สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โครงสร้างตัวกรองประสิทธิภาพสูง
สำหรับการหาขอบภาพแบบโซเบลที่ปราศจากตัวคูณ

AN EFFICIENT FILTER STRUCTURE FOR MULTIPLIERLESS SOBEL EDGE DETECTION

ณรงค์ ระวินู

NARONG RAVINU

วพ
ณ213ค
2554

เลขหมู่.....
เลขทะเบียน...117866
วัน,เดือน,ปี....22 ส.ค. 2554

b.12370436
i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2554

KMITL- 2011 – EN – M – 010 - 032

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN EFFICIENT FILTER STRUCTURE FOR MULTIPLIERLESS SOBEL EDGE DETECTION

NARONG RAVINU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN TELECOMMUNICATIONS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2011

KMITL- 2011 – EN – M – 010 - 032

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2011

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	โครงสร้างตัวกรองประสิทธิภาพสูงสำหรับการหาขอบภาพแบบโซเบลที่ปราศจากตัวคูณ
นักศึกษา	ร้อยเอก ณรงค์ ระวินู
รหัสนักศึกษา	49060931
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2554
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

เทคนิคการตรวจจับขอบภาพในวิธี Sobel edge detection เป็นรูปแบบหนึ่งของการตรวจจับขอบภาพบนพื้นฐานของ 1st order Derivative แต่ในการออกแบบตัวกรองดิจิตอล 2 มิตินั้น ต้องใช้ตัวคูณจำนวนมาก เพื่อให้ได้ Output แต่ละค่า เนื่องจากค่าสัมประสิทธิ์ของวงจรกรองมีค่าที่ไม่ใช่ศูนย์อยู่เป็นจำนวนมาก ซึ่งจะส่งผลให้ความเร็ว ในการประมวลผลสัญญาณนั้นลดลง และยังทำให้วงจรที่ออกแบบกินพื้นที่มากอีกด้วย ดังนั้นจึงทำการปรับรูปแบบสมาชิกของ Sobel edge detection และใช้สมการ 2 มิติ สำหรับการคำนวณภาพเอาท์พุทให้เหมาะสมกับการประยุกต์ใช้งานแบบเวลาจริง ในการจำลองการทำงานได้เขียนโปรแกรมภาษา VHDL เพื่อจำลองการทำงานด้วยโปรแกรม ISE WebPACK เปรียบเทียบระหว่าง เทคนิคการตรวจจับขอบภาพแบบโซเบลที่สร้างด้วยโครงสร้างเลขคณิตกระจาย กับเทคนิคใหม่ที่นำเสนอ จากการเปรียบเทียบพบว่า ระบบสามารถประมวลผลได้เสร็จภายในเวลา 34.507 ns และใช้ทรัพยากรของ Altera's FPGA เพียง 64% ซึ่งหมายถึงความซับซ้อนในโครงสร้างฮาร์ดแวร์และความเร็วในการทำงาน ของเทคนิคใหม่ที่นำเสนอสามารถทำงานได้อย่างมีประสิทธิภาพกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	An Efficient Filter Structure for Multiplierless Sobel Edge Detection
Student	Capt. Narong Ravinu
Student ID.	49060931
Degree	Master of Engineering
Program	Telecommunications Engineering
Year	2011
Thesis Advisor	Assoc. Prof. Dr. Kobchai Dejhan

ABSTRACT

Image edge detection techniques is a method of Sobel edge detection. It is a form of image edge detection based on 1st order derivative. However, in the 2-D digital filter design requires a lot of times to get each Output. Because of the filter coefficients are non-zero value is a lot. This will result in speed In signal processing is reduced. It also makes the circuit design also eat a lot of space. Therefore, the fine form and a member of the Sobel edge detection using two-dimensional equations for calculating the image output suitable for real-time applications. In the simulation program was written to simulate the VHDL language running the ISE WebPACK for comparison between Techniques to detect image edges to create a chain Bell distributed arithmetic structure with new techniques being presented. The comparison showed that system can process within the time 34.507 ns. and use the resources of the Altera's FPGA only 64%. This means that the complexity of hardware structure and speed of work of the new technique proposed to function more effectively than.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดีด้วยคำปรึกษาจาก รศ.ดร. กอบชัย เดชหาญ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.ศรวัฒน์ ชิวปรีชา และ ผศ.ดร.ชูสิทธิ์ ประดับเพชร (มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา) ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากอาจารย์ทั้งสามท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ถ่ายทอดความรู้ให้กับข้าพเจ้า

ขอขอบคุณเจ้าหน้าที่บัณฑิตศึกษา คณะวิศวกรรมศาสตร์ ที่ให้ความช่วยเหลือในเรื่องต่างๆ และขอขอบคุณ เพื่อนๆ พี่ๆ น้องๆ ทุกคนในภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ ที่ให้คำแนะนำ และคอยเป็นกำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆ เรื่อง ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ณรงค์ ระวินู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย.....1

1.3 สมมติฐานของการวิจัย.....2

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....2

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....2

1.6 ขอบเขตการวิจัย.....3

1.7 ขั้นตอนการวิจัย.....3

บทที่ 2 ทฤษฎีการประมวลผลภาพ.....4

2.1 ความหมายของการประมวลผลภาพ.....4

2.2 กระบวนการประมวลผลภาพ.....4

2.2.1 กระบวนการในโดเมนความถี่ (Frequency Domain).....4

2.2.2 กระบวนการสเปเชียลโดเมน (Spatial Domain).....4

2.3 ความหมายและนิยามของภาพระบบดิจิตอล.....5

2.4 รูปร่างของภาพ.....6

2.5 การแทนภาพด้วยข้อมูลดิจิตอล.....7

2.6 ระบบการประมวลภาพทางดิจิตอล.....7

2.7 การสุ่มแบบสม่ำเสมอและควอนไทเซชั่น.....8

2.8 เทคนิคสำหรับการประมวลผลภาพ.....9

2.8.1 Image Digitization.....9

2.8.2 Image Enhancement and Restoration.....9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.8.3 Image Encoding....12
2.9 ฟิลเตอร์ (Filter)....13
2.9.1 การเฉลี่ยค่ารอบย่าน (Neighborhood Average)....13
2.9.2 ฟิลเตอร์แบบมัธยฐาน (Medial Filter)....13
2.9.3 ฟิลเตอร์แบบความถี่ต่ำผ่าน (Low-pass Filter)....14
2.9.4 ระบบตัวเลข....14
2.10 ทฤษฎีการกระจายทางคณิตศาสตร์ (DA)....16
2.11 โครงสร้างของวงจรกรองสัญญาณ 2 มิติโดยวิธีการกระจายทางคณิตศาสตร์....19
2.11.1 การกระจายทางคณิตศาสตร์กับการกรองความถี่สูงผ่าน....19
2.11.2 การกระจายทางคณิตศาสตร์กับการกรองความถี่ต่ำผ่าน....20
2.12 การหาขอบภาพ....21
2.12.1 รูปแบบของขอบภาพ (Edge Model)....21
2.12.2 การหาขอบภาพ (Edge Detection)....25
2.12.2.1 การสร้างเกรเดียนท์มุมฉาก (Orthogonal Gradient Generation)....25
2.12.2.2 โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์ (Robert Difference Operator)...27
2.12.2.3 พรีวิทท์โอเปอเรเตอร์ (Prewitt Operator)....28
2.12.2.4 โซเบลโอปอเรเตอร์ (Sobel Operator)....29
2.12.2.5 การสร้างเทมเพลของเกรเดียนท์สำหรับการหาขอบภาพ....31
2.13 ภาษาวีเอชดีแอล (VHDL)....34
2.13.1 การออกแบบระบบดิจิตอล....34
2.13.2 ประวัติความเป็นมาของภาษา VHDL....35
2.13.3 ข้อกำหนดของภาษา VHDL....36
2.13.3.1 ลักษณะทั่วไป....35
2.13.3.2 สนับสนุนการออกแบบแบบลำดับขั้น....37
2.13.3.3 ไลบรารี....37
2.13.3.4 ลำดับคำสั่ง....37
2.13.3.5 การกำหนดคุณสมบัติ....37
2.13.3.6 ชนิดของข้อมูล....38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.13.3.7 โปรแกรมย่อย....38
2.13.3.8 การควบคุมเวลา....38
2.13.3.9 การกำหนดแบบโครงสร้าง....38
2.13.4 ส่วนประกอบต่างๆ ของภาษา VHDL....38
2.13.4.1 หน่วยการออกแบบเอนทิตี....38
2.13.4.2 หน่วยการออกแบบสถาปัตยกรรม....40
2.13.4.3 หน่วยการออกแบบแพ็กเก็ต....44
2.13.4.4 หน่วยการออกแบบโครงแบบ....45
2.13.5 การออกแบบจากบนลงล่าง....44

บทที่ 3 Sobel Edge Detection....48
3.1 แนวความคิดพื้นฐานของ Sobel Edge Detection....48
3.2 แนวความคิดโครงสร้างตัวกรอง Sobel Edge Detection....51
3.3 การใช้การกระจายทางคณิตศาสตร์ (DA) เพื่อออกแบบ Sobel Edge Detection แบบปราศจากตัวคูณ....52
3.4 การใช้ Discrete Pascal Filter เพื่อออกแบบ Sobel Edge Detection แบบปราศจากตัวคูณ 55

บทที่ 4 ผลการทดลอง....58

บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ....63

เอกสารอ้างอิง....63
ภาคผนวก ก เทคโนโลยี เอฟพีจีเอ....64
ภาคผนวก ข ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่....80

ประวัติผู้เขียน....87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

4.1 ผลการใช้ทรัพยากร.....58

4.2 แสดงผลการเปรียบเทียบ ทรัพยากรที่ใช้ในการสังเคราะห์วงจรโดยใช้ FPGA ของบริษัท Altera เบอร์ EPF10K20RC240-4.....58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 เปรียบเทียบเมื่อลดความละเอียดของภาพลง.....9

2.2 การแสดง contrast enhancement จากภาพที่มีการกระจายของ gray tone ต่ำ ซึ่งภาพจะมืด ทำให้มีการกระจายสูงขึ้น จะได้ภาพที่ชัดเจน.....10

2.3 แสดงการ contrast enhancement เพื่อให้ได้ลักษณะโครงร่างของภาพออกมา.....11

2.3(a) สามารถใช้ Edge enhancement นับจำนวนเม็ดเลือดในภาพได้.....11

2.3(b) แสดงโครงร่างของเมล็ดข้าวได้ชัดเจนขึ้น.....11

2.4 แสดงการทำ filtering enhancement.....12

2.4 (a) แสดงการทำ sharpening ของพื้นผิวดวงจันทร์ซึ่งได้ภาพคมชัดขึ้น.....12

2.4 (b) แสดงการทำ high pass filter กับภาพเพื่อให้แสดงรายละเอียดขนสัตว์ได้ชัดเจนขึ้น.....12

2.5 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่สูงผ่าน.....20

2.6 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่ต่ำผ่าน.....20

2.7 แสดงรูปแบบของขอบใน 1 มิติ.....22

2.8 แสดงรูปแบบของขอบใน 2 มิติ.....23

2.9 แสดงรูปแบบของขอบในระบบ 2 มิติซึ่งอยู่ในขอบเขตที่ไม่ต่อเนื่อง.....24

2.10 แสดงการหาเกรเดียนท์มุมฉาก.....25

2.11 แสดงตัวอย่างของการหาขอบภาพโดยใช้ Robert.....28

2.11(a) ภาพตัวอย่างที่นำมาหาขอบภาพ.....28

2.11(b) แสดงขอบภาพที่ได้จากใช้ Robert.....28

2.12 ลักษณะของเทมเพลทที่มีขนาด 3×3.....28

2.13 แสดงตัวอย่างของการหาขอบภาพโดยใช้ Prewitt.....29

2.13(a) ภาพตัวอย่างที่นำมาหาขอบภาพ.....29

2.13(b) แสดงขอบภาพที่ได้จากใช้ Prewitt.....29

2.14 อิมพัลส์เรสปอนส์ของเกรเดียนท์โอเปอเรเตอร์ในแนวมุมฉาก.....30

2.15 แสดงตัวอย่างของการหาขอบภาพโดย Sobel.....31

2.15(a) ภาพตัวอย่างนี้นำมาหาขอบภาพ.....31

2.15(b) แสดงขอบภาพที่ได้จากการใช้ Sobel.....31

2.16 เทมเพลเกรเดียนท์ของอิมพัลส์เรสปอนส์ที่มีขนาด 3×3.....32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.17 แสดงการกระจายของทิศทางของภาพ (compass direction).....33

2.18 แสดงขั้นตอนการออกแบบระบบดิจิตอล.....34

2.19 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี.....39

2.20 แสดงรูปแบบของมัลติเพล็กซ์.....39

2.20(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล.....39

2.20(b) มุมมองของตัวเชื่อมประสาน (Interfacing).....39

2.21 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย.....40

2.21(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล.....40

2.21(b) มุมมองของตัวเชื่อมประสาน.....40

2.22 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก.....40

2.23 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม.....41

2.24 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน.....42

2.25 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์.....42

2.26 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง.....43

2.27 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม.....43

2.28 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กแก็ต.....44

2.29 โครงสร้างของบอดี้แพ็กแก็ต.....45

2.30 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ.....45

2.31 ขั้นตอนการออกแบบจากบนลงล่าง.....46

3.1 ภาพ Sobel Edge Detection.....50

3.2 Sobel Edge Detection Distributed Arithmetic Filter Structure.....54

3.3 The 3-point Lowpass Type DPT Flow Graph.....55

3.4 The 2nd order Low-pass Type 1-D DPF.....56

3.5 Proposed Multiplerless Sobel Edge Detection Filter.....57

3.6 Input Image Arrangement Circuit.....57

4.1 Original input image.....59

4.2 Horizontal gradient output.....59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.3 Vertical gradient output.....59

4.4 Sobel edge detection output image 60

4.5 ผลลัพธ์จากอัลกอลิธึมของ Sobel 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีในปัจจุบันได้มีการใช้ประโยชน์จากภาพถ่าย สำหรับใช้ในการแสดงข้อมูลและการวิเคราะห์ภาพในรูปแบบต่างๆ เช่น ภาพถ่ายดาวเทียมที่ใช้แสดงลักษณะของพื้นผิวของโลก การใช้กล้องจุลทรรศน์อิเล็กตรอนศึกษาดูรูปร่างและพื้นผิวของวัสดุที่มีขนาดเล็ก ตลอดจนการใช้ประโยชน์ในการรักษาความปลอดภัย จึงทำให้การแสดงข้อมูลและการวิเคราะห์ด้วยภาพเป็นที่นิยมกันอย่างแพร่หลาย เพราะสามารถเข้าใจผลลัพธ์ที่ได้อย่างมีประสิทธิภาพมากขึ้น แต่เนื่องจากคุณภาพของภาพที่ได้จากระบบถ่ายภาพต่างๆ เหล่านั้นขึ้นอยู่กับเทคนิคการเก็บข้อมูลภาพและกระบวนการจัดการภาพ (Image Processing) ดังนั้นในการปรับปรุงคุณภาพของภาพให้ดีขึ้นจึงจำเป็นต้องอาศัยเทคนิคที่ดีสำหรับการเก็บข้อมูลภาพและกระบวนการจัดการภาพ โดยทั่วไปเทคนิคการเก็บข้อมูลภาพส่วนใหญ่จะถูกกำหนดและออกแบบให้เหมาะสมกับงานแต่ละด้านอยู่แล้ว แต่กระบวนการจัดการทางภาพนั้น เพื่อให้ได้คุณภาพของภาพที่ดียังต้องมีการพัฒนาอย่างต่อเนื่องทั้งระบบที่เป็นอนาลอกและระบบดิจิตอล จากการศึกษาพบว่าในปัจจุบันวิทยาการทางด้านดิจิตอลได้เข้ามามีบทบาทอย่างมากกับชีวิตประจำวัน จึงเป็นเหตุผลที่ทำให้การพัฒนาระบบการจัดการภาพมุ่งไปสู่กระบวนการจัดการภาพด้วยเทคนิคทางดิจิตอล (Digital Image Processing Technique)[1] และเป็นงานที่นักวิทยาศาสตร์หลายๆ สาขาให้ความสนใจเป็นอย่างมาก เช่น ทางการแพทย์ ทางวิศวกรรม ทางการสื่อสาร ฯลฯ เป็นต้น แต่เนื่องจากขนาดของวงจรดิจิตอลที่มีขนาดใหญ่ทำให้เกิดการสูญเสียเวลาในการทำงาน รวมทั้งค่าใช้จ่ายที่สูงเนื่องจากความสิ้นเปลืองทางด้านสถาปัตยกรรม

เทคนิคการตรวจจับขอบภาพในวิธี Sobel edge detection เป็นเทคนิคแบบหนึ่งของการจรวจจับขอบภาพบนพื้นฐานของ 1st order Derivative หรือ gradient operator[1] โดยมักจะถูกนำไปใช้ในการตรวจจับขอบภาพ ซึ่งมีองค์ประกอบทางด้านฮาร์ดแวร์ที่มีขนาดใหญ่ ทำให้เกิดความสูญเสียในเรื่องของเวลาในการทำงาน (Speed) และมีค่าใช้จ่ายที่สูงมาก

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

การหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เป็นเรื่องที่ง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีคุณภาพต่ำ และมีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วทั้งภาพ สำหรับขอบภาพนั้นเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง หากความแตกต่างนี้มีค่ามาก ขอบภาพก็จะเห็นได้ชัด ถ้าความแตกต่างมีค่าน้อย ขอบภาพก็จะไม่ชัดเจน งานวิจัยนี้จึงได้เสนอแนวทางการตรวจจับภาพแบบโซเบล (Sobel edge detection) มาทำงานร่วมกับเทคนิคการลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปตัวบวก เพื่อทำให้เกิดเทคนิคการตรวจจับขอบภาพแบบใช้ทรัพยากรทางด้านฮาร์ดแวร์น้อยที่สุด อันเป็นผลทำให้ประหยัดเวลาในการทำงาน (Speed) รวมทั้งประหยัดค่าใช้จ่ายอีกด้วย

1.3 สมมติฐานของการวิจัย

ตัวกรองสัญญาณเชิงเลข 2 มิติ (2-D convolution equations)[3] ที่ใช้โครงสร้างเลขคณิตกระจาย มีคุณสมบัติที่ดีอยู่หลายประการ ซึ่งสามารถนำไปประยุกต์กับงานหลายด้านโดยเฉพาะงานในส่วนการคำนวณเอาท์พุทของการตรวจจับขอบภาพแบบโซเบล (Sobel Edge Detection) ทำให้ในปัจจุบันได้มีการสร้างวงจรสำเร็จรูปที่ทำหน้าที่เป็นตัวตรวจจับขอบภาพ แต่เนื่องจากวงจรเหล่านั้นยังมีโครงสร้างฮาร์ดแวร์ขนาดใหญ่ ใช้ทรัพยากรในโครงสร้างฮาร์ดแวร์อย่างสิ้นเปลือง เนื่องจากมีโครงสร้างตัวกรองที่มีตัวคูณ[4] ทำให้เกิดความล่าช้าในการประมวลผล และไม่สะดวกในการทำงาน ดังนั้นจำเป็นต้องใช้วงจรรวมหรือชิพที่ทำหน้าที่ปรับรูปแบบสมาชิกแต่ละตัวในหน้ากาก (kernels) ของ Sobel-edge detection และใช้สมการ 2 มิติ (2-D convolution equations) สำหรับการคำนวณภาพเอาท์พุท (Output Image) ให้เหมาะสมกับการประยุกต์ใช้งานแบบเวลาจริง

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

จากการศึกษาผลงานวิจัย แนวความคิดเดิมของการตรวจจับขอบภาพแบบโซเบล (Sobel-edge detection) การตรวจจับขอบภาพแบบโซเบลเป็นรูปแบบหนึ่งของการหาค่า Gradient หรือ (1^{st} order derivative) ของภาพที่ผ่านการกรองด้วยความถี่ต่ำ (Gaussian low-pass filter) ทำให้ใช้ตัวคูณจำนวน 9 ตัว และจำนวนตัวบวกถึง 8 ตัวต่อหน้ากาก (kernel) ดังนั้น ตัวกรองดิจิตอลขนาด 2 มิติ (2-D digital filter) ต้องใช้ตัวคูณถึง 18 ตัว และตัวบวกถึง 16 ตัว สำหรับการพิสูจน์โดยตรงเพื่อให้ได้ตัวตรวจจับขอบภาพแบบโซเบล ดังนั้นจึงได้ออกแบบโครงสร้างใหม่ของตัวตรวจจับขอบภาพแบบโซเบลที่มีเพียง 14 ตัวบวกเพื่อใช้ในการคำนวณเท่านั้น และจำลองการทำงานบนโปรแกรม ISE WebPACK

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

เนื่องจากการสร้างวงจรตรวจจับขอบภาพแบบโซเบล (Sobel-edge detection) โดยใช้ตัวกรองดิจิตอล 2 มิติแบบเรียลไทม์[2] ทำให้วงจรมีขนาดใหญ่ ใช้ทรัพยากรมาก ใช้เวลาในการประมวลผลมาก ดังนั้นจึงนำเสนอวิธีใหม่โดยการจัดโครงสร้างตัวกรองดิจิตอล ที่สามารถประหยัดทรัพยากร มีขนาดวงจรที่เล็กลง และประมวลผลได้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขอบเขตการวิจัย

1.6.1 ศึกษารูปแบบการทำงานของตัวตรวจจับขอบภาพแบบโซเบล

1.6.2 ออกแบบและสร้างตัวตรวจจับขอบภาพแบบโซเบลโดยใช้สถาปัตยกรรม และตัวตรวจจับขอบภาพแบบโซเบลแบบใหม่ที่นำเสนอ โดยใช้บอร์ด FPGA

1.6.3 เปรียบเทียบประสิทธิภาพจากวงจรที่ออกแบบ และสร้างจริงบนบอร์ด FPGA กับตัวตรวจจับขอบภาพแบบโซเบลแบบดั้งเดิมทั้งในเรื่องของความซับซ้อน (Computational Complexity) และ ความเร็วในการทำงาน (Speed)

1.7 ขั้นตอนของการวิจัย

1.7.1 การวิจัยมีขั้นตอนดังนี้

1.7.1.1 ศึกษาหลักการของวงจรกรองสัญญาณดิจิตอล 2 มิติ (2-D Digital filter)

1.7.1.2 ศึกษาหลักการการตรวจจับขอบภาพแบบโซเบล (Sobel-edge detection)

1.7.1.3 ศึกษาโครงสร้างพื้นฐานการกระจายทางคณิตศาสตร์ของตัวตรวจจับขอบภาพแบบโซเบลไม่ให้มีตัวคูณ

1.7.1.4 ศึกษาโครงสร้างของชิพ FPGA (Field Programmable Gate Array)

1.7.1.5 ศึกษาโปรแกรม ISE WebPack

1.7.1.6 สร้างแบบจำลองของตัวตรวจจับขอบภาพแบบโซเบลแบบปราศจากตัวคูณ และจำลองด้วยโปรแกรม ISE WebPACK

1.7.1.7 ทำการทดสอบเพื่อวิเคราะห์และเก็บผล

1.7.1.8 จัดทำวิทยานิพนธ์

1.7.2 เครื่องมือในการวิจัย

1.7.2.1 FPGA ของบริษัท XILINX

1.7.2.2 โปรแกรม ISE WebPACK

1.7.2.3 โปรแกรม MATLAB

1.7.3 วิธีเก็บรวบรวมข้อมูล

ทดสอบการทำงานของ Sobel edge detection โครงสร้างใหม่ที่ออกแบบด้วยชิพ FPGA และเปรียบเทียบผลการจำลองการทำงานด้วยโปรแกรม ISE WebPACK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีการประมวลผลภาพ

ในการประมวลสัญญาณภาพด้วยระบบคอมพิวเตอร์ จำเป็นต้องเปลี่ยนข้อมูลหรือสัญญาณที่อยู่ในรูปแบบอนาลอกให้เป็นสัญญาณดิจิตอล เพื่อประโยชน์ในการคำนวณและการประมวลผลได้ง่าย ในบทนี้จะกล่าว ถึงความหมายของภาพในระบบดิจิตอล, คณิตศาสตร์พื้นฐานที่เกี่ยวข้องกับการหาขอบภาพ และภาษาวีเอชดีแอล

2.1 ความหมายของ Image processing

Image processing สามารถเรียกอีกอย่างหนึ่งว่าการประมวลผลภาพ[2] ซึ่งหมายถึง การใช้ขั้นตอน หรือกรรมวิธีใดๆก็ตาม มากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อให้ได้ภาพใหม่ที่มีคุณภาพตามที่ต้องการ เช่น ความคมชัดหรือการประหยัดพื้นที่ในการเก็บข้อมูล โดยข้อมูลในเล่มนี้จะเรียก Image processing ว่ากระบวนการประมวลผลภาพ ซึ่งมีเนื้อที่กว้างมากในการศึกษา

2.2 กระบวนการประมวลผลภาพ

การประมวลผลภาพแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

2.2.1 กระบวนการในโดเมนความถี่ (Frequency Domain)

กระบวนการในโดเมนความถี่เป็นการนำการแปลงฟูเรียร์ มาประยุกต์ใช้โดยมีขั้นตอน 3 ขั้นตอนดังนี้

ก. การนำภาพมาหาผลการแปลงฟูเรียร์

ข. นำผลการแปลงฟูเรียร์ มาผ่านขั้นตอนการปรับปรุงภาพ

ค. การแปลงฟูเรียร์กลับ

2.2.2 กระบวนการสเปเชียลโดเมน (Spatial Domain)

สเปเชียลโดเมน เป็นกระบวนการที่กระทำกับจุดภาพโดยตรง อย่างเช่นตัวอย่างของกระบวนการในสเปเชียลโดเมนที่เลือกใช้ ได้แก่ ฮิสโตแกรมอิควอไลเซชั่น (Histogram Equalization), การบีบอัดไดนามิกเรนจ์ (Dynamic range compression) และ การแปลงระดับสีเทา (Gray sacling)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ความหมายและนิยามของภาพระบบดิจิตอล

ภาพในเชิงคณิตศาสตร์จะหมายถึง ฟังก์ชั่น 2 มิติ f(x,y) โดย x และ y เป็นแกนพิกัดในระนาบ 2 มิติ ค่าฟังก์ชั่น f(x,y) จะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพ ที่ตำแหน่ง (x,y) ซึ่งเรียกว่า ระดับสีเทา[5]

ภาพ 2 มิติที่แทนด้วยฟังชั่นที่จุด (x,y) คือความเข้มของแสงที่จุดนั้น เนื่องจากแสงเป็นพลังงาน รูปหนึ่ง ดังนั้น f(x,y) ต้องไม่เป็นศูนย์ และมีค่า (finite) นั่นคือ

$$0 < f(x,y) < \infty \tag{2.1}$$

โดยธรรมชาติของแสง ซึ่งจะต้องมีแหล่งกำเนิดและส่วนที่สะท้อนของแสง ดังนั้นเราสามารถแยกฟังก์ชั่น f(x,y) ออกเป็น 2 ส่วนคือ illumination component i(x,y) และ refectant component r(x,y) จะได้ว่า

$$f(x,y) = i(x,y)\ r(x,y) \tag{2.2}$$

เมื่อ

$$0 < i(x,y) < \infty \tag{2.3}$$

และ

$$0 < r(x,y) < \infty \tag{2.4}$$

สมการ (2.4) แสดงให้เห็นว่า ฟังก์ชั่นการสะท้อนถูกจำกัดขอบเขตระหว่าง 0 (ซึ่งหมายถึงการดูดซึมโดยสมบูรณ์) และ 1 (ซึ่งหมายถึงการสะท้อนโดยสมบูรณ์) ธรรมชาติของ i(x,y) ขึ้นอยู่กับแหล่งกำเนิดแสง ในขณะที่ r(x,y) ขึ้นอยู่กับวัตถุที่สะท้อนแสงมาเข้าตา

ดังกล่าวมาแล้ว ความเข้มของแสงที่จุด (x,y) เราเรียกว่าระดับสีเทา (λ) จากสมการที่ (2.2) ถึง (2.4) จะเห็นว่า (λ) ควรอยู่ในช่วง

$$L_{min} \leq (\lambda) \leq L_{max} \tag{2.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางทฤษฎี L_{min} ต้องมีค่าบวก ในขณะที่ L_{max} ต้องมีค่าน้อยกว่าอนันต์ แต่ในทางปฏิบัติ $L_{min} = i_{min}\, r_{min}$ และ $L_{max} = i_{max}\, r_{max}$ ช่วงของ (L_{min} , L_{max}) เราเรียกว่า ช่วงของระดับสีเทา ในทางปฏิบัติโดยใช้หลักคณิตศาสตร์ เรานิยมปรับช่วง (L_{min} , L_{max}) ให้เป็นช่วง (0, 1) โดย $\lambda = 0$ หมายถึงดำสนิท และ $\lambda = 1$ หมายถึงขาว

2.4 รูปร่างของภาพ

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular Image Model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำและการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์[6] สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (Array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (Pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

สมมุติให้ ภาพ เป็นตัวแปรแบบอะเรย์ขนาด M × N (M แถว และ N คอลัมน์) ที่ใช้เก็บ ภาพขนาด M × N จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็น ภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ ภาพ จะเห็นว่าเราใช้ ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

จากการใช้หน่วยความจำเพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก M × N × g เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640×480, 800×600 และ 1024×768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ ซึ่งมีอัตราส่วนแนวนอนต่อแนวตั้งเท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐาน จะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัส เช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640×520 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าภาพต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกันเนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยวได้ ดังนั้น ในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการคัดลอกข้อมูลทั้ง 8 บิต ส่งให้กับจอภาพซึ่งในกรณีที่จุดภาพมีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้ว ก็จะทำการคัดลอกข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือ ส่วนในกรณี จุดภาพที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการคัดลอกข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 640×480 และมีขนาด 16 บิตต่อจุดภาพ จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 640×480×16 บิต

2.5 การแทนภาพด้วยข้อมูลดิจิตอล

ภาพดิจิตอลเป็นภาพที่ถูกแปลงมาจากภาพอานาลอกอยู่ในรูปแบบตัวเลข โดยภาพอานาลอก จะถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็กๆที่เรียกว่า พิกเซล โดยในแต่ละพิกเซล จะถูกระบุตำแหน่งโดย (x,y) และค่าระดับสีเทาของพิกเซลนั้น คือค่า ของ f(x,y)

2.6 ระบบการประมวลภาพทางดิจิตอล

ระบบการประมวลภาพทางดิจิตอลประกอบด้วย 3 ส่วนใหญ่ๆ คือ

1. การเปลี่ยนสัญญาณอานาลอกเป็นสัญญาณดิจิตอล หรือที่เรียกว่า ดิจิไทเซอร์ (Digitizer)
2. ส่วนประมวลผล (Processing)
3. ส่วนแสดงผล (Display)

ดิจิไทเซอร์ ทำหน้าที่เปลี่ยนสัญญาณภาพให้เป็นตัวเลข เพื่อป้อนข้อมูลสัญญาณให้กับดิจิตอลคอมพิวเตอร์ อุปกรณ์ในส่วนนี้ได้แก่ กล้องโทรทัศน์ดิจิไทเซอร์

ส่วนประมวลผล คือ ดิจิตอลคอมพิวเตอร์ ซึ่งอาจใช้ตั้งแต่ขนาดไมโครคอมพิวเตอร์ ไปจนถึงเมนเฟรมคอมพิวเตอร์ เนื่องจากภาพที่ถูกดิจิไทซ์ จะได้ข้อมูลที่ป็น array ขนาดใหญ่ซึ่งยากต่อการเก็บไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์

ส่วนแสดงผล คือ ส่วนที่เปลี่ยนข้อมูลตัวเลขที่เก็บเป็น array ในคอมพิวเตอร์ ให้อยู่ในรูปที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การสุ่มแบบสม่ำเสมอและควอนไทเซชั่น

เพื่อที่จะประมวลสัญญาณภาพด้วยระบบคอมพิวเตอร์ ฟังก์ชั่นของภาพ f(x,y) จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่อง ทั้งระนาบของภาพ ซึ่งเราเรียกว่า Image sampling ค่าของฟังก์ชั่นที่ได้เรียกว่า gray – level quantization

สมมติว่าสัญญาณภาพต่อเนื่อง f(x,y) ถูกดิจิไทซ์ในระนาบ x – y เป็นช่วงเท่าๆกัน เราสามารถจัด f(x,y) ในรูปเมตริกซ์ขนาด NxN ได้ดังสมการ (2.6.1)

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1)..... & f(0,N-1) \\ f(1,0) & f(1,1)..... & f(1,N-1) \\ . & & \\ . & & \\ f(N-1,0) & f(N-1,1)... & f(N-1,N-1) \end{bmatrix} \quad (2.6)$$

ทางขวาของสมการ จะเรียกว่า ภาพดิจิตอล และทุกๆสมาชิกของเมตริกซ์จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิตอลข้างต้น จะเห็นว่า เราต้องการทราบขนาดของภาพ N×N พิกเซลและจำนวนระดับของระดับสีเทา ในทางปฏิบัติการทำควอทไทเซชั่นในระบบภาพดิจิตอลจะเป็นค่าของ 2 ยกกำลังจำนวนเต็ม คือ

$$N = 2^n \quad (2.7)$$

เมื่อ N คือขนาดของภาพ

และ

$$G = 2^m \quad (2.8)$$

เมื่อ G คือจำนวนระดับของสีเทา

เมื่อ G คือ จำนวนระดับของสีเทา ดังนั้นจำนวน bit (b) ที่ใช้เก็บภาพหนึ่งภาพที่ถูกดิจิไทซ์ คือ

$$B = N \times N \times m \text{ บิท} \quad (2.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(a) (b) (c)

รูปที่ 2.1 เปรียบเทียบเมื่อลดความละเอียดของภาพลง

2.8 เทคนิคสำหรับการประมวลผลภาพ

สามารถแบ่งออกเป็น 3 กลุ่มหลัก คือ

1. Image Digitization
2. Image Enhancement and Restoration
3. Image Encoding

2.8.1 Image Digitization

ดังได้กล่าวมาแล้วถึงความหมายของการดิจิไทซ์ภาพ ซึ่งขนาดความละเอียดของภาพที่ได้ก็ขึ้นอยู่กับการควอนไทซ์ภาพ ซึ่งเครื่องมือที่ใช้ทำขบวนการนี้ที่เรียกว่า Digitizer เช่นกล้องดิจิตอล

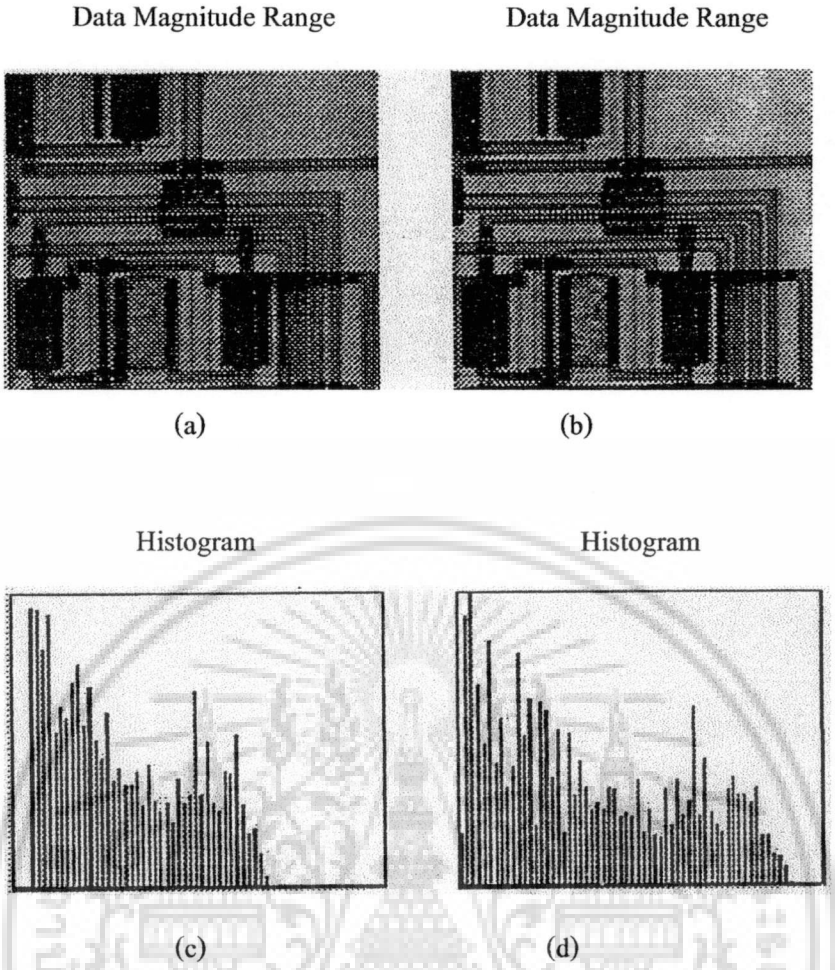
2.8.2 Image Enhancement and Restoration

การปรับปรุงคุณภาพของภาพ (Image Enhancement) เป็นการทำภาพให้อยู่ในรูปที่เหมาะสมขึ้น สำหรับงานเฉพาะอย่าง กล่าวคือ วิธีที่ทำภาพ หรือปรับปรุงภาพเอกซ์เรย์อาจจะไม่เป็นวิธีที่ดีเมื่อนำมาปรับปรุงภาพถ่ายดาวเคราะห์ที่ส่งมาจากการสำรวจอวกาศ

วิธีปรับปรุงคุณภาพของภาพ มีดังนี้

1. Contrast Enhancement เป็นวิธีที่ทำให้ภาพคมชัดขึ้น โดยอาศัยฮิสโตรแกรม อาจใช้แบบ Linear stretch , piecewise linear stretch หรือ equalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



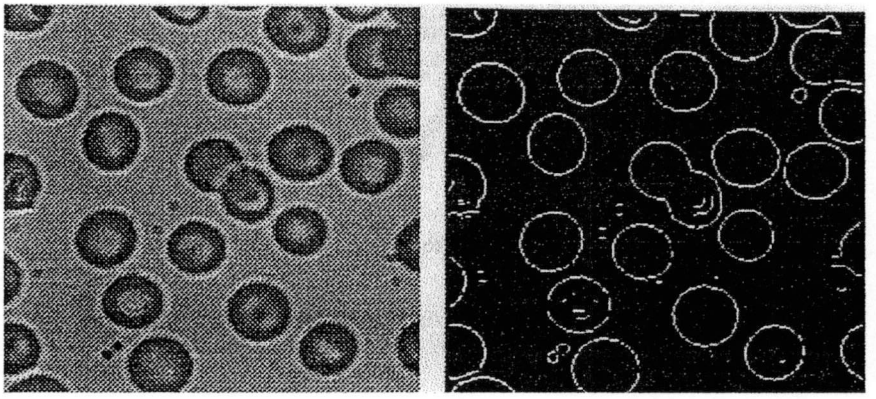
รูปที่ 2.2 การแสดง contrast enhancement จากภาพที่มีการกระจายของ gray tone ต่ำซึ่งภาพจะมืด ทำให้มีการกระจายสูงขึ้น จะได้ภาพที่ชัดเจน

จากรูปที่ 2.2 แสดงให้เห็นถึงการเปรียบเทียบระหว่างภาพ 2 ประเภทคือ ภาพที่แสดงเส้นตรง และภาพที่แสดงความเข้มข้นของสีแต่ละตำแหน่งในภาพ โดยเมื่อมีการปรับปรุงความเข้มแสงของภาพ (Contrast enhancement) จะทำให้เห็นความแตกต่างของภาพได้ชัดเจนมากขึ้น

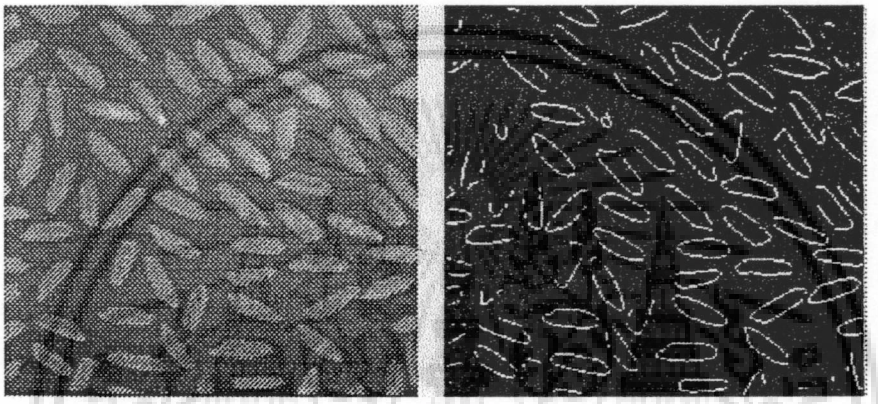
2. Pseudo-coler image processing เป็นการใช้เทคนิคของการทำ density slicing และการใส่สีเทียมให้กับภาพขาวดำ ที่มีระดับสีเทาต่างๆกัน

3. Edge enhancement เป็นการแยกความแตกต่างของจุดภาพที่ใกล้เคียงกัน เพื่อหาขอบเขตของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

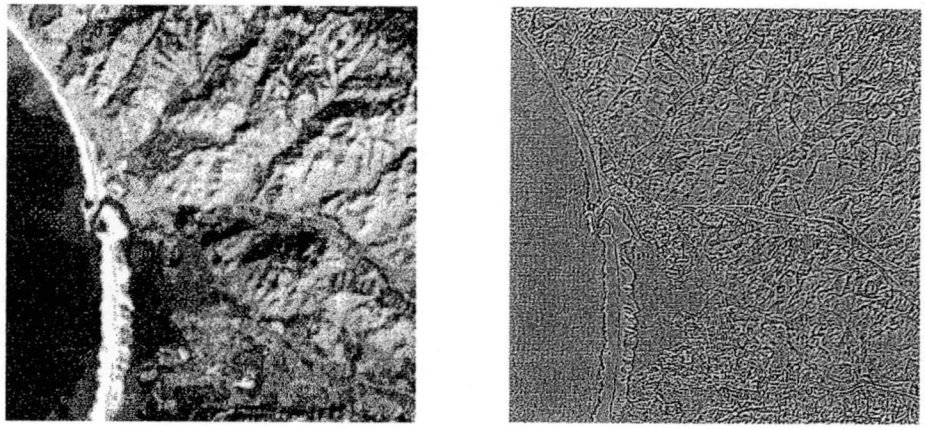
รูปที่ 2.3 แสดงการ contrast enhancement เพื่อให้ได้ลักษณะโครงร่างของภาพออกมา

จากรูปที่ 2.3 a) สามารถใช้ Edge enhancement นับจำนวนเม็ดเลือดในภาพได้ การตรวจจับระดับความแตกต่างของขอบภาพว่ามีกี่ตำแหน่ง แล้วทำการแปลงค่านั้นเป็นจำนวนเม็ดเลือด

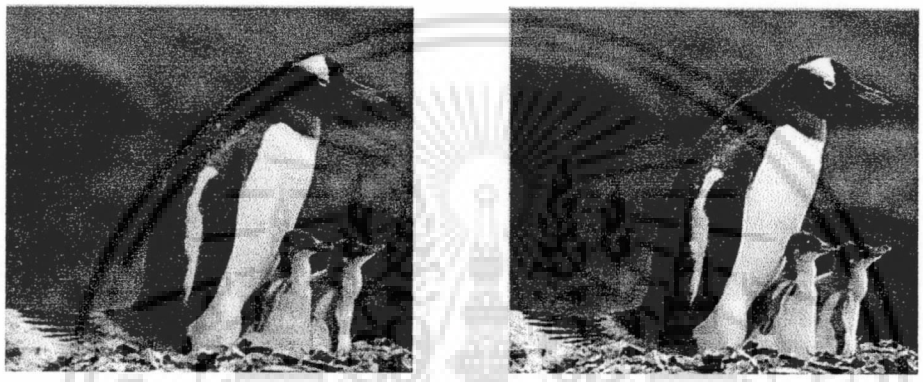
จากรูปที่ 2.3 b) สามารถใช้ Edge enhancement เพื่อแสดงความแตกต่างของขอบภาพให้ชัดเจนมากขึ้น และใช้จำนวนที่บ่งบอกความแตกต่างของขอบภาพนั้นเองเป็นค่าที่บอกถึงจำนวนเมล็ดข้าวสารได้อย่างถูกต้อง

4. Filtering เพื่อให้ภาพ smoothing หรือ sharpening โดยใช้ low-pass filter หรือ high-pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 2.4 แสดงการทำ filtering enhancement

a) แสดงการทำ sharpening ของพื้นผิวดวงจันทร์ซึ่งได้ภาพคมชัดขึ้น

b) แสดงการทำ high-pass filter กับภาพเพื่อให้แสดงรายละเอียดขนสัตว์ได้ชัดเจนขึ้น

5. Image Restoration เป็นขบวนการในการสร้างภาพกลับคืน โดยการหาค่าชดเชย และแก้ความคลาดเคลื่อน เนื่องจากข้อมูลในภาพผิดพลาดไปหรือเป็นขบวนการสร้างภาพกลับคืนจากภาพที่ถูกทำให้เสียไป เนื่องจากปรากฏการต่างๆ โดยใช้หลักการของ linear algebra

2.8.3 Image Encoding

เป็นการใช้เทคนิคต่างๆเพื่อเข้ารหัสข้อมูล เนื่องจากข้อมูลภาพที่จะได้ถูกเก็บในลักษณะเป็นจำนวนไบท์ ซึ่งถ้าภาพมีขนาดใหญ่ ก็ต้องใช้พื้นที่ในการเก็บมากด้วย ข้อจำกัดของเครื่องไมโครคอมพิวเตอร์ ที่มีขนาดหน่วยความจำจำกัด การเข้าโค้ดข้อมูลจึงมีประโยชน์ ในด้านการลดพื้นที่ในการเก็บข้อมูลภาพดังกล่าวมาก ผลของการเข้ารหัสข้อมูลนี้เรียกว่าเป็น Data compression

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้การเข้ารหัสยังมีประโยชน์ในการช่วยลดปริมาณข้อมูลภาพที่ใช้ในระบบการสื่อสาร เช่น การส่งภาพถ่ายจากยานอวกาศมายังโลก เป็นต้น

2.8 ฟิลเตอร์ (Filter)

ฟิลเตอร์ เป็นตัวกรองสัญญาณเบื้องต้นให้ได้ภาพที่มีความคมชัดมากยิ่งขึ้นเพื่อที่จะนำภาพไปประยุกต์ใช้ได้ดียิ่งขึ้นซึ่งมีมากมายหลายชนิดแต่ในที่นี้จะกล่าวเฉพาะที่นิยมใช้กันมากเท่านั้น

2.8.1 การเฉลี่ยค่ารอบย่าน (Neighborhood Average)

การเฉลี่ยค่ารอบย่าน เป็นเทคนิคที่ใช้กันในสเปเชียลโดเมนโดยตรงสำหรับการทำภาพให้ชัดเจนยิ่งขึ้น ให้ภาพอินพุท f(x,y) มีขนาด N×N และ g(x,y) เป็นผลของภาพที่ได้จากการปฏิบัติของระดับสีเทาทุกๆจุด (x,y) ได้จากการเฉลี่ยค่าระดับสีเทาของพิกเซลในภาพ f(x,y) ซึ่งสามารถอธิบายได้ดังนี้

$$g(x,y) = 1/M \quad \{ \quad f(n,m) \quad \} \qquad (2.10)$$

สำหรับ x,y = 0,1,...,n-1 และ m คือผลรวมของจำนวนจุดในย่านนั้นๆ

2.8.2 ฟิลเตอร์แบบมัธยฐาน (Medial Filter)

สิ่งสำคัญประการหนึ่งจากการเฉลี่ยค่ารอบย่าน คือ การเกิดความเบลอที่ขอบของภาพ และมีรายละเอียดสัญญาณรบกวนที่ชัดเจน อาจแก้ไขได้โดยการใช้เทรสโฮลด์ (Threshold) ซึ่งโดยทั่วไปแล้วค่าเทรสโฮลด์จะได้จากการลองผิดลองถูก แต่ถ้าใช้ฟิลเตอร์แบบมัธยฐานระดับสีเทาของแต่ละพิกเซล โดยการใช้ค่าเฉลี่ยมัธยฐานของระดับสีเทาในรอบย่านนั้นๆของพิกเซล จะทำให้ภาพที่ได้ประสิทธิภาพที่ได้ดีขึ้น การหาค่ามัธยฐาน (M) คือเซ็ตครึ่งหนึ่งมากกว่าอีกครึ่งหนึ่งน้อยกว่า M การหาค่ามัธยฐานสามารถหาได้โดยการทำการเรียงข้อมูลจากน้อยไปหามากแล้วนำค่าที่อยู่ตรงกลางนั้นมาใช้แทนค่า M

<u>ตัวอย่าง</u>

สมมติว่ามีย่านขนาด 3×3

(10, 20, 20, 20, 15, 20, 20, 25, 100)

สามารถเรียงข้อมูลให้ได้เป็น

(10, 15, 20, 20, 20, 20, 20, 25, 100)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นค่าที่อยู่ตรงกลางเป็นค่ามัธยฐาน มีค่าเท่ากับ 20 นั้นเอง ความคิดที่มีอิทธิพลต่อฟังก์ชั่นของฟิลเตอร์แบบมัธยฐานคือ จะทำให้จุดที่มีความแตกต่างมากกว่าย่านรอบๆของมัน

2.8.3 ฟิลเตอร์แบบความถี่ต่ำผ่าน (Low-pass Filter)

สัญญาณรบกวนในระดับสีเทาของภาพมักจะมีส่วนของสัญญาณความถี่สูงของการแปลงฟูเรียร์อย่างมาก ดังนั้นจึงต้องหาวิธีการกำจัดเฉพาะช่วงความถี่สูงที่ไม่ต้องการออกไป จากทฤษฎีการทำคอนโวลูชั่นจะได้ว่า

$$G(u,v) = H(u,v)\,F(u,v) \tag{2.11}$$

เมื่อ F(u,v) คือการแปลงฟูเรียร์ของภาพที่ต้องการให้เรียบ ปัญหาก็คือการเลือกทรานส์เฟอร์ฟังก์ชั่น H(u,v) ที่ให้ผลลัพธ์ G(u,v) โดยที่สามารถกำหนดจุดช่วงความถี่สูงของ F(u,v) ออกไปได้ การแปลงอินเวอร์สลาปาส (Inverse Laplace) ของ G(u,v)

$$g(x,y) = L^{-1}[H(u,v)\,F(u,v)] \tag{2.12}$$

จะทำให้ได้ g(x,y) เรียบขึ้นเพราะว่าส่วนประกอบความถี่สูงไม่สามารถผ่านออกไปได้แต่ช่วงความถี่ต่ำจะสามารถผ่านไปได้โดยปราศจากการลดทอนของสัญญาณซึ่งวิธีนี้เรียกว่า ฟิลเตอร์แบบความถี่ต่ำผ่าน ฟังก์ชั่น H(u,v) ถูกเรียกกว่าทรานส์เฟอร์ฟังก์ชั่นของฟิลเตอร์ (Filter Transfer Function) ซึ่งในการคำนวณจะพิจารณาเฉพาะ H(u,v) เท่านั้น ฟิลเตอร์แบบความถี่ต่ำผ่านนี้มีอยู่หลายชนิดด้วยกัน โดยที่แต่ละฟิลเตอร์จะทำหน้าที่กรองความถี่สูงออกไปให้เหลือแต่เฉพาะความถี่ต่ำเท่านั้น

2.8.4 ระบบตัวเลข

ก่อนอื่นจะกล่าวถึงพื้นฐานของเลขฐานสองก่อน เนื่องจากในระบบเลขฐานสิบนั้นมีเลขอยู่สิบค่าที่เป็นฐานคือ 0 ถึง 9 สัญลักษณ์ (+) และ (-) แทน บวก และ ลบ ตามลำดับ และจุดในเลขฐานสิบใช้เป็นตัวแยกระหว่างตัวเลขจำนวนเต็มและเศษส่วน แต่ในดิจิตอลฮาร์ดแวร์มีเลขอยู่สองค่าที่เป็นฐานคือ 0 และ 1 ซึ่งเลขทั้งสองค่าดังกล่าวใช้เป็นสัญลักษณ์แทนทั้ง +,- และจุด ถ้าแทนเลขไบนารี่ของตัวเลขจำนวนเต็มที่ ไม่มีเครื่องหมายซึ่งเป็นสัญลักษณ์เลขไบนารี่ จำนวนเลขไบนารี่ B (bit) สามารถแปลงเป็นเลขฐานสิบด้วยสูตร

$$X = \sum_{j=0}^{B-1} X^j 2^j \tag{2.13}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย X=0 หรือ 1 ดังนั้นเลขไบนารี่ 11011 แทนเป็นเลขฐานสิบได้ 26 ดังนี้ $0\times2^0 + 1\times2^1 + 0\times2^2 + 1\times2^3 + 1\times2^4 = 2+8+16$ ซึ่งคล้ายระบบเลขฐานสิบ นั้นคือ $26 = 2\times10^1 + 6\times10^0$ จากสมการข้างบน ค่าตัวเลขจำนวนเต็มที่มากที่สุดที่ถูกแทนโดยเลขฐานสอง B bit คือ 2B -1 ดังนั้น 10, 16 หรือ 32 บิท สามารถถูกแทนที่ด้วยเลขจำนวนเต็ม 1023,65,535 หรือ 4,150,998,095 ตวามลำดับ ค่าที่อธิบาย สามารถเปลี่ยนแปลงเป็นเศษส่วนได้ดังตัวอย่างต่อไปนี้ สมมติ 16 บิท ถูกแทนที่ด้วยเลขฐานสิบ และทุกจำนวนถูกสเกลด้วย $2^{-5} = 0.03125$ ดังนั้นค่าที่มากที่สุดที่สามารถแทนได้คือ $(2^{16} -1) \times 2^{-5} = 2047$ และค่าที่น้อยที่สุดคือ 0.03125 มันเปรียบเสมือนกับว่ามีจุดในตำแหน่งที่ 5 จาก Least Singnificant Bit (LSB) และคำนวณส่วนที่เป็นเศษส่วนในสมการข้างบนโดยการใช้ค่า 2 ยกกำลังด้วยค่าลบ ดังนั้นในกรณีนี้เลขไบนารี่ 0101100010110110 แทน $0\times10^{10} + 1\times10^9 + 0\times10^8 + 1\times10^7 + 1\times10^6 + 0\times10^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} + 0\times2^{-5}$ หรือเท่ากับ 709.6875

ในขบวนการดิจิตอล ทุกจำนวนมีขนาดน้อยกว่าหรือเท่ากับหนึ่งซึ่งสามารถทำได้โดยสมมติว่าจำนวนบิทมีจุดอยู่ทางซ้ายของ Most Significant Bit (MSB) หรือทุกจำนวนถูกสเกลด้วย 2^{-B} ซึ่งทำให้เลขไบนารี่มากที่สุด $1111...1,(2^B - 1)/2^{-B} = 1 - 2^{-B}$ ซึ่งน้อยกว่า 1

Sign-Magnitude Notation

ใน Sign-Magnitude Notation เลขที่มีค่าสัมปสิทธิ์ (absolute) เท่ากันจะแทนด้วยเลขไบนารี่ที่เหมือนกัน แต่จะมีบิทพิเศษเรียกว่า บิทเครื่องหลาย (Sign Bit) รวมเข้าไปทางซ้ายของ MSB เพื่อทำให้เกิดความแตกต่างระหว่างค่าบวกกับค่าลบ ตัวอย่างเช่น

+0.828125 = 0 110101

-0.828125 = 1 110101

ดังนั้นเลขไบนารี่ B บิท ใน Sign-Magnitude Notation ซึ่งสมมติว่ามีจุดอยู่ระหว่าง Sign bit และ MSB มีค่าอยู่ระหว่าง $-(1 - 2^{-B+1}) < X < (1-2^{-B+1})$ ทำให้จำนวนค่าบวกมีจำนวนเท่ากับจำนวนค่าลบและค่า 0 แทนได้ทั้ง + 0 หรือ - 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2'S Complement Notation

ใน 2'S Complement Notation จำนวนบวกเหมือนกับ Sign-Magnitude Notation ซึ่งบิทแรก เป็นบิทเครื่องหมาย จำนวนลบแทนโดยการเอาค่า absolute ลบออกจาก 2 เพราะค่าของ absolute ต้องน้อยกว่า 1 ดังนั้นบิทแรกจะเป็น 1 เสมอ เปรียบเทียบด้วยตัวอย่างค่า 2'S Complement ของ -0.375 หาได้โดย 2 – 0.375 = 1.625 ซึ่งค่าไบนารี่คือ 1.101 ในการแปลงเลข 2'S Complement โดยการสมมติว่าบิทแรก (sign bit) ถูกคูณด้วย -1 ดังนั้นถ้า X ถูกแทนที่ด้วยไบนารี่ B บิท (X^0,X^1,...,X^{B-1}) ใน 2'S Complement Notation , ค่าของ X ในระบบฐานสิบ หาได้จาก

$$X = -X^0 + \sum_{j=0}^{B-1} X^j 2^{-j} \qquad (2.14)$$

ดังนั้นตามสมการเลขไบนารี่ 2'S Complement 1101 แทนเป็นเลขระบบฐานสิบ

$$-1 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3} = -1 + 0.5 + 0.12 = -0.375$$

จากสมการการแปลง 2'S Complement จำนวนที่มีค่ามากที่สุด คือ $1\text{-}2^{-B+1}$ (0111...1) และจำนวนที่มีค่าน้อยที่สุด คือ -1 (1000...0) นอกจากนั้น ค่า 0 แทนด้วยค่าเดียวคือ (000...0), และมีค่าที่เป็นลบมากกว่าค่าที่เป็นบวกอยู่หนึ่งจำนวนคือ (-1)

2.9 ทฤษฎีการกระจายทางคณิตศาสตร์ (Distribute Arithmetic)

การกระจายทางคณิตศาสตร์ (Distribute Arithmetic) หรือเรียกย่อๆว่า DA ใช้กันมากในการจัดรูปแบบคณิตศาสตร์[8,9,10,11] โดยเฉพาะจะปรากฏอยู่ในงานการประมวลผลสัญญาณดิจิตอล โดยจัดสมการให้อยู่ในรูปแบบการคำนวณแบบบิท เพื่อให้สมการเปลี่ยนเป็นวงจรดิจิตอลได้โดยเฉพาะความเร็วในการคูณซึ่งจะออกแบบระบบการคูณแบบเปิดตารางค่าสัมประสิทธิ์ ที่ถูกเก็บไว้ในหน่วยความจำแบบอ่านได้อย่างเดียว การประยุกต์ใช้การกระจายทางคณิตศาสตร์ (DA) นี้จะแปลงสมการผลบวกของผลคูณให้การกระทำการคำนวณแบบบิท ซึ่งจะทำให้หาผลลัพธ์ได้เร็ว อุปกรณ์ที่สำคัญที่ใช้ใน DA มีชิฟท์รีจิสเตอร์ (shift register) เพื่อเลื่อนข้อมูลจากเวิร์ดเป็นบิท รอม (Read only Memory : ROM) สำหรับเก็บค่าสัมประสิทธิ์ตัวรวมข้อมูล (Accumulator) เพื่อรวมข้อมูลจากรอม เมื่อเปรียบเทียบอุปกรณ์รอบข้างระหว่าง DA กับ Standard CPU ก็มีลักษณะคล้ายกัน คือมี DAC (Digital to Analog Converter), ADC (Analog to Digital Converter) และระบบสัญญาณนาฬิกาเช่นกัน แต่ DA จะไม่มี CPU ประมวลสัญญาณเรื่องของความเร็ว DA จะเร็วกว่าเพราะว่า ถ้าสมมติให้คำนวณหาค่า $Y = A_1X_1 + A_2X_2 + A_3X_3 + A_4X_4$ TMS3210 จะใช้เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ประมาณ 640 ns เพราะคำสั่งคูณ 16×16 บิทใช้เวลา 160 ms ส่วน DA ถ้าใช้รอมความเร็วสูงและข้อมูลเข้ามีขนาด 16 บิท ต่อ 1 เวิร์ด ส่งเข้าขาแอดเดรสของรอมแบบ 1 BAAT [8] DA ใช้สัญญาณนาฬิกาประมวลผล 16 ลูก, 2 BAAT สัญญาณนาฬิกา 8 ลูก, 8 BAAT ใช้สัญญาณนาฬิกา 2 ลูก แต่ถ้าเป็น 16 BAAT จะใช้สัญญาณนาฬิกา 1 ลูก เท่านั้น เป็นกรรมวิธีทางคณิตศาสตร์ที่จัดรูปแบบการคำนวณที่ส่วนใหญ่อยู่ในรูปผลบวกของผลคูณ (Sum of Product) ของเลขฐานสิบให้กระจายออกเป็นบิทหรือในรูปแบบเลขฐานสองเพื่อให้ค่าการกระทำคณิตศาสตร์เปลี่ยนไปในรูปของวงจรดิจิตอลได้

วงจรดังกล่าวอาจเป็นวงจรซึ่งใช้แทนทรานเฟอร์ฟังก์ชั่น ทางดิจิตอลฟิลเตอร์ (digital filtering) แบบต่างๆที่มีลักษณะเชิงอินพุทและ หรือเอาท์พุทคูณกับค่าคงที่ การใช้ DA ในการประมวลผล จะมีประสิทธิภาพมากโดยเฉพาะสมการแบบผลบวกของผลคูณของสัญญาณดิจิตอลที่เป็นอินพุทกับค่าสัมประสิทธิ์ของสมการซึ่งจะได้กล่าวถึงรายละเอียดต่อไป

จุดประสงค์ของ DA คือ เปลี่ยนสมการคณิตศาสตร์ที่ส่วนใหญ่อยู่ในรูปของผลบวกของผลคูณเป็นวงจรดิจิตอลให้หาผลลัพธ์ออกมาได้อย่างรวดเร็ว เมื่อเทียบกับไมโครโปรเซสเซอร์มาตรฐานในงานประมวลผลสัญญาณดิจิตอลที่มีสัญญาณนาฬิกาเท่ากันเพราะว่า DA ใช้หลักการเปิดตารางคูณค่าสัญญาณอินพุทกับค่าสัมประสิทธิ์ที่ถูกเก็บไว้ในหน่วยความจำ ลองพิจารณาตัวอย่างผลบวกของผลคูณต่อไปนี้

$$Y = \sum_{k=0}^{K} A_k X_k \qquad (2.15)$$

สมการที่ 2.15 ให้ A_k เป็นค่าสัมประสิทธิ์ X_k เป็นข้อมูลอินพุทซึ่ง X_k เป็นตัวเลขคอมพลีเมนท์ของสอง (2'S complement) และให้ Absolute ของ $X_k < 1$ ดังนั้นแสดงค่าของ X_k ในเทอมของบิทได้

$$X_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \qquad (2.16)$$

เมื่อ b_{kn} = บิทต่างๆของ Xk เวิร์ดมีค่าเป็น 0 หรือ 1

b_{k0} = บิทที่แสดงเครื่องหมาย

b_{kN-1} = บิทที่มีนัยสำคัญต่ำสุด (LSB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

117866

N = จำนวนบิทต่อข้อมูลอินพุท

เมื่อแทนค่า X_k จากสมการที่ (2.16) ลงในสมการที่ (2.15) ได้เป็นสมการที่ (2.17a) แสดงค่า Y ในเทอมของบิทของ X_k ดังนี้

$$Y = \sum_{k=1}^{K} A_k \left[-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right] \qquad (2.17\ a)$$

สมการที่ (2.17a) สับเปลี่ยนจัดลำดับการบวกใหม่ให้เป็นกลุ่มย่อยจะได้

$$Y = \sum_{n=1}^{N-1} \left(\sum_{k=1}^{K} A_k b_{kn} \right) 2^{-n} + \sum_{k=1}^{K} A_k \left(-b_{k0} \right) \qquad (2.17b)$$

$$Y = -2^{0}(b_{10}A_1 + b_{20}A_2 + b_{30}A_3 + \ldots\ldots + b_{k0}A_k$$

$$+2^{-1}(b_{11}A_1 + b_{21}A_2 + b_{32}A_3 + \ldots\ldots + b_{k1}A_k$$

$$+2^{-2}(b_{12}A_1 + b_{22}A_2 + b_{32}A_3 + \ldots\ldots + b_{k2}A_k$$

$$+2^{-3}(b_{13}A_1 + b_{23}A_2 + b_{33}A_3 + \ldots\ldots + b_{k3}A_k$$

$$+2^{-5}(b_{15}A_1 + b_{24}A_2 + b_{34}A_3 + \ldots\ldots + b_{k4}A_k$$

$$+2^{-(N-1)}(b_{1,N-1}A_1 + b_{2,N-1}A_2 + b_{3,N-1}A_3 + \ldots\ldots + b_{k,N-1}A_k \qquad (2.17c)$$

สมการที่ (2.17c) เป็นการกระจายทางคณิตศาสตร์ของค่าสัมประสิทธิ์กับบิทต่างๆของข้อมูลอินพุท ด้วยเหตุนี้สมการที่ (2.17c) จึงเรียกว่า Distributed Arithmetic (DA) พิจารณาวงเล็บใหญ่ของสมการที่ (2.17b) คือ

$$Y = \sum_{k=1}^{K} A_k b_{kn} \qquad (2.17d)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่า b_{kn} จะมีค่าเป็น 0 หรือ 1 เท่านั้น ดังนั้นผลลัพธ์สมการ (2.17d) จะมีค่าเป็นไปได้ทั้งหมด 2^K ค่าซึ่งค่าต่างๆเหล่านี้ถูกคำนวณแล้วเก็บไว้ในหน่วยความจำแบบอ่านได้อย่างเดียว (Rom) ข้อมูลบิทของอินพุทถูกใช้เป็นแอดเดรส (Address) ของรอมได้โดยตรง ผลลัพธ์ถูกส่งไปเก็บไว้ในตัวรวมข้อมูล (Accumulator) หลังจากที่มีการส่งข้อมูลอินพุทที่ทำเป็นแอกเดรสจำนวน N รอบก็จะได้ผลลัพธ์ Y ออกมา

2.10 โครงสร้างของวงจรกรองสัญญาณ 2 มิติโดยวิธีการกระจายทางคณิตศาสตร์

ตัวกรอง 2 มิติเชิงเลขนั้นมีประโยชน์ ประยุกต์ใช้โดยเฉพาะเป็นตัวเพิ่มคุณภาพหรือลดสัญญาณรบกวนของสัญญาณในลักษณะ 2 มิติ ของระบบการประมวลสัญญาณเชิงเลข เช่น สัญญาณภาพรังสีเอ็กซ์ สัญญาณภาพดาวเทียม เป็นต้น สัญญาณเหล่านี้จะต้องพิจารณาเป็นลักษณะช่วงๆสัญญาณซึ่งส่งมาเป็นลำดับสามารถเขียนให้อยู่ในรูปฟังก์ชั่นได้โดยมีตัวแปรจำนวนเต็ม 2 ตัว เช่น เป็นตัวแสดงลำดับของสัญญาณซึ่งถูกกำหนดให้ค่าทุกค่าเป็นจำนวนเต็ม การประยุกต์ DA กับวงจรกรองสัญญาณสองมิติ สมการความแตกต่างเชิงเส้น (Linear Differential Equation) ของวงจรกรองสัญญาณสองมิติ (Two-dimension Second Digital Filter) โดยทั่วไปอยู่ในรูปของสมการที่ 2.18

$$Y_{m,n} = \sum_{k=0}^{2}\sum_{l=0}^{2} a_{k,l} X_{m-k,n-l} \qquad (2.18)$$

2.10.1 การกระจายทางคณิตศาสตร์กับการกรองความถี่สูงผ่าน

ตัวกรองความถี่สูงผ่าน ความหมายโดยทั่วไปในการประมวลผลภาพ เป็นการเน้นส่วนที่เป็นความถี่สูงหรือระดับความแตกต่างความเข้มของแสงมาก ขณะที่ลดส่วนที่เป็นความถี่ต่ำหรือระดับความแตกต่างความเข้มของแสงน้อย เพราะว่าขอบ (Edge) หรือรายละเอียดของภาพเป็นส่วนของความถี่สูง ดังนั้นการกรองความถี่สูงผ่านจะเป็นการเพิ่มความแตกต่างของแสงแต่ละที่ (Local Contrast) และความคมชัด (Sharp) ของภาพ ตัวอย่างผลตอบสนองทางอิมพัลส์ (Impulse Response) ของตัวกรองความถี่สูงผ่านที่ใช้ปรับความแตกต่างของแสงให้ชัดเจนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	-1	0
-1	5	-1
0	-1	0

รูปที่ 2.5 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่สูงผ่าน

จากผลตอบสนองทางอิมพัลส์ (Impulse Response) ของตัวกรองความถี่สูงผ่าน เมื่อพิจารณาอยู่ในรูปของผลตอบสนองทางความถี่ (Frequency Response) จะอยู่ในรูปของฟูเรียร์ทรานส์ฟรอม (Fourier Transform) ตามสมการที่ 2.19

$$H(\omega_1,\omega_2)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}h(m,n)\,e^{-j\omega_1 m}e^{-j\omega_2 n} \tag{2.19}$$

เมื่อ

$h(m,n)$ คือผลตอบสนองทางอิมพัลส์ (Impulse Response)

$$-\pi \le \omega_1 \le \pi \quad \text{และ} \quad -\pi \le \omega_2 \le \pi$$

2.10.2 การกระจายทางคณิตศาสตร์กับการกรองความถี่ต่ำผ่าน

ปริมาณสัญญาณภาพส่วนใหญ่ จะรวมส่วนที่เป็นสัญญาณความถี่ต่ำไว้มากกว่าส่วนที่เป็นความถี่สูง ส่วนที่เป็นความถี่ต่ำที่เหมาะสมคือ จุดแต่ละจุดจะต้องมีความสัมพันธ์กันกับจุดที่อยู่ข้างเคียง ถ้าไม่สัมพันธ์กันภาพจะไม่คมชัดโดยเหตุที่ว่าสัญญาณรบกวนที่สุ่มขึ้นมา (Random noise) มีช่วงความกว้างของความถี่ที่แผ่ออกไป จากกลุ่มของความถี่ปกติตัวกรองความถี่ต่ำผ่านจะทำการลดส่วนที่เป็นความถี่สูงขณะที่จะรักษาส่วนที่เป็นความถี่ต่ำเอาไว้ และจะลดสัญญาณรบกวนที่มีปริมาณมากให้น้อยลง

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

รูปที่ 2.6 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลตอบสนองทางอิมพัลส์ (Impulse Response) ของตัวกรองความถี่ต่ำผ่าน เมื่อพิจารณาอยู่ในรูปของผลตอบสนองทางความถี่ (Frequency Response) จะอยู่ในรูปของฟูเรียร์ทรานส์ฟรอม (Fourier Transform) ตามสมการที่ 2.20

$$H(\omega_1,\omega_2)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}h(m,n)\,e^{-j\omega_1 m}e^{-j\omega_2 n} \qquad (2.20)$$

เมื่อ

$h(m,n)$ คือผลตอบสนองทางอิมพัลส์ (Impulse Response)

$$-\pi \le \omega_1 \le \pi \quad \text{และ} \quad -\pi \le \omega_2 \le \pi$$

2.11 การหาขอบภาพ

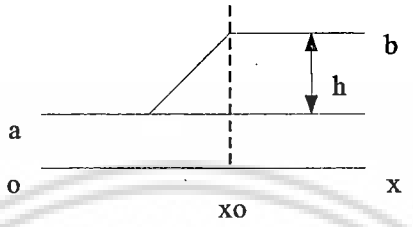
ขบวนการ หรือขั้นตอนที่สำคัญอย่างหนึ่งของการประมวลผลภาพคือการหาขอบภาพเพื่อมาปรับปรุงภาพ ซึ่งคำว่าขอบในที่นี้จะหมายถึงจุดที่มีระดับความเข้ม (Gray-level) เปลี่ยนแปลงอย่างเห็นได้ชัด และจุดที่มีการเปลี่ยนแปลงนี้จะเป็นกรอบของรายละเอียดในส่วนต่างๆของภาพ ซึ่งขอบที่กล่าวมานี้สามารถเกิดขึ้นได้ทุกทิศทางและอาจมีความเข้มไม่เท่ากันก็ได้ ส่วนที่เป็นขอบนี้จะแสดงสเปคตรัมที่ความถี่สูงเท่านั้น

2.11.1 รูปแบบของขอบภาพ (Edge Model)

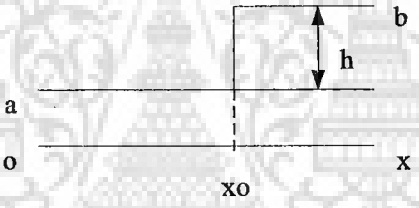
ในรูปที่ 2.7 (a) แสดงรูปร่างของขอบในหนึ่งมิติ ซึ่งแสดงในขอบเขตที่ต่อเนื่อง (continuous domain) โดยขอบที่มีลักษณะดังรูปนี้จะเรียกว่า แรมป์ (ramp) ซึ่งค่าของมันจะเพิ่มขึ้นตามระดับแอมปลิจูดของภาพจากต่ำไปสูง สิ่งที่สามารถบ่งบอกลักษณะของขอบคือ ความสูง ความชัน (slop) มุม (angle) และพิกัดในแนวนอน ณ จุดกึ่งกลางของความชัน ถ้าหากค่าของความชันของมุมมีค่าเท่ากับ 90 องศา จะเรียกขอบชนิดนี้ว่า ขอบแบบขั้นบันได (step edge) แสดงได้ดังรูปที่ 2.7(b) ซึ่งในระบบดิจิตอลจะนำขอบแบบขั้นบันไดไปใช้ในการสร้างภาพชนิดที่เรียกว่า มนุษย์เป็นผู้สร้างขึ้น (Artificial image) ตัวอย่างเช่น แบบการทดลอบ (Test pattern) และข้อมูลในทางกราฟิก (Bi level graphic data) และในขอบเขตที่มีความต่อเนื่องนี้ รูปแบบของภาพในระบบสองมิติจะถือว่าแอมปลิจูดที่มีลักษณะไม่ต่อเนื่องนั้นเป็นค่าคงที่ในจุดข้างเคียงทั้งแปดจุด(neighborhood orthogonal) ในรูปที่ 2.8(a) แสดงรูปร่างของขอบภาพในระบบสองมิติ นอกเหนือจากพารามิเตอร์ในระบบหนึ่งมิติแล้ว ในระบบสองมิตินี้ยังประกอบด้วยสิ่งที่สำคัญคือ ความชันของขอบที่มีความสัมพันธ์กับแกน และในรูปที่ 2.8(b) แสดงทางเดินของขอบของวัตถุที่มีรูปร่างแปดเหลี่ยมที่มีค่าแอมปลิจูดสูงกว่าบริเวณที่เป็นพื้นที่ (back ground) และในรูปที่ 2.9 แสดงลักษณะของขอบที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

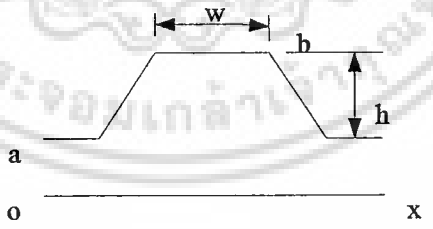
รูปแบบเป็นแรมป์และยูนิต (unit) ในขอบเขตที่ไม่ต่อเนื่องจากรูปจะพบรูปแบบของขอบแบบแรมป์ในแนวตั้งจะประกอบด้วยพิกเซล (pixel) ที่มีค่าเดียวกัน ซึ่งค่าแอมปลิจูดของมันมีค่าเท่ากับค่ากลางโดยเฉลี่ยของจุดข้างเคียง จากรูปที่แสดงให้เห็นจะมีขอบแบบแรมป์ที่อยู่ในแนวทแยงอยู่ 2 แบบ คือ ซิงเกิลพิกเซลทรานส์ซิชั่น (single pixel transition) ซึ่งจุดที่เป็นขอบเขตอยู่ระหว่างแอมปลิจูดสูงกับแอมปลิจูดต่ำและอีกแบบหนึ่งเรียกว่า สมูททรานส์ซิชั่น (smoothed transition) ซึ่งวิธีการหาขอบจะหาจากการหาค่าเฉลี่ยใน 2×2 จุดจากทุกๆจุดของขอบในแนวทแยง



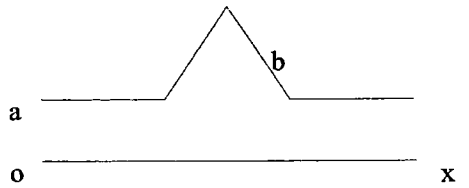
(a) Ramp edge



(b) Step edge



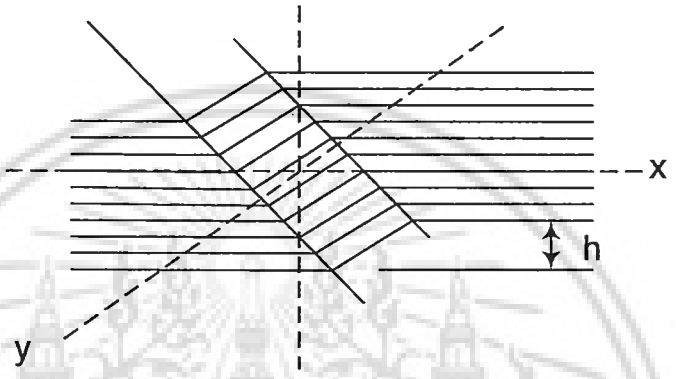
(c) Line



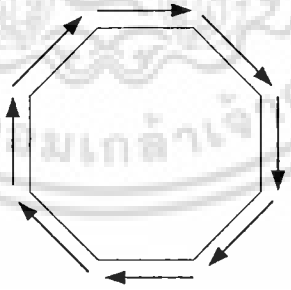
(d) Roof edge

รูป 2.7 แสดงรูปแบบของขอบใน 1 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) Edge model



(b) Edge Direction Definition

รูป 2.8 แสดงรูปแบบของขอบใน 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a a a a a b b b b b	a a a b b b b b b b	a a a a a a a a a a
a a a a a b b b b b	a a a a b b b b b b	a a a a a a a a a a
a a a a a b b b b b	a a a a a b b b b b	a a a a a b b b b b
a a a a a b b b b b	a a a a a a b b b b	a a a a a b b b b b
a a a a a b b b b b	a a a a a a a b b b	a a a a a b b b b b
Vertical step edge	Diagonal step edge	Corner step edge

a a a a c b b b b b	a a c b b b b b b b	a a a a a a a a a a
a a a a c b b b b b	a a a c b b b b b b	a a a a c c c c c c
a a a a c b b b b b	a a a a c b b b b b	a a a a c b b b b b
a a a a c b b b b b	a a a a a c b b b b	a a a a c b b b b b
a a a a c b b b b b	a a a a a a c b b b	a a a a c b b b b b
Vertical ramp edge	Diagonal ramp edge	Corner step edge

a a a a c b b b b b	a a d e b b b b b b	a a a a a a a a a a
a a a a c b b b b b	a a a d e b b b b b	a a a a d c c c c c
a a a a c b b b b b	a a a a d e b b b b	a a a a c b b b b b
a a a a c b b b b b	a a a a a d e b b b	a a a a c b b b b b
a a a a c b b b b b	a a a a a a d e b b	a a a a c b b b b b
Vertical ramp edge	Diagonal ramp edge	Corner step edge

รูปที่ 2.9 แสดงรูปแบบของขอบในระบบ 2 มิติซึ่งอยู่ในขอบเขตที่ไม่ต่อเนื่อง

จากรูปที่ 2.9 นี้จะพบว่าสำหรับขอบแบบขั้นบันได (discrete step edge) และขอบแบบคอนเนอร์แรมป์ (Corner ramp edge) ตำแหน่งของขอบอยู่ที่มีแอมปลิจูดสูงกว่ารอบข้าง สำหรับขอบแบบซิงเกิลพิกเซลทรานซิชั่น ตำแหน่งที่เหมาะสมของขอบที่จะอยู่ตรงที่จุดที่อยู่ในแนวที่มีการเปลี่ยนแปลงและในแบบของแนวทแยงจะมีจุด 2 จุดที่อยู่ใกล้กันในทิศทางเดินของขอบภาพ ดังนั้น ตำแหน่งของขอบภาพที่แท้จริงจะอยู่ตรงจุดที่มีแอมปลิจูดสูงกว่าอีกจุดหนึ่งในระหว่างจุดคู่นั้น และในรูปที่ 2.9 นี้จุดที่เป็นตำแหน่งของขอบภาพจะพิมพ์เป็นตัวเอียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 การหาขอบภาพ (Edge Detection)

วิธีการเบื้องต้นที่ใช้สำหรับการหาขอบภาพในลำดับแรก (First order Derivative Edge Detection) มีอยู่ 2 วิธีด้วยกัน วิธีแรกคือ การหาเกรเดียนท์ (gradient) ในทิศทางมุมฉากคือตามแนวนอนและแนวตั้ง และอีกวิธีเราจะทำการหาเกรเดียนท์ในหลายๆทิศทาง

2.11.2.1 การสร้างเกรเดียนท์มุมฉาก (Orthogonal Gradient Generation)

การกำหนดให้ G(x,y) คือเกรเดียนท์ใน 1 มิติ และ θ คือมุมพิจารณาโดยเทียบกับแนวระดับ ดังนั้นจะทำการคำนวณหาเกรเดียนท์สำหรับขอบภาพจาก

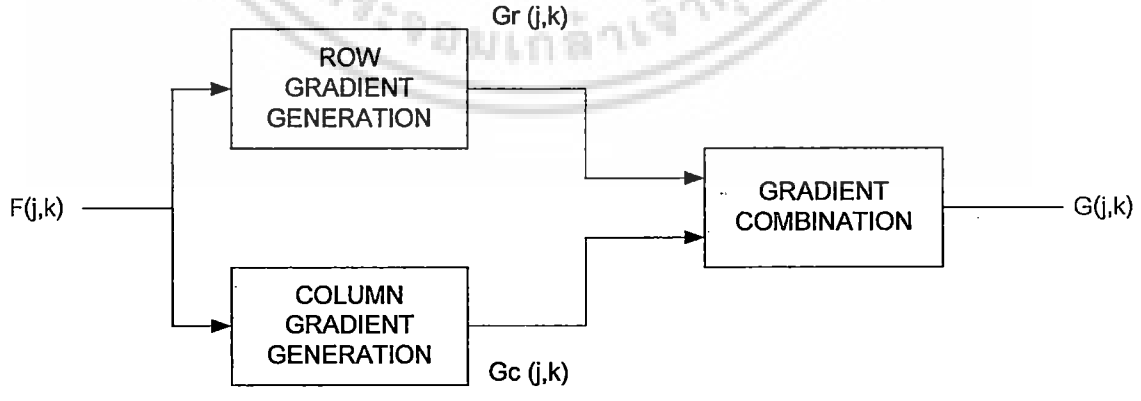
$$G(x,y) = \frac{\partial F(x,y)}{\partial x}\cos\theta + \frac{\partial F(x,y)}{\partial y}\sin\theta \qquad (2.21)$$

ในรูปที่ 2.10 แสดงการสร้างเกรเดียนท์ของขอบภาพ [G(j,k)] ในขอบเขตที่ไม่ต่อเนื่อง โดยจะแยกเป็นเกรเดียนท์ของแถว (row edge gradient) [Gr(j,k)] และเกรเดียนท์ของหลัก (column gradient) [Gc(j,k)] และค่าแอมปลิจูดของเกรเดียนท์ ซึ่งหมายความถึงความเข้มของภาพแสดงได้ดังนี้

$$G(j,k) = \{[Gr(j,k)] + [Gc(j,k)]\}^{\frac{1}{2}} \qquad (2.22)$$

การคำนวณค่าเกรเดียนท์บางครั้งอาจประมาณค่าจาก

$$G(j,k) = |Gr(j,k)| + |Gc(j,k)| \qquad (2.23)$$



รูปที่ 2.10 แสดงการหาเกรเดียนท์มุมฉาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทิศทางของเกรเดียนท์ตามแนวนอนคือ

$$\theta(j,k) = \tan^{-1}\left\{\frac{Gc(j,k)}{Gr(j,k)}\right\} \qquad (2.24)$$

ซึ่งวิธีการที่ง่ายที่สุดสำหรับการสร้างเกนเดียนท์คือ การหาค่าความแตกต่างของความเข้มในแต่ละจุดตามแนวแกนนอน (row) และตามแนวแกนตั้ง (column) ซึ่งเกรเดียนท์ตามแนวแกนนอนหาได้จาก

$$Gr(j,k) = F(j,k) - F(j,k-1) \qquad (2.25\ a)$$

และเกรเดียนท์ในแนวแกนตั้งหาได้จาก

$$Gc(j,k) = F(j,k) - F(j+1,k) \qquad (2.25\ b)$$

โดยเกรเดียนท์นี้จะนำมาใช้หาขอบภาพก็ต่อเมื่อ Gc และ Gr มีค่าเป็นบวกจากซ้ายไปขวาหรือจากล่างขึ้นบนของภาพ

ตัวอย่างเช่นจะหาค่าผลต่างความเข้มโดยใช้เกรเดียนท์ตามแนวแกนนอนของรูปที่ 2.9 เป็นขอบของภาพแบบแรมป์ในแนวดิ่ง (vertical ramp edge model) ได้ดังนี้

0 0 0 0 h 0 0 0 0

โดยที่ h = b-a ซึ่งหมายถึง ความสูงของขั้นบันไดและเกรเดียนท์ในแนวแกนนอนของขอบภาพแบบแรมป์ในแนวดิ่งคือ

0 0 0 0 h/2 h/2 0 0 0

จากค่าข้างบนนี้จะมีค่าที่เท่ากันอยู่ 1 คู่

สำหรับเกรเดียนท์ของขอบในแนวทแยง (Diagonal edge gradient) จะหาได้จากผลต่างของจุดแต่ละคู่ตามแนวทแยง ซึ่งในเรื่องนี้จะใช้วิธีการเรียกว่า โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์ (Robert Difference operator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2.2 โรเบิร์ตดิฟเฟอเรนซ์โอเปอเรเตอร์ (Robert Difference Operator)

สำหรับโอเปอเรเตอร์ชนิดนี้หาได้จาก

$$G(j,k) = \{[G_1(j,k)]+[G_2(j,k)]\} \tag{2.26 a}$$

หรืออาจหาได้จาก

$$G(j,k) = \left\{[G_1(j,k)]^2 +[G_2(j,k)]^2\right\}^{\frac{1}{2}} \tag{2.26 b}$$

เมื่อ

$$G_1(j,k) = F(j,k) - F(j+1,k+1) \tag{2.26 c}$$

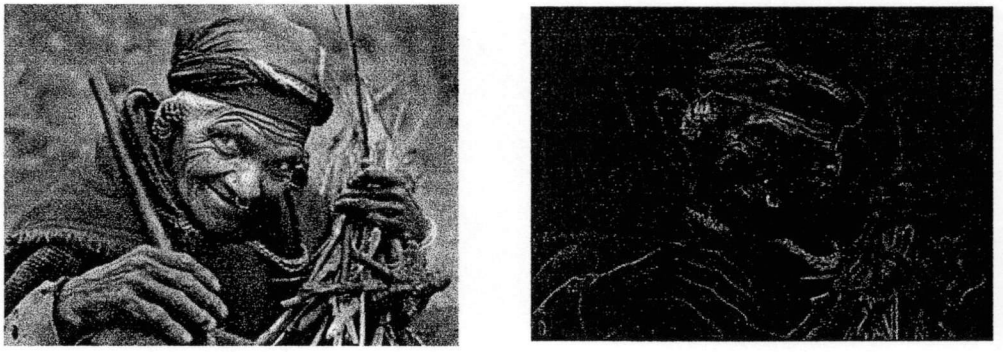
$$G_2(j,k) = F(j,k+1) - F(j+1,k) \tag{2.26 d}$$

และทิศทางของขอบภาพคือ

$$\theta(j,k) = \frac{\pi}{4} + \tan^{-1}\left\{\frac{Gc(j,k)}{Gr(j,k)}\right\} \tag{2.27}$$

การหาขอบในทิศทางเดียวเช่นนี้ จะไม่ค่อยสมบูรณ์นักสำหรับภาพที่มีความส่องสว่างเพียงเล็กน้อย ซึ่งการแก้ปัญหานี้ทำได้โดยการเกรเดียนท์ใน 2 มิติ คือ ทำการหาผลต่างใน 1 ทิศทาง แต่จะแสดงค่าออกมาเป็นค่าเฉลี่ยทั้งหมดในทิศทางมุมฉาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) (b)

รูปที่ 2.11 แสดงตัวอย่างของการหาขอบภาพโดยใช้ Robert

(a) ภาพตัวอย่างที่นำมาหาขอบภาพ

(b) แสดงขอบภาพที่ได้จากใช้ Robert

2.11.2.3 พรีวิทท์โอเปอเรเตอร์ (Prewitt Operator)

A0	A1	A2
A7	$F_{(j,k)}$	A3
A6	A5	A4

รูปที่ 2.12 ลักษณะของเทมเพลทที่มีขนาด 3×3

โอเปอเรเตอร์นี้จะใช้เทมเพลท (Template) ที่มีขนาด 3×3 ในการหาเกรเดียนท์โอเปอเรเตอร์ ถ้าสมมุติให้แต่ละจุดมีการเรียงลำดับตามรูป ดังนั้นเกรเดียนท์ของขอบคือ

$$G(j,k) = \left\{ \left[G_1(j,k) \right]^2 + \left[G_2(j,k) \right]^2 \right\}^{\frac{1}{2}} \qquad (2.28\ a)$$

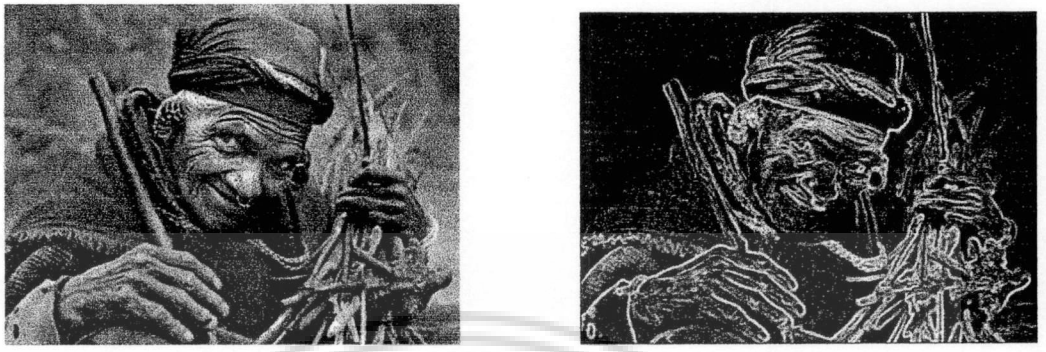
โดยที่

$$G_1(j,k) = \frac{1}{k+2} \left[(A2 + KA3 + A4) - (A0 + KA7 + A6) \right] \qquad (2.28\ b)$$

$$G_2(j,k) = \frac{1}{k+2} \left[(A0 + KA1 + A2) - (A6 + KA5 + A4) \right] \qquad (2.28\ c)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาโอเปอเรเตอร์แบบพรีวิทท์ค่า K ที่นำมาใช้มีค่าเท่ากับ 1 เมื่อแทนค่าลงไปในสูตรข้างต้นของเกรเดียนท์ในแนวนอนและแนวตั้ง สุดท้ายจะได้ยูนิตเกน (unit gain) ที่มีค่าเป็นบวกและลบของค่าเฉลี่ยความเข้มเพื่อหาตำแหน่งของขอบภาพ



(a) (b)

รูปที่ 2.13 แสดงตัวอย่างของการหาขอบภาพโดยใช้ Prewitt
(a) ภาพตัวอย่างที่นำมาหาขอบภาพ
(b) แสดงขอบภาพที่ได้จากใช้ Prewitt

2.11.2.4 โซเบลโอปอเรเตอร์ (Sobel Operator)

การหาขอบภาพของโซเบลนี้ จะแตกต่างจากแบบพรีวิทท์ตรงที่ค่าความเข้มของภาพในทิศเหนือ, ใต้, ตะวันออก, ตะวันตก จะมีค่าเพิ่มขึ้นอีกเท่าตัวในที่นี้คือค่า K ในสมการที่ 2.28 b และ 2.28 c มีค่าเท่ากับ 2 ดังนั้นค่าน้ำหนัก (weight) ที่เปลี่ยนไปแต่ละจุดคือตัวที่แสดงการกระจายของเกรเดียนท์

พรีวิทท์โอเปอเรเตอร์ จะใช้สำหรับการหาขอบภาพในแนวตั้งและแนวนอนได้ดีกว่าโอเปอเรเตอร์แบบโซเบล แต่โซเบลโอเปอเรเตอร์ใช้ในการหาขอบภาพในทิศทางแนวทแยงได้ดี การใช้โอเปอเรเตอร์แบบพรีวิทท์และโซเบล จะทำให้ได้ขอบภาพของวัตถุที่มีความชัดเจนกว่าการใช้โรเบิร์ตโอเปอเรเตอร์ ทั้งนี้เพราะว่ามีขนาดของโอเปอเรเตอร์ที่ใหญ่กว่าทำให้การเฉลี่ยค่าความเข้มในส่วนที่มีความส่องสว่างน้อยได้ดีกว่าเกรเดียนท์ในแนวนอนและแนวตั้งของดีแทคเตอร์ชนิดต่างๆ ที่กล่าวมาข้างต้นจะประกอบด้วยการนำความเข้มรอบๆ จุดที่ต้องการหาค่ามารวมกัน ดังนั้นการเดียนท์ในแนวนอนและแนวตั้งจะคำนวณจากความสัมพันธ์ดังนี้คือ

$$Gr(j,k) = F(j,k) \quad Hr(j,k) \tag{2.29 a}$$

$$Gc(j,k) = F(j,k) \quad Hc(j,k) \tag{2.29 b}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $Hr(j,k)$ คือ อิมพัลส์เรสปอนส์ในแนวแกนนอน

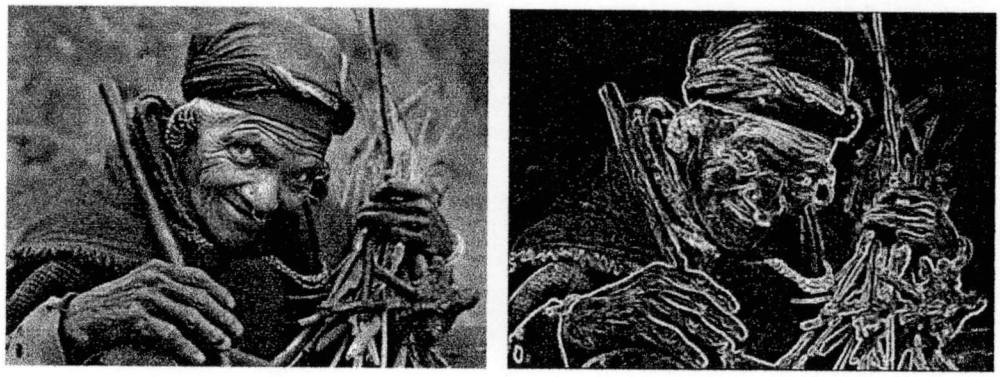
และ $Hc(j,k)$ เป็นแนวแกนตั้งของโอเปอเรเตอร์ชนิดต่างๆ

Operator	Row gradient	Column gradient
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Robert	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

รูปที่ 2.14 อิมพัลส์เรสปอนส์ของเกรเดียนท์โอเปอเรเตอร์ในแนวมุมฉาก

โอเปอเรเตอร์ที่กล่าวมาข้างต้นทั้งหมดนี้มีข้อจำกัดที่จะไม่สามารถหาขอบภาพได้อย่างถูกต้องแม่นยำในสภาพที่มีการรบกวนสูง (Noise) ซึ่งปัญหานี้อาจแก้ไขได้โดยการใช้โอเปอเรเตอร์ที่มีขนาดใหญ่ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) (b)

รูปที่ 2.15 แสดงตัวอย่างของการหาขอบภาพโดย Sobel
(a) ภาพตัวอย่างที่นำมาหาขอบภาพ
(b) แสดงขอบภาพที่ได้จากการใช้ Sobel

2.11.2.5 การสร้างเทมเพลทของเกรเดียนท์สำหรับการหาขอบภาพ (Edge Template Gradient)

การหาขอบภาพโดยใช้เทคนิคของเกรเดียนท์ ซึ่งเป็นวิธีที่ทำการหาการเปลี่ยนแปลงความเข้มของภาพในทิศทางมุมฉากคือในแนวแกนนอนและตามแนวแกนตั้ง ดังนั้นวิธีการหาทิศทางของขอบภาพนั้นจะทำได้โดยการคำนวณค่าเวกเตอร์ จากผลบวกเกรเดียนท์เหล่านั้น และต่อไปจะคำนวณทิศทางของขอบภาพในหลายๆ ทิศทางโดยการคอนโวลูชั่นระหว่างภาพกับเทมเพลทเกรเดียนท์ของ อิมพัลส์เรสปอนส์ (template gradient impulse array) ซึ่งเทมเพลทเหล่านั้นจะได้มาจาก

$$G(j,k) = \text{MAX}\left[|G_1(j,k)|,...,|G_m(j,k)|,...,|G_M(j,k)|\right] \quad (2.30\text{ a})$$

โดยที่

$$G_m(j,k) = F(j,k) * H_m(j,k) \quad (2.30\text{ b})$$

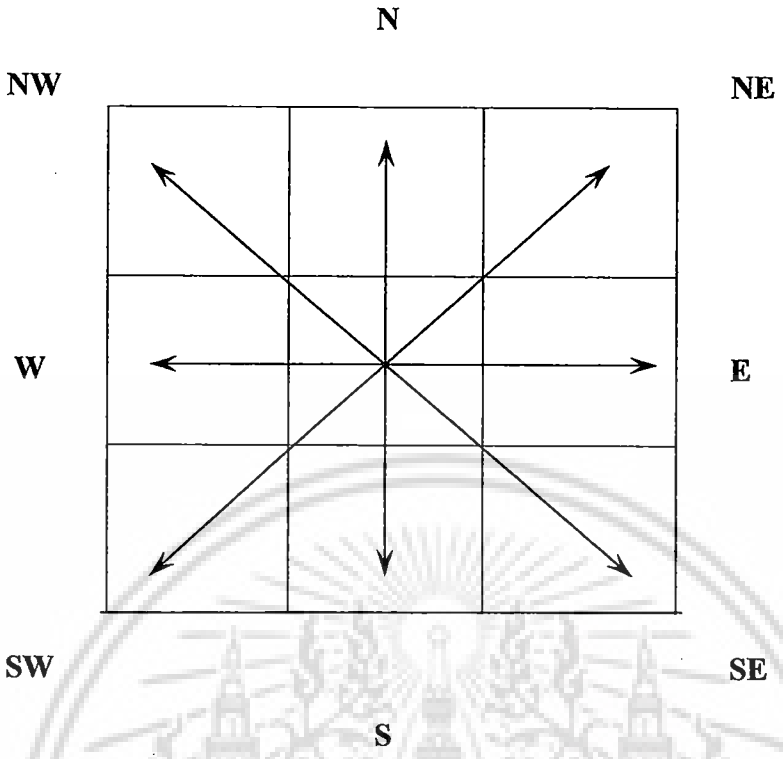
โดยที่ m คือจำนวนทิศทางในการคำนวณ ดังนั้นเกรเดียนท์จะได้มาจากการคอนโวลูชั่นระหว่างภาพกับเกรเดียนท์ของอิมพัลส์เรสปอนส์คือ $H_m(j,k)$ สุดท้ายขอบภาพที่จะได้อยู่ในทิศทางที่มีผลต่างความเข้มมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gradient direction	Prewitt compass gradient	Kirsch compass gradient
East(H_1)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$
Northeast (H_2)	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$
North(H_3)	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$
Northwest (H_4)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$
West (H_5)	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$
Southwest (H_6)	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$
South (H_7)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$
Southeast (H_8)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$
Scale factor	1/5	1/15

รูปที่ 2.16 เทมเพลทเกรเดียนท์ของอิมพัลส์เรสปอนส์ที่มีขนาด 3×3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงการกระจายของทิศทางของภาพ (compass direction)

ในรูปที่ 2.17 แสดงการเกรเดียนท์ 8 ของ โอเปอร์เรเตอร์ชนิดต่าง ๆ และในคริท์ (kirsch) นั้นทิศทางของเกรเดียนท์จะหาได้จาก

$$G(j,k) = \underset{i=0}{\overset{7}{MAX}} \left[\left| 5Si - 3Ti \right| \right] \qquad (3.11a)$$

โดยที่

$$Si = Ai + Ai + Ai + 1 + Ai + 2 \qquad (3.11b)$$

$$Ti = Ai + 3Ai + 4Ai + 5Ai + 6Ai + 7 \qquad (3.11c)$$

และเคิร์ทเกรเดียนท์อาจคำนวณได้จากสมการที่ 2.30 b

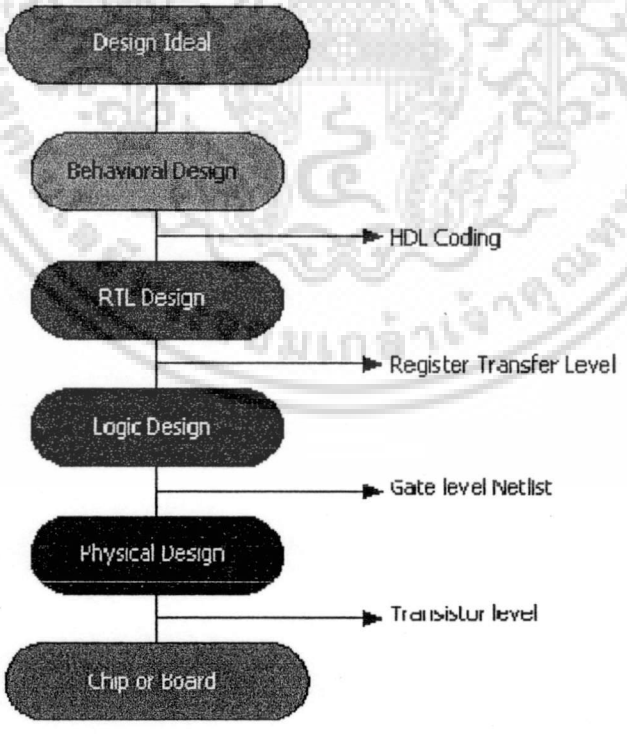
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 ภาษาวีเอชดีแอล (VHDL)

ความซับซ้อนในระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะ ส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยใน การออกแบบหรือ CAD มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการ ออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ (HDL : Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่อง เพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

2.12.1 การออกแบบระบบดิจิตอล

ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำหนดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ ที่ใช้งานได้จะต้องผ่านขั้นตอนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้น ก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป รูปที่ 2.18 แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิตอลทั่วไป ขั้นแรกผู้ออกแบบจะกำหนดแนวความคิดในการออกแบบแล้วทำการพัฒนาให้สามารถนำมาใช้ได้อย่างสมบูรณ์ ซึ่งภาย ในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบในเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะเป็นผังงานแสดงแบบหรือ รหัสคำสั่งเทียม (Pseudo code) ก็ได้



รูปที่ 2.18 แสดงขั้นตอนการออกแบบระบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปเป็นการออกแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิก ที่จำเป็นทั้งหมดเพื่อนำมาประกอบเป็นระบบที่สมบรูณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วยบัสหนึ่งหรือสอง ทิศทาง (Unidirectional or Bidirectional Bus) ส่วนกระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่าง รีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่ ขั้นตอนถัดมาเป็นการออกแบบวงจรลอจิก ซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐานและฟลิปฟลอป (flip-flop) มาประกอบเป็นอุปกรณ์ย่อยต่างๆ เช่น รีจิสเตอร์เก็บข้อมูล บัสวงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการโยงใยระหว่างเกทและ ฟลิปฟลอปนั่นเองการออกแบบในขั้นตอนถัดไป เป็นการเปลี่ยนเครือข่ายการโยงใยที่ได้จากขั้นตอนที่แล้วให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ Layout ซึ่งขั้นตอนนี้จะเกี่ยวข้องโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์ เพื่อ แทนเกทและฟลิปฟลอปต่างๆ และในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือสารที่โรงงานเพื่อผลิตออกมาเป็น วงจรรวมในที่สุด

2.12.2 ประวัติความเป็นมาของภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ด แวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และ สามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษา VHDL เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกร ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น VHDL ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มี ความซับซ้อนได้ทั้งหมด ดังนั้น VHDL จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง

วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกา หรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้น ประกอบกับเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้ จากการนำวงจรดิจิตอลหลายๆ วงจรมาทำการผลิตอยู่บนแผ่นซิลิกอนที่มีพื้นที่เพียง 1 - 2 ตารางเซนติเมตรเท่านั้น ซึ่ง เป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือ ในการทำงานและความคงทนต่อสภาพ แวดล้อมสูง แต่เนื่องจากในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ ต้องใช้วิศวกร และเวลาในดำเนินการมาก ฉะนั้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DoD จึงจัดตั้งโครงการขึ้นมาเพื่อศึกษาวิธีการที่ช่วยในการพัฒนา วงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น ซึ่งโครงการดังกล่าวมีชื่อว่า "Very High Speed Integrated Circuits" หรือ VHSIC โดยในระยะแรกนั้นโครงการนี้ถือเป็นความลับทาง ด้านความมั่งคงของประเทศ และอยู่ภายใต้ความควบคุมดูแลของ United States International Traffic and Arms Regulations (ITAR) สำหรับมาตรฐานของภาษาที่ใช้บรรยาย พฤติกรรมวงจรหรือฮาร์ดแวร์ของระบบ สำหรับโครงการ VHSIC ที่ DoDได้ให้ไว้สามารถสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่อง คอมพิวเตอร์โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่นเดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรม ภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า "Hardware Description Language" หรือ HDL

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

2.12.3 ข้อกำหนดของภาษา VHDL

DoD ได้ตั้งข้อกำหนดสำหรับภาษา VHDL ในเดือนมกราคมปี ค.ศ.1983 ไว้ดังนี้

2.12.3.1 ลักษณะทั่วไป

DoD ได้กำหนดให้ VHDL เป็นภาษาสำหรับการออกแบบและบรรยายของฮาร์ดแวร์ ซึ่งหมายถึงความสามารถ ในการอธิบายและออกแบบในระดับสูง การจำลอง (Simulation) การสังเคราะห์ (Synthesis) และการทดสอบ (Testing) นอกจากนั้น VHDL ยังถูกกำหนดไว้สำหรับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรยายฮาร์ดแวร์ตั้งแต่ระดับบนซึ่งก็คือระบบจนถึง ระดับเกทอีกด้วย เนื่องจากในการทำงานของระบบดิจิตอลนั้น ทุกๆ องค์ประกอบภายในระบบไม่ว่าเล็กหรือใหญ่ จะทำงานไปพร้อมๆ กัน ซึ่งในเรื่องของความพร้อมเพรียงในการทำงานนี้ก็ถือเป็นข้อกำหนดที่สำคัญอย่างหนึ่งของ VHDL ด้วยเช่นกัน (สำหรับในภาษาที่ใช้ในการบรรยายฮาร์ดแวร์นั้นความพร้อมเพรียงจะหมายถึงทุกๆ คำสั่ง องค์ประกอบ เกทหรือวงจรลอจิกจะถูกนำมาปฏิบัติทั้งหมด ดังนั้นในที่สุดแล้วก็จะดูเหมือนว่าได้มีการปฏิบัติไป พร้อมๆ กัน)

2.12.3.2 สนับสนุนการออกแบบแบบลำดับขั้น

การออกแบบแบบลำดับขั้น เป็นลักษณะที่สำคัญอย่างหนึ่งสำหรับการออกแบบระบบที่มีหลายๆ ระดับ โดยในการ ออกแบบจะประกอบด้วยส่วนการบรรยายการเชื่อมต่อ และส่วนการบรรยายหน้าที่การทำงาน ซึ่งหน้าที่การทำงาน ของระบบสามารถกำหนดได้ด้วยตัวเอง หรืออาจถูกกำหนดโดยโครงสร้างที่ประกอบด้วยองค์ประกอบย่อยๆ ลง ไปได้เช่นกัน แต่ที่ระดับล่างสุดองค์ประกอบต้องถูกบรรยายหน้าที่การทำงานด้วยตัวมันเอง และไม่สามารถกำหนด การทำงานโดยลักษณะแบบโครงสร้างได้

2.12.3.3 ไลบรารี

VHDL ได้สนับสนุนการมีไลบรารีเพื่อระบบการจัดการที่ดี ผู้ออกแบบสามารถกำหนดลักษณะและการทำงานของ อุปกรณ์พื้นฐานไว้ในระบบไลบรารี หรือจะใช้ไลบรารีที่ระบบได้จัดเตรียมไว้แล้วก็ได้ โมเดลและการบรรยายที่ถูก ต้องควรจัดเก็บไว้ในไลบรารีหลังจากที่ได้ผ่านการคอมไพล์เรียบร้อยแล้วเพื่อให้ผู้ออกแบบคนอื่นๆ สามารถนำไป ใช้ได้ด้วย

2.12.3.4 ลำดับคำสั่ง

แม้ว่าการปฏิบัติคำสั่ง หรือกระบวนการโดยพร้อมเพรียงกันจะเป็นคุณสมบัติที่สำคัญของ VHDL ก็ตาม ตัวภาษา เองก็ยังมีการจัดเตรียมลักษณะการควบคุมแบบลำดับคำสั่งไว้ให้ด้วย เมื่อผู้ออกแบบได้กำหนดหน้าที่และองค์ประกอบ ที่ทำงานพร้อมกันของระบบไว้เรียบร้อยแล้ว ผู้ออกแบบยังสามารถบรรยายหน้าที่การทำงานซึ่งเป็นรายละเอียดภายใน ของแต่ละองค์ประกอบได้ในลักษณะเดียวกับการเขียนโปรแกรมที่ประกอบด้วยโครงสร้างแบบ case, if - then - else และ loop ทั่วๆ ไปได้ การบรรยายแบบลำดับคำสั่งทำให้การออกแบบหน้าที่การทำงานของอุปกรณ์กระทำได้ สะดวกและง่ายขึ้น อย่างไรก็ตามโครงสร้างทั้งหมดของ VHDL ก็ยังคงเป็นการทำงานแบบพร้อมเพรียงกันเช่นเดิม

2.12.3.5 การกำหนดคุณสมบัติ

นอกจากการกำหนดอินพุทและเอาท์พุทแล้ว เงื่อนไขอื่นๆ ก็มีผลต่อการปฏิบัติหน้าที่ของอุปกรณ์ฮาร์ดแวร์ด้วยเช่นกัน โดยสิ่งนี้รวมถึงสภาพแวดล้อมและลักษณะทางกายภาพของอุปกรณ์นั้นๆ ด้วย ซึ่งภาษาสำหรับการออกแบบที่ดีควร ให้ผู้ออกแบบกำหนดคุณสมบัติของอุปกรณ์ที่ใช้ได้ด้วย เช่น สามารถกำหนดขนาด ลักษณะทางกายภาพเวลา โหลด และเงื่อนไขทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาพแวดล้อมอื่นๆ ซึ่งความสามารถในการกำหนดคุณสมบัตินี้ก็เป็นส่วนหนึ่งที่มีอยู่ในภาษา VHDL ด้วยเช่นกัน

2.12.3.6 ชนิดของข้อมูล

VHDL สามารถกำหนดชนิดของข้อมูลไม่เพียงแต่ชนิด BIT และ BOOLEAN เท่านั้น แต่ยังสามารถกำหนดชนิด ของข้อมูลเป็นจำนวนเต็ม จำนวนจริง จุดทศนิยม และชนิดลำดับการนับ (Enumerate Type) หรือแม้แต่ชนิดของ ข้อมูลที่ผู้ออกแบบกำหนดขึ้นมาเองก็ได้

2.12.3.7 โปรแกรมย่อย

ความสามารถในการใช้ฟังก์ชันและโพรซีเจอร์ (Procedure) ก็เป็นข้อกำหนดอีกอย่างหนึ่งใน VHDL ซึ่งผู้ออกแบบ สามารถนำโปรแกรมย่อยมาใช้ในการเปลี่ยนแปลงชนิดของข้อมูล การกำหนดหน่วยของลอจิก การกำหนดตัวกระทำต่างๆ หรือหน้าที่อื่นๆ ตามที่ต้องการได้ เช่นเดียวกับการเขียนโปรแกรมทั่วไป

2.12.3.8 การควบคุมเวลา

VHDL อนุญาตให้ผู้ออกแบบสามารถกำหนดเวลาในการส่งผ่านข้อมูลหรือสัญญาณได้ตามต้องการ การตรวจสอบ การออกแบบเกทหรือการหน่วงเวลาก็สามารถกระทำได้โดยการกำหนดช่วงเวลาที่แน่นอนหรือกำหนดให้มีการรอคอย เหตุการณ์ (Event) นอกจากนี้ก็ยังสามารถกำหนดรูปแบบของสัญญาณนาฬิกาได้อีกด้วย

2.12.3.9 การกำหนดแบบโครงสร้าง

การกำหนดโครงสร้างขององค์ประกอบต่างๆ สามารถกระทำได้ในทุกระดับของการออกแบบ โดยการกำหนดโครง สร้างขององค์ประกอบร่วมที่เกิดจากองค์ประกอบย่อยซึ่งแตกต่างกันหรือ เหมือนกันก็เป็นข้อกำหนดอย่างหนึ่งของ VHDL เช่นกัน

2.12.4 ส่วนประกอบต่างๆ ของภาษา VHDL

ในการเขียนรูปแบบบรรยายระบบดิจิตอล ในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเอชดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

- หน่วยการออกแบบเอนทิตี (Entity Design Unit)
- หน่วยการออกแบบสถาปัตยกรรม (Architecture Design Unit)
- หน่วยการออกแบบแพ็กเก็ต (Package Design Unit)
- หน่วยการออกแบบโครงแบบ (Configuration Design Unit)

2.12.4.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้ เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้น ที่เรียกว่า หน่วยการออกแบบเอนทิตี ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบ กำหนดทิศ

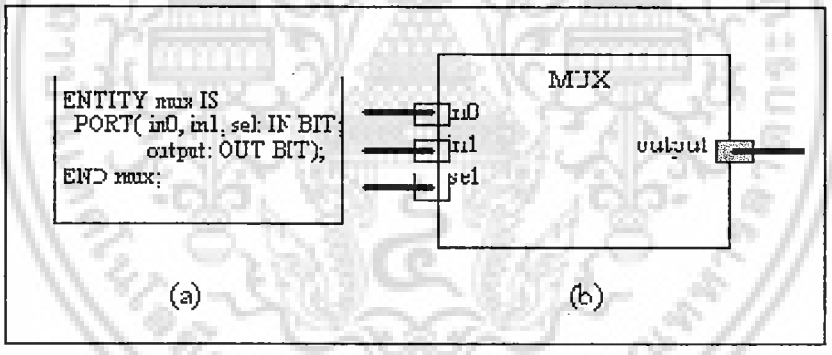
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหล่ผ่านจุดต่อเหล่านั้น รูปที่ 2.18 แสดงให้เห็นโครงสร้างอย่างง่ายๆ ของ หน่วยการออกแบบเอนทิตี

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

รูปที่ 2.19 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

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



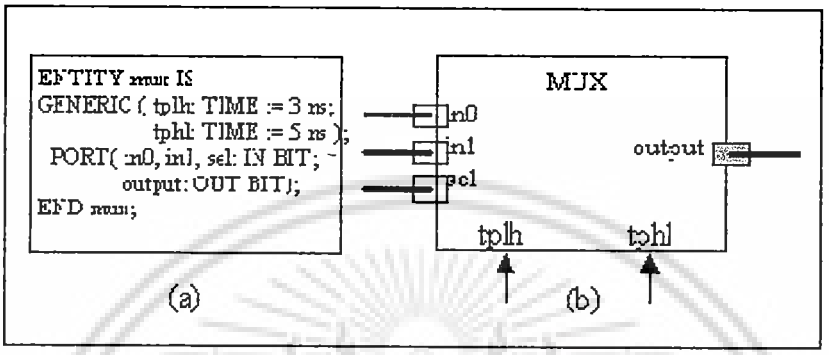
รูปที่ 2.20 แสดงรูปแบบของมัลติเพลกซ์
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล
(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

ในรูปที่ 2.19 เป็นหน่วยการออกแบบเอนทิตี ที่บรรยายอุปกรณ์ที่มีชื่อว่ามัลติเพลกซ์ หรือ MUX ในส่วนหัวของเอนทิตี มีการกำหนดจุดต่อ 4 จุดภายใต้ชุดคำสั่ง PORT โดยที่ 3 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ in0, in1, sel ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลเข้าของข้อมูล (IN) ที่แสดงด้วยรูปสี่เหลี่ยมโปร่งในรูปที่ 2.20 ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (OUT) ที่แสดงด้วยรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สี่เหลี่ยมทึบในรูปที่ 2.20 ส่วนประเภทของข้อมูลที่จะไหล เข้าและออก นั้นเป็นประเภท BIT ที่สามารถมีค่าได้เพียงสองค่าคือ '0' และ '1' เท่านั้น

นอกจากนั้น ผู้ออกแบบยังสามารถกำหนดค่าพารามิเตอร์ทางฟิสิกส์ที่เป็นข้อมูลเพิ่มเติมอื่นๆ ลงในส่วนหัวของเอนทิตีได้อีก เช่น ข้อมูลเกี่ยวกับความเร็วในการทำงานของอุปกรณ์ อันได้แก่ ค่าเวลาหน่วงแพร่กระจาย (Propagation delay time) พารามิเตอร์เหล่านี้ เรียกว่า เจนเนริก (Generic) ที่กำหนดด้วยคำสั่ง GENERIC จากตัวอย่างในรูปที่ 4



รูปที่ 2.21 รูปแบบมัลติเพลกซ์ที่ประกอบด้วยข้อมูลค่าเวลาหน่วงแพร่กระจาย
(a) หน่วยการออกแบบเอนทิตีในรูปของวีเอชดีแอล
(b) มุมมองของตัวเชื่อมประสาน

ในบางกรณีสามารถที่จะใช้ภาษาวีเอชดีแอล สร้างรูปแบบที่ปราศจากช่องทางไหล เข้าและออกของข้อมูล ได้ ซึ่งส่วนใหญ่จะพบในการสร้างรูปแบบ สำหรับตรวจสอบการทำงานของอีกรูปแบบหนึ่ง คือ วีเอชดีแอลสำหรับการทดสอบเปรียบเทียบ (Test bench)

```
ENTITY test_bench IS
END test_bench;
```

รูปที่ 2.22 หน่วยการออกแบบเอนทิตีที่ไม่มีการกำหนดช่องทางที่ต่อกับภายนอก

2.12.4.2 หน่วยการออกแบบสถาปัตยกรรม

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงานพฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดใน หน่วยการออกแบบเอนทิตี รูปที่ 2.22 แสดงให้เห็นถึงโครงสร้างอย่างง่ายๆ ของหน่วยการออกแบบสถาปัตยกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.23 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำ ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (OF <entity design unit> IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่นประเภท (type) ต่างๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ตัวคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า และไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง PORT) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า BEGIN กับ END ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (concurrent statement) เท่านั้น หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้นๆ ที่เป็นส่วนเผื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

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

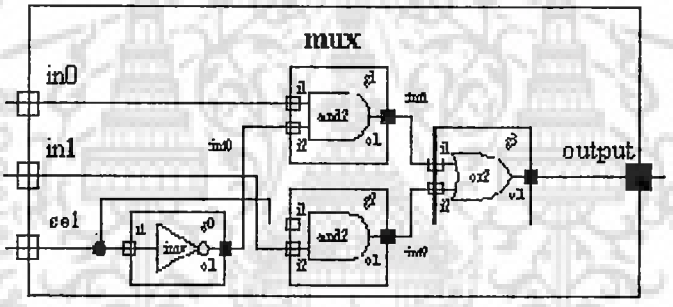
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.24 แสดงหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ตามฟังก์ชันบูลีน

$$output = (\overline{sel}.in0) + (sel.in1)$$

รูปที่ 2.23 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (*in0, in1*) กับข้อมูลที่ไหลออก (*output*) ประกอบด้วยชุดคำสั่งแบบแข่งขนานเพียงชุดเดียว ซึ่งเขียนเป็นประเภทการไหลของข้อมูลของมัลติเพลกซ์ หรือ ระดับการถ่ายโอนข้อมูลระหว่างเรจิสเตอร์ (RTL: Register Transfer Level)



รูปที่ 2.25 แสดงโครงสร้างภายในสถาปัตยกรรมของมัลติเพลกซ์

รูปที่ 2.24 เป็นหน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง โดยใช้ อินเวอร์เตอร์ (inv ที่ตำแหน่ง g0), แอนด์เกต 2 อินพุตจำนวน 2 ตัว (and2 ที่ตำแหน่ง g1 และ g2) และ ออร์เกต 2 อินพุต (or2 ที่ตำแหน่ง g3) มาสร้างตามฟังก์ชันบูลีนของรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.26 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทโครงสร้าง

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

รูปที่ 2.27 หน่วยการออกแบบสถาปัตยกรรมของมัลติเพลกซ์ประเภทพฤติกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าเขียนบรรยายส่วนของสถาปัตยกรรมของมัลติเพลกซ์ในลักษณะของ ประเภทพฤติกรรม ประเภทการไหลของข้อมูล ประเภทโครงสร้างหรือประเภทผสมที่นำเอาแต่ละประเภทมาเขียนไว้ในส่วนของสถาปัตยกรรม ก็ตามต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งนี้ก็เป็นข้อดีของภาษาวีเอชดีแอล

2.12.4.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดย หน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือ จากหน่วยการออกแบบแพ็กเก็ตอื่นๆ นอกจากนั้นสิ่งที่นิยมทำกันมากคือรูปแบบมาตรฐานต่างๆ เช่น อุปกรณ์มาตรฐาน (เช่น ไอซีตระกูล 74XX เป็นต้น) จะถูกเก็บไว้ในแพ็กเก็ต ที่ทุกคนสามารถเข้าถึง โดยปกติแล้ว แพ็กเก็ตจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กเก็ต (Package declaration) และ ส่วนของบอดี้แพ็กเก็ต (Package body) เนื่องจาก แพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอล สามารถกระทำได้ด้วยชุดคำสั่ง USE

1. Package declaration

ส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อ ของสิ่งที่ประกาศอยู่ภายในแพ็กเก็ต สำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเอง สิ่งใดๆ ถูกประกาศในส่วนของ ส่วนบอดี้แพ็กเก็ต แต่ไม่ถูกประกาศในส่วนการประกาศแพ็กเก็ต จะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือ จุดเชื่อมต่อ หรือ พอร์ต ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้ว แพ็กเก็ต สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้เช่น ใช้สำหรับประกาศ ชนิด (Type) หรือ สัญญาณ เช่นเดียวกันกับ ส่วนบอดี้แพ็กเก็ต ที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ต นั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

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

รูปที่ 2.28 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Package body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ต แล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ต ทั้งนี้รวมทั้ง การกำหนดค่าคงที่ต่างๆ อันได้แก่ตังคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต) ฉะนั้นส่วนบอดี้แพ็กเก็ต จึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเก็ต ไม่มีการประกาศชื่อ ที่เป็นโปรแกรมย่อย หรือ ค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.27

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

รูปที่ 2.29 โครงสร้างของบอดี้แพ็กเก็ต

2.12.4.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตี หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

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

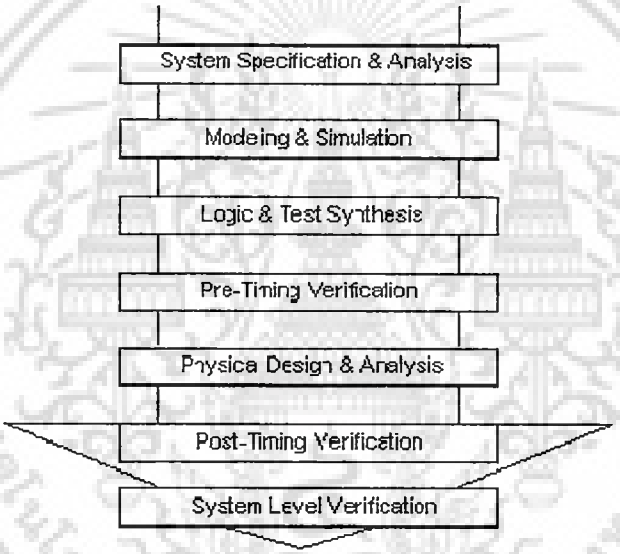
รูปที่ 2.30 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.12.5 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน เช่น วงจรรวม (ASIC: Application Specific Integrated Circuit) วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของของ บล็อกไดอะแกรมเสียก่อน ก่อนที่จะวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาติให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามที่ต้องการ และเพิ่มเติมในรายละเอียดที่ละชั้น นี้คือ หลักการออกแบบจากบนลงล่าง (Top-Down Design) ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Botom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากให้การออกแบบในลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอลกับหลักการออกแบบจากบนลงล่าง จึงเป็นทางออกให้กับวิศวกรออกแบบที่จะสามารถออกแบบและพัฒนา วงจรที่มีซับซ้อนได้มากขึ้น และช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ



รูปที่ 2.31 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.30 แสดงให้ถึงขั้นตอนของการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายๆ เทคโนโลยี เช่น พีแอลดี (PLD: Programmable Logic Device) อันได้แก่ พีแอลเอ (PLA: Programmable Logic Array), เอฟพีจีเอ (FPGA: Filed Programmable Gate Array), ซีพีแอลดี (CPLD: Cell Programmable Logic Device) เป็นต้น นอกนั้นยังมี เซมิคัลตัมไอซี (Semi-Custom IC) ได้แก่ เกตอะเรย์ (Gate array), เซลล์มาตรฐาน (Standard Cell) ขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษาวีเอชดีแอล หรือภาษาเอชดีแอลอื่นๆ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3. หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจรหรือสังเคราะห์ ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกต และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสท์ (Netlist) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในรูประดับเกตหรือเน็ทลิสท์แล้ว ข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงาน ในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี เวลาหน่วงของการแพร่กระจาย (Propagation delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ นาโนวินาที (10^{-9} นาที) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่างๆ จำนวน 10,000 เกต ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรรวมทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้า ที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวม (ASIC)

6. หลังจากที่ได้วงจรจริงมาแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วยอินพุตและเอาต์พุตแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้ง เป็นการควบคุมคุณภาพของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Sobel edge detection ประสิทธิภาพสูง

ในปัจจุบัน การประมวลผลภาพแบบเวลาจริง (Real time) โดยใช้ตัวกรองดิจิตอล 2 มิติ (2-D Digital filter) ได้ก้าวหน้าไปอย่างรวดเร็วจากการนำไปใช้งานด้านต่างๆ เช่น การประมวลผลด้านฟิสิกส์ธรณีวิทยา (geophysical) และการประมวลผลภาพทางชีวะการแพทย์ (biomedical) อย่างไรก็ตาม ในการออกแบบตัวกรองดิจิตอลแบบ 2 มิติ (2-D Digital filter) ต้องใช้ตัวคูณจำนวนมากในการคำนวณค่าเอาท์พุทหนึ่งค่า เพราะว่ามันมีค่าสัมประสิทธิ์ที่ไม่ใช่ศูนย์อยู่เป็นจำนวนมาก จำนวนของตัวคูณจะไปลดความเร็วของช่วงเวลาในการประมวลผล จนการประมวลผลสัญญาณลดลงเนื่องจากกระบวนการหาค่าผลคูณต้องใช้ขั้นตอนในการทำงานที่มาก (คือการบวกและเลื่อนข้อมูลหลายๆครั้ง) นอกจากนั้นยังกินพื้นที่ของวงจรในการออกแบบ (พิจารณาในแง่การออกแบบวงจรรวม) [8-11] จึงได้เสนอการออกแบบโครงสร้างการกระจายทางคณิตศาสตร์ (DA) สำหรับตัวกรองดิจิตอลแบบ 2 มิติ (2-D digital filter) ให้ไม่มีตัวคูณอยู่ในตัวกรองดิจิตอลแบบ 2 มิติ (2-D digital filter)

3.1 แนวความคิดพื้นฐานของ Sobel edge detection

The first order derivative หรือ gradient operator ของภาพ $f(x,y)$ อธิบายได้โดย

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.1}$$

ค่าสำคัญ ที่ใช้สำหรับตรวจจับขอบภาพคือ Magnitude of gradient คือ

$$mag(\nabla f) = |\nabla f| = \left[G_x^2 + G_y^2 \right]^{1/2} \tag{3.2}$$

ในการกำหนดสมาชิก สำหรับการตรวจจับขอบภาพแบบโซเบลจะสมมุติรูปภาพอินพุทให้มีขนาด 3×3 ตามที่ปรากฏในเมตริกซ์นี้

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

โดย $z_5 = f(x,y)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดที่ใช้สร้าง Sobel edge detection มีพื้นฐานมาจาก 1st order derivative ของตัวกรองภาพ low-pass แบบ Gaussian lowpass filter [5] ดังนั้น 1st order derivative ที่ตำแหน่ง z_5 คือ

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{3.3 a}$$

และ

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{3.3 b}$$

จากตัวอย่าง จะได้ สมการที่ 3.3 a และสมการ 3.3 b สามารถเขียนสมาชิกของหน้ากากที่ใช้สำหรับการตรวจจับขอบภาพแบบโซเบล (Sobel edge detection) ได้ดังนี้

-1	-2	-1
0	0	0
1	2	1

Horizontal Gradient (G_x) Kernel

-1	0	1
-2	0	2
-1	0	1

Vertical Gradient (G_y) Kernel

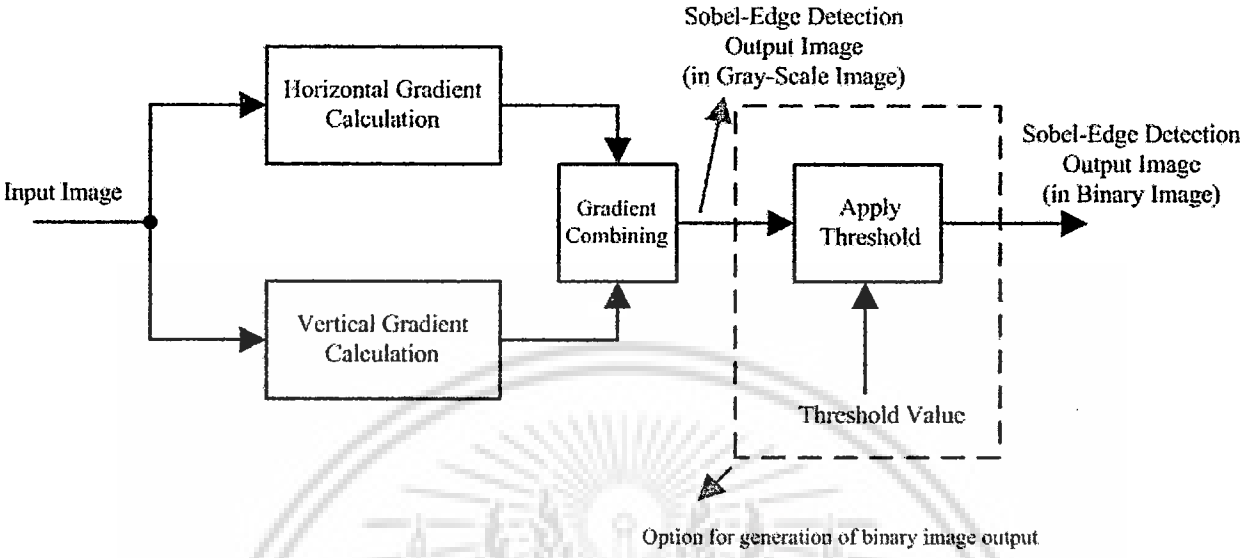
ในทางปฏิบัติ เพื่อหลีกเลี่ยงจำนวนเชิงซ้อนของตาราง และ square root เช่นในสมการที่ 3.2 สามารถเขียนค่า magnitude of gradient ในสมการที่ 3.2 ได้เป็น

$$|\nabla f| \approx |G_x| + |G_y|$$

ขั้นตอนนี้เรียกว่าการหาผลรวมของเกรเดียนท์ (Gradient Combining)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถสรุปการทำงานของการตรวจจับขอบภาพแบบโซเบล (Sobel edge detection) ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 การตรวจจับขอบภาพแบบโซเบล (Sobel Edge Detection)

สำหรับระบบคอซอล (Causal system) ของสมการคอนโวลูชั่นแบบ 2 มิติ (2-D convolution) ในการคำนวณภาพผลลัพธ์ของสมาชิก ขนาด 3×3 คือ

$$y(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}h(k,l)\,x(m-k,n-l) \tag{3.4}$$

โดย $x(m,n)$ และ $y(m,n)$ เป็นสัญญาณภาพของ input และ output, และให้ $h(k,l)$ เป็นสัมประสิทธิ์ที่อยู่ในแต่ละหน้ากาก

การตรวจจับขอบภาพแบบโซเบล (Sobel edge detection) จะได้ 2 สูตร ดังนั้นสมการทั้ง 2 ต้องการคารคำนวณจากค่า vertical and horizontal gradient ดังแสดงข้างล่าง

$$y_{hor}(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}h(k,l)\,x(m-k,n-l) \tag{3.5 a}$$

และ

$$y_{ver}(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}v(k,l)\,x(m-k,n-l) \tag{3.5 b}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากพิจารณาให้ดีจากสมการที่ 3.5 a และ 3.5 b ตัวตรวจจับขอบภาพแบบโซเบล ต้องใช้ตัวคูณ 18 ตัวคูณ และตัวบวก 16 ตัวบวก

3.2 แนวความคิดโครงสร้างตัวกรอง Sobel edge detection ประสิทธิภาพสูง

การตรวจจับขอบภาพแบบโซเบล (Sobel edge detection) ยังคงถูกใช้อย่างแพร่หลายเพื่อหาขอบภาพในการประยุกต์ใช้งานการประมวลผลสัญญาณภาพดิจิตอล (Digital image processing) แนวความคิดเดิมของ Sobel edge detection จะมีพื้นฐานจากเรื่อง gradient (หรือ 1st order derivative) ของ Gaussian low-pass filter และสามารถแปลงไปเป็น Sobel edge detection ที่มีสมาชิกขนาด 3×3 [5] สมาชิกของ Sobel edge detection คือการนำ vertical gradient convolution kernel และ horizontal gradient convolution kernel มาหารกัน. จนกระทั่งสมาชิกมีขนาด 3×3 หลังจากนั้นขั้นตอนการออกแบบจึงเริ่มออกแบบจากสมาชิกเหล่านั้น และใช้สมการ 2-D convolution เพื่อพิสูจน์โดยตรง แต่ละสมาชิก (kernel) จะต้องใช้จำนวนตัวคูณถึง 9 ตัว และจำนวนตัวบวกถึง 8 ตัว ดังนั้นตัวกรองดิจิตอลขนาด 2 มิติ (2-D digital filter) จึงต้องการตัวคูณถึง 18 ตัว และตัวบวกถึง 16 ตัว เพื่อการพิสูจน์ตัวกรอง Sobel edge detection โดยตรง จาก 2 สมการที่แตกต่างกัน (โดยไม่ครอบคลุม gradient combining)

วิธีที่เป็นที่นิยมที่สามารถใช้แก้ปัญหาของตัวคูณจำนวนมากๆ ในการพิสูจน์สมการผลต่างคือการกระจายทางคณิตศาสตร์ (DA) [1-4] ส่วนสำคัญของการดำเนินงานของ DA คือ ROMs (หรือหน่วยความจำชนิดอื่นๆ) และ scaling accumulator ดังนั้นในส่วนของการกระจายทางคณิตศาสตร์ (DA) จึงถูกเรียกว่า ROM-Accumulator structure ถึงแม้ว่า การดำเนินการของ DA จะไม่ต้องการตัวคูณ แต่ DA ต้องใช้หน่วยความจำอย่างเช่น ROM เพื่อเก็บค่าต่างๆ ของผลลัพธ์ที่คำนวณได้เป็นค่าย่อยๆ [2-4] สำหรับการพิสูจน์ Sobel edge detection นั้น ต้องใช้ ROMs จำนวน 2 ตัว เพื่อดำเนินการ ROM ตัวที่ 1 สำหรับการคำนวณค่าย่อยในส่วน vertical gradient และ ROM อีก 1 ตัวสำหรับการคำนวณค่าย่อยในส่วน horizontal gradient ซึ่ง ROM แต่ละตัวต้องการ 512 addresses เพื่อเก็บค่าต่างๆที่เป็นไปได้ทั้งหมดของ output หน่วยความจำ ROMs addresses คือ Input ที่เป็นสัญญาณภาพและมันทำให้เกิด delayed index. Output ของ ROMs แต่ละตัวต้องคำนวณค่า vertical gradient output และ horizontal gradient output โดยใช้ scaling-accumulatorในการคำนวณ ซึ่งต้องใช้ scaling-accumulator 2 ชุดสำหรับ ROMs 2 ตัว

แต่โครงสร้างใหม่ของตัวกรอง Sobel edge detection ต้องการเพียง 14 ตัวบวกเพื่อการคำนวณเท่านั้น (โดยไม่มีตัวคูณ ซึ่งเหมือนกับการกระจายทางคณิตศาสตร์พื้นฐาน DA-based จะเสนอการออกแบบ hardware และทำการเปรียบเทียบกับ DA-based สำหรับ 4 ส่วนของ 2nd order 1-D DPF (วงจรกรองปาสคาลเต็มหน่วย) ถูกนำมาใช้เพื่อหา hardware ของโครงสร้างตัวกรอง Sobel edge detection, 2nd order DPF ในส่วนอื่นๆเป็นโครงข่ายของตัวบวก 3 ตัว และตัวบวกอื่นๆอีก 2 ตัว ใช้สำหรับรวบรวมคู่ของ 2nd DPF เพื่อผลิต gradient output อื่นๆ ดังนั้นโครงข่ายตัวบวกที่มีประสิทธิภาพสูงสำหรับ Sobel edge detection จึงเป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การใช้การกระจายทางคณิตศาสตร์ (DA) เพื่อออกแบบ Sobel Edge Detection แบบปราศจากตัวคูณ

การพิสูจน์ โครงสร้าง 2 มิติ ของการกระจายทางคณิตศาสตร์ในตัวกรองดิจิตอล สามารถอธิบายได้ตามนี้ พิจารณา ตัวกรองเชิงเลข 2 มิติ บรรยายโดยสมการ linear difference equation ดังสมการที่ 3.4 อีกครั้ง

$$y(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}x(m-k,n-l)\,h(k,l) \tag{3.6}$$

โดยให้ x(m,n) และ y(m-k,n-l) เป็นสัญญาณภาพด้าน input และ output, และให้ h(k,l) เป็น filter coefficients kernel ขนาด 3×3

สัญญาณทั้งหมดมีขอบเขต ± 1 และกำหนดให้สัญญาณ input ในรูปจำนวนเชิงซ้อน (2's complement format) B bits โดย

$$x(m,n)=\sum_{j=1}^{B-1}x_j(m,n)\,2^{-j}-x_0(m,n) \tag{3.7}$$

โดยที่ $x_j(m,n)$ คือ $x(m,n)$ ในระดับบิท ดังนั้นสมการที่ (3.6) สามารถเขียนใหม่ได้เป็น

$$y(m,n)=\sum_{k=0}^{2}\sum_{l=0}^{2}h(k,l)\left(\sum_{j=1}^{B-1}x_j(m-k,n-l)\,2^{-j}-x_0(m-k,n-l)\right) \tag{3.8}$$

เราจัดเรียงสมการที่ 3.8 ใหม่ ได้เป็น

$$y(m,n)=\sum_{j=1}^{B-1}\left[\sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)x_j(m-k,n-l)\,2^{-j}\right)\right]-\sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)x_0(m-k,n-l)\right) \tag{3.9}$$

กำหนดฟังก์ชั่น $F(\cdot)$ ได้ดังนี้

$$F_j\left(x_j(m,n),x_j(m,n-1),\cdots,x_j(m-2,n-2)\right)=h(0,0)x_j(m,n)+h(0,1)x_j(m,n-1)+\cdots+h(2,2)x_j(m-2,n-2) \tag{3.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

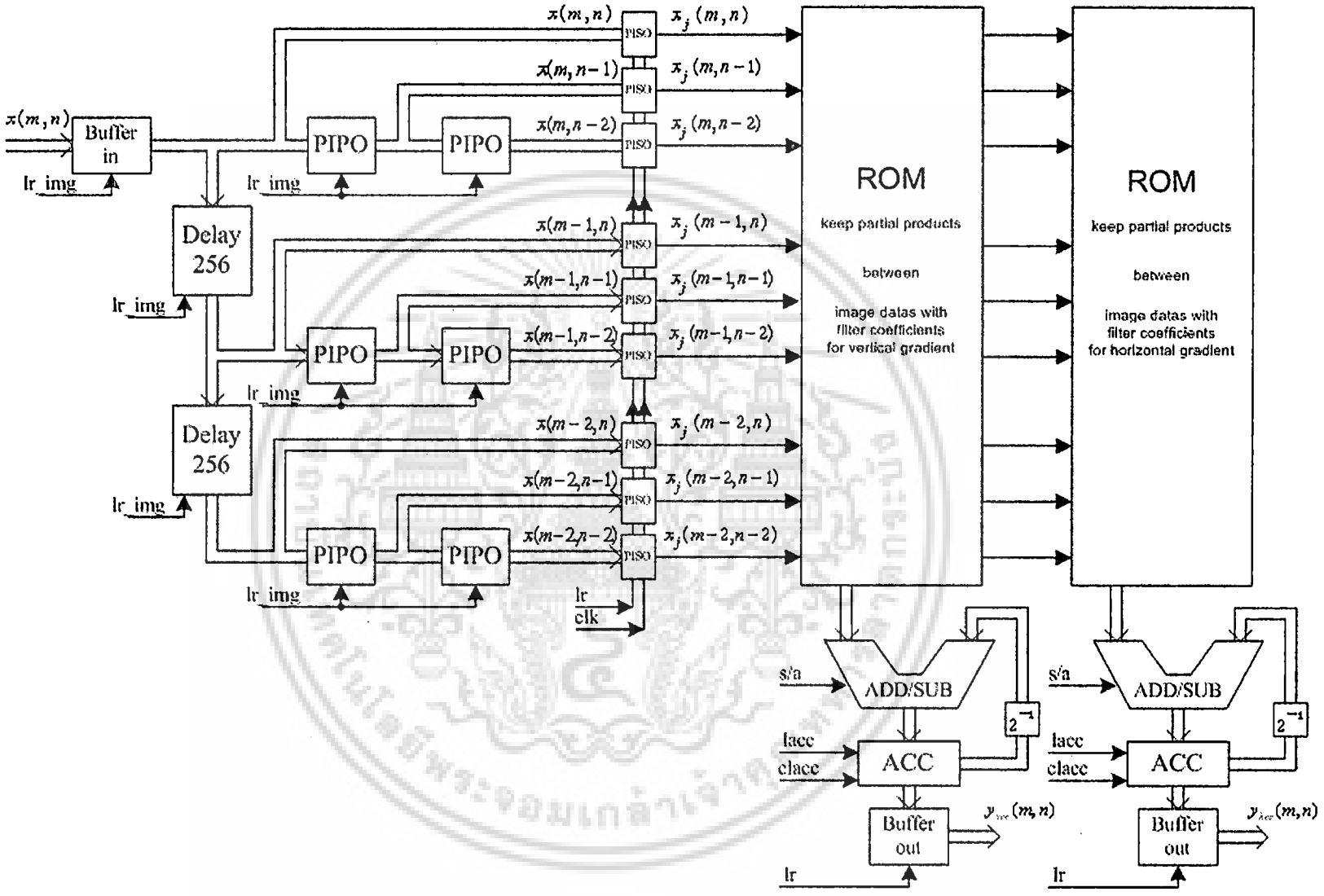
แต่ละ $x_j(m\text{-}k,n\text{-}l)$, ซึ่ง $j = 0,1,\ldots, B\text{-}1$ เป็นเพียง 0 หรือ 1 เท่านั้น, เรายังคำนวณค่าของฟังก์ชั่นนั้นที่เรียกว่าผลคูณย่อย (partial products) ที่ถูกเก็บไว้ในหน่วยความจำ และใช้สัญญาณ input signals เป็นตัวกำหนด addressing ดังนั้น ผลลัพธ์สามารถคำนวณได้โดยใช้การเลื่อน และการบวกของ partial products ดังในสมการที่ (3.11)

$$y(m,n) = \sum_{j=1}^{B-1} F_j(\cdot)2^{-j} - F_0(\cdot) \tag{3.11}$$

สมการนี้สามารถเขียนใหม่ให้อยู่ในรูป hardware โดยใช้โครงสร้างเลขคณิตกระจาย (distributed arithmetic : DA) ดังแสดงในรูปที่ 3.2 และสามารถแสดงให้เห็นว่าตัวคูณไม่ได้ถูกใช้ในโครงสร้างนี้

ดังนั้น จากสมการที่ 3.5 a และ 3.5 b โครงสร้างสถาปัตยกรรมของเลขคณิตกระจาย ของตัวตรวจจับขอบภาพแบบโซเบล สามารถแสดงได้ดังรูปที่ 3.2 โครงสร้างนี้ต้องการสัญญาณควบคุมเพื่อใช้ควบคุมวงจรย่อยต่างๆที่อยู่ด้านใน และสัญญาณภาพที่ป้อนเข้ามาจะกระจายลงไปในระดับ bit ก่อนที่จะใช้ไปใน memory addresses. รายละเอียดเพิ่มเติมของตัวตรวจจับขอบภาพแบบโซเบลโดยใช้การกระจายทางคณิตศาสตร์เพื่อ สามารถดูได้ใน [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2 โครงสร้างของการกระจายทางคณิตศาสตร์เพื่อตรวจจับขอบภาพแบบโซเบล [4]

3.4 การใช้ Discrete Pascal Filter เพื่อออกแบบ Sobel Edge Detection แบบปราศจากตัวคูณ

การนำเสนอโครงสร้างตัวตรวจจับขอบภาพแบบโซเบล ซึ่งพื้นฐานที่เรียกว่า low-pass-type 2^{nd} order 1-D DPF (Discrete Pascal Filter: DPF) คือรูปแบบการกรองที่ย่อยมาจาก discrete Pascal transform (DPT) มีสองประเภทที่เป็นรูปแบบย่อย high-pass และ low-pass, โดยขึ้นอยู่กับประเภทของ Pascal transformation matrix. Pascal transformation matrix สามารถแปลงไปเป็น binary matrices ได้ โดยผลของการแปลง Pascal matrix factorization ทำให้สามารถเขียนประสิทธิภาพของ hardware for 1-D DPT ได้ใหม่ซึ่งเรียกว่า DPT butterfly unit. โครงสร้างนี้สามารถทำงานได้โดยปราศจากการคูณและใช้เพียงการบวกเท่านั้น, ซึ่งสามารถลดความซับซ้อนในขั้นตอนการคำนวณได้ โครงสร้าง DPF ได้ถูกดัดแปลงมาจากโครงสร้าง DPT ด้วยการวางอุปกรณ์ delay elements ให้เกิดความสัมพันธ์กันในแต่ละ input [6-7]

Discrete Pascal Transforms X ของสัญญาณ vector x หนึ่งมิติ สามารถเขียนได้ดังนี้

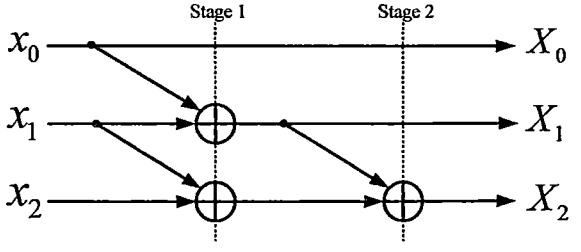
$$X = Px \tag{3.12}$$

ให้ Xคือ Output vector ที่ถูกแปลง, x คือ input vector และ P คือ Pascal transformation matrix

ตัวอย่าง 3-point low-pass type DPT และ 3×3 low-pass type Pascal transformation matrix ใช้ดังนี้

$$P_3^{(LP)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.13}$$

จากผลการวิจัย [17] ที่ผ่านมา แสดงถึงการคิดโครงสร้างที่มีประสิทธิภาพของ 3-point low-pass type DPT ซึ่งแสดงดังในรูปที่ 3.3 โครงสร้างนี้ได้ถูกแปลงมาจาก Pascal transformation matrix factorization ไปเป็น binary matrices (สามารถดูรายละเอียดเพิ่มเติมได้ที่ [17])



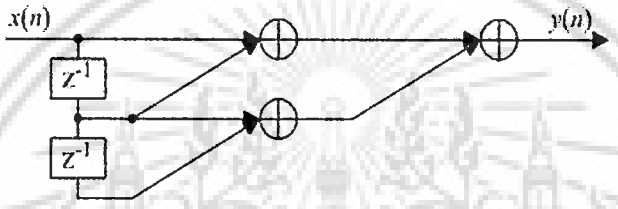
รูปที่ 3.3 The 3-point Lowpass Type DPT Flow Graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ the 2nd order low-pass type 1-D DPF, มีรูปแบบที่เป็น low-pass filtering สามารถดูรายละเอียดได้ใน [18] และสามารถสรุป 2nd order transfer function ได้ในสมการที่ 3.14

$$H^{(LP)}(z) = 1 + 2z^{-1} + z^{-2} \tag{3.14}$$

ดังนั้น DPF สามารถแปลงจาก DPT โดยให้ความสัมพันธ์ระหว่าง inputs และพิจารณาเฉพาะที่ 1 output เท่านั้น การใช้ DPF ในการควบคุมการคำนวณจะเปลี่ยนจากการคูณในรูป matrix ใน DPT ไปอยู่ในรูปของ DPF ด้วยการเปลี่ยนแปลงเล็กน้อย, การคิด filter structure ของ 1-D DPF of low-pass type สามารถเขียนได้ใหม่ตามข้างล่างนี้

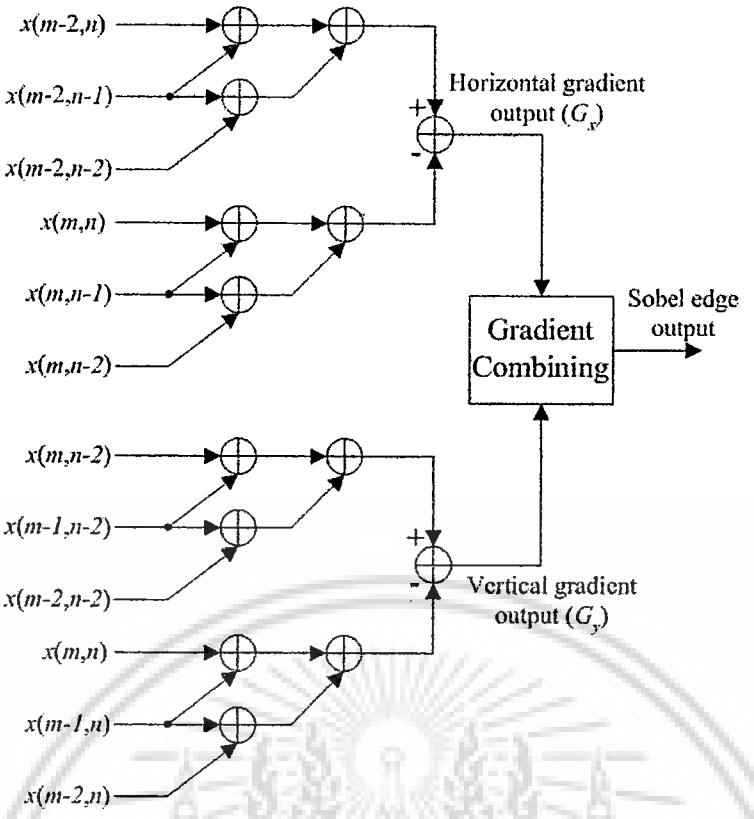


รูปที่ 3.4 The 2nd order Low-pass Type 1-D DPF

การเสนอโครงสร้างตัวกรองของเราจะเริ่มต้นจากนิยามของ gradient operator ซึ่งจะทำงานที่ภาพอินพุทไม่ชัดเจน จาก Gaussian low-pass filtering สามารถแสดงได้ตามสมการที่ 3.3 a และ 3.3 b จากจุดนั้น เราสามารถใช้แนวคิดของ low-pass-type 1-D DPF ซึ่งเมื่อเทียบกันกับ Gaussian low-pass filtering establish จะได้ตัวกรอง Sobel edge detection ปราศจากตัวคูณแบบใหม่

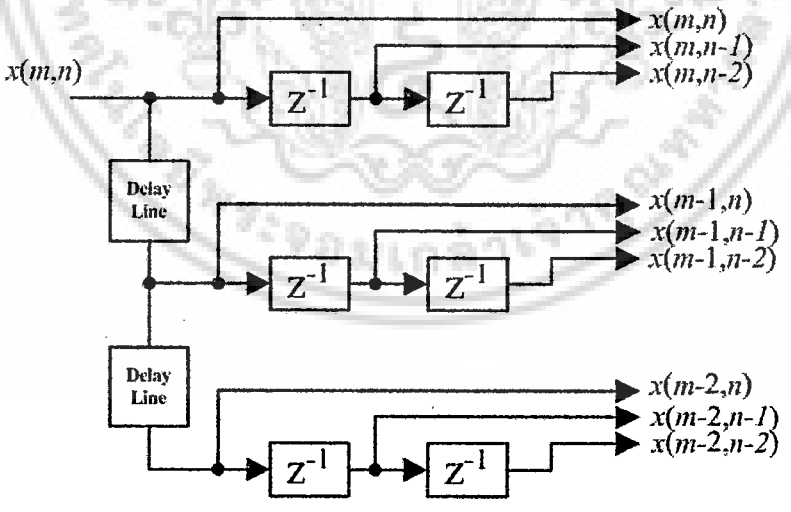
เนื่องจากการทำงานของ DPF มีลักษณะคล้าย Gaussian filtering (บางครั้ง, อาจเรียกได้ว่า binomial filter ซึ่งใกล้เคียงกับชนิดของ Gaussian filter) แต่สามารถเขียนให้มีโครงสร้างที่สวยงามโดยปราศจากการคูณได้ ดังนั้นจึงเสนอโครงสร้างของตัวกรอง Sobel edge detection ใหม่แบบไม่มีตัวคูณและมีเพียงตัวบวกแค่ 14 ตัวเท่านั้นที่ใช้ในการคำนวณ vertical gradient output และ horizontal gradient output ดังแสดงในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 The Proposed Multiplerless Sobel Edge Detection Filter

ในรูปที่ 3.5 สัญญาณภาพ input ทั้งหมดถูกป้อนมายัง adder network จะได้รับการสร้างและเรียงวงจรขึ้นใหม่ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 Input Image Arrangement Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

เมื่อทำการเปรียบเทียบโครงสร้างที่ได้เสนอโดยการพิสูจน์โดยตรง โดยการใช้ 2 difference equations โครงสร้างที่เสนอนั้นมีภาระในการคำนวณต่ำมาก ในขณะที่หากคิดแบบเดิมจะต้องใช้ 18 ตัวคูณ และ 16 ตัวบวก. สำหรับการเปรียบเทียบกับการดำเนินการแบบ DA โครงสร้างที่เราเสนอไม่ต้องใช้ ROMs ใดๆ ในการเก็บค่าย่อยที่ได้จากการคำนวณ และค่าต่างๆในการทำงาน (ใช้เพียง adder network) คือง่ายกว่า DA scaling-accumulator operation. ในตารางที่ 4.1 แสดงการเปรียบเทียบการคำนวณจำนวนเชิงซ้อน โดยพิจารณาเฉพาะส่วนที่ใช้ในการคำนวณเท่านั้น (ไม่รวมถึงการจัดการภาพทางด้านอินพุท)

ตารางที่ 4.1 ผลการใช้ทรัพยากร

	Direct (Eq. 5)	**DA-Based**	**Proposed**
No. of Multipliers	18 multipliers	None	None
No. of Adders (2-input)	16 adders	None	14 adders
No. of Memory bits	None	8,192 bits (to store pre-computed values)	None
No. of bit-serial shifter	None	9 shift registers	None
No. of Scaling Accumulator	None	2 Scaling Acc. Units	None

ตารางที่ 4.2 แสดงผลการเปรียบเทียบการพิสูจน์ระหว่าง hardware ของ multiplierless Sobel edge detection filter โดยใช้ DA-based และโครงสร้างที่นำเสนอโดยใช้กระบวกการภาพอินพุทขนาด 256×256. ใช้บอร์ดของ EPF10K20RC240-4 Altera's FPGA ในการดำเนินการโดยมีภาษา VHDL เป็นสื่อ

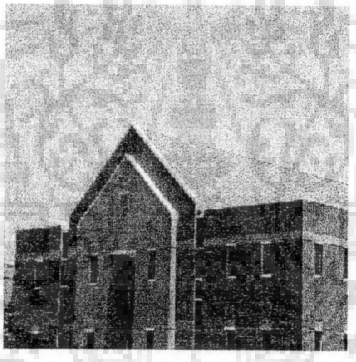
ตารางที่ 4.2 แสดงผลการเปรียบเทียบ Synthesized Circuits Using EPF10K20RC240-4 Altera's FPGA Device

	DA-Based Sobel edge Detector [4]	**Proposed Sobel edge Detector**
No. of Embedded Array Blocks (EABs)	6 EABs (12,288 Memory Bits)	2 EABs (4,096 Memory Bits)
No. of Logic Elements (LEs)	280 LEs	244 LEs
Maximum Frequency	37 MHz	32 MHz

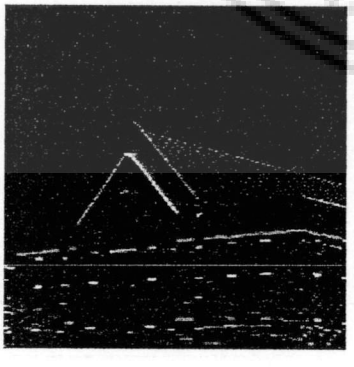
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 DA-based Sobel edge detection filter ต้องใช้ 4 EABs (Embedded Array Block) ในตัว FPGA เพื่อทำหน้าที่ memory (ROM) สำหรับเก็บค่าย่อที่คำนวณได้ของผลิตภัณฑ์ย่อย, และ 2 EABs เพื่อใช้ทำหน้าที่เป็น delay line (RAM) ในวงจรจัดการภาพด้านอินพุท. โครงสร้างที่นำเสนอต้องการเพียง 2 EABs for delay line (1 EAB equivalent to 2,048 memory bits) เท่านั้น วงจรส่วนอื่นที่ออกแบบโดย VHDL เช่น shift register, adder, scaling accumulator จะใช้ LEs (Logic Elements) ในการดำเนินการ ซึ่งโครงสร้างตัวกรองที่เสนอไปนั้นใช้น้อยกว่า DA-based. ประสิทธิภาพด้านความเร็วของการทำงานของวงจรจะแสดงที่ความถี่สูงสุด โครงสร้างตัวกรองที่นำเสนอนั้นมีความถี่ maximum สูงน้อยกว่าความถี่ของ DA-based. อย่างไรก็ตาม ค่าความถี่ maximum ของโครงสร้างตัวกรองที่นำเสนอสามารถปรับปรุงโดยใช้ตัวบวกประเภทอื่น ซึ่ง such carry-look ahead adder instead of carry-ripple adder (ซึ่งจะใช้ในการทำงานนี้) ใช้งานร่วมกันด้วยดีในด้านเทคนิค.

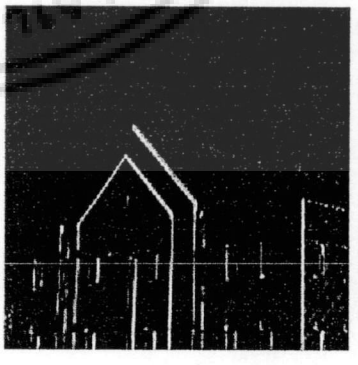
ผลการ Simulation ของ Sobel edge detection แสดงได้ดังรูปที่ 4.2-4.4 ตามลำดับ



รูปที่ 4.1 Original input image



รูปที่ 4.2 Horizontal gradient output



รูปที่ 4.3 Vertical gradient output

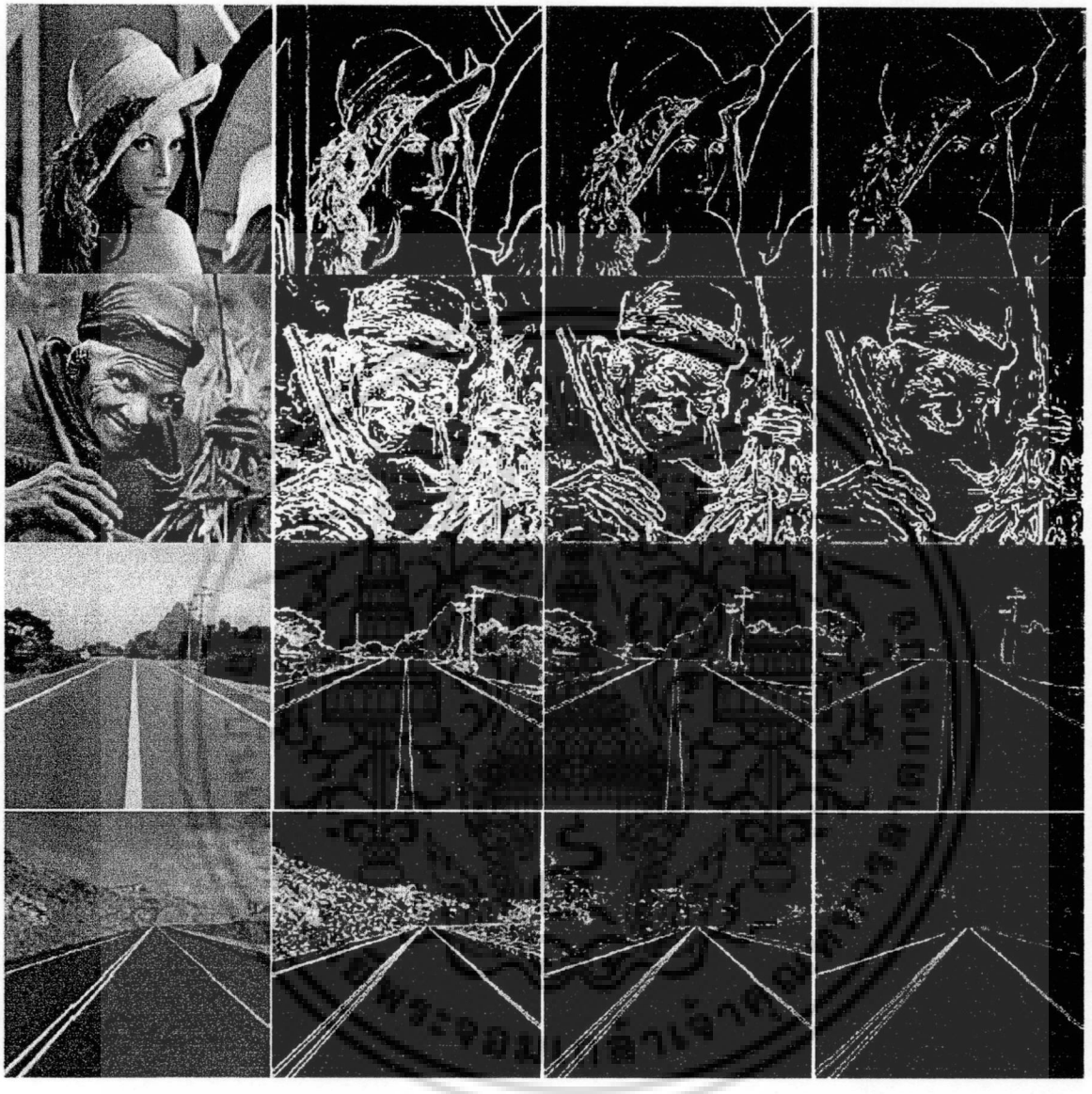
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 Sobel edge detection output image

จาก input image ในรูปที่ 4.2 และรูปที่ 4.3 แสดง horizontal gradient และ vertical gradient output, ตามลำดับ. Sobel edge detection output หลังจากการรวม gradient จะแสดงในรูปที่ 4.4 เมื่อรวมทั้งหมด, direct realization, DA-based และการนำเสนอโครงสร้าง Sobel edge detection filter จะให้ผลเดียวกันเหมือนกับ edge detected output, ความแตกต่างคือจะเป็นกังวลว่าต้องทำอย่างไรให้โครงสร้างถูกต้องซึ่งค่อนข้างยาก การนำเสนอ Sobel-edge detection filter มีความซับซ้อนต่ำ

จากรูปที่ 4.5 นี้ขอใช้ภาพถนนหลายๆ ภาพ ที่มีเส้นกึ่งกลางถนนในลักษณะที่เป็นเส้นตรงเพื่อให้เห็นผลการจำลองภาพที่ออกมา จะได้เห็นความแตกต่างกันมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ผลลัพท์จากอัลกอลิธึมของ Sobel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

ตัวกรอง Sobel edge detection แบบปราศจากตัวคูณที่นำเสนอไปนั้น, วิธีการในการคิดโครงสร้างนี้ ขึ้นอยู่กับการหา gradient ของภาพเบลอโดย Gaussian lowpass filter. lowpass ชนิด 1-D DPF สามารถใช้การคิด Sobel edge detector ในแบบที่นำเสนอด้วยโครงสร้าง hardware ที่ง่าย และโครงสร้างมีภาระในการคำนวณที่ไม่สูงมาก. โครงสร้าง Hardware ที่ได้นี้เหมาะสำหรับการนำไปใช้ออกแบบบน ASIC (Application Specific IC) เพื่อให้ได้ชิพเดี่ยวแบบ Sobel edge detection ด้วยการประมวลผลที่ความเร็วสูงและสามารถลดการสิ้นเปลืองพลังงานโดยรวมได้อีกด้วย (since using less hardware components)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Rafael C.Gonzalez and Richard E.Wood, "Digatal Image Processing," Prentice Hall, 2001.

[2] J.S.Lim, "Two-Dimensional Signal and Image Processing," Prentice Hall, 1990.

[3] S.Brown and Z.Vrenesic, "Fundamentals of digital logic with VHDL," McGraw-Hill, 2000.

[4] Douglas L.Perry and Thomas J.wilderotter, "A Designer's Guide to VHDL Synthesis," Kluwer Academic Publishers, 1998.

[5] S.Sjoholm and L.Lindh, "VHDL for Designers," Prentice Hall, 1997.

[6] The MathWorks, "Image Processing Toolbox User's Guide," 2001.

[7] K.S.Lin, "Texas Instruments Digital Signal Processing," Prenctice Hall, pp.(F)2-(F)9, 1998.

[8] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," *IEEE Trans. on ASSP.*, Vol. ASSP-22, No. 6, pp. 456-462, December 1974.

[9] J. Jaggernauth, A. C. P. Loui and A. N. Venetsanopoulos, "Real-Time Image Processing by Distributed Arithmetic Implementation of Two-Dimensional Digital Filters," *IEEE Trans. on ASSP.*, Vol. ASSP-33, No. 6, pp. 1546-1555, December 1985.

[10] S. Thongplew and K. Dejhan, "Implementation of 2-D Digital Filter-based Distributed Arithmetics," *Ladkrabang Engineering Journal.*, Vol. 11, No. 1, pp. 1-6, June 1994.

[11] S. Chivapreecha, K. Dejhan and C. Pienvijarnpong, "Hardware Implementation of Sobel-Edge Detection Distributed Arithmetic Digital Filter," *Proc. the 25th Asian Conference on Remote Sensing (ACRS'2004)*, Chiang-Mai, Thailand, pp.284-289, November 22-26, 2004.

[12] อภิรักษ์ ปานชื่น และ อมรเทพ สนินัด, "วงจรกรองสัญญาณเชิงเลขที่ใช้โครงสร้างเลขคณิตกระจายด้วยอุปกรณ์ FPGA," คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ.2546

[13] อนุชิต ธูปเหลือง, "การสร้างแผงวงจรแปลงสัญญาณภาพที่มีตัวประมวลผลสัญญาณดิจิตอล," คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ,พ.ศ.2546

[14] ชำนาญ ปัญญาใส และ วัชรากร หนูทอง, "ภาษาวีเอชดีแอล สำหรับการออกแบบวงจรดิจิตอล," ซีเอ็ดยูเคชั่น, กรุงเทพฯ, พ.ศ.2547

[15] บุญอนันต์ เกียงเอีย, "การออกแบบเครื่องจับสัญญาณภาพทางการแพทย์โดยใช้เอฟพีจีเอ," คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ,พ.ศ.2545

[16] ณรงค์ ทองฉิม และ เจริญ วงษ์ชุ่มเย็น, "ออกแบบไอซีดิจิตอลด้วย FPGA และ CPLD ภาคปฏิบัติ โดยใช้ภาษา VHDL ซอฟท์แวร์ตระกูร ISE WebPACK," ซีเอ็ดยูเคชั่น, กรุงเทพฯ, พ.ศ.2552

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[17] N. Ronnarongrit, S. Chivapreecha and K. Dejhan, "Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization," *Proc. The 4th International Colloquium on Signal Processing and Its Application (CSPA 2008)*, Kuala Lumpur, Malaysia, March 7-9, 2008

[18] S. Chivapreecha, U. Nithirochananont, and K. Dejhan, "Investigation of Frequency Characteristic in Discrete Pascal Transform and Its Applications," *Proc. The 4th International Colloquium on Signal Processing and Its Application (CSPA 2008)*, Kuala Lumpur, Malaysia, March 7-9, 2008.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เทคโนโลยี เอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

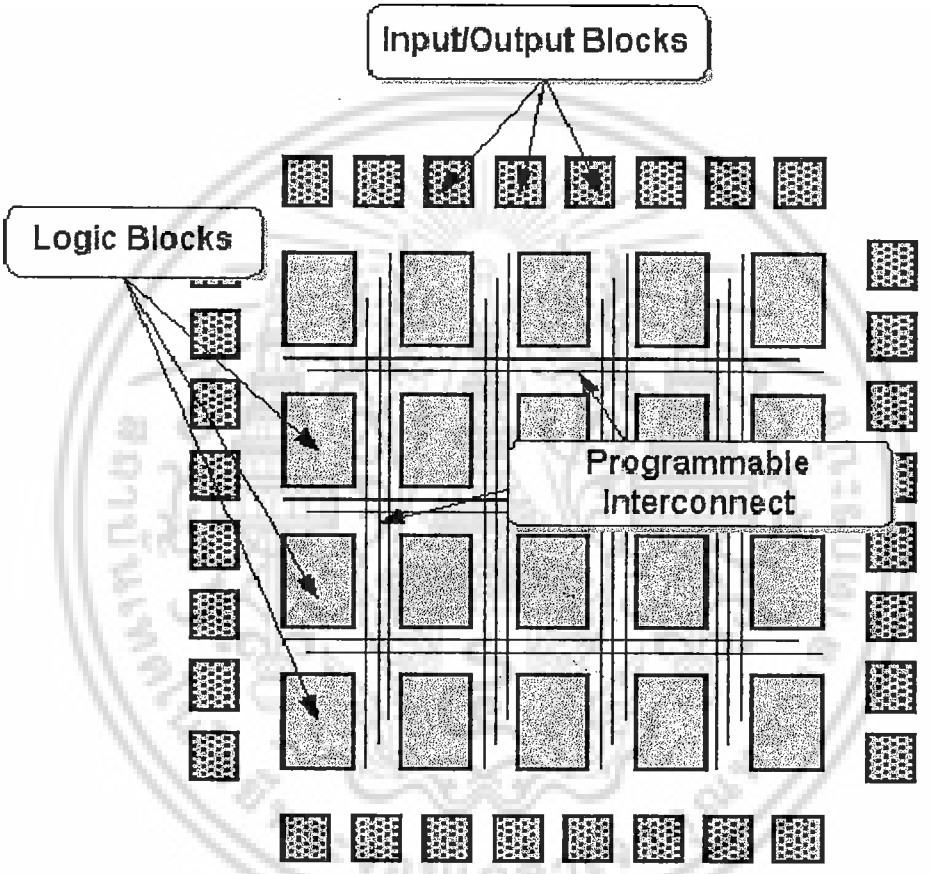
องค์ประกอบพื้นฐานของอุปกรณ์ FPGA จะกล่าวถึงรายละเอียดของส่วนต่างๆ ดังนี้

1. Programming Technology

2. Logic Cell

3. Programmable I/O Cells

4. Programmable Interconnects



โดยที่จะมีการยกตัวอย่าง FPGA ชิพที่มีการใช้งานจริงประกอบกันด้วย

1. Programming Technology (เทคโนโลยีการโปรแกรมอุปกรณ์)

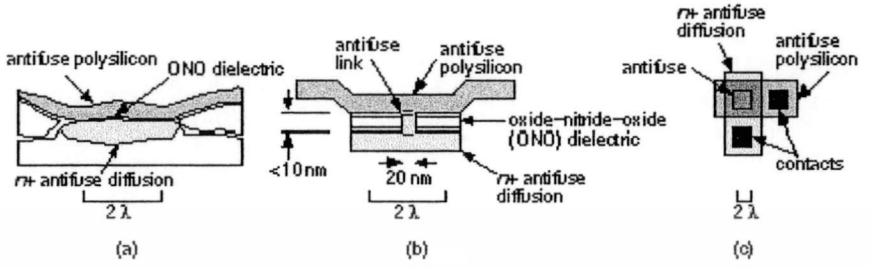
เทคนิคหรือวิธีการกำหนด (Configure) ลอจิกหรือการเชื่อมต่อสัญญาณลอจิก การโปรแกรมแบ่งได้เป็น 4 ชนิด ดังนี้

- Anti-fuse
- SRAM
- EPROM และ EEPROM
-Flash

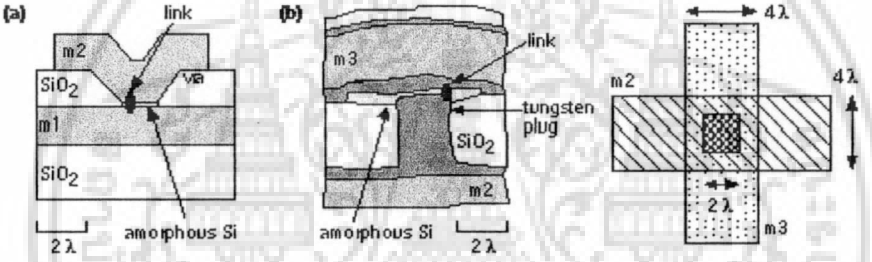
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Anti-fuse

คุณสมบัติของการทำงาน Anti-fuse คือตรงข้ามกับการทำงานของฟิวส์ไฟฟ้าทั่วไปคือ เมื่อสภาวะปกติ Anti-fuse จะเป็นวงจรเปิด Anti-fuse ทำงาน 8 ทำหน้าที่ให้วงจรปิด โดยการป้อนกระแสค่าสูงเข้าไปยังจุดที่เชื่อมต่อนั้นดังรูป



บริษัทที่ใช้เทคนิคการโปรแกรมลักษณะนี้ได้แก่ บริษัท Actel ในตระกูล ACT โดยใช้ Diffusion-Poly Anti-fuse เรียกว่า PLICE และ Quick Logic ใช้ metal-metal anti-fuse เรียกว่า ViaLink



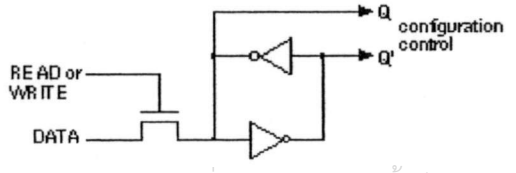
ข้อดี ของการโปรแกรมลักษณะนี้ก็คือ โครงสร้างจะอยู่ถาวร

ข้อเสีย โปรแกรมได้เพียงครั้งเดียว

Static RAM

มีการใช้วิธีนี้ค่อนข้างแพร่หลายใน FPGA ของบริษัท Xilinx สำหรับการกำหนด คอนฟิกูเรชั่นของชิพ เอฟ พี จี เอ จึงคอนฟอกูเรชั่นบิต 0 หรือ 1 จะถูกเก็บได้ใน SRAM จึงจำเป็นต้องมีแหล่งจ่ายให้เลี้ยงตลอดเวลา เมื่อแหล่งจ่ายไฟหมดไปจะทำให้ข้อมูลหายไปด้วย จึงจำเป็นต้องใช้ PROM ในการเก็บข้อมูลบิต

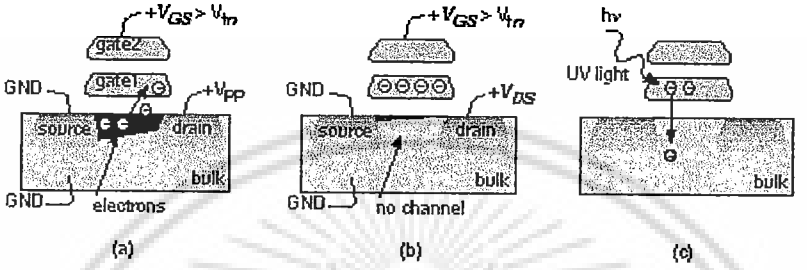
ข้อดี ของ เอฟพีจีเอ ชนิดพีจีเอ สามารถที่จะโปรแกรมซ้ำๆ ได้ แต่ต้องใช้ชิพ (EPROM) เพิ่มเติม เพื่อเก็บข้อมูลบิต ทำให้พื้นที่บอดเพิ่มขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

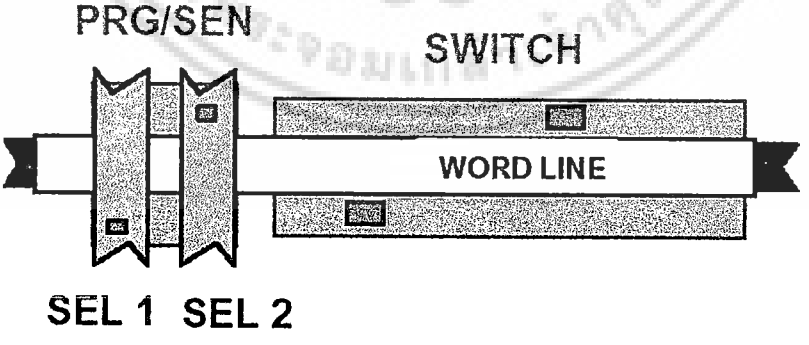
EPROM และ EEPROM

การคอนฟิกลอจิกทำได้โดยใช้เทคโนโลยี่ EPROM หรือ EEPROM ซึ่งใช้ในชิป EPLD ตระกูล Max ของบริษัท Altera และ Xilinx EPLD ตระกูล XC95xx วิธีการโปรแกรมโดยการป้อนศักย์ไฟฟ้าสูงๆ (>12v) ไปยัง+Vpp ตรงบริษัท Drain ทำให้อิเล็กตรอนมีพลังงานมากพอที่กระโดดเข้าไปยัง เกตลอย "gate1" ส่งผลให้เกิดค่า Vth ของทรานซิสเตอร์สูงขึ้นในสภาวะปกติ ทรานซิสเตอร์จะอยู่ในสภาวะปิด (off) (ไม่ทำงาน) เมื่อยิงรังสี uv ไปยังอิเล็กตรอนที่ gate1 แล้วจะกระโดดกลับมาอยู่ในแชลแนล ทำให้ทรานซิสเตอร์ทำงานเป็นปกติ

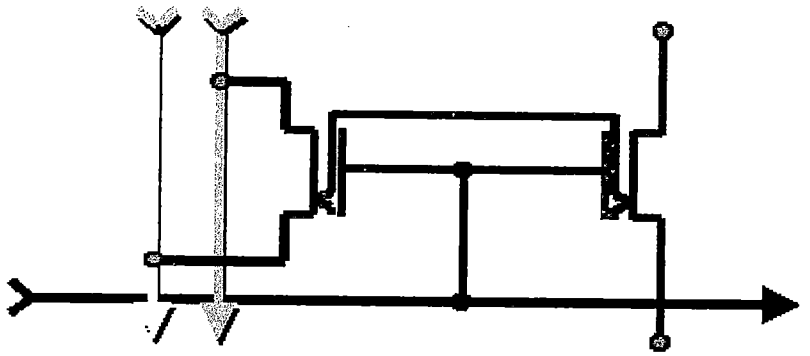


Flash-based FPGA

ลักษณะการโปรแกรม FPGA ชนิดนี้โดยอาศัยคุณสมบัติของหน่วยความจำแบบแฟลช ซึ่งเป็นหน่วยความจำแบบ non-volentine ข้อมูลที่โปรแกรมลงไปในFPGA จะไม่มีการสูญหายไปแม้จะไม่มีการป้อนกระแสไฟ จึงไม่จำเป็นต้องใช้ชิปหน่วยความจำเพื่อเก็บบิตไฟล์ของวงจรอย่าง FPGA ชนิด SRAM โดยทั่วไปขนาดของหน่วยความจำแบบแฟลชจะเล็กกว่าหน่วยความจำแบบ SRAM ทำให้ความจุของ FPGA ชนิดนี้มีความจุเกตสูงกว่า วิธีการโปรแกรมสามารถใช้ได้ทั้งเครื่องโปรแกรมไอซีและการโปรแกรมบนบอร์ดในภาคสนาม (in-system program) ผู้ผลิต FPGA ชนิด flash ที่สำคัญได้แก่บริษัท Actel ในตระกูล ProASIC โดยมีความจุของเกตตั้งแต่ 1 แสนเกต ถึง ล้านเกต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



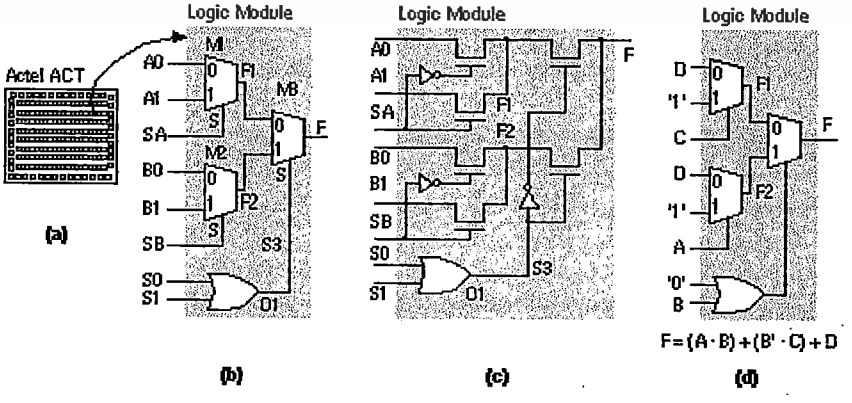
2. Programmable Logic Cell

ดังที่ได้เกริ่นไว้ในบทที่ 1 ในการแบ่งโครงสร้างภายในของ FPGA เป็นแบบ coarsed grain กับ Fine grain ในหัวข้อนี้จะครอบคลุมถึงโครงสร้างภายในของชิปที่มีใช้งานในปัจจุบัน และเทคโนโลยีล่าสุดโครงสร้างสถาปัตยกรรมที่มีการพัฒนาขึ้นประกอบด้วย

- Multiplexer-based cell
- Look-up-table (LUT)
- PAL
- Logic-Tile

Actel ACT

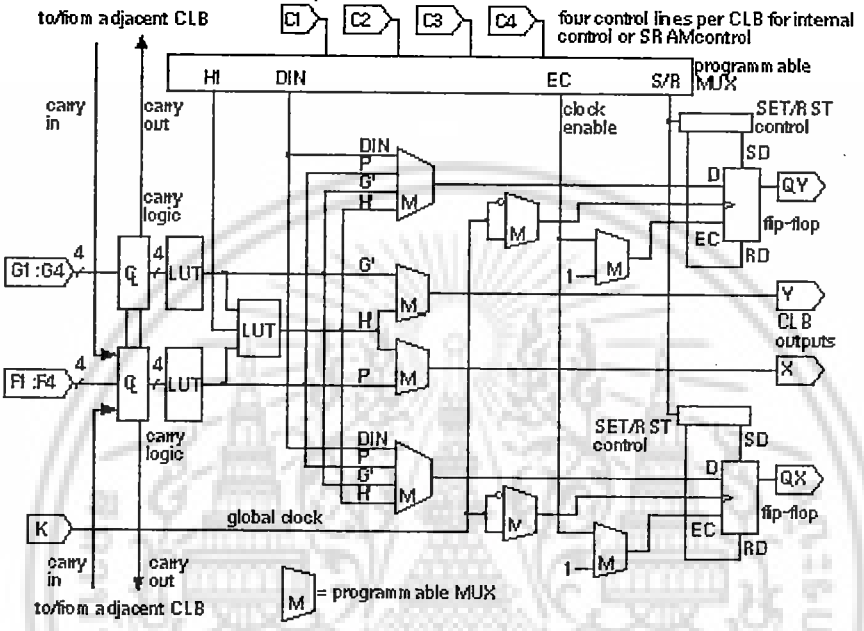
ตระกูล Act1 บริษัท Actel ใช้โครงสร้างสถาปัตยกรรมแบบ *Multiplexure* เรียกว่า Act Logic Module (LM) ซึ่งมีลักษณะดังรูป การสร้างฟังก์ชั่นโดยการกำหนดค่า input ของมัลติเพลกเซอร์ ต่างๆ เช่นในตัวอย่างการหาค่า F=(A.B)+(B.C)+D ในตระกูล Act2 และ Act3 จะเพิ่มลอจิกโมดูลที่ชื่อว่า C-module และ S-module ซึ่ง C-module คล้ายคลึงกับ LM ของ Act1 แต่สามารถใช้กับลอจิกฟังก์ชั่นถึง 5 อินพุท ส่วน S-module ประกอบด้วย C-module และ Sequential element



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Xilinx , CLB (Configurable Logic Block)

โครงสร้างของ FPGAของ Xilinx จะเป็นรูปแมตริกร์ ประกอบด้วย CLB และ Switching Matrix และมี I/O ล้อมรอบ ส่วน Logic block ของ Xilinx จะถูกเรียกว่า LCA (Logic Cell array) ซึ่งการกำหนดฟังก์ชั่นของลอจิกจะใช้ LUT ภายใน CLB ประกอบด้วยทั้ง combinational logic และ flip-flop ในตัวอย่างนี้จะแสดง LCA ของXilinx ตระกูล XC4000 อินพุท C1-C4 ยังใช้กำหนดการใช้งานของ F' และ G' LUT ให้เป็น SRAM ขนาด 32 บิท

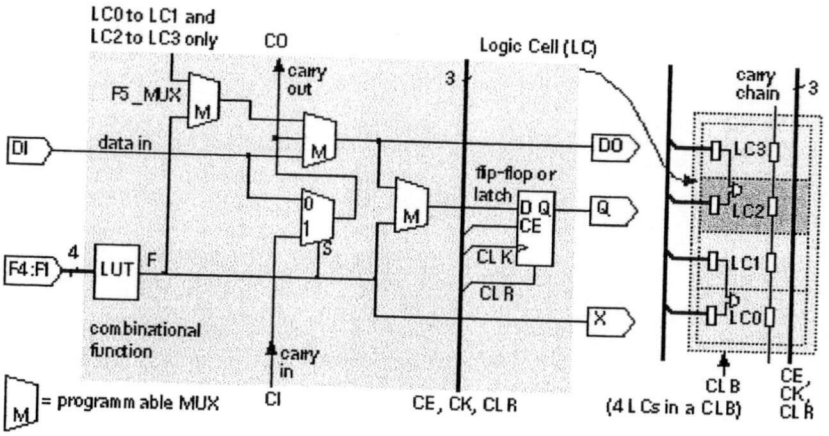


ในรูปแสดง CLB ของXilinx ตระกูล XC4000 ซึ่งประกอบLUT แบบ 4 อินพุท สำหรับแม็พค่าอินพุท จำนวน 2 ตัว มีMUX สำหรับคอนโทรลลอจิก C1-C4 เข้ากับ 4 อินพุทดังนี้ H1 ของ LUT DIN, CE และ S/R สำหรับ Flip-flop การใช้ LUT ใน CLB ของ Xilinx เพื่อใช้สร้างลอจิกมีทั้งข้อดีและข้อเสียด้วยกล่าวคือ การทำinverter อาจต้องใช้NAND ขนาด 5อินพุท แต่ LUT ยังช่วยให้การกำหนด tining ของ synchronous logic ได้สะดวกและเหมาะกับเทคโนโลยีการโปรแกรมแบบ SRAM นอกจากนี้ยังสามารถใช้ LUT มาสร้างเป็น SRAM 16X1 หรือ 32X1 ได้การตั้งชื่อ part ของXilinx จะมีค่าความหน่วงของลอจิกใน CLBเป็นเกณฑ์ เช่น XC4010-3 มีค่า t_{ILO}=3 ns เป็นต้น

XC5200 Logic CLB

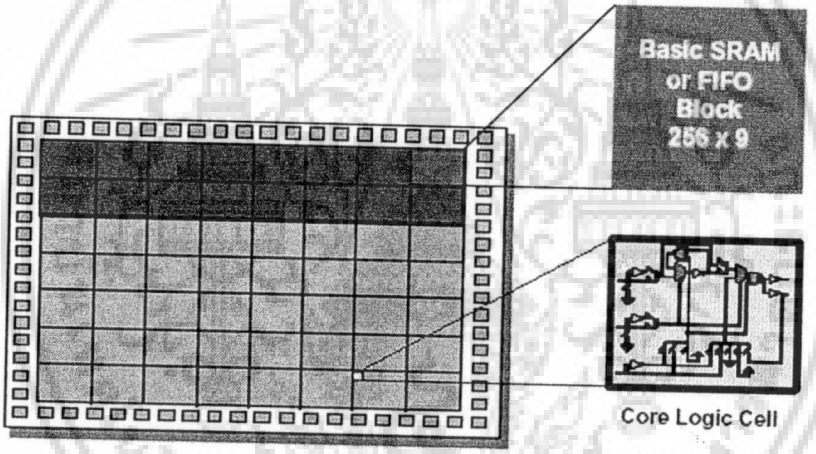
โครงสร้างพื้นฐานของ Xilinx ตระกูล XC5200 ซึ่งมีลักษณะใกล้เคียงกับตระกูล XC3000/4000 แต่มีง่ายกว่า ซึ่งเรียกว่า Logic Cell (LC) Xilinx ยังคงเรียก CLB ใน XC5200 นั้นหมายถึงกลุ่มของ LC จำนวน 4 ตัว (LC0-LC3) XC5200 LC ประกอบ LUT ขนาด 4 อินพุท,FT, และ Mux สำหรับสลับสายสัญญาณวงจร carry ถูกแยกออกมาจาก LUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

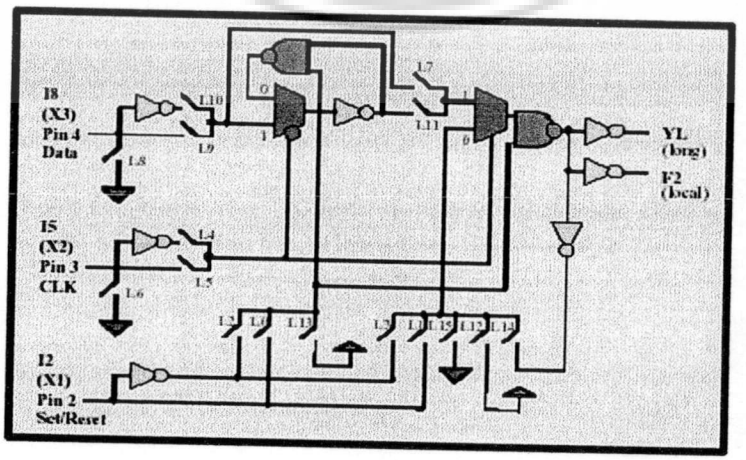


Actel ProASIC Flash based FPGA

โครงสร้างสถาปัตยกรรมของเอฟพีจีเอ Actel ProASIC ประกอบด้วยแผงอาเรย์ของลอจิกเซล และมีไอโอล้อมรอบ ภายในเซลลอจิกยังมีหน่วยความจำแบบSRAMหรือ FIFO ขนาด 256x9 บิต



ส่วนที่ทำหน้าที่ฟังก์ชั่นลางลอจิกเรียกว่า *Logic Tile* ซึ่งสามารถนำไปสร้างเป็นฟลิบฟล็อบหรือฟังก์ชันทางลอจิกแบบ 3 อินพุทได้



โครงสร้างของลอจิกของชิป ProAsic นั้นเรียกว่า Logic-Tile

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. I/Os ใน FPGA

ส่วน I/O ใน FPGA

ทำหน้าในการส่งและรับสัญญาณระหว่าง ลอจิกภายในกับโลกภายนอกส่วนของ I/O จึงต้องมีคุณสมบัติต่าง ๆ ซึ่งประกอบด้วยDc output สำหรับขับโหลดต้านทานที่กระแสที่ DC หรือความถี่ต่ำๆ (โดยทั่วไปน้อยกว่า1MHz) ซึ่ง I/O จะสามารถให้ศักย์ไฟฟ้าและกระแสเพียงพอที่จะขับโหลดเหล่านั้น เช่นโหลดที่เป็น LED รีเลย์หรือมอเตอร์ AC output สำหรับการขับโหลดที่เป็นตัวเก็บประจุไฟฟ้าด้วยสัญญาณความถี่สูง (โดยทั่วไปมากกว่า1MHz) เช่นโหลดที่เป็นไอซีข้างเดี่ยว บัสสัญญาณ โดยที่ I/O นี้สามารถที่จะขับสัญญาณได้เร็วเพียงพอ DC- input ตัวอย่างของแหล่งสัญญาณได้แก่สวิตซ์ สัญญาณจากเซ็นเซอร์ หรือ สัญญาณจากไอซีอื่น ๆ ซึ่งตัวI/O นี้สามารถแปลค่าลอจิกของอินพุทได้ถูกต้องหรือไม่ Clock input ได้แก่สัญญาณนาฬิกาของระบบ ซึ่ง clock input นี้สามารถส่งสัญญาณนาฬิกาได้รวดเร็วและถูกต้องหรือไม่ Power input ซึ่งให้ตัวจ่ายพลังงานให้กับส่วนของ I/O และลอจิกภายในชิปโดยที่ไฟเลี้ยงไม่ตกหรือเกิดสัญญาณรบกวนได้

จากที่ได้กล่าวมาแล้วเป็นส่วนที่จะต้องคำนึงถึงของ I/O ของชิป FPGA ในผู้ผลิตแต่ละรายจะมีการออกแบบ I/O ที่แตกต่างกันซึ่งมักจะมีข้อกำหนดเฉพาะเป็นของตนเองดังจะกล่าวต่อไป

Xilinx I/O Block

เซล I/O ของXilinx เรียกว่า I/O Block (IOB) สามารถแสดงได้ดังรูป ซึ่งเป็นตัวอย่าง IOB ของตระกูล XC4000 ในส่วนที่เป็น output

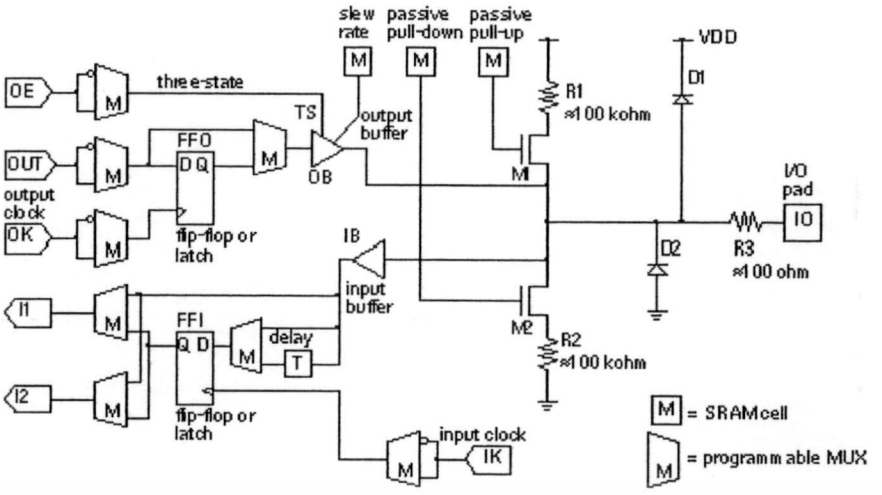
ประกอบด้วย

1. ตัวต้านทานสำหรับ pull-down, pall-up
2. ตัวคอนโทรล three state , OE
3. FF หรือ Latch สำหรับต่อตรงออกเอาท์พุท
4. สัญญาณคอนโทรล slew-rate ของเอาท์พุท

ในส่วนของอินพุทประกอบด้วย

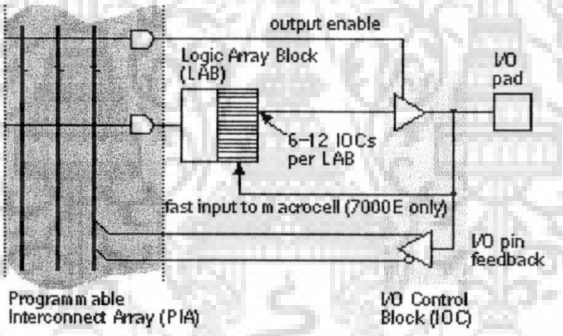
1. ส่วนคอนฟิก ระบบของอัพเพอร์อินพุทให้เป็น TTL หรือ CMOS
2. FF หรือ Latch สำหรับต่ออินพุทเข้าโดยตรง
3. สวิตซ์ในตัวหน่วงสำหรับกำจัดค่า hold time ของอินพุท

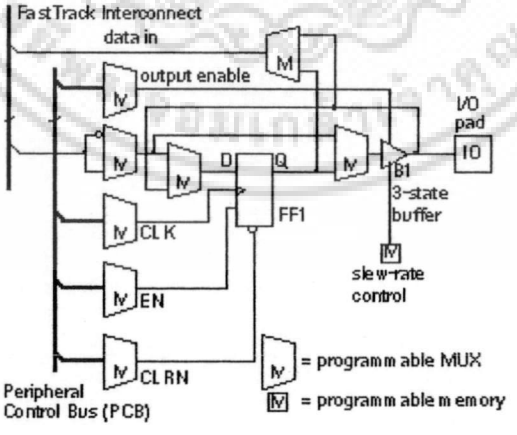
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Altera FLEX

ในส่วนของ Alter FLEX จะใช้ I/O Element (IOE) ดังรูป IOE จะต่อเข้ากับ interconnect โดยตรง และจะมี PCB (Peripheral Control Bus) สำหรับเป็นตัวสัญญาณคอนโทรล เช่น clock, reset, preset, output enable





<u>สรุป</u> จากการสังเกตจะพบว่าใน I/O cell จะมีทางเลือกต่าง ๆ ประกอบ พลังขับต่างๆ กับTTL/Cmos compatibility การต่อตรงหรือผ่านรีจีสเตอร์ มีตัวต้านทานสำหรับ pull-up/pull-down, มี Boundary Scan ในชิปเอฟพีจีเอ ใหม่ๆ จะมีให้เลือกเช่น สนับสนุน PCI หรือระดับโวลเตจแบบ low differential (LVDS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ทรัพยากรอื่นๆ</u> ในเอฟพีจีเอ

- ตัวกำหนดสัญญาณนาฬิกา เช่นDigital phone lock loop (DPLL) ใน Xilinx หรือ DLL (Delay lock loop) ใน Altera

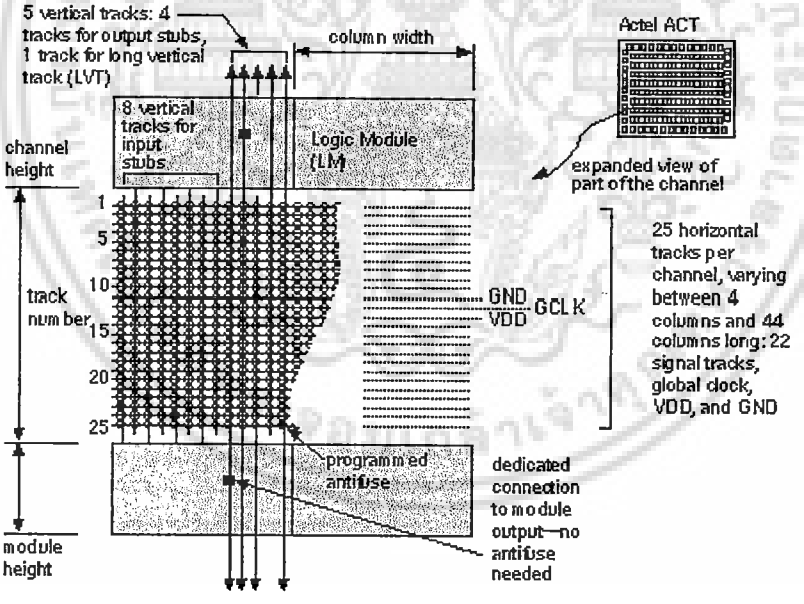
- Embedded core ในเอฟจีเอ บางรุ่นจะมีการผังวงจรหรือชิบ (Embedded) ตัวอื่นอยู่ด้วย เช่น μr หรือมีแรมบล็อกขนาดใหญ่ให้ใช้งานได้เลย

4. Interconnect

FPGA ที่มีอยู่ทั่วไปจะประกอบด้วย programmable interconnect แบบใดแบบหนึ่งซึ่งโครงสร้างและความซับซ้อนจะเป็นตัวบ่งบอกถึงเทคโนโลยีการโปรแกรมและสถาปัตยกรรมของเซลล์ลอจิก วัสดุที่ใช้สำหรับการเชื่อมต่อมักจะใช้อลูมิเนียม ซึ่งมีค่า sheet resistance ประมาณ 50 mW /

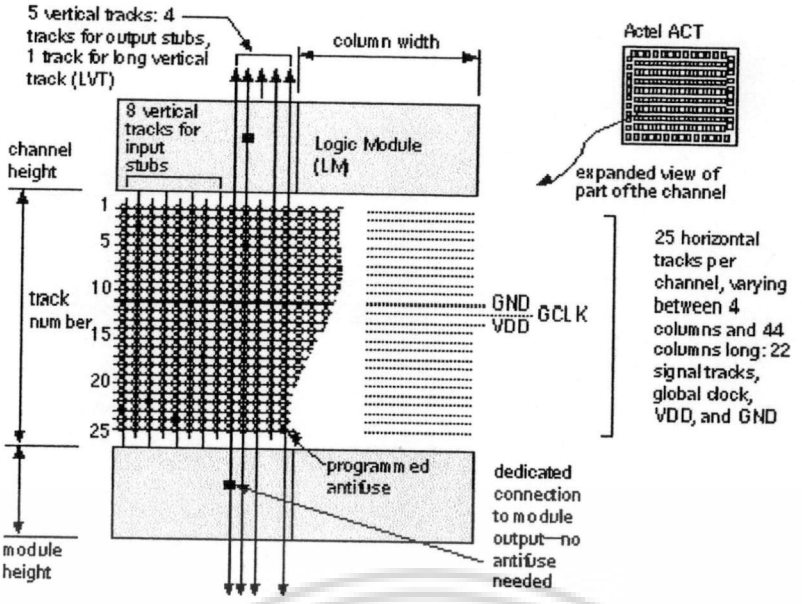
Actel Act

การเชื่อมต่อภายในของเอฟพีจีเอของ ACTEL ตระกูล ACTจะคล้ายกับ gate array ซึ่งมีลักษณะเป็นแถว (channel) ดังรูปที่....



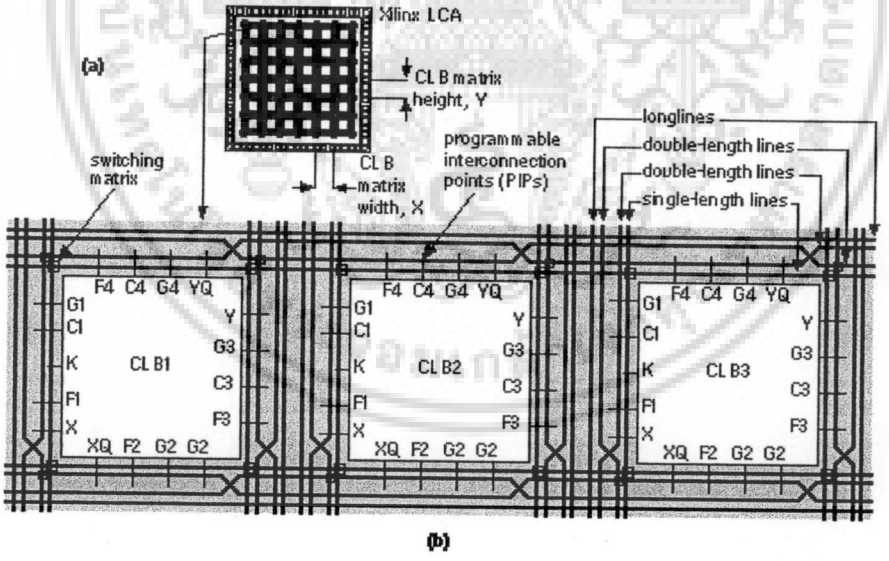
จึงประกอบด้วยส่วนสีเหลี่ยมขนาดเล็กเรียกว่า wiring channel แชนแนลแนวนอนสายสัญญาณจะวิ่งไปตามแนวนอน ส่วนในลอจิกโมดูลแนวตั้งสัญญาณจะวิ่งในแนวตั้งเหนือเซลลอจิก หรือภายในแต่ละแชนแนลจะประกอบด้วยแทร็กต่างๆ ที่ให้เส้นลวดวิ่งผ่านได้ 1 track = 1 wire การเชื่อมต่อระหว่างลอจิกโมดูลกับแชนแนลจะใช้หัวต่อเรียกว่า Stub ทรัพยากรสำหรับใช้ Routing ใน Actel Act1 ประกอบด้วยแชนแนลในแนวนอนแต่ละแชนแนลขนาด 25 แทร็ก และแชนแนลในแนวตั้งขนาด 8 แทร็กด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



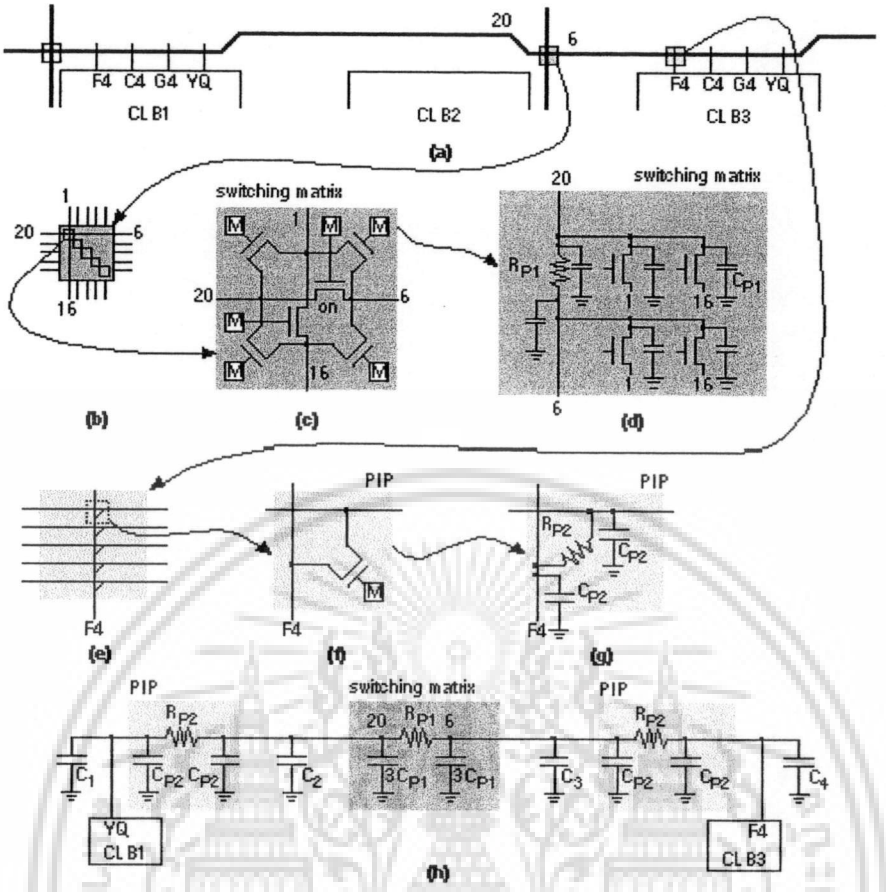
Xilinx LCA

โครงสร้าง interconnect ชิง Xilinx LCA ประกอบด้วยเส้นในแนวนอนกับแนวตั้งวิ่งระฟว่าง CLB เมตริกซ์สลับสาย(switching metric) สายสัญญาณยาววิ่งทั่วชิป ซึ่งสามารถใช้เป็นบัสภายในได้ ตัวเชื่อมต่อโดยตรงระหว่าง CLB โดยไม่ต้องผ่านเมตริกซีสลับสาย Programmable interconnect Point (PIP) ซึ่งเป็ฯพาสทรานซีสเตอร์ใช้สำหรับเชื่อมต่ออินพุทของCLBกับสายสัญญาณภายนอก



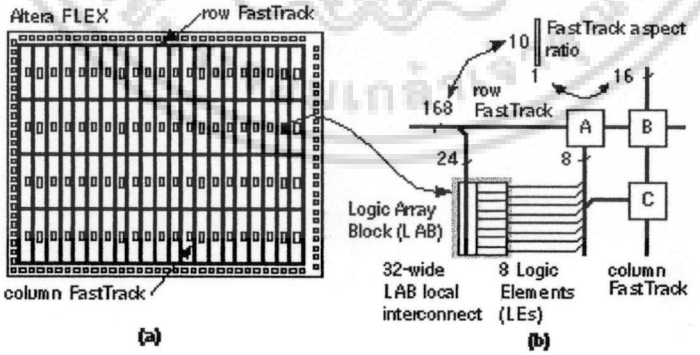
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงวิธีการเชื่อมต่อสัญญาณภายใน XCA in Xilinx โดยใช้ Switching matrix และ PIP



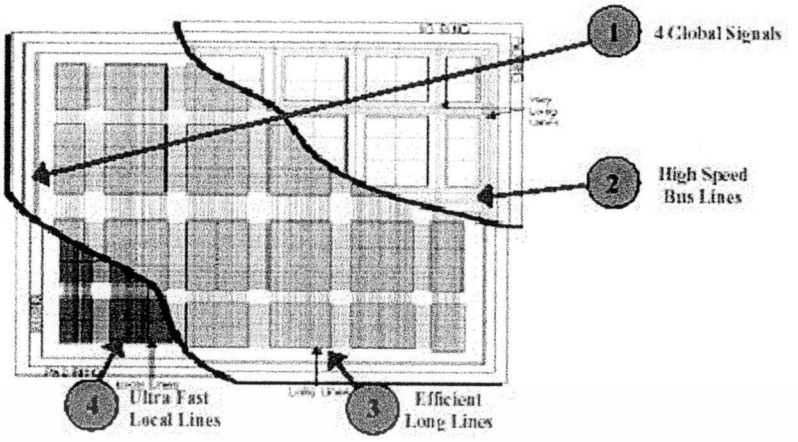
Altera Flex

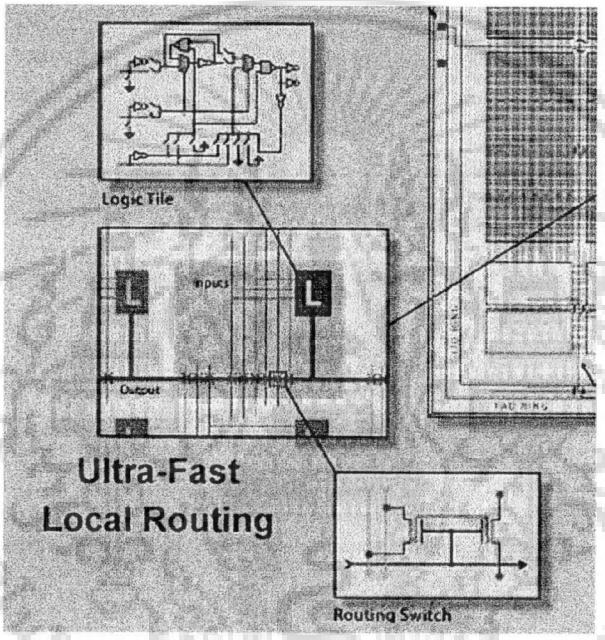
Altera Flex ใช้โครงสร้างที่เรียกว่า Fast Track โดยทีประกอบ Fast Track ขนาด 168 แชนแนลต่อแถวและ 16 แชนแนลต่อคอลัมน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Actel ProAsic





Summary: Programmable Interconnect

	Actel (ACT 1)	Altera MAX (MAX 7000)	Altera FLEX 10k	Xilinx (XC4000)	Xilinx LCA (XC5200)
Interconnect between logic cells (tracks = trks)	Channeled array with segmented routing, long lines: 25 trks/ch. (horiz.); 13 trks/ch. (vert.); < 4 antifuses/path	Fixed cross-bar PIA (Programmable Interconnect Architecture)	Row and column FastTrack between LABs	Switch box, PIPs (Programmable Interconnect Points), 3-state internal bus, and long lines	Switch box, PIPs (Programmable Interconnect Points), 3-state internal bus, and long lines
Interconnect delay	Variable	Fixed	Fixed with small variation in delay in row FastTrack	Variable	Variable
Interconnect inside logic cells	Poly–diffusion antifuse	EEPROM	LAB local interconnect between LEs. 16-bit SRAM LUT in LE.	32-bit SRAM LUT	16-bit SRAM LUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การเลือกใช้งาน FPGA

การเลือกใช้ FPGA องค์ประกอบที่จะต้องพิจารณา

1. System gates
2. Typical gates
3. Max I/Os
4. Max Flip-flop
5. Logic module
6. Speed
7. Packaging
8. Sourcing
9. Development tool
10. Power consumption
11. Cost

6. Design flow Basic & Design tools FPGA Synthesis

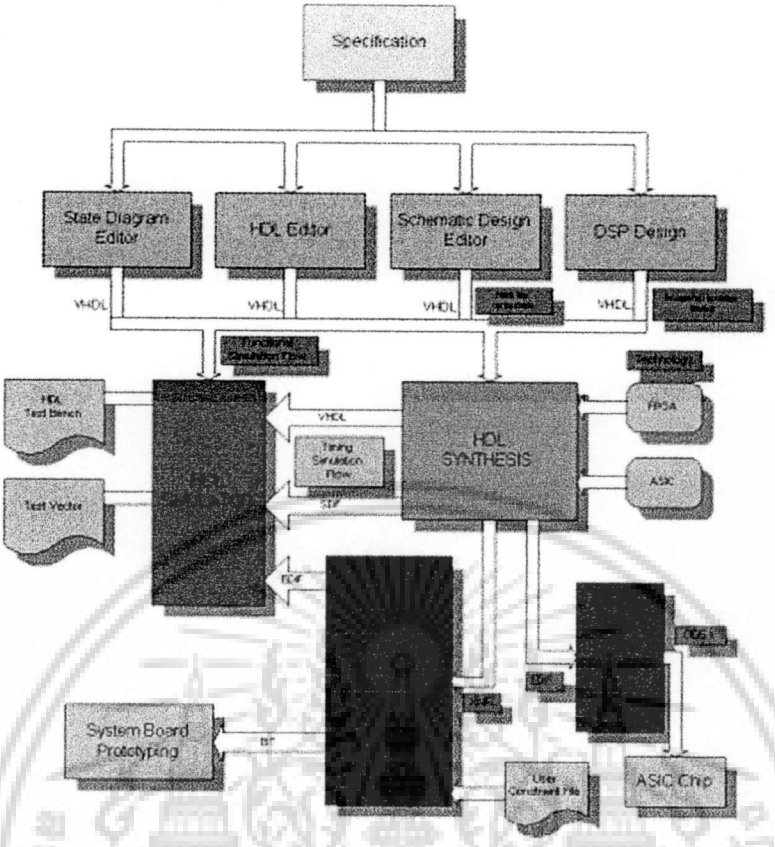
เนื่องจากวิวัฒนาการของเทคโนโลยีมีการเปลี่ยนแปลงที่เร็วมาก จากแรงกดดันจากการแข่งขันทางธุรกิจ ผู้ผลิตจำเป็นต้องออกผลิตภัณฑ์ใหม่ตลอดเวลา ชื่อบริษัท ชื่อและเวอร์ชั่นของซอฟต์แวร์อาจจะมีการเปลี่ยนแปลงดังเหตุผลข้างต้น ผู้เรียนสามารถดูข้อมูลล่าสุดจากหัวข้อ Resources & References

องค์ประกอบที่สำคัญส่วนหนึ่งของการออกแบบวงจรรวมทั้งการออกแบบ ASIC และการออกแบบ FPGA คือซอฟต์แวร์ช่วยในการออกแบบวงจรรวมหรือที่เรียกทั่วไปว่า EDA (Electronic Design Automation Tools) และนักออกแบบจำเป็นต้องเข้าใจขั้นตอนการออกแบบด้วย (Design Flow)

การออกวงจรรวมโดยใช้ FPGA มักจะใช้วิธีการออกแบบจากบนลงล่างหรือที่เรียกว่า Top-down design methodology (TDM) คือการออกแบบเริ่มต้นจากการมองที่ระบบใหญ่ก่อนแล้วทำการแบ่งย่อย (patition)ไปเป็นระบบย่อย(subsystem) การออกแบบวิธี TDM นี้มักจะใช้วิธีการหรือซอฟต์แวร์แบบ HDL-based (Hardware Description Language) ซึ่งเริ่มต้นออกแบบโดยใช้ภาษาชั้นสูงเพื่อบรรยายพฤติกรรมของฮาร์ดแวร์ ภาษาที่นิยมใช้ได้แก่ VHDL และ Verilog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบโดยทั่วไปสามารถแสดงได้ดังรูป



ขั้นตอนต่างๆ สามารถสรุปได้ดังนี้

1. Design Specification
2. Design Entry
3. RTL (HDL) Simulation
4. Logic Synthesis
5. Gate Simulation
6. Place & Route
7. Post Layout Simulation
8. FPGA Programming (Implementation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์สำหรับออกแบบ FPGA

1. Bundle Package

- Integrated design flow
- Schednatic design entry
- Simulation (Logic, Formational)
- Synthesis
- FPGA Implementation

1 Altera Max + Plus II

2 Xilinx & Foundation ISE

3 ALDEC ActiveHDL

2. Individual Package

1 Design Entry

- Renior/HDL Designer

2 Simulation

- Model Simulation

3 Synthesis

- Synplify
- Leonardo
- FPGA Express

4 FPGA Implementation

- Xilinx Foundation ISE, Alliance
- Altera Quartus II
- Actel Design System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Chusit Pradabpet, Narong Ravinu, Sorawat Chivapreecha, B. Knobnob and Kobchai Dejhan, **"An Efficient Filter Structure for Multiplierless Sobel Edge Detection"** 2009 Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), Monash University Sunway Campus, Malaysia, July.25-26, 2009.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

An Efficient Filter Structure for Multiplierless Sobel Edge Detection

C. Pradabpet, N. Ravinu, S. Chivapreecha, B. Knobnob and K. Dejhan, *Senior Member, IEEE*,

***Abstract*—This paper presents a multiplierless Sobel edge detection filter structure based on the gradient of Gaussian lowpass filter concept. This concept is same as the idea that used for evaluation of Sobel edge detection kernels. The proposed Sobel edge detection filter structure consists of 4 sets of the so-called lowpass-type 2nd order 1-D DPF (Discrete Pascal Filter) and the filter structure does not need any multipliers, only 14 adders are needed for realization. Consequently, the high efficiency multiplierless Sobel edge detector can be achieved.**

I. Introduction

At present, the real-time image processing using 2-D digital filter has become rapidly growing in many applications such as geophysical and biomedical image processing. However, the design of 2-D digital filter needs many multiplications for one output sample because there are many non-zero elements in the filter coefficient. The number of multipliers will decrease the speed of processing time since the multiplying stage has a lot of consumption process and uses large area in design. Therefore, in [1-4] propose a design of distributed arithmetic (DA) structure realization for 2-D digital filter to give the multiplierless 2-D digital filter.

The Sobel edge detection is also the widely used for edge detection in digital image processing applications. The original idea of Sobel edge detection is based on the gradient (1st order derivative) of Gaussian lowpass filter, and can evaluate to Sobel edge detection convolution kernels size 3x3 [5]. The Sobel edge detection kernels are divided to vertical gradient and horizontal gradient convolution kernel. Since each kernel has size 3x3, then if the design process starts from these kernels and use 2-D convolution equations for direct realization, 9 multipliers and 8 adders are needed for each kernel. Therefore, 2-D digital filter need up to 18 multipliers and 16 adders for direct realization of Sobel edge detection filter, from 2 difference equations (not included the gradient combining).

The popular method that can use to solve the problem of many multipliers in difference equation realization is DA [1-4]. The main part of DA implementation is ROMs (or any kind of memory) and scaling accumulator, so in the past DA is called ROM-Accumulator structure. Although, the DA implementation does not need multipliers but DA requires the memory such as ROM to store the pre-computed values of partial products [2-4]. For Sobel edge detection realization needs 2 ROMs for implementation, 1 ROM for vertical gradient pre-computed values, and 1 ROM for horizontal gradient pre-computed values. Each ROM needs 512 addresses to store all possible values of partial product, the ROMs addresses are the input image and its delayed index. Output of each ROMs must be computed the vertical gradient output and horizontal gradient output by using the scaling-accumulator, which need 2 units of scaling-accumulator for 2 ROMs.

But, the new Sobel edge detection filter structure which needs only 14 additions for computation (without any multipliers same as DA-based Sobel edge detection) will be proposed for hardware realization and comparison the complexity with the DA-based. The 4 units of 2nd order 1-D DPF are used for the proposed hardware of Sobel edge detection filter structure realization. Each 2nd order DPF is a network of 3 adders and another 2 adders are used for summing the pair of 2nd DPF to produce each gradient output. Consequently, the high efficiency adder network for Sobel edge detection can be obtained.

C. Pradabpet is with the Faculty of Science and Technology, Phranakhon Si Ayutthaya Rajabhat University, 13000, Thailand. (Corresponding author to E-mail: pchoos@ aru.ac.th; chusit@ icn.ist.hokudai.ac.jp).

N. Ravinu, S. Chivapreecha, B. Knobnob and K. Dejhan are with the Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), 10520, Thailand.

II. Principle of Sobel Edge Detection

A. Sobel Edge Detection Concept

The first order derivative or gradient operator of image $f(x,y)$ is defined by

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

Important quantity that used for edge detection is the magnitude of gradient, which is

$$mag(\nabla f) = |\nabla f| = \left[G_x^2 + G_y^2\right]^{1/2} \quad (2)$$

I In order to evaluate the convolution kernels for Sobel edge detection, assume input image size 3×3 as shown in this matrix.

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

where $z_5 = f(x,y)$

The concept that used for establish Sobel edge detection is based on 1st order derivative of the lowpass filtered image by Gaussian lowpass filter [5]. Therefore, the 1st order derivative at point z_5 is

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (3a)$$

and

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (3b)$$

Results from eqs. (3a) and (3b) can give the Sobel edge detection kernels as

-1	-2	-1
0	0	0
1	2	1

Horizontal Gradient Kernel

-1	0	1
-2	0	2
-1	0	1

Vertical Gradient Kernel

In practical, to avoid the complexity of squaring and square root as in eq. (2), the magnitude of gradient in eq. (2) can be approximated by $|\nabla f| \approx |G_x| + |G_y|$ this step is called "Gradient Combining". Finally, we can summarize the Sobel edge detection as shown in Fig. 1.

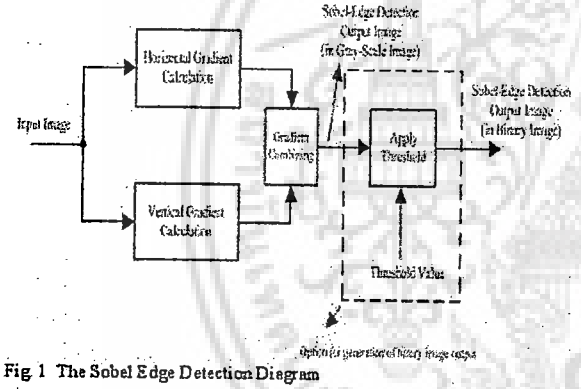


Fig 1 The Sobel Edge Detection Diagram

For the causal system, 2-D convolution equation for output image computation of the convolution kernel size 3×3 is

$$y(m,n) = \sum_{k=0}^{2}\sum_{l=0}^{2} h(k,l)\, x(m-k, n-l) \quad (4)$$

where $x(m,n)$ and $y(m,n)$ is input and output image, respectively and $h(k,l)$ are the coefficients inside the kernel.

The Sobel edge detection has 2 convolution kernels, so 2 convolution equations are needed for vertical and horizontal gradient computation as follows,

$$y_{hor}(m,n) = \sum_{k=0}^{2}\sum_{l=0}^{2} h(k,l)\, x(m-k, n-l) \quad (5a)$$

and

$$y_{ver}(m,n) = \sum_{k=0}^{2}\sum_{l=0}^{2} v(k,l)\, x(m-k, n-l) \quad (5b)$$

If consider directly from eqs. (5a) and (5b), 18 multipliers and 16 adders are needed for Sobel edge detection filter realization.

III. Multiplierless Sobel Edge Detection Filter by Distributed Arithmetic

Realization for 2-D distributed arithmetic digital filter can be explained as follows. Consider, the 2-D digital filter is described by linear difference equation as in eq. (4) again.

$$y(m,n) = \sum_{k=0}^{2}\sum_{l=0}^{2} x(m-k, n-l)\, h(k,l) \quad (6)$$

where $x(m,n)$ and $y(m,n)$ are the input and output image, respectively when $h(k,l)$ are filter coefficients kernel size 3x3. Assume, all signals are bounded by ± 1 and define the input signal in two's complement format, B bits accuracy including sign bit by

$$x(m,n) = \sum_{j=1}^{B-1} x_j(m,n)\, 2^{-j} - x_0(m,n) \quad (7)$$

where $x_j(m,n)$ are $x(m,n)$ in bit level. Thus, eq. (6) can be rewritten as

$$y(m,n) = \sum_{k=0}^{2}\sum_{l=0}^{2} h(k,l)\left(\sum_{j=1}^{B-1} x_j(m-k, n-l)\, 2^{-j} - x_0(m-k, n-l)\right) \quad (8)$$

Rearrange the summations in eq. (8) yields

$$y(n,n) = \sum_{j=1}^{B-1}\left[\sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)\, x_j(m-k, n-l)\, 2^{-j}\right)\right] - \sum_{k=0}^{2}\sum_{l=0}^{2}\left(h(k,l)\, x_0(m-k, n-l)\right) \quad (9)$$

Define the function $F(\cdot)$ as follow,

$$F_j\left(x_j(m,n), x_j(m,n-1), \cdots, x_j(m-2, n-2)\right) = h(0,0)\, x_j(m,n) + h(0,1)\, x_j(m, n-1) + \cdots + h(2,2)\, x_j(m-2, n-2) \quad (10)$$

Since each $x_j(m-k,n-l)$, where $j = 0,1,\ldots, B-1$ is only 0 or 1, we also pre-compute the values of this function that called partial products stored in memory and using the input signals for addressing. Therefore, the output can be computed by shifting and adding of partial products as in eq. (11).

$$y(m,n) = \sum_{j=1}^{B-1} F_j(\cdot)\, 2^{-j} - F_0(\cdot) \quad (11)$$

This equation can be realized to be hardware by using the distributed arithmetic and can be shown that the multiplier has not used in this structure.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Then, from eqs. (5a) and (5b) the hardware architecture of distributed arithmetic Sobel edge detection filter can be shown in Fig.2, this hardware needs the control signals to control each circuit components inside and the input image data will be distributed into bit level before used to be the memory addresses. The further more details for Sobel edge detection distributed arithmetic filter can see in [4].

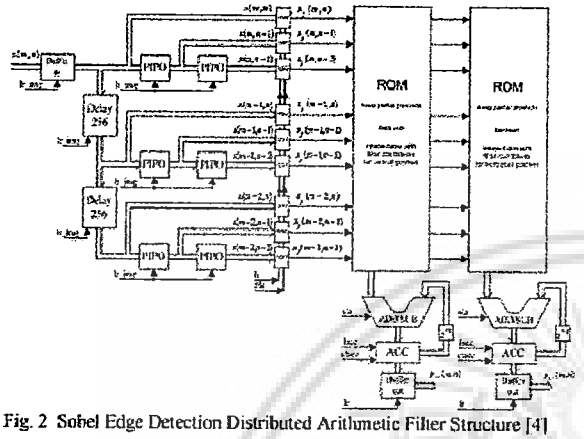


Fig. 2 Sobel Edge Detection Distributed Arithmetic Filter Structure [4]

IV. The Proposed Multiplierless Sobel Edge Detection Filter Based on DPF

The proposed Sobel edge detection filter structure which is based on the so-called lowpass-type 2nd order 1-D DPF (Discrete Pascal Filter) where the discrete Pascal filter is the filtering characteristics hidden in the discrete Pascal transform (DPT). There are two types of the hidden filtering characteristics; highpass and lowpass, depends on the type of Pascal transformation matrix. The Pascal transformation matrix can be factorized into binary matrices, result of the Pascal matrix factorization achieves the efficient hardware realization for 1-D DPT using the so-called DPT butterfly unit. This structure can operate without multiplication and only addition is used, which can reduce the computational complexity. The DPF structure is modified from DPT structure, by placing the delay elements to establish relationship between each input [6-7]. The discrete Pascal transform **X** of the one-dimensional signal vector **x** is defined as

$$\mathbf{X} = \mathbf{P}\mathbf{x} \tag{12}$$

where **X** is the transformed output vector, **x** is input vector and **P** is Pascal transformation matrix.

For example, 3-point lowpass type DPT, 3×3 lowpass type Pascal transformation matrix is used as follows,

$$\mathbf{P}_3^{(LP)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{13}$$

In [7] shows the efficiency structure realization for 3-point lowpass type DPT which is shown in Fig. 3. This structure is obtained from Pascal transformation matrix factorization into binary matrices (the details can see in [7]).

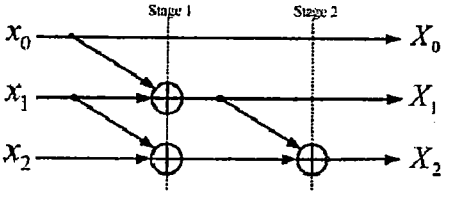


Fig. 3 The 3-point Lowpass Type DPT Flow Graph

For the 2nd order lowpass type 1-D DPF, which the characteristic is lowpass filtering can be investigated in [6] and can be summarized to the 2nd order transfer function in eq. (14).

$$H^{(LP)}(z) = 1 + 2z^{-1} + z^{-2} \tag{14}$$

Therefore, the DPF 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, the filter structure realization of 1-D DPF of lowpass type can be realized as follows,

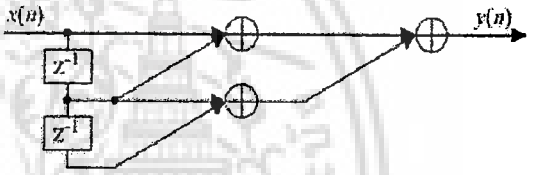


Fig. 4 The 2nd order Lowpass Type 1-D DPF

Our proposed filter structure realization is start from the definition of gradient operator, which will operate to the blurred input image from Gaussian lowpass filtering as shown in eqs. (3a) and (3b). From that point, we can use the idea of the lowpass-type 1-D DPF which is equivalent to Gaussian lowpass filtering establish the new multiplierless Sobel edge detection filter.

Since, the behavior of DPF like Gaussian filtering (sometimes, called binomial filter which is the approximation version of Gaussian filter) but can be realized to the beautiful structure without any multiplications. Therefore, our proposed new Sobel edge detection filter structure will be multiplierless and only 14 additions are needed to compute the vertical gradient output and horizontal gradient output as shown in Fig. 5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

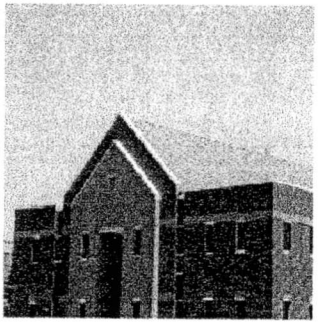


Fig. 7 Original input image

Fig. 8 and Fig. 9 show the horizontal gradient and vertical gradient output, respectively. The Sobel edge detection output after gradient combining step is shown in Fig. 10.

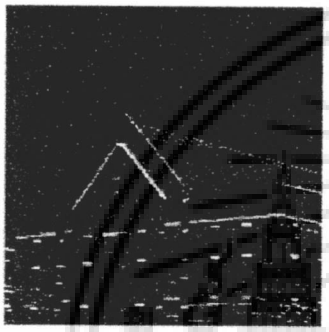


Fig. 8 Horizontal gradient output

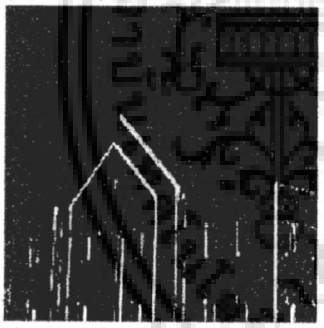


Fig. 9 Vertical gradient output

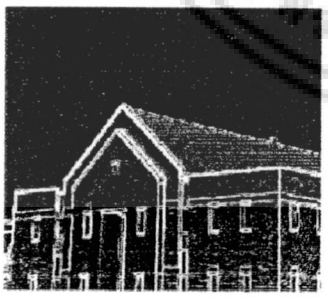


Fig. 10 Sobel edge detection output image

From the input image in Fig. 7, Fig. 8 and Fig. 9 show the horizontal gradient and vertical gradient output, respectively. The Sobel edge detection output after gradient combining is shown in Fig. 10. Altogether, direct realization, DA-based and the proposed Sobel edge detection filter structure will give the same edge detected output, the difference is concerned only about how to simplify the hardware structure realization, which the proposed Sobel-edge detection filter has lowest complexity,

CONCLUSIONS

The multiplierless Sobel edge detection filter is proposed, the method to realize this structure is based on finding the gradient of blurred image by Gaussian lowpass filter. The lowpass type 1-D DPF can be used for realization of our proposed Sobel edge detector with the simple hardware structure and very low computational burden. The obtained hardware structure is suitable for implementing on an ASIC (Application Specific IC) to obtain a single chip for Sobel-edge dectection with a high processing speed and can reduce power consumption (since using less hardware components).

REFERENCES

[1] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," IEEE Trans. on ASSP., Vol. ASSP-22, No. 6, pp. 456-462, December 1974.

[2] J. Jaggernauth, A. C. P. Loui and A. N. Venetsanopoulos, "Real-Time Image Processing by Distributed Arithmetic Implementation of Two-Dimensional Digital Filters," IEEE Trans. on ASSP., Vol. ASSP-33, No. 6, pp. 1546-1555, December 1985.

[3] S. Thongplew and K. Dejhan, "Implementation of 2-D Digital Filter-based Distributed Arithmetics," Ladkrabang Engineering Journal., Vol. 11, No. 1, pp. 1-6, June 1994.

[4] S. Chivapreecha, K. Dejhan and C. Pienvijarnpong, "Hardware Implementation of Sobel-Edge Detection Distributed Arithmetic Digital Filter," Proc. the 25th Asian Conference on Remote Sensing (ACRS'2004), Chiang-Mai, Thailand, pp.284-289, November 22-26, 2004.

[5] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice-Hall, 2002.

[6] S. Chivapreecha, U. Nithirochananont, and K. Dejhan, "Investigation of Frequency Characteristic in Discrete Pascal Transform and Its Applications," Proc. The 4th International Colloquium on Signal Processing and Its Application (CSPA 2008), Kuala Lumpur, Malaysia, March 7-9, 2008.

[7] N. Ronnarongrit, S. Chivapreecha and K. Dejhan, "Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization," Proc. The 4th International Colloquium on Signal Processing and Its Application (CSPA 2008), Kuala Lumpur, Malaysia, March 7-9, 2008.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้