สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เม้าส์สามมิติ

Mouse 3D

T050399

โดย

นาย ชาญนสิริ เกียรติมาลาวงศ์

นาย ชาตรี คิรินทร์รัตนะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50399
วัน,เดือน,ปี 13 พ.ค. 2547
.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เม้าส์สามมิติ

Mouse 3D

โดย

นาย ชาญนสิริ เกียรติมาลาวงศ์ รหัส 42010083

นาย ชาตรี คิรินทร์รัตนะ รหัส 42010085

อาจารย์ที่ปรึกษา

อาจารย์ เทอดศักดิ์ ลิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2545

ภาควิชา อิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง MOUSE 3D

ผู้จัดทำ

1. นาย ชาญนสิริ เกียรติมาลาวงศ์ รหัส 42010083
2. นาย ชาตรี ศิรินทร์รัตนะ รหัส 42010085

..... อาจารย์ที่ปรึกษา

(เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เม้าส์สามมิติ

MOUSE 3D

ผู้จัดทำ

1. นาย ชาญนสิริ เกียรติมาลาวงศ์ รหัส 42010083
2. นาย ชาตรี ศิรินทร์รัตนะ รหัส 42010085

ปริญญานิพนธ์ฉบับนี้ได้รับการตรวจสอบแล้วพร้อมที่จะทำการสอบได้

.....

(เทอดศักดิ์ ลิ่วหาทอง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เม้าส์สามมิติ

ชาญนสิริ เกียรติมาลาวงศ์
ชาตรี คิรินทร์รัตนะ
อ. เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอระบบการหาพิกัดที่ถูกต้องในรูปของสามมิติ จากตัวส่งสัญญาณอัลตร้าโซนิกไปยังตำแหน่งรับสัญญาณอัลตร้าโซนิก โดยอาศัยผลต่างของเวลาที่ใช้ในการส่งจากตัวส่ง ไปยังตัวรับทั้ง4ตัว ที่ติดตั้งไว้ที่พิกัดที่รู้ค่าของตำแหน่งแน่นอน เริ่มต้นโดยการใช้ผลต่างของเวลา ระหว่างตัวรับซึ่งวัดโดยใช้บอร์ดFPGA และส่งค่าไปยังเครื่องคอมพิวเตอร์โดยอาศัยพอร์ตขนาน และจึงนำค่าที่ได้ไปคำนวณหาพิกัดของตัวส่งและแสดงพิกัดขึ้นทางเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mouse 3D

Channasiri Kaitmalawong

Chartre Kirimurttana

Thurdsak Leauhatong Advisor

2002

ABSTACT

The thesis presents a systems for acculately estimating position of an ultrasonic transmitter in 3D space. The position is estimated by using differences in the time of flights(DTOFs) from the transmitter to four receivers fixed in known coordinates. First the DTOFs of the recivers are measured by using a field program gate array(FPGA). Then they are transferred to Personal Computer(PC) via a PC 's parallel port . Finally the coordinate of the transmitter is calculated and presented in PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ

Abstract

สารบัญ

สารบัญภาพ

บทที่1 บทนำ 1

บทที่2 ส่วนประกอบของโครงงาน 2

บทที่3 ทฤษฎีที่ใช้ในการออกแบบ 4

3.1 ทฤษฎีเกี่ยวกับคลื่นอัลตร้าโซนิค 4

3.2 อัลตร้าโซนิคทรานสดิวเซอร์ 10

3.3 ความรู้เบื้องต้นของพอร์ทขนาน 15

3.4 Field Programmable Gate Array (FPGA) 23

บทที่4 แนวความคิดในการออกแบบ 29

4.1 เป้าหมายของโครงงาน 29

4.2 ตำแหน่งของตัวรับและตัวส่ง 29

4.3 การคำนวณหาพิกัด X,Y,Z 31

4.4 ประโยชน์ของ T_{max} 32

บทที่5 การออกแบบและการทำงานของเม้าส์สามมิติ 34

5.1 แนวทางการออกแบบเม้าส์สามมิติ 34

5.2 การทำงานของเม้าส์สามมิติ 36

5.3 วงจรภาคประมวลผล 42

5.4 หลักการทำงานของส่วนFPGA 46

บทที่6 การทดลองและผลการทดลอง 48

6.1 ทดสอบหาระยะทางและมุมที่ตำแหน่งต่างๆของภาคส่งและภาครับ 48

6.2 ทดสอบการทำงานของส่วนแสดงผล 49

บทที่7 สรุปผลการทดลอง 53

7.1 สรุปผลการทดลอง 53

7.2 ปัญหาที่พบขณะทดลอง 54

7.3 แนวทางการแก้ปัญหา 54

กิตติกรรมประกาศ

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่1 แนวคิดในการออกแบบ 1

รูปที่2.1 แสดงการวางตำแหน่งของตัวส่งและตัวรับ 2

รูปที่2.2 ภาพแสดงการทำงานของส่วน FPGA 3

รูปที่2.3 ภาพแสดงการทำงานของส่วนประมวลและแสดงผล 3

รูปที่3.1 แสดงถึงช่วงความถี่ต่างๆที่ถูกนำไปใช้งาน 4

รูปที่3.2 แสดงลักษณะการเกิดคลื่นตามยาว 5

รูปที่3.3 แสดงลักษณะการเกิดคลื่นตามขวาง 6

รูปที่3.4 แสดงการบีสต์ของคลื่นเสียง 2 คลื่น 7

รูปที่3.5 แสดงลักษณะการเกิดคลื่นนิ่ง 7

รูปที่3.6 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบๆ จะเสมือนเป็นแหล่งกำเนิดตัวใหม่ 8

รูปที่3.7 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน 8

รูปที่3.8 แสดงเมื่อคลื่นวงกลมกระทบกับฉากที่เป็นเส้นตรงคลื่นที่สะท้อนกลับมา จะเป็นคลื่นวงกลมเช่นเดียวกัน 8

รูปที่3.9 ผลของเปียโซอิเลคทริค 11

รูปที่3.10 วงจรเสมือนของเปียโซอิเลคทริค 11

รูปที่3.11 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์ 15

รูปที่3.12 วงจรภายในของพอร์ต Data 18

รูปที่3.13 รายละเอียดวงจรภายในของพอร์ต Control 20

รูปที่3.14 โครงสร้างภายใน FPGA ตระกูล MAX7000S 24

รูปที่3.15 โครงสร้างภายในของ FPGA ตระกูล FLEX10K 24

รูปที่4.1 แสดงการวางของตัวส่งและตัวรับ 29

รูปที่4.2 แสดงระยะห่างของตัวรับแต่ละตัว 32

รูปที่5.1 แสดงวงจรภาคส่ง 36

รูปที่5.2 แสดงสัญญาณที่จุดA ในรูปที่5.1 37

รูปที่5.3 แสดงสัญญาณที่จุดB ในรูปที่5.2 38

รูปที่5.4 แสดงภาครับคลื่น Ultrasonic 38

รูปที่5.5 แสดงสัญญาณที่จุดC ในรูปที่5.4 39

รูปที่5.6 แสดงสัญญาณที่จุดD ในรูปที่5.4 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่5.7 แสดงสัญญาณที่จุดE ในรูปที่5.4 40
รูปที่5.8 แสดงสัญญาณที่จุดF ในรูปที่5.4 41
รูปที่5.9 แสดงสัญญาณที่จุดG ในรูปที่5.4 41
รูปที่5.10 แสดงการวางตำแหน่งของตัวส่งและตัวรับ 42
รูปที่6.1 แสดงหน้าต่างขณะเปิดใช้งานของโปรแกรม 49
รูปที่6.2 แสดงภาพสามเหลี่ยมที่จุดเริ่มต้น 50
รูปที่6.3 แสดงการเคลื่อนของภาพในแนวแกนX 50
รูปที่6.4 แสดงภาพสามเหลี่ยมวัตถุที่จุดเริ่มต้น 51
รูปที่6.5 แสดงการเคลื่อนของภาพในแนวแกนY 51
รูปที่6.6 แสดงภาพสี่เหลี่ยมที่จุดเริ่มต้น 52
รูปที่6.7 แสดงการเคลื่อนของภาพในแนวแกนZ 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 สัญญาณสำคัญๆของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์ 16

ตารางที่ 3.2 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน 17

ตารางที่ 3.3 แสดงแอดเดรสของพอร์ตขนาน 22

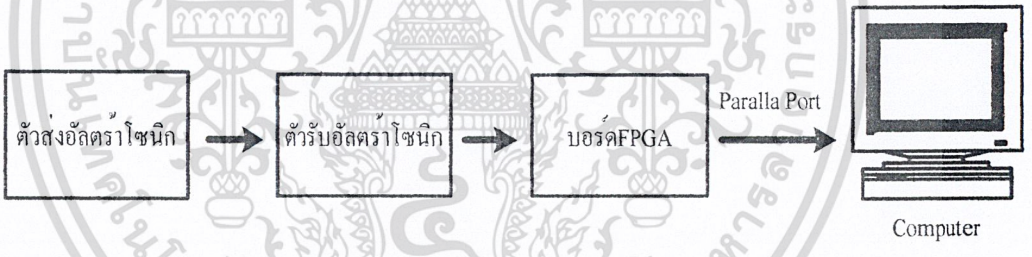
ตารางที่ 6.1 แสดงการทดลองวัดค่าตำแหน่งและมุมของภาคส่งและภาครับ 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ปริญญานิพนธ์ฉบับนี้เสนอการออกแบบระบบ เพื่อใช้บอกตำแหน่งของตัวชี้ตำแหน่ง (X,Y,Z) บนจอคอมพิวเตอร์ โดยอาศัยหลักการทำงานของคลื่น Ultrasonic เริ่มต้นด้วยการส่งสัญญาณจากตัวส่ง(ตัวชี้ตำแหน่ง) มาที่ตัวรับทั้ง 4 ตัว ซึ่งจัดวางอยู่ในรูปสี่เหลี่ยม จะรับสัญญาณของตัวส่งได้ในเวลาที่ต่างกัน เราจึงนำผลต่างของเวลาที่รับได้ส่งเข้าไปที่ FPGA ไปเปรียบเทียบกับสัญญาณนาฬิกา เพื่อนับผลต่างของเวลาในรูปสัญญาณนาฬิกา

จำนวนผลต่างของสัญญาณนาฬิกา ที่ได้ส่งต่อไปคำนวณระยะทางที่คอมพิวเตอร์ผ่านทางพอร์ตขนาน โดยค่าของสัญญาณนาฬิกาที่ต่างกันจะนำไปคำนวณระยะทางในพิกัด X,Y,Z โดยใช้โปรแกรม Visual Basic และแสดงผลออกเป็นภาพสามมิติที่ต้องการ ต่อไป



รูปที่ 1 แนวคิดในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ส่วนประกอบของโครงงาน

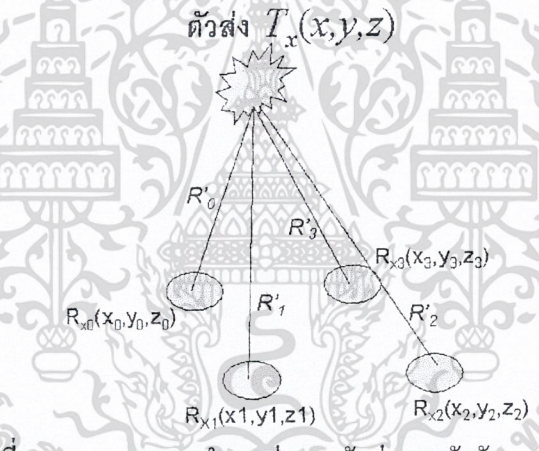
จากที่ได้กล่าวถึงการทำงานของMouse3D ไปพอสังเขปแล้วในบทนำ ในบทนี้จะกล่าวถึงการทำงานโดยละเอียดในแต่ละส่วน โดยจะกล่าวเป็นถึงส่วนประกอบของโครงงานให้ทราบเสียก่อนว่าประกอบไปด้วยอะไรบ้างแล้วจะกล่าวถึงหลักการทำงานต่อไป

2.1 ส่วนประกอบของโครงงาน

ในโครงงานนี้จะประกอบไปด้วย 3ส่วนหลักๆคือ

2.1.1 ชุดรับ-ส่งสัญญาณอัลตร้าโซนิค

ซึ่งจะแบ่งออกไปอีกคือ ตัวส่ง และ ตัวรับ สัญญาณอัลตร้าโซนิค



รูปที่ 2.1 แสดงการวางตำแหน่งของตัวส่งและตัวรับ

ตัวส่งสัญญาณอัลตร้าโซนิค

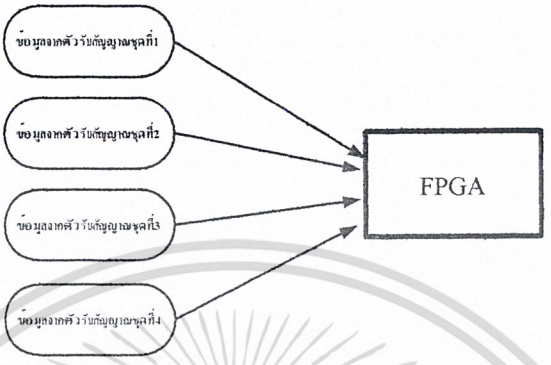
ส่วนนี้ทำหน้าที่ส่งคลื่นสัญญาณอัลตร้าโซนิค(ความถี่40kHz) ออกไป เพื่อเป็นตัวกำหนดพิกัดของตัวส่งสัญญาณว่าอยู่ที่ตำแหน่งใด โดยวงจรส่วนนี้มีหลักการทำงานซึ่งจะแสดงให้หัวข้อต่อไป

ตัวรับสัญญาณอัลตร้าโซนิค

วงจรส่วนนี้ทำหน้าที่รับสัญญาณอัลตร้าโซนิคที่ตัวส่งๆออกมา โดยตัวรับจะแบ่งออกเป็น4จุดวางตัวเป็นรูป 4 เหลี่ยม ตัวรับแต่ละตัวจะได้รับสัญญาณที่เวลาแตกต่างกันเราจึงเอาหลักการนี้ไปเปรียบเทียบกับสัญญาณนาฬิกาที่ชุดนับสัญญาณต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

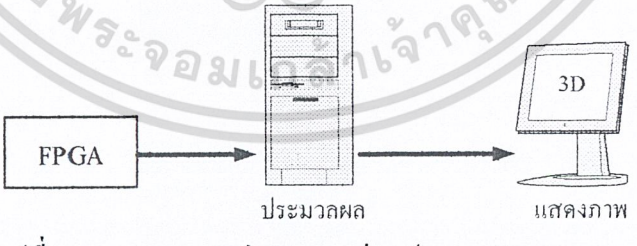
2.1.2 บอร์ดFPGA



รูปที่ 2.2 ภาพแสดงการทำงานของส่วน FPGA

บอร์ดนี้เป็นบอร์ดทดลองเกี่ยวกับการSimmulate พวกลอจิกที่เกี่ยวข้องกับอุปกรณ์ทางอิเล็กทรอนิคส์ นำได้นำบอร์ดตัวนี้เข้ามาเพื่อทำหน้าที่เป็นวงจร Counter โดยนำผลต่างของเวลาที่ภาครับสัญญาณอัลตร้าโซนิคมาเปรียบเทียบกับสัญญาณนาฬิกาบนบอร์ดFPGA จะได้ผลข้อมูลที่เป็นจำนวนพัลส์ของตำแหน่งทั้ง4ตัวของตัวรับเพื่อส่งไปประมวลผลผ่านทางพอร์ทขนานของตัวบอร์ดต่อไป

2.1.3 ภาคคำนวณและประมวลผลโดยเครื่องคอมพิวเตอร์



รูปที่ 2.3 ภาพแสดงการทำงานของส่วนประมวลและแสดงผล

ภาคนี้ทำงาน โดยการนำค่าCounterที่นับได้ส่งเข้าทางพอร์ตขนานของบอร์ดFPGA เพื่อเข้ามาคำนวณพิกัดของตัวส่งสัญญาณ ส่วนนี้ประกอบด้วย โปรแกรมVisual Basic เป็นหน้าต่างแสดงการทำงานของโปรแกรมนี้ และนำค่าพิกัดที่คำนวณได้ไปขึ้นเป็นภาคสามมิติต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
ทฤษฎีที่ใช้ในการออกแบบ

3.1 ทฤษฎีเกี่ยวกับคลื่นอัลตร้าโซนิค

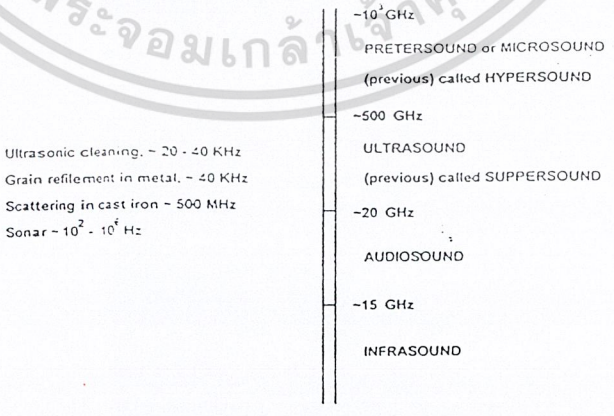
เครื่องชี้ตำแหน่ง 3 มิตินี้ อาศัยหลักการวัดระยะทางด้วยคลื่นอัลตร้าโซนิค แล้วนำค่าระยะทางที่วัดได้มาคำนวณหาพิกัดตำแหน่ง โดยทฤษฎีของคลื่นอัลตร้าโซนิคจะกล่าวโดยสังเขปดังนี้

3.1.1 คุณสมบัติและธรรมชาติของคลื่นอัลตร้าโซนิค

หลักการของคลื่นเสียง คือ ฟิลด์ของพลังงานจะถูกส่งออกมาจากตัวส่งไปซึ่งถ้าเรานำตัวรับชนิดเดียวกันกับตัวส่งซึ่งมีความถี่ตรงกันแล้วก็จะสามารถรับสัญญาณได้ ซึ่งคลื่นอัลตร้าโซนิคจะเป็นคลื่นที่มีทิศทางและจะพุ่งออกจากตัวส่งซึ่งจะมีขอบเขตประมาณ 30 องศาจากตัวส่ง เวลาที่ใช้ในการเดินทางในอากาศที่จะคำนวณระยะทางได้เพราะฉะนั้นการศึกษาถึงธรรมชาติของคลื่น จะสามารถทำให้เข้าใจในการนำไปประยุกต์ใช้ได้อย่างถูกต้อง

3.1.2 คลื่นอัลตร้าโซนิค

คลื่นอัลตร้าโซนิค คือ คลื่นเสียงที่มีความถี่สูงเกินที่หูมนุษย์จะได้ยิน โดยทั่วไปแล้วหูมนุษย์จะได้ยินคลื่นความถี่ประมาณ 20 Hz ถึง 20 kHz ดังนั้นแล้วคลื่นอัลตร้าโซนิคจึงหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20kHz จนถึง 104 GHz โดยคลื่นที่มีความถี่สูงกว่า 104 GHz และแอมปลิจูดสูงๆ จะเรียกว่า “ไฮเปอร์ซาวนด์”



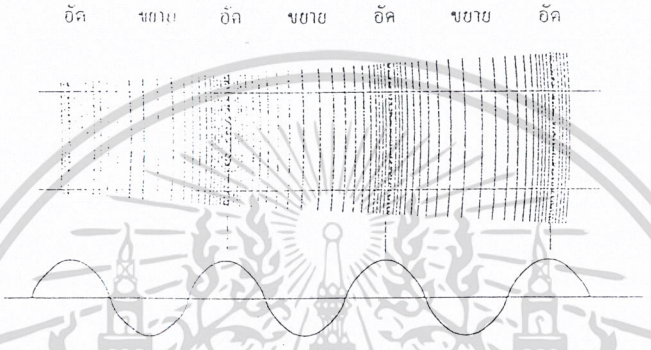
รูปที่ 3.1 แสดงถึงช่วงความถี่ต่างๆที่ถูกนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ชนิดของคลื่นอัลตร้าโซนิค

คลื่นอัลตร้าโซนิคที่เดินทางผ่านตัวกลาง ต่างๆ มีหลายชนิดด้วยกันซึ่งแต่ละชนิดแตกต่างกันตามการเคลื่อนของอนุภาคในตัวกลางนั้น

3.1.3.1 คลื่นตามยาว(Longtudinal Wave) คือคลื่นอนุภาคตัวกลางที่มีการเคลื่อนที่ไปในทิศทางการเคลื่อนที่ของคลื่น



รูปที่ 3.2 แสดงลักษณะการเกิดคลื่นตามยาว

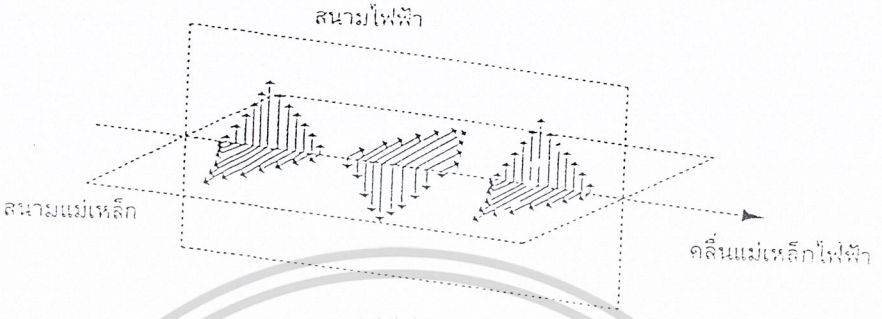
จากรูป 3.2 จะเห็นว่ามีส่วนที่เคลื่อนที่อัด (Compression) ซึ่งก็คือคลื่นช่วงของอนุภาคตัวกลางมีความชันสูงและคลื่นขยาย(Reaction) คือที่อนุภาคของตัวกลางมีความดันต่ำและเมื่อนำค่าของความดันที่เปลี่ยนแปลงตามระยะทางมาเขียนกราฟจะได้รูปซายน์ โดยยอดคลื่นจะตรงกับส่วนอัดและท้องคลื่นจะตรงกับส่วนขยาย ระยะทางระหว่างส่วนอัดถึงส่วนอัด หรือส่วนขยายถึงส่วนขยายคือ 1 ความยาวคลื่น และมีคาบเวลาเป็น (T) ซึ่งเท่ากับ 1/f โดยจุดที่เป็นแกนนั้นยาว 1 บรรยากาศ

3.1.3.2 คลื่นตามขวาง(Tranverse Wave) คือที่ทุกๆจุดบนคลื่น มีการเคลื่อนที่ไปในทิศทางที่ตั้งฉากกับทิศทางการเคลื่อนที่ คลื่นชนิดนี้จะเดินทางผ่านตัวกลางที่เป็นของแข็ง คลื่นชนิดนี้ไม่สามารถเดินทางผ่านตัวกลางที่เป็นของแข็งและก๊าซได้

คลื่นตามขวางที่ลักษณะเหมือนกับการเกิดขั้วลบและขั้วบวก ซึ่งเป็นเหตุผลที่ว่าการเปลี่ยนตำแหน่งของอนุภาคเป็นไปในทิศทางเดียว เช่น ในระนาบที่ตั้งฉากกับทิศทางของคลื่น ที่เคลื่อนที่ไปจากต้นกำเนิดของคลื่นตามขวาง เป็นพื้นที่หน้าเรียบของระนาบที่เกิดจากการเปลี่ยนแปลงของอนุภาคอันเนื่องมาจากการแกว่ง และความหนาของตัวกลาง จะไม่เปลี่ยนแปลงโดยการเคลื่อนที่ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นแบบตามขวาง ความเร็วของคลื่นชนิดนี้จะน้อยกว่าความเร็วของคลื่นชนิดตามยาว ในขณะที่เดินทางผ่านตัวกลางชนิดเดียวกัน ดังนั้นที่ความถี่เดียวกันความยาวของคลื่นตามขวางจะน้อยกว่าคลื่นตามยาวเสมอ



รูปที่ 3.3 แสดงลักษณะการเกิดคลื่นตามขวาง

3.1.3.3 คลื่นผิวหน้า (Surface Wave or Rayleigh) คือ คลื่นชนิดหนึ่งซึ่งคล้ายกับคลื่นตามขวางจะต่างกันตรงที่ว่าการเปลี่ยนตำแหน่งของอนุภาค ไม่เป็นไปในทิศทางที่ตั้งฉากกับทิศทางที่เคลื่อนที่เพียงอย่างเดียว แต่มีการเปลี่ยนแปลงในทิศทางเดียวกับทิศทางการเคลื่อนที่ด้วย จึงทำให้คลื่นเคลื่อนที่ไปตามระนาบในแนวนอน ด้วยเหตุนี้คลื่นจึงเดินทางผ่านไปเฉพาะบนผิวของตัวกลางเท่านั้น

3.1.4 คุณสมบัติที่สำคัญของคลื่นอัลตร้าโซนิค

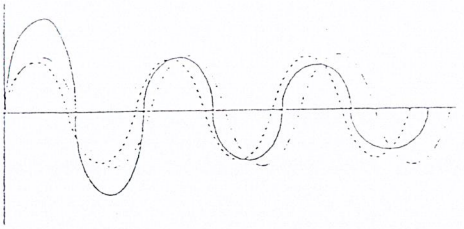
คุณสมบัติโดยทั่วไปของคลื่นเสียงจะแสดงได้ 4 แบบคือ

3.1.4.1 การแทรกสอดของเสียง(Interference)

การแทรกสอดของเสียงเกิดจากการรวมกันของคลื่น 2 คลื่นขึ้นไป ขณะเมื่อพบกันในตัวกลางเดียวกัน ซึ่งทำให้เกิดผลได้หลายหลักษณะคือ

ก.) *การบีตส์*(Beats) ของคลื่นเสียงเป็นปรากฏการณ์ที่เกิดจากการรวมคลื่นที่มีความถี่ต่างกันหรือต่างเฟสกัน เคลื่อนที่ไปในตัวกลางเดียวกันแล้วรวมเป็นคลื่นใหม่ ซึ่งทำให้แอมปลิจูดเปลี่ยนไปดังแสดงในรูปที่3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

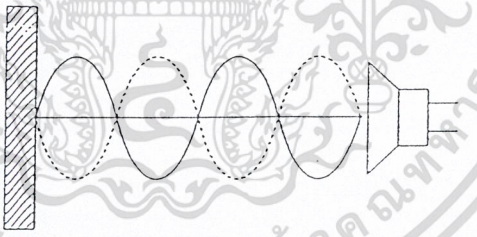


รูปที่ 3.4 แสดงการบีสต์ของคลื่นเสียง 2 คลื่น

จากรูปที่ 3.4 เป็นการแสดงบีตส์ของคลื่น 2 คลื่น ซึ่งคลื่นที่มีแอมปลิจูดเปลี่ยนไป ซึ่งประโยชน์ของการบีตส์ของคลื่นนั้น จะนำไปใช้ในการเปรียบเทียบความถี่ของคลื่นให้แสดงผลออกมาในลักษณะของแอมปลิจูดที่แตกต่างกันนี้

ข.) *การเกิดคลื่นนิ่ง*(Standing Wave) เกิดจากการแทรกสอดของคลื่นชุดที่มีแอมปลิจูดเท่ากันและความถี่เท่ากัน แต่มีทิศทางตรงกันข้ามหรือเฟสตรงกันข้าม ซึ่งจะทำให้เกิดคลื่นนิ่งดังรูป เสียงจะมีความเข้มข้นสูงสุดที่ตำแหน่ง ก. และเบาที่สุดที่ตำแหน่ง ข. ซึ่งระยะระหว่างขั้วทั้ง2 เท่ากับ

$$d = n\lambda/2; (n=1,2,3\ldots)$$

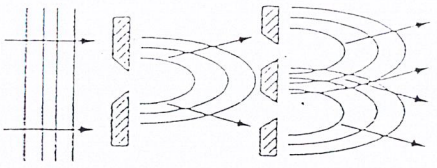


รูปที่ 3.5 แสดงลักษณะการเกิดคลื่นนิ่ง

3.1.4.2 การเลี้ยวเบนของคลื่นเสียง (Diffraction)

คือ คลื่นเสียงจะเลี้ยวเบนอ้อมสิ่งกีดขวางที่มีลักษณะเป็นมุมหรือช่องแคบ ซึ่งปรากฏการณ์เช่นนี้พบในชีวิตประจำวันอยู่ตลอดเวลาเช่น ในกรณีที่เราได้ยินแตรรถที่อยู่คนละมุมถนนของตึกหรือการได้ยินเสียงลอดผ่านช่องเล็กๆ จากอีกห้องหนึ่ง ดังแสดงในรูป 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบๆจะเสมือนเป็นแหล่งกำเนิดตัวใหม่

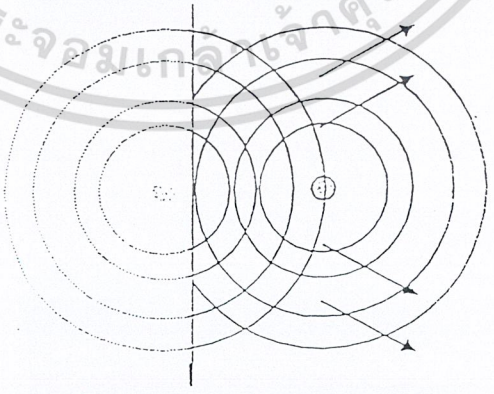
3.1.4.3 การหักเหของคลื่น (Refraction)

คลื่นเสียงเมื่อเดินทางผ่านตัวกลางที่มีความหนาแน่นต่างกันจะเกิดการหักเหของคลื่น ซึ่งทำให้ความเร็วของคลื่นเสียงเปลี่ยนไป โดยที่ความถี่ยังคงที่อยู่

รูปที่ 3.7 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน

3.1.4.4 การสะท้อนของคลื่นเสียง (Reflection)

คลื่นเสียงสามารถสะท้อนได้เมื่อตกกระทบตัวกลางโดยที่มุมสะท้อนและจะทำให้เกิดเสียงก้อง ซึ่งเสียงก้องที่สะท้อนกลับมาในเวลาที่มากกว่า 50 msec จะทำให้เราได้ยินเสียงนี้เป็นครั้งที่2



รูปที่ 3.8 แสดงเมื่อคลื่นวงกลมกระทบกับฉากที่เป็นเส้นตรง
คลื่นที่สะท้อนกลับมาจะเป็นคลื่นวงกลมเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.5 การเกิดคลื่นอัลตร้าโซนิค

อัลตร้าโซนิคเป็นคลื่นที่เกิดจากการเปลี่ยนแปลงพลังงานรูปอื่นให้เป็นพลังงานกล โดยการสั่นไปมา หรือ เกิดจากการเปลี่ยนแปลงพลังงานไฟฟ้ากับพลังงานกลให้เกิดคลื่นย่านอัลตร้าโซนิคกระจายออกไปในอากาศ ดังนั้นจึงถือได้ว่าคลื่นที่เกิดขึ้นเป็นคลื่นกล(Mechanical Wave) อัลตร้าโซนิคสามารถสร้างได้โดยตัวทรานสดิวเซอร์ ซึ่งเป็นอุปกรณ์ที่ใช้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลหลักการสร้างมีหลายวิธีดังนี้

ก. *แบบเปียโซอิเลคทริค*(Piezo-electric Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล โดยมีความถี่เรโซแนนท์คงที่อยู่ค่าหนึ่ง

ข. *แบบแมกนิโตสตริคทีฟ* (Magnetrostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้น

ค. *แบบอิเลคโทรสตริคทีฟ* (Electrostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล

*** สำหรับเปียโซอิเลคทริคเป็นแบบที่นิยมใช้เพราะมีราคาถูกและหาซื้อง่าย

3.1.4.6 ความถี่และความยาวคลื่น (Frequency and Wave Length)

ความถี่ คือ จำนวนของการออสซิลเลทที่สมบูรณ์จากแหล่งกำเนิดคลื่นภายในหนึ่งวินาที คลื่นที่ถูกส่งจากแหล่งกำเนิดจะเดินทางด้วยความถี่เดียวกัน

ความยาวคลื่น คือระยะทางที่คลื่นเดินทางระหว่างการสั่นที่สมบูรณ์ หรือการเดินทางครบ 1 รอบ(Cycle) สามารถกล่าวได้ว่า ความยาวคลื่นเป็นระยะทางระหว่างการอัดอย่างต่อเนื่อง (Success Compression) หรือการเบาบางของอากาศ (Rare Fraction) การอัดคือการที่บริเวณนั้นมีความหนาแน่นของโมเลกุล และแรงดันมากกว่าบริเวณรอบๆ ส่วนการเบาบางเป็นบริเวณเฉพาะที่การเกิดการลดความหน้าแน่นของโมเลกุลและแรงดันสัมพันธ์กับแรงดันของอากาศปกติ

ความถี่และความยาวคลื่นมีความสัมพันธ์กันตามสมการข้างล่าง

$$C=f\lambda$$

C : ความเร็วของการเดินทาง

f : ความเร็ว

λ : ความยาวคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.7 ความเร็วของคลื่นอัลตร้าโซนิค

ความเร็วที่ยอมรับของคลื่นในอากาศที่อุณหภูมิปกติความสัมพันธ์ที่ใช้จะเป็นดังสมการ

$$V = 331.45 + 0.607T \text{ (m/sec)}$$

V : ความเร็วของคลื่นในตัวกลางอากาศ

T : อุณหภูมิของคลื่นในอากาศ(องศาเซลเซียส)

3.1.4.8 ปริมาณพลังงานของคลื่นอัลตร้าโซนิค

ปริมาณพลังงานของคลื่นอัลตร้าโซนิคจะถูกวัดในรูปความเข้มของคลื่นอัลตร้าโซนิคจะมีหน่วยเป็น วัตต์ต่อตารางเซนติเมตร เป็นการไหลของพลังงานผ่านพื้นที่ 1 ตารางเซนติเมตร ซึ่งตั้งฉากกับทิศทางการเดินทางของคลื่นใน 1 วินาที

3.1.4.9 การลดทอนของคลื่นอัลตร้าโซนิค

เมื่อคลื่นเดินทางผ่านตัวกลาง ลำคลื่น(Beam) ของคลื่นอัลตร้าโซนิคจะสูญเสียความเข้มซึ่งเกิดจากการลู่ออกของลำคลื่นอัลตร้าโซนิค หรือเกิดจากการกระจายพลังงานของคลื่นออกจากลำคลื่นเนื่องจากความไม่ต่อเนื่องของตัวกลาง และอาจเกิดจากการดูดซับพลังงานส่วนหนึ่งของคลื่นโดยตัวกลางที่คลื่นเคลื่อนที่ผ่าน พลังงานที่ดูดซับนี้จะเปลี่ยนเป็นพลังงานความร้อน การดูดซับพลังงานความร้อนนี้ขึ้นอยู่กับลักษณะของวัตถุ ความยืดหยุ่นและความหนาแน่น รวมทั้งความถี่ของคลื่นที่ใช้ยิ่งความถี่สูงพลังงานยิ่งถูกดูดซับมาก

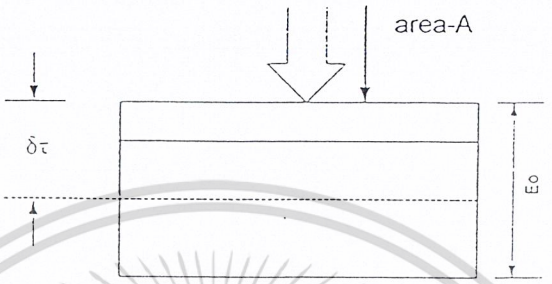
3.2 อัลตร้าโซนิคทรานสดิวเซอร์

3.2.1 ปรากฏการณ์เปียโซอิเลคทริค

เปียโซอิเลคทริคเป็นปรากฏการณ์ธรรมชาติอย่างหนึ่ง ซึ่งทำให้พลังงานสามารถเปลี่ยนแปลงจากรูปหนึ่งไปเป็นอีกรูปหนึ่งได้ กล่าวคือถ้าป้อนแรงกลให้แก่ Solek Crystalline Dielectric ดังในรูปที่ 3.9 ก็จะทำให้เกิดความเค้น (Stress) ภายในคริสตอลและทำให้ผลึกของคริสตอลผิดรูปไปเช่นพวกควอทซ์(Quartz) ผลก็คือประจุเปลี่ยนไป การผิดรูปร่างของแลททิสเป็นผลให้ความสัมพันธ์ระหว่างการแทนที่ (Displacement) ของประจุบวกและลบในแลททิสเปลี่ยนไป การแทนที่ของประจุภายในจะเท่ากับประจุภายนอกของขั้วที่ตรงข้ามของคริสตอล เรียกว่าผลของเปียโซอิเลคทริค

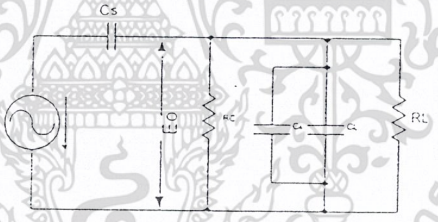
เอกสารนี้มีการสงวนลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นได้รับอนุญาตจากเจ้าของลิขสิทธิ์ ทั้งนี้ยกเว้นการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดประจุทำได้โดยการต่ออิเลคโทรด (electrod) เข้าที่ผิวนอก แล้ววัดความต่างศักย์ระหว่างขั้วทั้งสอง ขาด(Magnetude) และการมีขั้ว (Polarity) ของประจุบนผิวที่ถูกเหนี่ยวนำ(induce) เป็นสัดส่วนโดยตรงกับขนาดและทิศทางของแรงที่มากระทำ



รูปที่ 3.9 ผลของเปียโซอิเลคทริค

วงจรเสมือนของเปียโซอิเลคทริคทรานสดิวเซอร์



รูปที่ 3.10 วงจรเสมือนของเปียโซอิเลคทริค

C_c: คาปาซิสแตนท์ของชิ้นส่วนเปียโซอิเลคทริค

R_c : Leagage Resistance ของเปียโซอิเลคทริค ปกติจะมีค่าสูงประมาณ 10^8 - 10^{10}

ความต้านทานระหว่างเทอร์มินอล โดยทั่วไปจะเป็นความต้านทานทางโหลด R_L (ในช่วง $10^6 - 10^7$)

C_L : คาปาซิแตนท์ ของ Subsequent Stage(Load) ร่วมกับ Connection Cable

ที่ความถี่ปานกลางและความถี่สูงโวลท์เตจ E_0 ที่คร่อมโหลดหาได้โดย C_c และ C_L ค่าเหล่านี้มาได้โดยการใช้การแบ่งโวลท์เตจ โดยโวลตท์เตจ E_0 ได้จากสมการ

$$E_0 = E(C_c/(C_c+C_L))$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขึ้นกับความถี่ ถ้าเอาท์พุทโวลท์เตจมีค่ามากสามารถทำให้ลดลงได้โดยการเพิ่มค่า C_L เช่นการนำค่า C_c ต่อขนานเข้าไปที่ความถี่ต่ำโวลท์เตจ E_0 คร่อมโหลดหาได้โดยค่ารีแอคแตนท์ของ C_c และอิมพีแดนซ์ของ C_L และ R_L ที่ต่อขนานกันโวลท์เตจ E_0 ขึ้นกับความถี่และจะลดลงถ้าความถี่ลดลง การตอบสนองต่อความถี่สามารถปรับปรุงได้โดยการใช้ทรานสดิวเซอร์ที่มีค่าคาปาซิแตนท์สูงหรือโดยการเพิ่มความต้านทานโหลด RL

3.2.2 วัสดุเปียโซอิเลคทริค

วัสดุเปียโซอิเลคทริคที่ใช้กันอย่างกว้างขวาง เช่น Quartz , Tourmaline sulphate, Barium Tilanate และ Ziconnate (TZI) โดยทั่วไปแล้วพวกควอทซ์ และคริสตอลที่เป็นเปียโซอิเลคทริคธรรมชาติมักจะมีขั้วของมันเองตามธรรมชาติ แต่พวกวัสดุเปียโซอิเลคทริคที่สังเคราะห์ขึ้นมาเช่น แบเรียม ไททาเนท เซรามิค จะต้องนำมาทำการอบคริสตอลภายในแรงดัน และวางวัสดุที่ได้นี้ในสนามไฟฟ้าที่มีแรงดันของไฟฟ้ากระแสตรงมาก หลังจากชิ้นส่วนนี้ถูกนำในสนามไฟฟ้าแล้ว คริสตอลนี้จะมีขั้วตามแนวของทิศทางของสนาม และประพฤติตัวตามคุณสมบัติของเปียโซอิเลคทริคสำหรับชิ้นส่วนที่ทำจากวัสดุสังเคราะห์นี้ ไม่มีข้อจำกัดทางขนาดโดยโครงสร้างของคริสตอล และยังสามารถทำให้มีรูปร่างและขนาดต่างๆ และทิศทางของขั้วก็จะถูกสร้างขึ้นระหว่างขั้นตอนการผลิต

3.2.3 ชนิดของเปียโซอิเลคทริคทรานสดิวเซอร์

เปียโซอิเลคทริคทรานสดิวเซอร์สามารถแบ่งได้เป็น 2 ชนิดคือ

1. *แบบ Generation-Action Tranducer* ใช้เป็นตัวรับโดยแรงดันไฟฟ้าที่เกิดขึ้นจะหาได้จากแรงดันและความถี่ที่มากระทำต่อวัสดุเปียโซอิเลคทริค
2. *แบบ Motor-Action Transducer* ใช้เป็นตัวส่งโดยการเปลี่ยนแปลงของรูปร่างที่ทำให้เกิดคลื่นอัลตร้าโซนิค โดยจะขึ้นอยู่กับขนาดความสูงและความถี่ของแรงดันไฟฟ้าที่ให้ในทั้งสองกรณีค่าแรงดันไฟฟ้าที่เกิดขึ้นกับขนาดของวัสดุ

3.2.4 การทำงานของทรานสดิวเซอร์ตัวส่งตัวรับ

เมื่อเซรามิคได้รับสัญญาณแรงดันมาตกคร่อมจะทำให้ชิ้นสารเซรามิคโค้งงอทำให้เกิดการอัดอากาศโดยรอบเกิดเป็นคลื่นขึ้นมา ดังนั้นถ้าป้อนสัญญาณเป็นห้วงๆ (Electrically Pulse) จากการออสซิสเลทก็จะให้ชิ้นสาร โก่งงอมากน้อยหรือทิศทางใดตามขนาดและทิศทางการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดของสัญญาณไฟฟ้าจากการออสซิสเลทนั้นออกไป โดยทั่วไปกำลังเอาท์พุทที่ออกมาจะตก ประมาณ 10% ของกำลังไฟฟ้าที่ป้อนให้แต่เอาท์พุทจะสูงที่ค่านี้ โดยประมาณก็ต่อเมื่อความถี่ของสัญญาณออสซิสเลเตอร์ที่ป้อนเข้าชิ้นสารเซรามิค ตรงกับความถี่เรโซแนนท์เป็นความถี่ทางกลตามธรรมชาติของชิ้นสารเซรามิคนั้นๆ ส่วนที่ความถี่อื่นๆกำลังเอาท์พุทจะลดลงกว่านี้ ส่วนการทำงานของทรานสดิวเซอร์ตัวรับมักมีการทำงานตรงข้ามกับตัวส่ง คือเมื่อมีคลื่นเสียงที่มีความถี่ตรงกับความถี่เรโซแนนท์ของชิ้นสารเซรามิคเข้ามา จะทำให้ชิ้นสารโก่งตัวไปมาและเกิดสัญญาณแรงดันไฟฟ้าที่มีขนาดเล็กขึ้นคร่อมขั้วทั้งสองของตัวมัน

คุณสมบัติทั่วไปของอัลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคทริคก็คือ มีความต้านทานไฟฟ้าตรงสูงมากอาจถึง 100 เมกกะโอห์ม เรียกว่าถ้าเอามัลติมิเตอร์ธรรมดา ตั้งสเกลวัดค่าความต้านทานสูงๆเข็มจะ ไม่กระดิกเลยแต่ขณะที่มันทำงานความต้านทานไฟฟ้าสลับจะลดลง

3.2.5 ข้อควรระวังในการใช้งานตัวรับ-ตัวส่งทรานสดิวเซอร์

1. ไม่ควรให้ตัวต้านทานทรานสดิวเซอร์ได้รับการกระแทกหรือตกจากที่สูง เพื่อป้องกันโครงสร้างภายในไม่ให้เสียหาย

2. ทรานสดิวเซอร์ที่มีขายกันโดยทั่วไปจะทนแรงดันตกคร่อมตัวมันได้ไม่เกิน 10 V_{rms} ดังนั้นขนาดของสัญญาณที่จะป้อนให้กับตัวทรรานสดิวเซอร์ก็ควรอยู่ภายในขีดจำกัดนี้

3. ความถี่เรโซแนนท์ (คือความถี่ที่ตัวมันสามารถทำงานอย่างมีประสิทธิภาพที่สุด) ของทรานสดิวเซอร์ 40 KHz (Bandwidth) ที่มีขายโดยทั่วไปจะผิดพลาดไม่เกิน $\pm$1 KHz และมีแถบความถี่ประมาณ 4.5 KHz(Bandwidth) สำหรับตัวส่ง และมีแถบความถี่ประมาณ 5.0 KHz (Bandwidth) สำหรับตัวรับ จะเห็นได้ว่าแถบความถี่ของตัวรับจะกว้างกว่าตัวส่งเล็กน้อย เพื่อให้แน่ใจว่าตัวรับจะสามารถรับความถี่ทั้งหมดที่ออกจากตัวส่งได้

4. อุณหภูมิที่ใช้งานของตัวทรานสดิวเซอร์ควรอยู่ภายในช่วง –20 ถึง 60 องศาเซลเซียส

5. ตัวส่งและตัวรับจะมีทิศทางคล้ายกันมาก กล่าวคือที่ตำแหน่งเบนจากแนวแกนของตัวส่งไปประมาณ 30 องศา ความแรงของคลื่นเสียงที่ถูกส่งออกไปจะลดลงจากแนวแกนประมาณ 10 dB ทำนองเดียวกัน ถ้าคลื่นเสียงพุ่งเข้ามาในแนวที่เบี่ยงเบนไป จากแกนของตัวรับประมาณ 30 องศาความไวหรือขนาดของแรงดันที่ออกมาจะลด จึงควรจะทำให้ตัวรับและตัวส่งอยู่ในแนวที่พุ่งตรงมากที่สุดอย่างไรก็ตามในกรณีที่อยู่ในห้องจะเกิดการเบียงเบนกันได้มาก เพราะคลื่นเสียงอัลตร้าโซนิค จะสะท้อนกับกำแพงและวัสดุที่อยู่ภายในห้อง ทำให้คลื่นเสียงเข้าไปหาตัวรับได้หลายทาง

อัลตร้าโซนิคนี้สามารถทำให้เป็นลำแคบได้โดยที่ใช้เลนส์ที่เรียกว่า Planacone acelenes วางข้างหน้าของทรานสดิวเซอร์ แต่การทำให้เป็นลำแคบ(Beam) นี้ จะทำให้ near field สั้นลงและ far

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

field กระจายกว้างขึ้น การทำให้เป็นลำคลื่นเหมาะสำหรับการใช้ทรานสดิวเซอร์ที่สร้างความถี่สูงที่เหมาะสำหรับที่ใช้งานในงานแพทย์(Ultrasound) ที่มีความถี่ตั้งแต่ 2-5 MHz เพื่อใช้ตรวจในเนื้อเยื่อมีการส่งเป็นลำแคบ

6. ในกรณีที่ใช้งานตัวรับจะต้องมีความต้านทานต่อขนานกับตัวรับที่ทำหน้าที่เป็นโหลด ตามปกติแล้วตัวต้านทานนี้ควรมีค่าอยู่ระหว่าง 10 KΩ ถึง 100KΩ จากการทดลองพบว่าถ้าเปลี่ยนจาก 100KΩ มาเป็น 10KΩ ความไวจะลดลงประมาณ 10-12 dB แต่ความถี่จะกว้างขึ้น ถ้าความต้านทานต่ำลงไปอีก ความถี่เรโซแนนท์ (ความถี่กลาง) จะลดลงไปจากที่ระบุไว้ ถ้าการใช้งานมีสัญญาณรบกวนมากควรใช้โหลดที่มีค่าความต้านทานสูงสักหน่อย เพื่อให้ตัวส่งมีความไวสูงและมีความถี่แคบ

7. ตามปกติแล้วสามารถนำตัวส่งและตัวรับมาใช้งานแทนกันได้ ในการใช้งานส่วนใหญ่ตัวส่งและตัวรับรุ่นใด ก็สามารถใช้แทนกันได้ในงานส่วนใหญ่ ขอเพียงแต่ให้มีความถี่เรโซแนนท์เดียวกันเท่านั้นเอง อย่างไรก็ตามในบางกรณีอาจต้องเปลี่ยนแปลงค่าความต้านทานสมมูลย์ทางไฟฟ้าทางด้านไฟฟ้าสลับเพื่อให้เกิดลักษณะผลตอบสนองทางความถี่สอดคล้องกับของเดิม

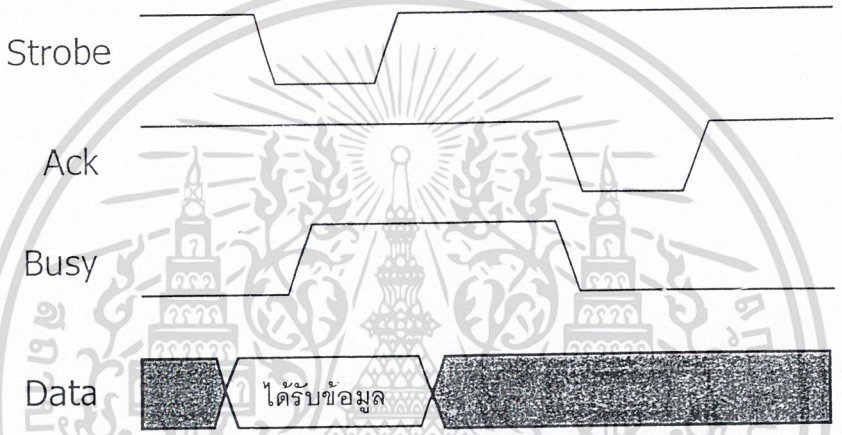
8. ประโยชน์การใช้งานคลื่นอัลตร้าโซนิค คลื่นอัลตร้าโซนิคเป็นคลื่นที่มีทิศทาง ทำให้สามารถเล็งคลื่นไปตามเป้าหมายที่ต้องการ ได้เจาะจง ยิ่งคลื่นที่มีความถี่สูงขึ้นความยาวคลื่นจะยิ่งสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิด(ที่ทำให้เสียงออกมา) ของตัวที่ทำให้เกิดเสียงความถี่นี้ เช่น คลื่นความยาว 300Hz ในอากาศ จะมีความยาวคลื่นถึงประมาณ 1 เมตรกว่าๆ ซึ่งจะยาวกว่าช่องเปิดที่ให้เสียงออกมาจากตัวกำเนิดเสียงโดยทั่วไปมากมาย คลื่นจะหักเหที่ขอบด้านนอกของตัวกำเนิดเสียงที่ทำให้เกิดการกระจายทิศทางของคลื่น แต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอัลตร้าโซนิค อย่างเช่น 40 KHz จะมีความยาวคลื่นในอากาศเพียง 8 มิลลิเมตร เท่านั้น ซึ่งเล็กกว่าตัวที่ทำให้เกิดคลื่นความถี่นี้มากๆ คลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบ จึงพุ่งออกมาเป็นลักษณะลำแคบๆหรือที่เราเรียกว่า “มีทิศทาง” นั้นเองการมีทิศทางของคลื่นเสียงอัลตร้าโซนิคทำให้เราสามารถนำเอาไปใช้งานได้หลายอย่าง เช่น คลื่นวัดความหนาแน่นของวัตถุ โดยสังเกตระยะที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกทำแผนที่ใต้ท้องทะเล ส่วนการใช้งานทางด้านการแพทย์ อาจใช้ความถี่ในช่วง 1MHz ถึง 10MHz ขณะที่ความถี่เป็น GHz ก็มีใช้กันหลายๆการใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่อากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ความรู้เบื้องต้นของพอร์ตขนาน

พอร์ตขนาน(Parallel port) เนื่องด้วยการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนาน ทำให้พอร์ตขนานมีอัตราการถ่ายทอดข้อมูลสูงกว่า การถ่ายทอดข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้นพอร์ตขนานจึงสามารถรองรับการถ่ายทอดข้อมูล 8 บิตได้โดยไม่ต้องต่อส่วนเพิ่มเติมใดๆ

3.3.1 ลักษณะทางกายภาพของพอร์ตขนาน



รูป 3.11 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

เพื่อให้เข้าใจหลักการทำงานของพอร์ตขนาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานภายในอย่างไร ในรูปที่ 3.11 แสดงไดอะแกรมเวลาของติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริงๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Strobe ออกไปด้วยเพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลใหม่มาที่ขา Data แล้วจากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy หรือเพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว

สัญญาณข้อมูลขนาด 8 บิต สัญญาณ Strobe และสัญญาณ ACK (Acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้ว ส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่นๆร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึง3อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์ , พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการณ์ที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม(เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลมากกว่านี้ได้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือสัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่นกระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซตเครื่องพิมพ์คอมพิวเตอร์ คอมพิวเตอร์จะต้องส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังตารางที่ 3.1

จากตารางที่3.1 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซตเครื่องพิมพ์	คอมพิวเตอร์

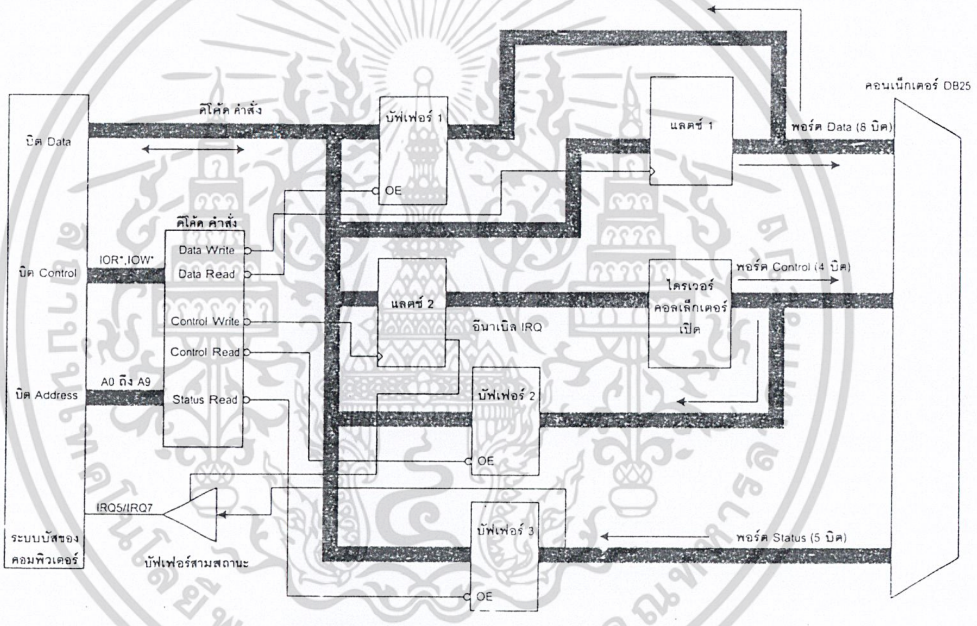
ตารางที่3.1 สัญญาณสำคัญๆของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงาน ดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่างๆจากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

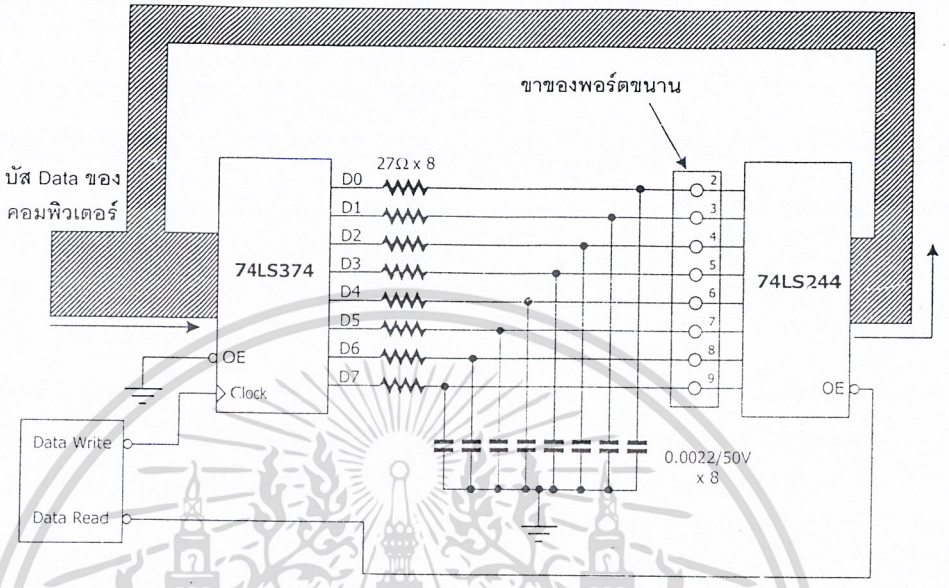


ตารางที่ 3.2 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50399

3.3.2 พอร์ตดาต้า (Data Port)



รูปที่ 3.12 วงจรภายในของพอร์ต Data

จากรูป3.12 แสดงให้เห็นว่าพอร์ต Data ประกอบไปด้วยบัฟเฟอร์ 1 ตัว และไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตช์ 1 ทั้ง 8 บิตเอาต์พุตของไอซีแลตช์1 คือ D0-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัฟเฟอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานค่า 27Ω และตัวเก็บประจุ 0.0022μF เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก “0” เป็นลอจิก “1” หรือจากลอจิก “1” เป็นลอจิก “0” เป็นไปอย่างช้าๆ เนื่องจากการเปลี่ยนแรงดันที่รวดเร็วทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่นๆได้ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในรูป3.12 ทำให้เอาต์พุตของพอร์ต Data มีคุณสมบัติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กระแสซิงก์สูงสุด 24 mA
- กระแสซอร์สสูงสุด 2.6 mA
- ระดับแรงดันของลอจิก "1" ต่ำสุดเท่ากับ 2.4 V
- ระดับแรงดันสูงสุดสำหรับลอจิก "0" เท่ากับ 0.5 V

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Dataread ออกมาเพื่ออีนาเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน (Standard Parallel Port : SPP) พอร์ต Data จะต้องใช้เพื่อการส่งค่าออกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทาง (Bidirectional Parallel Port) สามารถอ่านค่าจากพอร์ต Data ได้ด้วย แต่ก่อนที่จะอ่านค่าต้องจำไว้เสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก "1" ทั้งหมดก่อน

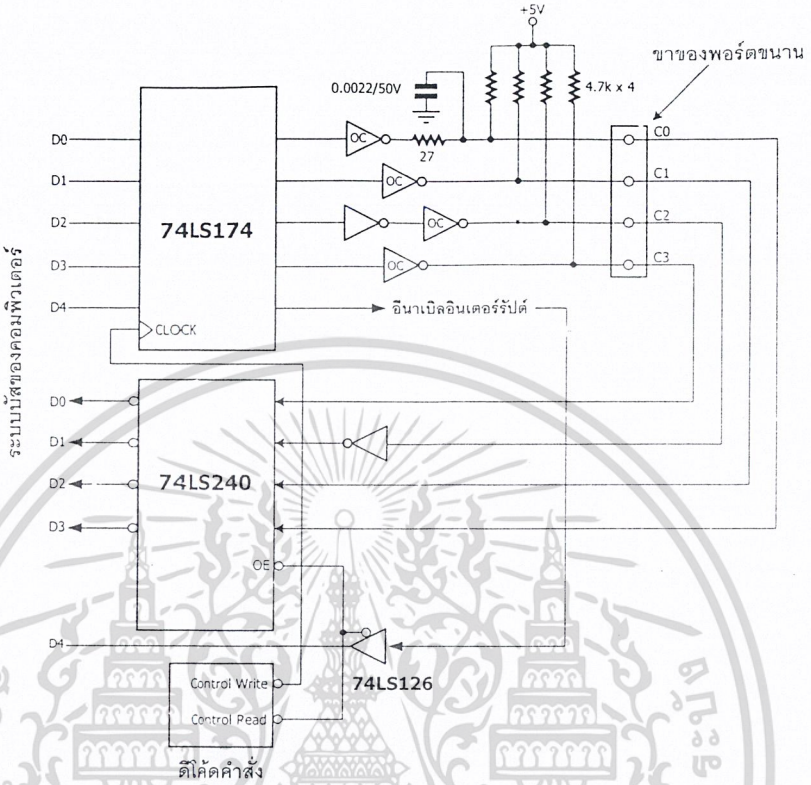
3.3.3 พอร์ต Control

พอร์ต Control ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์จากตารางที่3.2 จะเห็นว่าพอร์ต Control ประกอบไปด้วยบิตเอาท์พุต4บิตที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอินาเบิลอินเตอร์รัปต์ไม่ได้ถูกต่อออกไปรูปที่3.13 แสดงบล็อกไออะแกรมของพอร์ต Control เอาต์พุตของพอร์ต Control มีอินเวอร์เตอร์แบบคอลเล็กเตอร์เปิดต่อร่วมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัปไว้ด้วยตัวต้านทานค่า 4.7 KΩ สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัวทำให้ที่เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

สถานะของพอร์ต Control สามารถอ่านกลับได้โดยการใช้บัฟเฟอร์เบอร์ 74LS240 ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ต Control คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read

เนื่องจากเอาต์พุตของพอร์ต Control เป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่ามีลอจิก"1" เสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

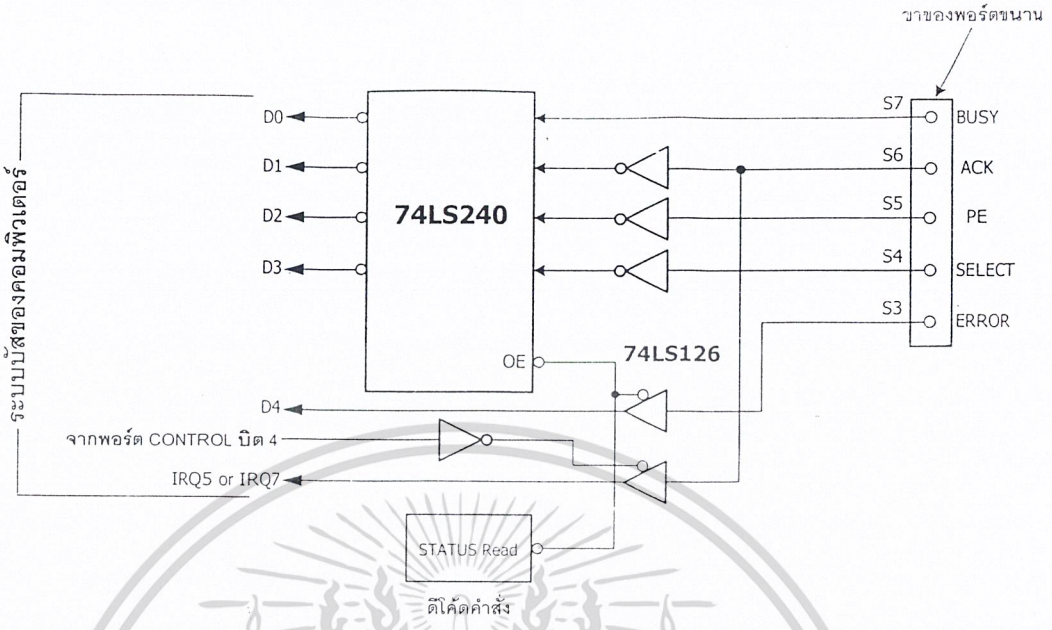


รูปที่ 3.13 วงจรภายในของพอร์ต Control

3.3.4 พอร์ตแสดงสถานะหรือพอร์ต Status

พอร์ต Status เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ รูปที่ 3.14 แสดงรายละเอียดภายในของพอร์ต Status จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกันและจะเรียกชื่อเป็น S3,S4,S5,S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Statusนั่นเอง สำหรับS7 จะมีข้อแตกต่างจากบิตอื่นๆที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลผ่านจากขาอินุตไปยัง 74LS240 ซึ่งเอาท์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกนากนี้ในการใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ต Control บิต4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงวงจรภาพในของพอร์ตแสดงสถานะ

3.3.5 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐานผู้ใช้งานสามารถนำพอร์ตอินพุต 5 บิต(พอร์ต Status) พอร์ตเอาต์พุต 4 บิต(พอร์ต Control)และพอร์ตเอาต์พุตอีก 8 บิต(พอร์ต Data)ไปใช้งานได้โดยตรง โดยที่ 4บิตของพอร์ตเอาต์พุตหรือพอร์ต Control นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุต ขนาด 4 บิต ได้ด้วยดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

3.3.6 การเขียนโปรแกรมติดต่อกับพอร์ตขนาน

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่นๆ คือ เมื่อต้องการติดต่อจะต้องกำหนดแอดเดรสที่ต้องการติดต่อด้วย ตารางที่3.3 แสดงแอดเดรสของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่งคือ แอดเดรสของรีจิสเตอร์ Data , รีจิสเตอร์ Status และรีจิสเตอร์ Control โดยแอดเดรสนี้จะมีอยู่ทั้งหมด 3 ชุดสำหรับพอร์ตขนาน 3 ชุดคือ LPT1,LPT2 และ LPT3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อพอร์ต	LPT1:		LPT2:		LPT3:	
	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
DATA	888	378H	956	3BCH	632	278H
STATUS	889	379H	957	3BDH	633	279H
CONTROL	890	37AH	958	3BEH	634	27AH

ตารางที่3.3 แสดงแอดเดรสของพอร์ตขนาน

3.3.7 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual BASIC

การเขียนโปรแกรมด้วย Visual BASIC ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ QBASIC แต่Visual BASIC จะไม่มีคำสั่งสำหรับการติดต่อกับพอร์ตโดยตรงคือ คำสั่ง Inp() และคำสั่ง OUT เหมือนกับ QBASIC ดังนั้นเพื่อให้สามารถติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมบางตัวเข้าไป โดยโปรแกรมที่เพิ่มเข้าไปนี้จะอยู่ในรูปของ DLL (Dynamic Linked Library)

ไฟล์ DLL นี้จะมี inpout32.dll โดยจะเก็บไฟล์ที่เก็บไฟล์นั้นจะต้องเก็บไว้ที่ไดเรกตอรี่ SYSTEM ซึ่งอยู่ภายในไดเรกตอรี่ที่เก็บโปรแกรมวินโดวส์ โดยส่วนใหญ่จะมีชื่อเป็น Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 FPGA(Field Programmable Gate Array)

โครงสร้างภายในของ FPGA ประกอบไปด้วยอะเรย์ของลอจิกเกทต่างๆมากมาย ซึ่งในปัจจุบันความจุเกทภายในตัวชิพ FPGA ได้เพิ่มขึ้นจากระดับไม่กี่พันตัวจนระดับล้านตัว ซึ่งสามารถรองรับวงจรดิจิตอลที่มีความสลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการออกแบบพัฒนาและทดสอบก็ทำได้ง่าย ซึ่งในปัจจุบันการออกแบบวงจรโดยใช้ FPGA กำลังเป็นที่นิยมและมีแนวโน้มที่จะนำมาใช้งานมากขึ้นเรื่อยๆ

3.4.1 FPGAแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรมซึ่งมีอยู่ 2 แบบคือ

การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรมโดยการใช้หน่วยความจำ

1. การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ
   1.1 *Fuse* เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน
   1.2 *Anti Fuse* เป็นวิธีการโปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรมแล้ว จะเชื่อมต่อจะเชื่อมถึงกัน
2. การโปรแกรมโดยใช้หน่วยความจำ

2.1 EEPROM Based FPGA

FPGA ที่ใช้การโปรแกรมแบบนี้มันเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกทต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกท แต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิท ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

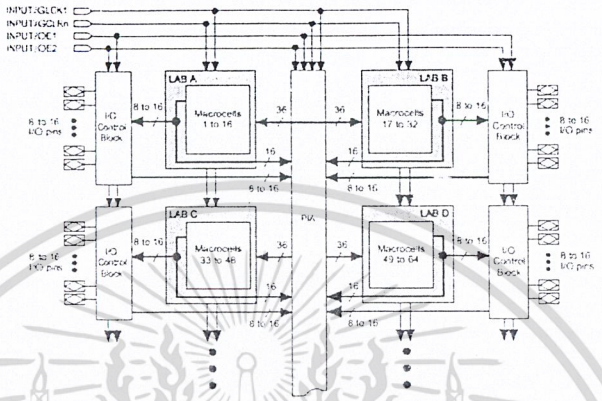
2.2 SRAM Based FPGA

FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถโปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกทในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 – 1,000,000 เกท) ซึ่งข้อดีของ SRAM Based FPGA คือใช้เวลาในการโปรแกรมน้อย (ระดับ ns) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อนส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิพในขณะที่เริ่มต้นใช้งาน

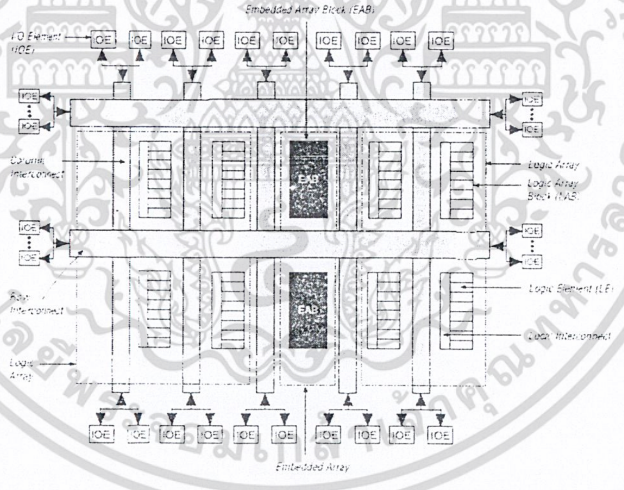
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอะเรย์ของบล็อคลอจิกสามารถทำการโปรแกรมได้ดังรูปที่ 3.22 และ 3.23



รูปที่ 3.14 โครงสร้างภายใน FPGA ตระกูล MAX7000S



รูปที่3.15 โครงสร้างภายใน ของ FPGA ตระกูล FIEX10K

3.4.3 รายละเอียดของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

1. *การสังเคราะห์วงจร (Logic Synthesis)*

ในขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรมของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA ของบริษัท XILINX และบริษัท ALTIRA จะมีซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตามเทคโนโลยี ที่เลือกใช้ ในการสังเคราะห์วงจรนั้น วงจรระดับเกท (Gate Level) จะไม่เหมาะสมกับโครงสร้างที่มีอยู่ในอุปกรณ์ FPGA นั้นๆ จึงทำให้ผลที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดลแต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหน่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอฟแวร์การสังเคราะห์วงจรก็จะมีการรายงานผลว่าโมเดลที่ออกแบบไปนั้นเป็นอย่างไร เช่นมีค่าความหน่วง(Delay) เท่าใด ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

2. การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆสำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่ง คือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำ โดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกท(gate), ฟลิป-ฟลอบ (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGA

หลังจากทำขั้นตอนนี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงลอจิก (Logic Delay) ส่วนซอฟต์แวร์จะรวมเอาซอฟต์แวร์ย่อยอื่นๆ อีก เพื่อให้การทำ PPR (Partitioning Placement & Routing) เป็นไปอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควรจะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวงวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆ ให้อยู่ใกล้กันโดยเฉพาะส่วนที่มีการเชื่อมต่อสัญญาณด้วยกันนอกจากนั้นการกำหนดตำแหน่งขา I/O (I/O pin) ตามตำแหน่งขา I/O ของ FPGA บนแผ่น PCB ก็จะมีผลโดยตรงเลยคือซอฟต์แวร์จะวาง I/O ลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่งบางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา I/O ควรกำหนดตำแหน่งให้เหมาะสมหรือไม่ก็ให้ซอฟต์แวร์จัดการเอง

4 การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆภายในอุปกรณ์ FPGA ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมด (เนื่องจากจำนวนทรัพยากรสำหรับเชื่อมต่อสัญญาณนั้นมีอยู่จำกัด) หรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับ

ผู้ออกแบบสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟท์แวร์หรือผู้ออกแบบจะทำการเชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับทางเวลา จะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

5 ความหน่วงด้านเวลา (Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วงที่เกิดขึ้นนั้นแยกได้เป็นสองประเภทคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 ความหน่วงทางลอจิก (Logic delay) เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เอง

5.1.2 ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ (Routing delay)เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA

โดยปกติแล้วค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะ ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณมันจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น และเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น

ค่าความหน่วงที่ได้หลังจากการวางอุปกรณ์ และเชื่อมต่อสัญญาณแล้วจะมีค่าความหน่วงที่ค่อนข้างแน่นอน ซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้น เป็นไปตามข้อกำหนดหรือไม่

5 การจำลองการทำงานของวงจร (Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่สำคัญอีกขั้นตอน หนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการทำงานของโมเดลว่าถูกต้องหรือไม่มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะมีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจรควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดลเกิดขึ้นตอนไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้เกิดข้อผิดพลาดตรงขั้นตอนนั้นๆ นั่นคือการทำการจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจสอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตามข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือถ้ามีจะแก้ไขให้ถูกต้อง

ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์การเชื่อมต่อสัญญาณ(Post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้องตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน (Standard Delay Format : SDF)ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่า วงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงานของ วงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่างๆ

6 การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR(Partitinoing ,Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด(Download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็นข้อมูลวงจร(Configuration data) ซึ่งอยู่ในรูปของบิตสตรีม (Bit stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการ ซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับอุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือ

ในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้นในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ Serial PROM ด้วยเพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานโมเดลครั้งต่อไปไม่ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยใช้วิธี EPROM หรือ Anti fuse ก็ไม่จำเป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

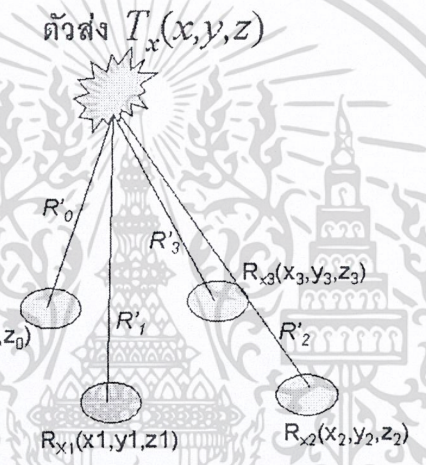
บทที่4

แนวความคิดในการออกแบบ

4.1 เป้าหมายของโครงงาน

โครงงานนี้สร้างเครื่องหาพิกัดของเมาส์สามมิติโดยใช้หลักการวัดค่าเวลาที่แตกต่างกัน(differences in the time of flights) Ultrasonic ใช้เดินทางจากตัวส่ง(เมาส์)ไปยังตัวรับUltrasonic 4 ตัวที่วางอยู่ ณ ตำแหน่งที่ทราบค่าแน่นอน

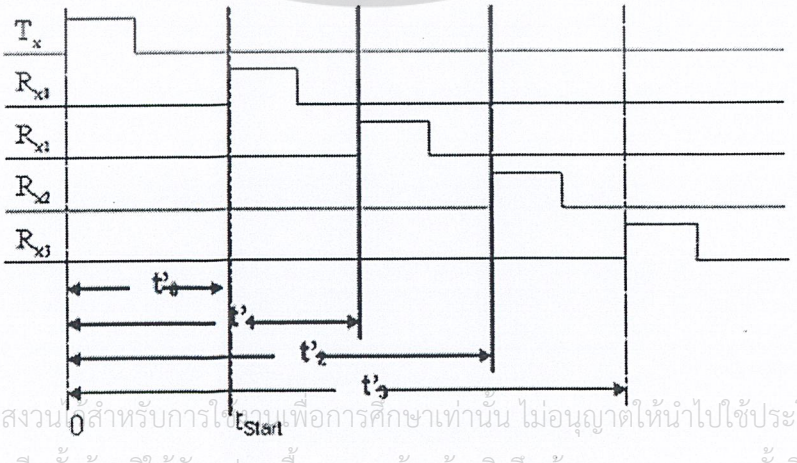
4.2 ตำแหน่งของตัวรับและตัวส่ง



รูปที่4.1 แสดงการวางของตัวส่งและตัวรับ

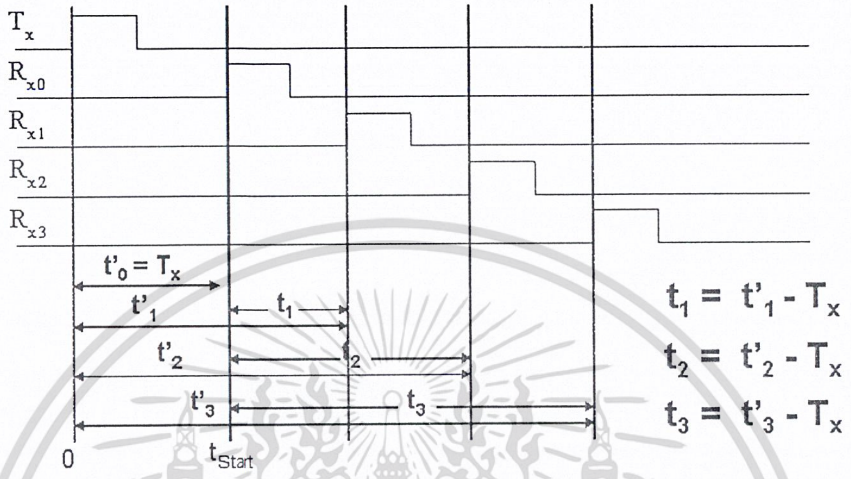
โดยตัวส่งจะลอยอยู่ในอากาศซึ่งมีพิกัด(X,Y,Z) จะส่งสัญญาณอัลตร้าโซนิกมาที่ตัวรับทั้ง4 ตัว ณ ตำแหน่งต่างๆ คือ R_1,R_2,R_3และR_4

Timing Diagramของสัญญาณของตัวส่งและตัวรับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Timing Diagram ของการทำงานแบบAsynchronous



การวัดเวลาการเดินทางของUltrasonicแบบดั้งเดิม

โดยทั่วไปการวัดเวลาในการเดินทางของสัญญาณเสียงจะต้องเริ่มจับเวลาตั้งแต่ตัวส่งเริ่มส่งสัญญาณไปจนตัวรับๆสัญญาณ ได้ทำให้วงจรภาคส่งและรับทำงานแบบ Synchronous ดังนั้นวงจรภาคส่งและรับจึงไม่อิสระต่อกัน

การวัดเวลาการเดินทางของUltrasonicที่ใช้ในโครงงานนี้

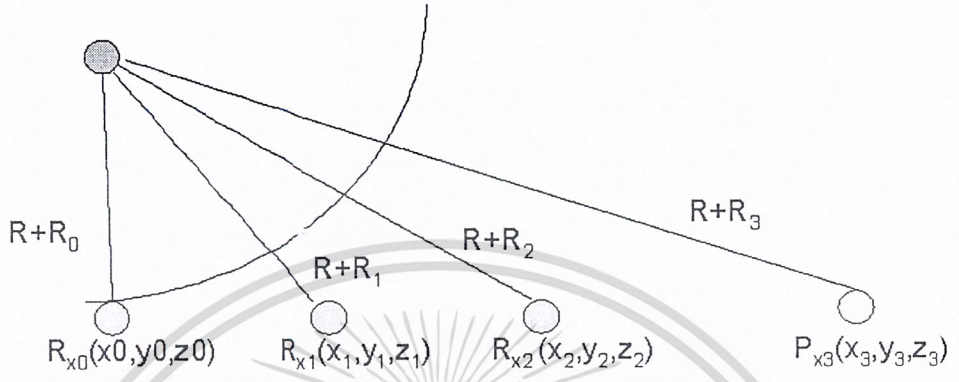
- โครงงานนี้ต้องการให้วงจรภาคส่งและรับเป็นอิสระต่อกัน หรือทำงานแบบ Asynchronous ดังนั้นเราจึงไม่สามารถจับเวลาตั้งแต่สัญญาณเริ่มส่งจากภาคส่งได้
- ในการหาตำแหน่ง(X,Y,Z) ของภาคส่ง เราจะใช้วิธีวัดหาเวลาที่แตกต่างกันที่สัญญาณ Ultrasonic ใช้เดินทางจากภาคส่งไปยังภาครับแต่ละตัว

การคำนวณหาค่า R0,R1,R2และ R3

ค่าเวลาที่แตกต่างกันเราจะสามารถคำนวณหาระยะทางได้โดยใช้ตัวนับในการจับเวลาเพื่อหาผลต่างของระยะทาง โดยตัวนับจะเริ่มทำงานพร้อมกัน แต่จะหยุดนับที่เวลาต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การคำนวณหาค่าพิกัด X,Y,Z



$$(x-x_0)^2 + (Y-Y_0)^2 + Z^2 = (R+R_0)^2$$

$$(x-x_1)^2 + (Y-Y_1)^2 + Z^2 = (R+R_1)^2$$

$$(x-x_2)^2 + (Y-Y_2)^2 + Z^2 = (R+R_2)^2$$

$$(x-x_3)^2 + (Y-Y_3)^2 + Z^2 = (R+R_3)^2$$

การกำหนดช่วงเวลาในการรับสัญญาณ

ในการรับสัญญาณ ตัวรับต้องรับสัญญาณได้ครบทั้ง 4 ตำแหน่งภายในช่วงเวลาหนึ่ง โดยเรากำหนดให้ช่วงเวลาที่มากที่สุดที่ตัวรับตัวสุดท้ายสามารถรับสัญญาณได้เป็น Tmax

การหาค่า T_{max}

$$T_{max} = d_{max} / V_s$$

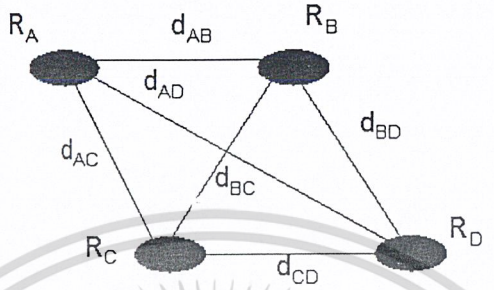
T_{max} = เวลามากที่สุดที่ตัวรับทั้ง 4 ตัวต้องรับค่าได้

d_{max} = ค่าระยะห่างระหว่างตัวรับ 2ตัวที่มากที่สุด

V_s = ความเร็วของคลื่นในตัวกลางอากาศ (331.45 + 0.607T m/sec)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า d_{max}

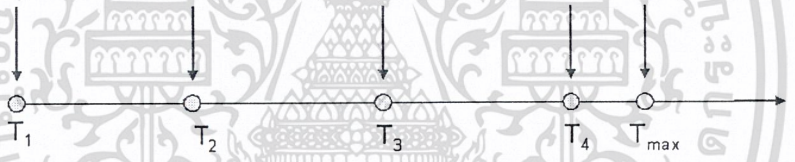


รูปที่4.2 แสดงระยะห่างของตัวรับแต่ละตัว

4.4 ประโยชน์ของ T_{max}

4.4.1 ตรวจสอบการเกิดปัญหาของภาครับ

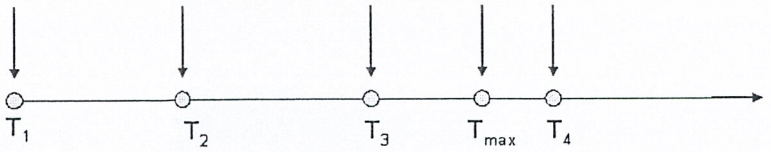
4.4.1.1 แสดงช่วงเวลาที่Senserรับสัญญาณได้ครบ ($T_1, T_2, T_3, T_4 \leq T_{max}$)



T_1, T_2, T_3, T_4 = เวลาที่ตัวรับทั้ง 4 ตัวรับสัญญาณได้ ตามลำดับ

T_{max} = เวลาที่มากที่สุดที่ ตัวรับทั้ง 4 ตัวต้องรับสัญญาณได้

4.4.1.2 แสดงช่วงเวลาที่Senserรับสัญญาณได้ไม่ครบ ($T_4 \geq T_{max}$)



T_1, T_2, T_3, T_4 = เวลาที่ตัวรับทั้ง 4 ตัวรับสัญญาณได้ ตามลำดับ

T_{max} = เวลาที่มากที่สุดที่ ตัวรับทั้ง 4 ตัวต้องรับสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 ใช้ในการกำหนดความถี่ให้กับบอร์ด FPGA

ความถี่ของบอร์ดFPGA

$$F_{Counter} < 2^{N} / T_{max}$$

$F_{Counter}$ = ความถี่ในการนับของบอร์ด FPGA

N = จำนวนบิตของข้อมูลในการนับ

T_{max} = เวลามากที่สุดที่ตัวรับทั้ง 4 ตัวต้องรับค่าได้

4.4.3 ใช้ในการคำนวณขนาด Pulse ของภาคส่งที่เหมาะสม

ขนาด Pulse ของภาคส่ง

$$T_{Pulse} > T_{max} + T_{Process}$$

T_{Pulse} = ขนาดของพัลส์ภาคส่ง

T_{max} = เวลามากที่สุดที่ตัวรับทั้ง 4 ตัวต้องรับค่าได้

$T_{Process}$ = เวลาที่ใช้ในการProcess ของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
การออกแบบและการทำงานของเม้าส์สามมิติ

5.1 แนวทางการออกแบบเม้าส์สามมิติ

5.1.1 จุดประสงค์ก่อนออกแบบ

1. วงจรภาคส่งและรับคลื่นอัลตร้าโซนิคสามารถทำงานได้
2. ควบคุมให้สามารถวัดระยะทางระหว่างตัวรับ ตัวส่งได้
3. นำระยะทางที่ได้มาคำนวณหาพิกัดตำแหน่งบนระนาบ 3 มิติ
4. สามารถแสดงผลของพิกัดตำแหน่งบนเครื่องคอมพิวเตอร์ได้

5.1.2 ขั้นตอนการออกแบบ

เพื่อใช้บอกตำแหน่งของตัวชี้ตำแหน่ง บนจอคอมพิวเตอร์ โดยมีหลักการทำงานประมาณได้ดังนี้ เริ่มด้วย ตัวชี้ตำแหน่งส่งสัญญาณไปที่ตัวรับ โดยตัวรับจะมีอยู่ 4 ตำแหน่ง คือ P0,P1,P2 และ P3 ตามลำดับ ที่ตำแหน่ง$(X_0,Y_0,0)$,$(X_1,Y_1,0)$,$(X_2,Y_2,0)$ และ $(X_3,Y_3,0)$ ตามลำดับ การหาพิกัด (X,Y,Z) ทำได้โดยการวัดระยะทางระหว่างตัวชี้ไปยังจุดอ้างอิงทั้ง4 คือ R+R0,R+R1,R+R2 และ R+R3 โดยใช้ FPGA เป็นตัวควบคุม จากนั้นก็จะส่งข้อมูลของระยะทางทั้ง 4 ไปยังคอมพิวเตอร์ โดยออกทางพอร์ทขนานของบอร์ด FPGA เพื่อให้คอมพิวเตอร์นำค่าที่ได้ไปคำนวณหาพิกัด (X,Y,Z) และนำไปแสดงผลเป็นภาพสามมิติต่อไป

ภาคส่งคลื่น

เนื่องจากตัวส่งสัญญาณ Ultrasonic จะมีความถี่ใช้งานที่ 40 KHz วงจรภาคส่งจึงเป็นวงจรผลิตความถี่ 40 KHz ป้อนให้กับตัวส่ง แต่เนื่องจากต้องเป็นตัวป้อนสัญญาณเพื่อเป็นจุดอ้างอิงให้กับ FPGA ด้วย ซึ่งสัญญาณที่จะป้อนให้กับ FPGA ต้องเป็นสัญญาณพัลส์ที่มีความถี่ต่ำๆ ดังนั้น วงจรภาคส่งนี้จะทำหน้าที่ผลิตความถี่ 40 KHz และผลิตสัญญาณพัลส์แล้วนำมาผสมกัน ป้อนให้กับตัวส่งอัลตร้าโซนิคและสัญญาณพัลส์ที่ผลิตขึ้นมานี้ก็จะถูกส่งให้กับ FPGA ด้วยพร้อมๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาครับคลื่น

วงจรในส่วนของภาครับคลื่น จะต้องสามารถรับสัญญาณที่ส่งมาจากตัวส่ง ซึ่งเป็นสัญญาณที่ถูกผสมกันระหว่างความถี่สูง 40 KHz กับสัญญาณพัลส์ความถี่ต่ำ 70 Hz แล้วทำการกรองเอาเฉพาะสัญญาณพัลส์ 70 Hz เพื่อเป็นสัญญาณให้กับ FPGA โดยจะสร้างวงจรภาครับขึ้นมา 4 ชุด เมื่อวงจรภาครับตัวใดตัวหนึ่งได้รับสัญญาณเป็นตัวแรก เคาน์เตอร์ตัวอื่นจะเริ่มนับและหยุดเมื่อวงจรภาครับชุดนั้นได้รับสัญญาณครบทั้ง 4จุด หรือ รอจนถึงเวลาที่ได้คำนวณไว้ว่าถ้าเวลาผ่านไปเท่านี้ภาครับยังรับสัญญาณได้ไม่ครบให้ตัวรับเริ่มรับค่าใหม่ ค่าที่ได้จะนำไปคำนวณหาตำแหน่งของตัวชี้ต่อไป

ภาคประมวณผล

วงจรในส่วนนี้จะรวมไปถึงส่วนของการแสดงผลบนเครื่องคอมพิวเตอร์ด้วย โดยแยกเป็น

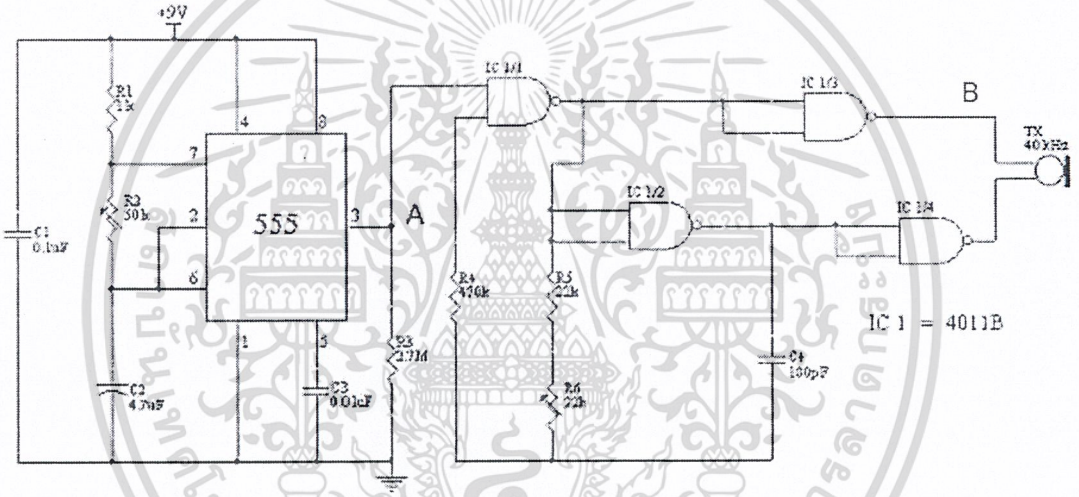
1. วงจรในส่วนของ FPGA ที่ทำหน้าที่รับสัญญาณจากวงจรภาครับ เปรียบเทียบและนับระยะเวลาที่เกิดขึ้นระหว่างตัวส่งกับตัวรับแต่ละจุดโดยใช้การเขียนโปรแกรมด้วยภาษา VHDL ค่าเวลาที่ได้จะส่งให้กับ MCS-51ซึ่งทำหน้าที่ส่งค่าเวลาที่นับได้ให้กับคอมพิวเตอร์คำนวณผ่านทางพอร์ตสื่อสารแบบขนาน
2. ส่วนของการคำนวณและแสดงผล ซึ่งเป็นการทำงานบนเครื่องคอมพิวเตอร์ จะรับค่าการนับจากบอร์ดFPGA ทำการคำนวณหาพิกัดตำแหน่ง และแสดงผลโดยใช้โปรแกรม Visual Basic แล้วนำพิกัดที่คำนวณได้ไปขึ้นภาพสามมิติต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การทำงานของเม้าส์ 3 มิติ

วงจรภาคส่งคลื่น

ภาคส่งสัญญาณจะทำหน้าที่ในการให้กำเนิดสัญญาณ ซึ่งคลื่นที่เราใช้ในที่นี้จะเป็นคลื่นอัลตราโซนิก โดยตัวส่งสัญญาณที่เราใช้จะสามารถทำงานได้ดีที่สุดที่ความถี่ 40 kHz ดังนั้นเราจะนำสัญญาณความถี่ต่ำ 70 Hz ซึ่งเป็นสัญญาณข้อมูล ไปผสมกับสัญญาณความถี่ 40 kHz เพื่อให้ตัวส่งสัญญาณอัลตร้าโซนิกสามารถส่งสัญญาณไปได้ไกลที่สุด โดยเราจะใช้วงจรดังรูป 5.1



รูปที่ 5.1 แสดงภาคส่งคลื่น Ultrasonic

การทำงานของวงจรเราสามารถแบ่งได้เป็น 2 ส่วน โดยส่วนแรกคือ ส่วนกำเนิดสัญญาณความถี่ต่ำ 70 Hz ในส่วนนี้เราจะใช้ไอซีเบอร์ 555 มาต่อเป็นวงจรอะสเตเบิ้ลมัลติไวเบรเตอร์ ซึ่งให้สัญญาณสี่เหลี่ยมออกมาที่ขาเอาต์พุตของวงจร คือขา 3 แล้วเราจะนำสัญญาณที่ได้นี้ไปผสมกับสัญญาณความถี่ 40 kHz โดยค่าของ R และ C ที่ใช้ในวงจรจะสามารถคำนวณได้จากสมการ

$$f = \frac{1.44}{(R1 + 2R2)C2}$$

และจากสมการ Duty Cycle

$$D = \frac{R1 + R2}{R1 + 2R2} \times 100\ \%$$

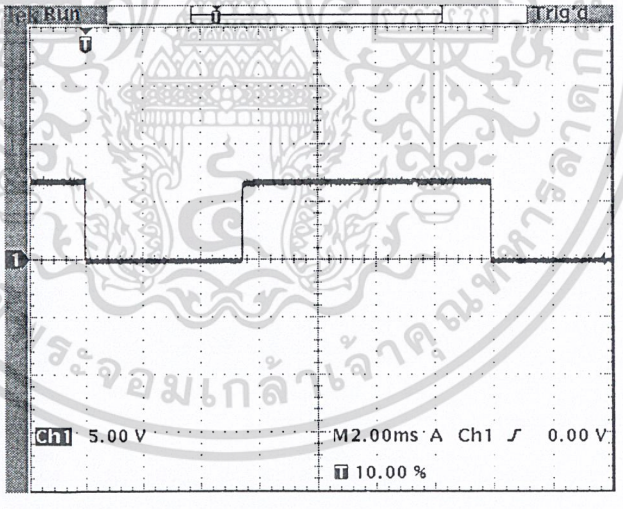
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากทั้งสองสมการ เราสามารถกำหนดความถี่และ Duty Cycle ของวงจรได้โดยการเปลี่ยนค่าของ R1 , R2 และ C2 โดยในที่นี้เราจะใช้ความถี่ที่ 70 Hz และกำหนดให้มี D ประมาณ 50%

วงจรส่วนที่ สอง เราจะใช้ไอซีเบอร์ 4011B ซึ่งภายในประกอบด้วยแนนด์เกต (NAND gate) 4 ตัว โดยวงจรนี้จะเอาเกต 2 ตัวมาทำเป็นวงจรออสซิเลเตอร์สร้างคลื่นรูปสี่เหลี่ยมขึ้นมา เกตอีก 2 ตัวที่เหลือต่อเป็นภาคบัฟเฟอร์เพื่อกั้นโหลดไม่ให้มีผลต่อวงจรออสซิเลเตอร์ บัฟเฟอร์แต่ละตัวจะนำเอาสัญญาณสี่เหลี่ยม 2 สัญญาณ ซึ่งมีเฟสต่างกันมาป้อนให้แต่ละขาของตัวส่ง ในลักษณะของพุชพูล เพื่อเพิ่มขนาดสัญญาณคร่อมตัวส่งให้สูงขึ้นเป็นประมาณ 2 เท่าเมื่อเทียบกับการต่อตัวส่งเข้ากับเอาต์พุตของบัฟเฟอร์เพียงตัวเดียว ความถี่ของการออสซิเลตก็สามารถปรับได้โดยการปรับค่า R6

จากรูปที่5.1 รูปสัญญาณ ณ จุดต่างๆเป็นดังต่อไปนี้

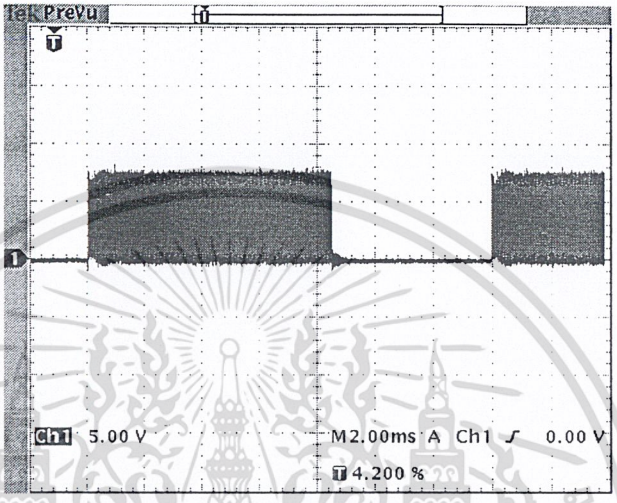
1) รูปสัญญาณ ณ จุดA



รูปที่5.2 แสดงสัญญาณที่จุดA ในรูปที่5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

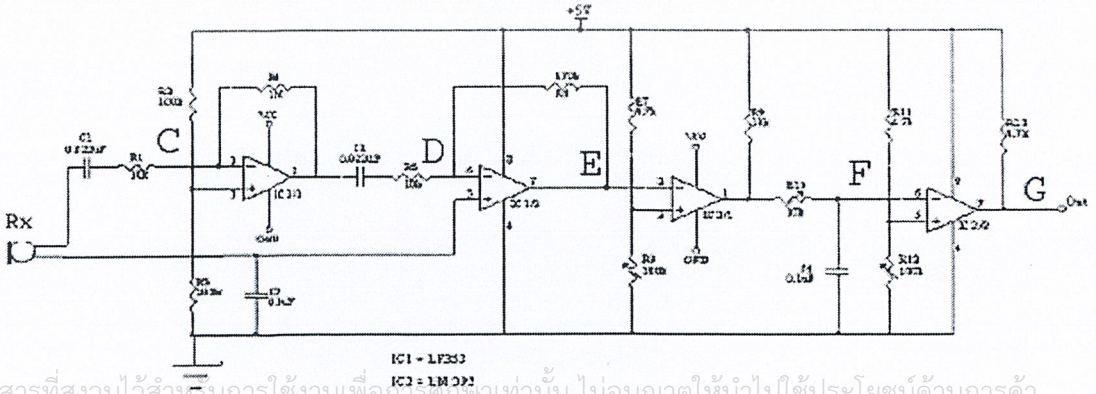
2) รูปสัญญาณ ณ จุด B



รูปที่5.3 แสดงสัญญาณที่จุดB ในรูปที่ 5.1

วงจรภาครับคลื่น

หน้าที่ของวงจรภาครับคลื่นคือการแปลงที่รับมาในรูปของสัญญาณที่ถูกผสมกันระหว่างสัญญาณความถี่สูง (40kHz) กับสัญญาณความถี่ต่ำ (70kHz) ให้เป็นพัลส์ขนาด 5V แล้วหลังจากนั้นก็จะทำการส่งเป็นสัญญาณ Interrupt ให้ FPGA เพื่อทำการประมวณผลต่อไป วงจรภาครับแสลงดังรูป 5.2



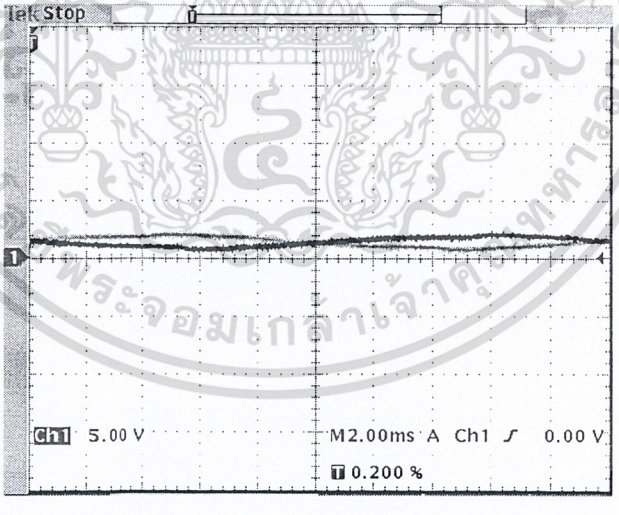
รูปที่ 5.4 แสดงภาครับคลื่น Ultrasonic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรจะเริ่มจาก เมื่อมีสัญญาณคลื่นความถี่ประมาณ 40 kHz มาตกกระทบที่ตัวรับ จะทำให้เกิดเป็นแรงดันค่าน้อยๆขึ้นที่ขั้วของตัวรับ ค่าแรงดันดังกล่าวจะถูกส่งไปยังวงจรขยายสัญญาณ โดยวงจรขยายจะมีอยู่ 2 ชุด ชุดแรกคือ IC 1/1 จะมีอัตราขยายประมาณ 100 เท่า กำหนดโดย R1 และ R4 และชุดที่สอง IC 1/2 จะมีอัตราขยายประมาณ 47 เท่า กำหนดโดย R5 และ R6 หลังจากนั้นจะถูกนำไปผ่านวงจรคอมพาราเตอร์ ซึ่งสัญญาณที่ได้ยังคงเป็นลักษณะของสัญญาณความถี่สูงรวมมากับสัญญาณความถี่ต่ำ ดังนั้นจะนำสัญญาณนี้ไปผ่านวงจร Low Pass Filter ซึ่งจะสร้างจาก R และ C เพื่อที่จะกรองความถี่ 40 kHz ทิ้งไปเหลือแต่สัญญาณความถี่ 70 Hz เท่านั้น โดยจะใช้ความถี่ Cutoff ประมาณ 1kHz ดังนั้นจะเหลือแต่สัญญาณความถี่ต่ำเท่านั้น แล้วจะนำสัญญาณดังกล่าวไปผ่านวงจรคอมพาราเตอร์อีกครั้ง เพื่อให้พัลส์ที่ได้ตรงแล้วจึงส่งไปยัง FPGA เพื่อนำไปประมวลผลและควบคุมต่อไป

จากรูปที่5.4 รูปสัญญาณ ณ จุดต่างๆ เป็นดังต่อไปนี้

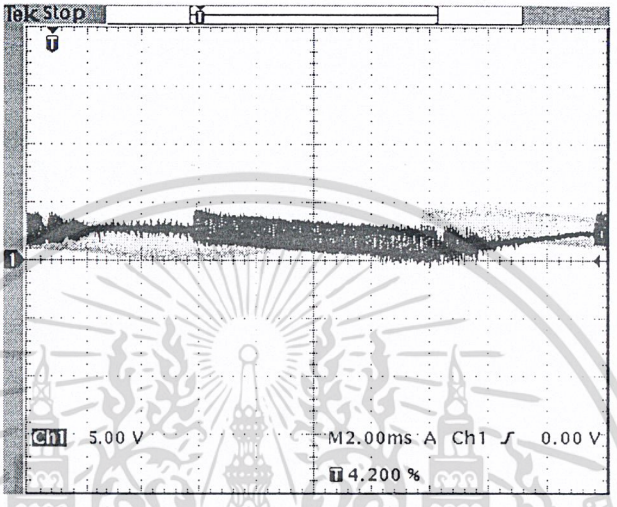
1) รูปสัญญาณ ณ จุด C



รูปที่ 5.5 แสดงสัญญาณที่จุดC ในรูปที่ 5.4

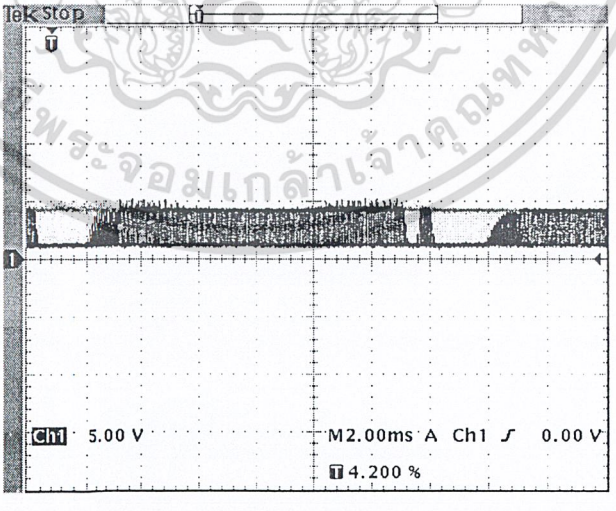
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) รูปสัญญาณ ณ จุด D



รูปที่5.6 แสดงสัญญาณที่จุดD ในรูปที่5.4

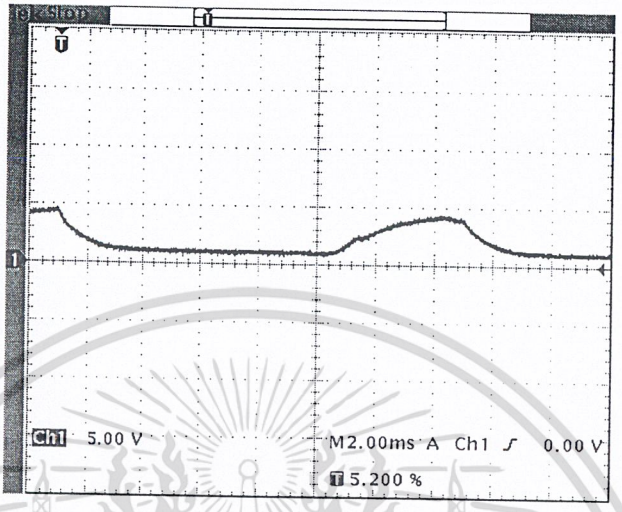
3) รูปสัญญาณ ณ จุด E



รูปที่5.7 แสดงสัญญาณที่จุดE ในรูปที่5.4

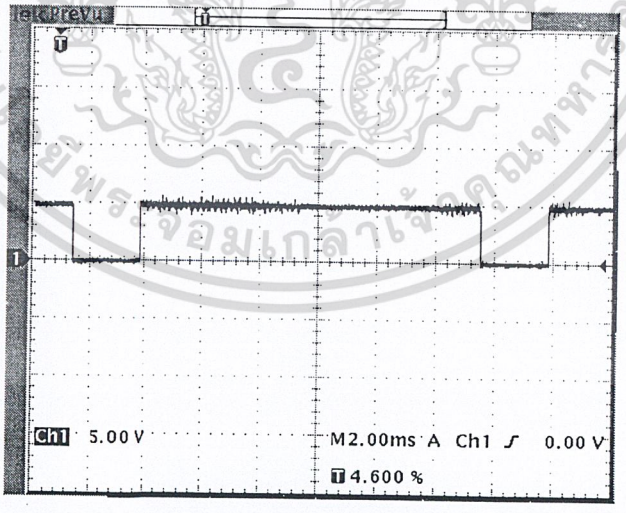
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) รูปสัญญาณ ณ จุด F



รูปที่5.8 แสดงสัญญาณที่จุดF ในรูปที่5.4

5) รูปสัญญาณ ณ จุด G



รูปที่5.9 แสดงสัญญาณที่จุดG ในรูปที่5.4

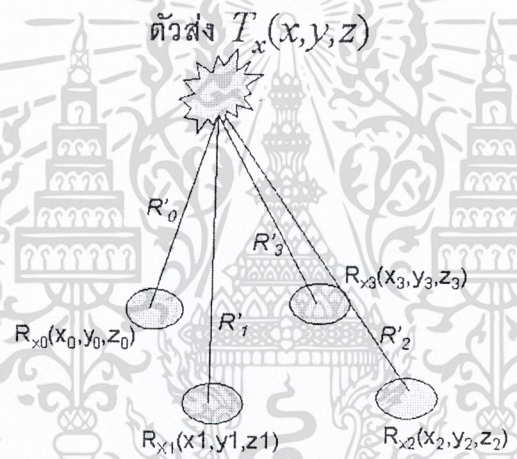
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 วงจรภาคประมวลผล

วงจรในส่วนของภาคประมวลผลจะประกอบไปด้วยส่วน FPGA , Visual Basic ซึ่งในส่วนของ FPGAจะเขียนด้วยโปรแกรม MAX+plus2 โดยใช้ภาษา VHDL ซึ่งเป็นตัวนับ (Counter) 4ตัว และเชื่อมต่อกับคอมพิวเตอร์โดยผ่านทางพอร์ทขนาน ซึ่งค่าที่ได้จะส่งไปคำนวณในโปรแกรม Visual Basic และแสดงเป็นภาพสามมิติต่อไป

5.3.1 การคำนวณหาพิกัดตำแหน่ง

การคำนวณหาพิกัดตำแหน่ง อาศัยหลักการทางคณิตศาสตร์ โดยมีวิธีการคำนวณดังนี้



รูปที่ 5.10 แสดงการวางตำแหน่งของตัวส่งและตัวรับ

สมมติว่าตัวส่งอยู่ที่ตำแหน่ง (x,y,z) และเราทำการวางตำแหน่งของตัวรับทั้ง4 ตัวตามจุดดังรูป เมื่อส่งสัญญาณUltrasonic จากตัวส่งไปยังตัวรับทั้ง 4ตัว ระยะเวลาที่คลื่นเดินทางระหว่างตัวส่งกับตัวรับจะถูกส่งไปที่บอร์ด FPGA เพื่อนับค่าเวลาที่ต่างกัน และส่งไปที่คอมพิวเตอร์ ผ่านทางพอร์ทขนาน

สัญญาณจากภาคส่งคลื่นจะมีลักษณะเป็น Sin Wave ที่มีความถี่ 40KHz เป็นช่วงๆ มีลักษณะคล้ายๆพัลส์ ความถี่ 70 Hz ส่งไปที่ภาครับ โดยภาครับจะแปลงสัญญาณที่ได้เป็นพัลส์ที่มีความถี่ 70 Hz เพื่อไปใช้งานต่อไป

เมื่อตัวรับตัวแรกได้รับสัญญาณดังกล่าวแล้วจะทำให้เคาน์เตอร์เริ่มทำงาน โดยรับค่าไปเรื่อยๆจนครบทุกตัว แล้วนำค่าที่ได้ไปคำนวณหาค่า R_0 ,R_1 ,R_2 ,R_3 ออกมาแล้วนำค่า R_0 ,R_1 ,R_2 ,R_3 ไปคำนวณหาค่า x,y,z ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาพิกัด

1. การคำนวณเพื่อให้ได้ค่า R_0 , R_1 , R_2 , R_3

เมื่อเราได้ค่าที่เคาน์เตอร์แต่ละตัวนับได้ (n) เราสามารถหาเวลาที่ใช้ได้ (t) จากสูตรต่อไปนี้

$$T = N/f$$

โดย T คือ เวลาที่แตกต่างกันระหว่างตัวรับตัวแรกกับตัวรับตัวนั้น

N คือ จำนวนที่เคาเตอร์นับได้

f คือ ความถี่ที่เคาเตอร์ใช้นับ

และเราสามารถหาค่า R_0 , R_1 , R_2 , R_3 ได้จากสูตรต่อไปนี้

จาก $S = VT$

จะได้ $R_0 = VN_0/f$

$R_1 = VN_1/f$

$R_2 = VN_2/f$

$R_3 = VN_3/f$

โดย V คือ ความเร็วของคลื่นอัลตราโซนิค

R_0 , R_1 , R_2 , R_3 คือ รัศมีที่แตกต่างกันระหว่างตัวรับตัวนั้นกับตัวรับที่อยู่ใกล้ตัวส่งมากที่สุด

หลังจากนั้นเราจะนำค่า R_0 , R_1 , R_2 , R_3 ไปคำนวณหาค่า X , Y , Z ต่อไป

2. การคำนวณหาค่า X , Y , Z

จากรูปที่ 3.3 เราสามารถคำนวณได้ดังนี้

$$(X-X_0)^2+(Y-Y_0)^2+Z^2 = (R+R_0)^2 \quad \text{.....(1)}$$

$$(X-X_1)^2+(Y-Y_1)^2+Z^2 = (R+R_1)^2 \quad \text{.....(2)}$$

$$(X-X_2)^2+(Y-Y_2)^2+Z^2 = (R+R_2)^2 \quad \text{.....(3)}$$

$$(X-X_3)^2+(Y-Y_3)^2+Z^2 = (R+R_3)^2 \quad \text{.....(4)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก (1)

$$X^2 - 2X_0X + X_0^2 + Y^2 - 2Y_0Y + Y_0^2 + Z^2 = R^2 + 2R_0R + R_0^2$$

$$X^2 + Y^2 + Z^2 - R^2 = 2X_0X + 2Y_0Y + 2R_0R - X_0^2 - Y_0^2 + R_0^2 \quad \text{.....(5)}$$

ในทำนองเดียวกันจาก (2) , (3) , (4)

$$X^2 + Y^2 + Z^2 - R^2 = 2X_1X + 2Y_1Y + 2R_1R - X_1^2 - Y_1^2 + R_1^2 \quad \text{.....(6)}$$

$$X^2 + Y^2 + Z^2 - R^2 = 2X_2X + 2Y_2Y + 2R_2R - X_2^2 - Y_2^2 + R_2^2 \quad \text{.....(7)}$$

$$X^2 + Y^2 + Z^2 - R^2 = 2X_3X + 2Y_3Y + 2R_3R - X_3^2 - Y_3^2 + R_3^2 \quad \text{.....(8)}$$

จากนั้นเราเอาสมการที่ (5) = (6) , (5) = (7) , (5) = (8) จะได้สมการใหม่อีก 3 สมการ

จาก (5) = (6) จะได้

$$2X_0X + 2Y_0Y + 2R_0R - X_0^2 - Y_0^2 + R_0^2 = 2X_1X + 2Y_1Y + 2R_1R - X_1^2 - Y_1^2 + R_1^2$$

$$2(X_0 - X_1)X + 2(Y_0 - Y_1)Y + 2(R_0 - R_1)R = X_0^2 - X_1^2 + Y_0^2 - Y_1^2 - R_0^2 + R_1^2$$

ในทำนองเดียวกันอีก 2 สมการที่ได้ คือ

$$2(X_0 - X_2)X + 2(Y_0 - Y_2)Y + 2(R_0 - R_2)R = X_0^2 - X_2^2 + Y_0^2 - Y_2^2 - R_0^2 + R_2^2$$

$$2(X_0 - X_3)X + 2(Y_0 - Y_3)Y + 2(R_0 - R_3)R = X_0^2 - X_3^2 + Y_0^2 - Y_3^2 - R_0^2 + R_3^2$$

กำหนดให้

$$A_1 = 2(X_0 - X_1) \qquad A_2 = 2(X_0 - X_2) \qquad A_3 = 2(X_0 - X_3)$$

$$B_1 = 2(Y_0 - Y_1) \qquad B_2 = 2(Y_0 - Y_2) \qquad B_3 = 2(Y_0 - Y_3)$$

$$C_1 = 2(R_0 - R_1) \qquad C_2 = 2(R_0 - R_2) \qquad C_3 = 2(R_0 - R_3)$$

$$D_1 = X_0^2 - X_1^2 + Y_0^2 - Y_1^2 - R_0^2 + R_1^2$$

$$D_2 = X_0^2 - X_2^2 + Y_0^2 - Y_2^2 - R_0^2 + R_2^2$$

$$D_3 = X_0^2 - X_3^2 + Y_0^2 - Y_3^2 - R_0^2 + R_3^2$$

จะได้

$$A_1X + B_1Y + C_1R = D_1$$

$$A_2X + B_2Y + C_2R = D_2$$

$$A_3X + B_3Y + C_3R = D_3$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix}$$

กำหนดให้

$$U = \begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}$$

จะได้

$$X = \frac{\begin{vmatrix} D_1 & B_1 & C_1 \\ D_2 & B_2 & C_2 \\ D_3 & B_3 & C_3 \end{vmatrix}}{U}$$

$$Y = \frac{\begin{vmatrix} A_1 & D_1 & C_1 \\ A_2 & D_2 & C_2 \\ A_3 & D_3 & C_3 \end{vmatrix}}{U}$$

$$R = \frac{\begin{vmatrix} A_1 & B_1 & D_1 \\ A_2 & B_2 & D_2 \\ A_3 & B_3 & D_3 \end{vmatrix}}{U}$$

$$Z = \sqrt{R^2 - (X - X_0)^2 - (Y - Y_0)^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 หลักการทำงานของส่วนFPGA

Block1 (ส่วนกำหนดสัญญาณเริ่มทำงาน)

มีหน้าที่หลักในการกำหนดสัญญาณเริ่มทำงานให้กับ วงจรส่วนCounter โดยในเริ่มแรกจะเข้ามาที่ Block1 ทั้ง4 สัญญาณผ่านเข้ามาทาง Input1-Input4 ซึ่งสัญญาณ Inputทั้ง 4ตำแหน่งนั้น สัญญาณที่ผ่านเข้ามาตำแหน่งแรกจะเป็นตัวกำหนดสัญญาณเริ่มต้น(Start) ให้กับวงจรภาค Counter ต่อไป

Block2(ส่วนวงจรภาคCounter 16 bit)

มีหน้าที่หลักคือการนับสัญญาณ Clock เพื่อส่งต่อไปที่ส่วนควบคุมการทำงาน เริ่มโดยการรับสัญญาณStart จากส่วนกำหนดสัญญาณเริ่มต้น วงจรส่วน Counterจะเริ่มนับและจะหยุดนับเมื่อได้รับสัญญาณจากOutputของBlockที่1 ซึ่งจะสังเกตได้ว่าจะมีวงจรส่วนCounter ชุดหนึ่ง คือ ชุดแรกที่หยุดนับทันทีที่ได้รับสัญญาณ Start วงจรชุดแรกจะนับได้เท่ากับ0 และเมื่อวงจร Counter ชุดต่อมาได้รับสัญญาณOutput ที่มาจากชุดนั้น วงจรCounter ชุดนั้นก็จะหยุดนับ เช่น วงจร Counter ชุด B ได้รับสัญญาณ Output จากชุดB วงจรCounter จากชุด B ก็จะหยุดรับแล้วส่งค่าที่นับได้ไปยังการทำงาน Block ต่อไป โดยการส่งไปยังชุดต่อไปนี้เนื่องจากเป็น Counter แบบ 16 bit แต่การส่งข้อมูลไปทางเครื่องคอมพิวเตอร์ ต้องส่งทีละ 8 bit จึงต้องแยกข้อมูลออกเป็น 2 ชุดคือชุด Lowbyte กับ Highbyte ข้อมูลต่างๆก่อนที่จะส่งไปยัง Block ต่อไป จะส่งสัญญาณ Send ไปก่อนเพื่อบอกว่าข้อมูลที่จะเก็บนั้นเป็นข้อมูลชุดใดจะได้เก็บได้อย่างถูกต้อง

Block3(ส่วนควบคุมการทำงานโดยติดต่อกับคอมพิวเตอร์)

หลักการทำงานเป็นส่วนเชื่อมต่อกับเครื่องคอมพิวเตอร์เพื่อเลือกข้อมูลส่งไปประมวลผลที่เครื่องคอมพิวเตอร์ เพื่อคำนวณพิกัดของตัวส่งสัญญาณ

การทำงานของส่วนนี้เริ่มด้วยการรับสัญญาณ Send เพื่อเป็นสัญญาณในการบ่งบอกชุดข้อมูลที่จะเก็บข้อมูล เมื่อ Block 3 ได้รับสัญญาณ Send จากวงจรภาค Counter แล้วก็จะดึงข้อมูลที่เป็น Lowbyte และ Highbyte มาเก็บไว้ ทำเช่นนี้จนได้ข้อมูลครบทั้ง4 ชุด วงจรส่วนควบคุมนี้จะส่งสัญญาณ Finishไปบอกเครื่องคอมพิวเตอร์เพื่อให้คอมพิวเตอร์ส่งข้อมูลมาเพื่อเลือกชุดข้อมูลที่เก็บค่าที่นับได้ที่ประกอบด้วย Lowbyte และ Highbyte ไปที่เครื่องคอมพิวเตอร์ทำเช่นนี้จนคอมพิวเตอร์ได้รับข้อมูลครบทั้ง 4 ตำแหน่ง เป็นอันสิ้นสุดการทำงานของFPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร OR-GATE

เป็นส่วนควบคุมในกรณีวงจรส่วน Counter นับได้เกิน 65535 ซึ่งเกินกว่า 16 bit บอร์ด FPGA ก็จะส่งความผิดพลาดนี้ไปที่เครื่องคอมพิวเตอร์ต่อไป

การทำงานของส่วนประมวลผลผ่านทางโปรแกรม Visual Basic

เริ่มต้นโดยการรับสัญญาณFinish(ขาS3) จากบอร์ด FPGA ผ่านทางพอร์ทขนาน เพื่อเป็นการบอกว่าข้อมูลทั้ง4 ตำแหน่งได้เข้ามาเก็บไว้แล้วๆพร้อมที่จะส่งออกแล้ว เมื่อคอมพิวเตอร์ได้รับสัญญาณก็จะส่งสัญญาณเพื่อเลือกชุดข้อมูลที่เก็บค่าของตำแหน่งที่ต้องการไปทางบอร์ด FPGA เมื่อบอร์ดFPGA ได้รับสัญญาณControl จึงจะส่งข้อมูลชุดที่ถูกเลือกผ่านเข้าเครื่องคอมพิวเตอร์ผ่านทางพอร์ทขนาน ข้อมูลที่จะส่งมานั้นมีขนาด16 บิตเนื่องจากส่งได้ทีละ8บิตจึงต้องแบ่งข้อมูลออกเป็นสองช่วงโดยช่วงแรกเรียกว่า Lowbyte และ อีกชุดหนึ่งเรียกว่า Highbyte ข้อมูลที่จะถูกส่งออกไปก่อนช่วงแรกคือ Lowbyte แล้วตรวจสอบสัญญาณที่บอกว่าส่งข้อมูลเสร็จแล้วคือสัญญาณบิตที่(ขา S4) ของStatus ว่าเป็น"1" หรือไม่ถ้าเป็นจึงส่งข้อมูลHighbyte ออกไป แล้วตรวจสอบว่าบิตที่4(ขา S4) ของ Status เป็น"0" หรือไม่ถ้าเป็นก็เป็นอันสิ้นสุดการส่งข้อมูลข้อมูล16 บิตนี้ ทำเช่นนี้จนได้ข้อมูลครบทั้ง4 ตำแหน่งแล้วส่งพิกัดที่คำนวณได้ขึ้นเป็นภาพสามมิติ ก็เป็นอันจบการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่6

การทดลองและผลการทดลอง

ผลการทดลอง

6.1 ทดสอบหาระยะทางและมุมที่ตำแหน่งต่างๆของภาคส่งและภาครับ

ระยะทางระหว่างตัวส่งและตัวรับ(cm)	มุม 0 องศา	มุม 30 องศา	มุม 45 องศา	มุม 60 องศา	มุม 90 องศา
30	O	O	O	O	O
60	O	O	O	O	X
90	O	O	O	O	X
120	O	O	O	O	X
150	O	O	O	O	X
180	O	O	O	X	X
210	O	O	O	X	X
240	O	O	O	X	X
270	O	O	X	X	X
300	O	O	X	X	X
330	O	X	X	X	X
360	X	X	X	X	X

O = รับได้ X = รับไม่ได้

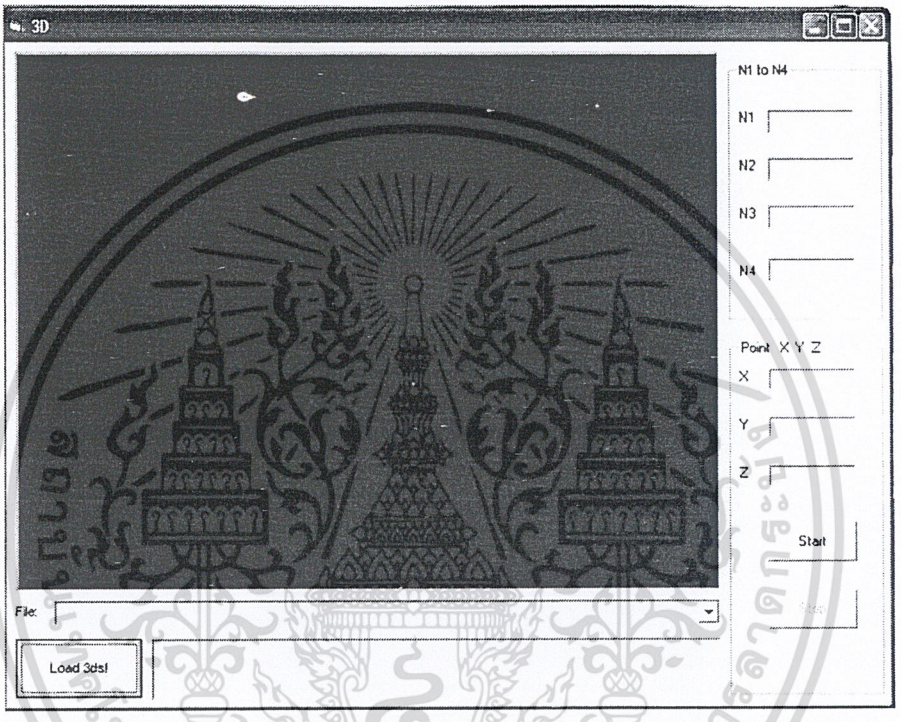
ตารางที่ 6.1 แสดงการทดสองวัดค่าตำแหน่งและมุมของภาคส่งและภาครับ

จากผลการทดลองจะเห็นว่า ระยะห่างระหว่างตัวรับและตัวส่ง มีผลโดยมีองค์ประกอบของมุมและระยะห่าง ของตัวรับและตัวส่งนั้น ถ้ามุมที่ตัวส่งกับตัวรับ ต่างกันมากขึ้นสัญญาณที่รับได้ก็จะรับได้น้อยลงจนกระทั่งรับไม่ได้เลย เมื่อวัดระยะห่างจากตัวรับและตัวส่งเมื่อห่างมากยิ่งขึ้น การรับของสัญญาณและคุณภาพของสัญญาณจะน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

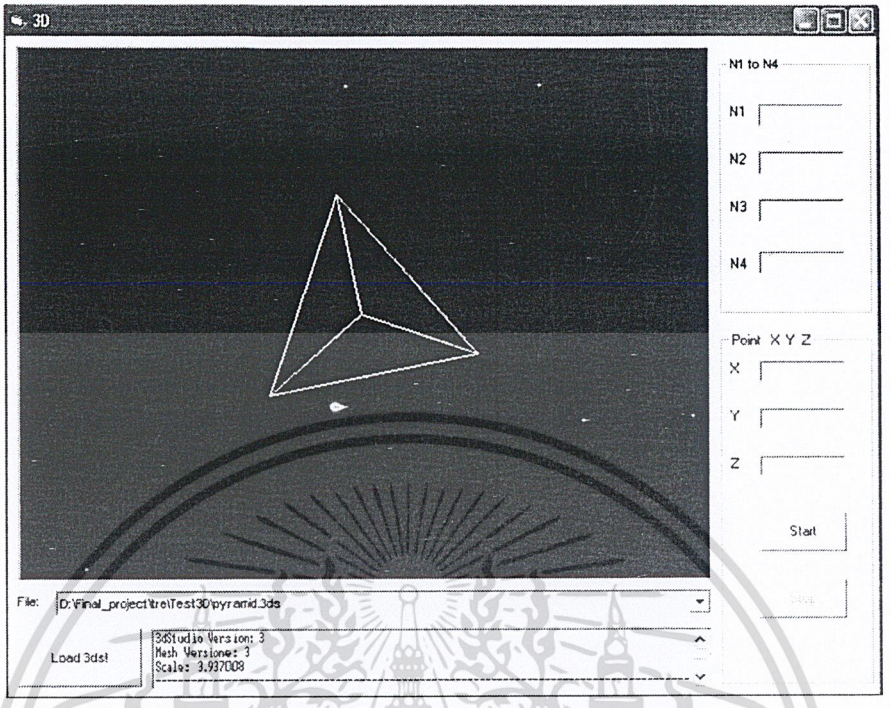
6.2 ทดสอบการทำงานของส่วนแสดงผล

ส่วนการแสดงผลเราจะแสดงตำแหน่งพิกัดในแนว (X,Y,Z) โดยใช้โปรแกรม Visual Basic ในการเขียน หน้าต่างของโปรแกรมขณะเปิดใช้งาน แสดงได้ดังรูป6.1

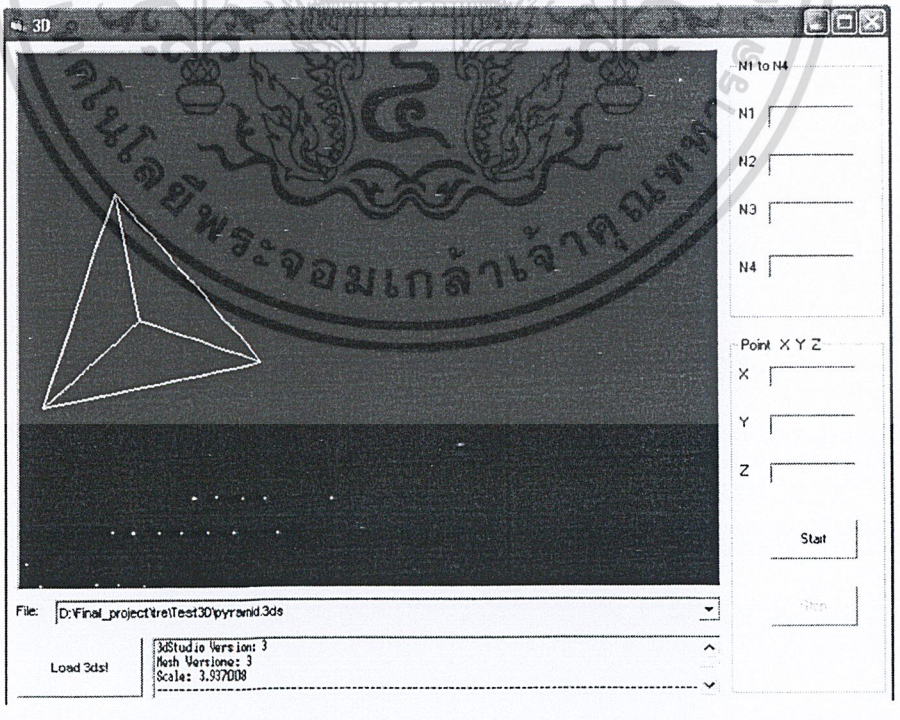


รูปที่6.1 แสดงหน้าต่างขณะเปิดใช้งานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

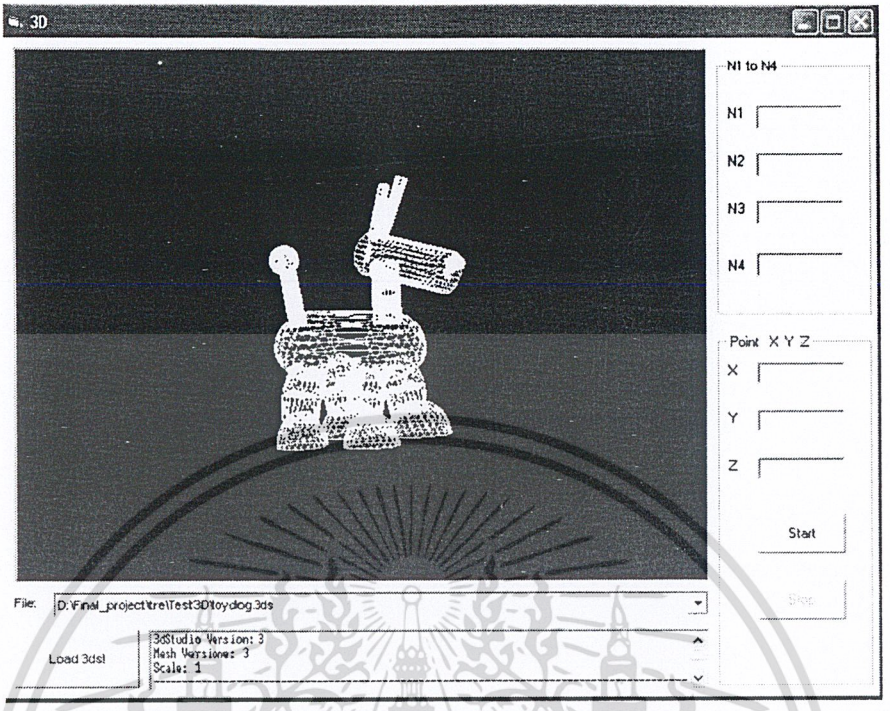


รูปที่6.2 แสดงภาพสามเหลี่ยมที่จุดเริ่มต้น

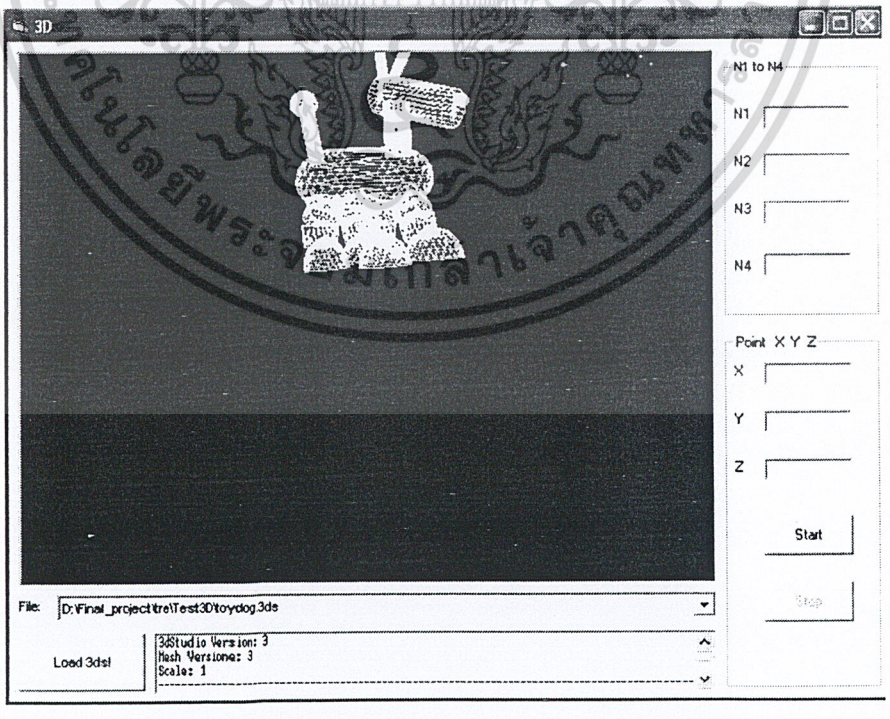


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่6.3 แสดงการเคลื่อนที่ของภาพในแนวแกน X



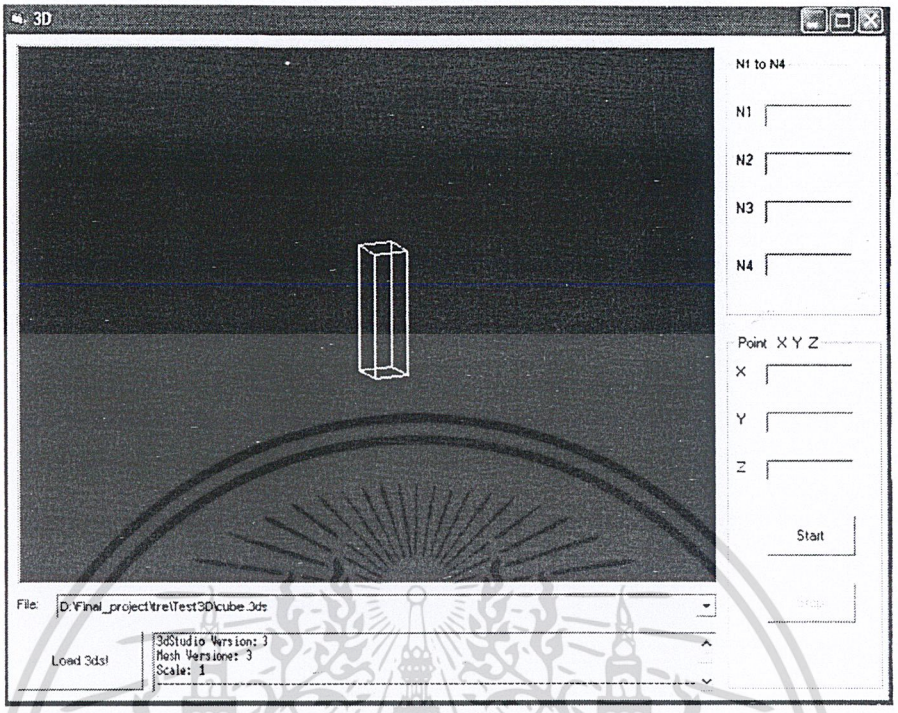
รูปที่6.4 แสดงภาพวัตถุที่จุดเริ่มต้น



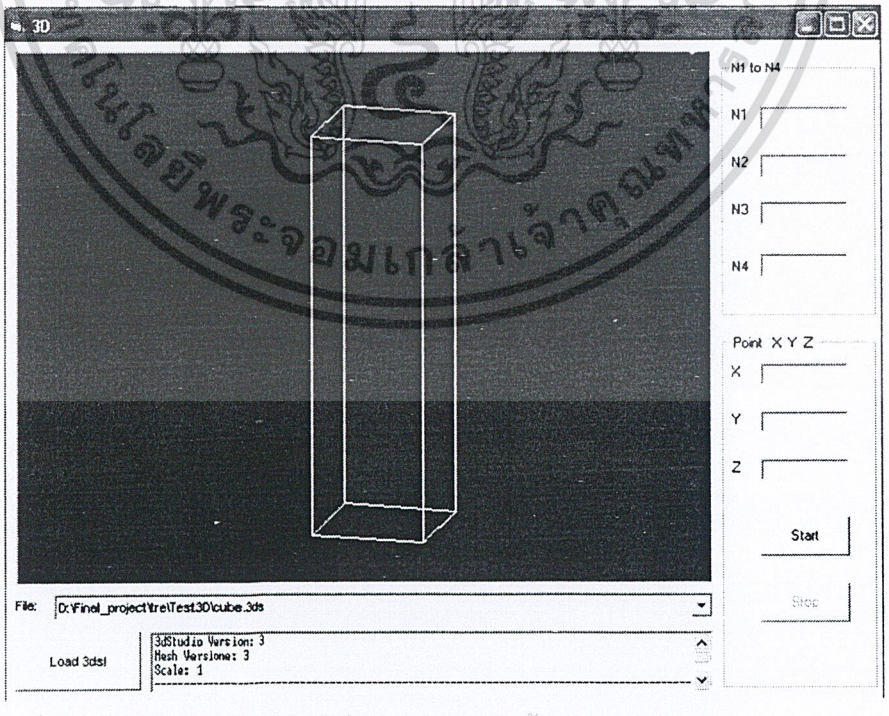
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่6.5 แสดงการเคลื่อนที่ของภาพในแนวแกน Y

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6.6 แสดงภาพสี่เหลี่ยมที่จุดเริ่มต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.7 แสดงการเคลื่อนที่ของภาพในแนวแกน Z

บทที่7
สรุปผลการทดลอง

7.1 สรุปผลการทดลอง

ภาคส่งสัญญาณ

ภาคส่งนี้มีขนาดเล็กวงจรไม่ซับซ้อน ได้ค่าความถี่ที่ต้องการอย่างถูกต้อง แต่มีปัญญาอยู่ที่ถ้าตัวส่งสัญญาณ Ultrasonic ยึดไม่ดีจะทำให้เกิดปัญหาที่ภาครับ แต่เมื่อลงบนแผ่นปริ้นต์แล้วปัญหาจึงหมดไป

ภาครับสัญญาณ

มีปัญหามากในเรื่องของสัญญาณรบกวน ที่แทรกมากับสัญญาณที่ต้องการส่งเมื่อตัวรับอยู่ห่างจากตัวส่งระยะหนึ่ง ดังนั้นในการออกแบบวงจรจึงต้องออกแบบให้สัญญาณรบกวนเข้ามาได้น้อยที่สุดเพื่อว่าเมื่อตัวส่งอยู่ไกลแล้ว ตัวรับและภาครับยังสามารถแยกแยะสัญญาณได้ถูกต้อง ซึ่งเมื่อทำปริ้นซ์แล้วปัญหาดังกล่าวจะน้อยลง

ภาคประมวลผล

ในส่วนนี้จะประกอบด้วย FPGA , Visual Basic ในส่วนของ Visual Basic ได้เขียนโปรแกรมเพื่อคำนวณค่าพิกัด(X,Y,Z) จากจำนวนของClock ที่นับได้เทียบกับคริสตอลบนบอร์ด FPGA นับค่าได้ผ่านสมการที่ได้คำนวณไว้ก่อนหน้านี้จะได้พิกัดของตัวส่งออกมา แล้วนำพิกัดที่คำนวณได้แสดงออกในรูปของภาพสามมิติ ในส่วนที่กล่าวมาข้างต้น พิจารณาที่โปรแกรม Visual Basic ได้เขียนโปรแกรมไว้แล้ว โปรแกรมเขียนขึ้นได้อย่างสมบูรณ์ ขาดอยู่ตรงที่ค่า Clock ที่ผ่านเข้ามาที่เครื่องคอมพิวเตอร์ผ่านทางพอร์ตขนาน ยังส่งมาได้ไม่ถูกต้องเนื่องจากมีปัญหาในการเชื่อมต่อพอร์ต จึงทำให้ค่าที่คำนวณได้ผิดพลาดไป เมื่อมาพิจารณาส่วนแสดงภาพสามมิติ จะพิจารณาได้ว่าถ้าเราเปลี่ยนค่าพิกัดที่รับได้ ตำแหน่งของภาพที่ออกมาก็จะเปลี่ยนตามจึงสรุปได้ว่าโปรแกรมที่รับค่าจากพอร์ตขนานที่ใช้ Visual Basic ในการคำนวณอยู่ในเกณฑ์ที่ใช้ได้อย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ปัญหาที่พบขณะทดลอง

1. ภาษา VHDL ที่ออกแบบมีความยากพอสมควร ซึ่งผู้ออกแบบต้องใช้เวลาในการศึกษาการเขียน

2. โปรแกรมที่ใช้ในการออกแบบมีข้อจำกัดมาก ทำให้การออกแบบมีความยุ่งยากมากขึ้น และตัวโปรแกรมเองก็ไม่ลองรับคำสั่งการทำงานบ้างคำสั่ง

3. ลักษณะของคลื่นที่ส่งมามีข้อผิดพลาดอยู่มากทำให้การคำนวณหาตำแหน่งเกิดความผิดพลาด

4. ชิฟ FPGA เบอร์ FLEX10K20CT144-4 เป็นชิฟชนิด SRAM ทำให้เวลาเก็บข้อมูลต้องมีไฟฟ้าเลี้ยงเสมอ

7.3 แนวทางการแก้ปัญหา

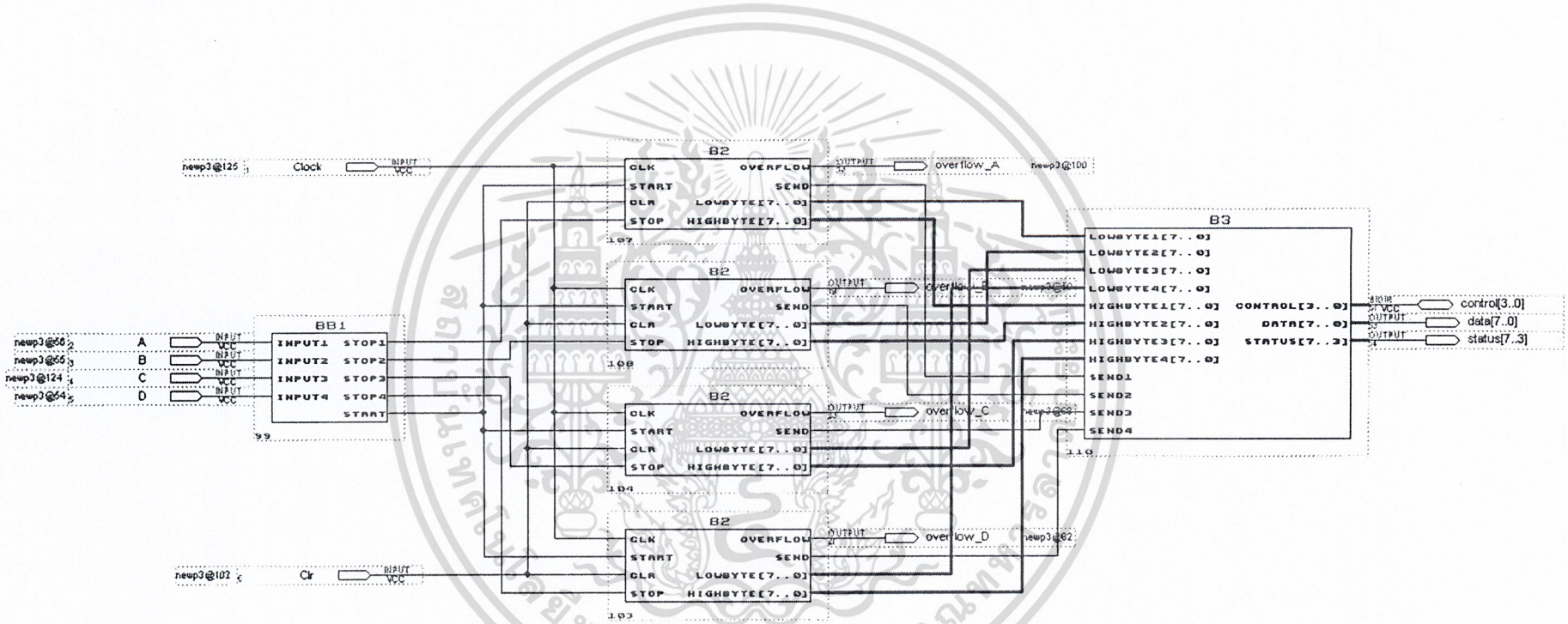
1. ผู้ออกแบบต้องมีความเข้าใจในตัวโครงสร้างภาษา VHDL มาก จึงจะอำนวยความสะดวกในการออกแบบได้ดี

2. การออกแบบที่มีประสิทธิภาพต้องออกแบบให้ประหยัดเนื้อที่เกตของFPGA มากที่สุด

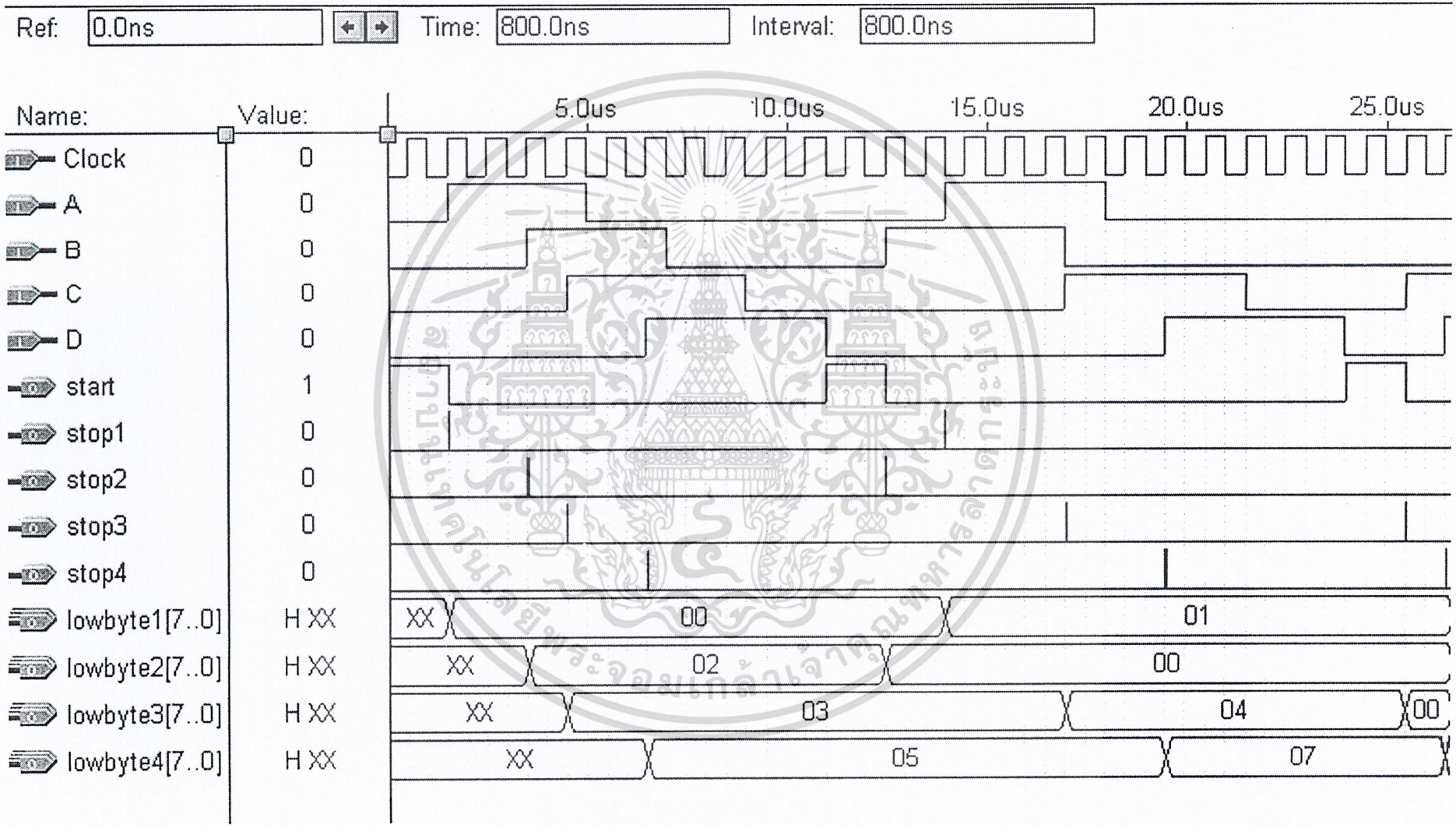
3. ควรเลือกใช้โปรแกรมที่เกี่ยวกับการเขียนภาษาVHDL หลายๆโปรแกรมเพื่อเปรียบเทียบความสามารถในการออกแบบ

4. การต่อวงจรภาคส่ง และ ภาครับ ควรทำให้มีNoise น้อยที่สุดเพื่อความถูกต้องของสัญญาณ

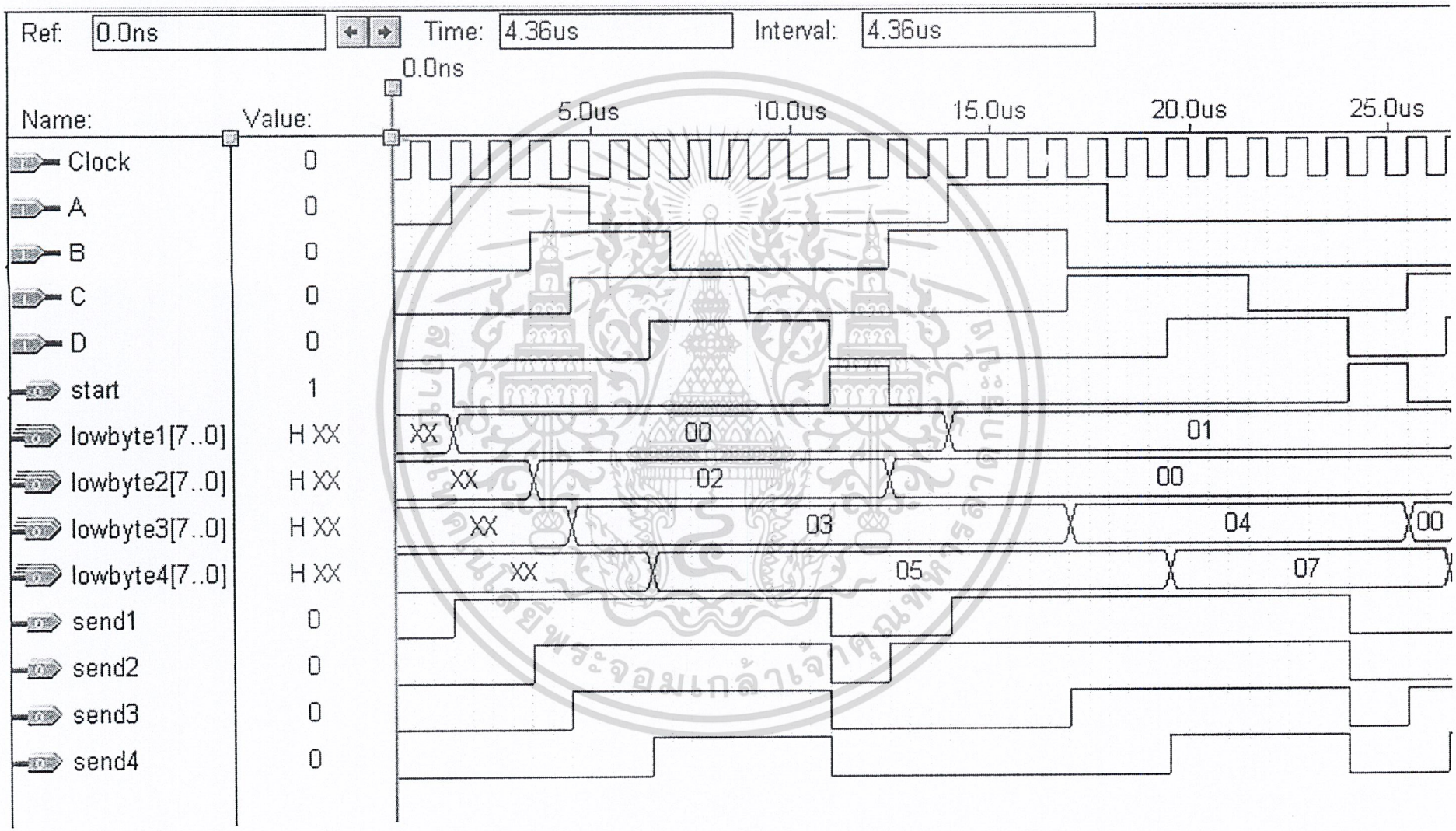
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



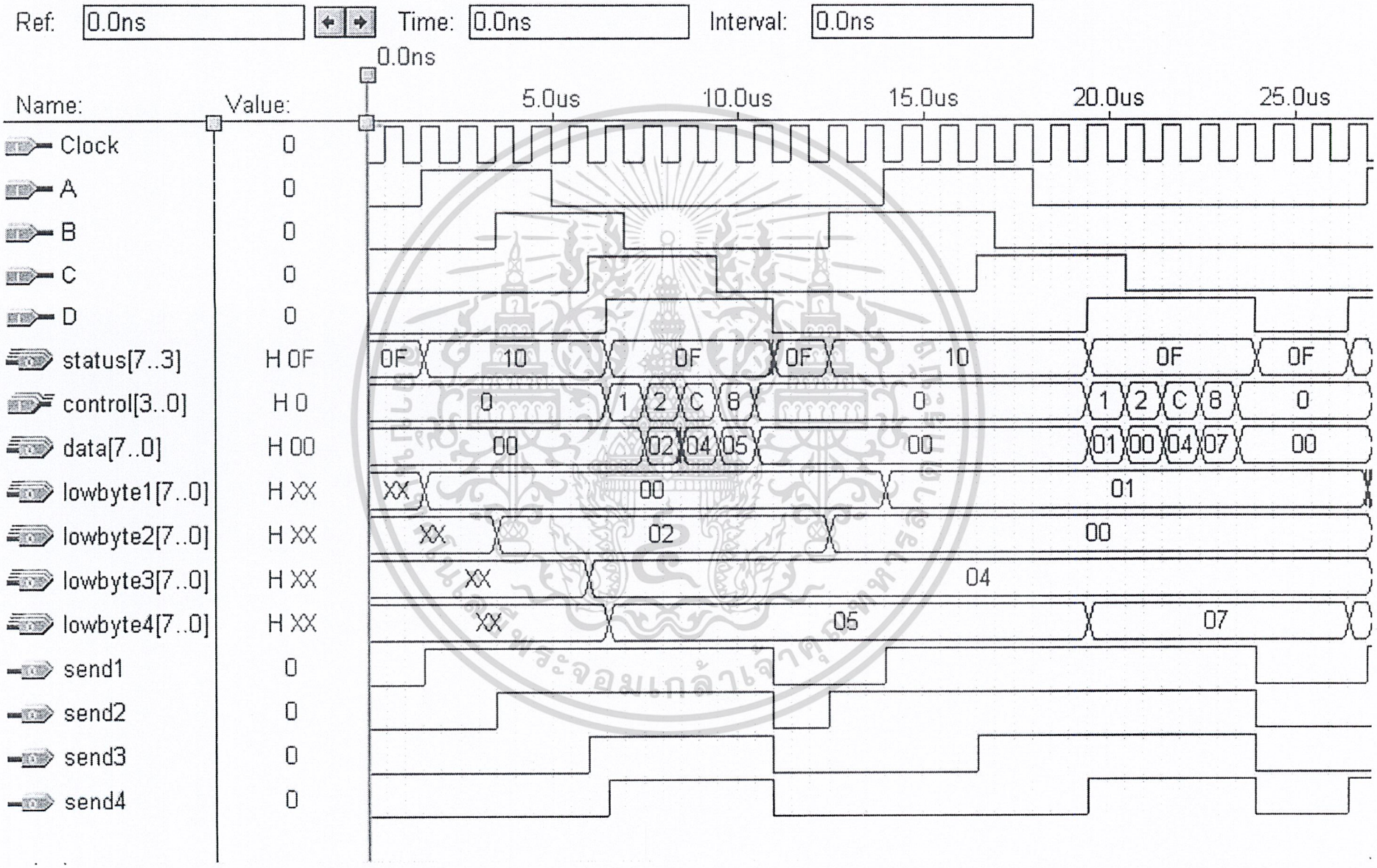
แผนภาพการทำงานของส่วน FPGA



Timing Diagram ของสัญญาณเมื่อผ่านBlock Start



Timing Diagram ของสัญญาณเมื่อผ่านBlock Counter



Timing Diagram ของสัญญาณเมื่อผ่านBlock Control

```
entity bb1 is
port (  input1 : in bit;
                input2 : in bit;
                input3 : in bit;
                input4 : in bit;
                output1: out bit;
                output2: out bit;
                output3: out bit;
                output4: out bit;
                start  : out bit);
end bb1 ;

architecture ff of bb1 is
begin
        process(input1,input2,input3,input4)

                variable sp1,sp2,sp3,sp4,np1,np2,np3,np4 : integer range 0 to 2;

         begin
                if (input1'EVENT and input1 ='1')  then np1 := np1+1; end if;
                if (input2'EVENT and input2 ='1')  then np2 := np2+1; end if;
                if (input3'EVENT and input3 ='1')  then np3 := np3+1; end if;
                if (input4'EVENT and input4 ='1')  then np4 := np4+1; end if;

                if np1=1 then np1:=0; output1<='1'; else output1<='0'; end if;
                if np2=1 then np2:=0; output2<='1'; else output2<='0'; end if;
                if np3=1 then np3:=0; output3<='1'; else output3<='0'; end if;
                if np4=1 then np4:=0; output4<='1'; else output4<='0'; end if;


                if (input1='0') and (input2='0') and (input3='0') and (input4='0') then
start<='1';
                else start<='0'; end if;
        end process;
end ff;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity b2 is
port (   clk       : in  std_logic;
                start     : in  std_logic;
                clr       : in  std_logic;
          input           : in  std_logic;
                overflow : out bit;
                send     : out bit;
                Lowbyte          : out std_logic_vector(7 downto 0);
                Highbyte : out std_logic_vector(7 downto 0));
end b2;

architecture RTL of b2 is
begin
        process(clk,input,start)
        variable     counterLow,counterHigh,ss : integer range 0 to 256 ;

begin
        if (clk'EVENT and clk ='1') then
                if (input='0' and start='0') then counterLow:=counterLow + 1;        end
if;
                if counterLow = 256  then counterLow:= 0;
                                                                counterHigh:=
counterHigh +1; end if;
           if counterHigh > 255 then overflow <='1'; end if;
         end if;

                if (input='1' and start='0') then
        Lowbyte <= conv_std_logic_vector(counterLow,8);
           Highbyte <= conv_std_logic_vector(counterHigh,8);
                ss :=1; end if;

                if (start='1') then counterLow :=0; counterHigh:=0; ss:=0; end if;
         if ss=1 then send <= '1'; else send <='0';       end if;
        end process;
end RTL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity b3 is
port (   Lowbyte1       : in std_logic_vector(7 downto 0);
                Lowbyte2       : in std_logic_vector(7 downto 0);
                Lowbyte3       : in std_logic_vector(7 downto 0);
                Lowbyte4       : in std_logic_vector(7 downto 0);
                Highbyte1      : in std_logic_vector(7 downto 0);
                Highbyte2      : in std_logic_vector(7 downto 0);
                Highbyte3      : in std_logic_vector(7 downto 0);
                Highbyte4      : in std_logic_vector(7 downto 0);
                get_send1   : in bit;
                get_send2   : in bit;
                get_send3   : in bit;
                get_send4   : in bit;
           control        : inout std_logic_vector(3 downto 0);
           Data                 : out    std_logic_vector(7 downto 0);
           status               : out    std_logic_vector(7 downto 3) );

end b3;
architecture aaa of b3 is
 signal m1,m2,m3,m4 : std_logic_vector(7 downto 0);
 signal n1,n2,n3,n4 : std_logic_vector(7 downto 0);
begin
        process(Lowbyte1,Lowbyte2,Lowbyte3,Lowbyte4,Highbyte1,Highbyte2,

        Highbyte3,Highbyte4,get_send1,get_send2,get_send3,get_send4)
                variable y1,y2,y3,y4,x : integer range 0 to 4;
        begin
                control <= "ZZZZ";
        if (get_send1='1') then y1:=1; m1 <= lowbyte1;
                                                                n1 <= highbyte1;
        else y1:=0; end if;
        if (get_send2='1') then y2:=1; m2 <= lowbyte2;
                                                                n2 <= highbyte2;
        else y2:=0; end if;
        if (get_send3='1') then y3:=1; m3 <= lowbyte3;
                                                                n3 <= highbyte3;
        else y3:=0; end if;
        if (get_send4='1') then y4:=1; m4 <= lowbyte4;
                                                                n4 <= highbyte4;
        else y4:=0; end if;

                x:=y1+y2+y3+y4;
        case x is
                when 1 to 3 => status <="10000";
                when others => status <="01111";
        end case;
        case control is
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                when "0001" => Data <= m1;
                when "0101" => Data <= n1;

                when "0010" => Data <= m2;
                when "1010" => Data <= n2;

                when "1100" => Data <= m3;
                when "0111" => Data <= n3;

                when "1000" => Data <= m4;
                when "0011" => Data <= n4;

                when others => Data <= "00000000";
        end case;
end process;
end aaa;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As
Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal
hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As
Long) As Long
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As
Long
Dim bClick As Integer
Dim c As POINTAPI

Private Sub Command2_Click()
Pic.Refresh
End Sub

Private Sub CmdLoad_Click()
Dim TempX As Double
Dim TempY As Double
Dim Lista() As String
ReDim Lista(0)
'Pic.Cls
Lst.Clear
AzzeraVar
ReadFile TxtFile, True, Lista
TraslaX Pic.ScaleWidth / 2
Traslay Pic.ScaleHeight / 2
Dim I As Integer
For I = 1 To UBound(Lista)
Lst.AddItem Lista(I)
Next
Render tHdc
BitBlt Pic.hdc, 0, 0, Pic.ScaleWidth, Pic.ScaleHeight, tHdc, 0, 0, vbSrcCopy
Pic.SetFocus

End Sub

Private Sub Start_Click()
Start.Enabled = False
Timer1.Enabled = True
Stop1.Enabled = True
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
GetKey Pic, KeyCode
End Sub

Private Sub Form_Load()
CreaBit Pic
TxtFile.AddItem App.Path + "\pyramid.3ds"
```

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TxtFile.AddItem App.Path + "\cube.3ds"
TxtFile.AddItem App.Path + "\toydog.3ds"
bDone = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
DeleteAll
End
End Sub

Private Sub Pic_KeyDown(KeyCode As Integer, Shift As Integer)
GetKey Pic, KeyCode
End Sub

Private Sub Pic_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
bClick = True
c.X = X
c.Y = Y
End Sub

Private Sub Pic_Paint()
'Render Pic.hdc
End Sub

Private Sub Stop1_Click()
Start.Enabled = True
Timer1.Enabled = False
Stop1.Enabled = False
End Sub

Private Sub Timer1_Timer()
   Dim s As Double
   Dim bL1, bL2, bL3, bL4 As Integer
   Dim bH1, bH2, bH3, bH4 As Integer
   Dim exc As Double
   Dim n0, n1, n2, n3 As Double
   Dim r0, r1, r2, r3 As Double
   Dim c1, c2, c3 As Double
   Dim d1, d2, d3 As Double
   Dim I As Double
   Dim u As Double
   Dim v As Double
   Dim w As Double
   Dim X As Double
   Dim Y As Double
   Dim Z As Double
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   s = Inp(&H379) And &H8
If s = 8 Then

 Out &H37A, &HA
 bL1 = Inp(&H378)
 Out &H37A, &HE
 bH1 = Inp(&H378)

 Out &H37A, &H9
 bL2 = Inp(&H378)
 Out &H37A, &H1
 bH2 = Inp(&H378)

 Out &H37A, &H7
 bL3 = Inp(&H378)
 Out &H37A, &HC
 bH3 = Inp(&H378)

 Out &H37A, &H3
 bL4 = Inp(&H378)
 Out &H37A, &H8
 bH4 = Inp(&H378)

 Out &H37A, &H4
 Out &H37A, &HB

Text1.Text = bH1
exc = Convert(Text1.Text)
n0 = exc + bL1
Text1.Text = n0

Text2.Text = bH2
exc = Convert(Text2.Text)
n1 = exc + bL2
Text2.Text = n1

Text3.Text = bH3
exc = Convert(Text3.Text)
n2 = exc + bL3
Text3.Text = n2

Text4.Text = bH4
exc = Convert(Text4.Text)
n3 = exc + bL4
Text4.Text = n3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If (n0 - n1 - n2 + n3) <> 0 Then
```

```
        r0 = (340 * n0) / 251250
        r1 = (340 * n1) / 251250
        r2 = (340 * n2) / 251250
        r3 = (340 * n3) / 251250

        c1 = 2 * (r0 - r1)
        c2 = 2 * (r0 - r2)
        c3 = 2 * (r0 - r3)

        d1 = (r1 ^ 2) - (r0 ^ 2) - 625
        d2 = (r2 ^ 2) - (r0 ^ 2) - 625
        d3 = (r3 ^ 2) - (r0 ^ 2) - 1250

        u = 2500 * (c1 + c2 - c3)

        X = 50 * ((c2 * d1) + (c3 * d2) - (c2 * d3) - (c1 * d2)) / u
        Text5.Text = X

        Y = 50 * ((c1 * d2) + (c3 * d1) - (c2 * d1) - (c1 * d3)) / u
        Text6.Text = Y

        v = 2500 * (d1 + d2 - d3) / u

        w = (v ^ 2) - (X ^ 2) - (Y ^ 2)
            If w > 0 Then
              Z = Sqr(w)
              Text7.Text = Z
            Else
              Text7.Text = " "
            End If
    Else:   Text5.Text = " "
          Text6.Text = " "
          Text7.Text = " "
    End If
Else
End If

If X > 0 Then
 If Y > 0 Then
  If Z > 0 Then
    GotoX X
    GotoY Y
    GotoZ Z
  End If
 End If
End If
Render tHdc
BitBlt Pic.hdc, 0, 0, Pic.ScaleWidth, Pic.ScaleHeight, tHdc, 0, 0, vbSrcCopy
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Function Convert(aaa As Double) As Double
 Dim a, b, c, d As Double
 Dim Z(1 To 8) As Double
 b = aaa Mod 128
c = b / 64
d = Int(c)
Z(2) = d * 16384
a = b Mod 64
b = a / 32
c = Int(b)
Z(3) = c * 8192
b = a Mod 32
c = b / 16
d = Int(c)
Z(4) = d * 4096
a = b Mod 16
c = a / 8
d = Int(c)
Z(5) = d * 2048
b = a Mod 8
c = b / 4
d = Int(c)
Z(6) = d * 1024
a = b Mod 4
c = a / 2
d = Int(c)
Z(7) = d * 512
b = a Mod 2
d = Int(b)
Z(8) = d * 256
Convert = Z(1) + Z(2) + Z(3) + Z(4) + Z(5) + Z(6) + Z(7) + Z(8)

End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

Project นี้จะไม่สามารถสำเร็จออกมาได้ หาไม่ได้รับความร่วมมือกันทำงานของเหล่าผู้ร่วมงานทุกคน ขอขอบคุณเหล่าเพื่อนพ้องที่คอยให้คำแนะนำ ขอขอบคุณอาจารย์เทอดศักดิ์ที่ให้คำแนะนำและปรึกษาในด้านการทำงานทั้งในโปรเจ็กและสิ่งที่ควรทำในอนาคต และที่สำคัญที่สุด คือ คุณพ่อ และ คุณแม่ ที่พวกเราเคารพรัก และทำให้มีพวกเราในวันนี้ ถ้ามีอะไรผิดพลาดประการใดทางผู้จัดทำต้องขอวภัยมา ณ ที่นี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 Flash Microcontroller " , บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด กรุงเทพ
2. กฤษดา ใจเย็น และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตขนาน" , บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด กรุงเทพ
3. "เปิดโลก FPGA กับ WIZARD PLD-A01" บริษัท แอสทรอน ลอจิก รีเสริช์แอนด์ดิวิล็อปเมนต์
4. สัจจะ จรัสรุ่งรวีวร , "คู่มือการเขียนโปรแกรม และใช้งาน Visual Basic6" , สำนักพิมพ์ อินโฟเพรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้