

รายงานผลการวิจัยฉบับสมบูรณ์  
โครงการวิจัยที่ได้รับทุนสนับสนุนจากเงินรายได้คณะฯ ปี 2550

การออกแบบระบบประมวลผลสัญญาณภาพเชิงเลขแบบเวลาจริง  
โดยใช้อุปกรณ์ FPGA

RC4  
TA  
1634  
ศ 144 ก

สาขา.....  
เลขทะเบียน..... 84525  
วัน,เดือน,ปี..... 13 ต.ค. 2551

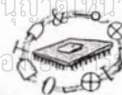
โดย

นายสรวัฒน์ ชิวปรีชา : หัวหน้าโครงการวิจัย

11๐๙51๖๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของ

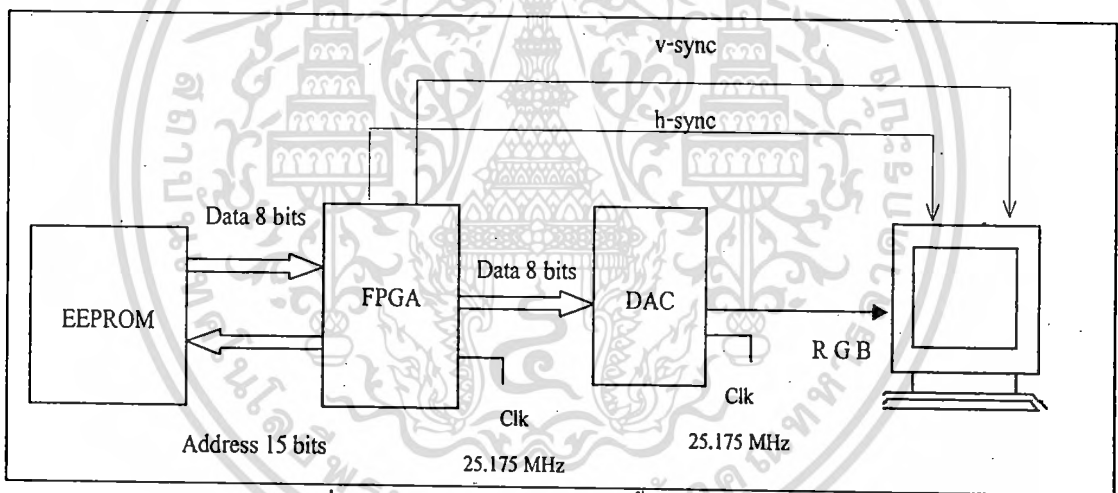


By Sorawat Chiyapreecha  
Telecommunication Eng., Dept.  
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

1 ความเป็นมาและความสำคัญของโครงการ

โครงการวิจัยนี้เป็นการสร้างระบบประมวลผลสัญญาณภาพเชิงเลข (Digital Image Processing System) แบบเวลาจริง (Real-Time) โดยใช้อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์หลักในการออกแบบและสร้างระบบดังกล่าว นอกจากนี้ระบบจะต้องประกอบไปด้วยภาค Input และ ภาค Output โดยในเบื้องต้นจะใช้หน่วยความจำในการเก็บข้อมูลภาพไว้ สำหรับทำหน้าที่สร้างสัญญาณ Input ให้กับระบบ และใช้จอ Monitor แบบ VGA สำหรับการแสดงผลของสัญญาณ Output ที่ผ่านการประมวลผลสัญญาณมา ซึ่งในส่วนของทฤษฎีที่จะนำมาใช้ในการประมวลผลสัญญาณภาพนั้นเรานำการแปลงฟูริเยร์ (Discrete Pascal Transform) ซึ่งได้นำกล่าวไว้ในภาคผนวก มาเป็นอัลกอริทึมในการประมวลผลสัญญาณภาพต่อไป โดยภาพรวมของโครงการที่ได้ดำเนินการไปแล้วแสดงได้ดังรูปที่ 1



รูปที่ 1.1 โครงสร้างโดยรวมของชิ้นงานในโครงการ

2 วัตถุประสงค์โครงการวิจัย

- 2.1 เพื่อศึกษาการทำงานและสร้างระบบประมวลผลสัญญาณภาพเชิงเลขแบบเวลาจริงโดยใช้ FPGA ทำหน้าที่เป็นตัวประมวลผลสัญญาณ รวมทั้งรับค่าสัญญาณภาพอินพุตและแสดงผลสัญญาณภาพเอาต์พุตทางหน้าจอ VGA
- 2.2 ศึกษากระบวนการแปลงสัญญาณภาพจากรูปแบบที่เป็นสัญญาณภาพเชิงอุปมานให้เป็นสัญญาณภาพเชิงเลขสำหรับการนำไปประมวลผล

- 2.3 ศึกษาการแสดงผลสัญญาณภาพที่ผ่านการประมวลผลสัญญาณ โดยจะทำการแสดงผลภาพที่ผ่านการประมวลผลบนหน้าจอ VGA โดยตรง โดยปราศจากการใช้งานคอมพิวเตอร์เพื่อช่วยในการแสดงผล
- 2.4 เพื่อที่จะสามารถนำเอาระบบประมวลผลสัญญาณภาพเชิงเลขแบบเวลาจริงดังกล่าวไปประยุกต์ใช้กับรูปแบบการประมวลผลสัญญาณในหลายๆ รูปแบบ และนำไปประยุกต์ใช้งานด้านต่างๆ ในหลายๆ ด้าน
- 2.5 เพื่อนำความรู้ที่ได้รับจากกระบวนการทำงานวิจัยนี้ไปประยุกต์และปรับใช้ในการเรียนการสอนให้กับนักศึกษา โดยเฉพาะในวิชาที่เกี่ยวกับการประมวลผลสัญญาณและการออกแบบระบบดิจิทัลสำหรับประมวลผลสัญญาณ



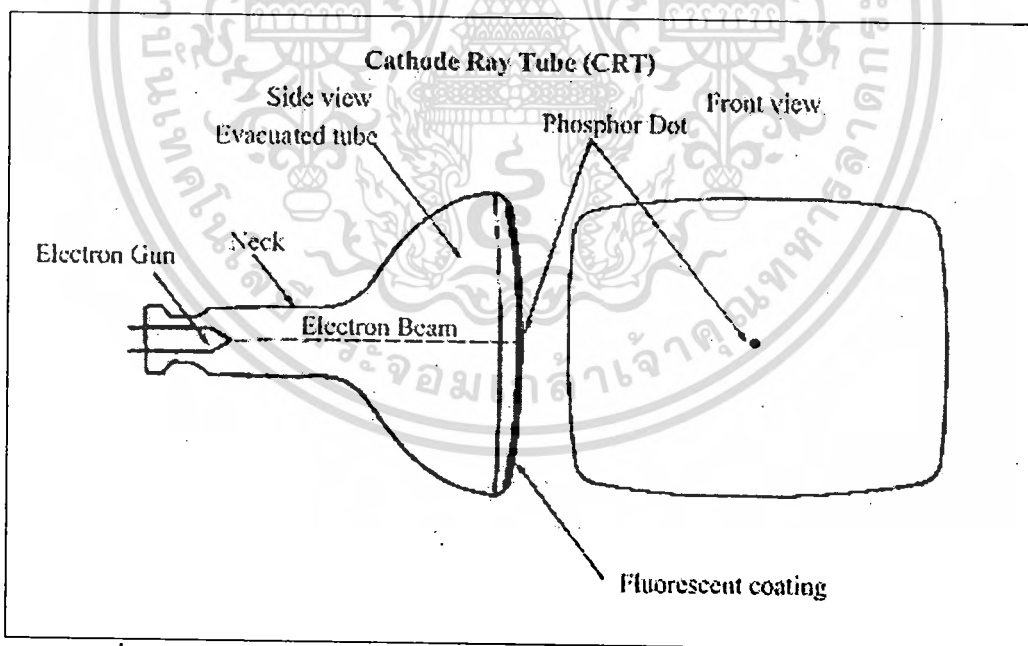
ความรู้เบื้องต้นเกี่ยวกับ VGA + อุปกรณ์ภายนอกอื่นๆ ที่ประกอบกันในโครงการ

การกำเนิดการแสดงผลภาพให้กับจอภาพระบบ VGA (VGA Video Display Generator)

สัญญาณที่ใช้ในระบบภาพ VGA ประกอบด้วย 5 สัญญาณด้วยกัน โดยมี 2 สัญญาณที่เกี่ยวข้องกับระดับสัญญาณลอจิก TTL นั่นคือ สัญญาณ Horizontal Sync และ Vertical Sync ซึ่งใช้สำหรับสังเคราะห์สัญญาณภาพ ส่วนอีก 3 สัญญาณนั้นจะเกี่ยวข้องกับสัญญาณอนาล็อก โดยมีค่า  $0.7 - 1.0 V_{p-p}$  จะใช้ในควบคุมสัญญาณสี ซึ่งสัญญาณสีประกอบด้วยสีแดง, สีเขียว และสีน้ำเงิน สัญญาณสีเหล่านี้เรียกรวมว่า สัญญาณ RGB

เทคโนโลยีการแสดงผลภาพจอ (Video Display Technology)

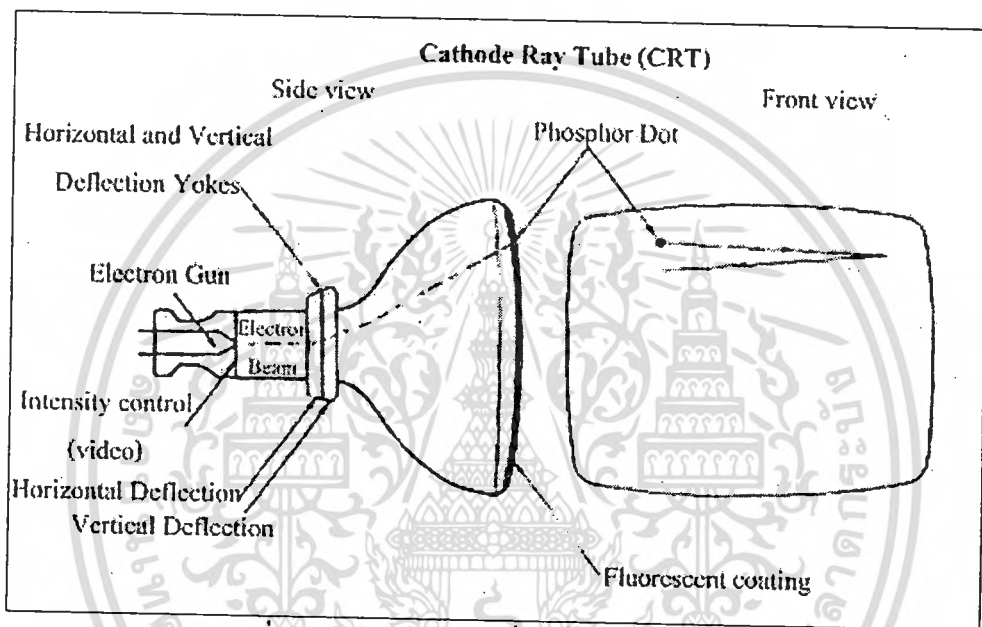
เทคโนโลยีที่ใช้โดยส่งการแสดงผลเป็นสัญญาณภาพนิ่งที่เหมือนธรรมชาติในรูปแบบของสัญญาณภาพ จะใช้หลอดภาพอิเล็กตรอนิกส์ (Cathode ray tube: CRTs) ซึ่งเกิดขึ้นครั้งแรกเมื่อปี ค.ศ. 1897 โดยนักวิทยาศาสตร์ชาวเยอรมันชื่อ Karl Ferdinand Braun โดยพัฒนาเป็นเครื่องมือที่เรียกว่า ออสซิลโลสโคป (Oscilloscope) ที่ใช้เป็นเครื่องวัดค่าสัญญาณทางไฟฟ้า และเริ่มผลิตออกมาในเชิงพาณิชย์ครั้งแรกในปี ค.ศ. 1922 ในรูปแบบของจอโทรทัศน์จนถึงทุกวันนี้ CRTs เป็นอุปกรณ์ที่ใช้กันแพร่หลายทั่วไปส่วนมากจะมีใช้ในรูปแบบชุดโทรทัศน์บ้างหรือใช้เป็นตัวเชื่อมต่อในการแสดงผลของระบบคอมพิวเตอร์ก็ดี



รูปที่ 2.1 แสดงถึงอุปกรณ์ CRT ซึ่งก็คือหลอดแก้วที่ยังอิเล็กตรอนไปยังฉากด้านหน้า

และฉากนี้เองที่พื้นผิวด้านในจะเคลือบสารฟลูออเรสเซนต์อยู่ และที่ตรงส่วนท้ายของหลอดภาพ จะมีปืนยิงลำอิเล็กตรอน เมื่อปืนนี้ยิงไปกระตุ้นที่ฉากก็จะทำให้เกิดจุดสว่างขึ้น

ซึ่งหลอดอิเล็กตรอนนี้จะยิงลำอิเล็กตรอน (Electron beam) กวาดสแกนบนฉากตามแนวนอนที่ต่อเนื่องกัน และเพื่อที่จะกำเนิดมาเป็นภาพ จะมีแกนเหล็กสำหรับเบี่ยงเบนลำอิเล็กตรอน (Deflection yoke) ขดลวดเลี้ยวเบนนี้ จะแบ่งเป็นแกนแนวตั้งกับแนวนอน (จะอยู่ที่คอของหลอด CRT) โดยจะอาศัยสนามแม่เหล็ก (หรืออาจจะใช้ สนามไฟฟ้าสถิตก็ได้ แต่ไม่เป็นที่นิยม) เบี่ยงเบนลำอิเล็กตรอนให้ไปตามตำแหน่งที่ต้องการบนฉากของจอภาพ ส่วนข้อมูลของสัญญาณ RGB ที่ออกมาเป็นสัญญาณภาพจะใช้ควบคุมความแรงของอิเล็กตรอน และแสงที่เกิดขึ้น บนจอภาพนั้นเกิดจากการที่มีสัญญาณภาพเข้ามาซึ่งจะทำให้ลำอิเล็กตรอนทำงาน และสามารถพุ่งไปปะทะจุดสี เรืองแสงบนฉากของหลอดภาพ CRT ซึ่งบนฉากของ CRT จะเคลือบด้วยสารเรืองแสง 3 สี (สีแดง, สีเขียว และสี น้ำเงิน)



รูปที่ 2.2 การควบคุมการเคลื่อนที่ของลำอิเล็กตรอน

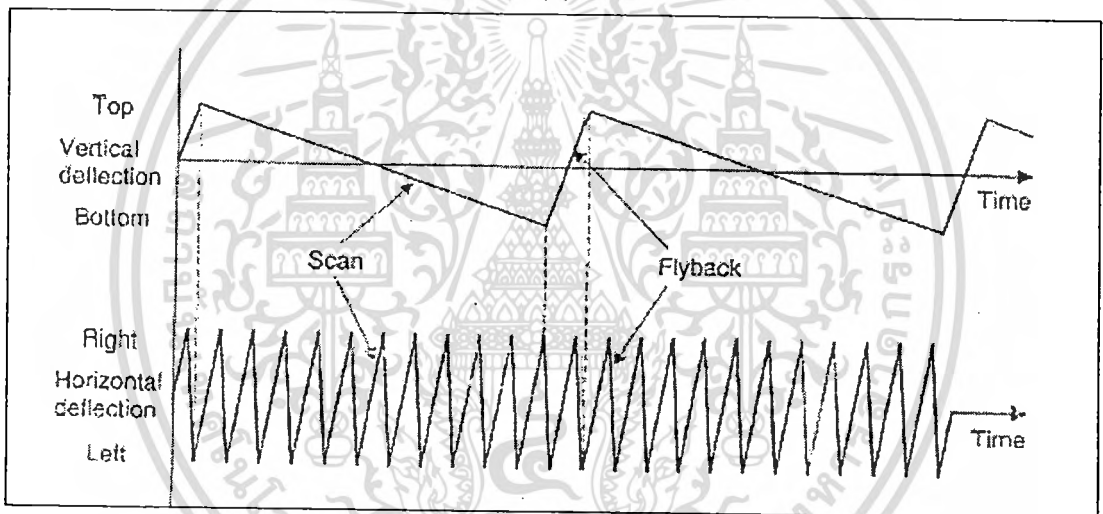
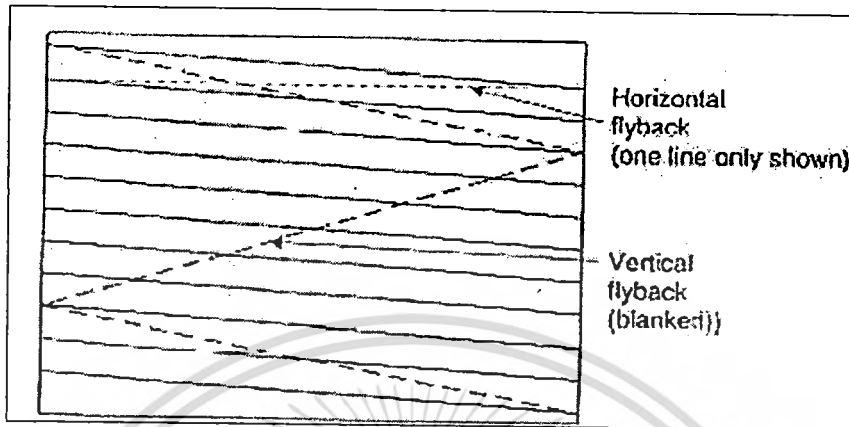
### การกวาดตรวจภาพ (Scanning)

การที่จะนำภาพ 2 มิติเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งทำได้โดยทำภาพนั้นเป็นสัญญาณไฟฟ้าซึ่งจะง่ายต่อการเคลื่อนย้าย แต่ปัญหาคือจะทำการเปลี่ยนภาพ 2 มิติไปเป็นโวลต์เดจที่เปลี่ยนตามเวลาปัญหานี้แก้ได้โดย ทฤษฎีของการกวาดตรวจ

วิธีการสแกนของ CRT ที่นิยมใช้กันอย่างแพร่หลายในการออกแบบทางการค้าโทรทัศน์และจอภาพ แสดงผล มีชื่อเรียกว่า การตรวจกวาดแบบราสเตอร์ (Raster scan) ซึ่งวิธีนี้จะต้องมีการให้กำเนิดขงกทั้งในแนวแกน ตั้งและแนวนอนเพื่อให้ลำอิเล็กตรอนสามารถเคลื่อนที่ได้

ในการสแกนแบบนี้จะเริ่มทำจากซ้ายไปขวา และจากข้างบนลงไปข้างล่างของจอภาพ โดยที่ลำอิเล็กตรอน จะถูกเลี้ยวเบนให้ไปอยู่ทางมุมซ้ายบนและเมื่อกวาดไปทางขวาของฉาก ลำอิเล็กตรอนก็จะสะบัดกลับโดยตอนที่ กลับนั้นก็จะมีการเลื่อนตำแหน่งลงไปด้วย การสะบัดกลับนี้เรียกว่า Retrace หรือ

fly back ซึ่งในส่วนของการสลับกลับของลำอิเล็กตรอนนั้นความเข้มของลำอิเล็กตรอนจะลดลง เมื่อลดความเข้มลงอย่างเพียงพอแล้วเวลาที่เคลือบด้วยฟลูออเรสเซนต์ก็จะไม่ถูกกระตุ้นทำให้ไม่เกิดแสงสว่างจึงมองไม่เห็นเส้นที่ลากกลับ ดังรูป



รูปที่ 2.3 การสแกนภาพแบบราสเตอร์

(ก) การเคลื่อนที่ของการสแกนภาพ

(ข) กราฟแสดงความสัมพันธ์ของการเคลื่อนที่ของการสแกนกับเวลา

### การซิงโครไนซ์ (Synchronization)

สิ่งสำคัญที่ทำให้การสแกนภาพหนึ่งได้ถูกต้องนั้นเป็นหน้าที่ของสัญญาณซิงค์ซึ่งแบ่งได้เป็นสัญญาณ Horizontal และ Vertical sync นั้นทำงานไปพร้อม ๆ กันได้นั้นเพราะสัญญาณนี้จะส่งข้อมูล timing แนบไปกับสัญญาณ video

การซิงโครไนซ์ในแนวแกนอนและแกนตั้ง

- Horizontal Sync จะเริ่มทำงานที่การเริ่ม retrace หรือจบการ trace จะไม่ทำตอนเริ่ม trace
- Vertical Sync เริ่มทำงานที่การเริ่ม retrace ในแนวแกนตั้ง (ก็คือจะทำงานตอน beam อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้า



ที่ล่างขวาของจอภาพ)

ถ้าหากไม่มีสัญญาณ Vertical sync จะทำให้ภาพที่ถูกสร้างทวนซ้ำขึ้นมาจะไม่สามารถคงตำแหน่งในแนวแกนตั้ง ภาพจะวิ่งขึ้น-ลง

แต่ถ้าหากสัญญาณ Horizontal sync ไม่ซิงก์กันจะทำให้ภาพนั้นจะเลื่อนซ้าย-ขวา และบางส่วนของเหมือนภาพถูกแยกออก

โดยทั่วไปแล้วจอภาพส่วนใหญ่จะใช้สัญญาณอนาล็อก RGB มาตรฐานที่  $0.7 V_{p-p}$  ส่วนสัญญาณซิงก์ก็จะขึ้นอยู่กับประเภทลักษณะที่ใช้ดังต่อไปนี้

1. RGB + HSYNC + VSYNC จะใช้สายในการเชื่อมต่อ 5 เส้น มาตรฐานของสัญญาณ RGB ที่ใช้คือ  $0.7 V_{p-p}$  และสัญญาณซิงก์ใช้มาตรฐาน TTL ลักษณะการใช้งานแบบนี้จะง่ายและเป็นที่ยอมรับมากในพวกโปรเจกเตอร์ในสมัยก่อน
2. RGB + Composite sync จะใช้สายในการเชื่อมต่อ 4 เส้น เพราะจะใช้สัญญาณซิงก์ 1 เส้น (เรียก CSYNC) ใช้มากในพวกโปรเจกเตอร์ในสมัยก่อน
3. RGB + Sync on green จะใช้สายในการเชื่อมต่อ 3 เส้น สัญญาณ R และ B ยังคงใช้  $0.7 V_{p-p}$  แต่สัญญาณ G จะส่งรวมไปกับสัญญาณ SYNC ซึ่งจะใช้งานที่  $-0.3 V_{p-p}$

#### อัตราการสแกนภาพ (Scan Rate)

ในมาตรฐานของระบบภาพของ USA อัตราการกวาดเส้นในแนวนอน (The horizontal scanning frequency or line rate) ของลำอิเล็กตรอนสำหรับโทรทัศน์คือ  $31.5 \text{ kHz}$  ซึ่ง และถ้าต้องการให้เคลื่อนที่ไปพร้อมกันในแนวตั้ง (The vertical scanning frequency or frame rate or field rate) มีอัตราการกวาดที่  $60 \text{ Hz}$  ซึ่งก็จะทำให้เกิดเส้นกวาดในแนวนอนจากบนสุดจนถึงล่างสุดของจอทั้งหมดเท่ากับ  $31500 \div 60 = 525$  เส้น

จำนวนสัญลักษณ์ที่จะแสดงได้ใน 1 แถวนั้นจะขึ้นอยู่กับความถี่ของสัญญาณภาพที่ใช้ ส่วนจำนวนบรรทัดที่จะแสดงผลบนจอภาพนั้นจะขึ้นอยู่กับอัตราการสแกนภาพ ซึ่งสามารถเพิ่มได้ วิธีที่เห็นได้ชัดที่สุดก็คือการเพิ่มความถี่ในการกวาดในแนวนอน ซึ่งก็จะทำให้เกิดเส้นสแกนแนวนอนมากขึ้นตัวอย่างเช่น

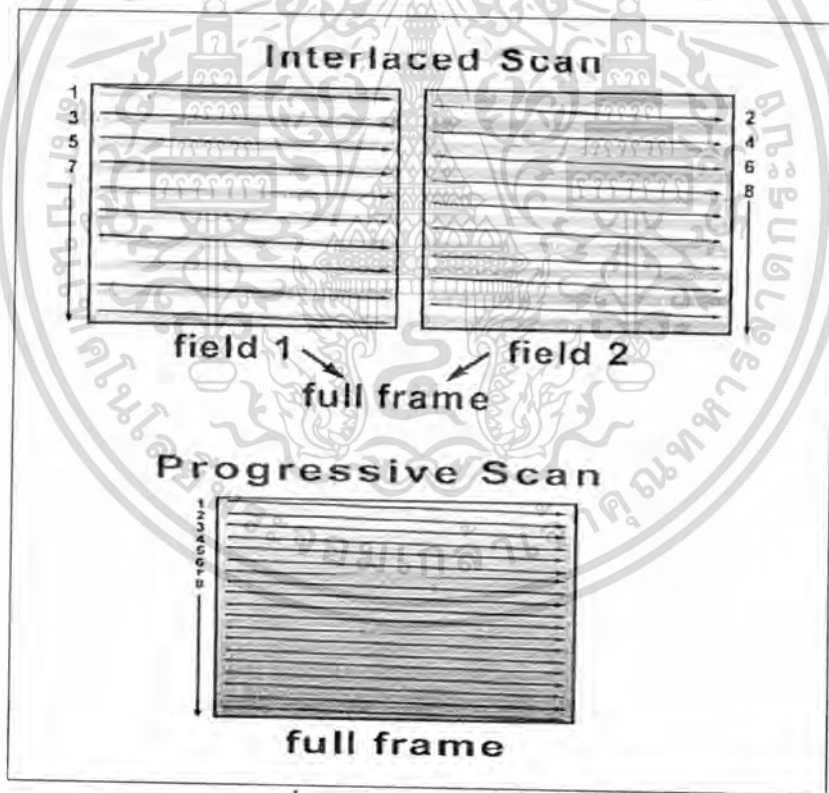
ถ้าเพิ่มความถี่ในแนวนอนจาก  $31.5 \text{ kHz}$  เป็น  $36 \text{ kHz}$  แล้วจะทำให้เกิดเส้นกวาดแนวนอนเท่ากับ  $36000 \div 60 = 600$  เส้น แต่อย่างไรก็ตามการเพิ่มความถี่การกวาดแนวนอนนั้น ก็จะทำให้ผลลัพธ์ที่ได้ไม่เป็นมาตรฐานเช่นกัน นอกจากนี้ยังต้องเพิ่มความถี่ Video เพื่อให้จำนวนสัญลักษณ์เหมาะสมกับจำนวนตัวสัญลักษณ์ต่อเส้นกวาดแนวนอน

#### อัตราการแสดงภาพใหม่ (Refresh rate)

คือ อัตราส่วนของจำนวนครั้งที่จุดภาพ (Pixel) ถูกฉายลงบนหน้าจอภาพ ซึ่งจะเริ่มจากซ้ายไปขวาและบนลงล่าง ในเวลา 1 วินาที ซึ่งเป็นค่าที่สำคัญเพราะว่ามีผลกระทบต่ออารมณ์มองเห็นภาพที่แสดงบนหน้าจอภาพ

ปัญหาหลักของการเพิ่มเส้นสแกนนั้นคือ ถ้าอัตราการ Refresh แนวนั้นต่ำกว่ากับความถี่ power line ของ CRT ก็จะทำให้ลำอิเล็กตรอนเลี้ยวเบนด้วยสนามไฟฟ้าที่มีรูปแบบไม่ถูกต้อง ผลกระทบนี้เป็นเหตุให้เกิดแสงภาพของจอกระตุกสั่น การแก้ปัญหาก็คือเราควรใช้ refresh rate นั้นให้เท่าความถี่ของ power line ดังนั้นใน USA ที่ความถี่การกวาดในแนวนอนเท่ากับ  $31.5\text{ kHz}$  ควรใช้ความถี่ในการ refresh rate มากกว่าหรือเท่ากับ  $60\text{ Hz}$  ในขณะที่ประเทศไทยก็ใช้ตั้งแต่ความถี่  $50\text{ Hz}$  ขึ้นไป เพราะถ้าอัตราการ Refresh น้อยไปก็จะทำให้ภาพนั้นสั่นกระพริบ เป็นอันตรายต่อสายตา ถ้าอัตราการ Refresh นี้ก็จะขึ้นกับสายตาของแต่ละคน และก็ยังจะขึ้นอยู่กับขนาดของจอภาพด้วยเช่นกัน โดยที่จอภาพขนาดใหญ่จะมีการสั่นของภาพได้ง่ายกว่าจอภาพที่มีขนาดเล็ก ฉะนั้นจอภาพขนาดใหญ่ควรใช้อัตราการ Refresh สูงกว่าจอภาพขนาดเล็ก

อย่างไรก็ตามยังมีวิธีการลดการกระพริบของการสแกนภาพหนึ่งแต่ละภาพ ซึ่งจะนิยมใช้วิธีการสแกนไขว้กันหรือ Interlace scanning ตามรูปที่ 2.4 โดยที่ภาพหนึ่งเฟรมจะประกอบด้วยภาพหนึ่ง 2 ฟิลด์ โดยที่เริ่มต้นด้วยการสแกนภาพหนึ่งฟิลด์เส้นที่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งกลางของแนวเส้นสุดท้าย ลำอิเล็กตรอนก็จะกลับไปเริ่มสแกนใหม่แต่จะสแกนที่ฟิลด์ของเส้นคู่ หลังจากทีสแกนเสร็จทั้ง 2 ฟิลด์แล้วก็จะได้ภาพหนึ่งเฟรม



รูปที่ 2.4 การสแกนแบบไขว้กัน

การกวาดสแกนสัญญาณภาพหรือการเริ่มสแกนใหม่ จะเป็นไปตามกระบวนการของสัญญาณ Vertical Sync ซึ่งจะแสดงถึงการเริ่มแสดงภาพใหม่หรือ Frame ใหม่ ส่วนสัญญาณ Horizontal Sync จะแสดงถึงการเริ่มใหม่ในแถว สัญญาณ Vertical sync ก็จะทำให้การ reset ตำแหน่งลำอิเล็กตรอนไปที่มุมซ้ายบน (จุด (0, 0) นั่นเอง) และเริ่มทำกระบวนการเดิมไปเรื่อย ๆ ในขณะที่กลับไปตำแหน่งเริ่มต้นสัญญาณ RGB จะมีค่าเป็น "000" หรือสีดำ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่เอกสารนี้เป็นการทำ



## ถ้ามีจอคอมพิวเตอร์ พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

นั่นเอง แต่ถ้าสัญญาณ Sync ทั้ง 2 ค่านั้นไม่สอดคล้องกันแล้ว ส่วนใหญ่ในปัจจุบันจอภาพจะมี LED แสดงสถานะ ถ้า LED มีสีเขียวแสดงว่ามีการตรวจพบสัญญาณ Sync และถ้า LED เป็นสีเหลืองแสดงว่าตรวจไม่พบสัญญาณ Sync ในการ์ด VGA จะมีหน่วยความจำซึ่งทำหน้าที่เก็บค่าของสีทุก pixel และเวลาการแสดงผล หน่วยความจำนี้ เมื่อถูกอ่านออกไปแล้วจะทำหน้าที่เหมือนเป็นตัวผลิตสัญญาณ RGB แล้วส่งให้ลำโวลีกรอนอิงกวาดไปบนฉากร

### องค์ประกอบภาพ

หากเราตัดภาพจากหนังสือพิมพ์สักภาพหนึ่ง แล้วขยายขึ้นด้วยกล้องหรือแว่นขยาย จะพบว่าภาพมีองค์ประกอบมาจากจุดสีด้ามากมายมาเรียงกันประกอบขึ้นเป็นภาพ จุดเหล่านี้เองที่เรียกว่าองค์ประกอบภาพหรือเรียกว่า Picture element หรือเรียกว่า พิกเซล (pixel)

ในการทำงานเดียวกัน ภาพที่ปรากฏบนทางจอภาพก็ใช้หลักการเดียวกัน ภาพที่เกิดขึ้นบนจอภาพประกอบด้วยเส้นแนวนอนเส้นเล็ก ๆ เป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิทและสว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสง ความแตกต่างกับการเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองเราก็ดจัดว่าเป็นองค์ประกอบภาพ

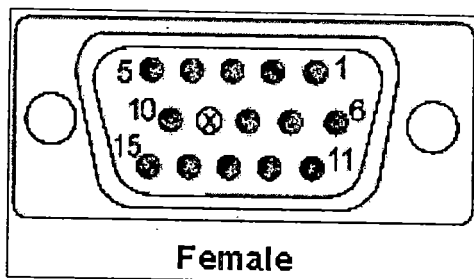
มาตรฐานระบบสแกน 525 เส้น แต่ละเส้นจะแบ่งได้เป็น 800 เส้น ดังนั้นจึงกล่าวได้ว่าหากจะหาองค์ประกอบภาพจะได้  $525 \times 800 = 420,000$  Pixels ใช้แบนวิทที่ 6 MHz ยิ่งภาพมีจำนวนเส้นสแกนมากเท่าไร รายละเอียดของภาพหรือความคมชัดก็จะมากขึ้นเท่านั้น แต่การออกแบบก็จะยากขึ้นด้วยเช่นกัน เนื่องจากภาพที่มีรายละเอียดมากก็ต้องการแบนวิทมากขึ้นด้วย จากที่กล่าวมาว่าองค์ประกอบของภาพมีหลาย pixel แต่ในความเป็นจริงแล้วเราไม่สามารถมองเห็นได้ครบทุก pixel เนื่องจากจะมีบางส่วนหายไประหว่างการสแกนภาพ

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

ปัจจุบันสำหรับโทรทัศน์ธรรมดาเราพบว่ามีเพิ่มเส้นภาพให้มากขึ้น และแน่นอนจำนวนพิกเซลก็มากขึ้นด้วย อย่างระบบโทรทัศน์แบบรายละเอียดสูงหรือ HDTV จะมีจำนวนเส้นสแกนเท่ากับ 725 เส้น ส่วนในโปรเจกเตอร์จะมีจำนวนเส้นสแกน 2200 เส้น

### VGA Connector (DE-15)

จะเป็น Video Port ซึ่งเป็นพอร์ตสำหรับต่อสายสัญญาณภาพกับจอคอมพิวเตอร์ ลักษณะของพอร์ตจะเป็นพอร์ตแบบตัวเมีย มีรู 15 รู ดังรูปที่ 2.5



รูปที่ 2.5 VGA Connector

ตารางแสดงตำแหน่งของ VGA Connector

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)
Pin Number	Signal Name	Notes
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

ตารางแสดงค่า ID ที่จะเลือกใช้ของบิตต่างๆ

Bit				Description
3	2	1	0	
n/a	n/c	n/c	n/c	No monitor connected
n/a	n/c	n/c	GND	Mono monitor which does not support 1024x768
n/a	n/c	GND	n/c	Color monitor which does not support 1024x768
n/a	GND	GND	n/c	Color monitor which supports 1024x768

## ROM: Read Only Memory

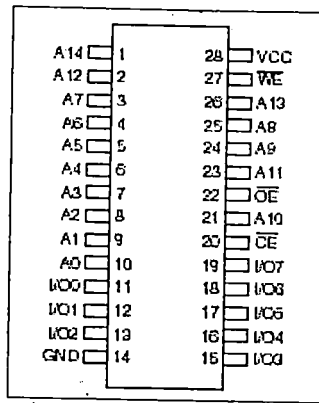
รอม คือ หน่วยความจำชนิดหนึ่งที่มีโปรแกรมหรือข้อมูลอยู่แล้ว และพร้อมที่จะนำมาต่อกับไมโครโปรเซสเซอร์ได้โดยตรง ซึ่งโปรแกรมหรือข้อมูลนี้จะไม่สูญหายไป แม้ว่าจะไม่มีการจ่ายไฟเลี้ยงให้กับระบบ ข้อมูลที่เก็บอยู่ในรอมจะสามารถอ่านค่าออกมาได้ แต่ไม่สามารถเขียนข้อมูลเข้าไปได้ เว้นแต่จะใช้วิธีการพิเศษซึ่งขึ้นกับชนิดของหน่วยความจำที่ใช้ โดยในที่นี้ใช้หน่วยความจำชนิด EEPROM ในการเก็บข้อมูลของภาพ input สำหรับการทดสอบ



รูปที่ 2.6 ลักษณะ EEPROM ที่นำมาใช้งาน

ตารางแสดง ชื่อของตัวสัญญาณต่าง ๆ ที่ใช้งานในหน่วยความจำภายนอก (EEPROM)

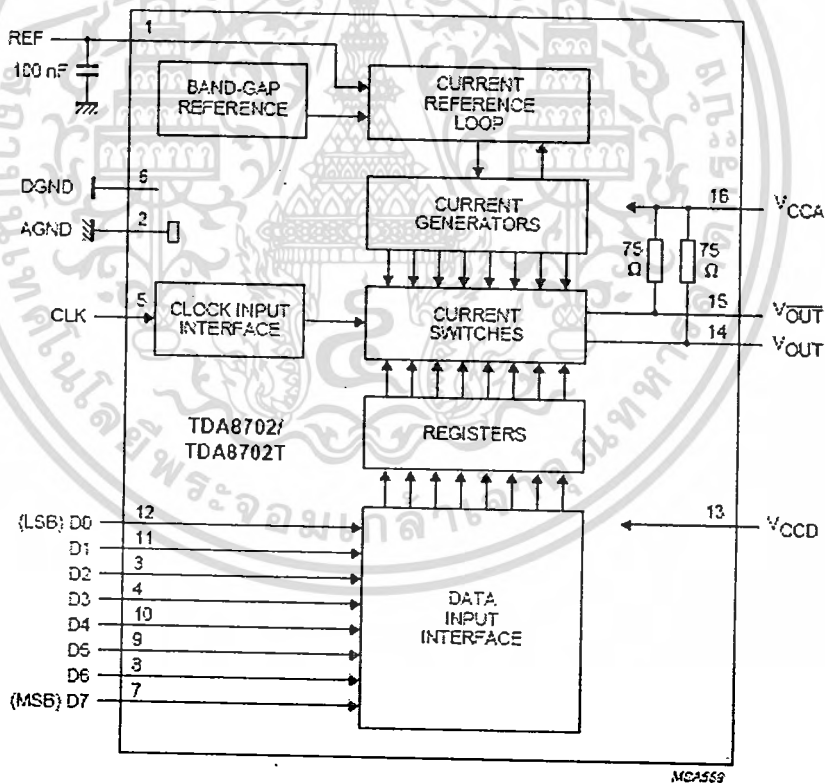
Name	Description
A0-A14	Address input
Q0-Q7	Data input
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
Vcc	Positive Power Supply
GND	Ground



รูปที่ 2.7 EEPROM เบอร์ 28C256

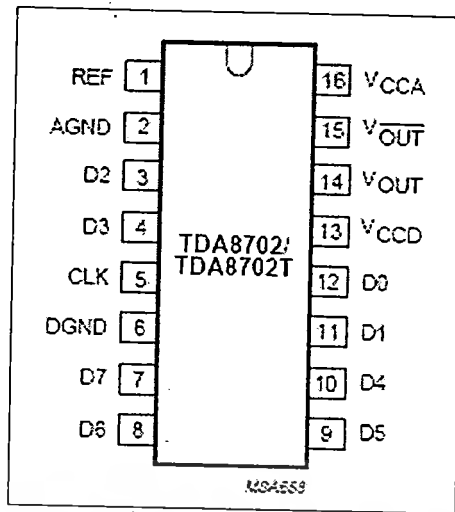
วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก

เป็นส่วนที่ทำหน้าที่แปลงสัญญาณ video ขนาด 8 บิต ที่เป็นสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก ซึ่งทำการแปลงระดับสัญญาณดิจิทัลในช่วง '00000000' ถึง '11111111' ให้ระดับแรงดันอยู่ในช่วง 0-0.8 Volt โดยใช้ไอซีเบอร์ TDA8702 โดยมีอัตราการแปลงข้อมูลสูงสุด 30 MHz



รูปที่ 2.8 บล็อกไดอะแกรมของ TDA8702





รูปที่ 2.9 ลักษณะของ DAC

ตารางแสดง การใช้งานของแต่ละ PIN ของ DAC

SYMBOL	PIN	DESCRIPTION
REF	1	voltage reference (decoupling)
AGND	2	analog ground
D2	3	data input; bit 2
SYMBOL	PIN	DESCRIPTION
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)

## การประมวลผลภาพ (Image Processing)

การประมวลผลภาพเป็นการเปลี่ยนภาพต่าง ๆ ที่ได้จากกล้องหรือการสแกน ตลอดจนข้อมูลภาพในรูปแบบดิจิทัล แล้วนำไปผ่านกระบวนการบางอย่างเพื่อให้เกิดเป็นภาพใหม่ เช่น ภาพที่มีความคมชัดสดใส หรือภาพที่เบลอ เป็นต้น

ซึ่งในโครงการนี้ เราจะอาศัยหลักการแปลงฟูริเยร์เพื่อเพิ่มหน่วยมาประยุกต์สร้างเป็นวงจรกรองฟูริเยร์แล้วนำมาทำการประมวลผลภาพโดยเราตั้งสมมติฐานไว้ว่าผลลัพธ์ที่ได้ออกมานั้นจะมีคุณสมบัติเป็นไปในลักษณะของตัวกรองความถี่สูงผ่าน (High pass filter) และอาจจะสามารถนำไปประยุกต์ใช้ในการตรวจจับขอบภาพ (edge detection) ได้อีกด้วย

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

ในโครงการนี้จะทำการอธิบายถึงทฤษฎีทวินามและความสัมพันธ์ของทฤษฎีดังกล่าวกับสามเหลี่ยมปาสคาลว่ามีความสัมพันธ์กับทฤษฎีทวินามอย่างไร ยกตัวอย่างการอธิบายทฤษฎีทวินาม เช่น ถ้าต้องการหาค่าหรือกระจายเทอม  $(x+y)^2$  สามารถทำการกระจายได้ดังสมการ

$$(x+y)^2 = x^2 + 2xy + y^2 \quad (2.1)$$

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

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

$$= x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \dots + y^n \quad (2.3)$$

$$= a_0 x^n + a_1 x^{n-1} y + a_2 x^{n-2} y^2 + \dots + a_{n-1} x y^{n-1} + a_n y^n \quad (2.4)$$

ซึ่งการกระจายเทอมในสมการ (2.2) - (2.4) บางครั้งเรียกได้ว่าเป็นการกระจายเทอมไบนอมิอัล (Binomial Expansion) โดย  $a_k$  เรียกว่าสัมประสิทธิ์ไบนอมิอัล (Binomial coefficient) ซึ่งจะมีค่าดังสมการ

$$a_k = \binom{n}{k} \quad (2.5)$$

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

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

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

สำหรับค่าของสัมประสิทธิ์ไบนอมิยัลนั้นสามารถนำมาจัดเรียงเป็นรูปฟอร์มที่เป็นสามเหลี่ยม ที่รู้จักกันในชื่อว่า สามเหลี่ยมปาสคาล ได้ดังนี้

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

(ก) สามเหลี่ยมปาสคาล

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

(ข) สามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

รูปที่ 2.10 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก

แต่ละแถว (n) ของสามเหลี่ยมปาสคาลสามารถสร้างได้ด้วยวิธีอื่นที่ไม่ใช่จากสมการที่ (2.4) โดยการบวกของ 2 องค์ประกอบที่อยู่ในแบบทแยงขึ้นไปในแถวข้างบน เช่น ในแถวที่ 4 (n=3) องค์ประกอบตัวที่ 3 (ที่ n=3, เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปเผยแพร่โดยไม่ได้รับอนุญาต) ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าคุณทหารลาดกระบัง

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

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

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

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

$$= \binom{n-1}{k} + \binom{n-1}{k-1} \quad (2.10)$$

โดยสมาชิกในสามเหลี่ยมปาสคาลในแถวที่  $n$  ใด ๆ สามารถคำนวณได้ด้วยการหาสัมประสิทธิ์ไบนอมิยัลดังในสมการที่ (2.7) ถึง (2.10) สำหรับ  $k=1, 2, 3, \dots, n$  ได้

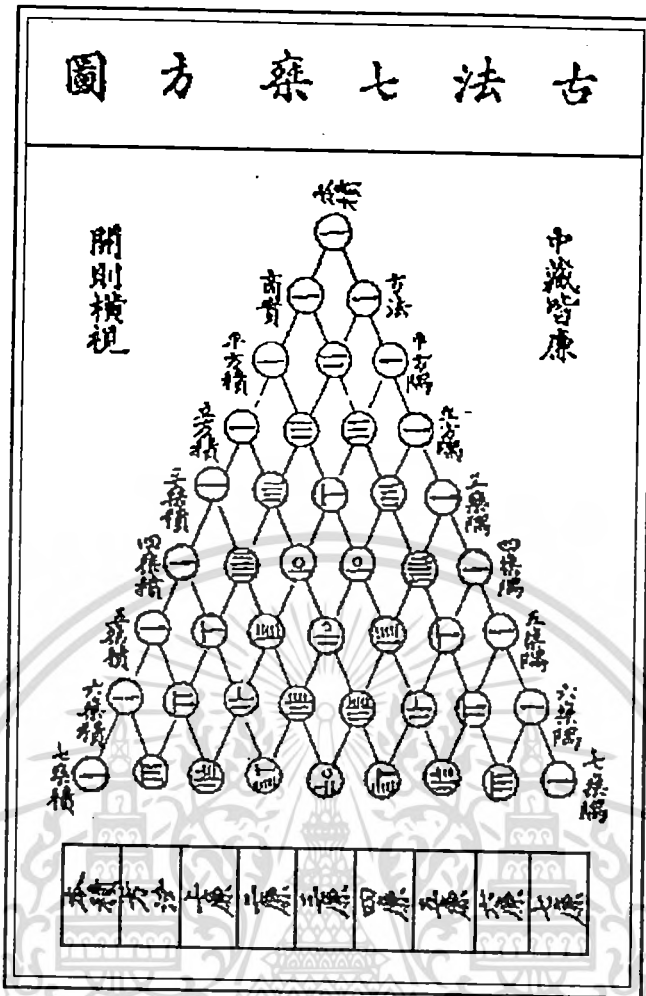
ดังนั้นเมื่อเราจะทำการกระจายเทอมของ  $(x+y)^n$  ใด ๆ เราจะสามารถหาค่าของสัมประสิทธิ์ไบนอมิยัลจากสมการที่ (2.4) ได้โดยง่ายจากสามเหลี่ยมปาสคาลนั่นเอง เช่น  $(x+y)^4$

$$(x+y)^4 = 1x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + 1y^4 \quad (2.11)$$

โดยสัมประสิทธิ์ในที่นี้มาจากสามเหลี่ยมปาสคาลในรูปที่ 2.20 ในแถวที่ 5 ( $n=4$ )

ที่มาของชื่อสามเหลี่ยมปาสคาล

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



รูปที่ 2.11 แสดงสามเหลี่ยมของ ยาง ซูย

การแปลงปาสคาลในหนึ่งมิติ (1-D Fast Pascal Transform)

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

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

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

โดยการสร้างเมตริกซ์ของการแปลงปาสคาลนั้นสามารถสร้างได้ดังนี้

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

โดย  $N$  คือ ขนาดของเมตริกซ์ของการแปลงปาสคาลซึ่งจะเป็นเมตริกซ์จัตุรัส

ส่วนฟังก์ชัน  $i^j$  จะเรียกว่าเป็นพอลลิงแฟกทอเรียลพาวเวอร์ (falling factorial powers) ซึ่งมีความหมายคือ

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

และสมการที่ (2.13) นี้เองถือว่าเป็นฟังก์ชันพื้นฐาน (Basis function) ของการแปลงดีสครีตไพสคาล ทรานสฟอร์มนี้ ซึ่งผลที่ได้จากการสร้างเมทริกซ์ของการแปลงไพสคาล ดังในสมการที่ (2.13) ซึ่งจะเหมือนกับการนำสมาชิกในสามเหลี่ยมไพสคาล ในรูปที่ 2.20 (ข) มาจัดวางเป็นเมทริกซ์และทำการสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับ

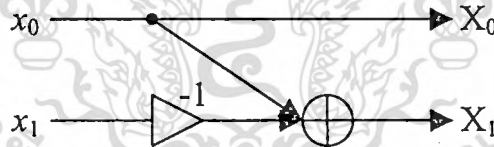
การสร้างเมทริกซ์การแปลงไพสคาลขนาดต่าง ๆ

เมื่อ  $N=2$  จะได้

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

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

สามารถสร้างบัตเตอร์ฟลายยูนิท (butterfly unit) อย่างง่ายได้ดังนี้



รูปที่ 2.12 แสดงการสร้างบัตเตอร์ฟลายยูนิทสำหรับการแปลงไพสคาลเต็มหน่วย

เมื่อ  $N=3$  จะได้

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

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

ในการสร้างบัตรเตอร์ฟลายยูนิตสำหรับ  $N=3$  นี้แน่นอนว่าต้องมีส่วนประกอบของบัตรเตอร์ฟลายยูนิต ดัง  
 ในกรณีที่  $N=2$  แต่ในทางปฏิบัติแล้ว เราใช้เมตริกซ์ที่มีองค์ประกอบในลักษณะเป็นไบนารีเมตริกซ์  $(1,0,-1)$  ดังที่  
 เห็นในกรณี  $N=2$  ซึ่งเมตริกซ์  $P$  คือ เมตริกซ์ไบนารี  $(1,0,-1)$  เราจะสามารถสร้าง บัตรเตอร์ฟลายยูนิตจากไบนารี  
 เมตริกซ์ได้โดยง่าย ทั้งนี้เราสามารถจัดปาสคาลเมตริกซ์ต่าง ๆ ให้อยู่ในรูปไบนารีเมตริกซ์  $(1,0,-1)$  ได้โดยใช้ผล  
 คูณของไบนารีเมตริกซ์ ดังสมการต่อไปนี้

$$P = \prod_{k=N-1}^1 [q_{ij}]_k \quad ; i, j = 0, 1, 2, \dots, N-1 \quad (2.15)$$

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

### วิธีการแยกองค์ประกอบของปาสคาลเมตริกซ์

#### 1. พิจารณาแนวเส้นทแยงมุมหลัก

$$P = [q_{ij}]_{N-1} [q_{ij}]_{N-2} \dots [q_{ij}]_1 \quad (2.16)$$

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

$$\text{Diag}([q_{ij}]_{N-1}) = [1 \ 1 \ 1 \ \dots \ 1 \ -1] \quad (2.17)$$

โดยที่จะมี '1' เท่ากับจำนวน  $N-1$

เช่น

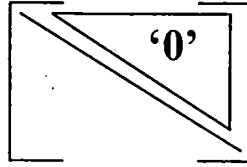
แสดงที่  $k=2$

$$\text{Diag}([q_{ij}]_2) = [1 \ 1 \ -1 \ -1 \ \dots \ -1] \quad (2.18)$$

แสดงที่  $k=1$

$$\text{Diag}([q_{ij}]_1) = [1 \ -1 \ -1 \ -1 \ \dots \ -1] \quad (2.19)$$

2. ท้องค้ประกอบเหนือเส้นทแยงมุมหลัก (Upper Triangle) หรือส่วนบนของสามเหลี่ยมมีค่าเท่ากับ '0'



รูปที่ 2.13 แสดงตำแหน่งองค์ประกอบ '0' ในส่วนของสามเหลี่ยมบนของแต่ละเมทริกซ์  $[q_{ij}]_k$

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

4. นอกเหนือจากนี้ให้มีค่าเท่ากับ '0'

ตัวอย่าง  $N = 3$  จะได้

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

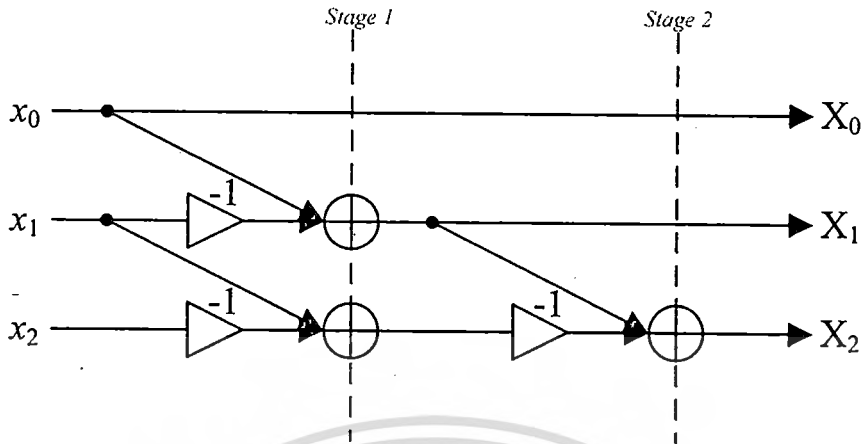
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{bmatrix} X_0 = x_0 \\ X_1 = x_0 - x_1 \\ X_2 = x_0 - 2x_1 + x_2 \end{bmatrix}$$

สามารถจัดรูปองค์ประกอบได้ดังนี้

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

Stage 2                      Stage 1

สามารถออกแบบโดยใช้บิตเตอร์หลายยูนิต ได้ดังนี้



รูปที่ 2.14 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงฟูริเยร์แบบเติมหน่วย เมื่อ  $N=3$

ตัวอย่าง  $N=4$  จะได้

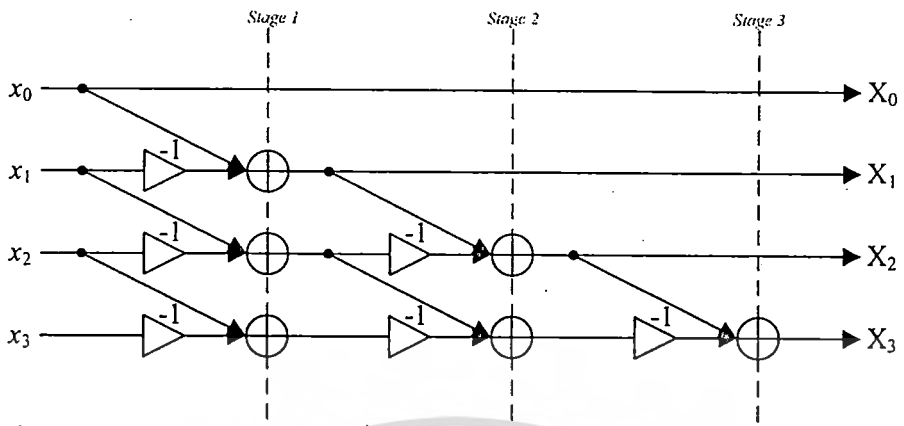
$$P = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

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

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

Stage 3                      Stage 2                      Stage 1

สามารถออกแบบโดยใช้บิตเตอร์หลายยูนิต ได้ดังนี้



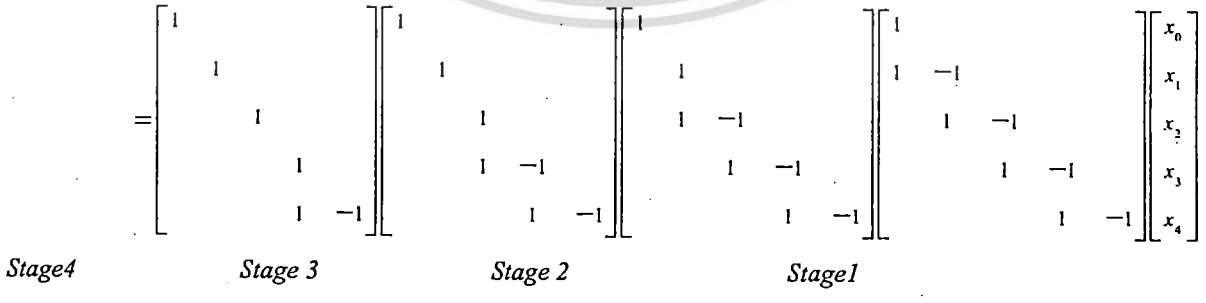
รูปที่ 2.15 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงปาสคาลเต็มหน่วย เมื่อ  $N=4$

ตัวอย่าง  $N=5$  จะได้

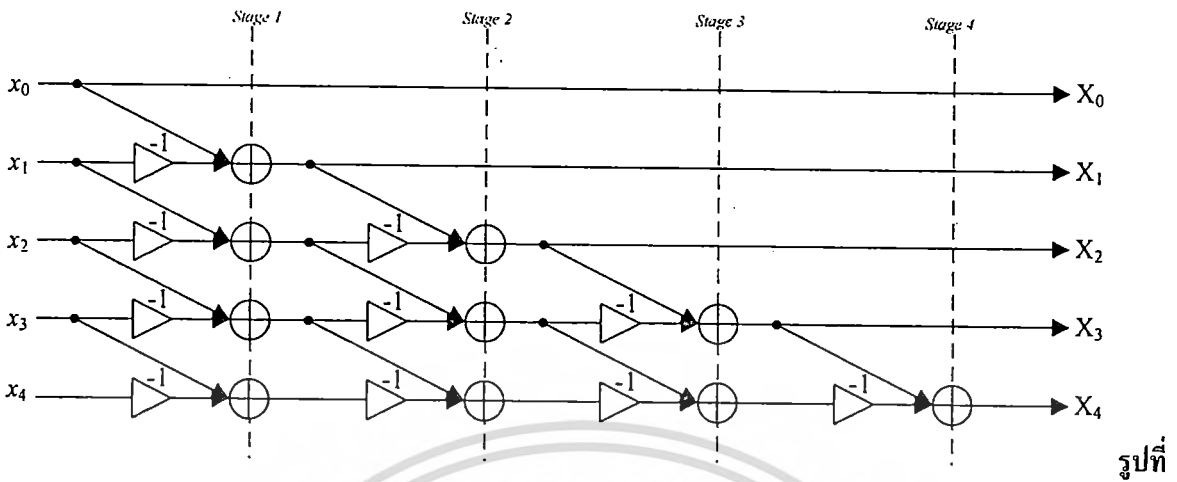
$$P = \begin{bmatrix} 1 & & & & & \\ 1 & -1 & & & & \\ 1 & -2 & 1 & & & \\ 1 & -3 & 3 & -1 & & \\ 1 & -4 & 6 & -4 & 1 & \end{bmatrix}$$

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

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



สามารถออกแบบโดยใช้บิตเตอร์หลายยูนิต ได้ดังนี้



2.16 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงฟูรีเยร์แบบเติมหน่วย เมื่อ  $N = 5$

คุณสมบัติที่ตีประการหนึ่งของการแปลงฟูรีเยร์แบบเติมหน่วยนี้คือ

$$P^{-1} = P \quad (2.20)$$

ดังนั้นทำให้การแปลงกลับดีสครีตฟูรีเยร์ (Inverse DFT) สามารถทำได้เหมือนทำการแปลงฟูรีเยร์ดีสครีตฟูรีเยร์ (Forward DFT) ทุกประการ ถ้ามองในลักษณะของการออกแบบฮาร์ดแวร์นั้นจะเห็นได้ว่าสามารถใช้วงจรเดิมที่ใช้ตอนแปลงมาสำหรับการแปลงกลับได้เช่นเดียวกัน ไม่ต้องทำการปรับเปลี่ยนวงจรแต่อย่างใด

$$x = P^{-1} X = PX \quad (2.21)$$

การแปลงฟูรีเยร์แบบเติมหน่วยในสองมิติ (2-D Discrete Pascal Transform)

สำหรับการแปลงฟูรีเยร์แบบเติมหน่วยแบบ 2 มิติ (2-D DFT) สามารถทำการแปลงได้ดังนี้

$$X = PxP^T \quad (2.22)$$

เมื่อ  $X$  = เมทริกซ์ผลลัพธ์จากการแปลง

$x$  = เมทริกซ์ของข้อมูลอินพุต

$P$  = เมทริกซ์ของการแปลงฟูรีเยร์

และกรณีแปลงกลับ (Inverse Transforms) สามารถแสดงได้ดังสมการที่ (2.23)

$$x = PXP^T \quad (2.23)$$

เพื่อความง่ายและสะดวกในการอธิบายหลักการแปลงปาสคาลเต็มหน่วยในสองมิติ ให้พิจารณาตัวอย่างเมื่อ  $N = 3$

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

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

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

สมมติให้เมตริกซ์นี้ซึ่งผ่านการแปลงปาสคาลเต็มหน่วยในหนึ่งมิติ ในแนวหลัก (column operation) มาเป็นดังนั้นก็จะได้ว่า

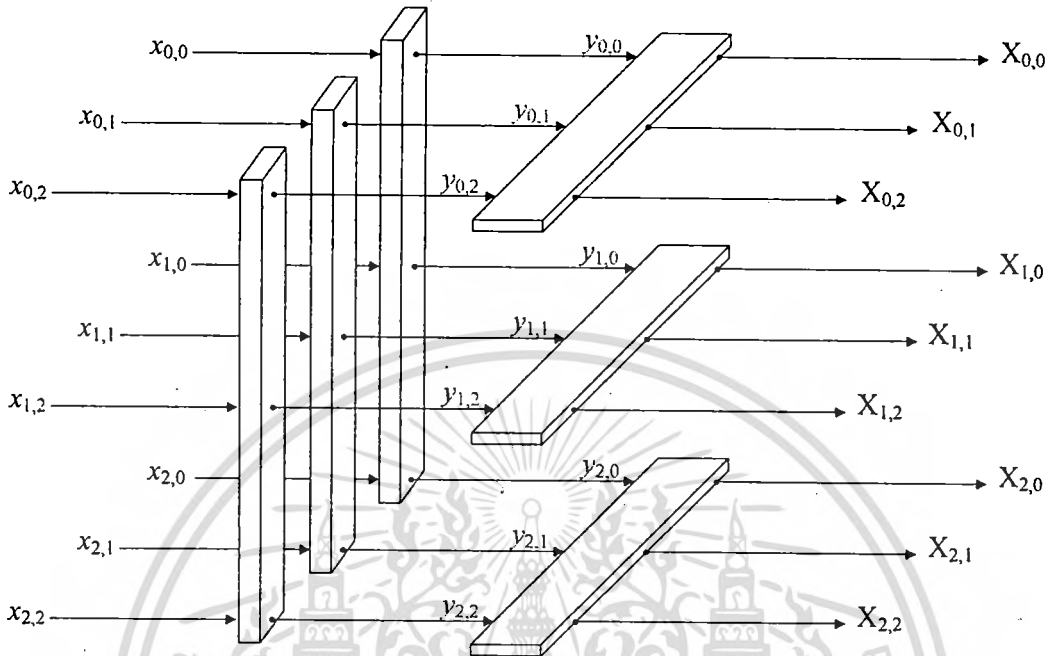
$$\therefore \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} y_{0,0} & y_{0,1} & y_{0,2} \\ y_{1,0} & y_{1,1} & y_{1,2} \\ y_{2,0} & y_{2,1} & y_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0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

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

เมตริกซ์นี้คือการดำเนินการในส่วนของแถว (row operation) สำหรับอินพุตที่ผ่านการดำเนินการในส่วนของคอลัมน์มาแล้ว ผลลัพธ์ทั้งหมดคือ การดำเนินการแปลงดีสครีตปาสคาลทรานสฟอร์มในสองมิติ ในที่สุดจะได้ว่า

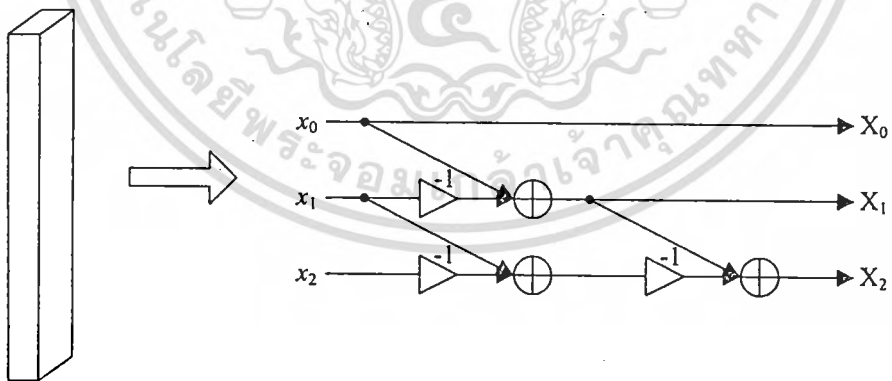
$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & x_{0,0} - x_{0,1} & x_{0,0} - 2x_{0,1} + x_{0,2} \\ [x_{0,0} - x_{1,0}] & [x_{0,0} - x_{1,0}] - [x_{0,1} - x_{1,1}] & [x_{0,0} - x_{1,0}] - 2[x_{0,1} -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

เราสามารถสรุปเป็นไดอะแกรม ได้ดังนี้



รูปที่ 2.17 แสดงกราฟการไหลของสัญญาณสำหรับการแปลงฟูรีเยร์แบบ 2 มิติ เมื่อ  $N=3$

โดยสัญญาณที่ใช้ในรูปที่ 2.17 มีความหมายดังในรูปที่ 2.18



รูปที่ 2.18 แสดงโครงสร้างแบบวงจรภายในที่ใช้ในสัญลักษณ์ตามรูปที่ 2.27 เมื่อ  $N=3$

การตรวจสอบหาคุณลักษณะทางความถี่ (คุณสมบัติการกรองสัญญาณ) ของการแปลงฟูรีเยอร์หน่วย  
การกรองความถี่สูงผ่านอาจจะเป็นลักษณะเฉพาะที่สำคัญของการแปลงฟูรีเยอร์หน่วยซึ่งเป็น  
สมมติฐานที่เราตั้งขึ้น โดยต้องทดสอบและออกแบบการทดสอบนั้น ซึ่งแนวคิดนี้มาจาก [1]  
จากที่ได้กล่าวมาแล้ว ต่อไปจะเป็นการพิสูจน์การกรองความถี่สูงผ่านของฟูรีเยอร์หน่วย

### ฟูรีเยอร์หน่วยฟังก์ชันในหนึ่งมิติ

ตัวอย่างเช่น : เมทริกซ์ขนาด 3 x 3

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

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้ โดยการเปลี่ยนตัวแปรเป็น

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

$$x(n-2) = Y_0 \quad (2.24)$$

$$x(n-2) - x(n-1) = Y_1 \quad (2.25)$$

$$x(n-2) - 2x(n-1) + x(n) = Y_2 \quad (2.26)$$

จากสมการที่ (2.25) จะได้

$$Y_0 - x(n-1) = Y_1 \quad (2.27)$$

จากสมการที่ (2.26) และ (2.27) จะได้

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

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

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

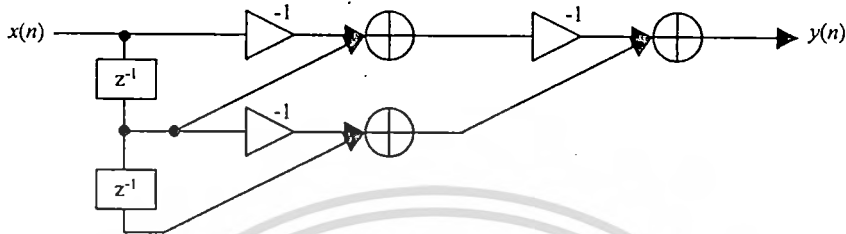
เราพิจารณา  $Y_2$  สำหรับ  $y(n)$  เพราะเราพิจารณา  $x_2$  เป็น  $x(n)$  ดังนั้น

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

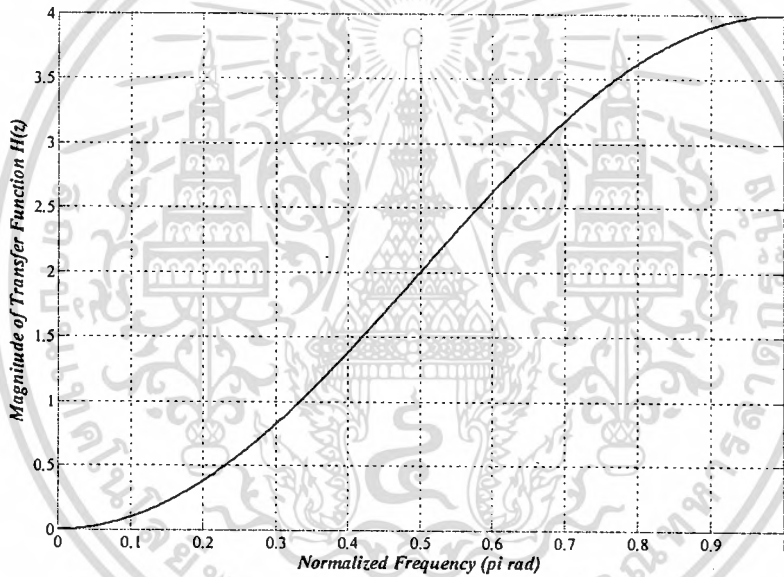
ดังนั้นจะได้ทรานสเฟอร์ฟังก์ชันคือ

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

และจะได้โครงสร้างทางฮาร์ดแวร์ของวงจรกรองปาสดาล ดังในรูปที่ 2.19 และแสดงคุณลักษณะทางความถี่ (เชิงขนาด) ได้ดังรูปที่ 2.20 ซึ่งจะเห็นได้ว่ามีคุณลักษณะเป็นวงจรกรองความถี่สูงผ่าน



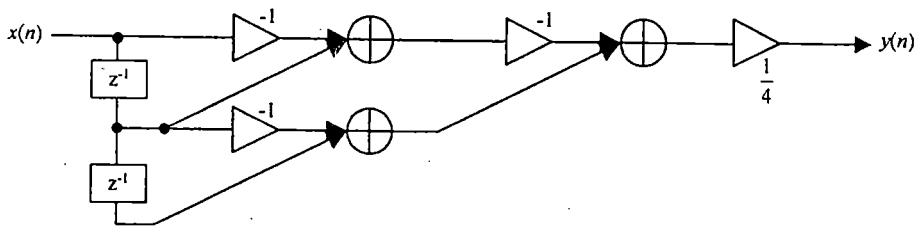
รูปที่ 2.19 โครงสร้างของวงจรกรองปาสดาลจากสมการที่ (2.28)



รูปที่ 2.20 ผลตอบสนองทางแอมพลิจูดจากสมการที่ (2.28)

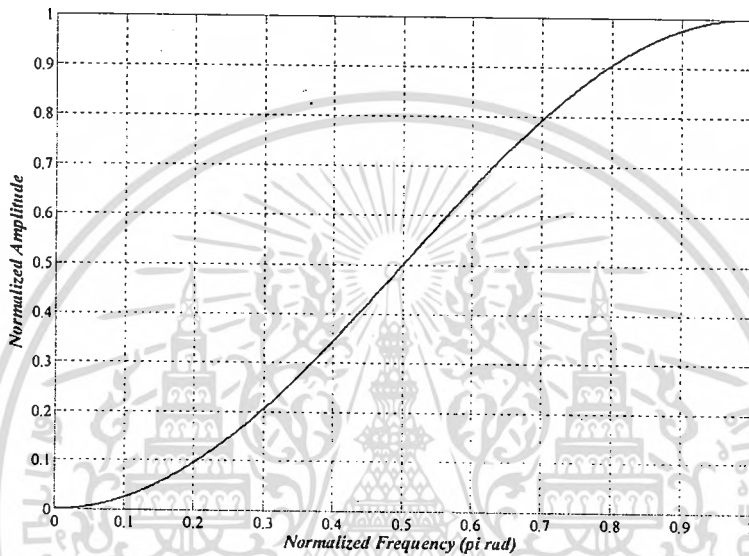
จากรูปที่ 2.20 ถ้าต้องการค่าสูงสุดของผลตอบสนองทางแอมพลิจูดเป็น 1 ต้องปรับค่า ทรานสเฟอร์ฟังก์ชันก่อนหน้าโดยปรับลงด้วยผลรวมของค่าสัมบูรณ์ของสัมประสิทธิ์ทั้งหมด จะได้ทรานสเฟอร์ฟังก์ชันใหม่คือ

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



รูปที่ 2.21 โครงสร้างของวงจรกรองปาสคาลจากสมการที่ (2.29)

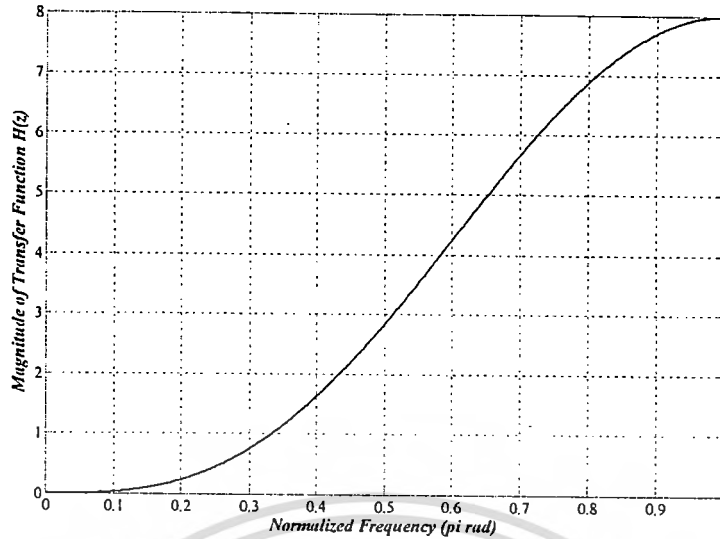
ในการปรับค่าสูงสุดของผลตอบสนองทางแอมพลิจูดนั้น ในส่วนของโครงสร้างฮาร์ดแวร์ จะทำได้โดยการเลื่อนบิตเอาต์พุตไปทางขวามือซึ่งให้ผลเสมือนกับการหารนั่นเอง จะได้ผลตอบสนองความถี่ใหม่ตามรูปที่ 2.22



รูปที่ 2.22 ผลตอบสนองทางความถี่จากสมการที่ (2.29)

กรณีที่  $N=4$ , order ของ filter คือ  $N-1 = 3^{\text{th}}$  order ของวงจรกรองปาสคาล ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน (Normalized transfer function) จะได้สมการดังนี้

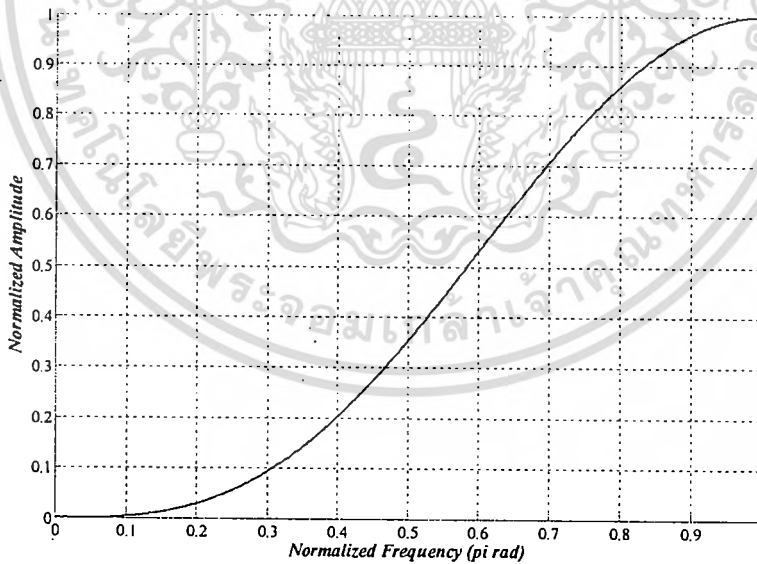
$$H(z) = 1 - 3z^{-1} + 3z^{-2} - 1z^{-3} \quad (2.30)$$



รูปที่ 2.33 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $3^{\text{th}}$  order

เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

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



รูปที่ 2.24 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $3^{\text{th}}$  order และทำการนอร์มอลไลซ์แล้ว

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

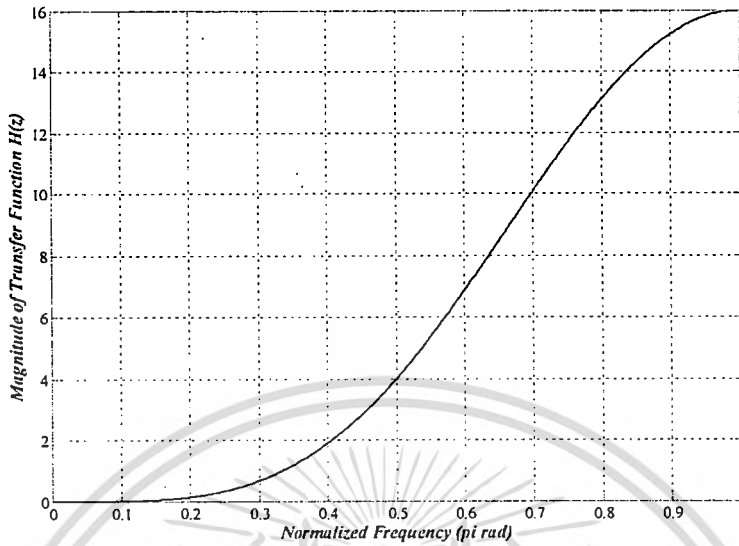
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์อื่นใด  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง King Mongkut's Institute of Technology Ladkrabang



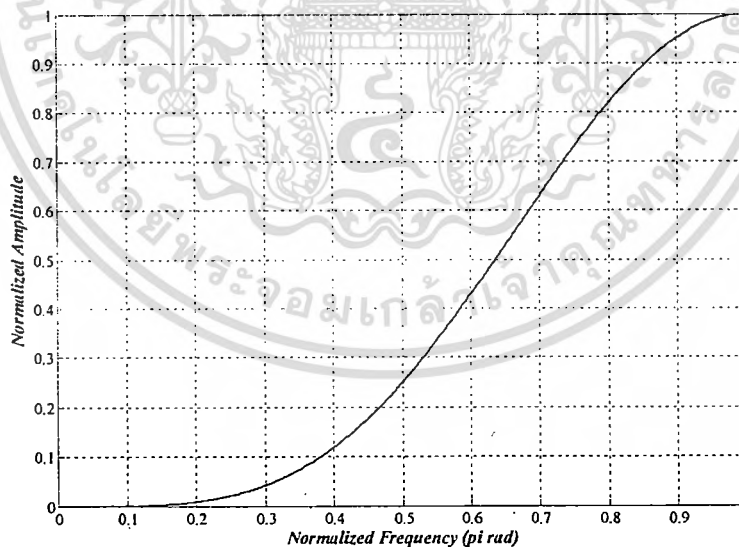
By Sorawat Chivapreecha  
 Telecommunication Eng. Dept.  
 Faculty of Engineering  
 King Mongkut's Institute of Technology Ladkrabang

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



รูปที่ 2.25 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N<sup>th</sup> order มีค่า 4<sup>th</sup> order เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชันจะได้สมการดังนี้

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

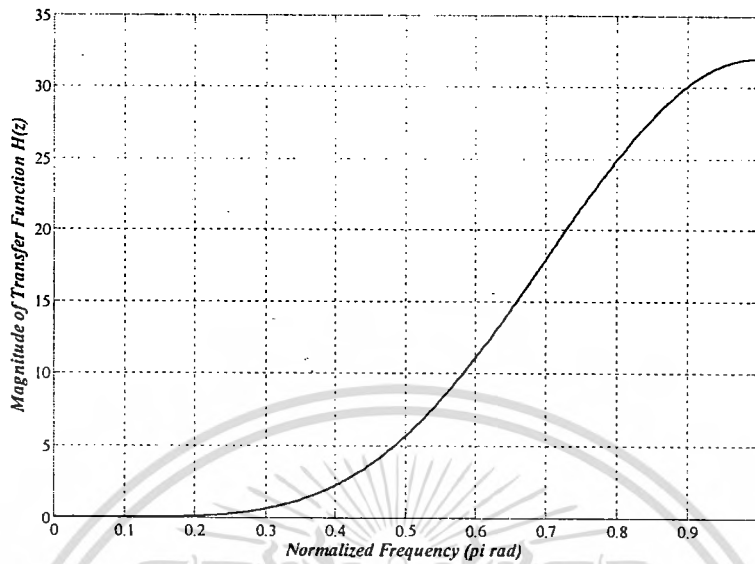


รูปที่ 2.26 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า N<sup>th</sup> order มีค่า 4<sup>th</sup> order และทำการนอร์มอลไลซ์แล้ว

กรณีนี้ N=6, order ของฟิลเตอร์ คือ N-1 = 5<sup>th</sup> order ของวงจรถองปาสคาล

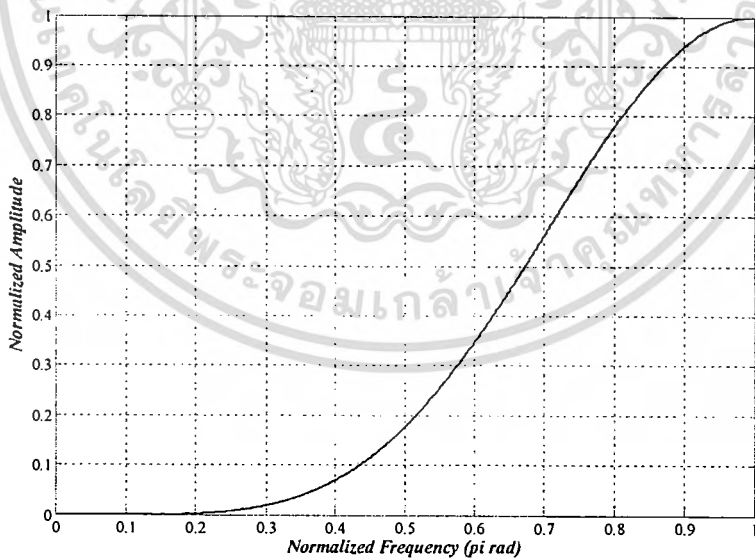
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

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



รูปที่ 2.27 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $5^{\text{th}}$  order เมื่อทำการนอร์มอลไลซ์ทรานสเฟอร์ฟังก์ชัน จะได้สมการดังนี้

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

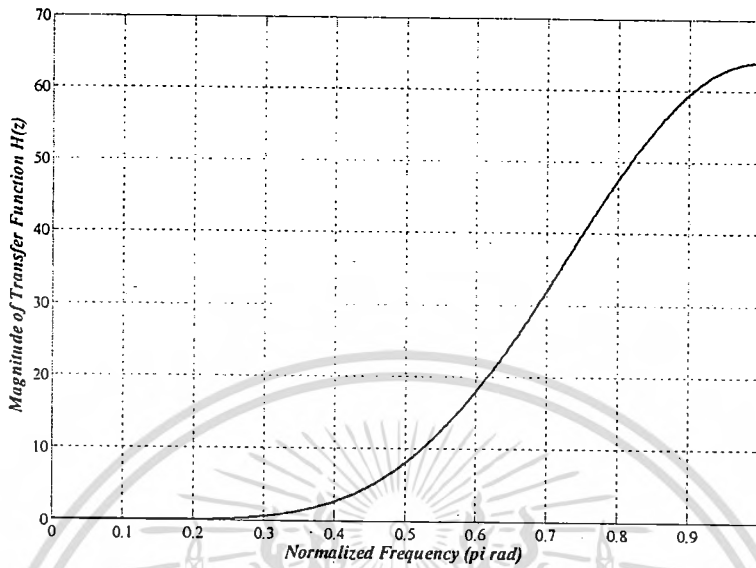


รูปที่ 2.28 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $5^{\text{th}}$  order และทำการนอร์มอลไลซ์แล้ว

กรณีนี้  $N=7$ , order ของฟิลเตอร์ คือ  $N-1 = 6^{\text{th}}$  order ของวงจรถองปาสคาล

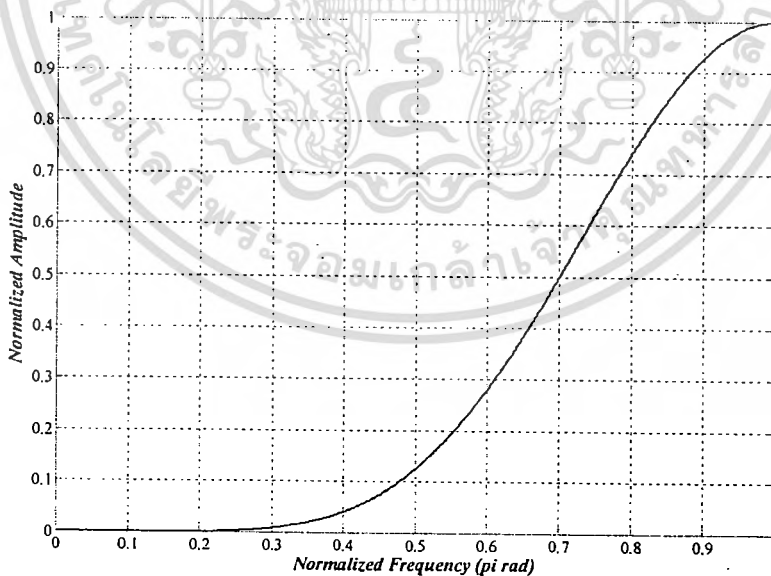
ก่อนทำการนอร์มอลไลซ์ทรานสเฟอ์ฟังก์ชัน จะได้สมการดังนี้

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



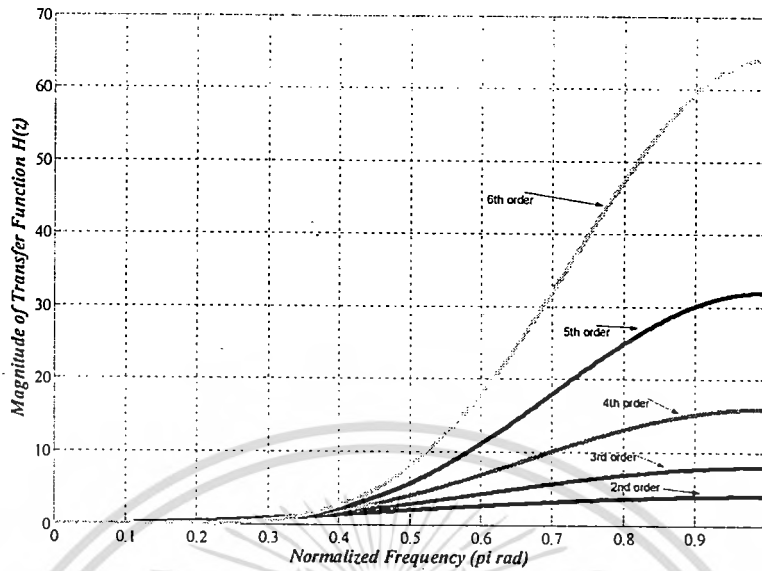
รูปที่ 2.29 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $6^{\text{th}}$  order เมื่อทำการนอร์มอลไลซ์ทรานสเฟอ์ฟังก์ชัน จะได้สมการดังนี้

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

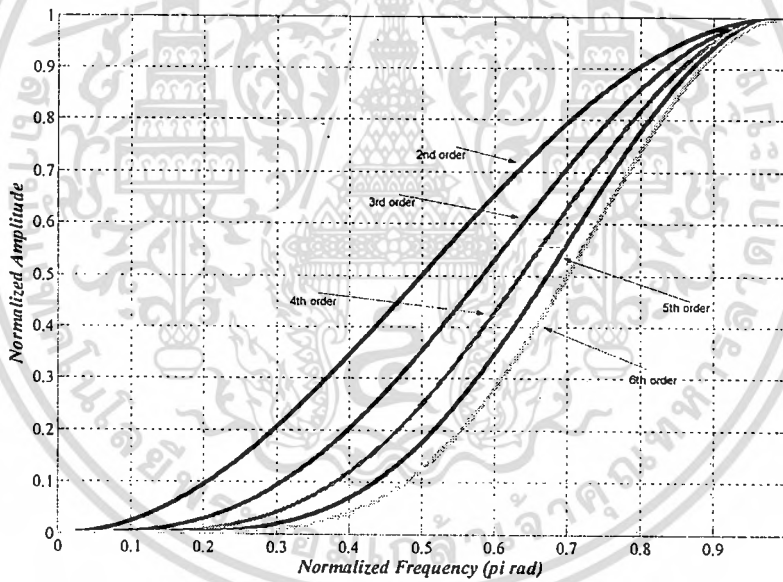


รูปที่ 2.30 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่า  $6^{\text{th}}$  order และทำการนอร์มอลไลซ์แล้ว

เมื่อทำการแสดงผลกราฟที่  $N^{\text{th}}$  order มีค่าต่าง ๆ กัน

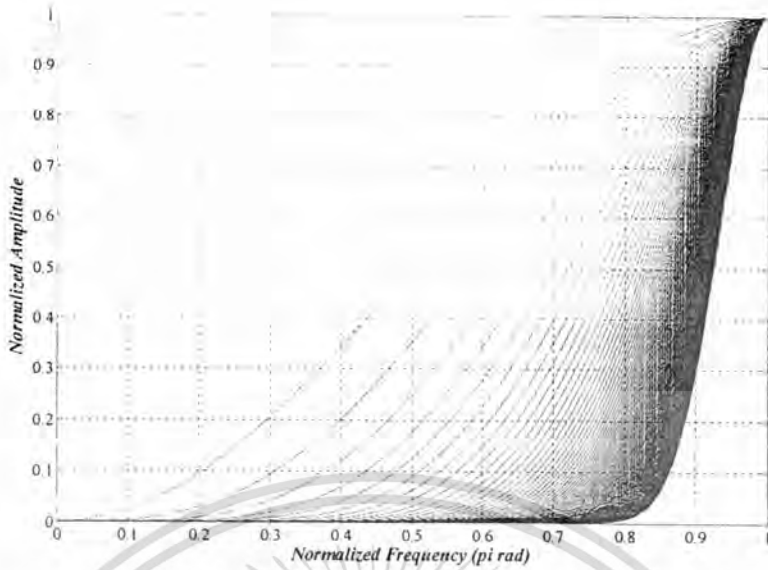


รูปที่ 2.31 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่าตั้งแต่  $2^{\text{nd}}$  order ถึง  $6^{\text{th}}$  order



รูปที่ 2.32 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อค่า  $N^{\text{th}}$  order มีค่าตั้งแต่  $2^{\text{nd}}$  order ถึง  $6^{\text{th}}$  order

เมื่อทำการนอร์มอลไลซ์แล้ว



รูปที่ 2.33 กราฟแสดงผลตัวกรองความถี่สูงผ่าน เมื่อทำการปรับค่า  $N^{\text{th}}$  order ตั้งแต่  $2^{\text{nd}}$  order ถึง  $99^{\text{th}}$  order

จากรูปที่ 2.32 และ 2.33 จะสังเกตเห็นได้ว่า เมื่ออันดับของวงจกรองสูงขึ้น ค่าของความชันที่ได้จะมีค่าที่ค่อนข้างจะเปลี่ยนแปลงน้อยซึ่งคิดจากกรณีของวงจกรองสัญญาณทั่ว ๆ ไป ที่เมื่ออันดับของวงจกรองสูงขึ้น ความชันของผลตอบสนองทางขนาดจะเพิ่มขึ้นแต่สิ่งที่เปลี่ยนแปลงเมื่ออันดับของวงจกรองที่ได้เปลี่ยนแปลงก็คือ ความกว้าง-แถบของย่านผ่านสัญญาณซึ่งคุณลักษณะนี้จะคล้ายกับวงจกรองสัญญาณที่เรียกว่าวงจกรองเกาส์เซียน (Gaussian Filter) เมื่ออันดับของวงจกรองเปลี่ยนแปลง จะมีผลทำให้ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของ

ผลตอบสนองอิมพัลส์ (Impulse response) ของวงจกรองปาสคาล 2 มิติ (2-D Pascal Filter)

ตัวอย่างเช่น : เมตริกซ์ขนาด  $3 \times 3$

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

เราสามารถปรับปรุงอินพุต และเอาต์พุตเวกเตอร์เหล่านี้ โดยการเปลี่ยนตัวแปรเป็น

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

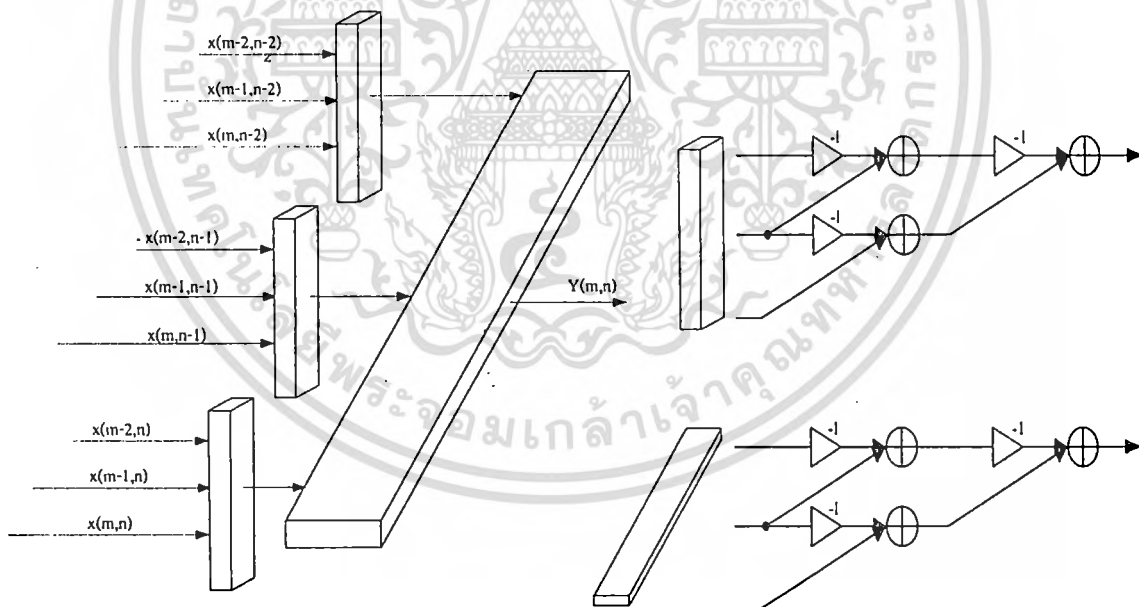
จากตัวอย่างในปาสคาลทรานสเฟอร์ฟังก์ชันในหนึ่งมิติ เมื่อเราพิจารณา  $Y_2$  สำหรับ  $y(m,n)$  จะได้ดังนี้

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

ดังนั้นจะได้ผลคอบสนองอิมพัลส์ของระบบประมวลผลสัญญาณ 2 มิติ ในหน้ากาคอนโวลูชัน (Convolution Mask) และเราจะเรียกหน้ากาคที่ได้นี้ว่าหน้ากาคอนโวลูชันปาสคาล (Pascal Convolution Mask) ดังแสดงในรูปที่ 2.34

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

รูปที่ 2.34 แสดงหน้ากาคอนโวลูชันของวงจรรองปาสคาล 2 มิติ ซึ่งมีขนาด 3x3



รูปที่ 2.35 โครงสร้างของวงจรรองปาสคาล 2 มิติ ตามสมการที่ (2.40)

## การกรองข้อมูลภาพ (Image Filtering)

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

การกรองข้อมูลภาพคือการประมวลผลภาพอย่างหนึ่งที่ทำเป็นประจำ เนื่องจากในการใช้งานจริง ภาพที่ได้มามักมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่น ๆ ประปนอยู่ด้วย การกรองข้อมูลภาพสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้น เหมาะแก่การประมวลผลในขั้นต่อไป

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

ซึ่งตัวกรองแบ่งออกได้เป็น 4 ประเภทตามลักษณะการเลือกความถี่คือ

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

โดยที่ค่าพารามิเตอร์หลักในการกำหนดคุณสมบัติของตัวกรองคือ ค่าความถี่คัทออฟ (cut-off frequency) ความถี่คัทออฟคือ ความถี่ที่ระบุจุดตัดของสัญญาณว่าจะให้ผ่าน หรือไม่ผ่าน

## 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

## ค่าเทรชโฮล

ในบางครั้งภาพระดับสีเทาที่พิจารณามีความเด่นชัดของวัตถุในภาพที่แยกตัวออกจากพื้นหลังอย่างชัดเจน ในกรณีนี้เราอาจแยกภาพของวัตถุออกจากพื้นหลังโดยการเปลี่ยนภาพเกรย์สเกลให้เป็นภาพไบนารี แต่ภาพเกรย์สเกลมีระดับความเข้มของแสงได้มากกว่าสองระดับดังนั้นเราจึงต้องมีวิธีเลือกว่าพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลพื้นหลัง และพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลภาพ วิธีที่นิยมใช้วิธีหนึ่งก็คือการกำหนดค่าความเข้มของแสงค่าหนึ่งที่ใช้แยกแยะ เช่น หากความเข้มของแสงของพิกเซลที่พิจารณา มีค่าน้อยกว่าค่านี้พิกเซลนั้นก็จะเป็นพิกเซลพื้นหลัง มิฉะนั้นพิกเซลนั้นก็ถือเป็นพิกเซลภาพ ค่าสำหรับแยกประเภทของพิกเซลที่วันนี้เราเรียกว่าค่า เทรชโฮล (threshold)

## ค่าความเข้มแสงของภาพ (Contrast)

ค่าความเข้มแสงของภาพเป็นความแตกต่างในคุณสมบัติการมองเห็นซึ่งทำให้วัตถุ (หรือการดำเนินการของมันในภาพ) ซึ่งสามารถแยกแยะความแตกต่างได้อย่างชัดเจนจากวัตถุอื่นและภาพพื้นหลัง ในการรับรู้เรื่องการมองเห็นของโลกความเป็นจริงนั้น ค่าความเข้มแสงของภาพถูกกำหนดโดยความแตกต่างในสีและความสว่างของวัตถุและวัตถุอื่นภายในอาณาเขตของการมองเห็นเดียวกัน เพราะระบบการมองเห็นของมนุษย์นั้นจะมีความไวมากกว่าใจความแตกต่างของความสว่าง ในทำนองเดียวกันเราสามารถที่จะสัมผัสถึงโลกโดยที่ไม่คำนึงถึงการเปลี่ยนแปลงอย่างมากของความสว่างในแต่ละวันหรือจากสถานที่หนึ่งไปสถานที่หนึ่ง นั่นคือในแต่ละช่วงเวลา เมื่อความเข้มแสงมีการเปลี่ยนแปลงไป เรื่อย ๆ เรานั้นแทบจะไม่มีรู้สึกได้ถึงเปลี่ยนแปลงนั้นเลย

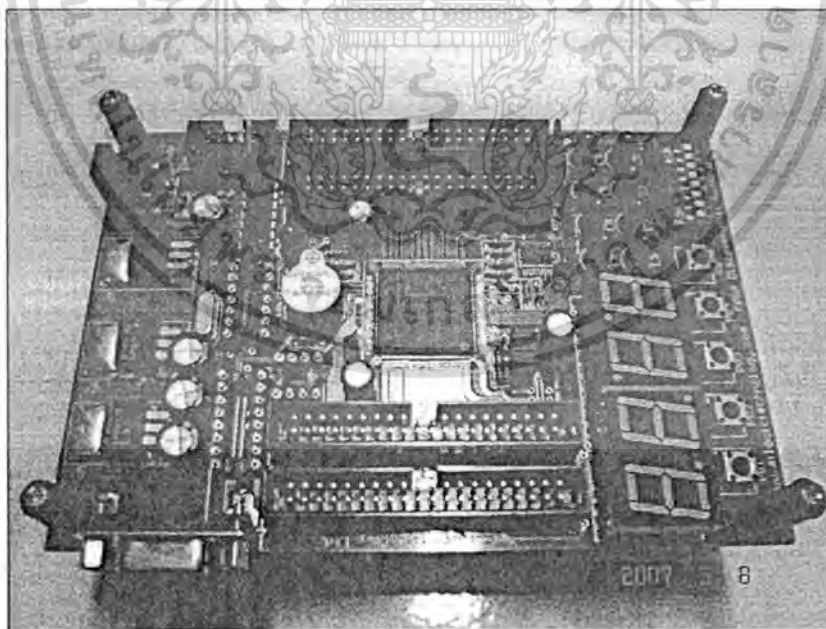
### บทที่ 3

#### Specifications ของ FPGA ที่ใช้งาน + การควบคุม VGA

##### การออกแบบวงจรเชิงเลขด้วยอุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการ โปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ออกแบบไว้ ในการทำ FPGA ซึ่งเป็นวิธีการออกแบบ IC (Integrated Circuit) แบบ Semicustom อีกวิธีหนึ่ง เมื่อเทียบกับการทำ ASICs (Application Specific Integrated Circuits) แล้วนั้นก็ยังมีทั้งข้อดีและข้อเสีย คือ การทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์ FPGA จะมีจำนวนเกต (Gate) ให้ใช้จำนวนจำกัดและการทำ FPGA ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (Code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (Download) นั้นน้อยกว่าการทำ ASIC มากและการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

การใช้งาน FPGA ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายใน หรือปรับปรุงโครงสร้างสถาปัตยกรรมภายในและยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ PPR (Partitioning Placement and Routing) สำหรับอุปกรณ์นั้น ๆ ด้วย ลักษณะของตัว FPGA และการนำไปใช้งานแสดงดังใน รูปที่ 3.1



รูปที่ 3.1 ลักษณะของตัว FPGA และการนำไปใช้งาน

สำหรับตัวอุปกรณ์ FPGA นั้นก็มีโครงสร้างพื้นฐานเทคโนโลยีที่ใช้สร้างตลอดจนเทคนิควิธีการ โปรแกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นอุปกรณ์ FPGA ของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วน ในการใช้งานนั้นอุปกรณ์ FPGA สามารถนำไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP: Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้น

#### การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

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

ในการเขียนโค้ดสิ่งที่ต้องคำนึงถึงคือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรได้และให้คุณสมบัติของวงจรตามที่กำหนด เพราะลักษณะการเขียนโค้ดจะมีผลโดยตรงกับวงจรที่ได้ เนื่องจากในการสังเคราะห์วงจรมันซอฟต์แวร์สังเคราะห์วงจร (Synthesis Tools) จะทำการสังเคราะห์ตามโค้ดที่เขียนถ้าอธิบายการทำงานของวงจรเดียวกันแต่เขียนโค้ดในลักษณะที่ต่างกันเมื่อสังเคราะห์แล้วจะได้วงจรที่ต่างกัน และจากวงจรที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันทั้งในด้านของขนาดหรือความเร็ว (Area and Time) ส่วนการเขียนโค้ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

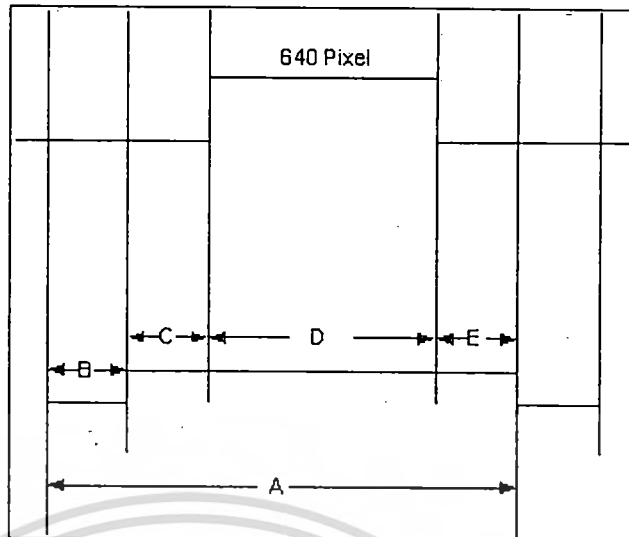
#### การสร้างสัญญาณควบคุม VGA

ในโครงการนี้ได้ใช้ไอซีเบอร์ XC3S200 โดยรับสัญญาณ clock จาก Oscillator 25.175 MHz ซึ่งใช้การเขียนภาษา VHDL

#### การสร้างสัญญาณสแกนทางแนวนอน (Horizontal Synchronization)

สัญญาณอ้างอิง

สัญญาณการสแกน  
ทางแนวนอน



รูปที่ 3.3 รูปสัญญาณการสแกนและสัญญาณอ้างอิงตำแหน่งทางแนวนอน

สัญญาณการสแกนทางแนวนอนจะเป็นตัวกำหนดการสแกนทางแนวนอนในแต่ละแถวจะมีค่าข้อมูล อยู่ในช่วงประมาณ 794 คือช่วง C+D+E นั่นเองโดยจะมีสัญญาณอ้างอิงเป็นสัญญาณ บอกถึงต้องการสแกน ในช่วง 0-639 ตามการสแกนตามมาตรฐานนั่นเอง

ตารางที่ 3.1 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวนอน และค่าการแปลงเวลาเป็นข้อมูลที่ใช้ในการเขียนโปรแกรม

Parameter	A	B	C	D	E
Time	31.77 $\mu$ s	3.77 $\mu$ s	1.89 $\mu$ s	25.17 $\mu$ s	0.94 $\mu$ s
Data	794.25	94.25	47.25	629.25	23.5

จากตารางที่ 3.1 จะแสดงช่วงค่าของ VGA Timing ของสัญญาณการสแกนทางแนวนอน ซึ่งจะใช้เวลาต่าง ๆ ตามช่วงที่กำหนดโดยประมาณ นอกจากนั้นเราจะแปลงค่าของเวลาเป็นค่าของข้อมูล เพื่อการใช้ในการเขียนโปรแกรมด้วย ภาษา VHDL จากค่าเวลาต่าง ๆ จะมีการคำนวณดังนี้

A คือเวลาทั้งหมดของการสแกนทางแนวนอน

B, G และ E คือ Guard Bands

D คือเวลาการสแกนช่วง ข้อมูล 640 Pixel/แถว

ช่วงข้อมูลจะมีการคำนวณดังนี้

ความถี่ที่ใช้งานมีค่า 25.175 MHz

การสแกนแต่ละ Pixel จะใช้เวลา

$$T_{\text{pixel}} = 1/25.175\text{MHz} = 39.721 \text{ หรือประมาณ } 40 \text{ ns}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง



By Sorawat Chivapreecha  
Telecommunication Eng., Dept.  
Faculty of Engineering  
King Mongkut's Institute of Technology Ladkrabang

เวลาทั้งหมดในการสแกนในแต่ละแถว

$$T_{row} = A = B+C+D+E = (T_{pixel} * 640) + B+C+E = 31.77 \mu s$$

ถ้าเทียบจากเวลาเป็นข้อ มูลจะมีการคำนวณดังนี้

$$A = 31.77 \mu s / 40 ns = 794.25$$

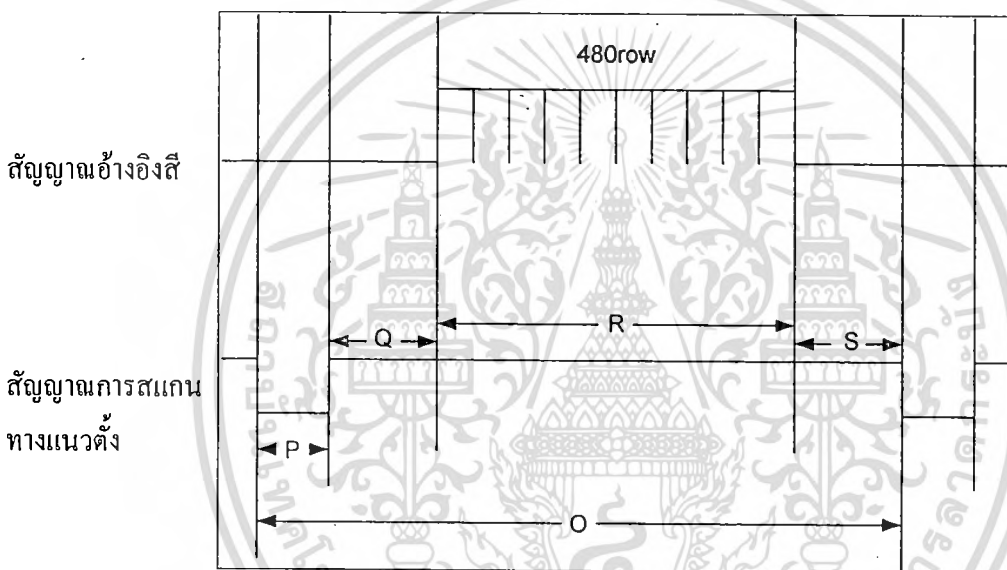
$$B = 3.77 \mu s / 40 ns = 94.25$$

$$C = 1.89 \mu s / 40 ns = 47.25$$

$$D = 25.17 \mu s / 40 ns = 629.25$$

$$E = 0.94 \mu s / 40 ns = 23.5$$

### สัญญาณการสแกนทางแนวตั้ง (Vertical synchronization)



รูปที่ 3.4 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวตั้ง

สัญญาณการสแกนทางแนวตั้งจะเป็นตัวกำหนดการสแกนทางแถวตั้ง โดยถ้ามีการสแกนทางแนวนอนครบ 523 แถวจะเกิดสัญญาณการสแกนทางแนวตั้ง 1 ลูก คือช่วง Q+R+S นั่นเอง โดยจะมีสัญญาณการอ้างอิงสีเป็นสัญญาณ บอกถึงต้องการสแกนในช่วง 0-479 ตามการสแกนมาตรฐานคือ

ตารางที่ 3.2 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวตั้ง และค่าการแปรเวลา 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 จะแสดงช่วงค่าของ VGA Timing ของสัญญาณการสแกนทางแนวตั้ง ซึ่งจะใช้เวลาตามช่วงที่กำหนดโดยประมาณ นอกจากนั้นเราจะแปลงค่าของเวลาเป็นค่าของข้อมูลเพื่อใช้ในการเขียนโปรแกรมด้วยภาษา VHDL จากค่าเวลาต่าง ๆ จะมีการคำนวณดังนี้

O คือเวลาทั้งหมดของการสแกนทางแนวตั้ง

P, Q และ S คือ Guard Bands

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

สัญญาณการสแกนทางแนวตั้ง 1 ลูก = สัญญาณการสแกนทางแนวนอน 523 แถว

เวลาที่ใช้ในการสแกนแต่ละแถวคือ  $31.77 \mu\text{s}$

$$T_{row} = 31.77 \mu\text{s}$$

เวลาทั้งหมดในการสแกนทางแนวตั้ง

$$T_{frame} = O = P+Q+R+S = (T_{row} * 480\text{rows}) + P+Q+S = 16.6 \text{ ms}$$

ถ้าเทียบจากเวลาจะได้ดังนี้

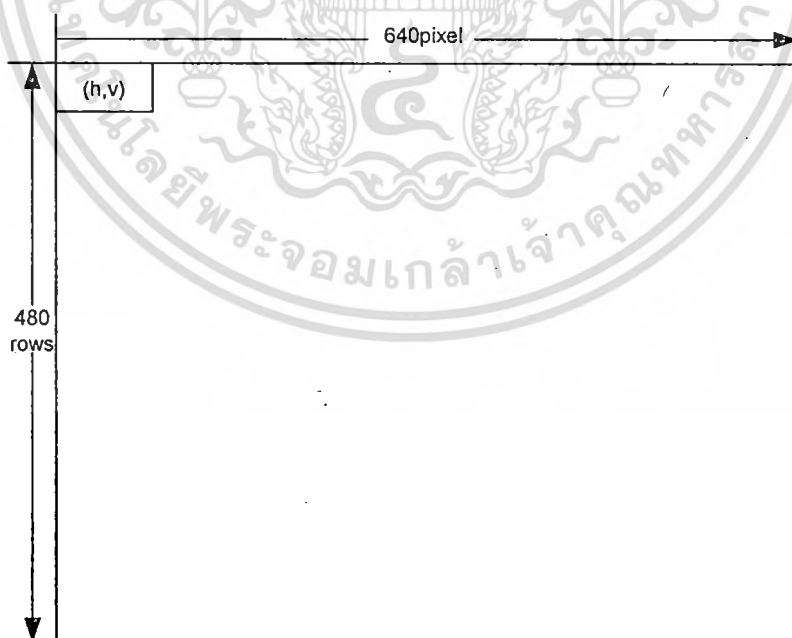
$$O = 16.6 \text{ ms} / 31.77 \mu\text{s} = 522.505$$

$$P = 0.06 \text{ ms} / 31.77 \mu\text{s} = 2.014$$

$$Q = 1.02 \text{ ms} / 31.77 \mu\text{s} = 32.105$$

$$S = 0.35 \text{ ms} / 31.77 \mu\text{s} = 11.016$$

สัญญาณอ้างอิงตำแหน่งทางแนวนอนและแนวตั้ง



รูปที่ 3.5 แสดงการอ้างอิงของจุดภาพ

จากการที่ได้กล่าวมาแล้วเราจะใช้ข้อมูลที่ได้จากการแปลงค่าจากช่วงเวลา มาสร้างสัญญาณตามที่ต้องการ ในการสร้างสัญญาณอ้างอิงตำแหน่ง จะใช้วงจรนับโดยมี Clock ตามการคำนวณของทั้งการสร้างสัญญาณทั้งทาง แนวนอนและแนวตั้งคือ

ที่ ขนาดภาพ 640\*480 Pixel และ refresh rate = 60 Hz

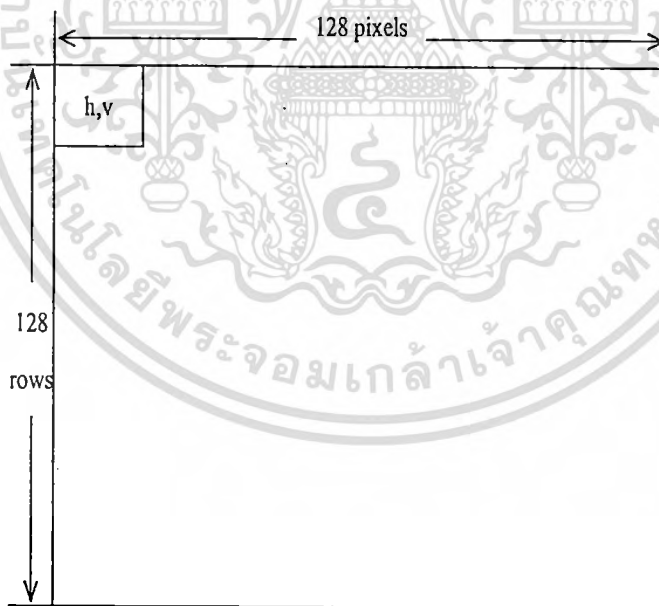
- Clock การอ้างอิงตำแหน่งทางแนวนอนใช้ความถี่ประมาณ 25.175 MHz คิดเป็นเวลาคือประมาณ 40 ns

- Clock การอ้างอิงตำแหน่งทางแนวตั้งจะใช้ความถี่ประมาณ 31.476 kHz คิดเป็นเวลาคือประมาณ 31.77

μs

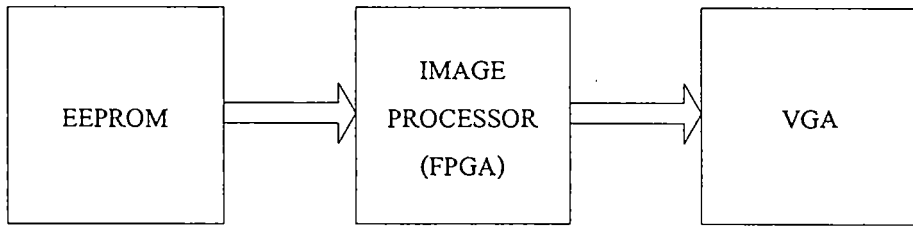
ส่วนการอ้างอิงสีนั้นจะสามารถกำหนดตำแหน่งได้ให้เกิดสีปรากฏบนหน้าจอ VGA ได้โดยดูจากรูปที่ 3.4 จะมีความกว้างทางแนวนอน 640 pixel แทนด้วย h และจะมีความยาว 480 rows แทนด้วย v ถ้าต้องการให้สีปรากฏ ที่ตำแหน่ง [h, v] เราก็จะใช้สัญญาณเป็นตัวอ้างอิงเมื่อถึงตำแหน่งที่ต้องการให้สีปรากฏออกไป ซึ่งตัวสัญญาณมี ลักษณะเป็นบิตในการอ้างอิงตำแหน่งสี ซึ่งเกิดจากการนับของ ดีฟลิปฟล็อปซึ่งก็จะใช้สัญญาณการอ้างอิง ตำแหน่งนั่นเอง แต่จะมีสัญญาณการสแกนทางแนวนอนและแนวตั้งเป็นตัวควบคุมอยู่ ดังแสดงในรูป 3.3 และ 3.4

ซึ่งดังที่ได้กล่าวมาแล้วนั้นเป็นการสร้างสัญญาณอ้างอิงทางตำแหน่งไม่ว่าจะเป็นทั้งทางแนวนอนและ แนวตั้งก็ตามจะเป็นการสร้างสัญญาณของหน้าจอวีซีเอทั้งหมด (ขนาด 640x480 pixel) แต่ในโครงการนี้เราจะ ทดลองทำการสร้างสัญญาณอ้างอิงทางตำแหน่งทางแนวนอน (Horizontal Sync) เป็น 128 pixels และสัญญาณ อ้างอิงทางตำแหน่งทางแนวตั้ง (Vertical Sync) เป็น 128 rows ก่อนในขั้นต้น ดังรูปที่ 3.6

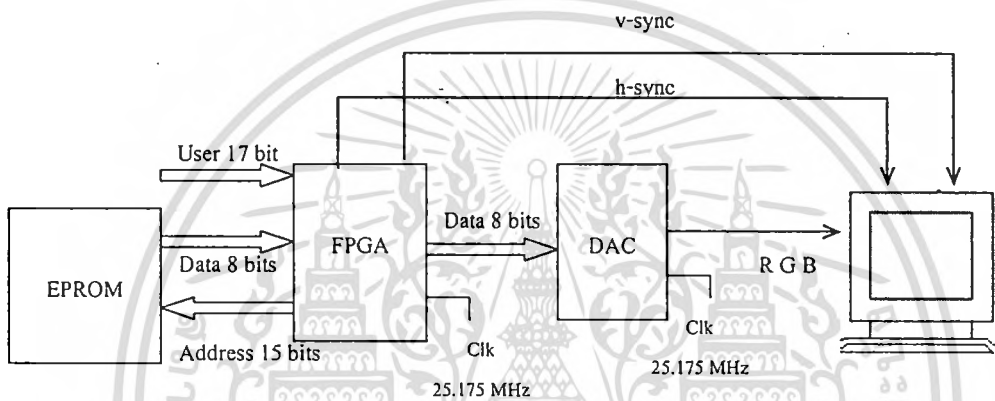


รูปที่ 3.6 แสดงการอ้างอิงของจุดภาพ (128x128)

การออกแบบวงจร



รูปที่ 3.7 วงจรรวมทั้งหมดของโครงการ



รูปที่ 3.8 วงจรรวมทั้งหมดที่จะนำข้อมูลของภาพจากหน่วยความจำภายนอกเข้าสู่เอพฟี่จีเอ เพื่อทำการประมวลผลภาพและแสดงผลทางวีซีเอ

วงจรที่สร้างจากเอพฟี่จีเอ

มีโครงสร้างหลัก ๆ ตามรูปที่ 3.8 โดยจะอาศัยหน่วยความจำภายนอก (EEPROM) เพื่อที่เก็บข้อมูลแต่ละพิกเซลของภาพ ซึ่งแต่ละพิกเซลนั้นจะมีขนาด 8 บิต ในโครงการนี้เราจะใช้ภาพขนาด 128x128 พิกเซล โดยวงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเอพฟี่จีเอที่ออกแบบด้วยภาษาวีเอชดีแอลจะทำการชี้ตำแหน่งแอดเดรสของข้อมูลให้กับหน่วยความจำภายนอกด้วยความเร็ว 25.175 MHz จากนั้นทำการรับข้อมูลภาพแต่ละพิกเซลจากหน่วยความจำภายนอกที่ส่งกลับมาให้กับเอพฟี่จีเอทำการประมวลผลสัญญาณภาพในลักษณะต่าง ๆ เช่น การปรับระดับความเข้มแสงของภาพ การขยายขนาดของภาพดั้งเดิม การจัดตำแหน่งการแสดงผลภาพบนหน้าจอวีซีเอ และการประมวลผลภาพผ่านโครงสร้างของพลาสมาฟิลเตอร์ โดยการใช้สมการทรานสเฟอร์ฟังก์ชัน ข้อมูลที่ได้จะทำการประมวลผลกับสัญญาณควบคุมการแสดงผลภาพ จากนั้นจะนำข้อมูลเข้าสู่วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกในระดับสี่เทา ทำการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อกที่มีระดับแรงดันต่าง ๆ กัน สัญญาณเอาต์พุตที่ได้จากวงจรนี้จะให้ระดับสีที่แตกต่างกัน จากนั้นนำสัญญาณที่ได้เชื่อมต่อเข้ากับขาอาร์จีบีของหน้าจอวีซีเอ นอกจากนี้แล้ววงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเอพฟี่จีเอยังสร้างสัญญาณควบคุมการสแกนจอภาพ

ทางแนวนอนและแนวตั้งอีกด้วย เมื่อนำสัญญาณควบคุมการสแกนจอภาพทางแนวนอนและแนวตั้งเชื่อมต่อกับหน้าจอดีวีดีก็จะสามารถแสดงผลภาพที่หน้าจอได้

ซึ่งส่วนประกอบของวงจรแสดงผลภาพโดยใช้บอร์ดเอฟพีจีเอส่วนหน้าจอดีวีดีจะประกอบด้วย วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง วงจรสร้างตำแหน่งของข้อมูล วงจรรับสัญญาณข้อมูลภาพจากหน่วยความจำภายนอกและวงจรประมวลผลภาพในรูปแบบต่างๆ โดยแต่ละส่วนนั้นมีรายละเอียดดังต่อไปนี้

### วงจรสร้างตำแหน่งของข้อมูล

วงจรสร้างตำแหน่งของข้อมูล ซึ่งทำหน้าที่สร้างตำแหน่งของข้อมูลให้กับหน่วยความจำภายนอก วงจรนี้สร้างจากวงจรมับ (Counter) ขนาด 14 บิต ซึ่งจะทำการนับค่าเพิ่มขึ้นทีละ 1 บิต จาก '0000 0000 0000 00' ถึง '1111 1111 1111 11' แล้ววนไปเรื่อยๆ ความเร็วในการนับนั้นจะขึ้นอยู่กับสัญญาณนาฬิกา (clk) ที่ป้อนให้กับวงจร ซึ่งในโครงงานนี้จะทำการทดลองโดยใช้สัญญาณนาฬิกาที่มีความถี่ 25.175 MHz เพื่อให้สัมพันธ์กับความถี่ที่ใช้ในการกำเนิดภาพบนหน้าจอดีวีดี

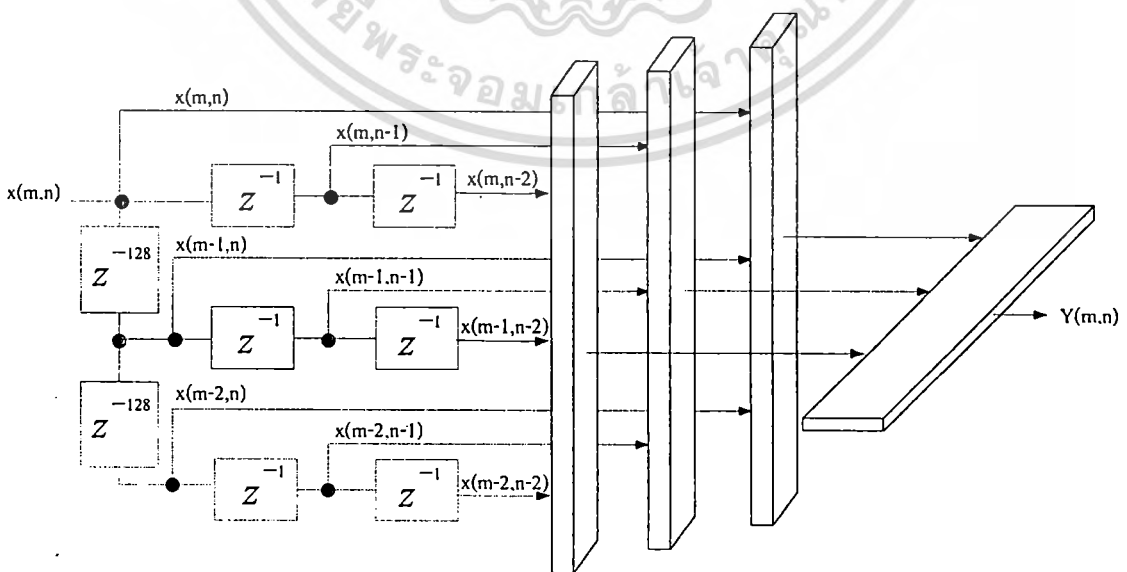
### วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง

วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง จะทำหน้าที่สร้างสัญญาณสแกนหน้าจอทางแนวนอน และสัญญาณสแกนหน้าจอทางแนวตั้งซึ่งเราสามารถสร้างวงจรดังกล่าวได้โดยการใช้วงจรมับ โดยอาศัยข้อมูลจากการคำนวณที่ผ่านมามาตามตารางที่ 3.1 และตารางที่ 3.2

### วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอ

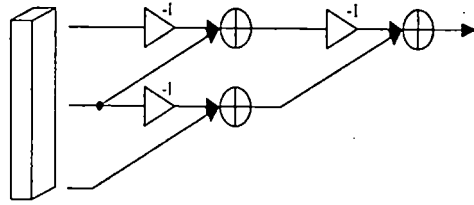
วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอมีหน้าที่รับสัญญาณข้อมูลภาพจากวงจรประมวลผลภาพโดยใช้ทฤษฎีการแปลงสามเหลี่ยมปาสคาล ทั้งนี้วงจรนี้จะทำหน้าที่ประมวลผลว่าจะให้ข้อมูลภาพนี้แสดงผลที่หน้าจอเมื่อใด เพื่อที่จะให้รูปภาพไม่เกิดการบิดเบี้ยวหรือบิดเบี้ยวไป

### วงจรประมวลผลภาพโดยใช้วงจรรองปาสคาล 2 มิติ



รูปที่ 3.9 วงจรสร้างลำดับของข้อมูลก่อนนำไปประมวลผลกับวงจรรองปาสคาล 2 มิติ

โดยสัญลักษณ์ที่ใช้ในรูปที่ 3.9 แสดงเป็นวงจรรายในดังรูปที่ 3.10

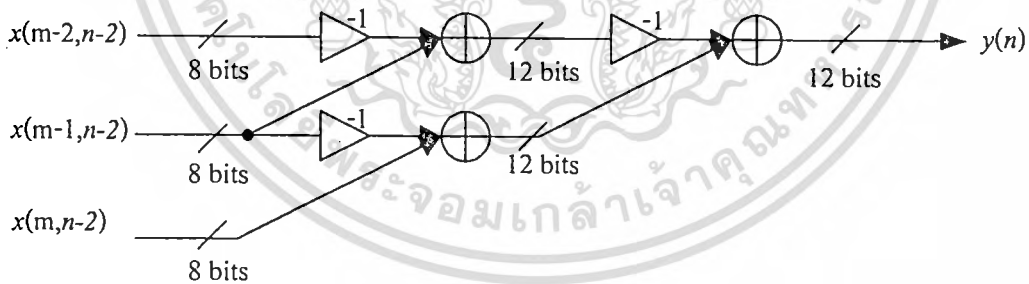


รูปที่ 3.10 แสดงบิตเตอร์หลายยูนิต เมื่อ  $N=3$

ในส่วนของเทอมนี้ นอกจากเราจะนำภาพที่มีอยู่ไปแสดงผลบนหน้าจอวีซีเอแล้ว ได้มีการทดลองนำภาพมาทำการประมวลผลโดยอาศัยหลักการของปาสคาลทรานสฟอร์มก่อนที่จะนำไปแสดงผลบนหน้าจอวีซีเออีกด้วย ซึ่งวงจรในส่วนของการประมวลผลภาพจะทำหน้าที่รับสัญญาณข้อมูลภาพจากหน่วยความจำภายนอกเข้ามา โดยที่ข้อมูลนั้นจะมีลักษณะเป็นบิตจำนวน 8 บิต ซึ่งข้อมูลที่เข้ามาจะต้องจัดลำดับสัญญาณข้อมูลเข้าให้ถูกต้องก่อนที่จะนำไปประมวลผล นั่นคือต้องสร้างลำดับข้อมูล

$X(m,n), X(m,n-1), X(m,n-2), X(m-1,n), X(m-1,n-1), X(m-1,n-2), X(m-2,n), X(m-2,n-1), X(m-2,n-2)$

ทั้งนี้กำหนดให้  $m$  เป็นตัวหน่วงข้อมูลมีค่าเท่ากับ 128 ค่า จำนวน 2 ชุด โดยอาศัยตัวเลื่อนข้อมูล (shift register) ต่ออนุกรมกัน 256 ตัว ดังรูปที่ 3.10 จากนั้นนำข้อมูลที่จัดเรียงแล้วส่งไปคูณกับแผ่นข้อมูล (Mask) ที่มีเมตริกซ์ขนาด  $3 \times 3$  เราสามารถออกแบบโครงสร้างของวงจรประมวลผลภาพได้จากสมการ ทรานสเฟอร์ฟังก์ชันตามสมการที่ (41) ในบทที่ 2 และจากรูปที่ 3.9 และ รูปที่ 3.10 สามารถออกแบบวงจรที่มีรายละเอียดดัง รูปที่ 3.11



รูปที่ 3.11 ตัวอย่างรายละเอียดของวงจรบิตเตอร์หลายยูนิต เมื่อ  $N=3$

ซึ่งในการเลื่อนจุดแต่ละจุดของภาพ เราทำการเลื่อนในลักษณะของการทำคอนโวลูชัน (Convolution) ข้อมูล  $Y(n)$  ที่ได้จะเข้าสู่วงจรประมวลผลข้อมูลร่วมกับสัญญาณแสดงผลบนหน้าจอ วีซีเอต่อไป

## วงจรรขยายขนาดของภาพและจัดตำแหน่งของภาพ

วงจรรขยายขนาดของภาพและจัดตำแหน่งของภาพ ทำหน้าที่ในการขยายขนาดของภาพ และสามารถจัดตำแหน่งของรูปภาพให้อยู่ในตำแหน่งต่าง ๆ ของหน้าจอตามความต้องการของผู้ใช้งาน

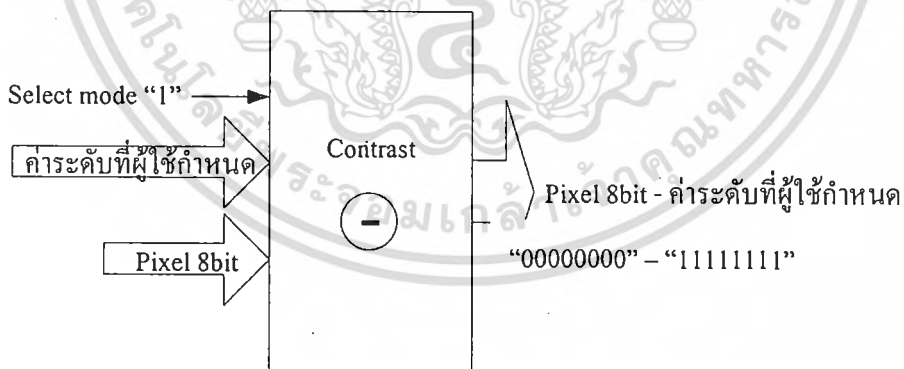
ในการจัดตำแหน่งของรูปภาพนั้นสามารถทำได้โดย ทำการหาตำแหน่งเริ่มต้นของพิกเซลแรก ที่ต้องการแสดงภาพ ให้เป็นจุดเริ่มต้นของการแสดงผลบนหน้าจอ จากนั้นนับตำแหน่งของข้อมูลภาพต่อไปเรื่อย ๆ จนครบทั้งภาพ ก็จะได้ภาพที่อยู่ในตำแหน่งที่ต้องการ

นอกจากนี้ในการขยายขนาดของภาพนั้น สามารถทำได้โดยควบคุมการชี้ตำแหน่งข้อมูลของหน่วยความจำภายนอกให้สัมพันธ์กับจำนวนพิกเซลที่แสดงบนหน้าจอของรูปภาพขนาดต่าง ๆ ที่ต้องการ เพียงเท่านี้ก็จะสามารถเลือกตำแหน่งในการแสดงภาพ และขนาดของภาพได้

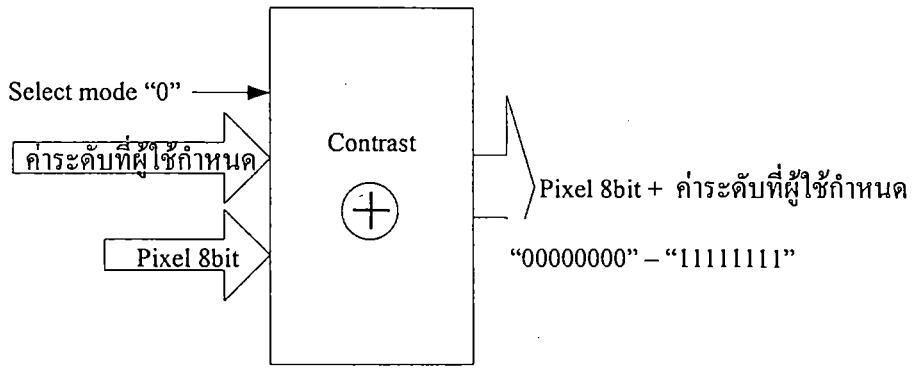
## วงจรรปรับระดับความเข้มแสง

วงจรรปรับระดับความเข้มแสง มีหน้าที่ในการปรับค่าความเข้มแสงในแต่ละพิกเซลของรูปภาพ ในการประมวลผลนั้นทำได้ โดยรับค่าระดับที่ต้องการปรับความเข้มแสงจากผู้ใช้งาน จากนั้นนำค่าที่รับมาทำการเพิ่มด้วยการบวกค่าที่รับมากับค่าความเข้มแสงของภาพนั้น หรือทำการลดค่าความเข้มแสงด้วยการลบค่าความเข้มแสงของภาพด้วยค่าที่รับมา ซึ่งผู้ใช้งานสามารถกำหนดได้ว่าต้องการเพิ่มหรือลดค่าความเข้มแสงของภาพนั้น ๆ

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



(ก)

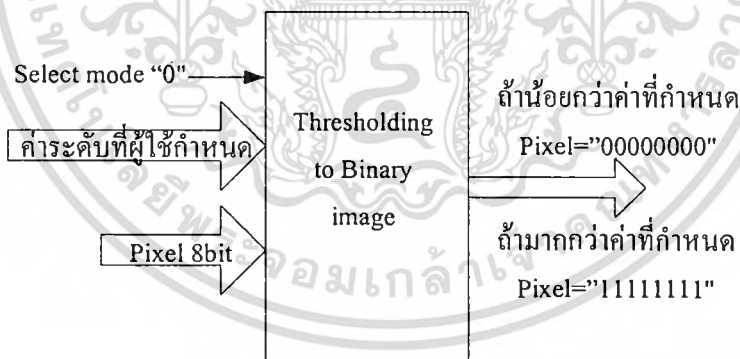


(ข)

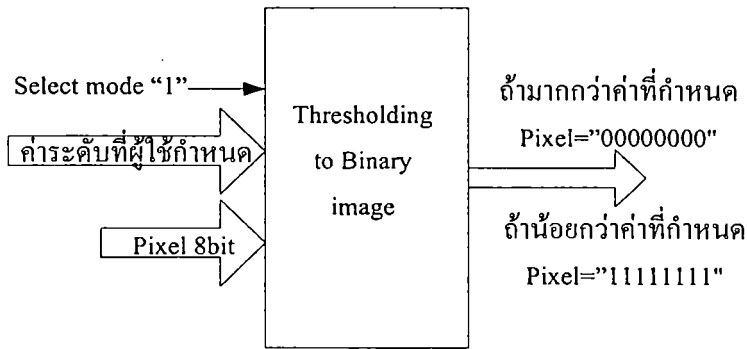
รูปที่ 3.12 วงจรปรับระดับความเข้มแสง

วงจรปรับความเข้มแสงของภาพเป็นสองระดับ

มีหน้าที่ในการปรับค่าความเข้มแสง ในแต่ละพิกเซล การประมวลผลภาพนั้นจะพิจารณาจากข้อมูลที่ใช้ป้อนให้กับวงจร โดยจะเปรียบเทียบค่าความเข้มแสงของพิกเซลนั้น ๆ เช่น ถ้าความเข้มของแสงพิกเซลนั้นมีค่าน้อยกว่า ค่าที่ผู้ใช้กำหนดพิกเซลนั้นจะแสดงค่า "00000000" หรือสีดำ แต่ถ้าค่าความเข้มของพิกเซลนั้นมีค่ามากกว่า ค่าที่ผู้ใช้กำหนดพิกเซลนั้นจะแสดงค่า "11111111" หรือสีขาว



(ค)

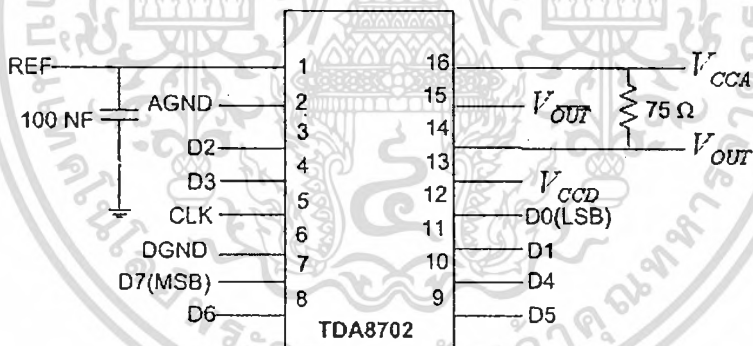


(ง)

รูปที่ 3.13 วงจรปรับความเข้มแสงของภาพเป็นสองระดับ

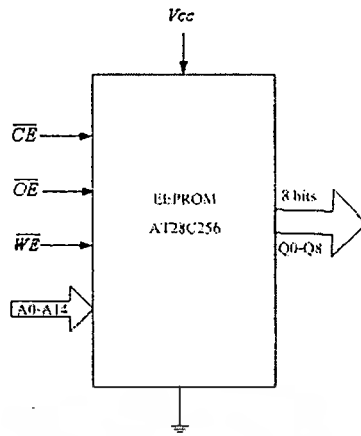
**วงจรแปลงระดับสัญญาณดิจิทัลเป็นอนาล็อก**

เป็นส่วนที่ทำหน้าที่แปลงสัญญาณภาพขนาด 8 บิต ซึ่งเป็นสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก ซึ่งทำการแปลงระดับสัญญาณดิจิทัลในช่วง '00000000' ถึง '11111111' ให้ระดับแรงดันอยู่ในช่วง 0-0.8 โวลต์ โดยใช้ไอซีเบอร์ TDA8702 MHz โดยมีอัตราการแปลงข้อมูลสูงสุด 30 MHz แต่ในโครงการนี้เราใช้คล็อกที่มีความถี่ 25.175 MHz



รูปที่ 3.14 DAC เบอร์ TDA8702

# วงจรหน่วยความจำภายนอก



รูปที่ 3.15 การทำงานของหน่วยความจำภายนอก



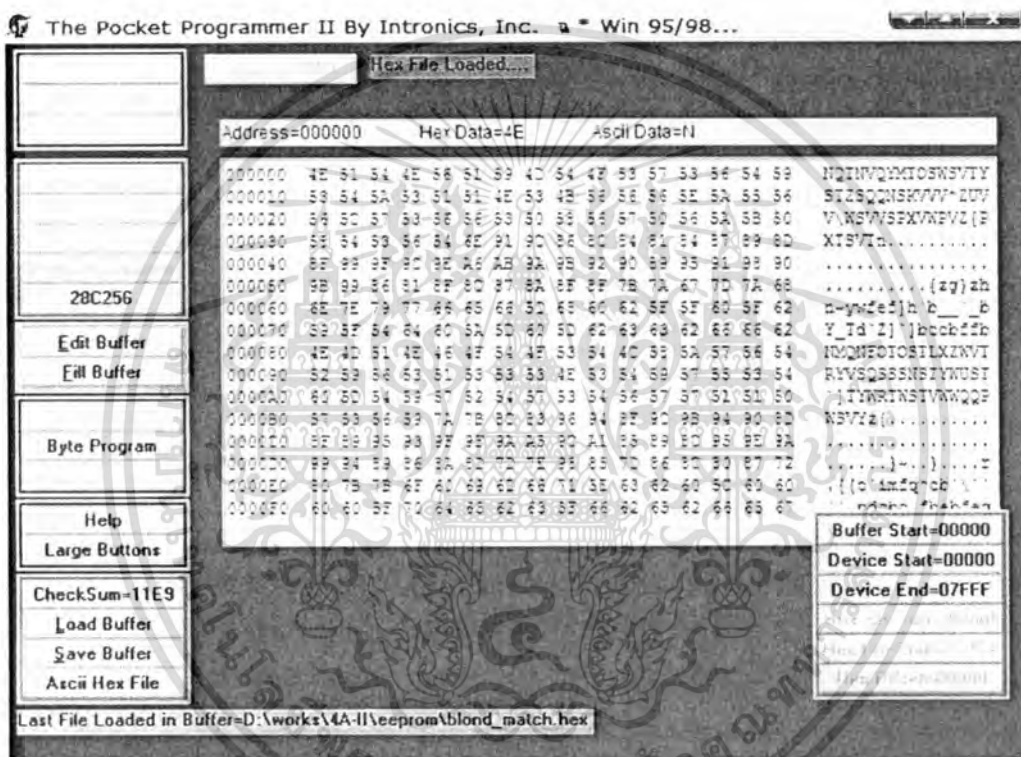
## บทที่ 4

### การทดลองและผลการทดลอง

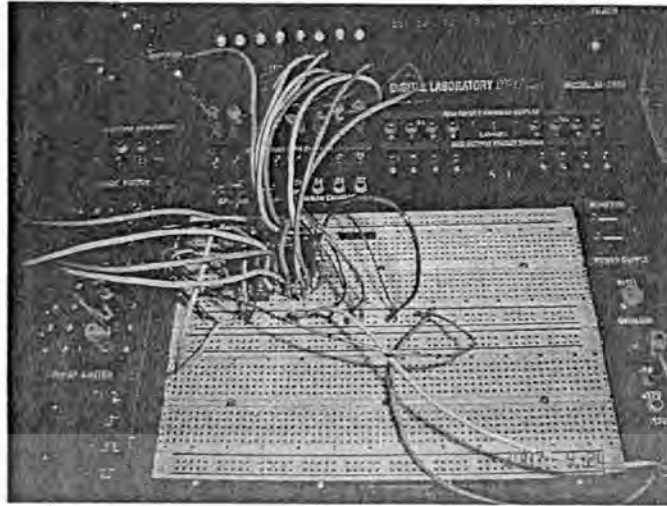
#### 4.1 ขั้นตอนการทดลอง

##### 4.1.1 การใช้งานหน่วยความจำภายนอก

ใช้หน่วยความจำภายนอก (EEPROM) เพื่อที่จะใช้เก็บข้อมูลแต่ละพิกเซลของภาพ ซึ่งในแต่ละ พิกเซลนั้นจะมีขนาด 8 บิต โดยในโครงงานนี้เราจะใช้ภาพขนาด 128x128 พิกเซล



รูปที่ 4.1 หน้าค่างการโปรแกรมของหน่วยความจำภายนอก



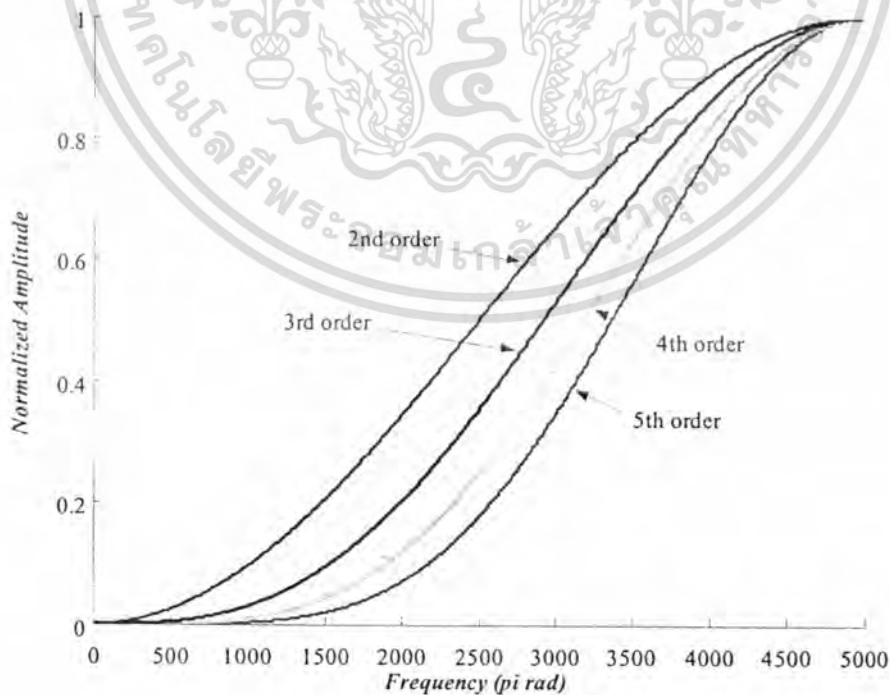
รูปที่ 4.2 รูปการทดลองเรียกใช้ข้อมูลจากหน่วยความจำภายนอก

ในรูปที่ 4.2 เป็นการทดลองเรียกใช้ข้อมูลจากหน่วยความจำภายนอกโดยป้อนค่าที่ชี้ตำแหน่งเป็น '0000000000000000' ให้กับขา A0-A14 (แอดเดรส) ซึ่งตำแหน่งนี้ทำการเก็บข้อมูลของภาพมีค่าเท่ากับ 9A ที่เป็นค่าในรูปแบบเลขฐานสิบหก ดังรูปที่ 4.1 ซึ่งผลที่ได้จากการทดลองพบว่าถูกต้อง

#### 4.1.2 ผลที่ได้จากโปรแกรม MATLAB

##### 4.1.2.1 ผลที่ได้จากการแปลงปาสคาลเต็มหน่วยและวงจรรองปาสคาลใน 1 มิติ

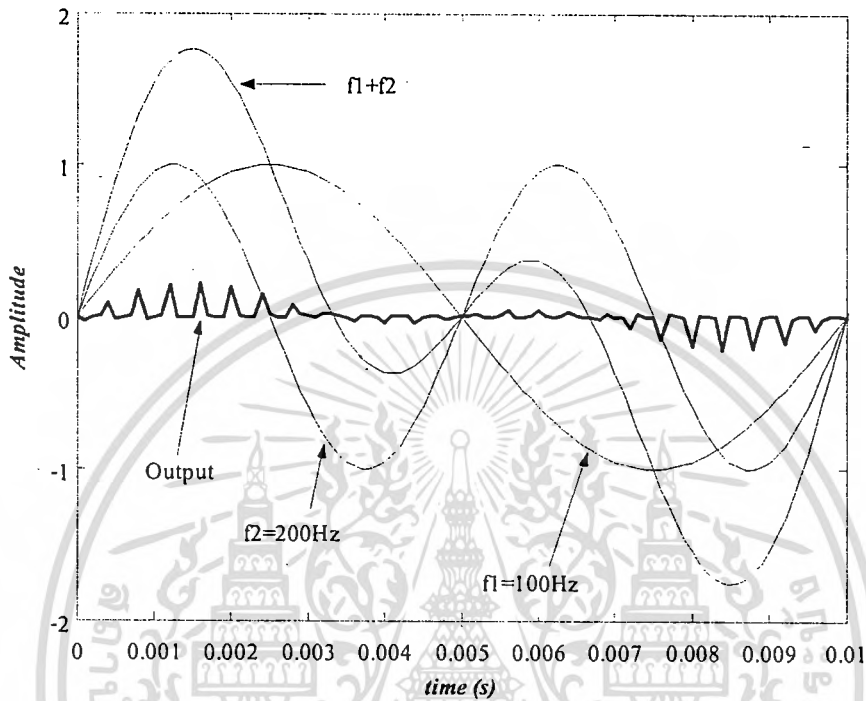
เราได้ทำการทดลองการแปลงปาสคาลแบบเต็มหน่วยและวงจรรองปาสคาลใน 1 มิติ โดยทดลองวงจรรองปาสคาลที่ 2<sup>nd</sup> order ถึง 5<sup>th</sup> order และได้พล็อตกราฟผลตอบสนองทางแอมพลิจูดและความถี่ ได้ดังรูป



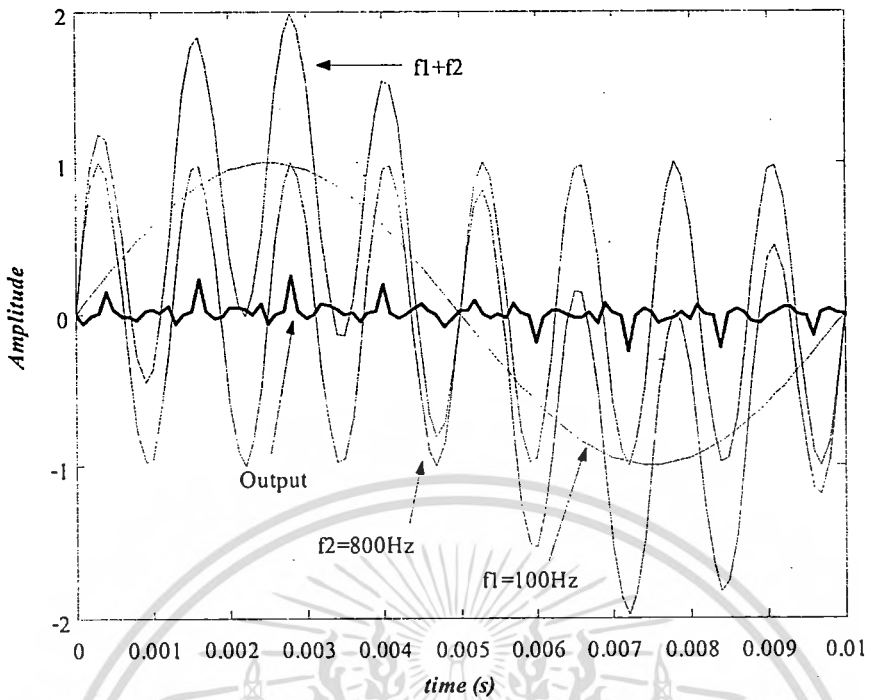
รูปที่ 4.3 แสดงผลตอบสนองทางขนาดของความเป็นตัวกรองความถี่สูงผ่าน

เมื่อทำการปรับค่า  $N^{\text{th}}$  order ตั้งแต่ 2<sup>nd</sup> order ถึง 5<sup>th</sup> order เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง

การจำลองการทำงานนี้ได้สร้างสัญญาณไซน์  $f_1$  และ  $f_2$  ที่มีความถี่ต่างกัน โดยนำสัญญาณทั้งสองมาบวกกันจะได้เป็นสัญญาณ  $f_1+f_2$  ดังรูป สัญญาณเอาต์พุตเป็นสัญญาณที่ผ่านการแปลงฟูริเยร์แล้ว ดังแสดงใน [1]

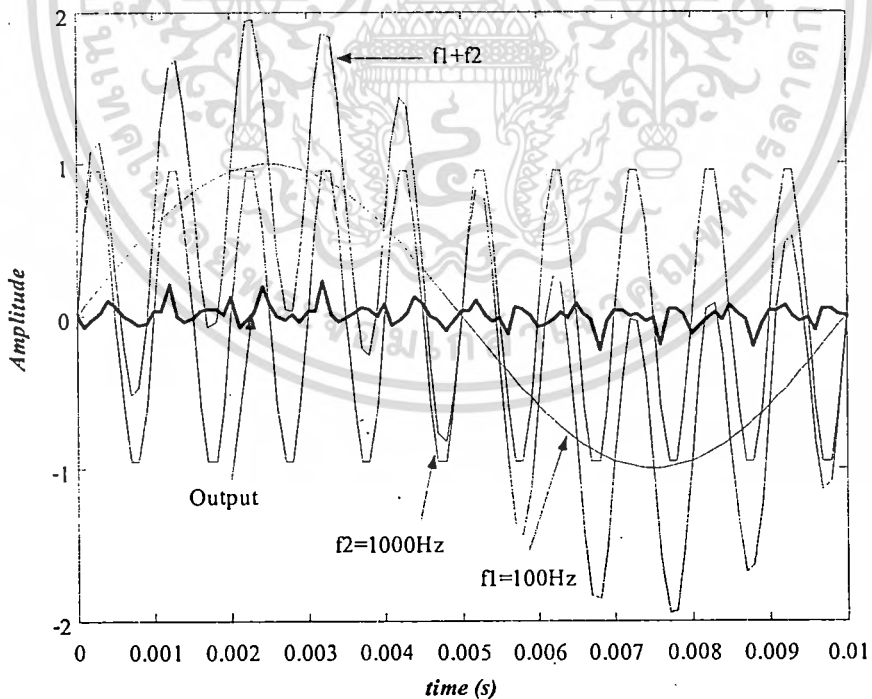


รูปที่ 4.4 ผลจากการจำลองการทำงานของการแปลงฟูริเยร์ดังใน [1]  
ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 200 \text{ Hz}$



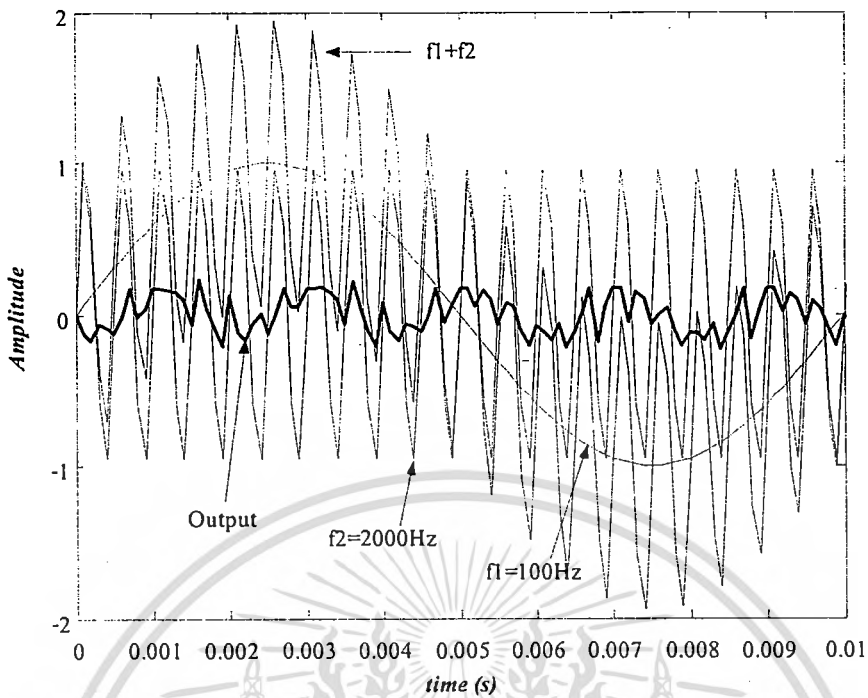
รูปที่ 4.5 ผลจากการจำลองการทำงานของเครื่องแปลงฟูริเยร์ด้วยเมทริกซ์ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 800 \text{ Hz}$

ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 800 \text{ Hz}$



รูปที่ 4.6 ผลจากการจำลองการทำงานของเครื่องแปลงฟูริเยร์ด้วยเมทริกซ์ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 1000 \text{ Hz}$

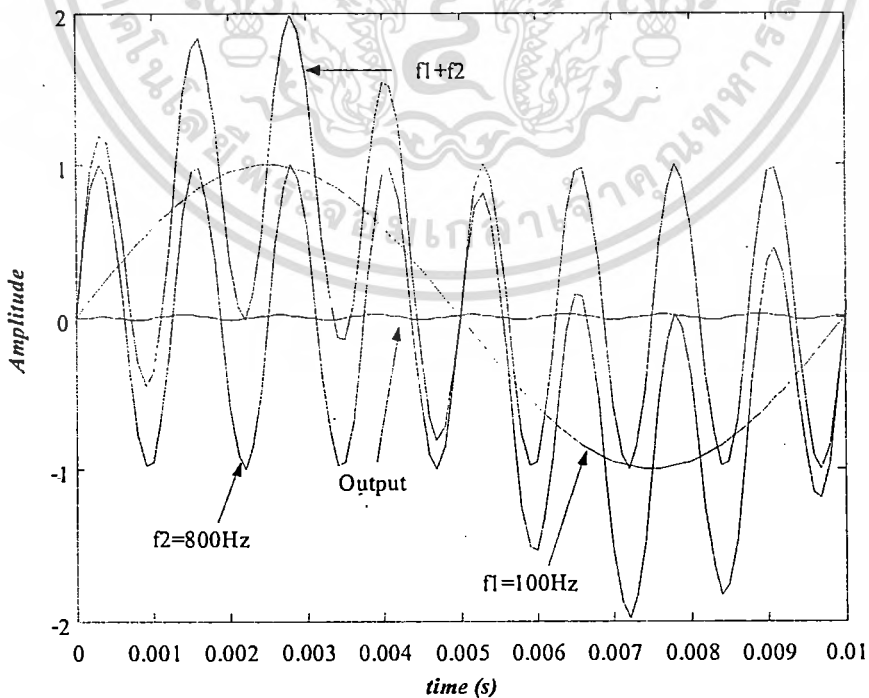
ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 1000 \text{ Hz}$



รูปที่ 4.7 ผลจากการจำลองการทำงานของ การแปลงฟูริเยร์ด้วยวิธี [1]

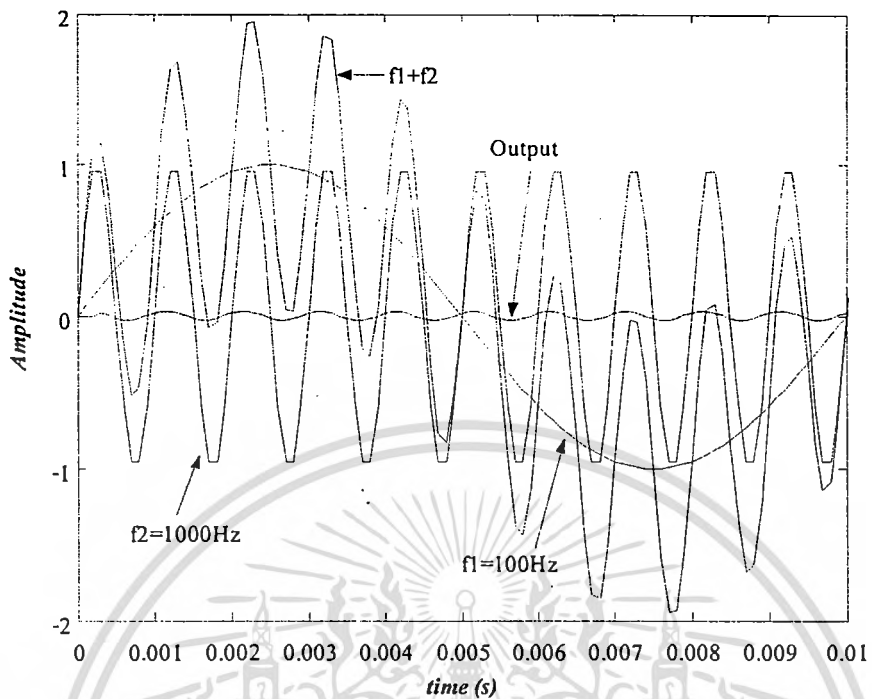
ขนาด 4x4 จาก MATLAB เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 2000 \text{ Hz}$

จะสังเกตเห็นว่า การแปลงฟูริเยร์ด้วยวิธีนี้ จะไม่สามารถเห็นถึงคุณสมบัติของการกรองความถี่สูงผ่านได้ จากนั้นได้ทดลองนำสัญญาณ  $f_1+f_2$  ผ่านวงจรกรองฟูริเยร์ จะได้ผลดังนี้



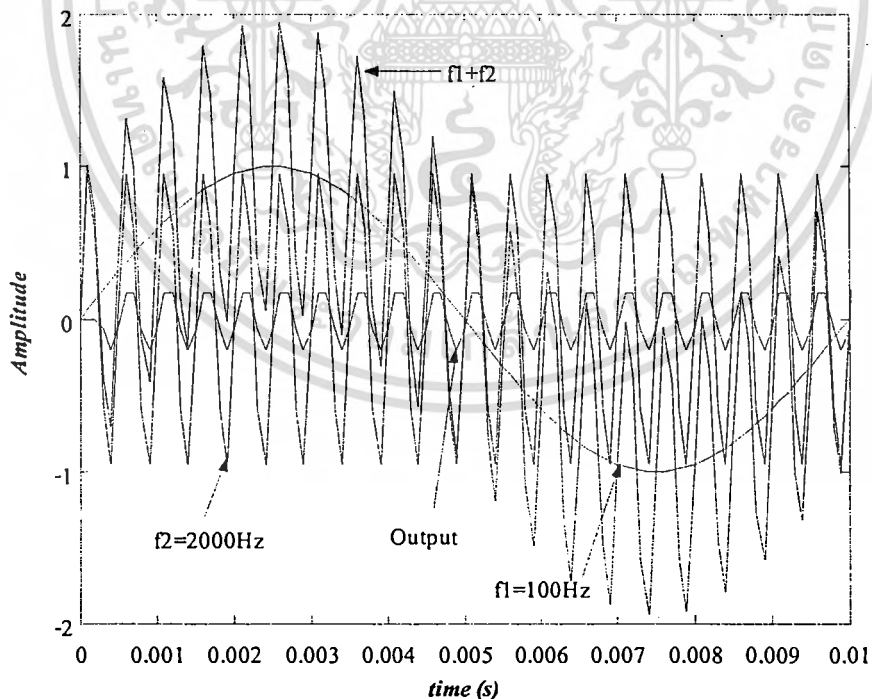
รูปที่ 4.8 ผลจากการจำลองการทำงานของ วงจรกรองฟูริเยร์ 1 มิติ อันดับที่ 3 จาก MATLAB

เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 800 \text{ Hz}$



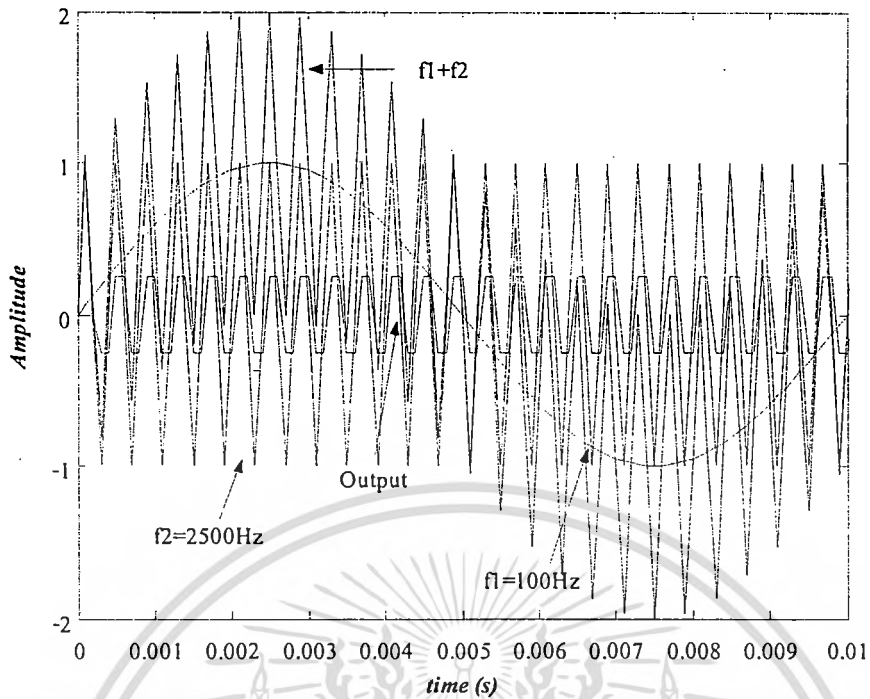
รูปที่ 4.9 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB

เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 1000 \text{ Hz}$



รูปที่ 4.10 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB

เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 2000 \text{ Hz}$



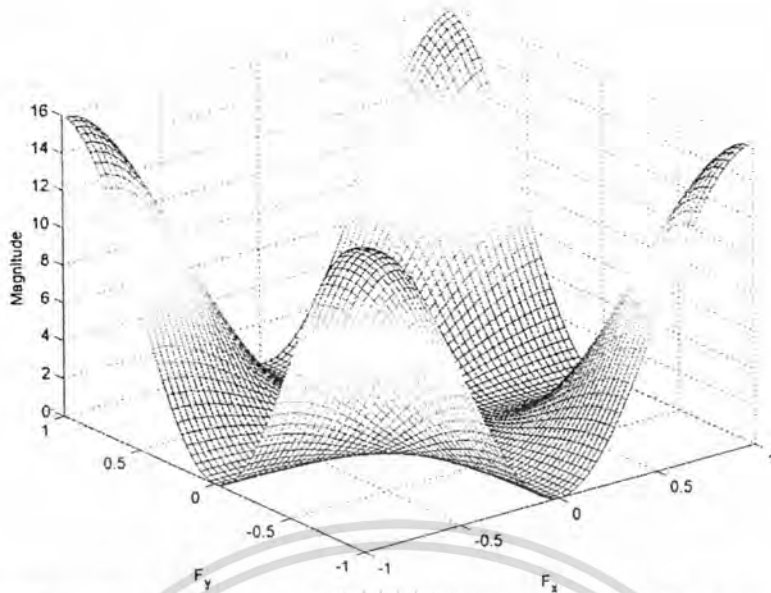
รูปที่ 4.11 ผลจากการจำลองการทำงานของวงจรกรองปาสคาล 1 มิติ อันดับที่ 3 จาก MATLAB

เมื่อ  $f_1 = 100 \text{ Hz}$  และ  $f_2 = 2500 \text{ Hz}$

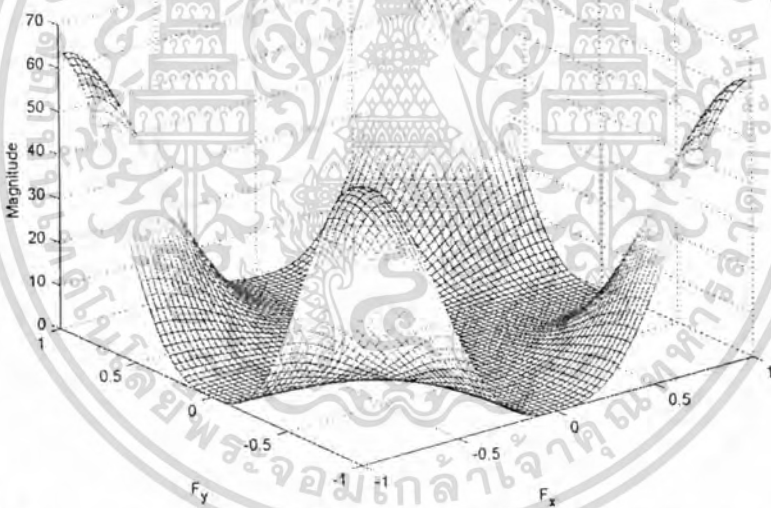
จากการทดลองพบว่าเมื่อนำสัญญาณที่มีองค์ประกอบความถี่สูงรวมกับความถี่ต่ำเมื่อผ่านปาสคาลฟิลเตอร์แล้ว สัญญาณเอาต์พุตที่ได้เป็นสัญญาณที่มีองค์ประกอบความถี่สูง นั่นก็คือสัญญาณความถี่สูงเท่านั้นที่สามารถผ่านออกมาได้ ซึ่งเป็นคุณสมบัติของตัวกรองความถี่สูงผ่าน

#### 4.1.2.2 ผลตอบสนองทางความถี่ที่ได้จากวงจรกรองปาสคาล 2 มิติ

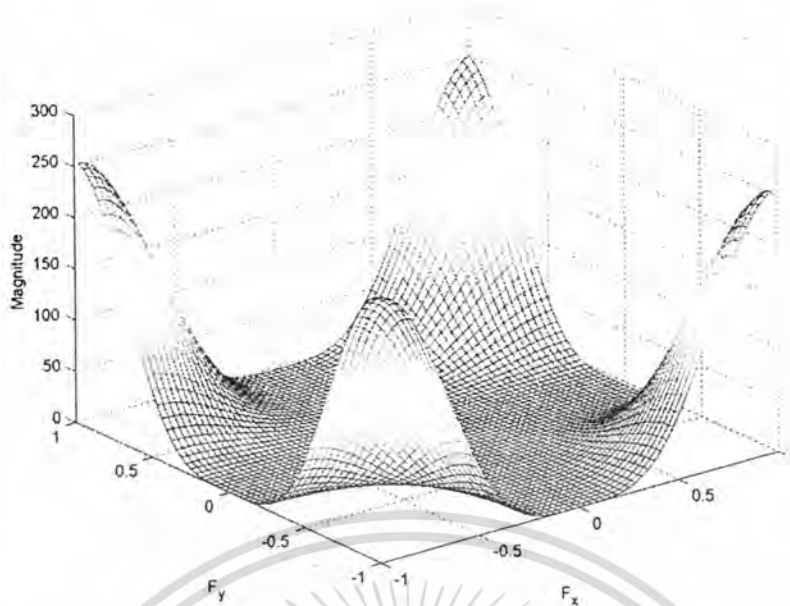
ในส่วนของวงจรกรองปาสคาล 2 มิตินี้ เราทำการแสดงผลตอบสนองทางความถี่ได้ดังรูป



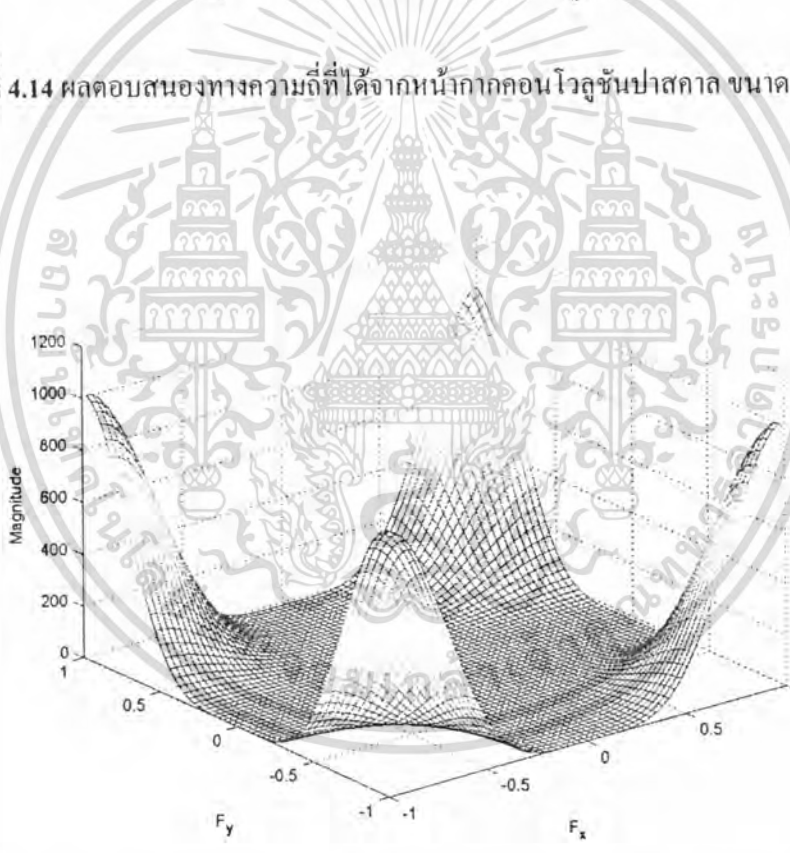
รูปที่ 4.12 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสกาล ขนาด 3x3



รูปที่ 4.13 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนโวลูชันปาสกาล ขนาด 4x4

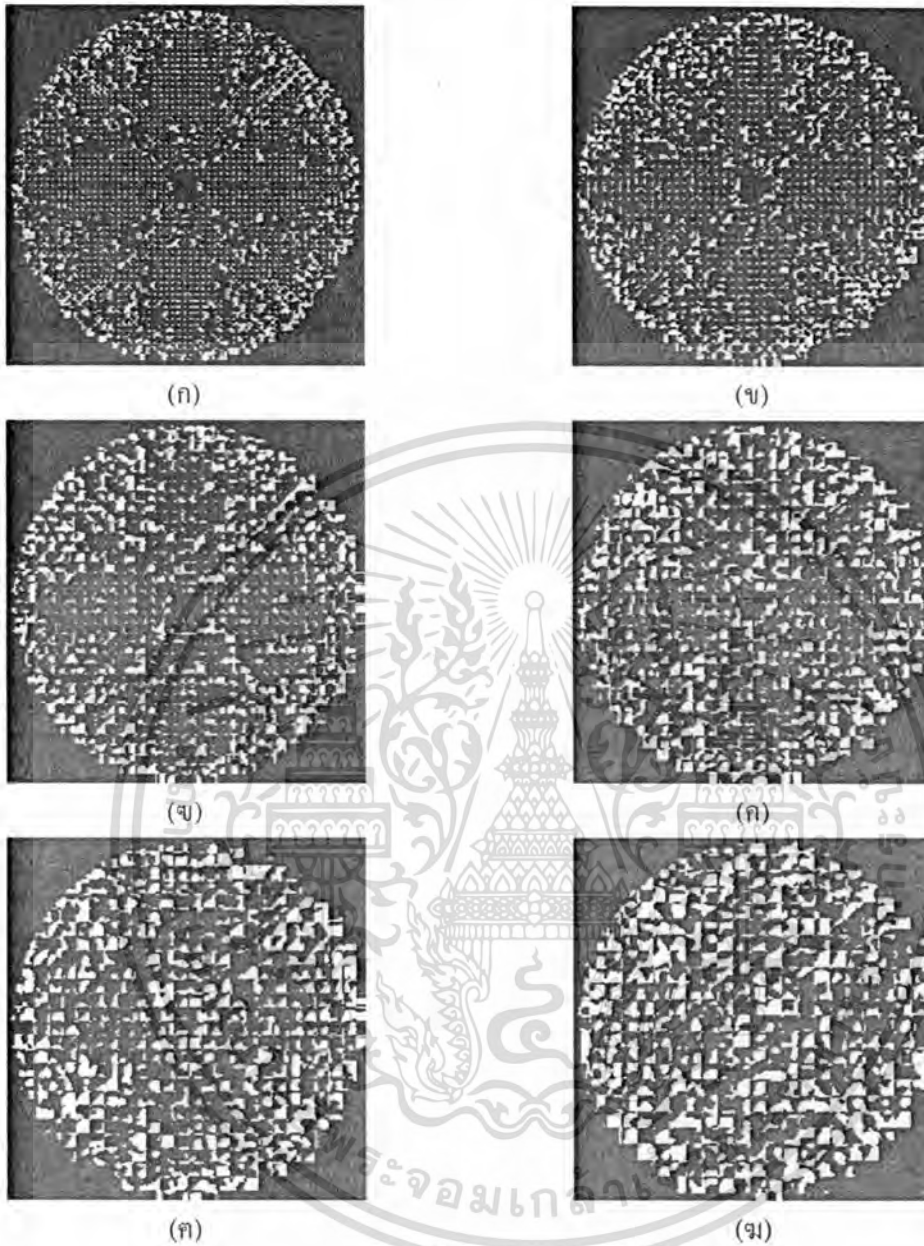


รูปที่ 4.14 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนไวลูชันพลาสติก ขนาด 5x5



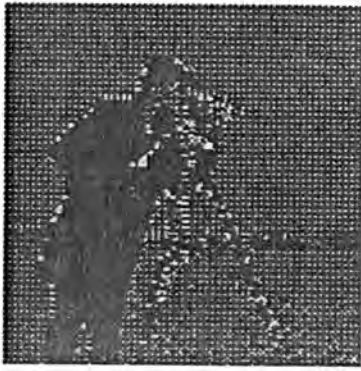
รูปที่ 4.15 ผลตอบสนองทางความถี่ที่ได้จากหน้ากากคอนไวลูชันพลาสติก ขนาด 6x6

จากนั้นทำการแปลงปาสคาลเต็มหน่วย 2 มิติ ดังใน [1] กับสัญญาณอินพุตที่เป็นภาพ จะได้ดังรูปที่ 4.16



รูปที่ 4.16 ภาพที่ได้จากการแปลงปาสคาลแบบเต็มหน่วย ตัวอย่างที่ 1

- |                            |                            |
|----------------------------|----------------------------|
| (ก) ปาสคาลเมตริกซ์ขนาด 3x3 | (ข) ปาสคาลเมตริกซ์ขนาด 4x4 |
| (ค) ปาสคาลเมตริกซ์ขนาด 5x5 | (ด) ปาสคาลเมตริกซ์ขนาด 6x6 |
| (จ) ปาสคาลเมตริกซ์ขนาด 7x7 | (ฉ) ปาสคาลเมตริกซ์ขนาด 8x8 |



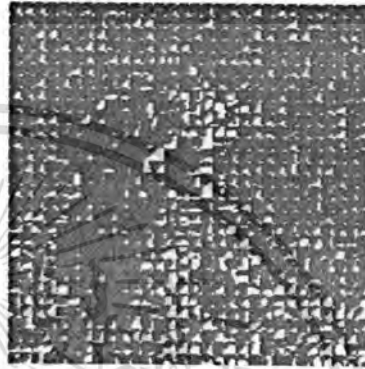
(ก)



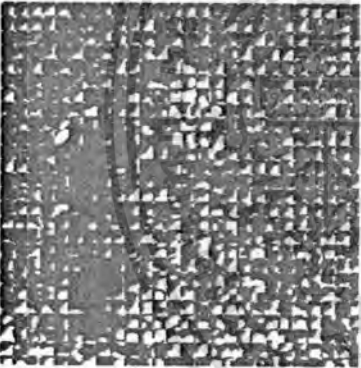
(ข)



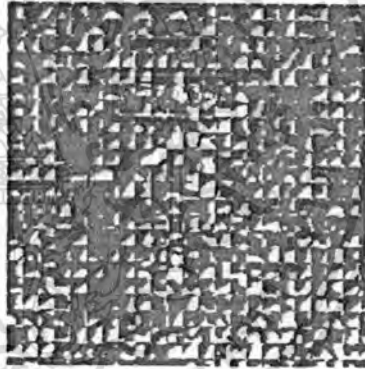
(ค)



(ง)



(ฉ)

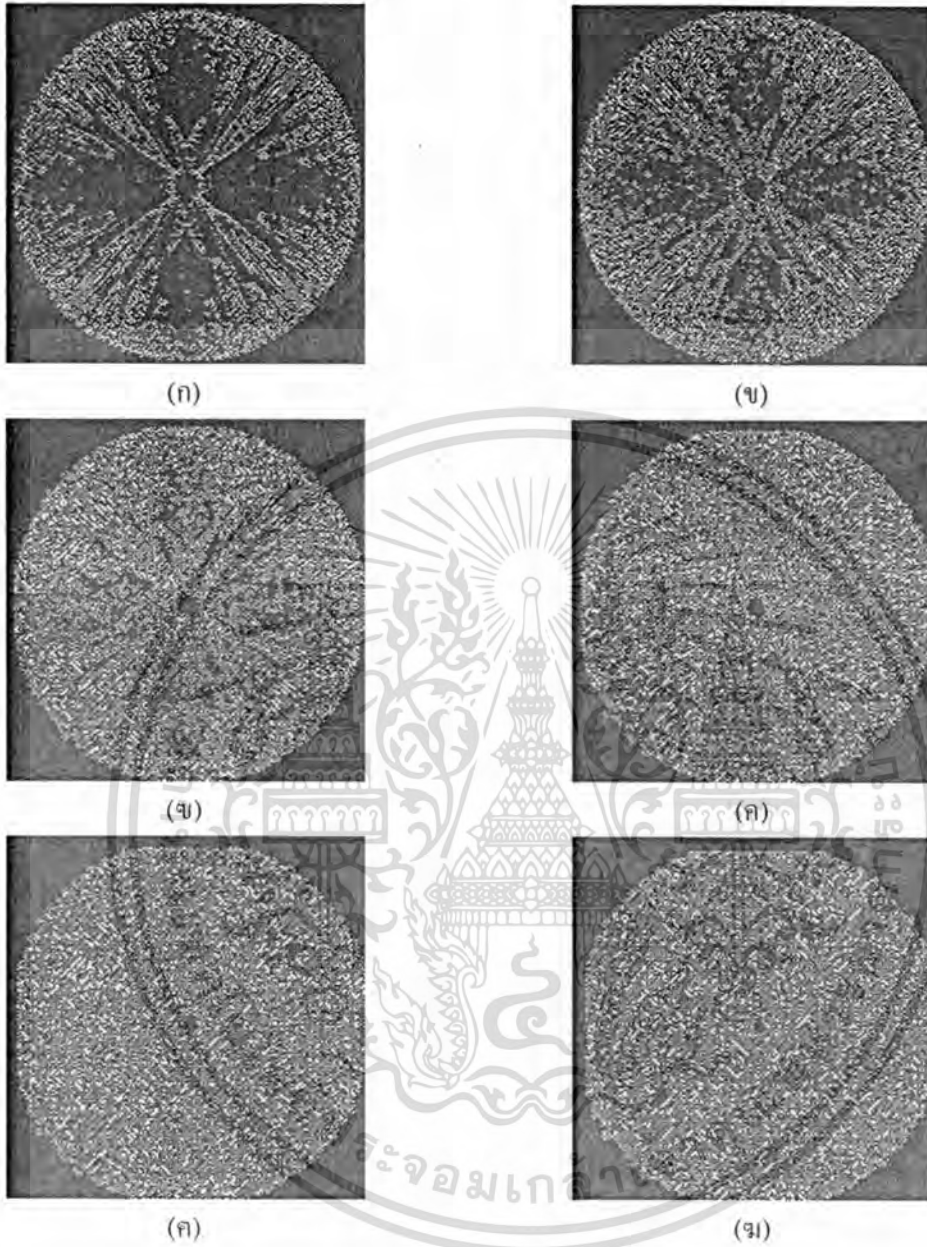


(ช)

รูปที่ 4.17 ภาพที่ได้จากการแปลงดีสครีตฟูริเยร์แบบเต็มหน่วย ตัวอย่างที่ 2  
(ก) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 3x3 (ข) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 4x4  
(ค) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 5x5 (ง) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 6x6  
(ฉ) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 7x7 (ช) หน้ากากคอนโวลูชันฟูริเยร์ ขนาด 8x8

จากการจำลองการทำงานของการแปลงฟูริเยร์แบบเต็มหน่วยในฟูริเยร์ขนาดต่าง ๆ นั้น พบว่าไม่สามารถเน้นขอบภาพได้

จากนั้นทดลองนำภาพมาทำการประมวลผลโดยผ่านวงจรกรองปาสคาล 2 มิติ



รูปที่ 4.18 ภาพที่ได้จากวงจรกรองปาสคาล 2 มิติ ตัวอย่างที่ 1

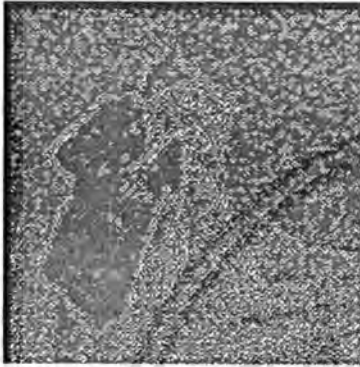
- (ก) หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 (ข) หน้ากากคอนโวลูชันปาสคาล ขนาด 4x4  
(ค) หน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 (ด) หน้ากากคอนโวลูชันปาสคาล ขนาด 6x6  
(e) หน้ากากคอนโวลูชันปาสคาล ขนาด 7x7 (f) หน้ากากคอนโวลูชันปาสคาล ขนาด 8x8



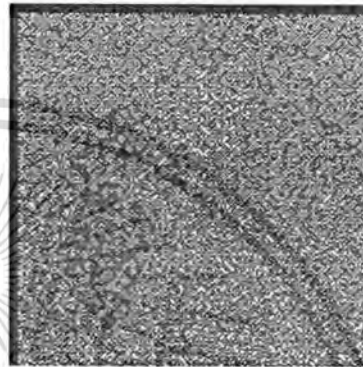
(ก)



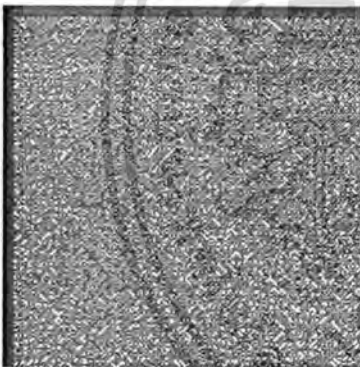
(ข)



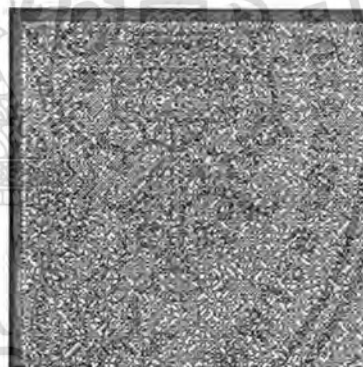
(ค)



(ด)



(จ)



(ฉ)

รูปที่ 4.19 การแปลงปาสคาลไคเตอร์ ตัวอย่างที่ 2

(ก) หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 (ข) หน้ากากคอนโวลูชันปาสคาล ขนาด 4x4

(ค) หน้ากากคอนโวลูชันปาสคาล ขนาด 5x5 (ด) หน้ากากคอนโวลูชันปาสคาล ขนาด 6x6

(จ) หน้ากากคอนโวลูชันปาสคาล ขนาด 7x7 (ฉ) หน้ากากคอนโวลูชันปาสคาล ขนาด 8x8

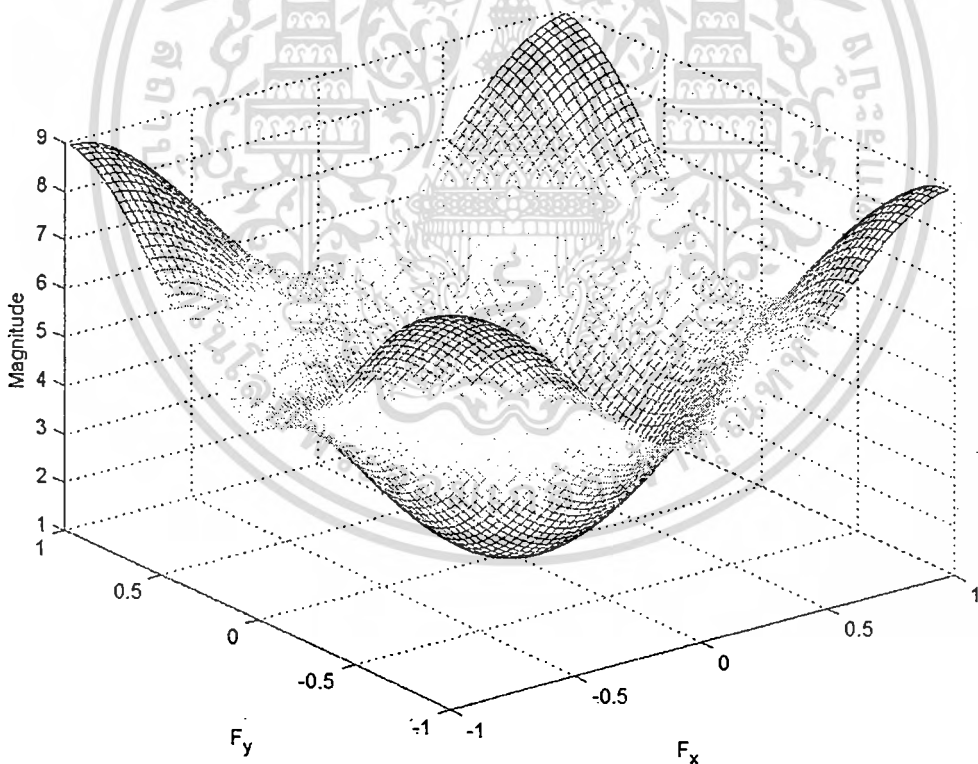
จะเห็นได้ว่าการกรองปาสคาล 2 มิติ โดยใช้หน้ากากคอนโวลูชันปาสคาล ขนาด 3x3 นั้นจะให้ผลลัพธ์  
ออกมาดีที่สุดและโครงสร้างของวงจรกรองเมื่อทำการ โปรแกรมลงบนเฟิร์มแวร์จะใช้อุปกรณ์ น้อยกว่า  
หน้ากากคอนโวลูชันปาสคาลขนาดอื่น ๆ

#### 4.1.2.2.1 เปรียบเทียบผลที่ได้จากวงจรกรองปาสคาลกับวงจรกรองความถี่สูงผ่านทั่วไป

ตัวกรองความถี่สูงผ่าน ความหมายโดยทั่วไปในการประมวลผลภาพเป็นการเน้นส่วนที่เป็นความถี่สูงหรือระดับความแตกต่างความเข้มของแสงมาก ขณะที่ลดส่วนที่เป็นความถี่ต่ำหรือระดับความแตกต่างความเข้มของแสงน้อย เพราะว่าขอบ (Edge) หรือรายละเอียดของภาพเป็นส่วนของความถี่สูง ดังนั้นการกรองความถี่สูงผ่านจะเป็นการเพิ่มความแตกต่างของแสงแต่ละที่ (Local Contrast) และ ความคมชัด (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



(ก)



(ข)

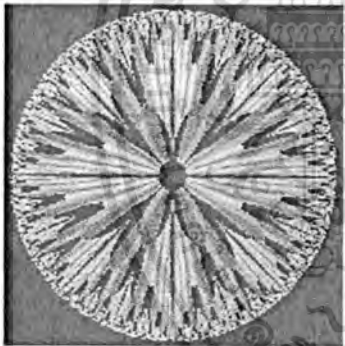


(ง)

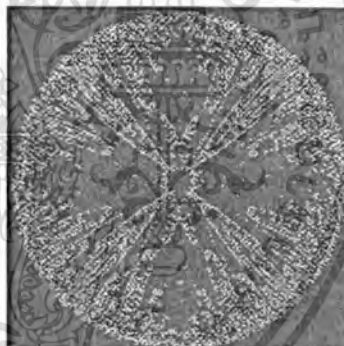


(ค)

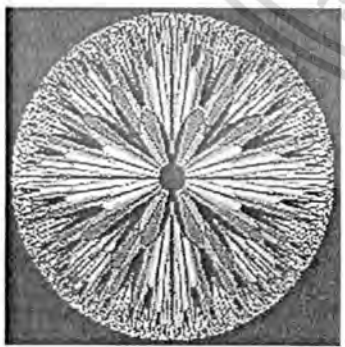
ตัวอย่างที่ 1



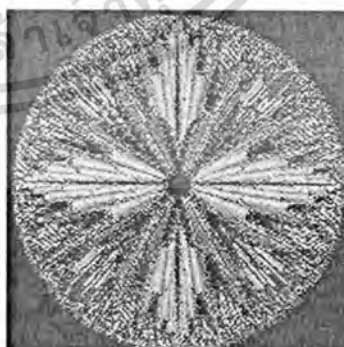
(ก)



(ข)



(ง)



(ค)

ตัวอย่างที่ 2

รูปที่ 4.22 เปรียบเทียบวงจรรองความถี่สูงผ่านทั่วไปกับวงจรรองปาสคาล

(ก) รูปต้นแบบ

(ข) รูปที่ผ่านวงจรรองปาสคาล

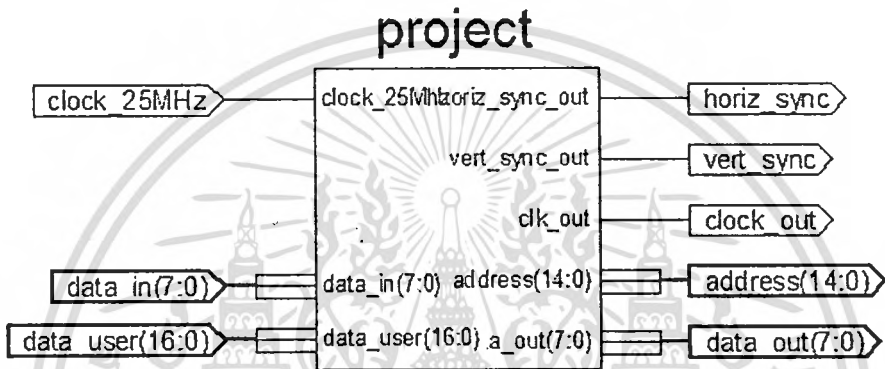
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง

(ข) รูปที่ผ่านวงจรตรวจสอบความถี่สูงผ่าน (ค) รูปต้นแบบบวกกับรูปที่ผ่านวงจรกรองปาสคาล

จากรูปที่ 4.22 (ข) ทั้งในตัวอย่างที่ 1 และตัวอย่างที่ 2 เป็นการจำลองการทำงานตัวกรองความถี่สูงผ่าน จะเห็นได้ว่ารูปที่ได้นั้นเป็นรูปที่คงค่าคล้ายเดิมไว้ และขอบภาพมีการเน้นชัดขึ้น ซึ่งแตกต่างจากการผ่านวงจรกรองปาสคาลที่กรองออกมาได้เป็นขอบภาพเท่านั้น ตามรูปที่ 4.22 (ข) ในตัวอย่างที่ 1 และที่ 2 จึงทดลองนำรูปต้นแบบบวกกับรูปที่ได้จากการผ่านวงจรกรองปาสคาล จะได้ว่ารูปที่ 4.22 (ค) ในตัวอย่างที่ 1 และตัวอย่างที่ 2

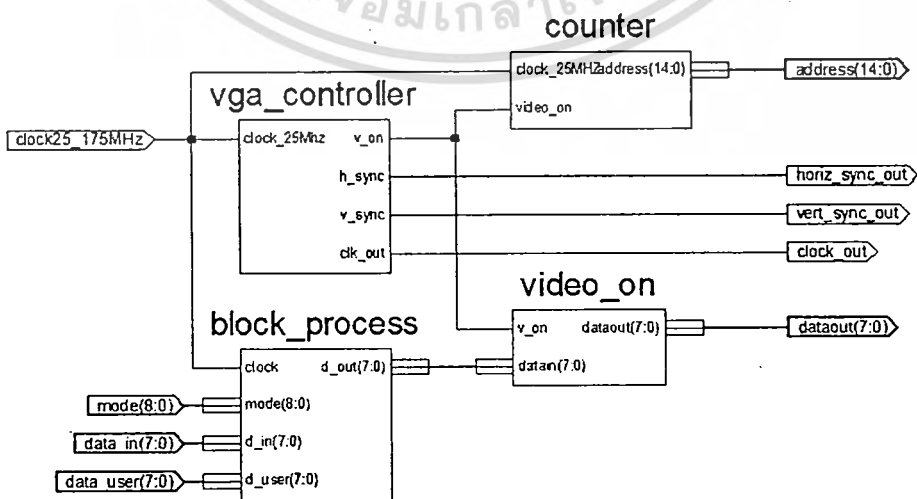
#### 4.1.3 การออกแบบวงจรส่วนต่าง ๆ โดยใช้ภาษาวีเอชดีแอล

##### 4.1.3.1 วงจรส่วนของตัวควบคุมวีจีเอ



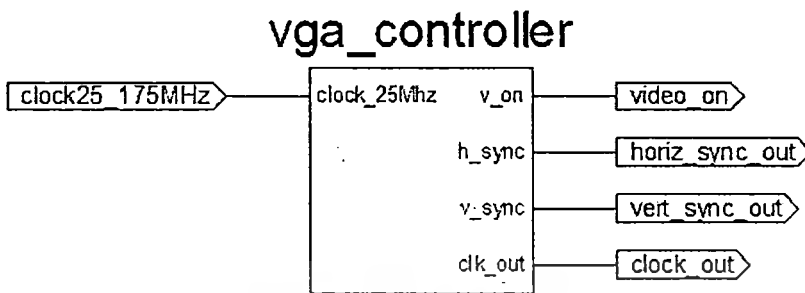
รูปที่ 4.23 สัญลักษณ์ส่วนของตัวควบคุมวีจีเอ

จากรูปที่ 4.23 เป็น โมดูลของวงจรที่ใช้ในการควบคุมการแสดงผลของหน้าจอวีจีเอ และการประมวลผลทั้งหมด ซึ่งภายในจะประกอบด้วย 4 วงจรหลัก ๆ ได้แก่ วงจรในส่วนของการสร้างสัญญาณควบคุมการสแกนทางแนวตั้งและแนวนอนให้กับวีจีเอ, วงจรประมวลผลข้อมูลร่วมกับสัญญาณกำหนดการแสดงผลบนหน้าจอ วงจรประมวลผลภาพ และวงจรสร้างตำแหน่งของข้อมูล ดังรูปที่ 4.24



รูปที่ 4.24 สัญลักษณ์ของวงจรภายในทั้งหมดที่ใช้ในการควบคุมวีซีเอ

4.1.3.2 วงจรในส่วนของการสร้างสัญญาณควบคุมการสแกนทางแนวตั้งและแนวนอนให้กับวีซีเอ



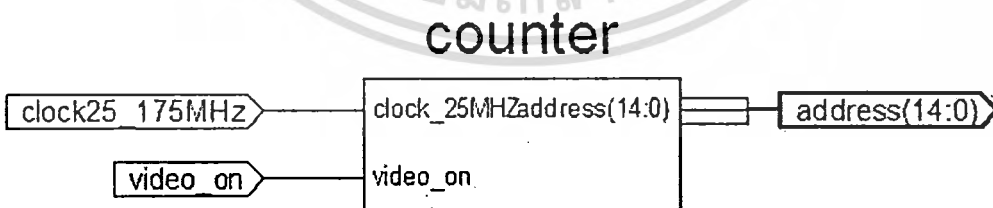
รูปที่ 4.25 สัญลักษณ์ส่วนของภาคขับสัญญาณ

4.1.3.3 วงจรส่วนของภาครับของหน่วยความจำภายนอก



รูปที่ 4.26 สัญลักษณ์ส่วนของภาครับของหน่วยความจำภายนอก

4.1.3.4 วงจรสร้างตำแหน่งของข้อมูล

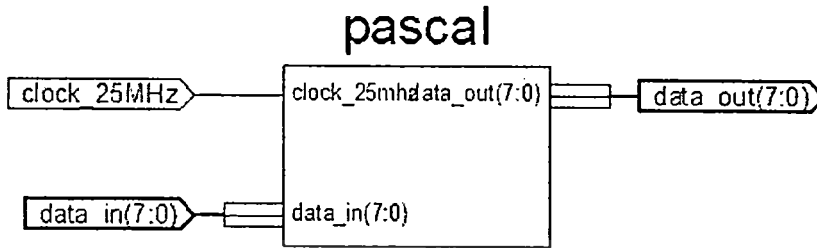


รูปที่ 4.27 สัญลักษณ์ส่วนของวงจรสร้างตำแหน่งของข้อมูล

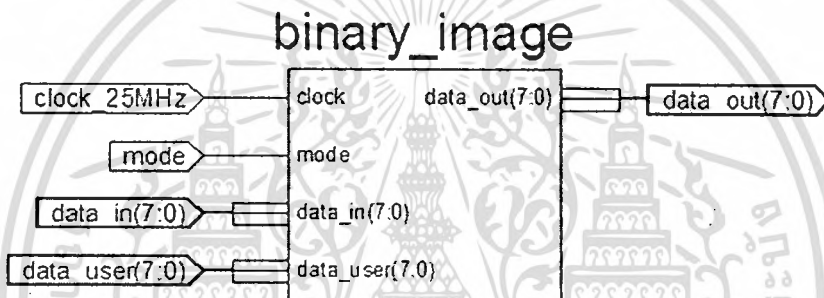
จากรูปที่ 4.27 จะเป็นวงจรสร้างตำแหน่งของข้อมูล ซึ่งจะอาศัยวงจรรนับ (Counter) มาช่วยในการออกแบบ

วงจร

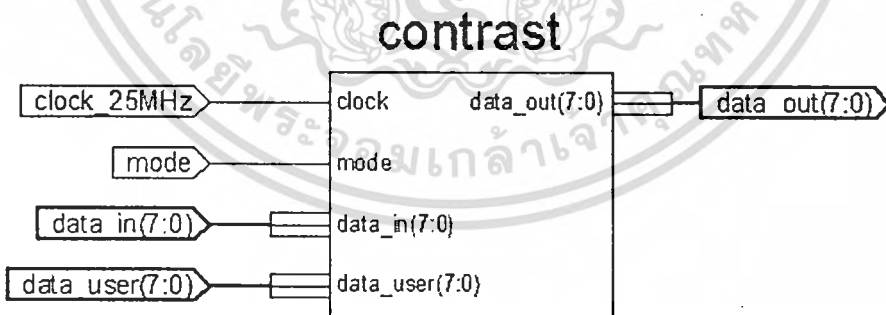
#### 4.1.3.5 วงจรที่ใช้ในการประมวลผลภาพ



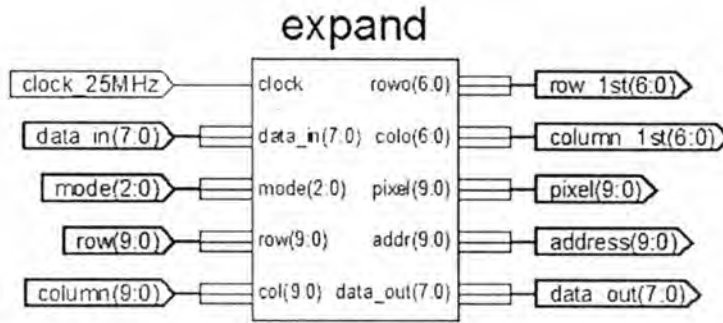
รูปที่ 4.28 สัญลักษณ์ส่วนของวงจรที่ใช้ในการประมวลผลภาพโดยอาศัยการแปลงปาดกาล



รูปที่ 4.29 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพเป็นสองระดับ



รูปที่ 4.30 สัญลักษณ์ส่วนของวงจรปรับความเข้มแสงของภาพ



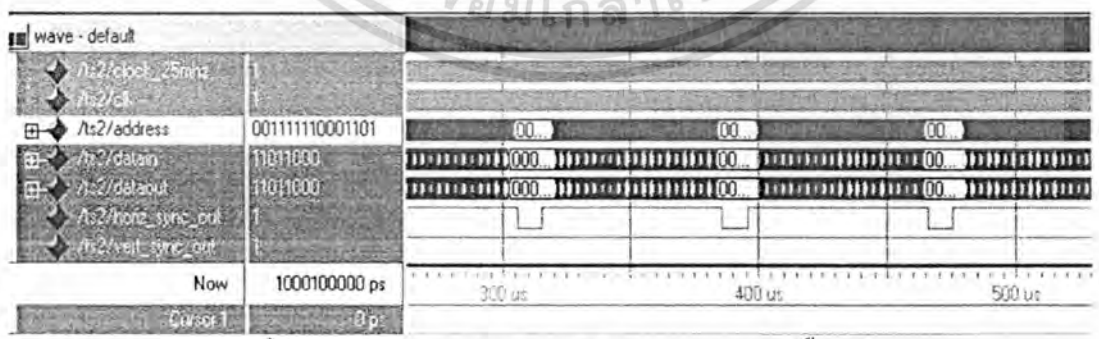
รูปที่ 4.31 สัญลักษณ์ส่วนของวงจรขยายขนาดของภาพและจัดตำแหน่งของภาพ

4.2 ผลการทดลอง

- 4.2.1 ผลการทดลองจากการจำลองการทำงานในวีเอชดีแอล
- 4.2.1.1 เมื่อทำการนำภาพที่ยังไม่ได้ทำการประมวลผลมาแสดงบนจอวีจีเอ
- 4.2.1.1.1 การทำงานของวงจรรวมทั้งหมด

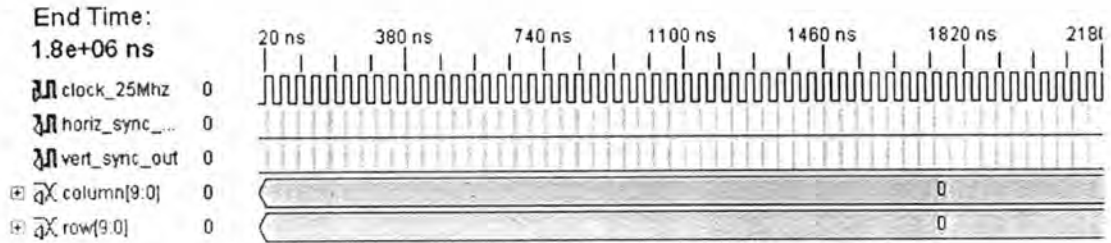


รูปที่ 4.32 ระบบทดสอบ (Testbench) ของวงจรรวมทั้งหมด



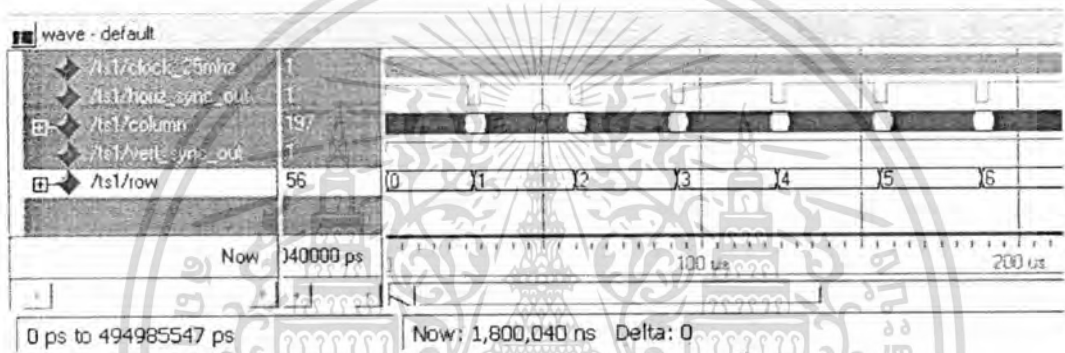
รูปที่ 4.33 แสดงผลการจำลองการทำงานของวงจรรวมทั้งหมด

#### 4.2.1.1.2 การสร้างสัญญาณซิงโครไนซ์และสัญญาณตี



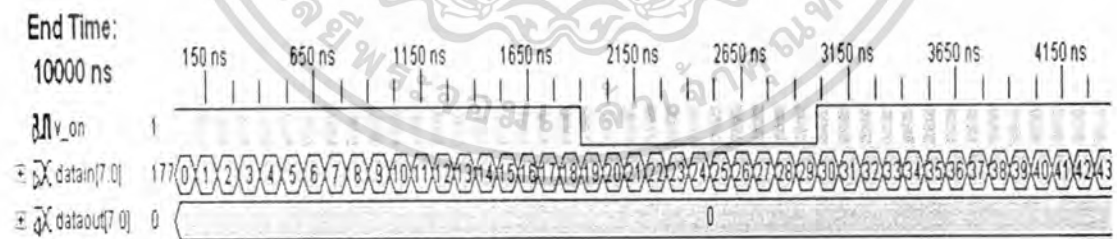
รูปที่ 4.34 ระบบทดสอบของสัญญาณอ้างอิงทางแนวตั้งและแนวนอน

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (Simulation) ได้ดังนี้

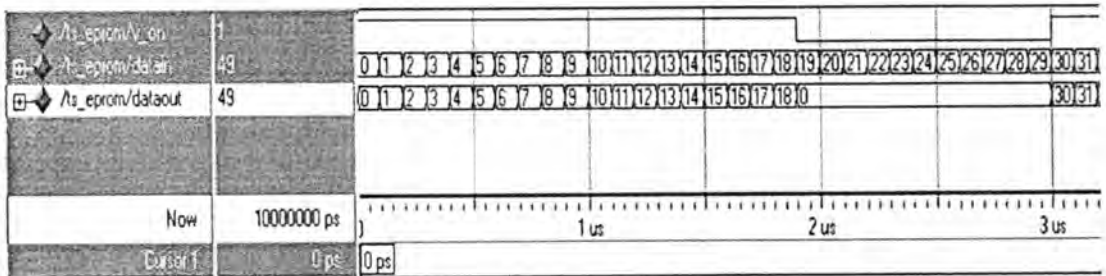


รูปที่ 4.35 แสดงผลการจำลองสัญญาณอ้างอิงทางแนวตั้งและแนวนอน

#### 4.2.1.1.3 การทำงานของภาครับของหน่วยความจำภายนอก



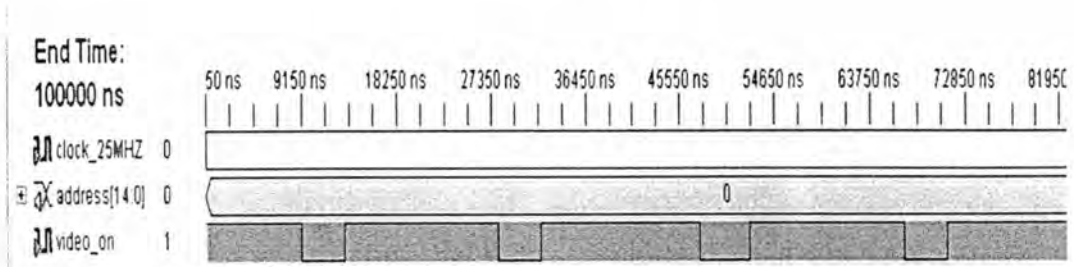
รูปที่ 4.36 ระบบทดสอบของภาครับของหน่วยความจำภายนอก



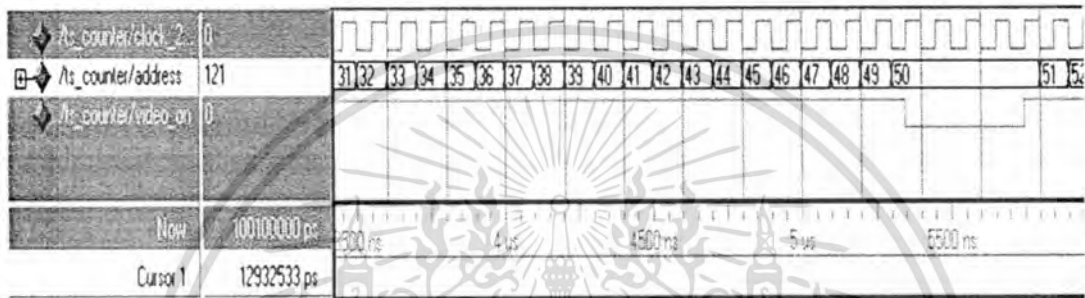
รูปที่ 4.37 แสดงผลการจำลองการทำงานของภาครับของหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้า

#### 4.2.1.1.4 การทำงานของวงจรสร้างตำแหน่งของข้อมูล



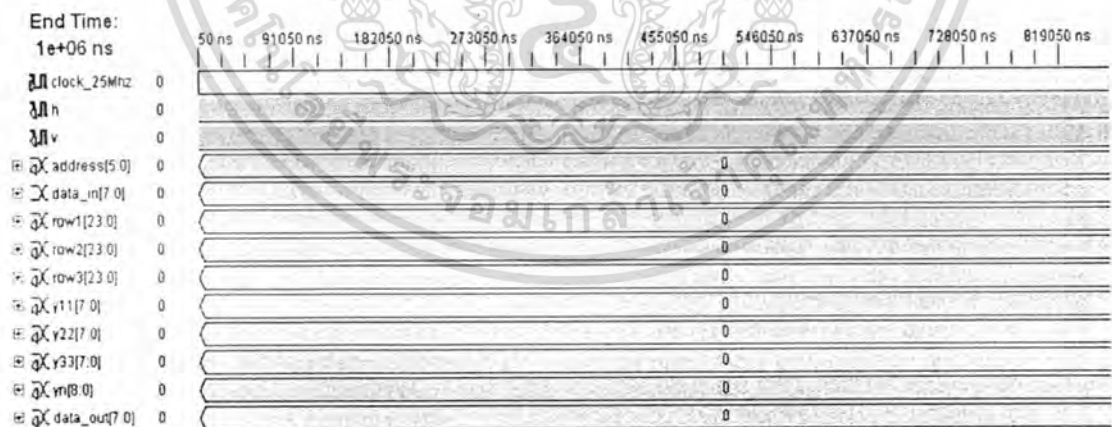
รูปที่ 4.38 ระบบทดสอบของวงจรสร้างตำแหน่งของข้อมูล



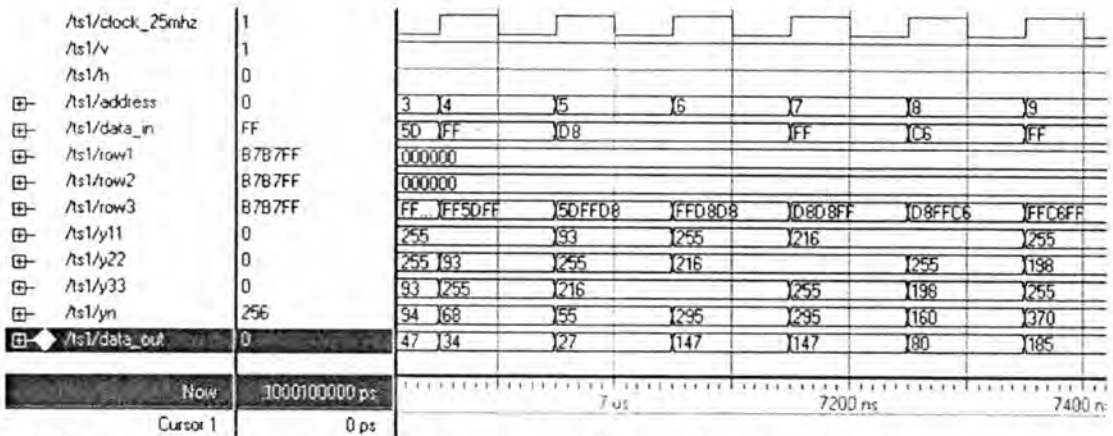
รูปที่ 4.39 แสดงผลการจำลองการทำงานของวงจรสร้างตำแหน่งของข้อมูล

#### 4.2.1.2 เมื่อทำการนำภาพที่ผ่านการประมวลผลเรียบร้อยแล้วมาแสดงบนจอวีจีเอ

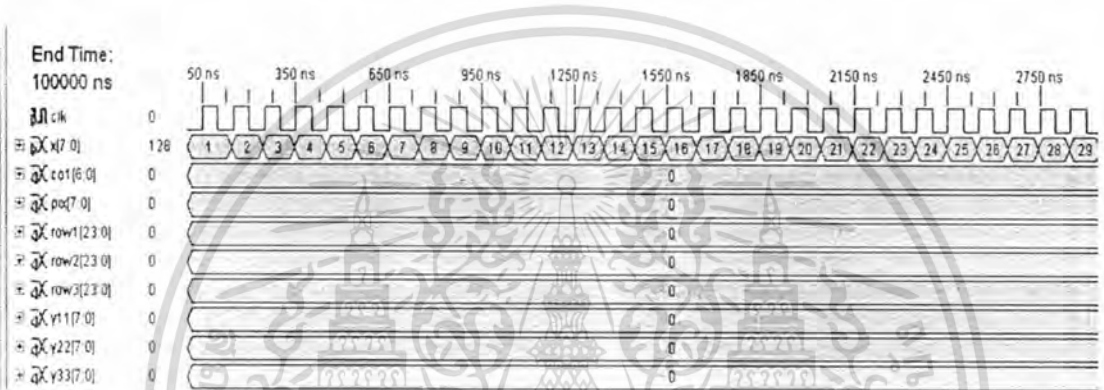
##### 4.2.1.2.1 การทำงานของวงจรรวมทั้งหมด



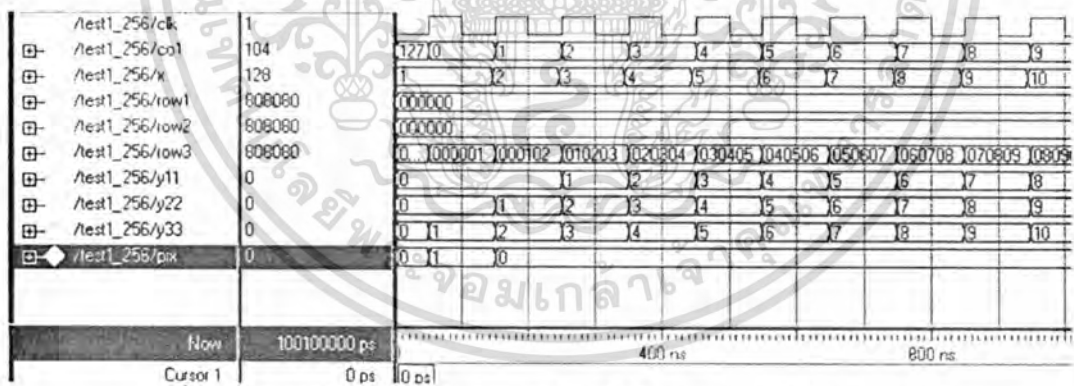
รูปที่ 4.40 ระบบทดสอบของวงจรรวมทั้งหมด (เมื่อมีส่วนของการประมวลผลภาพด้วย)



รูปที่ 4.41 แสดงผลการจำลองการทำงานของวงจรทั้งหมด (เมื่อมีส่วนของการประมวลผลภาพด้วย)



รูปที่ 4.42 ระบบทดสอบของวงจรในส่วนของการประมวลผลภาพ

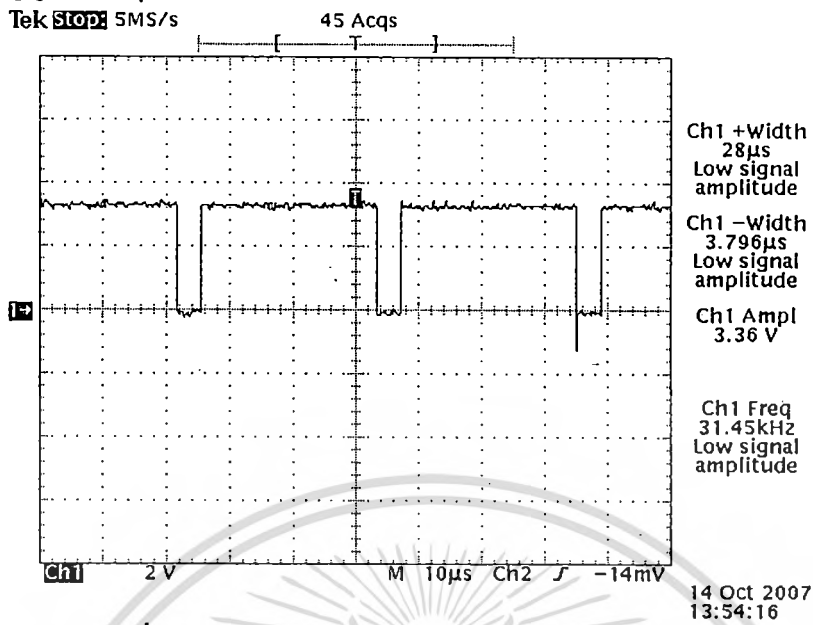


รูปที่ 4.43 ผลการจำลองการทำงานของวงจรในส่วนของการประมวลผลภาพ

## 4.2.2 ผลการทดลองที่ได้จากฮาร์ดแวร์

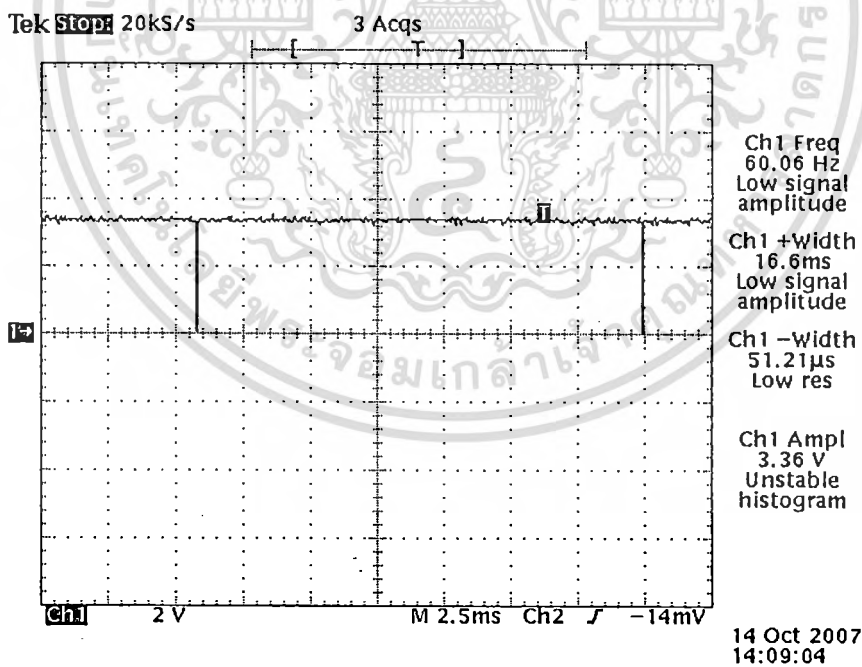
วัตถุประสงค์จริงที่ได้จากการจำลองสัญญาณ โดยใช้ฮาร์ดแวร์ไมโครคอนโทรลเลอร์มาวัดสัญญาณควบคุมการสแกนทางแนวนอน และสัญญาณควบคุมการสแกนทางแนวตั้ง ได้ดังนี้

#### 4.2.2.1 สัญญาณควบคุมการสแกนหน้าจอบางแนวอนและแนวตั้ง



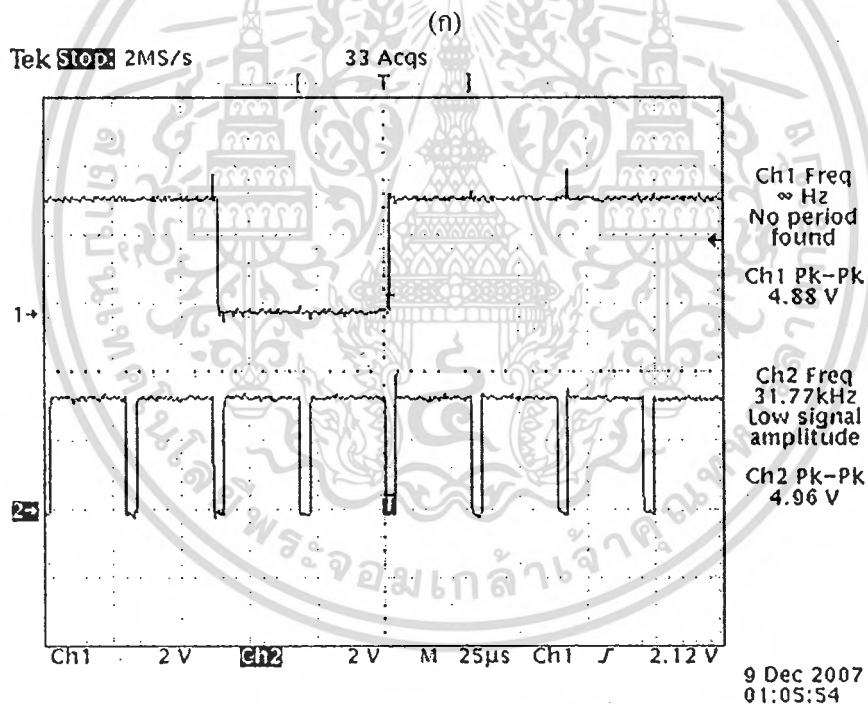
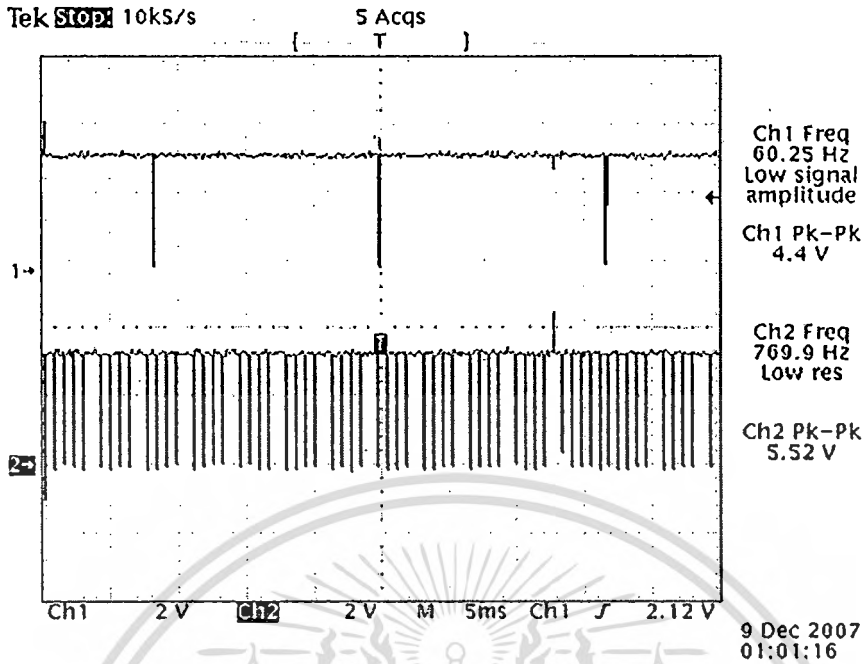
รูปที่ 4.44 สัญญาณสัญญาณควบคุมการสแกนทางแนวอน

จากรูปที่ 4.44 เป็นการวัดสัญญาณการควบคุมการสแกนทางแนวอนพบว่ามีความถี่เท่ากับ 31.45 kHz ส่วนจากการคำนวณนั้น จะมีค่าเท่ากับ 31.467 kHz



รูปที่ 4.45 สัญญาณสัญญาณควบคุมการสแกนทางแนวตั้ง

จากรูปที่ 4.45 เป็นการวัดสัญญาณการควบคุมการสแกนทางแนวตั้งพบว่ามีความถี่เท่ากับ 60.06 Hz ส่วนจากการคำนวณนั้น จะมีค่าเท่ากับ 60 Hz

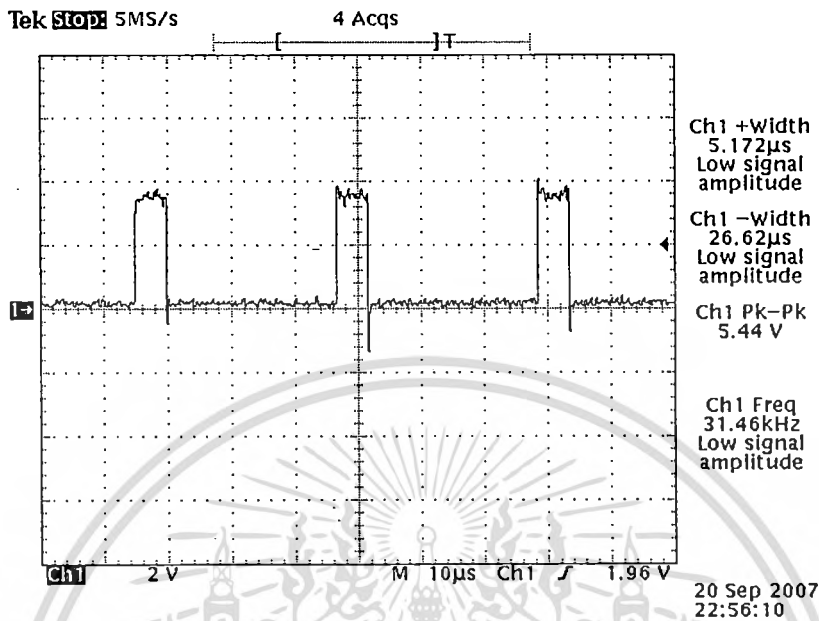


(ข)

รูปที่ 4.46 เมื่อทำการวัดสัญญาณสัญญาณควบคุมการสแกนทางแนวนอน  
เปรียบเทียบกับสัญญาณควบคุมการสแกนทางแนวตั้ง

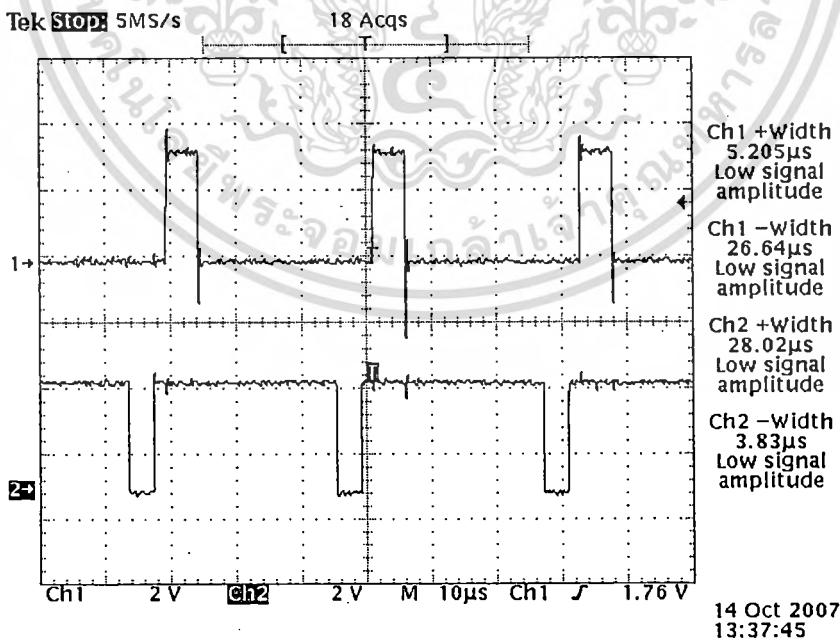
จากรูปที่ 4.46 เป็นการวัดสัญญาณสัญญาณควบคุมการสแกนทางแนวนอนเปรียบเทียบกับสัญญาณ  
ควบคุมการสแกนทางแนวตั้งและเมื่อขยายสัญญาณตามรูปที่ 4.46 (ข) เพื่อให้เห็นถึงลักษณะของสัญญาณทั้งสองได้  
อย่างชัดเจน

4.2.2.2 สัญญาณวิดีโอ (Video on) เทียบกับสัญญาณข้อมูล

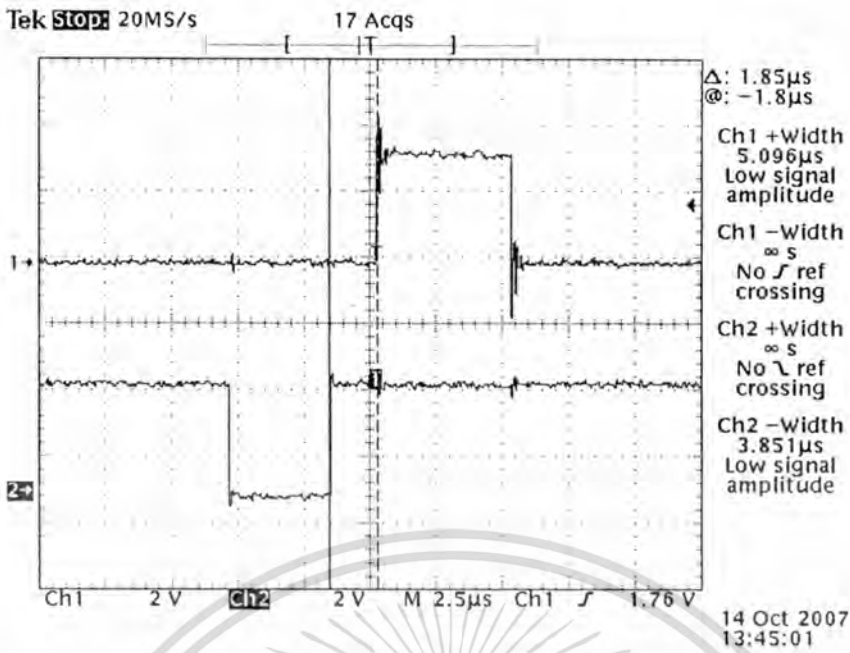


รูปที่ 4.47 สัญญาณวิดีโออน

จากรูปที่ 4.47 เป็นการวัดสัญญาณวิดีโออนพบว่าระดับสัญญาณช่วงที่มีการแสดงข้อมูลภาพขนาด 128 พิกเซลต่อแถว (ลอจิก '1') จะมีขนาดเท่ากับ 5.17  $\mu$ s ซึ่งค่าที่ได้จากการคำนวณมีขนาดเท่ากับ 5.08  $\mu$ s



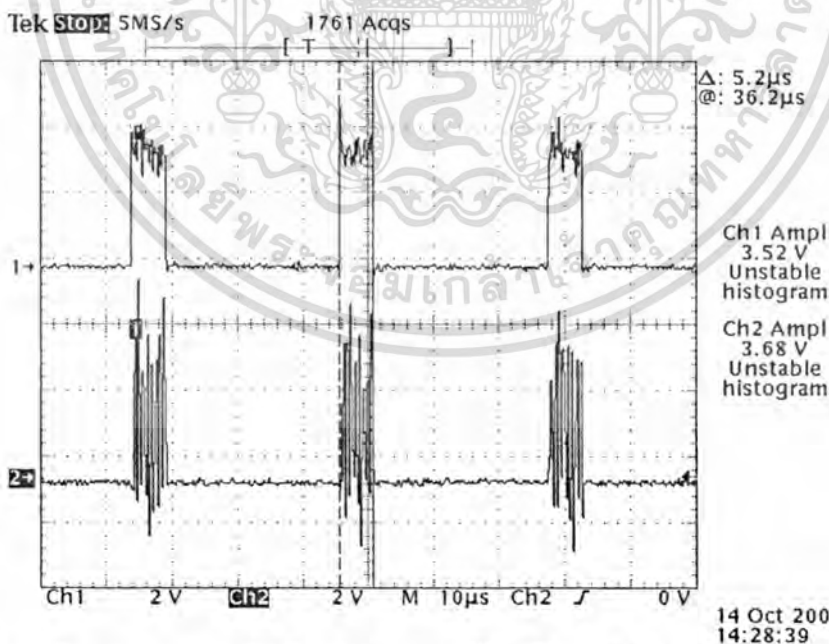
(ก)



(ข)

รูปที่ 4.48 สัญญาณวีดีโอออนวืดเทียบกับสัญญาณควบคุมการสแกนทางแนวนอน

จากรูปที่ 4.48 ในช่องสัญญาณที่ 1 เป็นสัญญาณวีดีโอออนและในช่องสัญญาณที่ 2 เป็นสัญญาณ ควบคุมการสแกนทางแนวนอนในรูปที่ 4.48 (ข) จะพบว่าคาร์ดเบนซ์ของสัญญาณสัญญาณควบคุมการสแกนทางแนวนอนมีขนาด 1.85 µs และทางทฤษฎีพบว่าจะมีค่า 1.89 µs

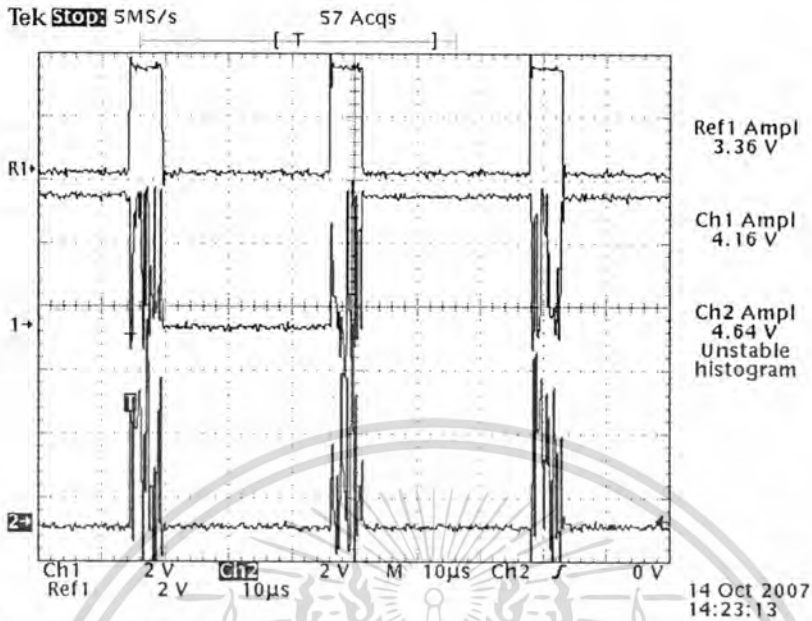


รูปที่ 4.49 สัญญาณ video on วืดเทียบกับแอดเดรสที่เฟฟฟี่เจส่งให้กับหน่วยความจำภายนอก

จากรูปที่ 4.49 จะเห็นได้ว่าข้อมูลที่ชี้ตำแหน่งแอดเดรสนั้นจะถูกส่งให้กับหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง

เมื่อสัญญาณวีดีโออนเป็นลอจิก '1' แต่หากสัญญาณวีดีโออนเป็นลอจิก '0' ข้อมูลก็จะไม่ถูกส่งออกไป

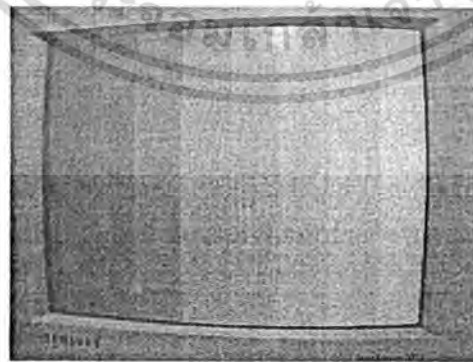


รูปที่ 4.50 สัญญาณวีดีโอวัดเทียบกับสัญญาณเอาต์พุตจากหน่วยความจำภายนอกให้กับเอฟพีจีและเอาต์พุตจากเอฟพีจีเอที่จะส่งให้กับวีจีเอ

จากรูปที่ 4.50 จะเห็นได้ว่าข้อมูลเอาต์พุตที่รับมาจากหน่วยความจำภายนอกจะถูกส่งจากเอฟพีจีเอไปให้หน้าจอวีจีเอเมื่อสัญญาณวีดีโออนเป็นลอจิก '1' แต่หากสัญญาณวีดีโออนเป็นลอจิก '0' ข้อมูลก็จะไม่ถูกส่งออกไป

#### 4.2.3 ผลการทดลองที่ได้จากหน้าจอวีจีเอ

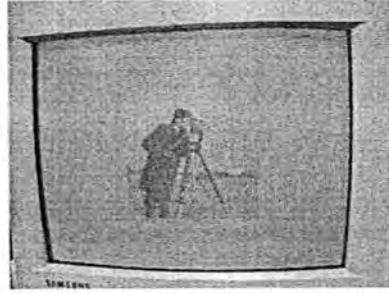
##### 4.2.3.1 ภาพที่แสดงผลทางหน้าจอวีจีเอเมื่อยังไม่ได้ผ่านการประมวลผลภาพ



รูปที่ 4.51 แสดงระดับแถบสีแบบเกรย์สเกล



(ก)



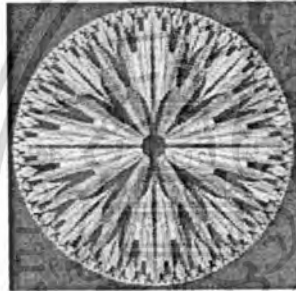
(ข)



(ค)



(ด)



(จ)



(ฉ)



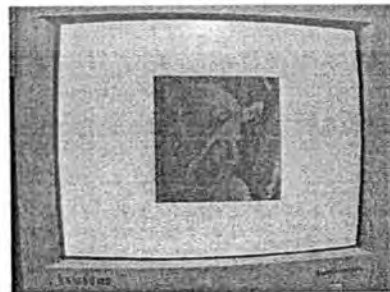
(ช)



(ค)



(ค)



(ค)

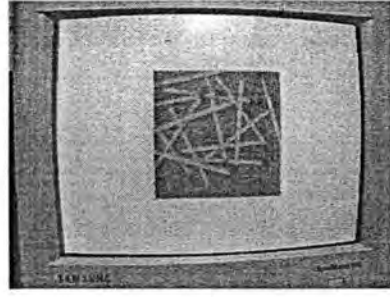
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง



By Sorawat Chivapreecha  
Telecommunication Eng. Dept.  
Faculty of Engineering  
King Mongkut's Institute of Technology Ladkrabang



(ซ)



(ฅ)

รูปที่ 4.52 การเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่นำมาแสดงผลทางหน้าจอวีซีเอ



(ก)



(ข)

รูปที่ 4.53 แสดงภาพต้นแบบแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1



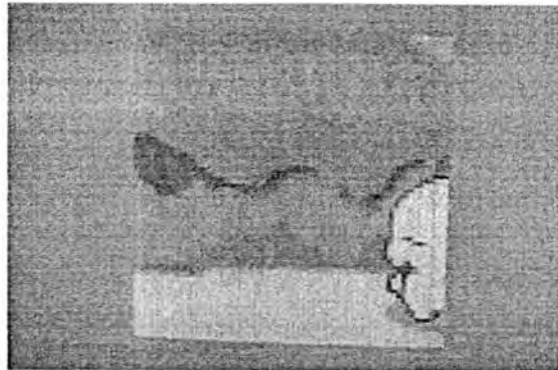
(ก)

(ข)

รูปที่ 4.54 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 1



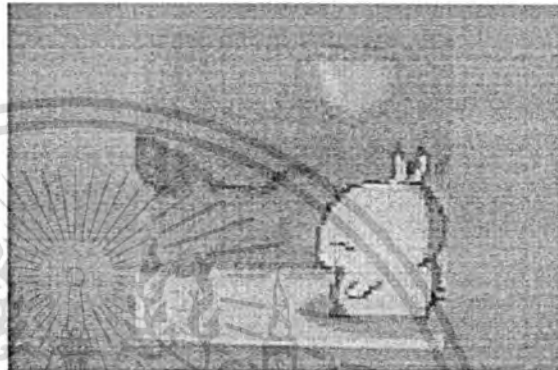
(ก)



(ข)



(ข)



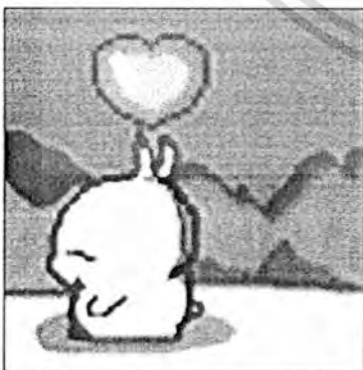
(ค)



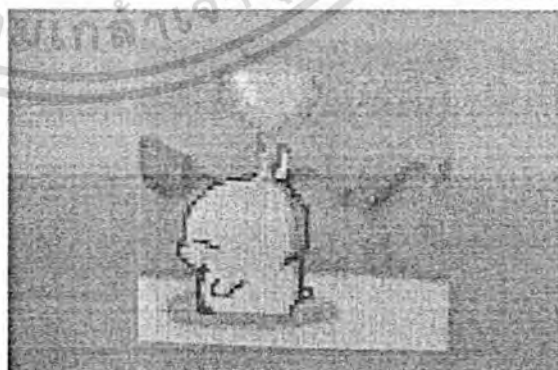
(ค)



(ง)



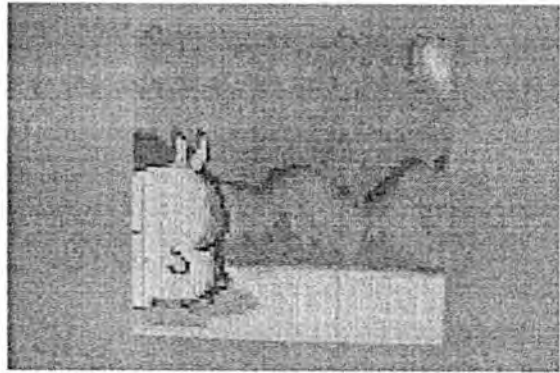
(ง)



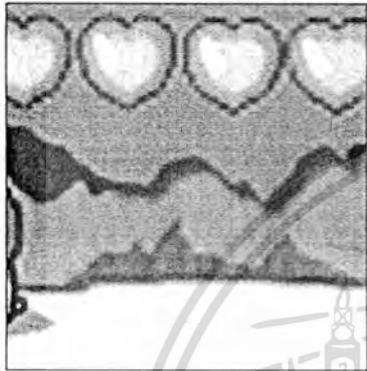
(จ)



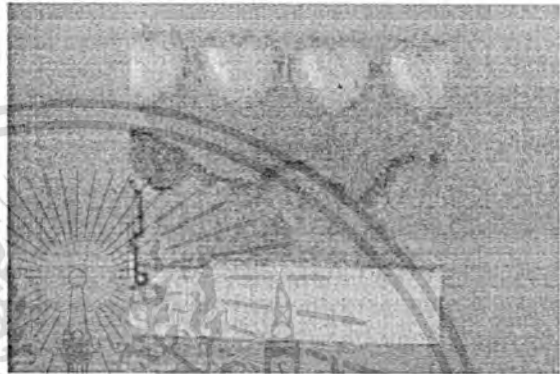
(ก)



(ข)



(ค)



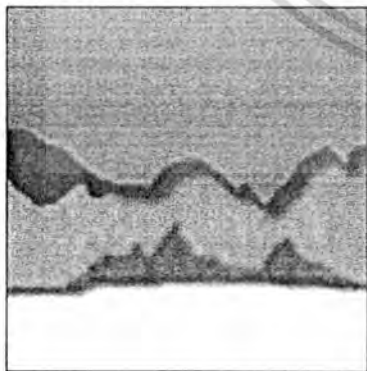
(ง)



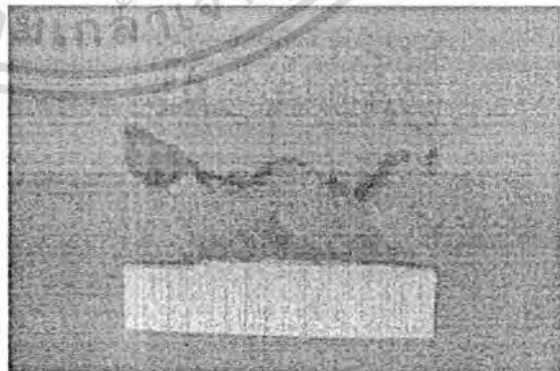
(ฉ)



(ช)

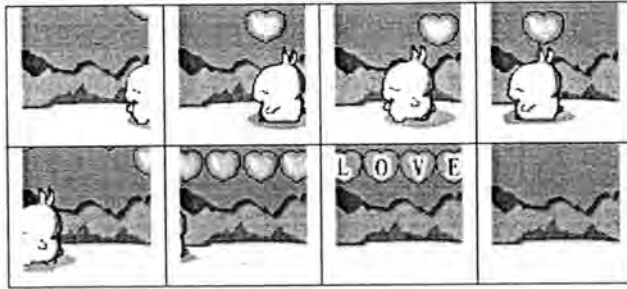


(ฉ)



(จ)

รูปที่ 4.55 แสดงภาพของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2



(๓)

รูปที่ 4.56 แสดงภาพต้นแบบของแต่ละเฟรมของภาพเคลื่อนไหว ตัวอย่างที่ 2

#### 4.2.3.2 ภาพที่แสดงผลทางหน้าจอวีซีเอเมื่อผ่านการประมวลผลภาพเรียบร้อยแล้ว

##### 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

โดยทำการเปรียบเทียบการปรับความเข้มแสงของภาพเป็นสองระดับจาก โปรแกรม MATLAB กับผลที่แสดงทางจอวีซีเอ



รูปที่ 4.57 ภาพต้นแบบ ตัวอย่างที่ 1



(ก)



(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "15"



(จ)



(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "63"



(ค)



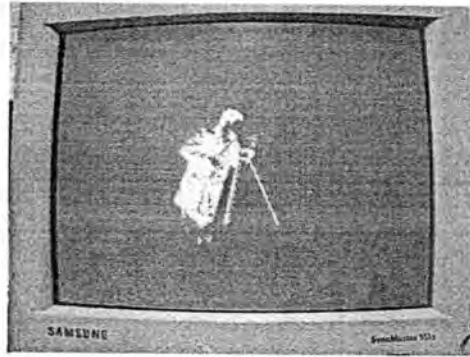
(ง)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "127"

รูปที่ 4.58 ในกรณีที่ค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1



(ก)

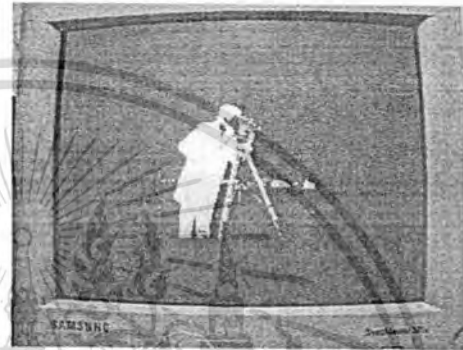


(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "15"



(จ)



(ค)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "63"



(ค)



(ง)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ "127"

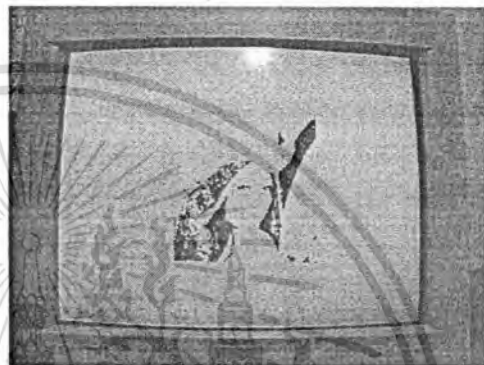
รูปที่ 4.59 ในกรณีที่ค่ามากกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 1



รูปที่ 4.60 ภาพต้นแบบ ตัวอย่างที่ 2



(ก)



(ข)

เมื่อป้อนค่าเทรชโฮลให้มีค่าเท่ากับ “7”



(ง)



(ค)

เมื่อป้อนค่าเทรชโฮลให้มีค่าเท่ากับ “31”



(จ)



(ฉ)

เมื่อป้อนค่าเทรชโฮลให้มีค่าเท่ากับ “63”

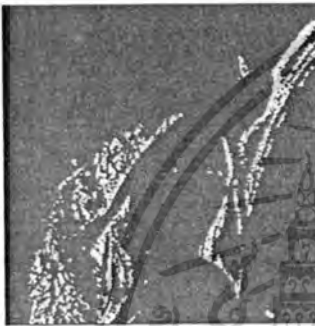


(ง)

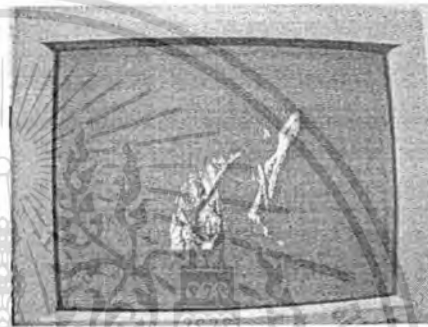


(จ)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “127”  
รูปที่ 4.61 ในกรณีที่มีค่าน้อยกว่าที่กำหนดให้แสดงสีดำ ตัวอย่างที่ 2



(ก)



(ข)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “7”



(ค)

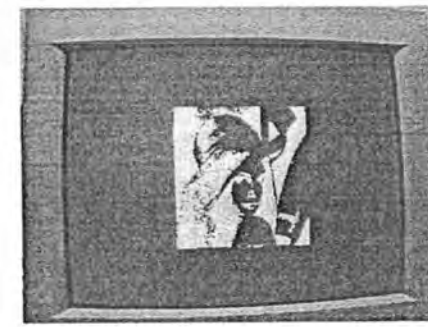


(ด)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “31”



(ค)

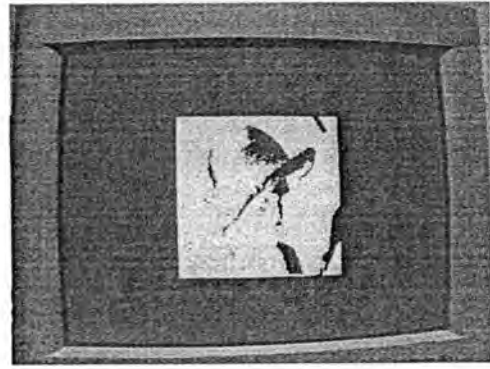


(ง)

เมื่อป้อนค่าเทรสโฮลให้มีค่าเท่ากับ “63”



(ง)



(จ)

เมื่อป้อนค่าเทรชโฮลให้มีค่าเท่ากับ “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 กับผลที่แสดงบนหน้าจอวีซีเอ



(ก)



(ข)

เมื่อลดระดับความเข้มแสงไป “15”



(ค)



(ง)

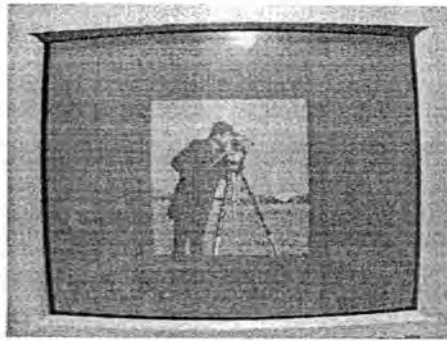
เมื่อลดระดับความเข้มแสงไป “31”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้า





(ค)

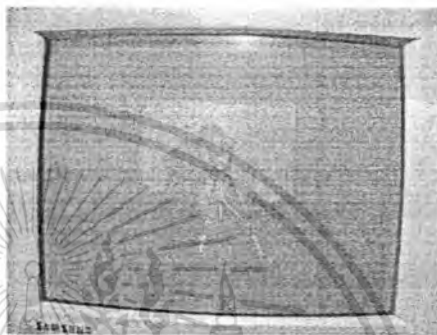


(ง)

เมื่อลดระดับความเข้มแสงไป“63”



(ง)



(จ)

เมื่อลดระดับความเข้มแสงไป“127”

รูปที่ 4.63 เมื่อทำการลดระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1



(ค)



(ง)

เมื่อเพิ่มระดับความเข้มแสงไป“15”



(ข)



(ค)

เมื่อเพิ่มระดับความเข้มแสงไป“31”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิง



By Sorawat Chivapreecha  
Telecommunication Eng. Dept.  
Faculty of Engineering  
King Mongkut's Institute of Technology Ladkrabang



(ค)

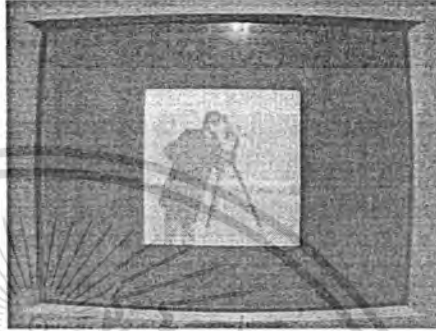


(ข)

เมื่อเพิ่มระดับความเข้มแสงไป“63”



(ง)



(จ)

เมื่อเพิ่มระดับความเข้มแสงไป“127”

รูปที่ 4.64 เมื่อทำการเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 1



(ก)

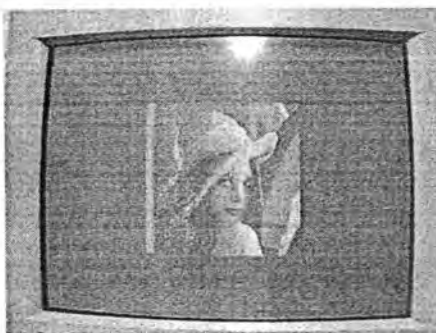


(ข)

เมื่อลดระดับความเข้มแสงไป“15”



(ข)

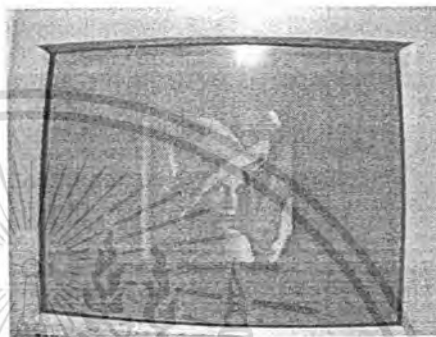


(ค)

เมื่อลดระดับความเข้มแสงไป“31”



(ค)



(ง)

เมื่อลดระดับความเข้มแสงไป“63”



(ง)



(จ)

เมื่อลดระดับความเข้มแสงไป“127”

รูปที่ 4.65 เมื่อทำลดเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2



(จ)



(ช)

เมื่อเพิ่มระดับความเข้มแสงไป“15”

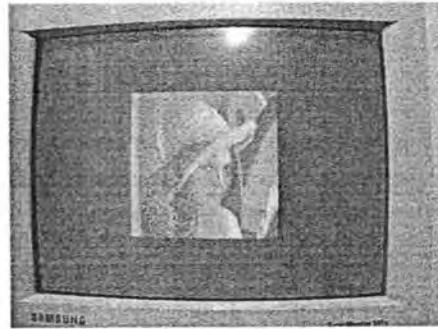
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรรณการวิจัย ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงแหล่งที่มาของเอกสาร



By Sorawat Chivapreecha  
Telecommunication Eng. Dept.  
Faculty of Engineering  
King Mongkut's Institute of Technology Ladkrabang



(ข)

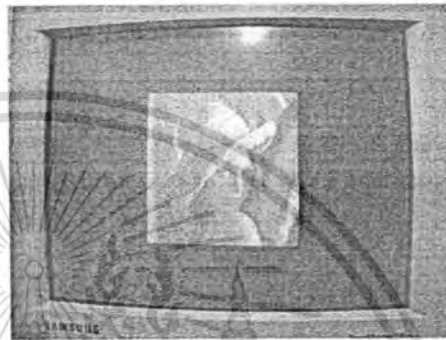


(ค)

เมื่อเพิ่มระดับความเข้มแสงไป“31”



(ด)



(ฉ)

เมื่อเพิ่มระดับความเข้มแสงไป“63”



(ง)

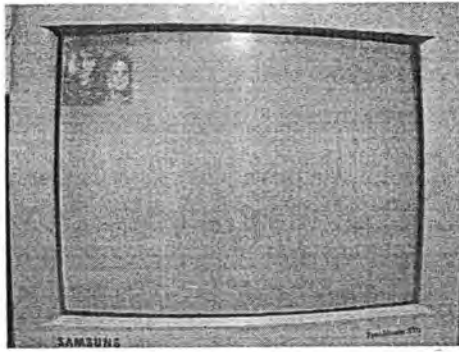


(จ)

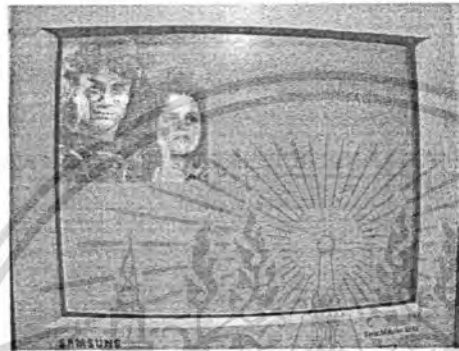
เมื่อเพิ่มระดับความเข้มแสงไป“127”

รูปที่ 4.66 เมื่อทำเพิ่มระดับความเข้มแสงตามค่าที่กำหนด ตัวอย่างที่ 2

4.2.3.2.3 เมื่อผ่านการประมวลผลด้วยวงจรมหาขนาดของภาพและจัดตำแหน่งของภาพ



(ก)



(ข)



(จ)

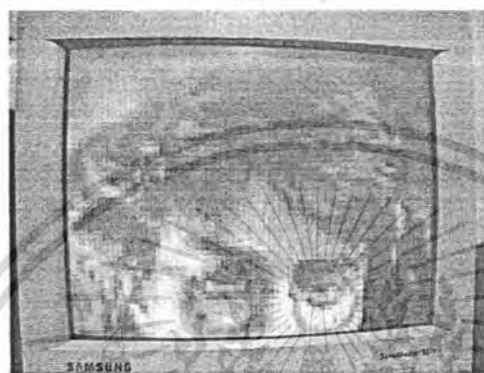


(ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึง



(ค)

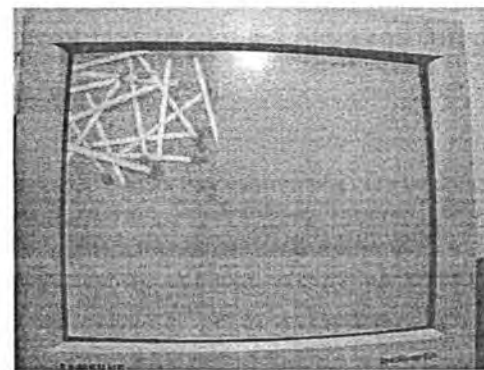


(ง)

รูปที่ 4.67 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 1



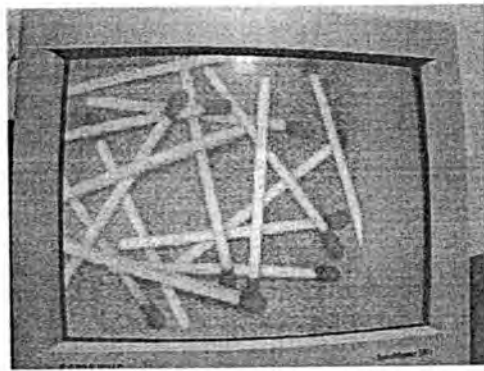
(ก)



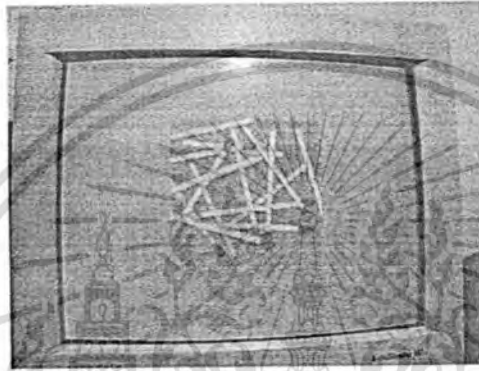
(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึง

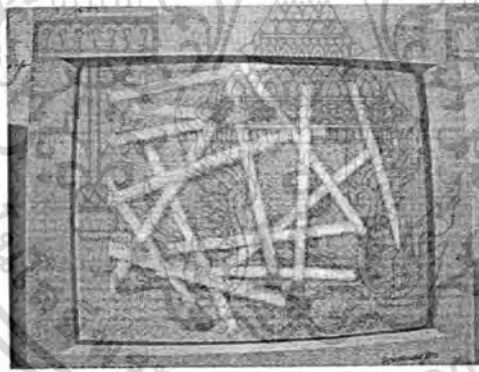
By Sorawat Chivapreecha  
Telecommunication Eng. (Dept.  
Faculty of Engineering  
King Mongkut's Institute of Technology Ladkrabang



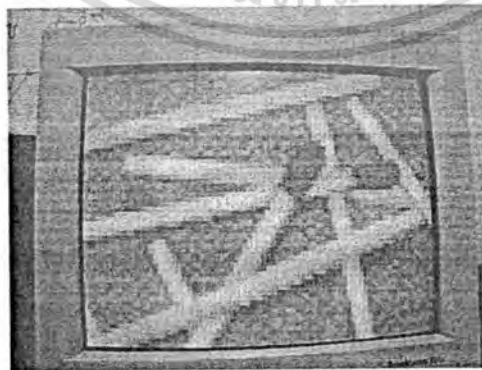
(ข)



(ค)



(ง)



(จ)

รูปที่ 4.68 เมื่อทำการขยายขนาดของภาพและจัดตำแหน่งของภาพ ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

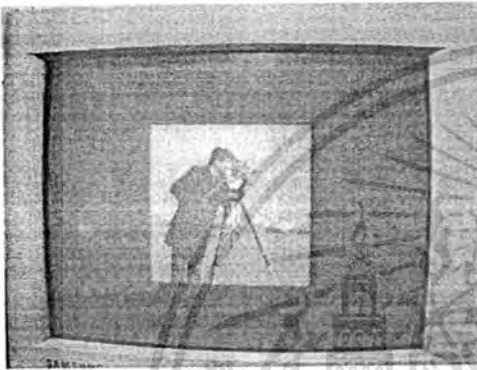
#### 4.2.3.2.4 การประมวลผลภาพโดยการผ่านวงจรกรองพาสกาล



ก (1)



ก (2)



ก (3)



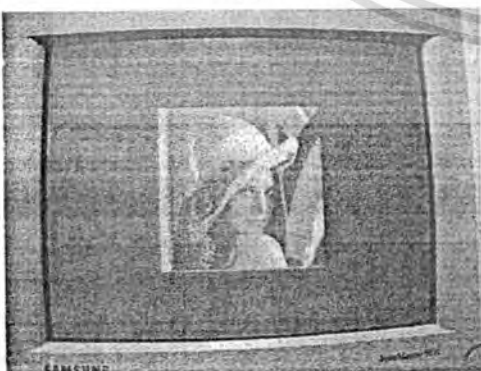
ก (4)



ข (1)



ข (2)

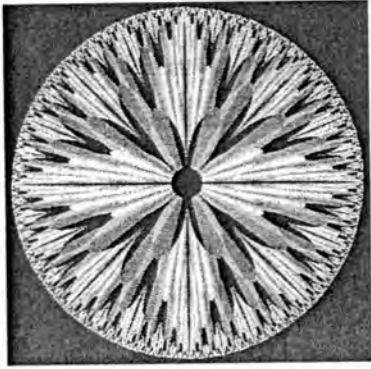


ข (3)

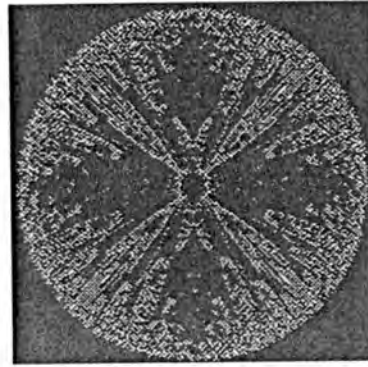


ข (4)

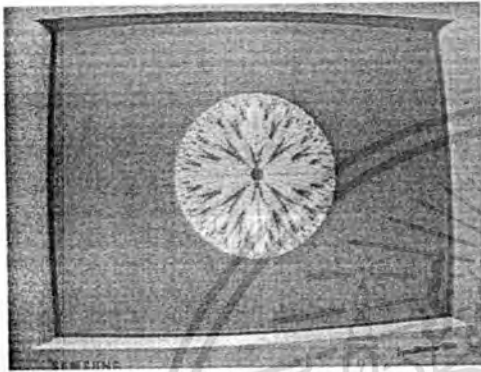
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้า



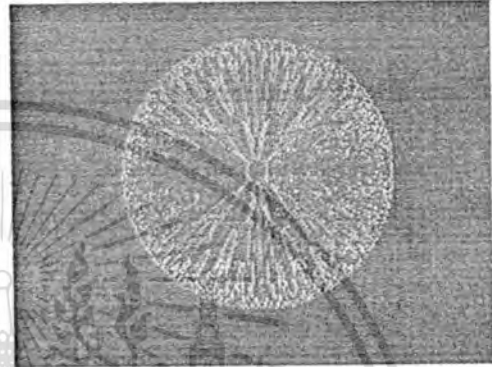
ข (1)



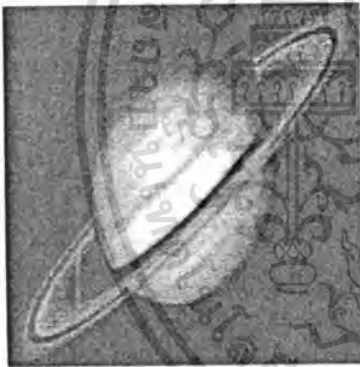
ข (2)



ข (3)



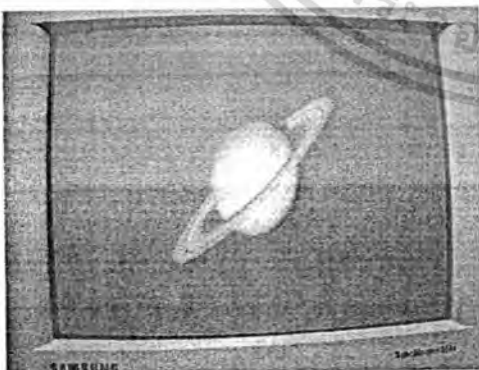
ข (4)



ค (1)



ค (2)



ค (3)



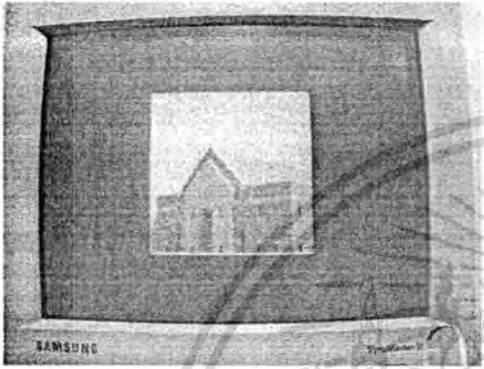
ค (4)



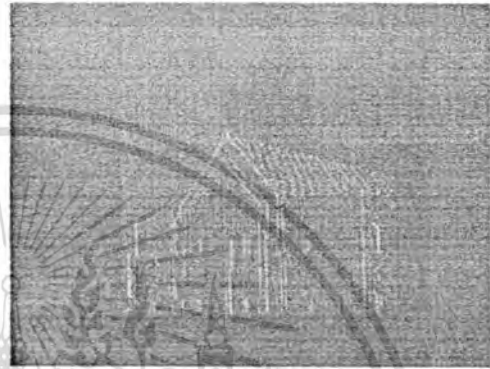
ก (1)



ก (2)



ก (3)



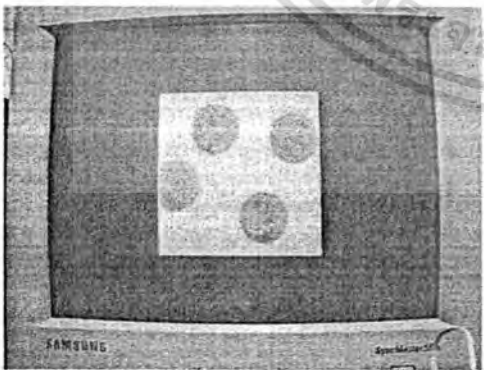
ก (4)



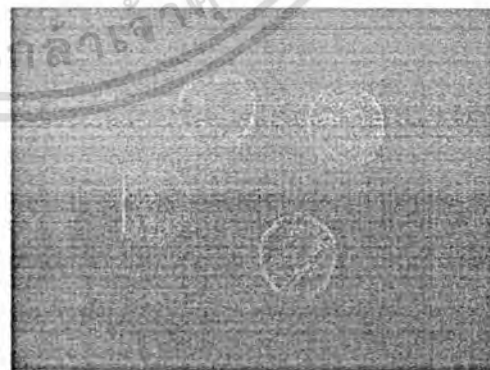
ข (1)



ข (2)



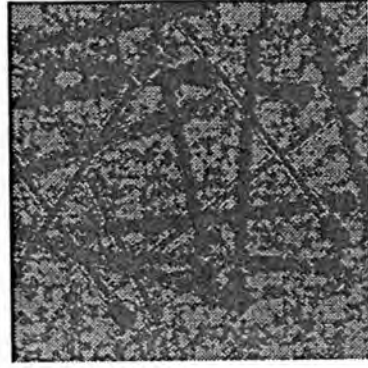
ข (3)



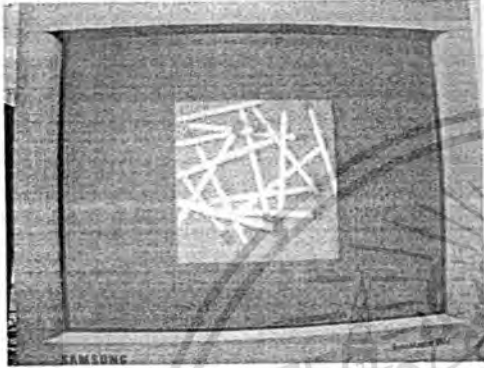
ข (4)



จ (1)



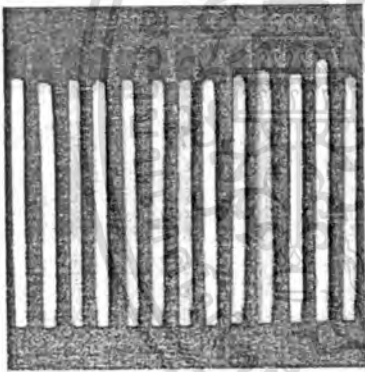
จ (2)



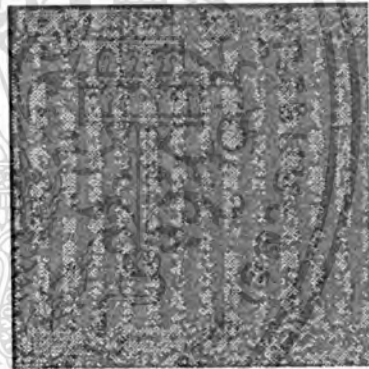
จ (3)



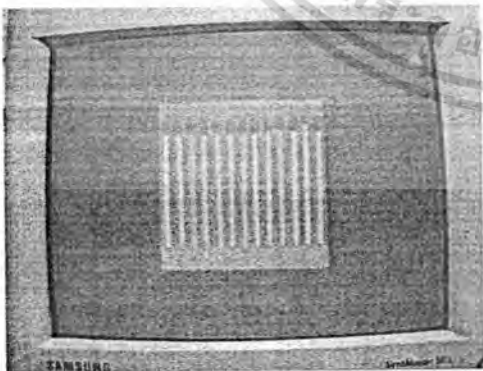
จ (4)



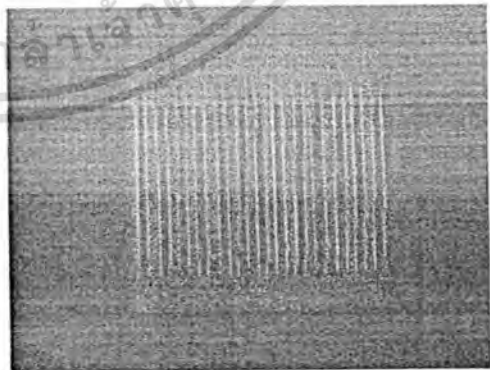
จ (1)



จ (2)



จ (3)



จ (4)



ก (1)



ก (2)



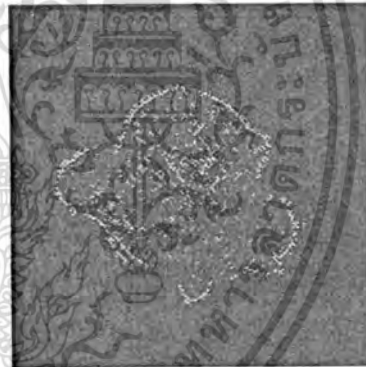
ข (1)



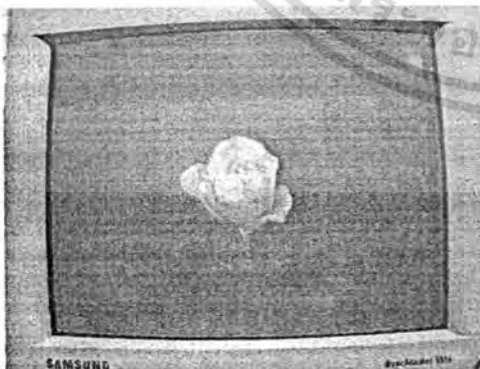
ข (2)



ช (1)



ช (2)



ช (3)

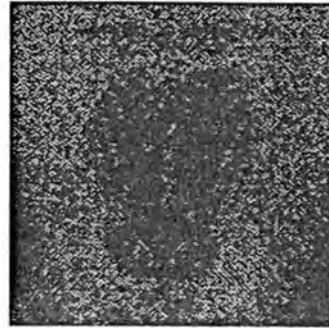


ช (4)

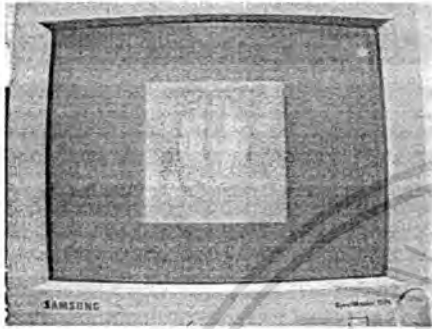
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้า



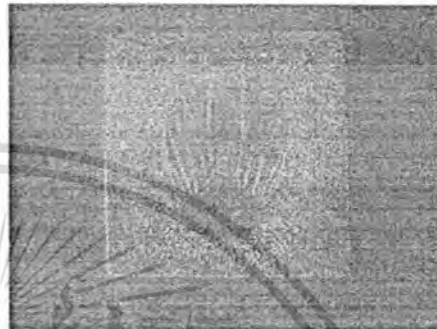
ซ (1)



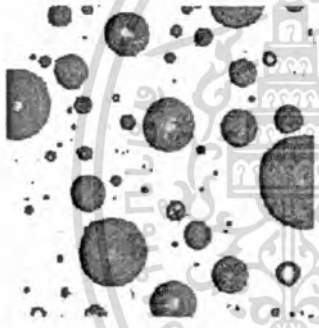
ซ (2)



ซ (3)



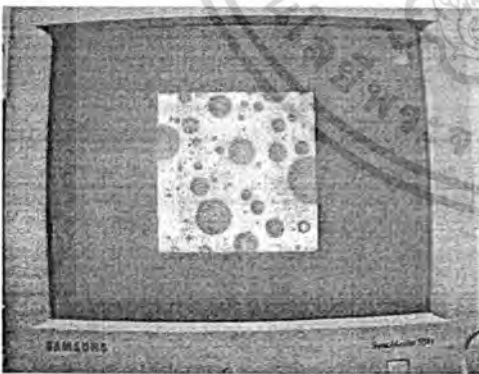
ซ (4)



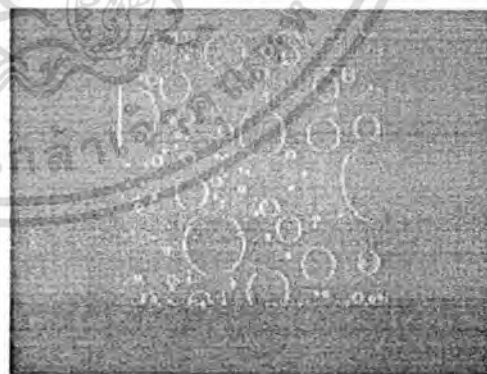
ฉ (1)



ฉ (2)



ฉ (3)



ฉ (4)

รูปที่ 4.69 ภาพเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่ผ่านวงจกรองพาสคาลเรียบร้อยแล้ว

- (1) ภาพต้นฉบับ
- (2) ภาพที่ผ่านวงจรองพาสคาลจากโปรแกรม MATLAB
- (3) ภาพต้นฉบับเมื่อนำไปแสดงผลบนจอวีจีเอ
- (4) ภาพที่ผ่านวงจรองพาสคาลเมื่อนำไปแสดงผลบนจอวีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึง

## บทที่ 5

### บทสรุปและวิจารณ์

โครงการนี้นำเสนอการประมวลผลภาพชนิดระดับสีเทาแบบเวลาจริง โดยการใช้อุปกรณ์ เอฟพีจีเอ ซึ่งได้ทำการศึกษาหลักการการทำงานของหน้าจอ วีจีเอ และทำการออกแบบวงจรที่ใช้ในการควบคุมแล้วแสดงผลบนหน้าจอวีจีเอ จากการทดลองสามารถสร้างสัญญาณการสแกนทางแนวนอนที่มีความถี่ 31.45 kHz และสามารถสร้างสัญญาณการสแกนทางแนวตั้งที่มีความถี่ 60.24 Hz ซึ่งจากการคำนวณตามทฤษฎีสัญญาณการสแกนทางแนวนอนมีความถี่เท่ากับ 31.476 kHz และสัญญาณการสแกนทางแนวตั้งมีความถี่เท่ากับ 60 Hz จะเห็นได้ว่าสัญญาณที่ได้มีความถี่ใกล้เคียงตามทฤษฎี ทั้งนี้ได้ทำการแสดงผลภาพระดับสีเทาแบบภาพนิ่งและแบบภาพเคลื่อนไหว

จากการศึกษาปาสคาลทรานสเฟอร์ม ได้ทำการจำลองการทำงานของวงจรกรองปาสคาลใน 1 มิติ โดยใช้ปาสคาลทรานสเฟอร์ฟังก์ชันในการออกแบบ พบว่ามีคุณสมบัติผลตอบสนองทางความถี่คล้ายกับตัวกรองความถี่สูงผ่าน คือสามารถกรองความถี่สูงออกมาได้ จากการจำลองการทำงานของปาสคาลฟิลเตอร์ใน 2 มิติ พบว่าเมื่อนำภาพระดับสีเทามาประมวลผลโดยใช้วงจรกรองปาสคาล ภาพที่ได้จะมีลักษณะขอบภาพชัดขึ้น เนื่องจากบริเวณที่เป็นขอบภาพมีการเปลี่ยนแปลงค่าความเข้มแสงมาก ๆ หรือมีความถี่สูง บริเวณที่มีความถี่สูงจึงสามารถผ่านวงจรกรองออกมาได้ จึงเห็นเป็นขอบภาพ และจากการทดลองพบว่าทรานสเฟอร์ฟังก์ชันที่ให้ผลดีที่สุดในขนาดเมตริกซ์ 3x3 ซึ่งมีโครงสร้างที่ง่ายต่อการออกแบบให้กับเอฟพีจีเอ จากผลการทดลองทำการรับข้อมูลภาพจากหน่วยความจำภายนอก เข้ามาผ่านกระบวนการประมวลผลด้วยวงจรกรองปาสคาลภายในเอฟพีจีเอ แล้วแสดงผลบนหน้าจอวีจีเอ พบว่าภาพที่แสดงบนหน้าจอมีลักษณะเห็นขอบภาพชัดขึ้น ซึ่งเป็นไปตามผลจากการจำลองการทำงานของวงจรกรองปาสคาลจากโปรแกรม MATLAB

นอกจากนี้แล้วยังมีการออกแบบวงจรประมวลผลภาพในลักษณะต่างๆ ได้แก่ การปรับค่าระดับความเข้มแสงให้มีค่าเพิ่มขึ้นหรือลดลง เพื่อเป็นการปรับความสว่าง หรือมืดให้กับภาพต้นแบบ และยังมีการออกแบบวงจรปรับค่าความเข้มแสงให้มีเพียง 2 ระดับ คือ “00000000” ซึ่งแสดงผลเป็นสีดำ และ “11111111” ซึ่งแสดงผลเป็นสีขาว โดยผู้ใช้งานสามารถป้อนค่าที่ใช้ในการประมวลผลมีค่าตั้งแต่ 0 – 255 ระดับ ทั้งนี้ยังมีการออกแบบวงจรขยายขนาดของภาพต้นแบบและจัดตำแหน่งในการแสดงผลภาพบนหน้าจอได้อีกด้วย

การประยุกต์ใช้งาน สามารถใช้วงจรกรองปาสคาลเป็นตัวกรองความถี่สูงผ่านได้ จากคุณสมบัติของตัวกรองความถี่สูงผ่านที่ได้นี้สามารถนำไปใช้ในการหาขอบภาพ ปรับคุณภาพของภาพเพื่อประยุกต์ในการใช้งานด้านอื่นๆ ต่อไป สำหรับวงจรปรับระดับ

## เอกสารอ้างอิง

- [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] 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.
- [3] 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
- [4] 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
- [5] Wikipedia, the free encyclopedia. "EPROM"  
Available : [http// Wikipedia.com](http://Wikipedia.com)
- [6] HWb, "VGA (15)"  
Available : [http// VGA\\_\(15\)\\_-HwB.htm](http://VGA_(15)_-HwB.htm)
- [7] Wikipedia, the free encyclopedia. "VGA USB 1394 firewire EVC"  
Available : [http//www.wikipedia.com](http://www.wikipedia.com)



ผลงานการวิจัย  
ที่ได้รับการตีพิมพ์เผยแพร่  
ในเอกสารการประชุมวิชาการ  
ระดับนานาชาติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Technical Programme

The 4<sup>th</sup> International Colloquium on  
Signal Processing and its Applications  
(CSPA 2008)

7-9 March 2008

Kuala Lumpur, Malaysia.

### Organizer

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

### Technical co-sponsor

IEEE UiTM SB, Malaysia Section  
IEEE Signal Processing, Malaysia Section

### Editors

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



# Investigation of Frequency Characteristic in Discrete Pascal Transform and Its Applications

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

**Abstract**-This paper presents the investigation of frequency or filtering characteristic that is hidden in the discrete Pascal transform. The Pascal matrix that used for signal transformation by discrete Pascal transform in this paper is divided into 2 types, the first type has hidden high-pass filtering characteristic and another type has hidden low-pass filtering characteristic. This paper will show the method to investigate those characteristics by the transfer function of discrete Pascal transform systems will be formulated and used for analysis the frequency characteristic of such systems. Moreover, the relationship between the dimension of Pascal matrix and frequency characteristic will be shown. Finally, the result from the investigation will be applied for both 1-D and 2-D signal filtering applications.

## I. INTRODUCTION

The discrete Pascal transform (DPT), that was proposed in [1-2] has an attraction for signal processing applications, especially application of the transform in digital image processing. Moreover, many previous papers [3-5] introduced the advantages of the Pascal matrix for designing digital filter.

As mentioned in [1-2], the DPT was used in bump and edge detection. So, it makes us think of the high frequency component in an image. Therefore, we make an assumption about the hidden highpass filtering characteristic in DPT. This paper will show the method to investigate the frequency characteristic that is hidden in DPT. The DPT that uses Pascal matrix as shown in [1], we will call the highpass type DPT, and we also present another one that we will call the lowpass type DPT. All frequency characteristic can be shown in the frequency response of both 1-D and 2-D filtering. Moreover, we found that the operation of direct DPT, as proposed in [1-2] has weakness. From the frequency or filtering characteristic investigation, we can improve the operation method from DPT to discrete Pascal filtering (DPF). The simulation results will show the differences between DPT and DPF output signal both highpass and lowpass 1-D and 2-D signal filtering.

## II. DISCRETE PASCAL TRANSFORM AND BASIS FUNCTION

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

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

where  $P$  is the Pascal transform matrix size  $N \times N$ , and  $x$  is input signal vector,  $X$  is a transformed output signal vector size  $N \times 1$ . In this paper, we divide the type of Pascal matrix into 2 types by basis function of the Pascal transform matrix. The first type, we call highpass type DPT which is the same as the one proposed in [1] and another type, we call lowpass type DPT. We will explain to the reason for using these type names in the next section.

### A. Basis Function of Highpass Type DPT

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

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

### B. Basis Function of Lowpass Type DPT

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

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

where  $N$  is dimension of Pascal transform matrix and

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

which we call binomial coefficient and is related to the Pascal's triangle. The only difference between eq. (2) and eq. (3) is that the term  $(-1)^k$  does not appear in eq. (3). This term will alternate the signs of the columns of highpass type Pascal transformation matrix.

For both eq. (2) and eq. (3),  $P(x, k)$  is an element in Pascal transformation matrix  $P$  as show in eq. (1) depending on the type needed,  $x^{(k)}$  is the row index and  $k^{th}$  is the column index. The function  $x^{(k)}$  is called falling factorial powers and can be shown as follow,

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

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

For example, we can show the 3<sup>rd</sup> order basis function of both highpass type and lowpass type as follows,

In the case of highpass type, there are 4 polynomials, those are:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

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

and we can summarize it to the recurrent formula by

$$P_{k+1}^{(HP)}(x) = -\frac{1}{k+1}(x-k)P_k^{(HP)}(x) \quad (5)$$

In the case of lowpass type, there are also 4 polynomials, those are:

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

So the recurrent formula can be shown as,

$$P_{k+1}^{(LP)}(x) = \frac{1}{k+1}(x-k)P_k^{(LP)}(x) \quad (6)$$

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

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

The basic properties of these P matrices are the followings,

1. The elements of the first column are equal to 1.
2. All matrices are lower triangular.
3. For highpass type  $P^{(HP)}$  matrix: the sum of the elements of each row (except the first row) is equal to zero.  
For lowpass type  $P^{(LP)}$  matrix: the sum of the elements of each row is equal to  $2^x$ .

4. For highpass type  $P^{(HP)}$  matrix: the inverse  $[P^{(HP)}]^{-1}$  matrix is equal to the forward  $P^{(HP)}$  matrix, or  $[P^{(HP)}]^{-1} = P^{(HP)}$ .

For lowpass type  $P^{(LP)}$  matrix: the inverse  $[P^{(LP)}]^{-1}$  matrix is equal to  $(-1)^{x+k}$  multiply to the forward  $P^{(LP)}$  matrix, or  $[P^{(LP)}]^{-1} = (-1)^{x+k} P^{(LP)}$ .

The two-dimensional (2-D) DPT can find the transformed output in the following [1]

$$X = P x P^T \quad (7)$$

where P is the Pascal transform matrix in  $N \times N$  dimension, and its transpose is  $P^T$ , X is a transformed output of an input image matrix x. In [1-2] does not explain about the practical usage, which normally the size of input image must be bigger than the size or dimension of the matrix operator. Only the

same size of input image and Pascal matrix operator are shown with small size of assumed input image. In the real applications, we use normal input image size such as  $256 \times 256$  pixels. So, it is impossible to use the same size for Pascal matrix operator. The block processing is necessary for this case same as when we use the DCT (Discrete Cosine Transform) matrix to transform the input image to the frequency domain. The DCT coefficients can be obtained by using DCT matrix size  $8 \times 8$  for transforming image. Similar to the DCT, we can use the smaller size of Pascal transform matrix to operate with the real input image and the DPT coefficients can be obtained. Therefore, it is not a problem to apply the smaller size of Pascal transform matrix to operate with the bigger size of input image, but the obtained coefficients does not give a satisfactory transformed output image, and we cannot see the edge of an image by this way.

If we focus on the effect that happens on the output image, the direct DPT cannot give the good results and cannot give the results we expected at the edge or high frequency components of an image. We will show the transformed output of the direct DPT on both 1-D signal and 2-D signal or image in section IV.

### III. THE HIDDEN FREQUENCY CHARACTERISTIC IN DPT AND ITS DISCRETE PASCAL FILTER

Since the DPT has the Pascal transform matrix performing as the key operator, so if we consider such matrix as the system and the input of this system would be input vector or input matrix x depending on whether the case is 1-D or 2-D signal, and the output of this system would be the transformed output X.

Therefore, the system function or transform function of this system must be formulated for investigating the characteristic, especially the frequency or filtering characteristic of both the highpass type and lowpass type Pascal transform matrix and both 1-D and 2-D signal case.

#### A. Frequency Response of Highpass Type DPT

Consider the 1-D DPT with  $N = 3$ , from eq. (1) we would get

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \quad (8)$$

In order to formulate the transfer function, we are necessary to make the relationship for each input data by giving

$$\begin{aligned}
x_2 &= x(n) && \text{present data} \\
x_1 &= x(n-1) && \text{1 sample delayed data} \\
x_0 &= x(n-2) && \text{2 samples delay data}
\end{aligned}$$

Also, from eq. (8) we would get

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x(n-2) \\ x(n-1) \\ x(n) \end{bmatrix} \quad (9)$$

Thus, we can summarize

$$X_0 = x(n-2) \quad (10a)$$

$$X_1 = -x(n-1) + x(n-2) \quad (10b)$$

$$X_2 = x(n) - 2x(n-1) + x(n-2) \quad (10c)$$

and we can show a system diagram to represent the highpass type DPT as below

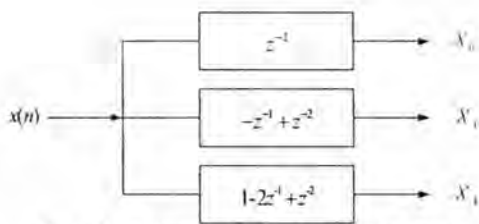


Fig. 1 Block diagram of highpass type DPT system

From Fig.1. we can see that this system has 3 outputs. But since we assume that input  $x_2$  is the present input data  $x(n)$ , so the output  $X_2$  should be the present output  $y(n)$ . Then from eq. (10c) we would get

$$y(n) = x(n) - 2x(n-1) + x(n-2) \quad (11)$$

Take the z-transform, the transfer function of highpass type DPT can be shown as

$$H^{(HP)}(z) = 1 - 2z^{-1} + z^{-2} \quad (12)$$

We can notice that the transfer function coefficients comes from the last row of highpass type Pascal transform matrix in eq. (8).

From the obtained transfer function as shown in eq. (12), we can investigate the frequency characteristic as shown in Fig. 2.

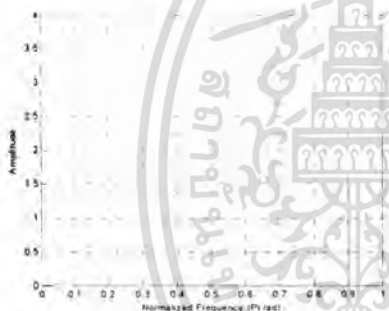


Fig. 2 Amplitude response of highpass type DPT at  $N = 3$

So, we can observe that the frequency characteristic of such Pascal transform matrix is highpass filter and this is the reason why we call it highpass type DPT. The order of transfer function is equal to the dimension of Pascal transform matrix minus 1. Therefore, from eq. (12), we will call  $2^{nd}$  order highpass type discrete Pascal filter (DPF) instead of DPT because the process of operating signal has changed from matrix multiplication as in eq. (1) and eq. (8) to the convolution between filter coefficients and input signal as in eq. (12). For the higher dimension  $N$  or higher order DPF can be considered in the same manner.

Fig. 3 shows the amplitude response of highpass type DPF by showing the order of filter from  $2^{nd}$  -  $99^{th}$  order. We can notice that this type of filter has something different from ordinary filter. For ordinary filter, when the order is increased, the slope of the filter will be also increased with the same cut-off frequency.

$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & & x_{0,0} - x_{0,1} \\ [x_{0,0} - x_{1,0}] & [x_{0,0} - x_{1,0}] - [x_{0,1} - x_{1,1}] & x_{0,0} - 2x_{0,1} + x_{0,2} \\ [x_{0,0} - 2x_{1,0} + x_{2,0}] & [x_{0,0} - 2x_{1,0} + x_{2,0}] - [x_{0,1} - 2x_{1,1} + x_{2,1}] & [x_{0,0} - 2x_{1,0} + x_{2,0}] - 2[x_{0,1} - 2x_{1,1} + x_{2,1}] + [x_{0,2} - 2x_{1,2} + x_{2,2}] \end{bmatrix} \quad (14)$$

But this type of filter, when the order is increased, the slope of adjacent order changes slightly. The distinct change is cut-off frequency. In the case of highpass type DPF, when higher order the cut-off frequency is increased. If focusing on bandwidth of passband, when higher order the passband will be narrower.

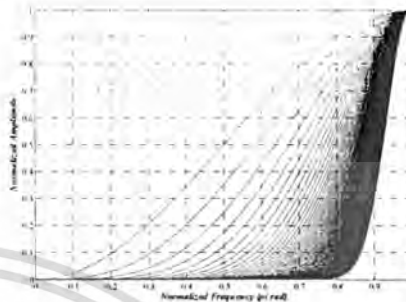


Fig. 3 Amplitude response of highpass type DPF from  $2^{nd}$  to  $99^{th}$  order

Consider the 2-D DPT with  $N = 3$ , from eq. (7) we can express.

$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{0,0} - x_{1,0} & x_{0,1} - x_{1,1} & x_{0,2} - x_{1,2} \\ x_{0,0} - 2x_{1,0} + x_{2,0} & x_{0,1} - 2x_{1,1} + x_{2,1} & x_{0,2} - 2x_{1,2} + x_{2,2} \end{bmatrix} \quad (13)$$

The first matrix multiplication is comparable to the column operation of 3 vectors of 1-D signal. After that multiply again to the transpose of its Pascal matrix, which is comparable to the row operation of 3 vectors of 1-D that is obtained from column operation. Therefore, the final result is 2-D operation, or 2-D highpass type DPT in eq. (14). Later, we make the relationship of each input data in the input matrix as

$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x(m-2, n-2) & x(m-2, n-1) & x(m-2, n) \\ x(m-1, n-2) & x(m-1, n-1) & x(m-1, n) \\ x(m, n-2) & x(m, n-1) & x(m, n) \end{bmatrix} \quad (15)$$

Similar to 1-D, since we consider  $x_{2,2}$  as the present input  $x(m, n)$ . So,  $X_{2,2}$  should be the present output  $y(m, n)$ . From eq. (14) we would get

$$\begin{aligned} \therefore y(m, n) &= x(m, n) - 2x(m, n-1) + x(m, n-2) \\ &\quad - 2x(m-1, n) + 4x(m-1, n-1) + 2x(m-1, n-2) \\ &\quad + x(m-2, n) - 2x(m-2, n-1) + x(m-2, n-2) \end{aligned} \quad (16)$$

Form eq. (16), we can show the convolution equation that is used for computing the output  $y(m, n)$  as follows,

$$y(m, n) = \sum_{k=0}^2 \sum_{l=0}^2 h(k, l)x(m-k, n-l) \quad (17)$$

where  $h(k,l)$  is the impulse response of 2-D digital filter. Also, we can summarize that the impulse response of the highpass type 2-D DPT with  $N = 3$  in this example is  $h^{(HP)}(k,l)$ , and call it the highpass type Pascal convolution mask as shown in Fig. 4.

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

Fig. 4 Highpass type Pascal convolution mask size  $3 \times 3$

In Fig. 5, it shows the amplitude response of highpass Pascal convolution mask



Fig. 5 Amplitude response of highpass type Pascal convolution mask size  $3 \times 3$

At this point, we can see that the operation of Pascal transform matrix is changed from multiplication to convolution in DPF (Discrete Pascal Filter), as from eq. (8) to eq. (11) in case of 1-D processing, and from eq. (13) to eq. (16) in case of 2-D processing.

#### B. Frequency Response of Lowpass Type DPT

Similar to previous sub-section, only change highpass type Pascal transform matrix  $P^{(HP)}$  to lowpass type Pascal transform matrix  $P^{(LP)}$  both in the case of 1-D and 2-D.

The lowpass type 1-D DPT with  $N = 3$ . We can summarize the  $2^{nd}$  order transfer function as

$$H^{(LP)}(z) = 1 + 2z^{-1} + z^{-2} \quad (18)$$

and can be shown the frequency response as in Fig. 6,

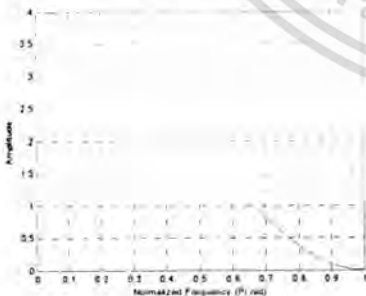


Fig. 6 Amplitude response of lowpass type DPT at  $N = 3$

Fig. 7 shows the amplitude response of lowpass type DPF from  $2^{nd}$  order to  $99^{th}$  order. The results are as same as in the case of highpass type. that is higher order will give the narrower passband.

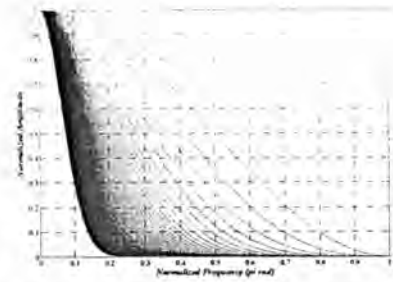


Fig. 7 Amplitude response of lowpass type DPF from  $2^{nd}$  to  $99^{th}$  order

The impulse response  $h^{(LP)}(k,l)$  of the lowpass type 2-D DPF can be summarized as shown in lowpass type Pascal convolution mask size  $3 \times 3$  in Fig. 8.

1	2	1
2	4	2
1	2	1

Fig. 8 Lowpass type Pascal convolution mask size  $3 \times 3$

The value of  $(1/16)$  come from sum of all elements in the mask which are used to scale the coefficients in the mask and is necessary for 2-D LPF (Lowpass Filter). The amplitude response can be shown as in Fig 9.

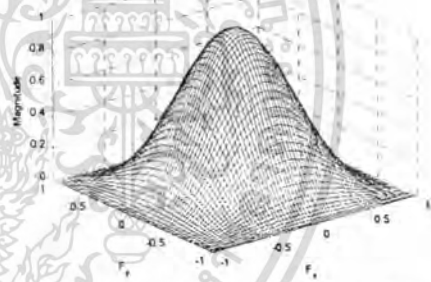


Fig. 9 Amplitude response of lowpass type Pascal convolution mask size  $3 \times 3$

#### IV. APPLICATION OF DISCRETE PASCAL FILTERING

From previous section, we found that the hidden frequency characteristic in DPT can help us improve the operation of DPT from multiplication to convolution and we call it DPF. In this section, we will show how DPF filter the signal of both 1-D and 2-D, and both highpass and lowpass filtering and compare their results with the directly DPT proposed in [1-2] in order to show the weakness of such method.

##### A. Highpass Filtering Results

The first simulation, we will apply the single sinusoidal frequency 100 Hz to highpass type 1-D DPT and the simulation results can be shown as in Fig. 10.

The second simulation, the composite signal that is created from low frequency sinusoidal signal 100 Hz and high frequency sinusoidal signal 1500 Hz, will be applied to the  $4^{th}$  order (if we

compare to DPT, it would be 5-point DPT) highpass type 1-D DPF and 5-point highpass type 1-D DPT to make a comparison.

From Fig. 11, the highpass type 1-D DPF output can ensure that the highpass type 1-D DPF can be applied to highpass filter by the convolution method, but the highpass type 1-D DPT, which uses matrix multiplication, cannot give the good or expected results to the transformed output. So, it is classed as weakness of direct DPT. The last simulation for this section, highpass type 2-D DPF and highpass type 2-D DPT will be shown. The input image size  $256 \times 256$  pixels shown in Fig. 12 and highpass type Pascal convolution mask size  $3 \times 3$ ,  $5 \times 5$  and  $8 \times 8$  will be used for 2-D DPF. The highpass type 2-D DPT will use highpass type Pascal transform matrix with the same size as highpass type Pascal convolution mask to make a comparison with 2-D DPF. From Fig. 13, the highpass type 2-D DPF with the mask size  $3 \times 3$  gives the best result. Higher mask size gives the poor result because the highpass type Pascal convolution mask does not has scaling coefficient. Also, when using of higher mask size, it will increase the high gain and result in the increase of noise in the output image. Therefore, the highpass type 2-D DPF with the mask size  $3 \times 3$  can detect the edge of an image better than the 2-D DPT's result.

### B. Lowpass Filtering Results

The first simulation, we apply the sinusoidal frequency 100 Hz to lowpass type 1-D DPT and the simulation results are as shown in Fig. 14. The second simulation, the composite signal that created from low frequency sinusoidal signal 100 Hz plus high frequency sinusoidal signal 4000 Hz will be applied to the 4<sup>th</sup> order lowpass type 1-D DPF and 5-point lowpass type 1-D DPT. From Fig. 15, the lowpass type 1-D DPF output can ensure that the lowpass type 1-D DPF can be applied to lowpass filter, but the lowpass type 1-D DPT cannot give the expected results.

The last simulation, lowpass type 2-D DPF and lowpass type 2-D DPT will be compared in the same manner as in highpass type case. From Fig. 16, the lowpass type 2-D DPF can smoothen an image, or lowpass filter an image, better than the lowpass type 2-D DPT's result.

## V. CONCLUSION

The investigation of frequency characteristic or filtering characteristic that is hidden in the DPT was proposed, both highpass type and lowpass type DPT and both 1-D and 2-D signal are considered. The results from frequency characteristic investigation can be used to upgrade the ordinary DPT to DPF. The DPF can give the good results in sense of signal filtering and can clarify the weakness of ordinary or direct DPT. Consequently, both 1-D and 2-D and both highpass type and lowpass type DPF can be applied to signal filtering applications.

### ACKNOWLEDGMENT

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

## REFERENCES

- [1] M. F. Aburdene and T.J. Goodman, "The Discrete Pascal Transform and Its Applications," *IEEE Signal Processing Letters*, vol. 12, No. 7, pp. 493-495, July 2005.
- [2] T.J. Goodman and M. F. Aburdene, "A hardware implementation of the discrete Pascal transform for image processing," *Proc. of SPIE-IT&T Electronic Imaging*, SPIE vol. 6064, pp.60640 H-1 – 6064 H-8, 2006
- [3] B. Psenicka, F. Garcia-Ugalde, and A. Hewera-Camacho, "The bilinear Z. Transform by Pascal Matrix and Its Application in the Design of Digital Filters," *IEEE Signal Processing Letters*, vol. 07, No. 11, pp. 368-357 Nov 2002.
- [4] S. Chivapreecha, S. Sryapong, S. Junnapiya, and K. Dejhan, "Bilinear s-z with Frequency Transformation Using Pascal Matrix Operation," *Proc. IEEE International Symposium on Communications and Information Technology (ISCIT 2005)*, pp. 735-742, Oct. 2005.
- [5] 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)*, CHing Mai, Thailand, July 10-13, 2006.

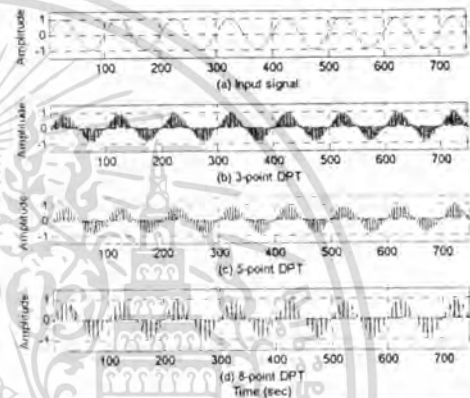


Fig. 10 Input signal and transformed output of highpass type 1-D DPT

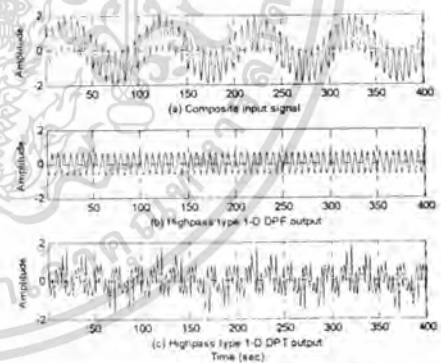


Fig. 11 The comparison result of highpass type 1-D DPF and 1-D DPT



Fig. 12 Original image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

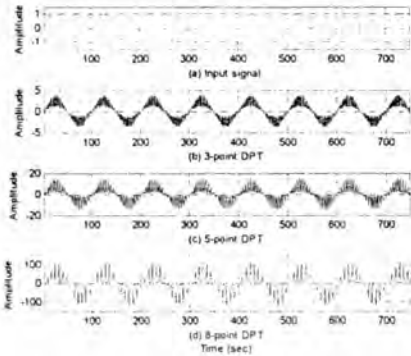


Fig. 14 Input signal and transformed output of lowpass type 1-D DPT

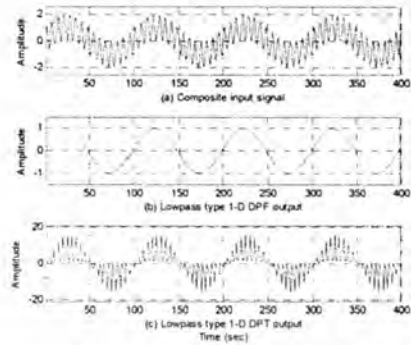


Fig. 15 The comparison result of lowpass type 1-D DPF and 1-D DPT

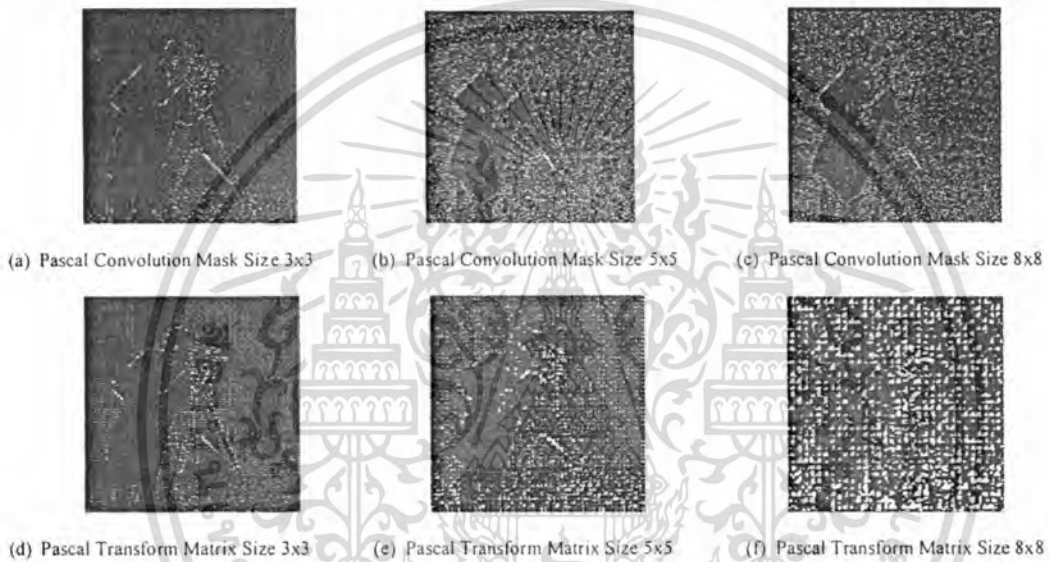


Fig. 13 The comparison result of highpass type 2-D DPF and 2-D DPT

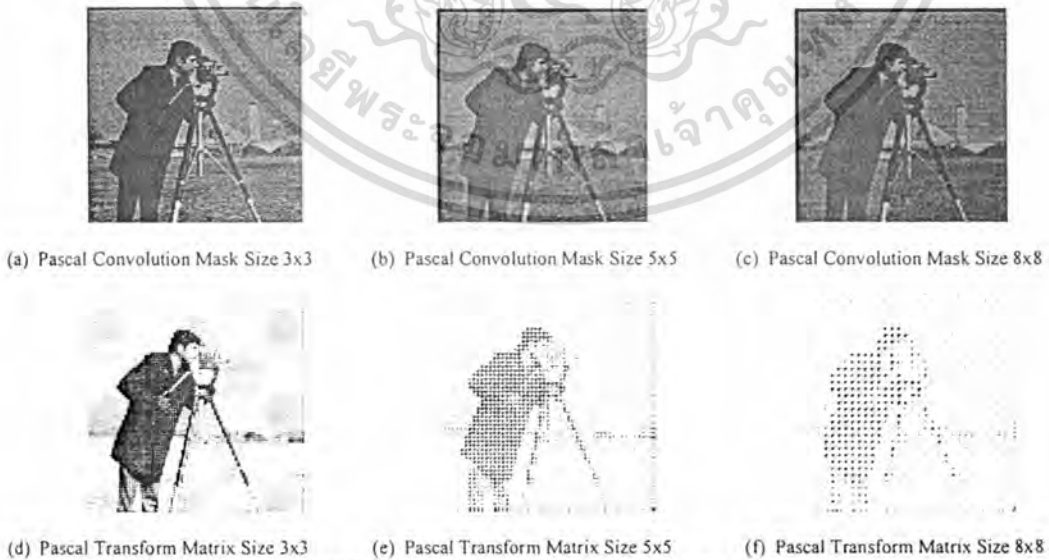


Fig. 16 The comparison result of lowpass type 2-D DPF and 2-D DPT

# Efficient Hardware Realization for Discrete Pascal Transform Using Matrix Factorization

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

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

## I. INTRODUCTION

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

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

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

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

## II. BASIS FUNCTION OF DISCRETE PASCAL TRANSFORM

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

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

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

### A. Basis Function of Highpass Type DPT

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

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

### B. Basis Function of Lowpass Type DPT

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

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

where  $N$  is dimension of Pascal transform matrix and

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

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

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

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

For example, we can show the 3<sup>rd</sup> order basis function as follows,

In the case of highpass type has 4 polynomials are:

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

and we can summarize to the recurrence formula by

$$P_{k+1}^{(HP)}(x) = -\frac{1}{k+1}(x-k)P_k^{(HP)}(x) \quad (5)$$

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

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

So, the recurrence formula can be shown as,

$$P_{k+1}^{(LP)}(x) = \frac{1}{k+1}(x-k)P_k^{(LP)}(x) \quad (6)$$

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

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

The basic properties of these P matrices are the following,

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

For lowpass type  $P^{(LP)}$  matrix: the inverse  $[P^{(LP)}]^{-1}$  matrix is equal to  $(-1)^{x+k}$  multiply to the forward  $P^{(LP)}$  matrix, or  $[P^{(LP)}]^{-1} = (-1)^{x+k} P^{(LP)}$

### III. EFFICIENT HARDWARE REALIZATION OF DPT AND ITS BUTTERFLY UNIT

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

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

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

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

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

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

Therefore,

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

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

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

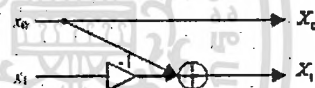


Fig. 1 A Butterfly unit of highpass type DPT

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

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

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

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

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

Therefore,

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

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

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

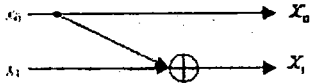


Fig. 2 A Butterfly unit of lowpass type DPT

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

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

### Lowpass type DPT

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

**First Step** for matrix elimination:

$$E_2^{(LP)} P_4^{(LP)} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & & \\ & 0 & P_3^{(LP)} & \end{bmatrix}$$

**Second Step:**

$$\begin{bmatrix} 1 & 1 & & \\ 0 & E_3^{(LP)} & & \\ 0 & & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & & \\ 1 & 0 & & \\ 0 & 1 & & \\ 0 & 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & P_2^{(LP)} & \end{bmatrix}$$

**Third Step:**

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

So that,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_3^{(LP)} \end{bmatrix} [E_2^{(LP)}] [P_4^{(LP)}] = [I] \quad (9)$$

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

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

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

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

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_2^{(LP)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & E_3^{(LP)} \end{bmatrix} [E_4^{(LP)}] = (-1)^{x+x} P_4^{(LP)} \quad (11)$$

we can show that

$$P_4^{(LP)} = (-1)^{x+x} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & E_3^{(LP)} \end{bmatrix} \times (-1)^{x+x} \begin{bmatrix} 1 & 0 \\ 0 & E_2^{(LP)} \end{bmatrix} \times (-1)^{x+x} [E_4^{(LP)}] \quad (12)$$

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

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

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

### Highpass type DPT

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

**First Step** for matrix elimination:

$$E_2^{(HP)} P_4^{(HP)} = \begin{bmatrix} 1 & & & \\ 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & & \\ & 0 & P_3^{(HP)} & \end{bmatrix}$$

**Second Step:**

$$\begin{bmatrix} 1 & 1 & & \\ 0 & E_3^{(HP)} & & \\ 0 & & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & & \\ 1 & 0 & & \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & P_2^{(HP)} \end{bmatrix}$$

**Third Step:**

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

So that,

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

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

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

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

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

Also, we can show that

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

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

$$P_4^{(HP)} = \begin{bmatrix} 1 & -1 & & \\ 1 & -2 & 1 & \\ 1 & -3 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix} \quad (16)$$

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

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

### B. DPT Flow Graph

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

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

$$= [q_{xN-1}] [q_{xN-2}] \dots [q_{x0}]$$

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

Therefore, from eq. (13)

$$P_4^{(HP)} = [q_{x3}^{(HP)}]_1 [q_{x2}^{(HP)}]_2 [q_{x1}^{(HP)}]_1$$

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

and from eq. (16)

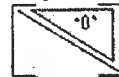
$$P_4^{(HP)} = [q_{x3}^{(HP)}]_1 [q_{x2}^{(HP)}]_2 [q_{x1}^{(HP)}]_1$$

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

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

For Lowpass type:

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



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

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

4. Otherwise are '0'.

For Highpass type:

1. Consider in the main diagonal,

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

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

or consider in this manner

for stage  $l = N-1$

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

for stage  $l = 2$

$Diag([q_{xk}]_2) = [1 \ 1 \ -1 \ -1 \ \dots \ -1]$ ; '1' has 2 elements  
others are '-1'

for stage  $l = 1$

$Diag([q_{xk}]_1) = [1 \ -1 \ -1 \ -1 \ \dots \ -1]$ ; '1' has 1 element  
others are '-1'

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

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

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

4. Otherwise are '0'

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

The lowpass type 4-point DPT can be computed by

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

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

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

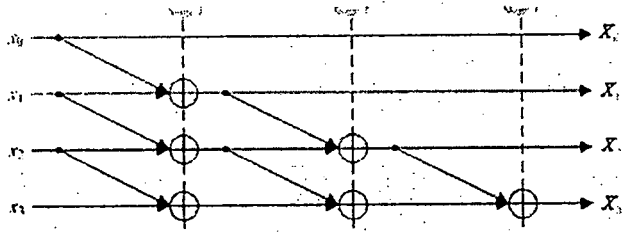


Fig. 3 The lowpass type 4-point DPT flow graph

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

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

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

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

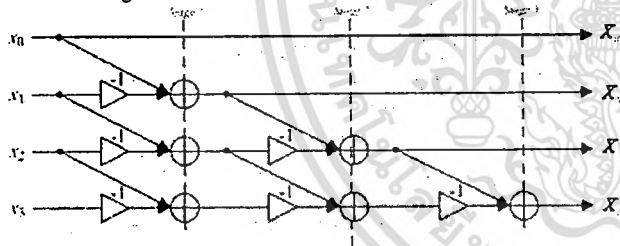


Fig. 4 The highpass type 4-point DPT flow graph

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

### C. Computational Complexity of DPT

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

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

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

the number of multiplications and number of additions would be

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

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

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

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

TABLE I  
COMPUTATIONAL COMPLEXITY OF  $N$ -POINT DPT

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

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

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

$$X = PxP^T \quad (22)$$

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

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

$$X = [P^{(111)}][x][P^{(111)}]^T = \begin{bmatrix} 1 & & \\ 1 & -1 & \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} \\ x_{1,0} & x_{1,1} & x_{1,2} \\ x_{2,0} & x_{2,1} & x_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ & & 1 \end{bmatrix} \quad (23)$$

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

This matrix is a result of column operation for each input column vector, and assume to be  $[x_{i,j}]$

$$\times \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ & & 1 \end{bmatrix}$$

Thus,

$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} y_{0,0} & y_{0,1} & y_{0,2} \\ y_{1,0} & y_{1,1} & y_{1,2} \\ y_{2,0} & y_{2,1} & y_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ & & 1 \end{bmatrix} \quad (24)$$

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

This matrix is a result of row operation for each input row vector which is passed from column operation. Therefore, total results mean 2-D operation (2-D DPT)

Finally,

$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & & & & & \\ [x_{0,0} - x_{1,0}] & & & & & \\ [x_{0,0} - 2x_{1,0} + x_{2,0}] & & & & & \\ & x_{0,1} - x_{1,1} & & & & \\ & [x_{0,1} - 2x_{1,1} + x_{2,1}] & & & & \\ & & x_{0,2} - x_{1,2} & & & \\ & & [x_{0,2} - 2x_{1,2} + x_{2,2}] & & & \\ & & & x_{0,2} - 2x_{1,2} + x_{2,2} & & \\ & & & & x_{0,2} - 2x_{1,2} + x_{2,2} & \\ & & & & & x_{0,2} - 2x_{1,2} + x_{2,2} \end{bmatrix} \quad (25)$$

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

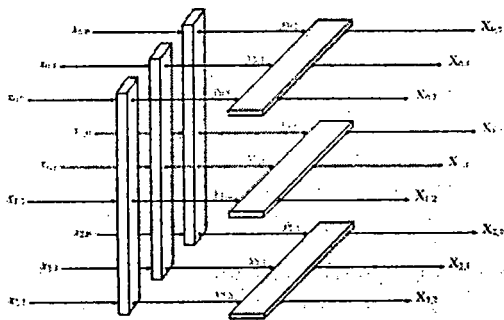
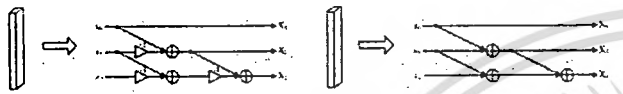


Fig. 5 The 2-D DPT flow graph

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



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

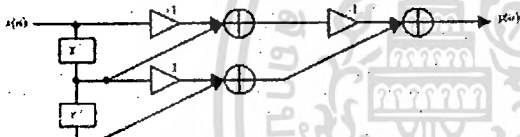


Fig. 6 The 2<sup>nd</sup> order highpass type 1-D DPF



Fig. 7 The 2<sup>nd</sup> order lowpass type 1-D DPF

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

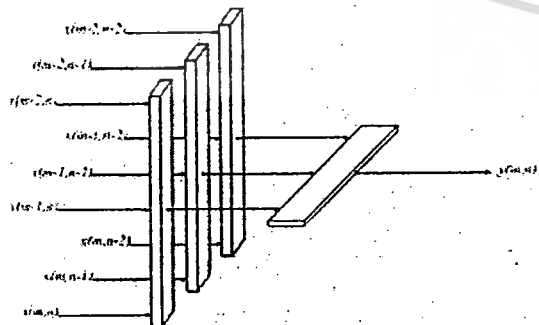
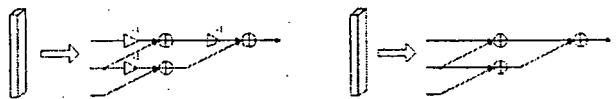


Fig. 8 The 2-D DPF with Convolution Mask Size 3x3

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



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

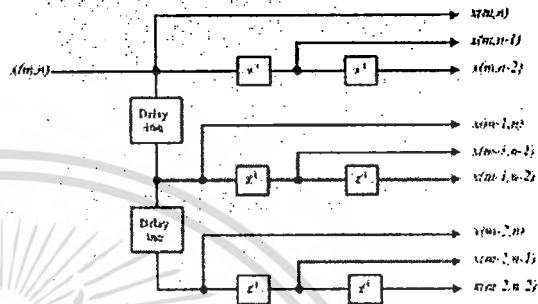


Fig. 9 The image input arrangement circuit

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

## V. CONCLUSIONS

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

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] M. F. Aburdene and T. J. Goodman, "The Discrete Pascal Transform and Its Applications," *IEEE Signal Processing Letters*, vol. 12, No. 7, pp. 493-495, July 2005.
- [2] T. J. Goodman and M. F. Aburdene, "A hardware implementation of the discrete Pascal transform for image processing," *Proc. of SPIE-IT&T Electronic Imaging*, SPIE vol. 6064, pp.60640 H-1 – 6064 H-8, 2006.
- [3] S. Chivapreecha, U. Nithirochananont, and K. Dejhan, "Invertigation of frequency characteristic in discrete Pascal transform and Its applications," *Submitted to The 4<sup>th</sup> International Colloquium on Signal Processing and Its Application (CSPA 2008)*, Kuala Lumpur, Malaysia, 2008.
- [4] A. N. Skodras, "Fast Discrete Pascal Transform," *Electronics Letters*, Vol. 42, No. 23, November 2006.
- [5] A. Edelman and G. Strang, "Pascal Matrices," *Am. Math Mon.*, pp. 189-197., 2004.