

รายงานผลการวิจัยฉบับสมบูรณ์

โครงการวิจัยที่ได้รับทุนสนับสนุนจากเงินรายได้คณะฯ ปี 2551

การออกแบบและสร้างวงจรกรองสัญญาณแบบปรับค่าความหน่วงเป็น
เศษส่วนโดยสมการพหุนามปาสคาลและการประยุกต์ใช้งาน
(Design and Implementation of Variable Fractional Delay Digital Filter
by Pascal Polynomial and Applications)

โดย

RCH

TK

7872

F5

ด147ก

ดร. ศรวัฒน์ ชิวปรีชา : หัวหน้าโครงการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๖. 12160684

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

โลกของเรานี้ในปัจจุบันนี้อยู่ในยุคโลกาภิวัตน์ ซึ่งจำเป็นที่จะต้องมีการใช้สื่อเพื่อการโฆษณาหรือประชาสัมพันธ์บางสิ่งบางอย่างแก่สังคมภายนอก โดยส่วนประกอบหลักในการประชาสัมพันธ์ยุคปัจจุบันนั้นก็คือ ภาพ ซึ่งส่วนใหญ่ภาพที่ถูกนำมาใช้เพื่อการประชาสัมพันธ์นั้นเป็นภาพดิจิทัล ซึ่งหากภาพต้นฉบับมีขนาดเล็ก แต่ต้องใช้ในการประชาสัมพันธ์บนบอร์ดขนาดใหญ่ เมื่อทำการพิมพ์ภาพ ภาพที่ได้นั้นอาจเบลอ, แดก และอาจส่งผลให้ข่าวสารนั้นขาดประสิทธิภาพในการสื่อสารเพื่อให้ผู้บริโภคเข้าใจ จึงต้องมีวิธีการในการขยายภาพ ซึ่งจะคงความชัดของภาพได้ ส่งผลให้ภาพนั้นมีความชัดเจนที่เหมาะสมกับการพิมพ์ขยายเพื่อการประชาสัมพันธ์

โครงการนี้ได้แบ่งการทำงานออกเป็น 2 ส่วน ได้แก่

1. การจำลองการทำงานบนซอฟต์แวร์ โดยใช้โปรแกรมแม่แบบในการจำลองการทำงาน ซึ่งกระบวนการในการอินเทอร์เน็ตไปเลทส์จะนำมาทำการจำลองการทำงาน มีดังต่อไปนี้

- 1.1 รีพีเรชั่นพิกเซล
- 1.2 บีลีเนียนอินเทอร์เน็ตไปเลทส์
- 1.3 ฟาสฟูเรียร์ทรานส์ฟอร์มเบสอินเทอร์เน็ตไปเลทส์
- 1.4 ดิสครีตปาสคาลทรานส์ฟอร์มเบสอินเทอร์เน็ตไปเลทส์

ในกระบวนการในการจำลองการทำงานในโปรแกรมแม่แบบ ในโครงการนี้ได้ทำการจำลองการทำงานของกระบวนการอินเทอร์เน็ตไปเลทส์ โดยใช้ภาพต้นแบบจากภาพพื้นฐานที่ได้เตรียมไว้ก่อนการประมวลผล

2. เป็นการนำกระบวนการในการอินเทอร์เน็ตไปเลทส์จะนำมาทำงานบนฮาร์ดแวร์ ซึ่งในส่วนนี้จะใช้เอฟพีจีเอในการประมวลผล โดยกระบวนการในการอินเทอร์เน็ตไปเลทส์ที่ได้นำมาทำการจำลองการทำงานทางฮาร์ดแวร์ มีดังต่อไปนี้

- 2.1 รีพีเรชั่นพิกเซล
 - 2.1.1 การรีพีเรชั่นพิกเซลโดยไม่ใช้ RAM
 - 2.1.2 การรีพีเรชั่นพิกเซลโดยใช้ RAM
- 2.2 บีลีเนียนอินเทอร์เน็ตไปเลทส์

ในกระบวนการทำงานทางฮาร์ดแวร์นั้น เราจะทำการเก็บภาพต้นแบบไว้ในอีอีพรอมก่อนที่จะนำภาพนั้นมาทำการประมวลผล และได้ทำการทดลองใช้ไมโครคอนโทรลเลอร์ในการตรวจจับภาพเพื่อใช้เป็นภาพต้นแบบสำหรับการอินเทอร์เน็ตไปเลทส์ด้วย โดยรูปแบบของโครงสร้างของฮาร์ดแวร์ที่ใช้ในการจำลองการทำงานในกระบวนการอินเทอร์เน็ตไปเลทส์เป็นดังนี้

RCH

TK

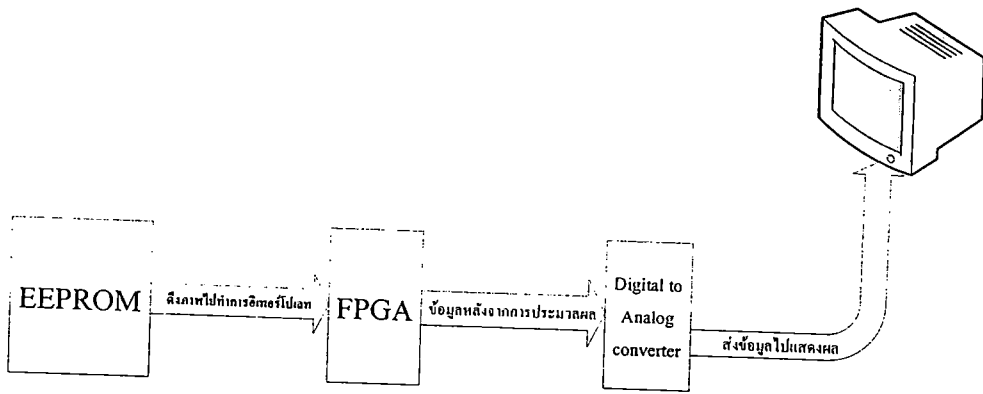
๗8๗2

• F5

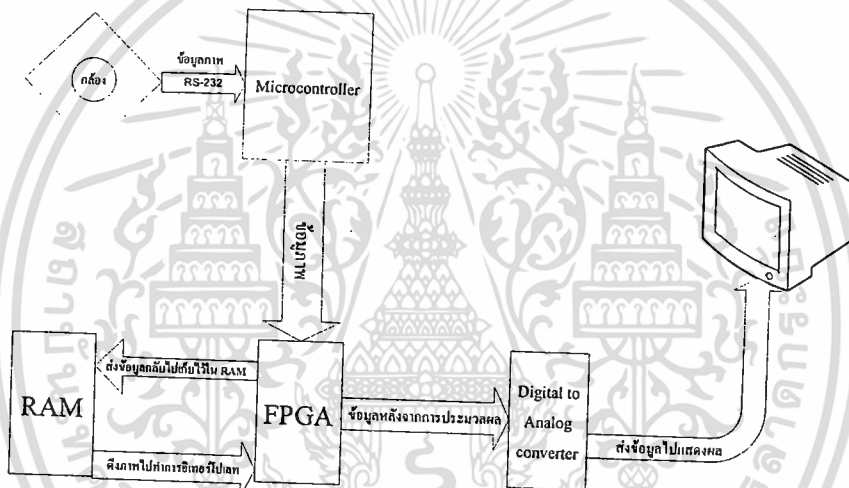
๐๑14๗๗

เลขหมู่.....
เลขทะเบียน.....105802.....
วัน,เดือน,ปี.....- 2 S.ศ. 2552.....

b. 1๐1๖๐๖๘๗
i.....



รูปที่ 1.1 โครงสร้างโดยรวมของชิ้นงานที่ใช้อีอีพรอมเป็นแหล่งเก็บข้อมูลต้นแบบ



รูปที่ 1.2 โครงสร้างโดยรวมของชิ้นงานที่ใช้ไมโครคอนโทรลเลอร์จับภาพเป็นแหล่งข้อมูลต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

การประมวลผลสัญญาณดิจิทัล หรือ ที่เรียกกันติดปากสั้น ๆ ว่า ดีเอสพี (DSP - digital signal processing) เป็นการศึกษาการประมวลผลสัญญาณที่อยู่ในรูปดิจิทัล (digital) โดยทั่วไปการประมวลผลสัญญาณ อาจแบ่งได้ตาม:

รูปแบบของตัวแทนสัญญาณ : การประมวลผลสัญญาณดิจิทัล (digital signal processing) และ การประมวลผลสัญญาณแอนะล็อก (analog signal processing)

คุณสมบัติของสัญญาณ : การประมวลผลสัญญาณไม่สุ่ม (deterministic signal processing) และ การประมวลผลสัญญาณสุ่ม (stochastic/statistical signal processing)

ลักษณะการประมวลผลสัญญาณ : เชิงเส้น (linear signal processing) และ ไม่เป็นเชิงเส้น (nonlinear signal processing) และ อื่นๆ ที่แบ่งตามคุณลักษณะเฉพาะของสัญญาณ หรือ ลักษณะเฉพาะของการประมวลผล เช่น adaptive signal processing, multirate/multiresolution signal processing, chaotic signal processing ฯลฯ

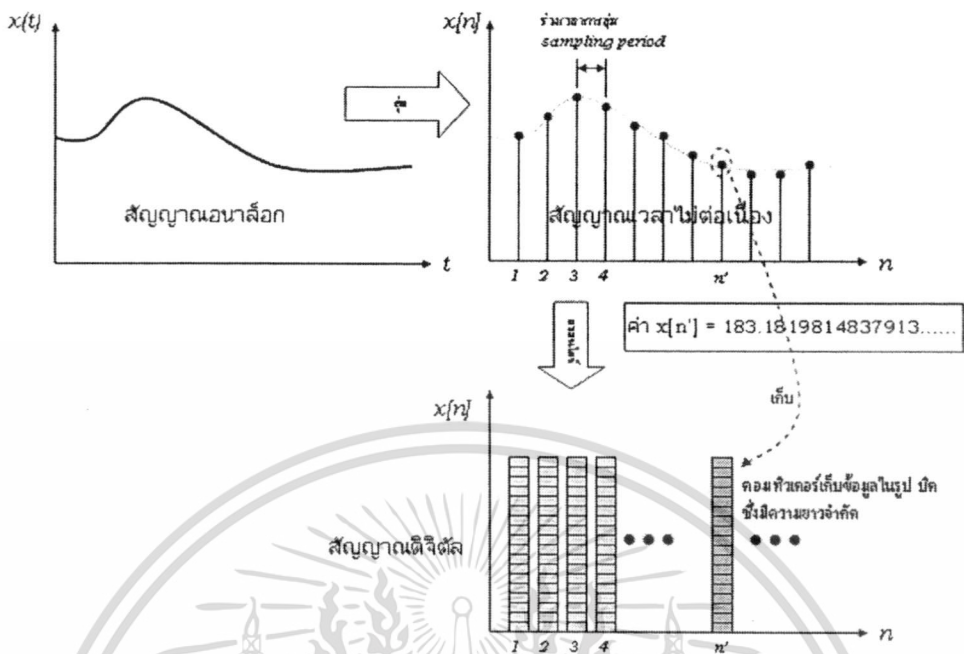
ดีเอสพีนี้อาจแบ่งออกได้เป็นในส่วนของซอฟต์แวร์และฮาร์ดแวร์ หรือตามการประยุกต์เป็น การประมวลผลสัญญาณเสียง (audio signal processing) การประมวลผลภาพดิจิทัล (digital image processing) และ การประมวลผลคำพูด (speech processing) ถึงแม้ว่าในดีเอสพีนั้น สัญญาณที่เราพิจารณากันจะเป็นดิจิทัล แต่โดยทั่วไปสัญญาณเหล่านี้จากแหล่งกำเนิด จะอยู่ในรูปเดิมที่เป็นแอนะล็อก การได้มาซึ่งสัญญาณดิจิทัลซึ่งเป็นตัวแทนสัญญาณแอนะล็อกนี้ จะต้องผ่านกระบวนการแปลงสัญญาณแอนะล็อกเป็นดิจิทัล (Analog-to-Digital Conversion - ADC) หรือการดิจิไทซ์ (digitization) ซึ่งประกอบด้วยการสุ่มตัวอย่าง (sampling) (อย่าสับสนกับคำว่า สุ่ม ที่มาจาก random หรือ stochastic) และการควอนไทซ์ (quantization) ให้อยู่ในรูปดิจิทัลก่อนที่จะทำการประมวลผลต่อไป

ตารางที่ 2.1 แสดงตัวอย่างการนำการประมวลผลสัญญาณดิจิทัลไปใช้งานด้านต่างๆ

ด้านการศึกษา	การประมวลผลทางเสียง	เครื่องมือวัด	ด้านทหาร	การประมวลผลภาพ	ด้านชีวิตการแพทย์
-Adaptive equalisation	-Speech recognition	-Spectrum analysis	-RADAR processing	-Pattern recognition	-patient monitoring
-Echo Cancellation	-Speech synthesis	-noise reduction	-Missile guidance	-robotic vision	-EEG signal (brain) analysis
-Video conferencing	-Text to speech	-data compression		-image enhancement	-ECG (heart) analysis
-Data communication	-Digital audio Equalisation			-facsimile	
				-Video animation	-X-ray storage and enhancement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้เป็นการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การสุ่มสัญญาณ (signal sampling) และการควอนไทซ์ (quantization)



รูปที่ 2.1 แสดงกระบวนการสุ่มสัญญาณและควอนไทซ์

2.2 โดเมนของเวลาและสเปซ (temporal and spatial domain)



รูปที่ 2.2 สัญญาณในรูปของตัวแปรเวลา หรือ temporal signal: เสียง



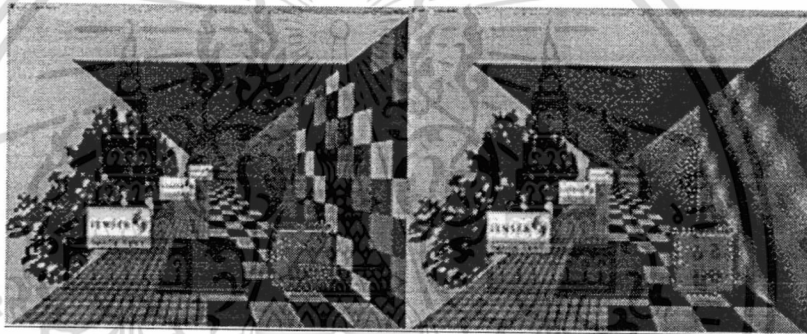
เอกสารนี้เป็นเอกสารรูปที่ 2.3 สัญญาณในรูปตัวแปรของสเปซหรือ spatial signal: รูปภาพ ระเบียบด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลภาพเป็นการเปลี่ยนภาพต่างๆ ที่ได้จากกล้องหรือการสแกน ตลอดจนข้อมูลภาพ ในรูปดิจิทัล แล้วนำไปผ่านกระบวนการบางอย่างเพื่อให้เกิดเป็นภาพใหม่ เช่น ภาพที่มีความคมชัดสดใส หรือภาพที่เบลอ เป็นต้น

ซึ่งในโครงการนี้ เราจะทำการขยายภาพให้มีขนาดเพิ่มมากขึ้นโดยกระบวนการ interpolate ในแบบต่างๆ

2.3 Bilinear

เป็นเทคนิคยอดนิยมที่ใช้กันโดยทั่วไป โดยหลักการทำงานคือ การรวมเอา texture โดยรอบ 4 จุด มาเฉลี่ยสีกัน ทำให้ texture กลมกลืนกันไป หรือ ถ้าไม่มีการทำ bi-linear แล้ว texture ที่ถูกขยายขนาด ออกมาจะมีลักษณะเป็นลายเหลี่ยมๆ (patch-work blanket) เนื่องจากแต่ละ texture จะสามารถมองเห็นเป็น แผ่นๆ (patches) ได้



รูปที่ 2.4 ภาพ point mapping และภาพที่มี bilinear

แต่ bilinear filtering ก็ยังเป็นแค่เทคนิคเดียว ซึ่งยังไม่สมบูรณ์นัก เพราะยังไม่สามารถสร้างความลึกให้กับภาพได้ ทั้งนี้เนื่องจาก วัตถุที่อยู่ไกลออกไป จะมีขนาดเล็กลง และขนาดที่เล็กลงนั้น ทำให้ texture ที่ถูก map อยู่บนภาพนั้นถูกบีบอัดมากขึ้น และเมื่อมีการเคลื่อนไหวจะทำให้เห็นเหมือนเกิดจุด หรือการเคลื่อนตัวของ texture ลองดูตัวอย่างภาพจาก Indy 3D ในส่วนบริเวณพื้น ซึ่งใช้แค่ bilinear เพียงอย่างเดียว จะเห็นว่า พื้นที่ควรจะได้ลายเส้นไปจนสุดสายตา จะมีการหายไป ของเส้นบางเส้น ทั้งนี้ เนื่องจากการถูกบีบของ texture ทำให้รายละเอียดบางส่วนหายไป โดยปกติแล้ว bilinear Filtering จะถูกใช้ควบคู่กับเทคนิค Mipmapping ซึ่งเป็นการใช้ texture รูปเดียวกัน แต่แตกต่างกันด้านขนาดและรายละเอียด สำหรับใช้กับวัตถุที่อยู่ใกล้ หรือ ไกลจากผู้ชม เพื่อความเหมาะสมกับภาพที่จะมองเห็น เช่น texture ขนาดใหญ่ รายละเอียดสูงจะถูกนำมาใช้ เมื่อวัตถุอยู่ใกล้ผู้ชม และเมื่อวัตถุเคลื่อนไกลออกไป texture ก็จะเปลี่ยนเป็น texture ที่มีขนาดเล็กลง และมีรายละเอียดน้อยลงตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การแปลงฟูรีเยร์ (Fourier transform)

ตั้งชื่อตาม โจเซฟ ฟูรีเยร์ หมายถึงการแปลงเชิงปริพันธ์โดยเป็นการเขียนแทนฟังก์ชันใดๆ ในรูปผลบวก หรือปริพันธ์ ของฐาน ที่เป็นฟังก์ชันรูปคลื่น ไซน์หรือโคไซน์ สัญลักษณ์ $\tau\{\}$ เป็นการแปลงทางคณิตศาสตร์ที่โดยมากมักใช้ในการวิเคราะห์ความถี่ โดยมีรากฐานจากอนุกรมฟูรีเยร์ (สังเคราะห์ความถี่) ความหมายโดยคร่าวๆ ก็คือถ้าเรามีสัญญาณที่เปลี่ยนแปลงในเวลาต่างๆ แล้วเราแปลงฟูรีเยร์สัญญาณเราจะได้ความถี่หรือสเปกตรัมของความถี่ ซึ่งอย่างที่ทราบกันอยู่แล้ว ความถี่ก็คือส่วนกลับของเวลา จะขอยกตัวอย่างการวิเคราะห์ความถี่ในสัญญาณคลื่นที่มีลักษณะผสมกันซึ่งถ้าเรามองด้วยตาจะแยกไม่ออกว่ามี ความถี่ใดรวมอยู่ในคลื่นนั้นบ้าง

ซึ่งถ้าสมมติว่าเราไม่ทราบความถี่ของคลื่นมาก่อนแล้วกราฟของคลื่นเหมือนดังรูปแรก เมื่อแปลงฟูรีเยร์ เราก็สามารถบอกความถี่ของคลื่นที่ผสมกันได้ว่ามีความถี่เท่ากับ 100,150 Hz ดังรูปที่สอง รูปที่สามพยายามจะบอกว่าค่าที่ได้จากการแปลงฟูรีเยร์จะอยู่ในรูปของจำนวนเชิงซ้อน คือมีทั้งส่วนจริงและส่วนจินตภาพซึ่งสมการของการแปลงฟูรีเยร์หนึ่งมิติก็คือ

$$\tau\{x\} = F(u) = \int_{-\infty}^{\infty} x e^{-i2\pi ux} dx \quad (2.1)$$

และสมการอินเวอร์สฟูรีเยร์ก็คือ

$$\tau^{-1}\{F(x)\} = f(u) = \int_{-\infty}^{\infty} F(x) e^{i2\pi ux} dx \quad (2.2)$$

2.4.1 การแปลงฟูรีเยร์แบบต่อเนื่อง

โดยปกติแล้วคำ "การแปลงฟูรีเยร์" จะใช้หมายถึง การแปลงฟูรีเยร์ต่อเนื่อง ซึ่งเป็นการเขียนแทนฟังก์ชัน $f(t)$ ที่สามารถหาปริพันธ์ของกำลังสองได้ ด้วยผลบวกของ ฟังก์ชันเอกซ์โปเนนเชียลเชิงซ้อน ซึ่งมีความถี่เชิงมุม ω และ ขนาด(หรือ แอมพลิจูด) เป็นจำนวนเชิงซ้อน $F(\omega)$

$$f(t) = F^{-1}(F)(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \quad (2.3)$$

ความสัมพันธ์ด้านบนคือ การแปลงกลับของ การแปลงฟูรีเยร์แบบต่อเนื่อง ส่วนการแปลงฟูรีเยร์นั้นปกติจะเขียน $F(\omega)$ ในรูปของ $f(t)$ คู่ของฟังก์ชันดั้งเดิมและผลของการแปลงของฟังก์ชันนั้น บางครั้งก็เรียกคู่ของการแปลง (transform pair) ดูข้อมูลเพิ่มเติมที่ การแปลงฟูรีเยร์ต่อเนื่อง ภาคขยายของการแปลงนี้คือ การแปลงฟูรีเยร์แบบไม่เป็นจำนวนเต็ม (fractional Fourier transform) ซึ่งค่ายกกำลังของการแปลง (จำนวนการแปลงซ้ำ) นั้น ไม่จำเป็นจะต้องเป็นจำนวนเต็ม สามารถเป็นค่าจำนวนจริงใดๆ

เมื่อ $f(t)$ เป็น ฟังก์ชันคู่ (ฟังก์ชันคี่) เทอมไซน์ (โคไซน์) จะไม่ปรากฏ ซึ่งคงเหลือไว้แต่ การแปลงโคไซน์ และ การแปลงไซน์ ตามลำดับ อีกกรณีหนึ่งคือ เมื่อ $f(t)$ เป็นฟังก์ชันค่าจริง จะทำให้

$$F(-\omega) = F^*(\omega) \quad (2.4)$$

การแปลงฟูรีเยร์ต่อเนื่องนั้นเป็นภาคขยาย ของแนวความคิดที่เกิดก่อนหน้านั้น คือ อนุกรมฟูรีเยร์ ซึ่งเป็นการเขียนแทน ฟังก์ชันคาบ (หรือฟังก์ชัน ในโดเมนจำกัด) $f(x)$ (มีคาบ 2π) ด้วย อนุกรม ของ เอกสฟังก์ชันรูปคลื่น: ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f(x) = \sum_{n=-\infty}^{\infty} F_n e^{inx} \quad (2.5)$$

ซึ่ง F_n เป็น ค่าจำนวนเชิงซ้อนของขนาดหรือค่าจริงของขนาดเมื่อฟังก์ชันเป็นฟังก์ชันค่าจริงอนุกรมฟูรีเยร์
 ยังอาจเขียนในรูป:

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)] \quad (2.6)$$

โดย a_n และ b_n เป็นค่าจำนวนจริงของขนาดของอนุกรมฟูรีเยร์

2.4.2 การแปลงฟูรีเยร์ไม่ต่อเนื่อง

สำหรับการคำนวณด้วยเครื่องคอมพิวเตอร์ ค่าสัญญาณในทั้งสองโดเมนจำเป็นต้องมีค่าเป็น
 ดิจิตอล ซึ่งคือฟังก์ชันค่าไม่ต่อเนื่อง $x[n]$ บนโดเมนไม่ต่อเนื่อง แทนที่จะเป็นโดเมนต่อเนื่อง ในช่วงจำกัด
 หรือ เป็นคาบ ในกรณีนี้เราจะใช้ การแปลงฟูรีเยร์ไม่ต่อเนื่อง (discrete Fourier transform-DFT) ซึ่งเขียน
 แทน $x[n]$ ด้วยผลบวกของฟังก์ชันคาบ

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi ink}{N}} \quad n = 0, \dots, N-1 \quad (2.7)$$

2.4.2.1 การแปลงฟูรีเยร์เต็มหน่วย (DFT)

ผลการแปลงฟูรีเยร์เต็มหน่วยของสัญญาณ $x(n)$ พิจารณาได้จากอนุกรมฟูรีเยร์เต็มหน่วย (discrete
 Fourier series หรือ DFS) ของสัญญาณเป็นคาบ $\tilde{X}(n)$ ที่มีคาบเท่ากับ N แสดงได้ดังสมการ (2.8)

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{j \left(\frac{2\pi}{N}\right) kn} \quad (2.8)$$

โดยที่ $\tilde{X}(k)$ คือค่าสัมประสิทธิ์ของอนุกรมฟูรีเยร์ ซึ่งคำนวณได้จาก $\tilde{x}(n)$ ตามความสัมพันธ์ดัง
 สมการ(2.9)

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j \left(\frac{2\pi}{N}\right) kn} \quad (2.9)$$

ค่าสัมประสิทธิ์ฟูรีเยร์ของ $x(n)$ จึงมีค่าเท่ากับหนึ่งคาบของ $\tilde{X}(k)$ แสดงดังสมการ (2.10) ซึ่ง
 เรียกว่า ผลการแปลงฟูรีเยร์เต็มหน่วย หรือ DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \left(\frac{2\pi}{N}\right) kn} \quad , \quad 0 \leq k \leq N-1 \quad (2.10)$$

สเปกตรัมขนาดของ $X(k)$ จะแสดงองค์ประกอบทางความถี่ของสัญญาณ $x(n)$ ที่ให้พลังงาน
 สูงสุด หรือเรียกว่า ฟอर्मานต์ (formant) ในกรณีสัญญาณเสียงคนตรี นำอัลกอริทึมการแปลงฟูรีเยร์อย่าง
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวดเร็ว (fast Fourier transform หรือ FFT) มาใช้ในการคำนวณ DFT โดยกำหนดให้ N เป็นค่ากำลังของสองและ $W_N = e^{-j(2\pi/N)}$ จะได้ $X(k)$ มีค่าตามสมการ (2.11)

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad , \quad 0 \leq k \leq N-1 \quad (2.11)$$

อัลกอริทึม FFT จะทำการแบ่ง $x(n)$ ออกเป็นสองลำดับเท่าๆกัน คือลำดับของเลขคู่ (แทน $n = 2r$) และลำดับของเลขคี่ (แทน $n = 2r + 1$) โดยที่ $0 \leq r \leq \left(\frac{N}{2}\right) - 1$ ดังสมการ (2.12)

$$X(k) = \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r+1)W_N^{(2r+1)k} \quad (2.12)$$

ให้ $W_N^2 = W_{N/2}$ จะได้ DFT จำนวน N จุด เป็นผลรวมของ DFT จำนวน $N/2$ จุด สองลำดับ แสดงดังสมการ (2.13)

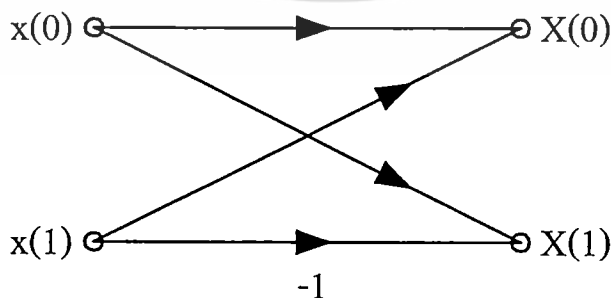
$$X(k) = \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r)W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\left(\frac{N}{2}\right)-1} x(2r+1)W_{N/2}^{rk} \quad (2.13)$$

ซึ่งแต่ละลำดับสามารถแยกออกเป็นผลรวมของ DFT จำนวน $N/4$ จุด สองลำดับ จนกระทั่งแยกได้เป็น DFT จำนวน 2 จุด $N/2$ ลำดับ กระบวนการนี้จะสิ้นสุด การคำนวณ DFT จำนวน 2 จุด แสดงได้ดังสมการ (2.14) และเนื่องจาก $W_1^{0k} = 1$ และ $W_2^k = (-1)^k$ จะได้ค่าของ $X(k)$ แสดงดังสมการ (2.15) และ (2.16) แทนด้วยกราฟที่เรียกว่า บัตเตอร์ฟลาย (butterfly) ได้ดังรูปที่ 2.5 และการคำนวณ DFT จำนวน 4 จุด แสดงดังรูปที่ 2.6

$$X(k) = x(0)W_k^{0k} + W_2^k [x(1)W_k^{0k}] \quad , \quad k = 0,1 \quad (2.14)$$

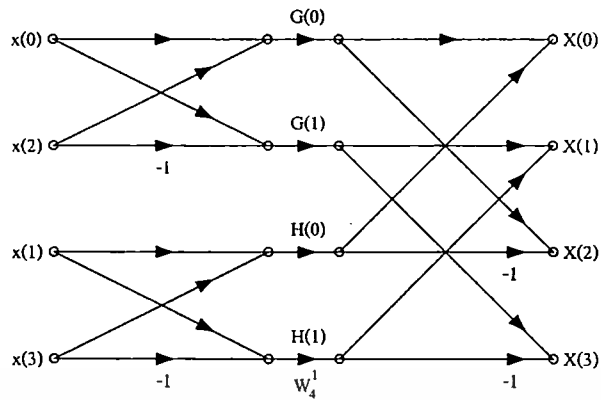
$$X(0) = x(0) + x(1) \quad (2.15)$$

$$X(1) = x(0) - x(1) \quad (2.16)$$



รูปที่ 2.5 กราฟของการคำนวณ DFT จำนวน 2 จุด (บัตเตอร์ฟลาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 กราฟของการคำนวณ DFT จำนวน 4 จุด

สำหรับการวิเคราะห์สัญญาณด้วย DFT การเพิ่มความถูกต้องของตำแหน่งเส้นสเปกตรัม ทำได้โดยเพิ่มขนาดของการคำนวณ DFT ให้เป็น R จุด ซึ่งจำนวนจุดที่เพิ่มขึ้นจะเป็นค่าศูนย์ ตัวอย่างต่อไปนี้ แสดงการคำนวณ DFT ของสัญญาณไซน์ซอซด์ $y(n)$ มีค่าตามสมการ (10) โดยกำหนดให้ $N = 16$, $f_1 = 0.22$ Hz, $f_2 = 0.34$ Hz และ R มีค่าเปลี่ยนแปลงจาก 16 จุด ถึง 128 จุด

$$y(n) = \left(\frac{1}{2}\right) \sin(2\pi f_1 n) + \sin(2\pi f_2 n), \quad 0 \leq n \leq N-1 \quad (2.17)$$

2.5 ทฤษฎีทวินาม (Binomial Theorem) และสามเหลี่ยมปาสกาล (Pascal's Triangle)

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

$$(x+y)^2 = x^2 + 2xy + y^2 \quad (2.18)$$

ถ้าเราต้องการกระจายเทอม $(x+y)^n$ ใดๆ จะหาได้โดยทฤษฎีทวินาม ดังสมการ

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k \quad (2.19)$$

$$= x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \dots + y^n \quad (2.20)$$

$$= a_0 x^n + a_1 x^{n-1} y + a_2 x^{n-2} y^2 + \dots + a_{n-1} x y^{n-1} + a_n y^n \quad (2.21)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการกระจายเทอมในสมการ (2.19) - (2.21) บางครั้งเรียกได้ว่าเป็นการกระจายเทอมไบนอมิเยล (Binomial Expansion) โดย a_k เรียกว่าสัมประสิทธิ์ไบนอมิเยล (Binomial coefficient) ซึ่งจะมีค่าดังสมการ

$$a_k = \binom{n}{k} \quad (2.22)$$

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

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!} = a_k \quad (2.23)$$

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

สำหรับค่าของสัมประสิทธิ์ไบนอมิเยลนั้นสามารถนำมาจัดเรียงเป็นรูปฟอร์มที่เป็นสามเหลี่ยมที่รู้จักกันในชื่อว่า 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

เอกสารนี้เป็นรูปที่ 2.7 แสดงสามเหลี่ยมปาสคาลและสามเหลี่ยมปาสคาลที่จัดรูปเป็นสามเหลี่ยมมุมฉาก ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แต่ละแถว (n) ของสามเหลี่ยมปาสคาลสามารถสร้างได้ด้วยวิธีอื่นที่ไม่ใช่จากสมการที่ (2.21) โดยการบวกของ 2 องค์ประกอบที่อยู่ในแบบทแยงขึ้นไปในแถวข้างบน เช่น ในแถวที่ 4 (n=3) องค์ประกอบตัวที่ 3 (ที่ n=3, k=1) มาจาก 1+2 ในแถวบน (n=2) และ 3 ตัวถัดไป (ที่ n=3, k=2) มาจาก 2+1 เป็นต้น ซึ่งสามารถพิสูจน์คุณสมบัติ ดังกล่าว (การบวกเพื่อสร้างองค์ประกอบในแถวต่อไป) ได้ดังนี้

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)!}{k!(n-k)!} \quad (2.24)$$

$$= \frac{(n-k)(n-1)!}{k!(n-k)!} + \frac{k(n-1)!}{k!(n-k)!} \quad (2.25)$$

$$= \frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} \quad (2.26)$$

$$= \binom{n-1}{k} + \binom{n-1}{k-1} \quad (2.27)$$

โดยสมาชิกในสามเหลี่ยมปาสคาลในแถวที่ n ใด ๆ สามารถคำนวณได้ด้วยการหาสัมประสิทธิ์ไบนอมิยัล ดังในสมการที่ (2.24) ถึง (2.27) สำหรับ k=1, 2, 3, ..., n ได้

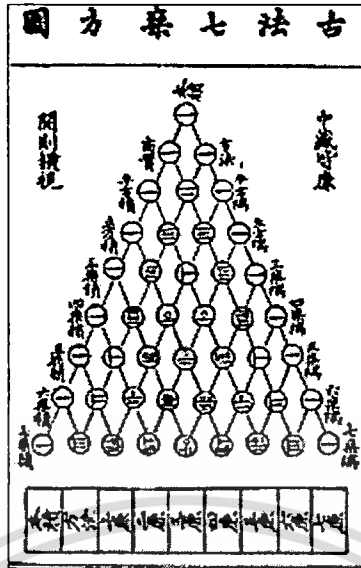
ดังนั้นเมื่อเราจะทำการกระจายเทอมของ $(x+y)^n$ ใด ๆ เราจะสามารถหาค่าของสัมประสิทธิ์ไบนอมิยัลจากสมการที่ (2.21) ได้โดยง่ายจากสามเหลี่ยมปาสคาลนั่นเอง เช่น $(x+y)^4$

$$(x+y)^4 = 1x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + 1y^4 \quad (2.28)$$

โดยสัมประสิทธิ์ในที่นี้มาจากสามเหลี่ยมปาสคาลในรูปที่ 2.7 ในแถวที่ 5 (n=4)

2.5.1 ที่มาของชื่อสามเหลี่ยมปาสคาล

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



รูปที่ 2.8 แสดงสามเหลี่ยมของ ยาง ฮุย

2.5.2 การแปลงปาสกาล

จะทำการแปลงสัญญาณที่ไม่ต่อเนื่องทางเวลาที่ $x[n]$ ($0 \leq n \leq N$) ไปที่สัญญาณที่ไม่ต่อเนื่องทางเวลาอื่น $X[k]$ ($0 \leq k \leq N$) ตามที่สมการด้านล่าง

$$X[k] = \sum_{j=0}^k (-1)^j \binom{k}{j} x[j] \quad (2.29)$$

ฟังก์ชันมาตรฐานของการแปลงปาสกาล คือ พหุนาม $P_k(x)$ ซึ่งคำนวณได้จาก

$$p_k = \frac{(-1)^k}{k!} x(x-1)(x-2)\dots(x-n+1) \quad (2.30)$$

ดังนั้น

$$x[n] = \sum_{k=0}^N X[k] P_k(n) \quad (2.31)$$

โดยที่ p_k เป็นฟังก์ชันต่อเนื่องแม้ว่าฟังก์ชันมาตรฐานของการแปลงปาสกาล $P_k(x)$ เป็นแบบไม่ต่อเนื่อง

2.5.3 การแปลงปาสกาลในหนึ่งมิติ (1-D Fast Pascal Transform)

การแปลงดิสครีตปาสกาลทรานสฟอร์ม (DPT: discrete Pascal Transform) ถูกนำเสนอแรก ๆ โดย โทมัส เจ กูดแมน (Thomas J. Goodman) ซึ่งมีรูปแบบสมการการแปลงดังนี้

$$X = P x \quad (2.32)$$

โดย X = เมทริกซ์ผลลัพธ์จากการแปลง

x = เมทริกซ์ของข้อมูลอินพุต

P = เมทริกซ์ของการแปลงปาสกาล (Pascal Transform Matrix)

โดยการสร้างเมทริกซ์ของการแปลงปาสกาลนั้นสามารถสร้างได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P = [P_{ij}] = (-1)^j \binom{i}{j} = \frac{(-1)^j}{j!} i^{\underline{j}}; i, j = 0, 1, 2, \dots, N-1 \quad (2.33)$$

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

$$i^{\underline{j}} = i(i-1)(i-2)\dots(i-j+1) \quad ; j \geq 1 \text{ and } i^{\underline{0}} = 1 \quad (2.34)$$

และสมการที่ (2.33) นี้เองถือว่าเป็นฟังก์ชันพื้นฐาน (Basis function) ของการแปลงดิสครีตปาสคาลทรานสฟอร์มนี้ ซึ่งผลที่ได้จากการสร้างเมทริกซ์ของการแปลงปาสคาล ดังในสมการที่ (2.33) ซึ่งจะเหมือนกับการนำสมาชิกในสามเหลี่ยมปาสคาล ในรูปที่ 2.7 (ข) มาจัดวางเป็นเมทริกซ์และทำการสลับเครื่องหมายในแต่ละหลักจากบวกไปเป็นลบสลับกันไปตามลำดับ

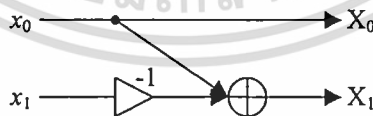
2.5.4 การสร้างเมทริกซ์การแปลงปาสคาลขนาดต่าง ๆ

เมื่อ $N=2$ จะได้

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

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

สามารถสร้างบัตเตอร์ฟลายยูนิท (butterfly unit) อย่างง่ายได้ดังนี้



รูปที่ 2.9 แสดงการสร้างบัตเตอร์ฟลายยูนิทเมื่อ $N=2$

เมื่อ $N=3$ จะได้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

$$P = \prod [q_{ij}]_k ; i, j = 0, 1, 2, \dots, N-1 \quad (2.35)$$

2.5.5 การแปลงปาสกาลเต็มหน่วยในสองมิติ (2-D Fast Pascal Transform)

สำหรับการแปลงปาสกาลเต็มหน่วยแบบ 2 มิติ (2-D DPT) สามารถทำการแปลงได้ดังนี้

$$X = PxP^T \quad (2.36)$$

เมื่อ X = เมตริกซ์ผลลัพธ์จากการแปลง

x = เมตริกซ์ของข้อมูลอินพุต

P = เมตริกซ์ของการแปลงปาสกาล

และกรณีแปลงกลับ (Inverse Transforms) สามารถแสดงได้ดังสมการที่ (2.23)

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

เพื่อความง่ายและสะดวกในการอธิบายหลักการแปลงดีสครีตปาสกาลทรานสฟอร์มในสองมิติ ให้พิจารณาตัวอย่างเมื่อ : $N = 3$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

สมมติให้เมตริกซ์นี้ซึ่งผ่านการแปลงดีสครีตปาซกาลทรานสฟอร์มในหนึ่งมิติ ในแนวหลัก (column operation) มาเป็น $[y_{a,b}]$ ดังนั้นจะได้ว่า

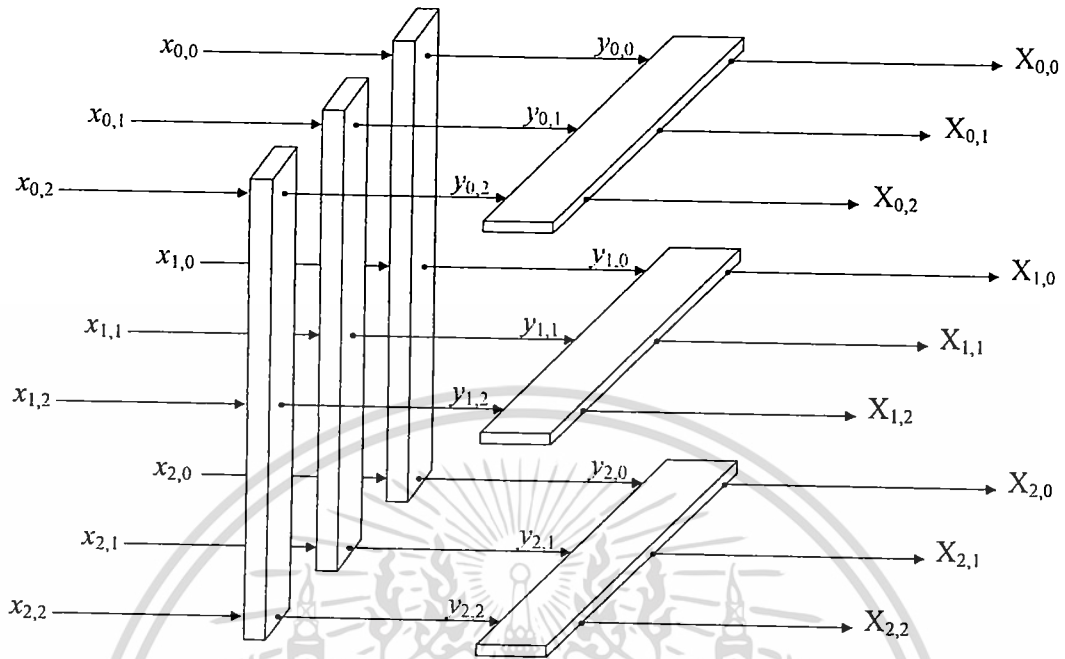
$$\therefore \begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} y_{0,0} & y_{0,1} & y_{0,2} \\ y_{1,0} & y_{1,1} & y_{1,2} \\ y_{2,0} & y_{2,1} & y_{2,2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0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

เมตริกซ์นี้คือการดำเนินการในส่วนของแถว (row operation) สำหรับอินพุตที่ผ่านการดำเนินการในส่วนของคอลัมน์มาแล้ว ผลลัพธ์ทั้งหมดคือ การดำเนินการแปลงดีสครีตปาซกาลทรานสฟอร์มในสองมิติ ในที่สุดจะได้ว่า

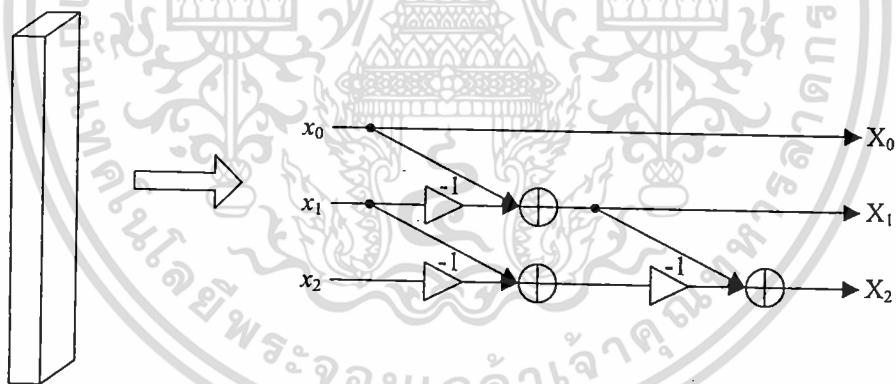
$$\begin{bmatrix} X_{0,0} & X_{0,1} & X_{0,2} \\ X_{1,0} & X_{1,1} & X_{1,2} \\ X_{2,0} & X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} x_{0,0} & x_{0,0} - x_{0,1} & x_{0,0} - 2x_{0,1} + x_{0,2} \\ [x_{0,0} - x_{1,0}] & [x_{0,0} - x_{1,0}] - [x_{0,1} - x_{1,1}] & [x_{0,0} - x_{1,0}] - 2[x_{0,1} -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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถสรุปเป็นไดอะแกรม ได้ดังนี้



รูปที่ 2.10 ไดอะแกรม เมื่อ $N=3$



รูปที่ 2.11 แสดงบิตเตอร์ฟลายยูนิทในไดอะแกรม เมื่อ $N=3$

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

การทำงานของจอภาพนั้นบนหน้าจอจะประกอบไปด้วยพิกเซล (Pixel) หรือจุดขนาดเล็กในการแสดงผล โดยจุดเหล่านี้มีขนาดเล็กมากและจะกระพริบอยู่ตลอดเวลา ในความถี่ต่าง ๆ เมื่อมีการแสดงผลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดหลอดภาพเล็ก ๆ เหล่านี้ที่อยู่กันเป็นกลุ่มจะเปิด ปิด เพื่อให้เกิดความแตกต่างกันจนออกมาเป็นตัวอักษร หรือรูปภาพที่ต้องการ

จอภาพที่ดีจะต้องมีขนาดของความละเอียด (Resolution) ที่สูงจะได้ความคมชัดมากขึ้น นอกจากนี้อัตราการแสดงภาพใหม่ (Refresh rate) ของจอภาพก็เป็นเรื่องสำคัญที่ควรจะต้องพิจารณาเพราะ หากความละเอียดสูงภาพจะคมชัดขึ้นหรือสามารถมีรายละเอียดมากขึ้นเป็นต้น ส่วนอัตราการแสดงภาพใหม่ จะเป็นอัตราที่บอกความถี่ในการแสดงผลและการสแกนภาพ เพื่อให้เกิดความคมชัดและไม่เป็นอันตรายต่อสายตาโดยทั่วไปนิยมตั้งไว้ที่ 50–60 Hz และนอกจากนั้นปริมาณสีที่สามารถแสดงผลได้ เป็นเรื่องที่สำคัญอีกประการหนึ่ง

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

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

จอภาพที่ใช้แสดงผลของคอมพิวเตอร์มักถูกเรียกว่ามอนิเตอร์ ซึ่งในการแสดงผลนั้นมักจะประกอบด้วยข้อมูลทั้งที่เป็นตัวหนังสือและภาพกราฟิก สำหรับเครื่องคอมพิวเตอร์แบบตั้งโต๊ะโดยทั่วไปจะใช้เทคโนโลยีจอภาพแบบหลอดรังสีแคโทด ในขณะที่คอมพิวเตอร์แบบพกพา เช่น โน้ตบุ๊ก จะใช้เทคโนโลยีแบบลิควิดคริสตัลดีสเพลย์ และไลท์อิมิตทิงไดโอด (LCD : Liquid Crystal Display, LED : Light-Emitting Diode) หรือแก๊สพลาสมา(GasPlasma)

จอภาพแบ่งออกได้ 2 ประเภทคือ

- จอภาพสีเดียวหรือ โมโนโครม เป็นจอภาพที่ใช้กันทั่วไปในสมัยก่อน แต่ปัจจุบันใช้เฉพาะในบางที่ เช่น เครื่องคิดเงินในร้านค้า หรือตู้เอทีเอ็มของบางธนาคาร เป็นต้น มีลักษณะการแสดงผลเป็นสีเดียว เช่น สีขาว สีเขียว หรือสีส้มและแสดงได้เฉพาะตัวอักษรเท่านั้น

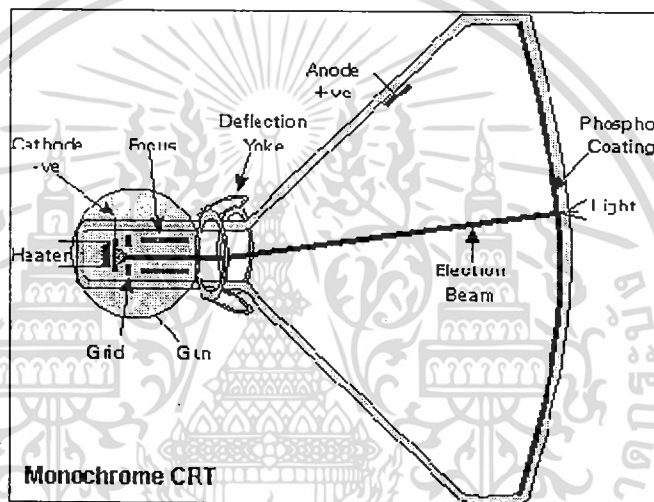
- จอภาพโมโนโครมแบบไดเรกต์ไดรฟ์ (Direct-drive) ใช้ได้กับทั้งหน้าจอ โมโนโครม และหน้าจอ อีจีเอ

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อใช้ในการเรียนการสอน มีอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จอภาพสีแบบคอมโพสิตจอภาพสีแบบนี้สามารถแสดงสีและกราฟิกได้ แต่คุณภาพความละเอียดของภาพกราฟิกจะต่ำ
- จอภาพสีแบบอาร์จีบี จอภาพสีแบบนี้คุณภาพดีในการแสดงข้อความและมีความละเอียดสูงในงานกราฟิกแบบที่ต้องการความละเอียดสูง
- จอภาพแบบหลายความถี่ (Variable Frequency Monitors) จอภาพสีแบบนี้จะมีพื้นฐานคล้ายคลึงกับจอภาพแบบอาร์จีบีแต่มีหลายความถี่จึงเป็นจอที่มีความสามารถสูงใช้งานได้กับหลากหลายระบบ เพราะเป็นจอที่สามารถเข้ารหัสสัญญาณสีได้มากกว่าหนึ่งแบบทั้งแบบดิจิทัลหรืออนาล็อกและใช้กับช่วงสัญญาณความถี่ที่กว้างขึ้น



รูปที่ 2.12 ส่วนประกอบของจอภาพโมโนโครม

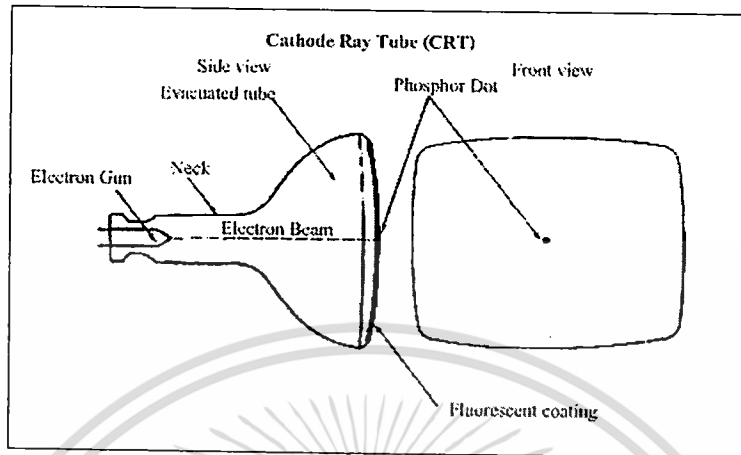
2.6 การกำเนิดการแสดงผลภาพให้กับจอภาพวีจีเอ

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

2.7 เทคโนโลยีการแสดงผลจอภาพ (Video Display Technology)

เทคโนโลยีที่ใช้โดยส่งการแสดงผลเป็นสัญญาณภาพนิ่งที่เหมือนธรรมชาติในรูปแบบของสัญญาณภาพจะใช้หลอดภาพรังสีแคโทดซึ่งเกิดขึ้นครั้งแรกเมื่อปี ค.ศ. 1897 โดยนักวิทยาศาสตร์ชาวเยอรมันชื่อ คาร์ล เฟอร์ดินานด์ บราวน์ (Karl Ferdinand Braun) โดยพัฒนาเป็นเครื่องมือที่เรียกว่า ออสซิลโลสโคป (Oscilloscope) ที่ใช้เป็นเครื่องวัดค่าสัญญาณทางไฟฟ้า และเริ่มผลิตออกมาในเชิงพาณิชย์ครั้งแรกในปี ค.ศ. 1922 ในรูปแบบของจอโทรทัศน์จนถึงทุกวันนี้ หลอดภาพรังสีแคโทดเป็นไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้กันแพร่หลายทั่วไปส่วนมากจะมีใช้ในรูปแบบชุดโทรทัศน์บ้างหรือใช้เป็นตัวเชื่อมต่อในการแสดงผลของระบบคอมพิวเตอร์ก็ดี



รูปที่ 2.13 แสดงถึงอุปกรณ์หลอดรังสีแคโทดซึ่งก็คือหลอดแก้วที่ยิงอิเล็กตรอนไปยังฉากด้านหน้า

ที่ฉากของจอภาพพื้นผิวด้านในจะเคลือบสารฟลูออเรสเซนต์ (Fluorescent) อยู่ และที่ตรงส่วนท้ายของหลอดภาพ จะมีปืนยิงลำอิเล็กตรอน เมื่อปืนนี้ยิงไปกระตุ้นที่ฉากก็จะทำให้เกิดจุดสว่างขึ้น

ซึ่งหลอดอิเล็กตรอนนี้จะยิงลำอิเล็กตรอน (Electron beam) กวาดสแกนบนฉากตามแนวนอนที่ต่อเนื่องกัน และเพื่อที่จะกำเนิดมาเป็นภาพ จะมีแกนเหล็กสำหรับเบี่ยงเบนลำอิเล็กตรอน (Deflection yoke) ขดลวดเล็กๆบนนี้จะแบ่งเป็นแกนแนวตั้งกับแนวนอน (จะอยู่ที่คอของหลอดรังสีแคโทด) โดยจะอาศัยสนามแม่เหล็ก (หรืออาจจะใช้สนามไฟฟ้าสถิตก็ได้ แต่ไม่เป็นที่นิยม) เบี่ยงเบนลำอิเล็กตรอนให้ไปตามตำแหน่งที่ต้องการบนฉากของจอภาพ ส่วนข้อมูลของสัญญาณอาร์จีบี ที่ออกมาเป็นสัญญาณภาพจะใช้ควบคุมความแรงของอิเล็กตรอน และแสงที่เกิดขึ้นบนจอภาพนั้นเกิดจากการที่มีสัญญาณภาพเข้ามาซึ่งจะทำให้ลำอิเล็กตรอนทำงาน และสามารถพุ่งไปปะทะจุดสีเรียงสแกนฉากของหลอดรังสีแคโทดซึ่งบนฉากของหลอดรังสีแคโทดจะเคลือบด้วยสารเรืองแสง 3 สี (สีแดง, สีเขียว และ สีน้ำเงิน)

2.8 การกวาดตรวจภาพ (Scanning)

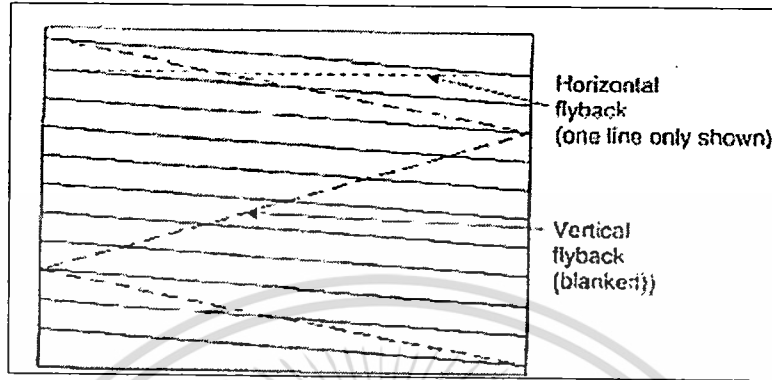
การที่จะนำภาพ 2 มิติเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งทำได้โดยทำภาพนั้นเป็นสัญญาณไฟฟ้าซึ่งจะง่ายต่อการเคลื่อนย้าย แต่ปัญหาที่จะทำการเปลี่ยนภาพ 2 มิติไปเป็นโวลต์เดจที่เปลี่ยนตามเวลาปัญหานี้แก้ได้โดยทฤษฎีของการกวาดตรวจ

วิธีการสแกนของหลอดรังสีแคโทดที่นิยมใช้กันอย่างแพร่หลายในการออกแบบทางการค้าโทรทัศน์และจอภาพแสดงผล มีชื่อเรียกว่า การตรวจกวาดแบบราสเตอร์ (Raster scan) ซึ่งวิธีนี้จะต้องมีการให้กำเนิดซิงค์ทั้งในแนวแกนตั้งและแนวนอนเพื่อให้ลำอิเล็กตรอนสามารถเคลื่อนที่ได้

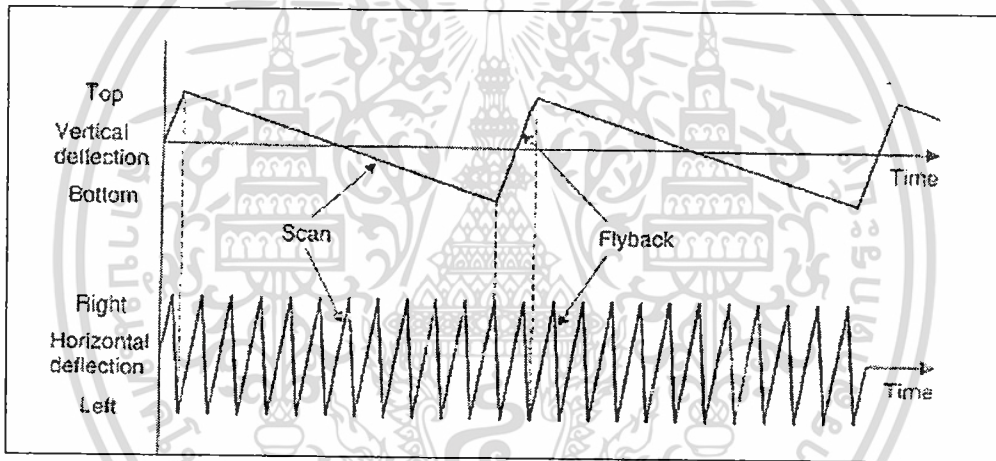
ในการสแกนแบบนี้จะเริ่มทำจากซ้ายไปขวา และจากข้างบนลงไปข้างล่างของจอภาพ โดยที่ลำอิเล็กตรอนจะถูกเบี่ยงเบนให้ไปอยู่ทางมุมซ้ายบนและเมื่อกวาดไปทางขวาของฉาก ลำอิเล็กตรอนก็จะ

สลับกลับโดยตอนที่กลับนั้นก็จะมี การเลื่อนตำแหน่งลงไปด้วย การสลับกลับนี้เรียกว่ารีเทรอส (Retrace) ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ ฟลายแบ็ก (fly back) ซึ่งในส่วนของการสลับกลับของลำอิเล็กตรอนนั้นความเข้มของลำอิเล็กตรอนจะลดลง เมื่อลดความเข้มลงอย่างเพียงพอแล้วเวลาที่เคลื่อนด้วยฟลูออเรสเซนต์ก็จะไม่ถูกกระตุ้นทำให้เกิดแสงสว่างจึงมองไม่เห็นเส้นที่ลากกลับ ดังรูปที่ 2.15



(ก) การเคลื่อนที่ของการสแกนภาพ



(ข) กราฟแสดงความสัมพันธ์ของการเคลื่อนที่ของการสแกนกับเวลา

รูปที่ 2.14 การสแกนภาพแบบราสเตอร์

2.8.1 การซิงค์โครไนซ์ (Synchronization)

สิ่งสำคัญที่ทำให้การสแกนภาพหนึ่งได้ถูกต้องนั้นเป็นหน้าที่ของสัญญาณซิงค์ซึ่งแบ่งได้เป็นสัญญาณควบคุมการสแกนทางแนวนอน และสัญญาณควบคุมการสแกนทางแนวตั้ง โดยทำงานไปพร้อม ๆ กันได้นั้นเพราะสัญญาณนี้จะส่งข้อมูลตามเวลาแนบไปกับสัญญาณวีดีโอ

สัญญาณควบคุมการสแกนในแนวแกนอนและแกนตั้ง

- สัญญาณควบคุมการสแกนทางแนวนอนจะเริ่มทำงานที่การเริ่มสลับกลับ หรือจบการสลับกลับ จะไม่ทำตอนเริ่มสลับกลับ
- สัญญาณควบคุมการสแกนทางแนวตั้งเริ่มทำงานที่การเริ่มสลับกลับในแนวแกนตั้ง (ก็คือจะทำงานตอนหลอดภาพยิงไปอยู่ที่ล่างขวาของจอภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหากไม่มีสัญญาณควบคุมการสแกนทางแนวตั้งจะทำให้ภาพที่ถูกสร้างทวนซ้ำขึ้นมาจะไม่สามารถคงตำแหน่งในแนวแกนตั้ง ภาพจะวิ่งขึ้น-ลง

แต่ถ้าหากสัญญาณควบคุมการสแกนทางแนวนอนไม่ซิงก์กันจะทำให้ภาพนั้นจะเลื่อนซ้าย-ขวา และบางส่วน จะเหมือนภาพถูกแยกออก

โดยทั่วไปแล้วจอภาพส่วนใหญ่จะใช้สัญญาณอนาล็อกอาร์จีบีมาตรฐานที่ $0.7 V_{p-p}$ ส่วนสัญญาณซิงก์ก็จะขึ้นอยู่กับประเภทลักษณะที่ใช้ดังต่อไปนี้

1. ในการเชื่อมต่อสัญญาณอาร์จีบี สัญญาณควบคุมการสแกนทางแนวตั้ง และสัญญาณควบคุมการสแกนทางแนวนอน จะใช้สายในการเชื่อมต่อ 5 เส้น มาตรฐานของสัญญาณอาร์จีบีที่ใช้คือ $0.7 V_{p-p}$ และสัญญาณซิงก์ใช้มาตรฐานทางลอจิก ลักษณะการใช้งานแบบนี้จะง่ายและเป็นที่ยอมรับ
2. ในการเชื่อมต่อสัญญาณอาร์จีบีและสัญญาณคอมโพสิต (เรียก CSYNC: Composite sync) จะใช้สายในการเชื่อมต่อ 4 เส้น เพราะจะใช้สัญญาณซิงก์ 1 เส้น ใช้มากในพวกโปรเจกเตอร์ในสมัยก่อน
3. ในการเชื่อมต่อสัญญาณอาร์จีบีและสัญญาณซิงค์ออนกรีน (Sync on green) จะใช้สายในการเชื่อมต่อ 3 เส้น สัญญาณสีแดง และสีน้ำเงิน ยังคงใช้ $0.7 V_{p-p}$ แต่สัญญาณสีเขียว จะส่งรวมไปกับสัญญาณคอมโพสิต ซึ่งจะใช้งานที่ $-0.3 V_{p-p}$

2.8.2 อัตราการสแกนภาพ (Scan Rate)

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

จำนวนสัญลักษณ์ที่จะแสดงได้ใน 1 แถวนั้นจะขึ้นอยู่กับความถี่ของสัญญาณภาพที่ใช้ ส่วนจำนวนบรรทัดที่จะแสดงผลบนจอภาพนั้นจะขึ้นอยู่กับอัตราการสแกนภาพ ซึ่งสามารถเพิ่มได้ วิธีที่เห็นได้ชัดที่สุดก็คือการเพิ่มความถี่ในการกวาดในแนวนอน ซึ่งก็จะทำให้เกิดเส้นสแกนแนวนอน มากขึ้น ตัวอย่างเช่น

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

2.8.3 อัตราการแสดงผลใหม่ (Refresh rate)

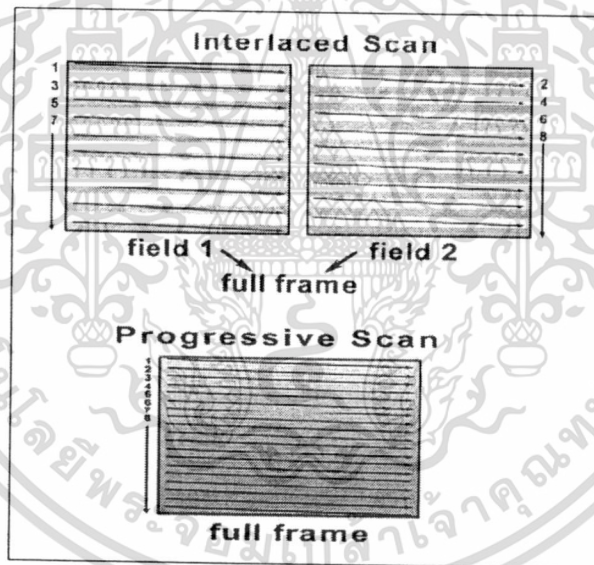
คือ อัตราส่วนของจำนวนครั้งที่จุดภาพถูกฉายลงบนหน้าจอภาพ ซึ่งจะเริ่มจากซ้ายไปขวาและบนลงล่างในเวลา 1 วินาทีซึ่งเป็นค่าที่สำคัญเพราะว่ามีผลกระทบโดยตรงต่อการมองเห็นภาพที่แสดงบนหน้าจอภาพ ปัญหาหลักของการเพิ่มเส้นสแกนนั้นคือถ้าอัตราการแสดงผลใหม่แนวตั้งนั้นต่ำกว่า

เอกลักรีนเป็นเอกลักษณ์ของภาพที่แสดงบนหน้าจอภาพ การเพิ่มเส้นสแกนนั้นคือถ้าอัตราการแสดงผลใหม่แนวตั้งนั้นต่ำกว่า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับความถี่ มาตรฐานของหลอดรังสีแคโทดก็จะทำให้ลำอิเล็กตรอนเลี้ยวเบนด้วยสนามไฟฟ้าที่มีรูปแบบ ไม่ถูกต้องผลกระทบนี้เป็นเหตุให้เกิดแสงภาพของจอกระตุกสั่น การแก้ปัญหาก็คือเราควรใช้อัตรา การแสดงภาพใหม่นั้นให้เท่าความถี่มาตรฐาน ดังนั้นในสหรัฐอเมริกา ที่ความถี่การกวาดในแนวนอน เท่ากับ 31.5 kHz ควรใช้ความถี่ในการแสดงภาพใหม่มากกว่าหรือเท่ากับ 60 Hz ในขณะที่ประเทศไทยก็ใช้ ตั้งแต่ความถี่ 50 Hz ขึ้นไป เพราะถ้าอัตราการแสดงภาพใหม่น้อยไปก็จะทำให้ภาพนั้นสั่นกระพริบเป็น อันตรายต่อสายตา ค่าอัตราการแสดงภาพใหม่นี้ก็จะขึ้นกับสายตาของแต่ละคน และก็ยังจะขึ้นอยู่กับขนาด ของจอภาพด้วยเช่นกัน โดยที่จอภาพขนาดใหญ่จะมีการสั่นของภาพได้ง่ายกว่าจอภาพที่มีขนาดเล็ก ฉะนั้น จอภาพขนาดใหญ่ควรใช้อัตราการแสดงภาพใหม่สูงกว่าจอภาพขนาดเล็ก

อย่างไรก็ตามยังมีวิธีการลดการกระพริบของการสแกนภาพหนึ่งแต่ละภาพ ซึ่งจะนิยมใช้วิธีการ สแกนไขว้กัน (Interlace scanning) ตามรูปที่ 2.15 โดยที่ภาพหนึ่งเฟรมจะประกอบด้วยภาพหนึ่ง 2 ฟิลด์ โดยที่เริ่มต้นด้วยการสแกนภาพหนึ่งฟิลด์เส้นที่ก่อน เมื่อเสร็จสิ้นถึงตำแหน่งกลางของแนวเส้นสุดท้าย ลำอิเล็กตรอนก็จะกลับไปเริ่มสแกนใหม่แต่จะสแกนที่ฟิลด์ของเส้นคู่ หลังจากทีสแกนเสร็จทั้ง 2 ฟิลด์แล้ว ก็จะได้ภาพหนึ่งเฟรม



รูปที่ 2.15 การสแกนแบบไขว้กัน

การกวาดสแกนสัญญาณภาพหรือการเริ่มสแกนใหม่ จะเป็นไปตามกระบวนการของสัญญาณ ควบคุมการสแกนทางแนวตั้ง ซึ่งจะแสดงถึงการเริ่มแสดงภาพใหม่หรือเฟรมใหม่ ส่วนสัญญาณควบคุม การสแกนทางแนวนอนจะแสดงถึงการเริ่มใหม่ในแถว สัญญาณควบคุมการสแกนทางแนวตั้งก็จะทำการ รีเซ็ตตำแหน่งลำอิเล็กตรอนไปที่มุมซ้ายบน (จุด (0, 0) นั่นเอง) และเริ่มทำกระบวนการเดิมไปเรื่อย ๆ ในขณะที่กลับไปตำแหน่งเริ่มต้นสัญญาณอาร์ซีบีจะมีค่าเป็น "000" หรือสีดำนั่นเอง แต่ถ้าสัญญาณซิงค์ ทั้ง 2 ค่านี้ไม่สอดคล้องกันแล้ว ส่วนใหญ่ในปัจจุบันจอภาพจะมีแอลอีดีแสดงสถานะ ถ้าแอลอีดีมีสีเขียว แสดงว่ามีการตรวจพบสัญญาณซิงค์ และถ้าแอลอีดีเป็นสีเหลืองแสดงว่าตรวจไม่พบสัญญาณซิงค์ ในการ์ดวีจีเอจะมีหน่วยความจำซึ่งทำหน้าที่เก็บค่าของสีทุกพิกเซล และเวลาการแสดงผลหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้เมื่อถูกอ่านออกไปแล้วจะทำหน้าที่เหมือนเป็นตัวผลิตสัญญาณอาร์จีบีแล้วส่งให้ลำอิลีคตรอนยิงกวาดไปบนฉากร

2.9 องค์ประกอบภาพ

ในภาพหนึ่งภาพจะมีองค์ประกอบมาจากจุดสีด้ามามากมายมาเรียงกันประกอบขึ้นเป็นภาพ ซึ่งจุดเหล่านี้เองที่เรียกว่าองค์ประกอบภาพ (Picture element) หรือเรียกว่าพิกเซล ภาพที่ปรากฏบนทางจอภาพก็ใช้หลักการเดียวกัน ภาพที่เกิดขึ้นบนจอภาพประกอบด้วยเส้นแนวนอนเส้นเล็ก ๆ เป็นจำนวนมาก แต่ละเส้นนั้นมีทั้งส่วนที่ดำสนิทและสว่างรวมกันอยู่ เส้นเหล่านี้เราได้มาจากการกวาดลำแสง ความแตกต่างกับการเส้นกวาดลำแสงหรือเส้นสแกนเหล่านี้เองเราก็จัดว่าเป็นองค์ประกอบภาพ

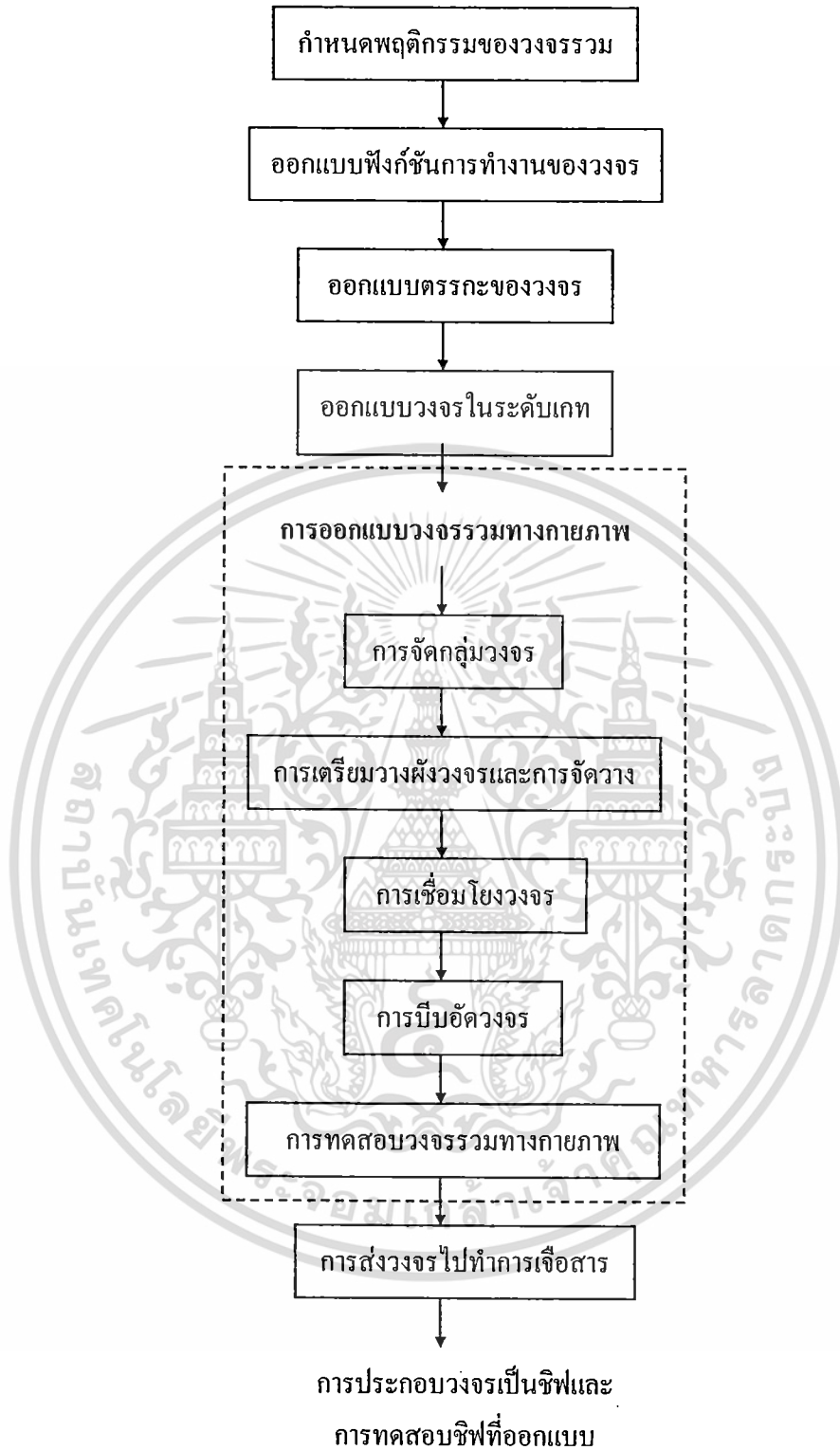
มาตรฐานระบบสแกน 525 เส้น แต่ละเส้นจะแบ่งได้เป็น 800 เส้น ดังนั้นจึงกล่าวได้ว่าหากจะหาองค์ประกอบภาพจะได้ $525 \times 800 = 420,000$ Pixels ใช้แบนวิทท์ 6 MHz ยิ่งภาพมีจำนวนเส้นสแกนมากเท่าไร รายละเอียดของภาพหรือความคมชัดก็จะมากขึ้นเท่านั้น แต่การออกแบบก็จะยากขึ้นด้วยเช่นกัน เนื่องจากภาพที่มีรายละเอียดมากก็ต้องการแบนวิทท์มากขึ้นด้วย จากที่กล่าวมาว่าองค์ประกอบของภาพมีหลายพิกเซล แต่ในความเป็นจริงแล้วเราไม่สามารถมองเห็น ได้ครบทุกพิกเซลเนื่องจากจะมีบางส่วนหายไประหว่างการสแกนภาพ

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

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

2.10 วงจรรวม (ASIC: Application Specific Integrated Circuit)

ความก้าวหน้าของอุตสาหกรรมอิเล็คทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่าง ๆ ซึ่งสามารถลดค่าใช้จ่าย ลดการสิ้นเปลืองขนาดพื้นที่ รวมทั้งพลังงานที่ใช้กับอุปกรณ์และในขณะเดียวกันยังมีการเพิ่มประสิทธิภาพและระดับความน่าเชื่อถือของวงจรที่สูงขึ้นอย่างเห็นได้ชัดจากเทคโนโลยีไมโครโปรเซสเซอร์และหน่วยความจำปัจจุบัน ในการพัฒนาเพิ่มความหนาแน่นและจำนวนฟังก์ชันที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิทัลได้พิจารณาถึงการผลิตในจำนวนมาก ๆ และการผลิตวงจรรวม ซึ่งวงจรรวมจะสามารถแบ่งได้เป็น 2 กลุ่ม คือ ฟิวล์โปรแกรมเมเบิล (Field Programmable) และ แมสโปรแกรมเมเบิล (Mask Programmable)



รูปที่ 2.16 ขั้นตอนการออกแบบวงจรรวม

ในการออกแบบวงจรรวมต้องผ่านขั้นตอนต่าง ๆ ดังรูปที่ 2.16 ซึ่งเริ่มต้นจากการกำหนดการทำงานของระบบใหญ่ของวงจรรวม (System Specification) ซึ่งประกอบระบบ ในแต่ละส่วนย่อยจะถูกเอกสา กำหนดคุณสมบัติเป็นฟังก์ชัน (Functional Design) จากนั้นนำมาทำการออกแบบวงจรตรรกะไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

การผลิตวงจรรวมเฉพาะกิจ แบ่งเป็น 2 ขั้นตอน คือ การออกแบบ และการเจือสาร

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

2.11 การเขียนภาษาวีเอชดีแอล

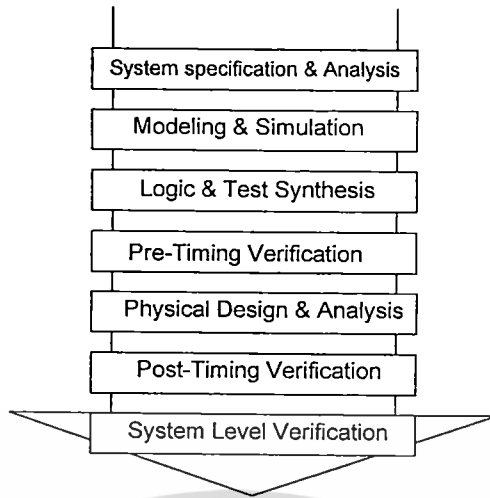
ปัจจุบันในการออกแบบระบบดิจิทัลที่เรารู้จักกันจะเป็นการออกแบบโดยใช้การวาดวงจร (Capture Schematic) โดยใช้โปรแกรมช่วยในการวาด (Schematic entry tools) ซึ่งผู้ออกแบบจะต้องมีทักษะสูงในการออกแบบ และต้องใช้เวลามากในการออกแบบระบบจำลองการทำงาน (Simulation) และตลอดจนถึงการแก้ไขความถูกต้องของระบบ (Debugging) ซึ่งในการออกแบบจะต้องอ้างอิงเทคโนโลยีที่ใช้ออกแบบระบบดิจิทัล (Technology Dependent) ถ้าต้องการเปลี่ยนเทคโนโลยีของระบบที่ออกแบบค่อนข้างทำได้ยากและใช้เวลามาก และเมื่อต้องการออกแบบระบบดิจิทัลที่มีความซับซ้อนสูงยิ่งทำได้ยากหรืออาจทำไม่ได้ โดยใช้กระบวนการออกแบบเก่า ๆ ที่กล่าวมา แต่ในการออกแบบระบบดิจิทัลในปัจจุบันได้มีกระบวนการออกแบบรูปแบบใหม่ที่มีประสิทธิภาพสูง รวดเร็ว และไม่ยึดติดกับเทคโนโลยีที่ใช้ออกแบบ กระบวนการดังกล่าว คือ การออกแบบจากบนลงล่าง (Top-down design) ซึ่งกระบวนการดังกล่าวจะใช้ภาษาบรรยายฮาร์ดแวร์ การออกแบบจำลองการทำงานสังเคราะห์วงจร (Synthesis) ในรูปแบบของเทคโนโลยีที่เราต้องการ และสามารถทดสอบวงจรที่ออกแบบได้บนฮาร์ดแวร์จำพวกชิพเอฟพีจีเอ หรือวงจรรวม ดังนั้นการออกแบบสามารถทำได้ง่ายและมีความสะดวกรวดเร็วมากยิ่งขึ้น

2.11.1 การออกแบบจากบนลงล่าง

ในการพัฒนางจรรวมเชิงเลขขนาดใหญ่ที่มีความซับซ้อน ผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อน จากนั้นจึงวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์ เพื่อให้ได้การทำงานตามที่ต้องการ โดยการออกแบบในลักษณะนี้เรียกว่า หลักการออกแบบจากบนลงล่าง ซึ่งถ้าเปรียบเทียบกับกระบวนการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาในการออกแบบมากกว่าเพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่าง ๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากในการออกแบบลักษณะนี้ ดังนั้นการใช้ภาษาวีเอชดีแอลกับหลักการออกแบบจากบนลงล่างจึงเป็นวิธีการที่เหมาะสมสำหรับการออกแบบและพัฒนางจรรวมที่มีความซับซ้อนมากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่าย

ในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.17 แสดงให้เห็นถึงขั้นตอนการออกแบบจากบนลงล่างทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย โดยขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียด ดังนี้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา
2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบ โดยใช้ภาษาวีเอชดีแอลสำหรับบรรยาย พฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด
3. ขั้นตอนการสังเคราะห์ซึ่งจะต้องทำการกำหนดเทคโนโลยีที่จะมารองรับวงจรที่ออกแบบและระบบช่วยออกแบบจะทำการสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกต (Gate Level) และการเชื่อมต่อกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ตลิสต์ (Net list) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้
4. หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกตหรือเน็ตลิสต์แล้ว ข้อมูลที่ได้นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชันแล้ว ยังมีข้อมูลที่เกี่ยวข้องกับเวลาด้วย ซึ่งจากความจริงที่ว่า อุปกรณ์อิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการเคลื่อนผ่าน (Propagation Delay time) เสมอ ถึงแม้ว่าจะเป็นเวลานี้น้อยมากในระดับนาโนวินาที แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่าง ๆ จำนวน 10,000 เกต ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้นจนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารทำงานในย่านความถี่สัญญาณนาฬิกาสูง ๆ ได้
5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้รับการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของอุปกรณ์เอพฟิซีเอ หรือวงจรรวม
6. หลังจากที่ได้วงจรจริงมาแล้วยังต้องมีความจำเป็นที่จะต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วยเพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่น ๆ ให้เป็นระบบ เพราะในขั้นตอนนี้วงจรที่ออกแบบจะประกอบด้วยอินพุทและเอาต์พุทแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. หลังจากที่น่าวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่น ๆ ให้เป็นระบบแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆอีกครั้ง ซึ่งเป็นการทดสอบการทำงานจริงขั้นสุดท้าย

2.11.2 องค์ประกอบของภาษาวีเอชดีแอล

ส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วย คือ

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

โดยหน่วยการออกแบบที่เป็นพื้นฐานที่ต้องออกแบบเสมอในการออกแบบวงจร นั่นคือ หน่วยการออกแบบเอนทิตี และหน่วยการออกแบบโครงแบบซึ่งทั้งสองหน่วยการออกแบบนี้ จะเรียกว่าคอมโพเนนต์ (Component)

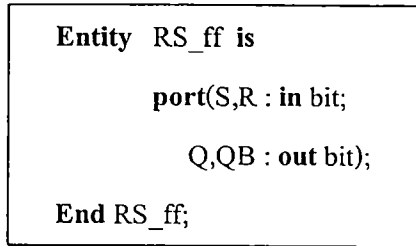
2.11.2.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างอุปกรณ์ภายนอกกับรูปแบบที่เขียนขึ้น รวมทั้งการส่งผ่านค่าพารามิเตอร์บางอย่างระหว่างวงจรถับอุปกรณ์ภายนอกโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางกรไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่าง ๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.18 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบเอนทิตี

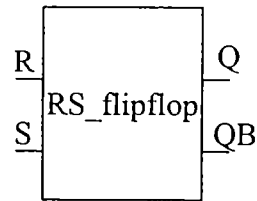
```
Entity component_name is
    Input and Output ports
    Physical and other parameters
End [component_name];
```

รูปที่ 2.18 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

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



(ก) หน่วยการออกแบบแบบเอนทิตี
ในรูปของภาษาวีเอชดีแอล



(ข) มุมมองของตัวเชื่อมประสาน (Interfacing)

รูปที่ 2.19 แสดงรูปแบบของอาร์เอสฟลิปฟลอป

ในรูปที่ 2.19 เป็นหน่วยการออกแบบเอนทิตีที่บรรยายอุปกรณ์ชื่ออาร์เอสฟลิปฟลอปในส่วนหัวของเอนทิตีมีการกำหนดจุดต่อ 4 จุด ภายใต้ชุดคำสั่ง port โดยที่ 2 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ R, S ซึ่งกำหนดด้วยทิศทางการติดต่อกับโลกภายนอกเป็นการไหลเข้าของข้อมูล ส่วนจุดเอาต์พุตเป็นจุดให้ข้อมูลไหลออก ได้แก่ Q, QB ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก ส่วนประเภทของข้อมูลที่จะไหลเข้าและออกนั้นเป็นประเภทบิตที่สามารถมีค่าได้เพียงสองค่าเท่านั้น คือ “0” และ “1” เท่านั้น

ประเภทของพอร์ตที่สามารถประกาศใช้ในเอนทิตี มี 4 ประเภทดังนี้

- พอร์ตอินพุตเข้า

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากอุปกรณ์ภายนอกเข้ามาภายในวงจร สามารถนำมาป้อนให้กับสัญญาณอื่นหรืออ่านค่าได้แต่ไม่สามารถถูกเขียนจากภายในวงจรได้

- พอร์ตอินพุตออก

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากวงจร ส่งออกไปยังอุปกรณ์ภายนอก สามารถเขียนจากภายในวงจรได้ แต่ไม่สามารถอ่านจากภายในวงจรได้

- พอร์ตอินพุตและเอาต์พุต (Bidirectional)

เป็นพอร์ต 2 ทิศทาง ที่สามารถส่งถูกเขียนและอ่านได้จากภายในวงจร

- พอร์ตบัฟเฟอร์ (Output with Internal Feedback)

เป็นพอร์ตเอาต์พุตประเภทหนึ่งที่สามารถอ่านค่ากลับ (Feedback) เข้ามาภายในวงจรได้

2.11.2.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรมคือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงาน พฤติกรรมต่าง ๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ต่าง ๆ ที่กำหนดในหน่วยการออกแบบเอนทิตี รูปที่ 2.20 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบสถาปัตยกรรม

```

Architecture identifier of component_name is
    [declaration]
Begin
    Specification of the functionality
    of the component in terms of its
    input lines and as influenced by
    physical and other parameters
End [identifier];

```

รูปที่ 2.20 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่ง que แสดงให้เห็นว่าหน่วยการออกแบบสถาปัตยกรรมนั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใด ๆ (of <entity design unit> is) ส่วนที่อยู่ระหว่าง Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเพื่อเลือกในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่าง ๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น สัญญาณ ค่าคงที่ โปรแกรมน้อย และอุปกรณ์ ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่งพอร์ต) นั้น จะถูกบรรยายในบริเวณ เนื้อที่ระหว่างคำว่า Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่ง ทุกคำสั่งที่อยู่ภายในบริเวณนี้จะป็นชุดคำสั่งแบบแข่งขันาน (Concurrent statement) เท่านั้น คือทุก ๆ สถานะจะทำงานพร้อมกัน ลำดับก่อนหลังจะไม่มีผลต่อการทำงานของรูปแบบ หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง End และชื่อของสถาปัตยกรรมนั้น ๆ โดยทั่วไปการเขียนรูปแบบโมเดล (Modeling styles) ระบบเชิงเลขด้วยภาษาวีเอชดีแอลสามารถเขียนได้ในลักษณะต่าง ๆ ดังนี้

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

```

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

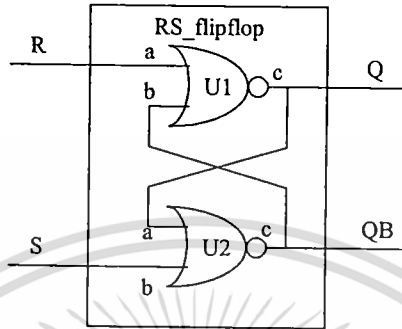
```

รูปที่ 2.21 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟล็อปตามฟังก์ชันบูลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Q = \overline{QB + R} \text{ และ } QB = \overline{Q + S} \quad (2.37)$$

รูปที่ 2.29 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (R, S) กับข้อมูลที่ไหลออก (Q, QB) ประกอบด้วยชุดคำสั่งแบบแข่งขันานาน 2 ชุด ซึ่งเขียนเป็นประเภทการไหลของข้อมูล หรือเรียกว่า ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL : Register Transfer Level)



รูปที่ 2.22 แสดงโครงสร้างภายในสถาปัตยกรรมของอาร์เอสฟลิปฟลอป

รูปที่ 2.30 เป็นหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะโครงสร้าง ซึ่งเปรียบเสมือนการนำอุปกรณ์ที่มีอยู่ในไลบรารีมาต่อเป็นวงจรตามต้องการโดยใช้อินพุต 2 อินพุต (nor2) จำนวนสองตัวมาสร้างตามฟังก์ชันบูลีน

```

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

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

End struc;
    
```

รูปที่ 2.23 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟลอปในลักษณะ โครงสร้าง

```

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

```

รูปที่ 2.24 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟล็อปในลักษณะพฤติกรรม

รูปที่ 2.24 เป็นการเขียนบรรยายการทำงานของรูปแบบในลักษณะพฤติกรรม ซึ่งจะเห็นได้ว่ามีลักษณะที่เหมือนกับการเขียนโปรแกรมทั่วไป โดยจะต้องมีการใช้งานส่วนที่เรียกว่า process และการทำงานของรูปแบบจะขึ้นอยู่กับ การเปลี่ยนแปลงของสิ่งที่อยู่ภายใน process (อินพุต R, S) ซึ่งเรียกว่า เซนซิวิตีลิสต์ (Sensitivity list) การเขียนในลักษณะนี้ลำดับก่อนหลังของชุดคำสั่งจะมีผลต่อการทำงานของรูปแบบที่เขียนขึ้น

```

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

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

```

รูปที่ 2.25 แสดงหน่วยการออกแบบสถาปัตยกรรมของอาร์เอสฟลิปฟล็อปในลักษณะผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าจะเขียนบรรยายส่วนของสถาปัตยกรรมของอาร์เอสฟลิปฟล็อปในลักษณะของพฤติกรรม การไหลของข้อมูล โครงสร้างหรือผสมที่นำเอาแต่ละลักษณะมาเขียนไว้ในส่วนของสถาปัตยกรรมก็ตาม ต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งถือว่าเป็นข้อดีของ ภาษาวีเอชดีแอล

การออกแบบวงจรดิจิทัลใด ๆ จะต้องออกแบบหน่วยการออกแบบพื้นฐานก็คือ ส่วนของเอนทิตี และส่วนของหน่วยของสถาปัตยกรรมที่มีความสัมพันธ์กัน ซึ่งมีด้วยกันหลายรูปแบบสำหรับการ ออกแบบวงจรเดียวกัน

2.11.2.3 หน่วยการออกแบบแพ็คเกจ

ข้อมูลต่าง ๆ ตลอดจนโปรแกรมย่อย เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบเชิงเลข สามารถเก็บไว้ในส่วนของแพ็คเกจได้และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดยหน่วยการออกแบบ เอนทิตีหน่วยการออกแบบสถาปัตยกรรมหรือจากหน่วยการออกแบบแพ็คเกจอื่น ๆ โดยปกติแล้วแพ็คเกจ จะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็คเกจ (Package Declaration) และส่วนของบอดีแพ็คเกจ (Package Body) เนื่องจากแพ็คเกจถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็คเกจไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอล สามารถกระทำได้ด้วยชุดคำสั่ง USE โดยสิ่งที่สามารถสร้างไว้ใน Package ได้แก่

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

ส่วนการประกาศแพ็คเกจ

เป็นส่วนที่มีความสำคัญที่สุดของแพ็คเกจ (ถ้ามองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ ส่วนการประกาศแพ็คเกจ เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็คเกจสำหรับ นำไปใช้ภายนอกตัวของแพ็คเกจเองสิ่งใด ๆ ที่ถูกประกาศไว้ในส่วนของบอดีแพ็คเกจแต่ไม่ได้ ถูกประกาศไว้ในส่วนการประกาศแพ็คเกจจะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือจุดเชื่อมต่อหรือพอร์ท ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้น โดยทั่วไปแล้วแพ็คเกจสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมี ส่วนบอดี (Package body) และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูล (Type) หรือสัญญาณ (Signal) ในทางกลับกันกับส่วนบอดีแพ็คเกจที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็คเกจ แต่แพ็คเกจนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

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

รูปที่ 2.26 แสดง โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็คเกจ

บอดีแพ็คเกจ

โครงสร้างที่ประกอบด้วยคำสั่งต่าง ๆ ในรูปของคำสั่งลำดับ (Sequential statement) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อยทั้งหลายที่ชื่อของโปรแกรมย่อยนั้น ๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็คเกจแล้วจะถูกเก็บไว้ในส่วนบอดีแพ็คเกจซึ่งรวมทั้งการกำหนดค่าที่ต่าง ๆ อันได้แก่ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็คเกจแต่ถูกกำหนดค่าในส่วนของบอดีแพ็คเกจ ฉะนั้นส่วนบอดีแพ็คเกจจึงไม่จำเป็นต้องมีถ้าในส่วนของการประกาศแพ็คเกจไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่ การเขียนบอดีแพ็คเกจนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.35

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

รูปที่ 2.27 แสดง โครงสร้างโดยทั่วไปของบอดีแพ็คเกจ

2.11.2.4 หน่วยการออกแบบโครงสร้าง

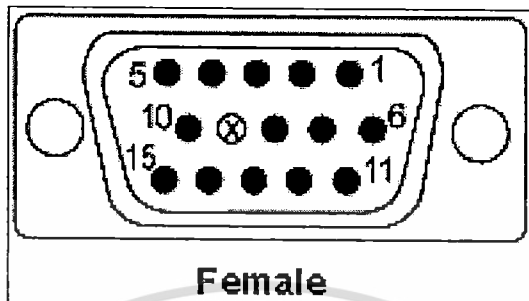
สิ่งที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิทัลไม่ว่าจะเป็นอะไรจะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งหน่วยเท่านั้นแต่ในขณะที่หน่วยการออกแบบเอนทิตีหนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบโครงสร้างมาเพื่อกำหนดการใช้โครงสร้าง ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

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

รูปที่ 2.28 แสดง โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงสร้าง

2.12 หัวเชื่อมต่อกับหน้าจอดีวีเอ (VGA Connector: DE-15)

พอร์ตวีดีโอ (Video Port) ซึ่งเป็นพอร์ตสำหรับต่อสายสัญญาณภาพกับจอคอมพิวเตอร์ ลักษณะของพอร์ตจะเป็นพอร์ตแบบตัวเมีย มีรู 15 รู ดังรูปที่ 2.29



รูปที่ 2.29 หัวเชื่อมต่อกับหน้าจอดีวีเอ

ตารางที่ 2.2 คุณสมบัติต่าง ๆ ของหัวเชื่อมต่อกับหน้าจอดีวีเอ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตารางแสดงค่าที่จะเลือกใช้ของบิตต่างๆ

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

2.13 รอม (ROM: Read Only Memory)

รอม คือ หน่วยความจำชนิดหนึ่งที่มีโปรแกรมหรือข้อมูลอยู่แล้ว และพร้อมที่จะนำมาต่อกับไมโครโปรเซสเซอร์ได้โดยตรง ซึ่งโปรแกรมหรือข้อมูลนี้จะไม่สูญหายไป แม้ว่าจะไม่มีการจ่ายไฟเลี้ยงให้กับระบบ ข้อมูลที่เก็บอยู่ในรอมจะสามารถอ่านค่าออกมาได้ แต่ไม่สามารถเขียนข้อมูลเข้าไปได้ เว้นแต่จะใช้วิธีการพิเศษซึ่งขึ้นกับชนิดของรอม ซึ่งชนิดของรอมสามารถแบ่งออกได้เป็น

2.13.1 แมนนวลรอม (Manual ROM)

ข้อมูลทั้งหมดที่อยู่ในรอมจะถูกโปรแกรมโดยผู้ผลิต (โปรแกรม มาจากโรงงาน) เราจะใช้รอมชนิดนี้เมื่อข้อมูลนั้นไม่มีการเปลี่ยนแปลง และมีความต้องการใช้งานเป็นจำนวนมาก ผู้ใช้ไม่สามารถเปลี่ยนแปลงข้อมูลภายในรอมได้

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

2.13.2 พรอม (PROM: Programmable Read-Only Memory)

ข้อมูลที่ต้องการโปรแกรมจะถูกโปรแกรมโดยผู้ใช้เอง โดยป้อนพัลส์แรงดันสูง (High Voltage Pulsed) ทำให้แผ่นโลหะหรือโพลีคริสตอลซิลิกอน (Polycrystalline Silicon) ที่อยู่ในตัวไอซีขาดออกจากกัน ทำให้เกิดเป็นลอจิก '1' หรือ '0' ตามตำแหน่งที่กำหนดในหน่วยความจำนั้น ๆ เมื่อพรอมถูกโปรแกรมแล้ว ข้อมูลภายในจะไม่สามารถเปลี่ยนแปลงได้อีก หน่วยความจำชนิดนี้จะใช้ในงานที่ใช้ความเร็วสูง ซึ่งสูงกว่าหน่วยความจำที่โปรแกรมได้กว่าหน่วยความจำชนิดอื่น ๆ

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

ข้อมูลจะถูกโปรแกรมโดยผู้ใช้ โดยการให้สัญญาณที่มีแรงดันสูงผ่านเข้าไปในตัวอีพรอม ซึ่งเป็นวิธีเดียวกันกับที่ใช้ในพรอม แต่ข้อมูลที่อยู่ในอีพรอมเปลี่ยนแปลงได้ โดยการลบข้อมูลที่อยู่ในอีพรอมออกก่อน แล้วค่อยโปรแกรมเข้าไปใหม่ การลบข้อมูลนี้ทำได้ด้วยการฉายแสงอัลตราไวโอเล็ตเข้าไปใน

เอกสารนี้เป็นเอกสารของบริษัทเอกชนที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำข้อมูลไปเผยแพร่หรือใช้เพื่อวัตถุประสงค์อื่นได้โดยไม่ได้รับอนุญาตจากบริษัทฯ

ตัวไอซีโดยผ่านทางกระแสไฟฟ้าที่อยู่บนตัวไอซี เมื่อฉายแสงครู่หนึ่ง (ประมาณ 5-10 นาที) ข้อมูลที่อยู่ภายในก็จะถูกลบทิ้ง ซึ่งช่วงเวลาฉายแสงนี้สามารถดูได้จากค่าชิทมาทากับตัวไอซีพร้อมและมีความเหมาะสมที่จะใช้ เมื่องานของระบบมีโอกาสที่จะปรับปรุงแก้ไขข้อมูลใหม่

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

อีเอพรอมหรือเรียกอีกชื่อหนึ่งว่าอีอีพรอม (EEPROM: Electrical Erasable EPROM) เนื่องจากมีการใช้ไฟฟ้าในการลบข้อมูลในรอมเพื่อเขียนใหม่ ซึ่งใช้เวลาสั้นกว่าของอีพรอม

การลบขึ้นอยู่กับพื้นฐานการใช้เทคโนโลยีที่แตกต่างกัน ดังนั้นอีเอพรอมจะอยู่บนพื้นฐานของเทคโนโลยีแบบเอ็นมอส ข้อมูลจะถูกโปรแกรมโดยผู้ใช้เหมือนในอีพรอมแต่สิ่งที่แตกต่างก็คือ ข้อมูลของอีเอพรอมสามารถลบได้โดยทางไฟฟ้าไม่ใช่โดยการฉายแสงแบบอีพรอม

2.14 แรม (RAM: Random-Access Memory)

แรม (RAM: Random Access Memory หน่วยความจำเข้าถึงโดยสุ่ม หรือหน่วยความจำชั่วคราว) เป็นหน่วยความจำหลักที่ใช้ในระบบคอมพิวเตอร์ยุคปัจจุบัน หน่วยความจำชนิดนี้ อนุญาตให้เขียนและอ่านข้อมูลได้ในตำแหน่งต่างๆ อย่างอิสระ และรวดเร็วพอสมควร ซึ่งต่างจากสื่อเก็บข้อมูลชนิดอื่นๆ อย่างเทป หรือดิสก์ ที่มีข้อจำกัดในการอ่านและเขียนข้อมูล ที่ต้องทำตามลำดับก่อนหลังตามที่จัดเก็บไว้ในสื่อ หรือมีข้อจำกัดแบบรอม ที่อนุญาตให้อ่านเพียงอย่างเดียว

ข้อมูลในแรม อาจเป็นโปรแกรมที่กำลังทำงาน หรือข้อมูลที่ใช้ในการประมวลผล ของโปรแกรมที่กำลังทำงานอยู่ ข้อมูลในแรมจะหายไปทันที เมื่อระบบคอมพิวเตอร์ถูกปิดลง เนื่องจากหน่วยความจำชนิดนี้ จะเก็บข้อมูลได้เฉพาะเวลาที่มีกระแสไฟฟ้าหล่อเลี้ยงเท่านั้น

2.14.1 ดีแรม (DRAM: Dynamic Random Access Memory)

ดีแรม คือ ไดนามิก ซึ่งตรงกันข้ามกับ SRAM หรือ สแตติกแรม (Static RAM); DRAM ต้องการการรีเฟรช (Refresh) เซลเก็บข้อมูล หรือการประจุไฟ (Charge) ในทุกๆ ช่วงมิลลิวินาที DRAM จะเก็บทุกๆ บิตในเซลล์เก็บข้อมูล ซึ่งประกอบขึ้นด้วยกลุ่มตัวเก็บประจุ (Capacitor) และทรานซิสเตอร์ (Transistor) คาปาซิเตอร์จะสูญเสียประจุไฟอย่างรวดเร็ว และนั่นเป็นเห็นเหตุผลว่า ทำไม DRAM จึงต้องการการรีเฟรช

2.14.2 เอสแรม (SRAM: Static Random-Access Memory)

เอสแรม ย่อมาจากคำว่า Static RAM (สแตติกแรม) เป็นหน่วยความจำประเภท RAM ที่พัฒนามาจากทรานซิสเตอร์ มีคุณสมบัติสามารถเก็บรักษาข้อมูลไว้ในหน่วยความจำถาวรเท่าที่ยังมีไฟเลี้ยง อยู่ และทำงานได้ที่ไม่ต้องการเขียนข้อมูลลงไปซ้ำหลายครั้งเพื่อรักษาข้อมูลภายใน (เรียกว่า รีเฟรช - Refresh) อยู่ตลอดเวลาเหมือน DRAM มีความเร็วในการเข้าถึงข้อมูลที่สูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

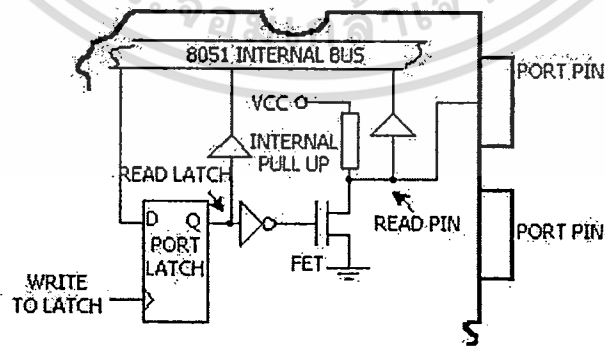
2.15 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์จึงถูกเลือก ใช้ในการทดลองด้วยเหตุผลในการใช้อุปกรณ์ที่ต่อรวมในวงจรมีน้อยชิ้น ราคาถูก หาแหล่งข้อมูลได้ง่ายมีการพัฒนาประสิทธิภาพ ในการประมวลผล เทียบเท่าไมโครโพรเซสเซอร์ขนาด 8 บิต -16บิต และ ใช้ในการพัฒนากับงานที่ไม่ต้องการความซับซ้อนซึ่งจะทำให้มีความสะดวกมากขึ้น ปัจจุบันมีไอซีของสองบริษัทที่เรานิยมใช้ในงานออกแบบ และสำหรับในการเรียนการสอนในสถาบันหลายแห่ง คือ ไอซีไมโครคอนโทรลเลอร์ของบริษัท Microchip ตระกูล 16FXXX และ ไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 จากหลายบริษัทที่ผลิตออกมามากมายหลายเบอร์ แต่คำสั่งจะใกล้เคียงกัน ดังนั้น การทดลองในหนังสือเล่มนี้จึงเลือกใช้งานไอซี ไมโครคอนโทรลเลอร์ของตระกูล MCS-51 ด้วยเหตุผลเพราะมีข้อมูล และตัวอย่างที่หลากหลายหามาใช้งานได้ง่าย แต่ก็ไม่ได้ตัดสินว่าดีกว่า ไอซีในตระกูลอื่นๆ ดังนั้น ในการออกแบบชุดทดลองที่ใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ของ MCS-51 ที่จะแนะนำให้สร้างขึ้นนี้ สามารถที่จะนำมาใช้งานร่วมกับไอซีไมโครคอนโทรลเลอร์ได้ทุกเบอร์ และทุกตระกูล

หน่วยความจำของไมโครคอนโทรลเลอร์ 8051 แบ่งเป็น 2 ประเภท คือ

1. หน่วยความจำโปรแกรม (Program Memory) หน่วยความจำประเภทนี้ คือ ROM ใช้เก็บโปรแกรมที่ใช้ในการควบคุมระบบ ซึ่งเป็นหน่วยความจำประเภท non-volatile
2. หน่วยความจำข้อมูล (Data Memory) หน่วยความจำประเภทนี้ได้แก่ RAM โครงสร้างของพอร์ต

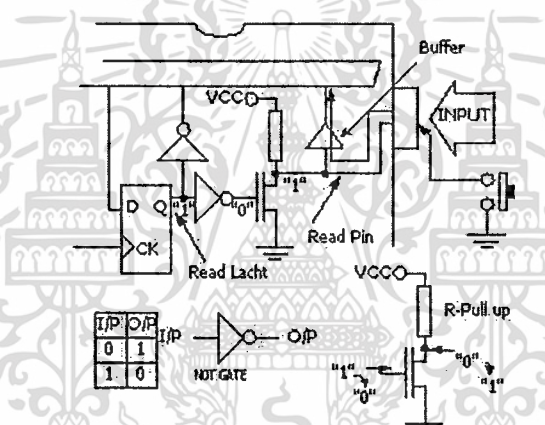
การที่จะให้ไมโครคอนโทรลเลอร์ติดต่อกับอุปกรณ์ภายนอกได้ เช่นการรับข้อมูลจากอุปกรณ์ภายนอก (INPUT) หรือการส่งข้อมูลให้กับอุปกรณ์ภายนอก (OUTPUT) ก็จะต้องติดต่อผ่านพอร์ต (PORT) หรืออาจกล่าวได้ว่าพอร์ตคือช่องทางสำหรับการโอนย้ายข้อมูลระหว่างไมโคร คอนโทรลเลอร์กับอุปกรณ์ภายนอกนั่นเอง



รูป 2.30 แสดงโครงสร้างภายในของพอร์ต

2.15.1 การใช้งานไมโครคอนโทรลเลอร์ให้เป็นพอร์ตอินพุต (INPUT PORT)

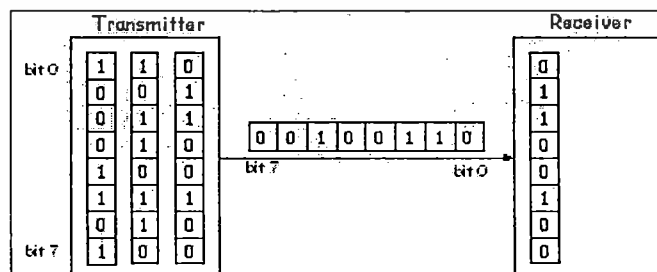
การกำหนดให้พอร์ตหรือบิตใดๆ ของไมโครคอนโทรลเลอร์เป็นพอร์ตอินพุต หรือส่วนในการรับค่าของข้อมูลเข้ามามีแสดงในรูป 5.2 จะเริ่มต้นด้วยการเขียนข้อมูลให้กับพอร์ตหรือบิตนั้นๆ เป็นสถานะลอจิก "1" แล้วจึงส่งไปแต่ละบิตของพอร์ตที่ต้องการใช้งาน เพื่อให้เป็นพอร์ตอินพุต เช่น ใช้คำสั่ง MOV P3, #0FFH หรือคำสั่ง SETB P1.5 จะทำให้วงจรส่วนของการคงสถานะข้อมูล หรือวงจรแลตช์ (Latch)* ซึ่งสร้างมาจากวงจร ดี ฟลิป-ฟล็อป (D flip-flop) จะให้เอาต์พุต Q มีสถานะลอจิกเป็น "1" แล้วจะผ่านไปที่วงจรกลับสัญญาณ (Inverter) เพื่อให้หยุดการทำงานของเฟต (FET)* ซึ่งจะทำให้ขาสัญญาณของพอร์ตถูกเชื่อมต่อเข้ากับวงจรพูลอัพ * ที่เป็นตัวต้านทานภายใน (Internal Pull up) โดยตรง ส่งผลให้ขาพอร์ตนั้นมีสถานะลอจิกเป็น "1" และสามารถจะรับสัญญาณที่เป็นลอจิก "0" จากอุปกรณ์ภายนอกได้ ข้อมูลจากอุปกรณ์ภายนอกที่อยู่ก็จะถูกส่งเข้ามาเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต เพื่อรอให้ซีพียูมาทำการอ่านค่าเข้าไป ดังนั้นอุปกรณ์ภายนอกที่เราจะนำมาเชื่อมต่อกับพอร์ตอินพุต จึงนิยมกำหนดให้ทำงานในสถานะลอจิก "0"



รูป 2.31 แสดงการกำหนดเป็นพอร์ตอินพุต โดยให้สถานะเป็น "1" ตำแหน่งบิตที่ต้องการ

2.16 การสื่อสารแบบอนุกรม (Serial Communication)

เป็นการสื่อสารโดยการส่งข้อมูลที่ละบิต ผ่านสายสัญญาณเส้นเดียว จนครบทั้ง 8 บิต หรือไปที โดยจะส่งบิตต่ำ (LSB) ออกไปก่อน สามารถแสดงให้เห็นหลักการส่งข้อมูลแบบอนุกรมได้ดังรูปที่ 2.32



รูปที่ 2.32 การส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.1 รูปแบบของการสื่อสารแบบอนุกรมและอัตราบอดการสื่อสาร

อัตราบอด (Baud Rate) คือความเร็วในการรับ-ส่งข้อมูลอนุกรมมีหน่วยเป็นบิตต่อวินาที ซึ่งจะบอกจำนวนบิตที่รับ-ส่งในเวลา 1 วินาที เช่น ส่งข้อมูลด้วยอัตรา 9600 บิตต่อวินาที ก็คือการส่งข้อมูลตัวอักษรขนาด 10 บิต (บิต Start 1 บิต บิตข้อมูล 8 บิต บิต Stop 1 บิต) ได้ 960 ตัวอักษรใน 1 วินาที ซึ่งตารางที่ 2.10.1 แสดงอัตราบอดของ UART ที่ใช้กันทั่ว ตารางที่ 2.4 เป็นการแสดงอัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

ตารางที่ 2.4 อัตราบอดที่ใช้กันทั่วไป

อัตราบอด	ช่วงเวลาของแต่ละบิต
110	9.91 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	0.833 ms
2400	0.417 ms
4800	0.208 ms
9600	0.104 ms
19200	0.52

2.16.2 การอินเทอร์เฟสตามมาตรฐาน RS-232

มาตรฐาน RS-232 เป็นมาตรฐานที่ได้รับการพัฒนามานานและถูกใช้งานกันอย่างแพร่หลาย เราใช้ RS-232 เชื่อมต่อ DTE (Data Terminal Equipment) เข้ากับ DCE (Data Communication Equipment) เช่น การต่อเทอร์มินัลเข้ากับโมเด็ม มาตรฐาน RS-232 กล่าวถึงลักษณะทางกล, ลักษณะของสัญญาณไฟฟ้าและลักษณะการทำงานที่ใช้การอินเทอร์เฟส ตัวอย่างของอุปกรณ์ที่ใช้ในการอินเทอร์เฟสตามมาตรฐาน RS-232 ได้แก่ เทอร์มินัล, พล็อตเตอร์, ลอจิกอานาไลเซอร์ (Logic Analyzer) และเครื่องพิมพ์ ถ้าการประยุกต์ใช้งานของเราต้องการทำอินเทอร์เฟสอุปกรณ์เข้ากับอินเทอร์เฟสมาตรฐาน RS-232 เราจำเป็นต้องแปลงระดับสัญญาณ TTL ให้เป็นระดับสัญญาณแบบอื่น

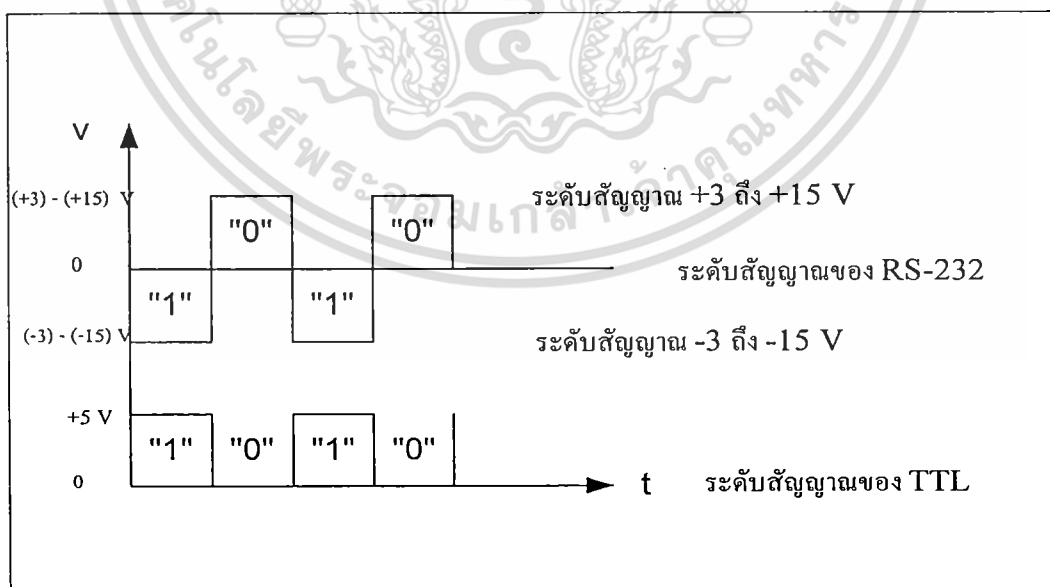
2.16.3 ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟซ

มาตรฐาน RS-232 ใช้สัญญาณเพียงเส้นเดียวในการส่งสัญญาณ โดยสัญญาณจะส่งไปในทิศทางเดียวกัน ในกรณีที่อัตราเร็วในการส่งข้อมูลมีค่าเท่ากับ 20 กิโลบิตต่อวินาที (kbps) ซึ่งค่านี้เป็นค่าสูงสุดที่ใช้ในการสื่อสารข้อมูล (ในปัจจุบันพัฒนาให้สามารถส่งข้อมูลได้มากกว่านี้) ระยะทางในการส่งข้อมูลไม่ควรเกิน 50 ฟุต (ตามข้อกำหนดในมาตรฐาน) สำหรับการแทนแรงดันของระดับสัญญาณจะแทนระดับสัญญาณของลอจิก "0" ด้วยค่าแรงดัน +3 โวลต์ ถึง +12 โวลต์ ส่วนลอจิก "1" จะแทนระดับสัญญาณด้วยค่าแรงดันระหว่าง -3 โวลต์ ถึง -12 โวลต์

2.16.4 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232 (x) เป็นมาตรฐานอุตสาหกรรม โดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส (Asynchronous) 2 ทิศทาง เพื่อให้มีการใช้งานในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่าง ๆ การรับส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุต โดยมีระดับ สัญญาณตั้งแต่ 3 โวลต์ จนถึง 15 โวลต์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลต์ จนถึง -15 โวลต์ สำหรับลอจิก "1" ดังแสดงในรูป 2.33

ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิทัลทั่ว ๆ ไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 - 5 โวลต์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า -3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจจะต่อวงจรจากทรานซิสเตอร์ได้

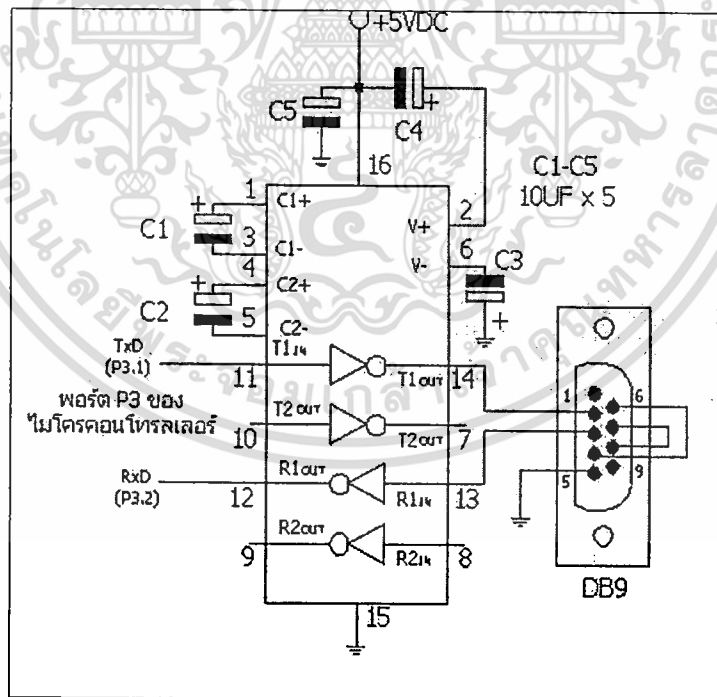
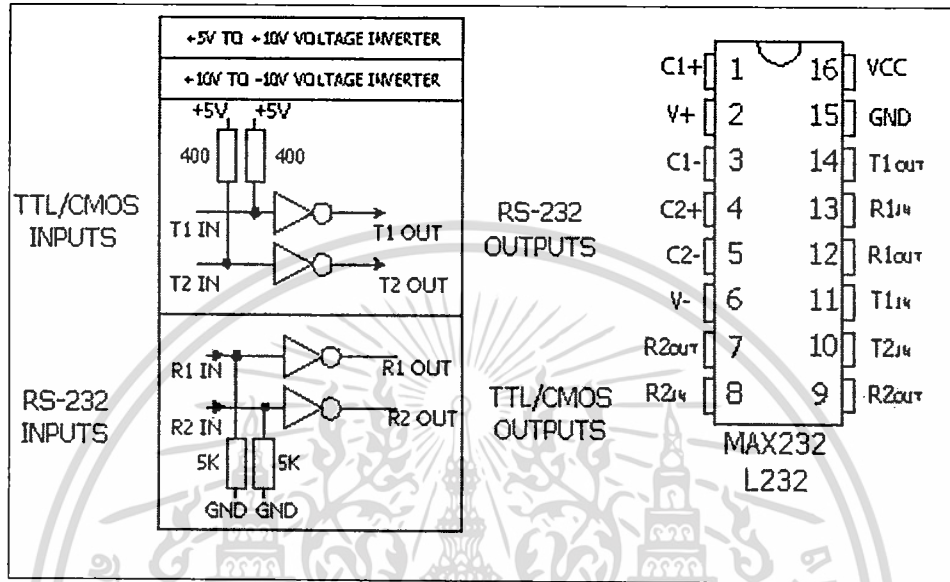


รูปที่ 2.33 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.5 ไอซี MAX232, L232

ไอซี MAX232, L232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้

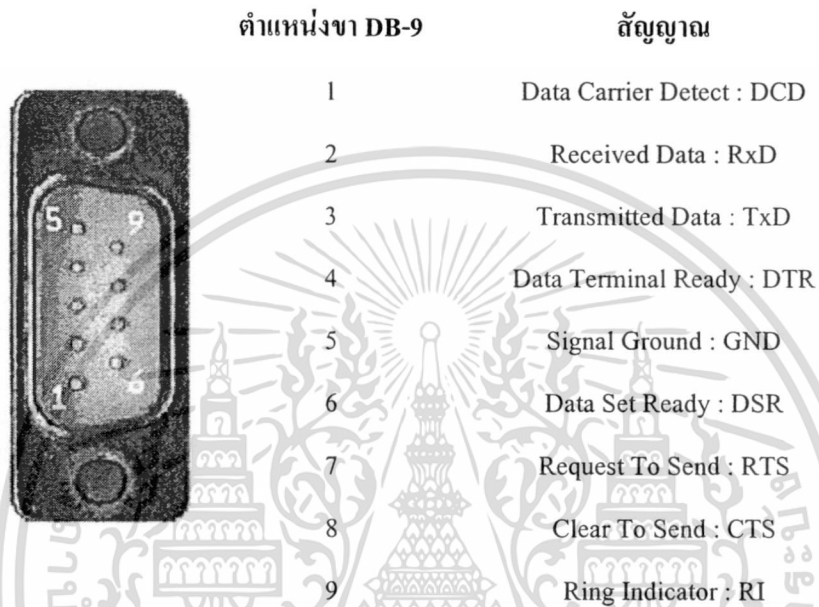


รูปที่ 2.34 แสดงตำแหน่งขาของไอซี MAX232, L232 และการต่อใช้งาน

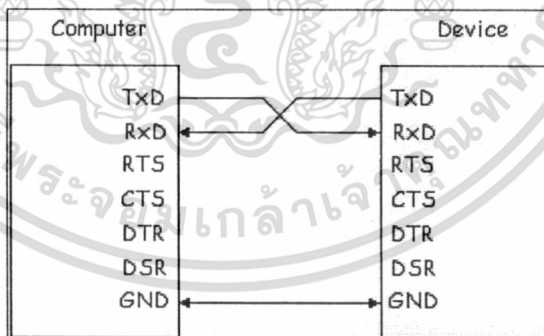
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.6 การออกแบบตัวแปลงสัญญาณ

การเชื่อมต่อกับพอร์ตสื่อสารของคอมพิวเตอร์ส่วนบุคคลจะเลือกใช้พอร์ตสื่อสารแบบอนุกรม 9 ขา (DB-9) ซึ่งสามารถทำการรับส่งข้อมูลได้แบบอนุกรม โดยลักษณะของสัญญาณจะเป็นไปตามมาตรฐาน RS-232 โดยลักษณะของการเชื่อมต่อของพอร์ตสื่อสารสำหรับคอนเน็คเตอร์ (connector) แบบ DB-9 สามารถแสดงให้เห็นได้ดังรูปที่ 2.35 และรูปที่ 2.36



รูปที่ 2.35 การจัดขาของ connector พอร์ตอนุกรมแบบ DB-9



รูปที่ 2.36 การต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น

2.17 โมดูลตรวจจับภาพ

- ใช้โมดูลกล้องทุกทิศทางซีมอส รุ่น OV6620
- ใช้ไมโครคอนโทรลเลอร์ SX28AC ความเร็วสูงในการจัดการสัญญาณภาพเป็นข้อมูลอนุกรม เพื่อให้ไมโครคอนโทรลเลอร์ขนาดเล็กสามารถทำงานกับสัญญาณหรือรูปภาพได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ติดต่อแบบอนุกรมทั้งในระดับที่ทีแอล และมาตรฐาน RS-232 สามารถเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรมได้ เลือกรับส่งข้อมูลอนุกรมได้ 4 ค่าคือ 115200, 38400, 19200, 9600 บิตต่อวินาที
- ตรวจจับภาพสีได้ด้วยอัตราเร็ว 17 เฟรมต่อวินาที
- สามารถค้นหาจุดศูนย์กลางภาพได้
- ให้ข้อมูลค่าเฉลี่ยสีแดง, เขียว และน้ำเงิน เพื่อแยกแยะความแตกต่างได้
- ถ่ายทอดข้อมูลจุดภาพแบบไบนารีเวลาจริงของแต่ละพิกเซลที่ตรวจจับได้ในภาพ
- สามารถปรับแต่งคุณสมบัติภาพของกล้องได้
- สามารถอ่านข้อมูลดิบของภาพออกมาได้
- ความละเอียดของภาพที่ตรวจจับได้คือ 80 x 143 จุด

โมดูลตรวจจับภาพ iV-CAM ใช้สำหรับตรวจจับภาพ ให้ข้อมูลของภาพที่ตรวจจับได้นั้นเป็นข้อมูลของแม่สีหลัก คือ สีแดง สีเขียว สีน้ำเงิน โดยสามารถให้ข้อมูลรายละเอียดในแต่ละจุดภาพหรือพิกเซล จึงสามารถนำข้อมูลเหล่านั้นมาประกอบกันเพื่อสร้างการจดจำรูปร่างของวัตถุได้

2.18 การตรวจสอบคุณภาพของภาพ

ในการเปรียบเทียบผลของการปรับปรุงคุณภาพของภาพนั้น ต้องมีมาตรการในการควบคุมคุณภาพของภาพ โดยจะใช้สองมาตรการคือ *Mean-Squared Error* และ *Peak Signal-to-Noise Ratio* โดยที่ค่า *Mean-Squared Error* (MSE) ระหว่างภาพ 2 ภาพ คือ $g(x, y)$ และ $\hat{g}(x, y)$ คือ

$$e_{MSE} = \frac{1}{MN} \sum_{n=1}^M \sum_{m=1}^N [\hat{g}(n, m) - g(n, m)]^2 \quad (2.38)$$

ปัญหาสำคัญปัญหาหนึ่งของ การหาค่า *Mean-Squared Error* คือ ระดับความเข้มของภาพ ก็คือ ในการหาค่า *Mean-Squared Error* ที่ 100.0 สำหรับภาพ 8 บิต (ค่าพิกเซลอยู่ในช่วง 0-255) ดูไม่น่าจะเป็นไปได้ ในการหาค่า *Mean-Squared Error* ที่ 100.0 สำหรับภาพ 10 บิต (ค่าพิกเซลอยู่ในช่วง 0-1023) ก็เกือบจะหาค่าไม่ได้

psnr เลี่ยงปัญหานี้โดยการ scaling ที่ mse ตามช่วงของภาพ

$$PSNR = -10 \log_{10} \frac{e_{MSE}}{S^2} \quad (2.39)$$

เมื่อ s คือค่าพิกเซลสูงสุด psnr มีค่าอยู่ในหน่วยเดซิเบล (DB) ค่า psnr นั้นยังไม่เป็นค่าในอุดมคติ แต่ก็ยังใช้ในทางปฏิบัติสิ่งที่เป็นหลักที่นำไปใช้ไม่ได้คือความแข็งแรงของสัญญาณนั้นถูกประเมินเป็น S^2 มากกว่าสัญญาณที่แท้จริงสำหรับภาพ PSNR เป็นการวัดที่ดีสำหรับการเปรียบเทียบผลของการซ่อมแซมภาพสำหรับภาพเดียวกัน แต่ระหว่างภาพที่ถูกเปรียบเทียบด้วย psnr นั้น ไม่มีความหมายที่ถูกต้องมากนัก ภาพที่มี PSNR 20 DB อาจจะถูกดีกว่าภาพอื่นที่มีค่า PSNR 30 DB ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

โครงการนี้แบ่งออกได้เป็นสองส่วนหลักคือ

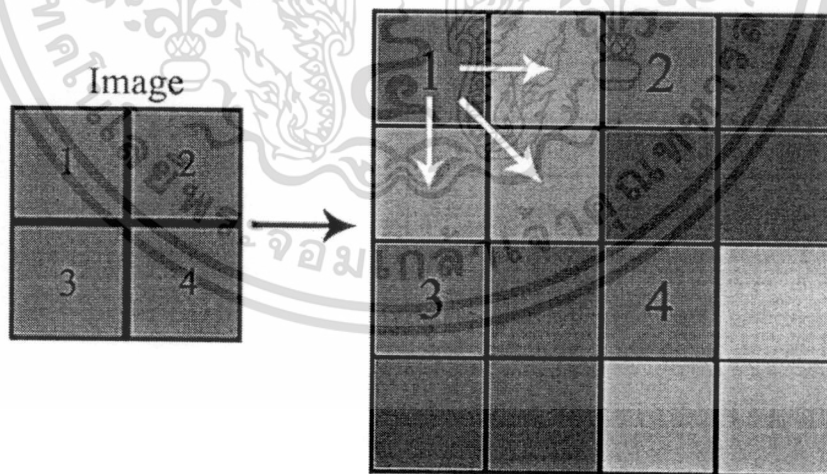
- การอินเทอร์โพลेटโดยใช้โปรแกรมเมทแลปในการช่วยประมวลผล
- การอินเทอร์โพลेट โดยใช้ FPGA ในการประมวลผล

โดยในส่วนของเมทแลปนั้นเป็นการศึกษาวิธีการนำข้อมูล เพื่อมาใช้ในการอินเทอร์โพลิตชัน ซึ่งในโครงการนี้ได้เลือกวิธีการทำอินเทอร์โพลิตชันแบบต่างๆกัน ดังนี้

1. Replication Pixel
2. Bilinear Interpolation
3. Fast Fourier Transform Based Interpolation
4. Discrete Pascal Transform Based Interpolation

3.1 Replication Pixel

วิธีการนี้เป็นการนำพิกเซลต้นแบบ มาทำการคัดลอก และวางไว้ที่บริเวณข้างเคียงของพิกเซลที่ทำการคัดลอก เพื่อให้ภาพมีขนาดใหญ่ขึ้น ดังตัวอย่างที่แสดงในรูปข้างล่างต่อไปนี้



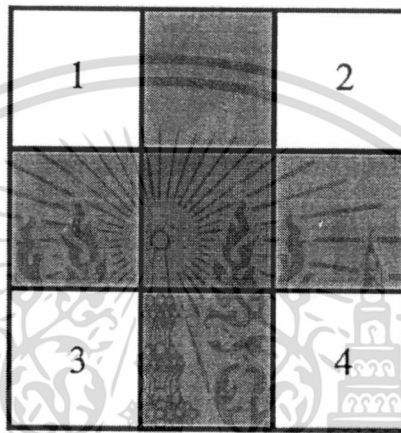
รูปที่ 3.1 แสดงการขยายด้วยวิธีการ Replication Pixel

3.2 Bilinear Interpolation

วิธีการนี้ จะทำการขยายข้อมูลด้วยวิธีการนำค่าบริเวณข้างเคียง(4 จุดพิกเซลที่ติดกันในข้อมูลภาพ) มาทำการคำนวณ เพื่อหาค่าสัมประสิทธิ์ ด้วยสมการ

$$Ax+By+Cxy+D = Z(x,y)$$

โดยที่ค่า x และ y คือจุดพิกัดของภาพในแต่ละพิกเซล และค่า Z คือค่าของสีในพิกเซลนั้นๆ ซึ่งเมื่อทำการแทนค่าลงไปจะได้สมการ 4 สมการ (ซึ่งเกิดจาก 4 พิกเซล) 4 ตัวแปร ทำให้เราสามารถแก้สมการเพื่อหาค่าสัมประสิทธิ์คือ A B C และ D ได้ และนำค่าที่ได้เพื่อมาคำนวณค่าในจุดใหม่



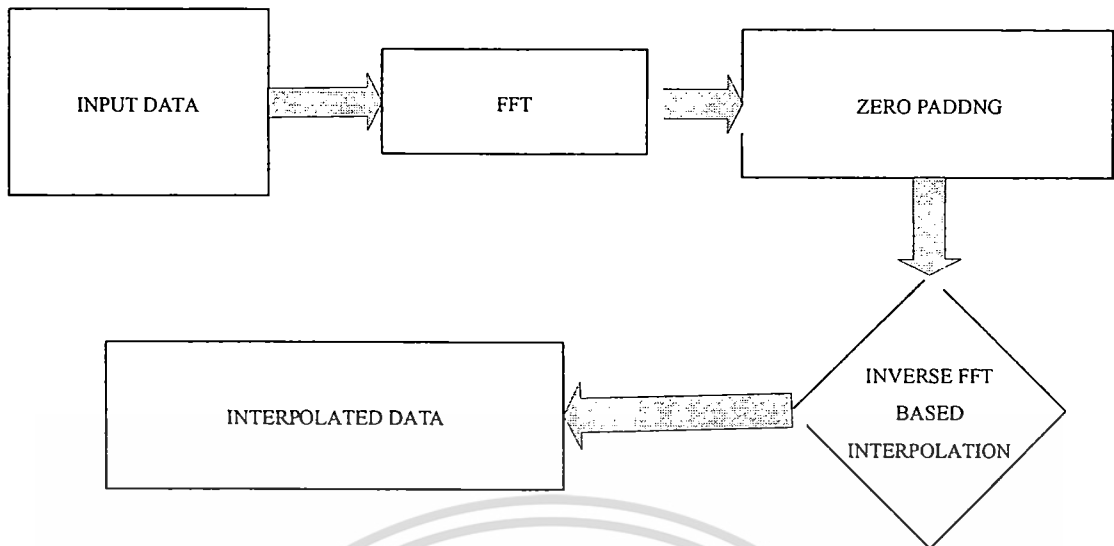
รูปที่ 3.2 แสดงการขยายภาพด้วยวิธีการ Bilinear

จากรูป ค่าที่ตำแหน่ง 1 2 3 และ 4 คือค่าของสีในพิกเซลต้นแบบที่จะนำมาทำการอินเทอร์โพลเลชัน โดยในตำแหน่งที่ 1 จะมีค่าพิกัด $(x,y) = (1,1)$, ตำแหน่งที่ 2 มีค่าพิกัด $(x,y) = (1,3)$, ตำแหน่งที่ 3 มีค่าพิกัด $(x,y) = (3,1)$, ตำแหน่งที่ 4 มีค่าพิกัด $(x,y) = (3,3)$ นำไปทำการแทนในสมการเพื่อหาค่าสัมประสิทธิ์ และนำค่าสัมประสิทธิ์ที่ได้มาคำนวณค่าของสีที่ตำแหน่งใหม่ แตกต่างกันไปตามพิกัดของ (x,y) เช่น ตำแหน่งที่อยู่ระหว่าง 1 และ 2 จะมีค่าพิกัด $(x,y) = (1,2)$ แทนลงไปในสมการเพื่อหาค่า Z

3.3 Fast Fourier transforms Interpolation

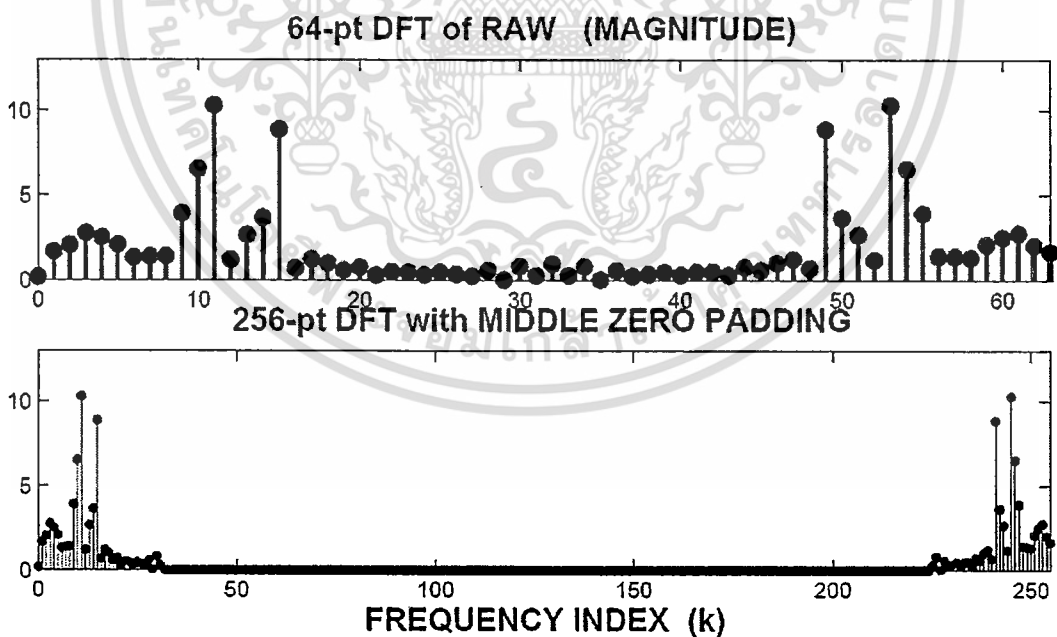
เป็นการแปลงสัญญาณจาก โดเมนของเวลา ไปเป็นโดเมนของความถี่ แล้วทำการเพิ่มค่า 0 ลงไปในตรงกลางของสเปกตรัม (Zero padding) ซึ่งหากเราเติมค่า 0 หลังสัญญาณในโดเมนของเวลา เมื่อทำการแปลง FFT แล้ว จะเป็นการเพิ่มจำนวนความหนาแน่นของสเปกตรัม แต่ในทางกลับกันหากเราเติมค่า 0 ลงตรงกลางของสเปกตรัมที่เกิดขึ้นจากการแปลง FFT แล้วทำการแปลงกลับมามีอยู่ในรูปของโดเมนเวลา จะพบว่า สัญญาณนั้นจะมีจำนวนของการสุ่มตัวอย่าง (Sampling) เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



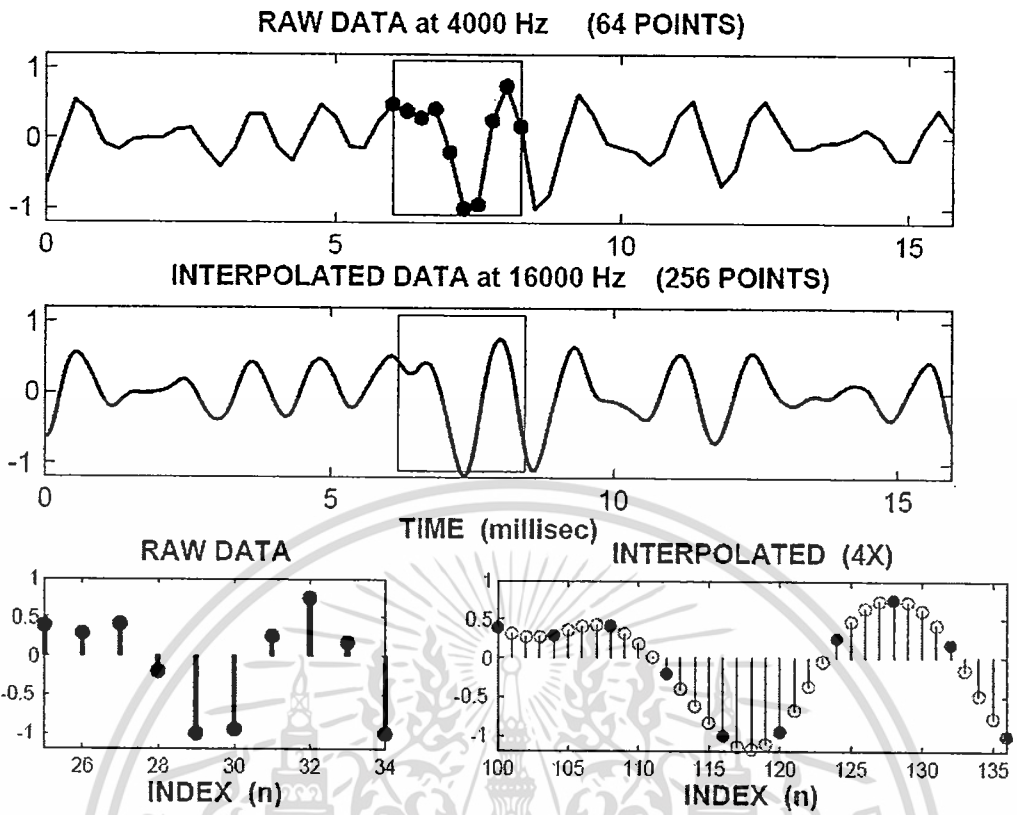
รูปที่ 3.3 ขั้นตอนในการทำการอินเทอร์โพลชันด้วย FFT

ซึ่งค่าของ Zero padding ที่ทำการเพิ่มลงไปโดเมนของสมปคตรีมนั้น จะต้องเพิ่มในจำนวนที่เท่ากับ จำนวนของข้อมูลเดิม (N) คูณกับ เลขจำนวนเท่าที่ต้องการขยาย (L) ลบด้วยหนึ่งเพราะฉะนั้น จำนวนข้อมูลของ Zero padding ที่ต้องทำการเพิ่มลงไปข้อมูลจะมีจำนวนเท่ากับ $N \times (L-1)$



รูปที่ 3.4 วิธีการเพิ่ม Zero padding ในแกนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



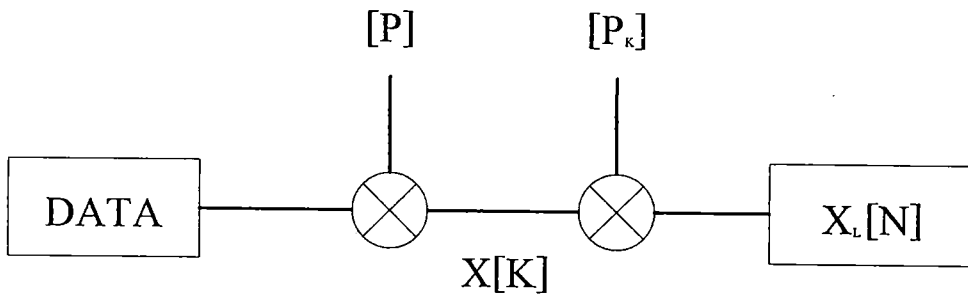
รูปที่ 3.5 ผลที่ได้จากการขยายด้วยวิธีการ FFT

3.4 Discrete Pascal transforms Interpolation

เราสามารถใช้กระบวนการแปลงฟูริเยร์ในการทำอินเทอร์โพลชัน ซึ่งการคำนวณในการแปลงฟูริเยร์จะใช้วิธีในการคำนวณแบบพหุนาม สัญญาณไม่ต่อเนื่องทางเวลา $x[n]$ สามารถแสดงผลได้เป็นผลรวมแบบถ่วงน้ำหนักของพหุนามแบบปาสคาล ที่ $X[k]$ เป็นขบวนการของสัมประสิทธิ์การแปลงแบบปาสคาล และ $P_k(x)$ เป็นพหุนามต่อเนื่องแบบปาสคาลกำลัง k เราสามารถสร้างฟังก์ชัน $x_L[n]$ ได้จากการเพิ่มอัตราการสุ่มตัวอย่างจากฟังก์ชัน $x[n]$ ดังนี้ ถ้า $x[n]$ มี N ตัวอย่าง $x_L[n]$ จะมี $N + (L - 1)(N - 1)$ ตัวอย่าง หรือ ประมาณ NL สำหรับ N ที่มีค่ามากๆ โดยเราสามารถคำนวณค่า $x_L[n]$ ได้จาก

$$x_L[n] = \sum_{k=0}^N X[k] P_k\left(\frac{n}{L}\right) \quad 0 \leq n \leq LN$$

โดยวิธีในการอินเทอร์โพลทแบบดิครีตปาสคาลทรานฟอร์ม จะมีขั้นตอนดังแสดงในภาพต่อไปนี้



รูปที่ 3.6 ขั้นตอนในการทำการอินเทอร์โพล์ด้วย DPT

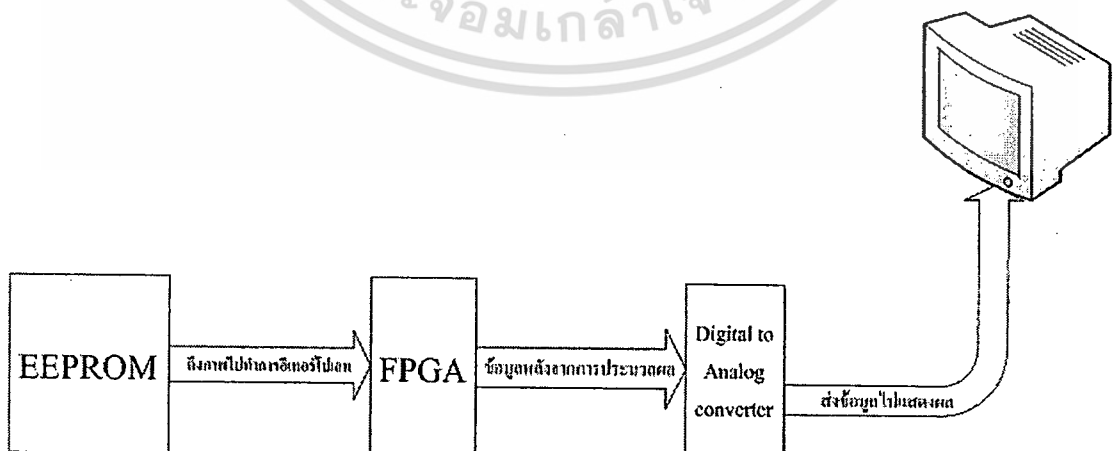
ซึ่งในการอินเทอร์โพล์ด้วยวิธีการนี้ สามารถทำการขยายได้สองแบบ แบบแรกคือทำการคำนวณและขยายที่เดียวทั้งภาพ (แบบโกลบอล) และแบบที่สองคือการแบ่งภาพออกเป็นส่วนย่อยๆ และนำมาขยายที่ละส่วนของภาพ (แบบโลคอล) ก่อนจะนำมาเรียงต่อกันให้เป็นภาพใหญ่

ในส่วนของฮาร์ดแวร์นั้นจะทำการนำวิธีการอินเทอร์โพล์ที่เหมาะสมกับการนำมาใช้ในการเขียนโปรแกรมลงบน FPGA โดยในที่นี้ได้ทำการเลือกใช้วิธีการในการอินเทอร์โพล์สามวิธีดังนี้

1. Replication Pixel (โดยใช้การหน่วงเวลา)
2. Replication Pixel
3. Linear

3.5 Replication Pixel (Using Delay)

เป็นวิธีการอินเทอร์โพล์โดยการทำการแสดงผลภาพข้อมูลเดิมในตำแหน่งของจอยวีจีโอถัดไป โดยจะทำให้เป็นการอินเทอร์โพล์ด้วยวิธีการสร้างข้อมูลซ้ำจากตำแหน่งก่อนหน้าโดยการใช้วิธีการอินเทอร์โพล์ด้วยวิธีนี้จะใช้หน่วยความจำภายนอกเพียงอย่างเดียวคือ EEPROM



รูปที่ 3.7 วงจรรวมที่ใช้ในการอินเทอร์โพล์แบบใช้การดีเลย์

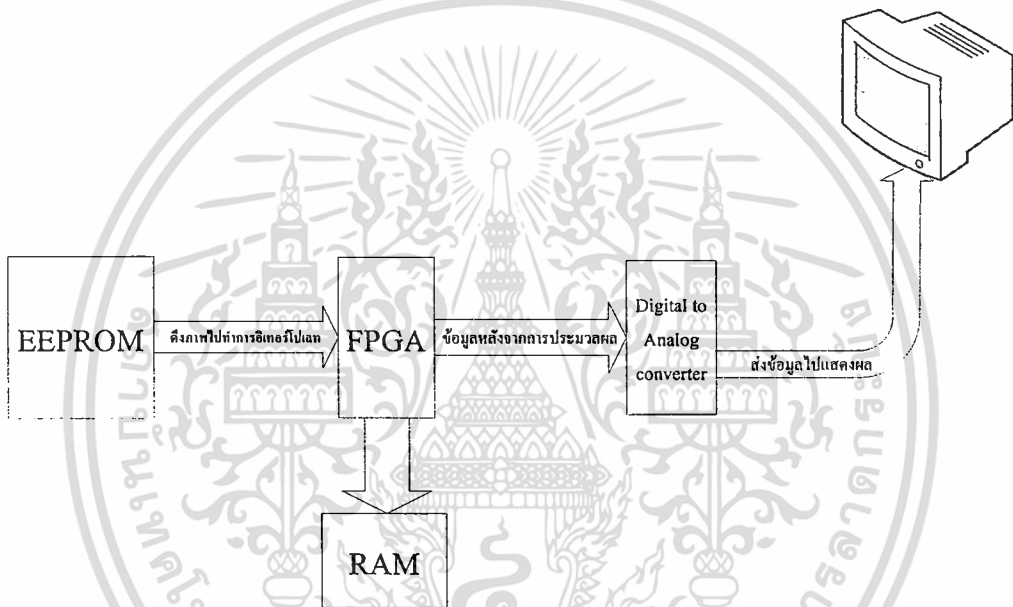
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 Replication Pixel

เป็นวิธีการที่เหมือนกับการทดลองในแมทแลป คือการคัดลอกพิกเซลต้นแบบมาไว้บริเวณข้างเคียง โดยจะทำการอ่านข้อมูลจาก EEPROM มาประมวลผลใน FPGA และเก็บลงใน RAM เพื่อเป็นที่เก็บข้อมูลในการอ่าน-เขียนข้อมูลเพื่อทำกระบวนการอินเทอร์โพล

3.7 Linear Interpolation

เป็นการอินเทอร์โพลโดยใช้วิธีการนำพิกเซลข้างเคียงมาเฉลี่ยค่าเพื่อนำมาสร้างข้อมูลใหม่ระหว่างข้อมูลทั้งสอง โดยจะทำการอ่านข้อมูลจาก EEPROM มาประมวลผลดังวิธีดังกล่าวแล้วนำไปเก็บลงใน RAM เพื่อเก็บข้อมูลในการอ่าน-เขียนข้อมูลเพื่อทำการอินเทอร์โพล



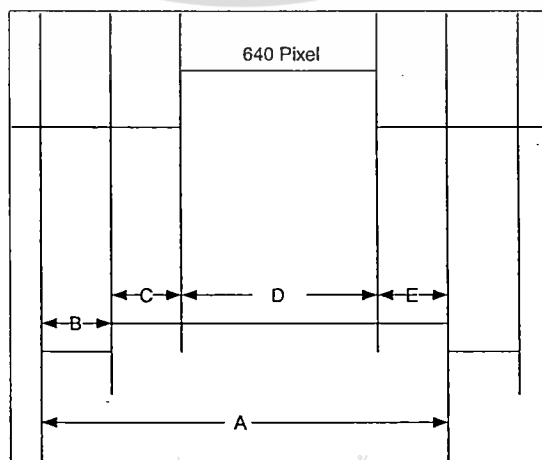
รูปที่ 3.8 วงจรรวมที่ใช้ในการอินเทอร์โพลแบบคัดลอกพิกเซลและแบบลิเนียร์

3.8 การสร้างสัญญาณความถี่วีจีโอ

3.8.1 การสร้างสัญญาณสแกนทางแนวนอน (Horizontal Synchronization)

สัญญาณอ้างอิง

สัญญาณการสแกน
ทางแนวนอน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น

รูปที่ 3.9 รูปสัญญาณการสแกนและสัญญาณการอ้างอิงตำแหน่งทางแนวนอน

สัญญาณการสแกนทางแนวนอนจะเป็นตัวกำหนดการสแกนทางแนวนอนในแต่ละแถวจะมีค่าข้อมูล อยู่ในช่วงประมาณ 794 คือช่วง C+D+E นั่นเองโดยจะมีสัญญาณอ้างอิงสี่เป็นสัญญาณ บอกถึง ต้องการสแกนในช่วง 0-639 ตามการสแกนตามมาตรฐานนั่นเอง

ตารางที่ 3.1 ตารางแสดงเวลาในช่วงต่าง ๆ ของสัญญาณการสแกนทางแนวนอน และค่าการแปลงเวลา เป็นข้อมูลที่ใช้ในการเขียนโปรแกรม

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

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

A คือเวลาทั้งหมดของการสแกนทางแนวนอน

B, G และ E คือ การ์ดแบนด์ (Guard Bands)

D คือเวลาการสแกนช่วงข้อมูล 640 พิกเซล/ แถว

ช่วงข้อมูลจะมีการคำนวณดังนี้

ความถี่ที่ใช้งานมีค่า 25.175 MHz

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

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

ถ้าเทียบจากเวลาเป็นข้อมูลจะมีการคำนวณดังนี้

$$A = 31.77 \mu\text{s} / 40 \text{ ns} = 794.25$$

$$D = 25.17 \mu\text{s} / 40 \text{ ns} = 629.25$$

$$B = 3.77 \mu\text{s} / 40 \text{ ns} = 94.25$$

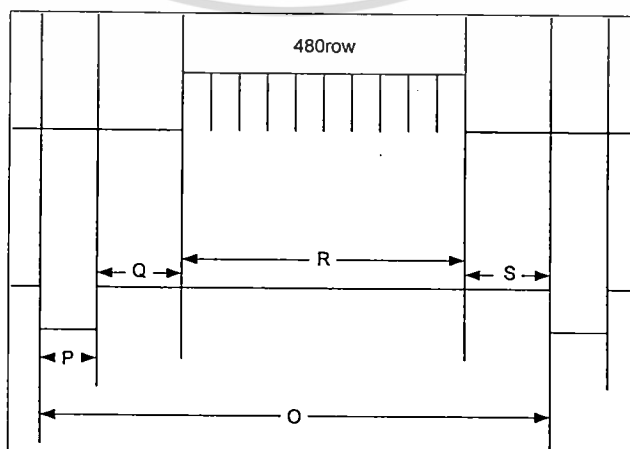
$$E = 0.94 \mu\text{s} / 40 \text{ ns} = 23.5$$

$$C = 1.89 \mu\text{s} / 40 \text{ ns} = 47.25$$

3.8.2 สัญญาณการสแกนทางแนวตั้ง (Vertical synchronization)

สัญญาณอ้างอิงสี่

สัญญาณการสแกน
ทางแนวตั้ง



รูปที่ 3.10 รูปสัญญาณการสแกนและสัญญาณอ้างอิงตำแหน่งทางแนวตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวงจำกัดเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณการสแกนทางแนวตั้งจะเป็นตัวกำหนดการสแกนทางแนวตั้งโดยถ้ามีการสแกนทางแนวอนครบ 523 แถวจะเกิดสัญญาณการสแกนทางแนวตั้ง 1 ลูก คือช่วง Q+R+S นั่นเองโดยจะมีสัญญาณการอ้างอิงที่เป็นสัญญาณ 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 จะแสดงช่วงค่าของวีจีโอใหม่มีมิ่งของสัญญาณการสแกนทางแนวตั้ง ซึ่งจะใช้เวลาตามช่วงที่กำหนดโดยประมาณ นอกจากนั้นเราจะแปลงค่าของเวลาเป็นค่าของข้อมูลเพื่อใช้ในการเขียนโปรแกรมด้วยภาษาวีเอชดีแอลจากค่าเวลาต่าง ๆ จะมีการคำนวณดังนี้

O คือเวลาทั้งหมดของการสแกนทางแนวตั้ง

P, Q และ S คือ การ์ดแบนด์

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

สัญญาณการสแกนทางแนวตั้ง 1 ลูก = สัญญาณการสแกนทางแนวอน 523 แถว
เวลาที่ใช้ในการสแกนแต่ละแถวคือ 31.77 μ s

$$Trow = 31.77 \mu s$$

เวลาทั้งหมดในการสแกนทางแนวตั้ง

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

ถ้าเทียบจากเวลาจะได้ดังนี้

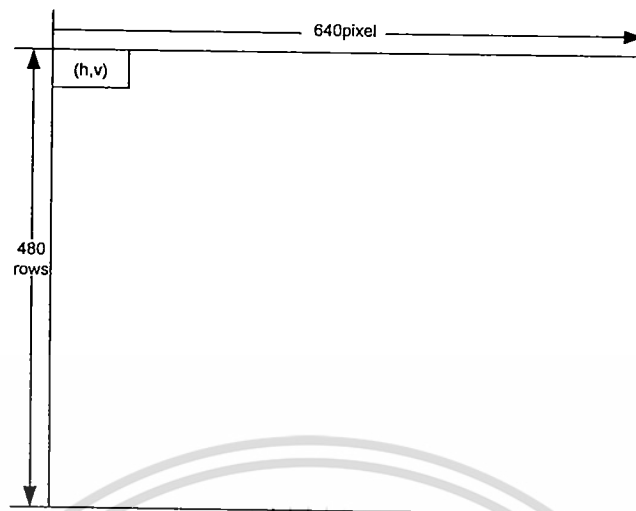
$$O = 16.6 ms / 31.77 \mu s = 522.505$$

$$P = 0.06 ms / 31.77 \mu s = 2.014$$

$$Q = 1.02 ms / 31.77 \mu s = 32.105$$

$$S = 0.35 ms / 31.77 \mu s = 11.016$$

3.8.3 สัญญาณอ้างอิงตำแหน่งทางแนวนอนและแนวตั้ง



รูปที่ 3.11 แสดงการอ้างอิงของจุดภาพ

จากการที่ได้กล่าวมาแล้วเราจะใช้ข้อมูลที่ได้จากการแปลงค่าจากช่วงเวลา มาสร้างสัญญาณตามที่ต้องการ ในการสร้างสัญญาณอ้างอิงตำแหน่ง จะใช้วงจรนับ โดยมีกลไกตามการคำนวณของทั้งการสร้างสัญญาณทั้งทางแนวนอนและแนวตั้งคือ

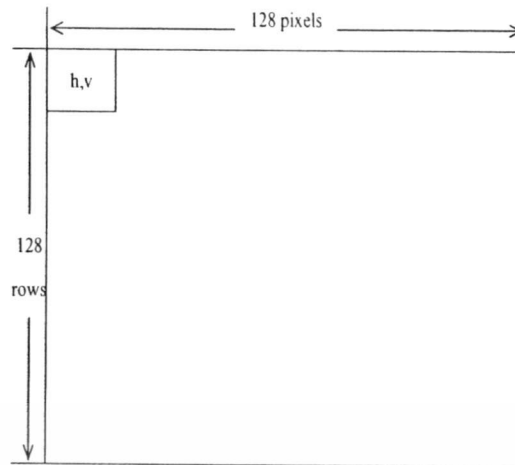
ที่ ขนาดภาพ 640x480 พิกเซล และ อัตราการแสดงผลภาพใหม่ = 60 Hz

- คลื่นการอ้างอิงตำแหน่งทางแนวนอนใช้ความถี่ประมาณ 25.175 MHz คิดเป็นเวลาคือประมาณ 40 ns

- คลื่นการอ้างอิงตำแหน่งทางแนวตั้งจะใช้ความถี่ประมาณ 31.476 KHz คิดเป็นเวลาคือประมาณ 31.77 μ s

ส่วนการอ้างอิงสีนั้นจะสามารถกำหนดตำแหน่งได้ให้เกิดสีปรากฏบนหน้าจอวีซีเอได้โดยดูจากรูปที่ 3.4 จะมีความกว้างทางแนวนอน 640 พิกเซล แทนด้วย h และจะมีความยาว 480 แถว แทนด้วย v ถ้าต้องการให้สีปรากฏ ที่ตำแหน่ง $[h, v]$ เราก็จะใช้สัญญาณเป็นตัวอ้างอิงเมื่อถึงตำแหน่งที่ต้องการให้สีปรากฏออกไป ซึ่งตัวสัญญาณมีลักษณะเป็นบัสในการอ้างอิงตำแหน่งสี ซึ่งเกิดจากการนับของดีฟลิปฟล็อปซึ่งก็จะใช้สัญญาณการอ้างอิงตำแหน่งนั่นเอง แต่จะมีสัญญาณการสแกนทางแนวนอนและแนวตั้งเป็นตัวควบคุมอยู่ ดังแสดงในรูป 3.9 และ 3.10

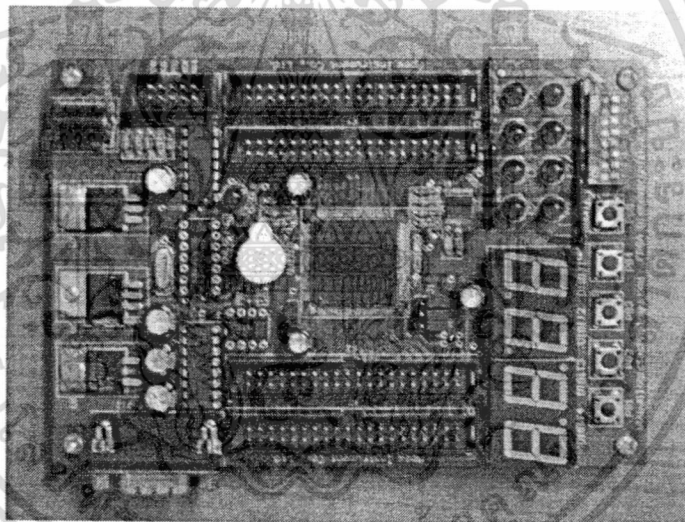
ซึ่งดังที่ได้กล่าวมาแล้วนั้นเป็นการสร้างสัญญาณอ้างอิงทางตำแหน่งไม่ว่าจะเป็นทั้งทางแนวนอนและแนวตั้งก็ตามจะเป็นการสร้างสัญญาณของหน้าจอวีซีเอทั้งหน้าจอ (ขนาด 640x480 พิกเซล) แต่ในโครงการนี้เราจะทดลองทำการสร้างสัญญาณอ้างอิงทางตำแหน่งทางแนวนอนเป็น 128 พิกเซล และสัญญาณอ้างอิงทางตำแหน่งทางแนวตั้งเป็น 128 แถว ก่อนในขั้นต้น ดังรูปที่ 3.12



รูปที่ 3.12 แสดงการอ้างอิงของจุดภาพ (128x128)

3.9 อุปกรณ์ที่ใช้ในการออกแบบ

3.9.1 บอร์ด FPGA รุ่น Discovery-III XC3S200

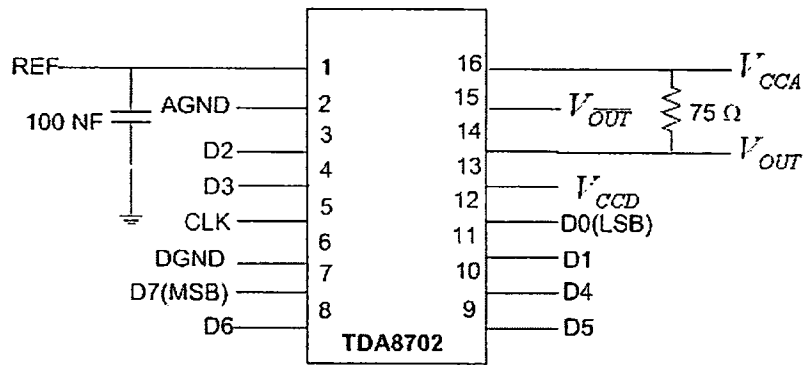


รูปที่ 3.13 บอร์ดทดลอง FPGA รุ่น Spartan III

3.9.2 ไอซี TDA8702

เป็นส่วนที่ทำหน้าที่แปลงสัญญาณภาพขนาด 8 บิต ซึ่งเป็นสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก ซึ่งทำการแปลงระดับสัญญาณดิจิทัลในช่วง '0000000' ถึง '1111111' ให้ระดับแรงดันอยู่ในช่วง 0-0.8 โวลต์ โดยใช้ไอซีเบอร์ TDA8702 โดยมีอัตราการแปลงข้อมูลสูงสุด 30 MHz แต่ในโครงการนี้เราใช้คล็อกที่มีความถี่ 25.175 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



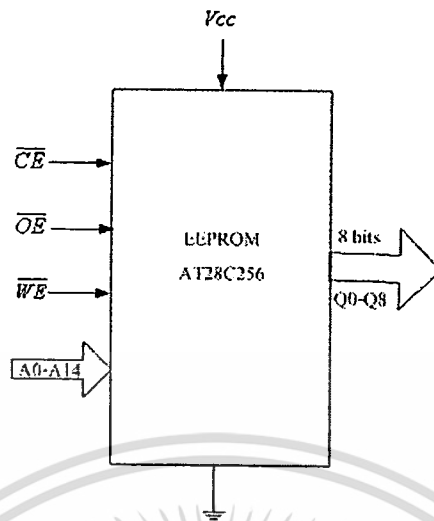
รูปที่ 3.14 การต่อวงจร IC เบอร์ TDA8702

ตารางที่ 3.3 การใช้งานของแต่ละ PIN ของ TDA 8702

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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.3 หน่วยความจำภายนอก EEPROM



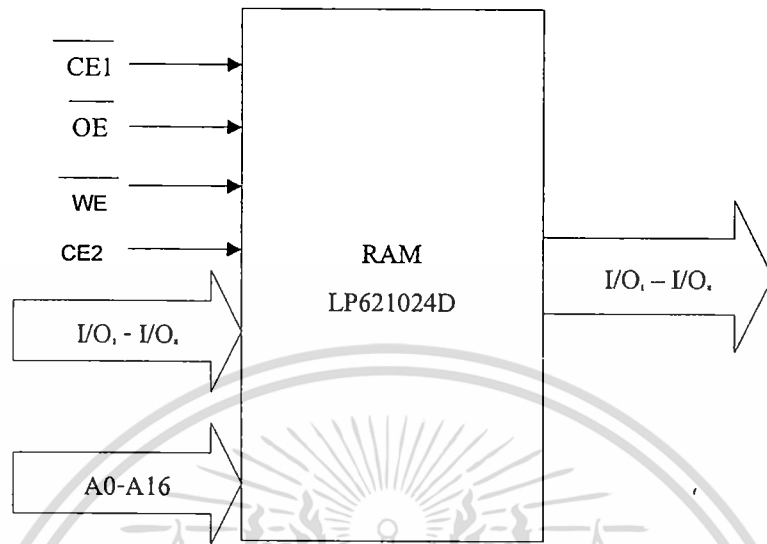
รูปที่ 3.15 การทำงานของหน่วยความจำภายนอก (EEPROM)

จากรูปที่ 3.15 เป็นรูปของหน่วยความจำภายนอก (EEPROM) จะอยู่ในสถานะ “Read” ก็ต่อเมื่อทำการป้อนสัญญาณ ‘low’ ให้กับขา \overline{CE} (Chip Enable) และขา \overline{OE} (Output Enable) และป้อนสัญญาณ ‘high’ ให้กับขา \overline{WE} ซึ่งใช้เป็นเบอร์ 28c256 ดังรูป

ตารางที่ 3.4 ชื่อของตัวสัญญาณต่าง ๆ ที่ใช้งานใน 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

3.9.4 หน่วยความจำภายนอก RAM



รูปที่ 3.16 การทำงานของหน่วยความจำภายนอก (RAM)

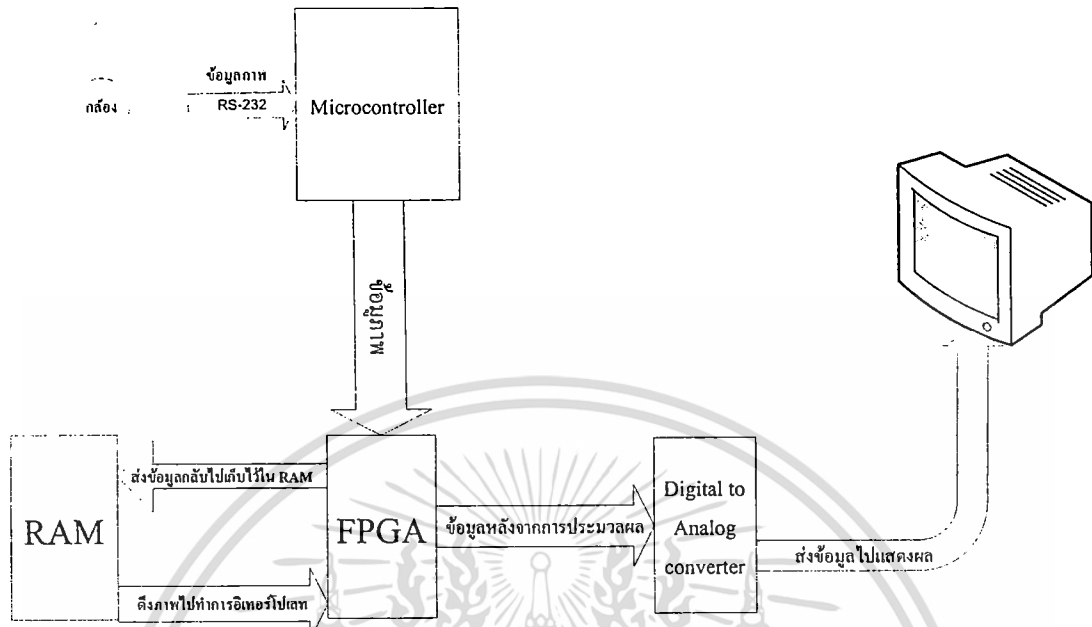
จากรูปที่ 3.16 เป็นรูปของหน่วยความจำภายนอก (RAM) จะอยู่ในสถานะ “Read” ก็ต่อเมื่อ ทำการป้อนสัญญาณ ‘high’ ให้กับ ขา WE (Write Enable) และจะอยู่ในสถานะ “Write” ก็ต่อเมื่อทำการป้อนสัญญาณ ‘low’ ให้กับขา WE1(Write Enable) และในตลอดเวลาจะป้อนสัญญาณ ‘high’ ให้กับขา OE (Output Enable) และ ขา CE1 (Chip Enable1) ส่วนขา CE2 (Chip Enable2) จะป้อนสัญญาณ ‘low’

ตารางที่ 3.5 ชื่อของตัวสัญญาณต่าง ๆ ที่ใช้งานใน RAM

Pin No.	Name	Description
1	NC	No Connection
2 - 12, 23, 25 - 28, 31	A0 - A16	Address Inputs
13 - 15, 17 - 21	I/O1 - I/O8	Data Input/Outputs
16	GND	Ground
22	CE1	Chip Enable
24	OE	Output Enable
29	WE	Write Enable
30	CE2	Chip Enable
32	VCC	Power Supply (+5V)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การออกแบบวงจร



รูปที่ 3.17 โครงสร้างโดยรวมทั้งหมดของโครงการ

3.10.1 วงจรที่สร้างจากเฟิร์มแวร์

มีโครงสร้างหลัก ๆ ตามรูปที่ 3.17 โดยจะอาศัยหน่วยความจำภายนอก (RAM) เพื่อที่จะเก็บข้อมูลแต่ละพิกเซลของภาพ ซึ่งแต่ละพิกเซลนั้นจะมีขนาด 8 บิต โดยวงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเฟิร์มแวร์ที่ออกแบบด้วยภาษาวีเอชดีแอลจะทำการชี้ตำแหน่งแอดเดรสของข้อมูลให้กับหน่วยความจำภายนอกด้วยความเร็ว 25.175 MHz จากนั้นทำการรับข้อมูลภาพแต่ละพิกเซลจากหน่วยความจำภายนอกที่ส่งกลับมาให้กับเฟิร์มแวร์ทำการประมวลผลสัญญาณภาพในลักษณะต่าง ๆ จากนั้นจะนำข้อมูลเข้าสู่วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกในระดับสี่เททา ทำการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อกที่มีระดับแรงดันต่าง ๆ กัน นอกจากนี้แล้ววงจรควบคุมการแสดงผลภาพโดยใช้บอร์ดเฟิร์มแวร์ยังสร้างสัญญาณควบคุมการสแกนจอภาพทางแนวนอนและแนวตั้งอีกด้วย เมื่อนำสัญญาณควบคุมการสแกนจอภาพทางแนวนอนและแนวตั้งเชื่อมต่อกับหน้าจอดีวีเอจี้จะสามารถแสดงผลภาพที่หน้าจอได้

ซึ่งส่วนประกอบของวงจรแสดงผลภาพโดยใช้บอร์ดเฟิร์มแวร์หน้าจอดีวีเอจี้จะประกอบด้วยวงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง วงจรสร้างตำแหน่งของข้อมูล วงจรรับสัญญาณข้อมูลภาพจากหน่วยความจำภายนอกและวงจรประมวลผลภาพในรูปแบบต่างๆ โดยแต่ละส่วนนั้นมีรายละเอียดดังต่อไปนี้

3.10.1.1 วงจรสร้างตำแหน่งของข้อมูล

วงจรสร้างตำแหน่งของข้อมูลจะเป็นวงจรรนับ(Counter) ขนาด14บิต เพื่อเรียกตำแหน่งจาก EEPROM และขนาด 16 บิต เพื่อเรียกตำแหน่งจาก RAM เพื่อสร้างตำแหน่งของข้อมูลจากหน่วยความจำภายนอก ซึ่งกำหนดความเร็วในการนับโดยใช้สัญญาณนาฬิกา (clk) โดยในโครงงานนี้จะใช้สัญญาณนาฬิกาขนาด 25.175 MHz เพื่อให้สัมพันธ์กับความเร็วที่ใช้ในการควบคุมหน้าจอและจะควบคุมการทำงานของแรม โดยการใช้ขา WE เป็นตัวกำหนดว่าจะอ่านค่าจากหน่วยความจำหรือเขียนข้อมูลลงไปในหน่วยความจำ

3.10.1.2 วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง

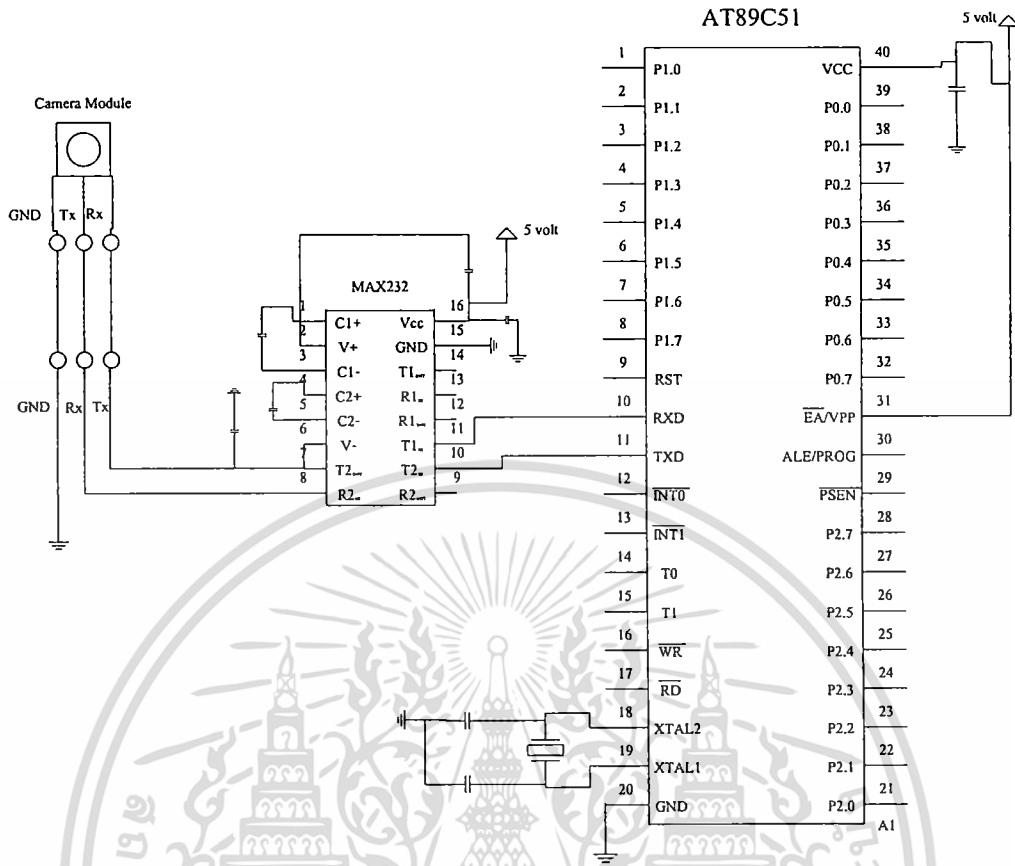
วงจรสร้างสัญญาณสแกนหน้าจอทางแนวนอนและแนวตั้ง จะทำหน้าที่สร้างสัญญาณสแกนหน้าจอทางแนวนอน และสัญญาณสแกนหน้าจอทางแนวตั้งซึ่งเราสามารถสร้างวงจรถัดกล่าวได้โดยการใช้วงจรรนับ โดยอาศัยข้อมูลจากการคำนวณที่ผ่านมตามตารางที่ 3.9 และตารางที่ 3.10

3.10.1.3 วงจรขยายขนาดของภาพ

วงจรขยายขนาดของภาพ ทำหน้าที่ในการขยายขนาดของภาพโดยวิธีการ Interpolate ตามหลักการดังที่กล่าวมาโดยในที่นี้ในการออกแบบวงจรจะใช้วิธีการ Replications pixel และ Linear interpolation ซึ่งสามารถออกแบบทางด้าน Hardware ได้โดยไม่มีข้อจำกัดมากนัก

3.10.2 วงจรรับข้อมูลภาพจากกล้อง iv-cam

เป็นส่วนที่ใช้ในการติดต่อกับโมดูลตรวจจับภาพ โดยทำหน้าที่ส่งคำสั่งเฉพาะไปให้โมดูลตรวจจับภาพทำการเก็บภาพ โดยในระหว่างการสื่อสาร โมโครคอนโทรลเลอร์จะทำหน้าที่รับข้อมูลภาพจากกล้องผ่านพอร์ตการสื่อสารอนุกรม แล้วทำการแปลงเป็นระดับสัญญาณดิจิทัลในช่วง '00000000' ถึง '11111111' ก่อนที่จะส่งข้อมูลดังกล่าวออกไปให้เอพพีจีเอ ผ่านพอร์ตเอาต์พุตขนานขนาด 8 บิต



รูปที่ 3.18 วงจรรับข้อมูลจากภาพจากกล้อง iv-cam

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

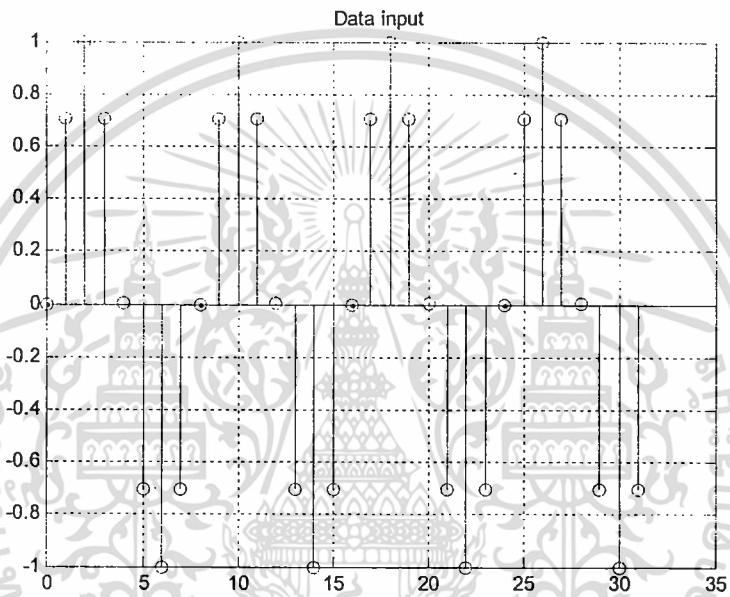
บทที่ 4

การทดลองและผลการทดลอง

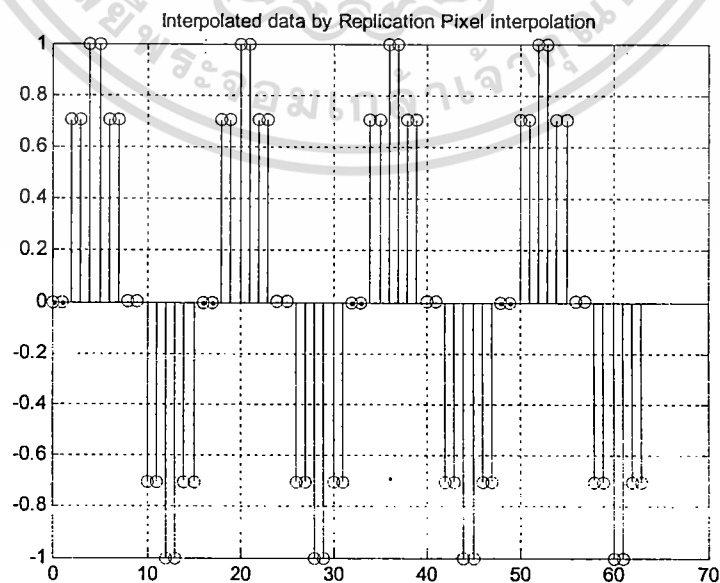
4.1 ผลการทดลองที่ได้จากการทำการทดลองโดยใช้ Matlab

4.1.1 ผลการทดลองด้วยวิธีการ Replication Pixel interpolation

4.1.1.1 ข้อมูล 1 มิติ



รูปที่ 4.1 สัญญาณชานันไม่ต่อเนื่อง



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยได้ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 ข้อมูล 2 มิติ

Original image input



รูปที่ 4.3 ภาพต้นแบบ (256x256)

Interpolated Image by Replication Pixel



รูปที่ 4.4 ภาพที่ผ่านการขยายแบบ Replication Pixel (512x512)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original image input



รูปที่ 4.5 ภาพต้นแบบ (128x128)

Interpolated Image by Replication Pixel

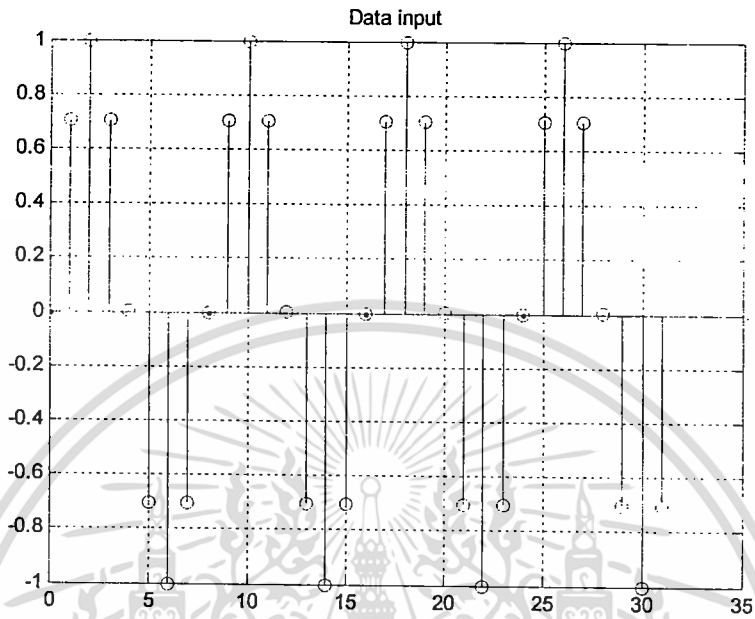


รูปที่ 4.6 ภาพที่ผ่านการขยายแบบ Replication Pixel (256x256)

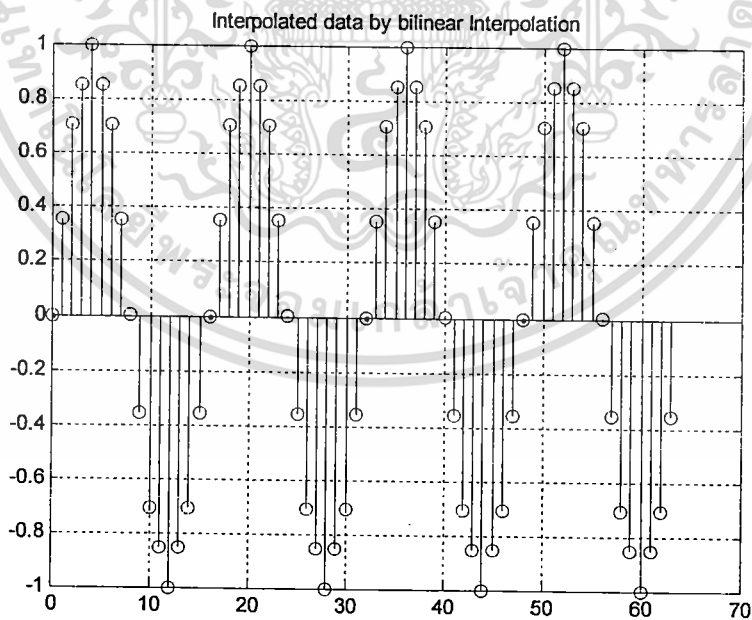
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองด้วยวิธีการ Bilinear Interpolation

4.1.2.1 ข้อมูล 1 มิติ



รูปที่ 4.7 สัญญาณชายนี่ไม่ต่อเนื่อง



รูปที่ 4.8 สัญญาณชายนี่ผ่านการขยายแบบ Bilinear Interpolation

เอกสารนี้เป็น 4.1.2.2 ข้อมูล 2 มิติ ทรัพยากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original image input



รูปที่ 4.9 ภาพต้นแบบ (256x256)

Interpolated Image by Bilinear Interpolation



รูปที่ 4.10 ภาพที่ผ่านการขยายแบบ Bilinear Interpolation (512x512)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original image input



รูปที่ 4.11 ภาพต้นแบบ (128x128)

Interpolated Image by Bilinear Interpolation



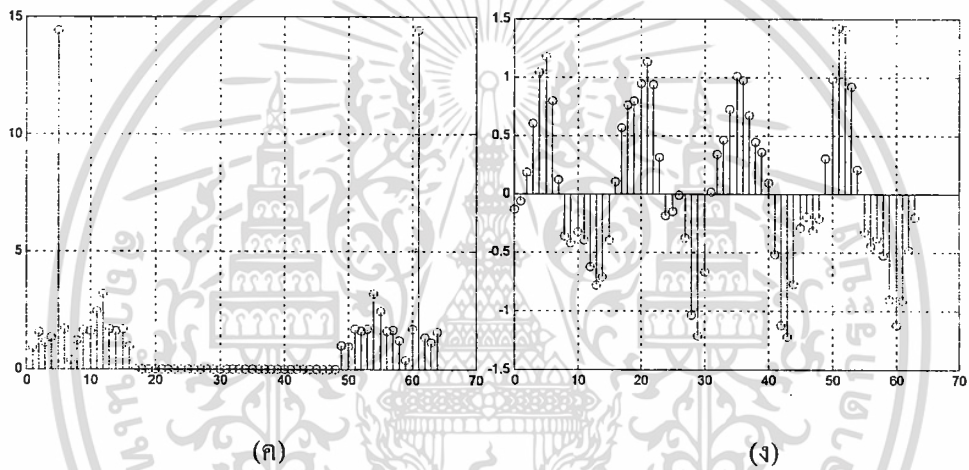
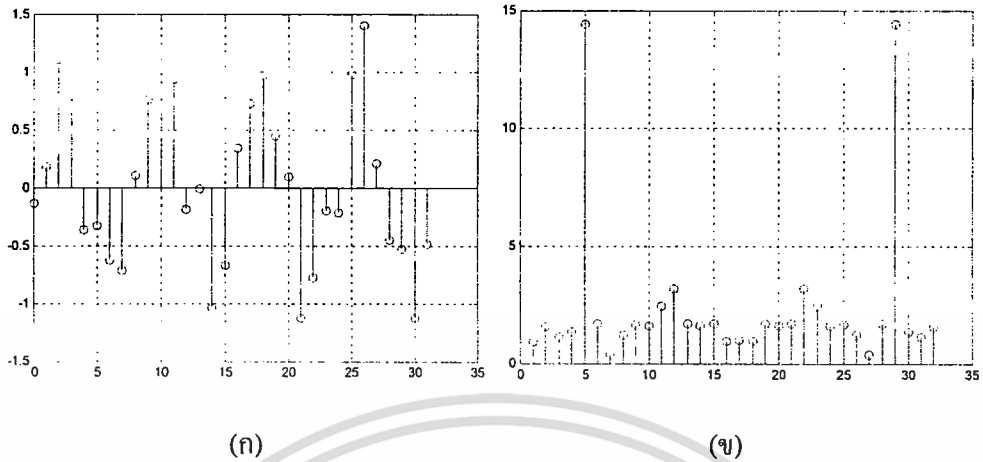
รูปที่ 4.12 ภาพที่ผ่านการขยายแบบ Bilinear Interpolation (256x256)

4.1.3 ผลการทดลองด้วยวิธีการ Fast Fourier Transform Interpolation

4.1.3.1 ข้อมูล 1 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการขยายให้ใหญ่ขึ้น 2 เท่า



(ก) สัญญาณชายน์ผสมกับสัญญาณรบกวนแบบสุ่ม

(ข) สเปกตรัมของสัญญาณ (ก)

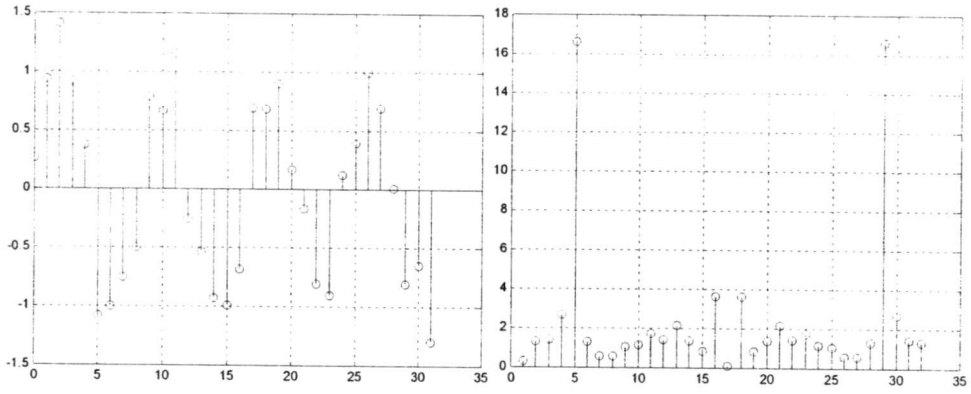
(ค) สเปกตรัมที่ผ่านการเพิ่ม Zero padding

(ง) สัญญาณที่ผ่านการขยายแบบ FFT Interpolation

รูปที่ 4.13 แสดงขั้นตอนการอินเทอร์โพลตสัญญาณแบบฟาสฟูรีเยร์ (ขยาย 2 เท่า)

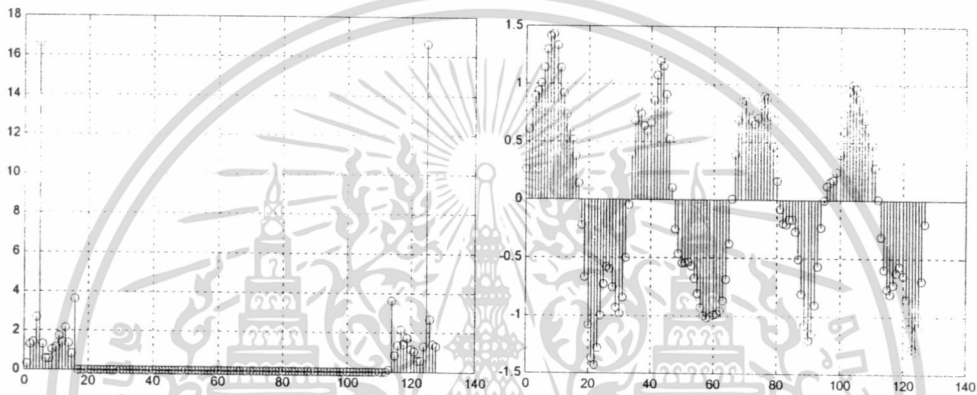
เมื่อทำการขยายให้ใหญ่ขึ้น 4 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

(ข)



(ค)

(ง)

(ก) สัญญาณซายน์ผสมกับสัญญาณรบกวนแบบสุ่ม

(ข) สเปกตรัมของสัญญาณ (ก)

(ค) สเปกตรัมที่ผ่านการเพิ่ม Zero padding

(ง) สัญญาณที่ผ่านการขยายแบบ FFT Interpolation

รูปที่ 4.14 แสดงขั้นตอนการอินเทอร์โพลตสัญญาณแบบฟาสฟูรีเยร์ (ขยาย 4 เท่า)

4.1.3.2 ข้อมูล 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original Image input



รูปที่ 4.15 ภาพต้นแบบ (256x256)

Interpolated Image by FFT Interpolation



รูปที่ 4.16 ภาพที่ผ่านการขยายแบบ FFT Interpolation (512x512)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original Image input



รูปที่ 4.17 ภาพต้นแบบ (128x128)

Interpolated Image by FFT Interpolation



รูปที่ 4.18 ภาพที่ผ่านการขยายแบบ FFT Interpolation (256x256)

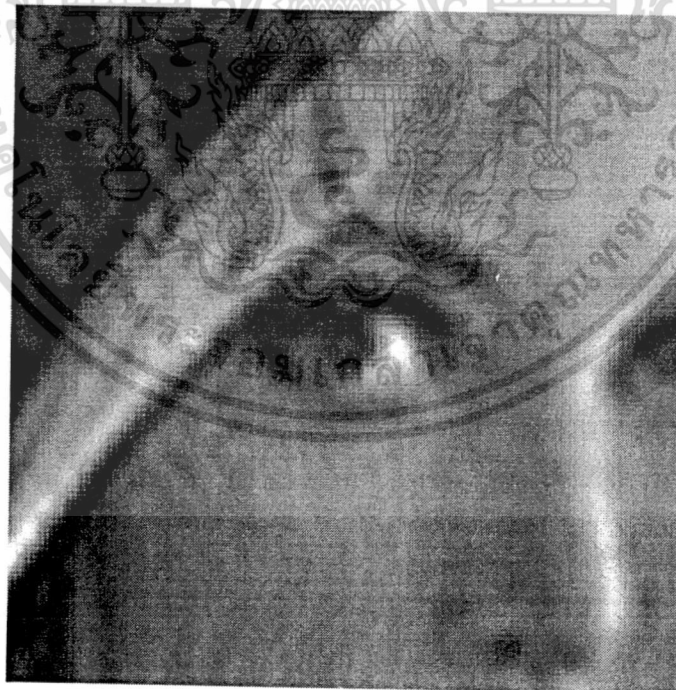
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interpolated Image by FFT Interpolation $L = 2$



รูปที่ 4.19 ภาพที่ผ่านการขยายแบบ FFT Interpolation (2 เท่า)

Interpolated Image by FFT Interpolation $L = 4$



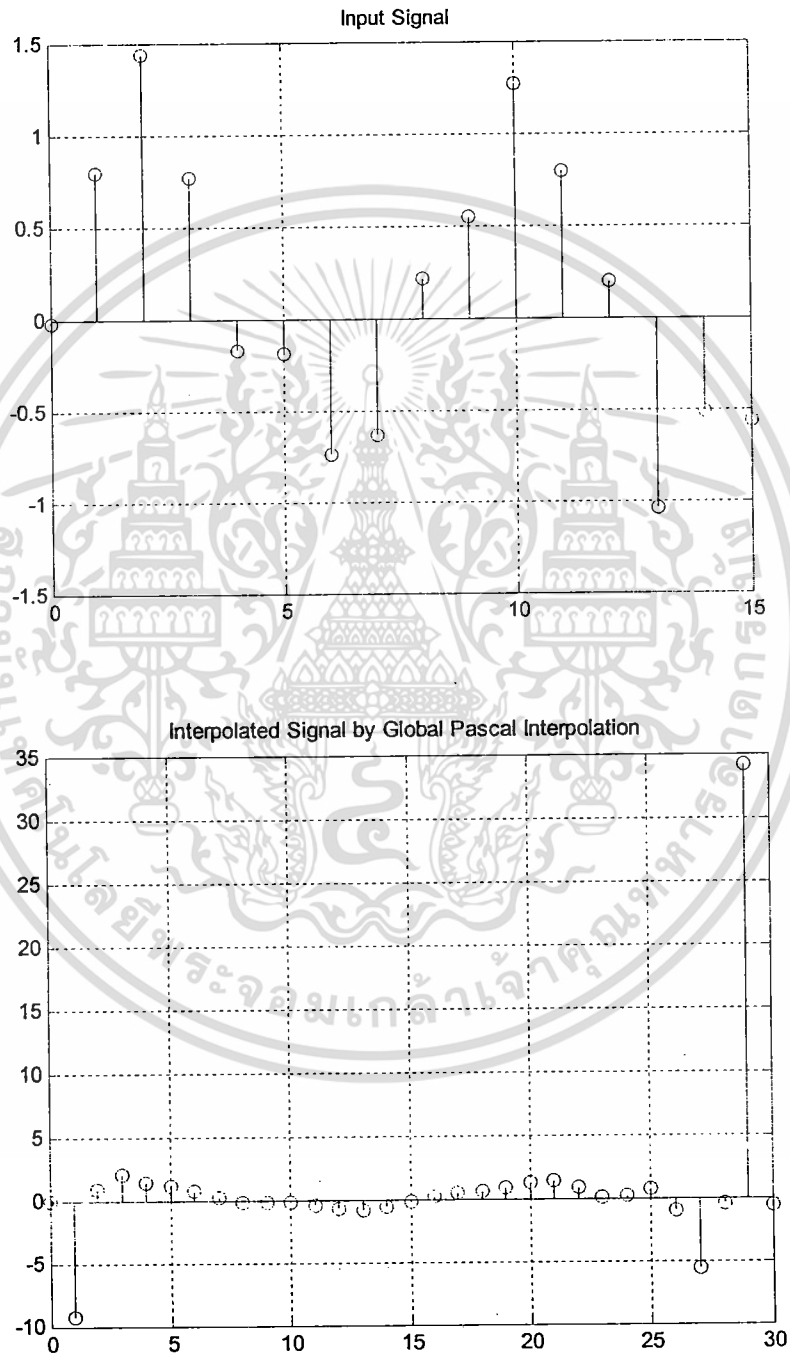
รูปที่ 4.20 ภาพที่ผ่านการขยายแบบ FFT Interpolation (4 เท่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ผลการทดลองด้วยวิธีการ Discrete Pascal Transform Interpolation

4.1.4.1 Global

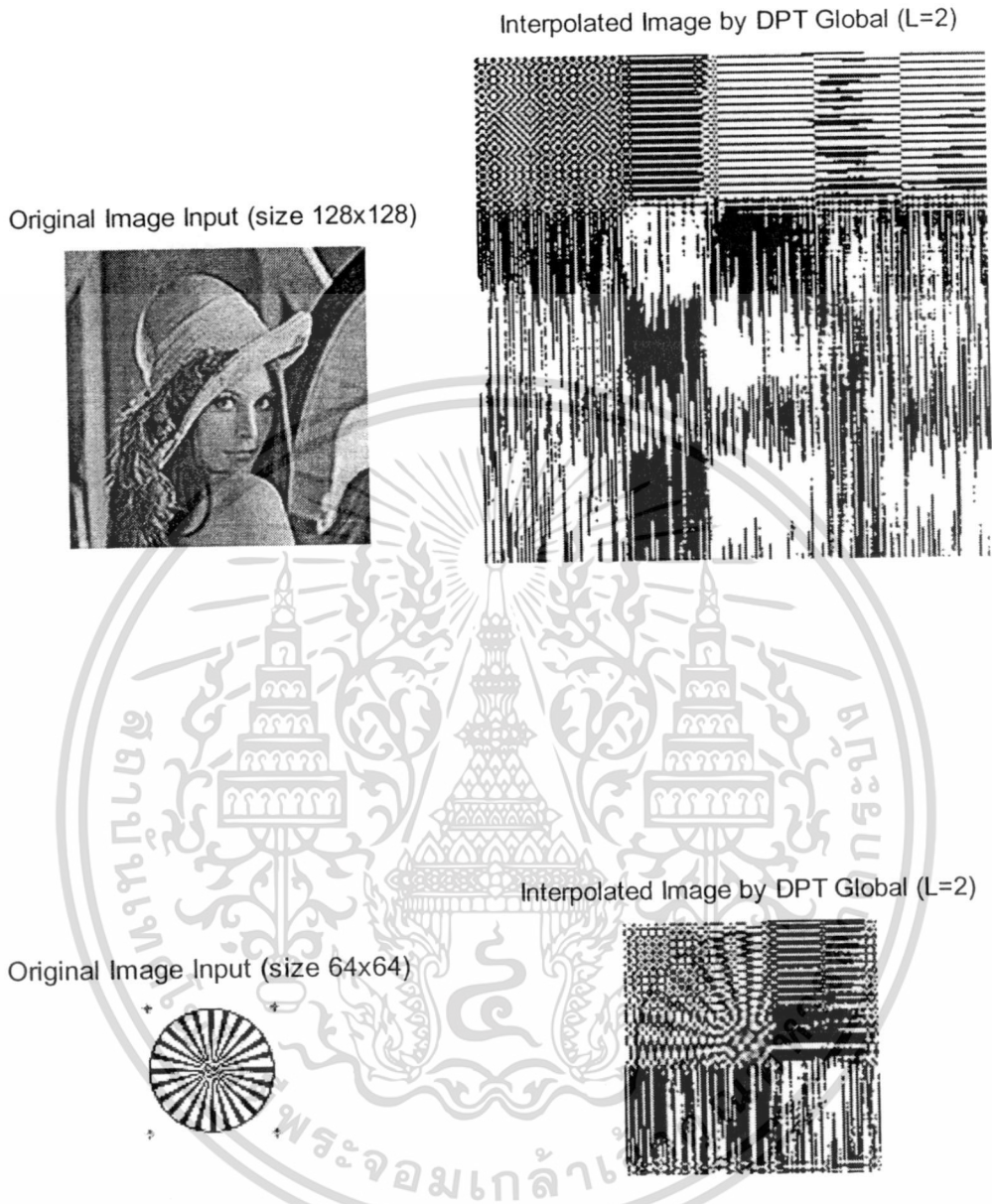
4.1.4.1.1 ข้อมูล 1 มิติ



รูปที่ 4.21 สัญญาณต้นแบบ และ เมื่อขยายด้วย DPT แบบ Global

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4.1.2 ข้อมูล 2 มิติ

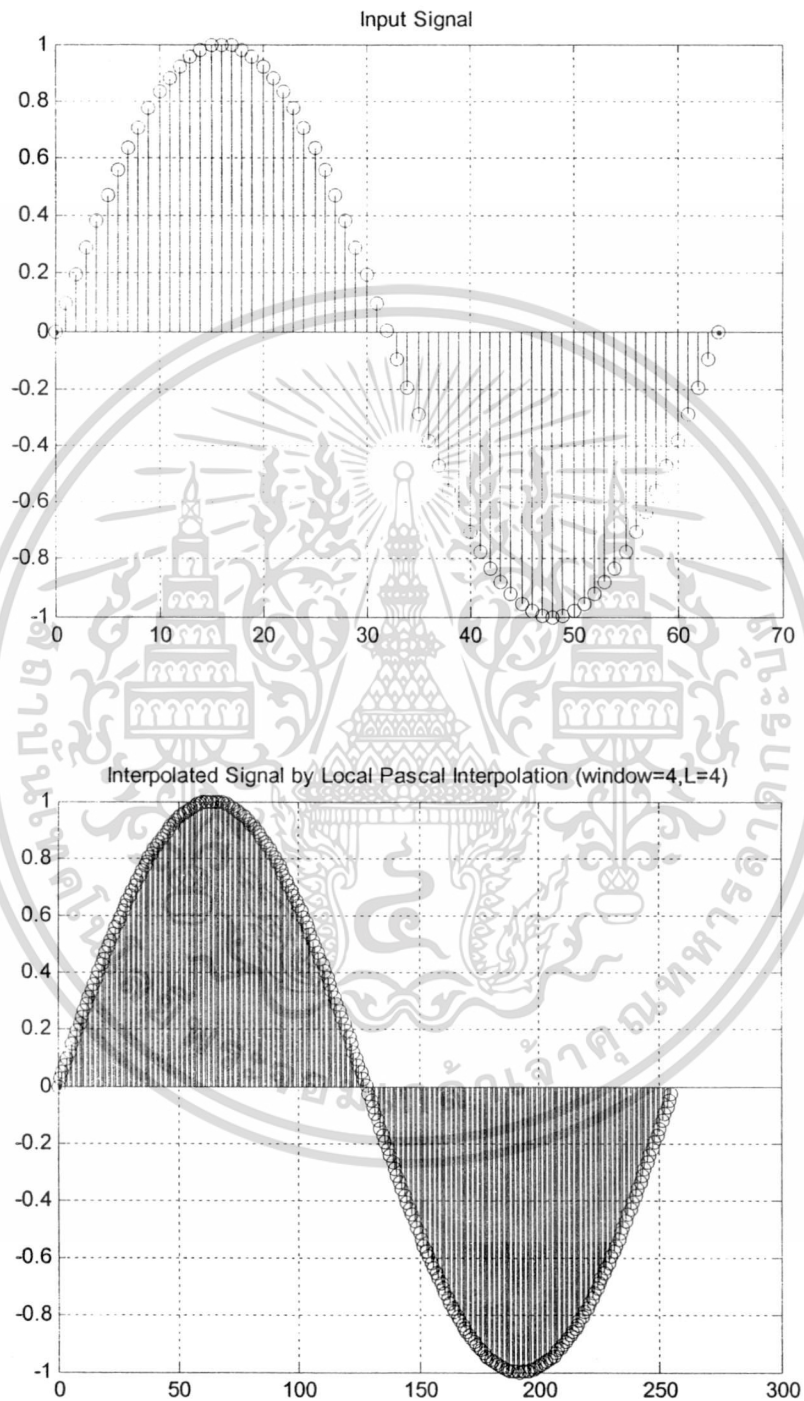


รูปที่ 4.22 ข้อมูล 2 มิติต้นแบบและที่ผ่าน Discrete Pascal Transform Interpolation แบบ Global

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

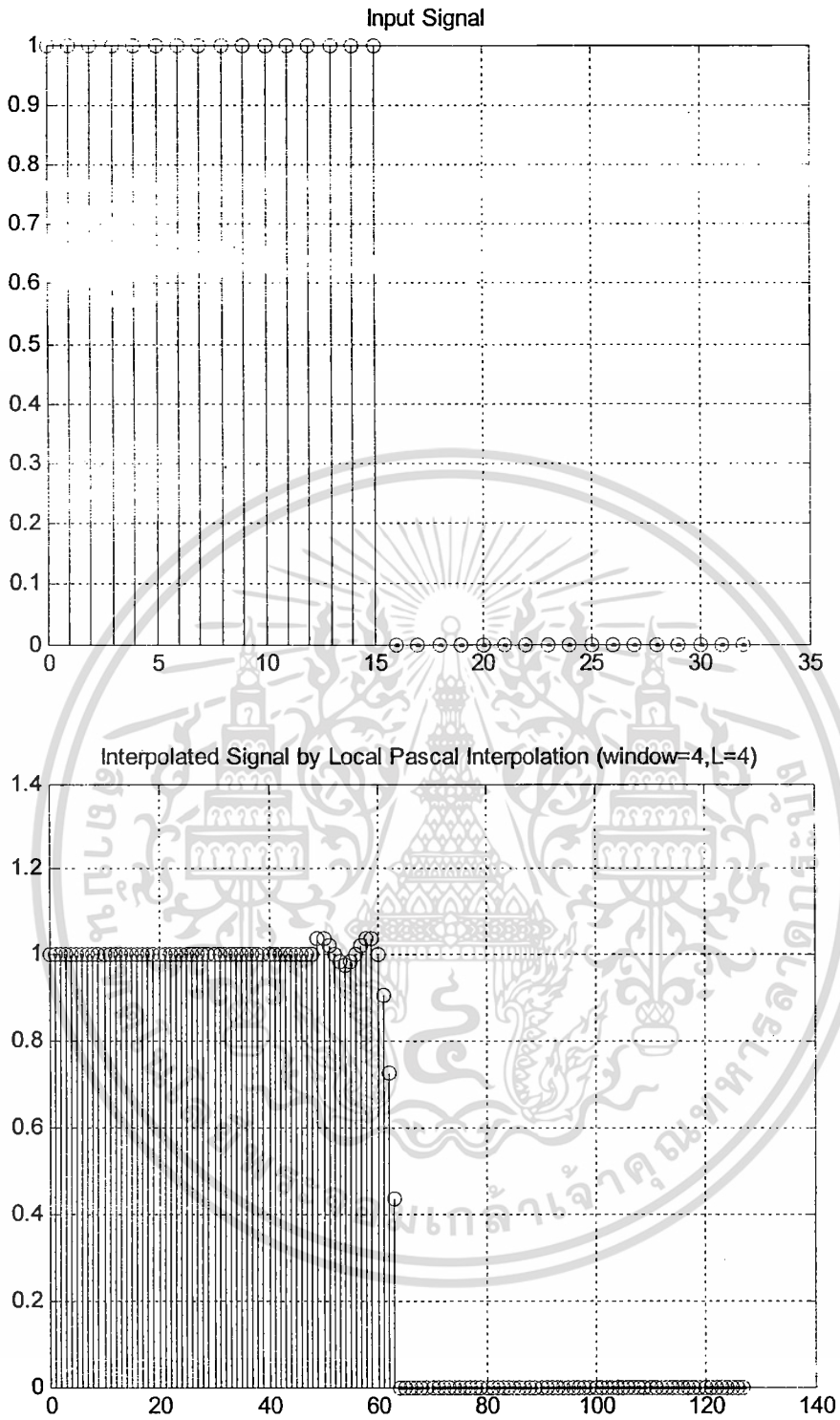
4.1.4.2 Local

4.1.4.2.1 ข้อมูล 1 มิติ



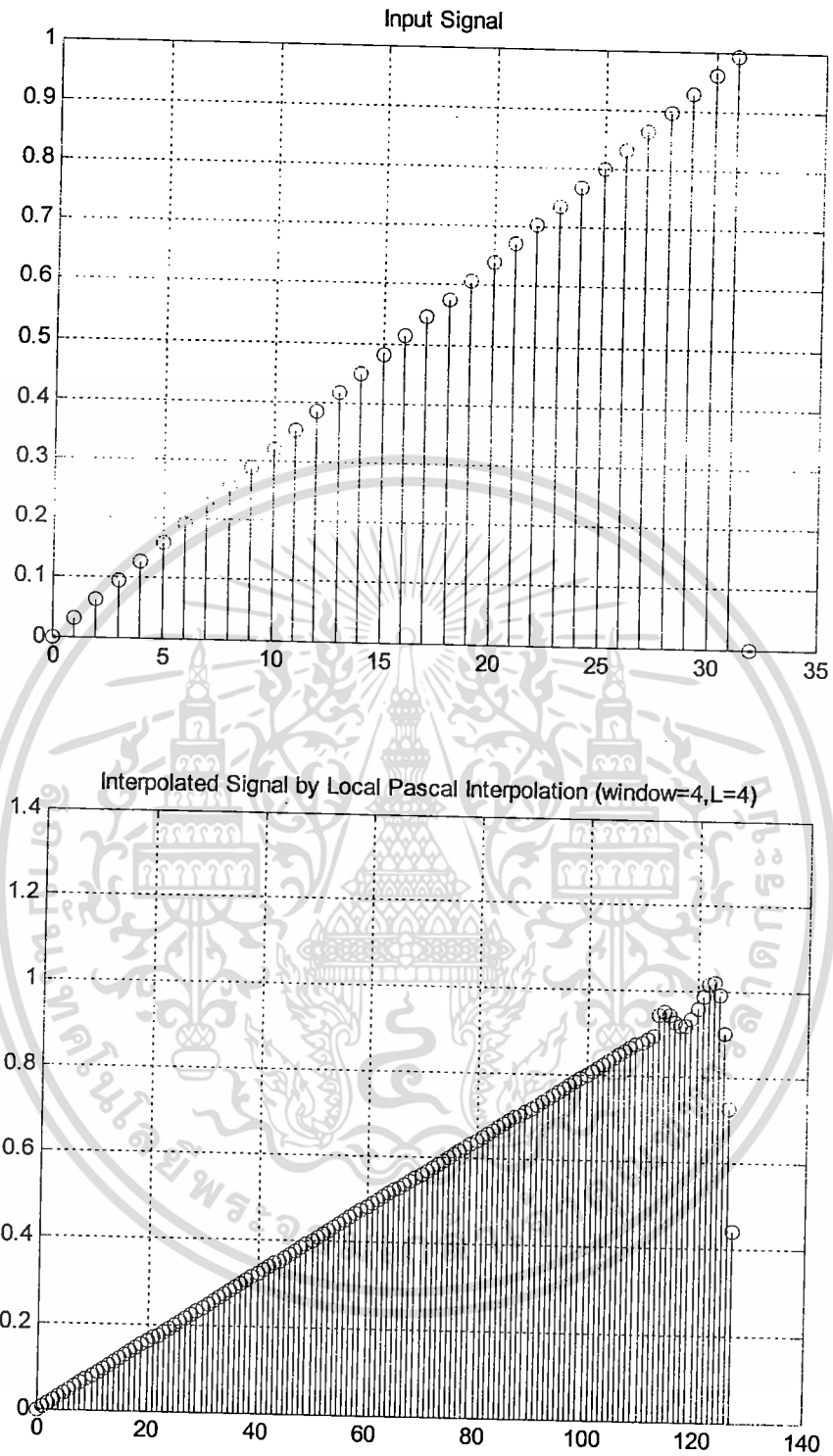
รูปที่ 4.23 สัญญาณชานด์ต้นแบบ และเมื่อขยายด้วย DPT แบบ Local

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



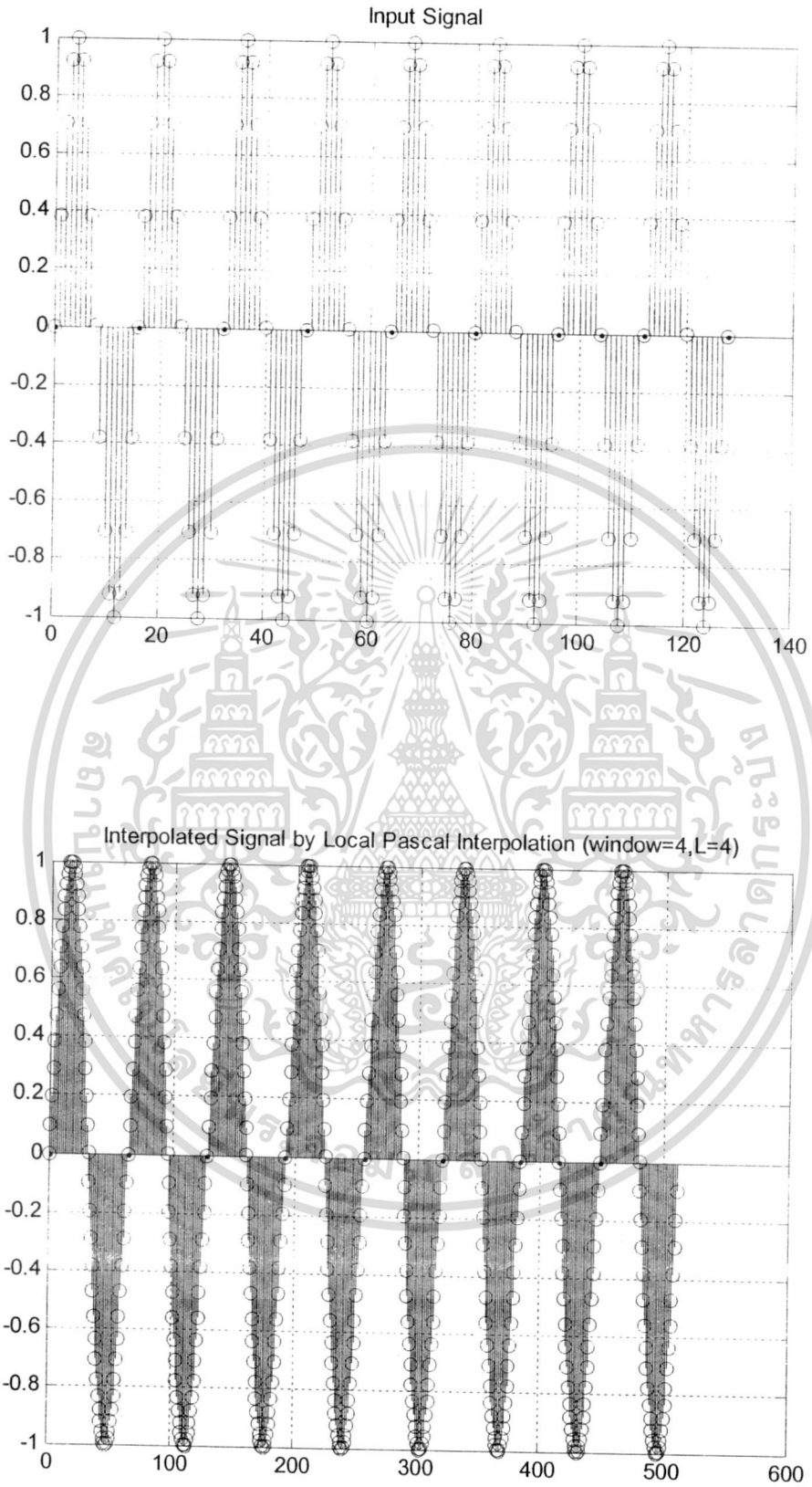
รูปที่ 4.24 สัญญาณพัลส์ต้นแบบ และเมื่อขยายด้วย DPT แบบ Local

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 สัญญาณสามเหลี่ยมต้นแบบ และเมื่อขยายด้วย DPT แบบ Local

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 ขบวนการสัญญาณขายนัดแบบ และเมื่อขยายด้วย DPT แบบ Local

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4.2.2 ข้อมูล 2 มิติ

Original Image Input (size 128x128)



รูปที่ 4.27 ภาพต้นฉบับ (128x128)

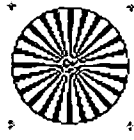
Interpolated Image by DPT Local ($L=2$)



รูปที่ 4.28 ภาพที่ผ่านการขยายแบบ DPT Interpolation (2 เท่า)

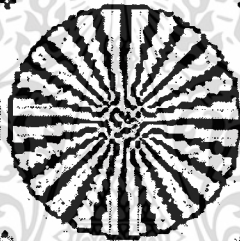
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original Image Input (size 64x64)



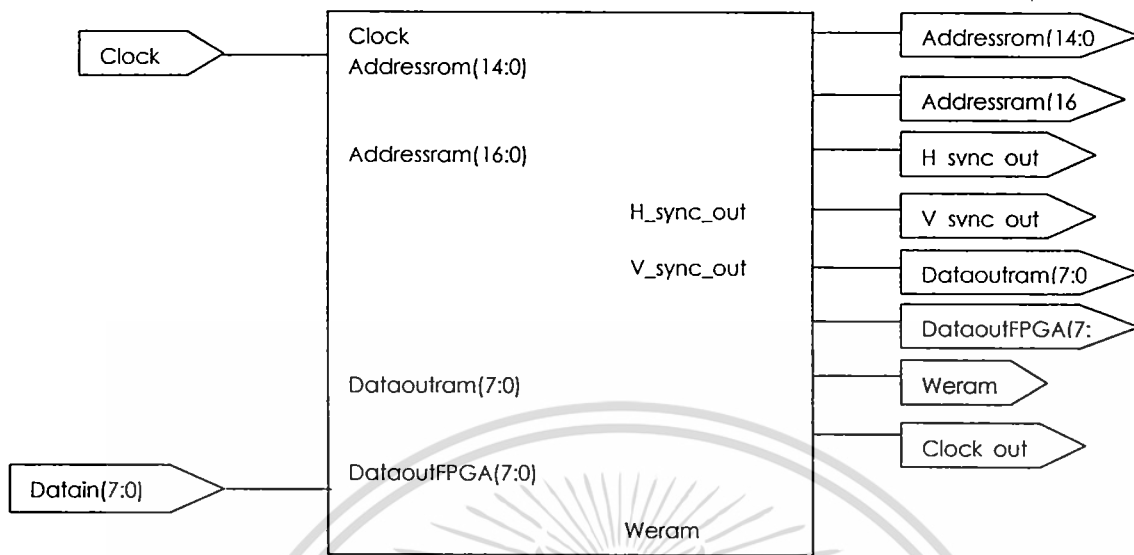
รูปที่ 4.29 ภาพต้นฉบับ (64x64)

Interpolated Image by DPT Local (L=2)



รูปที่ 4.30 ภาพที่ผ่านการขยายแบบ DPT Interpolation (2 เท่า)

เอกสาร 4.2 นี้ ออกแบบวงจร โดยใช้ภาษาวีเอชดีแอล เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

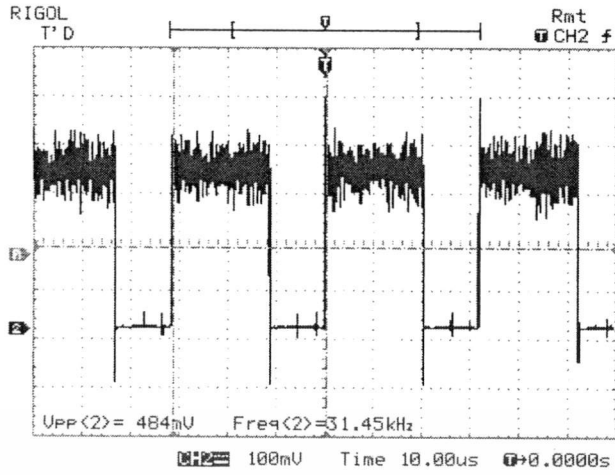


รูปที่ 4.31 Symbol ส่วนของโครงสร้างรวมของภาษาVHDL

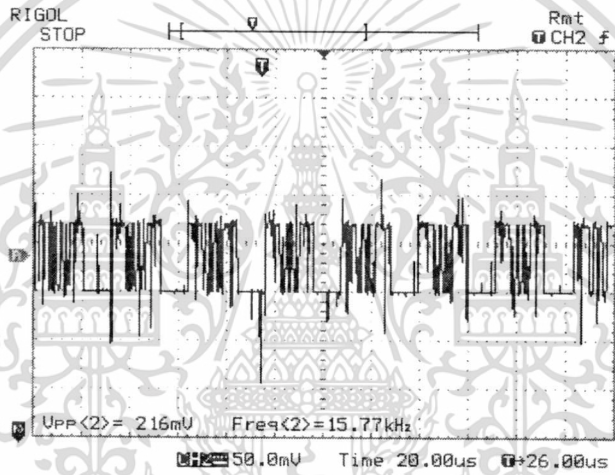
จากรูปที่ 4.30 เป็นโมดูลของวงจรที่ใช้ในการ Interpoate โดยจะอ่านข้อมูลจากรอมแล้วนำข้อมูลมาทำการ Interpolate ใส่แรมและในโครงสร้างนี้ก็จะประกอบไปด้วยโมดูลต่างๆที่ใช้ในการออกแบบ ซึ่งวงจรหลักๆคือ วงจรส่วนควบคุมหน้าจอดีวีเอ วงจรขยายสัญญาณภาพ และ วงจรอ่านตำแหน่งของหน่วยความจำภายนอก อยู่ในกระบวนการ

4.3 ผลการทดลองที่ได้จากออสซิลโลสโคป

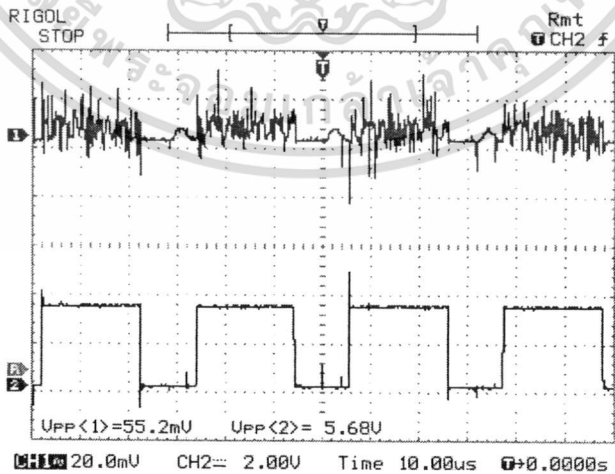
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 สัญญาณวิดีโอ

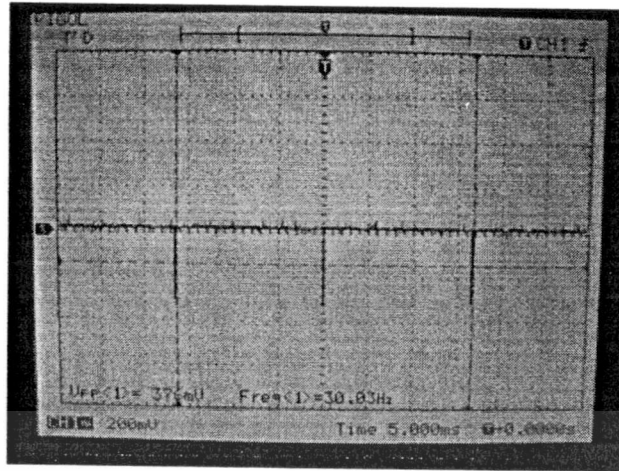


รูปที่ 4.33 สัญญาณข้อมูลที่ออกจาก TDA8702

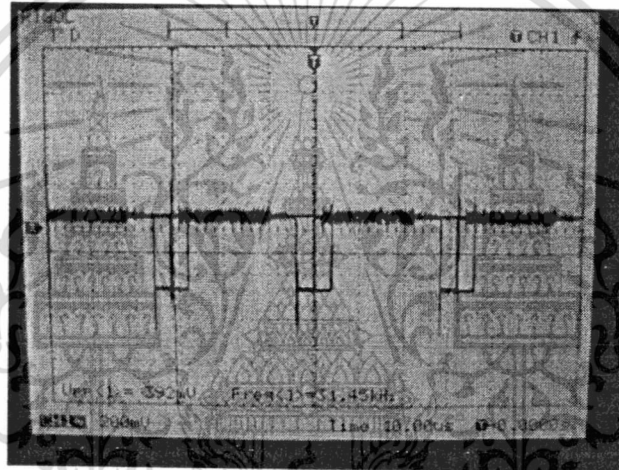


รูปที่ 4.34 สัญญาณวิดีโอวัดเทียบสัญญาณข้อมูลที่ออกจาก TDA8702

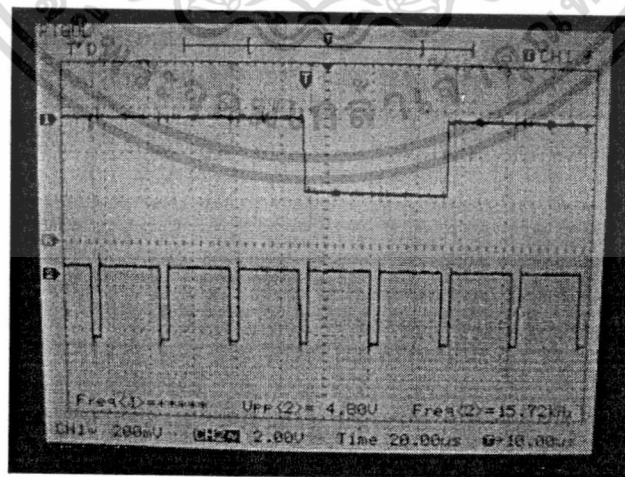
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 สัญญาณควบคุมการสแกนทางแนวตั้ง



รูปที่ 4.36 สัญญาณควบคุมการสแกนทางแนวนอน



รูปที่ 4.37 เมื่อทำการวัดสัญญาณควบคุมการสแกนทางแนวนอนเปรียบเทียบกับสัญญาณควบคุมการสแกนทางแนวตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

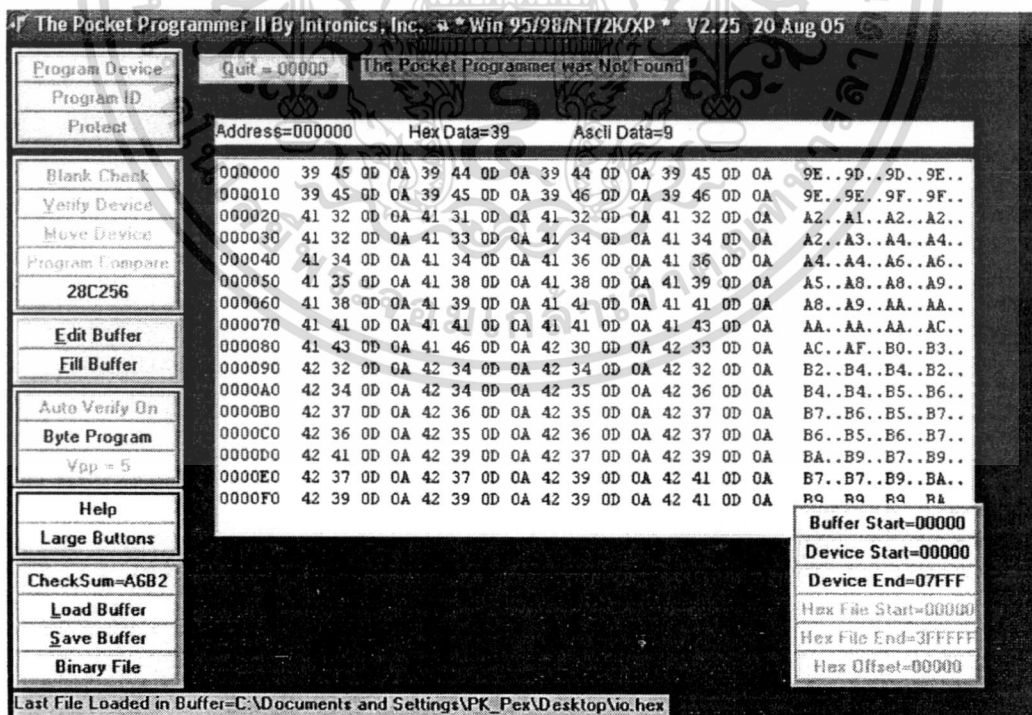
4.4 ขั้นตอนการทดลองที่ใช้ FPGA

การทดลองจะแบ่งเป็นขั้นตอนต่างๆดังนี้

1. การทำการทดลองโดยใช้หน่วยความจำภายนอกที่เป็น EEPROM เพียงอย่างเดียวเพื่อทำการอินเตอร์โพลด้วยวิธีการ Replication pixel
2. การทำการทดลองโดยใช้หน่วยความจำภายนอกทั้ง EEPROM และ RAM เพื่อทำการอินเตอร์โพลด้วยวิธีการ Linear interpolation และ Replication pixel
3. การใช้ FPGA ติดต่อกับกล้องโดยมีการเก็บค่าที่ได้จากกล้องไว้ใน RAM โดยใช้ไมโครคอนโทรลเลอร์ในการประมวลผลรับข้อมูลส่งให้ FPGA เพื่อทำการประมวลผลก่อนนำข้อมูลออกไปยังหน้าจอวีซีเอ

4.4.1 การใช้หน่วยความจำภายนอก (EEPROM)

เพื่อที่จะใช้เก็บข้อมูลแต่ละพิกเซลของภาพซึ่งแต่ละพิกเซลจะมีขนาด 8 บิต



รูปที่ 4.38 หน้าต่างโปรแกรมของหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

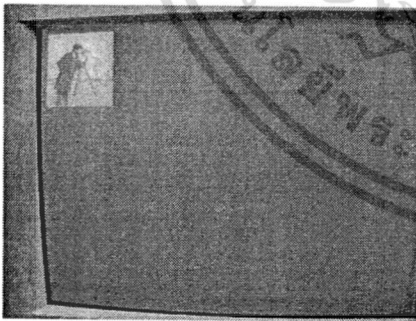
4.4.2 การแสดงข้อมูลภาพโดยใช้หน่วยความจำภายนอก EEPROM เพียงอย่างเดียว

เปรียบเทียบภาพขนาดปกติกับภาพที่ถูกขยายด้วยวิธีการ replication pixel โดยการนำเวลาการแสดงผลของข้อมูลที่ได้จาก EEPROM

รูปตัวอย่าง 1



(ก)



(ข)



(ค)

ก) รูปต้นแบบ

ข) ภาพที่แสดงผลทางหน้าจอวีซีเอ

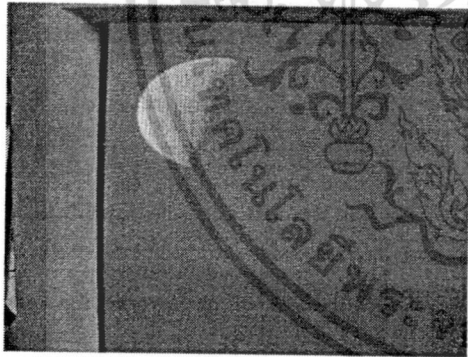
ค) ภาพที่ภาพที่ถูกขยายด้วยวิธีการ Replication pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

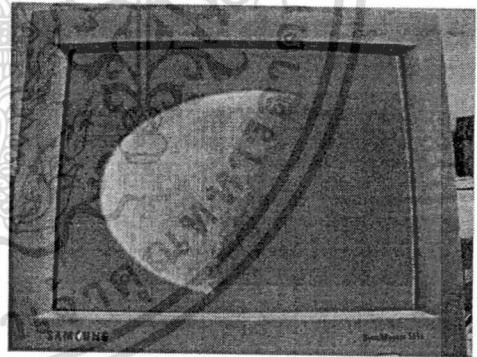
รูปตัวอย่าง 2



(ก)



(ข)



(ค)

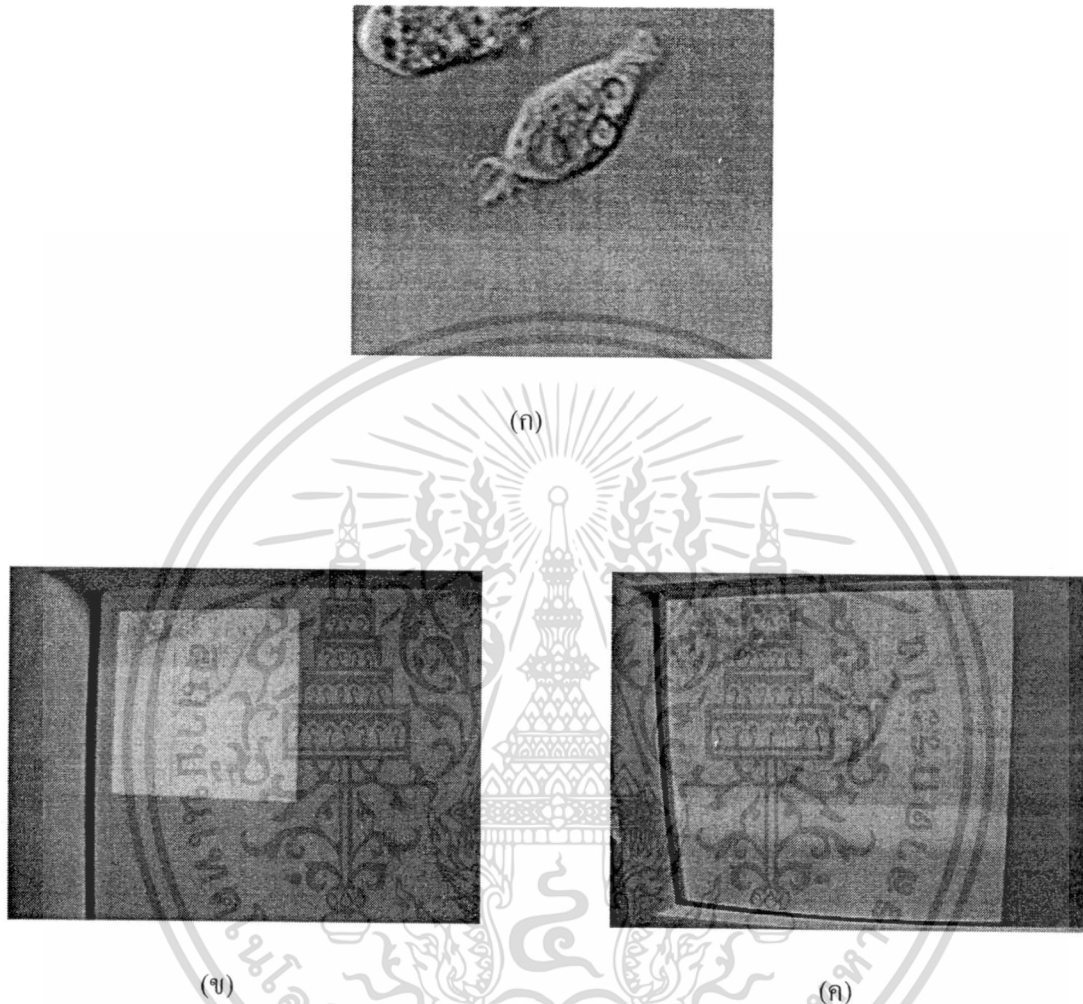
ก) รูปต้นแบบ

ข) ภาพที่แสดงผลทางหน้าจอวีจีเอ

ค) ภาพที่ ถูกขยายด้วยวิธีการ Replication pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปตัวอย่าง 3



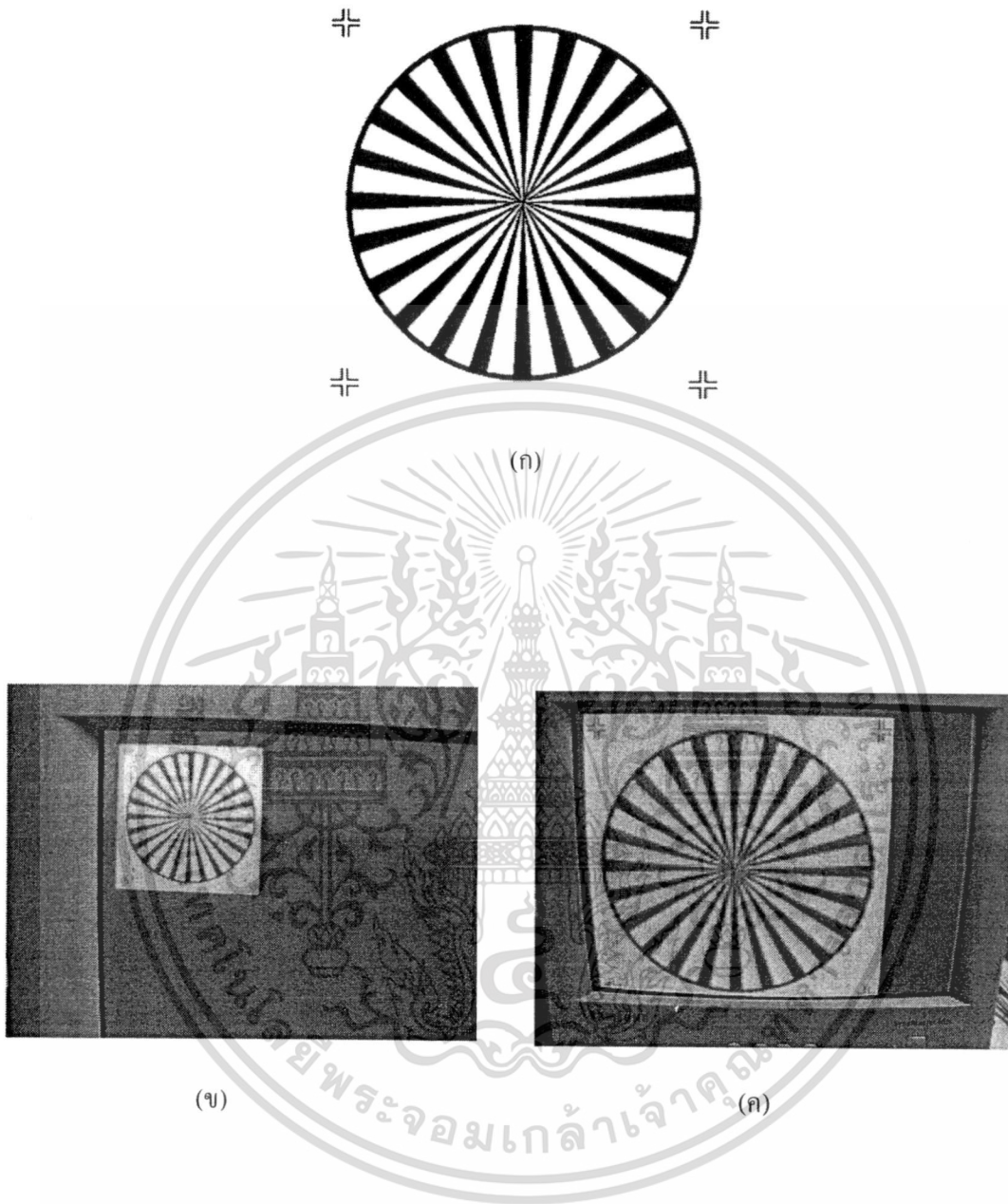
ก) รูปต้นแบบ

ข) ภาพที่แสดงผลทางหน้าจอวีจีเอ

ค) ภาพที่ ถูกขยายด้วยวิธีการ Replication pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปตัวอย่าง 4



ก) รูปต้นแบบ

ข) ภาพที่แสดงผลทางหน้าจอวีซีเอ

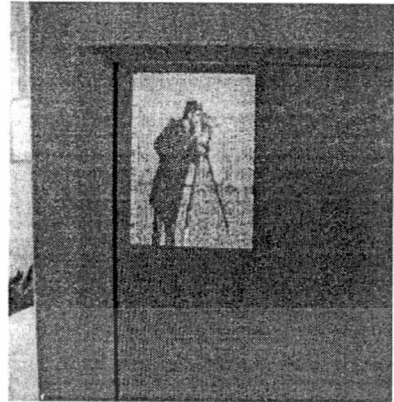
ค) ภาพที่ภาพที่ถูกขยายด้วยวิธีการ Replication pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

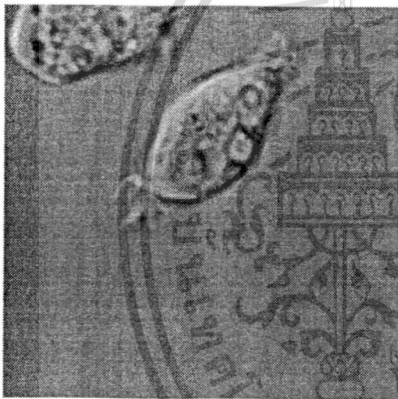
4.4.3 การแสดงผลข้อมูลภาพโดยใช้หน่วยความจำภายนอก EEPROM ร่วมกับ RAM



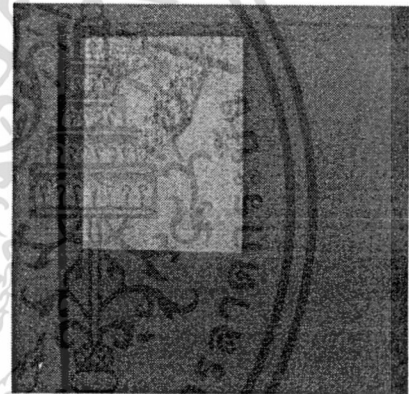
(ก)



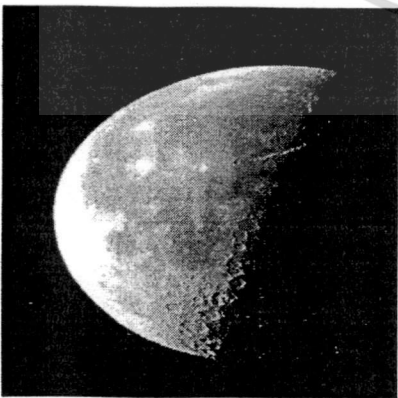
(ข)



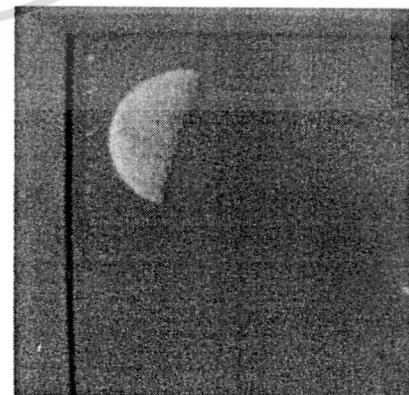
(ค)



(ง)

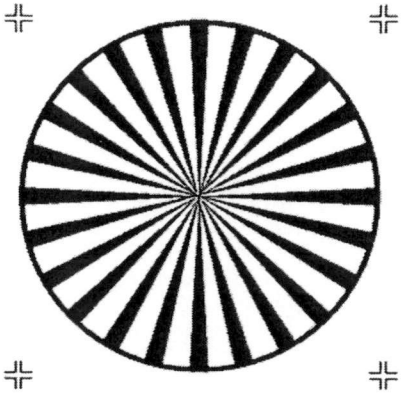


(จ)

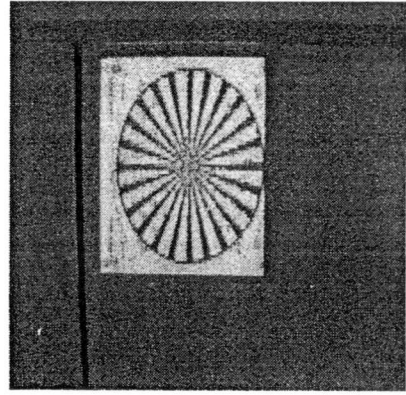


(ฉ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



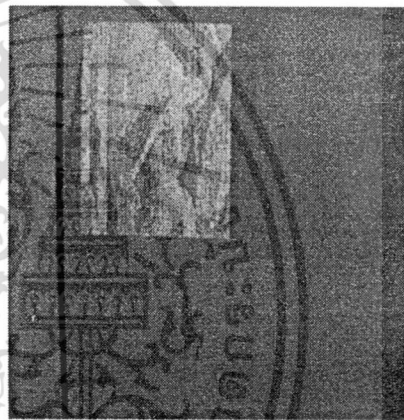
(ข)



(ช)



(ฉ)



(ญ)

รูปที่ 4.39 ภาพการเปรียบเทียบการแสดงผลข้อมูลจาก RAM ในขนาดปกติกับภาพต้นแบบ

ก) ภาพต้นแบบ

ข) ภาพที่แสดงผลข้อมูลจาก RAM

ค) ภาพต้นแบบ

ง) ภาพที่แสดงผลข้อมูลจาก RAM

จ) ภาพต้นแบบ

ฉ) ภาพที่แสดงผลข้อมูลจาก RAM

ช) ภาพต้นแบบ

ซ) ภาพที่แสดงผลข้อมูลจาก RAM

ฅ) ภาพต้นแบบ

ญ) ภาพที่แสดงผลข้อมูลจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

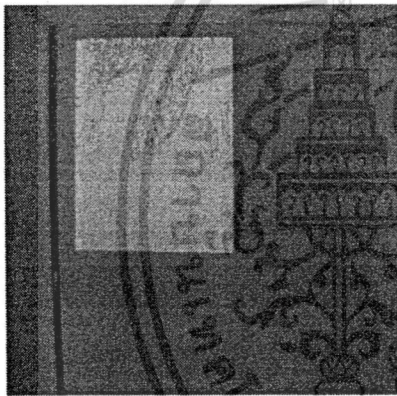
4.4.4 การขยายข้อมูลภาพโดยวิธี replication pixel ผ่าน RAM



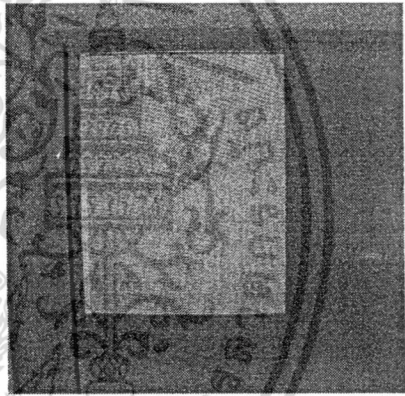
(ก)



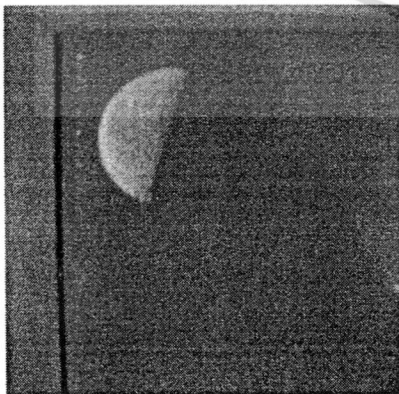
(ข)



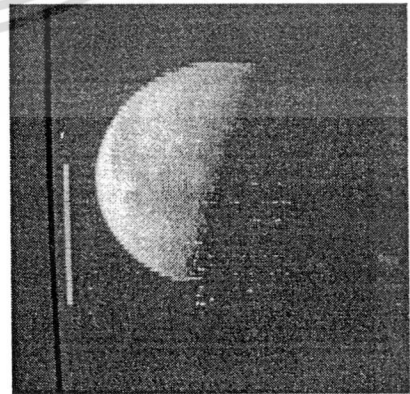
(ค)



(ง)

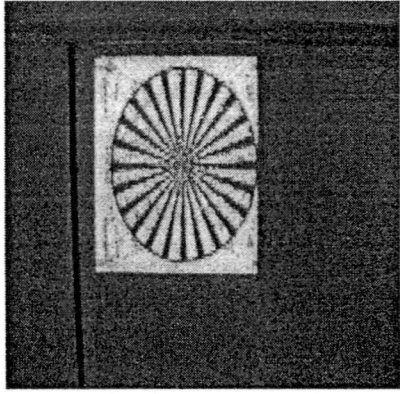


(จ)

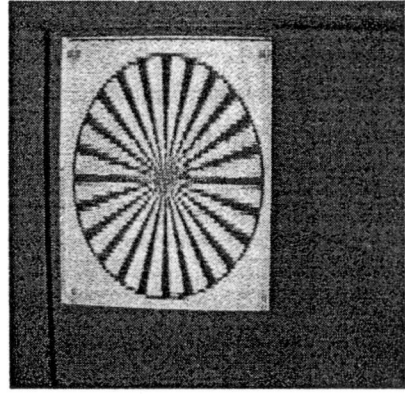


(ฉ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)



(ช)



(ฉ)



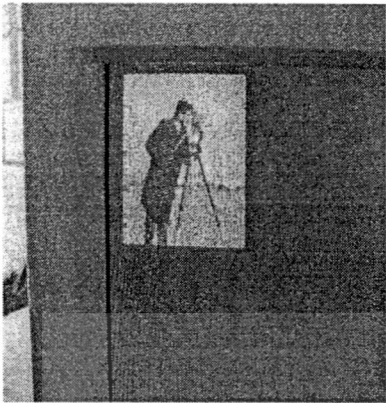
(ญ)

รูปที่ 4.39 ภาพการเปรียบเทียบการแสดงผลข้อมูลจาก RAM ในขนาดปกติกับภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM

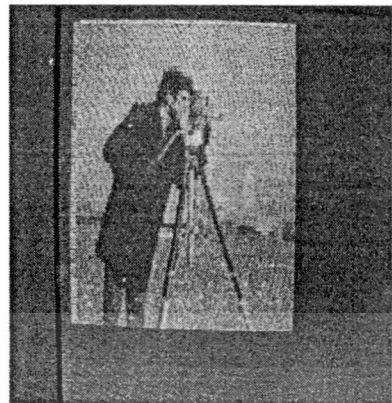
- ก) ภาพที่แสดงข้อมูลจาก RAM
- ข) ภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM
- ค) ภาพที่แสดงข้อมูลจาก RAM
- ง) ภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM
- จ) ภาพที่แสดงข้อมูลจาก RAM
- ฉ) ภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM
- ช) ภาพที่แสดงข้อมูลจาก RAM
- ช) ภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM
- ฉ) ภาพที่แสดงข้อมูลจาก RAM
- ญ) ภาพที่ถูกขยายด้วยวิธี replication pixel จาก RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.5 การขยายข้อมูลภาพโดยวิธี Linear Interpolation ผ่าน RAM



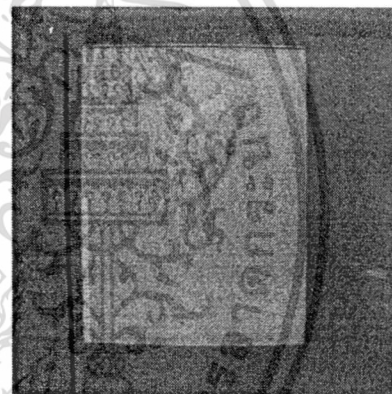
(ก)



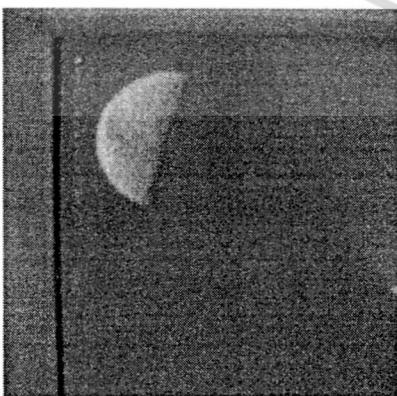
(ข)



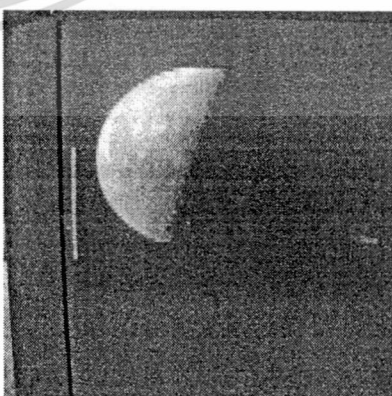
(ค)



(ง)

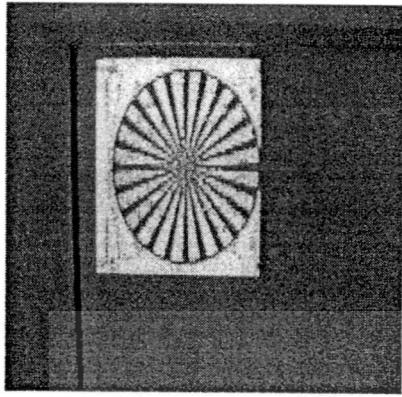


(จ)

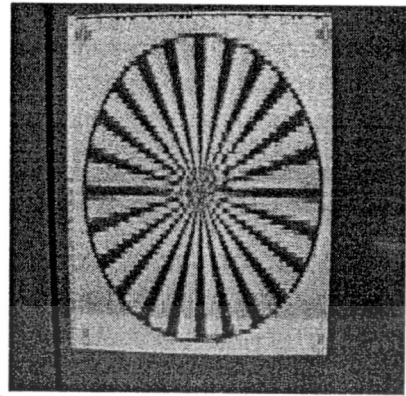


(ฉ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ซ)



(ซ)



(ฅ)



(ฅ)

รูปที่ 4.41 ภาพการเปรียบเทียบการแสดงผลข้อมูลจาก RAM ในขนาดปกติกับภาพที่ถูกขยายด้วยวิธี Linear Interpolation

- ก) ภาพที่แสดงข้อมูลจาก RAM ข) ภาพที่ถูกขยายด้วยวิธี Linear Interpolation จาก RAM
- ค) ภาพที่แสดงข้อมูลจาก RAM ง) ภาพที่ถูกขยายด้วยวิธี Linear Interpolation จาก RAM
- จ) ภาพที่แสดงข้อมูลจาก RAM ฉ) ภาพที่ถูกขยายด้วยวิธี Linear Interpolation จาก RAM
- ช) ภาพที่แสดงข้อมูลจาก RAM ซ) ภาพที่ถูกขยายด้วยวิธี Linear Interpolation จาก RAM
- ฅ) ภาพที่แสดงข้อมูลจาก RAM ญ) ภาพที่ถูกขยายด้วยวิธี Linear Interpolation จาก RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.6 การแสดงผลภาพจากกล้องโดยใช้ RAM เป็นตัวเก็บข้อมูล



รูปที่ 4.42 ภาพการเปรียบเทียบภาพต้นแบบกับภาพที่ได้จากการรับข้อมูลจากกล้อง

ก) ภาพต้นแบบ

ข) ภาพที่ได้จากการรับข้อมูลจากกล้อง

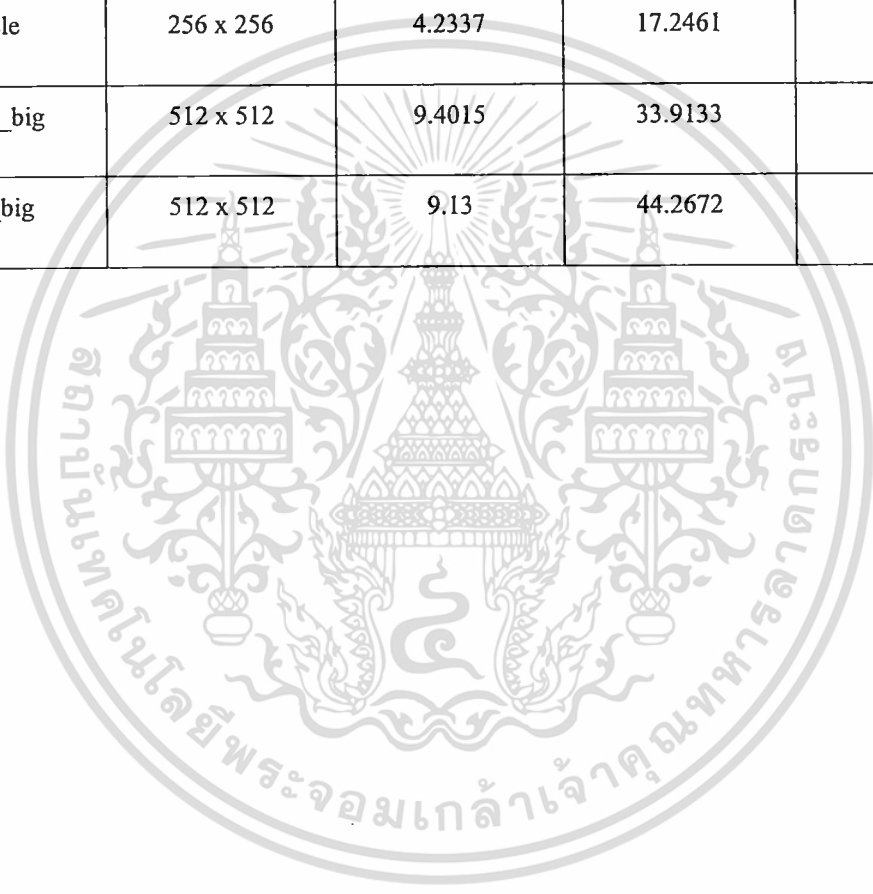
ค) ภาพต้นแบบ

ง) ภาพที่ได้จากการรับข้อมูลจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ตารางแสดงค่า PSNR ของการอินเทอร์โพลเทภาพแบบต่างๆ ใน FFT และ DPT

Image	Size	FFT	DPT	%
Lenna	128 x 128	9.3609	22.4507	13
Cell	128 x 128	9.1269	33.4278	24
Cameraman	256 x 256	8.0671	23.5262	15
Circle	256 x 256	4.2337	17.2461	13
Lenna_big	512 x 512	9.4015	33.9133	24
Cell_big	512 x 512	9.13	44.2672	35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและวิจารณ์การทดลอง

5.1 ส่วนผลการทดลองบนแมทแลป

จากการทดลองการอินเทอร์โพลสภาพดิจิทัลโดยใช้แมทแลปช่วยในการประมวลผล ได้ผลการทดลองเป็นดังนี้

1. Replication Pixel

สำหรับวิธีการนี้ เป็นส่วนที่นำไปทดลองโดยโปรแกรมลงบอร์ด FPGA ซึ่งผลที่ได้จะเป็นผลที่ไม่ดีนัก เนื่องจากการเพิ่มข้อมูลนั้น ไม่มีการลงรายละเอียดของน้ำหนักระหว่างข้อมูลเดิมแต่ใช้ข้อมูลเดิมซ้ำลงไปทำให้ผลที่ออกมาไม่ค่อยเหมาะสมนัก

2. Bilinear interpolation

สำหรับวิธีการนี้ ทำการเขียนโปรแกรมใน Matlab เพื่อจำลองผลการทดลอง โดยในส่วนของสัญญาณ เมื่อผ่านการขยายพบว่าการเพิ่มค่าแซมปลิง โดยค่าของแซมปลิงที่เพิ่มจะมีการเฉลี่ยจากสัญญาณแซมปลิงบริเวณข้างเคียง และในส่วนของภาพนั้น จะได้ภาพที่มีรายละเอียดที่ดีขึ้นกว่าการคัดลอกค่าเดิมซ้ำ เนื่องจากมีการเฉลี่ยค่าของสี่บริเวณข้างเคียง เพื่อนำมาเพิ่มลงในส่วนของภาพที่ขยาย

3. Fast Fourier Transform Interpolation

สำหรับวิธีการนี้ ทำการเขียนโปรแกรมใน Matlab เพื่อจำลองผลการทดลอง โดยในส่วนของสัญญาณ เมื่อผ่านการขยายโดยการแปลงข้อมูลให้อยู่ในรูปของโดเมนความถี่ พร้อมทั้งทำการเพิ่มค่าศูนย์ (Zero padding) และแปลงกลับมาในโดเมนของเวลาพบว่าการเพิ่มค่าแซมปลิง โดยค่าของแซมปลิงที่เพิ่มจะมีการเฉลี่ยจากสัญญาณแซมปลิงบริเวณข้างเคียง และส่วนของภาพนั้น จะได้ภาพที่มีรายละเอียดที่ดีขึ้นเนื่องจากมีจำนวนการแซมปลิงที่มากขึ้น ทำให้ภาพมีความละเอียด

4. Discrete Pascal Transform Interpolation

สำหรับวิธีการนี้ ทำการเขียนโปรแกรมใน Matlab เพื่อจำลองผลการทดลอง โดยแบ่งออกเป็นแบบโกบอล และแบบโลคอล ในส่วนของโกบอลนั้น เมื่อสัญญาณ ผ่านการขยายแล้วพบว่าการเกิดความผิดพลาดบริเวณของขอบสัญญาณ ส่วนภาพที่ได้จากโกบอลนั้นมีความผิดพลาดเกิดขึ้นที่บริเวณขอบเป็นขนาดใหญ่ อาจเกิดขึ้นเนื่องจากปาสคาลเมทริกซ์ที่นำมาทำการคูณด้วยนั้น มีขนาดใหญ่มากเกินไป ดังนั้นการอินเทอร์โพลด้วยวิธีโกบอลนี้ไม่สามารถใช้กับข้อมูลมีขนาดใหญ่ได้ จึงต้องแก้ไขด้วยวิธีการแบบโลคอลคือ แบ่งภาพออกเป็นส่วนๆ และทำการอินเทอร์โพลเช่นกันในแต่ละส่วนย่อย ให้แล้วเสร็จก่อน จึงค่อยนำมาประกอบกลับเป็นภาพขนาดใหญ่ ซึ่งผลที่ได้จะดีกว่าแบบโกบอลมาก หากข้อมูลมีขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Peak Signal to Noise Ratio

เป็นวิธีการวัดคุณภาพของภาพที่ได้จากการอินเทอร์โพลเลท ซึ่งในที่นี้ทำการเปรียบเทียบผลที่ได้จากการอินเทอร์โพลเลทแบบ DPT และ FFT ซึ่งค่าที่ได้จากวิธีการอินเทอร์โพลเลทแบบ DPT นั้นให้ค่าที่มากกว่าแบบ FFT ซึ่งหมายถึงว่าการอินเทอร์โพลเลทแบบ DPT นั้นจะให้ผลออกมาที่มีคุณภาพที่ดีกว่าแบบ FFT โดยที่หากเราทำการเพิ่มขนาดของภาพที่จะนำมาทำการอินเทอร์โพลเลทให้มีขนาดใหญ่ขึ้นจะพบว่า วิธีการอินเทอร์โพลเลทแบบ FFT นั้น ให้ค่า PSNR เกือบจะเท่าเดิม แต่ในวิธีของ DPT นั้นให้ค่าที่มากกว่าภาพเดียวกันที่มีขนาดเล็กกว่า

5.2 ส่วนผลการทดลองบนฮาร์ดแวร์ (FPGA)

การทดลองอินเทอร์โพลเลทด้วยวิธีต่างๆด้วย FPGA

1. Replication Pixel

จะแบ่งเป็นสองแบบคือแบบที่ทำการหน่วงเวลาเพื่อการแสดงผลข้อมูลเดิมและแบบที่ทำการเรียกข้อมูลเดิมขึ้นมาจริงๆซึ่งในแบบที่สองนั้นมีความจำเป็นในการใช้แรมเพื่อใช้อ่านและเขียนข้อมูลเพื่อประมวลผลทำให้เกิดความผิดพลาดของข้อมูลขึ้นเนื่องจากการอ่าน-เขียนแรมสลับกันไปมาทำให้ภาพอาจจะเกิดความผิดพลาดได้

2. Linear Interpolation

ในการอินเทอร์โพลเลทแบบนี้ปัญหาก็จะเกิดขึ้นจากการผิดพลาดของข้อมูลจากการอ่าน-เขียนแรมเช่นกันแต่คุณภาพนั้นจะดีกว่าแบบ Replication Pixel เนื่องจากการเฉลี่ยข้อมูลเกิดขึ้น

สำหรับการติดต่อกับตัวกล้องนั้นการติดต่อกับโมดูลกล้อง iv-cam โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวรับข้อมูลส่งให้กับ FPGA นั้นสรุปได้ว่า FPGA สามารถรับข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์เพื่อแสดงผลได้ แต่ว่าข้อมูลที่ได้นั้นอาจจะมีปัญหาเนื่องจากภาพที่แสดงผลได้นั้นเป็นแค่เพียงภาพที่มีลักษณะของภาพที่ถ่ายแต่ไม่มีรายละเอียดที่ชัดเจนซึ่งปัญหานี้จะเกิดขึ้นเนื่องจาก

1. ตัวโมดูลกล้อง iv-cam นั้นมีการจับพอร์ตคำสั่งส่งภาพเพียงสำหรับ Baud Rate ที่มีค่า 115200 ซึ่งไมโครคอนโทรลเลอร์ที่ใช้ไม่สามารถทำได้
2. การเก็บภาพของโมดูล iv-cam นั้นไม่เหมาะกับการนำมาประมวลผลภาพเนื่องจากภาพมีความละเอียดไม่มากนักเมื่อนำมาแปลงค่าให้เป็นค่าเกรย์สเกลอาจจะมีความผิดพลาดได้

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.
4. Wikipedia, the free encyclopedia. "EPROM"
Available : <http://www.wikipedia.com>
5. HWb, "VGA (15)"
Available : [http://VGA_\(15\)_-HwB.htm](http://VGA_(15)_-HwB.htm)
6. Wikipedia, the free encyclopedia. "VGA USB 1394 firewire EVC"
Available : <http://www.wikipedia.com>
7. Wikipedia, the free encyclopedia. "Peak Signal-to-noise Ratio"
Available : <http://www.wikipedia.com>
8. Electoday, Electronics and Microcontroller. "Image Processing"
Available : <http://www.electoday.com>
9. knowledge of image processing, "Measures of image quality"
Available : <http://ubiety.uwaterloo.ca/~tveldhui/papers/MAScThesis/node18.html>
10. ชำนาญ ปัญญาใส, วัชรกร หนูทอง, "ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล", ซีเอ็ดยูเคชั่น, กรุงเทพฯ, 2547.
11. มนัส สัจจวิไล, วรรัตน์ภัทรอมรกุล, "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์", สำนักพิมพ์ อินโฟเพรส, กรุงเทพฯ, 2543.
12. วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิมพรจิตรวิไล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51", อินโนเวตีฟ เอ็ดจอร์เม้นต์, กรุงเทพฯ, 2545.
13. อุปกรณ์คอมพิวเตอร์ Hardware, "ROM"
Available : <http://bcoms.net>