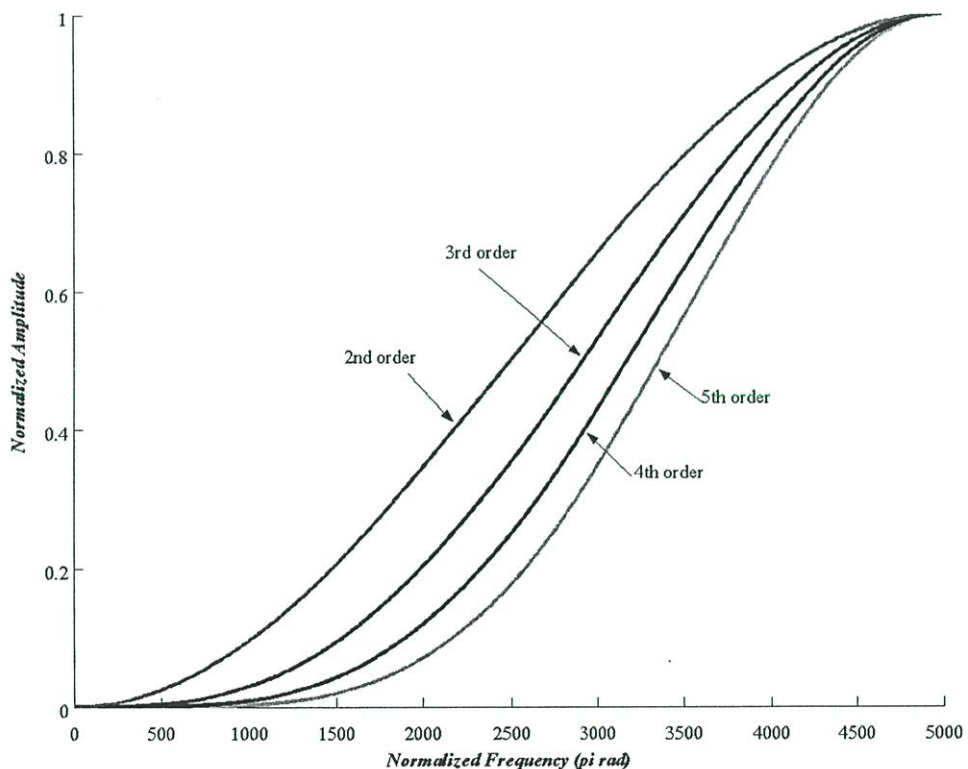
ผลการออกแบบและผลการจำลองการทำงาน

จากบทที่ 3 ทำให้เราได้โครงสร้างของการแปลงปาสคาลเต็มหน่วยซึ่งในบทนี้ทำการทดสอบคุณสมบัติของวงจรกรองโดยโปรแกรม MATLAB ดังที่จะแสดงต่อไป

5.1 ผลการออกแบบ

5.1.1 ผลที่ได้จากการแปลงสามเหลี่ยมปาสคาลใน 1 มิติ

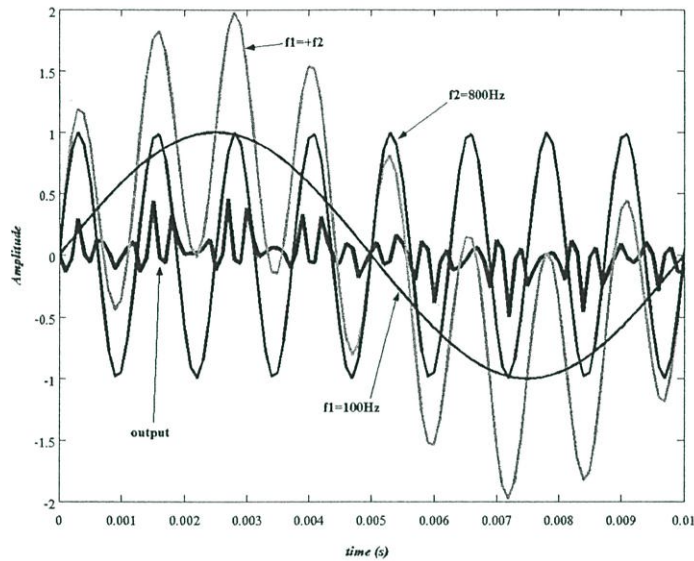
เราได้ทำการทดลองแปลงปาสคาลทรานสฟอร์มหนึ่งมิติ โดยทดลองทำที่ 2nd order ถึง 5th order ทั้งการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยและการแปลงปาสคาลฟิลเตอร์ และได้พล็อตกราฟผลตอบสนองทางแอมพลิจูดและความถี่ ได้ดังรูป



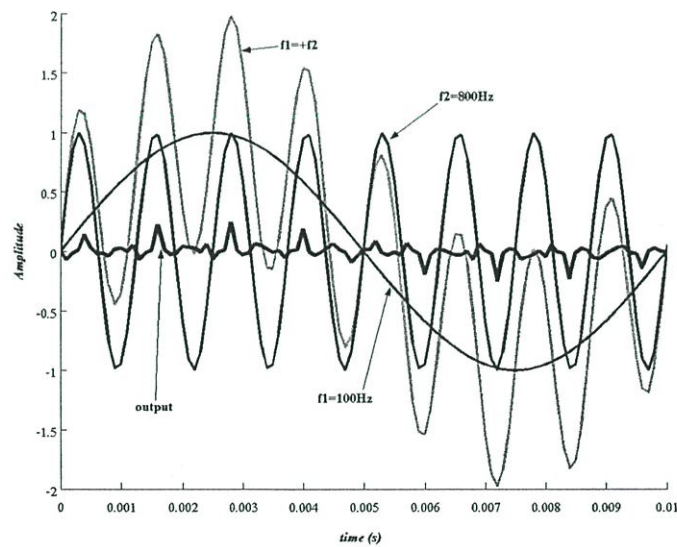
รูปที่ 5.1 แสดงผลตอบสนองทางขนาดของความเป็นตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2nd order ถึง 5th order

5.1.1.1 การจำลองการทำงานของข้อมูล

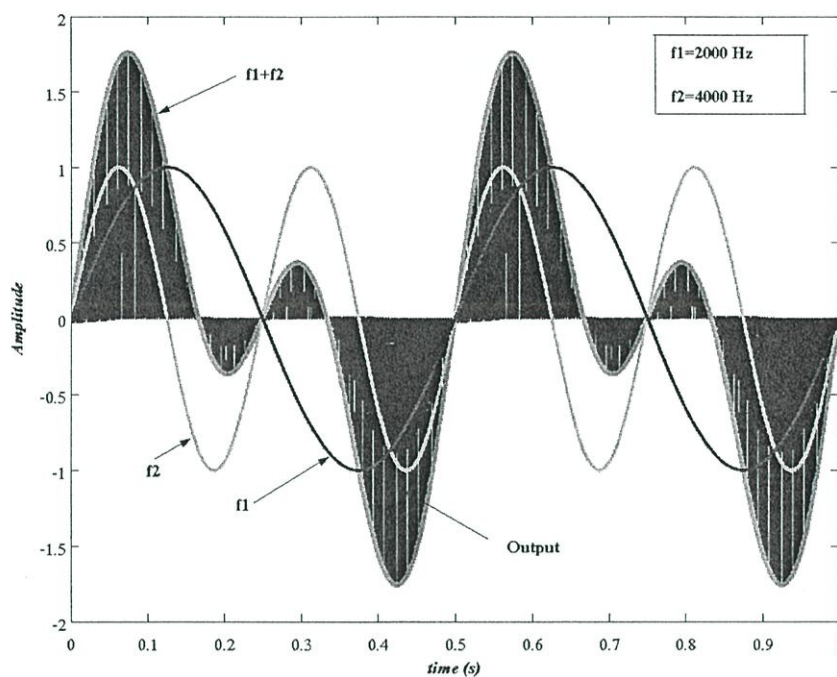
การจำลองการทำงานนี้ได้สร้างสัญญาณไซน์ f_1 และ f_2 ความถี่ 2,000 Hz และ 4,000 Hz ตามลำดับ โดยนำสัญญาณทั้งสองมาบวกกันจะได้เป็นสัญญาณ f_1+f_2 ดังรูป สัญญาณเอาต์พุตเป็นสัญญาณที่ผ่านการแปลงปาสคาลเต็มหน่วยแล้ว ดังแสดงใน [1]



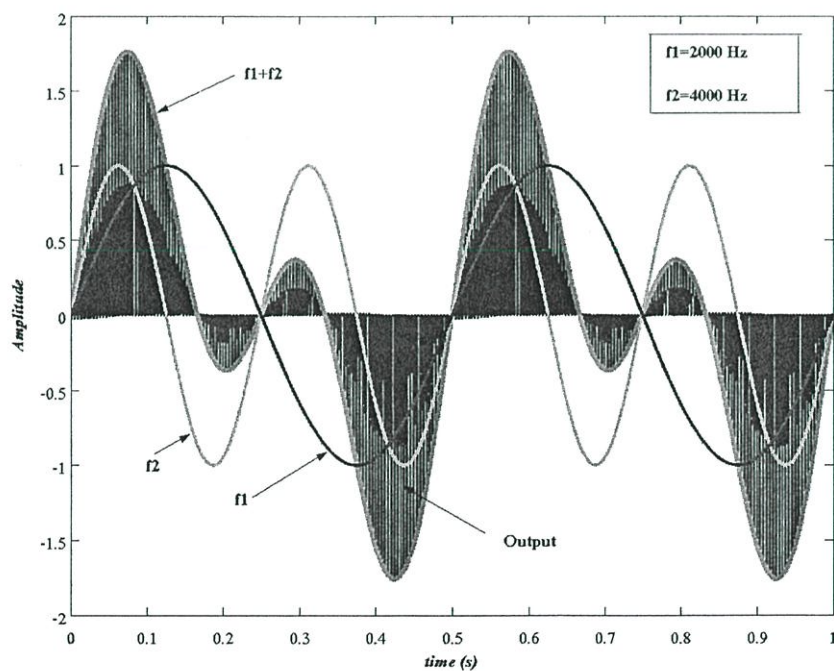
รูปที่ 5.2 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1] ขนาด 3x3 จาก MATLAB



รูปที่ 5.3 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1] ขนาด 4x4 จาก MATLAB



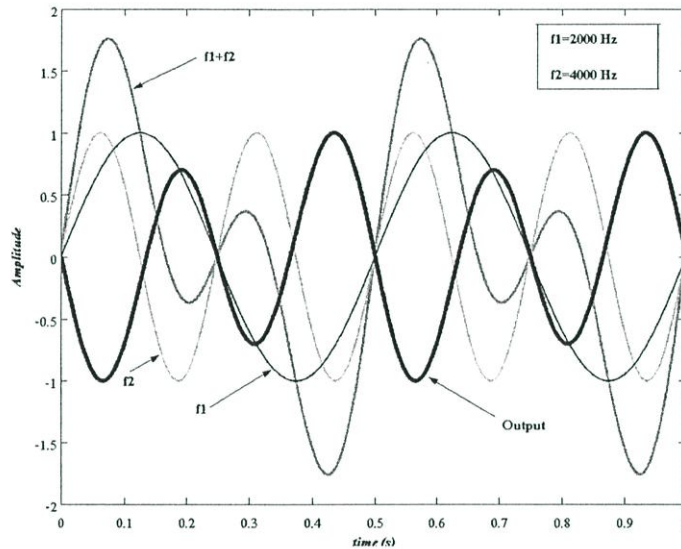
รูปที่ 5.4 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ด้วยหน่วยดัดแปลง [1] ขนาด 5x5 จาก MATLAB



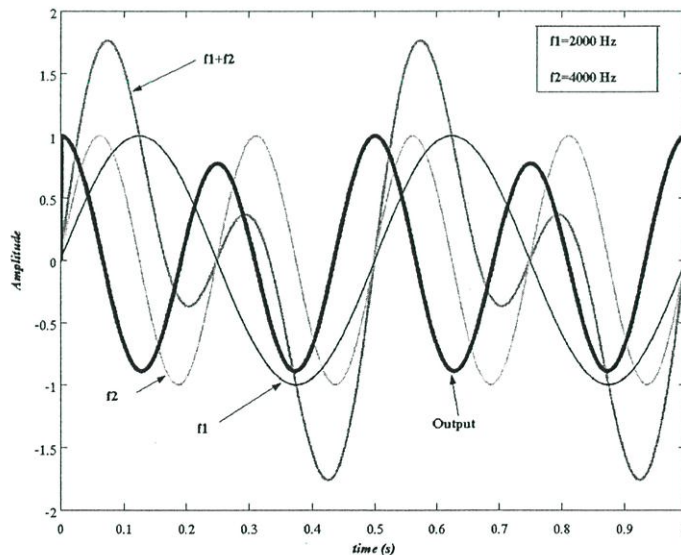
รูปที่ 5.5 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ด้วยหน่วยดัดแปลง [1] ขนาด 6x6 จาก MATLAB

จะสังเกตเห็นว่า การแปลงฟูรีเยอร์เต็มหน่วยดังใน[1] จะไม่สามารถเห็นถึงคุณสมบัติของการกรองความถี่สูงผ่านได้

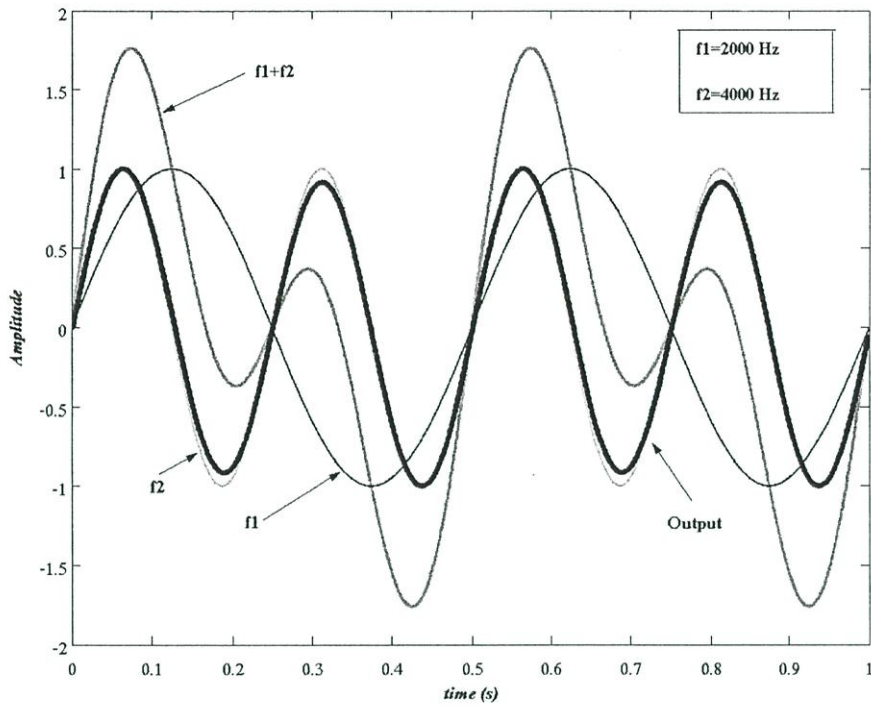
จากบทที่ 4 ที่ผ่านมาระบุได้แสดงถึงคุณสมบัติความถี่สูงผ่านและคุณสมบัติความถี่ต่ำผ่านของระบบการแปลงฟูรีเยอร์เต็มหน่วยซึ่งจะเรียกว่า”วงจรถ่ายฟูรีเยอร์ 1 มิติ” และได้แสดงวิธีหาค่าทรานสเฟอว์ฟังก์ชันใน order ต่างๆและได้ทำการจัดรูปแบบโครงสร้างใหม่ดังที่ได้กล่าวมาแล้วซึ่งในบทนี้ทำการทดสอบคุณสมบัติของวงจรถ่ายฟูรีเยอร์โดยโปรแกรม MATLAB ดังที่จะแสดงต่อไป



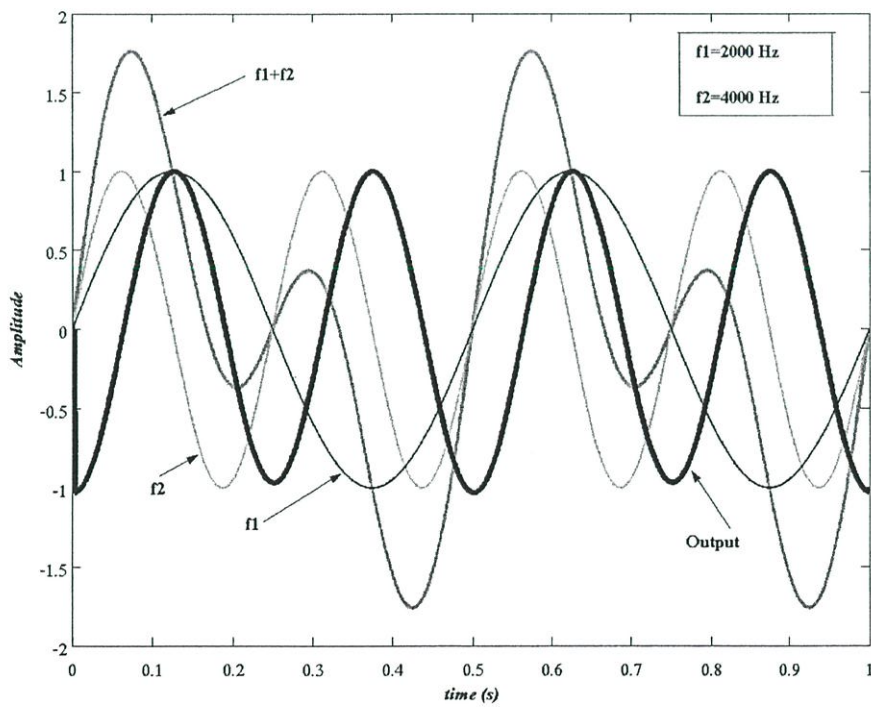
รูปที่ 5.6 ผลจากการจำลองการทำงานของวงจรถ่ายฟูรีเยอร์ 1 มิติ อันดับที่ 2 จาก MATLAB



รูปที่ 5.7 ผลจากการจำลองการทำงานของวงจรถ่ายฟูรีเยอร์ 1 มิติ อันดับที่ 3 จาก MATLAB



รูปที่ 5.8 ผลจากการจำลองการทำงานของวงจรกรองพาสคาล 1 มิติ อันดับที่ 4 จาก MATLAB



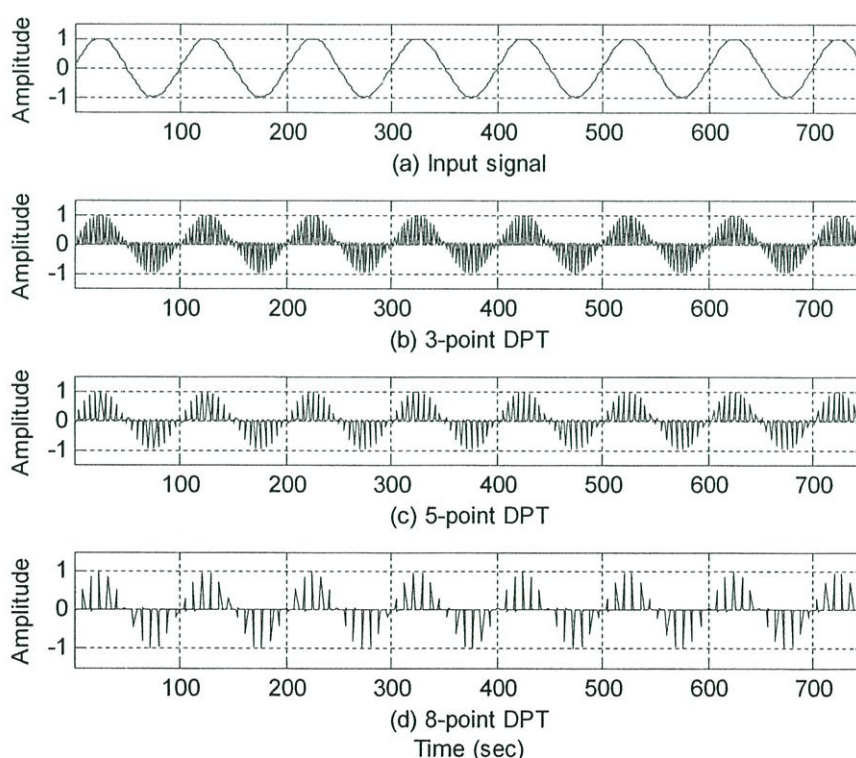
รูปที่ 5.9 ผลจากการจำลองการทำงานของวงจรกรองพาสคาล 1 มิติ อันดับที่ 5 จาก MATLAB

จากการทดลองพบว่าเมื่อนำสัญญาณที่มีองค์ประกอบความถี่สูงรวมกับความถี่ต่ำเมื่อผ่าน ปาสคาลฟิลเตอร์แล้ว สัญญาณเอาต์พุตที่ได้เป็นสัญญาณที่มีองค์ประกอบความถี่สูง นั่นก็คือ สัญญาณความถี่สูงเท่านั้นที่สามารถผ่านออกมาได้ ซึ่งเป็นคุณสมบัติของตัวกรองความถี่สูงผ่าน หนึ่งของความถี่ต่ำก็เช่นเดียวกัน

5.1.1.2 ผลการจำลองการทำงานชนิดความถี่สูงผ่าน

ขั้นที่ 1 การจำลองการทำงานนี้ได้สร้างสัญญาณไซน์ความถี่ 100 Hz ป้อนเข้าสู่ การแปลง ดิสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่าน แสดงผลดังรูปที่ 5.10 โดยที่

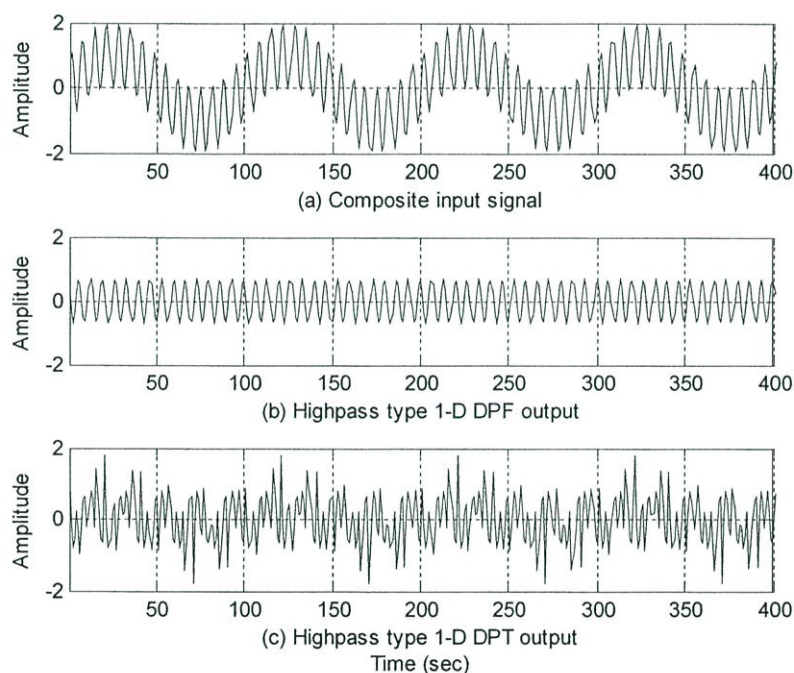
- สัญญาณอินพุต
- ผลจากการจำลองการทำงานของการแปลงดิสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่าน แบบ 3 จุด
- ผลจากการจำลองการทำงานของการแปลงดิสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่าน แบบ 5 จุด
- ผลจากการจำลองการทำงานของการแปลงดิสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่าน แบบ 8 จุด



รูปที่ 5.10 ผลจากการจำลองการทำงานของการแปลงดิสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่าน

ขั้นที่ 2 การจำลองการทำงานนี้ได้สร้างสัญญาณรวม (Composite signal) จากสัญญาณไซน์ความถี่ต่ำ 100 Hz ผสมกับไซน์ความถี่สูง 1500 Hz ป้อนเข้าสู่ วงจรการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4th order เปรียบเทียบกับการแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่าน โดยทดลองทำที่ 4th order แสดงผลดังรูปที่ 5.11 โดยที่

- สัญญาณรวมอินพุต
- ผลจากการจำลองการทำงานของการแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่าน โดยทดลองทำที่ 4th order
- ผลจากการจำลองการทำงานของการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4th order



รูปที่ 5.11 ผลเปรียบเทียบการจำลองการทำงานของการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4th order กับ การแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่าน โดยทดลองทำที่ 4th order

จากรูปที่ 5.11 จะเห็นได้ว่าผลของการแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่าน โดยทดลองทำที่ 4th order สามารถกรองความถี่สูงจากสัญญาณรวมที่ป้อนเข้าไปทดสอบได้ โดยวิธีการของการคอนโวลูชัน (Convolution method) แต่การแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4th order แบบตรงๆ จะไม่สามารถกรองความถี่สูง

จากสัญญาณรวมที่ป้อนเข้าไปทดสอบได้ ดังนั้นสามารถสรุปได้ว่าเราจะต้องใช้กระบวนการของการคอนโวลูชัน (Convolution method)

ขั้นสุดท้าย จะแสดงผลของการแปลงแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน และการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่สูงผ่าน โดยป้อนสัญญาณอินพุตเป็นภาพขนาด 256 x 256 จุด ดังแสดงในรูปที่ 5.12 และผลที่ได้จากการแปลงแสดงในรูปที่ 5.13 โดยที่

a) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 3 x 3

b) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 5 x 5

c) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 8 x 8

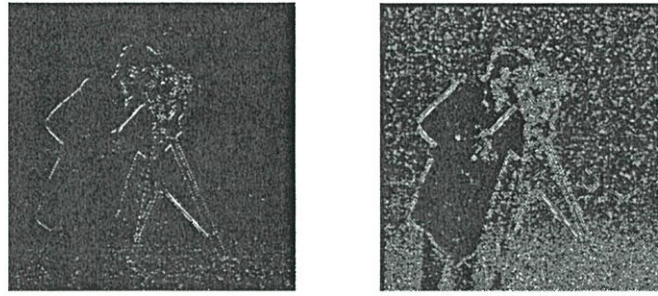
d) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้เมตริกขนาด 3 x 3

e) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้เมตริกขนาด 5 x 5

f) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้เมตริกขนาด 8 x 8

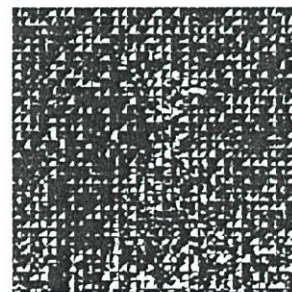
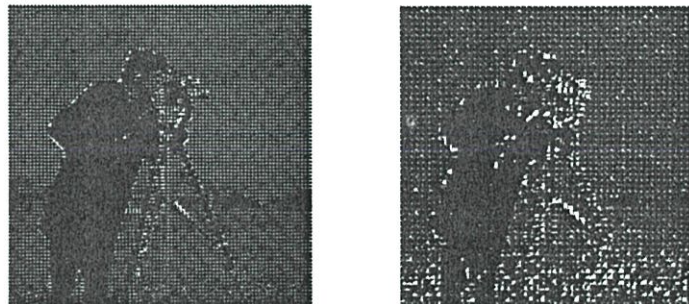


รูปที่ 5.12 ภาพมาตรฐานชื่อ Camera man ที่ถูกใช้เป็นสัญญาณอินพุตขนาด 256 x 256 จุด



a) Pascal Convolution Mask Size3x3 b) Pascal Convolution Mask Size5x5

c) Pascal Convolution Mask Size8x8



d) Pascal Transform Matrix Size3x3 b) Pascal Transform Matrix Size5x5

c) Pascal Transform Matrix Size8x8

รูปที่ 5.13 เปรียบเทียบผลของการแปลงแปลงปาสคาลฟิลเตอร์แบบ 2 มิติขนิคความถี่สูงผ่านและ
การแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติขนิคความถี่สูงผ่าน

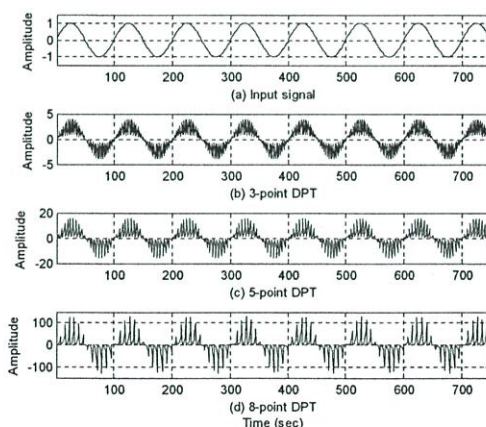
จากรูปที่ 5.13 ผลของการแปลงฟูริเยร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้ หน้ากากคอนโวลูชันฟูริเยร์ขนาด 3×3 ให้ผลที่ดีที่สุด สำหรับการใช้น้ำหนักคอนโวลูชันฟูริเยร์ขนาดอื่นๆที่ให้ผลออกมาไม่ดีนักเนื่องมาจากไม่ได้ทำการนอร์มอลไลซ์ค่าของสัมประสิทธิ์ ดังนั้นจึงทำให้ผลที่ได้จากการแปลงฟูริเยร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้ หน้ากากคอนโวลูชันฟูริเยร์ขนาดที่สูงกว่า 3×3 มีค่าของการรบกวนเพิ่มมากขึ้นเช่นกันสังเกตได้จากผลด้านบน

ดังนั้นเราสามารถประยุกต์ใช้การแปลงฟูริเยร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้ หน้ากากคอนโวลูชันฟูริเยร์ขนาด 3×3 สำหรับการตรวจสอบขอบภาพ (Edge detection) ได้ และเมื่อเปรียบเทียบกับผลการแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 2 มิติชนิดความถี่สูงผ่านซึ่งให้ผลที่ไม่สามารถคาดการณ์ได้

5.1.1.3 ผลการจำลองการทำงานชนิดความถี่ต่ำผ่าน

ขั้นที่ 1 การจำลองการทำงานนี้ได้สร้างสัญญาณไซน์ความถี่ 100 Hz ป้อนเข้าสู่ การแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่าน แสดงผลดังรูปที่ 5.14 โดยที่

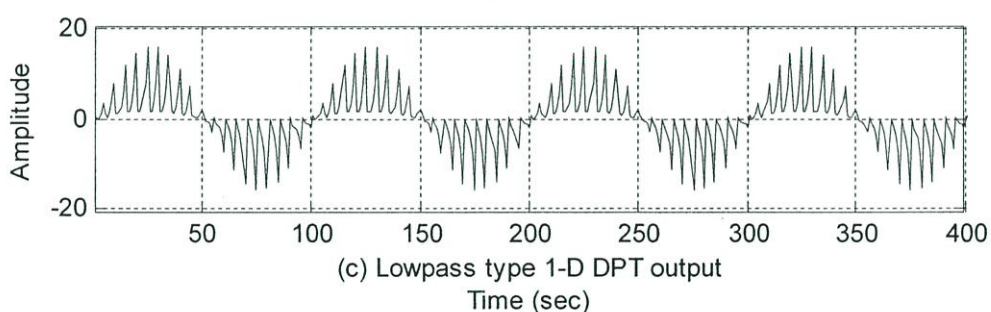
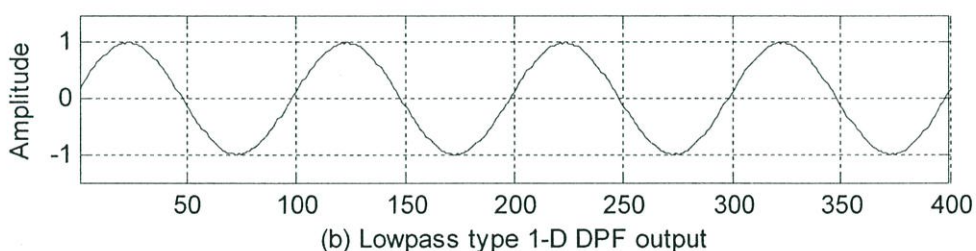
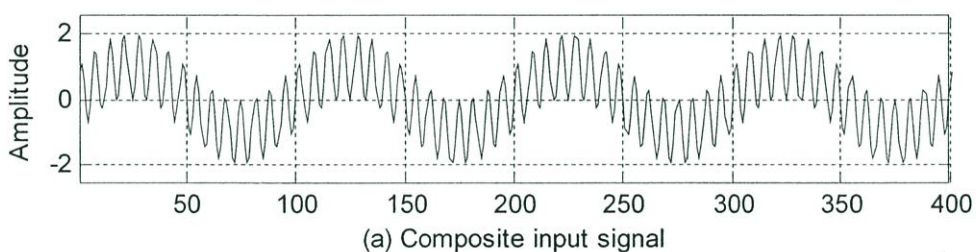
- สัญญาณอินพุต
- ผลจากการจำลองการทำงานของการแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่าน แบบ 3 จุด
- ผลจากการจำลองการทำงานของการแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่าน แบบ 5 จุด
- ผลจากการจำลองการทำงานของการแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่าน แบบ 8 จุด



รูปที่ 5.14 ผลจากการจำลองการทำงานของการแปลงดิสครีตฟูริเยร์เพิ่มเติมหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่าน

ขั้นที่ 2 การจำลองการทำงานนี้ได้สร้างสัญญาณรวม (Composite signal) จากสัญญาณไซน์ ความถี่ต่ำ 100 Hz ผสมกับไซน์ความถี่สูง 4000 Hz ป้อนเข้าสู่ วงจรการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่สูงผ่านแบบ 5 จุด หรือ 4th order เปรียบเทียบกับการแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่าน โดยทดลองทำที่ 4th order แสดงผลดังรูปที่ 5.15 โดยที่

- สัญญาณรวมอินพุต
- ผลจากการจำลองการทำงานของการแปลงปาสคาลฟิลเตอร์ชนิดความถี่ต่ำผ่าน โดยทดลองทำที่ 4th order
- ผลจากการจำลองการทำงานของการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่านแบบ 5 จุด หรือ 4th order



รูปที่ 5.15 ผลเปรียบเทียบการจำลองการทำงานของการแปลงดิจิตอลปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่านแบบ 5 จุด หรือ 4th order กับ การแปลงปาสคาลฟิลเตอร์ชนิดความถี่ต่ำผ่าน โดยทดลองทำที่ 4th order

จากรูปที่ 5.15 จะเห็นได้ว่าผลของการแปลงปาสคาลฟิลเตอร์ชนิดความถี่สูงผ่านโดยทดลองทำที่ 4th order สามารถกรองความถี่ต่ำจากสัญญาณรวมที่ป้อนเข้าไปทดสอบได้ โดยวิธีการของการคอนโวลูชัน (Convolution method) แต่การแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 1 มิติ ชนิดความถี่ต่ำผ่านแบบ 5 จุด หรือ 4th order แบบตรงๆ จะไม่สามารถกรองความถี่ต่ำจากสัญญาณรวมที่ป้อนเข้าไปทดสอบได้

ขั้นสุดท้าย จะแสดงผลของการแปลงแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่าน และการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยป้อนสัญญาณอินพุตเป็นภาพขนาด 256 x 256 จุด ดังแสดงในรูปที่ 5.12 และผลที่ได้จากการแปลงแสดงในรูปที่ 5.16 โดยที่

a) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 3 x 3

b) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 5 x 5

c) ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 8 x 8

d) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้เมตริกขนาด 3 x 3

e) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้เมตริกขนาด 5 x 5

f) ผลของการแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่าน โดยใช้เมตริกขนาด 8 x 8



a) Pascal Convolution Mask Size3x3 b) Pascal Convolution Mask Size5x5

c) Pascal Convolution Mask Size8x8



d) Pascal Transform Matrix Size3x3 b) Pascal Transform Matrix Size5x5

c) Pascal Transform Matrix Size8x8

รูปที่ 5.16 เปรียบเทียบผลของการแปลงแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่านและ การแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่าน

จากรูปที่ 5.16 ผลของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่ต่ำผ่าน จะให้ภาพที่สมูท (Smoothing) เมื่อเปรียบเทียบกับ การแปลงดีสครีตปาสคาลทรานสฟอร์มเต็มหน่วยแบบ 2 มิติชนิดความถี่ต่ำผ่านซึ่งให้ผลที่ไม่สามารถคาดการณ์ได้

5.2 บทสรุป

เนื้อหาในบทนี้ได้แสดงให้เห็นถึงผลจากการออกแบบและการจำลองการทำงานเพื่อทดสอบคุณสมบัติของแปลงปาสคาลทรานสฟอร์มหนึ่งมิติและสองมิติ ทั้งในรูปแบบชนิดความถี่สูงผ่านและความถี่ต่ำผ่าน จากผลแสดงให้เห็นว่าการแปลงปาสคาลฟิลเตอร์ในแบบหนึ่งมิติสามารถทำงานเป็นตัวกรองได้ดังผลการจำลองการทำงานข้างต้น และถ้าพิจารณาในการแปลงปาสคาลทรานสฟอร์มแบบสองมิติ คุณสมบัติที่ดีประการหนึ่งที่สังเกตเห็นได้จากผลการจำลองการทำงานคือ เราสามารถประยุกต์ใช้การแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้ หน้ากากคอนโวลูชันปาสคาลขนาด 3x3 สำหรับการตรวจสอบขอบภาพ (Edge detection) ข้อดีของการแปลงปาสคาลฟิลเตอร์แบบ 2 มิติคือ โครงสร้างทางฮาร์ดแวร์ไม่ซับซ้อนและยังปราศจากการใช้ตัวคูณ โดยสามารถสรุปจำนวนการใช้ตัวบวกหรือตัวลบในการสร้างวงจรของการแปลงปาสคาลทรานสฟอร์มแบบ N จุด ได้ดังตารางที่ 5.1

ตารางที่ 5.1 สรุปจำนวนการใช้ตัวบวกหรือตัวลบในการสร้างวงจรของการแปลงฟูรีเยร์แบบ N จุด

	Direct Matrix Multiplication	Pascal Matrix Multiplication	Butterfly Based DPT
M_N	N^2	$1 + \frac{N(N-3)}{2}$	0
A_N	$(N-1)N$	$\frac{(N-1)N}{2}$	$\frac{(N-1)N}{2}$

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการดำเนินการวิจัย

จากการนำเสนอการแปลงปาสคาลเต็มหน่วยสำหรับการประมวลผลสัญญาณเชิงเลข โดยรูปแบบของการดำเนินการของการแปลงปาสคาลเต็มหน่วยจะอาศัยปาสคาลเมตริกซ์เป็นตัวดำเนินการ ดังนั้นทำให้การแปลงสัญญาณจำเป็นต้องใช้จำนวนตัวคูณและตัวบวกเป็นจำนวนมาก ซึ่งขึ้นอยู่กับมิติของเมตริกซ์ตัวดำเนินการที่นำมาใช้ ได้ทำการสรุปการแยกองค์ประกอบของปาสคาลเมตริกซ์ให้อยู่ในรูปของเมตริกซ์เลขฐานสองให้ง่ายต่อการพิจารณาซึ่งประโยชน์ที่ได้จากการแยกองค์ประกอบของปาสคาลเมตริกซ์ให้อยู่ในรูปของเมตริกซ์เลขฐานสองจะทำให้การดำเนินการแปลงสัญญาณด้วยปาสคาลเมตริกซ์สามารถทำได้โดยปราศจากตัวคูณใช้เพียงตัวบวกหรือตัวลบสัญญาณเท่านั้น ทำให้โครงสร้างทางฮาร์ดแวร์ของวงจรที่ใช้ในการแปลงสัญญาณสามารถออกแบบได้อย่างมีประสิทธิภาพดังที่ได้แสดงในบทที่ผ่านมาแล้ว นอกจากนี้จะนำเสนอการประยุกต์ใช้งานการแปลงปาสคาลเต็มหน่วยสำหรับประมวลผลสัญญาณทั้งในกรณีสัญญาณ 1 มิติ และสัญญาณ 2 มิติ ซึ่งจากการประยุกต์ใช้งานนั้นเราจะเรียกชื่อการแปลงเป็นการแปลงปาสคาลฟิลเตอร์โดยเราแสดงให้เห็นถึง ผลการจำลองการทำงานชนิดความถี่สูงผ่าน และ ผลการจำลองการทำงานชนิดความถี่ต่ำผ่านทั้งในกรณีของสัญญาณ 1 มิติและสัญญาณ 2 มิติ จึงทำให้เราเห็นถึงจุดเด่นบางประการจากผลการจำลองการทำงานเช่นเราสามารถประยุกต์ใช้การแปลงปาสคาลฟิลเตอร์แบบ 2 มิติชนิดความถี่สูงผ่าน โดยใช้หน้ากากคอนโวลูชันปาสคาลขนาด 3x3 สำหรับการตรวจสอบขอบภาพ (Edge detection) ได้

6.2 แนวทางการพัฒนา

จากที่กล่าวไปข้างต้นว่างานวิจัยนี้ได้นำเสนอการแปลงปาสคาลเต็มหน่วยสำหรับการประมวลผลสัญญาณเชิงเลข และได้ทดสอบคุณสมบัติการเป็นตัวกรองทั้งแบบความถี่สูงผ่านและแบบความถี่ต่ำผ่านทั้งในกรณีของสัญญาณ 1 มิติและสัญญาณ 2 มิติ ดังนั้นจากคุณสมบัติข้างต้นสามารถนำไปพัฒนาหรือประยุกต์ใช้กับงานประมวลผลสัญญาณเชิงเลขในแบบอื่นๆ ได้

บรรณานุกรม

- [1] V. Biolkova and D. Biolek, "Generalized Pascal Matrix of First Order s-z Transforms," Proc. ICECS, Pafos Cyprus, 1999.
- [2] B. Psenicka, F. Garcia-Ugalde and A. Herrera-Camacho, "The Bilinear Z Transform by Pascal Matrix and Its Application in the Design of Digital Filters," IEEE Signal Processing Letters, Vol. 9, No. 11, pp. 368-370., November 2002.
- [3] B. Psenicka and F. Garcia-Ugalde, "Z Transform From Lowpass to Bandpass by Pascal Matrix," IEEE Signal Processing Letters, Vol. 1, No. 2, pp. 282-284., February 2004.
- [4] S. Chivapreecha, S. Sriyapong, S. Junnapiya and K. Dejhan, "Bilinear s-z with Frequency Transformation Using Pascal Matrix Operation," Proc. IEEE International Symposium on Communications and Information Technology (ISCIT2005), pp. 739-742., October 2005.
- [5] W. Mongkhonmalee, S. Chivapreecha, S. Tooprakai and K. Dejhan, "Biquad Digital Filter Design Using Pascal Matrix," Proc. International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2007), Vol. 1., pp. 21-22., July 2007.
- [6] M. F. Aburdene and T. J. Goodman, "The Discrete Pascal Transform and Its Applications," IEEE Signal Processing Letters, Vol. 12, No. 7, pp. 493-495., July 2005.
- [7] M. F. Aburdene, R. J. Kozick, R. S. Magargle, J. D. Maloney-Han and C. M. Coviello, "Discrete Polynomial Transform Representation Using Binary Matrices and Flow Diagrams," Proc. IEEE Int. Conf. Acoust., Speech Signal Processing., Vol. 2, pp. 1141-1144., 2001.
- [8] A. Edelman and G. Strang, "Pascal Matrices," Am. Math Mon., pp. 189-197., 2004.
- [9] A. N. Skodras, "Fast Discrete Pascal Transform," Electronics Letters, Vol. 42, No. 23, November 2006.
- [10] T. J. Goodman and M. F. Aburden, "A hardware implementation of the discrete Pascal transform for image processing," Proc. of SPIE-IT&T Electronic Imaging, SPIE vol. 6064, pp.60640 H-1 – 6064 H-8, 2006.
- [11] S. Chivapreecha, U. Nithirochananont, and K. Dejhan, "Invertigation of frequency characteristic in discrete Pascal transform and Its applications," Submitted to The 4th

International Colloquium on Signal Processing and Its Application (CSPA 2008), Kuala Lumpur, Malaysia, 2008.

ภาคผนวก

ภาคผนวก ก.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Sorawat Chivapreecha, **Narison Ronnarongrit**, Surapan Yimman, Chusit Pradabpet and Kobchai Dejhan,

“**Multiplierless Digital PID Controller Using FPGA**” International Conference on Control, Automation and Systems (ICCAS 2004), August 25-27, 2004, The Shangri-La Hotel, Bangkok, Thailand.

2. **Narison Ronnarongrit**, Sorawat Chivapreecha and Kobchai Dejhan,

“**Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization**” The 4th International Colloquium on Signal Processing and its Applications (CSPA 2008), March 7-9, 2008 , Kuala Lumpur, Malaysia.



ICCAS '04

ICCAS 2004

Final Program & Abstract Book

International Conference on Control, Automation and Systems

August 25-27, 2004

The Shangri-La Hotel, Bangkok, Thailand

Organized by:

Research Center for Communications and Information Technology, King Mongkut's Institute of Technology Ladkrabang

The Institute of Control, Automation, and Systems Engineers, Korea



ICASE

In association with:



Multiplierless Digital PID Controller Using FPGA

Sorawat Chivapreecha*, Narison Ronnarongrit*, Surapan Yimman**, Chusit Pradabpet*** and Kobchai Dejhan*

*Faculty of Engineering and Research Center for Communication and Information Technology King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand Tel: +66-2-326-4238, +66-2-326-4242, Fax. +66-2-326-4554; E-mail : sorawat@telecom.kmitl.ac.th, kobchai@telecom.kmitl.ac.th **King Mongkut's Institute of Technology North Bangkok, Bangkok 10800, Thailand Tel. +66-2-913-2500; E-mail : sym@kmitnb.ac.th ***Rajabhat Institute Phranakhon Si Ayutthaya, Phranakhon Si Ayutthaya 13000, Thailand Tel. +66-3532-2076; E-mail : c_pradabpet@hotmail.com

Abstract: This paper proposes a design and implementation of multiplierless digital PID (Proportional-Integral-Derivative) controller using FPGA (Field Programmable Gate Array) for controlling the speed of DC motor in digital system. The multiplierless PID structure is based on Distributed Arithmetic (DA). The DA is an efficient way to compute an inner product using partial products, each can be obtained by using look-up table. The PID controller is designed using MATLAB program to generate a set of coefficients associated with a desired controller characteristics. The controller coefficients are then included in VHDL (Very high speed integrated circuit Hardware Description Language) that implements the PID controller onto FPGA. MATLAB program is used to activate the PID controller, calculate and plot the time response of the control system. In addition, the hardware implementation uses VHDL and synthesis using FLEX10K Altera FPGA as target technology and use MAX+plusII program for overall development. Results in design are shown the speed performance and used area of FPGA. Finally, the experimental results can be shown when compared with the simulation results from MATLAB.

Keywords: PID Controller, Digital Control, Distributed Arithmetic FPGA

1. INTRODUCTION

Generally, an implementation of digital PID controller is widely used microprocessor or microcontroller which has disadvantage in speed of operations because the operations depend on software which has the sequence of operations and multiplication command needs many machine cycles for execute. Also, FPGA-based digital PID controller is proposed because the operations of FPGA are hardware concurrent operations. However, FPGA-based digital PID controller still needs multipliers for computation. These multipliers will decrease the speed of processing time since the multiplying stage is consumption process and use large silicon area in VLSI design. These multiplications can change to DA architecture, DA architecture was first proposed by Peled and Liu in 1974 [1]. The DA is a direct method for sum of products operations, partial products and can pre-compute by difference equation and stored in look-up table containing in memory, input signals can be used for addressing. The product can be computed by scaling accumulate of partial products from memory, therefore, the multipliers don't necessary for this method, the proposed digital PID controller will be multiplierless.

2. PRINCIPLE OF DIGITAL PID CONTROLLER

Considering a block diagram of the digital control system

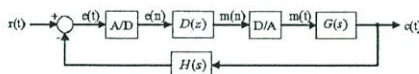


Fig.1 Block diagram of the digital control system

Fig.1, r(t) is the reference input or called set-point, c(t) is the control output, D(z) is the digital controller, G(s) is the plant transfer function and H(s) is the sensor transfer function. A digital PID controller can be derived as follows.

In continuous-time system, a transfers function for PID controller [2-3] is

m(t) = k_p e(t) + k_i \int e dt + k_d \frac{de}{dt} (1)

Where

- k_p is gain of proportional controller
k_i is gain of integral controller
k_d is gain of derivative controller
e(t) is error signal
m(t) is output signal

Use Laplace transform in equation (1) will give.

M(s) = k_p E(s) + \frac{k_i}{s} E(s) + k_d s (2)

Also, the transfer function of PID controller is

D(s) = \frac{M(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s (3)

Transform Eq.(3) to digital domain and will be obtained the transfer function of digital PID controller as follows.

D(z) = K_p + K_i \frac{T}{2} \frac{z+1}{z-1} + \frac{K_d}{T} \frac{z-1}{z} (4)

Eq. (4) can be realized to be direct form I structure by

D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} (5)

Normally, for digital PID controller b_2 = 0 and b_1 = -1 with

a_0 = K_p + \frac{K_i T}{2} + \frac{K_d}{T}
a_1 = -K_p + \frac{K_i T}{2} - \frac{2K_d}{T}
a_2 = \frac{K_d}{T}

Where K_p, K_i and K_d are the proportional, integral and derivative parameters of digital PID controller, and T is the sampling period. Fig.2 shows the direct form I structure of digital PID controller that correspond to Eq. (5).

All of possible values containing in ROM equal to 8 values, each value will be convert to 16 bit two's complement format before stored in ROM, addresses for tapping ROM come form sequence of bit level inputs $e_k(n)$, $e_k(n-1)$ and $m_k(n-1)$ respectively. Table 1 shows evaluation of values contention ROM

Table 1 Evaluation of values content in ROM

Address			Values contention In ROM
0	0	0	0
0	0	1	1
0	1	0	a_1
0	1	1	$a_1 + 1$
1	0	0	a_0
1	0	1	$a_0 + 1$
1	1	0	$a_0 + a_1$
1	1	1	$a_0 + a_1 + 1$

Also, the proposed multiplierless digital PID controller be shown in Fig. 4

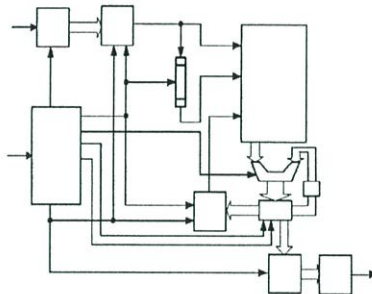


Fig. 4 Architecture of proposed multiplierless digital PID controller

The operation can be described by 5 steps as follow;

1. A/D that controlled by signal sc will be converted analog signal $e(t)$ to 8 bit digital signal $e(n)$
2. signal lr will be loaded input data to PISO (parallel in serial out shift register)
3. signal clk will be shifted data in each shift register, output of each shift register used for each ROM addressing, output of ROM will be accumulated by scaling accumulator that control by signal s/a , results will be loaded to ACC by signal $lacc$.
4. clk will be shifted continue and repeat in step 3 until shift to last bit, output of ROM will be subtracted from ACC, the results will be loaded to buffer by signal lr for D/A, respectively.
5. clear ACC using signal $clacc$ and repeat in step 1-5, respectively.

The timing diagram of control unit is used for the proposed hardware architecture can be shown in Fig. 5.

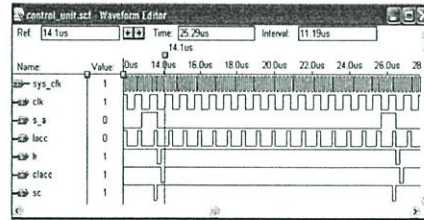


Fig.5 Timing diagram of control unit

4. SYNTHESIS AND EXPERIMENTAL RESULTS

An EPF10K10LC84-4 device in FLEX10K device family used for circuit synthesis. Device summary of multiplierless digital PID controller is shown in Fig. 6

```

** DEVICE SUMMARY **
Chip/      Input Output Bidir  Memory  Memory  Lcs
POF       Device   Pins  Pins  Pins  Bits  % Utilized  LCs  % Utilized
pid       EPF10K10LC84-4  17   17   0   128   2 %      228  39 %
User Pins:      17   17   0
    
```

Fig. 6 Device Summary

From device summary, 17 input pins used for 16 bit input data and one for system clock, 17 output pin for 16 bit output data and one for signal sc , 128 memory bits used for implementing ROM. Others component which using VHDL for design will use 228 LCs (logic cells) for implementing. Timing summary shows the maximum frequency of synthesized circuits is show in Fig. 7.

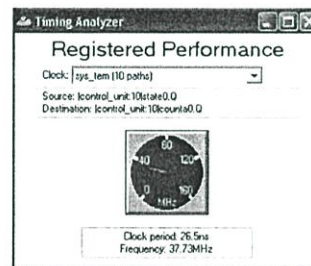


Fig. 7 Timing Summary

For experiment, the first test was to run the motor without controller. The output speed was only about half of the desired output is about 6.25 rps or 375 rpm. Fig. 7 shows the simulation result and Fig. 8 shows the experimental result.

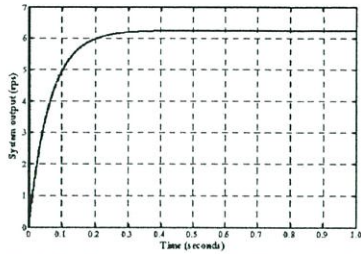


Fig.7 Simulation result in case of without controller

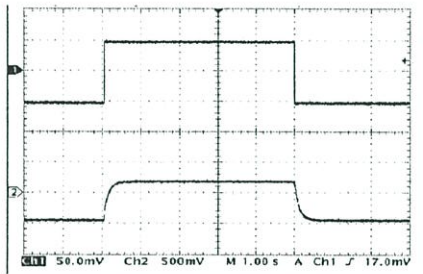


Fig. 8 Experiment result in case of without controller

Fig.8, motor speed at 375 rpm corresponding with output voltage about 750 mV. Fig. 9 and Fig. 10 shows the simulation result and experimental result of the second test that performed with a proposed multiplierless digital PID controller added.

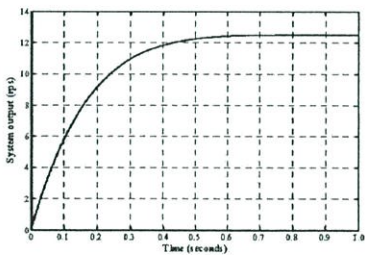


Fig. 9 Simulation result in case of with controller

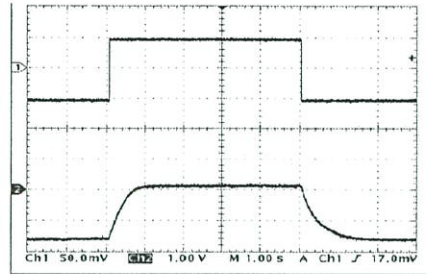


Fig. 10 Experimental result in case of with controller

Fig. 9, this achieved a steady state output speed of 750 rpm as desired output speed which its about 2 times of the first test. Also, the motor speed at 750 rpm will be corresponded to output voltage about 1.5V as shown in Fig. 10. However, the rise-time is longer with the controller added. The rise-time can be reduced by increasing K , but it possible to have an overshoot.

5. CONCLUSION

The proposed method in this paper can obtain the multiplierless digital PID controller using distributed arithmetic realization, The hardware is satiable for implementing on FPGA, high processing speed and reduce power consumption when compared with using discrete components in implementation. Form experimental results when compared with simulation results, it can be ensured that the proposed hardware can operated correctly.

REFERENCES

- [1] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. ASSP.*, vol. ASSP-22, pp. 456-462, Dec 1974.
- [2] B. C. Kuo, *Automatic control system*, Prentice-Hall, 1995.
- [3] F. Nekoogaran and G. Moriarty, *digital control using digital signal processing*, Prentice-Hall, 1998.
- [4] C.S. Barrns, "Digital filter structure described by disdtributed arithmetic," *IEEE Trans. Circuits and systems*, vol. CAS-24, No.12, pp. 674-680, Dec 1977.
- [5] S.A. White, "Application of distributed arithmetic to digital signal processing : A tutorial review," *IEEE ASSP. Magazine*, vol.6, No.3, pp. 4-13, July 1989.

International Colloquium on Signal Processing and its Applications, March 7-9, 2008, Kuala Lumpur, Malaysia



Technical Programme

The 4th International Colloquium on
Signal Processing and its Applications
(CSPA 2008)

7-9 March 2008

Kuala Lumpur, Malaysia.

Organizer

*Advanced Signal Processing Research Group
Faculty of Electrical Engineering
Universiti Teknologi MARA
Shah Alam Malaysia*

Technical co-sponsor

*IEEE UITM SB, Malaysia Section
IEEE Signal Processing, Malaysia Section*

Editors

*Mohd Nasir Taib
Mohd Hezri Fazalul Rahiman
Nooritawati Md Tahir
Ramlil Adnan
Mohd Faiz Hj Mohd Said
Zuhaina Zakaria
Abdul Hadi Abdul Razak*



Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization

Narison Ronnarongrit, Sorawat Chivapreecha and Kobchai Dejhan
 Faculty of Engineering and Research Center for Communication and Information Technology
 King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
 Tel: +66-2326-4238, +66-2326-4242, Fax: +66-2326-4554
 E-mail: sorawat@telecom.kmitl.ac.th, kobchai@telecom.kmitl.ac.th

Abstract—This paper presents the discrete Pascal transform for digital signal processing, the operations of discrete Pascal transform is based on Pascal transform matrix which perform as the operator for signal transformation. The Pascal transform matrix that is used in the discrete Pascal transform which is proposed in this paper is divided to 2 types. Generally, the matrix transformation is necessary to have many multipliers and adders, which is depending on the dimension of used matrix operator. The factorization of Pascal matrix into binary matrices will allow the transformation using Pascal matrix to operate without multipliers and only adders are used. Therefore, the hardware realization for transformation circuits can be efficient designed by using the butterfly unit for discrete Pascal transform to establish the whole structure. Moreover, the hardware structure of two dimensional discrete Pascal transform will be proposed for 2-D signal processing case.

I. INTRODUCTION

The discrete Pascal transform (DPT) that was presented by [1] is one of many discrete transforms such as DFT (Discrete Fourier Transform), FFT (Fast Fourier Transform), DCT (Discrete Cosine Transform), DWT (Discrete Wavelet Transform), etc. Most of these discrete transforms can be operated in the form of matrix operation as same as the DPT. The interesting problem is that matrix operation consumes many multiplications and many adders for implementation, all of these depend on the size or dimension of matrix operator.

With some properties of Pascal transform matrix where the elements in matrix operator are the binomial coefficients and related to the Pascal's triangle, we can factorize the Pascal transform matrix into binary (1,0,-1) matrices. Then we can represent Pascal transform matrix in the form of binary (1,0,-1) matrices, the transformed output, which is computed from Pascal transform matrix operation, can be computed without any multipliers. Only adders are used by using an efficient hardware structure that is obtained from these factorized binary (1,0,-1) matrices [2,4].

The Pascal transform matrix in this paper is divided into two types that are called highpass type and lowpass type, respectively [3]. In [4] proposes only the efficient hardware realization of highpass type DPT that uses highpass type Pascal transform matrix, but this paper also proposes the efficient hardware realization of lowpass type DPT. Moreover, we also propose the elimination matrix that is used for formulating the factorized binary (1,0,-1) matrices both highpass and lowpass

type which do not appear in [2,4]. Finally the efficient hardware realization of two-dimensional (2-D) DPT can be shown in the form of one-dimensional (1-D) DPT structure, and from the DPT hardware structure we will show the improvement of discrete Pascal filter (DPF) in both 1-D and 2-D case and both highpass and lowpass type.

II. BASIS FUNCTION OF DISCRETE PASCAL TRANSFORM

The discrete Pascal transform (DPT) X of the one-dimensional (1-D) signal vector x is defined as

$$X = Px \quad (1)$$

where P is the Pascal transform matrix size $N \times N$, and x is input signal vector, X is transformed output signal vector size $N \times 1$. In this paper, we divide the type of Pascal matrix into 2 types by basis function of the Pascal transform matrix. The first type, we call highpass type DPT which is the same as proposed in [1-4], and another type, we call lowpass type DPT. The reason for calling those type names was mentioned in [3].

A. Basis Function of Highpass Type DPT

From eq. (1), the Pascal matrix P in the case of highpass type DPT has the basis function as follows,

$$P_k^{(HP)}(x) = P^{(HP)}(x, k) = \frac{(-1)^k x^k}{k!} = (-1)^k \binom{x}{k} \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (2)$$

B. Basis Function of Lowpass Type DPT

In the same manner as eq. (2), the basis function of lowpass type DPT can be shown as

$$P_k^{(LP)}(x) = P^{(LP)}(x, k) = \frac{x^k}{k!} = \binom{x}{k} \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (3)$$

where N is dimension of Pascal transform matrix and

$$\binom{x}{k} = \frac{x!}{k!(x-k)!}$$

that we call binomial coefficient and is related to the Pascal's triangle. The only difference between eq. (2) and eq. (3) is that the term $(-1)^k$ appears only in eq. (2). This term will make alternating the sign of the columns of highpass type Pascal transformation matrix. Both eq. (2) and eq. (3), $P(x, k)$ are the element in Pascal transformation matrix P as shown in eq. (1) depending on the type needed, x^k is the row index and k^k is the column index.

The function $x^{(k)}$ is called falling factorial powers which can be shown as follows,

$$x^{(k)} = x(x-1)(x-2)\dots(x-k+2)(x-k+1) = \frac{x!}{(x-k)!}; k \geq 1 \quad (4)$$

where $x^{(0)} = 1$ and k is also called the order of basis function.

For example, we can show the 3rd order basis function as follows,

In the case of highpass type has 4 polynomials are:

$$\begin{aligned} F_0^{(k)}(x) &= 1 \\ F_1^{(k)}(x) &= -x \\ F_2^{(k)}(x) &= \frac{1}{2}x(x-1) = -\frac{1}{2}(x-1)F_1^{(k)}(x) \\ F_3^{(k)}(x) &= -\frac{1}{6}x(x-1)(x-2) = -\frac{1}{3}(x-2)F_2^{(k)}(x) \end{aligned}$$

and we can summarize to the recurrence formula by

$$F_{k+1}^{(k)}(x) = -\frac{1}{k+1}(x-k)F_k^{(k)}(x) \quad (5)$$

In the case of lowpass type has also 4 polynomials are:

$$\begin{aligned} F_0^{(k)}(x) &= 1 \\ F_1^{(k)}(x) &= x \\ F_2^{(k)}(x) &= \frac{1}{2}x(x-1) = \frac{1}{2}(x-1)F_1^{(k)}(x) \\ F_3^{(k)}(x) &= \frac{1}{6}x(x-1)(x-2) = \frac{1}{3}(x-2)F_2^{(k)}(x) \end{aligned}$$

So, the recurrence formula can be shown as,

$$F_{k+1}^{(k)}(x) = \frac{1}{k+1}(x-k)F_k^{(k)}(x) \quad (6)$$

Therefore, we can use these 4 polynomials that are obtained from each basis function to generate the element in Pascal transform matrix both of highpass and lowpass type as follows,

$$P^{**} = [P^{(k)}(x,k)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -3 & 3 & -1 \end{bmatrix}, \quad P^{**} = [P^{(k)}(x,k)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{bmatrix}$$

The basic properties of these P matrices are the following.

1. The elements of the first column are equal to 1.
2. All matrices are lower triangular.
3. For highpass type $P^{(k)}$ matrix; the sum of the elements of each row (except the first row) is equal to zero.
For lowpass type $P^{(k)}$ matrix; the sum of the elements of each row is equal to 2^k ; x^k is the row index
4. For highpass type $P^{(k)}$ matrix; the inverse $[P^{(k)}]^{-1}$ matrix is equal to the forward $P^{(k)}$ matrix, or $[P^{(k)}]^{-1} = P^{(k)}$.
For lowpass type $P^{(k)}$ matrix; the inverse $[P^{(k)}]^{-1}$ matrix is equal to $(-1)^{k+k}$ multiply to the forward $P^{(k)}$ matrix, or $[P^{(k)}]^{-1} = (-1)^{k+k} P^{(k)}$

III. EFFICIENT HARDWARE REALIZATION OF DPT AND ITS BUTTERFLY UNIT

In [1] proposes only the basic of DPT in the form of matrix operation and does not focus on hardware realization. In [2] proposes the hardware implementation of DPT but the method that used to decompose or factorize the Pascal transform matrix into binary (1,0,-1) matrices is not as compact as the one proposed in [4] which can give the very efficiently hardware structure with all compact factorized binary (1,0,-1) matrices. This section will show the method to factorize the Pascal transform matrix both highpass and lowpass type into binary (1,0,-1) matrices using elimination matrix where some details do not appear in [4].

A. Pascal Matrix Factorization to Binary (1,0,-1) Matrices

Consider matrix $P^{(k)}$ that has the dimension $N - 2$, which can be denoted to

$$P_1^{(k)} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}$$

and from equation of highpass type 2-point DPT can give

$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Therefore,

$$X_0 = x_0 \quad (7a)$$

$$X_1 = x_0 - x_1 \quad (7b)$$

from eq. (7a) and eq. (7b), we can make the data flow graph as below,

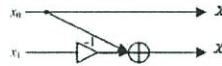


Fig. 1 A Butterfly unit of highpass type DPT

The data flow graph that is obtained from eq. (7a) and (7b), we call a butterfly unit of highpass type DPT. In practical, we do not need the constant multiplier (-1) since we can design the subtractor instead of adder. The reason to draw data flow graph as shown in Fig.1 is that we want to make distinguish structure compared with a butterfly unit of lowpass type DPT and, in this paper, both subtractor and adder will be classed all the same.

Consider again to $P^{(k)}$ that has the dimension $N - 2$,

$$P_1^{(k)} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

and from equation of lowpass type 2-point DPT can give

$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Therefore,

$$X_0 = x_0 \quad (8a)$$

$$X_1 = x_0 + x_1 \quad (8b)$$

from eq. (8a) and eq. (8b), we can make the data flow graph as below,

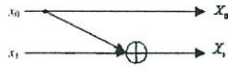


Fig. 2 A Butterfly unit of lowpass type DPT

The data flow graph that is obtained from eq. (8a) and (8b), also called a butterfly unit of lowpass type DPT. The butterfly unit shown in Fig. 1 and Fig. 2 will be the basic element that is used to establish any N -point DPT.

The method to factorize the Pascal transform matrix into binary (1,0,-1) matrices and to obtain the efficient hardware structure of any N -point DPT using Gaussian elimination [5] can be described as follows,

Lowpass type DPT

The lowpass type elimination matrix $E^{(L)}$ has entries $E_{xx}^{(L)} = -1$ and $E_{xx}^{(L)} = -1$ [5]. Assume $P_4^{(L)}$ for consideration,

First Step for matrix elimination:

$$E_4^{(L)} P_4^{(L)} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & & \\ & 0 & P_2^{(L)} & \end{bmatrix}$$

Second Step:

$$\begin{bmatrix} 1 & 1 & \\ 0 & E_1^{(L)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P_2^{(L)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \\ & 0 & 1 & 0 \\ & & 0 & P_2^{(L)} \end{bmatrix}$$

Third Step:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(L)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & P_2^{(L)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [I]$$

So that,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(L)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_1^{(L)} \end{bmatrix} [E_4^{(L)}] [P_4^{(L)}] = [I] \quad (9)$$

$$\therefore P_4^{(L)} = [E_4^{(L)}]^{-1} \begin{bmatrix} 1 & 0 \\ 0 & E_1^{(L)} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(L)} \end{bmatrix}^{-1} \quad (10)$$

Therefore, we can factorize the lowpass type Pascal transform matrix $P_4^{(L)}$ into binary (1,0) matrices as

$$P_4^{(L)} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

This method of matrix decomposition or factorization is the same as the one used in [2] which cannot give the compact form of binary matrices. Using the property of inverse of lowpass type Pascal transform matrix as mentioned in section II.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(L)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_1^{(L)} \end{bmatrix} [E_4^{(L)}] = (-1)^{m+n} P_4^{(L)} \quad (11)$$

we can show that

$$P_4^{(L)} = (-1)^{m+n} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(L)} \end{bmatrix} \times (-1)^{m+n} \begin{bmatrix} 1 & 0 \\ 0 & E_1^{(L)} \end{bmatrix} \times (-1)^{m+n} [E_4^{(L)}] \quad (12)$$

Finally, we can factorize $P_4^{(L)}$ into binary (1,0) matrices by this method as

$$P_4^{(L)} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (13)$$

And this is the compact form of binary matrices that is factorized from lowpass type Pascal transform matrix.

Highpass type DPT

The highpass elimination matrix $E^{(HP)}$ has entries $E_{xx}^{(HP)} = -1$ except $E_{00}^{(HP)} = 1$ and $E_{xx}^{(HP)} = 1$. Assume $P_4^{(HP)}$ for consideration,

First Step for matrix elimination:

$$E_4^{(HP)} P_4^{(HP)} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & & \\ & 0 & P_2^{(HP)} & \end{bmatrix}$$

Second Step:

$$\begin{bmatrix} 1 & 1 & \\ 0 & E_1^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P_2^{(HP)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \\ & 0 & 1 & 0 \\ & & 0 & P_2^{(HP)} \end{bmatrix}$$

Third Step:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(HP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P_2^{(HP)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [I]$$

So that

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_3^{(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_2^{(2)} \end{bmatrix} [E_4^{(2)}] [P_4^{(2)}] = [I] \quad (14)$$

$$\therefore P_4^{(2)} = [E_4^{(2)}]^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_3^{(2)} \end{bmatrix}^{-1}$$

Therefore, we can factorize the highpass type Pascal transform matrix $P_4^{(2)}$ into binary (1,0,-1) matrices as

$$P_4^{(2)} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Similar to the case of lowpass type, this form of binary matrices are not compact form. Using the property of inverse of highpass type Pascal transform matrix is equal to its matrix. Also, we can show that

$$P_4^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_3^{(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_2^{(2)} \end{bmatrix} [E_4^{(2)}] \quad (15)$$

Finally, we can factorize $P_4^{(2)}$ into binary (1,0,-1) matrices as follows,

$$P_4^{(2)} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (16)$$

and this is the compact form of binary matrices that is factorized from highpass type Pascal transformation matrix and is the same as binary (1,0,-1) matrices that is used in fast DPT[4].

The above method can be used to factorize any dimension N of both type of Pascal transform matrix into binary matrices with efficiency.

B. DPT Flow Graph

From previous, we can factorize both of highpass and lowpass type Pascal transform matrix into binary matrices. The Pascal transform matrix in dimension $N = 4$ or P_4 can be factorized to three binary matrices. We can summarize in general form as follows,

$$P = \prod_{l=1}^L [q_{x,l}] \quad ; x, k = 0, 1, 2, \dots, N-1 \quad (17)$$

$$= [q_{x,1}] [q_{x,2}] \dots [q_{x,L}]$$

where $[q_{x,k}]_l$ is binary matrix at stage l^{th} .

Therefore, from eq. (13)

$$P_4^{(2)} = [q_{x,1}^{(2)}] [q_{x,2}^{(2)}] [q_{x,3}^{(2)}]$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (18)$$

and from eq. (16)

$$P_4^{(2)} = [q_{x,1}^{(2)}] [q_{x,2}^{(2)}] [q_{x,3}^{(2)}]$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (19)$$

To simplify the method to factorize the Pascal transform matrix both lowpass and highpass type into binary matrices, we propose steps to create each binary matrix as follows,

For Lowpass type:

1. The main diagonal entries are '1' for all stage.
2. All entries in upper triangle are equal to '0'.



3. At stage l^{th} , consider for the rowth $x = N-1, N-2, \dots, l$ by

$$\boxed{\text{for each } x \Rightarrow \text{if } q_{x,x} = -1 \text{ then } q_{x,x-1} = -1}$$

4. Otherwise are '0'.

For Highpass type:

1. Consider in the main diagonal,

At stage l^{th} , consider for the rowth $x = N-1, N-2, \dots, l$ by

$$\boxed{q_{x,x} = -1, \text{ other are '1'}}$$

or consider in this manner

for stage $l = N - 1$

$Diag([q_{x,N-1}]) = [1 \ 1 \ 1 \ 1]$; '1' has $N-1$ elements others are '-1'

:

for stage $l = 2$

$Diag([q_{x,2}]) = [1 \ 1 \ -1 \ -1 \ \dots \ -1]$; '1' has 2 elements others are '-1'

for stage $l = 1$

$Diag([q_{x,1}]) = [1 \ -1 \ -1 \ -1 \ \dots \ -1]$; '1' has 1 element others are '-1'

2. All entries in upper triangle are '0'.

3. At stage l^{th} , consider in each row,

$$\boxed{\text{if } q_{x,x} = -1 \text{ then } q_{x,x-1} = -1}$$

4. Otherwise are '0'

The data flow graph showing hardware realization of DPT can be created from factorized binary matrices and its structure consists of many butterfly units which are the basic computational element in DPT.

The lowpass type 4-point DPT can be computed by

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} \Rightarrow \begin{matrix} X_0 = x_0 \\ X_1 = x_0 + x_1 \\ X_2 = x_0 + 2x_1 + x_2 \\ X_3 = x_0 + 3x_1 + 3x_2 + x_3 \end{matrix} \quad (20)$$

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

Therefore, the lowpass type 4-point DPT flow graph can be shown as in Fig. 3

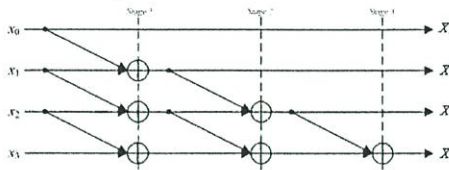


Fig. 3 The lowpass type 4-point DPT flow graph

We use 6 lowpass type butterfly units for realizing lowpass type 4-point DPT. This structure is suitable for hardware implementation and very easy to implement. Especially, in VLSI design this structure is very compatible to the pipeline technique by placing the pipeline register on each stage as shown in Fig. 3 to enhance the speed of computation of DPT. The highpass type 4-point DPT can be computed by

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -3 & 3 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \begin{bmatrix} X_0 - x_0 \\ X_1 - x_0 - x_1 \\ X_2 - x_0 - 2x_1 + x_2 \\ X_3 - x_0 - 3x_1 + 3x_2 - x_3 \end{bmatrix} \quad (21)$$

Therefore, the highpass type 4-point DPT flow graph can be shown in Fig. 4

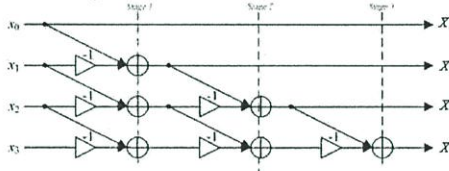


Fig. 4 The highpass type 4-point DPT flow graph

This type of DPT also uses 6 highpass type butterfly units for realization.

C. Computational Complexity of DPT

If we consider the number of operation of DPT directly from definition matrix multiplication in eq. (1), the number of multiplications (M_N) and the number of additions (or subtractions in the case of highpass type) (A_N) will be

$$M_N = N^2 \text{ and } A_N = (N-1)N$$

However, the Pascal transform matrix is a lower triangular matrix where all entries in the first column are '1' and the entries in main diagonal are equal to '1' or '-1'. Also, no multiplications are needed for operating those entries. Then,

the number of multiplications and number of additions would be

$$M_N = 1 + \frac{N(N-3)}{2} \text{ and } A_N = \frac{(N-1)N}{2}$$

Finally, from DPT flow graph (butterfly based) which is based on Pascal matrix factorization into binary matrices can compute the transformed results without multiplications. Thus

$$M_N = 0 \text{ and } A_N = \frac{(N-1)N}{2}$$

The summary of computational complexity of any N -point DPT can be shown as in Table I.

	Direct Matrix Multiplication	Pascal Matrix Multiplication	Butterfly Based DPT
M_N	N^2	$1 + \frac{N(N-3)}{2}$	0
A_N	$(N-1)N$	$\frac{(N-1)N}{2}$	$\frac{(N-1)N}{2}$

IV. TWO-DIMENSIONAL DPT AND ITS IMPROVEMENT TO DPF STRUCTURE

The two-dimension (2-D) DPT can be computed by [1-3]

$$X = P x P^T \quad (22)$$

where P is the Pascal transform matrix and its transpose P^T , X is transformed output of an input image matrix x, all of matrix size $N \times N$.

for example, we will consider highpass type 2-D DPT with $N = 3$

$$X = [P^{(HP)}]^{-1} [x] [P^{(HP)}]^T = \begin{bmatrix} 1 & & \\ & 1 & -1 \\ & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & -1 \\ 1 & & 1 \end{bmatrix} \quad (23)$$

Thus,

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \\ X_{20} & X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & 2x_{10} + x_{20} & x_{11} + x_{21} & x_{02} & x_{12} & 2x_{12} + x_{22} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ & 1 & 2 \\ & & 1 \end{bmatrix} \quad (24)$$

This matrix is a result of row operation for each input row based on the 2-D operation (2-D DPT). For each input column vector, the result is in $[x_{ij}]$.

Finally,

$$\begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} \end{bmatrix} = \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} & x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} & x_{20} & x_{21} & x_{22} \\ x_{20} & 2x_{10} + x_{20} & x_{11} + x_{21} & x_{20} & 2x_{10} + x_{20} & x_{11} + x_{21} \\ x_{02} & x_{12} & 2x_{12} + x_{22} & x_{02} & x_{12} & 2x_{12} + x_{22} \end{bmatrix} \quad (25)$$

We can summarize 2-D DPT flow graph with $N = 3$ as in Fig. 5

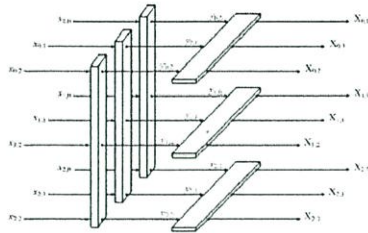
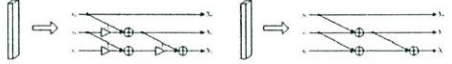


Fig. 5 The 2-D DPT flow graph

where the processing elements of highpass (left) and lowpass (right) type are as below, respectively



The DPF (Discrete Pascal Filter), which is mentioned in our another paper [3], can be formulated from the DPT by making the relation between the inputs and focus only on 1 output. Using DPF, the computational operation will be changed from matrix multiplication in DPT to convolution in DPF. With a little modification, we can obtain the hardware realization of 1-D DPF of both highpass type and lowpass type as follows

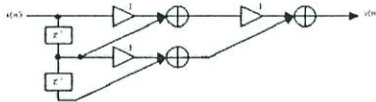


Fig. 6 The 2nd order highpass type 1-D DPF

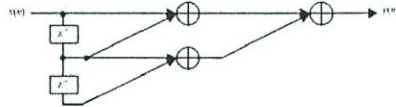


Fig. 7 The 2nd order lowpass type 1-D DPF

The order of DPF is equal to the dimension of Pascal transform matrix minus 1, $(N-1)$. 2-D DPF, which is based on 2-D DPT flow graph in Fig. 5, can be modified to

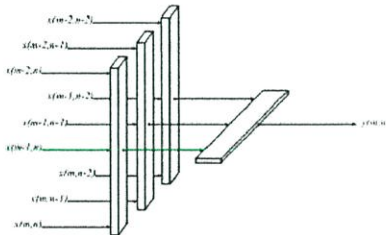


Fig. 8 The 2-D DPF with Convolution Mask Size 3×3

The processing element of highpass (left) and lowpass (right) type are as below, respectively



In practical, the 2-D DPF in Fig.8 used for image processing has to work along with image input arrangement circuit as shown in Fig. 9.

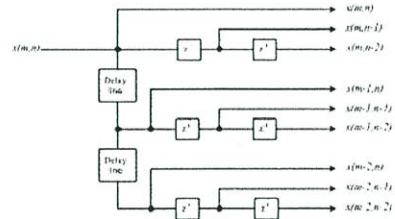


Fig. 9 The image input arrangement circuit

From all above described in section III and III can be applied to higher order or higher dimension DPT or DPF applications.

V. CONCLUSIONS

The DPT hardware realization of both 1-D and 2-D and both highpass type and lowpass type is proposed. The hardware realization using Pascal matrix factorization or decomposition into binary (1,0-1) matrices can give the efficient structure based on butterfly unit for DPT. The obtained DPT flow graph can be used for transformation without multiplications, only additions are needed. Moreover, the DPF which is improved from DPT is also proposed.

ACKNOWLEDGMENT

The authors would like to thank Prof. Tian-Bo Deng of Toho University, Japan, for his helpful comments and support us in Pascal matrices research.

REFERENCES

- [1] M. F. Aburdene and T. J. Goodman, "The Discrete Pascal Transform and Its Applications," *IEEE Signal Processing Letters*, vol. 12, No. 7, pp. 493-495, July 2005.
- [2] T. J. Goodman and M. F. Aburdene, "A hardware implementation of the discrete Pascal transform for image processing," *Proc. of SPIE/IT&T Electronic Imaging*, SPIE vol. 6064, pp.6064 H-1 – 6064 H-8, 2006.
- [3] S. Chivaprecha, U. Nithirochanonit, and K. Dejhan, "Investigation of frequency characteristic in discrete Pascal transform and Its applications," *Submitted to The 4th International Colloquium on Signal Processing and Its Application (CSPA 2008)*, Kuala Lumpur, Malaysia, 2008.
- [4] A. N. Skodras, "Fast Discrete Pascal Transform," *Electronics Letters*, Vol. 42, No. 23, November 2006.
- [5] A. Edelman and G. Strang, "Pascal Matrices," *Am. Math Mon.*, pp. 189-197, 2004.

ประวัติผู้เขียน

ชื่อ-นามสกุล นายนริศ รณรงค์ฤทธิ์
วัน เดือน ปีเกิด 20 มีนาคม 2525 ที่ จังหวัดราชบุรี
ที่อยู่ 20 ถ.โชคชัย ต.โพธาราม อ.โพธาราม จ.ราชบุรี 70120
ประวัติการศึกษา

2544-2547 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2547-ปัจจุบัน วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงาน

2547-2551 ตำแหน่ง Senior Network Engineer บริษัท MFEC Public Company Limited