สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

8

ระบบประมวลผลภาพเชิงเลขแบบเวลาจริงโดยใช้ FPGA และแสดงผลบน VGA

REAL-TIME DIGITAL IMAGE PROCESSING SYSTEM USING FPGA AND DISPLAY ON VGA

โดย

นางสาวชลธิชา ฟองสินธุ์

นางสาวแพรวไพลิน เริงเกษตรกิจ

นางสาวมุจลินท์ จันท์พิพัฒน์พงศ์

วพ.
ธ2245
2550

เลขหมู่.....
เลขทะเบียน 83299
วัน,เดือน,ปี 11 ส.ค. 2551

b 11962535
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบประมวลผลภาพเชิงเลขแบบเวลาจริงโดยใช้ FPGA และแสดงผลบน VGA

REAL-TIME DIGITAL IMAGE PROCESSING SYSTEM USING FPGA AND DISPLAY ON VGA

โดย

นางสาวชลธิชา ฟองสินธุ์ 47010150

นางสาวแพรวไพลิน เริงเกษตรกิจ 47010543

นางสาวมุจฉินท์ จันท์พิพัฒน์พงศ์ 47010594

อาจารย์ที่ปรึกษา

อ. ศรวัฒน์ ชิวปรีชา

ผศ. อัครพล ตรีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบประมวลผลภาพเชิงเลขแบบเวลาจริงโดยใช้ FPGA และแสดงผลบน VGA**

REAL-TIME DIGITAL IMAGE PROCESSING SYSTEM USING FPGA AND DISPLAY ON VGA

ผู้จัดทำ

1. นางสาวชลธิชา ฟองสินธุ์ 47010150

2. นางสาวแพรวไพลิน เริงเกษตรกิจ 47010543

3. นางสาวมุจลินท์ จันท์พิพัฒน์พงศ์ 47010594

อาจารย์ที่ปรึกษา

(อ. ศรวัฒน์ ชิวปรีชา)

อาจารย์ที่ปรึกษา

(ผศ. อัครพล ตรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบประมวลผลภาพเชิงเลขแบบเวลาจริงโดยใช้ FPGA และแสดงผลบน VGA

REAL-TIME DIGITAL IMAGE PROCESSING SYSTEM USING FPGA AND DISPLAY ON VGA

โดย นางสาวชลธิชา ฟองสินธุ์ 47010150
นางสาวแพรวไพลิน เริงเกษตรกิจ 47010543
นางสาวมุจลินท์ จันท์พิพัฒน์พงศ์ 47010594

อาจารย์ที่ปรึกษา อ. ศรวัฒน์ ชิวปรีชา
ผศ. อัครพล ตรีรัตน์

บทคัดย่อ

โครงงานนี้นำเสนอการประมวลผลภาพเกรย์สเกล (gray scale)แบบเวลาจริง โดยการใช้อุปกรณ์ FPGA (Field Programmable Gate Array) ซึ่งอธิบายการทำงานของฮาร์ดแวร์ด้วยภาษา VHDL และสามารถแสดงผลบนหน้าจอ VGA ได้ทันที โดยจะทำการอ่านข้อมูลภาพมาจากหน่วยความจำซึ่งเก็บค่าระดับความเข้มแสงของภาพไว้ จากนั้นจะทำการประมวลผลภาพ โดยในโครงงานนี้จะแบ่งการประมวลผลภาพที่ใช้เป็น 4 รูปแบบ คือ การปรับค่าความสว่างของภาพ (Contrast) การปรับระดับค่าเทรสโฮล (Threshold) เพื่อสร้างภาพให้เป็นภาพขาว-ดำ (Binary Image) การขยาย (Zoom) และจัดตำแหน่งภาพ สุดท้ายคือการกรองปาสคาลซึ่งประยุกต์มาจากการแปลงปาสคาลเต็มหน่วย (Discrete Pascal Transform) โดยในส่วนของการกรองปาสคาลนี้ยังได้นำเสนอทั้งทฤษฎี หลักการวิธีการตรวจสอบคุณสมบัติทางความถี่และรูปแบบโครงสร้างทางฮาร์ดแวร์ที่ได้มาจากการแยกเมตริกซ์องค์ประกอบของปาสคาลเมตริกซ์ผลที่ได้จากการประมวลผลภาพแบบต่าง ๆ จะถูกแสดงบนหน้าจอวีจีเอ โดยส่วนควบคุมการแสดงผลภาพนี้ได้ออกแบบไว้ในเอฟพีจีเอเช่นกัน

ABSTRACT

This project presents how to process gray scale image in real-time by using FPGA (Field Programmable Gate Array) for circuits design. Using VHDL describes the operation of hardware. It can be shown immediately while connect to VGA monitor. Image data which is stored by intensity of light in memory will be identified address by FPGA and then run the process. There are 4 types of image processing for example adjusting image contrast, changing threshold to create the binary image, image zooming and setting its position and Pascal filter which is applied from discrete Pascal transform. Moreover, we propose the principle and the method to verify the frequency response. The hardware structure from factorizing Pascal matrix to binary matrices can be shown. The images which were processed in each type will be shown to VGA controlling by FPGA.

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

สารบัญ II

สารบัญรูป VI

สารบัญตาราง XI

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ 1

1.3 ประโยชน์ที่ได้รับ 2

บทที่ 2 ทฤษฎีและหลักการ 3

2.1 การกำเนิดการแสดงผลภาพให้กับจอภาพวีจีเอ 5

2.2 เทคโนโลยีการแสดงผลจอภาพ (Video Display Technology) 5

2.3 การกวาดตรวจภาพ (Scanning) 6

2.4 การซิงก์โครไนซ์ (Synchronization) 7

2.5 อัตราการสแกนภาพ (Scan Rate) 8

2.6 อัตราการแสดงภาพใหม่ (Refresh rate) 8

2.7 องค์ประกอบภาพ 10

2.8 วงจรรวม (ASIC: Application Specific Integrated Circuit) 10

2.9 การเขียนภาษาวีเอชดีแอล 12

2.9.1 การออกแบบจากบนลงล่าง 12

2.9.2 องค์ประกอบของภาษาวีเอชดีแอล 14

2.9.2.1 หน่วยการออกแบบเอนทิตี 14

2.9.2.2 หน่วยการออกแบบสถาปัตยกรรม 15

2.9.2.3 หน่วยการออกแบบแพ็กเก็ต 19

2.9.2.4 หน่วยการออกแบบโครงแบบ 20

2.10 หัวเชื่อมต่อกับหน้าจอวีจีเอ (VGA Connector : DE-15) 21

2.11 รอม (ROM: Read Only Memory) 22

2.11.1 แมนนวลรอม (Manual ROM) 22

2.11.2 พรอม (PROM : Programmable Read-Only Memory) 22

2.11.3 อีพรอม (EPROM : Erasable Programmable Read-Only Memory) 22

2.11.4 อีเอพรอม (EAPROM :Electrically Alterable Read-Only Memory) 22

2.12 การประมวลผลภาพ (Image Processing) 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.13 ทฤษฎีทวินาม (Binomial Theorem) และสามเหลี่ยมปาสคาล (Pascal's Triangle) 23
2.13.1 ที่มาของชื่อสามเหลี่ยมปาสคาล 25
2.13.2 การแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform) 26
2.13.3 การสร้างเมตริกซ์การแปลงปาสคาลขนาดต่าง ๆ 27
2.13.4 วิธีการแยกองค์ประกอบของปาสคาลเมตริกซ์ 28
2.13.5 การแปลงปาสคาลเต็มหน่วยในสองมิติ (2-D Fast Pascal Transform) 32
2.13.6 การตรวจสอบหาคุณลักษณะทางความถี่ (คุณสมบัติการกรองสัญญาณ) ของการแปลงปาสคาลเต็มหน่วย 34
2.14 การกรองข้อมูลภาพ (Image Filtering) 44
2.15 ภาพไบนารี 45
2.16 ค่าเทรสโฮล 45
2.17 ค่าความเข้มแสงของภาพ (Contrast) 45
บทที่ 3 การคำนวณและการสร้าง 46
3.1 การสร้างสัญญาณควบคุมวีจีเอ 46
3.1.1 การสร้างสัญญาณสแกนทางแนวนอน (Horizontal Synchronization) 46
3.1.2 สัญญาณการสแกนทางแนวตั้ง (Vertical synchronization) 48
3.1.3 สัญญาณอ้างอิงตำแหน่งทางแนวนอนและแนวตั้ง 49
3.2 การออกแบบวงจร 50
3.2.1 วงจรที่สร้างจากเอฟพีจีเอ 51
3.2.1.1 วงจรสร้างตำแหน่งของข้อมูล 52
3.2.1.2 วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอน และแนวตั้ง 52
3.2.1.3 วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอ 52
3. 2.1.4 วงจรประมวลผลภาพโดยใช้วงจรกรองปาสคาล 2 มิติ 52
3.2.1.5 วงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ 54
3.2.1.6 วงจรปรับระดับความเข้มแสง 54
3.2.1.7 วงจรปรับความเข้มแสงของภาพเป็นสองระดับ 55
3.3 วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก 56
3.4 วงจรหน่วยความจำภายนอก 58
บทที่ 4 การทดลองและผลการทดลอง 60
4.1 ขั้นตอนการทดลอง 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.1.1 การใช้งานหน่วยความจำภายนอก 60

4.1.2 ผลที่ได้จากโปรแกรม MATLAB 61

4.1.2.1 ผลที่ได้จากการแปลงสามเหลี่ยมปาสคาลใน 1 มิติ 61

4.1.2.2 ผลตอบสนองทางความถี่ที่ได้จากวงจรกรองปาสคาล 2 มิติ 66

4.1.2.2.1 เปรียบเทียบผลที่ได้จากวงจรกรองปาสคาลกับ วงจรกรองความถี่สูงผ่านทั่ว ๆ ไป 72

4.1.3 การออกแบบวงจรส่วนต่าง ๆ โดยใช้ภาษาวีเอชดีแอล 74

4.1.3.1 วงจรส่วนของตัวควบคุมวีจีเอ 74

4.1.3.2 วงจรในส่วนของการสร้างสัญญาณควบคุมการแสกน ทางแนวตั้งและแนวนอนให้กับวีจีเอ 75

4.1.3.3 วงจรส่วนของภาครับของหน่วยความจำภายนอก 75

4.1.3.4 วงจรสร้างตำแหน่งของข้อมูล 75

4.1.3.5 วงจรที่ใช้ในการประมวลผลภาพ 76

4.2 ผลการทดลอง 77

4.2.1 ผลการทดลองจากการจำลองการทำงานในวีเอชดีแอล 77

4.2.1.1 เมื่อทำการนำภาพที่ยังไม่ได้ทำการประมวลผลมาแสดง บนจอวีจีเอ 77

4.2.1.1.1 การทำงานของวงจรรวมทั้งหมด 77

4.2.1.1.2 การสร้างสัญญาณซิงโครไนซ์และสัญญาณสี 78

4.2.1.1.3 การทำงานของภาครับของหน่วยความจำภายนอก 78

4.2.1.1.4 การทำงานของวงจรสร้างตำแหน่งของข้อมูล 79

4.2.1.2 เมื่อทำการนำภาพที่ผ่านการประมวลผลเรียบร้อยแล้ว มาแสดงบนจอวีจีเอ 79

4.2.1.2.1 การทำงานของวงจรรวมทั้งหมด 79

4.2.2 ผลการทดลองที่ได้จากออสซิลโลสโคป 80

4.2.2.1 สัญญาณควบคุมการสแกนหน้าจอทางแนวนอนและแนวตั้ง 80

4.2.2.2 สัญญาณวีดีโอออน (Video on) เทียบกับสัญญาณข้อมูล 83

4.2.3 ผลการทดลองที่ได้จากหน้าจอวีจีเอ 85

4.2.3.1 ภาพที่แสดงผลทางหน้าจอวีจีเอเมื่อยังไม่ได้ผ่านการ ประมวลผลภาพ 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2.3.2 ภาพที่แสดงผลทางหน้าจอวีจีเอเมื่อผ่านการประมวลผลภาพเรียบร้อยแล้ว 90
4.2.3.2.1 เมื่อผ่านการประมวลผลด้วยวงจรปรับระดับความเข้มแสงเป็นสองระดับ (Binary Threshold) 90
4.2.3.2.2 เมื่อผ่านการประมวลผลด้วยวงจรปรับระดับความเข้มแสง (Contrast) 95
4.2.3.2.3 เมื่อผ่านการประมวลผลด้วยวงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ 100
4.2.3.2.4 การประมวลผลภาพโดยการผ่านวงจรกรองปาสคาล 102
บทที่ 5 บทสรุปและวิจารณ์ 109
หนังสืออ้างอิง 111
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 โครงสร้างโดยรวมของชิ้นงานในโครงงาน 1

รูปที่ 2.1 ส่วนประกอบของจอภาพโมโนโครม 4

รูปที่ 2.2 แสดงถึงอุปกรณ์หลอดรังสีแคโทดซึ่งก็คือหลอดแก้วที่ยิงอิเล็กตรอนไปยังฉากด้านหน้า 5

รูปที่ 2.3 การควบคุมการเคลื่อนที่ของลำอิเล็กตรอน 6

รูปที่ 2.4 การสแกนภาพแบบราสเตอร์ 7

รูปที่ 2.5 การสแกนแบบไขว้กัน 9

รูปที่ 2.6 ขั้นตอนการออกแบบวงจรรวม 11

รูปที่ 2.7 แสดงขั้นตอนการออกแบบจากบนลงล่าง 13

รูปที่ 2.8 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี 14

รูปที่ 2.9 แสดงรูปแบบของอาร์เอสฟลิปฟลอป 15

รูปที่ 2.10 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม 16

รูปที่ 2.11 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปตามฟังก์ชันบูลีน $Q = \overline{QB + R}$ และ $QB = \overline{Q + S}$ 16

รูปที่ 2.12 แสดงโครงสร้างภายในสถาปัตยกรรมของอาร์เอสฟลิปฟลอป 17

รูปที่ 2.13 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะโครงสร้าง 17

รูปที่ 2.14 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะพฤติกรรม 18

รูปที่ 2.15 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะ 18

รูปที่ 2.16 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต 20

รูปที่ 2.17 แสดงโครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต 20

รูปที่ 2.18 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ 20

รูปที่ 2.19 หัวเชื่อมต่อกับหน้าจอวีจีเอ 21

รูปที่ 2.20 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก 24

รูปที่ 2.21 แสดงสามเหลี่ยมของ ยาง ฮุย 26

รูปที่ 2.22 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วย 27

รูปที่ 2.23 แสดงตำแหน่งองค์ประกอบ '0' ในส่วนของสามเหลี่ยมบนของแต่ละเมตริกซ์ $\left[q_{ij} \right]_k$ 29

รูปที่ 2.24 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ $N = 3$ 30

รูปที่ 2.25 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ $N = 4$ - 31

รูปที่ 2.26 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ $N = 5$ 32

รูปที่ 2.27 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วยแบบ 2 มิติ เมื่อ N=3 34

รูปที่ 2.28 แสดงโครงสร้างแบบวงจรภายในที่ใช้ในสัญลักษณ์คามรูปที่ 2.27 เมื่อ N=3 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 2.29 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (2.28) 36
รูปที่ 2.30 ผลตอบสนองทางแอมปลิจูดจากสมการที่ (2.28) 36
รูปที่ 2.31 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (2.29) 36
รูปที่ 2.32 ผลตอบสนองทางความถี่จากสมการที่ (2.29) 37
รูปที่ 2.33 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{th} order 37
รูปที่ 2.34 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{th} order
และทำการนอร์มอลไลซ์แล้ว 38
รูปที่ 2.35 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order 38
รูปที่ 2.36 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order
และทำการนอร์มอลไลซ์ แล้ว 39
รูปที่ 2.37 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order 39
รูปที่ 2.38 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order
และทำการนอร์มอลไลซ์แล้ว 40
รูปที่ 2.39 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order 40
รูปที่ 2.40 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order
และทำการนอร์มอลไลซ์แล้ว 41
รูปที่ 2.41 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order
มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order 41
รูปที่ 2.42 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order
มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order เมื่อทำการนอร์มอลไลซ์แล้ว 42
รูปที่ 2.43 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order
ตั้งแต่ 2^{nd} order ถึง 99^{th} order 42
รูปที่ 2.44 แสดงหน้ากากคอนโวลูชันของวงจรกรองปาสคาล 2 มิติ ซึ่งมีขนาด 3x3 43
รูปที่ 2.45 โครงสร้างของวงจรกรองปาสคาล 2 มิติ ตามสมการที่ (2.40) 44
รูปที่ 3.1 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวนอน 46
รูปที่ 3.2 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวตั้ง 48
รูปที่ 3.3 แสดงการอ้างอิงของจุดภาพ 49
รูปที่ 3.4 แสดงการอ้างอิงของจุดภาพ (128x128) 50
รูปที่ 3.5 วงจรรวมทั้งหมดของโครงงาน 50
รูปที่ 3.6 วงจรรวมที่จะนำข้อมูลของภาพจากหน่วยความจำภายนอกเข้าสู่เอฟพีจีเอ
เพื่อทำการประมวลผลภาพและแสดงผลทางวีจีเอ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.7 วงจรสร้างลำดับของข้อมูลก่อนนำไปประมวลผลวงจรกรองปาสคาล 2 มิติ 52
รูปที่ 3.8 แสดงบัตเตอร์ฟลายยูนิต เมื่อ N=3 53
รูปที่ 3.9 ตัวอย่างรายละเอียดของวงจรบัตเตอร์ฟลายยูนิต เมื่อ N=3 53
รูปที่ 3.10 วงจรปรับระดับความเข้มแสง 55
รูปที่ 3.1I วงจรปรับความเข้มแสงของภาพเป็นสองระดับ 56
รูปที่ 3.12 DAC เบอร์ TDA8702 56
รูปที่ 3.13 บล๊อคไดอะแกรมของ TDA8702 57
รูปที่ 3.14 ลักษณะของไอซีที่ใช้ในการแปลงสัญญาณดิจิตอลเป็นอนาล็อก 57
รูปที่ 3.15 การทำงานของหน่วยความจำภายนอก 58
รูปที่ 3.16 ลักษณะอีอีพรอมที่นำมาใช้งาน 59
รูปที่ 3.17 หน่วยความจำภายนอก เบอร์ 28C256 59
รูปที่ 4.1 หน้าต่างการโปรแกรมของหน่วยความจำภายนอก 60
รูปที่ 4.2 รูปการทดลองเรียกใช้ข้อมูลจากหน่วยความจำภายนอก 60
รูปที่ 4.3 แสดงผลตอบสนองทางขนาดของความเป็นตัวกรองความถี่สูงผ่าน
เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{rd} order ถึง 5^{th} order 61
รูปที่ 4.4 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 200 Hz 62
รูปที่ 4.5 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 800 Hz 62
รูปที่ 4.6 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 1000 Hz 63
รูปที่ 4.7 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 2000 Hz 63
รูปที่ 4.8 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB
เมื่อ f1 = 100 Hz และ f2 = 800 Hz 64
รูปที่ 4.9 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB
เมื่อ f1 = 100 Hz และ f2 = 1000 Hz 64
รูปที่ 4.10 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB
เมื่อ f1 = 100 Hz และ f2 = 2000 Hz 65
รูปที่ 4.11 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB
เมื่อ f1 = 100 Hz และ f2 = 2500 Hz 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 4.12 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 66
รูปที่ 4.13 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 4x4 66
รูปที่ 4.14 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 67
รูปที่ 4.15 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 6x6 67
รูปที่ 4.16 ภาพที่ได้จากการแปลงปาสคาลแบบเต็มหน่วย ตัวอย่างที่ 1 68
รูปที่ 4.17 ภาพที่ได้จากวงจรกรองปาสคาล 2 มิติ ตัวอย่างที่ 1 69
รูปที่ 4.18 ภาพที่ได้จากการแปลงดีสครีตปาสคาลแบบเต็มหน่วย ตัวอย่างที่ 2 70
รูปที่ 4.19 การแปลงปาสคาลฟิลเตอร์ ตัวอย่างที่ 2 71
รูปที่ 4.20 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่สูงผ่าน 72
รูปที่ 4.21 ผลตอบสนองทางความถี่ของตัวกรองความถี่สูงผ่านของรูปที่ 4.20 72
รูปที่ 4.22 เปรียบเทียบตัวกรองความถี่สูงผ่านทั่วไปกับวงจรกรองปาสคาล ตัวอย่างที่ 1 73
รูปที่ 4.23 สัญลักษณ์ส่วนของตัวควบคุมวีจีเอ 74
รูปที่ 4.24 สัญลักษณ์ของวงจรภายในทั้งหมดที่ใช้ในการควบคุมวีจีเอ 74
รูปที่ 4.25 สัญลักษณ์ส่วนของภาคขับสัญญาณ 75
รูปที่ 4.26 สัญลักษณ์ส่วนของภาครับของหน่วยความจำภายนอก 75
รูปที่ 4.27 สัญลักษณ์ส่วนของวงจรสร้างตำแหน่งของข้อมูล 75
รูปที่ 4.28 สัญลักษณ์ส่วนของวงจรที่ใช้ในการประมวลผลภาพโดยอาศัยการแปลงปาสคาล 76
รูปที่ 4.29 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพเป็นสองระดับ 76
รูปที่ 4.30 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพ 76
รูปที่ 4.31 สัญลักษณ์ส่วนของวงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ 77
รูปที่ 4.32 ระบบทดสอบ (Testbench) ของวงจรรวมทั้งหมด 77
รูปที่ 4.33 แสดงผลการจำลองการทำงานของวงจรรวมทั้งหมด 77
รูปที่ 4.34 ระบบทดสอบของสัญญาณอ้างอิงทางแนวตั้งและแนวนอน 78
รูปที่ 4.35 แสดงผลการจำลองสัญญาณอ้างอิงทางแนวตั้งและแนวนอน 78
รูปที่ 4.36 ระบบทดสอบของภาครับของหน่วยความจำภายนอก 78
รูปที่ 4.37 แสดงผลการจำลองการทำงานของภาครับของหน่วยความจำภายนอก 78
รูปที่ 4.38 ระบบทดสอบของวงจรสร้างตำแหน่งของข้อมูล 79
รูปที่ 4.39 แสดงผลการจำลองการทำงานของวงจรสร้างตำแหน่งของข้อมูล 79
รูปที่ 4.40 ระบบทดสอบของวงจรรวมทั้งหมด (เมื่อมีส่วนของการประมวลผลภาพด้วย) 79
รูปที่ 4.41 แสดงผลการจำลองการทำงานของวงจรรวมทั้งหมด
เมื่อมีส่วนของการประมวลผลภาพด้วย) 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 4.42 ระบบทดสอบของวงจรในส่วนของการประมวลผลภาพ 80
รูปที่ 4.43 ผลการจำลองการทำงานของวงจรในส่วนของการประมวลผลภาพ 80
รูปที่ 4.44 สัญญาณสัญญาณควบคุมการสแกนทางแนวนอน 81
รูปที่ 4.45 สัญญาณสัญญาณควบคุมการสแกนทางแนวตั้ง 81
รูปที่ 4.46 เมื่อทำการวัดสัญญาณสัญญาณควบคุมการสแกนทางแนวนอน
เปรียบเทียบกับสัญญาณควบคุมการสแกนทางแนวตั้ง 82
รูปที่ 4.47 สัญญาณวีดีโอออน 83
รูปที่ 4.48 สัญญาณวีดีโอออนวัดเทียบกับสัญญาณควบคุมการสแกนทางแนวนอน 84
รูปที่ 4.49 สัญญาณวีดีโอออนวัดเทียบกับแอดเดรสที่เอฟพีจีเอส่งให้กับหน่วยความจำภายนอก 84
รูปที่ 4.50 สัญญาณวีดีโอออนวัดเทียบกับสัญญาณเอาต์พุตจากหน่วยความจำภายนอกให้กับ
เอฟพีจีเอและเอาต์พุตจากเอฟพีจีเอที่จะส่งให้กับวีจีเอ 85
รูปที่ 4.51 แสดงระดับแถบสีแบบเกรย์สเกล 85
รูปที่ 4.52 การเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่นำมาแสดงผลทางหน้าจอวีจีเอ 87
รูปที่ 4.53 แสดงภาพต้นแบบแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1 87
รูปที่ 4.54 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1 87
รูปที่ 4.55 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2 89
รูปที่ 4.56 แสดงภาพต้นแบบของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2 90
รูปที่ 4.57 ภาพต้นแบบ ตัวอย่างที่ 1 90
รูปที่ 4.58 ในกรณีที่ค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1 91
รูปที่ 4.59 ในกรณีที่ค่ามากกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1 92
รูปที่ 4.60 ภาพต้นแบบ ตัวอย่างที่ 2 93
รูปที่ 4.61 ในกรณีที่ค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 2 94
รูปที่ 4.62 ในกรณีที่ค่ามากกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 2 95
รูปที่ 4.63 เมื่อทำการลดระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1 96
รูปที่ 4.64 เมื่อทำการเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1 97
รูปที่ 4.65 เมื่อทำลดเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2 98
รูปที่ 4.66 เมื่อทำเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2 99
รูปที่ 4.67 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 1 100
รูปที่ 4.68 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 2 102
รูปที่ 4.69 ภาพเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่ผ่านวงจรกรองปาสคาลเรียบร้อยแล้ว 108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 คุณสมบัติต่าง ๆ ของหัวเชื่อมต่อหน้าจอวีจีเอ 21
ตารางที่ 3.1 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวนอนและ
ค่าการแปลงเวลาเป็นข้อมูลที่ใช้ในการเขียนโปรแกรม 47
ตารางที่ 3.2 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวตั้งและ
ค่าการแปลงเวลาเป็นข้อมูลที่ใช้ในการเขียนโปรแกรม 48
ตารางที่ 3.3 การใช้งานของแต่ละ ขาของไอซีเบอร์ TDA8702 58
ตารางที่ 3.4 ชื่อของตัวสัญญาณต่าง ๆ ที่ใช้งานในหน่วยความจำภายนอก 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

โลกของเราในปัจจุบันนี้อยู่ในยุคโลกาภิวัฒน์ซึ่งจำเป็นที่จะต้องใช้การสื่อสารข้อมูลระบบดิจิตอลที่มีความแม่นยำสูง เพื่อพัฒนาให้ทันกับความต้องการของระบบการสื่อสารที่เป็นอยู่ ส่งผลให้ความซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันเพิ่มมากขึ้นทุกขณะส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบหรือ CAD (Computer-aided design) มาใช้ในขบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการออกแบบใหม่ๆ ได้ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ (HDL : Hardware Description Language) เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่องเพื่อช่วยให้การปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

โครงงานนี้จะเป็นการประยุกต์ใช้งานของอุปกรณ์ที่เรียกว่าเอฟพีจีเอ (FPGA : Field Programmable Gate Array) ซึ่งเป็นอุปกรณ์ที่ทำให้สามารถออกแบบวงจรรวมได้ง่ายและสะดวกรวดเร็วมากยิ่งขึ้น เราจะทำการนำภาพที่ได้เก็บไว้ในหน่วยความจำซึ่งถูกประมวลผลภาพภายในบอร์ดเอฟพีจีเอด้วยวิธีต่าง ๆ ได้แก่ การแปลงปาสคาล (Pascal transform) การปรับความเข้มแสงของภาพ (Contrast) การขยายขนาดของภาพ การปรับความเข้มแสงของภาพเป็นระดับสีขาว-ดำ และนำมาแสดงผลบนหน้าจอวีจีเอ

รูปที่ 1.1 โครงสร้างโดยรวมของชิ้นงานในโครงงาน

1.2 วัตถุประสงค์

1. ศึกษาการเขียนภาษาวีเอชดีแอล (VHDL : VHSIC Hardeware Describtion Language) ที่ใช้ในการออกแบบระบบฮาร์ดแวร์ดิจิตอลซึ่งเริ่มตั้งแต่การออกแบบการจำลองการทำงานจนถึงการสังเคราะห์วงจรเพื่อจะใช้ในการโปรแกรมลงในเอฟพีจีเอ
2. ศึกษาการทำงานของจอแสดงผลวีจีเอ (VGA : Video Graphics Array)
3. ศึกษาองค์ประกอบของภาพและการประมวลผลภาพของภาพเกรย์สเกล (gray scale)
4. สามารถประยุกต์เรียกใช้งานหน่วยความจำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ศึกษาการใช้งานและเก็บข้อมูลลงในหน่วยความจำเพื่อที่จะนำไปใช้ควบคู่กับเอฟพีจีเอ
6. ศึกษาการประมวลผลภาพโดยอาศัยหลักการของการแปลงปาสคาลเต็มหน่วย (Discrete Pascal Transform) ซึ่งมีรากฐานมาจากทฤษฎีบททวินามและนำมาประยุกต์เป็นการออกแบบวงจรกรองปาสคาล (Pascal Filter)

1.3 ประโยชน์ที่ได้รับ

1. สามารถประยุกต์ใช้งานอุปกรณ์เอฟพีจีเอได้อย่างกว้างขวาง
2. สามารถเขียนโปรแกรมควบคุมการทำงานของเอฟพีจีเอได้
3. สามารถควบคุมการทำงานของหน้าจอวีจีเอได้
4. สามารถนำการแสดงผลบนหน้าจอวีจีเอไปประยุกต์ใช้งานด้านต่าง ๆ ได้
5. สามารถทำการประมวลผลภาพด้วยวิธีต่าง ๆ เพื่อที่จะนำไปประยุกต์ใช้งานอื่น ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

ในอดีตจอภาพมีหลายแบบตั้งแต่ชนิดโมโนโครม (Monochrome) ซึ่งเป็นจอเขียวหรือจอขาวดำ, ซีจีเอ (CGA : Color Graphics Adapter) 4 สี, อีจีเอ (EGA : Enhanced Graphics Adapter) 16 สี และ วีจีเอ 256 สี ในปัจจุบันเราใช้จอภาพแสดงผลแบบซุปเปอร์วีจีเอ (SVGA) เป็นมาตรฐานหลักในการแสดงผลบนเครื่องพีซี

การทำงานของจอภาพนั้นบนหน้าจอจะประกอบไปด้วยพิกเซล (Pixel) หรือจุดขนาดเล็กในการแสดงผล โดยจุดเหล่านี้มีขนาดเล็กมากและจะกระพริบอยู่ตลอดเวลา ในความถี่ต่าง ๆ เมื่อมีการแสดงผล จุดหลอดภาพเล็ก ๆ เหล่านี้ที่อยู่กันเป็นกลุ่มจะเปิด ปิด เพื่อให้เกิดความแตกต่างกันจนออกมาเป็นตัวอักษรหรือรูปภาพที่ต้องการ

จอภาพที่ดีจะต้องมีขนาดของความละเอียด (Resolution) ที่สูงจะได้ความคมชัดมากขึ้น นอกจากนี้อัตราการแสดงภาพใหม่ (Refresh rate) ของจอภาพก็เป็นเรื่องสำคัญที่ควรจะต้องพิจารณาเพราะหากความละเอียดสูงภาพจะคมชัดขึ้นหรือสามารถมีรายละเอียดมากขึ้นเป็นต้น ส่วนอัตราการแสดงภาพใหม่ จะเป็นอัตราที่บอกความถี่ในการแสดงผลและการสแกนภาพ เพื่อให้เกิดความคมชัดและไม่เป็นอันตรายต่อสายตาโดยทั่วไปนิยมตั้งไว้ที่ 50 – 60 *Hz* และนอกจากนั้นปริมาณสีที่สามารถแสดงผลได้เป็นเรื่องที่จำเป็นอีกประการหนึ่ง

จอภาพคอมพิวเตอร์ (Monitor) ที่ใช้กับระบบคอมพิวเตอร์ในปัจจุบันจะเป็นจอภาพแบบวีจีเอหรือซุปเปอร์วีจีเอทั้งสิ้น โดยมีขนาดที่นิยมใช้กันตั้งแต่ขนาด 14 นิ้ว 15 นิ้ว 17 นิ้ว 20 นิ้ว และ 21 นิ้ว โดยที่ขนาดของหน้าจอที่ต้องการจะขึ้นอยู่กับลักษณะของงานที่ใช้ ถ้าต้องการงานที่มีรายละเอียดสูงจะต้องใช้จอที่มีขนาดใหญ่ แต่จอที่มีขนาดใหญ่ก็มีราคาแพงและอาจไม่คุ้มค่าในการลงทุน

ปัจจุบันจอภาพยังมีหลายรูปแบบเช่น จอแบบแบน (Flat LCD Screen) เป็นจอภาพที่ใช้กับเครื่องคอมพิวเตอร์แบบพกพาได้ (Portable) เช่น โน๊ตบุ๊ค (Laptop) เป็นหลัก จอภาพแบบนี้จะประหยัดไฟฟ้าและยังมีรังสีแผ่กระจายน้อยมากอีกด้วย แต่ปัจจุบันยังมีราคาแพงอยู่จึงเป็นปัญหาของการใช้งานในองค์กร ขนาดของจอภาพแบบนี้โดยทั่วไปจะมีขนาด 14-15 นิ้ว เป็นหลัก หรือจอภาพแบบหลอดรังสีแคโทด (CRT : Cathode-Ray Tube) ที่เป็นจอภาพสำหรับคอมพิวเตอร์แบบตั้งโต๊ะ

จอภาพที่ใช้แสดงผลของคอมพิวเตอร์มักถูกเรียกว่ามอนิเตอร์ ซึ่งในการแสดงผลนั้นมักจะประกอบด้วยข้อมูลทั้งที่เป็นตัวหนังสือและภาพกราฟิก สำหรับเครื่องคอมพิวเตอร์แบบตั้งโต๊ะโดยทั่ว ๆ ไปจะใช้เทคโนโลยีจอภาพแบบหลอดรังสีแคโทด ในขณะที่คอมพิวเตอร์แบบพกพา เช่น โน๊ตบุ๊ค จะใช้เทคโนโลยีแบบลิควิดคริสตัลดิสเพลย์ และไลท์อิมิทติงไดโอด (LCD : Liquid Crystal Display, LED : Light-Emitting Diode) หรือแก๊สพลาสมา(GasPlasma)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอภาพแบ่งออกได้ 2 ประเภทคือ

● จอภาพสีเดียวหรือโมโนโครม เป็นจอภาพที่ใช้กันทั่วไปในสมัยก่อน แต่ปัจจุบันใช้เฉพาะในบางที่ เช่น เครื่องคิดเงินในร้านค้า หรือตู้เอทีเอ็มของบางธนาคาร เป็นต้น มีลักษณะการแสดงผลเป็นสีเดียว เช่น สีขาว สีเขียว หรือสีส้มและแสดงได้เฉพาะตัวอักษรเท่านั้น

- จอภาพโมโนโครมแบบไดเรกไดร์ฟ (Direct-drive) ใช้ได้กับทั้งหน้าจอโมโนโครม และหน้าจออีจีเอ

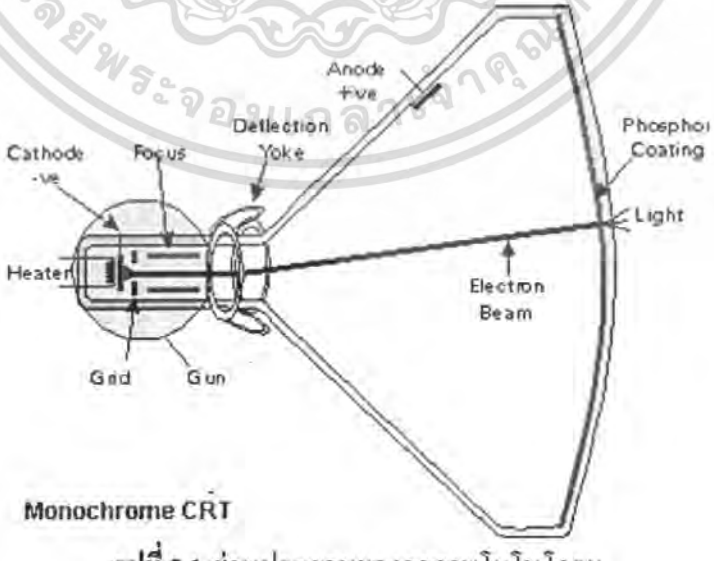
- จอภาพโมโนโครมแบบคอมโพสิท (Composite) จอภาพชนิดนี้จะใช้ได้กับเฉพาะซีจีเอเท่านั้น โดยเสียบสายสัญญาณผ่านทางคอมโพสิทวีดีโอเอาท์พุต (Composite video output) คือสัญญาณภาพรวม (Composite Video signal) โดยสัญญาณทั้งหมดจะส่งมาในสัญญาณนี้เพียงสัญญาณเดียว

● จอภาพสีเป็นจอภาพที่ใช้กันทั่วไปในปัจจุบัน ซึ่งมีลักษณะการแสดงผลอาศัยสัญญาณดิจิตอล แต่จะแยกออกเป็น 3 สัญญาณตามแม่สีของแสงคือ แดง เขียว และน้ำเงิน หรือ อาร์จีบี (RGB : Red Green Blue) ที่จะทำให้เกิดสีต่าง ๆ มากมายตามหลักการผสมของแม่สีนั่นเอง

- จอภาพสีแบบคอมโพสิทจอภาพสีแบบนี้สามารถแสดงสีและกราฟิกได้ แต่คุณภาพความละเอียดของภาพกราฟิกจะต่ำ

- จอภาพสีแบบอาร์จีบี จอภาพสีแบบนี้คุณภาพดีในการแสดงข้อความและมีความละเอียดสูงในงานกราฟิกแบบที่ต้องการความละเอียดสูง

- จอภาพแบบหลายความถี่ (Variable Frequency Monitors) จอภาพสีแบบนี้จะมีพื้นฐานคล้ายคลึงกับจอภาพแบบอาร์จีบีแต่มีหลายความถี่จึงเป็นจอที่มีความสามารถสูงใช้งานได้กับหลากหลายระบบ เพราะเป็นจอที่สามารถเข้ารหัสสัญญาณสีได้มากกว่าหนึ่งแบบทั้งแบบดิจิตอลหรืออนาล็อกและใช้กับช่วงสัญญาณความถี่ที่กว้างขึ้น



รูปที่ 2.1 ส่วนประกอบของจอภาพโมโนโครม

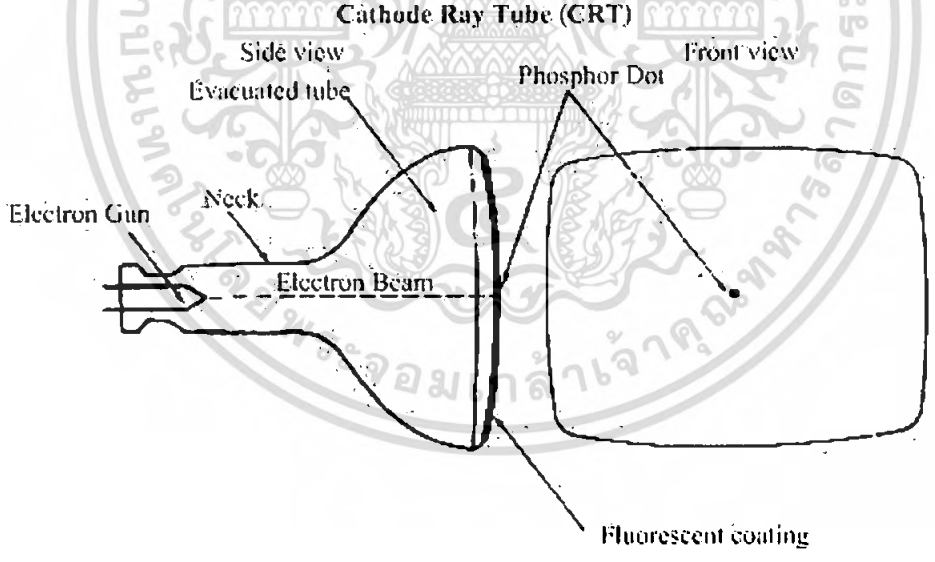
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การกำเนิดการแสดงผลภาพให้กับจอภาพวีจีเอ

สัญญาณที่ใช้ในระบบภาพวีจีเอ ประกอบด้วย 5 สัญญาณด้วยกัน โดยมี 2 สัญญาณที่เกี่ยวข้องกับระดับสัญญาณลอจิก (TTL) นั่นคือ สัญญาณควบคุมการสแกนหน้าจอทางแนวนอน (Horizontal Sync) และสัญญาณควบคุมการสแกนหน้าจอทางแนวตั้ง (Vertical Sync) ซึ่งใช้สำหรับสังเคราะห์สัญญาณภาพ ส่วนอีก 3 สัญญาณนั้นจะเกี่ยวข้องสัญญาณอนาล็อก โดยมีค่า 0.7 - 1.0 V_{p-p} จะใช้ในควบคุมสัญญาณสี ซึ่งสัญญาณสีประกอบด้วยสีแดง สีเขียว และสีน้ำเงิน สัญญาณสีเหล่านี้เรียกรวมว่า สัญญาณอาร์จีบี

2.2 เทคโนโลยีการแสดงผลจอภาพ (Video Display Technology)

เทคโนโลยีที่ใช้โดยส่งการแสดงผลเป็นสัญญาณภาพนิ่งที่เหมือนธรรมชาติในรูปแบบของสัญญาณภาพจะใช้หลอดภาพรังสีแคโทดซึ่งเกิดขึ้นครั้งแรกเมื่อปี ค.ศ. 1897 โดยนักวิทยาศาสตร์ชาวเยอรมันชื่อ คาร์ล เฟอร์ดินาน บัวร์น (Karl Ferdinand Braun) โดยพัฒนาเป็นเครื่องมือที่เรียกว่า ออสซิลโลสโคป (Oscilloscope) ที่ใช้เป็นเครื่องวัดค่าสัญญาณทางไฟฟ้า และเริ่มผลิตออกมาในเชิงพาณิชย์ครั้งแรกในปี ค.ศ. 1922 ในรูปแบบของจอโทรทัศน์จนถึงทุกวันนี้ หลอดภาพรังสีแคโทดเป็นอุปกรณ์ที่ใช้กันแพร่หลายทั่วไปส่วนมากจะมีใช้ในรูปแบบชุดโทรทัศน์บ้างหรือใช้เป็นตัวเชื่อมต่อในการแสดงผลของระบบคอมพิวเตอร์



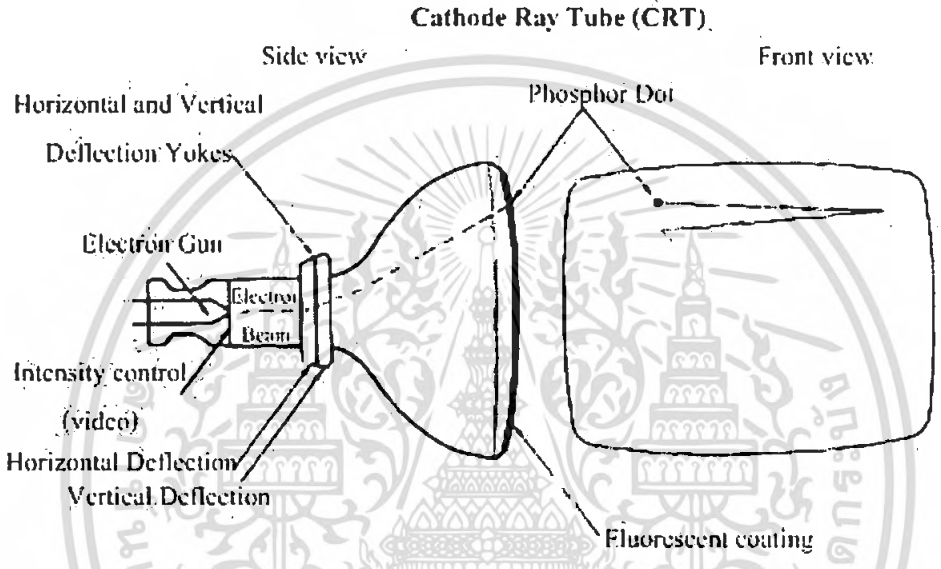
รูปที่ 2.2 แสดงถึงอุปกรณ์หลอดรังสีแคโทดซึ่งก็คือหลอดแก้วที่ยิงอิเล็กตรอนไปยังฉากด้านหน้า

ที่ฉากของจอภาพพื้นผิวด้านในจะเคลือบสารฟลูเรซเซนท์ (Fluorescent) อยู่ และที่ตรงส่วนท้ายของหลอดภาพ จะมีปืนยิงลำอิเล็กตรอน เมื่อปืนนี้ยิงไปกระตุ้นที่ฉากก็จะทำให้เกิดจุดสว่างขึ้น

ซึ่งหลอดอิเล็กตรอนนี้จะยิงลำอิเล็กตรอน (Electron beam) กวาดสแกนบนฉากตามแนวนอนที่ต่อเนื่องกัน และเพื่อที่จะกำเนิดมาเป็นภาพ จะมีแกนเหล็กสำหรับเบี่ยงเบนลำอิเล็กตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Deflection yoke) ขดลวดเลี้ยวเบนนี้จะแบ่งเป็นแกนแนวตั้งกับแนวนอน (จะอยู่ที่คอของหลอดรังสีแคโทด) โดยจะอาศัยสนามแม่เหล็ก (หรืออาจจะใช้สนามไฟฟ้าสถิตก็ได้ แต่ไม่เป็นที่นิยม) เบี่ยงเบนลำอิเล็กตรอนให้ไปตามตำแหน่งที่ต้องการบนฉากของจอภาพ ส่วนข้อมูลของสัญญาณอาร์จีบี ที่ออกมาเป็นสัญญาณภาพจะใช้ควบคุมความแรงของอิเล็กตรอน และแสงที่เกิดขึ้นบนจอภาพนั้นเกิดจากการที่มีสัญญาณภาพเข้ามาซึ่งจะทำให้ลำอิเล็กตรอนทำงาน และสามารถพุ่งไปปะทะจุดสีเรืองแสงบนฉากของหลอดรังสีแคโทดซึ่งบนฉากของหลอดรังสีแคโทดจะเคลือบด้วยสารเรืองแสง 3 สี (สีแดง, สีเขียว และสีน้ำเงิน)



รูปที่ 2.3 การควบคุมการเคลื่อนที่ของลำอิเล็กตรอน

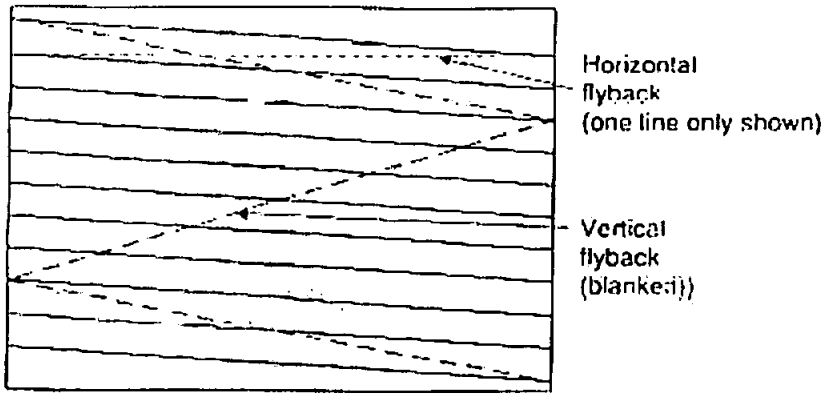
2.3 การกวาดตรวจภาพ (Scanning)

การที่จะนำภาพ 2 มิติเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งทำได้โดยทำภาพนั้นเป็นสัญญาณไฟฟ้าซึ่งจะง่ายต่อการเคลื่อนย้าย แต่ปัญหาคือจะทำการเปลี่ยนภาพ 2 มิติไปเป็นโวลต์เตจที่เปลี่ยนตามเวลาปัญหานี้แก้ได้โดยทฤษฎีของการกวาดตรวจ

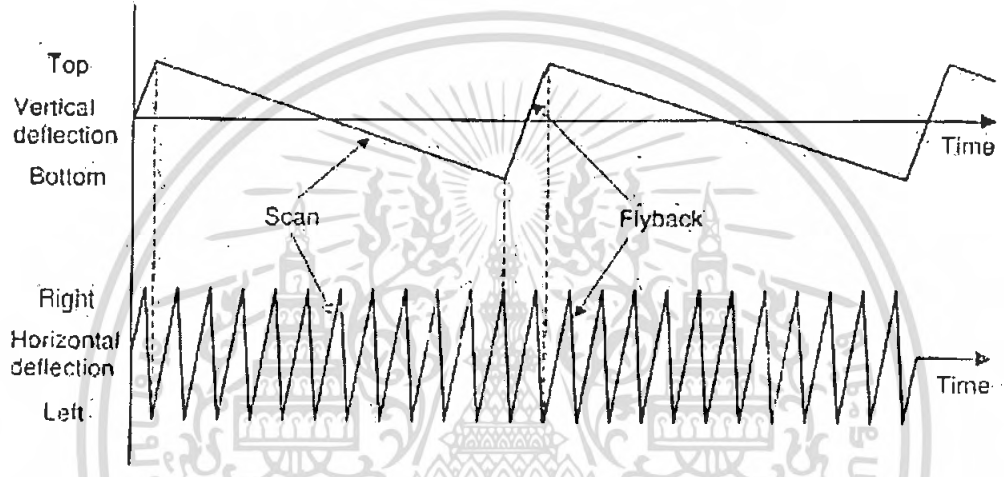
วิธีการสแกนของหลอดรังสีแคโทดที่นิยมใช้กันอย่างแพร่หลายในการออกแบบทางการค้าโทรทัศน์และจอภาพแสดงผล มีชื่อรียกว่า การตรวจกวาดแบบราสเตอร์ (Raster scan) ซึ่งวิธีนี้จะต้องมีการให้กำเนิดซิงก์ทั้งในแนวแกนตั้งและแนวนอนเพื่อให้ลำอิเล็กตรอนสามารถเคลื่อนที่ได้

ในการสแกนแบบนี้จะเริ่มทำจากซ้ายไปขวา และจากข้างบนลงไปข้างล่างของจอภาพ โดยที่ลำอิเล็กตรอนจะถูกเลี้ยวเบนให้ไปอยู่ทางมุมซ้ายบนและเมื่อกวาดไปทางขวาของฉาก ลำอิเล็กตรอนก็จะสะบัดกลับโดยตอนที่กลับนั้นก็จะมีการเลื่อนตำแหน่งลงไปด้วย การสะบัดกลับนี้เรียกว่ารีทราส (Retrace) หรือ ฟลายแบ็ค (fly back) ซึ่งในส่วนของการสะบัดกลับของลำอิเล็กตรอนนั้นความเข้มของลำอิเล็กตรอนจะลดลง เมื่อลดความเข้มลงอย่างเพียงพอแล้วฉากที่เคลือบด้วยฟลูเรซเซนท์ก็จะไม่ถูกกระตุ้นทำให้ไม่เกิดแสงสว่างจึงมองไม่เห็นเส้นที่ลากกลับ ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 2.4 การสแกนภาพแบบราสเตอร์

(ก) การเคลื่อนที่ของการสแกนภาพ

(ข) กราฟแสดงความสัมพันธ์ของการเคลื่อนที่ของการสแกนกับเวลา

2.4 การซิงก์โครไนซ์ (Synchronization)

สิ่งสำคัญที่ทำให้การสแกนภาพนิ่งได้ถูกต้องนั้นเป็นหน้าที่ของสัญญาณซิงก์ซึ่งแบ่งได้เป็นสัญญาณควบคุมการสแกนทางแนวนอน และสัญญาณควบคุมการสแกนทางแนวตั้ง โดยทำงานไปพร้อม ๆ กันได้นั้นเพราะสัญญาณนี้จะส่งข้อมูลตามเวลาแนบไปกับสัญญาณวีดีโอ

การทำงานของสัญญาณควบคุมการสแกนในแนวแกนนอนและแกนตั้ง

- สัญญาณควบคุมการสแกนทางแนวนอนจะเริ่มทำงานที่การเริ่มสะบัดกลับ หรือจบการสะบัดกลับ จะไม่ทำตอนเริ่มสะบัดกลับ
- สัญญาณควบคุมการสแกนทางแนวตั้งเริ่มทำงานที่การเริ่มสะบัดกลับในแนวแกนตั้ง (ก็คือจะทำงานตอนหลอดภาพยิงไปอยู่ที่ล่างขวาของจอภาพ)

ถ้าหากไม่มีสัญญาณควบคุมการสแกนทางแนวตั้งจะทำให้ภาพที่ถูกสร้างทวนซ้ำขึ้นมาจะไม่สามารถคงตำแหน่งในแนวแกนตั้ง ภาพจะวิ่งขึ้น-ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าหากสัญญาณควบคุมการสแกนทางแนวนอนไม่ซิงก์กันจะทำให้ภาพนั้นจะเลื่อนซ้าย-ขวา และบางส่วน จะเหมือนภาพถูกแยกออก

โดยทั่วไปแล้วจอภาพส่วนใหญ่จะใช้สัญญาณอนาล็อกอาร์จีบีมาตรฐานที่ 0.7 V_{p-p} ส่วนสัญญาณซิงก์ก็จะขึ้นอยู่กับประเภทลักษณะที่ใช้ดังต่อไปนี้

1. ในการเชื่อมต่อสัญญาณอาร์จีบี สัญญาณควบคุมการสแกนทางแนวดั้ง และสัญญาณควบคุมการสแกนทางแนวนอน จะใช้สายในการเชื่อมต่อ 5 เส้น มาตรฐานของสัญญาณอาร์จีบีที่ใช้คือ 0.7 V_{p-p} และสัญญาณซิงก์ใช้มาตรฐานทางลอจิก ลักษณะการใช้งานแบบนี้จะง่ายและเป็นที่นิยม

2. ในการเชื่อมต่อสัญญาณอาร์จีบีและสัญญาณคอมโพสิท (เรียก CSYNC : Composite sync) จะใช้สายในการเชื่อมต่อ 4 เส้น เพราะจะใช้สัญญาณซิงก์ 1 เส้น ใช้มากในพวกโปรเจคเตอร์ในสมัยก่อน

3. ในการเชื่อมต่อสัญญาณอาร์จีบีและสัญญาณซิงค์ออนกรีน (Sync on green) จะใช้สายในการเชื่อมต่อ 3 เส้น สัญญาณสีแดง และสีน้ำเงิน ยังคงใช้ 0.7 V_{p-p} แต่สัญญาณสีเขียว จะส่งรวมไปกับสัญญาณคอมโพสิท ซึ่งจะใช้งานที่ -0.3 V_{p-p}

2.5 อัตราการสแกนภาพ (Scan Rate)

ในมาตรฐานของระบบภาพของสหรัฐอเมริกา อัตราการกวาดเส้นในแนวนอน (The horizontal scanning frequency or line rate) ของลำอิเล็กตรอนสำหรับโทรทัศน์คือ 31.5 *kHz* และถ้าต้องการให้เคลื่อนที่ไปพร้อมกันในแนวตั้ง (The vertical scanning frequency or frame rate or field rate) มีอัตราการกวาดที่ 60 *Hz* ซึ่งก็จะทำให้เกิดเส้นกวาดในแนวนอนจากบนสุดจนถึงล่างสุดของฉากทั้งหมดเท่ากับ $31500 \div 60 = 525$ เส้น

จำนวนสัญลักษณ์ที่จะแสดงได้ใน 1 แถวนั้นจะขึ้นอยู่กับความถี่ของสัญญาณภาพที่ใช้ ส่วนจำนวนบรรทัดที่จะแสดงผลบนจอภาพนั้นจะขึ้นอยู่กับอัตราการสแกนภาพ ซึ่งสามารถเพิ่มได้ วิธีที่เห็นได้ชัดที่สุดก็คือการเพิ่มความถี่ในการกวาดในแนวนอน ซึ่งก็จะทำให้เกิดเส้นสแกนแนวนอนมากขึ้นตัวอย่างเช่น

ถ้าเพิ่มความถี่ในแนวนอนจาก 31.5 *kHz* เป็น 36 *kHz* แล้วจะทำให้เกิดเส้นกวาดแนวนอนเท่ากับ $36000 \div 60 = 600$ เส้น แต่อย่างไรก็ตามการเพิ่มความถี่การกวาดแนวนอนนั้น ก็จะทำให้ผลลัพธ์ที่ได้ไม่เป็นมาตรฐานเช่นกัน นอกจากนี้ยังต้องเพิ่มความถี่วีดีโอ เพื่อให้จำนวนสัญลักษณ์เหมาะสมกับจำนวนตัวสัญลักษณ์ต่อเส้นกวาดแนวนอน

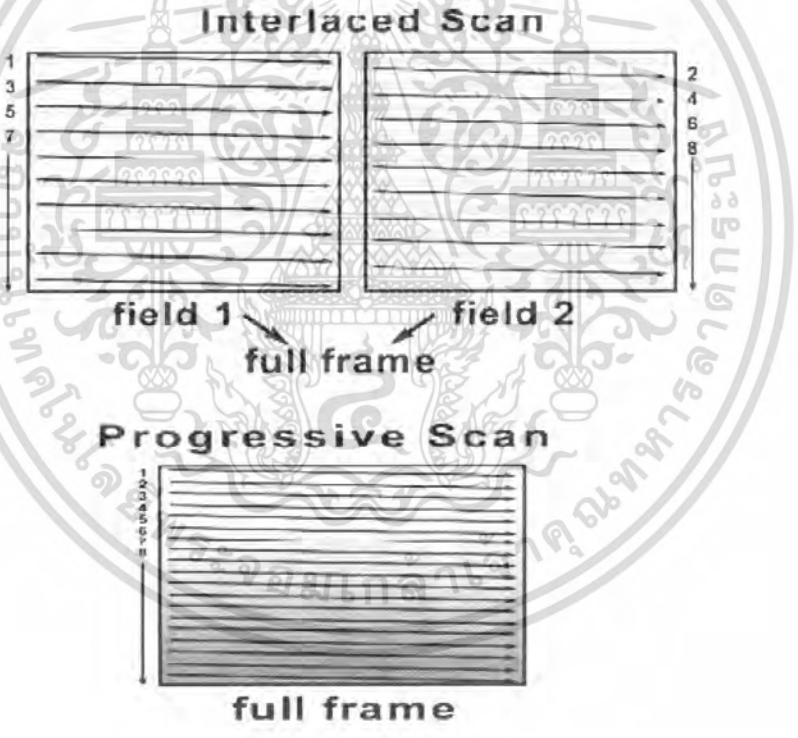
2.6 อัตราการแสดงภาพใหม่ (Refresh rate)

คือ อัตราส่วนของจำนวนครั้งที่จุดภาพถูกฉายลงบนหน้าจอภาพ.ซึ่งจะเริ่มจากซ้ายไปขวาและบนลงล่างในเวลา 1 วินาทีซึ่งเป็นค่าที่สำคัญเพราะว่ามีผลกระทบโดยตรงต่อการมองเห็นภาพที่แสดงบนหน้าจอภาพ ปัญหาหลักของการเพิ่มเส้นสแกนนั้นคือถ้าอัตราการแสดงภาพใหม่แนวตั้งนั้นต่ำกว่าความถี่มาตรฐานของหลอดรังสีแคโทดก็จะทำให้ลำอิเล็กตรอนเลี้ยวเบนด้วยสนามไฟฟ้าที่มีรูปแบบไม่ถูกต้องผลกระทบนี้เป็นเหตุให้เกิดแสงภาพของจอกระตุกสั่น การแก้ปัญหาก็คือเราควรใช้อัตรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงภาพใหม่นั้นให้เท่าความถี่มาตรฐาน ดังนั้นในสหรัฐอเมริกา ที่ความถี่การกวาดในแนวนอนเท่ากับ 31.5 *kHz* ควรใช้ความถี่ในการแสดงภาพใหม่มากกว่าหรือเท่ากับ 60 *Hz* ในขณะที่ประเทศไทยก็ใช้ตั้งแต่ความถี่ 50 *Hz* ขึ้นไป เพราะถ้าอัตราการแสดงภาพใหม่น้อยไปก็จะทำให้ภาพนั้นสั่นกระพริบเป็นอันตรายต่อสายตา ค่าอัตราการแสดงภาพใหม่นี้ก็จะขึ้นกับสายตาของแต่ละคน และก็ยังจะขึ้นอยู่กับขนาดของจอภาพด้วยเช่นกัน โดยที่จอภาพขนาดใหญ่จะมีการสั่นของภาพได้ง่ายกว่าจอภาพที่มีขนาดเล็ก ฉะนั้นจอภาพขนาดใหญ่ควรใช้อัตราการแสดงภาพใหม่สูงกว่าจอภาพขนาดเล็ก

อย่างไรก็ตามยังมีวิธีการลดการกระพริบของการสแกนภาพนิ่งแต่ละภาพ ซึ่งจะนิยมใช้วิธีการสแกนไขว้กัน (Interlace scanning) ตามรูปที่ 2.5 โดยที่ภาพหนึ่งเฟรมจะประกอบด้วยภาพนิ่ง 2 ฟิลด์ โดยที่เริ่มต้นด้วยการสแกนภาพนิ่งฟิลด์เส้นคี่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งกลางของแนวเส้นสุดท้าย ลำอิเล็กตรอนก็จะกลับไปเริ่มสแกนใหม่แต่จะสแกนที่ฟิลด์ของเส้นคู่ หลังจากที่สแกนเสร็จทั้ง 2 ฟิลด์แล้วก็จะได้ภาพหนึ่งเฟรม



รูปที่ 2.5 การสแกนแบบไขว้กัน

การกวาดสแกนสัญญาณภาพหรือการเริ่มสแกนใหม่ จะเป็นไปตามกระบวนการของสัญญาณควบคุมการสแกนทางแนวตั้ง ซึ่งจะแสดงถึงการเริ่มแสดงภาพใหม่หรือเฟรมใหม่ ส่วนสัญญาณควบคุมการสแกนทางแนวนอนจะแสดงถึงการเริ่มใหม่ในแถว สัญญาณควบคุมการสแกนทางแนวตั้งก็จะทำการรีเซ็ทตำแหน่งลำอิเล็กตรอนไปที่มุมซ้ายบน (จุด (0, 0) นั่นเอง) และเริ่มทำกระบวนการเดิมไปเรื่อย ๆ ในขณะที่กลับไปตำแหน่งเริ่มต้นสัญญาณอาร์จีบีจะมีค่าเป็น "000" หรือสีดำนั่นเอง แต่ถ้าสัญญาณซิงค์ทั้ง 2 ค่านั้นไม่สอดคล้องกันแล้ว ส่วนใหญ่ในปัจจุบันจอภาพจะมีแอลอีดีแสดงสถานะ ถ้าแอลอีดีมีสีเขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงว่ามีการตรวจพบสัญญาณซิงค์ และถ้าแอลอีดีเป็นสีเหลืองแสดงว่าตรวจไม่พบสัญญาณซิงค์ ในการ์ดวีจีเอจะมีหน่วยความจำซึ่งทำหน้าที่เก็บค่าของสีทุกพิกเซล และเวลาการแสดงผลหน่วยความจำนี้เมื่อถูกอ่านออกไปแล้วจะทำหน้าที่เหมือนเป็นตัวผลิตสัญญาณอาร์จีบีแล้วส่งให้ลำอิเล็กตรอนยิงกวาดไปบนฉาก

2.7 องค์ประกอบภาพ

ในภาพหนึ่งภาพจะมีองค์ประกอบมาจากจุดสีดำมากมายมาเรียงกันประกอบขึ้นเป็นภาพ ซึ่งจุดเหล่านี้เองที่เรียกว่าองค์ประกอบภาพ (Picture element) หรือเรียกว่าพิกเซล ภาพที่ปรากฏบนจอภาพจะใช้หลักการเดียวกัน ภาพที่เกิดขึ้นบนจอภาพประกอบด้วยเส้นแนวนอนเส้นเล็ก ๆ เป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิทและสว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสงความแตกต่างกับการเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองเราก็จัดว่าเป็นองค์ประกอบภาพ

มาตรฐานระบบสแกน 525 เส้น แต่ละเส้นจะแบ่งได้เป็น 800 เส้น ดังนั้นจึงกล่าวได้ว่าหากจะหาองค์ประกอบภาพจะได้ $525 \times 800 = 420,000$ Pixels ใช้แบนวิดท์ 6 *MHz* ยิ่งภาพมีจำนวนเส้นสแกนมากเท่าไร รายละเอียดของภาพหรือความคมชัดก็จะมากขึ้นเท่านั้น แต่การออกแบบก็จะยากขึ้นด้วยเช่นกัน เนื่องจากภาพที่มีรายละเอียดมากก็ต้องการแบนวิดท์มากขึ้นด้วย จากที่กล่าวมาว่าองค์ประกอบของภาพมีหลายพิกเซล แต่ในความเป็นจริงแล้วเราไม่สามารถมองเห็นได้ครบทุกพิกเซลเนื่องจากจะมีบางส่วนหายไประหว่างการสแกนภาพ

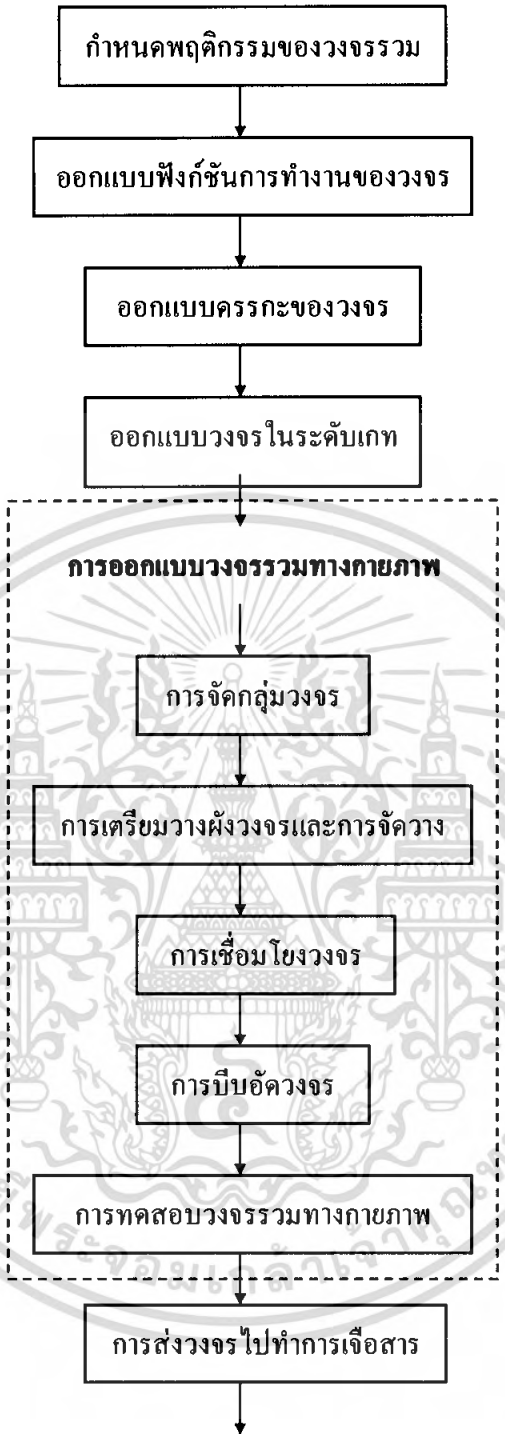
ระบบสแกนที่ใช้ในประเทศไทยคือ 625 เส้น ซึ่งจะต้องใช้แบนวิดท์เท่ากับ 7 *MHz* ความละเอียดก็จะมากขึ้นกว่าระบบสแกน 525 เส้นด้วย องค์ประกอบภาพที่ได้จะเท่ากับ $625 \times 851 = 531,875$ pixels

ปัจจุบันสำหรับโทรทัศน์ธรรมดาเราพบว่ามีการเพิ่มเส้นภาพให้มากขึ้น และแน่นอนจำนวนพิกเซลก็มากขึ้นด้วย อย่างระบบโทรทัศน์แบบรายละเอียดสูงหรือเอชดีทีวี (HDTV) จะมีจำนวนเส้นสแกนเท่ากับ 725 เส้น ส่วนในโปรเจคเตอร์จะมีจำนวนเส้นสแกน 2200 เส้น

2.8 วงจรรวม (ASIC: Application Specific Integrated Circuit)

ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่าง ๆ ซึ่งสามารถลดค่าใช้จ่าย ลดการสิ้นเปลืองขนาดพื้นที่ รวมทั้งพลังงานที่ใช้กับอุปกรณ์และในขณะเดียวกันยังมีการเพิ่มประสิทธิภาพและระดับความน่าเชื่อถือของวงจรที่สูงขึ้นอย่างเห็นได้ชัดจากเทคโนโลยีไมโครโปรเซสเซอร์และหน่วยความจำปัจจุบัน ในการพัฒนาเพิ่มความหนาแน่นและจำนวนฟังก์ชันที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิตอลได้พิจารณาถึงการผลิตในจำนวนมาก ๆ และ การผลิตวงจรรวม ซึ่งวงจรรวมจะสามารถแบ่งได้เป็น 2 กลุ่ม คือ ฟิลโปรแกรมเมเบิล (Field Programmable) และ แมสโปรแกรมเมเบิล (Mask Programmable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- **รูปที่ 2.6** ขั้นตอนการออกแบบวงจรรวม

ในการออกแบบวงจรรวมต้องผ่านขั้นตอนต่าง ๆ ดังรูปที่ 2.6 ซึ่งเริ่มต้นจากการกำหนดการทำงานของระบบใหญ่ของวงจรรวม (System Specification) ซึ่งประกอบระบบ ในแต่ละส่วนย่อยจะถูกกำหนดคุณสมบัติเป็นฟังก์ชัน (Functional Design) จากนั้นนำมาทำการออกแบบวงจรตรรกะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Logic Design) การออกแบบวงจร (Circuit) แบบทางกายภาพ (Physical Design) ไปสู่ขบวนการเจือสาร (Fabrication) ตามลำดับ เมื่อไอซีหรือชิพที่ผลิตแล้วส่งกลับมาก็จะทำการทดสอบการทำงานของชิพว่าตรงกับที่ออกแบบหรือไม่ (Packaging and Testing) เวลาทำงานของแต่ละขั้นตอนจึงเป็นสิ่งที่ต้องคำนึงถึง

การผลิตวงจรรวมเฉพาะกิจ แบ่งเป็น 2 ขั้นตอน คือ การออกแบบ และ การเจือสาร

ในการออกแบบทางกายภาพของวงจรรวมยังแบ่งขั้นตอนย่อยออกเป็น การจัดกลุ่มวงจร (Circuit Partitioning) การวางผัง (Floor Planning) การจัดวางตัวอุปกรณ์หรือกลุ่ม (Placement) การเชื่อมโยง (Routing) การบีบอัด (Compaction) และการทดสอบ (Verification)

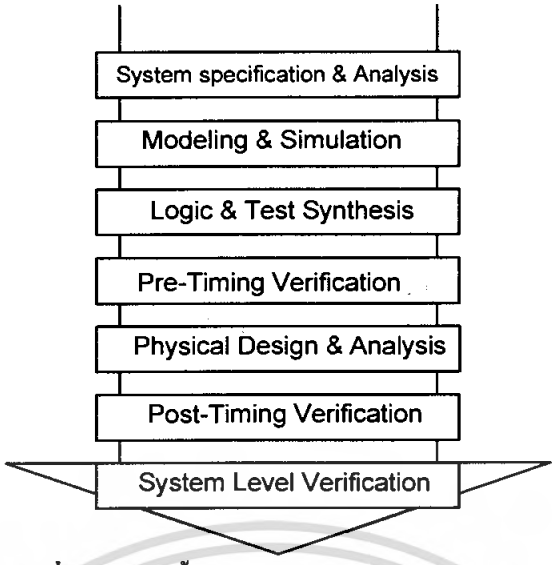
2.9 การเขียนภาษาวีเอชดีแอล

ปัจจุบันในการออกแบบระบบดิจิตอลที่เรารู้จักกันจะเป็นการออกแบบโดยใช้การวาดวงจร (Capture Schematic) โดยใช้โปรแกรมช่วยในการวาด (Schematic entry tools) ซึ่งผู้ออกแบบจะต้องมีทักษะสูงในการออกแบบ และต้องใช้เวลามากในการออกแบบระบบจำลองการทำงาน (Simulation) และตลอดจนถึงการแก้ไขความถูกต้องของระบบ (Debugging) ซึ่งในการออกแบบจะต้องอ้างอิงเทคโนโลยีที่ใช้ออกแบบระบบดิจิตอล (Technology Dependent) ถ้าต้องการเปลี่ยนเทคโนโลยีของระบบที่ออกแบบค่อนข้างทำได้ยากและใช้เวลามาก และเมื่อต้องการออกแบบระบบดิจิตอลที่มีความซับซ้อนสูงยิ่งทำได้ยากหรืออาจทำไม่ได้ โดยใช้กระบวนการออกแบบเก่า ๆ ที่กล่าวมา แต่ในการออกแบบระบบดิจิตอลในปัจจุบันได้มีกระบวนการออกแบบรูปแบบใหม่ที่มีประสิทธิภาพสูง รวดเร็ว และไม่ยึดติดกับเทคโนโลยีที่ใช้ออกแบบ กระบวนการดังกล่าว คือ การออกแบบจากบนลงล่าง (Top-down design) ซึ่งกระบวนการดังกล่าวจะใช้ภาษาบรรยายฮาร์ดแวร์ การออกแบบจำลองการทำงานสังเคราะห์วงจร (Synthesis) ในรูปแบบของเทคโนโลยีที่เราต้องการ และสามารถทดสอบวงจรที่ออกแบบได้บนฮาร์ดแวร์จำพวกชิพเอฟพีจีเอ หรือวงจรรวม ดังนั้นการออกแบบสามารถทำได้ง่ายและมีความสะดวกรวดเร็วยิ่งขึ้น

2.9.1 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมเชิงเลขขนาดใหญ่ที่มีความซับซ้อน ผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อน จากนั้นจึงวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์ เพื่อให้ได้การทำงานตามที่ต้องการ โดยการออกแบบในลักษณะนี้เรียกว่าหลักการออกแบบจากบนลงล่าง ซึ่งถ้าเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาในการออกแบบมากกว่าเพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่าง ๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากในการออกแบบลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอลกับหลักการออกแบบจากบนลงล่างจึงเป็นวิธีการที่เหมาะสมสำหรับการออกแบบและพัฒนาวงจรที่มีความซับซ้อนมากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.7 แสดงให้เห็นถึงขั้นตอนการออกแบบจากบนลงล่างทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย โดยขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียด ดังนี้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษาวีเอชดีแอลสำหรับบรรยาย พฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3. ขั้นตอนการสังเคราะห์ซึ่งจะต้องทำการกำหนดเทคโนโลยีที่จะมารองรับวงจรที่ออกแบบและระบบช่วยออกแบบจะทำการสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท (Gate Level) และ การเชื่อมต่อกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสต์ (Net list) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือเน็ทลิสต์แล้ว ข้อมูลที่ได้นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งจากความจริงที่ว่า อุปกรณ์อิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการเคลื่อนผ่าน (Propagation Delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาที แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชันต่าง ๆ จำนวน 10,000 เกทขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้นจนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาสูง ๆได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของอุปกรณ์เอฟพีจีเอ หรือวงจรรวม

6. หลังจากที่ได้วงจรจริงมาแล้วยังต้องมีความจำเป็นที่จะต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วยเพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่น ๆ ให้เป็นระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะในขั้นตอนนี้วงจรที่ออกแบบจะประกอบด้วยอินพุทและเอาท์พุทแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่น ๆ ให้เป็นระบบแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆอีกครั้ง ซึ่งเป็นการทดสอบการทำงานจริงขั้นสุดท้าย

2.9.2 องค์ประกอบของภาษาวีเอชดีแอล

ส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วย คือ

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

โดยหน่วยการออกแบบที่เป็นพื้นฐานที่ต้องออกแบบเสมอในการออกแบบวงจร นั่นคือ หน่วยการออกแบบเอนทิตี และหน่วยการออกแบบโครงแบบซึ่งทั้งสองหน่วยการออกแบบนี้ จะเรียกว่า คอมโพเนนท์ (Component)

2.9.2.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างอุปกรณ์ภายนอกกับรูปแบบที่เขียนขึ้น รวมทั้งการส่งผ่านค่าพารามิเตอร์บางอย่างระหว่างวงจรกับอุปกรณ์ภายนอกโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่าง ๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น0รูปที่ 2.8 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบเอนทิตี

```
Entity component_name is
    Input and Output ports
    Physical and other parameters
End [component_name];
```

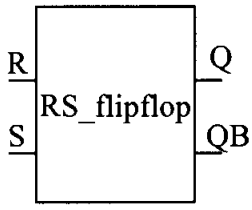
รูปที่ 2.8 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

ส่วนนี้จะขึ้นต้นด้วยคำว่า Entity และ is ระหว่างคำทั้งสองคำเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component name) หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูลรวมทั้งพารามิเตอร์อื่น ๆ และที่สำคัญคือหน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า End และเครื่องหมายอัฒภาคเสมอ (;)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Entity  RS_ff  is
        port(S,R : in bit;
             Q,QB : out bit);
End RS_ff;
```

(a) หน่วยการออกแบบเอนทิตี ในรูปของภาษาวีเอชดีแอล



(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

รูปที่ 2.9 แสดงรูปแบบของอาร์เอสฟลิปฟลอป

ในรูปที่ 2.9 เป็นหน่วยการออกแบบเอนทิตีที่บรรยายอุปกรณ์ชื่ออาร์เอสฟลิปฟลอปในส่วนหัวของเอนทิตีมีการกำหนดจุดต่อ 4 จุด ภายใต้ชุดคำสั่ง port โดยที่ 2 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้าได้แก่ R,S ซึ่งกำหนดด้วยทิศทางการติดต่อกับโลกภายนอกเป็นการไหลเข้าของข้อมูล ส่วนจุดเอาท์พุตเป็นจุดให้ข้อมูลไหลออก ได้แก่ Q,QB ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก ส่วนประเภทของข้อมูลที่จะไหลเข้าและออกนั้นเป็นประเภทบิตที่สามารถมีค่าได้เพียงสองค่าเท่านั้นคือ “0” และ “1” เท่านั้น

ประเภทของพอร์ตที่สามารถประกาศใช้ในเอนทิตี มี 4 ประเภทดังนี้

- พอร์ตอินพุตเข้า

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากอุปกรณ์ภายนอกเข้ามาภายในวงจร สามารถนำมาป้อนให้กับสัญญาณอื่นหรืออ่านค่าได้แต่ไม่สามารถถูกเขียนจากภายในวงจรได้

- พอร์ตอินพุตออก

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากวงจร ส่งออกไปยังอุปกรณ์ภายนอก สามารถเขียนจากภายในวงจรได้ แต่ไม่สามารถอ่านจากภายในวงจรได้

- พอร์ตอินพุตและเอาต์พุต (Bidirectional)

เป็นพอร์ต 2 ทิศทาง ที่สามารถส่งถูกเขียนและอ่านได้จากภายในวงจร

- พอร์ตบัฟเฟอร์ (Output with Internal Feedback)

เป็นพอร์ตเอาต์พุตประเภทหนึ่งที่สามารถอ่านค่ากลับ (Feedback) เข้ามาภายในวงจรได้

2.9.2.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรมคือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงาน พฤติกรรมต่าง ๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ต่าง ๆ ที่กำหนดในหน่วยการออกแบบเอนทิตี รูปที่ 2.10 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบสถาปัตยกรรม

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 2.10 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่าหน่วยการออกแบบสถาปัตยกรรมนั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใด ๆ (of <entity design unit> is) ส่วนที่อยู่ระหว่าง Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเผื่อเลือกในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่าง ๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น สัญญาณ ค่าคงที่ โปรแกรมย่อย และอุปกรณ์ ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่งพอร์ต) นั้น จะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (Concurrent statement) เท่านั้น คือทุก ๆ สถานะจะทำงานพร้อมกัน ลำดับก่อนหลังจะไม่มีผลต่อการทำงานของรูปแบบ หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง End และชื่อของสถาปัตยกรรมนั้น ๆ โดยทั่วไปการเขียนรูปแบบโมเดล (Modeling styles) ระบบเชิงเลขด้วยภาษาวีเอชดีแอลสามารถเขียนได้ในลักษณะต่าง ๆ ดังนี้

- ลักษณะการไหลของข้อมูล (Dataflow style) [RTL descriptions]
- ลักษณะพฤติกรรม (Behavioral style) [Algorithm descriptions]
- ลักษณะโครงสร้าง (Structural style) [Netlist description]
- ลักษณะผสม (Mixed Model style)

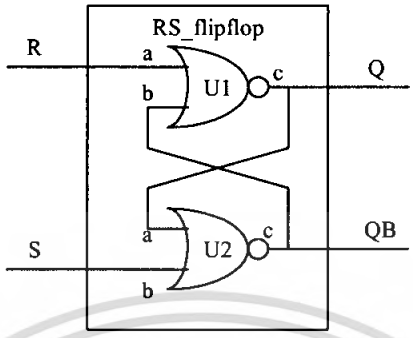
```
Architecture dataflow of RS_ff is
    Begin
        . Q <= not(QB or R);
          QB <= not(Q or S);
End dataflow;
```

รูปที่ 2.11 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปตามฟังก์ชันบูลีน

$Q = \overline{QB + R}$ และ $QB = \overline{Q + S}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (R,S) กับข้อมูลที่ไหลออก (Q,QB) ประกอบด้วยชุดคำสั่งแบบแข่งขนาน 2 ชุด ซึ่งเขียนเป็นประเภทการไหลของข้อมูล หรือเรียกว่าระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL : Register Transfer Level)



รูปที่ 2.12 แสดงโครงสร้างภายในสถาปัตยกรรมของอาร์เอสฟลิปฟลอป

รูปที่ 2.12 เป็นหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะโครงสร้าง ซึ่งเปรียบเสมือนการนำอุปกรณ์ที่มีอยู่ในไลบรารี่มาต่อเป็นวงจรตามต้องการโดยใช้นอร์เกท 2 อินพุท (nor2) จำนวนสองตัวมาสร้างตามฟังก์ชันบูลีน

```
Architecture struc of RS_ff is
        component nor2
                port(a,b : in bit;
                        c :out bit);
        end component;

Begin
        U1 : nor2 port map(R,QB,Q);
        U2 : nor2 port map(S,Q,QB);
End struc;
```

รูปที่ 2.13 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะโครงสร้าง

83299
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Architecture behave of RS_ff is
        Begin
                process(R,S)
                        begin
                                if R='0' and S='1' then
                                        Q  <= '1';
                                        QB <= '0';
                                elsif R='1' and S='0' then
                                        Q  <= '1';
                                        QB <= '0';
                                end if;
                end process;
End behave;
```

รูปที่ 2.14 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะพฤติกรรม

รูปที่ 2.14 เป็นการเขียนบรรยายการทำงานของรูปแบบในลักษณะพฤติกรรม ซึ่งจะเห็นได้ว่ามีลักษณะที่เหมือนกับการเขียนโปรแกรมทั่วไป โดยจะต้องมีการใช้งานส่วนที่เรียกว่า process และการทำงานของรูปแบบจะขึ้นอยู่กับการเปลี่ยนแปลงของสิ่งที่อยู่ภายใน process (อินพุท R,S) ซึ่งเรียกว่า เซนซิติวิตีลิสต์ (Sensitivity list) การเขียนในลักษณะนี้ลำดับก่อนหลังของชุดคำสั่งจะมีผลต่อการทำงานของรูปแบบที่เขียนขึ้น

```
Architecture mixed of RS_ff is
        component nor2
                port(a,b : in bit;
                     c : out bit);
        end component;

    Begin
        U1 : nor2 port map(R,QB,Q);
        QB <= not(Q or S);
End mixed;
```

รูปที่ 2.15 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าจะเขียนบรรยายส่วนของสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะของพฤติกรรม การไหลของข้อมูลโครงสร้างหรือผสมที่นำเอาแต่ละลักษณะมาเขียนไว้ในส่วนของสถาปัตยกรรมก็ตาม ต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งถือว่าเป็นข้อดีของภาษาวีเอชดีแอล

การออกแบบวงจรดิจิตอลใด ๆ จะต้องออกแบบหน่วยการออกแบบพื้นฐานก็คือ ส่วนของเอนทิตี และส่วนของหน่วยของสถาปัตยกรรมที่มีความสัมพันธ์กัน ซึ่งมีด้วยกันหลายรูปแบบสำหรับการออกแบบวงจรเดียวกัน

2.9.2.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่าง ๆ ตลอดจนโปรแกรมย่อย เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบเชิงเลข สามารถเก็บไว้ในส่วนของแพ็กเก็ตได้และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดยหน่วยการออกแบบเอนทิตีหน่วยการออกแบบสถาปัตยกรรมหรือจากหน่วยการออกแบบแพ็กเก็ตอื่น ๆ โดยปกติแล้วแพ็กเก็ตจะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็กเก็ต (Package Declaration) และส่วนของบอดี้แพ็กเก็ต (Package Body) เนื่องจากแพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่องโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอลสามารถกระทำได้ด้วยชุดคำสั่ง USE โดยสิ่งที่สามารถสร้างไว้ใน Package ได้แก่

- Subprogram
- Types
- Constants
- Signals
- Aliases
- Attributes
- Component
- Disconnection Specification

ส่วนการประกาศแพ็กเก็ต

เป็นส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็กเก็ตสำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเองสิ่งใด ๆ ที่ถูกประกาศไว้ในส่วนของบอดี้แพ็กเก็ตแต่ไม่ได้ถูกประกาศไว้ในส่วนการประกาศแพ็กเก็ตจะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือจุดเชื่อมต่อหรือพอร์ทที่มีหน้าที่ติดต่อกับโลกภายนอกฉะนั้นโดยทั่วไปแล้วแพ็กเก็ตสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ (Package body) และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของข้อมูล (Type) หรือสัญญาณ (Signal) ในทางกลับกันกับส่วนบอดี้แพ็กเก็ตที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
Package package_name is
   Package_declaration_part
End package_name;
```

รูปที่ 2.16 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

บอดี้แพ็กเก็ต

โครงสร้างที่ประกอบด้วยคำสั่งต่าง ๆ ในรูปของคำสั่งลำดับ (Sequential statement) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลายที่ชื่อของโปรแกรมย่อยนั้น ๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ตแล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ตทั้งนี้รวมทั้งการกำหนดค่าคงที่ต่าง ๆ อันได้แก่ ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ตแต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต ฉะนั้นส่วนบอดี้แพ็กเก็ตจึงไม่จำเป็นต้องมีถ้าในส่วนของการประกาศแพ็กเก็ตไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.17

```
Package body package_name is
   declarative part
End package_name;
```

รูปที่ 2.17 แสดงโครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต

2.9.2.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไรจะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งหน่วยเท่านั้นแต่ในขณะที่หน่วยการออกแบบเอนทิตีหนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

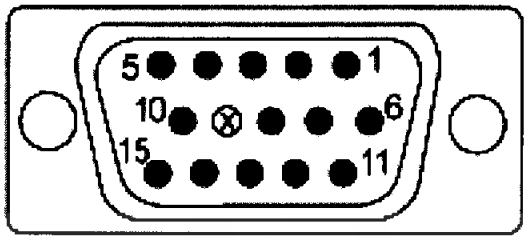
```
Configuration identifier of entity_name is
      Configuration_declarative_part
End;
```

รูปที่ 2.18 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 หัวเชื่อมต่อกับหน้าจอวีจีเอ (VGA Connector : DE-15)

พอร์ตวีดีโอ (Video Port) ซึ่งเป็นพอร์ตสำหรับต่อสายสัญญาณภาพกับจอคอมพิวเตอร์ ลักษณะของพอร์ตจะเป็นพอร์ตแบบตัวเมีย มีรู 15 รู ดังรูปที่ 2.19



Female

รูปที่ 2.19 หัวเชื่อมต่อกับหน้าจอวีจีเอ

ตารางที่ 2.1 คุณสมบัติต่าง ๆ ของหัวเชื่อมต่อหน้าจอวีจีเอ

Pin Number	Signal Name	Notes
1	RED Video	Analog Signal, usually 0.7 volt, peak-to-peak
2	GREEN Video	Analog Signal, usually 0.7 volt, peak-to-peak
3	BLUE Video	Analog Signal, usually 0.7 volt, peak-to-peak
4	ID2	Monitor ID Bit 2
5	Digital Ground	General "Ground" for the video system
6	RED Ground	Note that the video signals have a separate Ground path. This is needed anytime you get the pixel rate much over 20 or 30 MHz... which is just about aways.
7	GREEN Ground	
8	BLUE Ground	
9	KEY	(Not used)
10	SYNC Ground	TTL returns for the SYNC lines.
11	ID0	Monitor ID Bit 0
12	ID1 or SDA	Monitor ID Bit 1
13	Horizontal Sync	Digital level (0 to 5 volts, TTL output)
14	Vertical Sync	Digital level (0 to 5 volts, TTL output)
15	ID3 or SCL	Monitor ID Bit 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 รอม (ROM: Read Only Memory)

รอม คือ หน่วยความจำชนิดหนึ่งที่มีโปรแกรมหรือข้อมูลอยู่แล้ว และพร้อมที่จะนำมาต่อกับไมโครโปรเซสเซอร์ได้โดยตรง ซึ่งโปรแกรมหรือข้อมูลนี้จะไม่สูญหายไป แม้ว่าจะไม่มีการจ่ายไฟเลี้ยงให้กับระบบ ข้อมูลที่เก็บอยู่ในรอมจะสามารถอ่านค่าออกมาได้ แต่ไม่สามารถเขียนข้อมูลเข้าไปได้ เว้นแต่จะใช้วิธีการพิเศษซึ่งขึ้นกับชนิดของรอม ซึ่งชนิดของรอมสามารถแบ่งออกได้เป็น

2.11.1 แมนนวลรอม (Manual ROM)

ข้อมูลทั้งหมดที่อยู่ในรอมจะถูกโปรแกรมโดยผู้ผลิต (โปรแกรม มาจากโรงงาน) เราจะใช้รอมชนิดนี้เมื่อข้อมูลนั้นไม่มีการเปลี่ยนแปลง และมีความต้องการใช้งานเป็นจำนวนมาก ผู้ใช้ไม่สามารถเปลี่ยนแปลงข้อมูลภายในรอมได้

โดยรอมจะมีการใช้เทคโนโลยีที่แตกต่างกัน เช่น ไบโอโพลาร์ (Biopolar) ซีมอส (CMOS) เอ็นมอส (NMOS) และพีมอส (PMOS)

2.11.2 พรอม (PROM : Programmable Read-Only Memory)

ข้อมูลที่ต้องการโปรแกรมจะถูกโปรแกรมโดยผู้ใช้เอง โดยป้อนพัลซ์แรงดันสูง (High Voltage Pulsed) ทำให้แผ่นโลหะหรือโพลีคริสตอลซิลิกอน (Polycrystaline Silicon) ที่อยู่ในตัวไอซีขาดออกจากกัน ทำให้เกิดเป็นลอจิก '1' หรือ '0' ตามตำแหน่งที่กำหนดในหน่วยความจำนั้น ๆ เมื่อพีรอมถูกโปรแกรมแล้ว ข้อมูลภายในจะไม่สามารถเปลี่ยนแปลงได้อีก หน่วยความจำชนิดนี้จะใช้ในงานที่ใช้ความเร็วสูง ซึ่งสูงกว่าหน่วยความจำที่โปรแกรมได้กว่าหน่วยความจำชนิดอื่น ๆ

2.11.3 อีพรอม (EPROM : Erasable Programmable Read-Only Memory)

ข้อมูลจะถูกโปรแกรมโดยผู้ใช้ โดยการให้สัญญาณที่มีแรงดันสูงผ่านเข้าไปในตัวอีพรอม ซึ่งเป็นวิธีเดียวกันกับที่ใช้ในพรอม แต่ข้อมูลที่อยู่ในอีพรอมเปลี่ยนแปลงได้ โดยการลบข้อมูลที่อยู่ในอีพรอมออกก่อน แล้วค่อยโปรแกรมเข้าไปใหม่ การลบข้อมูลนี้ทำได้ด้วยการฉายแสงอุลตร้าไวโอเลตเข้าไปในตัวไอซีโดยผ่านทางกระจกใสที่อยู่บนตัวไอซี เมื่อฉายแสงครู่หนึ่ง (ประมาณ 5-10 นาที) ข้อมูลที่อยู่ภายในก็จะถูกลบทิ้ง ซึ่งช่วงเวลาฉายแสงนี้สามารถดูได้จากดาต้าชีทมากับตัวอีพรอมและมีความเหมาะสมที่จะใช้ เมื่องานของระบบมีโอกาสที่จะปรับปรุงแก้ไขข้อมูลใหม่

2.11.4 อีเอพรอม (EAPROM :Electrically Alterable Read-Only Memory)

อีเอพรอมหรือเรียกอีกชื่อหนึ่งว่าอีอีพรอม (EEPROM :Electrical Erasable EPROM) เนื่องจากมีการใช้ไฟฟ้าในการลบข้อมูลในรอมเพื่อเขียนใหม่ ซึ่งใช้เวลาสั้นกว่าของอีพรอม

การลบขึ้นอยู่กับพื้นฐานการใช้เทคโนโลยีที่แตกต่างกัน ดังนั้นอีเอพรอมจะอยู่บนพื้นฐานของเทคโนโลยีแบบเอ็นมอส ข้อมูลจะถูกโปรแกรมโดยผู้ใช้เหมือนในอีพรอมแต่สิ่งที่แตกต่างก็คือ ข้อมูลของอีเอพรอมสามารถลบได้โดยทางไฟฟ้าไม่ใช่โดยการฉายแสงแบบอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพเป็นการเปลี่ยนภาพต่าง ๆ ที่ได้จากกล้องหรือการสแกน ตลอดจนข้อมูลภาพในรูปดิจิตอล แล้วนำไปผ่านกระบวนการบางอย่างเพื่อให้เกิดเป็นภาพใหม่ เช่น ภาพที่มีความคมชัดสดใส หรือภาพที่เบลอ เป็นต้น

ซึ่งในโครงงานนี้ เราจะอาศัยหลักของการแปลงปาสคาลเต็มหน่วยมาประยุกต์สร้างเป็นวงจรกรองปาสคาลแล้วนำมาทำการประมวลผลภาพโดยเราตั้งสมมติฐานไว้ว่าผลลัพธ์ที่ได้ออกมานั้นจะมีคุณสมบัติเป็นไปในลักษณะของตัวกรองความถี่สูงผ่าน (High pass filter) และอาจจะสามารถนำไปประยุกต์ใช้ในการตรวจจับขอบภาพ (edge detection) ได้อีกด้วย

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

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

$$(x+y)^2 = x^2 + 2xy + y^2 \tag{2.1}$$

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

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k \tag{2.2}$$

$$= x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \ldots + y^n \tag{2.3}$$

$$= a_0 x^n + a_1 x^{n-1} y + a_2 x^{n-2} y^2 + \ldots + a_{n-1} x y^{n-1} + a_n y^n \tag{2.4}$$

ซึ่งการกระจายเทอมในสมการ (2.2) - (2.4) บางครั้งเรียกได้ว่าเป็นการกระจายเทอมไบโนเมียล (Binomial Expansion) โดย a_k เรียกว่าสัมประสิทธิ์ไบโนเมียล (Binomial coefficient) ซึ่งจะมีค่าดังสมการ

$$a_k = \binom{n}{k} \tag{2.5}$$

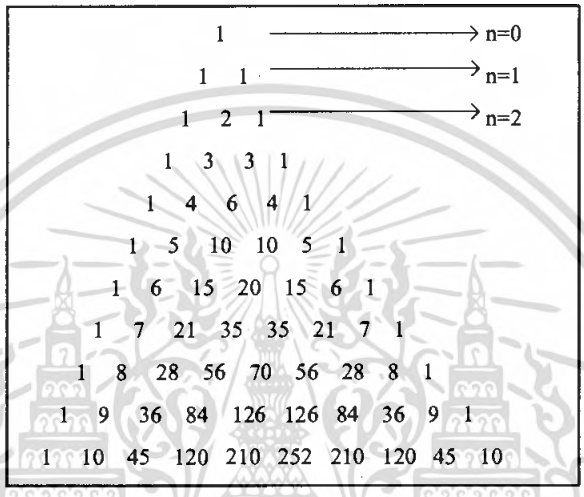
โดยความหมายของสัมประสิทธิ์ไบโนเมียลหรืออาจเรียกได้ว่าเป็นคอมไบเนทอเรียลนัมเบอร์ (Combinatorial number) คือจำนวนวิธีการของการหยิบของ k สิ่งออกจากผลทั้งหมดที่เป็นไปได้ n สิ่ง หรือที่รู้จักกันว่า การจัดหมู่ (Combination number) ดังที่ได้พบเห็นในวิชาความน่าจะเป็นและสถิติ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

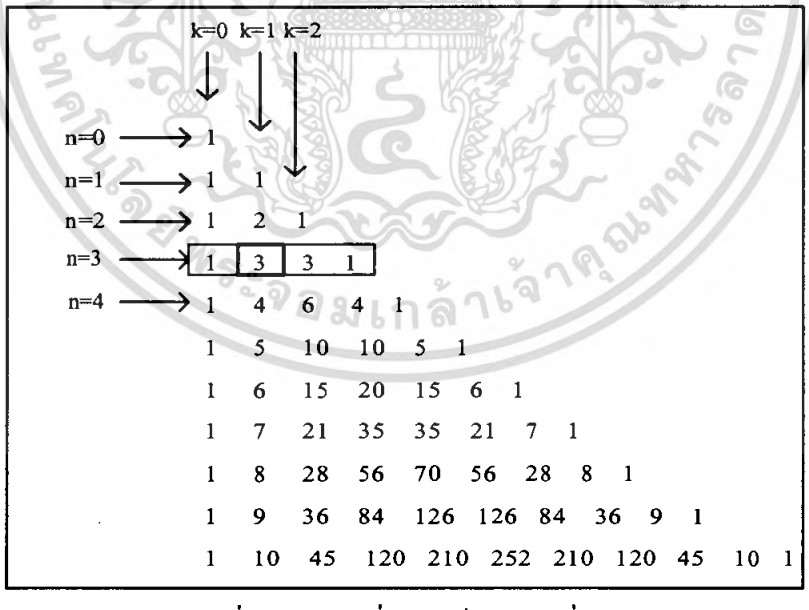
$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!} = a_k \tag{2.6}$$

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

สำหรับค่าของสัมประสิทธิ์ไบโนเมียลนั้นสามารถนำมาจัดเรียงเป็นรูปฟอร์มที่เป็นสามเหลี่ยมที่รู้จักกันในชื่อว่า สามเหลี่ยมปาสคาล ได้ดังนี้



(ก) สามเหลี่ยมปาสคาล



(ข) สามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

รูปที่ 2.20 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

แต่ละแถว (n) ของสามเหลี่ยมปาสคาลสามารถสร้างได้ด้วยวิธีอื่นที่ไม่ใช่จากสมการที่ (2.4) โดยการบวกของ 2 องค์ประกอบที่อยู่ในแบบทแยงขึ้นไปในแถวข้างบน เช่น ในแถวที่ 4 (n=3) องค์ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวที่ 3 (ที่ n=3, k=1) มาจาก 1+2 ในแถวบน (n=2) และ 3 ตัวถัดไป (ที่ n=3, k=2) มาจาก 2+1 เป็นต้น ซึ่งสามารถพิสูจน์คุณสมบัติ ดังกล่าว (การบวกเพื่อสร้างองค์ประกอบในแถวต่อไป) ได้ดังนี้

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)!}{k!(n-k)!} \tag{2.7}$$

$$= \frac{(n-k)(n-1)!}{k!(n-k)!} + \frac{k(n-1)!}{k!(n-k)!} \tag{2.8}$$

$$= \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} \tag{2.9}$$

$$= \binom{n-1}{k} + \binom{n-1}{k-1} \tag{2.10}$$

โดยสมาชิกในสามเหลี่ยมปาสคาลในแถวที่ n ใด ๆ สามารถคำนวณได้ด้วยการหาสัมประสิทธิ์ไบโนเมียล ดังในสมการที่ (2.7) ถึง (2.10) สำหรับ k=1, 2, 3, ..., n ได้

ดังนั้นเมื่อเราจะทำการกระจายเทอมของ $(x+y)^n$ ใด ๆ เราจะสามารถหาค่าของสัมประสิทธิ์ไบโนเมียลจากสมการที่ (2.4) ได้โดยง่ายจากสามเหลี่ยมปาสคาลนั่นเอง เช่น $(x+y)^4$

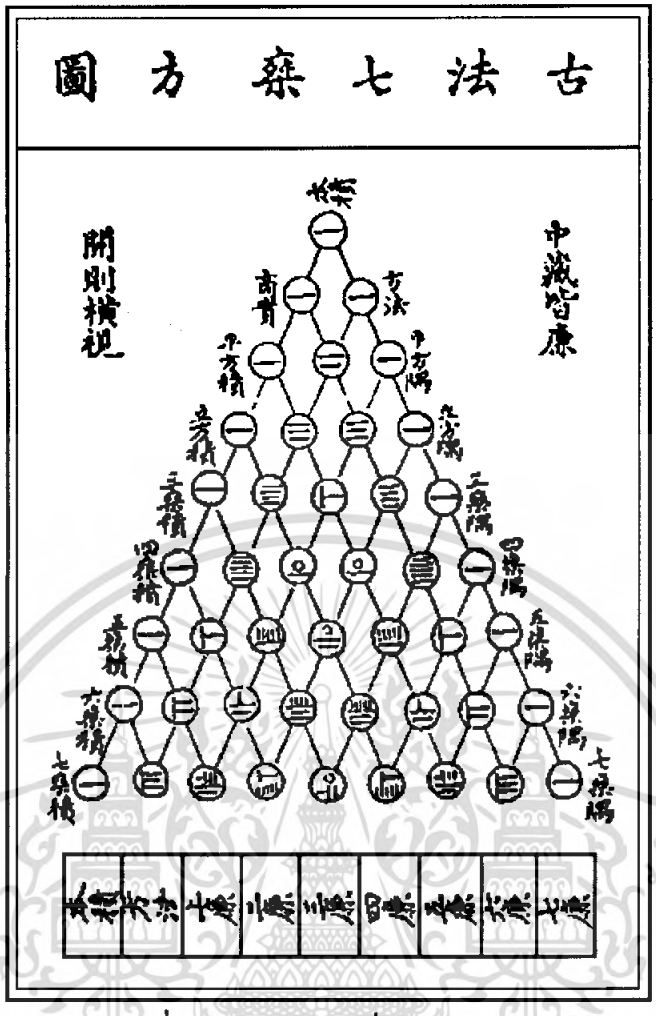
$$(x+y)^4 = \underline{1}x^4 + \underline{4}x^3y + \underline{6}x^2y^2 + \underline{4}xy^3 + \underline{1}y^4 \tag{2.11}$$

โดยสัมประสิทธิ์ในที่นี่มาจากสามเหลี่ยมปาสคาลในรูปที่ 2.20 ในแถวที่ 5 (n=4)

2.13.1 ที่มาของชื่อสามเหลี่ยมปาสคาล

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงสามเหลี่ยมของ ยาง ฮุย

2.13.2 การแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform)

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

$$X = P\,x \tag{2.12}$$

โดย X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุท

P = เมตริกซ์ของการแปลงปาสคาล (Pascal Transform Matrix)

โดยการสร้างเมตริกซ์ของการแปลงปาสคาลนั้นสามารถสร้างได้ดังนี้

$$P = \left[P_{ij}\right] = (-1)^j \binom{i}{j} = \frac{(-1)^j}{j!}\, i^{\underline{j}};\ i, j = 0, 1, 2.....N\text{-}1 \tag{2.13}$$

โดย N คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$i^{\underline{j}} = i(i-1)(i-2)\ldots(i-j+1) \quad ; \; j \geq 1 \text{ and } i^{\underline{0}} = 1 \tag{2.14}$$

และสมการที่ (2.13) นี้เองถือว่าเป็นฟังก์ชันพื้นฐาน (Basis function) ของการแปลงดีสครีตปาสคาลทรานสฟอร์มนี้ ซึ่งผลที่ได้จากการสร้างเมตริกซ์ของการแปลงปาสคาล ดังในสมการที่ (2.13) ซึ่งจะเหมือนกับการนำสมาชิกในสามเหลี่ยมปาสคาล ในรูปที่ 2.20 (ข) มาจัดวางเป็นเมตริกซ์และทำการสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับ

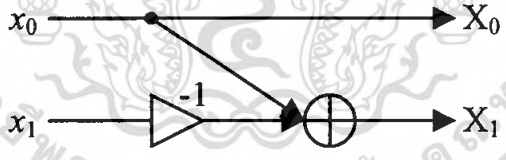
2.13.3 การสร้างเมตริกซ์การแปลงปาสคาลขนาดต่าง ๆ

เมื่อ $N = 2$ จะได้

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

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

สามารถสร้างบัตเตอร์ฟลายยูนิต (butterfly unit) อย่างง่ายได้ดังนี้



รูปที่ 2.22 แสดงการสร้างบัตเตอร์ฟลายยูนิตสำหรับการแปลงปาสคาลเต็มหน่วย

เมื่อ $N = 3$ จะได้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างบัตเตอร์ฟลายยูนิตสำหรับ N=3 นี้ แน่นอนว่าต้องมีส่วนประกอบของบัตเตอร์ฟลายยูนิต ดังในกรณีที่ N=2 แต่ในทางปฏิบัติแล้ว เราใช้เมตริกซ์ที่มีองค์ประกอบในลักษณะเป็นไบนารีเมตริกซ์ (1,0,-1) ดังที่เห็นในกรณี N=2 ซึ่งเมตริกซ์ P คือ เมตริกซ์ไบนารี (1,0,-1) เราจะสามารถสร้างบัตเตอร์ฟลายยูนิตจากไบนารีเมตริกซ์ได้โดยง่าย ทั้งนี้เราสามารถจัดปาสคาลเมตริกซ์ต่าง ๆ ให้อยู่ในรูปไบนารีเมตริกซ์ (1,0,-1) ได้โดยใช้ผลคูณของไบนารีเมตริกซ์ ดังสมการต่อไปนี้

$$\mathbf{P} = \prod_{k=N-1}^{1} \left[q_{ij} \right]_k \qquad ; i,j = 0, 1, 2, \ldots, N\text{-}1 \tag{2.15}$$

โดยเราเรียก $[q_{i\,j}]_k$ ว่าเมตริกซ์ $[q_{i\,j}]$ สเตจที่ k^{th}

2.13.4 วิธีการแยกองค์ประกอบของปาสคาลเมตริกซ์

1. พิจารณาแนวเส้นทแยงมุมหลัก

$$\mathrm{P} = \left[q_{ij} \right]_{N-1} \left[q_{ij} \right]_{N-2} \cdots \left[q_{ij} \right]_1 \tag{2.16}$$

เรียก $[q_{ij}]_k$ ว่าเมตริกซ์ $[q_{ij}]$ สเตจที่ k^{th}

<u>สเตจที่ $k = (N\text{-}1)$</u>

$$Diag([q_{ij}]_{N-1}) = [1\ 1\ 1\ \ldots\ 1\ {-1}] \tag{2.17}$$

โดยที่จะมี '1' เท่ากับจำนวน N-1

เช่น

<u>สเตจที่ $k = 2$</u>

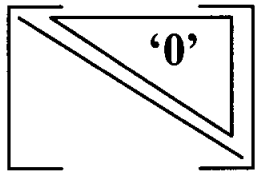
$$\mathrm{Diag}\,([q_{ij}]_2) = [1\ 1\ {-1}\ {-1}\ldots\ {-1}] \tag{2.18}$$

<u>สเตจที่ $k = 1$</u>

$$\mathrm{Diag}([q_{ij}]_1) = [1\ {-1}\ {-1}\ {-1}\ldots\ {-1}] \tag{2.19}$$

2. ทุกองค์ประกอบเหนือเส้นทแยงมุมหลัก (Upper Triangle) หรือส่วนบนของสามเหลี่ยมมีค่าเท่ากับ '0'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงตำแหน่งองค์ประกอบ '0' ในส่วนของสามเหลี่ยมบนของแต่ละเมตริกซ์ $\left[q_{ij} \right]_k$

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

4. นอกเหนือจากนี้ให้มีค่าเท่ากับ '0'

ตัวอย่าง N = 3 จะได้

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

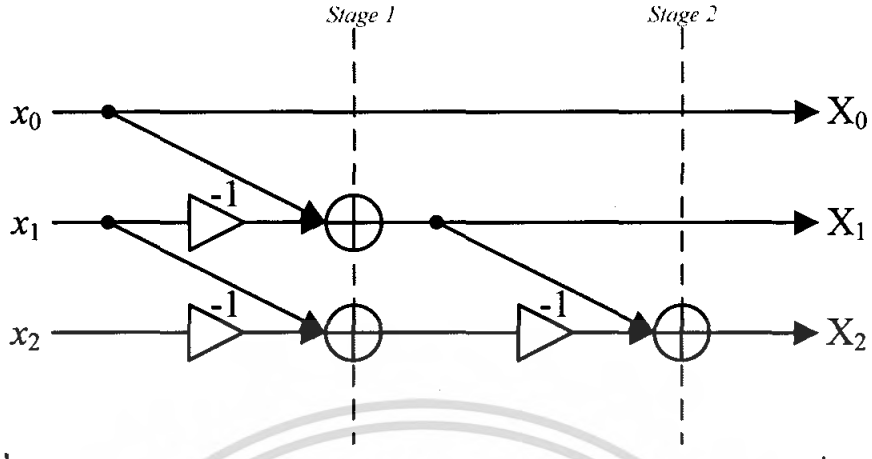
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \boxed{\begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 - x_1 \end{aligned}} \quad X_2 = x_0 - 2x_1 + x_2$$

สามารถจัดรูปองค์ประกอบได้ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถออกแบบโดยใช้บัตเตอร์ฟลายยูนิต ได้ดังนี้



รูปที่ 2.24 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ $N = 3$

ตัวอย่าง N = 4 จะได้

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

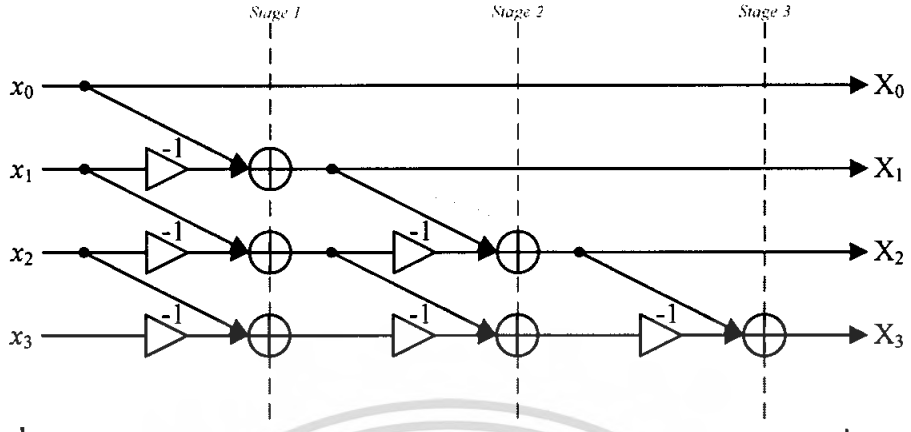
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \begin{aligned} X_0 &= x_0 \\ X_1 &= x_0 - x_1 \\ X_2 &= x_0 - 2x_1 + x_2 \\ X_3 &= x_0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

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

Stage 3 *Stage 2* *Stage 1*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถออกแบบโดยใช้บัตเตอร์ฟลายยูนิต ได้ดังนี้



รูปที่ 2.25 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงฟาสคาลเต็มหน่วย เมื่อ $N = 4$

ตัวอย่าง $N = 5$ จะได้

$$\mathbf{P} = \begin{bmatrix} 1 & & & & \\ 1 & -1 & & & \\ 1 & -2 & 1 & & \\ 1 & -3 & 3 & -1 & \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ 1 & -1 & & & \\ 1 & -2 & 1 & & \\ 1 & -3 & 3 & -1 & \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

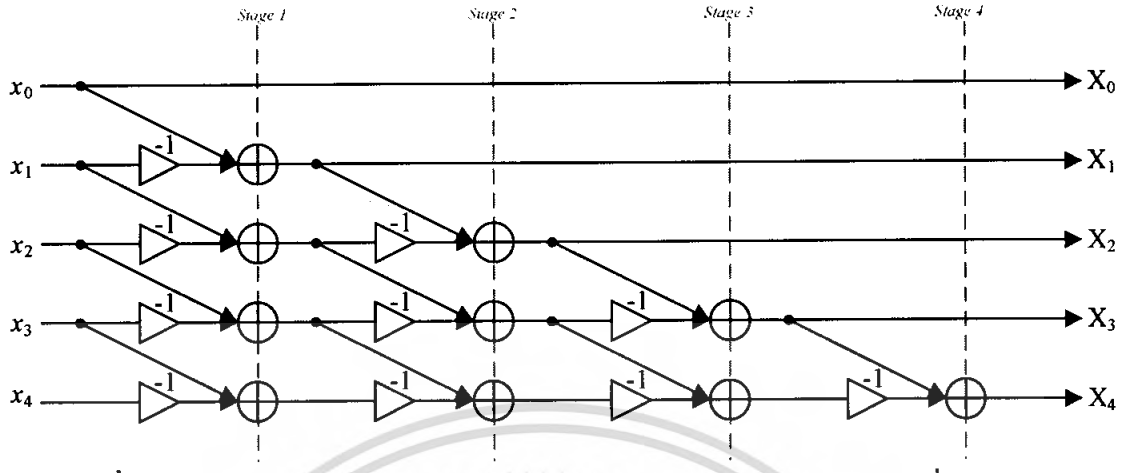
$$X_0 = x_0$$
$$X_1 = x_0 - x_1$$
$$\Rightarrow X_2 = x_0 - 2x_1 + x_2$$
$$X_3 = x_0 - 3x_1 + 3x_2 - x_3$$
$$X_4 = x_0 - 4x_1 + 6x_2 - 4x_3 + x_4$$

$$= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & -1 & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Stage 4 *Stage 3* *Stage 2* *Stage 1*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถออกแบบโดยใช้บัตเตอร์ฟลายยูนิต ได้ดังนี้



รูปที่ 2.26 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ $N = 5$

คุณสมบัติที่ดีประการหนึ่งของการแปลงปาสคาลเต็มหน่วยนี้คือ

$$P^{-1} = P \tag{2.20}$$

ดังนั้นทำให้การแปลงกลับดีสครีตปาสคาลทรานสฟอร์ม (Inverse DPT) สามารถทำได้เหมือนทำการแปลงฟอร์เวิร์ดดีสครีตปาสคาทรานสฟอร์ม (Forward DPT) ทุกประการ ถ้ามองในลักษณะของการออกแบบฮาร์ดแวร์นั้นจะเห็นได้ว่าสามารถใช้วงจรเดิมที่ใช้ตอนแปลงมาสำหรับการแปลงกลับได้เช่นเดียวกัน ไม่ต้องทำการปรับเปลี่ยนวงจรแต่อย่างใด

$$x = P^{-1}X = PX \tag{2.21}$$

2.13.5 การแปลงปาสคาลเต็มหน่วยในสองมิติ (2-D Discrete Pascal Transform)

สำหรับการแปลงปาสคาลเต็มหน่วยแบบ 2 มิติ (2-D DPT) สามารถทำการแปลงได้ดังนี้

$$X = PxP^T \tag{2.22}$$

เมื่อ X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุท

P = เมตรกซ์ของการแปลงปาสคาล

และกรณีแปลงกลับ (Inverse Transforms) สามารถแสดงได้ดังสมการที่ (2.23)

$$x = PXP^T \tag{2.23}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อความง่ายและสะดวกในการอธิบายหลักการแปลงปาสคาลเต็มหน่วยในสองมิติ ให้พิจารณาตัวอย่าง เมื่อ : $N = 3$

$$X = PxP^T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

สมมติให้เมตริกซ์นี้ซึ่งผ่านการแปลงปาสคาลเต็มหน่วยในหนึ่งมิติ ในแนวหลัก (column operation) มาเป็น $\begin{bmatrix} y_{a,b} \end{bmatrix}$

ดังนั้นจะได้ว่า

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

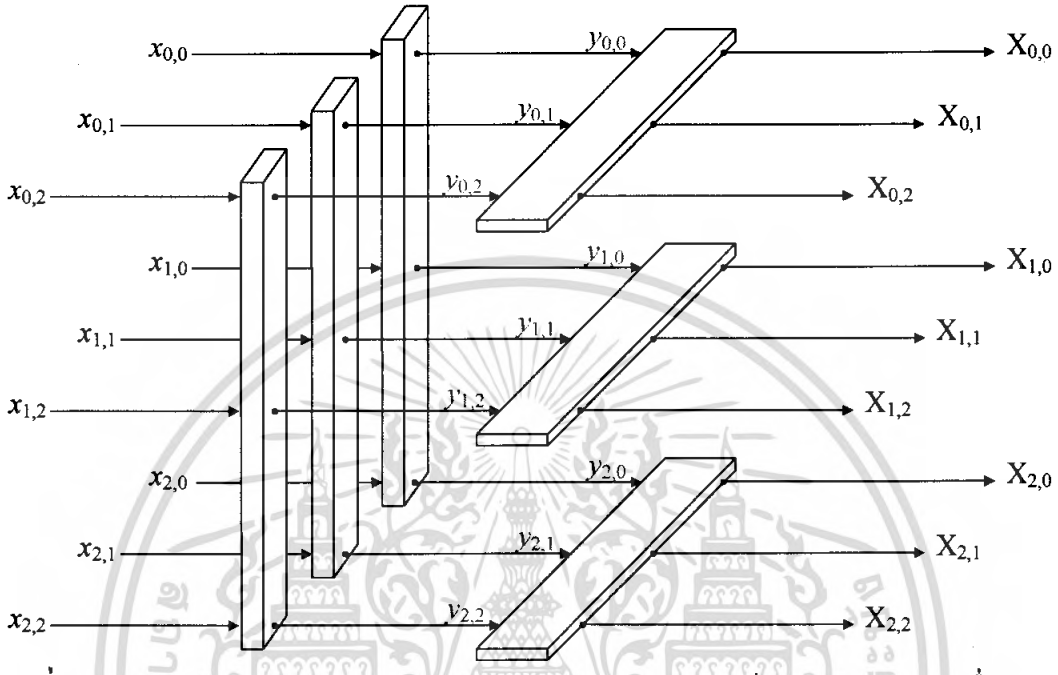
$$= \begin{bmatrix} y_{0,0} & y_{0,0} - y_{0,1} & y_{0,0} - 2y_{0,1} + y_{0,2} \\ y_{1,0} & y_{1,0} - y_{1,1} & y_{1,0} - 2y_{1,1} + y_{1,2} \\ y_{2,0} & y_{2,0} - y_{2,1} & y_{2,0} - 2y_{2,1} + y_{2,2} \end{bmatrix}$$

เมตริกซ์นี้คือการดำเนินการในส่วนของแถว (row operation) สำหรับอินพุทที่ผ่านการดำเนินการในส่วนของคอลัมน์มาแล้ว ผลลัพธ์ทั้งหมดคือ การดำเนินการแปลงดีสครีตปาสคาลทรานสฟอร์มในสองมิติ ในที่สุดจะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

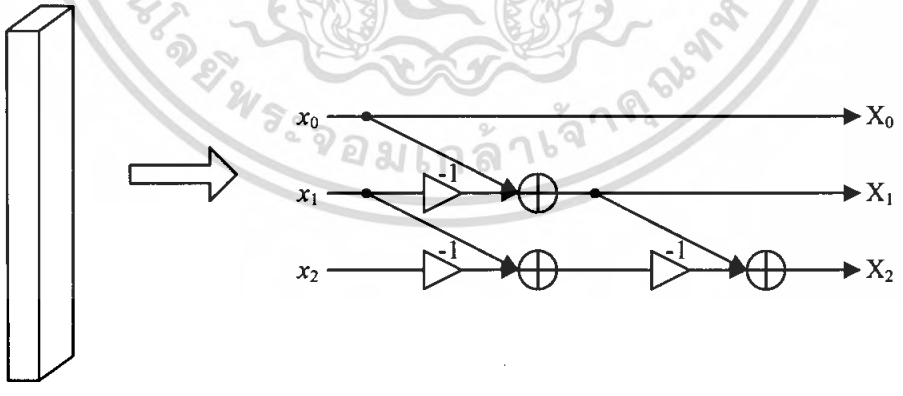
$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & x_{0,0}-x_{0,1} & x_{0,0}-2x_{0,1}+x_{0,2} \\ [x_{0,0}-x_{1,0}] & [x_{0,0}-x_{1,0}]-[x_{0,1}-x_{1,1}] & [x_{0,0}-x_{1,0}]-2[x_{0,1}-x_{1,1}]+[x_{0,2}-x_{1,2}]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

เราสามารถสรุปเป็นไดอะแกรม ได้ดังนี้



รูปที่ 2.27 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วยแบบ 2 มิติ เมื่อ N=3

โดยสัญญาณที่ใช้ในรูปที่ 2.27 มีความหมายดังในรูปที่ 2.28



รูปที่ 2.28 แสดงโครงสร้างแบบวงจรภายในที่ใช้ในสัญลักษณ์ดามรูปที่ 2.27 เมื่อ N=3

2.13.6 การตรวจสอบหาคุณลักษณะทางความถี่ (คุณสมบัติการกรองสัญญาณ) ของการแปลงปาสคาลเต็มหน่วย

การกรองความถี่สูงผ่านอาจจะเป็นลักษณะเฉพาะที่สำคัญของการแปลงปาสคาลเต็มหน่วยซึ่งเป็นสมมติฐานที่เราตั้งขึ้น โดยต้องทดสอบและออกแบบการทดสอบนั้น ซึ่งแนวคิดนี้มาจาก [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาแล้ว ต่อไปจะเป็นการพิสูจน์การกรองความถี่สูงผ่านของปาสคาลเมตริกซ์

<u>ปาสคาลทรานสเฟอร์ฟังก์ชันในหนึ่งมิติ</u>

ตัวอย่างเช่น : เมตริกซ์ขนาด 3 x 3

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

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้โดยการเปลี่ยนตัวแปรเป็น

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

ดังนั้น

$$x(n-2) = Y_0 \tag{2.24}$$

$$x(n-2) - x(n-1) = Y_1 \tag{2.25}$$

$$x(n-2) - 2x(n-1) + x(n) = Y_2 \tag{2.26}$$

จากสมการที่ (2.25) จะได้

$$Y_0 - x(n-1) = Y_1 \tag{2.27}$$

จากสมการที่ (2.26) และ (2.27) จะได้

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

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

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

เราพิจารณา Y2 สำหรับ y(n) เพราะเราพิจารณา x2 เป็น x(n) ดังนั้น

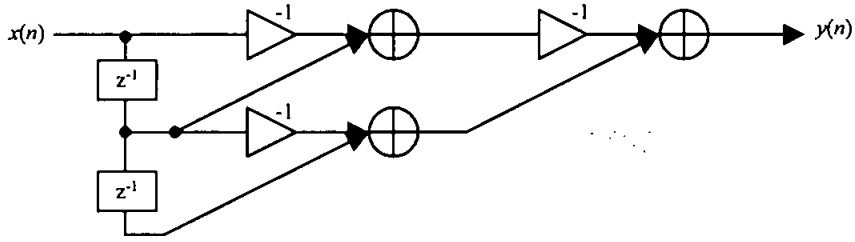
$$\therefore \quad y(n) = x(n) - 2x(n-1) + x(n-2)$$

ดังนั้นจะได้ทรานสเฟอร์ฟังก์ชันคือ

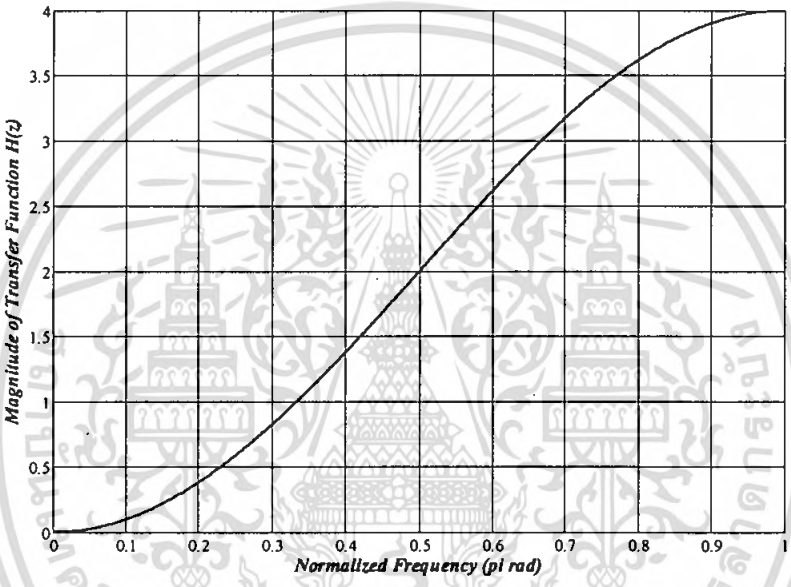
$$\therefore \quad H(z) = 1 - 2z^{-1} + 1z^{-2} \tag{2.28}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะได้โครงสร้างทางฮาร์ดแวร์ของวงจรกรองปาสคาล ดังในรูปที่ 2.29 และแสดงคุณลักษณะทางความถี่ (เชิงขนาด) ได้ดังรูปที่ 2.30 ซึ่งจะเห็นได้ว่ามีคุณลักษณะเป็นวงจรกรองความถี่สูงผ่าน



รูปที่ 2.29 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (2.28)

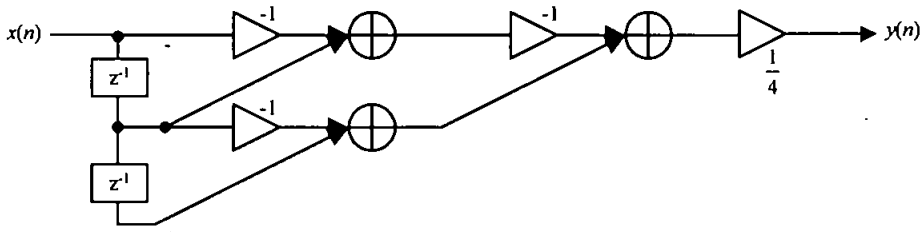


รูปที่ 2.30 ผลตอบสนองทางแอมปลิจูดจากสมการที่ (2.28)

จากรูปที่ 2.30 ถ้าต้องการค่าสูงสุดของผลตอบสนองทางแอมปลิจูดเป็น 1 ต้องปรับค่าทรานสเฟอร์ฟังก์ชันก่อนหน้าโดยปรับลงด้วยผลรวมของค่าสมบูรณ์ของสัมประสิทธิ์ทั้งหมด

จะได้ทรานสเฟอร์ฟังก์ชันใหม่คือ

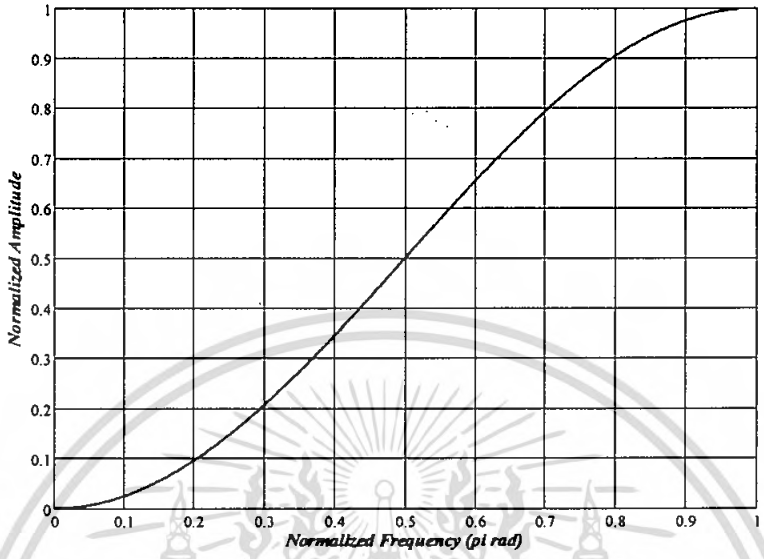
$$\therefore \quad H(z) = \frac{1}{4}\left(1 - 2z^{-1} + 1z^{-2}\right) \tag{2.29}$$



รูปที่ 2.31 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (2.29)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

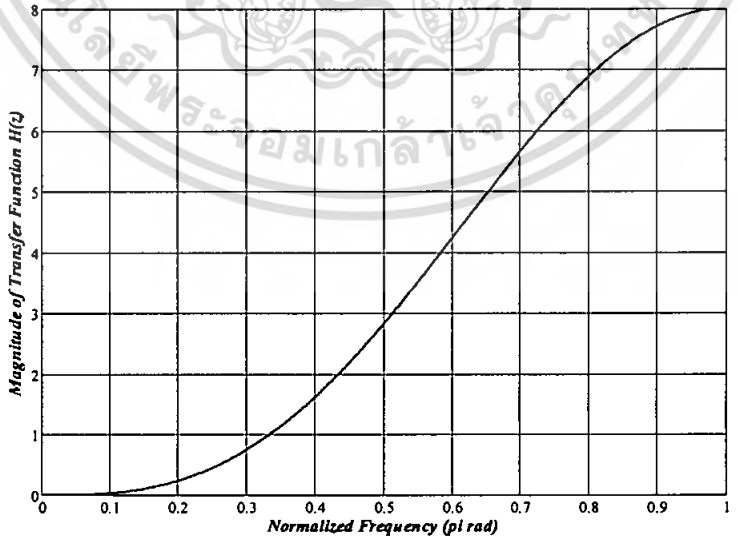
ในการปรับค่าสูงสุดของผลตอบสนองทางแอมปลิจูดนั้น ในส่วนของโครงสร้างฮาร์ดแวร์ จะทำได้โดยการเลื่อนบิตเอาต์พุตไปทางขวามือซึ่งให้ผลเสมือนกับการหารนั่นเอง จะได้ผลตอบสนองความถี่ใหม่ตามรูปที่ 2.32



รูปที่ 2.32 ผลตอบสนองทางความถี่จากสมการที่ (2.29)

กรณีที่ N=4, order ของ filter คือ N-1 = 3 th order ของวงจรกรองปาสคาล
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน (Normalized transfer function) จะได้สมการดังนี้

$$H(z) = 1 - 3z^{-1} + 3z^{-2} - 1z^{-3} \tag{2.30}$$

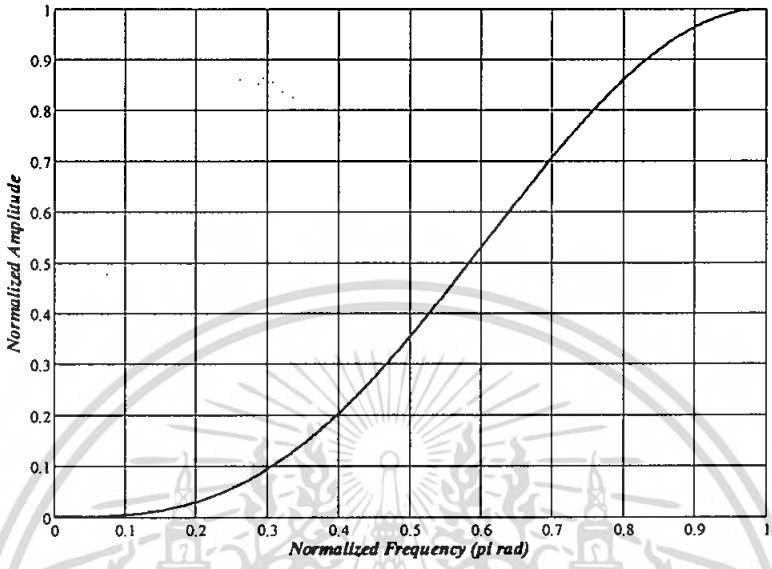


รูปที่ 2.33 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า Nth order มีค่า 3th order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

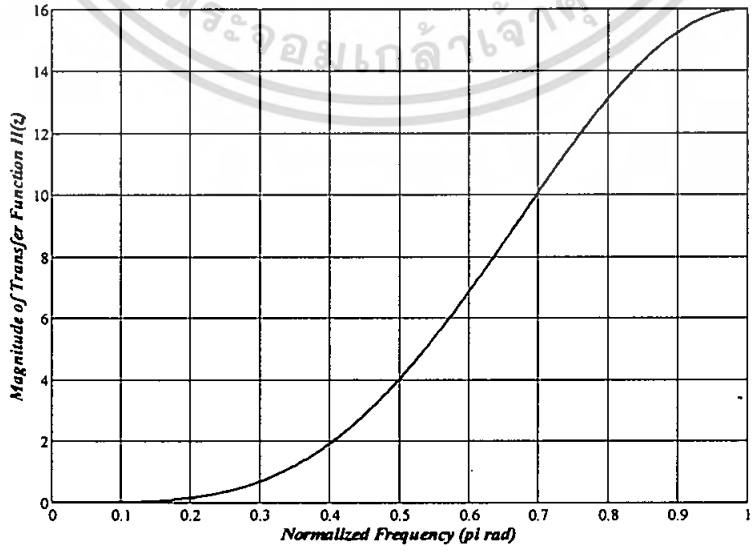
$$H(z) = \frac{1}{8}\left(1 - 3z^{-1} + 3z^{-2} - 1z^{-3}\right) \tag{2.31}$$



รูปที่ 2.34 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 3^{th} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่ N=5, order ของฟิลเตอร์ คือ N-1 = 5^{th} order ของวงจรกรองปาสคาล ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

$$H(z) = 1 - 4z^{-1} + 6z^{-2} - 4z^{-3} + 1z^{-4} \tag{2.32}$$

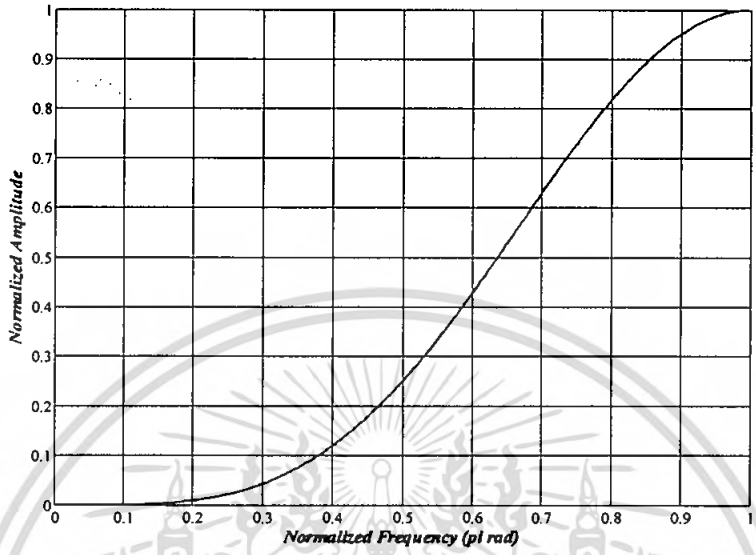


รูปที่ 2.35 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

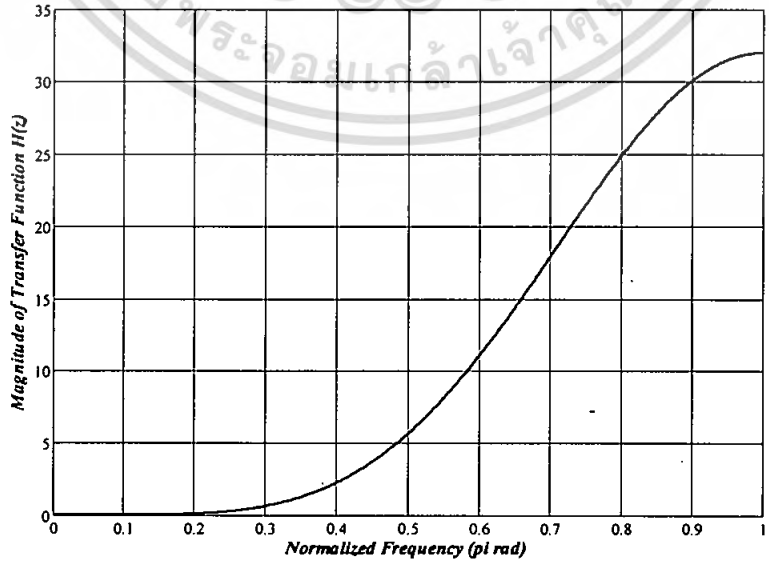
$$H(z) = \frac{1}{16}\left(1 - 4z^{-1} + 6z^{-2} - 4z^{-3} + 1z^{-4}\right) \tag{2.33}$$



รูปที่ 2.36 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 4^{th} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่ N=6, order ของฟิลเตอร์ คือ N-1 = 5^{th} order ของวงจรกรองปาสคาล ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

$$H(z) = 1 - 5z^{-1} + 10z^{-2} - 10z^{-3} + 5z^{-4} - 1z^{-5} \tag{2.34}$$

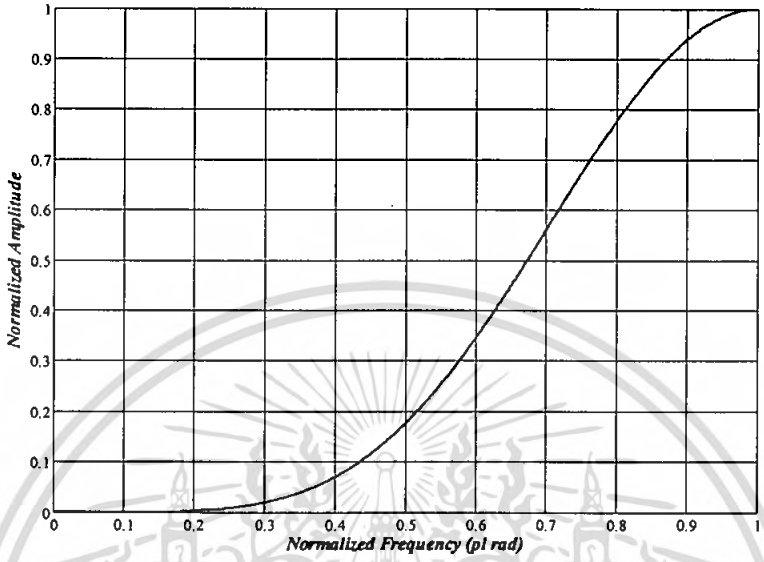


รูปที่ 2.37 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

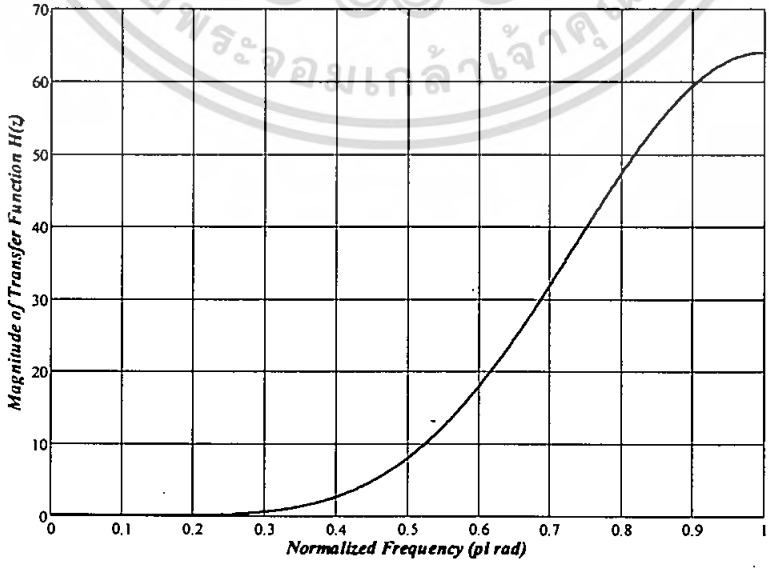
$$H(z) = \frac{1}{32}\left(1 - 5z^{-1} + 10z^{-2} - 10z^{-3} + 5z^{-4} - 1z^{-5}\right) \tag{2.35}$$



รูปที่ 2.38 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 5^{th} order และทำการนอร์มอลไลซ์แล้ว

กรณีที่ N=7, order ของฟิลเตอร์ คือ N-1 = 6^{th} order ของวงจรกรองปาสคาล ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

$$H(z) = 1 - 6z^{-1} + 15z^{-2} - 20z^{-3} + 15z^{-4} - 6z^{-5} + 1z^{-6} \tag{2.36}$$

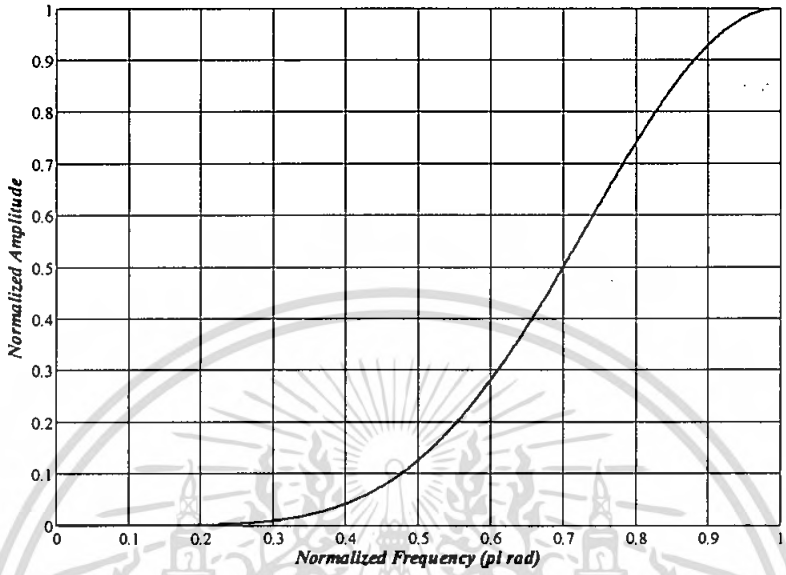


รูปที่ 2.39 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

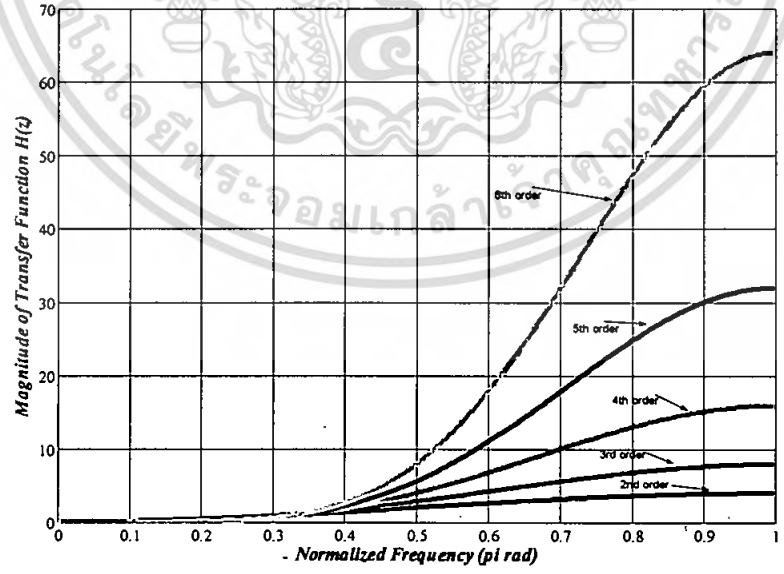
เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

$$H(z)=\frac{1}{64}(1-6z^{-1}+15z^{-2}-20z^{-3}+15z^{-4}-6z^{-5}+1z^{-6}) \quad (2.37)$$



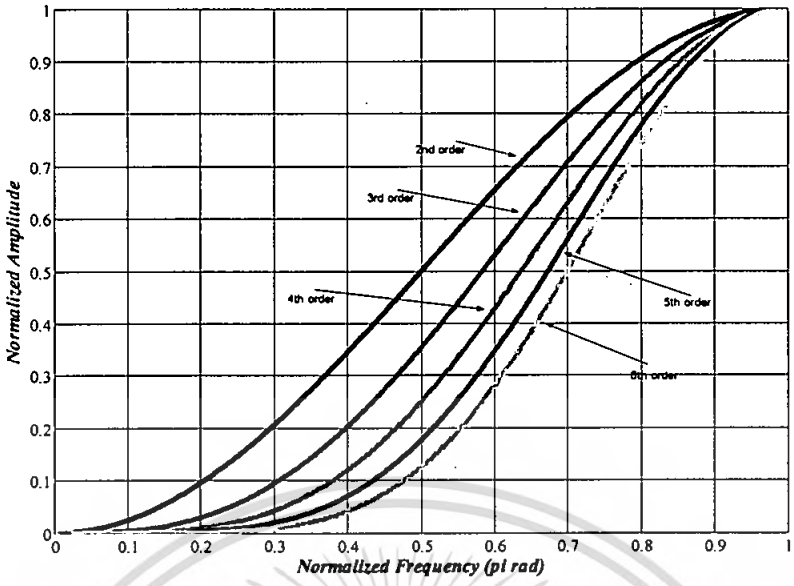
รูปที่ 2.40 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่า 6^{th} order และทำการนอร์มอลไลซ์แล้ว

เมื่อทำการแสดงผลกราฟที่ N^{th} order มีค่าต่าง ๆ กัน

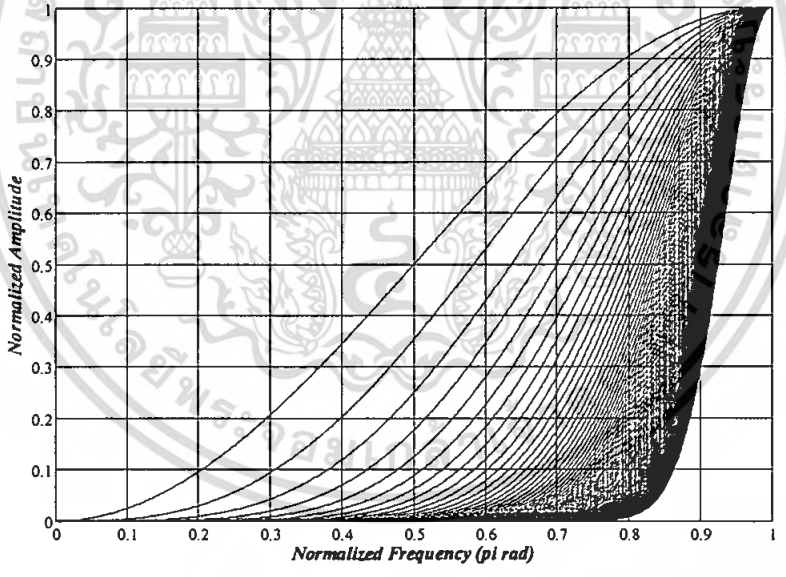


รูปที่ 2.41 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.42 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N^{th} order มีค่าตั้งแต่ 2^{nd} order ถึง 6^{th} order เมื่อทำการนอร์มอลไลซ์แล้ว



รูปที่ 2.43 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 99^{th} order

จากรูปที่ 2.42 และ 2.43 จะสังเกตเห็นได้ว่า เมื่ออันดับของวงจรกรองสูงขึ้น ค่าของความชันที่ได้จะมีค่าที่ค่อนข้างจะเปลี่ยนแปลงน้อยซึ่งผิดจากกรณีของวงจรกรองสัญญาณทั่ว ๆ ไป ที่เมื่ออันดับของวงจรกรองสูงขึ้น ความชันของผลตอบสนองทางขนาดจะเพิ่มขึ้นแต่สิ่งที่เปลี่ยนแปลงเมื่ออันดับของวงจรกรองที่ได้นี้เปลี่ยนไปก็คือความกว้าง-แคบของย่านผ่านสัญญาณซึ่งคุณลักษณะนี้จะคล้ายกับวงจรกรองสัญญาณที่เรียกว่าวงจรกรองเกาส์เซียน (Gaussian Filter) เมื่ออันดับของวงจรเปลี่ยนไป จะมีผลทำให้ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของผลตอบสนองทางขนาดนี้เปลี่ยนไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ผลตอบสนองอิมพัลส์ (Impulse response) ของวงจรกรองปาสคาล 2 มิติ (2-D Pascal Filter)</u>

ตัวอย่างเช่น : เมตริกซ์ขนาด 3 x 3

$$\text{จาก}\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \\ X_{20} & X_{21} & X_{22} \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.38}$$

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้โดยการเปลี่ยนตัวแปรเป็น

$$\begin{bmatrix} Y_{00} & Y_{01} & Y_{02} \\ Y_{10} & Y_{11} & Y_{12} \\ Y_{20} & Y_{21} & Y_{22} \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} x(m-2,n-2) & x(m-2,n-1) & x(m-2,n) \\ x(m-1,n-2) & x(m-1,n-1) & x(m-1,n) \\ x(m,n-2) & x(m,n-1) & x(m,n) \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.39}$$

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

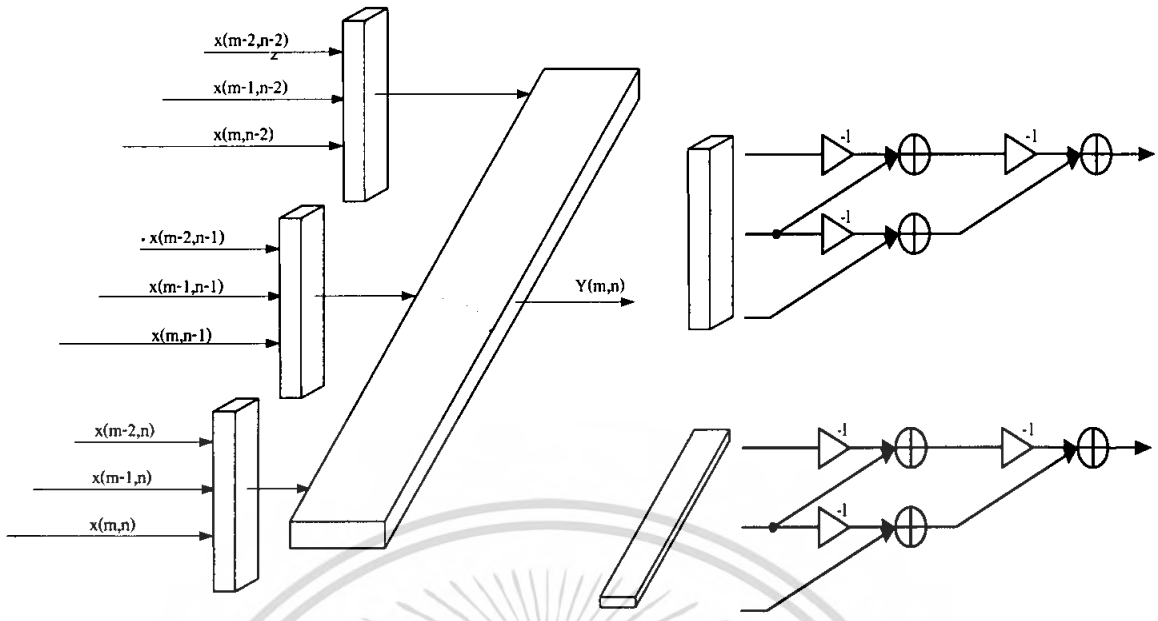
$$\begin{aligned} \therefore\ y(m,n) &= x(m,n)-2x(m,n-1)+x(m,n-2) \\ &\quad -2x(m-1,n)+4x(m-1,n-1)+2x(m-1,n-2) \\ &\quad +x(m-2,n)-2x(m-2,n-1)+x(m-2,n-2) \end{aligned} \tag{2.40}$$

ดังนั้นจะได้ผลตอบสนองอิมพัลส์ของระบบประมวลผลสัญญาณ 2 มิติ ในหน้ากากคอนโวลูชัน (Covolution Mask) และเราจะเรียกหน้ากากที่ได้นี้ว่าหน้ากากคอนโวลูชันปาสคาล (Pascal Convolution Mask) ดังแสดงในรูปที่ 2.44

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

รูปที่ 2.44 แสดงหน้ากากคอนโวลูชันของวงจรกรองปาสคาล 2 มิติ ซึ่งมีขนาด 3x3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.45 โครงสร้างของวงจรกรองปาสคาล 2 มิติ ตามสมการที่ (2.40)

2.14 การกรองข้อมูลภาพ (Image Filtering)

การกรองข้อมูลภาพคือ การนำภาพไปผ่านตัวกรองสัญญาณเพื่อให้ได้ภาพผลลัพธ์ออกมา ภาพผลลัพธ์ที่ได้จะมีคุณสมบัติแตกต่างจากภาพต้นแบบ วัตถุประสงค์หลักของการกรองข้อมูลภาพ คือการเน้น (enhance) หรือลดทอน (attenuate) คุณสมบัติบางประการของภาพ เพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ

การกรองข้อมูลภาพคือการประมวลผลภาพอย่างหนึ่งที่จำเป็นมาก เนื่องจากในการใช้งานจริง ภาพที่ได้มามักมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่น ๆ ปะปนอยู่ด้วย การกรองข้อมูลภาพสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้น เหมาะแก่การประมวลผลในขั้นต่อไป

ตัวกรองคือระบบ ๆ หนึ่งซึ่งรับสัญญาณเข้าประมวลผลสัญญาณและส่งสัญญาณออก โดยทั่วไป ตัวกรองจะถูกสร้างให้เป็นระบบเชิงเส้น (linear system) เนื่องจากออกแบบได้ง่าย และมีประสิทธิภาพดี ซึ่งในการกรองข้อมูลภาพ เรามักพิจารณาว่าภาพคือสัญญาณ 2 มิติที่ประกอบขึ้นจากสัญญาณความถี่ต่าง ๆ ผสมกันอยู่ในสัดส่วนที่ต่างกัน การออกแบบตัวกรองจึงเป็นการกำหนดว่าเราต้องการกำจัดสัญญาณความถี่ใดออกไป หรือต้องการเลือกสัญญาณความถี่ใดบ้าง

ซึ่งตัวกรองแบ่งออกได้เป็น 4 ประเภทตามลักษณะการเลือกความถี่คือ

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

โดยที่ค่าพารามิเตอร์หลักในการกำหนดคุณสมบัติของตัวกรองคือ ค่าความถี่คัทออฟ (cut-off frequency) ความถี่คัทออฟคือ ความถี่ที่ระบุจุดตัดของสัญญาณว่าจะให้ผ่าน หรือไม่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 ภาพไบนารี

ภาพไบนารี คือภาพซึ่งในแต่ละพิกเซลซึ่งจะมีความเข้มของแสงได้เพียงสองแบบเท่านั้น คือสว่างกับมืด โดยอาจเขียนแทนได้ด้วยเลข 0 และ 1 ด้วยความเข้มของแสงเพียงสองระดับ ภาพไบนารีจึงมีข้อจำกัดที่จะนำมาใช้แสดงภาพทั่ว ๆ ไป แต่อย่างไรก็ตามการมีความเข้มของแสงเพียงสองระดับก็ทำให้การประมวลผลทำได้อย่างมีประสิทธิภาพ การประมวลผลภาพไบนารีนำไปใช้มากในการประมวลผลเอกสาร การประมวลผลภาพในอุตสาหกรรมที่ต้องการความเร็วสูง เช่น การนับจำนวนชิ้นส่วนที่อยู่บนสายพานที่กำลังเคลื่อนที่ เป็นต้น การมีความเข้มเพียงสองระดับในภาพไบนารี ทำให้สามารถที่จะเลือกพิจารณาให้ความเข้มระดับหนึ่งแทนภาพของสิ่งที่เราสนใจ โดยเราจะขอเรียกพิกเซลที่มีความเข้มระดับนี้ว่าพิกเซลภาพ และความเข้มอีกระดับแทนพื้นหลังโดยพิกเซลที่มีความเข้มระดับนี้จะถูกเรียกว่าพิกเซลพื้นหลัง ในทางปฏิบัติในการพิจารณาภาพไบนารี เราจะไม่สนใจแต่ละพิกเซลแยกกันไป แต่เราจะสนใจกลุ่มของพิกเซลที่อยู่ติดกัน หรือที่เรียกว่าพิกเซลเพื่อนบ้าน

2.16 ค่าเทรสโฮล

ในบางครั้งภาพระดับสีเทาที่พิจารณามีความเด่นชัดของวัตถุในภาพที่แยกตัวออกจากพื้นหลังอย่างชัดเจน ในกรณีนี้เราอาจแยกภาพของวัตถุออกจากพื้นหลังโดยการเปลี่ยนภาพเกรย์สเกลให้เป็นภาพไบนารี แต่ภาพเกรย์สเกลมีระดับความเข้มของแสงได้มากกว่าสองระดับดังนั้นเราจึงต้องมีวิธีเลือกว่าพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลพื้นหลัง และพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลภาพ วิธีที่นิยมใช้วิธีหนึ่งก็คือการกำหนดค่าความเข้มของแสงค่าหนึ่งที่ใช้แยกแยะ เช่น หากความเข้มของแสงของพิกเซลที่พิจารณามีค่าน้อยกว่าค่านี้พิกเซลนั้นก็เป็นพิกเซลพื้นหลัง มิฉะนั้นพิกเซลนั้นก็ถือเป็นพิกเซลภาพ ค่าสำหรับแยกประเภทของพิกเซลที่ว่านี้เราเรียกว่า ค่าเทรสโฮล (threshold)

2.17 ค่าความเข้มแสงของภาพ (Contrast)

ค่าความเข้มแสงของภาพเป็นความแตกต่างในคุณสมบัติการมองเห็นซึ่งทำให้วัตถุ (หรือการดำเนินการของมันในภาพ) ซึ่งสามารถแยกแยะความแตกต่างได้อย่างชัดเจนจากวัตถุอื่นและภาพพื้นหลัง ในการรับรู้เรื่องการมองเห็นของโลกความเป็นจริงนั้น ค่าความเข้มแสงของภาพถูกกำหนดโดยความแตกต่างในสีและความสว่างของวัตถุและวัตถุอื่นภายในอาณาเขตของการมองเห็นเดียวกัน เพราะระบบการมองเห็นของมนุษย์นั้นจะมีความไวมากกว่าใจความแตกต่างของความสว่าง ในทำนองเดียวกันเราสามารถที่จะสัมผัสถึงโลกโดยที่ไม่คำนึงถึงการเปลี่ยนแปลงอย่างมากของความสว่างของแสงในแต่ละวันหรือจากสถานที่หนึ่งไปสถานที่หนึ่ง นั่นคือในแต่ละช่วงเวลา เมื่อความเข้มแสงมีการเปลี่ยนแปลงไปเรื่อย ๆ เรานั้นแทบจะไม่รู้สึกได้ถึงการเปลี่ยนแปลงนั้นเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

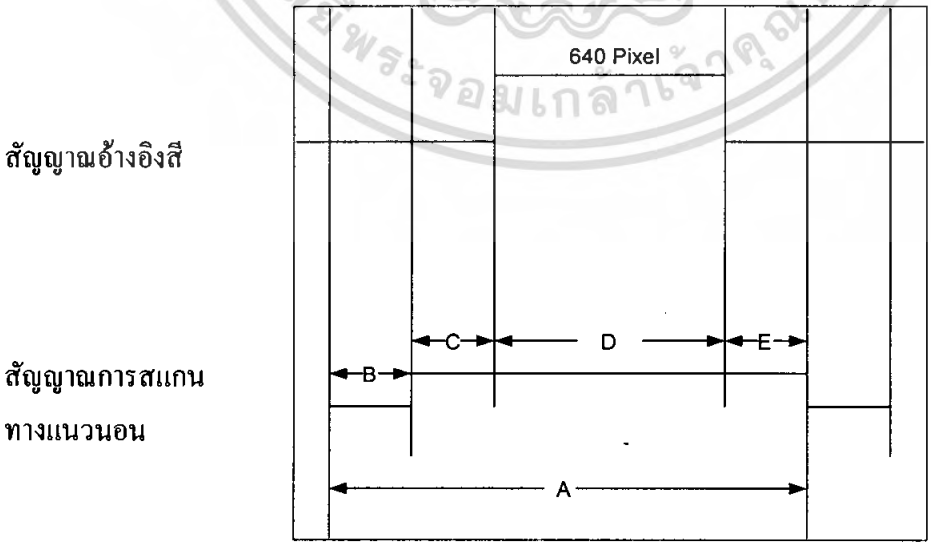
อุปกรณ์เอฟพีจีเอ เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์เอฟพีจีเอมีฟังก์ชันการทำงานตามที่ออกแบบไว้ ในการทำเอฟพีจีเอซึ่งเป็นวิธีการออกแบบไอซี (IC : Integrated Circuit) แบบเซมิคัสทอม (Semicustom) อีกวิธีหนึ่ง เมื่อเทียบกับการทำวงจรวมแล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือ การทำเอฟพีจีเอจะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์เอฟพีจีเอจะมีจำนวนเกท (Gate) ให้ใช้จำนวนจำกัดและการทำเอฟพีจีเอก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำเอฟพีจีเอก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (Code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดนั้นน้อยกว่าการทำวงจรรวมมากและการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

สำหรับตัวอุปกรณ์เอฟพีจีเอนั้นก็มีโครงสร้างพื้นฐานเทคโนโลยีที่ใช้สร้างตลอดจนเทคนิควิธีการโปรแกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นอุปกรณ์เอฟพีจีเอของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วน ในการใช้งานนั้นอุปกรณ์เอฟพีจีเอสามารถนำไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP: Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้น

ในโครงงานนี้ได้ใช้ชิพเบอร์ XC3S200 โดยรับสัญญาณนาฬิกาหรือคล็อค (clock) จากออสซิลเลเตอร์ 25.175 MHz ซึ่งใช้การเขียนภาษาวีเอชดีแอล

3.1 การสร้างสัญญาณควบคุมวีจีเอ

3.1.1 การสร้างสัญญาณสแกนทางแนวนอน (Horizontal Synchronization)



รูปที่ 3.1 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณการสแกนทางแนวนอนจะเป็นตัวกำหนดการสแกนทางแนวนอนในแต่ละแถวจะมีค่าข้อมูล อยู่ในช่วงประมาณ 794 คือช่วง C+D+E นั่นเองโดยจะมีสัญญาณการอ้างอิงสีเป็นสัญญาณ บอกถึงต้องการสแกนในช่วง 0-639 ตามการสแกนตามมาตรฐานนั่นเอง

ตารางที่ 3.1 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวนอน และค่าการแปลงเวลาเป็นข้อมูลที่ใช้ในการเขียนโปรแกรม

Parameter	A	B	C	D	E
Time	31.77 µs	3.77 µs	1.89 µs	25.17 µs	0.94 µs
Data	794.25	94.25	47.25	629.25	23.5

จากตารางที่ 3.1 จะแสดงช่วงค่าของวีจีเอไทม์มิ่ง (VGA Timing) ของสัญญาณการสแกนทางแนวนอน ซึ่งจะใช้เวลาต่าง ๆ ตามช่วงที่กำหนดโดยประมาณ นอกจากนั้นเราจะแปลงค่าของเวลาเป็นค่าของข้อมูล เพื่อการใช้ในการเขียนโปรแกรมด้วย ภาษาวีเอชดีแอลจากค่าเวลาต่าง ๆ จะมีการคำนวณดังนี้

A คือเวลาทั้งหมดของการสแกนทางแนวนอน

B, G และ E คือ การ์ดแบนด์ (Guard Bands)

D คือเวลาการสแกนช่วงข้อมูล 640 พิกเซล/ แถว

ช่วงข้อมูลจะมีการคำนวณดังนี้

ความถี่ที่ใช้งานมีค่า 25.175 *MHz*

การสแกนแต่ละพิกเซลจะใช้เวลา

Tpixel = 1/25.175 *MHz* = 39.721 หรือประมาณ 40 ns

เวลาทั้งหมดในการสแกนในแต่ละแถว

Trow = A = B+C+D+E = (Tpixel*640) +B+C+E = 31.77 µs

ถ้าเทียบจากเวลาเป็นข้อ มูลจะมีการคำนวณดังนี้

A = 31.77 µs /40 ns = 794.25

B = 3.77 µs /40 ns = 94.25

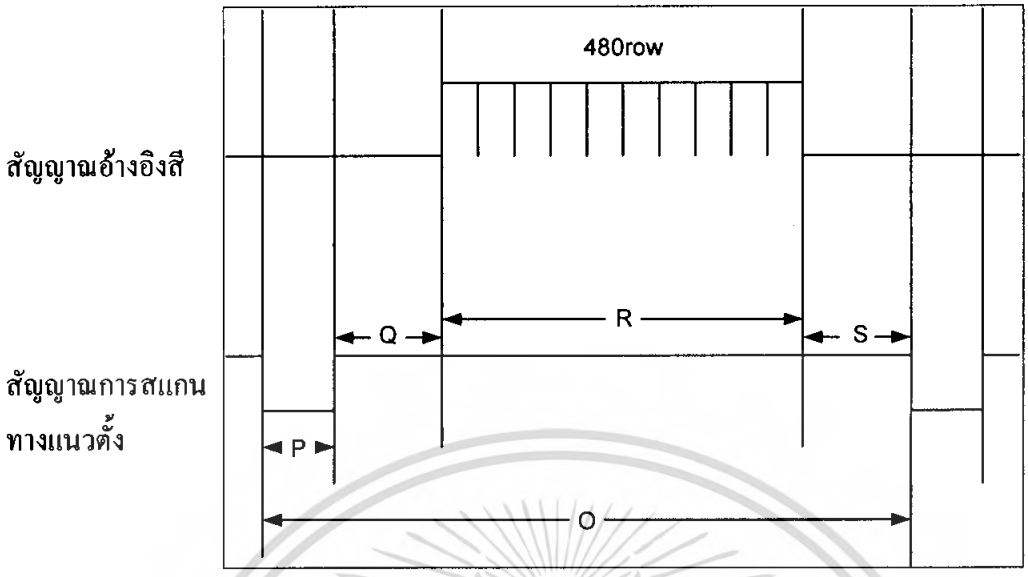
C = 1.89 µs /40 ns = 47.25

D = 25.17 µs /40 ns = 629.25

E = 0.94 µs /40 ns = 23.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 สัญญาณการสแกนทางแนวตั้ง (Vertical synchronization)



รูปที่ 3.2 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวตั้ง

สัญญาณการสแกนทางแนวตั้งจะเป็นตัวกำหนดการสแกนทางแถวตั้งโดยถ้ามีการสแกนทางแนวนอนครบ 523 แถวจะเกิดสัญญาณการสแกนทางแนวตั้ง1ลูก คือช่วง Q+R+S นั่นเองโดยจะมีสัญญาณการอ้างอิงสีเป็นสัญญาณ บอกถึงต้องการสแกนในช่วง 0-479 ตามการสแกนมาตรฐานคือ

ตารางที่ 3.2 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวตั้ง และค่าการแปลงเวลา เป็นข้อมูลที่ใช้ในการเขียนโปรแกรม

Parameter	O	P	Q	R	S
Time	16.6 ms	0.06 ms	1.02 ms	15.24 ms	0.35 ms
Data	522.505	2.014	32.105	479.697	11.016

จากตารางที่ 3.2 จะแสดงช่วงค่าของวีจีเอไทม์มิ่งของสัญญาณการสแกนทางแนวตั้ง ซึ่งจะใช้เวลาตามช่วงที่กำหนดโดยประมาณ นอกจากนั้นเราจะแปลงค่าของเวลาเป็นค่าของข้อมูลเพื่อใช้ในการเขียนโปรแกรมด้วยภาษาวีเอชดีแอลจากค่าเวลาต่าง ๆ จะมีการคำนวณดังนี้

O คือเวลาทั้งหมดของการสแกนทางแนวตั้ง

P, Q และ S คือ การ์ดแบนด์

R คือเวลาการสแกนช่วงข้อมูล 480 แถว/เฟรม คิดเป็น 307, 200 พิกเซลต่อเฟรม ช่วงข้อมูลจะมีการคำนวณและจะมีการได้สัญญาณสแกนทางแนวตั้งจะใช้เวลาดังนี้

สัญญาณการสแกนทางแนวตั้ง 1 ลูก = สัญญาณการสแกนทางแนวนอน 523 แถว เวลาที่ใช้ในการสแกนแต่ละแถวคือ 31.77 µs

Trow = 31.77 µs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาทั้งหมดในการสแกนทางแนวตั้ง

Tframe = O = P+Q+R+S = (Trow * 480rows) +P+Q+S = 16.6 ms

ถ้าเทียบจากเวลาจะได้ดังนี้

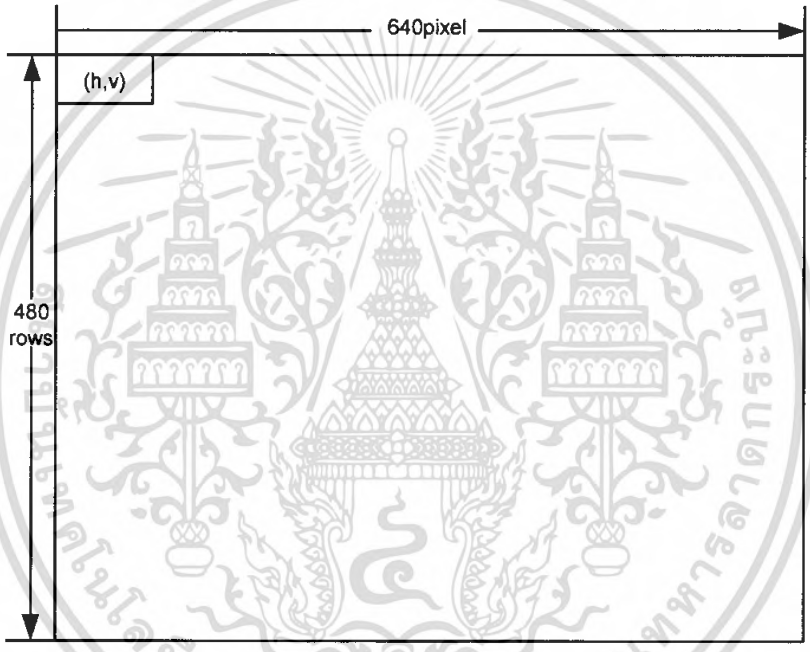
O = 16.6 ms/31.77 μs = 522.505

P = 0.06 ms/31.77 μs = 2.014

Q = 1.02 ms/31.77 μs = 32.105

S = 0.35 ms/31.77 μs = 11.016

3.1.3 สัญญาณอ้างอิงตำแหน่งทางแนวนอนและแนวตั้ง



รูปที่ 3.3 แสดงการอ้างอิงของจุดภาพ

จากการที่ได้กล่าวมาแล้วเราจะใช้ข้อมูลที่ได้จากการแปลงค่าจากช่วงเวลา มาสร้างสัญญาณตามที่ต้องการ ในการสร้างสัญญาณอ้างอิงตำแหน่ง จะใช้วงจรนับโดยมีคล็อคตามการคำนวณของทั้งการสร้างสัญญาณทั้งทางแนวนอนและแนวตั้งคือ

ที่ ขนาดภาพ 640*480 พิกเซล และ อัตราการแสดงภาพใหม่ = 60 *Hz*

- คล็อคการอ้างอิงตำแหน่งทางแนวนอนใช้ความถี่ประมาณ 25.175 *MHz* คิดเป็นเวลาคือประมาณ 40 ns

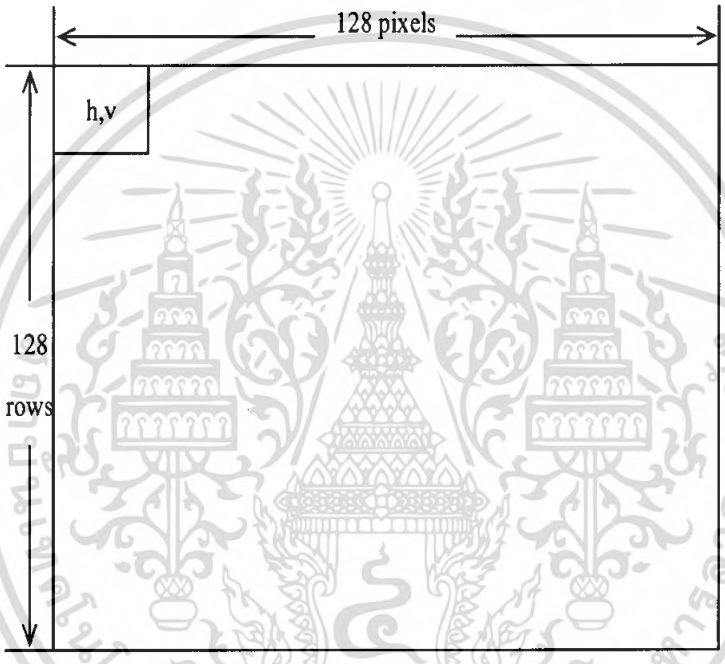
- คล็อคการอ้างอิงตำแหน่งทางแนวตั้งจะใช้ความถี่ประมาณ 31.476 *kHz* คิดเป็นเวลาคือประมาณ 31.77 μs

ส่วนการอ้างอิงสีนั้นจะสามารถกำหนดตำแหน่งได้ให้เกิดสีปรากฏบนหน้าจอวีจีเอได้โดยดูจากรูปที่ 3.3 จะมีความกว้างทางแนวนอน 640 พิกเซล แทนด้วย h และจะมีความยาว 480 แถว แทนด้วย v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

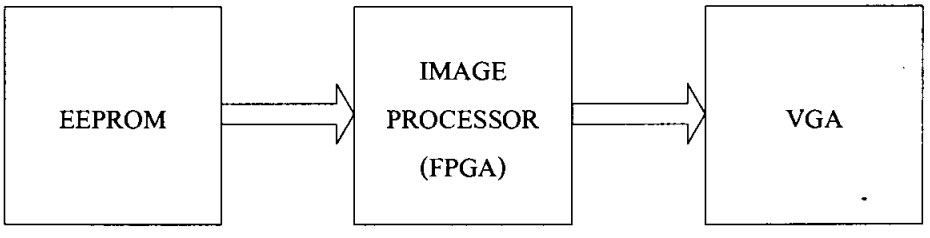
ถ้าต้องการให้สีปรากฏ ที่ตำแหน่ง [h, v] เราก็จะใช้สัญญาณเป็นตัวอ้างอิงเมื่อถึงตำแหน่งที่ต้องการให้สีปรากฏออกไป ซึ่งสัญญาณมีลักษณะเป็นบัสในการอ้างอิงตำแหน่งสี ที่เกิดจากการนับของดีฟลิปฟลอป ซึ่งก็จะใช้สัญญาณการอ้างอิงตำแหน่งนั่นเอง แต่จะมีสัญญาณการสแกนทางแนวนอนและแนวตั้งเป็นตัวควบคุมอยู่ ดังแสดงในรูป 3.1 และ 3.2

ซึ่งดังที่ได้กล่าวมาแล้วนั้นเป็นการสร้างสัญญาณอ้างอิงทางตำแหน่งไม่ว่าจะเป็นทั้งทางแนวนอนและแนวตั้งก็ตามจะเป็นการสร้างสัญญาณของหน้าจอวีจีเอทั้งหน้าจอ (ขนาด 640x480 พิกเซล) แต่ในโครงงานนี้เราจะทดลองทำการสร้างสัญญาณอ้างอิงทางตำแหน่งทางแนวนอนเป็น 128 พิกเซล และสัญญาณอ้างอิงทางตำแหน่งทางแนวตั้งเป็น 128 แถว ก่อนในขั้นต้น ดังรูปที่ 3.4



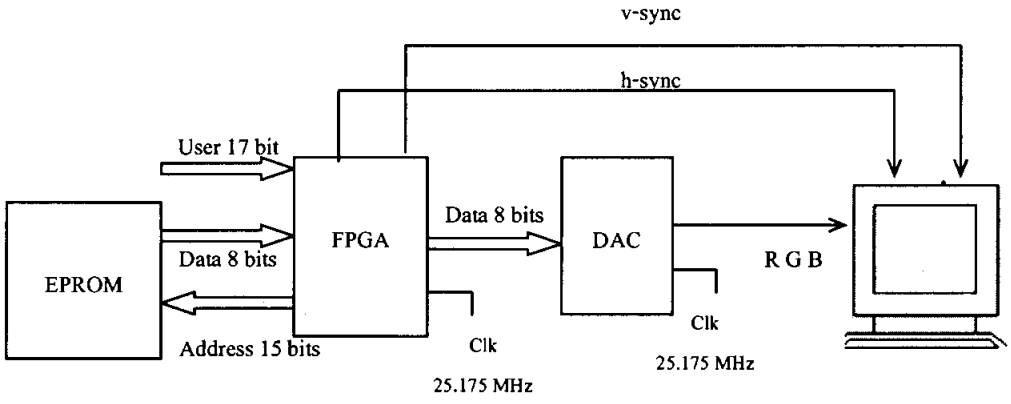
รูปที่ 3.4 แสดงการอ้างอิงของจุดภาพ (128x128)

3.2 การออกแบบวงจร



รูปที่ 3.5 วงจรรวมทั้งหมดของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรรวมที่จะนำข้อมูลของภาพจากหน่วยความจำภายนอกเข้าสู่เอฟพีจีเอ เพื่อทำการประมวลผลภาพและแสดงผลทางวีจีเอ

3.2.1 วงจรที่สร้างจากเอฟพีจีเอ

มีโครงสร้างหลัก ๆ ตามรูปที่ 3.6 โดยจะอาศัยหน่วยความจำภายนอก (EEPROM) เพื่อที่เก็บข้อมูลแต่ละพิกเซลของภาพ ซึ่งแต่ละพิกเซลนั้นจะมีขนาด 8 บิต ในโครงงานนี้เราจะใช้ภาพขนาด 128x128 พิกเซล ในวงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเอฟพีจีเอที่ออกแบบด้วยภาษาวีเอชดีแอล จะทำการชี้ตำแหน่งแอดเดรสของข้อมูลให้กับหน่วยความจำภายนอกด้วยความเร็ว 25.175 *MHz* จากนั้นทำการรับข้อมูลภาพแต่ละพิกเซลจากหน่วยความจำภายนอกที่ส่งกลับมาให้กับเอฟพีจีเอทำการประมวลผลสัญญาณภาพในลักษณะต่าง ๆ เช่น การปรับระดับความเข้มแสงของภาพ การขยายขนาดของภาพดั้งเดิม การจัดตำแหน่งการแสดงภาพบนหน้าจอวีจีเอ และทำการประมวลผลภาพผ่านโครงสร้างของปาสคาลฟิลเตอร์ โดยการใช้สมการทรานสเฟอร์ฟังก์ชัน ข้อมูลที่ได้จะทำการประมวลผลกับสัญญาณควบคุมการแสดงผลภาพ จากนั้นจะนำข้อมูลเข้าสู่วงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกในระดับสีเทา ทำการแปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาล็อกที่มีระดับแรงดันต่าง ๆ กัน สัญญาณเอาท์พุตที่ได้จากวงจรนี้จะให้ระดับสีที่แตกต่างกัน จากนั้นนำสัญญาณที่ได้เชื่อมต่อเข้ากับขาอาร์จีบีของหน้าจอวีจีเอ นอกจากนี้แล้ววงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเอฟพีจีเอยังสร้างสัญญาณควบคุมการสแกนจอภาพทางแนวนอนและแนวตั้งอีกด้วย เมื่อนำสัญญาณควบคุมการสแกนจอภาพทางแนวนอนและแนวตั้งเชื่อมต่อกับหน้าจอวีจีเอก็จะสามารถแสดงผลภาพที่หน้าจอได้

ซึ่งส่วนประกอบของวงจรแสดงผลภาพโดยใช้บอร์ดเอฟพีจีเอสู่หน้าจอวีจีเอจะประกอบด้วยวงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง วงจรสร้างตำแหน่งของข้อมูล วงจรรับสัญญาณข้อมูลภาพจากหน่วยความจำภายนอกและวงจรประมวลผลภาพในรูปแบบต่างๆโดยแต่ละส่วนนั้นมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1 วงจรสร้างตำแหน่งของข้อมูล

วงจรสร้างตำแหน่งของข้อมูล ซึ่งทำหน้าที่สร้างตำแหน่งของข้อมูลให้กับหน่วยความจำภายนอก วงจรนี้สร้างจากวงจรนับ (Counter) ขนาด 14 บิต ซึ่งจะทำการนับค่าเพิ่มขึ้นทีละ 1 บิต จาก "00000000 000000" ถึง "11111111111111" แล้ววนไปเรื่อย ๆ ความเร็วในการนับนั้นจะขึ้นอยู่กับสัญญาณนาฬิกาที่ป้อนให้กับวงจร ซึ่งในโครงงานนี้จะทำการทดลองโดยใช้สัญญาณนาฬิกาที่มีความถี่ 25.175 *MHz* เพื่อให้สัมพันธ์กับความถี่ที่ใช้ในการกำเนิดภาพบนหน้าจอวีจีเอ

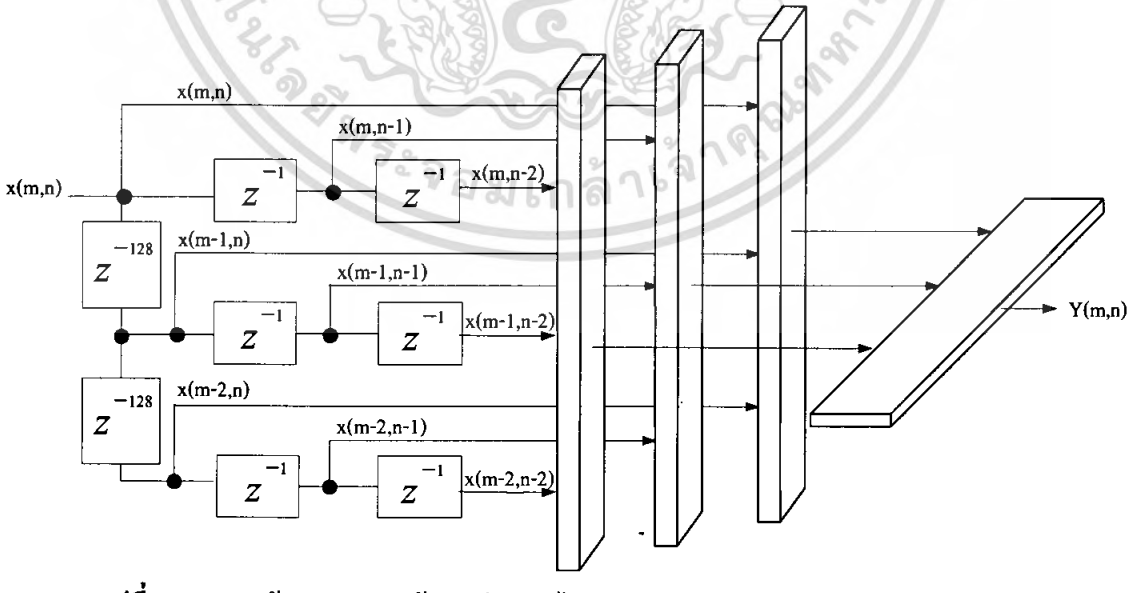
3.2.1.2 วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง

วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง จะทำหน้าที่สร้างสัญญาณสแกนหน้าจอทางแนวนอน และสัญญาณสแกนหน้าจอทางแนวตั้งซึ่งเราสามารถสร้างวงจรดังกล่าวได้โดยการใช้วงจรนับ โดยอาศัยข้อมูลจากการคำนวณที่ผ่านมาตามตารางที่ 3.1 และตารางที่ 3.2

3.2.1.3 วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอ

วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอมีหน้าที่รับสัญญาณข้อมูลภาพจากวงจรประมวลผลภาพโดยใช้ทฤษฎีการแปลงสามเหลี่ยมปาสคาล ทั้งนี้วงจรนี้จะทำหน้าที่ประมวลผลว่าจะให้ข้อมูลภาพนี้แสดงผลที่หน้าจอเมื่อใด เพื่อที่จะให้รูปภาพไม่เกิดการผิดเพี้ยนหรือบิดเบี้ยวไป

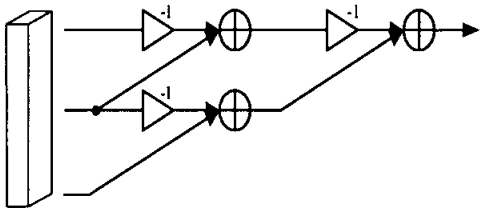
3. 2.1.4 วงจรประมวลผลภาพโดยใช้วงจรกรองปาสคาล 2 มิติ



รูปที่ 3.7 วงจรสร้างลำดับของข้อมูลก่อนนำไปประมวลผลกับวงจรกรองปาสคาล 2 มิติ

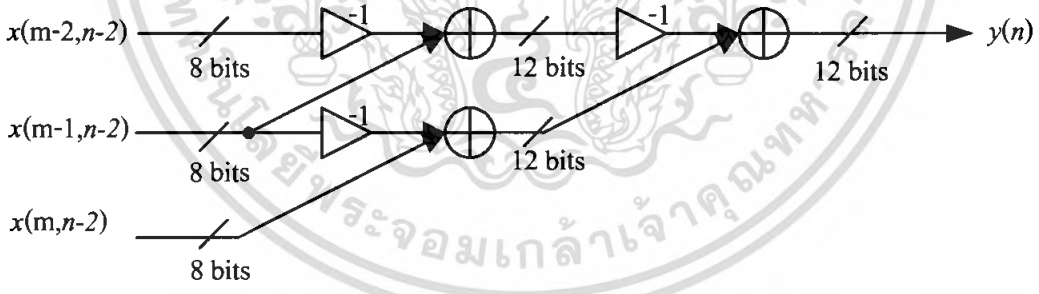
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสัญญาณที่ใช้ในรูปที่ 3.7 แสดงเป็นวงจรภายในดังรูปที่ 3.8



รูปที่ 3.8 แสดงบัตเตอร์ฟลายยูนิต เมื่อ N=3

นอกจากเราจะนำภาพที่มีอยู่ไปแสดงผลบนหน้าจอวีจีเอแล้ว ได้มีการทดลองนำภาพมาทำการประมวลผลโดยอาศัยหลักการของปาสคาลทรานสฟอร์มก่อนที่จะนำไปแสดงผลบนหน้าจอวีจีเออีกด้วย ซึ่งวงจรในส่วนของการประมวลผลภาพจะทำหน้าที่รับสัญญาณข้อมูลภาพจากหน่วยความจำภายนอกเข้ามา โดยที่ข้อมูลนั้นจะมีลักษณะเป็นบัสจำนวน 8 บิต ซึ่งข้อมูลที่เข้ามาจะต้องจัดลำดับสัญญาณข้อมูลเข้าให้ถูกต้องก่อนที่จะนำไปประมวลผล นั่นคือต้องสร้างลำดับข้อมูล x(m,n), x(m,n-1), x(m,n-2), x(m-1,n), x(m-1,n-1), x(m-1,n-2), x(m-2,n), x(m-2,n-1), x(m-2,n-2) ทั้งนี้กำหนดให้ m เป็นตัวหน่วงข้อมูลมีค่าเท่ากับ 128 ค่า จำนวน 2 ชุด โดยอาศัยตัวเลื่อนข้อมูล (shift register) ต่ออนุกรมกัน 256 ตัว ดังรูปที่ 3.8 จากนั้นนำข้อมูลที่จัดเรียงแล้วส่งไปคูณกับแผ่นข้อมูล (Mask) ที่มีเมตริกซ์ขนาด 3×3 เราสามารถออกแบบโครงสร้างของวงจรประมวลผลภาพได้จากสมการทรานสเฟอร์ฟังก์ชันตามสมการที่ (2.28) และจากรูปที่ 3.7 และ รูปที่ 3.8 สามารถออกแบบวงจรที่มีรายละเอียดดังรูปที่ 3.9



รูปที่ 3.9 ตัวอย่างรายละเอียดของวงจรบัตเตอร์ฟลายยูนิต เมื่อ N=3

ซึ่งในการเลื่อนจุดแต่ละจุดของภาพ เราทำการเลื่อนในลักษณะของการทำคอนโวลูชัน (Convolution) ข้อมูล $Y(n)$ ที่ได้จะเข้าสู่วงจรประมวลผลข้อมูลร่วมกับสัญญาณแสดงผลบนหน้าจอวีจีเอต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.5 วงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ

วงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ ทำหน้าที่ในการขยายขนาดของภาพ และสามารถจัดตำแหน่งของรูปภาพให้อยู่ในตำแหน่งต่าง ๆ ของหน้าจอตามความต้องการของผู้ใช้งาน

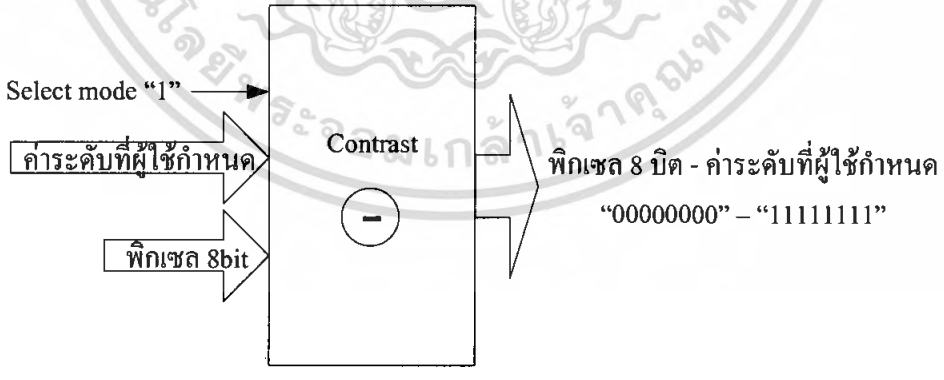
ในการจัดตำแหน่งของรูปภาพนั้นสามารถทำได้โดย ทำการหาตำแหน่งเริ่มต้นของพิกเซลแรกที่ต้องการแสดงภาพ ให้เป็นจุดเริ่มต้นของการแสดงผลบนหน้าจอ จากนั้นนับตำแหน่งของข้อมูลภาพต่อไปเรื่อย ๆ จนครบทั้งภาพ ก็จะได้ภาพที่อยู่ในตำแหน่งที่ต้องการ

นอกจากนี้ในการขยายขนาดของภาพนั้น สามารถทำได้โดยควบคุมการชี้ตำแหน่งข้อมูลของหน่วยความจำภายนอกให้สัมพันธ์กับจำนวนพิกเซลที่แสดงบนหน้าจอของรูปภาพขนาดต่าง ๆ ที่ต้องการเพียงเท่านี้ก็จะสามารถเลือกตำแหน่งในการแสดงภาพ และขนาดของภาพได้

3.2.1.6 วงจรปรับระดับความเข้มแสง

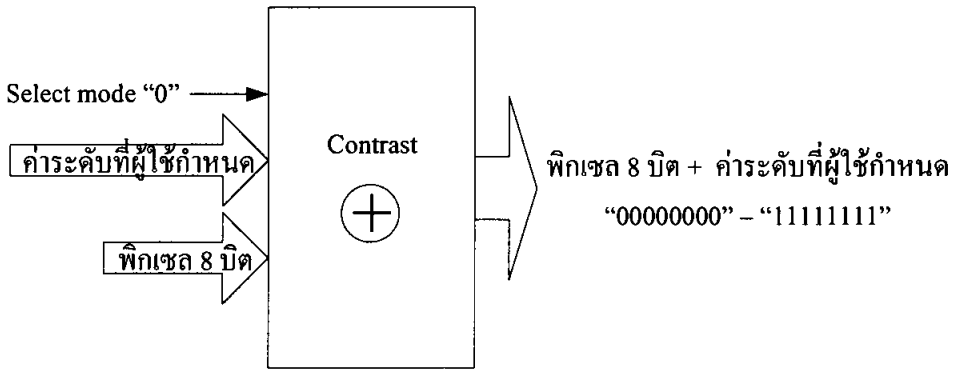
วงจรปรับระดับความเข้มแสง มีหน้าที่ในการปรับค่าความเข้มแสงในแต่ละพิกเซลของรูปภาพในการประมวลผลนั้นทำได้ โดยรับค่าระดับที่ต้องการปรับความเข้มแสงจากผู้ใช้งาน จากนั้นนำค่าที่รับมาทำการเพิ่มด้วยการบวกค่าที่รับมากับค่าความเข้มแสงของภาพนั้น หรือทำการลดค่าความเข้มแสงด้วยการลบค่าความเข้มแสงของภาพด้วยค่าที่รับมา ซึ่งผู้ใช้งานสามารถกำหนดได้ว่าต้องการเพิ่มหรือลดค่าความเข้มแสงของภาพนั้น ๆ

ทั้งนี้ในการประมวลผลนั้นหากค่าความเข้มแสงหลังจากประมวลผลแล้วมีค่าเกิน “11111111” จะให้พิกเซลนั้นมีค่าเท่ากับ “11111111” แต่ถ้าหลังจากการประมวลผลค่าความเข้มแสงมีค่าน้อยกว่า “00000000” จะให้พิกเซลนั้นมีค่าเท่ากับ “00000000” ซึ่งค่าที่ได้จะปรับระดับอยู่ที่ค่า 0-255 โดยที่ “11111111” มีค่าเป็นสีขาวและ “00000000” มีค่าเป็นสีดำ



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

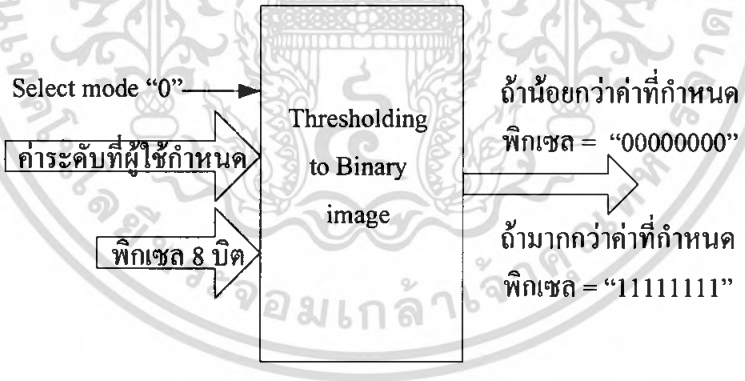


(ข)

รูปที่ 3.10 วงจรปรับระดับความเข้มแสง

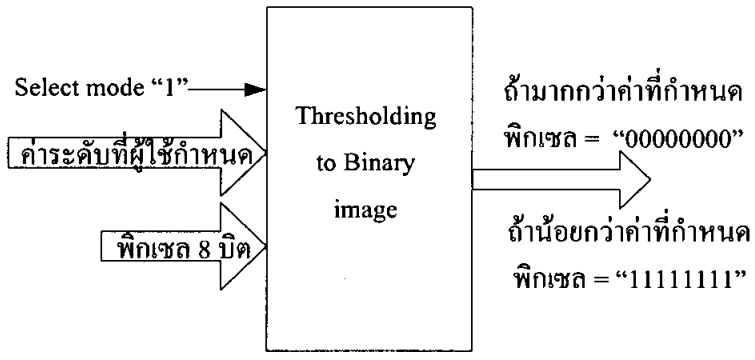
3.2.1.7 วงจรปรับความเข้มแสงของภาพเป็นสองระดับ

มีหน้าที่ในการปรับค่าความเข้มแสง ในแต่ละพิกเซล การประมวลผมภาพนั้นจะพิจารณาจากข้อมูลที่ผู้ใช้ป้อนให้กับวงจร โดยจะเปรียบเทียบค่าความเข้มแสงของพิกเซลนั้น ๆ เช่น ถ้าความเข้มของแสงพิกเซลนั้นมีค่าน้อยกว่า ค่าที่ผู้ใช้กำหนดพิกเซลนั้นจะแสดงค่า “00000000” หรือสีดำ แต่ถ้าค่าความเข้มของพิกเซลนั้นมีค่ามากกว่า ค่าที่ผู้ใช้กำหนดพิกเซลนั้นจะแสดงค่า “11111111” หรือสีขาว



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

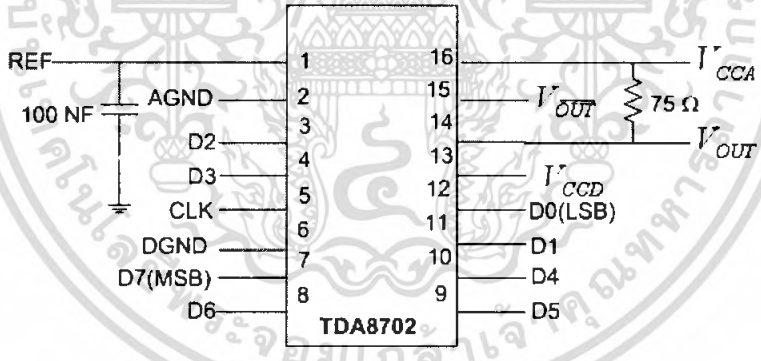


(ข)

รูปที่ 3.11 วงจรปรับความเข้มแสงของภาพเป็นสองระดับ

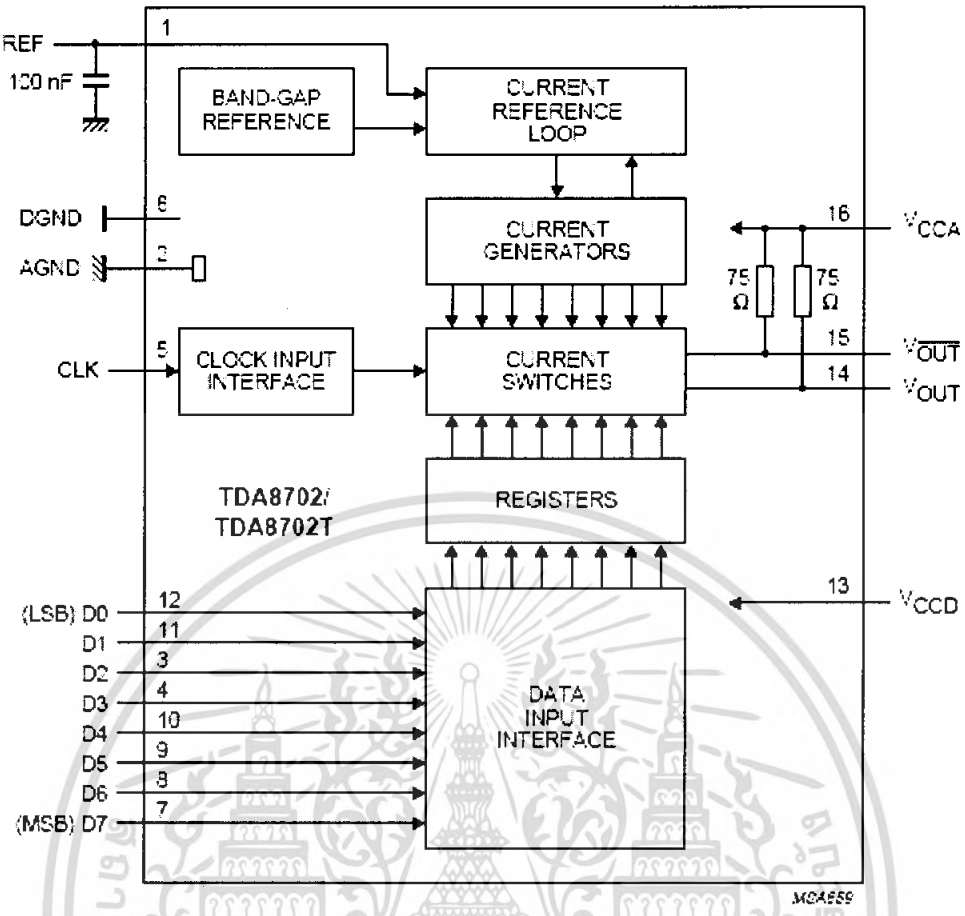
3.3 วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก

เป็นส่วนที่ทำหน้าที่แปลงสัญญาณภาพขนาด 8 บิต ซึ่งเป็นสัญญาณดิจิตอลให้เป็นสัญญาณอนาล็อก ซึ่งทำการแปลงระดับสัญญาณดิจิตอลในช่วง "00000000" ถึง "11111111" ให้ระดับแรงดันอยู่ในช่วง 0-0.8 โวลต์ โดยใช้ไอซีเบอร์ TDA8702 ซึ่งมีอัตราการแปลงข้อมูลสูงสุด 30 MHz แต่ในโครงงานนี้เราใช้สัญญาณนาฬิกาที่มีความถี่ 25.175 *MHz*

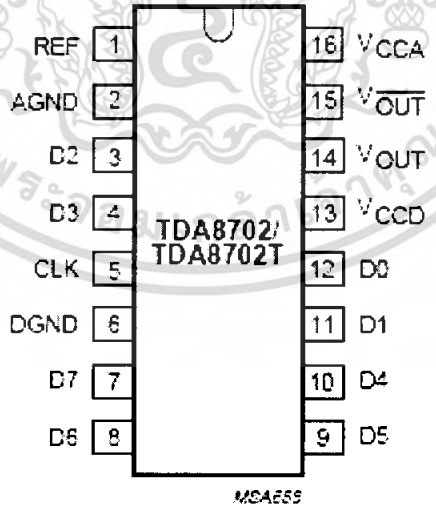


รูปที่ 3.12 DAC เบอร์ TDA8702

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 บล๊อคไดอะแกรมของ TDA8702



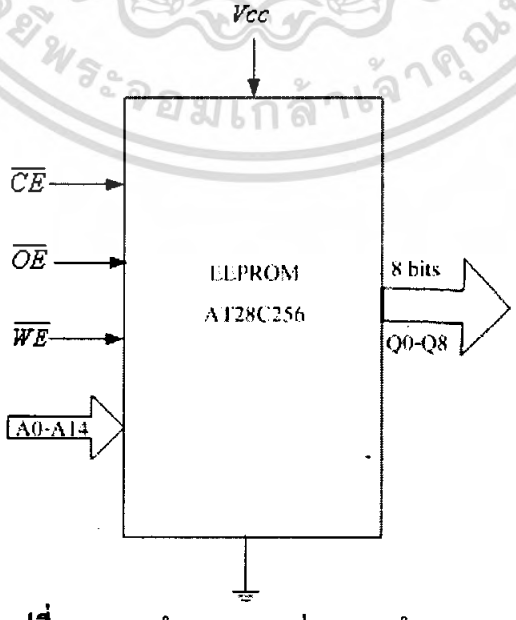
รูปที่ 3.14 ลักษณะของไอซีที่ใช้ในการแปลงสัญญาณดิจิตอลเป็นอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 การใช้งานของแต่ละขาของไอซี TDA8702

SYMBOL	PIN	DESCRIPTION
REF	1	voltage reference (decoupling)
AGND	2	analog ground
D2	3	data input; bit 2
D3	4	data input; bit 3
CLK	5	clock input
DGND	6	digital ground
D7	7	data input; bit 7
D6	8	data input; bit 6
D5	9	data input; bit 5
D4	10	data input; bit 4
D1	11	data input; bit 1
D0	12	data input; bit 0
VCCD	13	positive supply voltage for digital circuits (+5 V)
VOUT	14	analog voltage output
VOUT	15	complementary analog voltage output
VCCA	16	positive supply voltage for analog circuits (+5 V)

3.4 วงจรหน่วยความจำภายนอก



รูปที่ 3.15 การทำงานของหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.15 เป็นรูปของหน่วยความจำภายนอกจะอยู่ในสถานะ "Read" ก็ต่อเมื่อ ทำการป้อนสัญญาณ 'low' ให้กับ ขา $\overline{CE}$ (Chip Enable) และขา $\overline{OE}$ (Output Enable) และป้อนสัญญาณ 'high' ให้กับขา $\overline{WE}$ ซึ่งใช้เป็นเบอร์ 28c256 ดังรูป

รูปที่ 3.16 ลักษณะอีอีพรอมที่นำมาใช้งาน

ตารางที่ 3.4 ชื่อของตัวสัญญาณต่าง ๆ ที่ใช้งานในหน่วยความจำภายนอก

Name	Description
A0-A14	Address input
Q0-Q7	Data input
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
Vcc	Positive Power Supply
GND	Ground

รูปที่ 3.17 หน่วยความจำภายนอก เบอร์ 28C256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

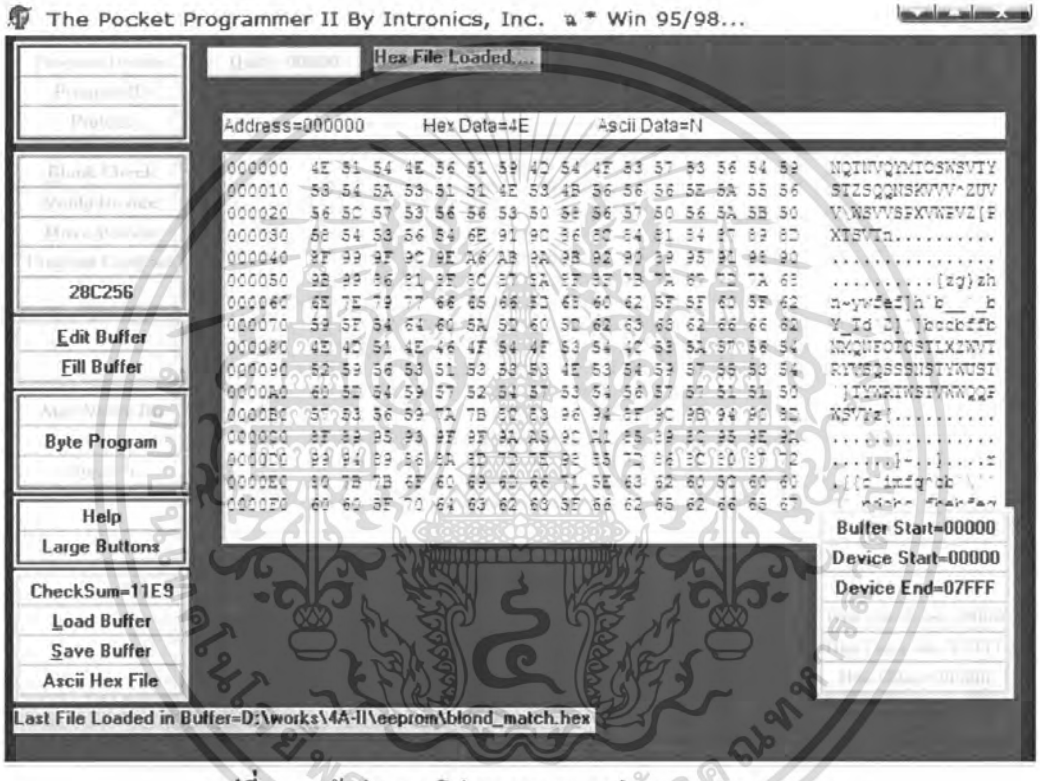
บทที่ 4

การทดลองและผลการทดลอง

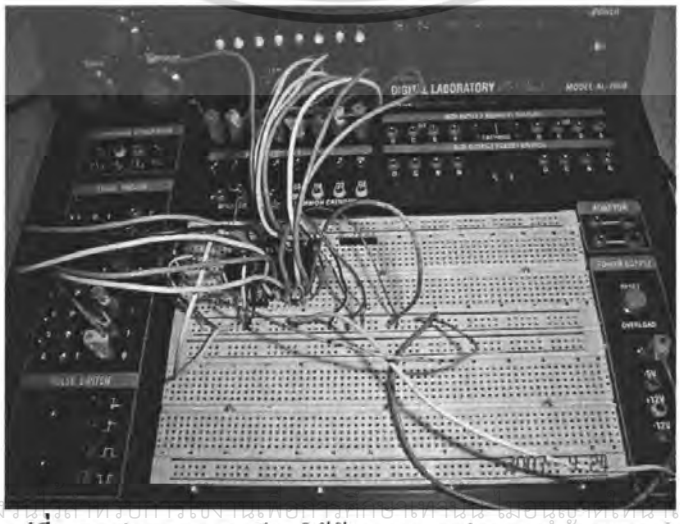
4.1 ขั้นตอนการทดลอง

4.1.1 การใช้งานหน่วยความจำภายนอก

ใช้หน่วยความจำภายนอก (EEPROM) เพื่อที่จะใช้เก็บข้อมูลแต่ละพิกเซลของภาพ ซึ่งในแต่ละพิกเซลนั้นจะมีขนาด 8 บิต โดยในโครงงานนี้เราจะใช้ภาพขนาด 128x128 พิกเซล



รูปที่ 4.1 หน้าต่างการโปรแกรมของหน่วยความจำภายนอก



รูปที่ 4.2 รูปการทดลองเรียกใช้ข้อมูลจากหน่วยความจำภายนอก

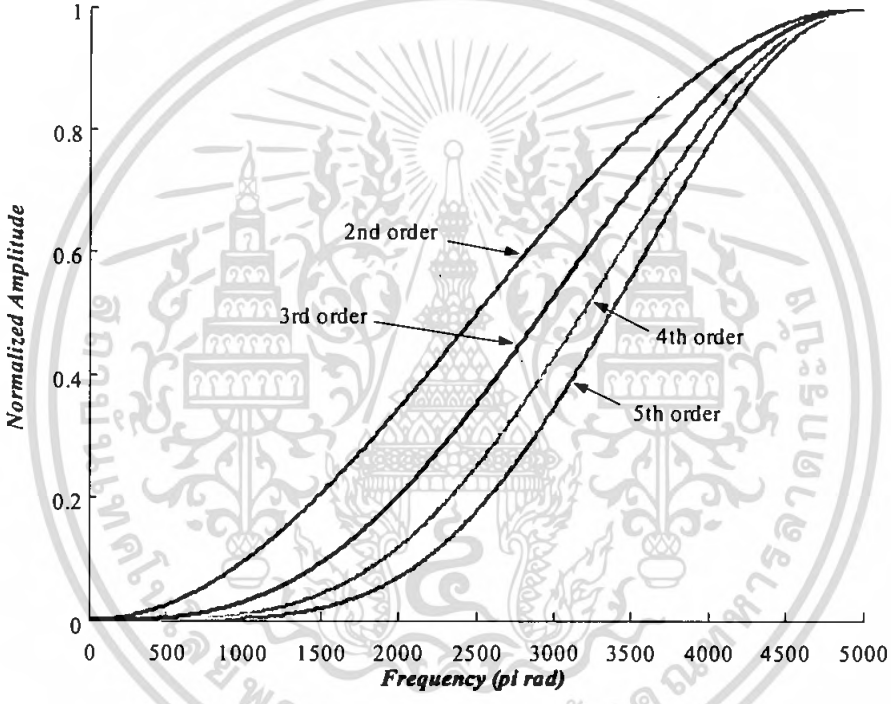
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4.2 เป็นการทดลองเรียกใช้ข้อมูลจากหน่วยความจำภายนอกโดยป้อนค่าที่ชี้ตำแหน่งเป็น "000000000000000" ให้กับขา A0-A14 (แอดเดรส) ซึ่งตำแหน่งนี้ทำการเก็บข้อมูลของภาพมีค่าเท่ากับ 4E ที่เป็นค่าในรูปแบบเลขฐานสิบหก ดังรูปที่ 4.1 ซึ่งผลที่ได้จากการทดลองพบว่าถูกต้อง

4.1.2 ผลที่ได้จากโปรแกรม MATLAB

4.1.2.1 ผลที่ได้จากการแปลงปาสคาลเต็มหน่วยและวงจรกรองปาสคาลใน 1 มิติ

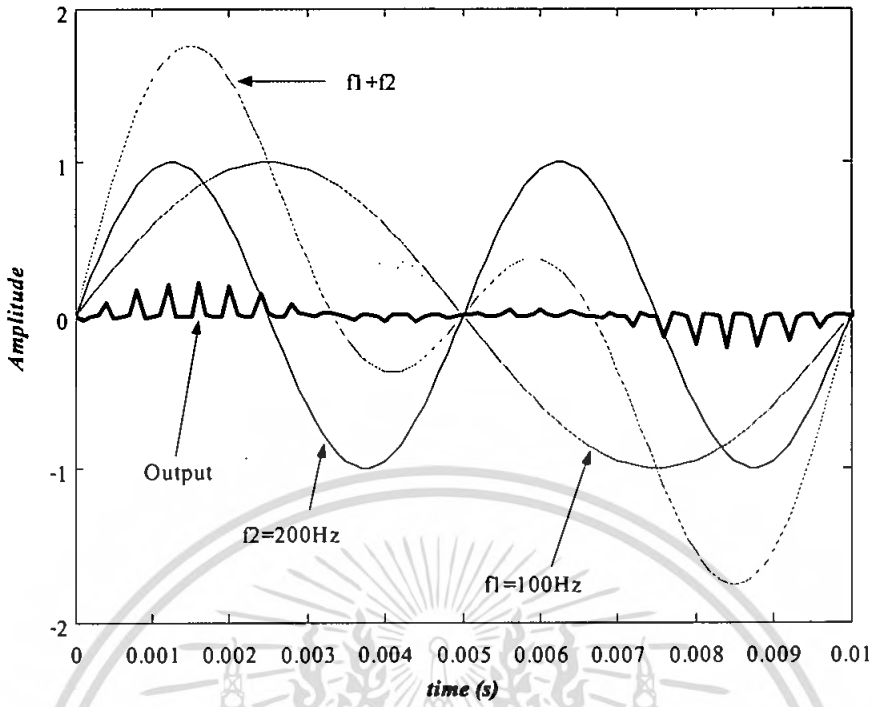
เราได้ทำการทดลองการแปลงปาสคาลแบบเต็มหน่วยและวงจรกรองปาสคาลใน 1 มิติ โดยทดลองวงจรกรองปาสคาลที่ 2^{nd} order ถึง 5^{th} order และได้พล็อตกราฟผลตอบสนองทางแอมปลิจูดและความถี่ ได้ดังรูป



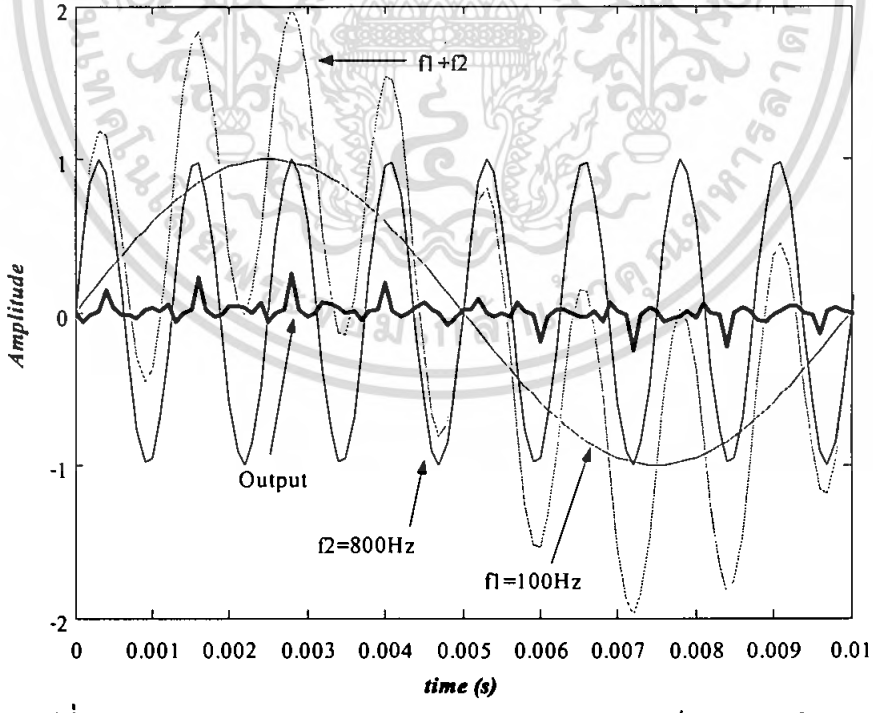
รูปที่ 4.3 แสดงผลตอบสนองทางขนาดของความเป็นตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า N^{th} order ตั้งแต่ 2^{nd} order ถึง 5^{th} order

การจำลองการทำงานนี้ได้สร้างสัญญาณไซน์ f1 และ f2 ที่มีความถี่ต่างกัน โดยนำสัญญาณทั้งสองมาบวกกันจะได้เป็นสัญญาณ f1+f2 สัญญาณเอาต์พุตเป็นสัญญาณที่ผ่านการแปลงปาสคาลเต็มหน่วยแล้ว ดังแสดงใน [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

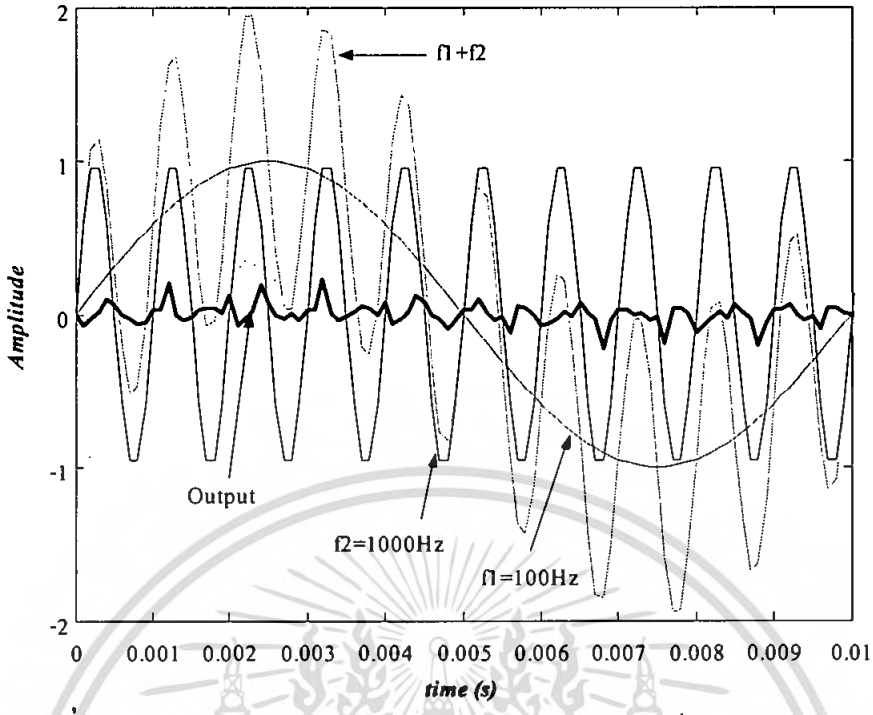


รูปที่ 4.4 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 200 Hz



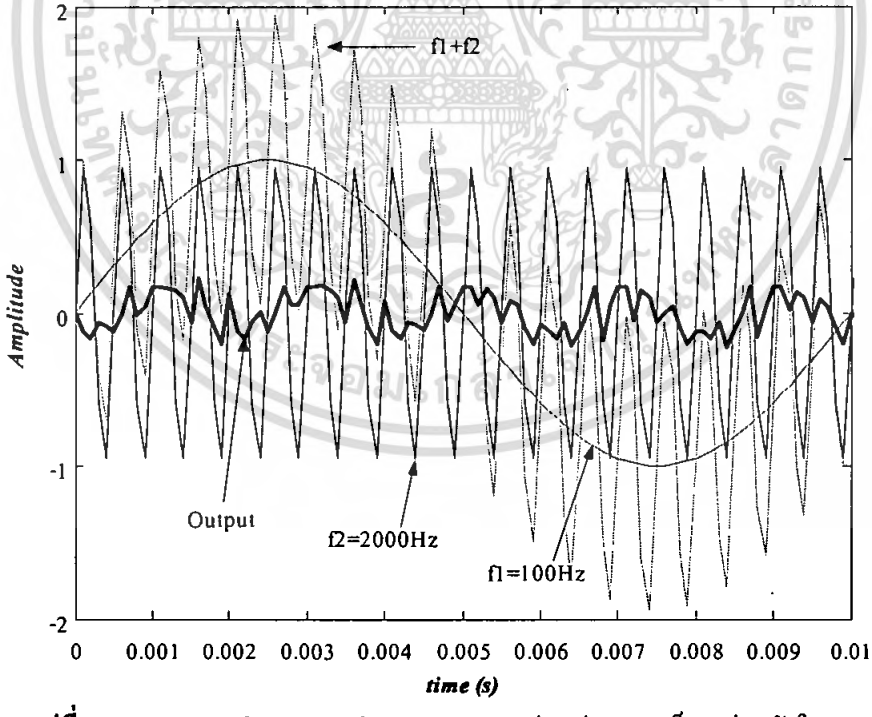
รูปที่ 4.5 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 800 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]

ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 1000 Hz

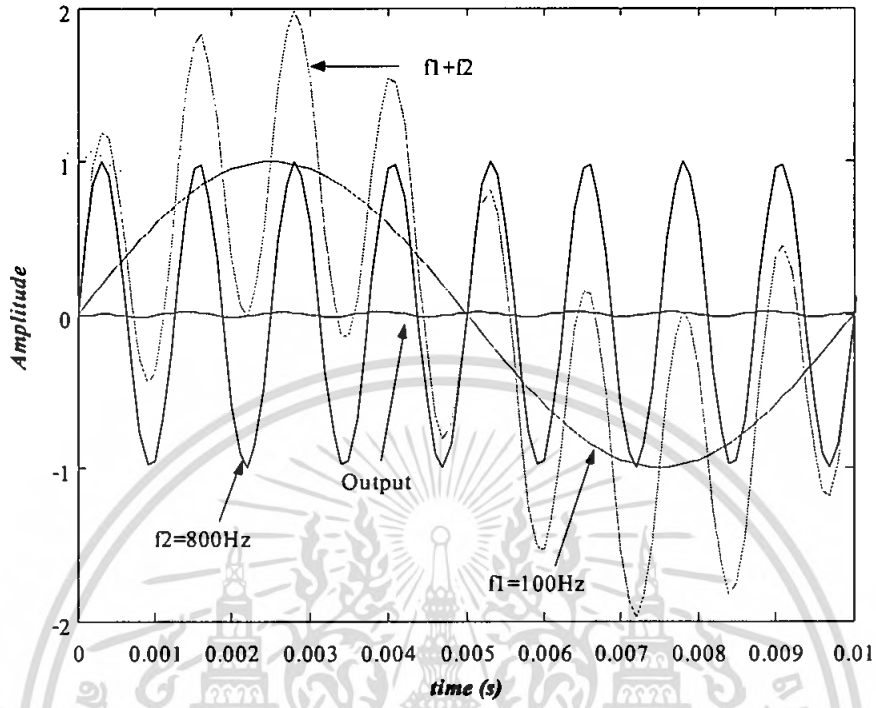


รูปที่ 4.7 ผลจากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยดังใน [1]

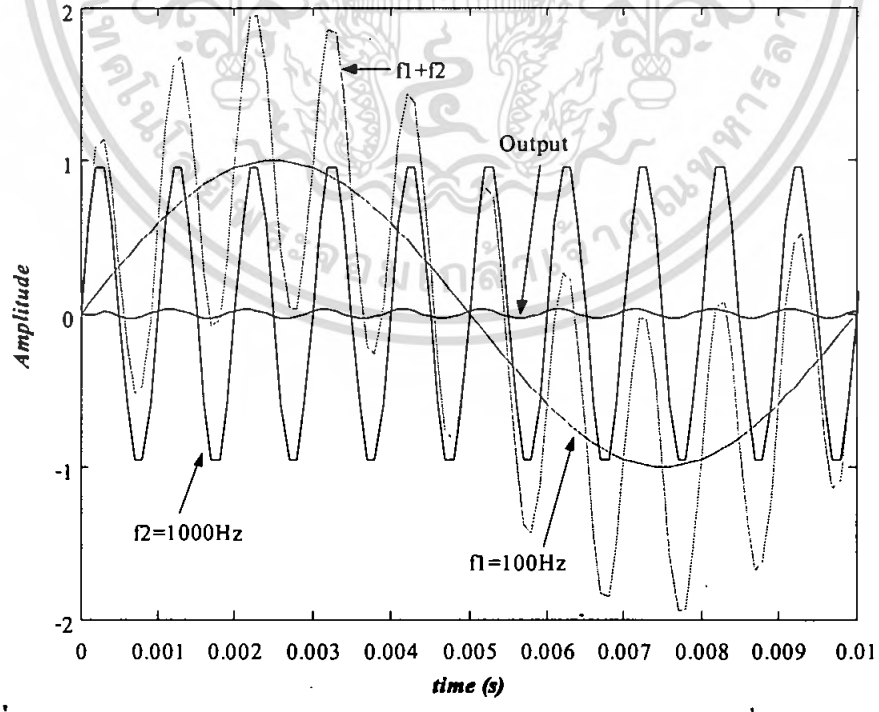
ขนาด 4x4 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 2000 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตได้ว่า การแปลงปาสคาลเต็มหน่วยจะไม่สามารถเห็นถึงคุณสมบัติของการกรองความถี่สูงผ่านได้ จากนั้นได้ทดลองนำสัญญาณ f1+f2 ผ่านวงจรกรองปาสคาล จะได้ผลดังนี้

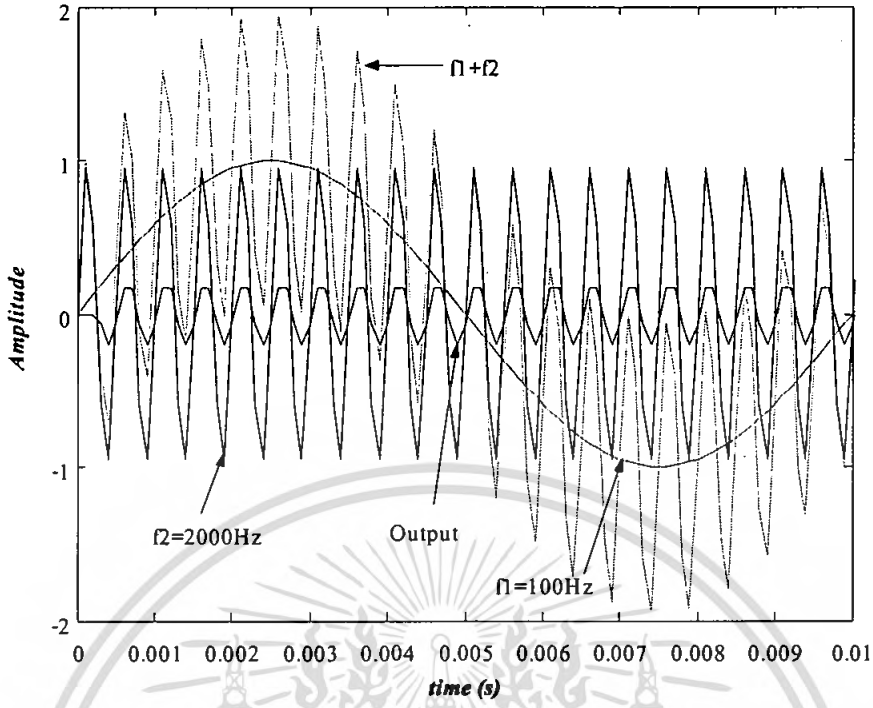


รูปที่ 4.8 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 800 Hz

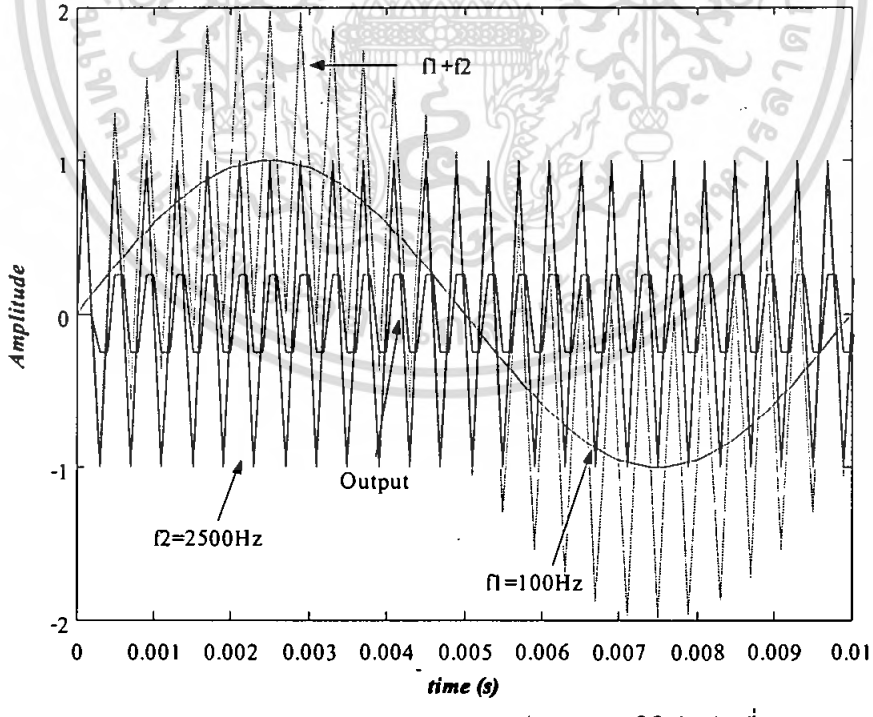


รูปที่ 4.9 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 1000 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 2000 Hz



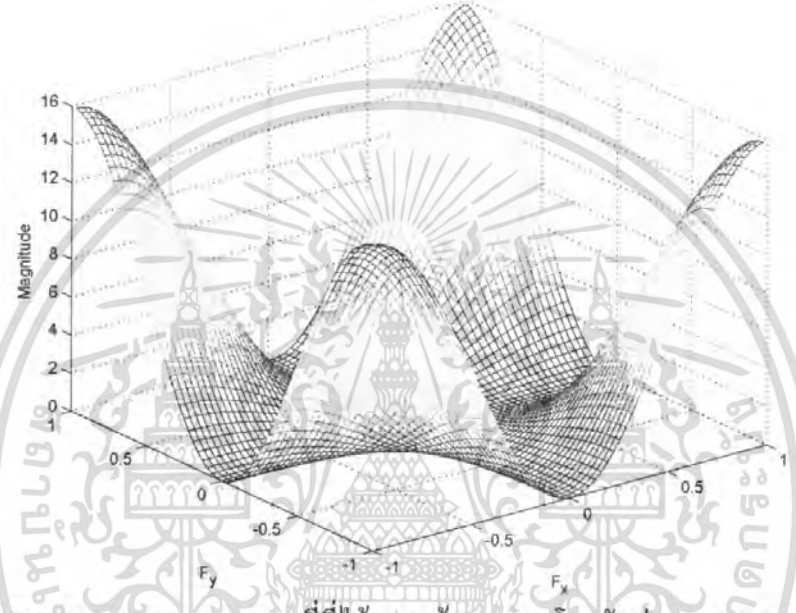
รูปที่ 4.11 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB เมื่อ f1 = 100 Hz และ f2 = 2500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

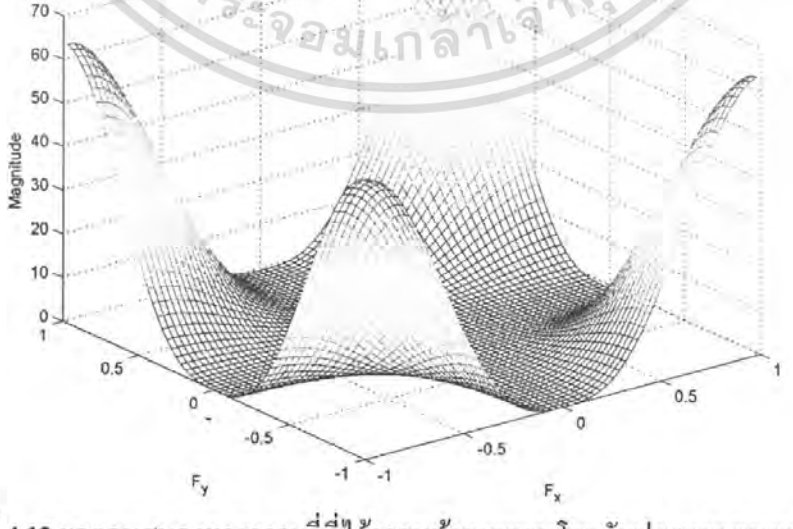
จากการทดลองพบว่าเมื่อนำสัญญาณที่มีองค์ประกอบความถี่สูงรวมกับความถี่ต่ำเมื่อผ่านปาสคาลฟิลเตอร์แล้ว สัญญาณเอาต์พุตที่ได้เป็นสัญญาณที่มีองค์ประกอบความถี่สูง นั่นก็คือสัญญาณความถี่สูงเท่านั้นที่สามารถผ่านออกมาได้ ซึ่งเป็นคุณสมบัติของตัวกรองความถี่สูงผ่าน

4.1.2.2 ผลตอบสนองทางความถี่ที่ได้จากวงจรกรองปาสคาล 2 มิติ

ในส่วนของวงจรกรองปาสคาล 2 มิตินี้ เราทำการแสดงผลตอบสนองทางความถี่ได้ดังรูป

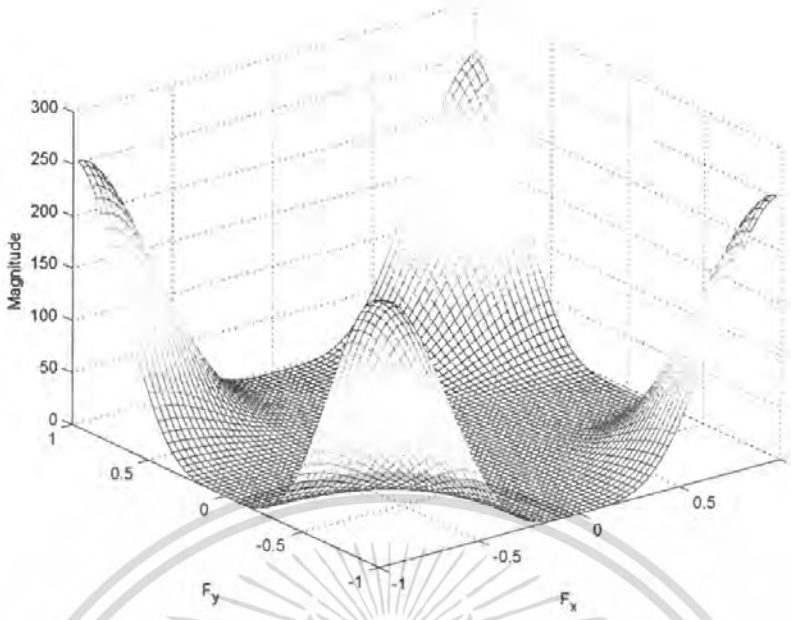


รูปที่ 4.12 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 3x3

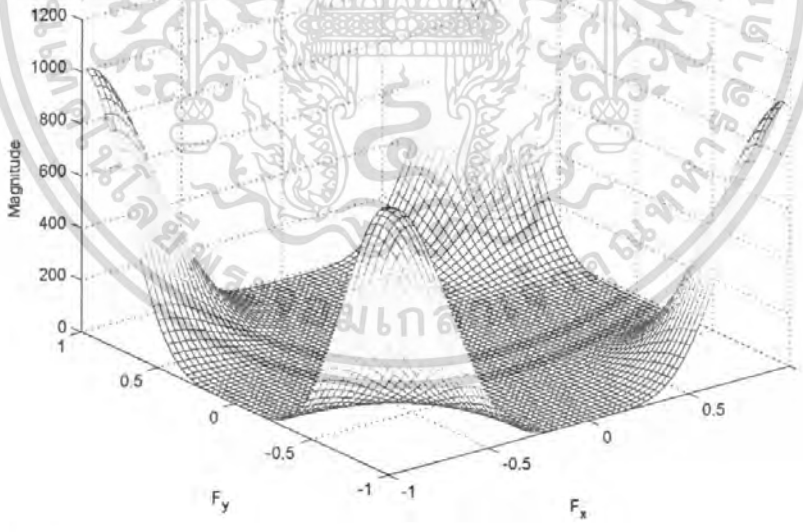


รูปที่ 4.13 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 4x4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



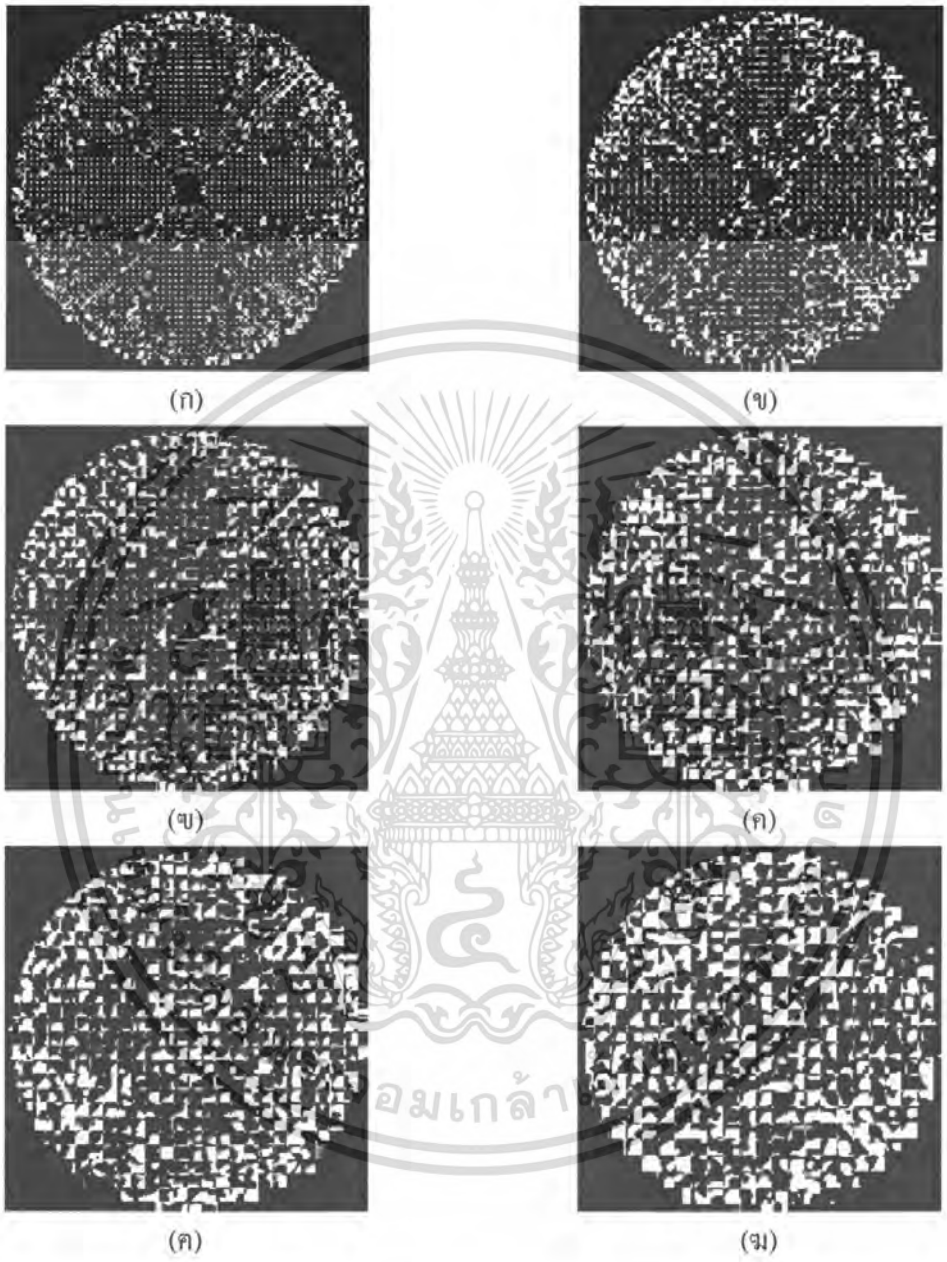
รูปที่ 4.14 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 5x5



รูปที่ 4.15 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสคาล ขนาด 6x6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการแปลงปาสคาลเต็มหน่วย 2 มิติ ดังใน [1] กับสัญญาณอินพุตที่เป็นภาพ จะได้ดังรูปที่ 4.16



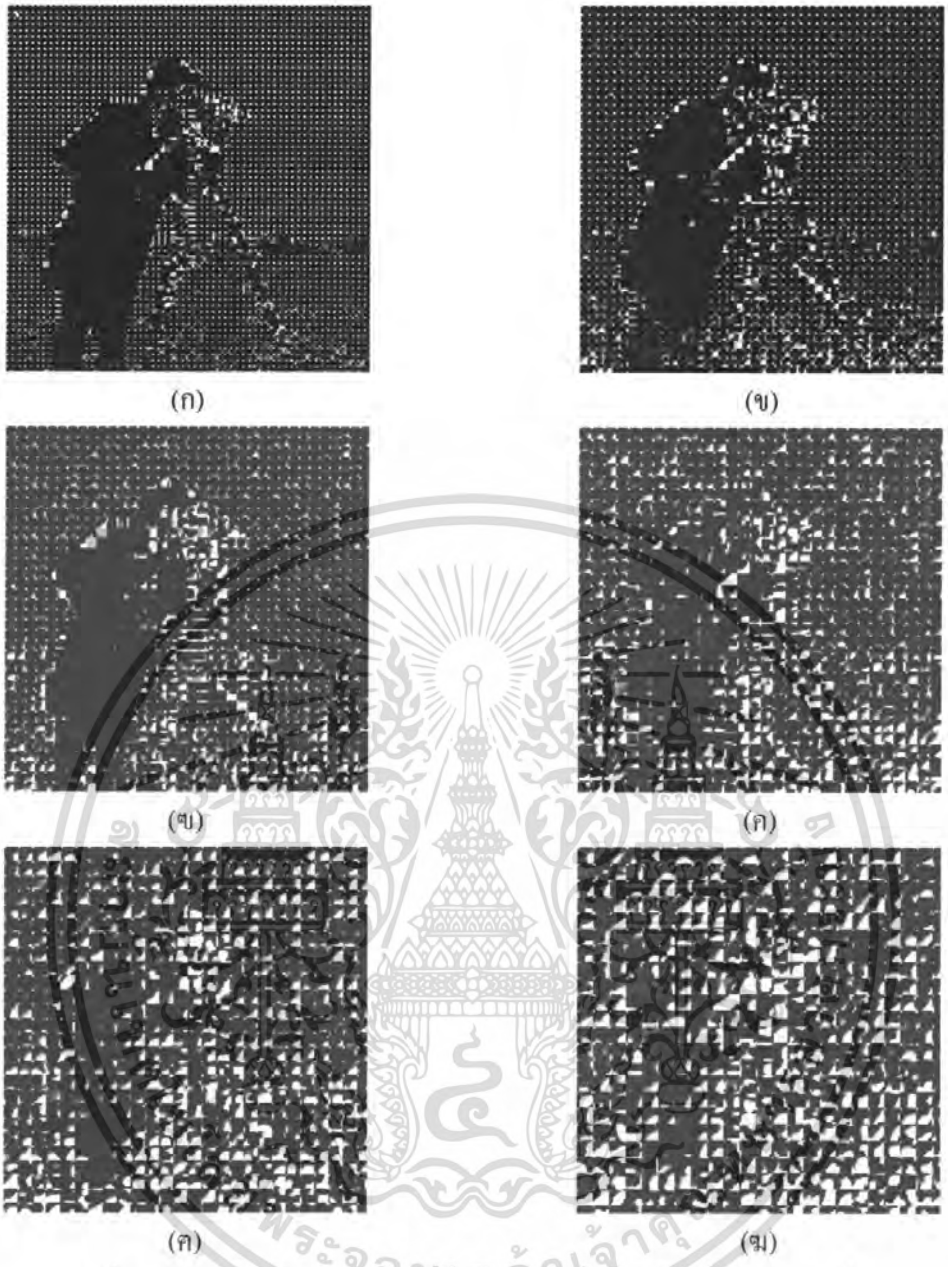
รูปที่ 4.16 ภาพที่ได้จากการแปลงปาสคาลแบบเต็มหน่วย ตัวอย่างที่ 1

(ก) ปาสคาลเมตริกซ์ขนาด 3x3 (ข) ปาสคาลเมตริกซ์ขนาด 4x4

(ฃ) ปาสคาลเมตริกซ์ขนาด 5x5 (ค) ปาสคาลเมตริกซ์ขนาด 6x6

(ฅ) ปาสคาลเมตริกซ์ขนาด 7x7 (ฆ) ปาสคาลเมตริกซ์ขนาด 8x8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

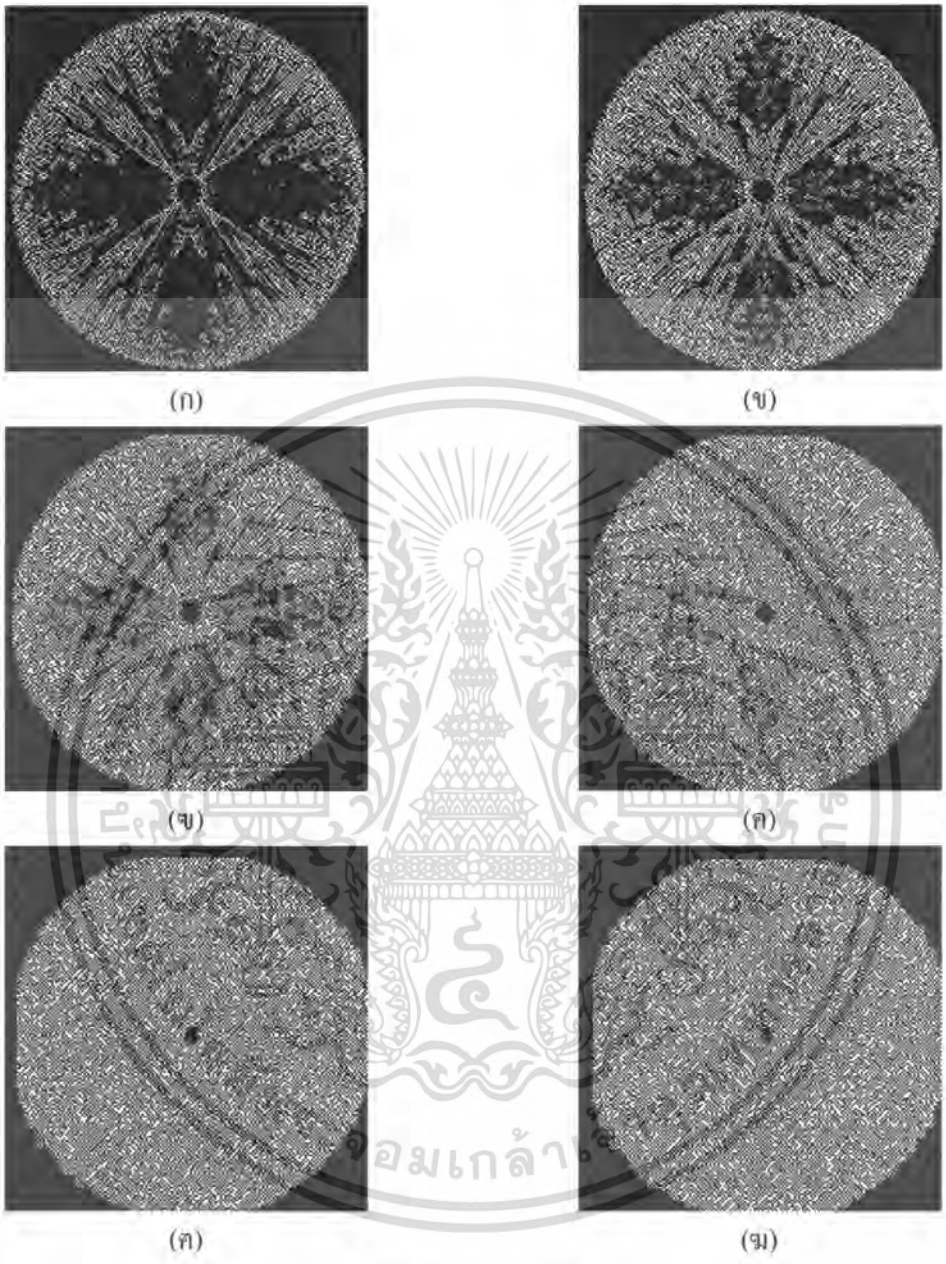


รูปที่ 4.17 ภาพที่ได้จากการแปลงดีสครีตปาสคาลแบบเต็มหน่วย ตัวอย่างที่ 2
(ก) หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 (ข) หน้ากากคอนโวลูชันปาสคาล ขนาด 4x4
(ฃ) หน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 (ค) หน้ากากคอนโวลูชันปาสคาล ขนาด 6x6
(ฅ) หน้ากากคอนโวลูชันปาสคาล ขนาด 7x7 (ฆ) หน้ากากคอนโวลูชันปาสคาล ขนาด 8x8

จากการจำลองการทำงานของการแปลงปาสคาลเต็มหน่วยในปาสคาลเมตริกซ์ขนาดต่าง ๆ นั้น พบว่าไม่สามารถเน้นขอบภาพได้

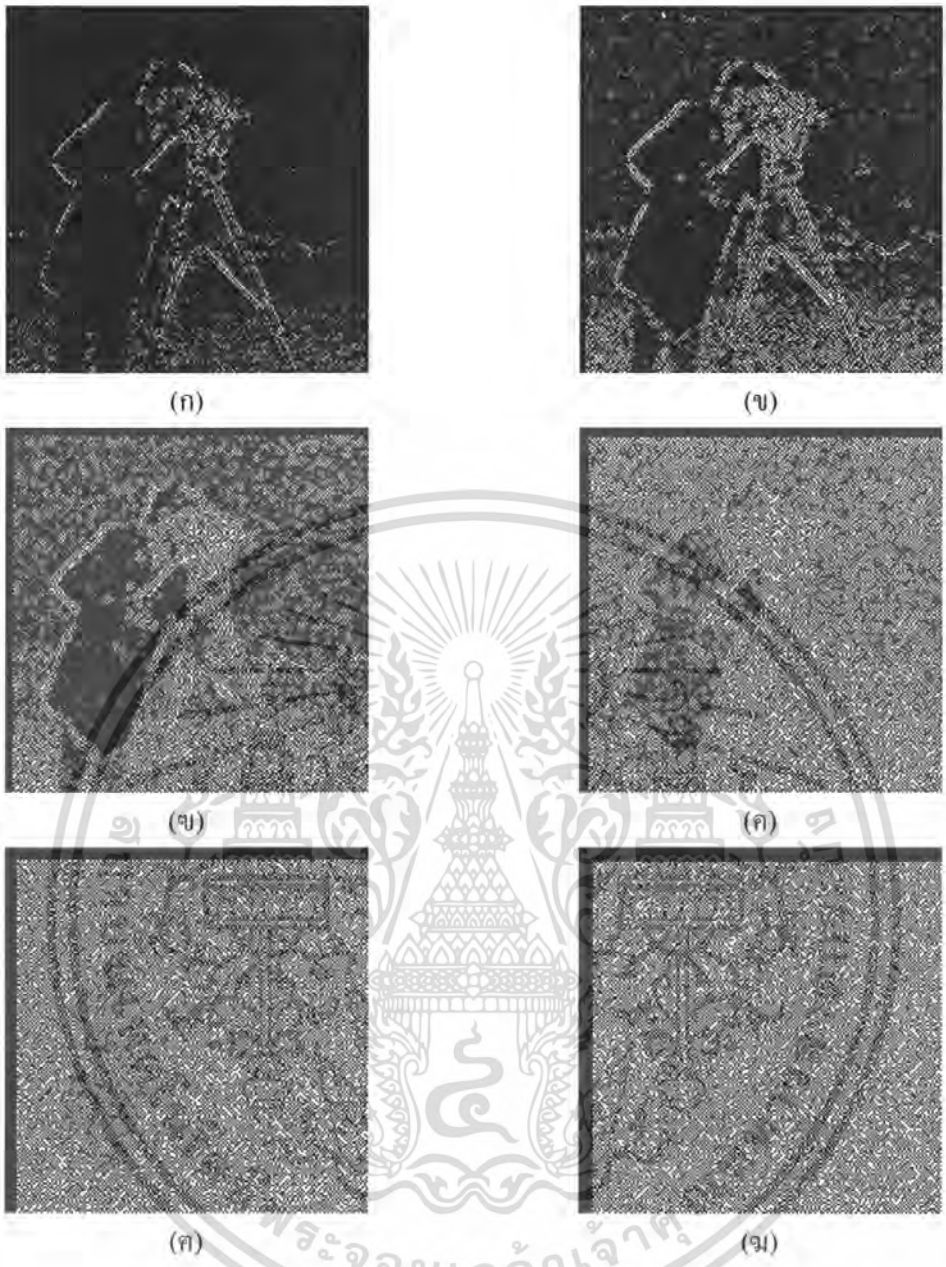
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทดลองนำภาพมาทำการประมวลผลโดยผ่านวงจรกรองปาสคาล 2 มิติ



รูปที่ 4.18 ภาพที่ได้จากวงจรกรองปาสคาล 2 มิติ ตัวอย่างที่ 1
(ก) หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 (ข) หน้ากากคอนโวลูชันปาสคาล ขนาด 4x4
(ฆ) หน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 (ค) หน้ากากคอนโวลูชันปาสคาล ขนาด 6x6
(ง) หน้ากากคอนโวลูชันปาสคาล ขนาด 7x7 (จ) หน้ากากคอนโวลูชันปาสคาล ขนาด 8x8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 การแปลงปาสคาลฟิลเตอร์ ตัวอย่างที่ 2

(ก) หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 (ข) หน้ากากคอนโวลูชันปาสคาล ขนาด 4x4
(ฃ) หน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 (ค) หน้ากากคอนโวลูชันปาสคาล ขนาด 6x6
(ฅ) หน้ากากคอนโวลูชันปาสคาล ขนาด 7x7 (ฆ) หน้ากากคอนโวลูชันปาสคาล ขนาด 8x8

จะเห็นได้ว่าการกรองปาสคาล 2 มิติ โดยใช้หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 นั้นจะให้ผลลัพธ์ออกมาดีที่สุดและโครงสร้างของวงจรกรองเมื่อทำการโปรแกรมลงบนชิพเอฟพีจีเอจะใช้อุปกรณ์น้อยกว่าหน้ากากคอนโวลูชันปาสคาลขนาดอื่น ๆ

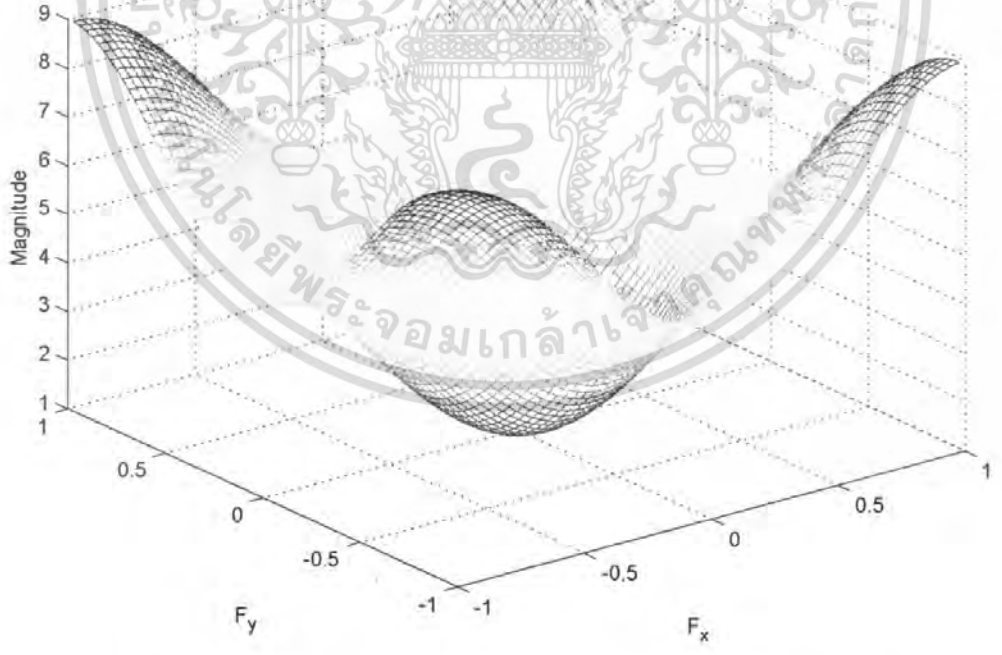
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2.1 เปรียบเทียบผลที่ได้จากวงจรกรองปาสคาลกับวงจรกรองความถี่สูงผ่านทั่ว ๆ ไป

ตัวกรองความถี่สูงผ่าน ความหมายโดยทั่วไปในการประมวลผลภาพเป็นการเน้นส่วนที่เป็นความถี่สูงหรือระดับความแตกต่างความเข้มของแสงมาก ขณะที่ลดส่วนที่เป็นความถี่ต่ำหรือระดับความแตกต่างความเข้มของแสงน้อย เพราะว่าขอบ (Edge) หรือรายละเอียดของภาพเป็นส่วนของความถี่สูง ดังนั้นการกรองความถี่สูงผ่านจะเป็นการเพิ่มความแตกต่างของแสงแต่ละที่ (Local Contrast) และความคมชัด (Shape) ของภาพ ตัวอย่างผลตอบสนองทางอิมพัลส์ (Impulse Response) ของตัวกรองความถี่สูงผ่านที่ใช้ปรับความแตกต่างของแสงให้ชัดเจนขึ้น

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

รูปที่ 4.20 ผลตอบสนองทางอิมพัลส์ของตัวกรองความถี่สูงผ่าน



รูปที่ 4.21 ผลตอบสนองทางความถี่ของตัวกรองความถี่สูงผ่านของรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

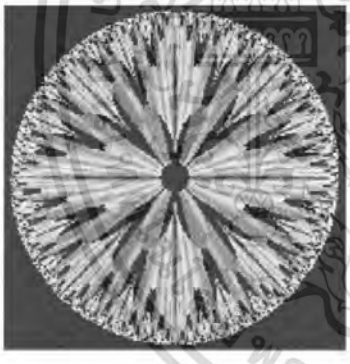
(ก)

(ข)

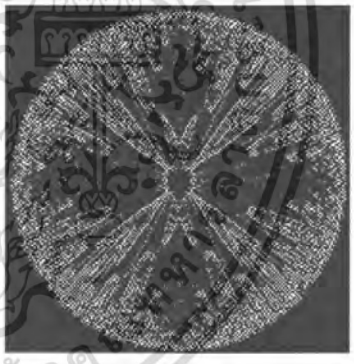
(ฆ)

(ค)

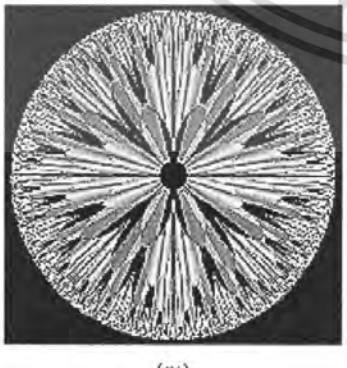
ตัวอย่างที่ 1



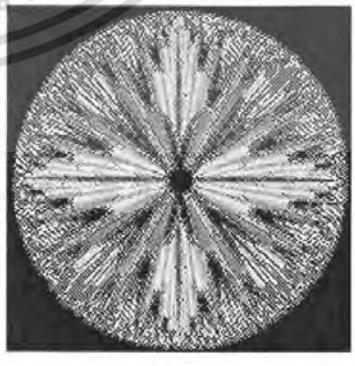
(ก)



(ข)



(ฆ)



(ค)

ตัวอย่างที่ 2

รูปที่ 4.22 เปรียบเทียบวงจรกรองความถี่สูงผ่านทั่วไปกับวงจรกรองปาสคาล

(ก) รูปต้นแบบ (ข) รูปที่ผ่านวงจรกรองปาสคาล

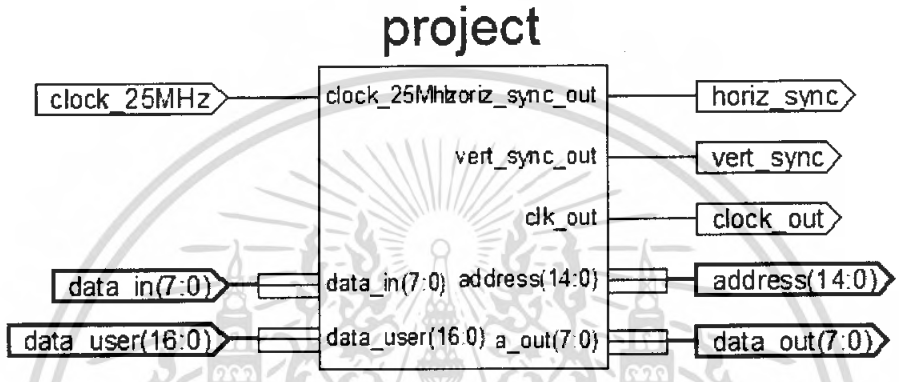
(ฆ) รูปที่ผ่านวงจรตัวกรองความถี่สูงผ่าน (ค) รูปต้นแบบบวกกับรูปที่ผ่านวงจรกรองปาสคาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.22 (ฆ) เป็นการจำลองการทำงานตัวกรองความถี่สูงผ่าน จะเห็นได้ว่ารูปที่ได้นั้นเป็นรูปที่คงค่าคล้ายเดิมไว้ และขอบภาพมีการเน้นชัดขึ้น ซึ่งแตกต่างจากการผ่านวงจรกรองปาสคาลที่กรองออกมาได้เป็นขอบภาพเท่านั้น ตามรูปที่ 4.22 (ข) จึงทดลองนำรูปต้นแบบบวกกับรูปที่ได้จากการผ่านวงจรกรองปาสคาล จะได้รูปที่ 4.22 (ค)

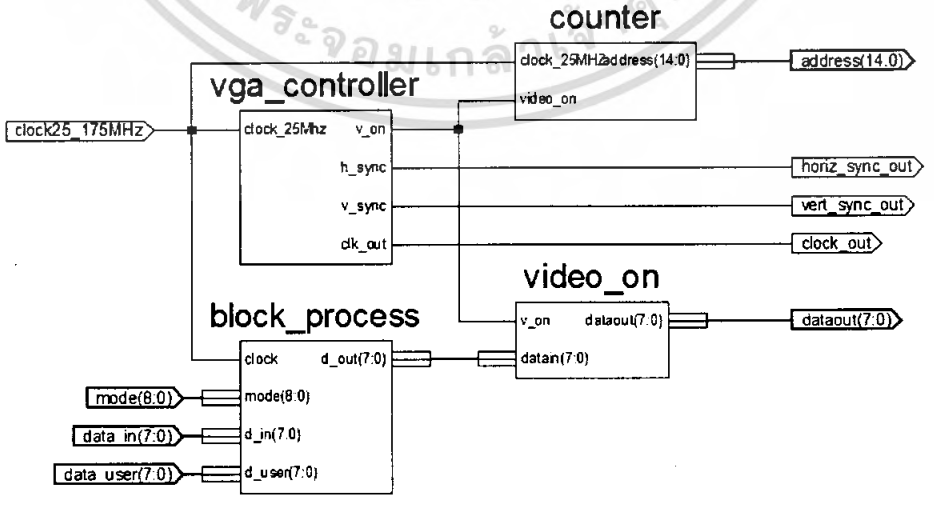
4.1.3 การออกแบบวงจรส่วนต่าง ๆ โดยใช้ภาษาวีเอชดีแอล

4.1.3.1 วงจรส่วนของตัวควบคุมวีจีเอ



รูปที่ 4.23 สัญลักษณ์ส่วนของตัวควบคุมวีจีเอ

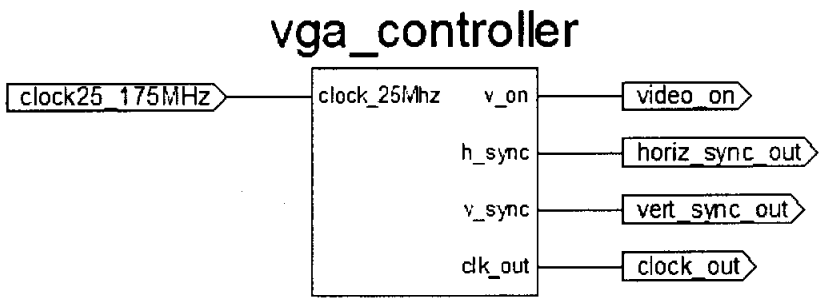
จากรูปที่ 4.23 เป็นโมดูลของวงจรที่ใช้ในการควบคุมการแสดงผลของหน้าจอวีจีเอ และการประมวลผลทั้งหมด ซึ่งภายในจะประกอบด้วย 4 วงจรหลัก ๆ ได้แก่ วงจรในส่วนของการสร้างสัญญาณควบคุมการแสกนทางแนวตั้งและแนวนอนให้กับวีจีเอ, วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอ, วงจรประมวลผลภาพ และวงจรสร้างตำแหน่งของข้อมูล ดังรูปที่ 4.24



รูปที่ 4.24 สัญลักษณ์ของวงจรภายในทั้งหมดที่ใช้ในการควบคุมวีจีเอ

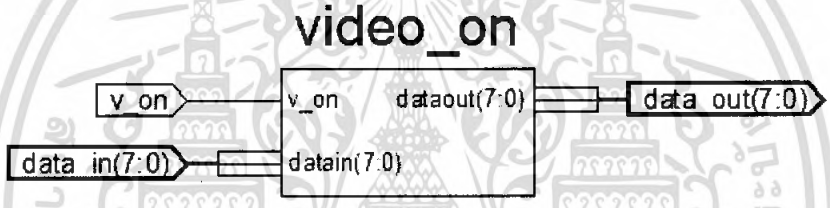
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3.2 วงจรในส่วนของการสร้างสัญญาณควบคุมการแสกนทางแนวตั้งและแนวนอนให้กับวีจีเอ



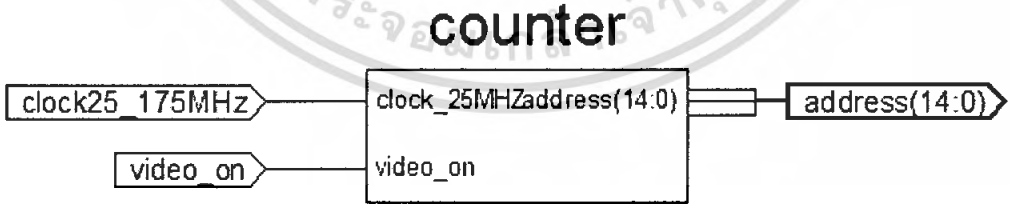
รูปที่ 4.25 สัญลักษณ์ส่วนของภาคขับสัญญาณ

4.1.3.3 วงจรส่วนของภาครับของหน่วยความจำภายนอก



รูปที่ 4.26 สัญลักษณ์ส่วนของภาครับของหน่วยความจำภายนอก

4.1.3.4 วงจรสร้างตำแหน่งของข้อมูล

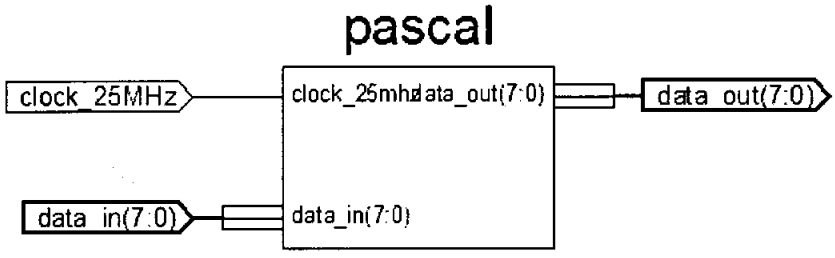


รูปที่ 4.27 สัญลักษณ์ส่วนของวงจรสร้างตำแหน่งของข้อมูล

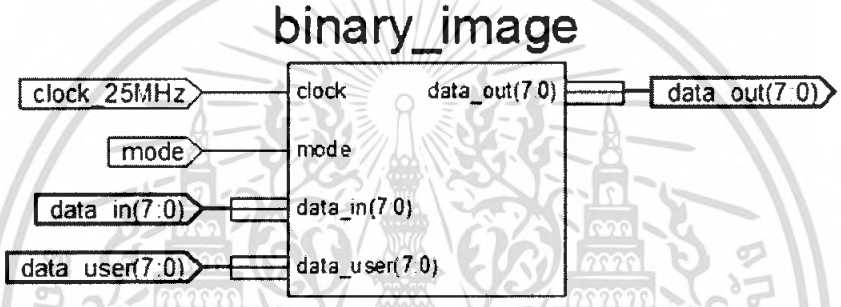
จากรูปที่ 4.27 จะเป็นวงจรสร้างตำแหน่งของข้อมูล ซึ่งจะอาศัยวงจรนับ (Counter) มาช่วยในการออกแบบวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

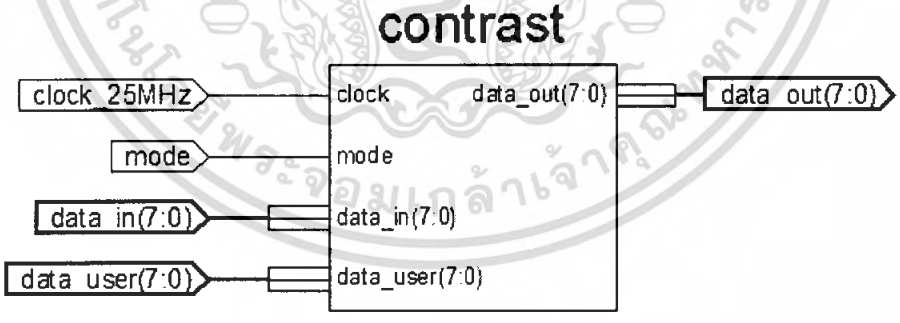
4.1.3.5 วงจรที่ใช้ในการประมวลผลภาพ



รูปที่ 4.28 สัญลักษณ์ส่วนของวงจรที่ใช้ในการประมวลผลภาพโดยอาศัยการแปลงปาสคาล

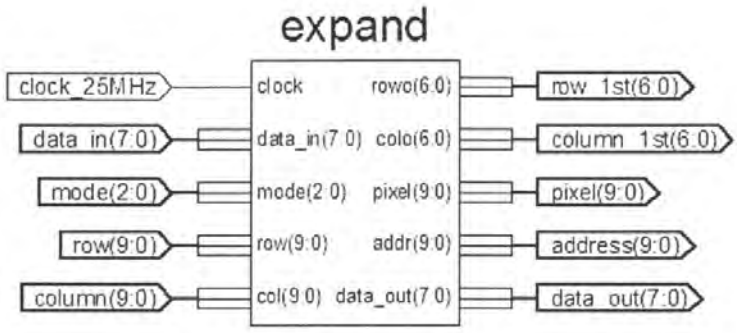


รูปที่ 4.29 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพเป็นสองระดับ



รูปที่ 4.30 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 สัญลักษณ์ส่วนของวงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ

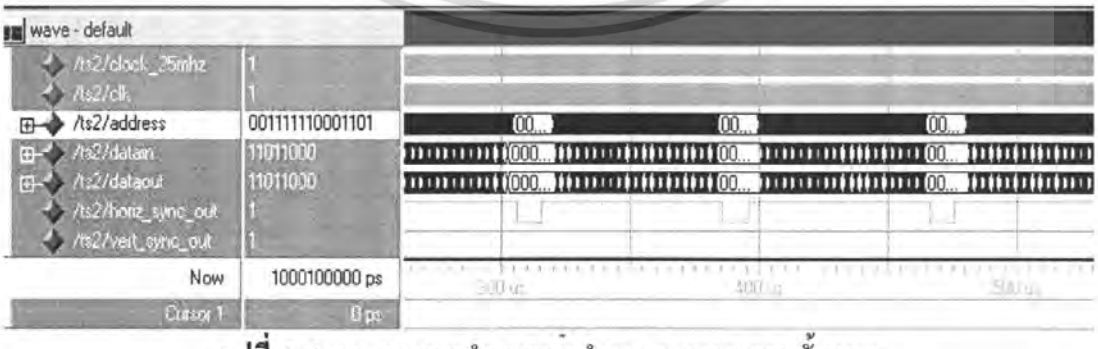
4.2 ผลการทดลอง

4.2.1 ผลการทดลองจากการจำลองการทำงานในวีเอชดีแอล

4.2.1.1 เมื่อทำการนำภาพที่ยังไม่ได้ทำการประมวลผลมาแสดงบนจอวีจีเอ

4.2.1.1.1 การทำงานของวงจรรวมทั้งหมด

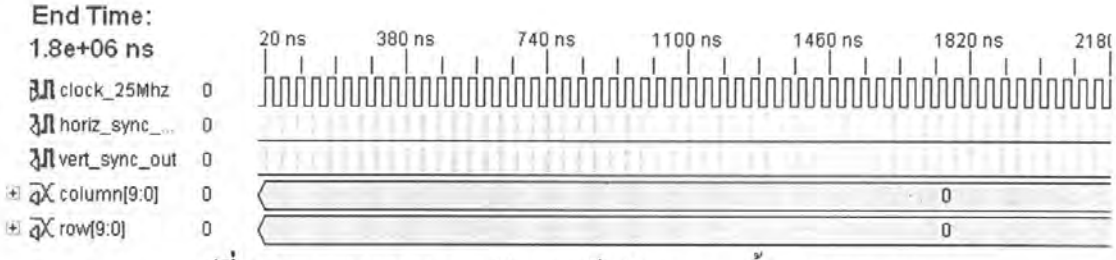
รูปที่ 4.32 ระบบทดสอบ (Testbench) ของวงจรรวมทั้งหมด



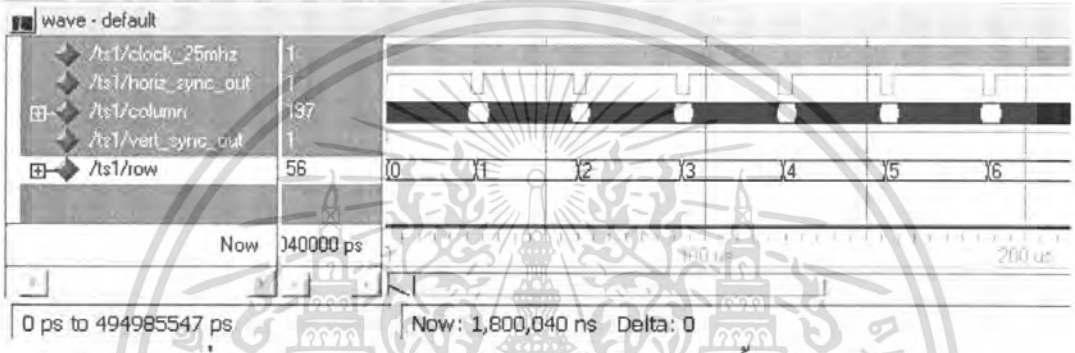
รูปที่ 4.33 แสดงผลการจำลองการทำงานของวงจรรวมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.1.2 การสร้างสัญญาณซิงโครไนซ์และสัญญาณสี

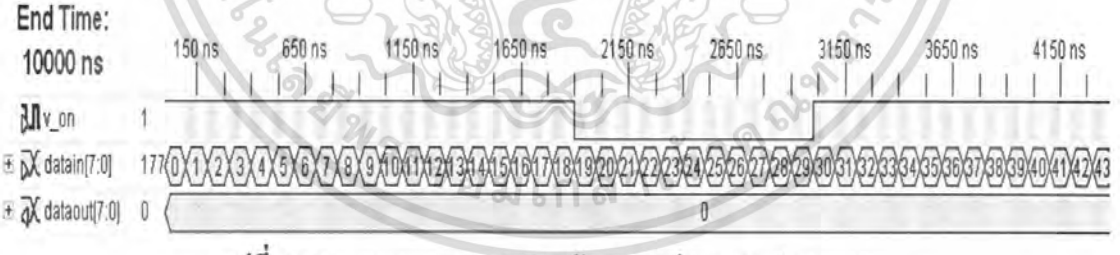


รูปที่ 4.34 ระบบทดสอบของสัญญาณอ้างอิงทางแนวตั้งและแนวนอน

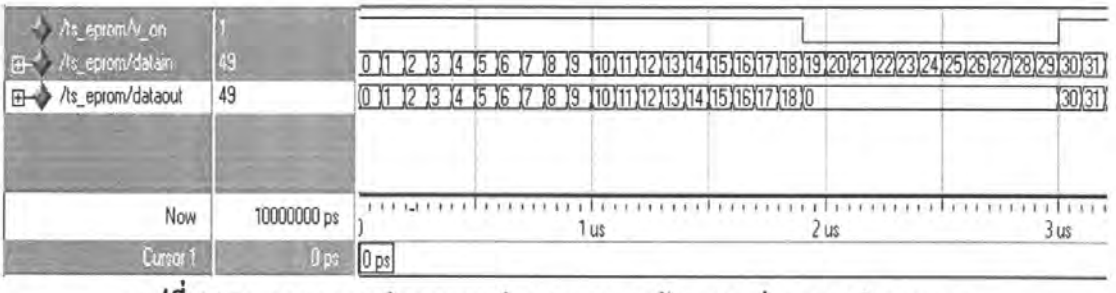


รูปที่ 4.35 แสดงผลการจำลองสัญญาณอ้างอิงทางแนวตั้งและแนวนอน

4.2.1.1.3 การทำงานของภาครับของหน่วยความจำภายนอก



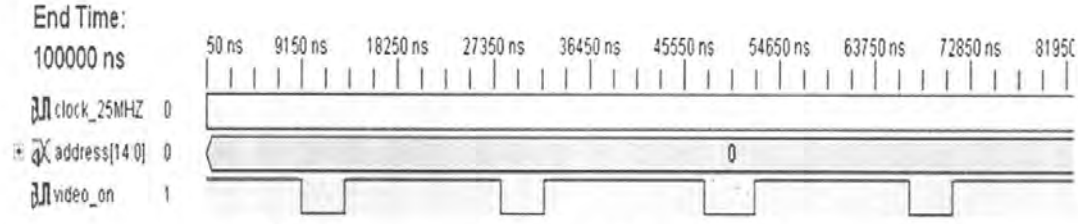
รูปที่ 4.36 ระบบทดสอบของภาครับของหน่วยความจำภายนอก



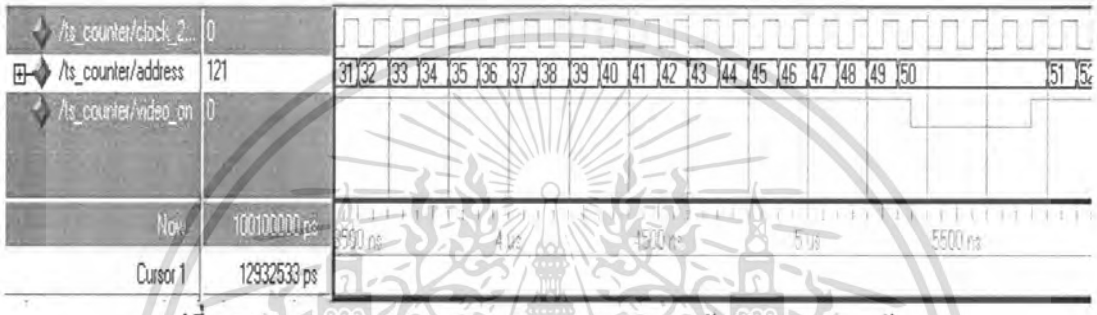
รูปที่ 4.37 แสดงผลการจำลองการทำงานของภาครับของหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.1.4 การทำงานของวงจรสร้างตำแหน่งของข้อมูล



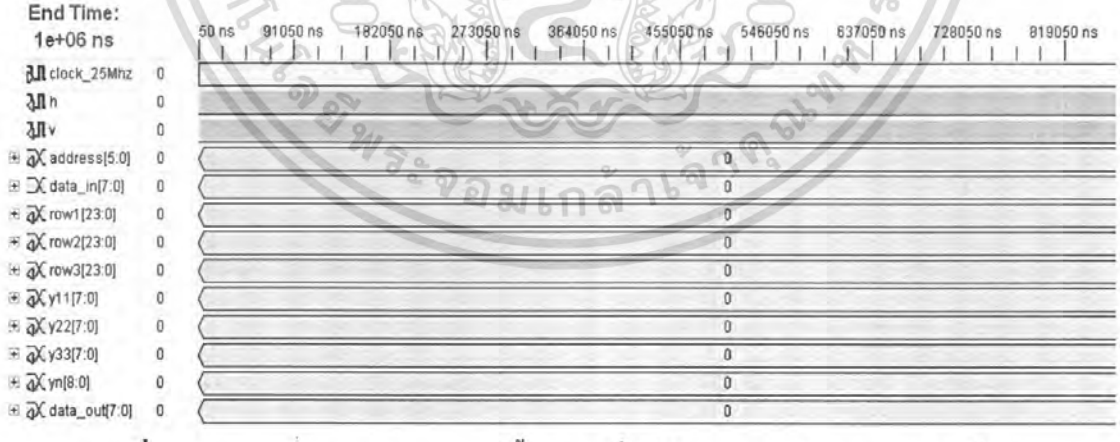
รูปที่ 4.38 ระบบทดสอบของวงจรสร้างตำแหน่งของข้อมูล



รูปที่ 4.39 แสดงผลการจำลองการทำงานของวงจรสร้างตำแหน่งของข้อมูล

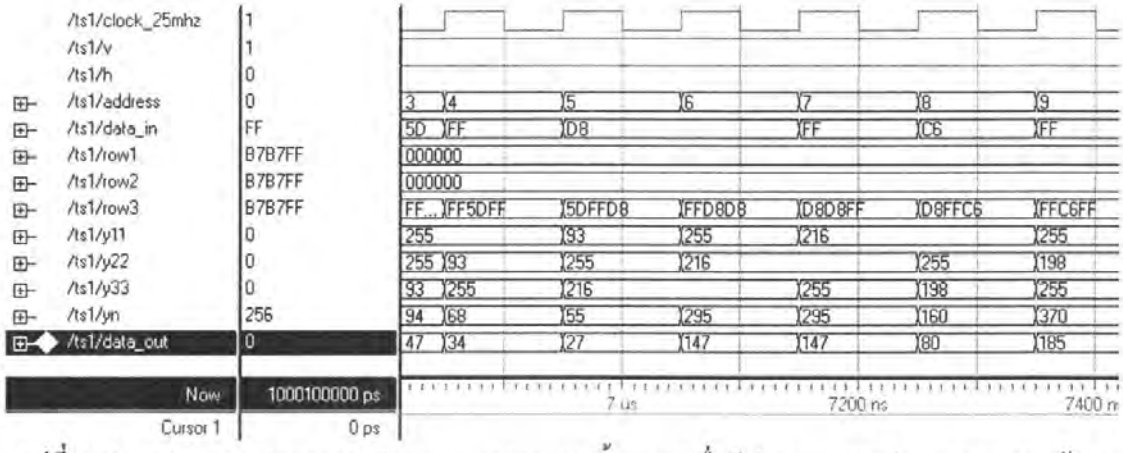
4.2.1.2 เมื่อทำการนำภาพที่ผ่านการประมวลผลเรียบร้อยแล้วมาแสดงบนจอวีจีเอ

4.2.1.2.1 การทำงานของวงจรรวมทั้งหมด

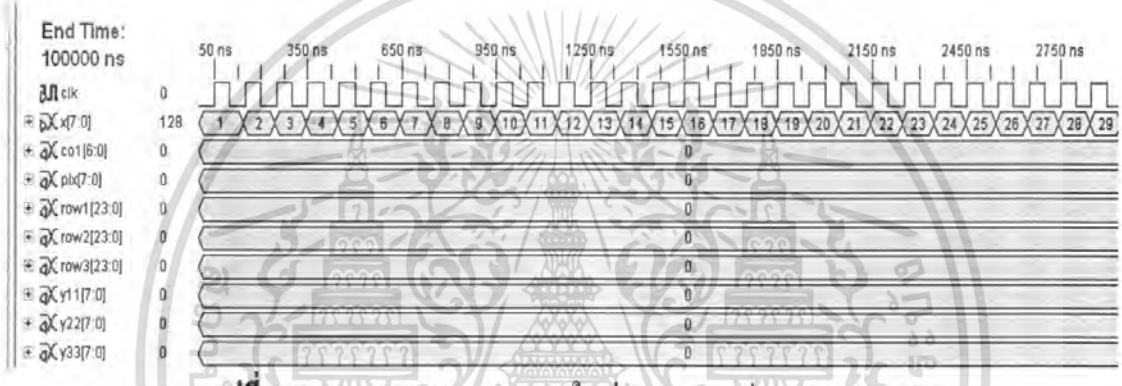


รูปที่ 4.40 ระบบทดสอบของวงจรรวมทั้งหมด (เมื่อมีส่วนของการประมวลผลภาพด้วย)

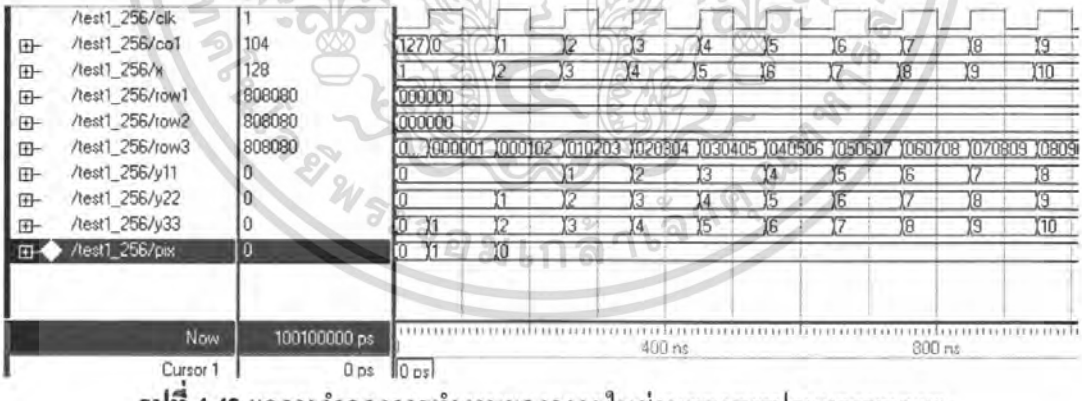
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 แสดงผลการจำลองการทำงานของวงจรรวมทั้งหมด (เมื่อมีส่วนของการประมวลผลภาพด้วย)



รูปที่ 4.42 ระบบทดสอบของวงจรในส่วนของการประมวลผลภาพ



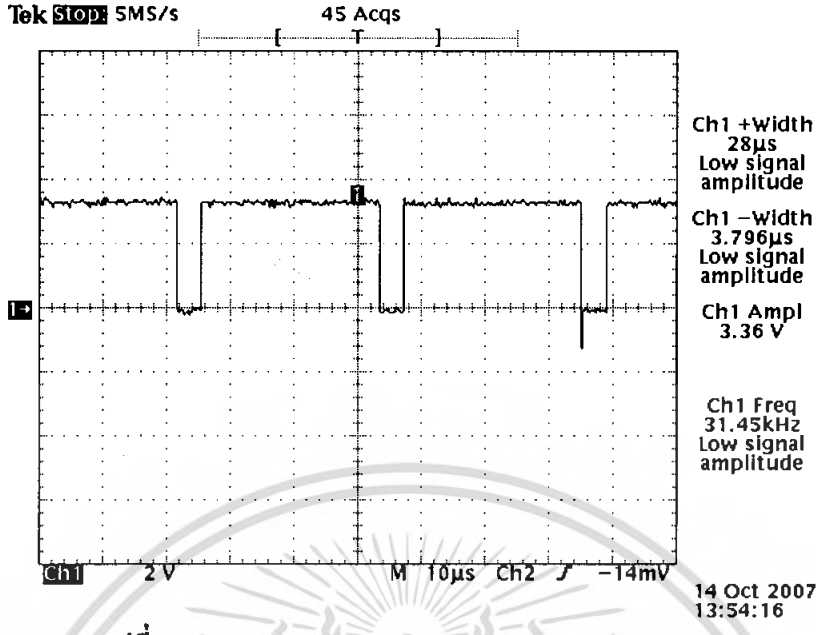
รูปที่ 4.43 ผลการจำลองการทำงานของวงจรในส่วนของการประมวลผลภาพ

4.2.2 ผลการทดลองที่ได้จากออสซิลโลสโคป

วัดสัญญาณจริงที่ได้จากการจำลองสัญญาณ โดยใช้ออสซิลโลสโคปมาวัดสัญญาณควบคุมการสแกนทางแนวนอน และสัญญาณควบคุมการสแกนทางแนวตั้ง ได้ดังนี้

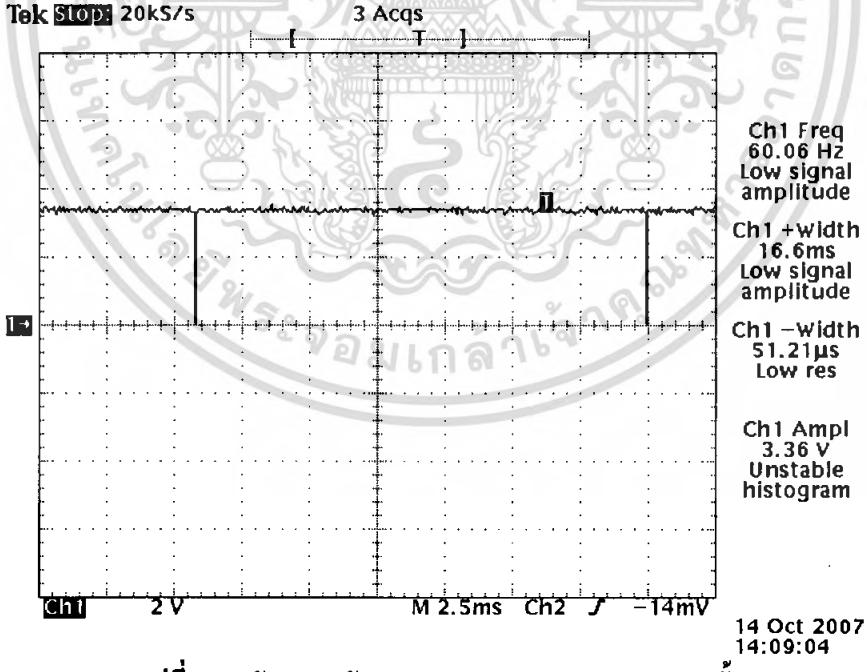
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 สัญญาณควบคุมการสแกนหน้าจอทางแนวนอนและแนวตั้ง



รูปที่ 4.44 สัญญาณสัญญาณควบคุมการสแกนทางแนวนอน

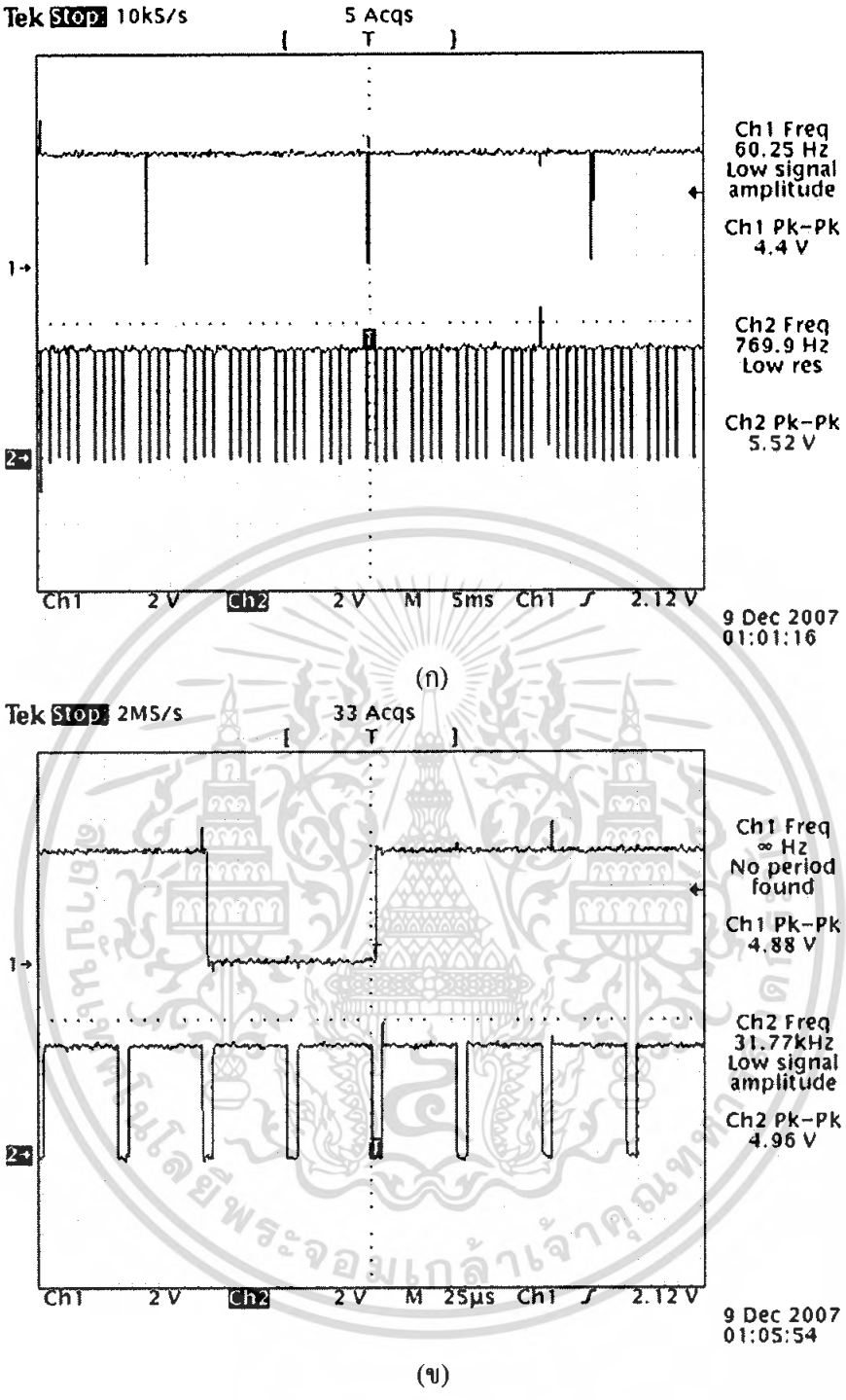
จากรูปที่ 4.44 เป็นการวัดสัญญาณการควบคุมการแสกนทางแนวนอนพบว่ามีความถี่เท่ากับ 31.45 kHz ส่วนจากการคำนวณนั้น จะมีค่าเท่ากับ 31.467 kHz



รูปที่ 4.45 สัญญาณสัญญาณควบคุมการสแกนทางแนวตั้ง

จากรูปที่ 4.45 เป็นการวัดสัญญาณการควบคุมการแสกนทางแนวตั้งพบว่ามีความถี่เท่ากับ 60.06 Hz ส่วนจากการคำนวณนั้น จะมีค่าเท่ากับ 60 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

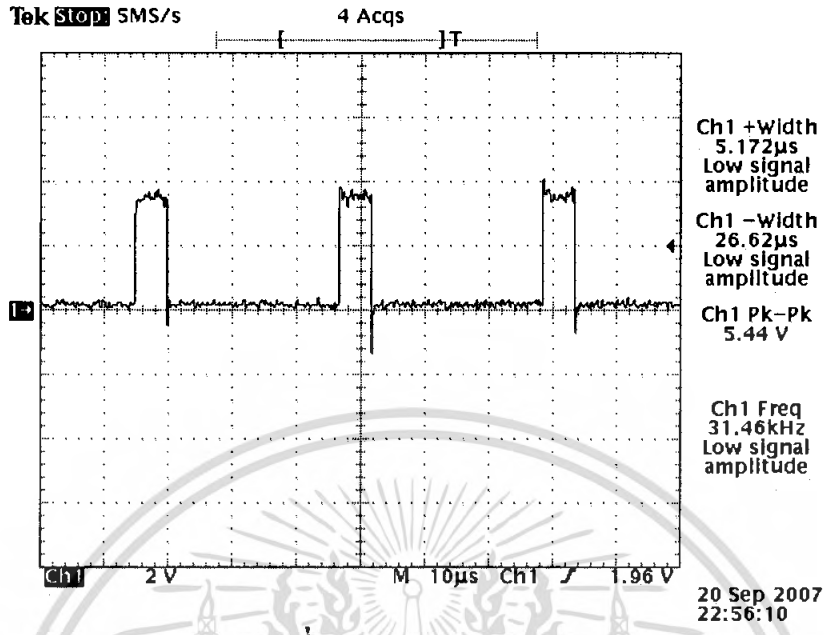


รูปที่ 4.46 เมื่อทำการวัดสัญญาณสัญญาณควบคุมการสแกนทางแนวนอน เปรียบเทียบกับสัญญาณควบคุมการสแกนทางแนวตั้ง

จากรูปที่ 4.46 เป็นการวัดสัญญาณสัญญาณควบคุมการสแกนทางแนวนอนเปรียบเทียบกับสัญญาณควบคุมการสแกนทางแนวตั้งและจากรูปที่ 4.46 (ข) ได้ทำการขยายสัญญาณเพื่อให้เห็นถึงลักษณะของสัญญาณทั้งสองได้อย่างชัดเจน

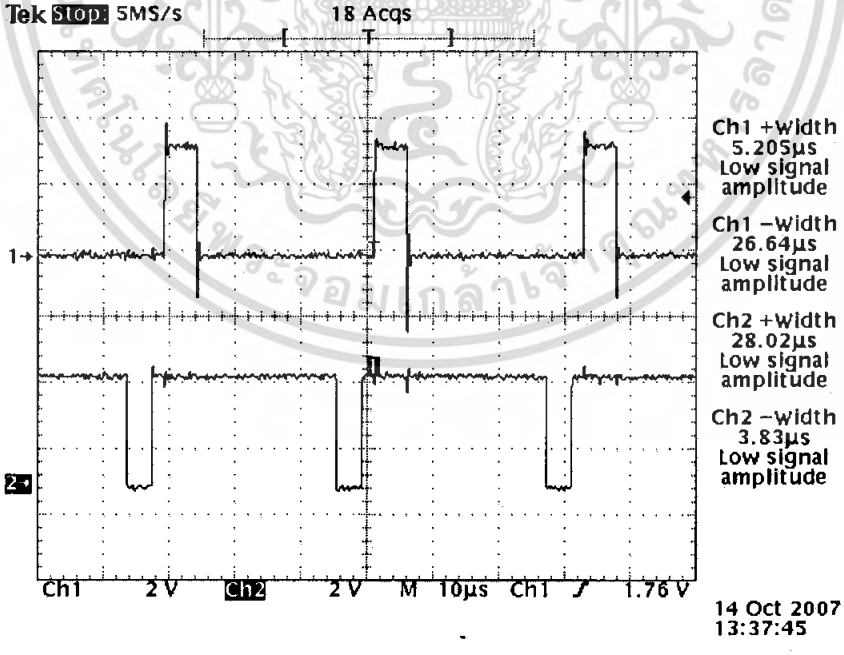
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 สัญญาณวีดีโอออน (Video on) เทียบกับสัญญาณข้อมูล



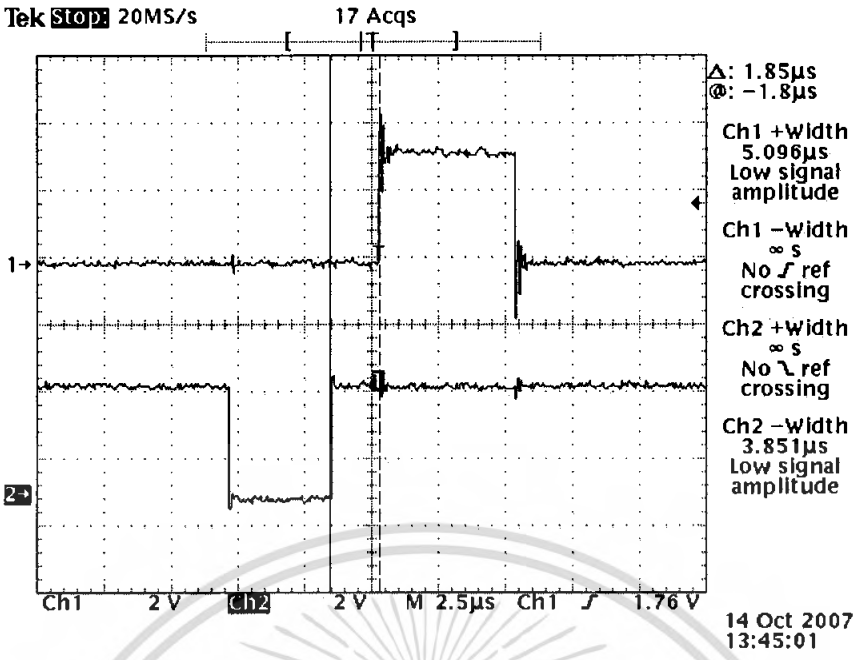
รูปที่ 4.47 สัญญาณวีดีโอออน

จากรูปที่ 4.47 เป็นการวัดสัญญาณวีดีโอออนพบว่าระดับสัญญาณช่วงที่มีการแสดงข้อมูลภาพขนาด 128 พิกเซลต่อแถว (ลอจิก "1") จะมีขนาดเท่ากับ 5.17 μs ซึ่งค่าที่ได้จากการคำนวณมีขนาดเท่ากับ 5.08 μs



(ก)

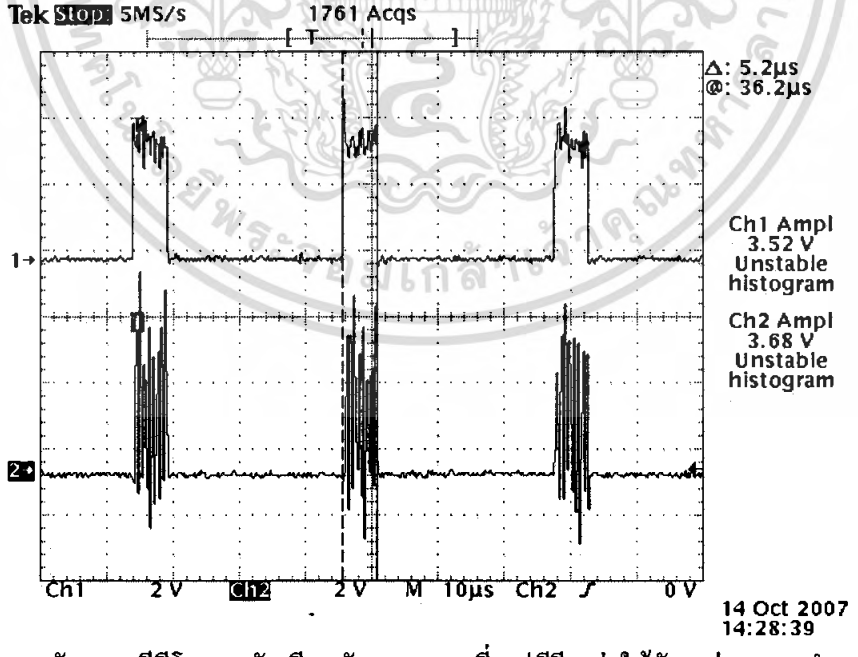
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

รูปที่ 4.48 สัญญาณวีดีโอออนวัดเทียบกับสัญญาณควบคุมการสแกนทางแนวนอน

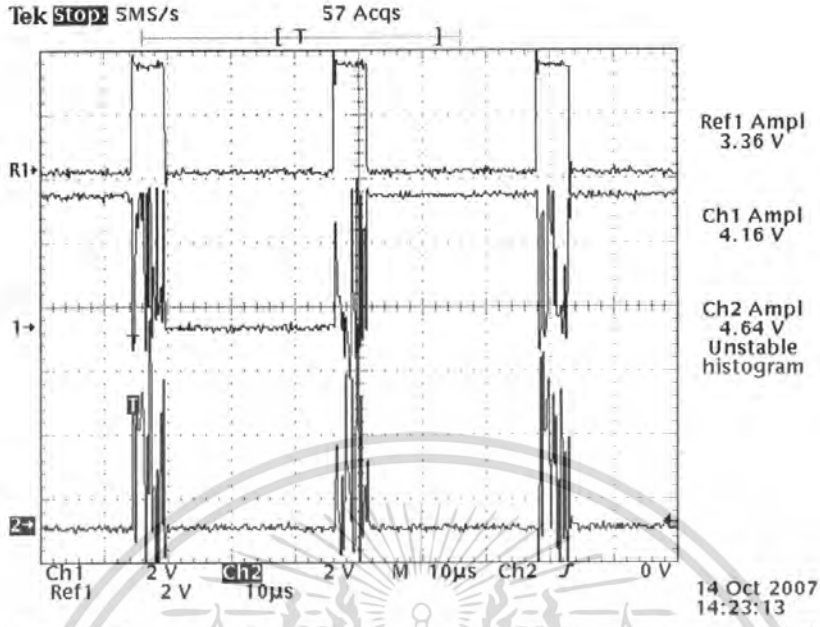
จากรูปที่ 4.48 ในช่องสัญญาณที่ 1 เป็นสัญญาณวีดีโอออนและในช่องสัญญาณที่ 2 เป็นสัญญาณสัญญาณควบคุมการสแกนทางแนวนอนในรูปที่ 4.48 (ข) จะพบว่าการ์ดแบนด์ของสัญญาณสัญญาณควบคุมการสแกนทางแนวนอนมีขนาด 1.85 μs และทางทฤษฎีพบว่าจะมีค่า 1.89 μs



รูปที่ 4.49 สัญญาณวีดีโอออนวัดเทียบกับแอดเดรสที่เอฟพีจีเอส่งให้กับหน่วยความจำภายนอก

จากรูปที่ 4.49 เห็นได้ว่าข้อมูลที่ชี้ตำแหน่งแอดเดรสนั้นจะถูกส่งให้กับหน่วยความจำภายนอกเมื่อสัญญาณวีดีโอออนเป็นลอจิก "1" แต่หากเป็นลอจิก "0" ข้อมูลก็จะไม่ถูกส่งออกไป

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

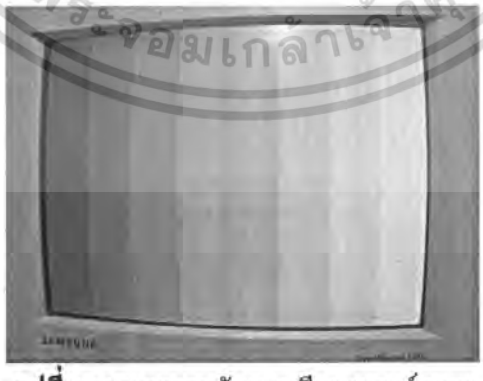


รูปที่ 4.50 สัญญาณวีดีโอออนวัดเทียบกับสัญญาณเอาต์พุตจากหน่วยความจำภายนอกให้กับเอฟพีจีเอ และเอาต์พุตจากเอฟพีจีเอที่จะส่งให้กับวีจีเอ

จากรูปที่ 4.50 จะเห็นได้ว่าข้อมูลเอาต์พุตที่รับมาจากหน่วยความจำภายนอกจะถูกส่งจากชิพเอฟพีจีเอไปให้หน้าจอวีจีเอเมื่อสัญญาณวีดีโอออนเป็นลอจิก "1" แต่หากสัญญาณวีดีโอออนเป็นลอจิก "0" ข้อมูลก็จะไม่ถูกส่งออกไป

4.2.3 ผลการทดลองที่ได้จากหน้าจอวีจีเอ

4.2.3.1 ภาพที่แสดงผลทางหน้าจอวีจีเอเมื่อยังไม่ได้ผ่านการประมวลผลภาพ



รูปที่ 4.51 แสดงระดับแถบสีแบบเกรย์สเกล

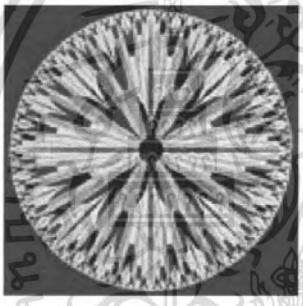
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก (1)

ก (2)

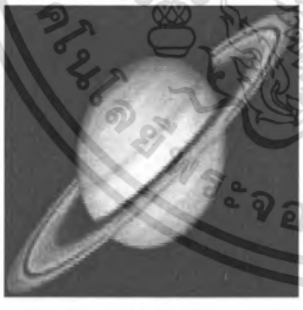
ข (1)

ข (2)



ฃ (1)

ฃ (2)



ค (1)

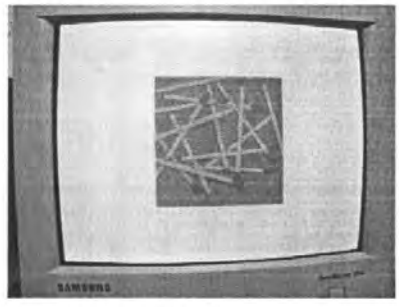
ค (2)

ศ (1)

ศ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฆ (1)



ฆ (2)

รูปที่ 4.52 การเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่นำมาแสดงผลทางหน้าจอวีจีเอ

(1) ภาพต้นแบบ

(2) ภาพที่แสดงบนหน้าจอวีจีเอ

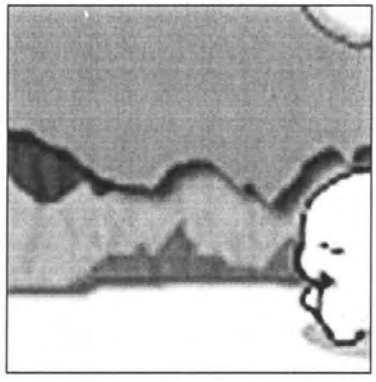
(ก) (ข)

รูปที่ 4.53 แสดงภาพต้นแบบแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1

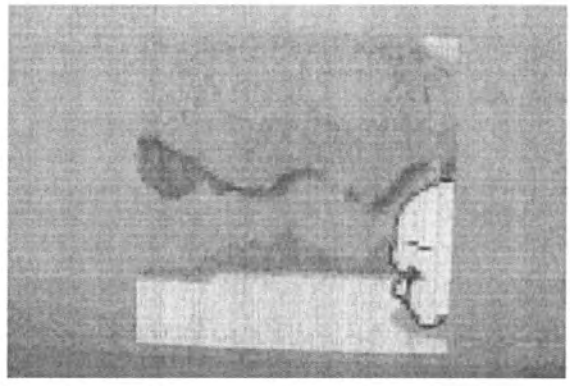
(ก) (ข)

รูปที่ 4.54 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

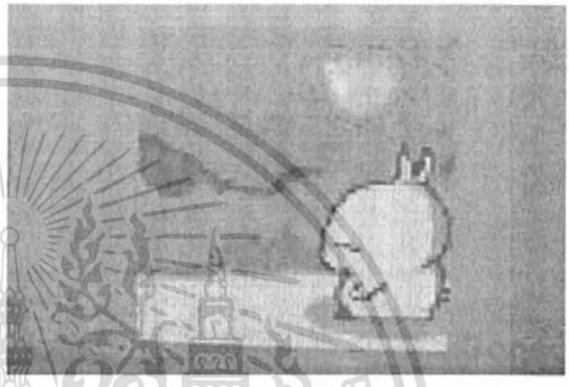


(ก)



(ข)

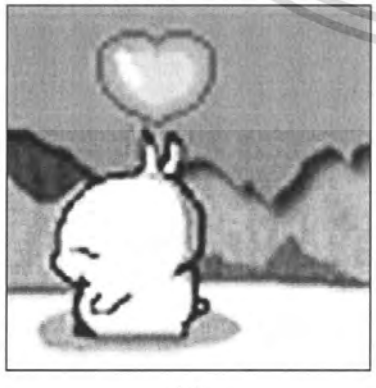
(ฆ)



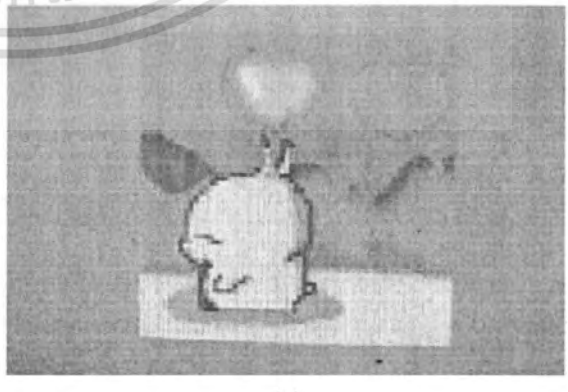
(ค)

(ค)

(ฆ)

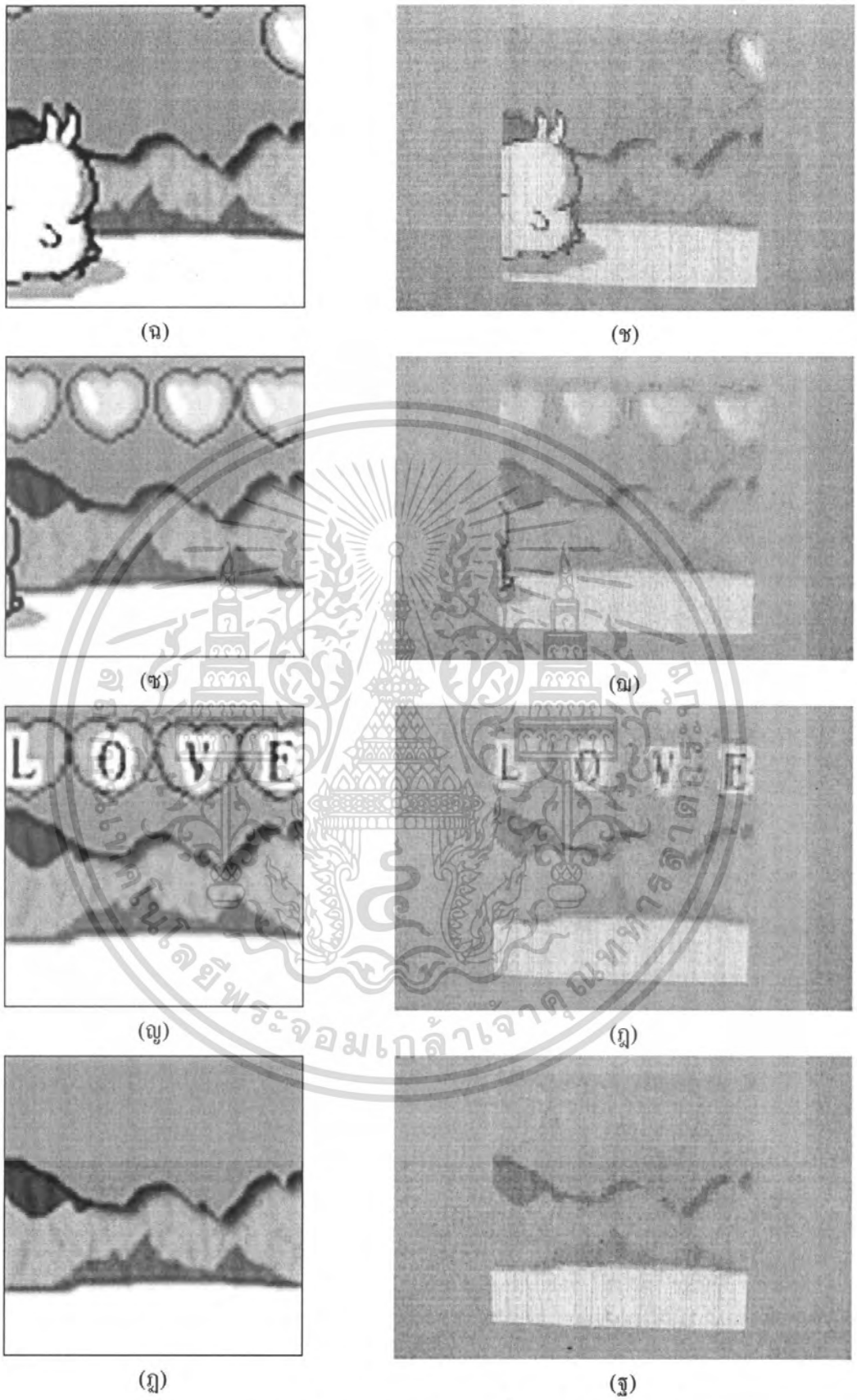


(ง)



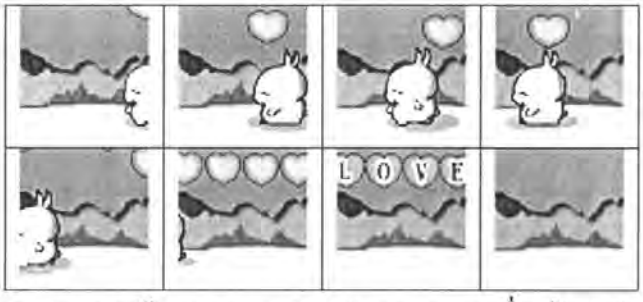
(จ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.55 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 แสดงภาพต้นแบบของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2

4.2.3.2 ภาพที่แสดงผลทางหน้าจอวีจีเอเมื่อผ่านการประมวลผลภาพเรียบร้อยแล้ว

4.2.3.2.1 เมื่อผ่านการประมวลผลด้วยวงจรปรับระดับความเข้มแสงเป็นสองระดับ (Binary Threshold)

โดยนำค่าที่ผู้ใช้งานกำหนดให้มาเปรียบเทียบกับค่าความเข้มแสงของพิกเซลของภาพนั้น ๆ ซึ่งจะแบ่งเป็น 2 กรณี ในกรณีแรกหากค่าความเข้มแสงของพิกเซลของภาพนั้นมีค่าน้อยกว่าค่าที่ผู้ใช้งานกำหนดให้ก็จะแสดงค่าความเข้มแสงเป็น "00000000" หรือสีดำ แต่ถ้ามากกว่าค่าที่ผู้ใช้งานกำหนดให้ก็จะแสดงค่าความเข้มแสงเป็น "11111111" หรือสีขาว ส่วนอีกกรณีหนึ่งนั้นหากค่าความเข้มแสงของพิกเซลของภาพนั้นมีค่ามากกว่าค่าที่ผู้ใช้งานกำหนดให้ก็จะแสดงค่าความเข้มแสงเป็น "00000000" หรือสีดำ แต่ถ้าน้อยกว่าค่าที่ผู้ใช้งานกำหนดให้ก็จะแสดงค่าความเข้มแสงเป็น "11111111" หรือสีขาว

โดยทำการเปรียบเทียบการปรับความเข้มแสงของภาพเป็นสองระดับจากโปรแกรม MATLAB กับผลที่แสดงทางจอวีจีเอ

รูปที่ 4.57 ภาพต้นแบบ ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก)

(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “15”

(ฆ)

(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “63”

(ค)

(ฆ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “127”

รูปที่ 4.58 ในกรณีที่ค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก)

(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “15”

(ฃ)

(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “63”

(ค)

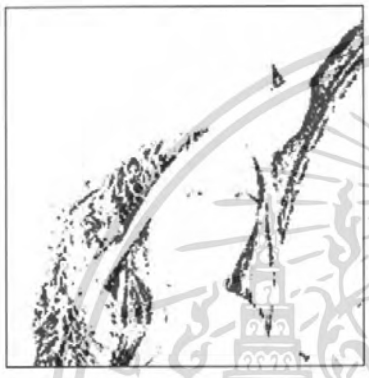
(ฆ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “127”

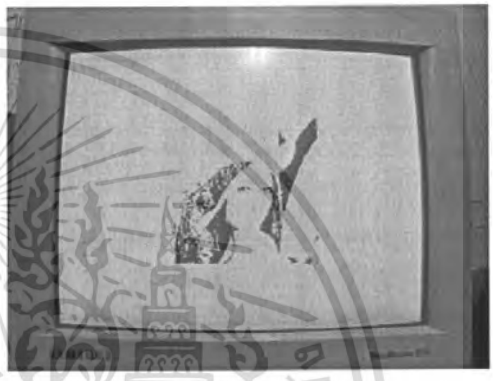
รูปที่ 4.59 ในกรณีที่ค่ามากกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.60 ภาพต้นแบบ ตัวอย่างที่ 2



(ก)



(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “7”

(ฃ)

(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “31”

(ฅ)

(ฆ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “63”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ง)

(จ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “127”

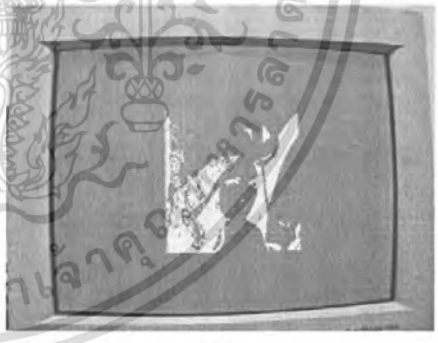
รูปที่ 4.61 ในกรณีที่ค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 2

(ก)

(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “7”

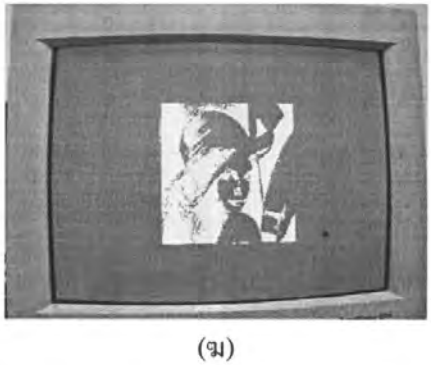
(ฃ)



(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “31”

(ฅ)



(ฆ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “63”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ง)

(จ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “127”

รูปที่ 4.62 ในกรณีที่ค่ามากกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 2

4.2.3.2.2 เมื่อผ่านการประมวลผลด้วยวงจรปรับระดับความเข้มแสง (Contrast)

โดยทำการรับค่าระดับที่ต้องการปรับความเข้มแสงจากผู้ใช้งาน จากนั้นนำค่าที่ได้รับมาทำการเพิ่มหรือลดค่าความเข้มแสงตามที่ต้องการ ซึ่งถ้าหากหลังการประมวลผลมีค่าความเข้มแสงเกิน “11111111” ก็จะคงค่าความเข้มแสงที่พิกเซลนั้นเป็น “11111111” แต่ถ้าภายหลังการประมวลผลค่าความเข้มแสงที่ได้มีค่าน้อยกว่า “00000000” ก็ให้คงค่าความเข้มแสงที่พิกเซลนั้นเป็น “00000000” ดังเดิม ดังนั้นค่าความเข้มแสงที่ได้ก็จะอยู่ในช่วง “00000000” (สีดำ) ถึง “11111111” (สีขาว) ซึ่งในที่นี้ได้ทำการเปรียบเทียบผลที่ได้จากโปรแกรม MATLAB กับผลที่แสดงบนหน้าจอวีจีเอ

(ก)

(ข)

เมื่อลดระดับความเข้มแสงไป“15”

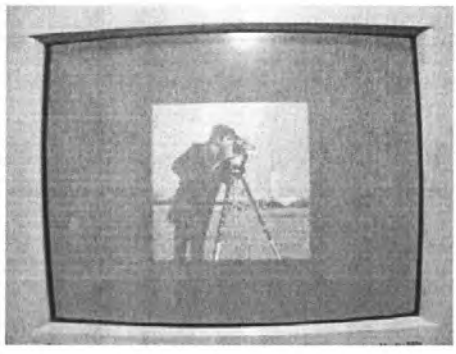
(ฆ)

(ค)

เมื่อลดระดับความเข้มแสงไป“31”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

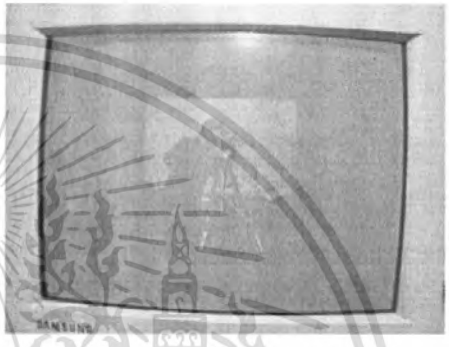
(ฅ)



(ฆ)

เมื่อลดระดับความเข้มแสงไป“63”

(ง)

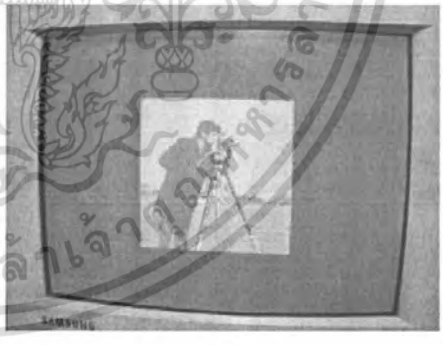


(จ)

เมื่อลดระดับความเข้มแสงไป“127”

รูปที่ 4.63 เมื่อทำการลดระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1

(ก)



(ข)

เมื่อเพิ่มระดับความเข้มแสงไป“15”

(ฃ)

(ค)

เมื่อเพิ่มระดับความเข้มแสงไป“31”

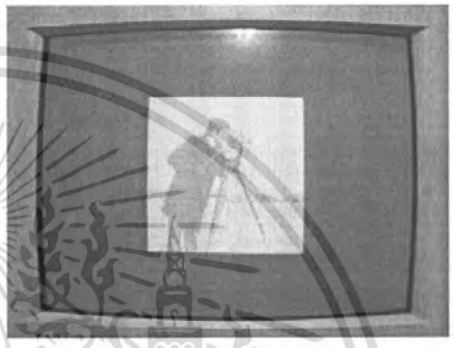
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ค)

(ฆ)

เมื่อเพิ่มระดับความเข้มแสงไป“63”

(ง)



(จ)

เมื่อเพิ่มระดับความเข้มแสงไป“127”

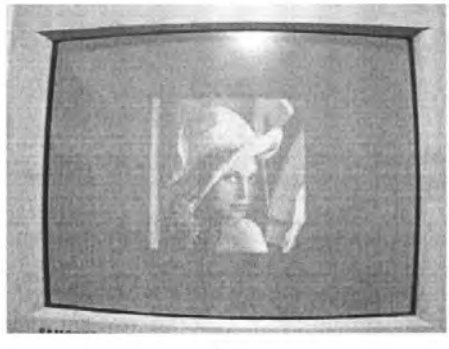
รูปที่ 4.64 เมื่อทำการเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1

(ก)

(ข)

เมื่อลดระดับความเข้มแสงไป“15”

(ฃ)

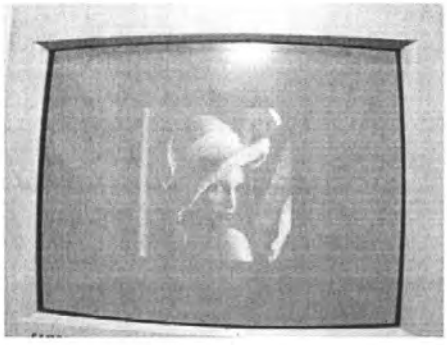


(ค)

เมื่อลดระดับความเข้มแสงไป“31”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

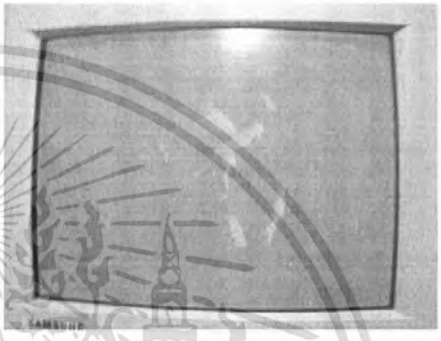
(ศ)



(ฆ)

เมื่อลดระดับความเข้มแสงไป“63”

(ง)



(จ)

เมื่อลดระดับความเข้มแสงไป“127”

รูปที่ 4.65 เมื่อทำลดเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2

(ก)

(ข)

เมื่อเพิ่มระดับความเข้มแสงไป“15”

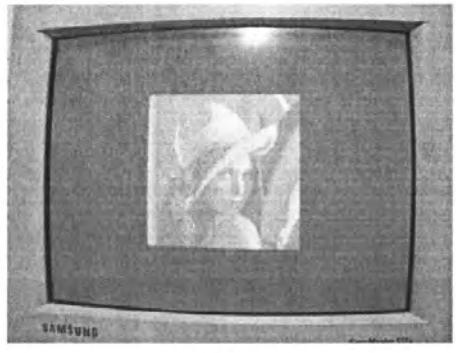
(ฃ)

(ค)

เมื่อเพิ่มระดับความเข้มแสงไป“31”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

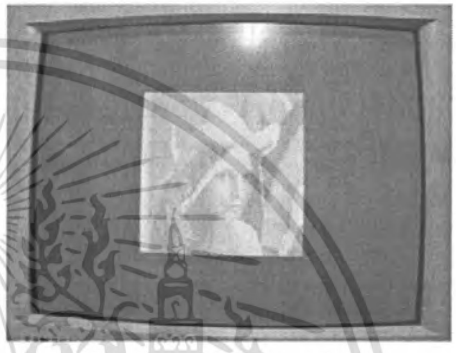
(ค)



(ฆ)

เมื่อเพิ่มระดับความเข้มแสงไป“63”

(ง)



(จ)

เมื่อเพิ่มระดับความเข้มแสงไป“127”

รูปที่ 4.66 เมื่อทำเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2

4.2.3.2.3 เมื่อผ่านการประมวลผลด้วยวงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ

(ก)

(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ข)

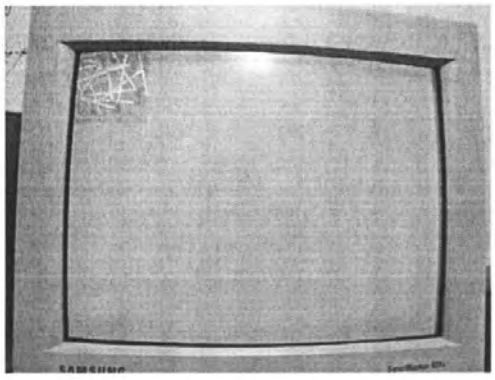
(ค)

(ค)

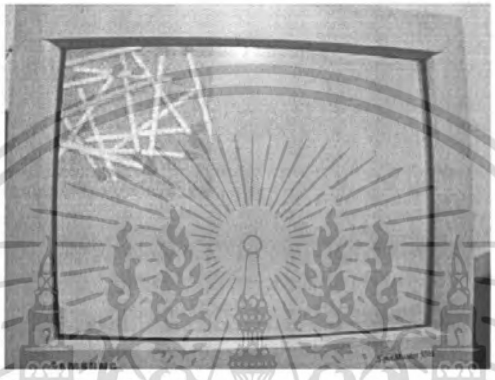
(ฆ)

รูปที่ 4.67 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

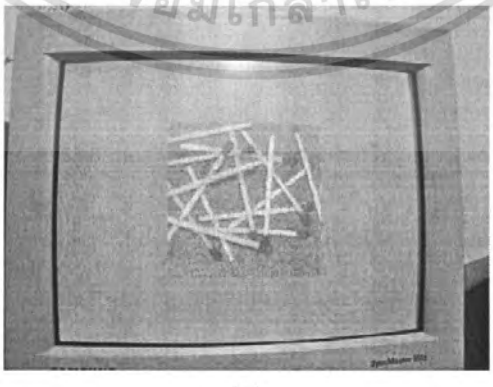


(ก)



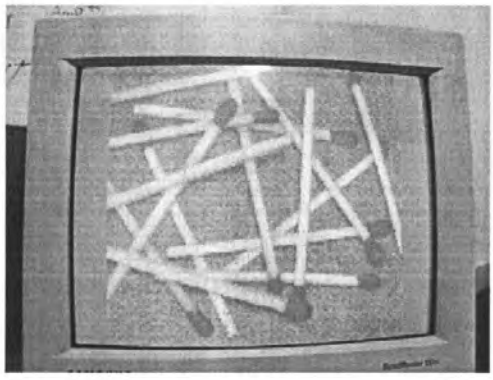
(ข)

(ค)

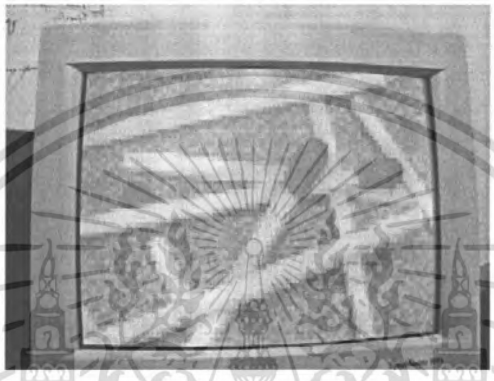


(ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ฅ)



(ฆ)

รูปที่ 4.68 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 2

4.2.3.2.4 การประมวลผลภาพโดยการผ่านวงจรกรองปาสคาล

ก (1)

ก (2)

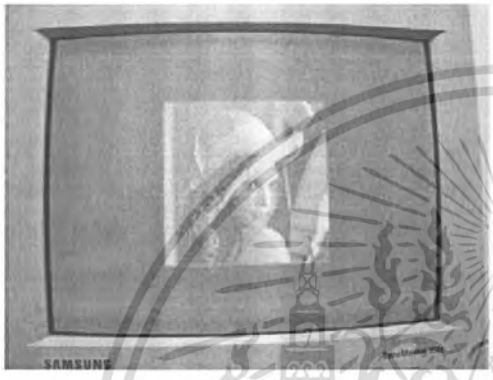
ก (3)

ก (4)

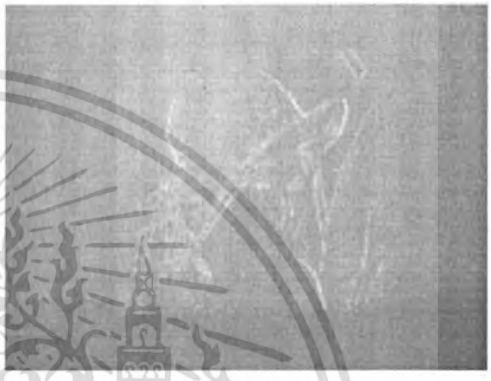
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข (1)

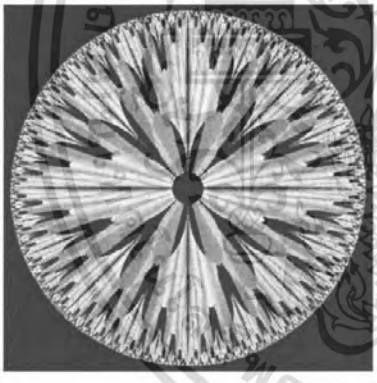
ข (2)



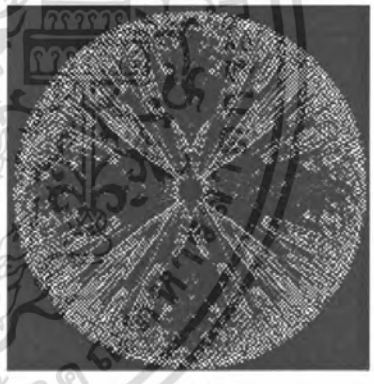
ข (3)



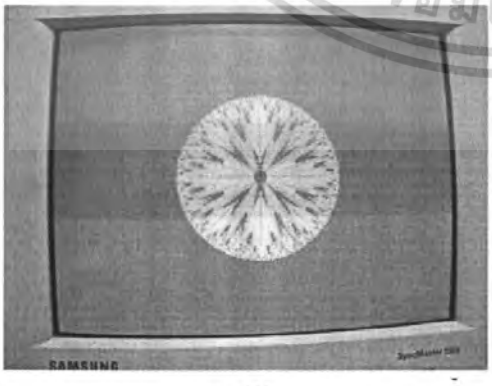
ข (4)



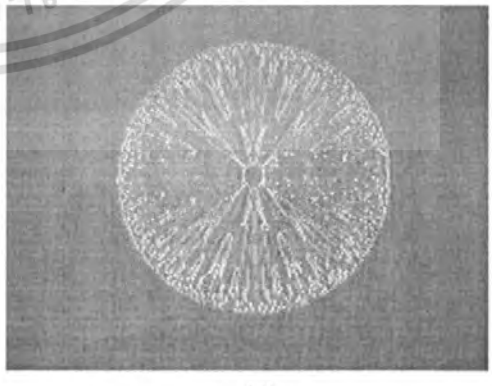
ฃ (1)



ฃ (2)

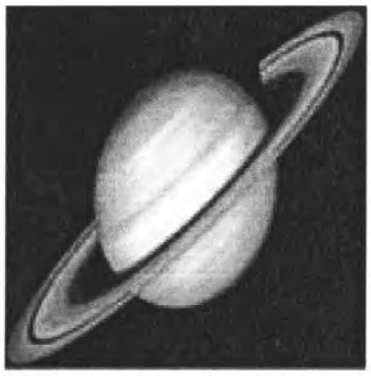


ฃ (3)



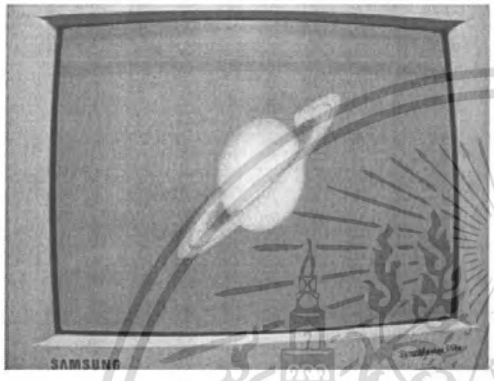
ฃ (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

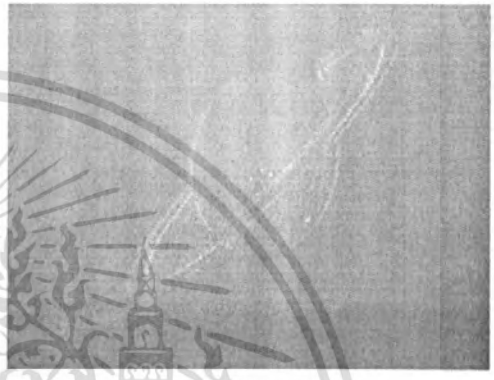


ค (1)

ค (2)



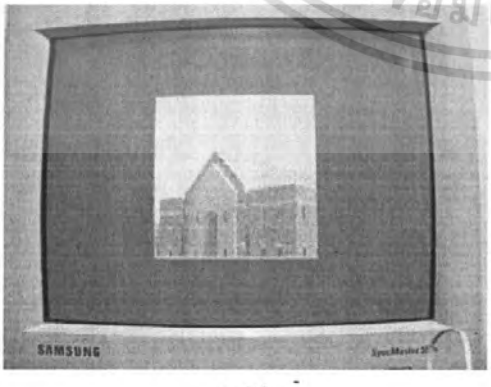
ค (3)



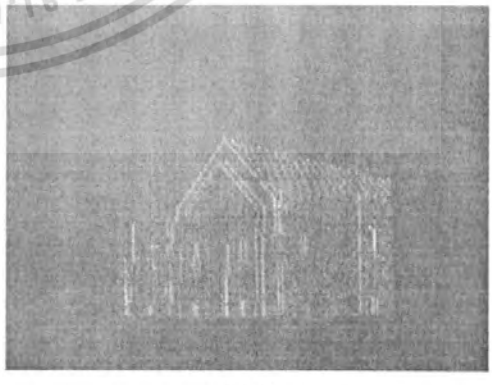
ค (4)

ค (1)

ค (2)



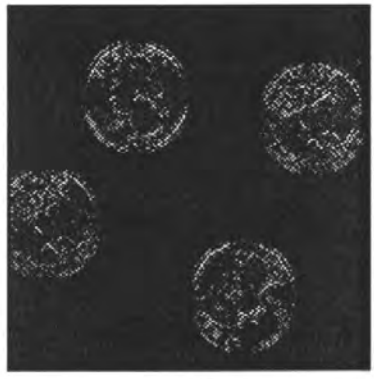
ค (3)



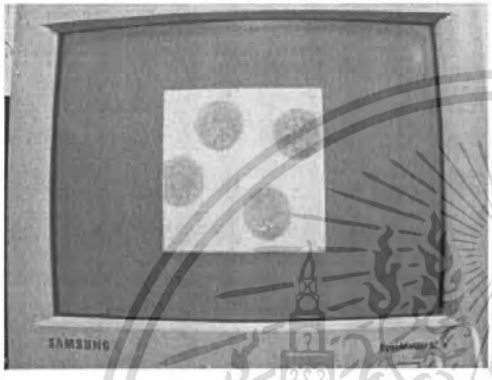
ค (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฆ (1)



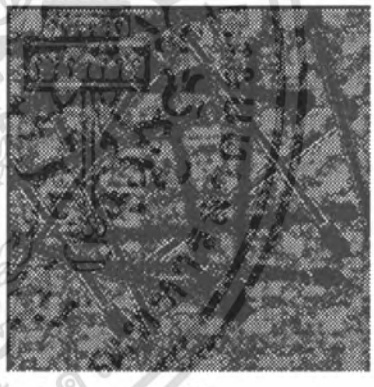
ฆ (2)



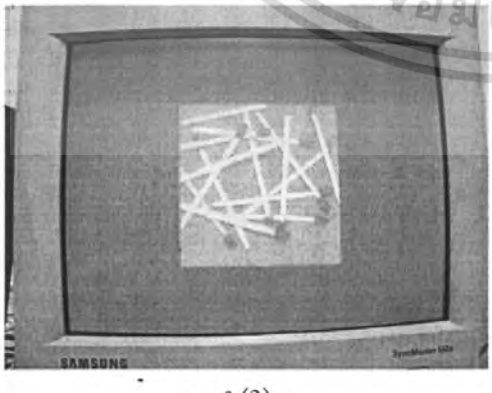
ฆ (3)

ฆ (4)

ง (1)



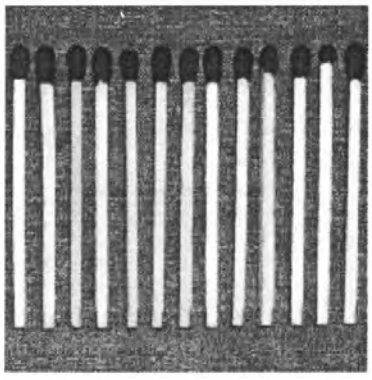
ง (2)



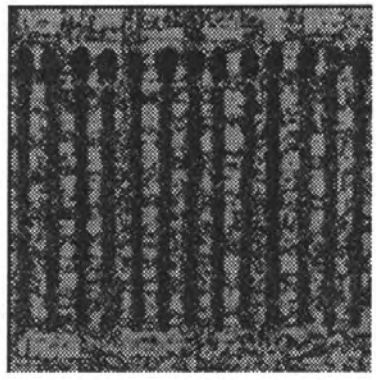
ง (3)

ง (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

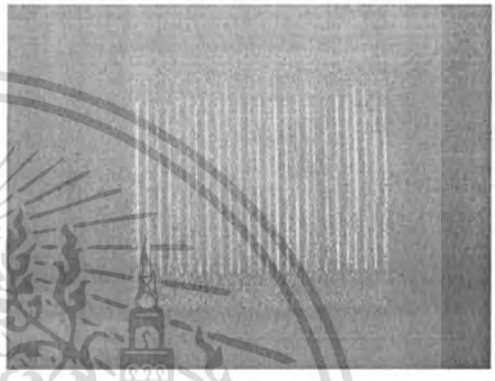


จ (1)



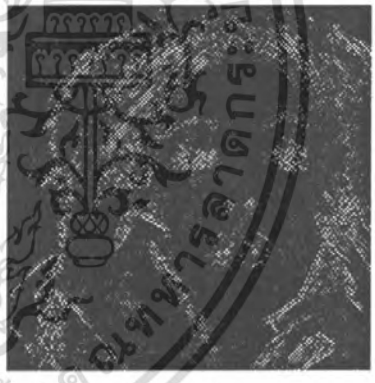
จ (2)

จ (3)



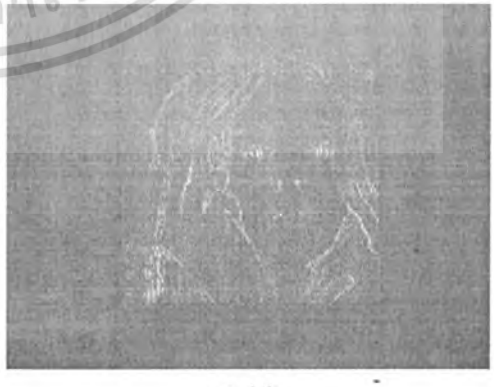
จ (4)

ฉ (1)



ฉ (2)

ฉ (3)



ฉ (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

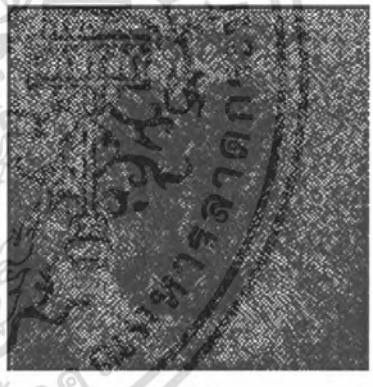
ช (1)

ช (2)

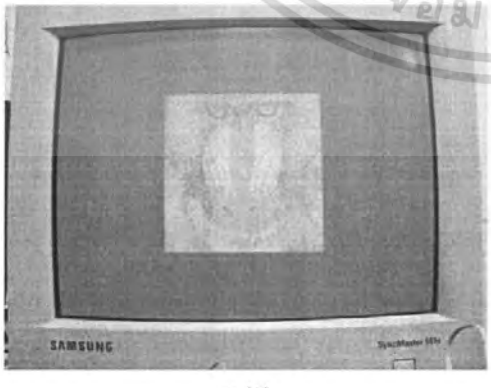
ช (3)

ช (4)

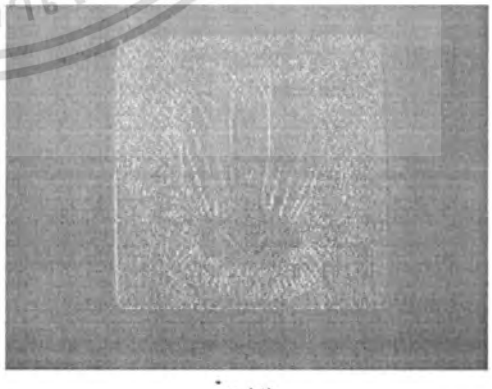
ช (1)



ช (2)

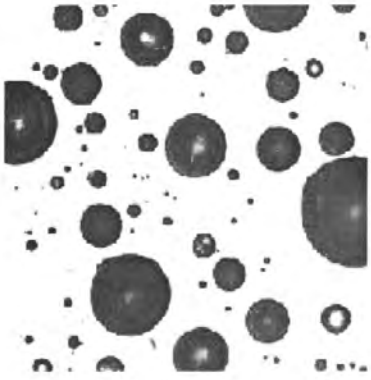


ช (3)

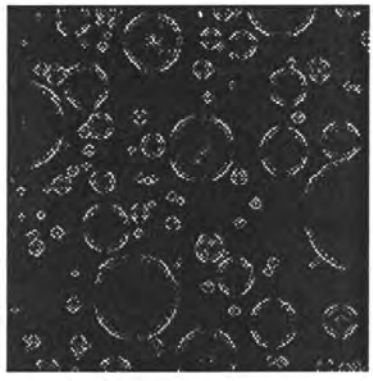


ช (4)

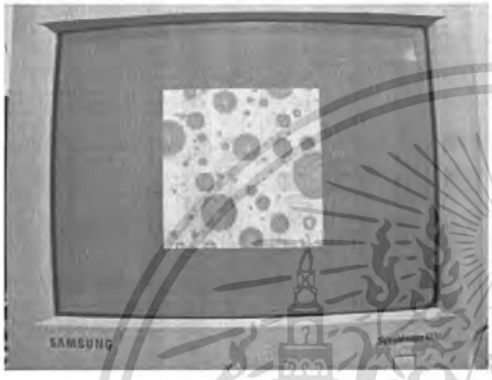
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ฌ (1)



ฌ (2)



ฌ (3)

ฌ (4)

รูปที่ 4.69 ภาพเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่ผ่านวงจรกรองปาสคาลเรียบร้อยแล้ว

(1) ภาพต้นฉบับ

(2) ภาพที่ผ่านวงจรกรองปาสคาลจากโปรแกรม MATLAB

(3) ภาพต้นฉบับเมื่อนำไปแสดงผลบนจอวีจีเอ

(4) ภาพที่ผ่านวงจรกรองปาสคาลเมื่อนำไปแสดงผลบนจอวีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

โครงงานนี้นำเสนอการประมวลผลภาพชนิดระดับสีเทาแบบเวลาจริง โดยการใช้อุปกรณ์เอฟพีจีเอ ซึ่งได้ทำการศึกษาหลักการการทำงานของหน้าจอวีจีเอ และทำการออกแบบวงจรที่ใช้ในการควบคุมแล้วแสดงผลบนหน้าจอวีจีเอ จากการทดลองสามารถสร้างสัญญาณการสแกนทางแนวนอนที่มีความถี่ 31.45 kHz และสามารถสร้างสัญญาณการสแกนทางแนวตั้งที่มีความถี่ 60.24 Hz ซึ่งจากการคำนวณตามทฤษฎีสัญญาณการสแกนทางแนวนอนมีความถี่เท่ากับ 31.476 kHz และสัญญาณการสแกนทางแนวตั้งมีความถี่เท่ากับ 60 Hz จะเห็นได้ว่าสัญญาณที่ได้มีความถี่ใกล้เคียงตามทฤษฎี ทั้งนี้ได้ทำการแสดงผลภาพระดับสีเทาแบบภาพนิ่งและแบบภาพเคลื่อนไหว

จากการศึกษาปาสคาลทรานสฟอร์ม ได้ทำการจำลองการทำงานของวงจรกรองปาสคาลใน 1 มิติ โดยใช้ปาสคาลทรานสเฟอร์ฟังก์ชันในการออกแบบ พบว่ามีคุณสมบัติผลตอบสนองทางความถี่คล้ายกับตัวกรองความถี่สูงผ่าน คือสามารถกรองความถี่สูงออกมาได้ แต่มีความแตกต่างกันในส่วนที่ตัวกรองความถี่สูงผ่านสามารถปรับเปลี่ยนความถี่ตัดผ่านได้ แต่ในวงจรกรองปาสคาลไม่สามารถทำได้

จากการจำลองการทำงานของปาสคาลฟิลเตอร์ใน 2 มิติ พบว่าเมื่อนำภาพระดับสีเทามาประมวลผลโดยใช้วงจรกรองปาสคาล ภาพที่ได้จะมีลักษณะขอบภาพชัดขึ้น เนื่องจากบริเวณที่เป็นขอบภาพมีการเปลี่ยนแปลงค่าความเข้มแสงมาก ๆ หรือมีความถี่สูง บริเวณที่มีความถี่สูงจึงสามารถผ่านวงจรกรองออกมาได้ จึงเห็นเป็นขอบภาพ และจากการทดลองพบว่าหน้ากากคอนโวลูชันปาสคาลขนาด 3 x 3 ให้ผลดีที่สุด และมีโครงสร้างที่ง่ายต่อการออกแบบให้กับเอฟพีจีเอ

จากผลการทดลองทำการรับข้อมูลภาพจากหน่วยความจำภายนอก เข้ามาผ่านกระบวนการประมวลผลด้วยวงจรกรองปาสคาลภายในเอฟพีจีเอ แล้วแสดงผลบนหน้าจอวีจีเอ พบว่าภาพที่แสดงบนหน้าจอมีลักษณะเห็นขอบภาพชัดขึ้น ซึ่งเป็นไปตามผลจากการจำลองการทำงานของวงจรกรองปาสคาลจากโปรแกรม MATLAB

นอกจากนี้แล้วยังมีการออกแบบวงจรประมวลผลภาพในลักษณะต่างๆ ได้แก่ การปรับค่าระดับความเข้มแสงให้มีค่าเพิ่มขึ้นหรือลดลง เพื่อเป็นการปรับความสว่าง หรือมืดให้กับภาพต้นแบบ และยังมีการออกแบบวงจรการปรับค่าความเข้มแสงให้มีเพียง 2 ระดับ คือ “00000000” ซึ่งแสดงผลเป็นสีดำ และ “11111111” ซึ่งแสดงผลเป็นสีขาว โดยผู้ใช้งานสามารถป้อนค่าที่ใช้ในการประมวลผลมีค่าตั้งแต่ 0-255 ระดับ ทั้งนี้ยังมีการออกแบบวงจรขยายขนาดของภาพต้นแบบและจัดตำแหน่งในการแสดงภาพบนหน้าจอได้อีกด้วย

ปัญหาที่พบจากการทำการทดลองในโครงงานนี้คือหน่วยความจำภายนอกที่นำมาใช้งานมีขนาดที่จำกัดทำให้ภาพที่เก็บมีขนาดที่จำกัดตามไปด้วย อีกประการหนึ่งพบปัญหาในส่วนของภาควงจรแปลงระดับสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกที่ใช้งานสำหรับภาพ ซึ่งมีคุณสมบัติแปลงสัญญาณดิจิตอลขนาด 8 บิต (0 - 255) พบว่าเมื่อทำการต่อวงจรร่วมกับวงจรส่วนอื่น ๆ ภาพที่ผ่านวงจรมีสีไม่เต็มระดับเมื่อเทียบกับภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่วงจรกรองปาสคาลมีคุณสมบัติของตัวกรองความถี่สูงผ่าน ทำให้สามารถนำมาประยุกต์ใช้งานเพื่อหาขอบภาพได้ ส่วนวงจรปรับระดับความเข้มแสง และวงจรปรับระดับความเข้มแสงเป็นสองระดับ สามารถนำไปใช้ในการปรับคุณภาพของภาพเพื่อประยุกต์ในการใช้งานด้านอื่น ๆ ต่อไป

ในการศึกษาโครงงานนี้ ทำให้มีความรู้เกี่ยวกับทฤษฎีต่าง ๆ ซึ่งสามารถนำมาใช้งานร่วมกันได้ รวมทั้งการออกแบบวงจรที่สามารถนำมาสร้างเป็นฮาร์ดแวร์ด้วยภาษาวีเอชดีแอล ที่มีความยืดหยุ่นในการออกแบบสูง ทั้งนี้ทำการเชื่อมต่อเอฟพีจีเอกับหน่วยความจำภายนอกและหน้าจอวีจีเอ และได้ความรู้เกี่ยวกับการประมวลผลภาพด้านต่าง ๆ มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Maurice F. Aburdene, Fellow, Thomas J. Goodman, "The Discrete Pascal Transform and Its Applications," IEEE Signal Processing Letters, VOL 2, No. 7, July 2005

2. N. Ronnarongrit, S. Chivapreecha and K. Dejhan, "Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization," Accepted to present on the 4th International Colloguim on Signal Processing and its Applications (CSPA) 2008, March 7-9, Kuala Lumper

3. S. Chivapreecha, U. Nithirochananont and K. Dejhan, "Investiyation of Frequency Characteristic in Discrete Pascal Transform and Its Applications," Accepted to present on the 4th International Colloguim on Signal Processing and its Applications (CSPA) 2008, March 7-9, Kuala Lumper

4. S. Chivapreecha, A. Jaruvamkul and K. Dejhan, "Modified Pascal Matrix for Biquad Digital Filter Design and Its Filter Structure Realization," Accepted to present on the 4th International Colloguim on Signal Processing and its Applications (CSPA) 2008, March 7-9, Kuala Lumper

5. S. Chivapreecha, S. Sriyapong, S. Junnapiya and K. Dejhan, "Bilinear s-z with frequency transformation using Pascal matrix operation," Proc. 2005 International Symposium on Communications and Information Technology (ISCIT 2005), Beijing, China, October 12-14, 2005.

6. S. Chivapreecha and K. Dejhan, "Pascal Matrix Operation for Bilinear s-z with Frequency Transformation," Proc. The 2006 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2006), Chiang Mai, Thailand, July 10-13, 2006

7. Wasan Mongkhommalee, Sorawat Chivapreecha, Siraphop Tooprakri and K. Dejhan, "Biquad Digital Filter Design Using Pascal Matrix," Proc. The 2007 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2007), Vol. 1, pp. 21-22, Busan, Korea, July 8-11, 2007

8. Wikipedia, the free encyclopedia. "EPROM"
Available : http:// Wikipedia.com

9. HWb, "VGA (15)"
Available : http:// VGA_(15)_-_HwB.htm

10. Wikipedia, the free encyclopedia. "VGA USB 1394 firewire EVC"
Available : http://www.wikipedia.com

11. Electoday, Electronics and Microcontroller. "Image Processing"
Available : http://www.electoday.com

12. ชำนาญ ปัญญาใส, วัชรากร หนูทอง, "ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล", ซีเอ็ดยูเคชั่น, กรุงเทพ, 2547.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. มนัส สังวรศิลปี, วรรัตน์ภัทรอมรกุล, "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์", สำนักพิมพ์อินโฟเพรส, กรุงเทพ, 2543.

14. อุปกรณ์คอมพิวเตอร์ Hardware, "ROM"

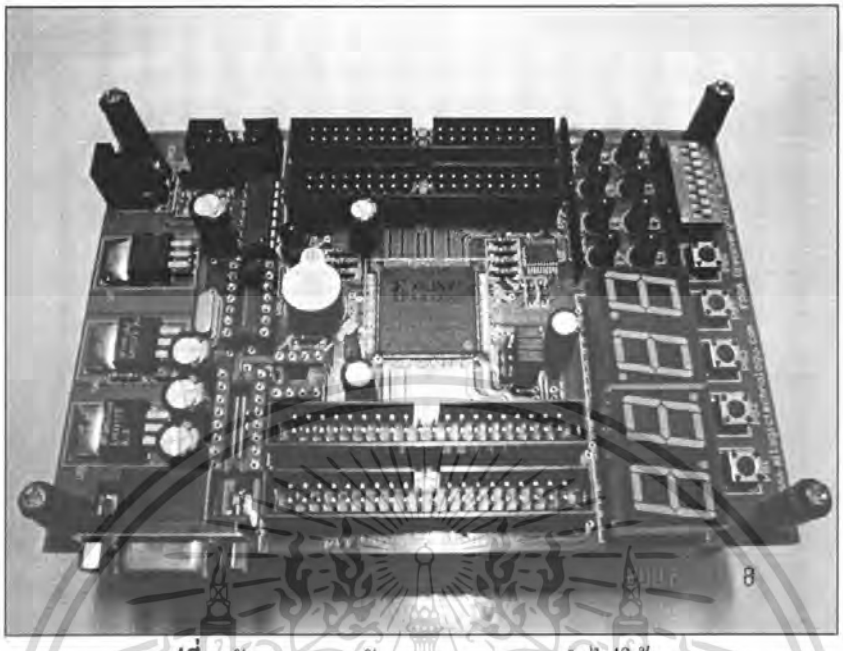
Available : http//bcoms.net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของบอร์ด FPGA Discovery-III XC3S200



รูปที่ 1 ลักษณะของตัว FPGA และการนำไปใช้งาน

Connector and Jumper

- **Expansion connector (K1 – K4)**

เป็นหัวต่อที่ใช้เชื่อมสัญญาณ I/O จาก FPGA ไปยังบอร์ดหรืออุปกรณ์ภายนอกที่มี I/O เป็น 3.3V โดยจะต่ออยู่กับขา CPLD ในกรณีที่ I/O ของ FPGA เป็น Output สามารถต่อออกจากบอร์ดไปขับ Input FPGA Discovery-III XC3S200 Board Manual V1.0 (REV2, 31/10/47) 2 ของอุปกรณ์ที่เป็นระบบ 3.3V และ 5V ได้โดยตรง แต่ถ้า I/O ของ FPGA เป็น Input นั้นจะรับได้เฉพาะ Input ที่เป็นระบบ 3.3V เท่านั้น (ถ้ารับมาจากระบบ 2.5V ต้องใช้ตัวความต้านทาน (R) มา Pull up) แต่ถ้า Input เป็นระบบ 5V จะต้องใช้บัฟเฟอร์ที่เป็นระบบ 3.3V มากั้นเพื่อป้องกันไม่ให้ I/O ของ FPGA ได้รับความเสียหาย บัฟเฟอร์ที่เป็นระบบ 3.3V อาจใช้ไอซีตระกูล 74HCxx หรือ 74ACxx (เร็วกว่า) มากินไฟเลี้ยง (Vcc) 3.3VDC และต้องต่อความต้านทาน 200 โอห์มที่ Input ของบัฟเฟอร์ทุกตัว (เพื่อจำกัดกระแสไม่ให้ Input ของบัฟเฟอร์เสียหาย คือ < 10 mA)

- **JTAG connector**

เป็นหัวต่อที่ใช้ต่อกับสายที่ใช้โปรแกรมข้อมูลลงตัว FPGA และ PROM โดยผ่ายทาง JTAG Cable

- **J1**

จัมเปอร์ J1 ประกอบด้วย M0, M1, M2 โดยปกติให้เซ็ทไว้ที่ลอจิก "L" หรืออยู่ในโหมด Master serial เนื่องจากเราสามารถโปรแกรม PROM หรือ FPGA โดยใช้สาย JTAG ได้อยู่แล้วโดยไม่ต้องสนใจตำแหน่งของจัมเปอร์แต่อย่างใด

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **J2**

จัมเปอร์ที่ใช้ควบคุมให้ FPGA ทำการ Pull up I/O ของ FPGA ทุกขาเมื่อใส่จัมเปอร์ และจะเป็น Hi Impedance เมื่อถอดจัมเปอร์ออก

- **J3**

จัมเปอร์ที่ใช้ตัด BUZZER ออกจาก FPGA เมื่อถอดจัมเปอร์ออก

Input

- **DIP switch (DIP SW)**

เป็นชุดของสวิทช์เลื่อนขนาดเล็กที่ใช้ป้อนข้อมูลเข้าสู่ FPGA โดยถ้าเลื่อนลง (Off) จะเป็น "1" ถ้าเลื่อนขึ้น (On) จะเป็น "0" โดยเชื่อมต่อกับขาของ FPGA ดังตารางด้านท้าย DIP SW ทุกตัวจึงทำงานแบบ Active Low

- **Push button switch (PB1 – PB5)**

เป็นสวิทช์กดติดปล่อยดับที่ให้สัญญาณเอ้าท์พุทเป็นระดับลอจิก "0" เมื่อกดสวิทช์ และเป็นระดับลอจิก "1" เมื่อปล่อยสวิทช์ โดยจะต่ออยู่กับขา FPGA ดังตารางด้านท้าย Push button switch ทุกตัวจึงทำงานแบบ Active Low

- **Changeable oscillator (OSC)**

เป็นตัวกำเนิดสัญญาณนาฬิกาที่สามารถเปลี่ยนค่าความถี่ที่ต้องการได้ โดยการถอดเปลี่ยนออสซิลเลเตอร์เดิม (3.3V) ที่ให้มาบนบอร์ดออก แล้วใส่ตัวใหม่(3.3V) เข้าไปแทนที่ที่ซอคเก็ตไอซีเบอร์ 74AC04 โดยที่เอาท์พุทของ OSC จะต่ออยู่กับขา FPGA ดังตารางด้านท้าย ซึ่งเป็นขา Global clock เหมาะสำหรับวงจรที่ต้องการความถี่ในการทำงานสูง ๆ

Output

- **7-Segment (DIGIT1 – DIGIT4)**

เป็นตัวแสดงผลเจ็ดส่วนจำนวน 4 หลักที่สามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับตัวแสดงผลเจ็ดส่วนทั้ง 4 หลัก) โดยเรียงจากซ้ายไปขวาคือ DIGIT4, DIGIT3, DIGIT2 และ DIGIT1 โดยตัวที่ 2 และ 1 จะทำการกลับตัวแสดงผลเจ็ดส่วนให้เพื่อใช้จุด (Dot) ในการทำนาฬิกาหรือ แสดงองศาในการวัดอุณหภูมิ เช่น 11:39 หรือ 20° ตัวแสดงผลเจ็ดส่วนทั้งหมดจะต่อขาคาต้าเข้าด้วยกันโดยมีขาไฟร่วม (Common cathode) แยกกันสี่ขา ดังนั้นผู้ใช้จึงจำเป็นต้องใช้เทคนิคในการสแกน (Scan) เพื่อให้ตัวแสดงผลทั้งเจ็ดส่วนสามารถแสดงผล พร้อมกันได้ทั้งหมด และตัวแสดงผลทั้งหมดเป็นแบบไฟร่วม (Common cathode) โดยจะต่ออยู่กับขา FPGA

- **LED แสดงผล**

LED แสดงผล L0 – L7 จะต่อแชร์กับ I/O ของ Connecter K3 และ K4 โดยที่ L2, L3, L6 และ L7 จะต่อแชร์กับ I/O ของ Connecter K3 โดยมี Resister "RNET3" 8P4R 470 Ohm จำกัดกระแส และ L0, L1, L4 และ L5 จะต่อแชร์กับ I/O ของ Connecter K4 โดยมี Resister "RNET3" 8P4R 470 Ohm จำกัดกระแส

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Buzzer**

เป็นออดความถี่เสียง (Buzzer) โดยที่จะมีเสียงดังเมื่อป้อนสัญญาณเป็น High "1" โดยจะต่ออยู่กับขาFPGA กรณีที่ต้องการใช้ I/O ของ Connecter K4 ที่แชร์อยู่กับออดหรือไม่ต้องการใช้ออดให้ถอดจัมเปอร์ J3 ออก

Misc

- **Jack สำหรับ DC Adaptor**

เป็นหัวต่อไฟเลี้ยงเพื่อป้อนให้แก่บอร์ดในการทำงาน ต่ออยู่กับอแดปเตอร์ที่มีไฟออกมาเป็น 4.7V – 9V โดยมีขั้วด้านในเป็น บวก "+" ด้านนอกเป็น ลบ "-"

- **Power LED (POWER)**

เป็นไดโอดเปล่งแสงว่าในขณะนั้น ๆ มีไฟเลี้ยงบอร์ดอยู่หรือไม่

- **RS-232C Port**

เป็นพอร์ต RS-232C ซึ่งหากไม่ต้องการใช้พอร์ต RS-232C แต่ต้องการใช้เป็น I/O ที่ Connector K1 และ K2 (ส่วนที่แชร์ I/O อยู่กับพอร์ต RS-232C) ให้ถอดไอซี MAX3232CPE ออกจาก Socket

หมายเหตุ ขา 15 ของไอซี MAX3232CPE ต้องบัดกรีลงกราวนด์ด้วย

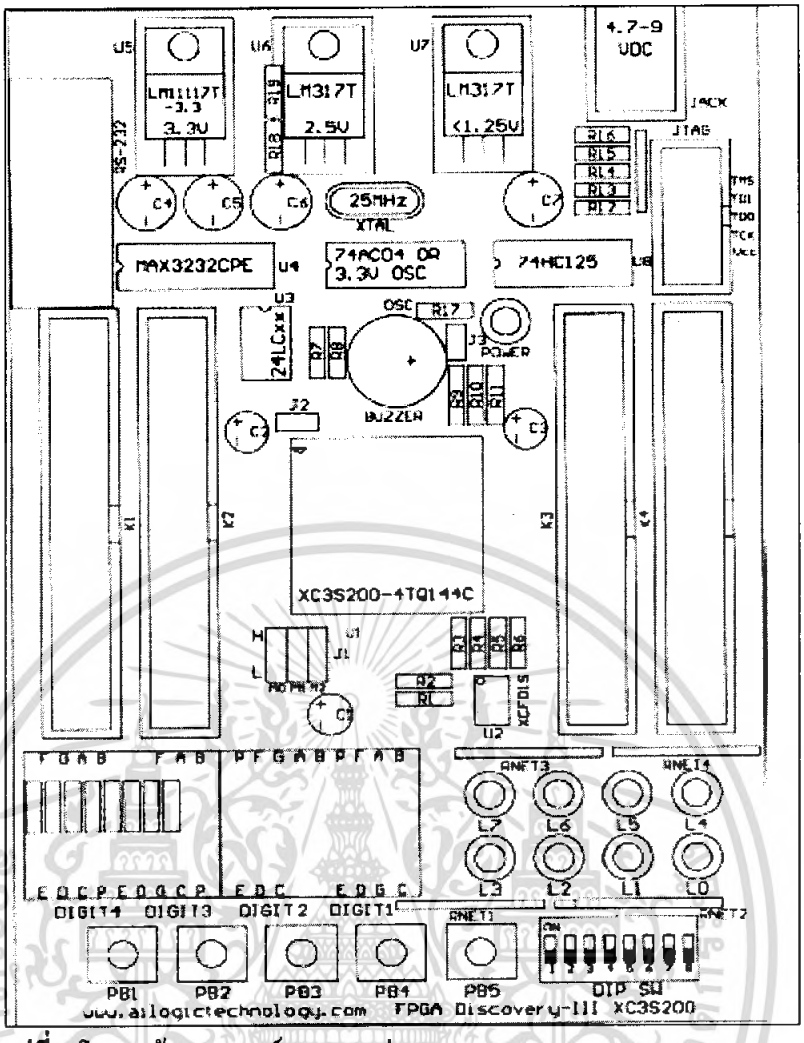
- **Platform Flash PROM เบอร์ XCF01S**

เป็น Serial PROM ที่สามารถโปรแกรมได้โดยตรงผ่านทางสาย JTAG สามารถโปรแกรมซ้ำได้ประมาณ 20,000 ครั้ง

- **I2C Socket**

เป็น Socket สำหรับใส่ไอซีแบบ I2C Serial EEPROM เบอร์ 24LCxx เช่น 24LC256 เป็นต้น ซึ่งสามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับ I2C Serial EEPROM) I/O 2 ขานี้จะมี Pull up resister 4.7 kOhm ต่ออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 โครงสร้างของบอร์ด FPGA รุ่น Spatan III (Discovery-III XC3S200)

การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

ในการออกแบบวงจรเชิงเลขนั้นทำได้โดยการวาดวงจร (Schematic) หรือใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์ (Hardware description language) ในขั้นตอนนี้เป็นขั้นตอนที่ไม่แตกต่างกันระหว่างการออกแบบด้วย FPGA และ ASIC ในกรณีที่ใช้ภาษาอธิบายฮาร์ดแวร์ แต่ในกรณีที่ออกแบบโดยวิธีการวาดวงจรจะแตกต่างกันโดยที่การทำวิธีนี้จะต้องคำนึงถึงเทคโนโลยีที่จะใช้ซึ่งแต่ละเทคโนโลยีก็มีความแตกต่างกันไป จะเห็นได้ว่าการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ ทำได้สะดวกกว่าเพราะการทำด้วยวิธีนี้ไม่ต้องคำนึงถึงเทคโนโลยีที่จะใช้ (Technology independence) และที่สำคัญการออกแบบด้วยวิธีนี้สามารถที่จะแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่าเพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี

ในการเขียนโค้ดสิ่งที่ต้องคำนึงถึงคือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรได้และให้คุณสมบัติของวงจรตามที่กำหนด เพราะลักษณะการเขียนโค้ดจะมีผลโดยตรงกับวงจรที่ได้ เนื่องจากในการสังเคราะห์วงจรนั้นซอร์ฟแวร์สังเคราะห์วงจร (Synthesis Tools) จะทำการสังเคราะห์ตามโค้ดที่เขียน ถ้าอธิบายการทำงานของวงจรเดียวกันแต่เขียนโค้ดในลักษณะที่ต่างกันเมื่อสังเคราะห์แล้วจะได้วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ต่างกัน และจากวงจรที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันทั้งในด้านของขนาดหรือความเร็ว (Area and Time) ส่วนการเขียนโค้ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

การจำลองการทำงานของวงจร (Simulation)

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะเป็นขั้นตอนที่ใช้ตรวจสอบฟังก์ชันการทำงานของวงจรว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหน เพื่อที่จะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะใช้ซอร์ฟแวร์สำหรับการจำลองการทำงานของวงจร เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจรควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดลเกิดขึ้นตอนไหนจะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้น ๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้เกิดข้อผิดพลาด นั่นคือการทำการจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจ สอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตามข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้องในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์ การเชื่อมต่อสัญญาณ (post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้องตรวจสอบคุณสมบัติอื่น ๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวม สามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงานของวงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่าง ๆ

การสังเคราะห์วงจร (Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอร์ฟแวร์สังเคราะห์วงจร (Synthesis tools) ทำการสังเคราะห์โค้ดเพื่อให้ได้เป็นวงจรขึ้นมา แต่ต้องตรวจสอบด้วยว่าซอร์ฟแวร์นั้น ๆ สนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการใช้หรือไม่โดย FPGA ที่นิยมใช้งาน เช่น ของบริษัท Xilinx ตระกูล XC4000 และ บริษัท Altera ตระกูล FLEX 10 K ซอร์ฟแวร์สังเคราะห์วงจรที่นิยมใช้ เช่น โปรแกรม Leonardo Spectrum ของบริษัท Exemplar Logic ซึ่งในขั้นตอนนี้ซอร์ฟแวร์สังเคราะห์วงจรจะแปลงโค้ดและทำการออปติไมซ์ (Optimization) เพื่อให้ได้วงจรตามเทคโนโลยีที่เลือกใช้ นอกจากนี้ยังสามารถกำหนดข้อบังคับสำหรับวงจรได้ เช่น ข้อบังคับในเรื่องของเวลา (Time Constraints) หรือข้อบังคับในเรื่องของพื้นที่ ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอนออปติไมซ์เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการออปติไมซ์คือการเทียบ (Mapping) วงจรให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างสถาปัตยกรรมภายในอุปกรณ์ FPGA ในกรณีของ Xilinx ตระกูล XC4000 และ Altera ตระกูล FLEX 10 K จะเทียบโดยใช้วิธี LUT (Look Up Table) เมื่อทำการสังเคราะห์วงจรเสร็จแล้วซอร์ฟแวร์สังเคราะห์วงจรจะมีการรายงานผลว่าวงจรที่ออกแบบไปนั้นเป็นอย่างไร เช่น มีความหน่วง (Delay) เท่าไร ใช้ทรัพยากรต่าง ๆ ใน FPGA อะไรบ้าง เป็นต้น

การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ให้เป็นส่วนย่อย ๆ สำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่น ๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้เพื่อช่วยลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอร์ฟแวร์ทำโดยซอร์ฟแวร์จะเทียบส่วนประกอบของวงจร เช่น เกท (Gate), ฟลิปฟลอป (Flipflop) ลงในทรัพยากรต่าง ๆ ที่มีอยู่ภายในอุปกรณ์ FPGA (CLBs, IOBs, BUFT) และ (Edge Decoder) หลังจากทำขั้นตอนนี้เสร็จแล้วสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนซอร์ฟแวร์ที่ใช้ในขั้นตอนนี้ขึ้นอยู่กับตัว FPGA ที่ใช้งาน เช่น FPGA ของบริษัท Xilinx จะใช้ Xilinx ISE Webpack 8.2i ซึ่งซอร์ฟแวร์ตัวนี้จะรวมเอาซอร์ฟแวร์ย่อยอื่น ๆ อีก เพื่อให้การทำ PPR (Partitioning, Placement and Routing) เป็นไปอย่างต่อเนื่อง ส่วน FPGA ของบริษัท Altera จะใช้ Altera MAX+II

การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควรจะอยู่ในตำแหน่งใดในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่น วงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทาง (Route) ได้ง่ายหรือช่วยลดความหน่วง

จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือตัว Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่าง ๆ ภายในอุปกรณ์ FPGA เช่น ระหว่าง CLBs หรือระหว่าง CLBs กับ IOBs ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอร์ฟแวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมดหรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับโดยสามารถทำขั้นตอนนี้ได้โดยใช้ซอร์ฟแวร์เช่นกันหรือทำการเชื่อมต่อสัญญาณด้วยตัวเอง (Manual Layout) ก็ได้แต่ทางที่ดีควรใช้ซอร์ฟแวร์ทำดีกว่าโดยให้ทำการค้นหาเส้นทางหลาย ๆ ครั้งเพื่อหาครั้งที่ดีที่สุด นอกจากนั้นการกำหนดข้อบังคับทางเวลา (Time Constraints) จะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่วงจรผ่านขั้นตอนต่าง ๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement and Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลดลงในอุปกรณ์ FPGA ได้แล้วในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้ให้เป็นข้อมูลวงจร (Configuration data) ซึ่งอยู่ในรูปของบิทสตรีม (Bit-Stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามวงจรที่ออกแบบไว้

จากที่อธิบายมาทั้งหมดจะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้น ทำได้สะดวกกว่าการทำ ASIC มากเพราะใช้เวลาน้อยกว่ามาก ส่วนสำคัญที่ใช้ในการทำ FPGA คือซอร์ฟแวร์ที่ใช้ตั้งแต่การเขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงในอุปกรณ์ FPGA ซึ่งซอร์ฟแวร์ที่ใช้ต้องเป็นซอร์ฟแวร์ที่ใช้ทำงานต่อเนื่องกัน

การเชื่อมต่อบอร์ดเอฟพีจีเอกับคอมพิวเตอร์

การออกแบบตัวกำเนิดสัญญาณแบบดิจิตอลโดยใช้ FPGA เป็นการออกแบบในคอมพิวเตอร์ด้วยภาษาวีเอชดีแอลจากโปรแกรม ISE WebPack และทำการ Simulated ด้วยโปรแกรม ModelSim โดยทำการสังเคราะห์ลงบอร์ดเอฟพีจีเอผ่านพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานจากโปรแกรม MATLAB

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 1 มิติ ขนาด 3x3

```
clear all
fs=10000;
f=2000;
f1=100;
n=0:9998;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
 P=pascal(3,1);
for s=1:3:9999;
u=P*sigt(s:s+2,1);
w(s,1)=u(1,1)/4;
w(s+1,1)=u(2,1)/4;
w(s+2,1)=u(3,1)/4;
end
plot(n/10000,w','LineWidth',3);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,sigt','green','LineWidth',2);
 plot(n/10000,sig,'black','LineWidth',2);
 plot(n/10000,sig1,'red','LineWidth',2);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 1 มิติ ขนาด 4x4

```
clear all
fs=10000;
f=2000;
f1=100;
n=0:10000-1;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
 P=pascal(4,1);
for s=1:4:10000;
u=(P*sigt(s:s+3,1));
w(s,1)=u(1,1)/sum(abs(P(4,:)));
w(s+1,1)=u(2,1)/sum(abs(P(4,:)));
w(s+2,1)=u(3,1)/sum(abs(P(4,:)));
w(s+3,1)=u(4,1)/sum(abs(P(4,:)));
end
 plot(n/10000,sigt','green','LineWidth',1);
hold on
 plot(n/10000,sig,'black','LineWidth',1);
 plot(n/10000,sig1,'red','LineWidth',1);
 plot(n/10000,w','LineWidth',2);axis([0 0.01 -2 2]);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 1 มิติ ขนาด 5x5

```
clear all
fs=10000;
f=2000;
f1=100;
n=0:10000-1;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
 P=pascal(5,1);
for s=1:5:10000;
u=(P*sigt(s:s+4,1));
w(s,1)=u(1,1)/sum(abs(P(5,:)));
w(s+1,1)=u(2,1)/sum(abs(P(5,:)));
w(s+2,1)=u(3,1)/sum(abs(P(5,:)));
w(s+3,1)=u(4,1)/sum(abs(P(5,:)));
w(s+4,1)=u(5,1)/sum(abs(P(5,:)));
end
plot(n/10000,w','LineWidth',3);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,sigt','green','LineWidth',2);
 plot(n/10000,sig,'black','LineWidth',2);
 plot(n/10000,sig1,'red','LineWidth',2);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 1 มิติ ขนาด 6x6

```
clear all
fs=10000;
f=2000;
f1=100;
n=0:10000-5;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
 P=pascal(6,1);
for s=1:6:10000-5;
u=(P*sigt(s:s+5,1));
w(s,1)=u(1,1)/sum(abs(P(6,:)));
w(s+1,1)=u(2,1)/sum(abs(P(6,:)));
w(s+2,1)=u(3,1)/sum(abs(P(6,:)));
w(s+3,1)=u(4,1)/sum(abs(P(6,:)));
w(s+4,1)=u(5,1)/sum(abs(P(6,:)));
w(s+5,1)=u(6,1)/sum(abs(P(6,:)));
end
plot(n/(10000-5),w','LineWidth',3);axis([0 0.01 -2 2]);
hold on
 plot(n/(10000-5),sigt','green','LineWidth',2);
 plot(n/(10000-5),sig,'black','LineWidth',2);
 plot(n/(10000-5),sig1,'red','LineWidth',2);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานของวงจรกรองป่าสคาล 1 มิติ ขนาด 3x3

```
clear all
fs=10000;
f=2500;
f1=100;
n=0:10000;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
P=PASCAL(3,1);
for s=1:10000-1;
u=(P(3,:)*sigt(s:s+2,1))/4;
w(s+2,1)=u(1,1);
end
p3=sig;
p1=sig1;
p2=sigt;
plot(n/10000,p2','green','LineWidth',1);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,p3,'black','LineWidth',1);
 plot(n/10000,p1,'red','LineWidth',1);
 plot(n/10000,w','blue','LineWidth',1);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

การจำลองการทำงานของวงจรกรองป่าสคาล 1 มิติ ขนาด 4x4

```
clear all
fs=10000;
f=2500;
f1=100;
n=0:10000;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
P=pascal(4,1);
w=zeros(10000,1);
for s=1:10000-2;
u=(P(4,:)*sigt(s:s+3,1))/sum(abs(P(4,:)));
w(s+3,1)=u(1,1);
end
p3=sig;
p1=sig1;
p2=sigt;
plot(n/10000,p2','green','LineWidth',1);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,p3,'black','LineWidth',1);
 plot(n/10000,p1,'red','LineWidth',1);
 plot(n/10000,w','blue','LineWidth',1);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ ขนาด 5x5

```
clear all
fs=10000;
f=2500;
f1=100;
n=0:10000;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';
P=pascal(5,1);
for s=1:10000-3;
u=(P(5,:)*sigt(s:s+4,1)/sum(abs(P(5,:))));
w(s+4,1)=u(1,1);
end
p3=sig;
p1=sig1;
p2=sigt;
plot(n/10000,p2','green','LineWidth',1);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,p3,'black','LineWidth',1);
 plot(n/10000,p1,'red','LineWidth',1);
 plot(n/10000,w','blue','LineWidth',1);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

การจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ ขนาด 6x6

```
clear all
fs=10000;
f=2500;
f1=100;
n=0:10000;
sig=sin(2*pi*f*(n/fs));
sig1=sin(2*pi*f1*(n/fs));
 sigt=sig+sig1;
 sigt=sigt';P=pascal(6,1);
for s=1:10000-4;
u=(P(6,:)*sigt(s:s+5,1)/sum(abs(P(6,:))));
w(s+5,1)=u(1,1);
end
p3=sig;
p1=sig1;
p2=sigt;
plot(n/10000,p2','green','LineWidth',1);axis([0 0.01 -2 2]);
hold on
 plot(n/10000,p3,'black','LineWidth',1);
 plot(n/10000,p1,'red','LineWidth',1);
 plot(n/10000,w','blue','LineWidth',1);
hold off
xlabel('time (s)')
ylabel('Amplitude')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 2 มิติ ขนาด 3x3

```
clear all
x=imread('cameraman.tif');
x=resizem(x,[128,128]);
n=3;
[m1,n1] = size(x);
m=3;
x=double(x);
y=PASCAL(3,1);
e=3;
for e1=1:fix(m1/3);
h=0;
for k=1:fix(m1/3);
  r=  [x(e-2:e,(k*3)-2:(k*3))];
f=y*r*y';
h=h+3;  t=0;
pic(e-(n-1):e-(n-3),h-2:h)=f(1:3,1:3);
end
e=e+3;
end
pic1=uint8(pic);
figure;imshow(pic1);
```

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 2 มิติ ขนาด 4x4

```
clear all
x=imread('cameraman.tif');
x=resizem(x,[128,128]);
n=4;
[m1,n1] = size(x);
m=4;
x=double(x);
y=PASCAL(4,1);
e=4;
for e1=1:fix(m1/4);
h=0;
for k=1:fix(m1/4);
  r=  [x(e-3:e,(k*4)-3:(k*4))];
f=y*r*y';
h=h+4;  t=0;
pic(e-(n-1):e-(n-4),h-3:h)=f(1:4,1:4);
end
e=e+4;
end
pic1=uint8(pic);
figure;imshow(pic1);
```

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 2 มิติ ขนาด 5x5

```
clear all
x=imread('cameraman.tif');
x=resizem(x,[128,128]);
n=5;
[m1,n1] = size(x);
m=5;
x=double(x);
y=PASCAL(5,1);
e=5;
for e1=1:fix(m1/5);
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
h=0;
for k=1:fix(m1/5);
  r=  [x(e-4:e,(k*5)-4:(k*5))];
f=y*r*y';
h=h+5;  t=0;
pic(e-(n-1):e-(n-5),h-4:h)=f(1:5,1:5);
end
e=e+5;
end
pic1=uint8(pic);
figure;imshow(pic1);
```

การจำลองการทำงานของการแปลงปาสคาลเต็มหน่วย 2 มิติ ขนาด 6x6

```
clear all
x=imread('cameraman.tif');
x=resizem(x,[128,128]);
n=6;
[m1,n1] = size(x);
m=6;
x=double(x);
y=PASCAL(6,1);
e=6;
for e1=1:fix(m1/6);
h=0;
for k=1:fix(m1/6);
  r=  [x(e-5:e,(k*6)-5:(k*6))];
f=y*r*y';
h=h+6;  t=0;
pic(e-(n-1):e-(n-6),h-5:h)=f(1:6,1:6);
end
e=e+6;
end
pic1=uint8(pic);
figure;imshow(pic1);
```

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 3x3

```
clear all
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
[m1,n1] = size(x);
n=3;
m=3;
y=PASCAL(3,1);
for e=n:m1;
for k=n:m1;
f1=(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3) ...
   *x(m-(n-3),k-(n-1))+y(n,2)*(x(m-(n-1),k-(n-2))+y(n,2) ...
   *x(m-(n-2),k-(n-2))+y(n,3)*x(m-(n-3),k-(n-2)))+y(n,3)...
   *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
   *x(m-(n-3),k-(n-3))));
    a=1:m1;
    i111(e,k)=f1;
end
m=m+1;
end
ii11=uint8(i111);
figure;imshow(ii11,'notruesize');
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 4x4

```
clear all
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
n=4;
[m1,n1] = size(x);
m=4;
y=PASCAL(4,1);
y=[y(4,1)*y(4,:);y(4,2)*y(4,:);y(4,3)*y(4,:);y(4,4)*y(4,:)];
y14=sum(sum(abs(y)));
y24=y14/2;
%y=y/8;
for e=n:m1;
for k=n:m1;
fcor2 =(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3) ...
       *x(m-(n-3),k-(n-1))+y(n,4)*x(m-(n-4),k-(n-1))+y(n,2)...
       *(x(m-(n-1),k-(n-2))+y(n,2)*x(m-(n-2),k-(n-2))+y(n,3)...
       *x(m-(n-3),k-(n-2))+y(n,4)*x(m-(n-4),k-(n-2)))+y(n,3)...
       *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
       *x(m-(n-3),k-(n-3))+y(n,4)*x(m-(n-4),k-(n-3)))+y(n,4)...
       *(x(m-(n-1),k-(n-4))+y(n,2)*x(m-(n-2),k-(n-4))+y(n,3)...
       *x(m-(n-3),k-(n-4))+y(n,4)*x(m-(n-4),k-(n-4))));
    a=1:m1;
    i112(e,k)=fcor2;
   end
m=m+1;
end
ii2=uint8(i112);
figure;imshow(ii2,'notruesize');
```

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 5x5

```
clear all
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
n=5;
[m1,n1] = size(x);
m=5;
y=PASCAL(5,1);
y=[y(5,1)*y(5,:);y(5,2)*y(5,:);y(5,3)*y(5,:);y(5,4)*y(5,:);...
  y(5,5)*y(5,:)];
y15=sum(sum(abs(y)));
y25=y15/2;
for e=n:m1;
for k=n:m1;
fcor3 =(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3)...
       *x(m-(n-3),k-(n-1))+y(n,4)*x(m-(n-4),k-(n-1))+y(n,5)...
       *x(m-(n-5),k-(n-1))+y(n,2)*(x(m-(n-1),k-(n-2))+y(n,2)...
       *x(m-(n-2),k-(n-2))+y(n,3)*x(m-(n-3),k-(n-2))+y(n,4)...
       *x(m-(n-4),k-(n-2))+y(n,5)*x(m-(n-5),k-(n-2)))+y(n,3)...
       *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
       *x(m-(n-3),k-(n-3))+y(n,4)*x(m-(n-4),k-(n-3))+y(n,5)...
       *x(m-(n-5),k-(n-3)))+y(n,4)*(x(m-(n-1),k-(n-4))+y(n,2)...
       *x(m-(n-2),k-(n-4))+y(n,3)*x(m-(n-3),k-(n-4))+y(n,4)...
       *x(m-(n-4),k-(n-4))+y(n,5)*x(m-(n-5),k-(n-4)))+y(n,5)...
       *(x(m-(n-1),k-(n-5))+y(n,2)*x(m-(n-2),k-(n-5))+y(n,3)...
       *x(m-(n-3),k-(n-5))+y(n,4)*x(m-(n-4),k-(n-5))+y(n,5)...
       *x(m-(n-5),k-(n-5))));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
a=1:m1;
i113(e,k)=fcor3;
end
m=m+1;
end
ii3=uint8(i113); figure;imshow(ii3,'notruesize');
```

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 6x6

```
clear a11
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
n=6;
[m1,nl] = size(x);
m=6;
y=PASCAL(6,1);
y=[y(6,1)*y(6,:);y(6,2)*y(6,:);y(6,3)*y(6,:);y(6,4)*y(6,:);...
  y(6,5)*y(6,:);y(6,6)*y(6,:)];
y16=sum(sum(abs(y)));
y26=y16/2;
for e=1:m1-(n-1);
for k=n:m1;
f4=(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3)...
   *x(m-(n-3),k-(n-1))+y(n,4)*x(m-(n-4),k-(n-1))+y(n,5)...
   *x(m-(n-5),k-(n-1))+y(n,6)*x(m-(n-6),k-(n-1))+y(n,2)...
   *(x(m-(n-1),k-(n-2))+y(n,2)*x(m-(n-2),k-(n-2))+y(n,3)...
   *x(m-(n-3),k-(n-2))+y(n,4)*x(m-(n-4),k-(n-2))+y(n,5)...
   *x(m-(n-5),k-(n-2))+y(n,6)*x(m-(n-6),k-(n-2)))+y(n,3)...
   *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
   *x(m-(n-3),k-(n-3))+y(n,4)*x(m-(n-4),k-(n-3))+y(n,5)...
   *x(m-(n-5),k-(n-3))+y(n,6)*x(m-(n-6),k-(n-3)))+y(n,4)...
   *(x(m-(n-1),k-(n-4))+y(n,2)*x(m-(n-2),k-(n-4))+y(n,3)...
   *x(m-(n-3),k-(n-4))+y(n,4)*x(m-(n-4),k-(n-4))+y(n,5)...
   *x(m-(n-5),k-(n-4))+y(n,6)*x(m-(n-6),k-(n-4)))+y(n,5)...
   *(x(m-(n-1),k-(n-5))+y(n,2)*x(m-(n-2),k-(n-5))+y(n,3)...
   *x(m-(n-3),k-(n-5))+y(n,4)*x(m-(n-4),k-(n-5))+y(n,5)...
   *x(m-(n-5),k-(n-5))+y(n,6)*x(m-(n-6),k-(n-5)))+y(n,6)...
   *(x(m-(n-1),k-(n-6))+y(n,2)*x(m-(n-2),k-(n-6))+y(n,3)...
   *x(m-(n-3),k-(n-6))+y(n,4)*x(m-(n-4),k-(n-6))+y(n,5)...
   *x(m-(n-5),k-(n-6))+y(n,6)*x(m-(n-6),k-(n-6))));
a=1:m1;
i114(e,k)=f4;
end
m=m+1;
end
ii4=uint8(i114); figure;imshow(ii4,'notruesize');
```

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 7x7

```
clear all
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
n=7;
[m1,nl] = size(x);
m=7;
y=PASCAL(7,1);
y=[y(7,1)*y(7,:);y(7,2)*y(7,:);y(7,3)*y(7,:);y(7,4)*y(7,:);...
   y(7,5)*y(7,:);y(7,6)*y(7,:);y(7,7)*y(7,:)];
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ ที่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y17=sum(sum(abs(y)));
y27=y17/2;
for e=n:m1;
for k=n:m1;
f5=(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3)...
   *x(m-(n-3),k-(n-1))+y(n,4)*x(m-(n-4),k-(n-1))+y(n,5)...
   *x(m-(n-5),k-(n-1))+y(n,6)*x(m-(n-6),k-(n-1))+y(n,7)...
   *x(m-(n-7),k-(n-1))+y(n,2)*(x(m-(n-1),k-(n-2))+y(n,2)...
   *x(m-(n-2),k-(n-2))+y(n,3)*x(m-(n-3),k-(n-2))+y(n,4)...
   *x(m-(n-4),k-(n-2))+y(n,5)*x(m-(n-5),k-(n-2))+y(n,6)...
   *x(m-(n-6),k-(n-2))+y(n,7)*x(m-(n-7),k-(n-2)))+y(n,3)...
   *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
   *x(m-(n-3),k-(n-3))+y(n,4)*x(m-(n-4),k-(n-3))+y(n,5)...
   *x(m-(n-5),k-(n-3))+y(n,6)*x(m-(n-6),k-(n-3))+y(n,7)...
   *x(m-(n-7),k-(n-3)))+y(n,4)*(x(m-(n-1),k-(n-4))+y(n,2)...
   *x(m-(n-2),k-(n-4))+y(n,3)*x(m-(n-3),k-(n-4))+y(n,4)...
   *x(m-(n-4),k-(n-4))+y(n,5)*x(m-(n-5),k-(n-4))+y(n,6)...
   *x(m-(n-6),k-(n-4))+y(n,7)*x(m-(n-7),k-(n-4)))+y(n,5)...
   *(x(m-(n-1),k-(n-5))+y(n,2)*x(m-(n-2),k-(n-5))+y(n,3)...
   *x(m-(n-3),k-(n-5))+y(n,4)*x(m-(n-4),k-(n-5))+y(n,5)...
   *x(m-(n-5),k-(n-5))+y(n,6)*x(m-(n-6),k-(n-5))+y(n,7)...
   *x(m-(n-7),k-(n-5)))+y(n,6)*(x(m-(n-1),k-(n-6))+y(n,2)...
   *x(m-(n-2),k-(n-6))+y(n,3)*x(m-(n-3),k-(n-6))+y(n,4)...
   *x(m-(n-4),k-(n-6))+y(n,5)*x(m-(n-5),k-(n-6))+y(n,6)...
   *x(m-(n-6),k-(n-6))+y(n,7)*x(m-(n-7),k-(n-6)))+y(n,7)...
   *(x(m-(n-1),k-(n-7))+y(n,2)*x(m-(n-2),k-(n-7))+y(n,3)...
   *x(m-(n-3),k-(n-7))+y(n,4)*x(m-(n-4),k-(n-7))+y(n,5)...
   *x(m-(n-5),k-(n-7))+y(n,6)*x(m-(n-6),k-(n-7))+y(n,7)...
   *x(m-(n-7),k-(n-7))));
a=1:m1;
i115(e,k)=f5;
end
m=m+1;
end
ii5=uint8(i115);
figure;imshow(ii5,'notruesize');
```

การจำลองการทำงานของวงจรกรองปาสคาล 2 มิติ ขนาด 8x8

```
clear all
x=imread('ccameraman.jpg');
x=imresize(x,[128 128]);
x=double(x);
[m1,n1] = size(x);
n=8;
m=8;
pasca1=8;
y=PASCAL(8,1);

y=[y(8,1)*y(8,:);y(8,2)*y(8,:);y(8,3)*y(8,:);y(8,4)*y(8,:);y(8,5)...
  *y(8,:);y(8,6)*y(8,:);y(8,7)*y(8,:);y(8,8)*y(8,:)];
y18=sum(sum(abs(y)));
y28=y18/2;
for e=n:m1;
for k=n:m1;
f6=(x(m-(n-1),k-(n-1))+y(n,2)*x(m-(n-2),k-(n-1))+y(n,3)...
   *x(m-(n-3),k-(n-1))+y(n,4)*x(m-(n-4),k-(n-1))+y(n,5)...
   *x(m-(n-5),k-(n-1))+y(n,6)*x(m-(n-6),k-(n-1))+y(n,7)...
   *x(m-(n-7),k-(n-1))+y(n,8)*x(m-(n-8),k-(n-1))+y(n,2)...
   *(x(m-(n-1),k-(n-2))+y(n,2)*x(m-(n-2),k-(n-2))+y(n,3)...
   *x(m-(n-3),k-(n-2))+y(n,4)*x(m-(n-4),k-(n-2))+y(n,5)...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    *x(m-(n-5),k-(n-2))+y(n,6)*x(m-(n-6),k-(n-2))+y(n,7)...
    *x(m-(n-7),k-(n-2))+y(n,8)*x(m-(n-8),k-(n-2)))+y(n,3)...
    *(x(m-(n-1),k-(n-3))+y(n,2)*x(m-(n-2),k-(n-3))+y(n,3)...
    *x(m-(n-3),k-(n-3))+y(n,4)*x(m-(n-4),k-(n-3))+y(n,5)...
    *x(m-(n-5),k-(n-3))+y(n,6)*x(m-(n-6),k-(n-3))+y(n,7)...
    *x(m-(n-7),k-(n-3))+y(n,8)*x(m-(n-8),k-(n-3)))+y(n,4)...
    *(x(m-(n-1),k-(n-4))+y(n,2)*x(m-(n-2),k-(n-4))+y(n,3)...
    *x(m-(n-3),k-(n-4))+y(n,4)*x(m-(n-4),k-(n-4))+y(n,5)...
    *x(m-(n-5),k-(n-4))+y(n,6)*x(m-(n-6),k-(n-4))+y(n,7)...
    *x(m-(n-7),k-(n-4))+y(n,8)*x(m-(n-8),k-(n-4)))+y(n,5)...
    *(x(m-(n-1),k-(n-5))+y(n,2)*x(m-(n-2),k-(n-5))+y(n,3)...
    *x(m-(n-3),k-(n-5))+y(n,4)*x(m-(n-4),k-(n-5))+y(n,5)...
    *x(m-(n-5),k-(n-5))+y(n,6)*x(m-(n-6),k-(n-5))+y(n,7)...
    *x(m-(n-7),k-(n-5))+y(n,8)*x(m-(n-8),k-(n-5)))+y(n,6)...
    *(x(m-(n-1),k-(n-6))+y(n,2)*x(m-(n-2),k-(n-6))+y(n,3)...
    *x(m-(n-3),k-(n-6))+y(n,4)*x(m-(n-4),k-(n-6))+y(n,5)...
    *x(m-(n-5),k-(n-6))+y(n,6)*x(m-(n-6),k-(n-6))+y(n,7)...
    *x(m-(n-7),k-(n-6))+y(n,8)*x(m-(n-8),k-(n-6)))+y(n,7)...
    *(x(m-(n-1),k-(n-7))+y(n,2)*x(m-(n-2),k-(n-7))+y(n,3)...
    *x(m-(n-3),k-(n-7))+y(n,4)*x(m-(n-4),k-(n-7))+y(n,5)...
    *x(m-(n-5),k-(n-7))+y(n,6)*x(m-(n-6),k-(n-7))+y(n,7)...
    *x(m-(n-7),k-(n-7))+y(n,8)*x(m-(n-8),k-(n-7)))+y(n,8)...
    *(x(m-(n-1),k-(n-8))+y(n,2)*x(m-(n-2),k-(n-8))+y(n,3)...
    *x(m-(n-3),k-(n-8))+y(n,4)*x(m-(n-4),k-(n-8))+y(n,5)...
    *x(m-(n-5),k-(n-8))+y(n,6)*x(m-(n-6),k-(n-8))+y(n,7)...
    *x(m-(n-7),k-(n-8))+y(n,8)*x(m-(n-8),k-(n-8))));
a=1:ml;
il16(e,k)=f6;
end
m=m+1;
end
ii6=uint8(il16);
figure;imshow(ii6,'notruesize');
x=uint8(x);
figure;imshow(x,'notruesize');
```

การปรับระดับความเข้มแสง (Contrast)

```
clear all
clc
x=imread('cameraman.tif');
x=imresize(x,[128 128]);
x=double(x);
con=input('contrast =');
x1=x+con;
x=uint8(x);
imcon=uint8(x1);
figure;imshow(x,'notruesize');
figure;imshow(imcon,'notruesize');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับThreshold

ค่าน้อยกว่าที่กำหนดให้เป็นสีดำ

```
clear all
clc
x=imread('cameraman.tif');
x=imresize(x,[128 128]);
x=double(x);
thres=input('threshold =');
for e=1:128;
for k=1:128;
if x(e,k) <=thres  ;
imthres(e,k)=0    ;
elseif x(e,k)>thres;
        imthres(e,k)=255;
    end
end
end
x=uint8(x);
imthres=uint8(imthres);
figure;imshow(imthres,'notruesize');
```

ค่าน้อยกว่าที่กำหนดให้เป็นสีขาว

```
clear all
clc
x=imread('cameraman.tif');
x=imresize(x,[128 128]);
x=double(x);
thres=input('threshold =');
for e=1:128;
for k=1:128;
if x(e,k) <=thres  ;
        imthres(e,k)=255    ;
    elseif x(e,k)>thres;
        imthres(e,k)=0;
    end
end
end
x=uint8(x);
imthres=uint8(imthres);
figure;imshow(imthres,'notruesize');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม VHDL

โปรแกรมรวมทั้งหมด

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity total is
port (clk : in std_logic;
      p1,p2,p3,p4,p5,p6,p7,p8,dip1,dip2,dip3,dip4,dip5:in std_logic;
      address : OUT STD_LOGIC_VECTOR(14 DOWNTO 0);
      pic_in : in std_logic_vector (7 downto 0);
      pic_out,out_bcd : out std_logic_vector(7 downto 0);
      hor,ver ,clk_out: out std_logic);
end total;

architecture Behavioral of total is
component vga_control
PORT(clock_25Mhz :IN  STD_LOGIC;
     v_on : out std_logic;
     pixel_row, pixel_column : OUT STD_LOGIC_VECTOR(9 DOWNTO 0);
     horiz_sync_out, vert_sync_out   :OUT STD_LOGIC);
end component;
-----------------------------------------------------------------------
component div
PORT( divv : in  STD_LOGIC_VECTOR (3 downto 0);
      row_mod1,column_mod1:OUT STD_LOGIC_VECTOR(9 DOWNTO 0);
      pixel_row, pixel_column : IN STD_LOGIC_VECTOR(9 DOWNTO 0));
end component;
-----------------------------------------------------------------------
component mid
PORT( ro2:in  STD_LOGIC_VECTOR (7 downto 0);
      Co2:in  STD_LOGIC_VECTOR (8 downto 0);
      Pixel2:in  STD_LOGIC_VECTOR (9 downto 0);
      row_address,column_address:OUT STD_LOGIC_VECTOR(8 DOWNTO 0);
      pixel_row, pixel_column : IN STD_LOGIC_VECTOR(9 DOWNTO 0));
end component;
-----------------------------------------------------------------------
component sel
port (ver : in std_logic;
      adr_out : out std_logic_vector (1 downto 0));
end component;
-----------------------------------------------------------------------
component pascal
port (clk : in std_logic;
      sh : in std_logic_vector(1 downto 0);
      bbb: in std_logic_vector (7 downto 0);
      pic : out std_logic_vector (7 downto 0));
end component;
-----------------------------------------------------------------------
component case_mid
Port (sel : in  STD_LOGIC_VECTOR (2 downto 0);
      div : out  STD_LOGIC_VECTOR (3 downto 0);
      ro : out  STD_LOGIC_VECTOR (7 downto 0);
      co : out  STD_LOGIC_VECTOR (8 downto 0);
      pixel : out  STD_LOGIC_VECTOR (9 downto 0);
      row33,colum33: in  STD_LOGIC_VECTOR (8 downto 0);
      addr : out  STD_LOGIC_VECTOR (13 downto 0));
end component;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--------------------------------------------------------------------------
component big
Port (sel_p : out std_logic;
      Seg3,seg4,shift: out std_logic_vector (1 downto 0);
      Seg1,seg2:out std_logic_vector (2 downto 0);
      Dip1,dip2,dip3,dip4,dip5,clk : in  STD_LOGIC;
      out_put : out  STD_LOGIC_VECTOR (7 downto 0);
      pic_a,pic_pas,p : in  STD_LOGIC_VECTOR (7 downto 0);
      sel_soom : out  STD_LOGIC_VECTOR (2 downto 0));
end component;
--------------------------------------------------------------------------
signal von,h,v,px : std_logic;
signal selpic : std_logic_vector (1 downto 0);
signal row,colum,row2,colum2:std_logic_vector (9 downto 0);
signal row3,colum3:std_logic_vector (8 downto 0);
signal pic,pix,aaa2,p,pixx : std_logic_vector (7 downto 0);
signal sel1:std_logic_vector (2 downto 0);
signal pixp,out_bcd1:std_logic_vector (7 downto 0);
signal div1:: STD_LOGIC_VECTOR (3 downto 0);
signal ro1: STD_LOGIC_VECTOR (7 downto 0);
signal co1: STD_LOGIC_VECTOR (8 downto 0);
signal pixel1: STD_LOGIC_VECTOR (9 downto 0);
signal addr1: STD_LOGIC_VECTOR (13 downto 0);
signal di3,di4,sh1: std_logic_vector (1 downto 0);
signal di1,di2: std_logic_vector (2 downto 0);
--------------------------------------------------------------------------
begin
u0 : vga_control
 PORT map (clock_25Mhz => clk,
           v_on      => von,
           pixel_row => row,
           pixel_column => colum,
           horiz_sync_out => h,
           vert_sync_out => v);

u1:case_mid
 Port map( sel => sel1,
           div => div1,
           ro => ro1,
           co => co1,
           pixel => pixel1,
           row33=> row3,
           colum33=> colum3,
           addr => addr1);

u2:div
port map (divv => div1,
          row_mod1=> row2,
          column_mod1=> colum2,
          pixel_row => row,
          pixel_column => colum);

u3:mid
PORT map( ro2=> ro1,
          Co2=> co1,
          Pixel2=> pixel1,
          row_address => row3,
          column_address=> colum3 ,
          pixel_row => row2 ,
          pixel_column => colum2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
u4:pascal
port map (clk => clk,
          sh => sh1,
          bbb => pic_in,
          pic => pixp);

u5:big
port map( seg1=> di1,
          seg2=> di2,
          seg3=> di3,
          seg4 => di4,
          dip1=> dip1,
          dip2=> dip2,
          dip3=> dip3,
          dip4=> dip4,
          dip5=> dip5,
          clk  => clk,
          out_put => pix,
          pic_a => pic_in,
          pic_pas => pixp,
          p => p,
          sel_p => px,
          shift => sh1,
          sel_soom => sel1);
--------------------------------------------------------------------------
hor <= h;
ver <= v;
p <= p8 & p7 & p6 & p5 & p4 & p3 & p2 & p1;
address <= px & addr1;
clk_out <= clk;
pic_out <= pixx  and (von & von & von & von & von & von & von & von);
out_bcd <= not out_bcd1;
--------------------------------------------------------------------------
process
begin
      if row3 > 0 then
        if colum3 > 0 then
           pixx <= pix;
        else pixx <= "00000000";
        end if;
      else pixx <= "00000000";
      end if;
end process;
out_bcd <= "10111111" when di1="000" else
                        "10000110" when di1="001" else
                        "11011011" when di1="010" else
                        "11001111" when di1="011" else
                        "11100110" when di1="100" else
                        "11110001";
end Behavioral;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมVGA controller

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY vga_control IS
       PORT(clock_25Mhz :IN  STD_LOGIC;
            v_on : out std_logic;
            pixel_row, pixel_column:OUT STD_LOGIC_VECTOR(9 DOWNTO 0);
            horiz_sync_out, vert_sync_out :OUT STD_LOGIC);
END vga_control;
ARCHITECTURE a OF vga_control IS
SIGNAL horiz_sync, vert_sync,data,clk : STD_LOGIC;
SIGNAL video_on, video_on_v, video_on_h : STD_LOGIC;
SIGNAL h_count, v_count,row,column:STD_LOGIC_VECTOR(9 DOWNTO 0);
signal counts: std_logic_vector (19 downto 0);
signal pic: std_logic_vector (7 downto 0);
BEGIN
PROCESS
BEGIN
      WAIT UNTIL(clock_25Mhz'EVENT) AND (clock_25Mhz='1');
                  IF (h_count = 799) THEN
                      h_count <= "0000000000";
                  ELSE
                      h_count <= h_count + 1;
                  END IF;

                  IF (h_count <= 755) AND (h_count >= 659) THEN
                      horiz_sync <= '0';
                  ELSE
                      horiz_sync <= '1';
                  END IF;

                  IF (v_count >= 524) AND (h_count >= 699) THEN
                      v_count <= "0000000000";
                  ELSIF (h_count = 699) THEN
                      v_count <= v_count + 1;
                  END IF;

                  IF (v_count <= 494) AND (v_count >= 493) THEN
                      vert_sync <= '0';
                  ELSE
                      vert_sync <= '1';
                  END IF;

                  IF (h_count <= 639) THEN
                      video_on_h <= '1';
                      column <= h_count;
                  ELSE
                      video_on_h <= '0';
                  END IF;

                  .
                  IF (v_count <= 479) THEN
                      video_on_v <= '1';
                      row <= v_count;
                  ELSE
                      video_on_v <= '0';
                  END IF;
END PROCESS;
```

เอกสารนี้ใช้ประโยชน์เพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
video_on <= video_on_H AND video_on_V;
v_on <= video_on;
pixel_row <= row  ;
pixel_column <= column  ;
horiz_sync_out <= horiz_sync;
vert_sync_out <= vert_sync;
END a;
```

โปรแกรมขยายจัดตำแหน่ง

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity case_mid is
    Port ( sel : in  STD_LOGIC_VECTOR (2 downto 0);
           div : out  STD_LOGIC_VECTOR (3 downto 0);
           ro : out  STD_LOGIC_VECTOR (7 downto 0);
           co : out  STD_LOGIC_VECTOR (8 downto 0);
           pixel : out  STD_LOGIC_VECTOR (9 downto 0);
           row33,colum33: in  STD_LOGIC_VECTOR (8 downto 0);
           addr : out  STD_LOGIC_VECTOR (13 downto 0));
end case_mid;

architecture Behavioral of case_mid is
signal s :std_logic_vector(2 downto 0);
begin
PROCESS(sel)
      BEGIN
      s <= sel;
      case sel is
            when "000"=>
            div <= "0001";
            ro <= "00000000";
            co <= "000000000";
            pixel <= "0010000000";
            addr <= row33(6 downto 0) & colum33(6 downto 0);

            when "001"=>
            div <= "0001";
            ro <= "00000000";
            co <= "000000000";
            pixel <= "0100000000";
            addr <= row33(7 downto 1) & colum33(7 downto 1);

            when "010"=>
            div <= "0001";
            ro <= "00000000";
            co <= "000000000";
            pixel <= "1000000000";
            addr <= row33(8 downto 2) & colum33(8 downto 2);

            when "011"=>
            div <= "0010";
            ro <= "00111000";
            co <= "001100000";
            pixel <= "0010000000";
            addr <= row33(6 downto 0) & colum33(6 downto 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            when "100"=>
            div <= "0010";
            ro <= "00000000";
            co <= "000100000";
            pixel <= "0100000000";
            addr <= row33(7 downto 1) & colum33(7 downto 1);
            when "101"=>
            div <= "1000";
            ro <= "00000000";
            co <= "000000000";
            pixel <= "0010000000";
            addr <= row33(6 downto 0) & colum33(6 downto 0);

            when others =>
            div <= "0001";
            ro <= "10110000";
            co <= "100000000";
            pixel <= "0010000000";
            addr <= row33(6 downto 0) & colum33(6 downto 0);
      end case;
      END PROCESS;
end Behavioral;
```

โปรแกรมหารพิกเซล

```
library IEEE;
use  IEEE.STD_LOGIC_1164.all;
use  IEEE.STD_LOGIC_ARITH.all;
use  IEEE.STD_LOGIC_UNSIGNED.all;
ENTITY div IS
PORT( row_mod1,column_mod1:OUT STD_LOGIC_VECTOR(9 DOWNTO 0);
      divv : in  STD_LOGIC_VECTOR (3 downto 0);
         pixel_row, pixel_column : IN STD_LOGIC_VECTOR(9DOWNTO 0));
END div;
ARCHITECTURE a OF div IS
SIGNAL row_mod,column_mod,row,column:integer range -1023 to 1024;
signal div : integer range 0 to 16;
begin
process
begin

if divv = "0001"then
      row_mod1 <= pixel_row;
      column_mod1 <= pixel_column;
elsif divv = "0010"then
      row_mod1 <= '0' & pixel_row(9 downto 1);
      column_mod1 <= '0' & pixel_column(9 downto 1);
else
      row_mod1 <= "000"& pixel_row(9 downto 3);
      column_mod1 <= "000"& pixel_column(9 downto 3);
end if;

end process;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมปาสคาล

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity pascal is
port (clk : in std_logic;
      sh : in std_logic_vector(1 downto 0);
      bbb : in std_logic_vector (7 downto 0);
      v_count,h_count : in std_logic_vector (6 downto 0);
      pic : out std_logic_vector (7 downto 0));
end pascal;

architecture Behavioral of pascal is
signal co:  std_logic_vector(1 downto 0)= "00";
signal m11,m12,m13,m21,m22,m23:std_logic_vector (7 downto 0);
signal m31,m32,m33:std_logic_vector (7 downto 0);
signal reg24bit : std_logic_vector (2079 downto 0);
signal mx1,mx2,mx3:std_logic_vector (23 downto 0);
signal m11_12bit,m12_12bit:std_logic_vector (11 downto 0);
signal m13_12bit,m21_12bit:std_logic_vector (11 downto 0);
signal m22_12bit : std_logic_vector (11 downto 0);
signal m23_12bit,m31_12bit: std_logic_vector (11 downto 0);
signal m32_12bit : std_logic_vector (11 downto 0);
signal m33_12bit : std_logic_vector (11 downto 0);
signal cmp_m11,cmp_m21,cmp_m12,cmp_m22:std_logic_vector(11 downto 0);
signal cmp_m23,cmp_m13:std_logic_vector(11 downto 0);
signal cmp_y1,cmp_y2,cmp_add1_1 :std_logic_vector (11 downto 0);
signal cmp_add1_3,cmp_add1_2 :std_logic_vector (11 downto 0);
signal cmp_add2_1,cmp_add2_2,cmp_add2_3:std_logic_vector(11downto 0);
signal cmp_add1,cmp_add2: std_logic_vector (11 downto 0);
signal add1_1,add1_2,add2_1: std_logic_vector (11 downto 0);
signal add2_2,add1_3: std_logic_vector (11 downto 0);
signal add2_3,add1_y : std_logic_vector (11 downto 0);
signal add2_y,add1,add2,y1,y2,y3,y: std_logic_vector (11 downto 0);

begin
process(clk,sh)
begin

if (clk'event and clk ='1') then
   co <= co + sh;
   if co(0)='0' then
   reg24bit <=   reg24bit (2071 downto 0)& bbb  ;
   end if;
end if;

m11 <= reg24bit(2071 downto 2064);
m12 <= reg24bit (2063 downto 2056);
m13 <= reg24bit (2055 downto 2048);
m21 <= reg24bit (1047 downto 1040);
m22 <= reg24bit (1039 downto 1032);
m23 <= reg24bit (1031 downto 1024);
m31 <= reg24bit (23 downto 16);
m32 <= reg24bit (15 downto 8);
m33 <= reg24bit (7 downto 0);
-----------------12bit-------------------------------------------------
m11_12bit <= "0000"& m11;
m12_12bit <= "0000"& m12;
m13_12bit <= "0000"& m13;
```

เอกสารนี้เป็นการนำเสนอผลงานเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m21_12bit <= "0000"& m21;
m22_12bit <= "0000"& m22;
m23_12bit <= "0000"& m23;
m31_12bit <= "0000"& m31;
m32_12bit <= "0000"& m32;
m33_12bit <= "0000"& m33;
---------------------------bank1-------------------------------------
cmp_m11 <= (not m11_12bit) + 1;
cmp_m21 <= (not m21_12bit) + 1;
add1_1  <= m21_12bit + cmp_m11;
add2_1  <= m31_12bit + cmp_m21;
cmp_add1_1 <= (not add1_1)+1;
y1 <= add2_1+cmp_add1_1;
---------------------------bank2-------------------------------------
cmp_m12 <= (not m12_12bit) + 1;
cmp_m22 <= (not m22_12bit) + 1;
add1_2  <= m22_12bit + cmp_m12;
add2_2  <= m32_12bit + cmp_m22;
cmp_add1_2 <= (not add1_2)+1;
y2 <= add2_2+cmp_add1_2;
---------------------------bank3-------------------------------------
cmp_m13 <= (not m13_12bit) + 1;
cmp_m23 <= (not m23_12bit) + 1;
add1_3  <= m23_12bit + cmp_m13;
add2_3  <= m33_12bit + cmp_m23;
cmp_add1_3 <= (not add1_3)+1;
y3 <= add2_3+cmp_add1_3;
---------------------------final-------------------------------------
cmp_y1 <= (not y1)+ 1;
cmp_y2 <= (not y2)+ 1;
add1  <= y2 +cmp_y1;
add2  <= y3+cmp_y2;
cmp_add1 <= (not add1)+1;
y <= add2+cmp_add1;
---------------------------------------------------------------------------
if y(11)='0' then
      if y >= 255 then
         pic <= "11111111";
      else pic <= y(7 downto 0);
      end if;
else pic <= "00000000";
end if;
END PROCESS;
end Behavioral;
```

โปรแกรมตำแหน่งรูปภาพ

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY mid IS
PORT(ro2:in  STD_LOGIC_VECTOR (7 downto 0);
     Co2:in  STD_LOGIC_VECTOR (8 downto 0);
     Pixel2:in  STD_LOGIC_VECTOR (9 downto 0);
     row_address,column_address : OUT STD_LOGIC_VECTOR(8 DOWNTO 0);
     pixel_row, pixel_column : IN STD_LOGIC_VECTOR(9 DOWNTO 0));
END mid;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ARCHITECTURE a OF mid IS
SIGNAL rp, rc : STD_LOGIC_VECTOR(9 DOWNTO 0);
SIGNAL row_point,column_point,co,ro,roo :integer range -1023 to 1024;
SIGNAL coo,pixel :integer range -1023 to 1024;

begin
process
begin
     roo <= conv_integer(ro2);
     coo <= conv_integer(co2);
     pixel <= conv_integer(pixel2);

     row_point <= roo;
     column_point <= coo;
     co <= column_point + pixel ;
     ro <= row_point + pixel;

IF (pixel_row >= row_point) and (pixel_row <= ro)  THEN
    IF (pixel_column >= column_point) and (pixel_column <= co)  THEN
        row_address <= pixel_row - row_point;
        column_address <= pixel_column - column_point;
    else
        row_address <= "000000000";
        column_address <= "000000000";
        end if;
        end if;
end process;
end a;
```

โปรแกรมเลือกตำแหน่งและขนาดของภาพ

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity big is
Port ( sel_p :out std_logic;
       Seg3,seg4,shift : out std_logic_vector (1 downto 0);
       Seg1,seg2: out std_logic_vector (2 downto 0);
       Dip1,dip2,dip3,dip4,dip5,clk : in  STD_LOGIC;
       out_put : out  STD_LOGIC_VECTOR (7 downto 0);
       pic_a,pic_pas,p : in  STD_LOGIC_VECTOR (7 downto 0);
      sel_soom : out  STD_LOGIC_VECTOR (2 downto 0));
end big;
architecture Behavioral of big is
signal ad_su,bw,dip : std_logic_vector (1 downto 0);
signal soom,se : std_logic_vector (2 downto 0);
signal row3,colum3,ap,aa,ba,bp : std_logic_vector (7 downto 0);
signal addr : std_logic_vector (14 downto 0);
signal cla,cl1a,c2a,clp,c11p,c2p : std_logic_vector (8 downto 0);
signal pic,pix,aaa2,ip,op,out_a,out_pas:std_logic_vector(7downto 0);
begin
process (clk)
begin
    if (clk'event and clk ='1') then
            if dip1='0' then
                  dip <= dip + '1';
            end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if dip2='0' then
                  ad_su <= ad_su + '1';
            end if;

            if dip3='0' then
                  bw <= bw + '1';
            end if;

            if dip4 ='0' then
                  soom <= soom + 1;
            end if;

            if dip5 ='0' then
                  se <= se + '1';
            end if;
-------------------------(+-)-------------------------------------
            if ad_su(0)='0'  then
              if c1a(8)='1' then
                 aa <= "11111111";
              else aa <=  c1a(7 downto 0);
              end if;
            end if;

            if ad_su(0)='1'  then
              if c2a(8)='1' then
                 aa <= "00000000";
              else aa <=  c2a(7 downto 0);
              end if;
            end if;
            if ad_su(0)='0'  then
               if c1p(8)='1' then
                 ap <= "11111111";
               else ap <=  c1p(7 downto 0);
               end if;
            end if;

            if ad_su(0)='1'  then
              if c2p(8) ='1' then
                 ap <= "00000000";
              else ap <=  c2p(7 downto 0);
              end if;
            end if;
-------------------------------binary-----------------------------
            if bw(0)='0' then
               if pic_a <= p  then
                  ba <=  "00000000";
               else ba <= "11111111" ;
               end if;
            elsif bw(0)='1' then
               if pic_a <= p  then
                  ba <=  "11111111";
               else ba <= "00000000" ;
               end if;
            end if;

            if bw(0)='0' then
               if pic_pas <= p  then
                  bp <=  "00000000";
               else bp <= "11111111" ;
               end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            elsif bw(0)='1' then
              if pic_pas <= p  then
                 bp <=  "11111111";
              else bp <= "00000000" ;
              end if;
            end if;
-------------------------------------------------------------------------
case se is
when "000"=>
sel_soom <= soom;
out_put <= aa;

when "001"=>
sel_soom <= soom;
out_put <= ba;

when "010"=>
sel_soom <= "111";
out_put <= pic_pas;
shift <= "00";

when "011"=>
sel_soom <= "011";
out_put <= ap;
shift <= "01";

when "100"=>
sel_soom <= "011";
out_put <= bp;
shift <= "01";

when others =>
sel_soom <= "111";
out_put <= pic_a;
end case;
end if;
end process;

c1a <= ('0' & pic_a) +  ('0' & p(7 downto 0));
c11a <= (not('0' & p )) + 1;
c2a <= ('0' & pic_a) + c11a;

c1p <= ('0' & pic_pas) +  ('0' & p(7 downto 0));
c11p <= (not('0' & p )) + 1;
c2p <= ('0' & pic_pas) + c11p;
-------------------------------------------------------------------------
sel_p <= dip(0);
seg4 <= ad_su;
seg3 <= bw;
seg2 <= soom ;
seg1 <= se ;
end Behavioral;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเลือกเฟรมรูปเคลื่อนไหว

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity sel is
port (ver   : in std_logic;
      adr_out : out std_logic_vector (1 downto 0));
end sel;

architecture a of sel is
SIGNAL  incnt :integer range -255 to 256;
signal address : std_logic_vector (1 downto 0);
begin
process

begin
      WAIT UNTIL(ver'EVENT) AND (ver='1');
            if incnt = 20 then
                  incnt <= 0;
                  address <= address + 1;
            else
                  incnt <= incnt+1;
            end if;
            adr_out <= '1' & address(0);

end process;
end a;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้