

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

**ระบบสั่งงานหน้าจอโปรเจกเตอร์ด้วยปลายนิ้วเพื่อ
ควบคุมหน้าจอวินโดว์แทนเมาส์**

**THE PROJECTOR TOUCH SCREEN CONTROL
IN WINDOWS**



โดย

นายกิตติศักดิ์ นวลศรี

นายภูริวัจน์ รอดน้อย

นายอักษร เพ็ชรผจญ

๒พ.
๗๖๗๗
๒๕๕๐

เลขที่.....

เลข..... 82430

วัน, เดือน, ปี..... 11 ก.ค. 2551

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

๑๑๙๔๒๙๒๑
๖.....
๑.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

ระบบสั่งงานหน้าจอโปรเจกเตอร์ด้วยปลายนิ้วเพื่อ
ควบคุมหน้าจอวินโดว์แทนเมาส์
**THE PROJECTOR TOUCH SCREEN CONTROL
IN WINDOWS**



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาววิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสั่งงานหน้าจอโปรเจกเตอร์ด้วยปลายนิ้วเพื่อ
ควบคุมหน้าจอวินโดวส์แทนเมาส์
THE PROJECTOR TOUCH SCREEN CONTROL
IN WINDOWS

ผู้จัดทำ

1. นาย กิตติศักดิ์ นवलศรี รหัส 48015241
2. นาย ภูริวิจน์ รอดน้อย รหัส 48015263
3. นาย อักษร เพ็ชรผจญ รหัส 48015280

ปริญญานิพนธ์ฉบับนี้ได้รับการตรวจสอบแล้วพร้อมที่จะทำการสอบได้

(จินภัทร นันทจิวารชย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2550

ภาควิชา อิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสั่งงานหน้าจอโปรเจกเตอร์ด้วยปลายนิ้วเพื่อควบคุมหน้าจอวินโดว์แทนเมาส์

(THE PROJECTOR TOUCH SCREEN CONTROL IN WINDOWS)

ผู้จัดทำ

1. นาย กิตติศักดิ์ นวลศรี รหัส 48015241
2. นาย ภูริวัฒน์ รอดน้อย รหัส 48015263
3. นาย อักษร เพ็ชรผจญ รหัส 48015280



(ชินภัทร นันทจิวารชัย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสั่งงานหน้าจอโปรเจกเตอร์ด้วยปลายนิ้วเพื่อ ควบคุมหน้าจอวินโดว์แทนเมาส์

นาย กิตติศักดิ์ นवलศรี รหัส 48015241

นาย ภูริวัฒน์ รอดน้อย รหัส 48015263

นาย อักษร เพ็ชรผจญ รหัส 48015280

อ.ชินภัทร นันทจิวงกรชัย อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

รายงานฉบับนี้นำเสนอการหาตำแหน่งของการเคาะบนแผ่นบอร์ดในระนาบพิกัด 2 มิติ โดยอาศัยผลต่างของเวลาจากตำแหน่งที่เคาะเดินทางมายังตัวรับที่ตำแหน่งต่าง ๆ ซึ่งเราทราบตำแหน่งพิกัดของตัวรับที่แน่นอนและอาศัยหลักการเดินทางของคลื่นเสียงบนแผ่นบอร์ด ผลต่างของเวลานี้จะใช้ FPGA เป็นตัววัดและผลต่างของเวลาที่ได้นี้จะถูกส่งให้กับ MCS-51 เพื่อคำนวณหาพิกัด X,Y และส่งต่อเข้าสู่คอมพิวเตอร์เพื่อที่จะควบคุมหน้าจอ WINDOWS ต่อไป

THE PROJECTOR TOUCH SCREEN CONTROL IN WINDOWS

Mr. Kitissak Nualse ID.48015241

Mr. Phooriwat Rodnoi ID.48015263

Mr. Aksorn Petpachon ID.48015280

Chinnapat Nantajivakornchai Advisor

Educational Year 2007

Abstract

This report presents a system for accurately estimating a position on a two dimension board. The position is estimated by using time differents from the tapped position to receivers at various positions which is counted by using a field program gate array (FPGA). The results will be transfered to MCS-51 to calculate the position x, y and delivered to computer in order to control the windows screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โปรเจกต์เรื่องนี้เป็นระบบสั่งงานหน้าจอ โปรเจ็กเตอร์ด้วยปลายนิ้วเพื่อควบคุมหน้าจอวินโดว แท็บเล็ต คงไม่อาจสำเร็จได้ หากไม่ได้รับการช่วยเหลือจากอาจารย์ที่ปรึกษา อาจารย์ ชินภัทร นันทจิ วากรัชย์ รวมถึงอาจารย์หลายท่านที่ไม่ได้กล่าวถึง ณ ที่นี้ ที่ได้คำแนะนำตลอดจนช่วยอบรมให้ความรู้ ทางวิชาการและการประกอบอาชีพทางวิศวกรรมขอบคุณเพื่อนๆภาคอิเล็กทรอนิกส์ทุกคน ที่ได้ให้ความช่วยเหลือทางด้านแนวความคิดและการดำเนินความสะดวกในการทำโปรเจกต์ ทั้งทางด้านอุปกรณ์ อิเล็กทรอนิกส์ เครื่องคอมพิวเตอร์ หนังสือและอาหารการกินต่างๆ คณะผู้จัดทำมีความประทับใจในความอบอุ่นและความมีน้ำใจเหล่านี้ และนำความรู้สึกเหล่านี้เป็นกำลังใจในการประกอบอาชีพและการ ดำรงชีวิต ตลอดจนการถ่ายทอดให้แก่รุ่นๆรุ่นต่อไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Abstract

สารบัญ

สารบัญภาพ

สารบัญตาราง

บทที่1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ส่วนประกอบของโครงการ	2
1.4 ขอบเขตของโครงการ	4
บทที่2 ทฤษฎีที่ใช้ในการออกแบบ	5
2.1 ปรัชญาการณของคติน	5
2.2 สารเปียโซอิเล็กทริกเซรามิก	11
2.3 Field Programmable Gate Array (FPGA)	18
2.4 ไมโครคอนโทรลเลอร์ MCS-51	24
2.5 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม	35
2.6 โปรแกรม Visual Basic (VB)	42
บทที่3 การออกแบบและการทำงาน	43
3.1 แนวทางการออกแบบ	43
3.2 ภาครับสัญญาณจากการเคาะ	45
3.3 หลักการทำงานของส่วน FPGA	46
3.4 วงจรภาคประมวลผล	50
3.5 หลักการทำงานของ MCS-51	54
3.6 หลักการทำงานของ Visual Basic	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง

4.1 ส่วนของตัวรับ	58
4.2 ส่วนของวงจรภาครับ	75
4.3 แสดงผลการ Simulate ของ FPGA	78
4.4 ส่วนของ MCS-51	81
4.5 ส่วนของ Visual Basic	83

บทที่ 5 สรุปผลการทดลอง

89

กิตติกรรมประกาศ

ภาคผนวก

เอกสารอ้างอิง

สารบัญภาพ

หน้า

รูปที่ 1 แนวคิดในการออกแบบ	1
รูปที่ 1.1 แสดงการวางตำแหน่งของตัวรับ	2
รูปที่ 1.2 ภาพแสดงการทำงานของส่วน FPGA	3
รูปที่ 1.3 ภาพแสดงการทำงานของส่วนประมวลผลและแสดงผล	4
รูปที่ 2.1 ลักษณะของคลื่น	5
รูปที่ 2.2 แสดงการบิดของคลื่นเสียง 2 คลื่น	7
รูปที่ 2.3 แสดงลักษณะการเกิดคลื่นนิ่ง	8
รูปที่ 2.4 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบๆ จะเสมือนเป็นแหล่งกำเนิดตัวใหม่	8
รูปที่ 2.5 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน	9
รูปที่ 2.6 แสดงเมื่อคลื่นวงกลมกระทบกับฉากที่เป็นเส้นตรงคลื่นที่สะท้อนกลับมาจะเป็นคลื่นวงกลมเช่นเดียวกัน	9
รูปที่ 2.7 แสดงสมบัติทางเปียโซอิเล็กทริกเซรามิก	13
รูปที่ 2.8 แสดงโหมดการสั่นของสารเปียโซอิเล็กทริกเซรามิก	14
รูปที่ 2.9 ผลของเปียโซอิเล็กทริก	15
รูปที่ 2.10 วงจรเสมือนของเปียโซอิเล็กทริก	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง

4.1 ส่วนของตัวรับ	58
4.2 ส่วนของวงจรภาครับ	75
4.3 แสดงผลการ Simulate ของ FPGA	78
4.4 ส่วนของ MCS-51	81
4.5 ส่วนของ Visual Basic	83

บทที่ 5 สรุปผลการทดลอง

89

กิตติกรรมประกาศ

ภาคผนวก

เอกสารอ้างอิง

สารบัญภาพ

หน้า

รูปที่ 1 แนวคิดในการออกแบบ	1
รูปที่ 1.1 แสดงการวางตำแหน่งของตัวรับ	2
รูปที่ 1.2 ภาพแสดงการทำงานของส่วน FPGA	3
รูปที่ 1.3 ภาพแสดงการทำงานของส่วนประมวลผลและแสดงผล	4
รูปที่ 2.1 ลักษณะของคลื่น	5
รูปที่ 2.2 แสดงการบิดของคลื่นเสียง 2 คลื่น	7
รูปที่ 2.3 แสดงลักษณะการเกิดคลื่นนิ่ง	8
รูปที่ 2.4 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบๆ จะเสมือนเป็นแหล่งกำเนิดตัวใหม่	8
รูปที่ 2.5 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน	9
รูปที่ 2.6 แสดงเมื่อคลื่นวงกลมกระทบกับฉากที่เป็นเส้นตรงคลื่นที่สะท้อนกลับมาจะเป็นคลื่นวงกลมเช่นเดียวกัน	9
รูปที่ 2.7 แสดงสมบัติทางเปียโซอิเล็กทริกเซรามิก	13
รูปที่ 2.8 แสดงโหมดการสั่นของสารเปียโซอิเล็กทริกเซรามิก	14
รูปที่ 2.9 ผลของเปียโซอิเล็กทริก	15
รูปที่ 2.10 วงจรเสมือนของเปียโซอิเล็กทริก	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 โครงสร้างภายใน FPGA ตระกูล MAX7000S	19
รูปที่ 2.12 โครงสร้างภายในของ FPGA ตระกูล FLEX10K	19
รูปที่ 2.13 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	25
รูปที่ 2.14 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51	26
รูปที่ 2.15 แสดงการใช้หน่วยความจำสำหรับเก็บโปรแกรม	27
รูปที่ 2.16 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก	27
รูปที่ 2.17 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51	28
รูปที่ 2.18 แสดงการจัดหน่วยความจำข้อมูล	29
รูปที่ 2.19 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี	30
รูปที่ 2.20 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม	31
รูปที่ 2.21 แสดงขาของ RS-232	36
รูปที่ 2.22 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนุกรม	37
รูปที่ 2.23 แสดงการเชื่อมต่อของสายสัญญาณ	39
รูปที่ 2.24 รูปคลื่นของสัญญาณที่ส่ง	40
รูปที่ 2.25 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232	41
รูปที่ 3.1 แสดงภาครับคลื่นจากการเคาะ	45
รูปที่ 3.2 แสดงภาพวงจรการทำงานภายใน Block Control	46
รูปที่ 3.3 แสดงภาพวงจรการทำงานภายใน Block Counter	47
รูปที่ 3.4 แสดงภาพวงจรในส่วนของ FPGA	48
รูปที่ 3.5 แสดงภาพวงจรทั้งหมดในส่วนของ FPGA	49
รูปที่ 3.6 แสดงการวางตำแหน่งของตัวรับ	50
รูปที่ 3.7 แสดงการต่อใช้งาน MCS-51	54
รูปที่ 3.8 Flow chart การทำงานของ MCS-51	55
รูปที่ 3.9 Flow chart การทำงานของ Visual Basic	56
รูปที่ 4.1 ลำโพงเล็ก	58
รูปที่ 4.2 สัญญาณที่ได้จากลำโพงเล็ก	58
รูปที่ 4.3 เปรียบไซกนาลเล็กและเปรียบไซกนาลใหญ่แบบเปลือย	59
รูปที่ 4.4 สัญญาณที่ได้ CH1. เปรียบไซกนาลเล็ก CH2. เปรียบไซกนาลใหญ่	59
รูปที่ 4.5 เปรียบไซกนาลใหญ่แบบมีฝากล่อง	60
รูปที่ 4.6 สัญญาณที่ได้ CH1. เปรียบไซแบบเปลือย CH2. เปรียบไซ แบบมีฝากล่อง ใช้นิ้วเคาะ	60
รูปที่ 4.7 เหล็กปลายแหลม เหล็กปลายหุ้มเทปกาว ปลายนิ้ว	61
รูปที่ 4.8 สัญญาณจาก CH1. เหล็กปลายหุ้มเทปกาว CH2. เหล็กปลายแหลม CH3. ปลายนิ้ว	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 สัญญาณจาก CH1. แผ่นกระจก CH2. แผ่นพีวเจอร์บอร์ด	62
รูปที่ 4.10 สัญญาณจาก CH1.แผ่นกระจก CH2.แผ่นไม้ (โต๊ะ)	62
รูปที่ 4.11 ลักษณะการเคาะ	63
รูปที่ 4.12 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 90 องศา	63
รูปที่ 4.13 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 180 องศา	64
รูปที่ 4.14 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 270 องศา	64
รูปที่ 4.15 ลักษณะระยะทางที่ใช้เคาะ	65
รูปที่ 4.16 สัญญาณที่ระยะ 0 กับ 1 cm(เปียโซขนาดเล็กแบบเปลือย)	65
รูปที่ 4.17 สัญญาณที่ระยะ 0 กับ 2 cm	66
รูปที่ 4.18 สัญญาณที่ระยะ 0 กับ 3 cm	66
รูปที่ 4.19 สัญญาณที่ระยะ 0 กับ 4 cm	67
รูปที่ 4.20 สัญญาณที่ระยะ 0 กับ 5 cm	67
รูปที่ 4.21 สัญญาณที่ระยะ 0 กับ 6 cm	68
รูปที่ 4.22 สัญญาณที่ระยะ 0 กับ 7 cm	68
รูปที่ 4.23 สัญญาณที่ระยะ 0 กับ 8 cm	69
รูปที่ 4.24 สัญญาณที่ระยะ 0 กับ 9 cm	69
รูปที่ 4.25 สัญญาณที่ระยะ 0 กับ 10 cm	70
รูปที่ 4.26 สัญญาณที่ระยะ 0 กับ 2 cm(เปียโซขนาดใหญ่แบบมีฝา)	71
รูปที่ 4.27 สัญญาณที่ระยะ 0 กับ 4 cm	71
รูปที่ 4.28 สัญญาณที่ระยะ 0 กับ 6 cm	72
รูปที่ 4.29 สัญญาณที่ระยะ 0 กับ 8 cm	72
รูปที่ 4.30 สัญญาณที่ระยะ 0 กับ 10 cm	73
รูปที่ 4.31 สัญญาณที่ระยะ 0 กับ 12 cm	73
รูปที่ 4.32 สัญญาณที่ระยะ 0 กับ 14 cm	74
รูปที่ 4.33 สัญญาณที่ระยะ 0 กับ 16 cm	74
รูปที่ 4.34 แสดงสัญญาณอินพุต A และเอาต์พุตทั้งสองชุด B, C	75
รูปที่ 4.35 สัญญาณเอาต์พุตของวงจรถ่ายเมื่อป้อนแรงดันอินพุตที่ 50 mV	75
รูปที่ 4.36 สัญญาณเอาต์พุตของวงจรถ่ายเมื่อป้อนแรงดันอินพุตที่ 100 mV	76
รูปที่ 4.37 สัญญาณเอาต์พุตของวงจรถ่ายเมื่อป้อนแรงดันอินพุตที่ 600 mV	76
รูปที่ 4.38 แสดงสัญญาณที่วัดได้จาก Logic Analyzer	77
รูปที่ 4.39 แสดง Timing Diagram ของ Block Control ที่ได้จากการ Simulate	78
รูปที่ 4.40 แสดง Timing Diagram ของ Block Counter ที่ได้จากการ Simulate	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.41 แสดง Timing Diagram ของวงจรทั้งหมดของ FPGA	80
รูปที่ 4.42 แสดงการทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8 บิต เข้าสู่คอมพิวเตอร์ โดยผ่าน MCS-51	81
รูปที่ 4.43 แสดงผลการทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8 บิต เข้าสู่คอมพิวเตอร์ โดยผ่าน MCS-51	82
รูปที่ 4.44 แสดงการป้อนค่า X,Y เพื่อคลิกซ้าย	83
รูปที่ 4.45 ผลการป้อนค่า X,Y เพื่อคลิกซ้าย	83
รูปที่ 4.46 แสดงการป้อนค่า X,Y เพื่อคลิกขวา	84
รูปที่ 4.47 ผลการป้อนค่า X,Y เพื่อคลิกขวา	84
รูปที่ 4.48 แสดงการป้อนค่า X,Y เพื่อคลิกดับเบิ้ลคลิก	85
รูปที่ 4.49 ผลการป้อนค่า X,Y เพื่อคลิกดับเบิ้ลคลิก	85
รูปที่ 4.50 แสดงการป้อนค่า N เพื่อคลิกซ้าย	86
รูปที่ 4.51 ผลการป้อนค่า N เพื่อคลิกซ้าย	86
รูปที่ 4.52 แสดงการป้อนค่า N เพื่อคลิกขวา	87
รูปที่ 4.53 ผลการป้อนค่า N เพื่อคลิกขวา	87
รูปที่ 4.54 แสดงการป้อนค่า N เพื่อคลิกดับเบิ้ลคลิก	88
รูปที่ 4.55 ผลการป้อนค่า N เพื่อคลิกดับเบิ้ลคลิก	88
รูปที่ 4.56 แผนภูมิแสดงค่าผิดพลาด(cm) ที่ได้จากการเคาะที่จุดต่าง ๆ	89
สารบัญตาราง	หน้า
ตารางที่ 2.1 ความเร็วเสียงที่เคลื่อนที่ในวัสดุต่าง ๆ	10
ตารางที่ 2.2 แสดงถึงหน้าที่ของขาต่าง ๆ	36
ตารางที่ 4.1 แสดงค่าผิดพลาด(cm) ที่ได้จากการเคาะที่จุดต่าง ๆ	89

บทที่ 1

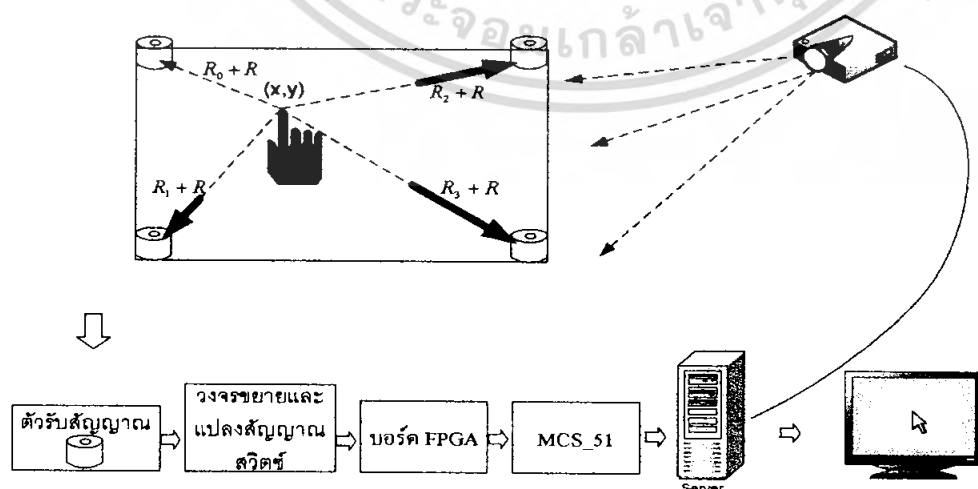
บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันเทคโนโลยีการใช้สื่อการเรียนการสอนหรือการนำเสนอผลงานในสถาบันการศึกษาหรือบริษัทต่างๆ ได้พัฒนาไปอย่างรวดเร็ว โดยสื่อที่ใช้งานง่ายและสะดวกกำลังเป็นที่นิยม คือ การใช้เครื่องโปรเจกเตอร์ฉายลงบนบอร์ดเพื่อนำเสนอข้อมูลจากเครื่องคอมพิวเตอร์ โดยข้อมูลที่ปรากฏลงบนบอร์ดนั้น จะมีหน้าตาเหมือนกับหน้าจอคอมพิวเตอร์ทุกประการ แต่มีขนาดใหญ่กว่าดังนั้นเมื่อผู้นำเสนอขึ้นที่บริเวณหน้าบอร์ดก็สามารถนำเสนอผลงานได้ แต่หากต้องการเสนอข้อมูลถัดไปหรือเปลี่ยนแปลงข้อมูลก็ต้องกลับไปเปลี่ยนแปลงที่เครื่องคอมพิวเตอร์ เพื่อจัดการกับปัญหาดังกล่าวเราจึงได้เพิ่มความสะดวกสร้างโปรเจกชันนี้ขึ้นมา โดยให้เราสามารถควบคุมหน้าจอวินโดว์ของเครื่องคอมพิวเตอร์ได้ โดยการติดต่อกับบอร์ดที่นำเสนออยู่ด้วยการใช้ปลายนิ้วเคาะลงไปที่ตำแหน่งต่างๆที่ต้องการลงบนบอร์ดที่ตรงกับตำแหน่งที่อ้างอิง โดยคอมพิวเตอร์เพื่อสั่งงานหน้าจอวินโดว์แทนการใช้เมาส์ นอกจากเพิ่มความสามารถในการติดต่อกับคอมพิวเตอร์ โดย การเคาะแล้ว ยังสามารถประยุกต์ใช้กับงานในด้านต่างๆที่เกี่ยวข้องกับการเคลื่อนที่ของคลื่นจากการสั่นสะเทือนได้อีกมากมาย เช่น การตรวจจับพิกัดจากการเดินของผู้บุกรุกหรือ การหาตำแหน่งการยิงปืนใหญ่ในสนามรบ

1.2 วัตถุประสงค์

- เพื่อนำไปใช้ในการหาค่าตำแหน่งพิกัดจากการสั่นสะเทือนโดยการเคาะในระนาบ 2 มิติ
- เพื่อศึกษาและเรียนรู้การทำงานของ FPGA, MCS-51 และ โปรแกรมต่างๆ



รูปที่ 1 แนวคิดในการออกแบบ

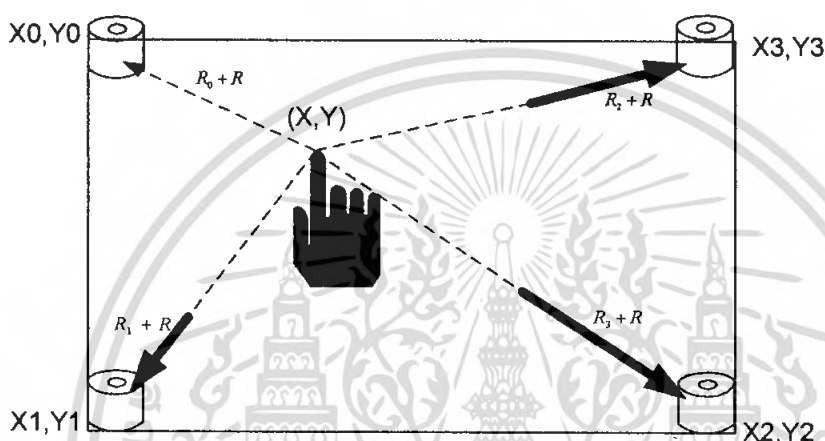
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ส่วนประกอบของโครงการงาน

ในโครงการงานนี้จะประกอบไปด้วย 3 ส่วนหลักๆ คือ

1.3.1 ชุดรับสัญญาณจากการเคาะ

ซึ่งประกอบไปด้วยเปียโซอิเล็กทริกจำนวน 4 ตัว

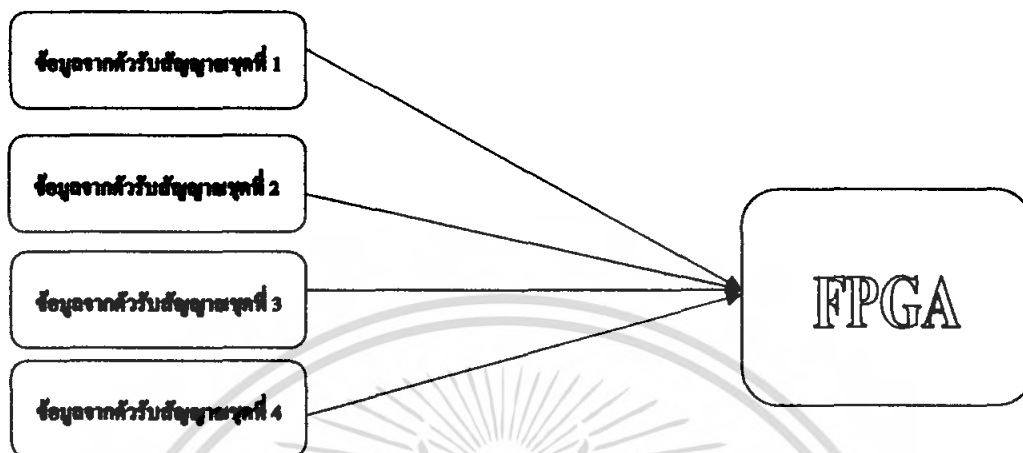


รูปที่ 1.1 แสดงการวางตำแหน่งของตัวรับ

ตัวรับสัญญาณจากการเคาะ

วงจรส่วนนี้ทำหน้าที่รับสัญญาณจากการเคาะที่ส่งออกมา โดยตัวรับจะแบ่งออกเป็น 4 จุด โดยวางตัวเป็นรูปสี่เหลี่ยมที่ตำแหน่งทราบค่าพิกัดที่แน่นอน ตัวรับแต่ละตัวจะได้รับสัญญาณที่เวลาแตกต่างกันขึ้นอยู่กับระยะทางเราจึงเอาหลักการนี้ไปเปรียบเทียบกับสัญญาณนาฬิกาที่ชุดนับสัญญาณต่อไป โดยสัญญาณที่ได้จากตัวรับซึ่งเป็นเปียโซอิเล็กทริกมีขนาดเล็กดังนั้นจึงมีวงจรขยายสัญญาณให้แรงขึ้นแล้วแปลงสัญญาณให้เป็นสัญญาณสวิตซ์แล้วจึงส่งไปยังภาค FPGA

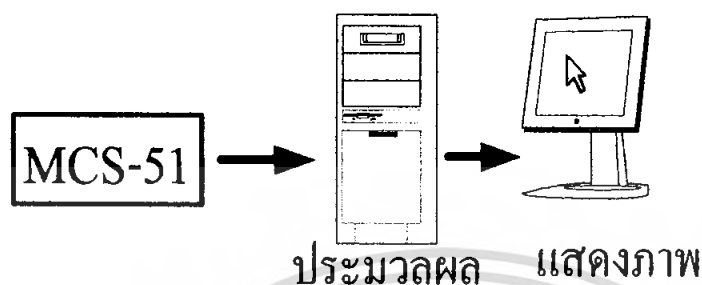
1.3.2 บอร์ดFPGA



รูปที่ 1.2 ภาพแสดงการทำงานของส่วน FPGA

FPGA สามารถรองรับวงจรดิจิทัลที่มีความสลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการออกแบบพัฒนาและทดสอบก็ทำได้ง่าย โดยได้นำบอร์ด FPGA เข้ามาเพื่อทำหน้าที่เป็นวงจร Counter นำผลต่างของเวลาที่ภาครับสัญญาณจากการเคาะมาเปรียบเทียบกับสัญญาณนาฬิกาบนบอร์ดFPGA เพื่อทำการนับ จะได้ผลข้อมูลที่เป็นจำนวน CLOCK ของตำแหน่งทั้ง 4 คิว หากตำแหน่งที่เคาะอยู่ใกล้ตัวรับตัวใดระยะเวลาที่คลื่นใช้เดินทางจะมีค่าน้อยมีผลทำให้จำนวน CLOCK ในจุดนั้นนับได้น้อยสัมพันธ์กัน จากนั้นข้อมูลของแต่ละชุดจะส่งไปประมวลผลโดยใช้ไมโครคอนโทรลเลอร์ต่อไป

1.3.3 ภาคคำนวณ ประมวลผลและส่งเข้าสู่เครื่องคอมพิวเตอร์



รูปที่ 1.3 ภาพแสดงการทำงานของส่วนประมวลผลและแสดงผล

ภาคนี้ทำงานโดยการรับค่าจำนวน Counter ที่นับได้จากบอร์ด FPGA เพื่อเข้ามาคำนวณ พิกัด X,Y โดยประกอบด้วย MCS-51 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่เราได้ทำการโปรแกรมป้อน สูตรทางคณิตศาสตร์เอาไว้แล้ว นอกจากนี้แล้ว MCS-51 ช่วยในการติดต่อกับคอมพิวเตอร์ได้ง่ายมากขึ้น โดยนำค่าพิกัดที่คำนวณได้ไปควบคุมการทำงานของเมาส์

1.4 ขอบเขตของโครงการ

1. ศึกษาถึงความไวและขนาดสัญญาณที่รับได้จากตัวรับชนิดต่างๆ
2. ศึกษาถึงวัตถุที่ใช้เคาะ
3. ศึกษาถึงตัวกลางที่ใช้เคาะ
4. ศึกษาถึงสัญญาณที่มุมตำแหน่งต่างๆ รอบตัวรับ
5. ศึกษาถึงระยะทางที่ใช้เคาะที่มีผลต่อเฟสของสัญญาณ
6. ศึกษาการทำงานในส่วนของ FPGA ในภาคต่างๆ
7. ศึกษาถึงการออกแบบของวงจรภาครับสัญญาณจากการเคาะ
8. ศึกษาการทำงานของ MCS-51
9. ศึกษาการทำงานของ Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่ใช้ในการออกแบบ

2.1 ปรัชญาการณ้ของคลื่น

2.1.1 คลื่น เกิดจากแหล่งกำเนิดคลื่นถ่ายทอดพลังงานออกไป โดยอนุภาคของตัวกลางไม่ได้เคลื่อนที่ไปด้วย เพียงแต่อนุภาคของตัวกลางมีการสั่นตามแนวสมดุค

2.1.2 ชนิดของคลื่น แบ่งได้หลายชนิดขึ้นอยู่กับว่าจะใช้เกณฑ์อะไรในการแบ่ง

1) แบ่งตามลักษณะของการสั่นของอนุภาค ได้เป็น 2 ประเภท

1.1) คลื่นตามยาว เช่น คลื่นเสียง, คลื่นอัดขยายในสปริงตามยาว

1.2) คลื่นตามขวาง เช่น คลื่นในเส้นเชือก, คลื่นน้ำ, คลื่นแม่เหล็กไฟฟ้า ฯลฯ

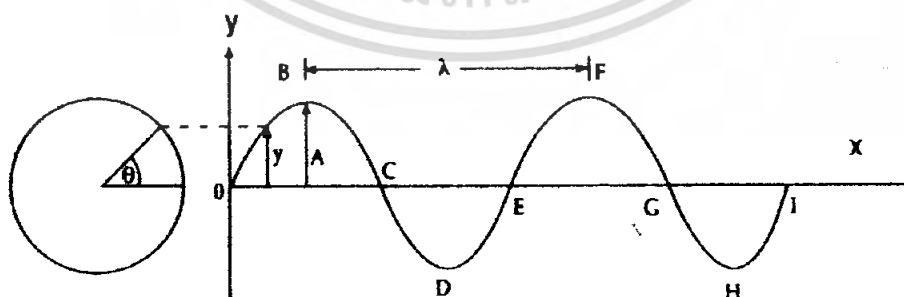
2) แบ่งตามลักษณะอาศัยตัวกลางหรือไม่ ได้ 2 ประเภท

2.1) คลื่นที่อาศัยตัวกลางในการเคลื่อนที่ หรือ เรียกว่า คลื่นกล เช่น คลื่นเสียง คลื่นน้ำ คลื่นในเส้นเชือก คลื่นของการสั่นของลวดสปริง ฯลฯ

2.2) คลื่นที่ไม่อาศัยตัวกลางในการเคลื่อนที่ หรือเรียกว่า คลื่นแม่เหล็กไฟฟ้า เช่น คลื่นแสง, คลื่นวิทยุ, ไมโครเวฟ ฯลฯ (คลื่นแม่เหล็กไฟฟ้ามีความเร็วแสง คือ 3×10^8 m/s)

3) แบ่งตามลักษณะการเกิดคลื่น 1) คลื่นคล เกิดคลื่น 2-3 ลูก 2) คลื่นเป็นช่วง คลื่นคลที่เกิดขึ้นเป็นช่วงๆ 3) คลื่นต่อเนื่อง เกิดขึ้นคลอย่างต่อเนื่องอย่างสม่ำเสมอ

2.1.3 ส่วนประกอบของคลื่น พิจารณาจากรูปที่



รูปที่ 2.1 ลักษณะของคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สันคลื่น (ยอดคลื่น) คือ ตำแหน่งที่สูงสุดของคลื่น คือ จุด B และ F

ท้องคลื่น คือ ตำแหน่งที่อยู่ต่ำที่สุด คือ จุด D และ H

การขจัด คือ ระยะจากแนวสมมูลถึงจุดบนคลื่น คือระยะขจัด y ใดๆ

แอมพลิจูด (ช่วงกว้าง) คือ การขจัดที่มากที่สุด (A)

ความยาวคลื่น คือ ระยะระหว่าง จุดสองจุดที่มีเฟสเดียวกันบนลูกคลื่นที่อยู่ติดกัน หรือระยะระหว่างสันคลื่นกับสันคลื่นที่อยู่ติดกัน (λ)

ความถี่ คือ จำนวนลูกคลื่นที่ผ่านจุดใดๆ ใน 1 หน่วยเวลา

คาบ คือ เวลาที่คลื่นหนึ่งลูกเคลื่อนที่ผ่านจุดๆ หนึ่ง (T)

เฟส คือ ตำแหน่งของจุดบนคลื่น หรือค่าของมุมที่บอกตำแหน่งของการขจัด

-เฟสตรงกัน จุดใดๆ บนคลื่นที่มีจุดต่างกัน $2n\pi$ เรเดียน เช่น 2π , 4π , 6π ,

-เฟสตรงข้าม จุดใดๆ บนคลื่นที่มีเฟสต่างกัน $(2n + 1)\pi$ เรเดียน เช่น π , 3π , 5π

-เฟสต่างกัน คือ จุดใดๆ บนคลื่นที่เฟสต่างกันใดๆ ที่ไม่ใช่เฟสตรงกันข้ามหรือตรงข้ามกัน

2.1.4 สูตรการคำนวณหาความเร็วของคลื่น

$$v = \frac{s}{t} \text{ หรือ } v = f\lambda \quad \text{----- (2.1)}$$

$$f = \frac{1}{T} \quad \text{----- (2.2)}$$

เมื่อ v = ความเร็วของคลื่น (m/s)

s = ระยะทาง (m)

t = เวลาของการเคลื่อนที่ (s)

f = ความถี่ (Hz)

T = คาบเวลา (s)

λ = ความยาวคลื่น (m)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 สมการทั่วไปของคลื่นการขจัด

$$y = A \sin \theta = A \sin \omega t \quad \text{----- (2.3)}$$

$$\text{เมื่อ } \theta = \omega t = \frac{2\pi t}{T} = \frac{2\pi x}{\lambda}$$

$$\omega = 2\pi f = \frac{2\pi}{T} \quad \text{----- (2.4)}$$

$$\text{ความต่างเฟสระหว่างจุด 2 จุด } \Delta\theta = \frac{2\pi\Delta t}{T} = \frac{2\pi\Delta x}{\lambda}$$

เมื่อ y = การขจัดใดๆ (m)

θ = มุมเฟส (เรเดียนหรือองศา)

A = แอมพลิจูด (m)

ω = ความเร็วเชิงมุม (rad / s)

t = เวลาใดๆ ที่คลื่นเคลื่อนที่ (s)

T = คาบเวลา (s / cycle)

x = ระยะทางใดๆ (m)

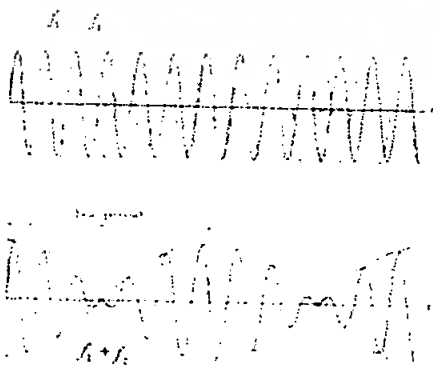
λ = ความยาวคลื่น (m)

2.1.6 คุณสมบัติโดยทั่วไปของคลื่นเสียง จะแสดงได้ 4 แบบ คือ

1) การแทรกสอดของเสียง(Interference)

การแทรกสอดของเสียงเกิดจากการรวมกันของคลื่น 2 คลื่นขึ้นไป ขณะเมื่อพบกันในตัวกลางเดียวกัน ซึ่งทำให้เกิดผลได้หลายลักษณะคือ

-การบีตส์(Beats) ของคลื่นเสียงเป็นปรากฏการณ์ที่เกิดจากการรวมคลื่นที่มีความถี่ต่างกัน หรือต่างเฟสกัน เคลื่อนที่ไปในตัวกลางเดียวกันแล้วรวมเป็นคลื่นใหม่ ซึ่งทำให้แอมพลิจูดเปลี่ยนแปลงดังแสดงในรูปที่ 2.2



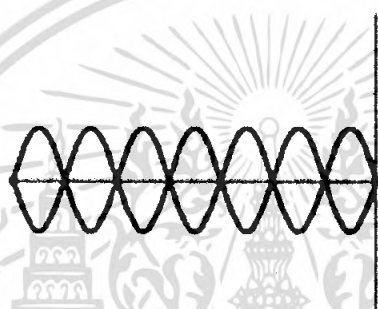
รูปที่ 2.2 แสดงการบีตของคลื่นเสียง 2 คลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 เป็นการแสดงบีตส์ของคลื่น 2 คลื่น ซึ่งคลื่นที่มีแอมพลิจูดเปลี่ยนไปซึ่งประโยชน์ของการบีตส์ของคลื่นนั้น จะนำไปใช้ในการเปรียบเทียบความถี่ของคลื่นให้แสดงผลออกมาในลักษณะของแอมพลิจูดที่แตกต่างกันนี้

-การเกิดคลื่นนิ่ง(Standing Wave) เกิดจากการแทรกสอดของคลื่นซูดที่มีแอมพลิจูดเท่ากันและความถี่เท่ากัน แต่มีทิศทางการเคลื่อนที่ตรงกันข้ามหรือเฟสตรงกันข้าม ซึ่งจะทำให้เกิดคลื่นนิ่งคังรูป เสียงจะมีความเข้มขึ้นสูงสุดที่ตำแหน่งคั่นทาง. และเบาที่สุดที่ตำแหน่งปลายทาง. ซึ่งระยะระหว่างขั้วทั้ง 2 เท่ากับ

$$d = n\lambda/2; (n=1,2,3\dots) \quad \text{----- (2.5)}$$



รูปที่ 2.3 แสดงลักษณะการเกิดคลื่นนิ่ง

2) การเลี้ยวเบนของคลื่นเสียง (Diffraction)

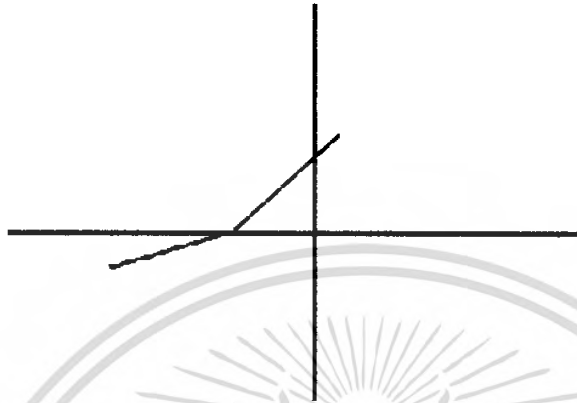
คือ คลื่นเสียงจะเลี้ยวเบนอ้อมสิ่งกีดขวางที่มีลักษณะเป็นมุมหรือช่องแคบ ซึ่งปรากฏการณ์เช่นนี้พบในชีวิตประจำวันอยู่ตลอดเวลาเช่น ในกรณีที่เราได้ยินแตรรถที่อยู่คนละมุมถนนของตึกหรือการได้ยินเสียงลอดผ่านช่องเล็กๆ จากอีกห้องหนึ่ง ดังแสดงในรูป 2.4



รูปที่ 2.4 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบๆจะเสมือนเป็นแหล่งกำเนิดตัวใหม่

3) การหักเหของคลื่น (Refraction)

คลื่นเสียงเมื่อเดินทางผ่านตัวกลางที่มีความหนาแน่นต่างกันจะเกิดการหักเหของคลื่น ซึ่งทำให้ความเร็วของคลื่นเสียงเปลี่ยนไป โดยที่ความถี่ยังคงที่อยู่



รูปที่ 2.5 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน

4) การสะท้อนของคลื่นเสียง (Reflection)

คลื่นเสียงสามารถสะท้อนได้เมื่อตกกระทบตัวกลางโดยที่มุมสะท้อนและจะทำให้เกิดเสียงก้อง ซึ่งเสียงก้องที่สะท้อนกลับมาในเวลาที่มากกว่า 50 ms จะทำให้เราได้ยินเสียงนี้เป็นครั้งที่ 2



รูปที่ 2.6 การสะท้อนของคลื่น

ตารางที่ 2.1 ความเร็วเสียงที่เคลื่อนที่ในวัสดุต่าง ๆ

Speed of Sound in Various Materials

solids		liquids	
	v (m/s)		v (m/s)
aluminum	6420	alcohol, ethyl	1207
beryllium	12,890	alcohol, methyl	1103
brass	4700	mercury	1450
brick	3650	water, distilled	1497
copper	4760	water, sea	1531
cork	500		
glass, crown	5100	gases (STP)	
glass, flint	3980		v (m/s)
glass, pyrex	5640	air, 000 °C	331
gold	3240	air, 020 °C	343
granite	5950	argon	319
iron	5950	carbon dioxide	259
lead	2160	helium	965
lucite	2680	hydrogen (H ₂)	1284
marble	3810	neon	435
rubber, butyl	1830	nitrogen	334
rubber, vulcanized	54	nitrous oxide	263
silver	3650	oxygen (O ₂)	316
steel, mild	5960	water vapor, 134 °C	494
steel, stainless	5790		
titanium	6070		
wood, ash	4670		
wood, elm	4120		
wood, maple	4110		
wood, oak	3850		

Sources: Unknown, but probably an old version of the CRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 สารเปียโซอิเล็กทริกเซรามิก

สารเปียโซอิเล็กทริกเซรามิกคือ สารเซรามิกชนิดหนึ่งที่มีสมบัติเปียโซอิเล็กทริก (piezoelectric effect) ซึ่งสมบัติเปียโซอิเล็กทริกนี้ถูกค้นพบครั้งแรกโดย Pierre Curie และ Jacques Curie เมื่อปี ค.ศ. 1880 ในการทดลองทางฟิสิกส์ของผลึกบางชนิดเช่น แร่ควอทซ์, Rochelle salt, ซิงก์เบรอน (zincblende) และทัวร์มาลีน (tourmaline) ฯลฯ แต่เนื่องจากผลึกแบบเดี่ยว (single crystal) ยังมีความเสถียรต่ำต่ออุณหภูมิ จนกระทั่งปี ค.ศ. 1940 จึงได้มีการพัฒนาสารเปียโซอิเล็กทริกเซรามิก (Piezoelectric ceramics) ขึ้น ซึ่งมีข้อดีกว่าสารเปียโซอิเล็กทริกแบบเดี่ยว คือ

- ประสิทธิภาพการเปลี่ยนรูปพลังงาน electromechanical สูง
- สามารถตกแต่งรูปทรงได้ตามต้องการ โดย machinability
- สามารถปรับคุณสมบัติด้วยการปรับปรุงส่วนประกอบของเนื้อสาร (high degree of freedom in characteristics design)
- มีเสถียรภาพสูง
- สามารถนำไปผลิตเป็นอุตสาหกรรมจำนวนมากๆ ได้

ทำให้มีการพัฒนาสารเปียโซอิเล็กทริกเซรามิกไปประยุกต์ใช้งานทางด้านอัลตราโซนิกส์ ทรานสดิวเซอร์ (ultrasonic transducer) อย่างมากมาย เช่นเซนเซอร์ตรวจหาปลา (fish finders), เครื่องทำความสะอาดด้วยคลื่นอัลตราโซนิกส์, เครื่องเชื่อมพลาสติกด้วยเครื่องอัลตราโซนิกส์ ฯลฯ สำหรับการพัฒนาสารเปียโซอิเล็กทริกเซรามิกไปประยุกต์ใช้งาน การวิเคราะห์สมบัติทางไฟฟ้าของสารนับเป็นสิ่งสำคัญเพื่อให้สาร เปียโซอิเล็กทริกเซรามิกถูกนำไปใช้งานได้เต็มที่ ประสิทธิภาพ และวิธีการอธิบายสมบัติทางไฟฟ้าของสารที่นิยม ก็คือการใช้วงจรไฟฟ้าสมมูลแบบ Butterworth Van dyke หรือ BVD ซึ่งมีสมทางไฟฟ้าที่ใกล้เคียงกับสารเปียโซอิเล็กทริกเซรามิก ในช่วงความถี่เรโซแนนซ์ โดยวงจร BVD สามารถคำนวณหาได้จากข้อมูลอิมพีแดนซ์ของสารเปียโซอิเล็กทริกในช่วงความถี่เรโซแนนซ์ (resonance frequency) นั่นเองเพื่อให้ นักวิจัยและพัฒนาสารเปียโซอิเล็กทริกเซรามิกได้เข้าใจถึงพื้นฐานของสมบัติทางไฟฟ้าของสารเปียโซอิเล็กทริกเซรามิกก่อนนำไปประยุกต์ใช้งาน งานวิจัยนี้นำเสนอวิธีการหาวงจรBVD ของสารเปียโซอิเล็กทริกเซรามิกชนิดเลดเซอร์โคเนตไททาเนต (Lead-Zirconate-Titanate) หรือ PZT ซึ่งผลิตด้วยกระบวนการทางเซรามิกแบบดั้งเดิม ลักษณะเป็นแผ่นกลม ผิวหน้าด้านบนและด้านล่างของสารถูกเคลือบด้วยโลหะเงิน มีรูปแบบการสั่นแบบ thickness mode หรือรูปแบบการสั่นในทิศทางขนานกับความหนาของสาร PZT เมื่อสารถูกกระตุ้นด้วยแรงดันไฟฟ้ากระแสสลับ

2.2.1 คุณสมบัติของสารเปียโซอิเล็กทริกเซรามิก

สารเปียโซอิเล็กทริกเซรามิกเป็นสารประกอบที่มีโครงสร้างแบบเพอร์รอฟสไกต์ (perovskite structure) จัดเรียงตัวกันเป็นผลึกแบบ polycrystalline ซึ่งต่างจากผลึกควอทซ์ที่มีลักษณะโครงสร้างเป็นผลึกเดี่ยว (single crystal) ตัวอย่างสารเปียโซอิเล็กทริกเซรามิกได้แก่ แบเรียมไททานเต (Barium Titanate), สารเลดไททานเต (Lead Titanate) และเลดเซอร์โคเนตไททานเต เป็นต้น สมบัติเปียโซอิเล็กทริก (piezoelectricity) ของสารสามารถแบ่งได้เป็น 2 ลักษณะ คือ สมบัติเปียโซอิเล็กทริกแบบทางตรง (direct effect) และสมบัติเปียโซอิเล็กทริกแบบทางอ้อม (reverse effect)

1) สมบัติเปียโซอิเล็กทริกแบบทางตรง (direct effect)

คือสมบัติของสารที่สามารถสร้างประจุไฟฟ้าออกมาที่ผิวหน้า เมื่อมีแรงกด (stress) มากระทำต่อสาร ดังรูปที่ 2.7 โดยสัญลักษณ์ P แทนเวกเตอร์ของความเป็นขั้วไฟฟ้าในเนื้อสาร (polarization) T คือแรงที่มากกระทำต่อพื้นที่หน้าตัดหรือแรงเครียด (strain) มีหน่วยเป็นนิวตันต่อตารางเมตร และ D คือประจุที่เกิดขึ้นภายในเนื้อสาร (dielectric displacement) จะได้ว่า

$$D = Q/A = dT \quad \text{----- (2.6)}$$

โดยที่

d คือค่าคงที่เปียโซอิเล็กทริก (piezoelectric constant) ของสารมีหน่วยเป็นคูลอมต่อนิวตัน (coulombs / Newton)

Q คือ ประจุไฟฟ้ามีหน่วยเป็นคูลอม

A คือ พื้นที่หน้าตัดของเนื้อสารมีหน่วยเป็นเมตร

2) สมบัติเปียโซอิเล็กทริกแบบทางอ้อม (reverse effect)

เป็นสมบัติของสารที่สามารถยืดหรือหดตัวได้เมื่ออยู่ใต้สนามไฟฟ้า ดังรูปที่ 2.8 ถ้าให้ S คือความเค้นที่เกิดภายในเนื้อสารและ E คือแรงดันไฟฟ้าที่มากกระทำต่อพื้นที่หน้าตัด สามารถเขียนสมการได้ดังนี้

$$S = dE \quad \text{----- (2.7)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสัญลักษณ์ d สามารถใช้แทนค่าคงที่เพียโซอิเล็กทริกทั้งแบบทางตรงและทางอ้อมได้ เนื่องจากเป็นอัตราส่วนของการเปลี่ยนรูปพลังงาน จากสมการ 2.1 และ 2.2 เมื่อพิจารณาถึงสมบัติของค่าความยืดหยุ่น (elastic) และค่าไดอิเล็กทริกของเนื้อสาร สามารถเขียนเป็นสมการเพียโซอิเล็กทริกคือ

$$D = dT + \epsilon^T E \quad \text{----- (2.8)}$$

$$S = s^E T + dE \quad \text{----- (2.9)}$$

โดยที่

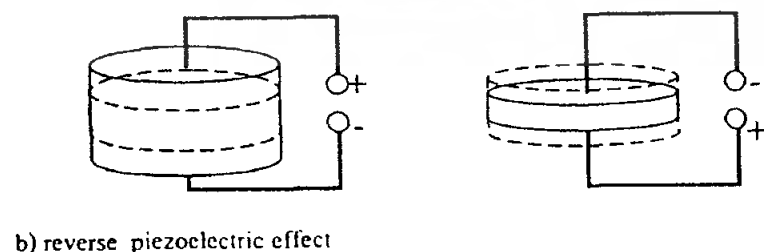
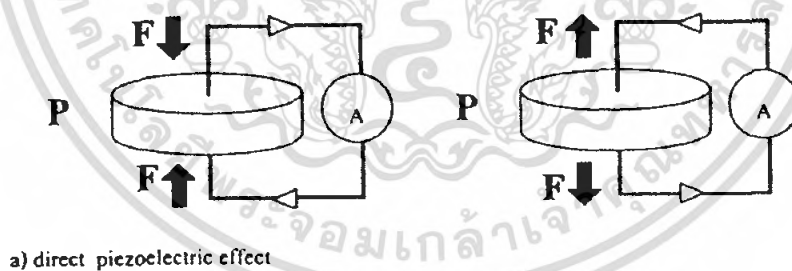
D คือค่า electric flux density หรือ electric displacement

S คือค่าความเค้น ($S = \Delta L/L$)

ϵ^T คืออัตราความจุกระแสไฟฟ้า (relative permittivity) หรือค่าคงที่ไดอิเล็กทริกของสาร (free dielectric constant) (ซึ่งบางครั้งใช้ K เป็นสัญลักษณ์) และสัญลักษณ์ด้วยกำลัง

T แทนสถานะที่ไม่มีแรงกด

s^E คือสัมประสิทธิ์ความยืดหยุ่นของสาร (elastic coefficient) และสัญลักษณ์ด้วยกำลัง E แทนสถานะที่ไม่มีสนามไฟฟ้า

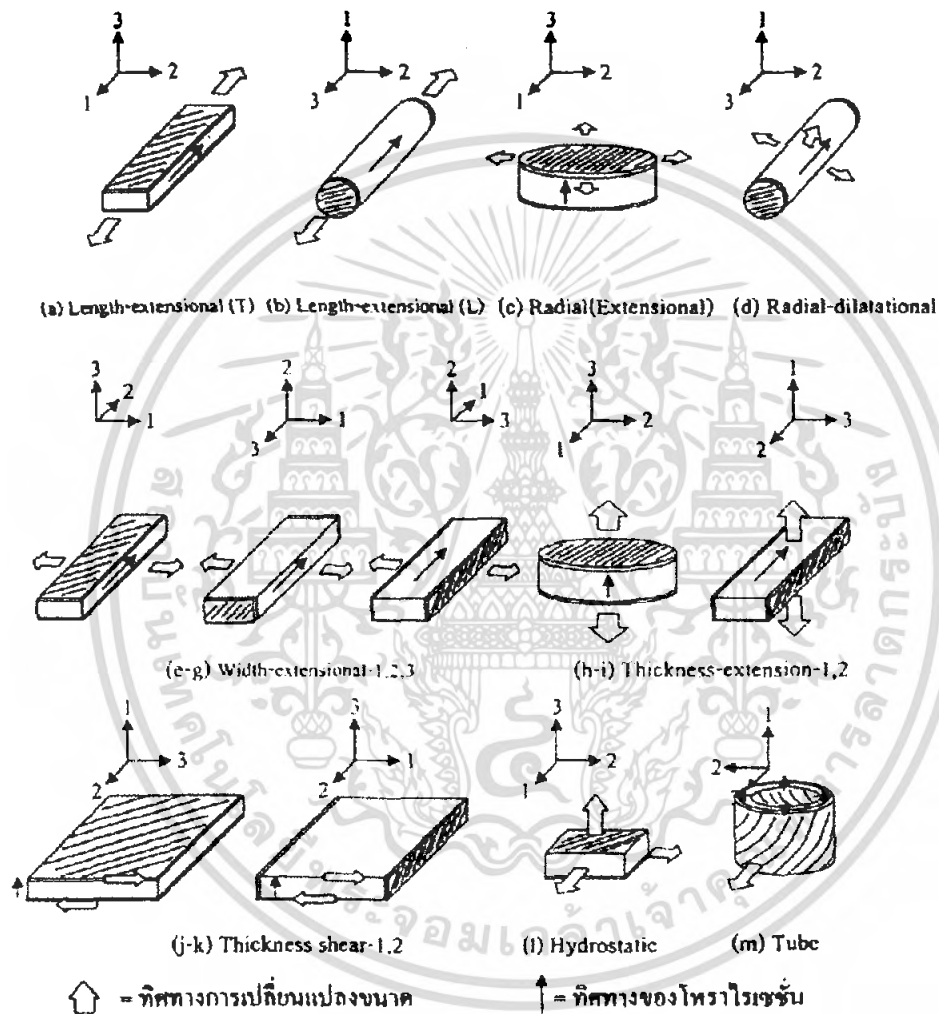


รูปที่ 2.7 แสดงสมบัติทางเพียโซอิเล็กทริกเซรามิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 โหมดการทำงานของเป็ยโซอิเล็กทริกเซรามิก

ในทิศทางการสั้นของสารเป็ยโซอิเล็กทริกเซรามิกเมื่อถูกกระตุ้นด้วยแรงดันไฟฟ้า กระแสลับมีหลายรูปแบบขึ้นกับรูปทรงและชนิดของสารเป็ยโซอิเล็กทริกเซรามิก ดังแสดงในรูปที่ 2.8



รูปที่ 2.8 แสดงโหมดการสั้นของสารเป็ยโซอิเล็กทริกเซรามิก

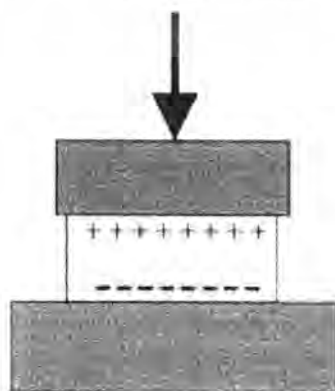
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการนำสารเปียโซอิเล็กทริกเซรามิกไปใช้งานทางด้านคลื่นอุลตราโซนิกส์แบบทางตรง ได้แก่ เครื่องจุดเตาแก๊ส, ตัวเซนเซอร์วัดแรงดัน แบบทางอ้อม ได้แก่ เครื่องเชื่อมพลาสติก ฟิล์มคลื่นอุลตราโซนิกส์, เครื่องทำความสะอาดด้วยคลื่นอุลตราโซนิกส์ นอกจากนี้สารเปียโซอิเล็กทริกยังสามารถนำไปใช้เป็นอุปกรณ์รับและส่งคลื่นอุลตราโซนิกส์ในตัวเดียวกันได้เช่น อุปกรณ์ตรวจหาปลา และ โซนาทรานสดิวเซอร์ ซึ่งสารเปียโซอิเล็กทริกเซรามิกในแต่ละตัวอย่างข้างต้นจะมีรูปทรง ชนิด และรูปแบบหรือโหมดการสั่นแตกต่างกันออกไป ตัวอย่างเช่นอุปกรณ์ทรานสดิวเซอร์ในเครื่องทำความสะอาดด้วยคลื่นอุลตราโซนิกส์ จะเป็นสารเปียโซอิเล็กทริกเซรามิกที่ทำจากสารเลดเซอร์โคเนตไททาเนต มีรูปแบบแบบการสั่นที่มีทิศทางแนวตั้งฉากกับผิวหน้าของสารหรือมีทิศทางการสั่นขนานกับความหนา เรียกว่า thickness mode ดังรูปที่ 2.8 มีลักษณะเป็นแผ่นวงกลม และมีอัตราส่วนของขนาดเส้นผ่านศูนย์กลางต่อความหนาเท่ากับ 10 หรือมากกว่า

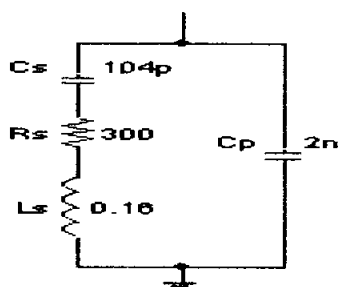
2.2.3 ปฏิกิริยาการเปียโซอิเล็กทริก

เปียโซอิเล็กทริกเป็นปรากฏการณ์ธรรมชาติอย่างหนึ่ง ซึ่งทำให้พลังงานสามารถเปลี่ยนแปลงจากรูปหนึ่งไปเป็นอีกรูปหนึ่งได้ กล่าวคือถ้าป้อนแรงกลให้แก่ Solck Crystalline Dielectric ดังในรูปที่ 2.9 ก็จะทำให้เกิดความเค้น (Stress) ภายในคริสตอลและทำให้ผลึกของคริสตอลผิดรูปไปเช่นพวกควอทซ์(Quartz) ผลก็คือประจุเปลี่ยนไป การผิดรูปร่างของแลททิซเป็นผลให้ความสัมพันธ์ระหว่างการแทนที่ (Displacement) ของประจุบวกและลบในแลททิซเปลี่ยนไป การแทนที่ของประจุภายในจะเท่ากับประจุภายนอกของขั้วที่ตรงข้ามของคริสตอล เรียกว่าผลของเปียโซอิเล็กทริก

การวัดประจุทำได้โดยการต่ออิเล็กโทรด (electrode) เข้าที่ผิวนอก แล้ววัดความต่างศักย์ระหว่างขั้วทั้งสอง ขนาด(Magnitude) และการมีขั้ว (Polarity) ของประจุบนผิวที่ถูกเหนี่ยวนำ (induce) เป็นสัดส่วน โดยตรงกับขนาดและทิศทางของแรงที่มากระทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.9 ผลของเปียโซอิเล็กทริก ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 วงจรเสมือนของเพียโซอิเล็กทริก

C_s : คาปาซิแตนซ์ของชิ้นส่วนเพียโซอิเล็กทริก

R_s : Leakage Resistance ของเพียโซอิเล็กทริก ปกติจะมีค่าสูงประมาณ $10^8 - 10^{10}$

ความต้านทานระหว่างเทอร์มินอลโดยทั่วไปจะเป็นความต้านทานทางโหลด R_L (ในช่วง $10^6 - 10^7$)

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

ที่ความถี่ปานกลางและความถี่สูงโวลต์เตจ E_0 ที่คร่อมโหลดหาได้โดย C_s และ C_L ค่าเหล่านี้มาได้โดยการแบ่งโวลต์เตจ โดยโวลต์เตจ E_0 ได้จากสมการ

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

โดยขึ้นกับความถี่ ถ้าเอาที่ทุกโวลต์เตจมีค่ามากสามารถทำให้ลดลงได้โดยการเพิ่มค่า C_L เช่นการนำค่า C_s ต่อขนานเข้าไปที่ความถี่ต่ำโวลต์เตจ E_0 คร่อมโหลดหาได้โดยค่ารีแอกแตนซ์ของ C_s และอิมพีแดนซ์ของ C_L และ R_L ที่ต่อขนานกัน โวลต์เตจ E_0 ขึ้นกับความถี่และจะลดลงถ้าความถี่ลดลง การตอบสนองต่อความถี่สามารถปรับปรุงได้โดยใช้ทรานสดิวเซอร์ที่มีค่าคาปาซิแตนซ์สูงหรือโดยการเพิ่มความต้านทานโหลด RL

2.2.4 วัสดุเพียโซอิเล็กทริก

วัสดุเพียโซอิเล็กทริกที่ใช้กันอย่างกว้างขวาง เช่น Quartz , Tourmaline sulphate, Barium Titanate และ Zirconate (TZI) โดยทั่วไปแล้วพวกควอทซ์ และคริสตอลที่เป็นเพียโซอิเล็กทริกธรรมชาติมักจะมีข้อของมันเป็นเองตามธรรมชาติ แต่พวกวัสดุเพียโซอิเล็กทริกที่สังเคราะห์ขึ้นมาเช่น แบเรียม ไททานาต เซรามิก จะต้องนำมาทำการอบคริสตอลภายในแรงดัน และวางวัสดุที่ได้นี้ในสนามไฟฟ้าที่มีแรงดันของไฟฟ้ากระแสตรงมาก หลังจากชิ้นส่วนนี้ถูกนำไปในสนามไฟฟ้าแล้วคริสตอลนี้จะมีข้อตามแนวของทิศทางของสนาม และประพืดตัวตามคุณสมบัติของเพียโซอิเล็กทริกสำหรับชิ้นส่วนที่ทำจากวัสดุสังเคราะห์นี้ ไม่มีข้อจำกัดทางขนาดโดยโครงสร้างของคริสตอลและยังสามารถทำให้มีรูปร่างและขนาดต่างๆ และทิศทางของขั้วก็จะถูกสร้างขึ้นระหว่างขั้นตอนการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.5 ชนิดของเปียโซอิเล็กทริกทรานสดิวเซอร์

เปียโซอิเล็กทริกทรานสดิวเซอร์สามารถแบ่งได้เป็น 2 ชนิดคือ

- แบบ *Generation-Action Transducer* ใช้เป็นตัวรับโดยแรงดันไฟฟ้าที่เกิดขึ้นจะหาได้จากแรงดันและความถี่ที่มากกระทำต่อวัสดุเปียโซอิเล็กทริก

- แบบ *Motor-Action Transducer* ใช้เป็นตัวส่งโดยการเปลี่ยนแปลงของรูปร่างที่ทำให้เกิดคลื่นอัลตราโซนิค โดยจะขึ้นอยู่กับขนาดความสูงและความถี่ของแรงดันไฟฟ้าที่ให้ในทั้งสองกรณี ค่าแรงดันไฟฟ้าที่เกิดขึ้นกับขนาดของวัสดุ

2.2.6 การทำงานของทรานสดิวเซอร์ตัวส่งตัวรับ

เมื่อเซรามิกได้รับสัญญาณแรงดันมาตกคร่อมจะทำให้ชั้นสารเซรามิก ไค้งงอทำให้เกิดการอัดอากาศโดยรอบเกิดเป็นคลื่นขึ้นมา ดังนั้นถ้าป้อนสัญญาณเป็นห้วงๆ (Electrically Pulse) จากการออสซิลเลทก็จะให้ชั้นสาร ไค้งงอมากขึ้นหรือทิศทางใดตามขนาดและทิศทางการเปลี่ยนแปลงขนาดของสัญญาณไฟฟ้าจากการออสซิลเลทนั้นออกไป โดยทั่วไปกำลังเอาท์พุทที่ออกมาจะตกประมาณ 10% ของกำลังไฟฟ้าที่ป้อนให้แต่เอาท์พุทจะสูงที่ค่านี้ โดยประมาณก็ต่อเมื่อความถี่ของสัญญาณออสซิลเลเตอร์ที่ป้อนเข้าชั้นสารเซรามิก ตรงกับความถี่เรโซแนนท์เป็นความถี่ทางกลตามธรรมชาติของชั้นสารเซรามิกนั้นๆ ส่วนที่ความถี่อื่นๆกำลังเอาท์พุทจะลดลงกว่านี้ ส่วนการทำงานของทรานสดิวเซอร์ตัวรับมักมีการทำงานตรงข้ามกับตัวส่ง คือเมื่อมีคลื่นเสียงที่มีความถี่ตรงกับความถี่เรโซแนนท์ของชั้นสารเซรามิกเข้ามา จะทำให้ชั้นสาร ไค้งงอตัวไปมาและเกิดสัญญาณแรงดันไฟฟ้าที่มีขนาดเล็กขึ้นคร่อมขั้วทั้งสองของตัวมัน

คุณสมบัติทั่วไปของทรานสดิวเซอร์แบบเปียโซอิเล็กทริกก็คือ มีความต้านทานไฟฟ้าตรงสูงมากอาจถึง 100 เมกกะ โอห์ม เรียกว่า ถ้าเอาอิมพีแดนซ์ธรรมดา คั้งสเกลวัดค่าความต้านทานสูงๆ เข็มจะไม่กระดิกเลยแต่ขณะที่มันทำงานความต้านทานไฟฟ้าสลับจะลดลง

82430

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 FPGA(Field Programmable Gate Array)

โครงสร้างภายในของ FPGA ประกอบด้วยแอเรย์ของลอจิกเกตต่างๆ มากมาย ซึ่งในปัจจุบันความจุเกตภายในตัวชิพ FPGA ได้เพิ่มขึ้นจากระดับไม่กี่พันตัวจนระดับล้านตัว ซึ่งสามารถรองรับวงจรดิจิทัลที่มีความสลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการออกแบบพัฒนาและทดสอบก็ทำได้ง่าย ซึ่งในปัจจุบันการออกแบบวงจรโดยใช้วิธี FPGA กำลังเป็นที่นิยมและมีแนวโน้มที่จะนำมาใช้งานมากขึ้นเรื่อย ๆ

2.3.1 FPGA แบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรมซึ่งมีอยู่ 2 แบบคือ

การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรมโดยการใช้หน่วยความจำ

1. การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

1.1 Fuse เป็นวิธีการ โปรแกรมที่สามารถทำได้เพียงครั้งเดียวซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน

1.2 Anti Fuse เป็นวิธีการ โปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรมแล้ว จะเชื่อมต่อถึงกัน

2. การโปรแกรมโดยใช้หน่วยความจำ

2.1 EEPROM Based FPGA

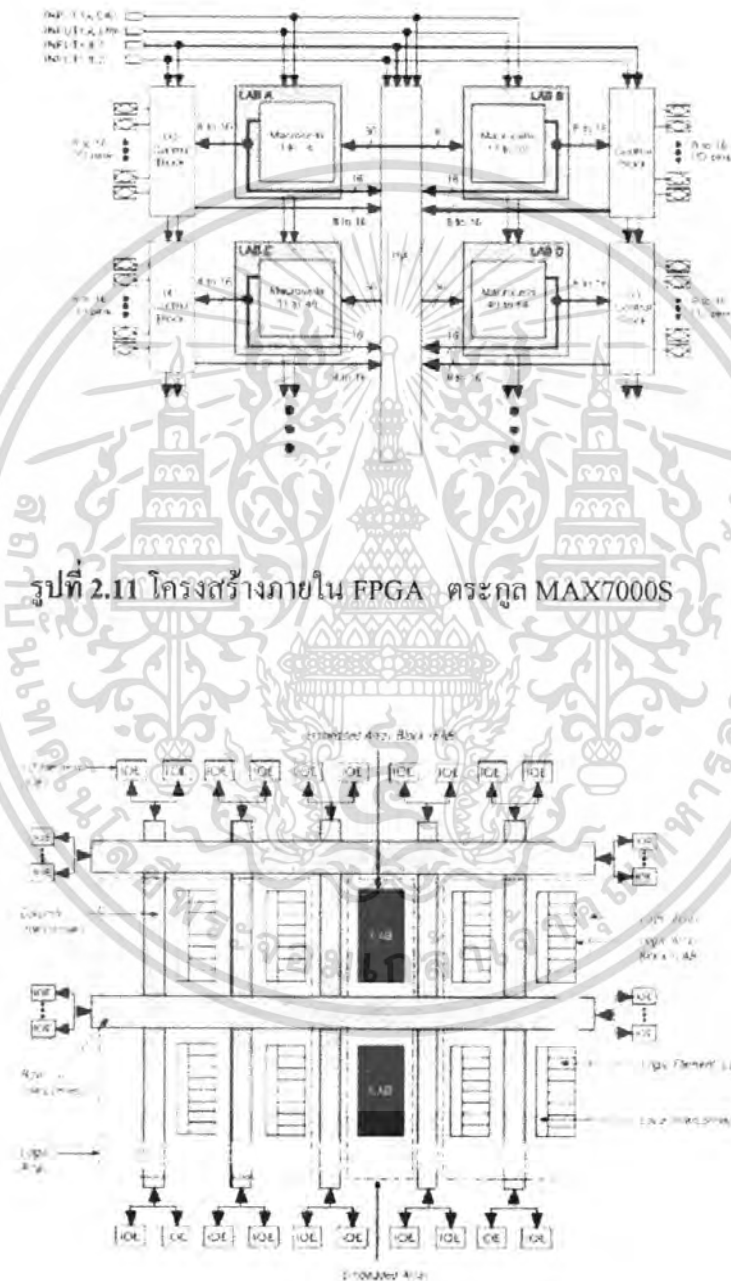
FPGA ที่ใช้การโปรแกรมแบบนี้มันเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกตต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกต แต่ละข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิต ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

2.2 SRAM Based FPGA

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

2.3.2 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอาร์เรย์ ของบล็อกลอจิกสามารถทำการโปรแกรมได้ดังรูปที่ 2.11 และ 2.12



รูปที่ 2.11 โครงสร้างภายใน FPGA ตระกูล MAX7000S

รูปที่ 2.12 โครงสร้างภายในของ FPGA ตระกูล FIEX10K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 รายละเอียดของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

1. การสังเคราะห์วงจร (Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรมของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น FPGA ของบริษัท Xilinx และบริษัท Altera จะมีซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตามเทคโนโลยี ที่เลือกใช้ ในการสังเคราะห์วงจรนั้น วงจรระดับเกต (Gate Level) จะไม่เหมาะสมกับโครงสร้างที่มีอยู่ในอุปกรณ์ FPGA นั้นๆ จึงทำให้ผลที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดลแต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอฟต์แวร์การสังเคราะห์วงจรก็จะมีผลการรายงานผลว่าโมเดลที่ออกแบบไปนั้นเป็นอย่างไร เช่นมีค่าความหน่วง (Delay) เท่าใดใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

2. การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่นๆภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่ง คือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในคอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำ โดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกต (gate), ฟลิป-ฟลอป (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGA

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

3. การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าจะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณ ได้ไม่หมด

การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆ ให้อยู่ใกล้กัน โดยเฉพาะส่วนที่มีการเชื่อมต่อสัญญาณด้วยกันนอกจากนั้นการกำหนดตำแหน่งขา 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 ขั้นตอนนี้จำทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณ ได้ไม่หมด (เนื่องจากจำนวนทรัพยากรสำหรับเชื่อมต่อสัญญาณนั้นมีอยู่จำกัด) หรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับ

ผู้ออกแบบสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟต์แวร์หรือผู้ออกแบบเชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับทางเวลา จะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

5. ความหน่วงด้านเวลา (Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้ความหน่วงน้อยที่สุดได้ สำหรับความหน่วงที่เกิดขึ้นนั้นแยกได้เป็นสองประเภทคือ

5.1.1 ความหน่วงทางลอจิก (Logic delay) เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เอง

5.1.2 ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ (Routing delay) เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA

โดยปกติแล้วค่าความหน่วงลอจิกไม่ควรเกิน 50 % ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณมันจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น และเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น

ค่าความหน่วงที่ได้หลังจากการวางอุปกรณ์ และเชื่อมต่อสัญญาณแล้วจะมีค่าความหน่วงที่ค่อนข้างแน่นอน ซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้น เป็นไปตามข้อกำหนดหรือไม่

6. การจำลองการทำงานของวงจร (Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการทำงานของโมเดลถูกต้องหรือไม่มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้ซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจรควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดลเกิดขึ้นตอนไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนี้ๆ นั่นคือการทำงานจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (function test) ยังไม่สามารถตรวจสอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตามข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือถ้ามีจะแก้ไขให้ถูกต้อง

ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์การเชื่อมต่อสัญญาณ (Post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากที่ตรวจสอบฟังก์ชันการทำงานแล้วยังต้องตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่า วงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงานของ วงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนนี้ต่างๆ

7. การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆจนกระทั่งผ่านการทำ PPR (Partitioning, Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่ดาวน์โหลด (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 เพราะมีข้อมูลวงจรเก็บข้อมูลอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยวิธี EPROM หรือ Anti fuse ก็ไม่จำเป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

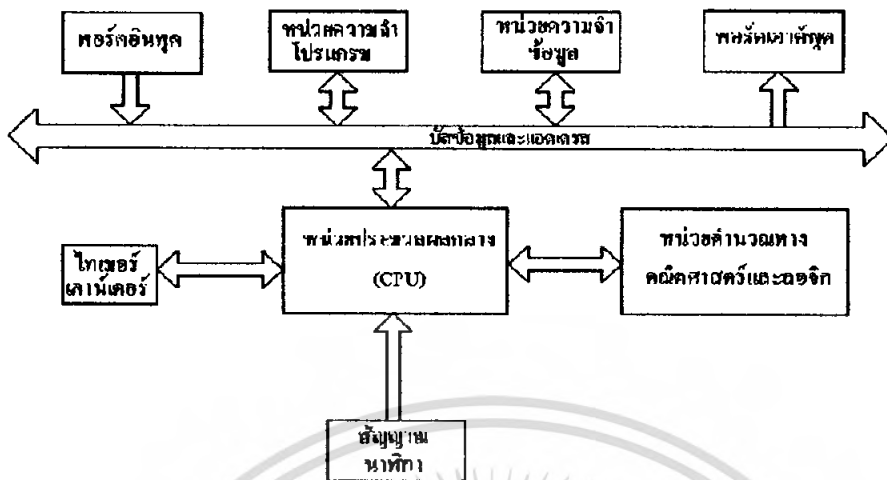
2.4 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มี อุปกรณ์สนับสนุนประกอบอยู่ภายในหลายอย่าง ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอน-โทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้น และมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากเราต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่นเพิ่มเติม เช่น ไอซี 8255 หรือหน่วยความจำภายนอกเรายังสามารถนำมาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีกด้วย

2.4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.13 ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (BOOLEAN PROCESSOR)
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ อีพรอม (เบอร์ 8451)
- หน่วยความจำแบบ แรม ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรถับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบคู่เพื่อกเต็ม(FULL DUPLEX)
- วงจรถบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับวงจรผลิตสัญญาณนาฬิกาภายในซึ่งโครงสร้างการทำงานทั้งหมดของไมโครคอนโทรลเลอร์จะ อาศัยหลักการการทำงานที่เกี่ยวข้องกัน โดยอาศัยหลักการการทำงานที่เป็นไปตามโครงสร้างเสมอ

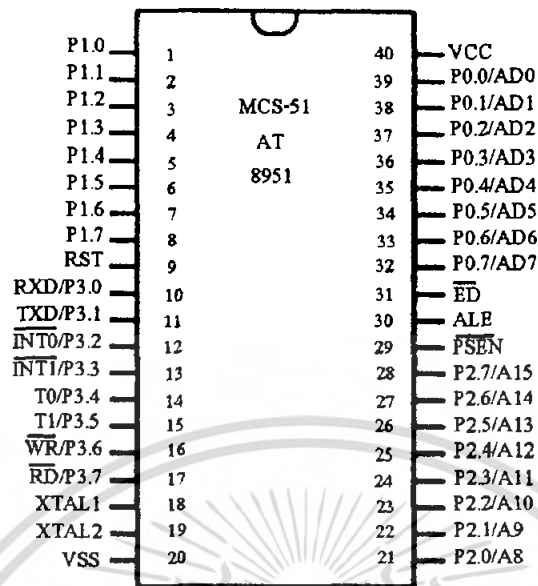


รูปที่ 2.13 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้มักจะมีรูปร่างของไอซีเป็นแบบขนาด 40 ขาดังแสดงในรูปที่ 2.14 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขาอย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า ALTERNATE FUNCTION ในรูปที่ 2.14) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่นขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุตหรืออินพุตตามปกติภายในไมโครคอนโทรลเลอร์ MCS-51 ซึ่งประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซี MCS-51 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจรควบคุมรีจิสเตอร์ (REGISTER) หรือหน่วยความจำภายในของไอซี MCS-51 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นสัญญาณที่เรียกว่าบัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ระหว่างการประมวลผลคำสั่ง หน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมาก็คือเป็นการควบคุมการรับหรือส่งข้อมูลระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการทำงานร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และลอจิก หรือเรียกว่า ARITHMETIC AND LOGIC UNIT :ALU

2.4.2 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51

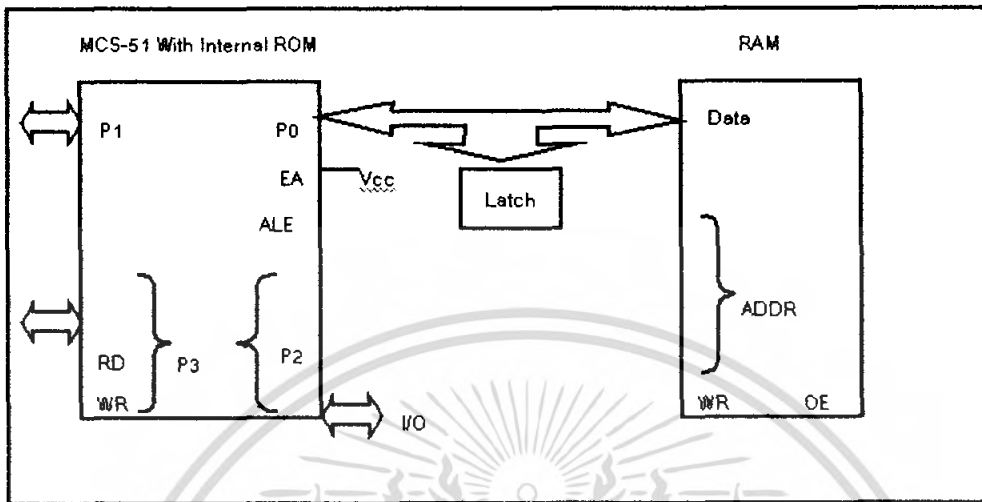
ไมโครคอนโทรลเลอร์ MCS-51 แยกการจัดการหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน คือ หน่วยความจำโปรแกรม (PROGRAM MEMORY) และหน่วยความจำข้อมูล (DATA MEMORY) หน่วยความจำทั้งสองนี้มีหน้าที่แตกต่างกัน และใช้วิธีการอ้างแอดเดรสสัญญาณการติดต่อแยกออกจากกัน



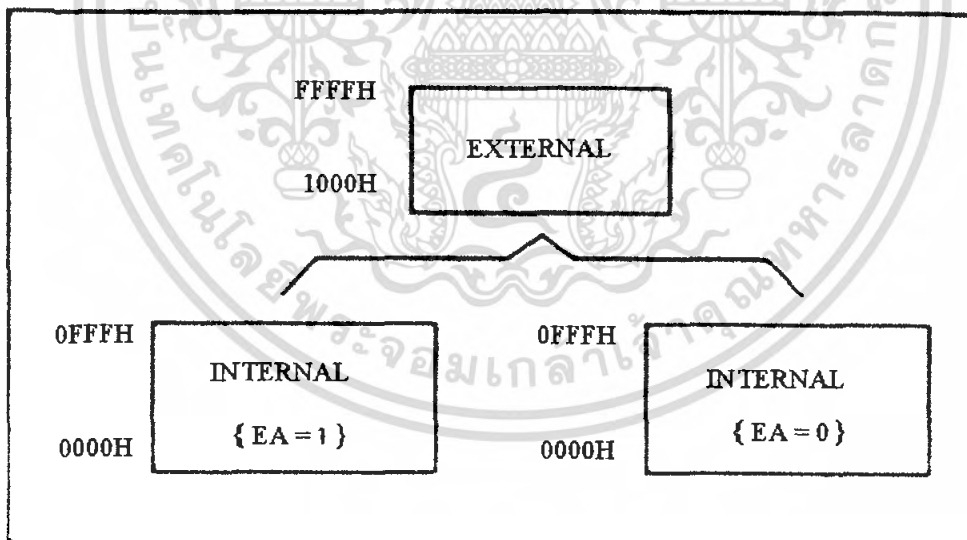
รูปที่ 2.14 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51

2.4.3 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไมโครคอนโทรลเลอร์ MCS-51 ของหน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบรอม (READ ONLY MEMORY) หรืออีพรอม (ERASABLE PROGRAMABLE READ ONLY MEMORY) ในไมโครคอนโทรลเลอร์ MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้นคือ หน่วยความจำโปรแกรมภายใน (INTERNAL PROGRAM MEMORY) ซึ่งเป็นหน่วยความจำรอม หรืออีพรอม ที่อยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก (EXTERNAL PROGRAM MEMORY) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ



รูปที่ 2.15 แสดงการใช้หน่วยความจำสำหรับเก็บโปรแกรม



รูปที่ 2.16 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมภายในและภายนอก

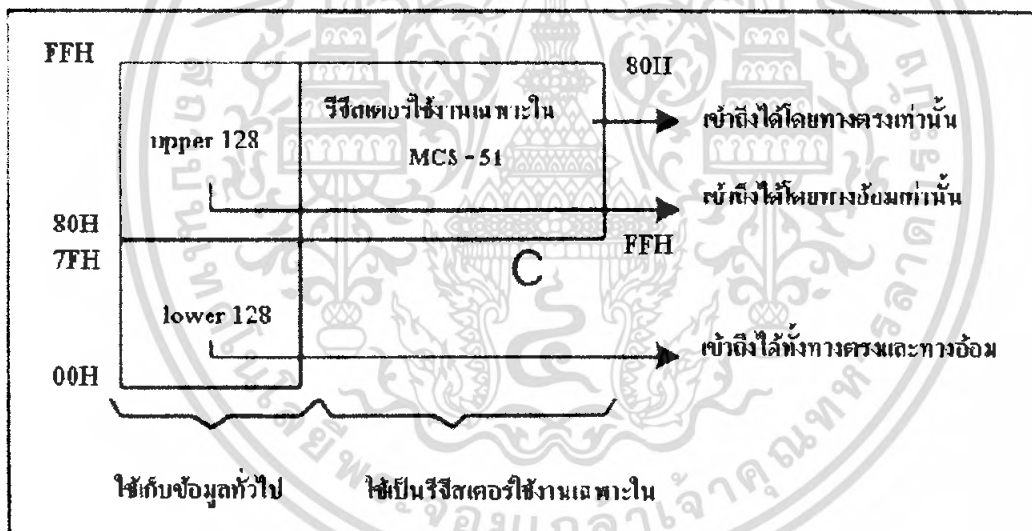
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ ของตระกูล 8051 นี้สามารถขยายให้ใช้งานในหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้วการอ้างตำแหน่งแอดเดรสที่มีทั้งในหน่วยความจำโปรแกรมภายใน และภายนอกนั้นจะต้องทำการควบคุมระดับลอจิกของสัญญาณ ในขณะที่นั้นด้วยขนาดหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ภายในตระกูล 8051 จะแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานลักษณะต่างๆ

-8051 และ 8052 มีหน่วยความจำแบบรวม 4 และ 8 กิโลไบต์

-8751 มีหน่วยความจำแบบอีพรอม ขนาด 4 กิโลไบต์ข้อมูลที่จัดเก็บภายในนี้ ซึ่งสามารถใช้แสงอุลตราไวโอเลตลบ และนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง

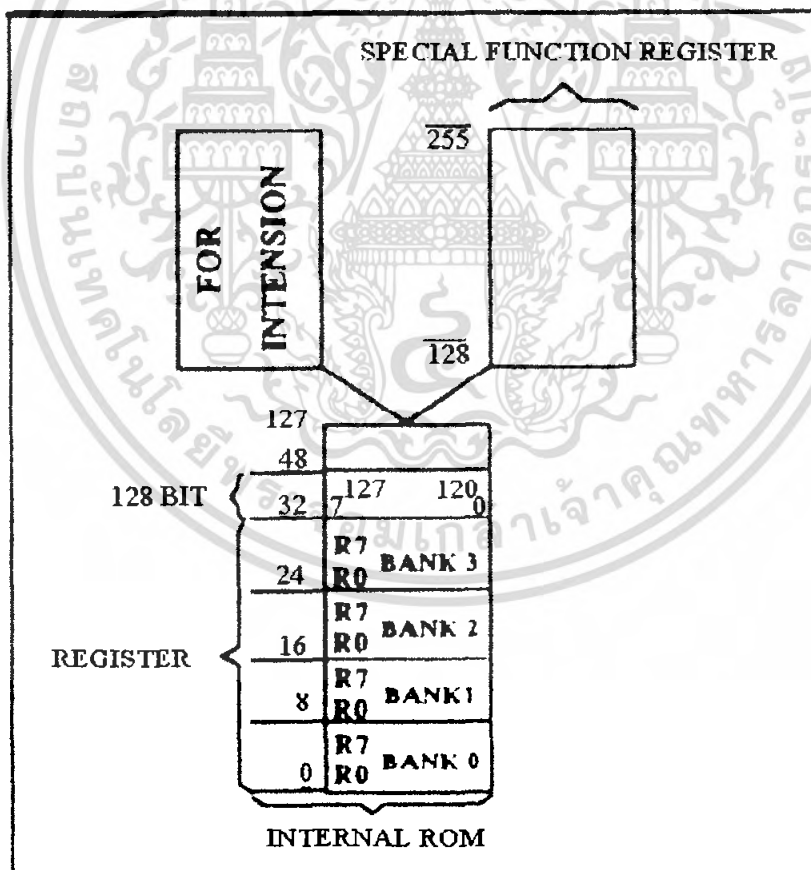
-8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซี ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำโปรแกรมภายนอกเสมอ



รูปที่ 2.17 แสดงหน่วยความจำสำหรับเก็บข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51

2.4.4 หน่วยความจำข้อมูล

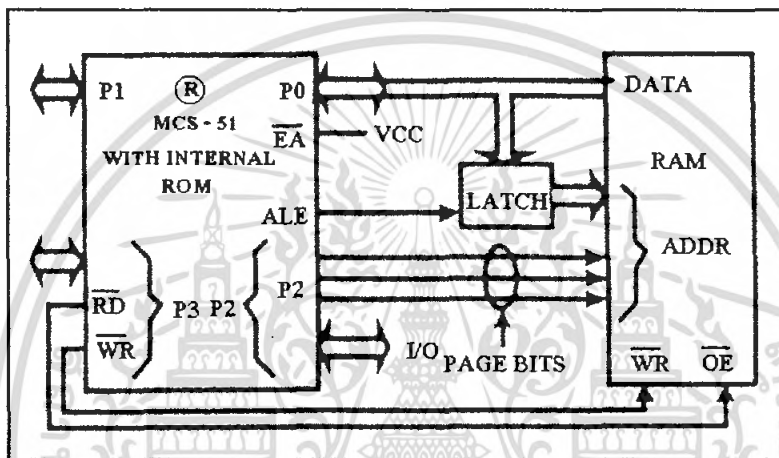
หน่วยความจำข้อมูล (DATA MEMORY) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำแรม สามารถเขียนหรืออ่านข้อมูลได้ (READ OR WRITE MEMORY) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่งโดยพื้นฐานแล้ว หน่วยความจำข้อมูลจัดเป็นหน่วยความจำแรมแบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำ นั่นคือ ตามลักษณะของหน่วยความจำโปรแกรมภายใน ซึ่งก็เป็นแรมที่อยู่ภายในตัวไอซีในตระกูลของไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่ม เต็มเข้าไปในวงจร ลักษณะเดียวกับ การนำไอซีอีพรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 2.18 แสดงการจัดหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

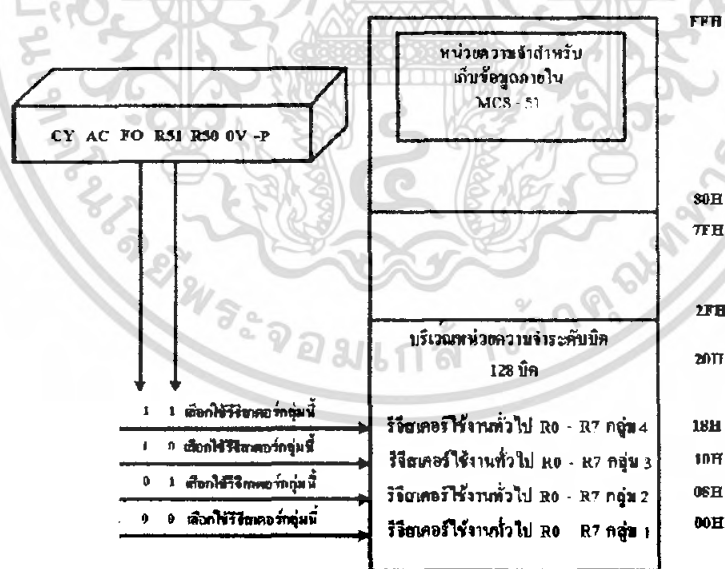
โดยที่หน่วยความจำสำหรับเก็บข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 นี้สามารถแบ่งออกเป็น 2 ส่วนคือ ในส่วนที่เป็นหน่วยความจำสำหรับเก็บข้อมูลภายในไอซี และหน่วยความจำสำหรับเก็บข้อมูลภายนอกไมโครคอนโทรลเลอร์ MCS-51 ทุก ๆ เบอร์จะมีหน่วยความจำเก็บข้อมูลทั่ว ๆ ไป ภายในไอซีอย่างน้อยคือ 128 ไบต์ ไปจนถึง 256 ไบต์ ทั้งนี้ขึ้นกับเบอร์ของไอซี หน่วยความจำสำหรับเก็บข้อมูลภายในไอซีในบริเวณ 128 ไบต์เรียกว่า LOWER 128 และในบริเวณ 128 ไบต์หลัง ที่มีเพิ่มในบางเบอร์มีชื่อเรียกว่า UPPER



รูปที่ 2.19 แสดงการต่อกับหน่วยความจำข้อมูลภายนอกไอซี

2.4.5 รีจิสเตอร์ที่เกี่ยวข้องกับไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ในกลุ่มนี้ จะเป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้งาน เพื่อเก็บข้อมูลของตัว แอคเตสเป็นสำคัญ โดยค่าที่อยู่ภายในแอคเตสนี้จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัส แอคเตส ในส่วนของไมโครคอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดใน กลุ่มนี้ประกอบด้วยรีจิสเตอร์ใช้งานทั่วไป (GENERAL-PURPOSE REGISTERS) รีจิสเตอร์ใน กลุ่มนี้จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผล การทำงานจากหน่วย ประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุด นอกจากนี้ โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังใช้เป็นการเก็บข้อมูลตัวแปรภายในโปรแกรม จะเห็นได้ว่าชื่อ ของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบงก์ใด ก็จะมีชื่อว่า R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นใน การใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่า ต้องการรีจิสเตอร์นั้นๆ จากแบงก์ใดๆ ซึ่งการกำหนด เลือกแต่ละกลุ่มของรีจิสเตอร์นี้ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในแฟลช (PSW) เท่านั้นอย่างไรก็ตาม โดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 ถึง R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วย ความจำแรม



รูปที่ 2.20 แสดงการเลือกใช้รีจิสเตอร์ใช้งานทั่วไป R0-R7 ในแต่ละกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **รีจิสเตอร์หน้าที่พิเศษ** เป็นรีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรส ที่เป็นของรีจิสเตอร์นั้นก็ได้ การจัดพื้นที่หน่วย ความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่า รีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่มีจำนวนเป็นทวีคูณของค่า 8 จะสามารถเข้าถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

- **แอกคิวมูลเตอร์ (ACCUMULATOR) หรือ ACC** เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในหน่วยประมวลผลกลาง และเก็บผลลัพธ์ที่ได้จากการทำงานเท่านั้น การทำงานของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานในโปรแกรมซึ่งใช้เรียกเป็น รีจิสเตอร์ A

2.4.6 ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยคำสั่งทั้งหมดจำนวนมาก ซึ่งสามารถจะจัดกลุ่มคำสั่งเหล่านี้ตามลักษณะและหน้าที่การทำงานที่คล้ายคลึงกัน เพื่อความสะดวกต่อการศึกษาทำความเข้าใจและใช้งาน ดังนี้

2.4.6.1 กลุ่มการถ่ายเทข้อมูล คือ กลุ่มคำสั่งในการโอนย้ายข้อมูล ทำหน้าที่ในโอนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือหน่วยความจำภายในแรม โดยมีรายละเอียดดังนี้ ชุดคำสั่งในการถ่ายเทแรม ภายในนั้น ซึ่งเวลาที่ใช้ในหนึ่งคำสั่งนั้นจะเป็นเวลา เมื่อขณะที่ความถี่ในการทำงานของหน่วยประมวลผลกลางที่ความถี่ 12 เมกะเฮิร์ตซ์ และรายละเอียดของแต่ละคำสั่งมีดังนี้ MOV :จะทำงานในลักษณะเป็นการถ่ายเทข้อมูลที่มีขนาดเป็นไบต์ หรือ บิตก็ได้ จากแหล่งกำเนิดเข้าสู่ตัวรับข้อมูลในฟิลด์ โอเปอร์เรนด์ PUSH:จะทำงานโดยเพิ่มค่ารีจิสเตอร์ SP ก่อนแล้วจึงทำการถ่ายเทข้อมูล 1 ไบต์จากแหล่งกำเนิดไปบริเวณสแต็กตามตำแหน่งที่รีจิสเตอร์ SP กำหนด POP:การถ่ายเทข้อมูลขนาด 1 ไบต์ จากบริเวณตำแหน่งที่รีจิสเตอร์ SP กำหนดไปยังรีจิสเตอร์ที่โอเปอร์เรนด์ กำหนดและหลังจากนั้น รีจิสเตอร์ SP จะลดค่าลง XCH:คำสั่งแลกเปลี่ยนไบต์ระหว่างแหล่งกำเนิดโอเปอร์เรนด์กับรีจิสเตอร์ AXCHD คำสั่งในการแลกเปลี่ยนขนาดนิบเปิดทางอันดับต่ำของแหล่งกำเนิดโอเปอร์เรนด์กับนิบเปิดอันดับต่ำของแอกคิวมูลเตอร์ ตัวอย่างเช่น ทำการเลื่อนข้อมูลไป 2 ไบต์ทางขวามือ ซึ่งจะมี 2 วิธีคือ ใช้คำสั่ง MOV หรือใช้คำสั่ง XCH รายละเอียดการใช้คำสั่งทั้ง 2 แบบ

2.4.6.2 กลุ่มคำสั่งทางคณิตศาสตร์ เช่น การบวก ลบ คูณ และหารข้อมูลภายในตัวรีจิสเตอร์ต่างๆ ช่วงเวลาการทำงานของแต่ละคำสั่งนั้นจะกำหนดที่ความถี่ของสัญญาณนาฬิกาที่ 12 เมกะเฮิร์ตซ์ คำสั่งทางคณิตศาสตร์ส่วนใหญ่ใช้เวลา 1 ms ยกเว้นคำสั่ง INC DPTR ซึ่งใช้เวลา 2 ms โดยที่คำสั่งการคูณและหารใช้เวลา 4 ms โดยมีรายละเอียดการใช้คำสั่งมีดังนี้ INC:เป็นการบวกหนึ่งกับโอเปอร์แรนด์และใส่ค่าใหม่กลับเข้าที่ตัวโอเปอร์แรนด์นั้น ๆ DEC:เป็นการลบออกจากตัวเลขที่อยู่ในแหล่งกำเนิด โอเปอร์แรนด์ และนำผลลัพธ์ที่ได้มาเก็บไว้ที่ตัวโอเปอร์แรนด์นั้น ADD:เป็นการบวกในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์แรนด์ ADDC:เป็นการบวกค่าต่างๆ ในแอกคิวมูลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์แรนด์และบวกกับบิตทดด้วย SUBB:เป็นการนำเลขที่แหล่งกำเนิดโอเปอร์แรนด์ ลบออกจากตัวเลขใน A และนำค่าบิตตัวทศมาลบออกอีกและผลลัพธ์ที่ได้นำมาใส่ลงในแอกคิวมูลเตอร์ A MUL:เป็นการคูณแบบไม่คิดตัวเครื่องหมายของตัวเลขที่อยู่ใน แอกคิวมูลเตอร์กับ เลขในรีจิสเตอร์ B แล้วได้ผลลัพธ์ 2 ไบต์ นำมาเก็บไว้ที่ AB โดย A จะรับอันดับต่ำ ส่วน B จะรับอันดับสูง DIV:เป็นคำสั่งในการหารแบบไม่คิดเครื่องหมายที่อยู่ในแอกคิวมูลเตอร์แล้วหารตัวเลขในรีจิสเตอร์ B แล้วนำผลลัพธ์ไปเก็บในแอกคิวมูลเตอร์ และเศษของการหารตัวเลขจะเก็บไว้ในรีจิสเตอร์ B DA:สำหรับการบวกกันทางตัวเลข BCD เป็นการปรับค่ารวมซึ่งเป็นผลมาจากการบวกกันทางไบนารีของระบบตัวเลข BCD ขนาด 2 หลักสองจำนวนการปรับค่าตัวเลขผลรวมด้วยการใช้คำสั่ง DA จะได้ผลลัพธ์กลับมาที่แอกคิวมูลเตอร์ รายละเอียดการใช้คำสั่ง

2.4.6.3 กลุ่มคำสั่งทางตรรกศาสตร์หรือแบบลอจิก ทำหน้าที่เกี่ยวกับการประมวลผลแบบลอจิก ต่างๆ เช่นการ AND OR หรือ EX-OR ระหว่างข้อมูลในรีจิสเตอร์ A นั่นเองโดยมีการใช้คำสั่งดังนี้ CPL:เป็นการใช้คำสั่งกลับค่าหรือคอมพลิเมนต์ ข้อมูลในแอกคิวมูลเตอร์จะไม่มีผลใดๆ ต่อค่าของแฟล็ก หรือการอ้างถึงตำแหน่งแอดเดรสนั้นตามบิตนั้นๆ RL, RLC, RR, RRC, SWAP: ทั้ง 5 คำสั่งนี้เป็นคำสั่งในการทำงานการวนบิตบนตัวของแอกคิวมูลเตอร์ ซึ่ง RL เป็นการวนบิตทางขวา, RLC เป็นการทำการวนทางซ้ายผ่านบิตทด, RRC เป็นการวนขวาผ่านบิตทด และ SWAP เป็นการวนซ้ายสี่ครั้ง ANL:เป็นการ ADD กันทางตรรกศาสตร์ ระหว่างแหล่งกำเนิดสองโอเปอร์แรนด์ ซึ่งจะสั่งให้ทำงานในรูปแบบของตรรกศาสตร์ทางข้อมูลขนาดเป็นไบต์หรือบิตกลุ่มคำสั่งแบบบูลีนหรือแบบบิตซึ่งมีความสามารถ ของไมโครคอนโทรลเลอร์ MCS-51 ที่จะดำเนินการประมวลผลแบบบิตแทนที่จะเป็นข้อมูล ทั้งไบต์เช่นปกติ โดยมีชุดคำสั่งที่จัดการโดยตรง ทุกคำสั่งจะเข้าถึงข้อมูลโดยตรงในระดับบิต โดยมีการบิตแอดเดรสได้ตั้งแต่ 00H - 7FH ในพื้นที่ 128 บิต หน่วยความจำข้อมูลภายในและบิตแอดเดรส 80H-FFH ในบริเวณกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6.4 กลุ่มคำสั่งในการกระโดดไปยังตำแหน่งต่างๆภายในโปรแกรม ซึ่งจะเปลี่ยนลำดับของการประมวลผลภายใน โปรแกรมไปยังส่วนต่างๆแทนที่จะดำเนินการไปเป็นลำดับต่อเนื่องโดยที่คำสั่ง JMP จะแบ่งเป็น 3 ลักษณะ คือ SJMP, LJMP, AJMP ซึ่งในแต่ละคำสั่งจะมีข้อแตกต่างของการกระโดดไปยังแอดเดรสไกลสุดที่ต่างกันคำสั่ง JMP ซึ่งเป็นแบบ โมณีซิกที่สามารถจะใช้ได้ โดยมีรายละเอียดการใช้งานของคำสั่งดังต่อไปนี้ SMP:จะเป็นการกระโดดแบบการย้ายอันดับตำแหน่งของแอดเดรสตำแหน่งเดิมซึ่งจะสามารถกระโดดได้ -128 ถึง +127 ไบต์ AJMP: ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 2 กิโลไบต์ ซึ่งจะใช้น้อยความจำเพียง 2 ไบต์เท่านั้น ในการกำหนด LJMP:ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 64 กิโลไบต์ ซึ่งจะใช้น้อยความจำเพียง 3 ไบต์เท่านั้นในการกำหนด JMP @A+DPTR:เป็นการควบคุมการกระโดดไปยังโปรแกรมที่ต้องการเฉพาะภายในส่วนต่างๆ

2.4.7 โครงสร้างการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51

จากแผนภาพโครงสร้างระบบอินเตอร์รัปต์ของ ไมโครคอนโทรลเลอร์ MCS-51 สัญญาณที่เข้ามาทำ การอินเตอร์รัปต์ MCS-51 นั้นเกิดขึ้นได้ 5 ลักษณะตามตารางข้อมูล ในรูปที่ 2.6 โดยจะเห็นได้ว่าสามารถที่จะกำหนดเลือกเพื่อยินยอม (หรืออีนาเบิล : ENABLE) และห้าม (หรือดิสเอเบิล:DISABLE) ไม่ให้มีการอินเตอร์รัปต์แต่ละประเภทได้ โดยการกำหนดบิตของข้อมูลที่เกี่ยวข้อง ซึ่งมักจะอยู่ภายในรีจิสเตอร์ TCONและSCON นอกจากนี้ยังมีตำแหน่งบิตภายในรีจิสเตอร์ IE(INTERRUPT ENABLE REGISTER) ซึ่งทำหน้าที่เสมือนกับเป็นสวิตซ์หลักที่เกี่ยวข้องกับสัญญาณอินเตอร์รัปต์ทั้งหมด หากว่ากำหนดไม่ให้เกิดการอินเตอร์รัปต์ แล้วการกำหนดบิตเพื่อห้ามหรือยินยอมของแต่ละอินเตอร์รัปต์ก็จะไม่มีผลใดๆ เกิดขึ้นยังแสดงให้เห็นว่าสัญญาณอินเตอร์รัปต์ แต่ละประเภทยังสามารถกำหนดระดับความสำคัญ (PRIORITY) ของการอินเตอร์รัปต์ได้สองลักษณะ คือระดับความสำคัญสูงหรือต่ำ (HIGH OR LOW PRIORITY) กล่าวคือ ขณะที่กำลังประมวลผลอยู่ภายในส่วนของ โปรแกรมย่อยบริการอินเตอร์รัปต์ ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเตอร์รัปต์ที่มีระดับความสำคัญสูงกว่า แต่หากว่าเป็นสัญญาณอินเตอร์รัปต์ที่มีระดับความสำคัญต่ำ เช่นเดียวกันแล้วก็ต้องรอให้เสร็จสิ้นการประมวลผลที่ ดำเนินการอยู่ก่อน

2.4.8 การรีเซต

โดยความหมายของการรีเซตเป็นการบังคับให้มีการเริ่มต้นใหม่อีกครั้งหนึ่ง ซึ่งมักจะกระทำโดยการกำหนดสถานะของสัญญาณที่ขารีเซตของไอซี MCS-51 ให้เป็นระดับลอจิกที่เหมาะสมเท่านั้น การรีเซตด้วยวิธีนี้ถือว่าการอินเตอร์รัปต์อย่างหนึ่งได้ แต่จะมีลักษณะต่างออกไปจากการอินเตอร์รัปต์ของสัญญาณนี้ได้ ซึ่งมีศัพท์เฉพาะเรียกว่า NON-MASKABLE INTERRUPT นอกจากนี้การดำเนินการของโปรแกรมก็แตกต่างออกไปด้วย โดยจะไม่มี การเก็บค่าของคำสั่งที่กำลังจะไปทำในลำดับต่อไปภายในรีจิสเตอร์ PC เมื่อมีการรีเซตเกิดขึ้น โปรแกรมจะถูกสั่งให้กระโดดไปยังแอดเดรส 0000 ทันทันที ซึ่งตำแหน่งนี้จะเป็นตำแหน่งเริ่มต้นของการทำงานของไมโครคอนโทรลเลอร์ MCS-51 เมื่อเริ่มจ่ายไฟให้กับระบบเมื่อใดก็ตามที่มีการรีเซตเกิดขึ้นค่าสถานะต่างๆ ภายในไมโครคอนโทรลเลอร์จะถูกกำหนดกลับไปเป็นค่าเริ่มต้นใหม่อีกครั้ง

2.5 การใช้งานไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม

การติดต่อกับพอร์ตอนุกรมนั้นจะยากกว่าการติดต่อกับกับพอร์ตขนาน การสื่อสารของอุปกรณ์ที่ต่อกับพอร์ตอนุกรมจะถูกเปลี่ยน (Convert) เป็นสัญญาณแบบขนาน แล้วจึงนำไปประมวลผลต่อ ซึ่งจะ ใช้ Universal Asynchronous Receive / Transmitter (UART) เป็นตัวทำหน้าที่ ส่วนทางด้านโปรแกรม ก็มีรีจิสเตอร์ที่ต้องจัดการมากกว่า Standard Parallel Port (SPP) อีกหลายตัว **ข้อดีของพอร์ตอนุกรม**

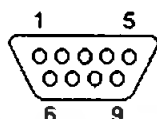
ระยะของสายอนุกรมสามารถมีความยาวได้มากกว่าสายของขนานมาก ทั้งนี้เพราะสัญญาณของพอร์ตอนุกรม ซึ่งส่วนใหญ่ใช้มาตรฐาน RS-232C จะมีค่า -3 Volt ถึง -15 Volt สำหรับ Logic "1" หรือ "Mark" และมีค่า +3 Volt ถึง +15 Volt สำหรับ Logic "0" หรือ "Space" (สำหรับช่วง +3 Volt ถึงถึง -3 Volt เป็นช่วง Undefined) ส่วนสัญญาณของสายขนานนั้น Logic "1" จะมีค่า +5 Volt และ logic "0" จะมีค่า 0 Volt ทำให้สัญญาณของสายอนุกรมสามารถรับการสูญเสียของสาย (Cable loss) ได้มากกว่าสัญญาณของสายขนานปกติสายขนานจะไปได้เพียง 5 ฟุต ส่วนสาย RS-232 จะไปได้ถึง 50 ฟุตที่ความเร็วสูงสุดของมัน

สายอนุกรมจะใช้จำนวนสายไฟน้อยกว่าสายขนานถ้าอยู่ในลักษณะ Null Modem จะใช้สายเพียง 3 เส้น ขณะที่แบบขนาน จะต้องใช้สาย 19 ถึง 25 เส้น

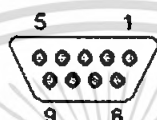
การสื่อสารแบบไร้สายเช่นการใช้ Infra Red การส่งพร้อมกันทีละ 8 บิต แบบขนาน จะทำให้ไม่สามารถแยกแยะได้ว่า Bit ไหนเป็น Bit0 หรือ Bit1 ... เป็นต้น ปัจจุบันอุปกรณ์ IrDA มีความเร็วไม่ต่ำกว่า 115.2K Baud แต่มี Pulse length เพียง 3/16 ของ RS-232 เพื่อประหยัดพลังงาน เพราะส่วนมากใช้ในอุปกรณ์แบบพกพาเช่น Laptop หรือ Palmtop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันไมโครคอนโทรลเลอร์ มักมีการผนวกพอร์ตการสื่อสารแบบอนุกรมไว้ด้วย เพราะใช้จำนวนขาน้อยกว่าแบบขนาน RS-232C กำหนด Baud rate ไว้ไม่เกิน 20K Baud ปัจจุบันได้แก้ไขให้รองรับกับเทคโนโลยีใหม่ได้ จึงมีการปรับปรุงถึง RS-232E ซึ่งมีรายละเอียดอีกหลายอย่าง



(Male at the computer side)



(Female at the cable side)

รูปที่ 2.21 แสดงขาของ RS-232

ตารางที่ 2.2 แสดงถึงหน้าที่ของขาต่างๆ

Pin	Name	RS232	V.24	Dir	Description
1	CD	CF	109	←	Carrier Detect
2	RXD	BB	104	←	Receive Data
3	TXD	BA	103	→	Transmit Data
4	DTR	CD	108.2	→	Data Terminal Ready
5	GND	AB	102	—	Signal Ground
6	DSR	CC	107	←	Data Set Ready
7	RTS	CA	105	→	Request to Send
8	CTS	CB	106	←	Clear to Send
9	RI	CE	125	←	Ring Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abbreviation	Full Name	Originator	Function
TD	Transmit Data	DTE	Serial data output (TXD) from DTE.
RD	Receive Data	DCE	Serial data input (RXD) to DTE.
CTS	Clear To Send	DCE	Tell DTE that DCE is ready to exchange data.
(D)CD	(Data) Carrier Detect	DCE	Carrier from remote DCE is detected.
DSR	Data Set Ready	DCE	Tell DTE that DCE is ready to establish a link.
DTR	Data Terminal Ready	DTE	Tell DCE that DTE is ready to establish a link.
RTS	Ready To Send	DTE	Tell DCE that DTE is ready to exchange data.
RI	Ring Indicator	DCE	Ringling signal from the phone line is detected.

รูปที่ 2.22 แสดงสัญญาณต่างๆที่ส่งในรูปแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 หน้าที่ของสัญญาณต่าง ๆ มีดังนี้

Protective ground เป็นจุดที่ต่อกับตัวเปลือกของอุปกรณ์ และไม่ต่อกับสัญญาณใด ๆ ในระบบเพื่อใช้ต่อลงดิน เป็นการป้องกันอันตรายจากไฟลัดวงจร และใช้เป็น Shield ป้องกันการรบกวน

Signal ground หรือ **Common return** เป็นจุดสำคัญที่สุดที่ต้องมีในระบบ RS-232 เพราะเป็นจุดอ้างอิงของทุกสัญญาณ ยกเว้น Protective ground

Transmit data เป็นสัญญาณข้อมูลที่ส่งจาก DTE ไปยัง DCE ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่ง (DTE) หรือ -3 V ถึง -15 V ที่ด้านรับ (DCE) การส่งข้อมูลจะเกิดขึ้นได้ต้องมีสัญญาณควบคุมที่เกี่ยวข้อง "On" ก่อนคือสัญญาณ RTS, CTS, (D)CD, DTR และ DSR

Receive data เป็นสัญญาณข้อมูลจาก DCE มายัง DTE ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่งหรือ -3 V ถึง -15 V ที่ด้านรับ ในขณะที่ไม่ได้ส่งข้อมูลมันจะมีสถานะเป็น Logic "1" หรือ "Mark" หรือ "Off" หรือ -5 V ถึง -15 V ที่ด้านส่ง (DCE) หรือ -3 V ถึง -15 V ที่ด้านรับ (DTE) กรณีที่เป็นการสื่อสารแบบ Half-duplex สัญญาณ RD จะอยู่สถานะ "Off" ขณะที่สัญญาณ RTS อยู่ในสถานะ "On" และสัญญาณ RD จะยังคงอยู่ในสถานะ "Off" อีกชั่วระยะเวลาหนึ่งหลังจากที่สัญญาณ RTS เปลี่ยนจากสถานะ "On" มาเป็น "Off" แล้วเพื่อให้การรับ-ส่งข้อมูลเสร็จสมบูรณ์

Request to send เป็นสัญญาณจาก DTE ส่งไปให้ DCE เพื่อขอส่งข้อมูลไปปรกติจะอยู่ในสถานะ "Off" เมื่อต้องการส่งข้อมูลจะเปลี่ยนเป็นสถานะ "On" จนกว่าการส่งเสร็จสิ้นจึงเปลี่ยนกลับมาที่สถานะ "Off" ตามเคม ทั้งนี้ทาง DTE ต้องได้รับสัญญาณ CTS จึงจะสามารถส่งข้อมูล TD ไปยัง DCE ได้ และสัญญาณ RTS ที่กลับสู่สถานะ "Off" จะไม่สามารถเปลี่ยนเป็น "On" ใหม่ ขณะที่สัญญาณ CTS อยู่ในสถานะ "On" ต้องรอจนกว่าสัญญาณ CTS เปลี่ยนมาอยู่ในสถานะ "Off" ก่อน เพื่อป้องกันการเกิด Overrun

Clear to send เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าพร้อมรับการส่งข้อมูลจาก DTE

Data set ready เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่า DCE สามารถเชื่อมโยงไปยังปลายทางและพร้อมที่จะติดต่อแล้ว

Data Terminal ready เป็นสัญญาณที่ DTE ส่งให้ DCE เพื่อแจ้งว่า DTE พร้อมหรือต้องการจะติดต่อสื่อสาร ซึ่งสัญญาณ DTR นี้ต้องเกิดก่อนทาง DCE จึงจะทำการติดต่อไปยังปลายทางและเมื่อติดต่อได้แล้วจึงส่งสัญญาณ DSR มายัง DTE เพื่อแจ้งให้รู้ว่าพร้อมรับการสื่อสารแล้ว และถ้า DTR เปลี่ยนเป็น Off แปลว่า DTE ไม่ต้องการติดต่อสื่อสารแล้วทาง DCE ก็จะปิดช่องสื่อสารและเปลี่ยนสัญญาณ DSR เป็น Off ทันที คู่สัญญาณระหว่าง RTS กับ CTS เป็นเรื่องของความพร้อมเกี่ยวกับช่องสื่อสารระหว่าง DTE กับ DCE ส่วนคู่สัญญาณ DTR กับ DSR เป็นเรื่องของความพร้อมเกี่ยวกับตัวอุปกรณ์ DTE กับ DCE

Data carrier detect เป็นสัญญาณที่ DCE ส่งให้ DTE เพื่อแจ้งว่าได้รับสัญญาณพาหะจาก DCE ที่อยู่อีกด้านหนึ่งของการสื่อสาร ซึ่งหมายความว่า ช่องการสื่อสารระหว่าง DCE ทั้ง 2 ไม่ขาดตอน พร้อมทั้งจะทำการสื่อสารได้ ซึ่งอุปกรณ์ DTE หรือโปรแกรมที่ควบคุมการสื่อสารมักจะตรวจ สัญญาณนี้ ถ้าไม่อยู่ที่ on แสดงว่าช่องการสื่อสารขาด ก็จะไม่ทำการรับหรือส่งข้อมูล

Ring Indicator เป็นสัญญาณจาก DCE แจ้งให้ DTE รู้ว่ามีการเรียกจาก DCE ที่อยู่อีกด้านหนึ่งของการสื่อสาร ซึ่งมักจะใช้ในระบบ automatic answering

2.5.2 Null Modems

Null Modem ใช้สำหรับเชื่อมโยงระหว่าง DTE 2 ตัวเข้าด้วยกันโดยตรง ซึ่งโดยมากใช้ในการถ่ายข้อมูลระหว่างคอมพิวเตอร์ หรือใช้ในการพัฒนาระบบไมโคร โปรเซสเซอร์



รูปที่ 2.23 แสดงการเชื่อมต่อของสายสัญญาณ

2.5.3 Flow Control

การติดต่อระหว่าง DTE และ DCE จะมีการควบคุมการไหลของข้อมูล เพื่อไม่ให้เกิดการ over flow ขึ้นได้ ซึ่งมีอยู่ 2 แบบคือ Hardware flow control และ Software flow control สำหรับ Software flow control มักจะเรียกว่า Xon/Xoff flow control ซึ่งใช้รหัส ASCII 17 เป็นสัญญาณ Xon และใช้รหัส ASCII 19 เป็นสัญญาณ Xoff หลักการทำงานที่ง่าย ๆ คือ Modem จะมี Buffer อยู่เมื่อ Modem รับข้อมูลจาก Computer จน Buffer ใกล้เคียงจะเต็มมันก็จะส่งสัญญาณ Xoff ไปให้ Computer เพื่อให้ Computer หยุดส่งข้อมูลให้มันชั่วคราว และเมื่อ Buffer มีที่ว่างถึงระดับหนึ่ง Modem ก็จะส่งสัญญาณ Xon ไปให้ Computer เพื่อให้ Computer ส่งข้อมูลให้มันต่อ การควบคุมโดยวิธีนี้ประหยัดสายสัญญาณ เพราะรับ-ส่งผ่าน TD และ RD แต่อาจทำให้การสื่อสารช้าลงอย่างเห็นได้ชัดในกรณีที่ใช้กับการสื่อสารที่มี speed ต่ำ เพราะแต่ละตัวอักษร ASCII ที่รับ-ส่ง จะมีขนาด 10 Bit

ส่วน Hardware flow control มักจะเรียกว่า RTS/CTS flow control จะใช้สายสัญญาณของ Serial Port ในการควบคุม ทำให้ไม่บั่นทอนความเร็วของข้อมูล หลักการทำงานคือ เมื่อ Modem มีที่ว่างเพื่อรับข้อมูล มันก็จะส่งสัญญาณ CTS ไปให้ Computer และเมื่อมันใกล้เคียงจะเต็ม มันก็จะหยุดส่งสัญญาณ CTS ไปให้ Computer

2.5.4 รูปคลื่น สัญญาณ RS-232

การสื่อสารโดย RS-232 เป็นการสื่อสารแบบ asynchronous หมายความว่าสัญญาณ clock ที่ใช้ควบคุมจังหวะไม่ได้ส่งไปพร้อมกับ Data แต่จะใช้ start bit เป็นตัว sync. ในแต่ละ word ของการสื่อสารและใช้สัญญาณ clock ภายในของแต่ละคำเป็นตัวให้จังหวะเอง



รูปที่ 2.24 รูปคลื่นของสัญญาณที่ส่ง

แสดงลักษณะของสัญญาณจาก UART เมื่อใช้ format แบบ 8N1 คือ 8 data bits ไม่มี parity bit และมี 1 stop bit ขณะที่ idle จะอยู่ในสถานะ "Mark" หรือ logic "1" การส่งจะเริ่มจากการส่ง start bit คือ logic "0" และตามด้วย LSB bit จนหมด data bits และถ้ามี parity bit ก็จะส่งที่จุดนี้ แล้วลงท้ายด้วย stop bit ซึ่งมีค่าเป็น logic "1" ในรูปได้แสดง bit ที่ต่อถัดจาก stop bit ซึ่งมีค่าเป็น logic "0" หมายความว่า เป็น start bit ของ การส่ง word ถัดไป แต่ถ้ายังไม่มีการส่ง word ถัดไป ก็ต้องอยู่ในสถานะของ logic "1" ซึ่งเป็นสถานะของ idle และถ้าสายอยู่ในสถานะของ logic "0" นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

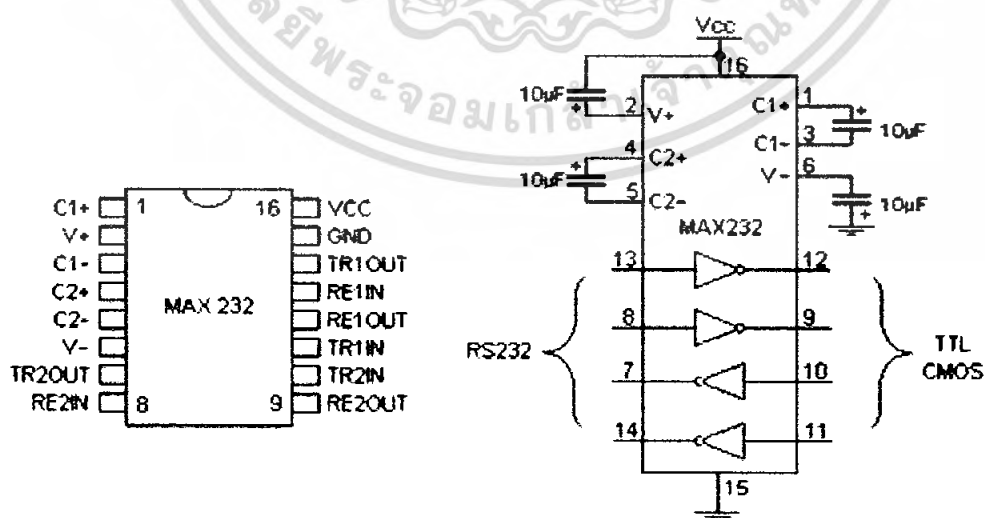
กว่าเวลาของการส่ง 1 full word ระบบจะถือว่าเป็นสัญญาณ “Break” เพื่อหยุดการสื่อสาร ดังนั้น ต้องไม่ลืมที่จะตั้งในสายกลับสู่สถานะ idle เมื่อสิ้นสุดการส่ง การรับ-ส่งข้อมูลในลักษณะนี้เรียกว่า แบบ frame คือมีกรอบปิดล้อมข้อมูลไว้ด้วย start bit และ stop bit

2.5.5 ตัวแปลงสัญญาณ RS-232

สัญญาณ RS-232 มีค่าแรงไฟต่างจากที่ใช้ใน UART ดังนั้นจึงต้องมีตัวแปลงสัญญาณเพื่อแปลงระดับสัญญาณให้เหมาะสมก่อนที่ จะเชื่อมต่อกับพอร์ตอนุกรม หรือ RS-232 port ของคอมพิวเตอร์

สัญญาณ RS-232 นั้น logic “0” จะมีค่า +3 V ถึง +25 V และ logic “1” จะมีค่า -3 V ถึง -25 V ส่วนค่าระหว่าง -3 V ถึง +3 V เป็นค่า undefined ระดับสัญญาณนี้ใช้กับทุกสัญญาณไม่ใช่ เฉพาะสัญญาณรับ-ส่งข้อมูลเท่านั้นแต่ยังรวมถึงสัญญาณควบคุมต่าง ๆ เช่น DTR, RTS, CTS เป็นต้น

IC ที่ใช้มักจะเป็นเบอร์ 1488 (RS-232 Driver) และ 1489 (RS-232 Receiver) โดยภายใน แต่ละตัวจะประกอบด้วย inverter 4 ตัวและต้องการไฟเลี้ยง 2 ชุดคือ +7.5 V ถึง +15 V และ -7.5 V ถึง -15 V ซึ่งอาจจะมีปัญหาในเครื่องที่มีไฟเลี้ยง +5 V เพียงชุดเดียว แต่ก็ยังมี IC อีกตัวหนึ่งคือ เบอร์ MAX-232 ซึ่งมีวงจร charge pump สามารถสร้างไฟ +10 V และ -10 V จากไฟ +5 V ได้ พร้อมทั้งมี 2 Tx และ 2 Rx อยู่ใน package เดียวกัน และรองรับ baud rate ได้ถึง 120 Kbps จึง สะดวกมากเพราะใช้ IC เพียงตัวเดียว รูปข้างล่างคือ MAX-232



รูปที่ 2.25 แสดง โครงสร้างภายในและตำแหน่งขาต่างๆของ Max232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการที่เราจะนำข้อมูลมาใช้งานก็ต้องแปลงเป็น Parallel ก่อนซึ่งเป็นหน้าที่ของ UART ซึ่งปัจจุบันไมโครคอนโทรลเลอร์มักจะมี serial communication interface (SCI) อยู่ในตัว แต่อาจจะมึงานบางอย่างที่ไม่ได้ใช้ไมโครคอนโทรลเลอร์และต้องการประมวลผลข้อมูลกับพอร์ตอนุกรม เช่น ต่อ ADC เข้ากับ UART หรือต่อ LCD display เข้ากับ Serial comm. ก็ต้องใช้ UART ช่วย เช่น เบอร์ 8250 หรือ 16550A หรือเบอร์อื่น ๆ ที่ได้กล่าวมาแล้ว แต่มี UART อีกพวกหนึ่งที่แยก Tx bus กับ Rx bus ออกจากกันทำให้มีความคล่องตัวมากขึ้น

2.6 โปรแกรม Visual Basic (VB)

เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่ นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจกต์) นั้นหมายความว่า ไม่ว่าจะ เป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจกต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก้ไขคุณสมบัติของออบเจกต์นั้นได้โดยตรง ในทุกๆ ออบเจกต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจกต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันได้ ขึ้นอยู่กับชนิดของออบเจกต์ ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเจอร์ (procedure) แต่ละโพรซีเจอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจกต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

บทที่ 3

การออกแบบและการทำงาน

3.1 แนวทางการออกแบบ

3.1.1 จุดประสงค์ก่อนออกแบบ

1. วงจรภาครับสัญญาณจากการเคาะสามารถทำงานได้
2. ควบคุมให้สามารถวัดระยะเวลาทางระหว่างตัวรับกับจุดที่เคาะได้
3. นำผลต่างของเวลาที่รับได้ส่งเข้าไปที่ FPGA ไปเปรียบเทียบกับสัญญาณนาฬิกา เพื่อที่จะนำความต่างของสัญญาณนาฬิกาไปหาพิกัด X,Y โดยใช้ MCS-51
4. เขียนโปรแกรมในส่วนการควบคุมเม้าส์โดยใช้ Visual Basic

3.1.2 ขั้นตอนการออกแบบ

ใช้หลักการของการเคลื่อนที่ของคลื่นที่ได้จากการเคาะเคลื่อนที่ไปบนผิวของวัตถุตัวกลาง โดยที่ปลายทางมีตัวรับคอยรับสัญญาณอยู่ โดยมีความสัมพันธ์กันทางหลักฟิสิกส์ที่ว่า ระยะเวลาที่คลื่นใช้เคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่งสัมพันธ์กับระยะทาง $s = vt$

ดังนั้นหลักการการบอกตำแหน่งของการเคาะ บนจอโปรเจกเตอร์ มีหลักการทำงานดังนี้ เริ่มด้วย การเคาะบนจอที่สร้างขึ้นโดยแผ่นกระจก สัญญาณจากการเคาะจะถูกส่งไปที่ตัวรับที่ตำแหน่งระยะทางที่เรากำหนดไว้ นั่นคือที่มุมทั้ง 4 ของบอร์ด โดยตัวรับจะมีอยู่ 4 ตำแหน่ง คือ P0,P1,P2,P3 ตามลำดับ ที่ตำแหน่ง $(X_0, Y_0), (X_1, Y_1), (X_2, Y_2)$ และ (X_3, Y_3) ตามลำดับ การหาพิกัด (X,Y) ทำได้โดยการวัดระยะทางที่อยู่ในรูปของเวลาระหว่างจุดที่เคาะยังจุดอ้างอิงทั้ง 4 คือ $R+R_0, R+R_1, R+R_2, R+R_3$ โดยใช้ FPGA เป็นตัวควบคุมการนับโดยใช้วงจรวอร์นเคอร์ หากตำแหน่งที่เคาะอยู่ใกล้ตัวรับใดระยะทางที่คลื่นเคลื่อนที่ไปยังตัวรับนั้นจะน้อยมีผลให้จำนวนที่นับของเคอร์วนั้นจะน้อย ค่าจำนวนข้อมูลดังกล่าวที่ได้จะส่งให้กับ MCS-51 เพื่อทำหน้าที่คำนวณค่าพิกัด X,Y จากนั้นก็จะส่งข้อมูล ไปยังคอมพิวเตอร์ นำไปควบคุมการทำงานของเม้าส์ต่อไป

ภาครับสัญญาณจากการเคาะ

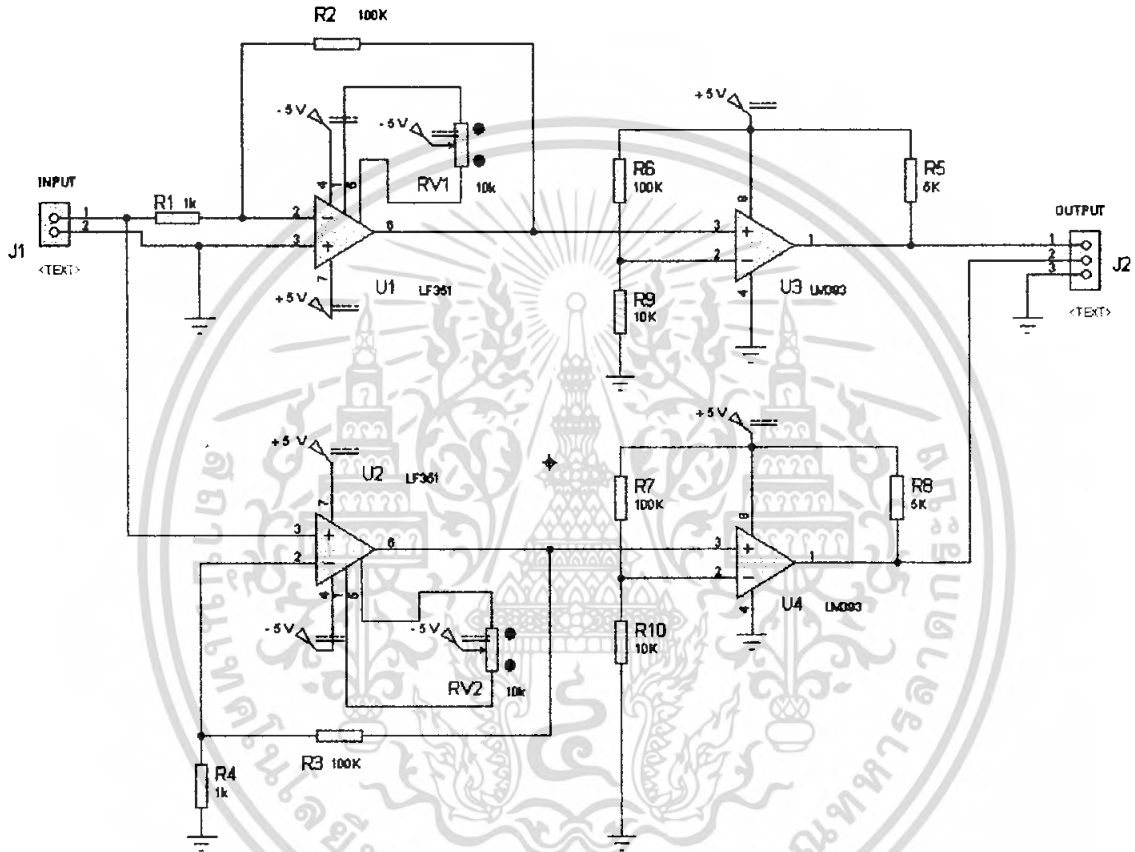
วงจรในส่วนของภาครับสัญญาณจากการเคาะ จะต้องสามารถรับสัญญาณการเคาะที่ได้จากเปียโนที่มีขนาดแรงดันแอมพลิจูดประมาณมากกว่า 200 mV เพื่อต้องการเวลาที่ถูกต้องจึงรับสัญญาณที่มีการเปลี่ยนแปลงลูกแรกเท่านั้น โดยวงจรภาครับ ทำหน้าขยายสัญญาณให้แรงขึ้นแล้วเปลี่ยนจากสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล เพื่อส่งให้กับวงจรเกทใน FPGA เมื่อวงจรภาครับตัวใดตัวหนึ่งได้รับสัญญาณเป็นตัวแรก เคา์นเตอร์ตัวอื่นจะเริ่มนับและหยุดเมื่อวงจรภาครับชุดนั้นได้รับสัญญาณครบทั้ง 4 จุด หรือ รอจนถึงเวลาที่ได้คำนวณไว้ว่าถ้าเวลาผ่านไปเท่านี้ภาครับยังรับสัญญาณได้ไม่ครบให้ตัวรับเริ่มรับค่าใหม่ ค่าที่ได้จะนำไปคำนวณหาตำแหน่งของตัวชี้ต่อไป

ภาคประมวลผล

วงจรในส่วนของ FPGA ที่ทำหน้าที่รับสัญญาณจากวงจรภาครับ นำสัญญาณที่เป็นผลต่างของเวลามาเปรียบเทียบและนับระยะเวลาที่เกิดขึ้นระหว่างตัวส่งกับตัวรับแต่ละจุด โดยใช้วงจรเคา์นเตอร์ทำการนับ ค่าเวลาที่ได้จะส่งให้กับ MCS-51 เพื่อทำหน้าที่คำนวณค่าพิกัด X,Y โดยใช้สมการทางคณิตศาสตร์ จากนั้นจะส่งพิกัดที่ได้สู่คอมพิวเตอร์ผ่านทางพอร์ตสื่อสารแบบ RS-232 จากนั้นใช้โปรแกรม Microsoft Visual Basic 6.0 นำพิกัดระนาบ X, Y ความคลุมการเคลื่อนที่ของเมาส์เพื่อสั่งงานหน้าจอวินโดว์ ต่อไป

3.2 ภาครับสัญญาณจากการเคาะ

หน้าที่ของวงจรภาครับสัญญาณจากการเคาะคือขยายสัญญาณให้มีความแรงขึ้นหลายๆเท่า จากนั้นก็ทำการเปรียบเทียบแรงดัน และให้สัญญาณดังกล่าวเปลี่ยนเป็นพัลส์รูปสี่เหลี่ยม แล้วหลังจากนั้นก็ทำการส่งสัญญาณไปในส่วนของ FPGA ต่อไป



รูปที่ 3.1 แสดงภาครับสัญญาณจากการเคาะ

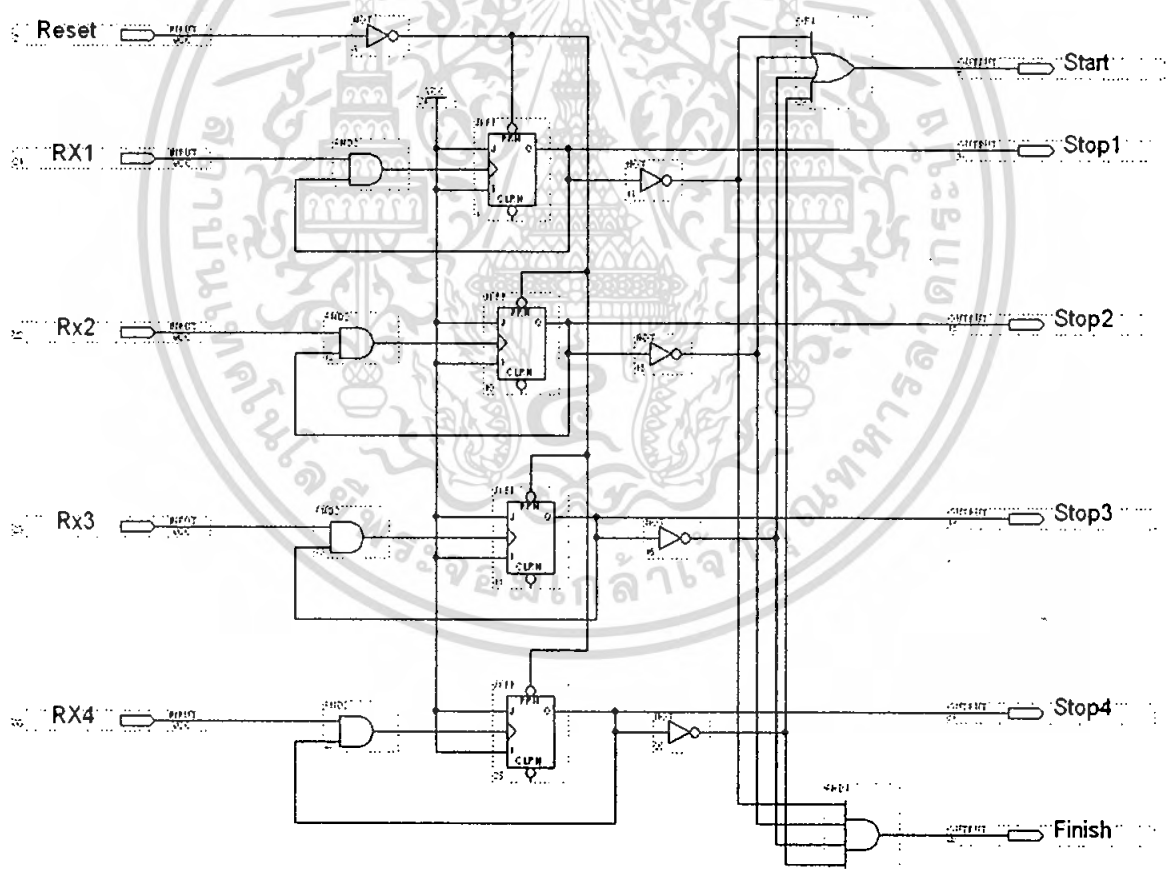
การทำงานในส่วนนี้สามารถแบ่งออกได้เป็น 2 จุดด้วยกันคือ จุดแรกได้แก่ภาครับสัญญาณเข้ามาแล้วทำการขยายสัญญาณแบบกลับเฟส โดยในที่นี้ใช้ LF351 เป็นตัวขยายสัญญาณ input ที่รับเข้ามาซึ่งจะได้เกนซ์ประมาณ 100 เท่า ส่วนจุดที่สองภาครับสัญญาณเข้ามาแล้วทำการขยายสัญญาณเช่นกันแต่จะขยายแบบไม่กลับเฟส เพื่อขยายสัญญาณให้ได้ทั้งซีกบวกและซีกลบ แล้วจึงใช้วงจร COMPARATOR LM393 ทำการเปรียบเทียบแรงดันจากภาครับขยายทั้งสองดังกล่าว โดยมีแรงดันที่ขาลบเป็นตัวอ้างอิงไว้เปรียบเทียบแล้วสัญญาณที่ออกจาก COMPARATOR ทั้งสองจุดจะนำไปรวมกันในส่วนของ FPGA

3.3 หลักการทำงานของส่วนFPGA

ในส่วนของ FPGA นี้จะมีอยู่ 2 ส่วนที่สำคัญคือส่วนของภาค Control และ ภาคCounter แล้วนำทั้งสองส่วนนี้มาประกอบเข้าด้วยกัน โดยหลักการแล้วเราต้องการให้ส่วนของ FPGA นี้ทำหน้าที่นับผลต่างและแสดงผลต่างที่นับได้ในแต่ละภาครับที่รับจากการเคาะ โปรเจคเตอร์ได้ทีละเวลาต่างกัน ก็เพื่อนำส่วนของผลต่างทางเวลานี้ไปเป็นข้อมูลในการคำนวณหาพิกัด X, Y ต่อไปในส่วนของ การคำนวณ

ภาค Control

ภาคนี้จะทำหน้าที่ในการควบคุมการเริ่มนับและการหยุดนับของ ภาคCounter ในแต่ละส่วนของภาครับ ดังนั้นภาคนี้จึงต้องทำหน้าที่ผลิตสัญญาณการเริ่มต้นการนับ และสัญญาณการหยุดนับ โดยจะต้องสัมพันธ์กับสัญญาณ Rx ในแต่ละตัวแต่ละภาครับที่จะรับคลื่นได้จากภาคส่ง

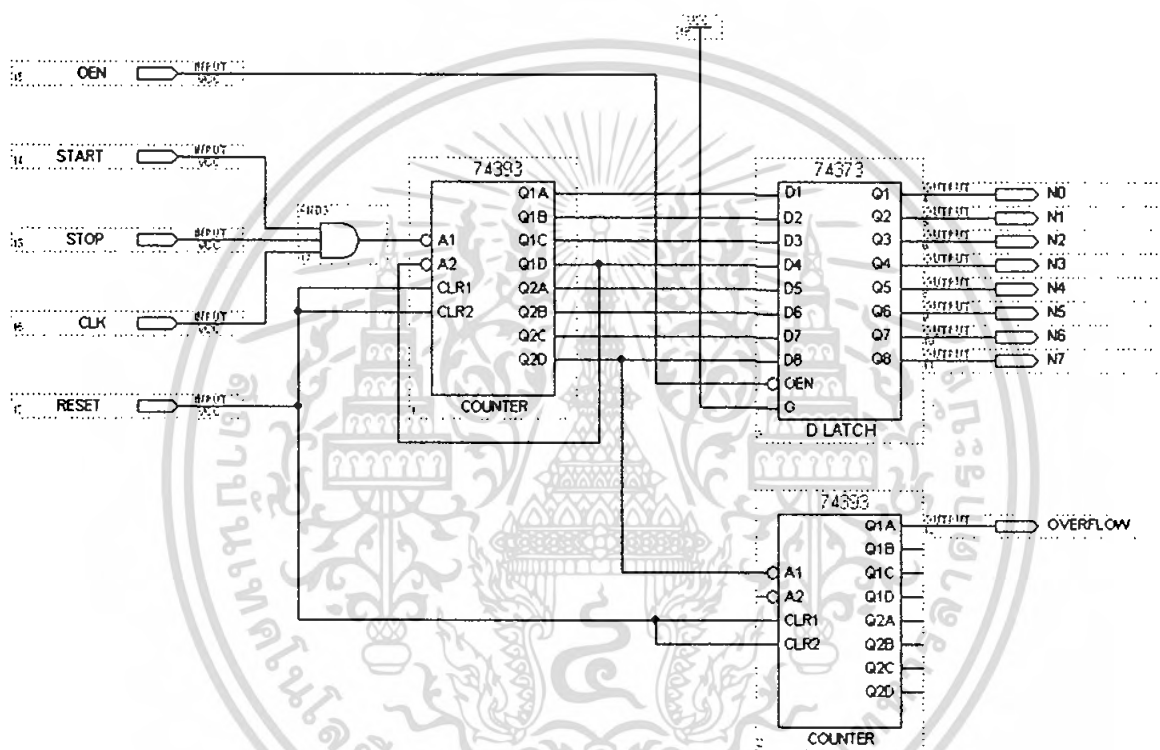


รูปที่ 3.2 แสดงภาพวงจรการทำงานภายใน Block Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาค Counter

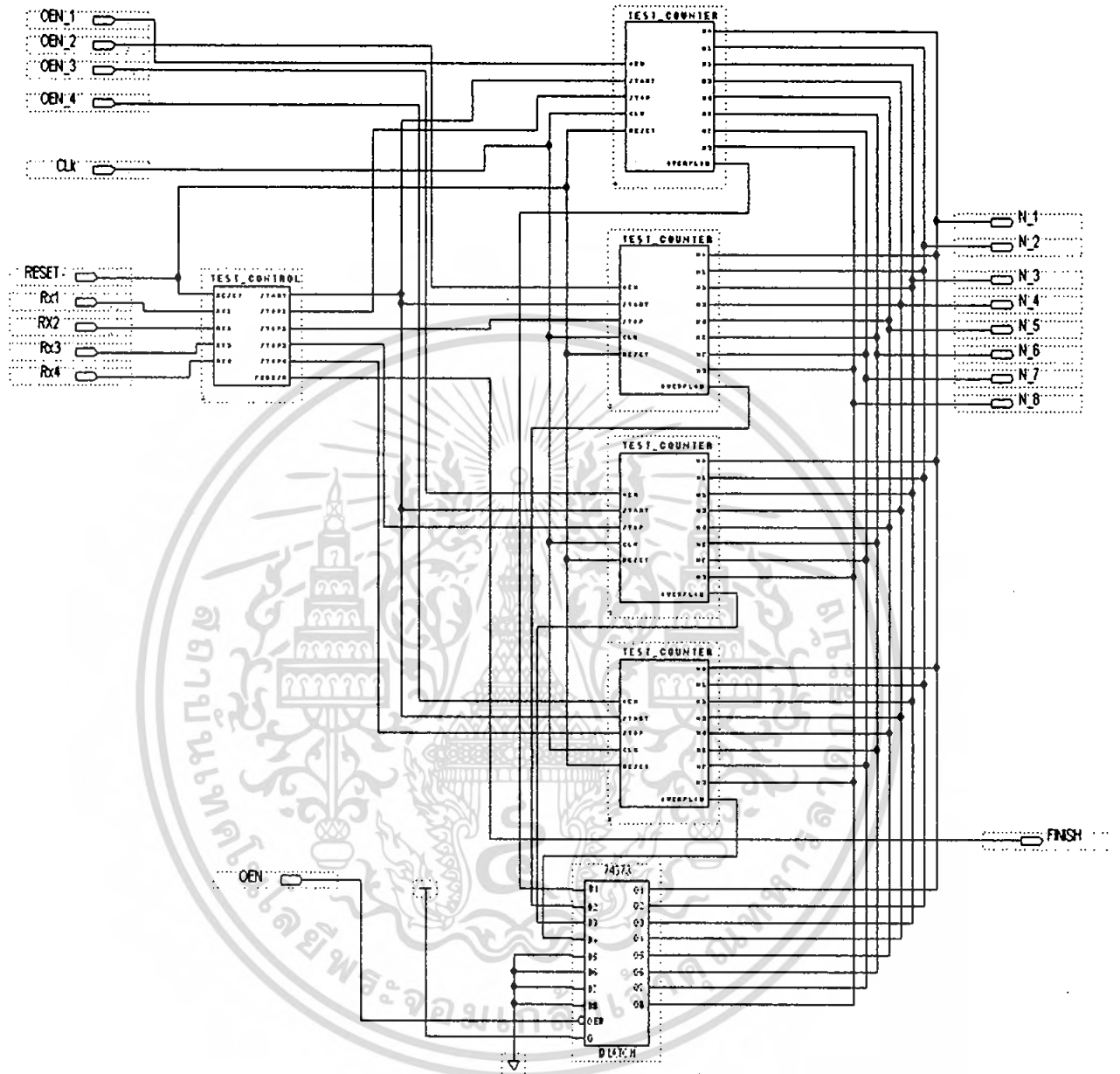
ภาคนี้จะทำหน้าที่หลักคือ นับผลต่างทางเวลาของสัญญาณ Rx ในแต่ละตัวในรูปของสัญญาณนาฬิกา ภาคนี้จะประกอบด้วย IC สองชนิดด้วยกันคือ 74393 คือ IC Counter ชนิดพื้นฐาน 16 8บิต และ IC 74373 คือ IC ดีแลตซ์ ดังนั้นภาคนี้จะนับได้ 0-255 ถ้าเกินนี้ก็จะส่งสัญญาณ Overflow ออกมาเสมือนว่าภาครับคลื่นภาคนั้นไม่สามารถที่จะรับคลื่นที่ส่งมาจากภาคส่งได้



รูปที่ 3.3 แสดงภาพวงจรการทำงานภายใน Block Counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

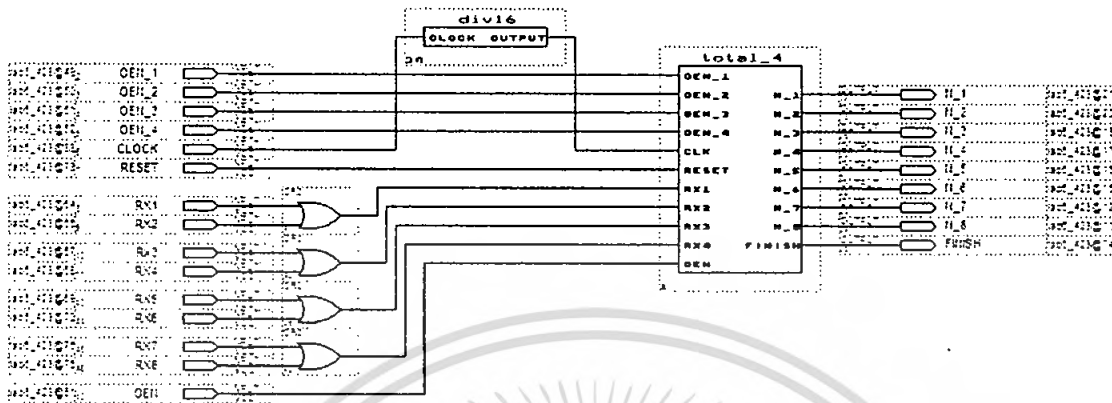
วงจร FPGA



รูปที่ 3.4 แสดงภาพวงจรในส่วนของ FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรวม FPGA



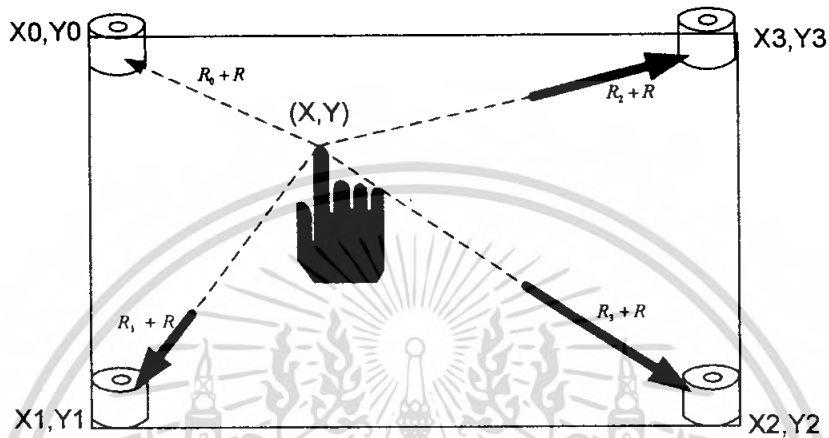
รูปที่ 3.5 แสดงภาพวงจรทั้งหมดในส่วนของ FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจรภาคประมวลผล

3.4.1 การคำนวณหาพิกัดตำแหน่ง

การคำนวณหาพิกัดตำแหน่ง อาศัยหลักการทางคณิตศาสตร์ โดยมีวิธีการคำนวณดังนี้



รูปที่ 3.6 แสดงการวางตำแหน่งของตัวรับ

สมมติว่าจุดที่เคาะอยู่ที่ตำแหน่ง (x,y) และเราทำการวางตำแหน่งของตัวรับทั้ง 4 ตัว ตามจุดที่กำหนดที่ทราบค่าแน่นอนคือจุด $(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$ ดังรูป เมื่อมีการเคาะโปรเจกเตอร์สัญญาณคลื่นจะถูกส่งไปยังตัวรับทั้ง 4 ตัว เปรียบเสมือนการโยนก้อนหินลงในน้ำ ระยะเวลาที่คลื่นเดินทางจากตำแหน่งเคาะ ไปยังตัวรับแต่ละตัว ไม่เท่ากันขึ้นอยู่กับขนาดระยะทาง ระยะเวลาของสัญญาณจะถูกส่งไปที่บอร์ด FPGA เพื่อนับค่าเวลาที่ต่างกัน และส่งไปที่ MCS-51 เพื่อใช้คำนวณ

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

การคำนวณหาพิคัด

1. การคำนวณเพื่อให้ได้ค่า R_0, R_1, R_2, R_3

เมื่อเราได้ค่าที่เคาน์เตอร์แต่ละตัวนับได้ (n) เราสามารถหาเวลาที่ใช้ได้ (t) จากสูตรต่อไปนี้

$$T = N/f \quad \text{----- (3.1)}$$

โดย T คือ เวลาที่แตกต่างกันระหว่างตัวรับตัวแรกกับตัวรับตัวนั้น

N คือ จำนวนที่เคาน์เตอร์นับได้

f คือ ความถี่ที่เคาน์เตอร์ใช้นับ

และเราสามารถหาค่า R_0, R_1, R_2, R_3 ได้จากสูตรต่อไปนี้

$$\begin{aligned} \text{จาก} \quad R &= VT && \text{----- (3.2)} \\ \text{จะได้} \quad R_0 &= VN_0/f \\ R_1 &= VN_1/f \\ R_2 &= VN_2/f \\ R_3 &= VN_3/f \end{aligned}$$

โดย V คือ ความเร็วของคลื่นในวัสดุที่เป็นตัวกลางในการเคาะ

R_0, R_1, R_2, R_3 คือ รัศมีที่แตกต่างกันระหว่างตัวรับตัวนั้นกับตัวรับที่อยู่ใกล้ตัวส่งมากที่สุด

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

2. การคำนวณหาค่า X, Y

โดยจุด $(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$ เป็นตำแหน่งที่เราทราบค่าแน่นอน

จากรูปที่ 3.6 เราสามารถคำนวณได้ดังนี้

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดรูปสมการด้านบน จะได้

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

$$X^2 - 2X_1X + X_1^2 + Y^2 - 2Y_1Y + Y_1^2 = R^2 + 2R_1R + R_1^2$$

$$X^2 - 2X_2X + X_2^2 + Y^2 - 2Y_2Y + Y_2^2 = R^2 + 2R_2R + R_2^2$$

$$X^2 - 2X_3X + X_3^2 + Y^2 - 2Y_3Y + Y_3^2 = R^2 + 2R_3R + R_3^2$$

$$X^2 + Y^2 - R^2 = 2X_0X + 2Y_0Y + 2R_0R - X_0^2 - Y_0^2 + R_0^2 \dots\dots\dots(5)$$

$$X^2 + Y^2 - R^2 = 2X_1X + 2Y_1Y + 2R_1R - X_1^2 - Y_1^2 + R_1^2 \dots\dots\dots(6)$$

$$X^2 + Y^2 - R^2 = 2X_2X + 2Y_2Y + 2R_2R - X_2^2 - Y_2^2 + R_2^2 \dots\dots\dots(7)$$

$$X^2 + Y^2 - R^2 = 2X_3X + 2Y_3Y + 2R_3R - X_3^2 - Y_3^2 + R_3^2 \dots\dots\dots(8)$$

นำสมการที่ (5) = (6) ; (5) = (7) ; (5) = (8)

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

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

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

กำหนดให้

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

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

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

$$D_1 = X_0^2 - X_1^2 + Y_0^2 - Y_1^2 + R_1^2 - R_0^2 \quad D_2 = X_0^2 - X_2^2 + Y_0^2 - Y_2^2 + R_2^2 - R_0^2$$

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

จะได้ 3 สมการที่ไม่ทราบค่าตัวแปร 3 ตัวคือ X, Y, R

$$A_1X + B_1Y + C_1R = D_1$$

$$A_2X + B_2Y + C_2R = D_2$$

$$A_3X + B_3Y + C_3R = D_3$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

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

จากสมการสามารถคำนวณค่าที่ได้ในระนาบ 2 มิติคือ

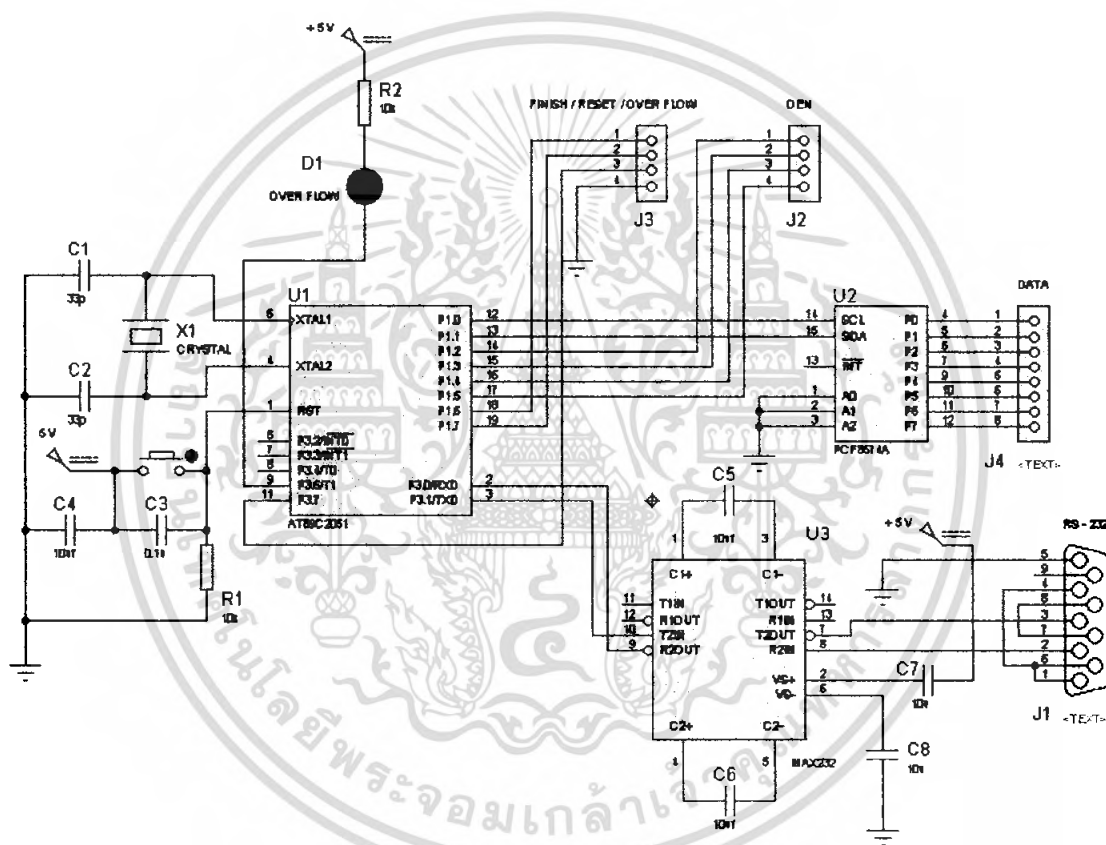
X, Y : พิกัดของตำแหน่งที่เกาะ

R : เป็นระยะทางที่คลื่นใช้เคลื่อนที่ถึงตัวรับตัวแรกซึ่งเป็นระยะที่สั้นที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 หลักการทำงานของ MCS-51.

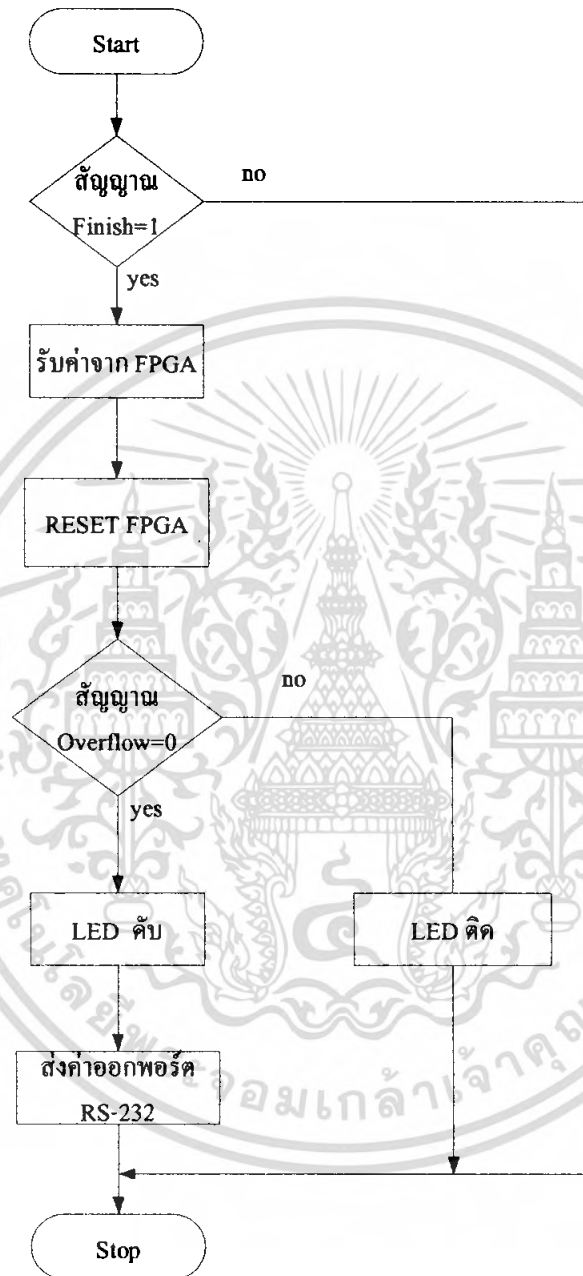
MCS-51 จะคอยเช็คสัญญาณ Finish อยู่ตลอดเวลาเป็น High หรือเป็น Low ถ้าเป็น Low ให้สิ้นสุดการทำงาน แต่ถ้าเป็น High ให้ทำการรับค่า N1, N2, N3, N4, Overflow จาก FPGA แล้ว จากนั้นทำการ Reset FPGA และเช็คสัญญาณ Overflow ว่าเป็น High หรือเป็น Low ถ้าเป็น High ให้ LED ติด แล้วสิ้นสุดการทำงาน แต่ถ้าเป็น Low ให้ LED ดับ และส่งค่า ค่า N1, N2, N3, N4 ออกทางพ्लीต RS-232



รูปที่ 3.7 แสดงการต่อใช้งาน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow chart การทำงานของ MCS-51

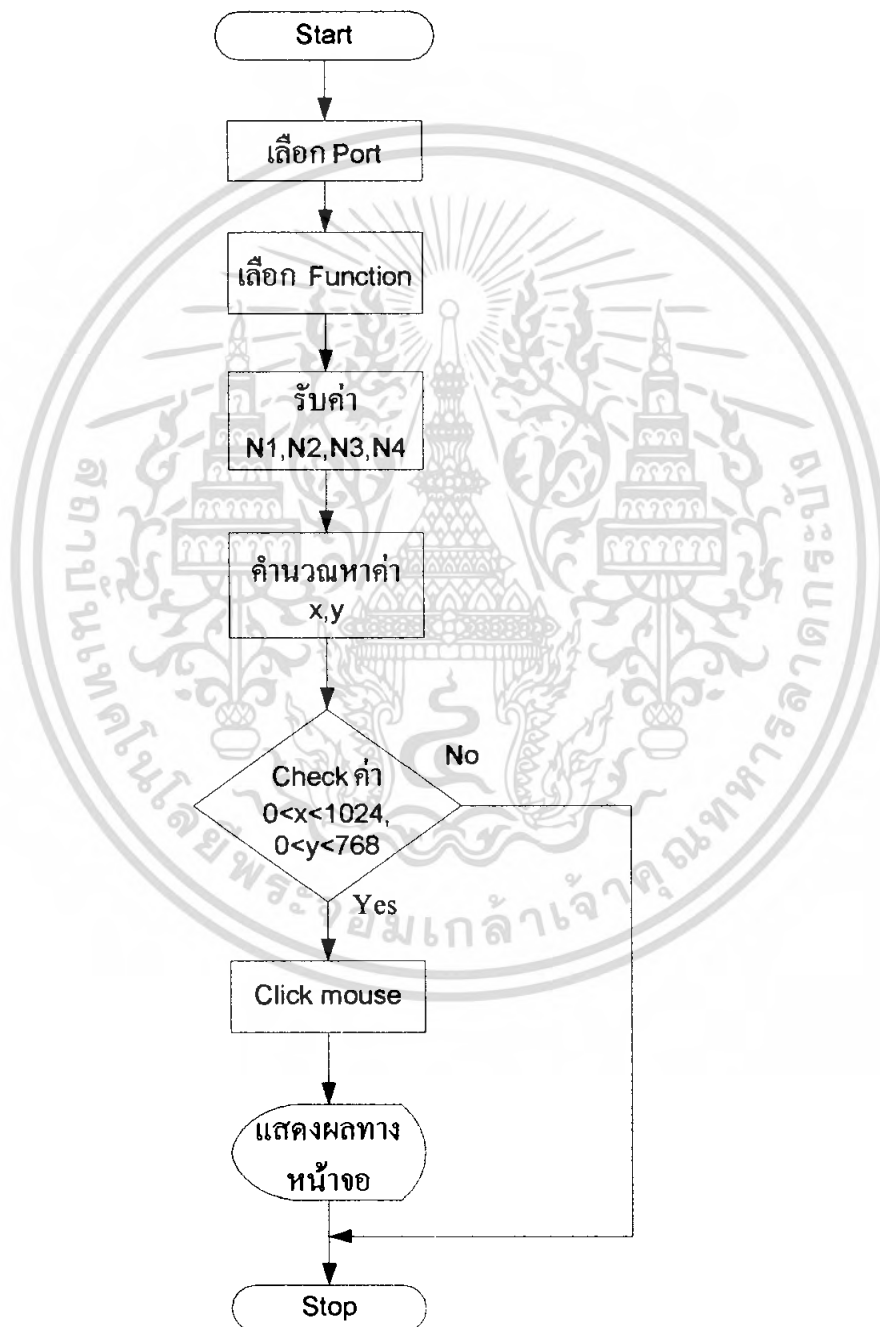


รูปที่ 3.8 แสดง Flow chart การทำงานของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 หลักการทำงานของ Visual Basic

Visual Basic จะทำการรับค่าที่ส่งมาจาก MCS-51 เข้ามาทาง com port แล้วทำการนำค่า N1, N2, N3, N4 มาคำนวณหาค่า X, Y แล้วทำการเช็คค่า X และ Y ว่า $0 < X < 1024$, $0 < Y < 768$ ถ้า X และ Y ตรงตามเงื่อนไขให้ทำการคลิกเมาท์ และแสดงผลทางหน้าจอ



รูปที่ 3.9 แสดง Flow chart การทำงานของ Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทที่ 4 เป็นการทดลอง และแสดงผลการทดลองในรูปแบบต่าง ๆ โดยแยกออกมาเป็นส่วนๆ ในการทดลองและบันทึกผล ซึ่งสามารถแยกออกเป็นข้อหัวดังนี้

1. ส่วนของตัวรับ
 - 1.1. เปรียบเทียบความไวและขนาดสัญญาณที่รับได้จากตัวรับชนิดต่างๆ
 - 1.2. เปรียบเทียบวัตถุที่ใช้เคาะ
 - 1.3. เปรียบเทียบตัวกลางที่ใช้เคาะ
 - 1.4. เปรียบเทียบสัญญาณที่มุมตำแหน่งต่างๆ รอบ เปียโซ
 - 1.5. เปรียบเทียบระยะทางที่ใช้เคาะ
2. ส่วนของวงจรภาครับสัญญาณเคาะ
 - 2.1. สัญญาณที่จุดต่างๆของวงจรภาครับสัญญาณจากการเคาะ
 - 2.2. สัญญาณเข้าที่พุดของวงจรภาครับเมื่อป้อนแรงดันอินพุตที่ระดับต่างๆ
 - 2.3. เปรียบเทียบค่าหน่วงเวลาของสัญญาณเข้าที่พุดของวงจรทั้ง 4 จุด
3. ส่วนของ FPGA ในภาคต่างๆ
 - 3.1 ผลการ Simulate ส่วนของ Block Control
 - 3.2 ผลการ Simulate ส่วนของ Block Counter
 - 3.3 ผลการ Simulate วงจรทั้งหมดของ FPGA
4. ส่วนของ MCS-51
 - 4.1 ทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8บิต เข้าสู่คอมพิวเตอร์ โดยผ่าน MCS-51
5. ส่วนของโปรแกรม Visual Basic 6.0 เพื่อควบคุมเมาส์
 - 5.1. เมื่อป้อนค่าตำแหน่ง X , Y เพื่อแทนค่าจำลองการป้อนค่าจากการนับใน FPGA
 - 5.2 เมื่อป้อนค่าด้วยจำนวน N เพื่อแทนค่าจำลองการป้อนค่าจากการนับใน FPGA

4.1 ส่วนของตัวรับ

4.1.1 เปรียบเทียบความไวและขนาดสัญญาณที่รับได้จากตัวรับสัญญาณที่ได้จากการเกาะชนิดต่างๆ

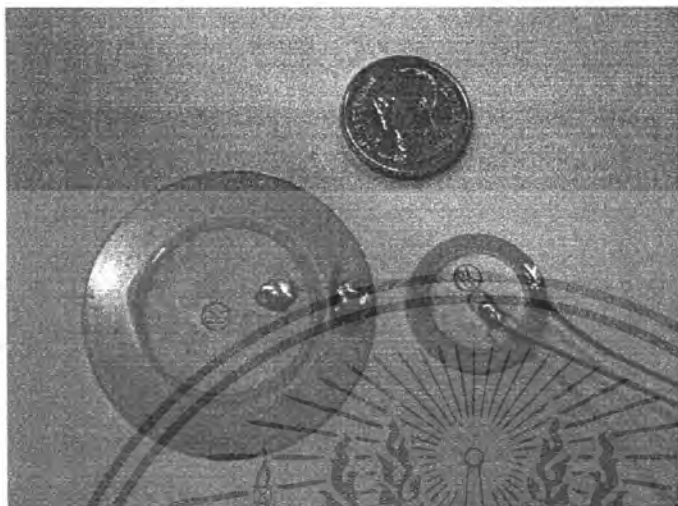
4.1.1.1 ลำโพงเล็ก



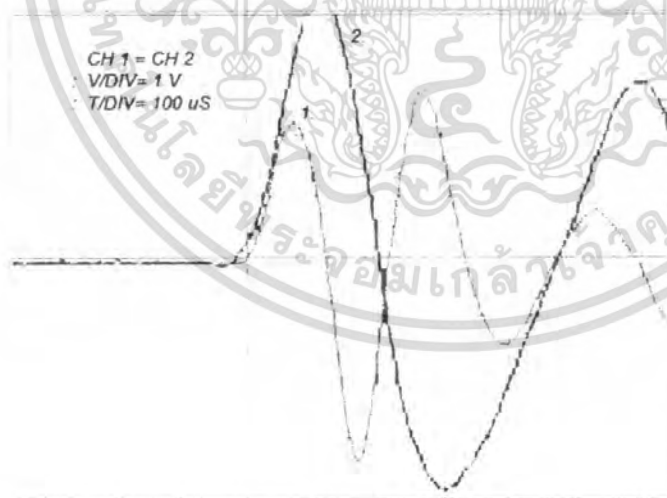
รูปที่ 4.2 สัญญาณที่ได้จากลำโพงเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 เปียโซขนาดเล็กและเปียโซขนาดใหญ่แบบเปลี่ยน



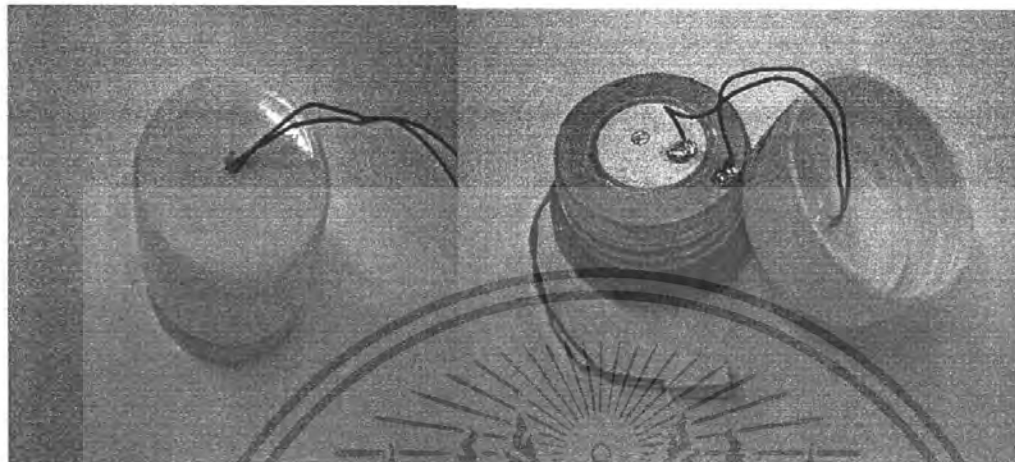
รูปที่ 4.3 เปียโซขนาดเล็กและเปียโซขนาดใหญ่แบบเปลี่ยน



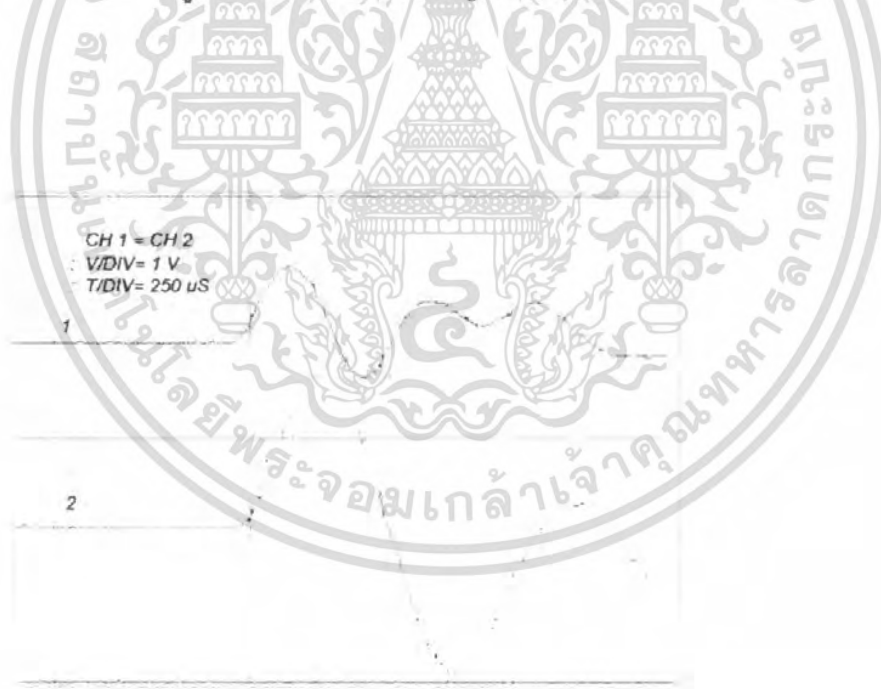
รูปที่ 4.4 สัญญาณที่ได้ CH1. เปียโซขนาดเล็ก CH2. เปียโซขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3 เปียโซขนาดใหญ่แบบมีฝาครอบ



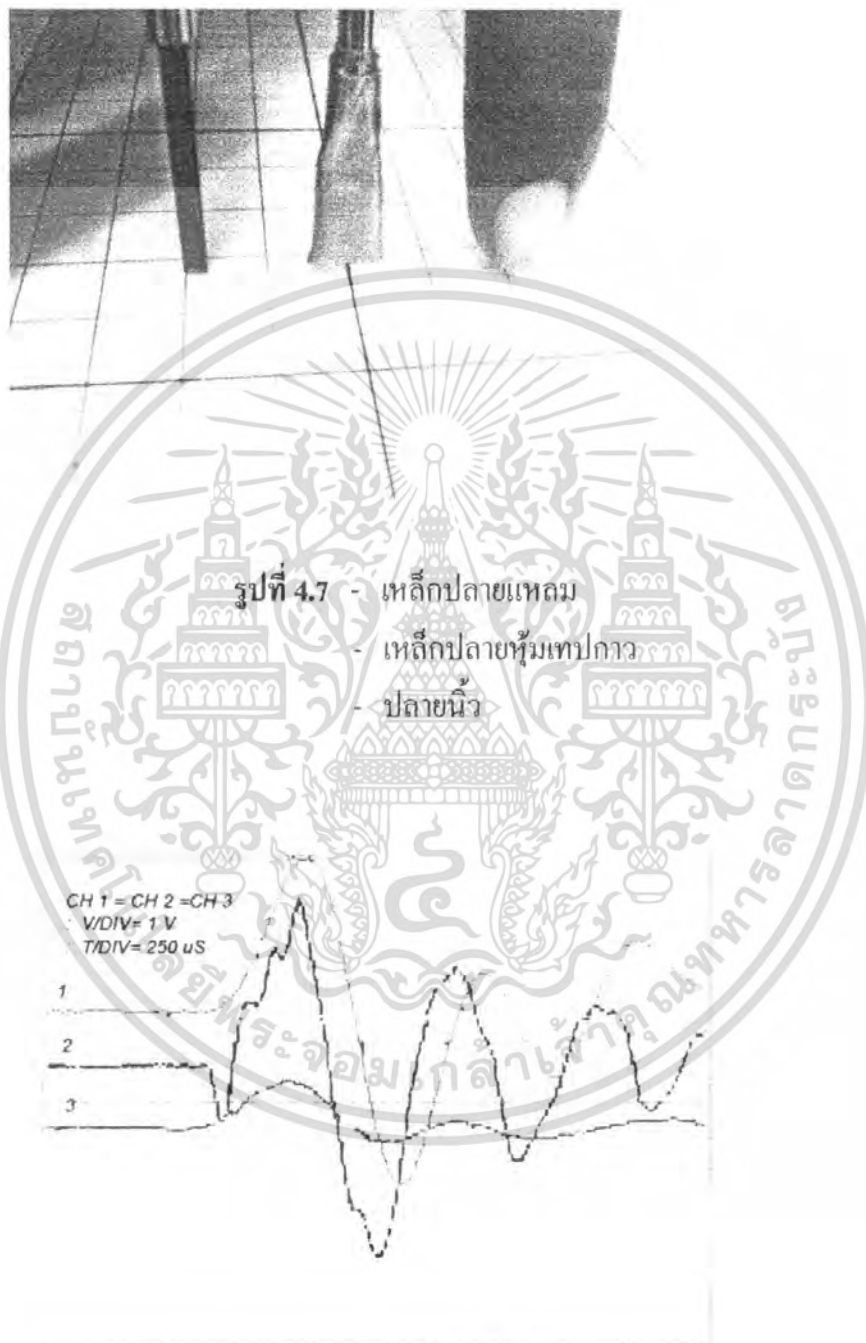
รูปที่ 4.5 เปียโซขนาดใหญ่แบบมีฝาครอบ



รูปที่ 4.6 สัญญาณที่ได้ CH1. เปียโซแบบเปลือย CH2.เปียโซ แบบมีฝาครอบ ใช้นิ้วเคาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 เปรียบเทียบวัตถุที่ใช้เกาะ

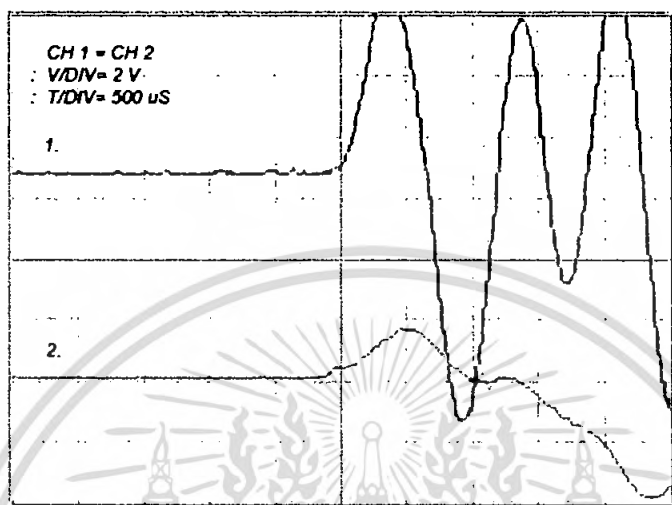


รูปที่ 4.8 สัญญาณจาก CH1. เหล็กปลายหุ้มเทปกาว
CH2. เหล็กปลายแหลม
CH3. ปลายไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

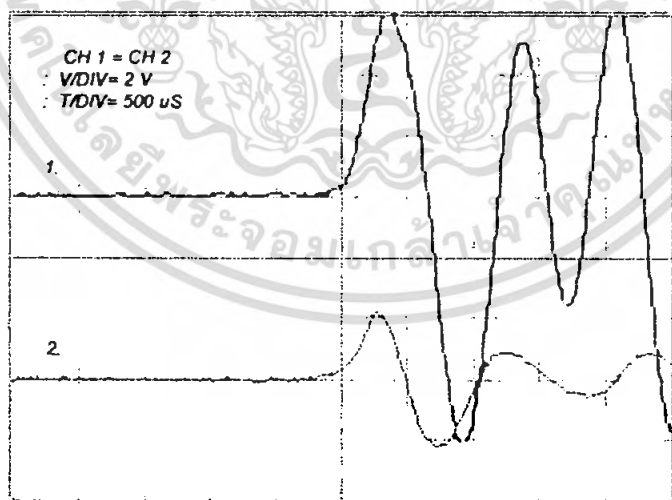
4.1.3 เปรียบเทียบตัวกลางที่ใช้เกาะ

4.1.3.1 แผ่นกระจกกับแผ่นพีวีเจอร์บอร์ด



รูปที่ 4.9 สัญญาณจาก CH1. แผ่นกระจก CH2. แผ่นพีวีเจอร์บอร์ด

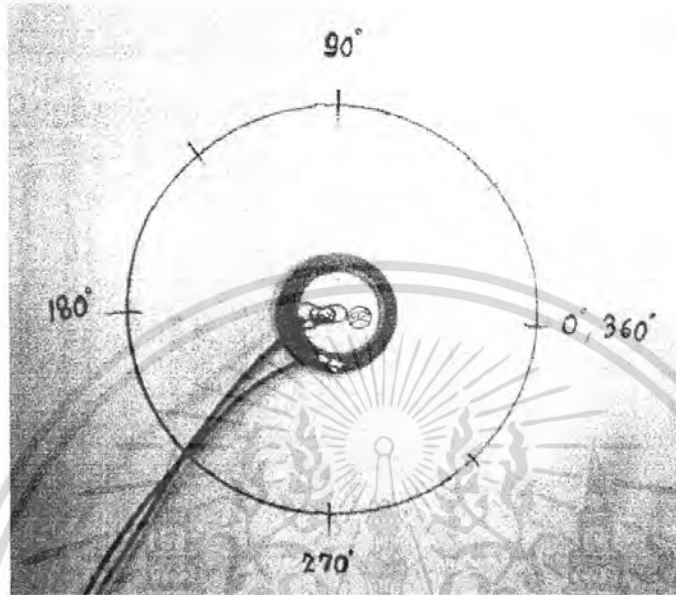
4.1.3.2 แผ่นกระจกกับแผ่นไม้ (โต๊ะ)



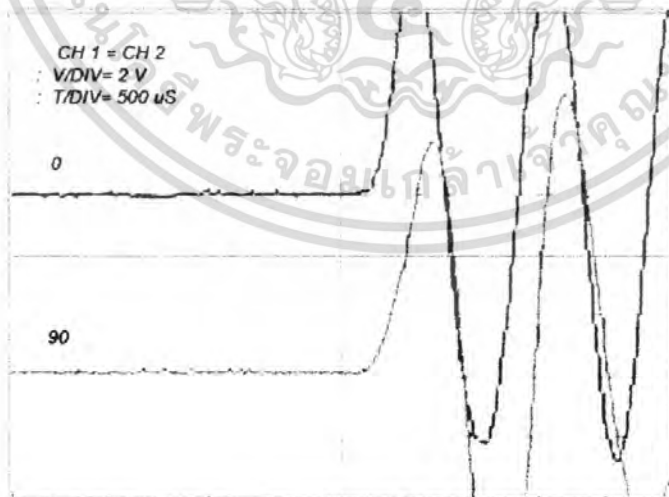
รูปที่ 4.10 สัญญาณจาก CH1.แผ่นกระจก CH2.แผ่นไม้ (โต๊ะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 เปรียบเทียบสัญญาณที่มุมตำแหน่งต่างๆ รอบๆ เปียโซ

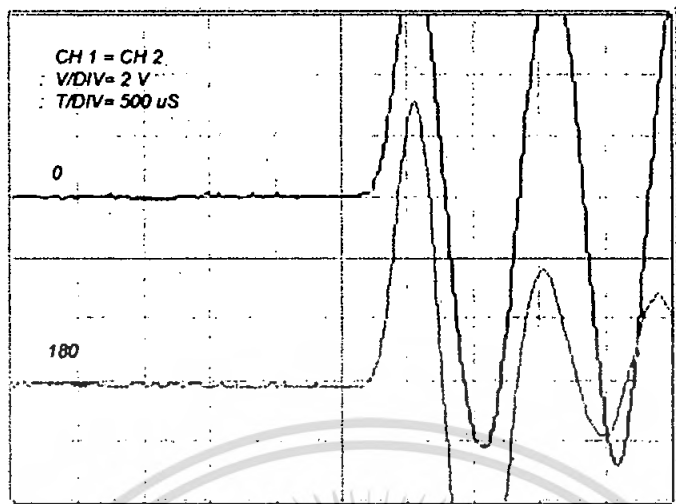


รูปที่ 4.11 ลักษณะการเคาะ

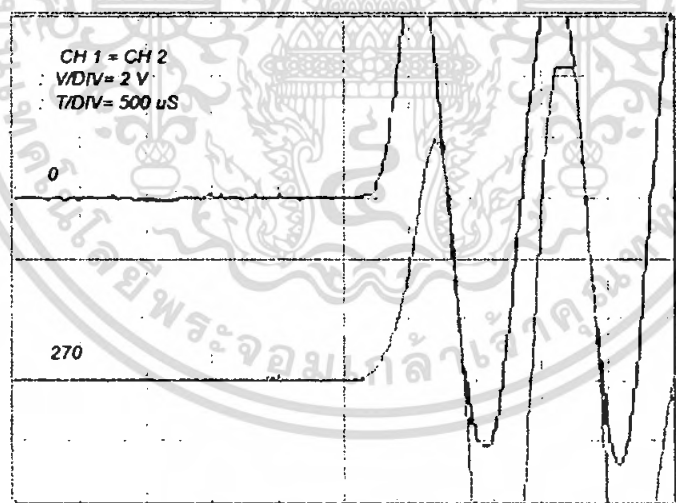


รูปที่ 4.12 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



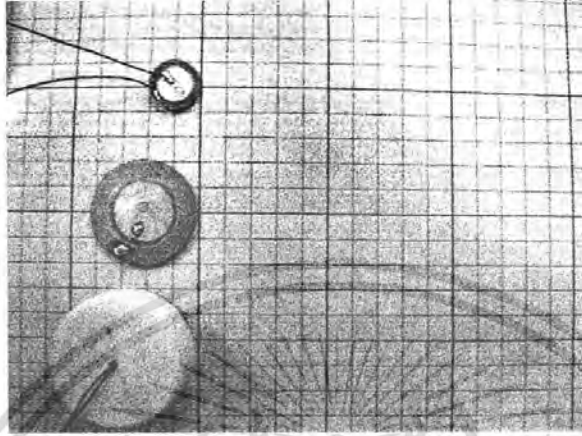
รูปที่ 4.13 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 180 องศา



รูปที่ 4.14 เปรียบเทียบสัญญาณที่มุม 0 องศา กับ 270 องศา

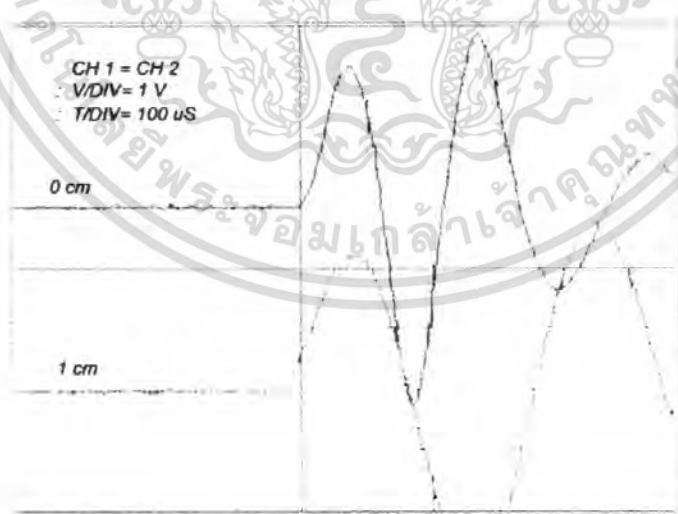
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 เปรียบเทียบระยะทางที่ใช้เกาะ



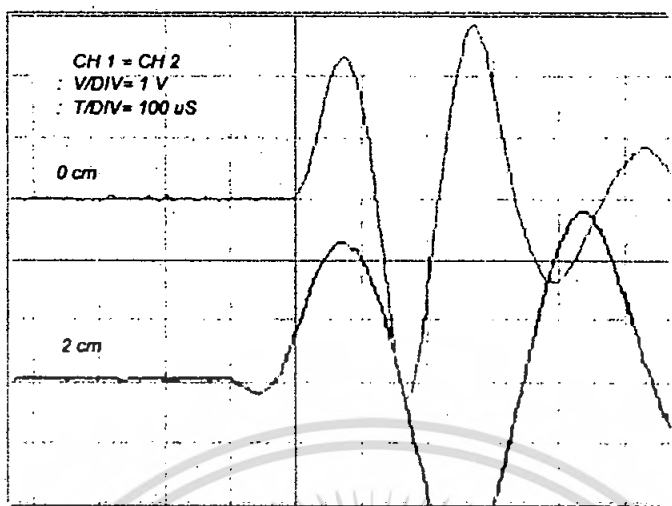
รูปที่ 4.15 ลักษณะระยะทางที่ใช้เกาะ

4.1.5.1 เปรียบเทียบความถี่ของสัญญาณที่ระยะต่างๆ

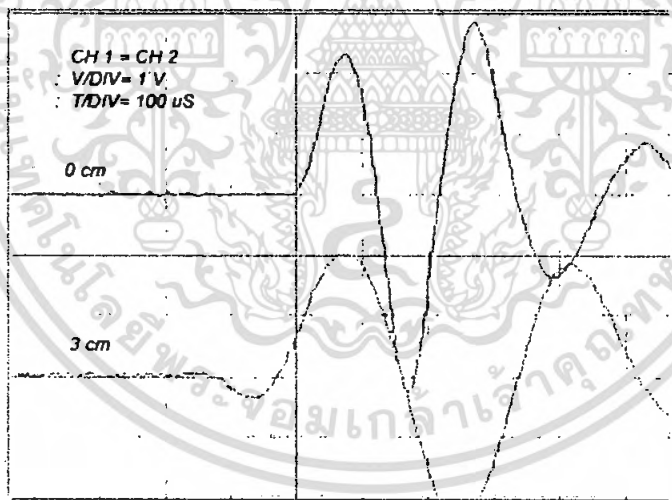


รูปที่ 4.16 สัญญาณที่ระยะ 0 กับ 1 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

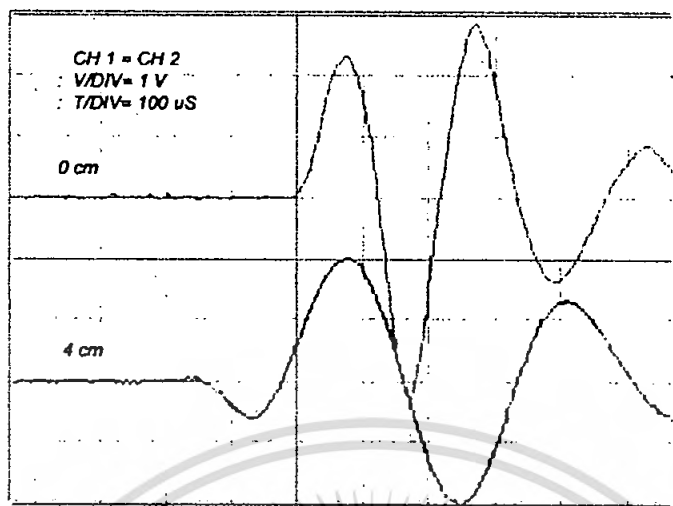


รูปที่ 4.17 สัญญาณที่ระยะ 0 กับ 2 cm

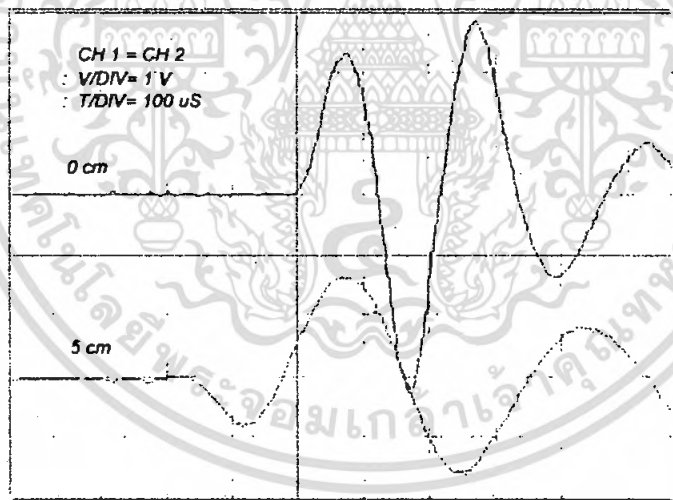


รูปที่ 4.18 สัญญาณที่ระยะ 0 กับ 3 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

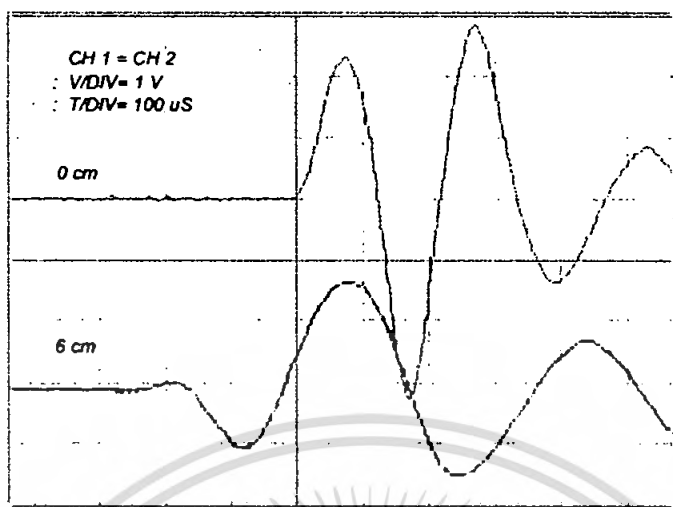


รูปที่ 4.19 สัญญาณที่ระยะ 0 ถึง 4 cm

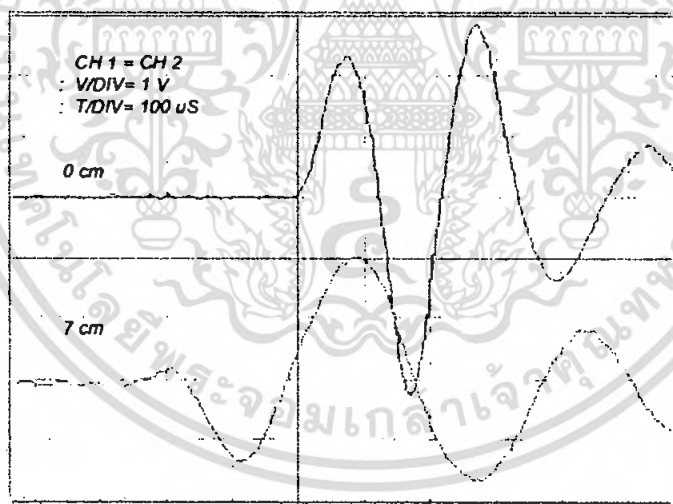


รูปที่ 4.20 สัญญาณที่ระยะ 0 ถึง 5 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

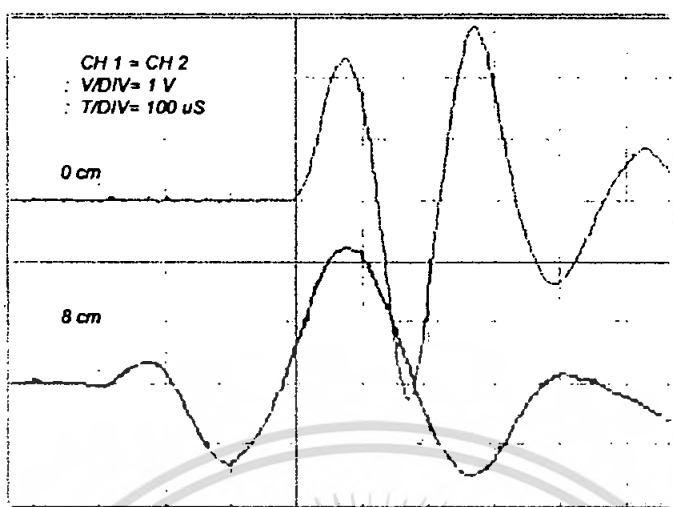


รูปที่ 4.21 สัญญาณที่ระยะ 0 กับ 6 cm

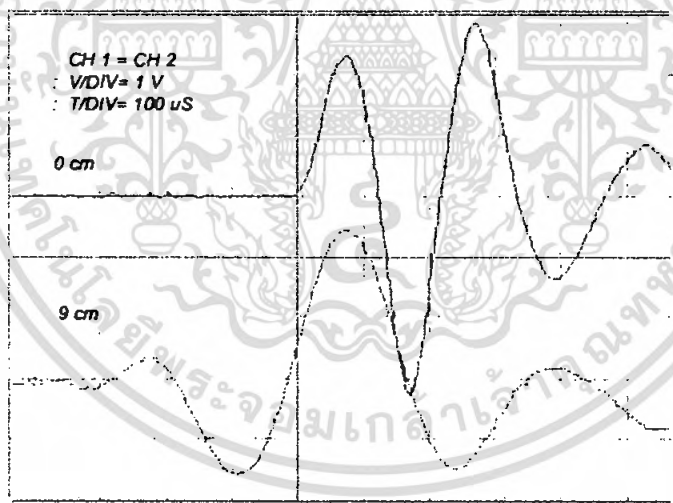


รูปที่ 4.22 สัญญาณที่ระยะ 0 กับ 7 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

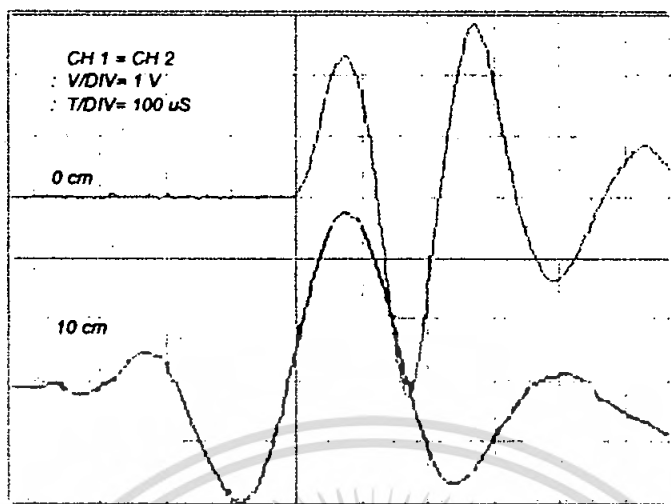


รูปที่ 4.23 สัญญาณที่ระยะ 0 กับ 8 cm



รูปที่ 4.24 สัญญาณที่ระยะ 0 กับ 9 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

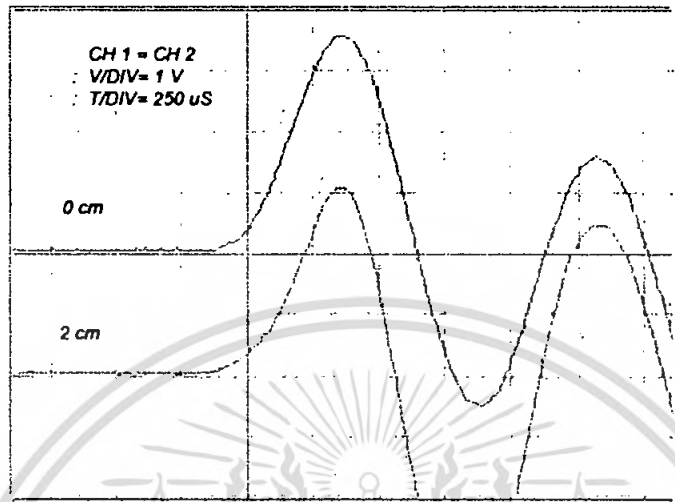


รูปที่ 4.25 สัญญาณที่ระยะ 0 กับ 10 cm

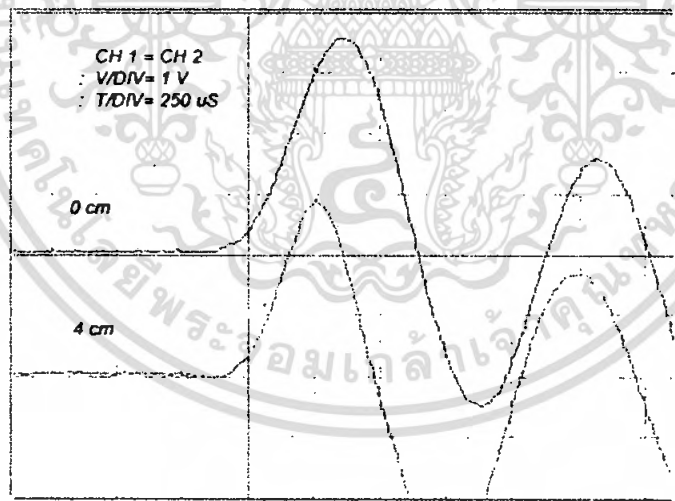


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5.2 เปรียบไซอิเล็กทริกขนาดใหญ่แบบมีฝากล่องที่ระยะต่างๆ

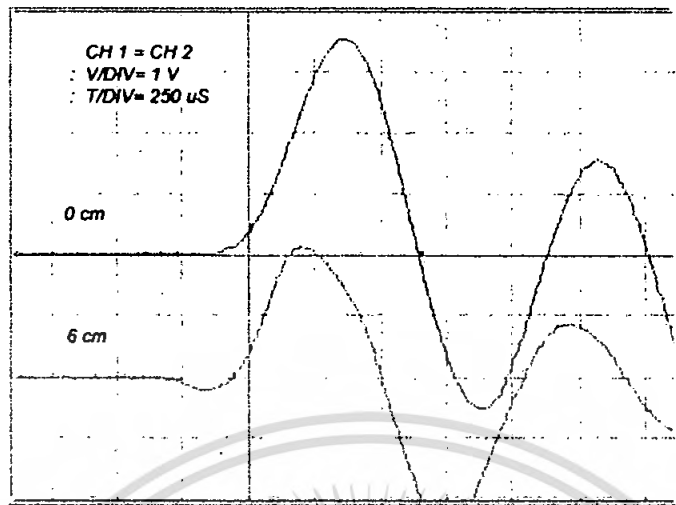


รูปที่ 4.26 สัญญาณที่ระยะ 0 กับ 2 cm

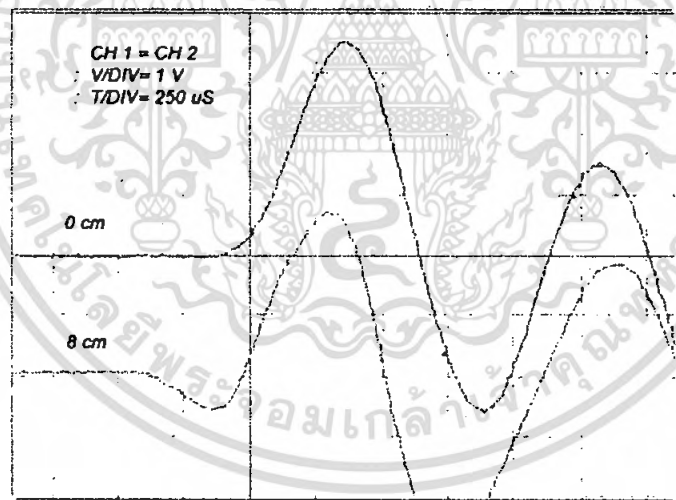


รูปที่ 4.27 สัญญาณที่ระยะ 0 กับ 4 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

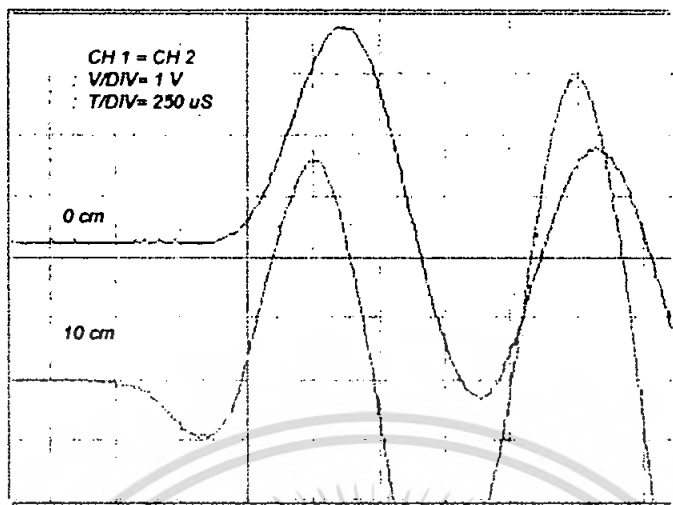


รูปที่ 4.28 สัญญาณที่ระยะ 0 กับ 6 cm

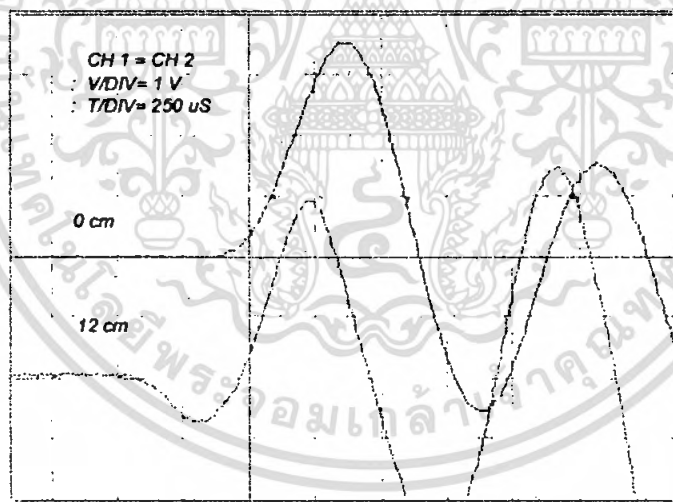


รูปที่ 4.29 สัญญาณที่ระยะ 0 กับ 8 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

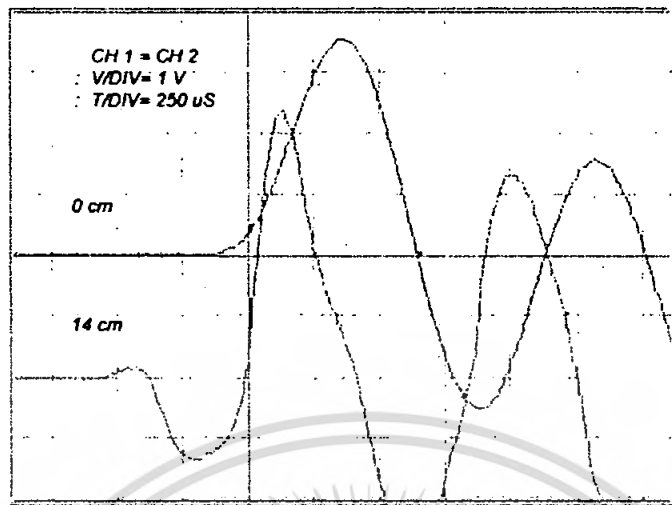


รูปที่ 4.30 สัญญาณที่ระยะ 0 กับ 10 cm

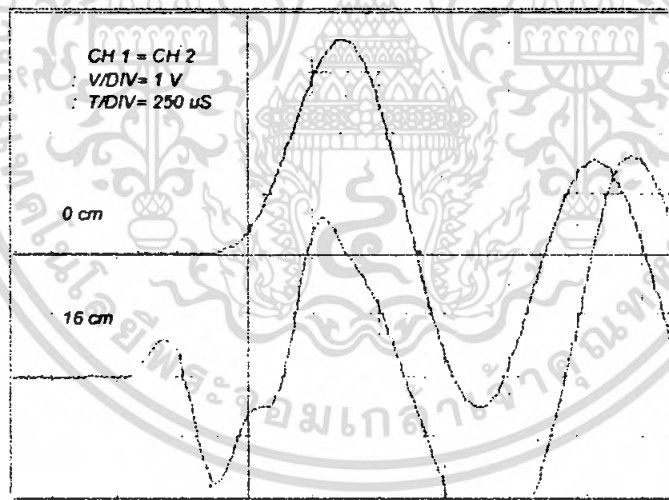


รูปที่ 4.31 สัญญาณที่ระยะ 0 กับ 12 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 สัญญาณที่ระยะ 0 กับ 14 cm

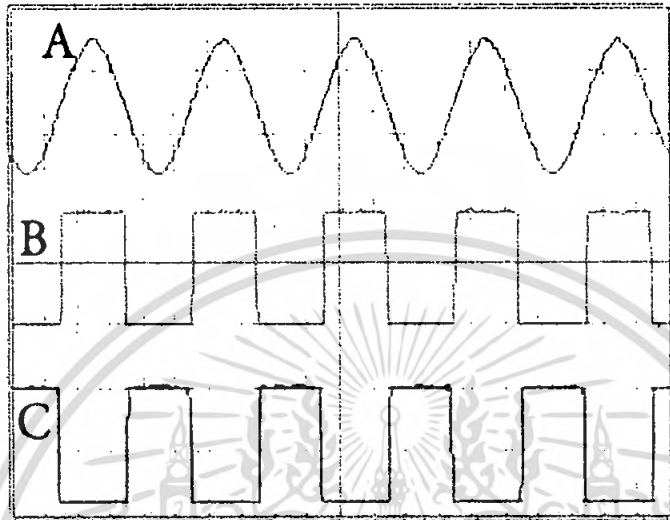


รูปที่ 4.33 สัญญาณที่ระยะ 0 กับ 16 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของวงจรภาครับ

4.2.1 สัญญาณที่จุดต่างๆของวงจรภาครับสัญญาณจากการเคาะ



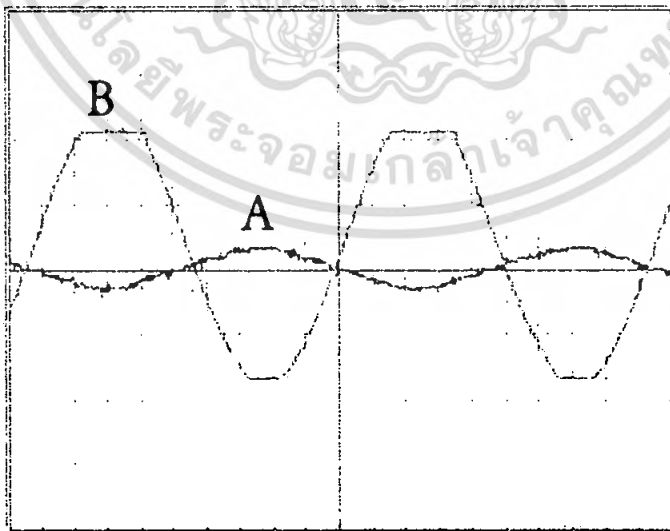
รูปที่ 4.34 แสดงสัญญาณอินพุต A และเอาต์พุตทั้งสองจุด B, C

A : INPUT ; V/Div = 500 mV , T/Div = 250 μ S

B : OUTPUT NONINVERTING AMP ; V/Div = 2 V , T/Div = 250 μ S

C : OUTPUT INVERTING AMP ; V/Div = 2 V , T/Div = 250 μ S

4.2.2 สัญญาณเอาต์พุตของวงจรภาครับเมื่อป้อนแรงดันอินพุตที่ระดับต่างๆ

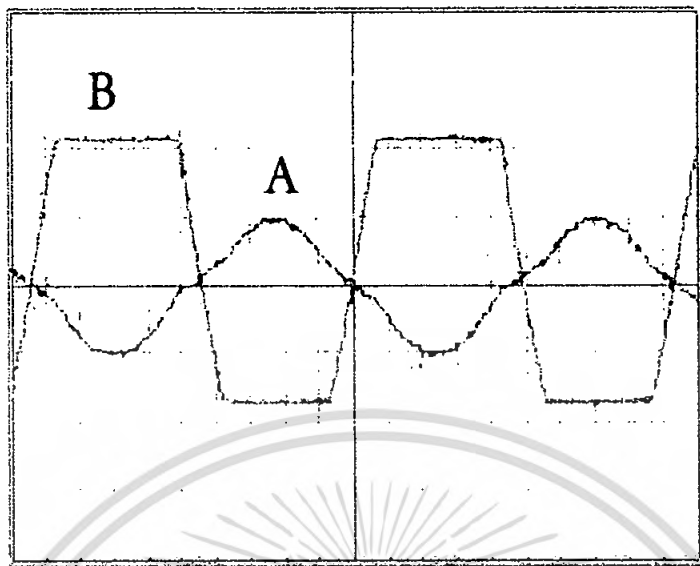


รูปที่ 4.35 สัญญาณเอาต์พุตของวงจรภาครับเมื่อป้อนแรงดันอินพุตที่ 50 mV

A : INPUT ; V/Div = 50 mV , T/Div = 250 μ S

B : OUTPUT ; V/Div = 2 V , T/Div = 250 μ S

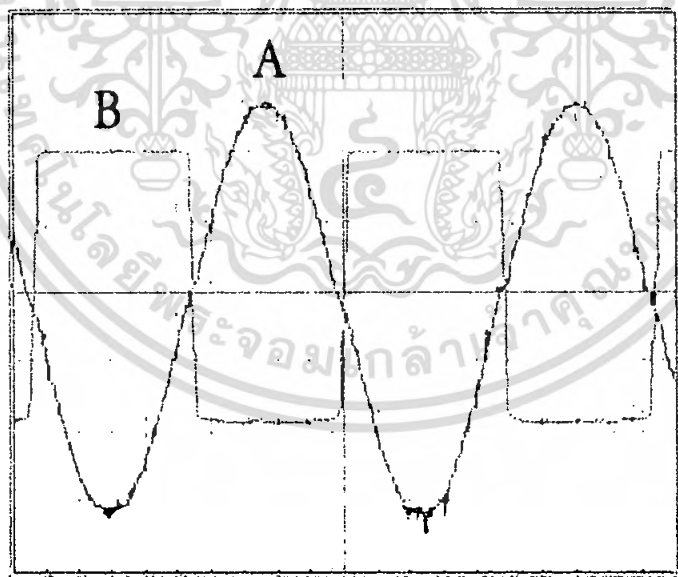
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.36 สัญญาณเข้าที่พุดของวงจรภาครับเมื่อป้อนแรงดันอินพุตที่ 100 mV

A : INPUT ; V/Div = 50 mV , T/Div = 250 uS

B : OUTPUT ; V/Div = 2 V , T/Div = 250 uS



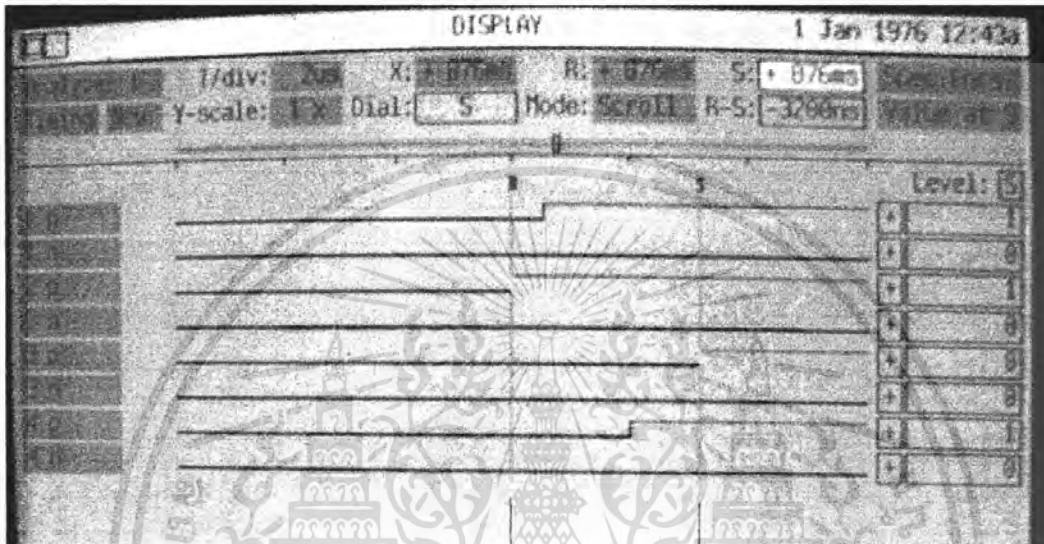
รูปที่ 4.37 สัญญาณเข้าที่พุดของวงจรภาครับเมื่อป้อนแรงดันอินพุตที่ 600 mV

A : INPUT ; V/Div = 100 mV , T/Div = 250 uS

B : OUTPUT ; V/Div = 2 V , T/Div = 250 uS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 เปรียบเทียบค่าหน่วยเวลาของสัญญาณเข้าที่พุดของวงจรทั้ง 4 ชุด เมื่อป้อนสัญญาณอินพุตที่มีขนาดเท่ากัน



รูปที่ 4.38 แสดงสัญญาณที่วัดได้จาก Logic Analyzer

โดย

$T/Div = 2 \text{ us}$

$R-S = 3200 \text{ nS}$ ซึ่งเป็นเวลาที่ต่างกันมากที่สุด ของวงจรแต่ละชุด

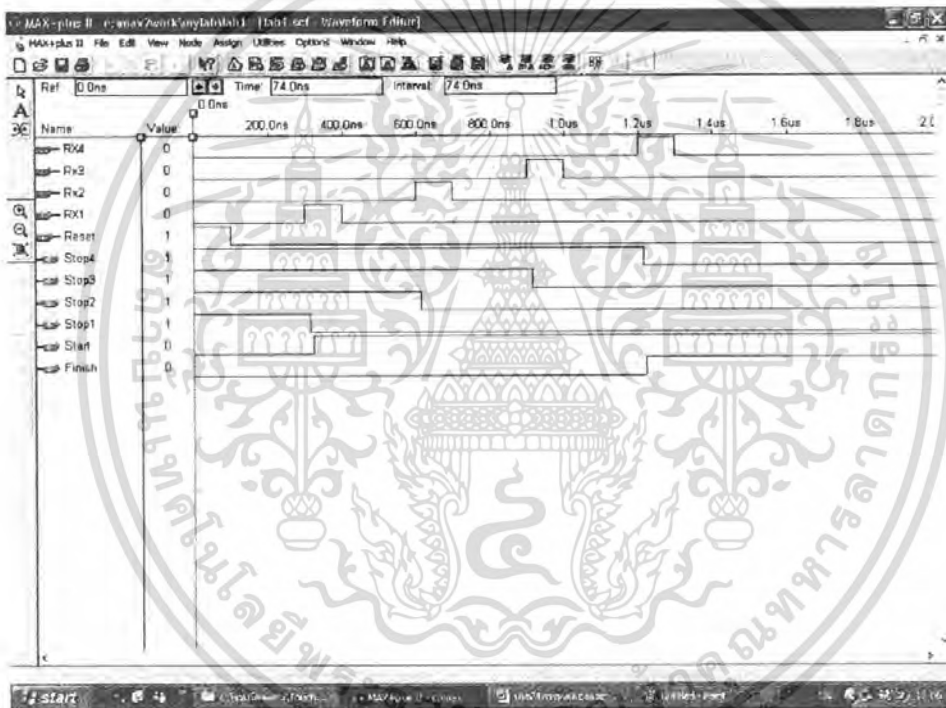
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 แสดงผลการ Simulate ของ FPGA

ในหัวข้อนี้ได้แสดงผลการ Simulate วงจรที่ได้ออกแบบไว้ในแต่ละส่วนการทำงาน ซึ่งส่วนหลักๆ ได้แก่ ส่วนของ Block Control และ Block Counter

4.3.1 ผลการ Simulate ส่วนของ Block Control

ในส่วนของ Block Control นี้เป็นส่วนที่เริ่มรับสัญญาณจากการเคาะ และไปควบคุมการทำงานในส่วน Block Counter ให้เริ่มนับและหยุดนับ

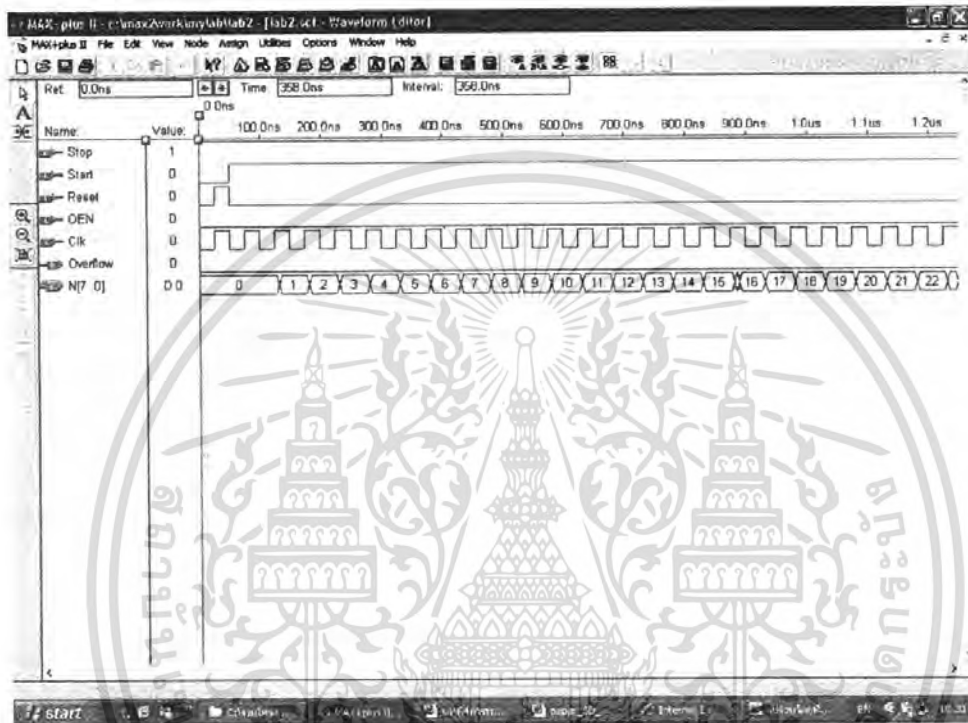


รูปที่ 4.39 แสดง Timing Diagram ของ Block Control ที่ได้จากการ Simulate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ผลการ Simulate ส่วนของ Block Counter

ในส่วนของ Block Counter นี้ได้ทำหน้าที่เป็นตัวจับเวลาของสัญญาณที่ส่งจากการเคาะจนกระทั่งตัวรับรับสัญญาณได้

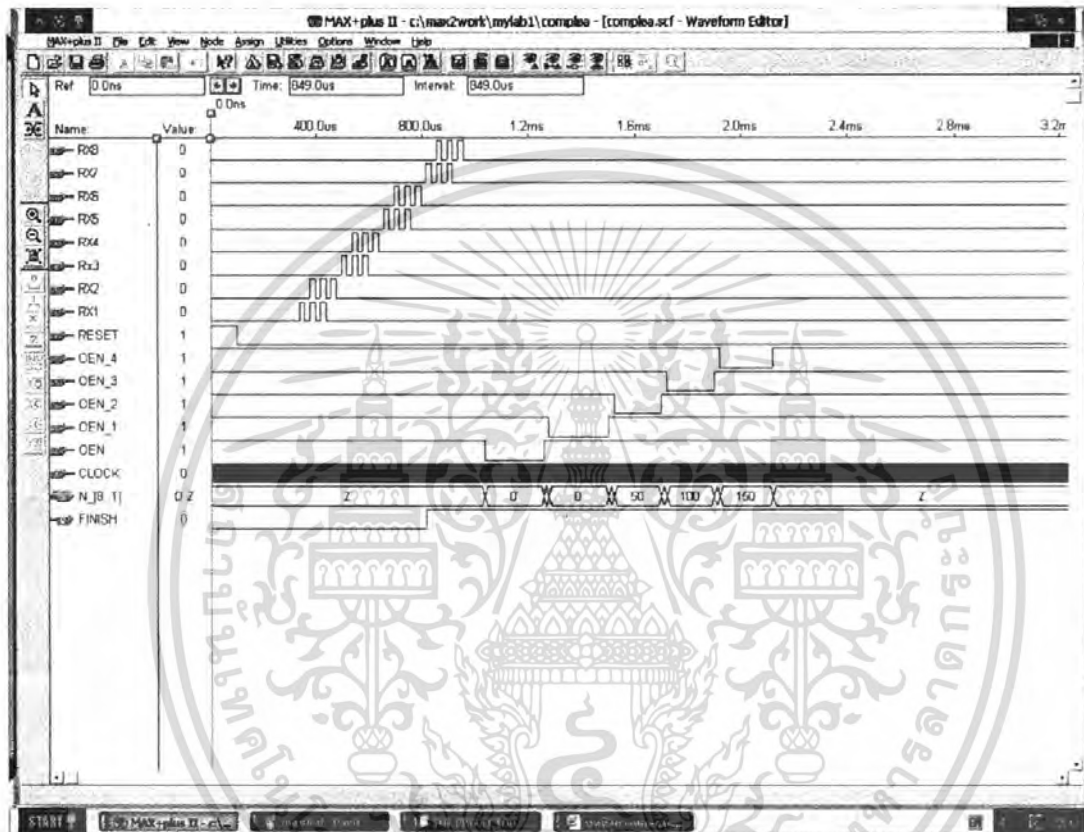


รูปที่ 4.40 แสดง Timing Diagram ของ Block Counter ที่ได้จากการ Simulate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลการ Simulate วงจรทั้งหมดของ FPGA

ในหัวข้อนี้ได้แสดงภาพวงจรการทำงานทุกๆ ส่วนของ FPGA ที่ได้นำมาประกอบกันอย่างสมบูรณ์และผลการ Simulate วงจรที่ได้ประกอบแล้วดังนี้

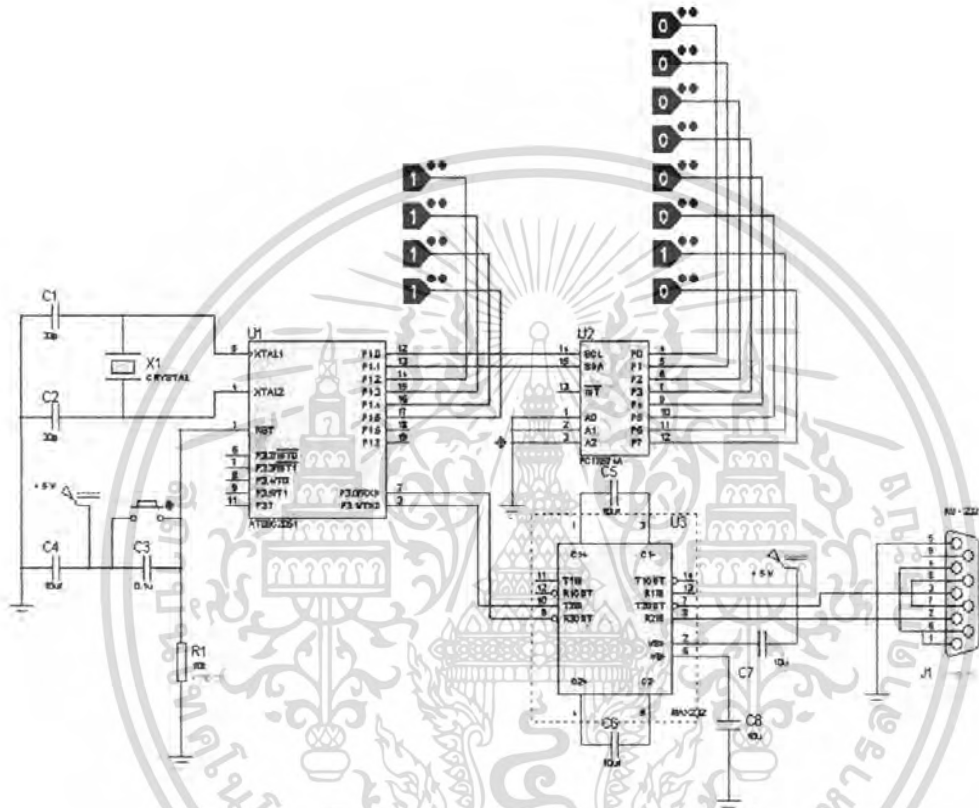


รูปที่ 4.41 แสดง Timing Diagram ของวงจรทั้งหมดของ FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

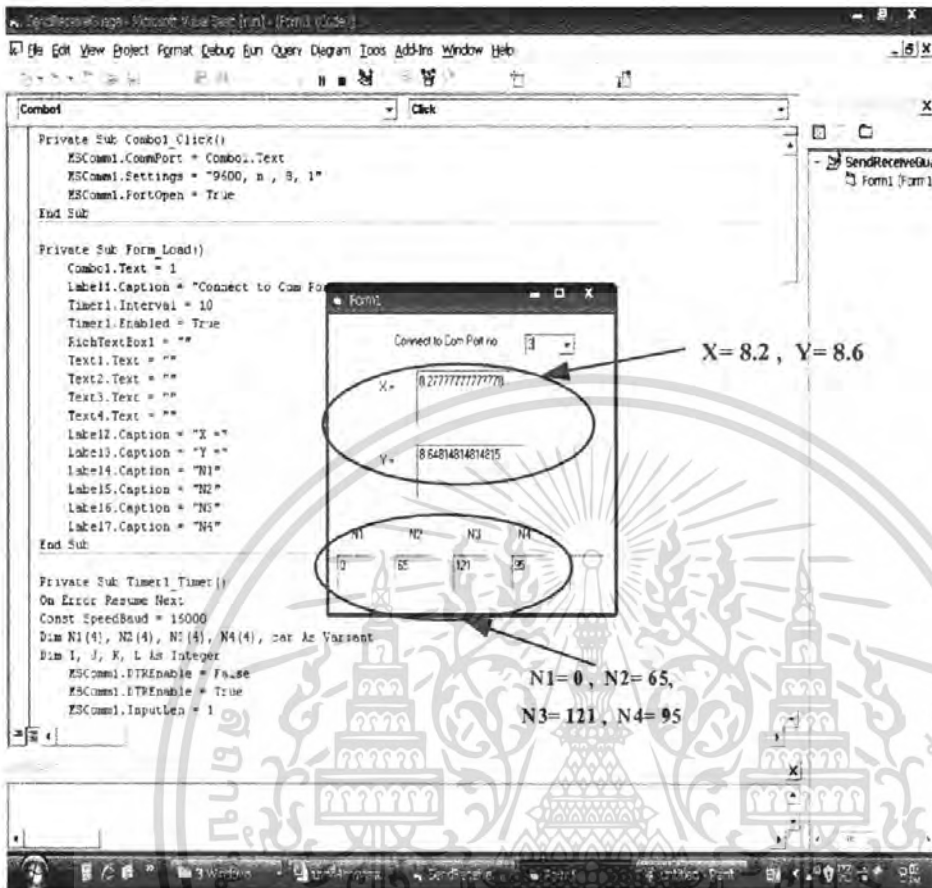
4.4 ส่วนของ MCS-51

4.4.1. ทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8 บิต เข้าสู่คอมพิวเตอร์โดยผ่าน MCS-51



รูปที่ 4.42 แสดงการทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8 บิต เข้าสู่คอมพิวเตอร์ โดยผ่าน MCS-51 ซึ่งเราได้ใส่ค่า $N1=0$, $N2=65$, $N3=121$, $N4=95$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.43 แสดงผลการทดลองส่งค่าข้อมูลจากคิฟสวิทซ์ 8 บิต เข้าสู่คอมพิวเตอร์
โดยผ่าน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ส่วนของ Visual Basic

ทดลองการทำงานของโปรแกรมจำลองการคลิกเมาส์

4.5.1 เมื่อป้อนค่าตำแหน่ง X, Y



รูปที่ 4.44 แสดงการป้อนค่า X,Y เพื่อคลิกซ้าย

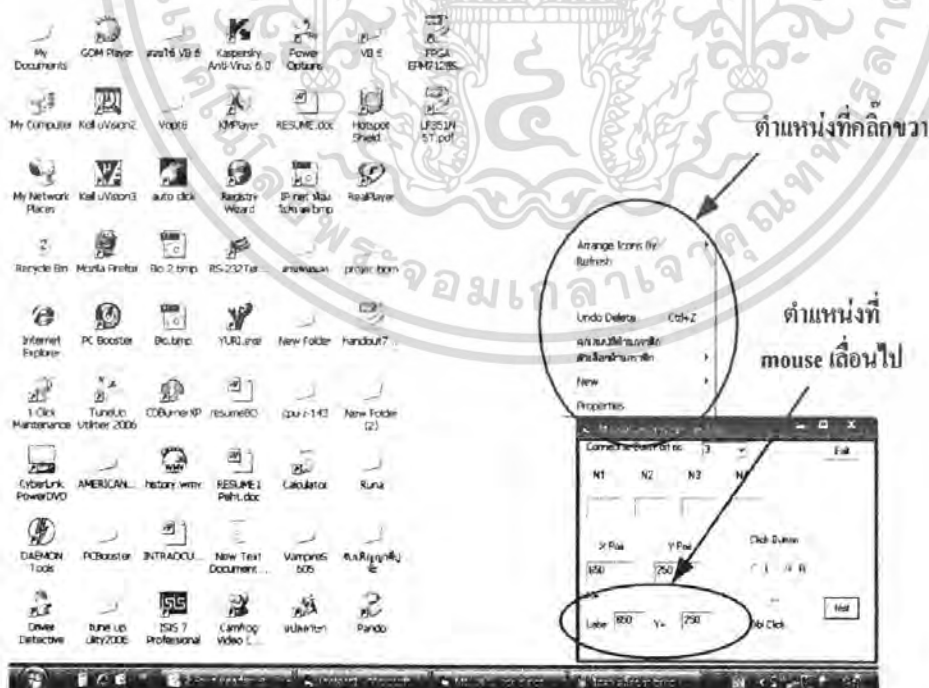


รูปที่ 4.45 ผลการป้อนค่า X,Y เพื่อคลิกซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

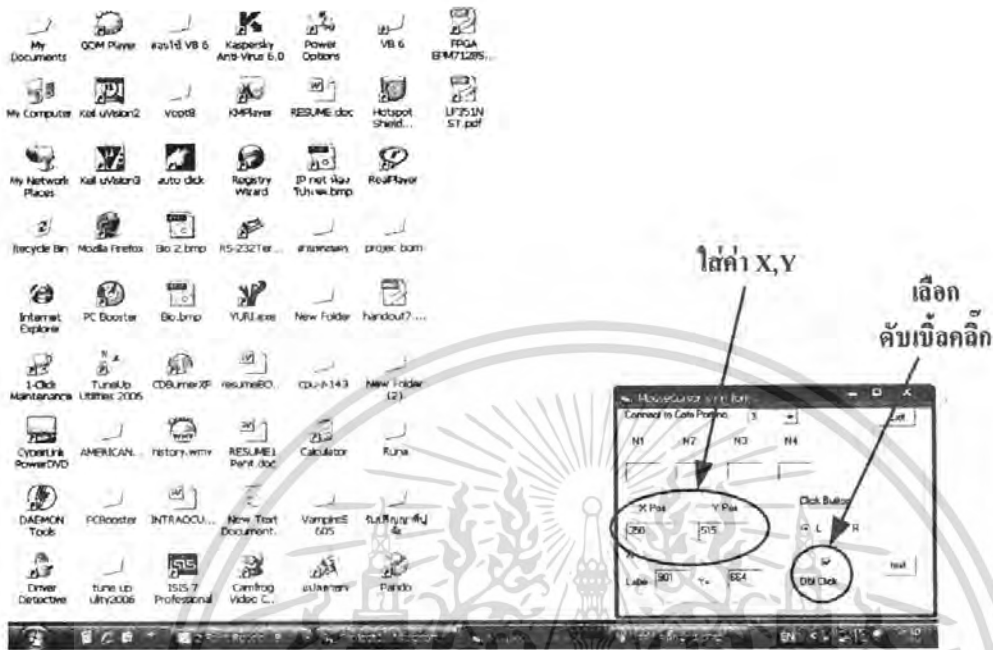


รูปที่ 4.46 แสดงการป้อนค่า X,Y เพื่อคลิกขวา

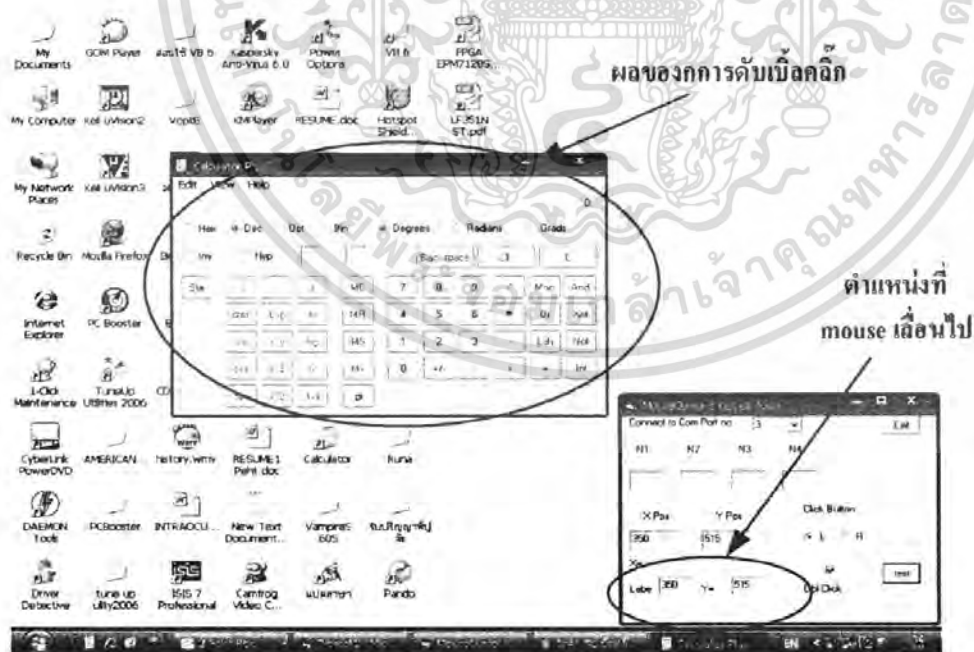


รูปที่ 4.47 ผลการป้อนค่า X,Y เพื่อคลิกขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



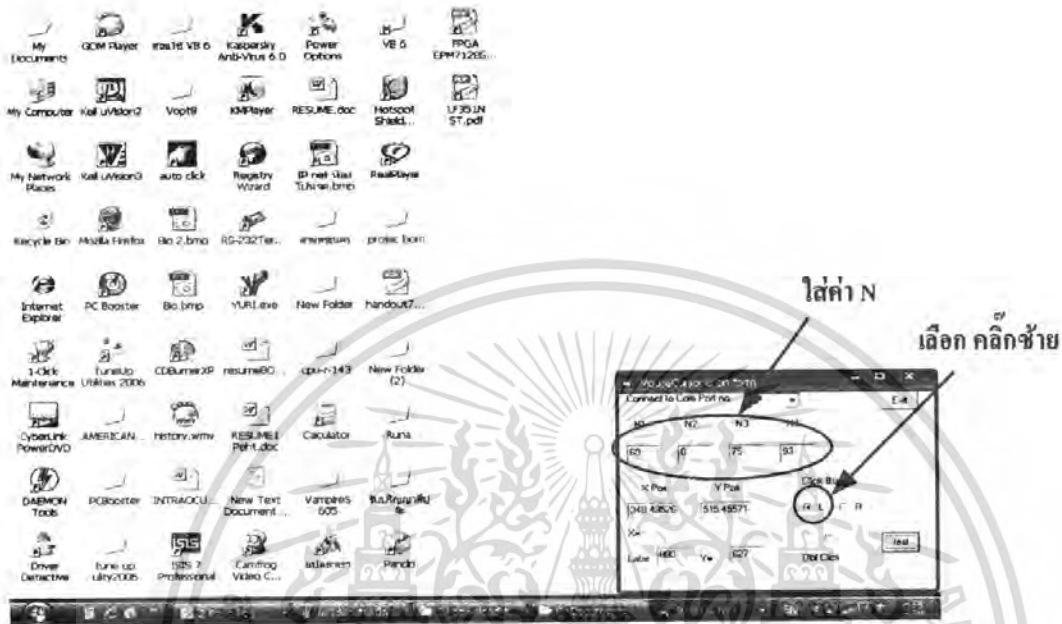
รูปที่ 4.48 แสดงการป้อนค่า X,Y เพื่อคลิกคีย์บอร์ด



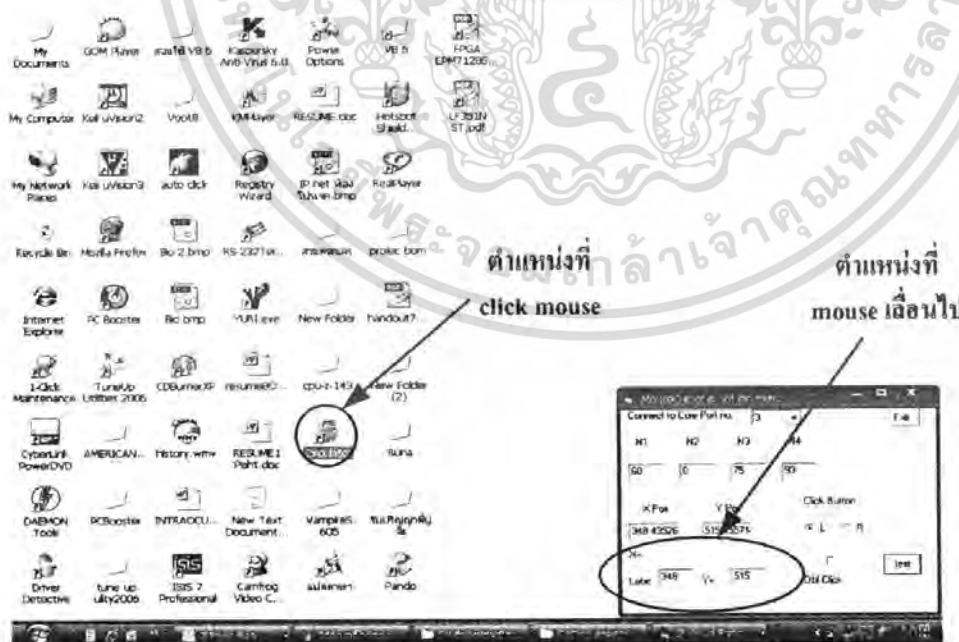
รูปที่ 4.49 ผลการป้อนค่า X,Y เพื่อคลิกคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 เมื่อป้อนค่าด้วยจำนวน N เพื่อแทนค่าจำลองการป้อนค่าจากการนับใน FPGA



รูปที่ 4.50 แสดงการป้อนค่า N เพื่อคลิกซ้าย

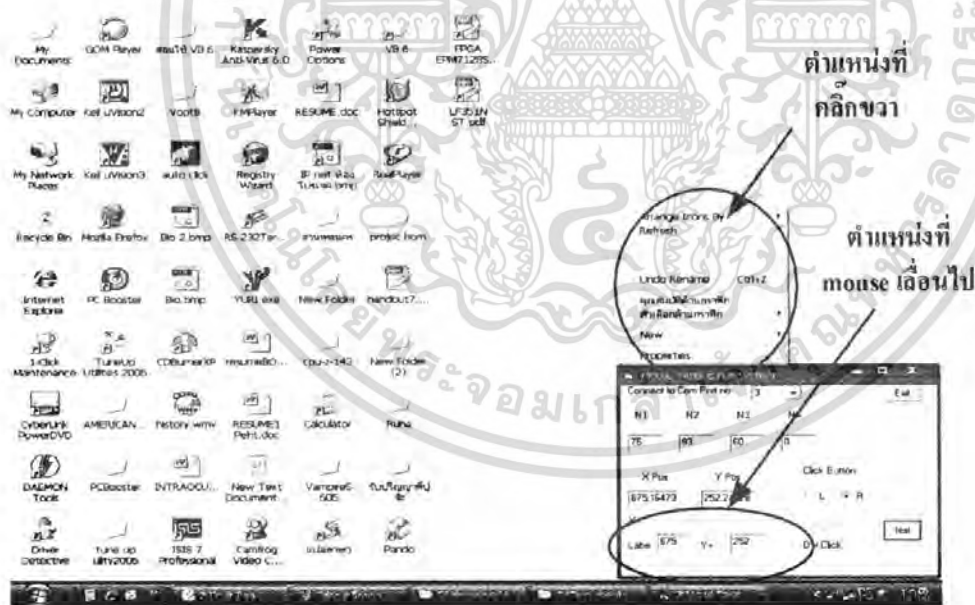


รูปที่ 4.51 ผลการป้อนค่า N เพื่อคลิกซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.52 แสดงการป้อนค่า N เพื่อคลิกขวา



รูปที่ 4.53 ผลการป้อนค่า N เพื่อคลิกขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.54 แสดงการป้อนค่า N เพื่อคลิกคัมเบตคลิก

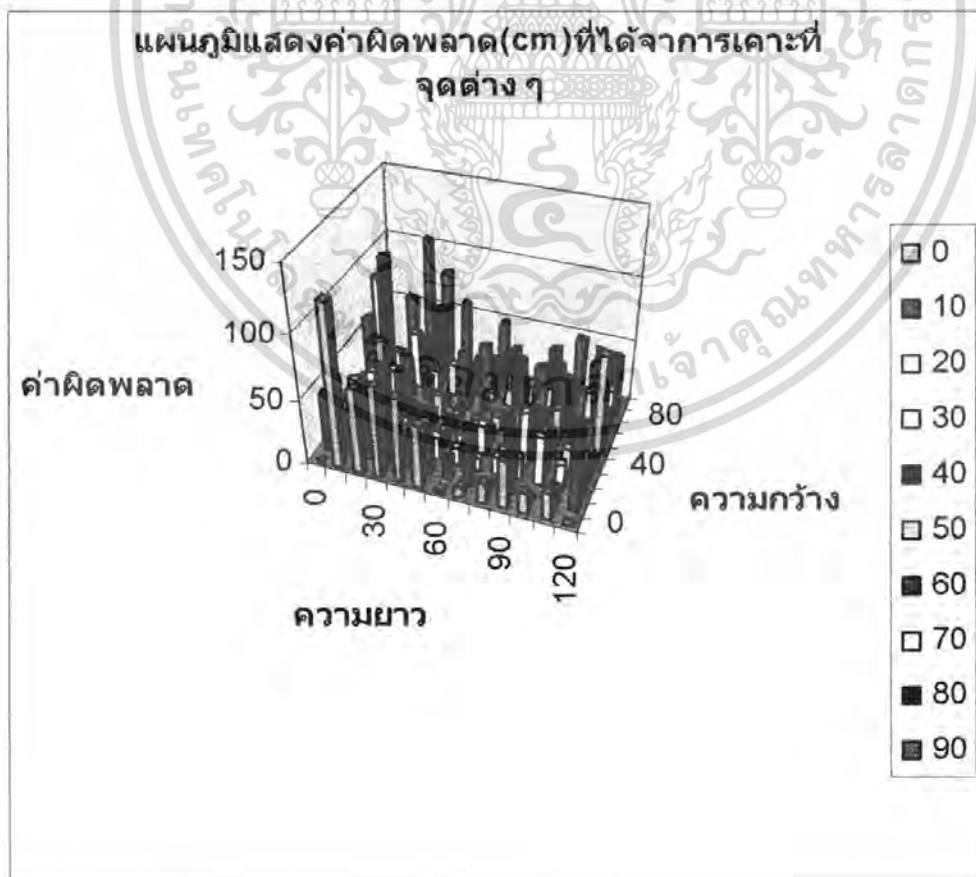


รูปที่ 4.55 ผลการป้อนค่า N เพื่อคลิกคัมเบตคลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงค่าผิดพลาด(cm) ที่ได้จากการเกาะที่จุดต่าง ๆ

Y X	0	10	20	30	40	50	60	70	80	90
0	-	43.046	48.093	19.104	17.029	18.601	59.033	83.546	91.706	-
10	126.142	30.594	50.219	9.055	7.615	52.392	29.12	26.248	11.401	53.15
20	66.128	58.412	59.933	14.866	14.56	16.124	31.016	43.416	31.622	102.5
30	71.847	77.006	64.381	12.041	15.297	3.162	17.804	16.278	58.821	79.25
40	67.067	22.36	13.416	15.811	16.155	18.384	6	24.207	13.928	56.54
50	50.159	25.298	18.357	13.416	12.206	9.219	47.507	6.403	22.36	24.21
60	2.236	16.124	14.035	14	5	5	16.031	7.81	27.018	48.01
70	8.062	44.181	8.944	25.238	11.704	11.661	2	13.892	37.735	20.12
80	15.132	4.472	14.422	8.2462	25.317	18.973	31.304	21.931	13.038	17.2
90	38.832	30.805	17.691	43.566	36.013	12.165	13	22.472	44.553	21.93
100	17.262	1	38.327	5.83	22.09	28.861	43.68	22.022	20	46.87
110	22.203	14.866	20.396	22.36	31.4	18.11	12	25.961	39.051	40.2
120	-	59.135	37.121	23.345	8.602	78.006	25.942	31.325	48.026	-



เอกสารนี้เป็นเอกสารที่ 4.56 แผนภูมิแสดงค่าผิดพลาด(cm) ที่ได้จากการเกาะที่จุดต่าง ๆ ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

จากการทดลองคุณสมบัติต่างๆของตัวรับ ตัวกลางที่ใช้รองรับการเคาะและวัตถุที่ใช้ในการเคาะ ตลอดจนศึกษาปัจจัยต่างๆที่มีผลต่อขนาดและเฟสของรูปคลื่นของสัญญาณ เพื่อที่จะนำไปวิเคราะห์ในการสร้างและออกแบบชิ้นงานให้ตรงกับความต้องการและมีประสิทธิภาพสูงสุด ซึ่งผลที่ได้สรุปได้ดังนี้

5.1 สรุปเปรียบเทียบความไวและขนาดสัญญาณที่รับได้จากตัวรับชนิดต่างๆ

- 1.1 ลำโพง มีความไวต่อการเคาะน้อยมาก โดยขนาดสัญญาณมีเพียง 50 mVp-p เมื่อขยับแผ่นไดอะแฟรมจากการใช้ลมเป่าแรงๆ เท่านั้น
- 1.2 เปียโซขนาดเล็กแบบเปลือย มีความไวต่อการเคาะปานกลาง 5 Vp-p แต่ต้องเคาะด้วยวัตถุปลายแหลมแรงๆ เท่านั้น
- 1.3 เปียโซขนาดใหญ่แบบเปลือย มีความไวต่อการเคาะปานกลาง 8 Vp-p โดยเคาะแรงๆ ด้วยวัตถุของแข็งทั่วไป เช่น แท่งไม้ คินสอไม้
- 1.4 เปียโซขนาดใหญ่แบบมีฝาครอบ มีความไวต่อการเคาะสูง 5 Vp-p โดยเคาะเบาๆ ด้วยปลายนิ้ว

ดังนั้น ชนิดตัวรับที่มีความไวและขนาดสัญญาณที่เหมาะสมแก่การนำไปใช้งานคือ เปียโซขนาดใหญ่แบบมีฝาครอบ

5.2 สรุปเปรียบเทียบวัตถุที่ใช้เคาะ

จากผลการทดลองสังเกตได้ว่าขนาดแอมพลิจูดที่ได้จากการเคาะ โดยใช้เหล็กปลายแหลม และเหล็กหุ้มเทปขาวจะมีขนาดแอมพลิจูดมากถึง 6 Vp-p ซึ่งต่างจากการเคาะด้วยปลายนิ้วที่ได้เพียง 1.5 Vp-p สังเกตว่าเหล็กปลายแหลมจะให้รูปสัญญาณที่ไม่สมมูลสมมาตรกัน ทั้งนี้เกิดขึ้นเนื่องจากการกระทบกันโดยตรงระหว่างของแข็งที่มีความหนาแน่นสูงนั่นคือเหล็กกับกระจกที่เป็นบอร์คอรอบรับเสียง ต่างจากเหล็กหุ้มเทปขาวและนิ้วซึ่งมีการลักษณะที่ไม่แข็งจนเกินไปทำให้สัญญาณที่ได้มีลักษณะสมมาตร แต่สิ่งที่เราสนใจในการทดลองนี้คือ เคาะด้วยปลายนิ้วมีจะให้ขนาดแรงดันเพียงพอต่อการนำไปใช้งานเท่านั้นซึ่งจากการทดลองนั้นก็จะได้แรงดัน 1.5 Vp-p ซึ่งหากนำไปผ่านวงจรขยาย 100 เท่าก็เพียงพอต่อการนำไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 สรุปเปรียบเทียบตัวกลางที่ใช้เคาะ

จากผลการทดลองเคาะที่ตำแหน่งระยะห่างเดียวกันบนวัตถุตัวกลางชนิดต่างๆ สังเกตได้ว่า

- แผ่นฟิวเจอร์บอร์ด มีขนาดแอมพลิจูด 6 Vp-p รูปสัญญาณไม่แน่นอน
- แผ่นไม้(โต๊ะ) มีขนาดแอมพลิจูด 4 Vp-p รูปสัญญาณแน่นอน
- แผ่นกระจก มีขนาดแอมพลิจูดมากถึง 14 Vp-p รูปสัญญาณแน่นอนสมมาตรกัน

สรุปได้ว่าวัตถุที่นำมาใช้รองรับเป็นบอร์ดสำหรับการเคาะ หากมีความหนาแน่นของเนื้อวัตถุสูง จะให้ขนาดของแอมพลิจูดที่สูงและแน่นอน ดังนั้นเราควรนำกระจกมาใช้เป็นวัตถุรองรับการเคาะในงานนี้จึงเหมาะสมที่สุด

5.4 สรุปเปรียบเทียบสัญญาณที่มุมตำแหน่งต่างๆ รอบๆ เปียโซ

จากผลการทดลองเคาะรอบๆ ตัวรับที่มุมต่างๆ ที่ระยะห่างเท่ากันสังเกตได้ว่า ขนาดแอมพลิจูดและขนาดเฟสมีค่าคงที่เท่ากันทุกจุดรอบตัวรับนั่นคือที่ตำแหน่งค่ามุมใดๆ ไม่มีผลต่อตัวรับ

5.5 สรุปเปรียบเทียบระยะทางที่ใช้เคาะ

จากผลการทดลองสังเกตได้ว่าที่ระยะเคาะที่ตำแหน่ง 0 cm ซึ่งใกล้ตัวรับมากที่สุด สัญญาณจะมีแอมพลิจูดมากที่สุดและลูกคลื่นลูกแรกมีเฟสเป็นซิกบววก่อน จากนั้นเมื่อระยะทางห่างออกไปจากตัวรับขนาดแอมพลิจูดของสัญญาณจะเล็กลงไปเรื่อยๆ และที่สำคัญคือมุมเฟสของลูกคลื่นลูกแรกจะมีการค่อยๆ กลับเฟสจากซิกบวกลูกแรกแล้วค่อยๆ เป็นซิกลบไปเรื่อยๆ แล้ววนกลับมาเป็นซิกบวกลูกถัดมา นั่นคือสัญญาณมีการเปลี่ยนมุมเฟสอยู่ตลอดเวลา สัมพันธ์กับระยะทางที่ห่างออกไปดังนั้นจากข้อมูลดังกล่าวเราจึงต้องออกแบบวงจรที่ทำงานได้ทั้งซิกบวกลูกแรกและซิกลบลูกแรกที่ผ่านมาเพื่อนำสัญญาณเริ่มต้นดังกล่าวไปหาผลต่างทางเวลาใช้คำนวณต่อไป

5.6 สรุปผลจากการทดลอง วงจรภาครับสัญญาณเคาะ

สามารถขยายสัญญาณที่ได้จากการเคาะโดยขยายอินพุตได้ทั้งซิกบวกลูกแรกและซิกลบ โดยใช้วงจรขยายแบบกลับเฟสและขยายแบบไม่กลับเฟสสองชุด เพื่อนำสัญญาณส่งไป SUM กันใน FPGA โดยใช้ OR Gate จากการทดลองป้อนอินพุตที่ระดับแรงดันต่างๆ สังเกตได้ว่าอินพุตต่ำสุดที่วงจรขยายได้แล้วได้เข้าพุทที่เป็นลอจิก 1 คือ 600 mV ซึ่งมีผลทำให้ไม่สามารถขยายสัญญาณที่เคาะตำแหน่งไกลๆ จากเปียโซได้ และจากการเปรียบเทียบค่าหน่วงเวลาของสัญญาณเข้าที่พุทของวงจรทั้ง 4 ชุด เมื่อป้อนสัญญาณอินพุตที่มีขนาดเท่ากัน ผลปรากฏว่าวงจรทั้ง 4 ชุด มีค่าหน่วงเวลาที่ต่างกันมากที่สุด คือ 3200 ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7 สรุปผลจากการทดลอง Simulate การทำงานในส่วนของ FPGA ในภาคต่างๆ ได้ผลดังนี้

ในส่วนของFPGA มีอยู่ 2 ส่วนที่สำคัญ คือภาค control และภาค counter

ภาค control ทำหน้าที่ในการเริ่มนับและหยุดนับเมื่อมีสัญญาณเริ่มต้นเข้ามา โดยจะรับค่าได้เพียงครั้งเดียวเท่านั้น จะรับได้อีกครั้งก็ต่อเมื่อมีการรีเซ็ต

ภาค counter ทำหน้าที่รับผลต่างทางเวลาสัญญาณที่เข้ามาในแต่ละตัวซึ่งจากการออกแบบนี้ค่าที่นับได้จะมีขนาด 8 บิต นับได้จำนวน 0- 255 ซึ่งหากเราต้องการความละเอียดของข้อมูลก็เพียงเพิ่มจำนวนบิตในการนับ

5.8 สรุปผลจากการทดลองส่วนของ MCS-51

จากการทดลองส่งค่าข้อมูลจากคิฟสวิทช์ 8 บิต เข้าสู่คอมพิวเตอร์โดยผ่าน MCS-51 เพื่อตรวจสอบ โปรแกรมที่เขียนขึ้นและตรวจสอบการส่งข้อมูลจาก MCS-51 เข้าสู่คอมพิวเตอร์ว่ามีค่าถูกต้องหรือไม่

จากผลการทดลองผลปรากฏว่า MCS-51 สามารถส่งข้อมูล 8 บิต ได้โดยไม่มีข้อผิดพลาดใด

5.9 สรุปผลจากการทดลองส่วนของโปรแกรม Visual Basicเพื่อควบคุมเมาส์

จากการทดสอบการจำลองการคลิกเมาส์ที่เขียนขึ้นโดย Visual Basic ด้วยการป้อนค่าตำแหน่ง X , Y แทนค่าที่ได้จากการคำนวณ และป้อนค่าด้วยจำนวน N เพื่อแทนค่าจำลองการป้อนค่าจากการนับใน FPGA

จากผลการทดลองผลปรากฏว่า ได้ค่าตำแหน่งการคลิก ที่ถูกต้องและแม่นยำ ในรูปการคลิกแบบต่างๆ

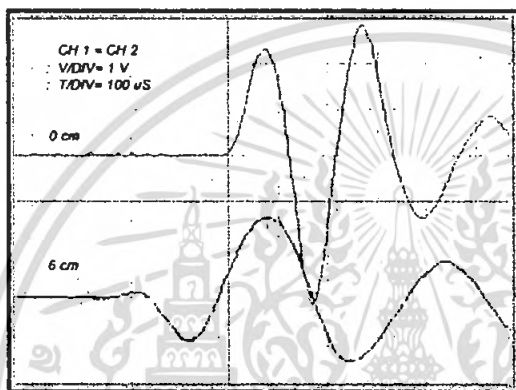
5.10 สรุปรวม

จากการทดลองทั้งหมดในโปรเจกชันนี้สามารถสรุปเป็นคุณสมบัติความสามารถได้ คือ

สามารถใช้ปลายนิ้วแตะลงบนกระจกเพื่อควบคุมการคลิกแทนเมาส์ได้โดยใช้ปลายนิ้วแตะเบาๆ ก็ทำให้วงจรรับรู้ได้ถึงตำแหน่งการสัมผัส แต่ค่าความถูกต้องและความแม่นยำยังไม่ดีพอ จึงทำให้ค่าตำแหน่งที่ได้จากการแตะมีความห่างจากตำแหน่งจริงมีค่าเฉลี่ยประมาณจากกราฟคือ 40 cm และค่าความแม่นยำของการแตะคือ เตะตำแหน่งเดิม 10 ครั้งจะ ได้ค่าเดิมเพียง 3 ครั้งสังเกตได้ว่าน้ำหนักที่ใช้ในการแตะ มีผลต่อค่าที่ผิดพลาดไปคือ

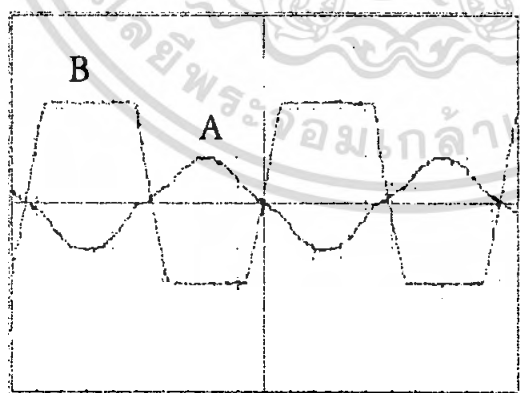
หากทำการเคาะแรงๆ จำนวนเวลาที่ FPGA นับได้จะมีค่าน้อยส่งผลให้ค่าตำแหน่งที่เคาะมีความใกล้เคียงตำแหน่งจริงมาก แต่หากทำการเคาะเบาๆ จำนวนเวลาที่ FPGA นับได้จะมีค่ามากส่งผลให้ค่าตำแหน่งที่เคาะห่างจากตำแหน่งจริงมาก ซึ่งเมื่อเราทำการศึกษาถึงสาเหตุของปัญหาดังกล่าวปรากฏว่า

1. เกิดจากเฟรซีและวงจรของภาครับที่ทำให้เกิดช่วงของเวลาที่หน่วงเวลาคือเฟรซีที่มีการกลับเฟสไปเรื่อยๆ ตามระยะห่างที่เคาะออกไปเรื่อยๆ โดยจะกลับเฟสหนึ่งครั้งเมื่อมีการเคาะห่างออกไป 6 cm



รูปที่ 5.1 แสดงการกลับเฟสเมื่อเคาะห่างออกไป 6 cm

2. เมื่อเราทำการเคาะในตำแหน่งที่ไกลออกไป(ระยะไกลสุดของการเคาะ) สัญญาณที่ได้มีขนาดเบาเกินไปทำให้เกิดการ delay ทางเวลาขึ้นในวงจรขยาย



A : INPUT ; V/Div = 50 mV , T/Div = 250 μs

B : OUTPUT ; V/Div = 2 V , T/Div = 250 μs

รูปที่ 5.2 แสดงสัญญาณที่วัดได้จากวงจรภาครับเมื่อป้อนอินพุตต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปสัญญาณที่วัดได้จากวงจรถ่ายเมื่อป้อนอินพุตต่างๆ แล้วเอาพุทเกิดการ Delay เป็น Slop สังเกตได้ว่ามีค่า Delay สูงถึง 100 μ S ซึ่งหมายความว่าถ้าจะให้วงจรใน FPGA นับ Clock พลาคไปถึง $100/16 = 7$ ลูก (1 ลูกใช้เวลา 16 μ S) ซึ่งทำให้ระยะที่เคาะผิดพลาคห่างออกไปถึง $7 * 1.5 = 10.5$ cm (1 ลูก ได้ระยะทาง 1.5 cm)

5.11 แนวทางการพัฒนาต่อ

เนื่องจากค่าของตำแหน่งเคาะที่ผิดพลาคไปเกิดจากเวลาที่ Delay ไปในส่วนของเปียโซซึ่งเราจะต้องแก้ปัญหาดังต่อไปนี้คือ

1. ใช้ตัวรับสัญญาณที่มีคุณสมบัติเหมือนกันให้มากที่สุด
2. ออกแบบวงจรขยายที่มีสามารถขยายแรงดันต่างๆ และวงจรที่ขยายนั่นลดทอนต่อสัญญาณรบกวนได้
3. ออกแบบให้สัญญาณรบกวนต่างๆที่เกิดขึ้น เช่น แรงสั่นสะเทือนจากภายนอกให้รบกวนต่อชิ้นงานให้น้อยที่สุด

เอกสารอ้างอิง

เกรียงวิทย์ ศรีนครินทร์เลิศ, “ฟิสิกส์ KEY ENTRANCE” , หน้า 55-58

เฉลิมชัย จีรพันธุ์, “ การสร้างและวิเคราะห์วงจรไฟฟ้าสมมูล BVDของสารเป็ยโซอิเล็กทริก”,

วพ ๓417ก2549,หน้า4-8,ปริญญาานิพนธ์ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร
บัณฑิต สาขาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ศิริชัย เรืองเดช, สุรเดช สนงาม “เม้าส์สามมิติ”, ปพ ศ452ม 2546ม, ปริญญาานิพนธ์ของการศึกษาตาม
หลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต ภาควิชาอิเล็คทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมหลักใน MCS-51

```

#include <reg51.h>
#include <stdio.h>
#include <intrins.h>
#include <i2cG.h>
#define PCF8574A_ID 0x70
void start_rd();
sbit P1_2 = P1^2;
sbit P1_3 = P1^3;
sbit P1_4 = P1^4;
sbit P1_5 = P1^5;
sbit P1_6 = P1^6;
sbit P1_7 = P1^7;
void start_rd()
{
    i2c_start();
    i2c_wbyte(PCF8574A_ID+1);
}
void main()
{
    unsigned int ij;
    int    N1,N2,N3,N4;
    unsigned char u;
    SCON = 0x50;
    TMOD = 0x21;
    TH1 = 0xFD;
    TL1 = 0xFD;
    TI = 1;
    TR1 = 1;
    while(1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{   if(P1_6==0)
    {
        u = 1;
    }
    if(P1_6==1)
    {   if(u==1)
        {   P1_2=0;
            start_rd();
            N1 = i2c_rdbyte();
            i2c_stop();
            P1_2=1;

            P1_3=0;
            start_rd();
            N2 = i2c_rdbyte();
            i2c_stop();
            P1_3=1;

            P1_4=0;
            start_rd();
            N3 = i2c_rdbyte();
            i2c_stop();
            P1_4=1;

            P1_5=0;
            start_rd();
            N4 = i2c_rdbyte();
            i2c_stop();
            P1_5=1;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("A%d",N1);
```

```
printf("B%d",N2);
```

```
printf("C%d",N3);
```

```
printf("D%d",N4);
```

```
printf("E");
```

```
u=getchar();
```

```
for(j=0;j<35000;j++);
```

```
P1_7 = 1;
```

```
for(i=0;i<10000;i++);
```

```
P1_7 = 0;
```

```
}
```

```
}
```

```
}
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม I2C

```

#include<intrins.h>
sbit SCL = P1^0;
sbit SDA = P1^1;
void i2c_delay(void)
{
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
}
void i2c_high(void)
{
    SCL = 1;
    i2c_delay();
}
void i2c_low(void)
{
    SCL = 0;
    i2c_delay();
}
void i2c_start(void)
{
    SDA = 1;
    i2c_high();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    SDA = 0;
    i2c_delay();
    i2c_low();
    SDA = 1;
}

void i2c_stop(void)
{
    SDA = 0;
    i2c_high();
    SDA = 1;
}

bit i2c_wrbyte(unsigned dat)
{
    unsigned char i;
    bit outbit;
    for (i=1;i<=8;i++)
    {
        outbit = dat & 0x80;
        SDA = outbit;
        dat = dat << 1;
        i2c_high();
        i2c_low();
    }
    SDA = 1;
    i2c_high();
    outbit = SDA;
    i2c_low();
    return(outbit);
}

unsigned char i2c_rdbyte()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    unsigned char i,d;
    bit inbit;
    dat = 0x00;

    for(i=1;i<=8;i++)
    {
        i2c_high();

        inbit = SDA;
        dat = dat << 1;
        dat = dat | inbit;
        i2c_low();
    }

    SDA = 1;
    i2c_high();
    inbit = SDA;
    i2c_low();

    return(dat);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Visual Basic

Option Explicit

Private Type POINTAPI

x As Long

y As Long

End Type

Private Declare Function GetAsyncKeyState Lib "user32" _

(ByVal vKey As Long) As Integer

Private Const VK_LBUTTON = &H1

Private Const VK_RBUTTON = &H2

Private Type RECT

Left As Long

Top As Long

Right As Long

Bottom As Long

End Type

Const MOUSEEVENTF_LEFTDOWN = &H2

Const MOUSEEVENTF_LEFTUP = &H4

Const MOUSEEVENTF_MIDDLEDOWN = &H20

Const MOUSEEVENTF_MIDDLEUP = &H40

Const MOUSEEVENTF_MOVE = &H1

Const MOUSEEVENTF_ABSOLUTE = &H8000

Const MOUSEEVENTF_RIGHTDOWN = &H8

Const MOUSEEVENTF_RIGHTUP = &H10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Declare Sub mouse_event Lib "user32" (ByVal dwFlags As Long, ByVal dx As Long,
ByVal dy As Long, ByVal cButtons As Long, ByVal dwExtraInfo As Long)
Private Declare Function GetMessageExtraInfo Lib "user32" () As Long
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
Private Declare Function SetCursorPos Lib "user32" (ByVal x As Long, ByVal y As Long) As
Long
Private Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, lpRect As
RECT) As Long

Private Sub Command1_Click()
Dim Rec As RECT
GetWindowRect Form1.hwnd, Rec
SetCursorPos Rec.Right - 15, Rec.Top + 15
mouse_event MOUSEEVENTF_LEFTDOWN Or MOUSEEVENTF_LEFTUP, 0&, 0&,
Command1, GetMessageExtraInfo
End Sub

Private Sub Command2_Click()
If optLeft(0).Value = True Then
Call Click_Mouse("LEFT", Val(txtXpos(0)), Val(txtYpos(0)))
If Check5.Value = 1 Then
Call Click_Mouse("LEFT", Val(txtXpos(0)), Val(txtYpos(0)))
End If
Else
Call Click_Mouse("RIGHT", Val(txtXpos(0)), Val(txtYpos(0)))
End If

Call DelayClick
End Sub

Private Sub Click_Mouse(Event_Click As String, Pos_X As Long, Pos_Y As Long)
SetCursorPos Pos_X, Pos_Y

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Event_Click = "LEFT" Then
mouse_event MOUSEEVENTF_LEFTDOWN Or MOUSEEVENTF_LEFTUP, 0&, 0&,
Command1, GetMessageExtraInfo
Else
mouse_event MOUSEEVENTF_RIGHTDOWN Or MOUSEEVENTF_RIGHTUP, 0&, 0&,
Command1, GetMessageExtraInfo
End If
End Sub

```

```

Private Sub DelayClick()
Dim cntDelay As Integer
For cntDelay = 1 To 20000
Next cntDelay
End Sub

```

```

Private Sub Combo1_Click()
MSComm1.CommPort = Combo1.Text
MSComm1.Settings = "9600, n , 8, 1"
MSComm1.PortOpen = True
End Sub

```

```

Private Sub Form_Load()

```

```

Me.ScaleMode = vbPixels
Combo1.Text = 1
Label1.Caption = "Connect to Com Port no."
Timer1.Interval = 10
Timer1.Enabled = True
Text1.Text = ""
Text2.Text = ""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text3.Text = ""
Text4.Text = ""
txtXpos(0) = ""
txtYpos(0) = ""
Label4.Caption = "N1"
Label5.Caption = "N2"
Label6.Caption = "N3"
Label7.Caption = "N4"

```

```
End Sub
```

```

Private Sub Timer1_Timer()
On Error Resume Next
Const SpeedBaud = 16000
Dim N1(4), N2(4), N3(4), N4(4), car As Variant
Dim I, J, K, L As Integer
Dim Rec As RECT, Point As POINTAPI
GetWindowRect Me.hwnd, Rec
GetCursorPos Point
Text6.Text = Point.x
Text7.Text = Point.y
If Point.x >= Rec.Left And Point.x <= Rec.Right And Point.y >= Rec.Top And Point.y <=
Rec.Bottom Then
Me.Caption = "MouseCursor is on form."
Else
Me.Caption = "MouseCursor is not on form."
End If
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1
MSComm1.Output = "1"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

car = MSComm1.Input
If car = "A" Then
For I = 1 To 4
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1
car = MSComm1.Input
If car = "B" Then
For J = 1 To 4
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1
car = MSComm1.Input
If car = "C" Then
For K = 1 To 4
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1
car = MSComm1.Input
If car = "D" Then
For L = 1 To 4
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1
car = MSComm1.Input
If car = "E" Then
Text1.Text = N1(1) & N1(2) & N1(3) & N1(4)
Text2.Text = N2(1) & N2(2) & N2(3) & N2(4)
Text3.Text = N3(1) & N3(2) & N3(3) & N3(4)
Text4.Text = N4(1) & N4(2) & N4(3) & N4(4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim Delta, x, x0, x1, x2, x3, y, y0, y1, y2, y3, r, r0, r1, r2, r3, r4, a1, a2, a3, a4, b1, b2, b3,

b4, c1, c2, c3, c4, d1, d2, d3, d4 As Variant

$$r0 = (\text{CDbl}(\text{Text1.Text}) * 1.318)$$

$$r1 = (\text{CDbl}(\text{Text2.Text}) * 1.222)$$

$$r2 = (\text{CDbl}(\text{Text3.Text}) * 1.288)$$

$$r3 = (\text{CDbl}(\text{Text4.Text}) * 1.156)$$

$$x0 = 0$$

$$y0 = 0$$

$$x1 = 0$$

$$y1 = 90$$

$$x2 = 120$$

$$y2 = 90$$

$$x3 = 120$$

$$y3 = 0$$

$$a1 = (2 * x0) - (2 * x1)$$

$$a2 = (2 * x0) - (2 * x2)$$

$$a3 = (2 * x0) - (2 * x3)$$

$$b1 = (2 * y0) - (2 * y1)$$

$$b2 = (2 * y0) - (2 * y2)$$

$$b3 = (2 * y0) - (2 * y3)$$

$$c1 = (2 * r0) - (2 * r1)$$

$$c2 = (2 * r0) - (2 * r2)$$

$$c3 = (2 * r0) - (2 * r3)$$

$$d1 = (x0 * x0) - (x1 * x1) + (y0 * y0) - (y1 * y1) + (r1 * r1) - (r0 * r0)$$

$$d2 = (x0 * x0) - (x2 * x2) + (y0 * y0) - (y2 * y2) + (r2 * r2) - (r0 * r0)$$

$$d3 = (x0 * x0) - (x3 * x3) + (y0 * y0) - (y3 * y3) + (r3 * r3) - (r0 * r0)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta = ((a1 * b2 * c3) + (b1 * c2 * a3) + (c1 * a2 * b3) - (a3 * b2 * c1) - (b3 * c2 * a1) - (c3 * a2 * b1))$$

$$x = ((d1 * b2 * c3) + (b1 * c2 * d3) + (c1 * d2 * b3) - (d3 * b2 * c1) - (b3 * c2 * d1) - (c3 * d2 * b1)) / \Delta$$

$$y = ((a1 * d2 * c3) + (d1 * c2 * a3) + (c1 * a2 * d3) - (a3 * d2 * c1) - (d3 * c2 * a1) - (c3 * a2 * d1)) / \Delta$$

$$r = ((a1 * b2 * d3) + (b1 * d2 * a3) + (d1 * a2 * b3) - (a3 * b2 * d1) - (b3 * d2 * a1) - (d3 * a2 * b1)) / \Delta$$

```
txtXpos(0) = (x * 8.533)
```

```
txtYpos(0) = (y * 8.533)
```

```
If optLeft(0).Value = True Then
```

```
    Call Click_Mouse("LEFT", Val(txtXpos(0)), Val(txtYpos(0)))
```

```
If Check5.Value = 1 Then
```

```
    Call Click_Mouse("LEFT", Val(txtXpos(0)), Val(txtYpos(0)))
```

```
End If
```

```
Else
```

```
    Call Click_Mouse("RIGHT", Val(txtXpos(0)), Val(txtYpos(0)))
```

```
End If
```

```
Call DelayClick
```

```
End If
```

```
N4(L) = car
```

```
Next L
```

```
End If
```

```
N3(K) = car
```

```
Next K
```

```
End If
```

```
N2(J) = car
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next J
End If
N1(I) = car
Next I
End If
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้