สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การ์ดลอจิกอนาไลซ์เซอร์

นายนพรัตน์ จิตจาตุรันต์
นางสาวบุษยา ตั้งเจริญ

โครงงานพิเศษเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

วศ.
ว61847ก
2547

เลขหมู่.....
เลขทะเบียน 58707
วัน,เดือน,ปี 31 ต.ค. 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b. 11 มิ.ย. 56
i.....

Logic Analyzer PCI Card

Mr. Nopparat Jitjaturunt
Miss Busaya Tangcharoen

A Special Project Submitted in Partial Fulfillment of the Requitement for the Degree of

Bachelor of Science

Departmemt of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

Academic Year 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง การ์ดลอจิกอนาไลซ์เซอร์

นักศึกษา นายนพรัตน์ จิตจาตุรันต์

นางสาวบุษยา ตั้งเจริญ

ภาควิชา ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**

สาขาวิชา ฟิสิกส์ประยุกต์-เครื่องมือวิทยาศาสตร์และอุตสาหกรรม

อาจารย์ที่ปรึกษา รศ.ดร.จิติ หนูแก้ว

อาจารย์ที่ปรึกษาร่วม อ.เบญจพล ตันฮู้

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ		ลายมือชื่อ
ประธานกรรมการ	ดร.ปิติพร ถนอมงาม	
กรรมการ	รศ.วิชาญ เตชิตธีระ	
กรรมการ	อ.สุรชาติ กมลดิลก	
กรรมการที่ปรึกษา	รศ.ดร.จิติ หนูแก้ว	
กรรมการที่ปรึกษาร่วม	อ.เบญจพล ตันฮู้	

.....

(รองศาสตราจารย์ วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง	การ์ดลอจิกอนาไลเซอร์
นักศึกษา	นายนพรัตน์ จิตจาตุรันต์ นางสาวบุษยา ตั้งเจริญ
ภาควิชา	ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**
สาขาวิชา	ฟิสิกส์ประยุกต์-เครื่องมือวิทยาศาสตร์และอุตสาหกรรม
ปีการศึกษา	2547
อาจารย์ที่ปรึกษา	รศ.ดร. จิติ หนูแก้ว
อาจารย์ที่ปรึกษาร่วม	อ. เบญจพล ตันฮู้

บทคัดย่อ

โครงการนี้จะเป็นการศึกษาหลักการทำงานของเครื่องลอจิกอนาไลซ์เซอร์ ซึ่งเป็นอุปกรณ์สำหรับเก็บข้อมูลทางดิจิตอล โดยการทำงานของลอจิกอนาไลซ์เซอร์ที่สร้างขึ้นจะมีการเชื่อมต่อผ่านอุปกรณ์ไอโอระหว่าง พีซีไอการ์ด กับ บอร์ดไมโครคอนโทรลเลอร์เบอร์ T89C51RD2 และ มีช่องการรับข้อมูลแบบดิจิตอล 8 ช่อง มีหน่วยความจำสำหรับการพักข้อมูลขนาด 1024 ไบต์ ซึ่งจากการทดลองพบว่า ลอจิกอนาไลซ์เซอร์ที่สร้างขึ้น มีอัตราการสุ่มข้อมูลที่ 200 กิโลเฮิรตซ์ สามารถวัดความถี่ของสัญญาณที่มากสุดคือ 50 กิโลเฮิรตซ์ และเมื่อนำไปเปรียบการทำงานกับเครื่องลอจิกอนาไลซ์เซอร์รุ่น HP1663C พบว่ามีประสิทธิภาพของการทำงานใกล้เคียงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Logic Analyzer PCI Card
Name	Mr. Nopparat Jitjaturunt
	Miss Busaya Tangcharoen
Department	Applied Physics **Faculty of Science**
Program	Applied Physics-Science and Industry Instrumentation
Academic Year	2004
Special Project Advisor	Assoc. Dr. Jiti Nukaew
Special Project co-advisor	Mr. Benchapol Tunhoo

ABSTRACT

The purpose of this special project is to a study of the principle of Logic Analyzer that use to keep digital data. This Logic Analyzer interface via I/O module, PCI Card and T89C51RD2 microprocessor board. This system has 8 input data channels and 1024 bytes of buffer memory. The result of experiment has sampling rate at 200 KHz and maximum frequency at 50 KHz. When the result of experiment compare with HP1663C Logic Analyzer, the efficiency are good.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการพิเศษนี้สำเร็จลุล่วงด้วยดี เนื่องด้วยความอนุเคราะห์จากคนหลายฝ่ายดังนี้

คุณพ่อและคุณแม่	ผู้ให้กำลังใจและการดูแล
รศ.ดร.จิติ หนูแก้ว	ผู้ให้ความรู้ด้านวิชาการ
อ.เบญจพล ตันฮู้	ผู้ให้ความรู้ด้านวิชาการ
รศ.วิชิต ศิริโชติ	ผู้ให้คำแนะนำ
บริษัท อีทีที จำกัด	ผู้เอื้อเฟื้ออุปกรณ์ในการทำโครงงานพิเศษ
คณะกรรมการทุกท่าน	ที่กรุณาตรวจรายงานโครงงานพิเศษ
สำนักหอสมุดกลาง	ที่ให้ข้อมูลความรู้
พี่ๆภาควิชาฟิสิกส์ประยุกต์	ที่คอยให้คำแนะนำ
และเพื่อนๆที่น่ารักทุกคน	ที่คอยเป็นกำลังใจและช่วยเหลือซึ่งกันและกัน

ท้ายที่สุดนี้ขอขอบคุณทุกๆท่านที่ได้กล่าวนามและไม่ได้กล่าวนามด้วยความจริงใจ

นายนพรัตน์ จิตจาตุรันต์

นางสาวบุษยา ตั้งเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญเรื่อง

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญเรื่อง	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ที่มาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตการดำเนินงาน	1
1.4 ประโยชน์ที่ได้รับ	2
บทที่2 ทฤษฎีและหลักการ	3
2.1 ระบบดิจิตอล (Digital System)	3
2.2 ลอจิกอนาไลซ์เซอร์	6
2.2.1 ลักษณะการทำงานทั่วไปของลอจิกอนาไลซ์เซอร์	6
2.2.2 หลักการสุ่ม (Sampling) สัญญาณ	7
2.2.3บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์	9
2.3 สถาปัตยกรรมพีซีไอ	11
2.3.1 โครงสร้างแบบบัส	11
2.3.2 การเคลื่อนย้ายข้อมูล	17
2.3.3 การตัดสินเลือกผู้ควบคุมบัส	19
2.4 โมดูลไอโอ	22
2.4.1 โมดูลฟังก์ชัน	22
2.4.2 โครงสร้างโมดูลไอโอ	24
2.4.3 การควบคุมไอโอด้วยโปรแกรม	26
2.4.4 เรื่องทั่วไปเกี่ยวกับการควบคุมไอโอด้วยโปรแกรม	27
2.4.5 การออกคำสั่งไอโอ	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 คำสั่งไอโอที่ถูกประมวลผล 29
2.4.7 ไอโอที่ใช้กลไกอินเทอร์รัพท์ 31
2.4.8 กระบวนการอินเทอร์รัพท์ 32
บทที่ 3 ขั้นตอนการดำเนินงาน 36
3.1 การใช้ PCI Logic Card สำหรับการวิเคราะห์สัญญาณ 36
3.1.1 ใช้โปรแกรม Labview รับส่งข้อมูลผ่านทาง I/O ของ PCI Logic Card 36
3.1.2 ใช้โปรแกรม Labview รับค่าสัญญาณจาก Function Generator 37
3.2 การใช้ Logic Analyzer สำหรับการวิเคราะห์สัญญาณ 42
บทที่ 4 ผลการทดลอง 43
4.1 การใช้ PCI Logic Card สำหรับการวิเคราะห์สัญญาณ 43
4.1.1 การรับค่าผ่าน I/O 43
4.1.2 รับค่าจาก Function Generator ผ่านบอร์ดไมโครคอนโทรเลอร์เบอร์ T89C51RD2 44
4.2 การใช้ Logic Analyzer สำหรับการวิเคราะห์สัญญาณ 68
4.2.1 รับค่าผ่าน I/O ที่ความถี่ต่างๆ
4.2.1 รับค่าผ่านวงจรหารหลายช่องสัญญาณ 68
4.2.2 โปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่างๆ 70
บทที่ 5 สรุปและข้อเสนอแนะ 71
5.1 โปรแกรมรับค่าจาก Function Generator ที่ความถี่ต่างๆ 71
5.2 โปรแกรมรับค่าผ่านวงจรหารหลายช่องสัญญาณ 72
5.3 โปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่างๆ 73
5.4 สรุปโดยรวม 74
5.5 ข้อเสนอแนะและแนวทางการพัฒนา 74
เอกสารอ้างอิง 75
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 2.1	แสดงสัญลักษณ์ระหว่างเลขฐานสิบและฐานสิบหก	4
ตารางที่ 2.2	ตารางแสดงความสัมพันธ์ระหว่างเลขฐานแปดหรือฐานสิบหก กับเลขฐานสองที่สมมูลของเลขจำนวนฐานแปดและฐานสิบหก	5
ตารางที่ 2.3	สายสัญญาณที่สำคัญของสถาปัตยกรรมพีซีไอ	14
ตารางที่ 2.4	สายสัญญาณส่วนที่เพิ่มเติมของสถาปัตยกรรมพีซีไอ	15
ตารางที่ 2.5	การแปลความหมายคำสั่งอ่านข้อมูลในสถาปัตยกรรมพีซีไอ	17
ตารางที่ 2.6	เทคนิคไอโอแบบต่างๆ	26
ตารางที่ 5.1	ตารางแสดงผลการเปรียบเทียบที่ความถี่ 5 และ 50 kHz	71
ตารางที่ 5.2	ตารางแสดงผลการเปรียบเทียบที่ความถี่ 5 ,10 และ 50 kHz	72
ตารางที่ 5.3	ตารางแสดงผลการเปรียบเทียบจากการวัดวงจรเกท	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 หลักการทำงานของลอจิกอนาไลซ์เซอร์ 6
รูปที่ 2.2 การสุ่มสัญญาณ 7
รูปที่ 2.3 การสุ่มที่มีขนาดของสัญญาณสุ่มน้อยกว่า 2 เท่าของสัญญาณอินพุท 8
รูปที่ 2.4 บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์ 9
รูปที่ 2.5 ตัวอย่างสถาปัตยกรรมพีซีไอ 12
รูปที่ 2.6 การอ่านข้อมูลในสถาปัตยกรรมพีซีไอ 17
รูปที่ 2.7 PCI Arbiter ในสถาปัตยกรรมพีซีไอ 19
รูปที่ 2.8 การควบคุมบัสระหว่างมาสเตอร์ 2 ตัว 21
รูปที่ 2.9 อัตราการถ่ายเทข้อมูลของอุปกรณ์ไอโอโดยทั่วไป 24
รูปที่ 2.10 ไดอะแกรมแบบบล็อกสำหรับโมดูลไอโอ 25
รูปที่ 2.11 เทคนิค 3 แบบสำหรับการนำเข้าข้อมูล 1 บล็อก 28
รูปที่ 2.12 ข้อมูลในหน่วยความจำและข้อมูลไอโอ 30
รูปที่ 2.13 กระบวนการอินเทอร์รัพท์อย่างง่าย 32
รูปที่ 2.14 การเปลี่ยนแปลงของหน่วยความจำและรีจิสเตอร์สำหรับอินเทอร์รัพท์ 35
รูปที่ 3.1 PCI Logic Card (ET –PCI8255V3) 36
รูปที่ 3.2 I/O Board 37
รูปที่ 3.3 (a) แสดงวงจรบอร์ดคอนโทรลเลอร์เบอร์ T89C51RD2 38
(b) บอร์ดคอนโทรลเลอร์เบอร์ T89C51RD2 38
รูปที่ 3.4 แสดงวงจรหาร 39
รูปที่ 3.5 แสดงวงจร AND, OR, NAND, NOR และ EXCLUSIVE-OR GATE 40
รูปที่ 3.4 Logic Analyzer 42
รูปที่ 4.1 โปรแกรมควบคุมไฟวิ่ง 43
รูปที่ 4.2 ผลจากการควบคุมไฟวิ่ง 43
รูปที่ 4.3 (a) แสดงการทำงานของโปรแกรมแอสเซมบลีบนบอร์ดไมโครคอนโทรเลอร์ 44
(b) โฟลวชาตแสดงการทำงานของโปรแกรม Labview บน PC 44
รูปที่ 4.4 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 1 45
รูปที่ 4.5 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 2 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 3 46
รูปที่ 4.7 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 1 47
รูปที่ 4.8 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 2 47
รูปที่ 4.9 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 3 48
รูปที่ 4.10 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 1 49
รูปที่ 4.11 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 2 49
รูปที่ 4.12 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 3 50
รูปที่ 4.13 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 1 51
รูปที่ 4.14 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 2 51
รูปที่ 4.15 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 3 52
รูปที่ 4.16 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 1 53
รูปที่ 4.17 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 2 53
รูปที่ 4.18 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 3 54
รูปที่ 4.19 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 4 54
รูปที่ 4.20 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 5 55
รูปที่ 4.21 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 1 56
รูปที่ 4.22 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 2 56
รูปที่ 4.23 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 3 57
รูปที่ 4.24 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 4 57
รูปที่ 4.25 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 5 58
รูปที่ 4.26 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 1 59
รูปที่ 4.27 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 2 59
รูปที่ 4.28 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 3 60
รูปที่ 4.29 แสดงสั ญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 4 60
รูปที่ 4.30 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 5 61
รูปที่ 4.31 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 1 62
รูปที่ 4.32 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 2 62
รูปที่ 4.33 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 3 63
รูปที่ 4.34 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 4 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.35 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 5 64
รูปที่ 4.36 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 1 65
รูปที่ 4.37 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 2 65
รูปที่ 4.38 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 3 66
รูปที่ 4.39 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 4 66
รูปที่ 4.40 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 5 67
รูปที่ 4.41 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 5 kHz 68
รูปที่ 4.42 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 10 kHz 68
รูปที่ 4.43 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 50 kHz 69
รูปที่ 4.44 แสดงสัญญาณที่วัดได้จากวงจรเกทที่ 5 และ 10 kHz 70
รูปที่ 4.45 แสดงสัญญาณที่วัดได้จากวงจรเกทที่ 8 และ 5 kHz 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของโครงงานพิเศษ

เนื่องจากปัจจุบัน ผู้ผลิตระบบคอมพิวเตอร์ส่วนมากได้ยกเลิกระบบบัสแบบไอซ่า(ISA) กับเมนบอร์ด (Mainboard) รุ่นใหม่ แต่ความต้องการในการนำคอมพิวเตอร์พีซี (PC) ไปใช้ในงานควบคุมยังมีอยู่ จึงได้พัฒนาระบบพีซีไอ การ์ด (PCI Card) ซึ่งเป็นการ์ดต่อขยายระบบคอมพิวเตอร์พีซี เพื่อทำหน้าที่เป็นอุปกรณ์รับส่งสัญญาณดิจิตอลกับอุปกรณ์ภายนอก เป็นไอโอ การ์ด (I/O Card) แบบพีซีไอ บัส (PCI BUS) นำมาทดแทนซึ่งเป็น ไอโอ การ์ด แบบไอซ่าบัส (ISA BUS) ได้

เครื่องวิเคราะห์ลอจิกเป็นอุปกรณ์ทางด้านดิจิตอลและไมโครโปรเซสเซอร์ เนื่องจากสัญญาณของวงจรประเภทนี้มีลักษณะไม่เป็นคาบ จึงไม่สามารถใช้ออสซิลโลสโคปวัดได้ และเนื่องจากข้อมูลสำคัญในวงจรดิจิตอลมีระดับลอจิกเพียง 2 ระดับ คือ 1 หรือ 0 เท่านั้น กับเวลาที่ข้อมูลเปลี่ยนแปลง จึงไม่จำเป็นต้องทราบรายละเอียดของสัญญาณ ดังนั้นเครื่องวิเคราะห์ลอจิกจึงเหมาะสมและมีความจำเป็นต้องใช้ในการวิเคราะห์

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาระบบสถาปัตยกรรมพีซีไอ
2. เพื่อศึกษาหลักการเครื่องวิเคราะห์ลอจิก (Logic Analyzer)
3. เพื่อศึกษาพีซีไอ การ์ด PCI Card
4. เพื่อศึกษาเอพีไอ API (Application Programming Interfacing) เช่น โปรแกรม LABVIEW สำหรับใช้ในงานควบคุม

1.3 ขอบเขตการดำเนินงาน

1. ศึกษาทฤษฎีทางระบบดิจิตอล เกี่ยวกับระดับลอจิกและสัญญาณทางดิจิตอล
2. ศึกษาระบบสถาปัตยกรรมพีซีไอ
3. ศึกษาหลักการทำงานของเครื่องวิเคราะห์ลอจิก (Logic Analyzer)
4. ศึกษาหลักการทำงานและการทดสอบไอโอ(I/O) ของ Logic PCI Card
5. ศึกษาการเขียนโปรแกรมติดต่อ และควบคุม Hardware ด้วยโปรแกรม LABVIEW
6. ทำการทดลองและเก็บผลจากตัวอย่าง Logic PCI Card
7. วิเคราะห์และสรุปผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่ได้รับ

1. สามารถพัฒนาบอร์ดการเชื่อมต่อด้วย PCI Card
2. เข้าใจหลักการทางด้าน PCI Card และสามารถนำไปประยุกต์ใช้ได้
3. เข้าใจหลักการของเครื่องวิเคราะห์ลอจิก
4. สามารถนำความรู้ทางด้านโปรแกรม LABVIEW มาใช้ติดต่อและควบคุมทาง Hardwareได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 **ระบบดิจิตอล** (Digital System)

- ระบบเลขฐานสิบ

แสดงด้วยการจัดเรียงของสัญลักษณ์ 10 สัญลักษณ์ คือ 0 – 9 ที่เรียกว่า Decimal Digit ตำแหน่งของแต่ละตัวเลขในลำดับจะมีค่าน้ำหนัก (Weight) ที่แน่นอน และแต่ละตัวเลขจะเป็นตัวคูณ (Multiplier) ของน้ำหนักของหลักนั้น ตำแหน่ง (Positional) น้ำหนักของแต่ละตำแหน่งเป็นค่าของฐาน คือ 10 แล้วยกกำลัง ค่าของเลขจำนวนนั้น จะเป็นผลรวมของผลคูณที่เกิดจากการคูณตัวเลขแต่ละตัวด้วยน้ำหนักของหลักของตัวเลขนั้นๆ

- ระบบเลขฐานสอง

มีเพียง 2 สัญลักษณ์ คือ 0 และ 1 ตัวเลขหนึ่งตัวของเลขฐานสองเรียกว่า บิต (Bit) จำนวนของเลขจำนวนฐานสองประกอบด้วยบิตเรียงเป็นลำดับโดยที่แต่ละบิตมีค่าเป็น 0 หรือ 1 และถูกคูณด้วย 2 ยกกำลังค่าต่างๆ น้ำหนักของแต่ละตำแหน่งบิตจะมากกว่าน้ำหนักของหลักที่อยู่ถัดไปทางขวาเป็น 2 ยกกำลัง ค่าของเลขฐานสอง เป็นผลรวมทั้งหมดของผลคูณแต่ละบิตกับน้ำหนักของบิตนั้น

เช่นเดียวกันกับระบบเลขฐานสิบที่บิตตำแหน่งทางขวาของหลักหน่วย(2^0) จะมีค่าน้ำหนักที่เป็นฐานยกกำลังด้วยค่าลบ ในการแยกเลขจำนวนเต็มและเศษของเลขจำนวนฐานสองก็จะใช้จุดฐานสอง (Binary point) แทนที่จะใช้จุดทศนิยมซึ่งเป็นฐานสิบ

- ระบบเลขฐานแปด

ระบบเลขฐานแปดซึ่งมีฐานเท่ากับ 8 ประกอบด้วยเลขฐานแปด 0, 1, 2, 3, 4, 5, 6 และ 7 และเนื่องจากฐานเป็นมีค่าเป็น 2 ยกกำลัง (2^3) จึงเป็นง่ายที่จะแปลงจากเลขฐานแปดเป็นเลขฐานสอง ให้สังเกตว่าในระบบเลขฐานแปดไม่มี 8 และ 9

ในระบบเลขฐานแปดมีโครงสร้างเช่นเดียวกับเลขฐานสองและเลขฐานสิบ แต่ละหลักของเลขฐาน 8 จะถูกคูณด้วย 8 ยกกำลัง

- ระบบเลขฐานสิบหก

ระบบเลขฐานสิบหกมีฐานเป็น 16 ได้นับความนิยมมากสำหรับการป้อน หรือ รับข้อมูลในระบบดิจิตอล เนื่องจากว่าฐานเป็น 2 ยกกำลัง (2^4) จึงทำให้การแปลงระหว่างเลขฐานสิบหกและเลขฐานสองกลับไปมาทำได้ง่าย และเห็นได้ว่าการแปลงข้อมูลเป็นเลขฐานสอง 4 บิต ที่เรียกว่านิบเบิล (Nibble) ซึ่งเป็นส่วนเบื้องต้นของข้อมูลฐานสองค่ามากๆนั้นสามารถทำได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเลขฐานสิบหกประกอบด้วย 16 สัญลักษณ์ เราใช้ 10 สัญลักษณ์ของฐาน 10 คือ 0 ถึง 9 แต่ต้องมีสัญลักษณ์ใหม่เพิ่มอีก 6 สัญลักษณ์ สำหรับใช้เป็นสัญลักษณ์ทั้งหมดของฐานสิบหก เพื่อให้ได้ตามที่ต้องการจึงใช้พยัญชนะ (Alphabet) ภาษาอังกฤษ 6 ตัวแรกเป็นสัญลักษณ์ อักษรแต่ละตัวจะมีค่าตรงกับเลขฐานสิบดังนี้

ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

ตารางที่ 2.1 แสดงสัญลักษณ์ระหว่างเลขฐานสิบและฐานสิบหก

การแปลงระหว่างระบบจำนวนเลข

- เลขฐานสิบเป็นเลขฐานสอง

วิธีที่เป็นระบบสำหรับการแปลงเลขจำนวนเต็มฐานสิบเป็นเลขฐานสองก็คือ วิธีการหารเลขจำนวนนั้นด้วย 2 ผลหารที่ได้ก็หารด้วย 2 ต่อไป แล้วนำผลหารที่ได้หารด้วย 2 ต่อไปเรื่อยๆ ลำดับของเศษที่เหลือจากกี่หารเหล่านี้จะเป็นเลขฐานสองที่สมมูลกับค่าของเลขฐานสิบ โดยเศษของการหารครั้งแรกจะเป็นบิตนัยสำคัญต่ำสุดและเศษการหารครั้งสุดท้ายจะเป็นบิตนัยสำคัญสูงสุด วิธีการหารด้วย 2 จะกระทำไปจนกระทั่งสุดท้ายคือหาร 1 ด้วย 2 ให้ผลหารเป็น 0 และมีเศษเป็น 1

- เลขฐานสิบเป็นฐานแปดและฐานสิบหก

กระบวนการแปลงเลขฐานสิบเป็นเลขฐานแปดหรือเลขฐานสิบหกก็เป็นเช่นเดียวกับการแปลงเลขฐานสิบเป็นเลขฐานสอง ในการแปลงแนะนำให้ใช้วิธีการหารด้วย 8 (สำหรับฐานแปด) หรือ (สำหรับฐานสิบหก) ซ้ำๆกัน และเก็บเศษที่เหลือเอาไว้เป็นลำดับ ในการแปลงเป็นเลขฐานสิบหก ถ้าเศษที่เหลือมีค่ามากกว่า 9 ก็ให้เขียนเป็นหลักของฐานสิบหกที่มีค่าสมมูล (A ถึง F)

- การแปลงระหว่างฐานสอง ฐานแปด และฐานสิบหก

ในการหาค่าของเลขฐานสองที่สมมูลกับเลขจำนวนในฐานซึ่งอยู่ในรูปแบบของ 2^n นั้น แต่ละหลักของเลขจะถูกแทนด้วยเลขฐานสอง n บิตที่สมมูลกัน เนื่องจากเลขฐานแปดมีฐาน $8=2^3$ ดังนั้นแต่ละหลักของเลขฐานแปดจึงถูกแทนได้ด้วยเลขฐานสอง 3 บิต ที่สมมูลกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทำนองเดียวกันเลขฐานสิบซึ่งมีฐาน 16 = 2^4 สามารถแปลงเป็นเลขฐานสองได้โดยการแทนแต่ละตัวเลขฐานสิบหกด้วยเลขฐานสอง 4 บิตที่มีสมมูลกัน

ในการเขียนเลขฐานสองที่สมมูลแต่ละหลักของฐานแปดหรือฐานสิบหกต้องเขียนให้ครบ 3 หรือ 4 บิตตามลำดับ ดังตาราง

ตัวเลขฐานแปดหรือฐานสิบหก	เลขฐานสองที่สมมูลของเลขจำนวนฐานแปด	เลขฐานสองที่สมมูลของเลขจำนวนฐานสิบหก
0	000	0000
1	001	0001
2	010	0010
3	011	0011
4	100	0100
5	101	0101
6	110	0110
7	111	0111
8		1000
9		1001
A		1010
B		1011
C		1100
D		1101
E		1110
F		1111

ตารางที่ 2.2 ตารางแสดงความสัมพันธ์ระหว่างเลขฐานแปดหรือฐานสิบหกกับเลขฐานสองที่สมมูลของเลขจำนวนฐานแปดและฐานสิบหก

จากความสัมพันธ์ที่ได้กล่าวมายังสามารถนำไปใช้ในการแปลงเลขฐานสองให้เป็นฐานแปดหรือฐานสิบหกที่สมมูลกันได้ ด้วยการจัดกลุ่มเลขฐานสองเป็นย่อยๆ กลุ่มละ 3 หรือ 4 บิต ที่แทนด้วยเลขฐานแปด และฐานสิบหกที่สมมูลกันตามลำดับ

ข้อสำคัญที่ต้องจำไว้สองข้อสำหรับการจัดกลุ่มย่อยของบิต คือ

1. การจัดกลุ่มต้องเริ่มต้นที่จุดฐานสอง ด้วยเหตุนี้การจัดกลุ่มจะเริ่มโดยการเคลื่อนออกไปทางซ้ายผ่านส่วนที่เป็นจำนวนเต็มและเคลื่อนออกไปทางขวาผ่านส่วนที่เป็นเศษส่วนของเลขฐานสอง
2. ใส่ 0 เข้าไปข้างหน้าและตามหลังได้ตามที่จำเป็น เพื่อให้เลขฐานสองขึ้นต้นและจบด้วยกลุ่มของ 3 หรือ 4 บิต

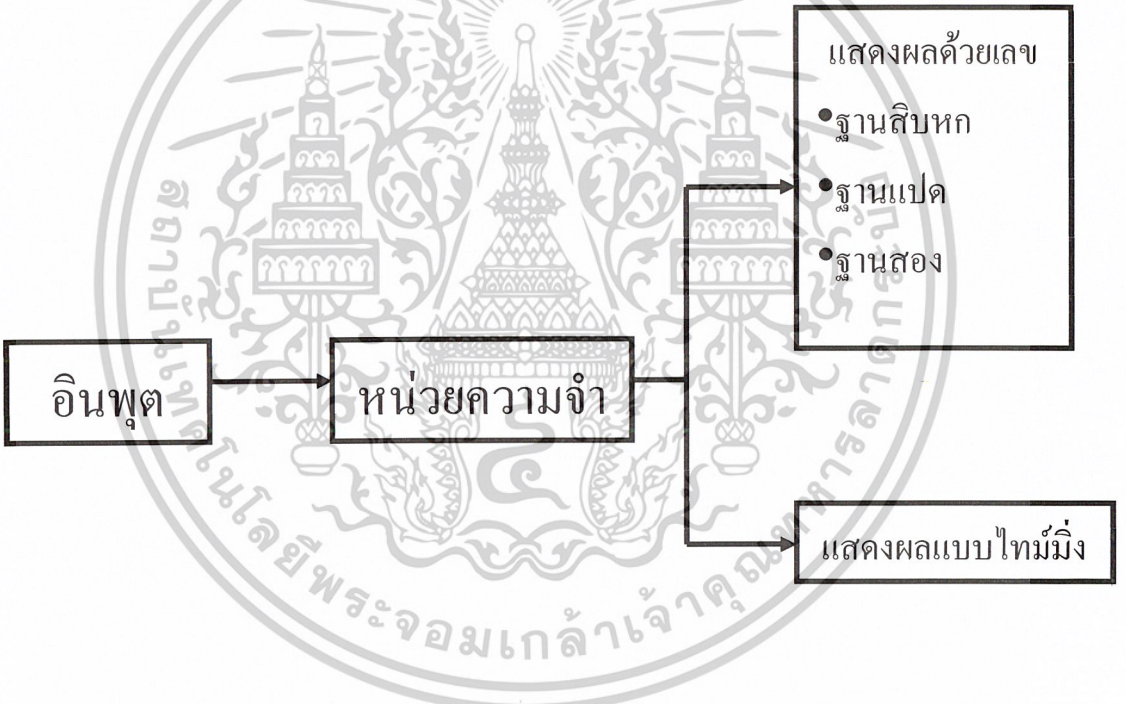
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ลอจิกอนาไลซ์เซอร์

ลอจิกอนาไลซ์เซอร์ เป็นเครื่องมือวิเคราะห์สัญญาณทางตรรกะ ที่สามารถวัดสัญญาณดิจิตอลได้พร้อมๆกันถึง 16 สัญญาณ หรือ 16 ช่อง และมีหน่วยความจำที่ใช้สำหรับเก็บข้อมูล สามารถที่จะนำข้อมูลมาแสดงผลได้อีก หลังจากที่สัญญาณเกิดขึ้นแล้ว

2.2.1 ลักษณะการทำงานทั่วไปของลอจิกอนาไลซ์เซอร์

รูปที่2.1 แสดงให้เห็นหลักการทำงานของลอจิกอนาไลซ์เซอร์ โดยสัญญาณอินพุตที่ผ่านการสุ่มได้มาจากสัญญาณนาฬิกาภายในหรือภายนอก จากนั้นนำเอาข้อมูลไปเก็บในหน่วยความจำและนำข้อมูลในหน่วยความจำดังกล่าวไปแสดงผลที่เอาต์พุตเช่น จอมอนิเตอร์ โดยอาจจะแสดงในรูปของตัวเลข (Stage Display) หรือไทม์มิ่ง (Timing Display)

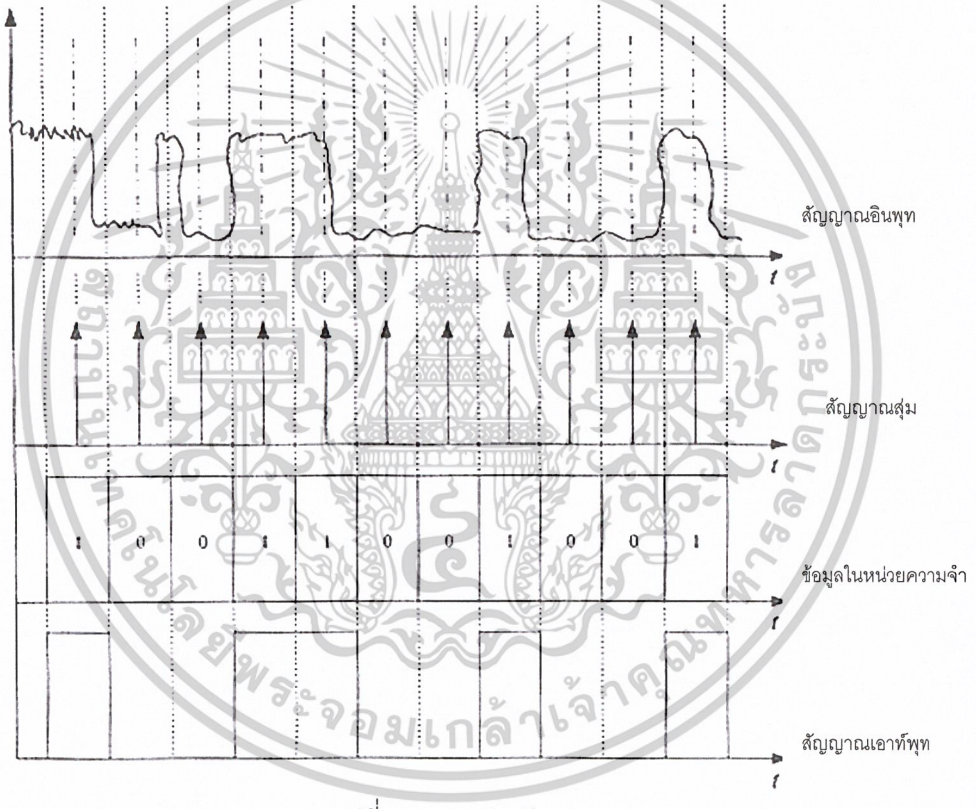


รูปที่ 2.1 หลักการทำงานของลอจิกอนาไลซ์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 หลักการสุ่ม (Sampling) สัญญาณ

การสุ่มสัญญาณ หมายถึง การสุ่มสัญญาณเป็นช่วง จากรูปที่2.2 จะเห็นว่าสัญญาณการสุ่มจะเป็นพัลซ์แคบๆ สัญญาณการสุ่มข้อมูลจะต้องเป็นสัญญาณนาฬิกา ที่มีความถี่มากกว่าสัญญาณที่เข้ามาทำการสุ่มอย่างน้อย 2 เท่า หรือมากกว่า เพื่อที่จะทำให้สัญญาณที่สุ่มออกมา มีประสิทธิภาพและมีค่าความถูกต้องสูง โดยในที่นี้จะทำการสุ่มสัญญาณเข้าไปเก็บในหน่วยความจำ คือนำสัญญาณที่ได้เป็นช่วงๆ ไปเก็บในหน่วยความจำ จากนั้นนำข้อมูลจากหน่วยความจำไปทำการแสดงผลบนจอของออสซิลโลสโคป และแสดงผลออกมาในรูปไทม์มิ่งหรือเลขไบนารี ซึ่งมีลักษณะของสัญญาณดังรูปที่ 2.2

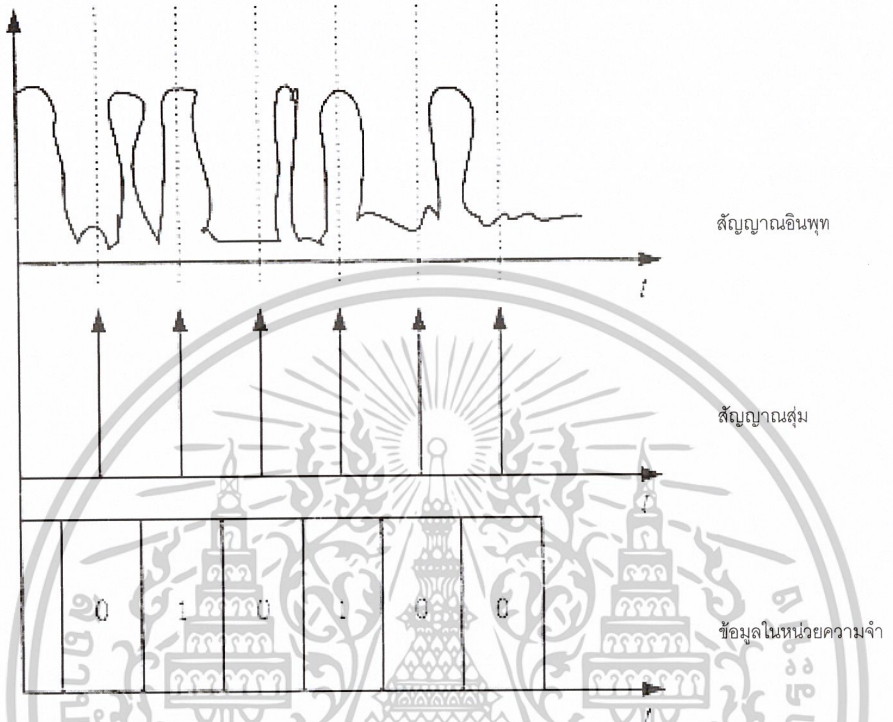


รูปที่ 2.2 การสุ่มสัญญาณ

รูปที่2.2 ถ้าสัญญาณการสุ่มตรงกับสัญญาณอินพุตที่มีค่าแรงดัน 2-5 โวลต์(TTL) จะให้สถานะเป็นลอจิก1 นำค่าลอจิก1 ดังกล่าวไปเก็บในหน่วยความจำ และถ้าสัญญาณการสุ่มตรงกับสัญญาณอินพุตที่มีค่าแรงดัน 0-0.8 โวลต์ เป็นลอจิก 0 นำค่าลอจิก 0 ดังกล่าวไปเก็บไว้ในหน่วยความจำและจะทำการสุ่มเช่นนี้ไปเรื่อยๆ จนกระทั่งข้อมูลเต็มหน่วยความจำ จากนั้นนำข้อมูลที่ได้ไปแสดงผลบนจอออสซิลโลสโคป แสดงผลออกมาในรูปไทม์มิ่งหรือเลขไบนารีก็ได้ รูปที่ 2.2 แสดงให้เห็นว่าสัญญาณเอาต์พุตที่ได้จะมีค่าความถูกต้องใกล้เคียงกับสัญญาณอินพุตมาก คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

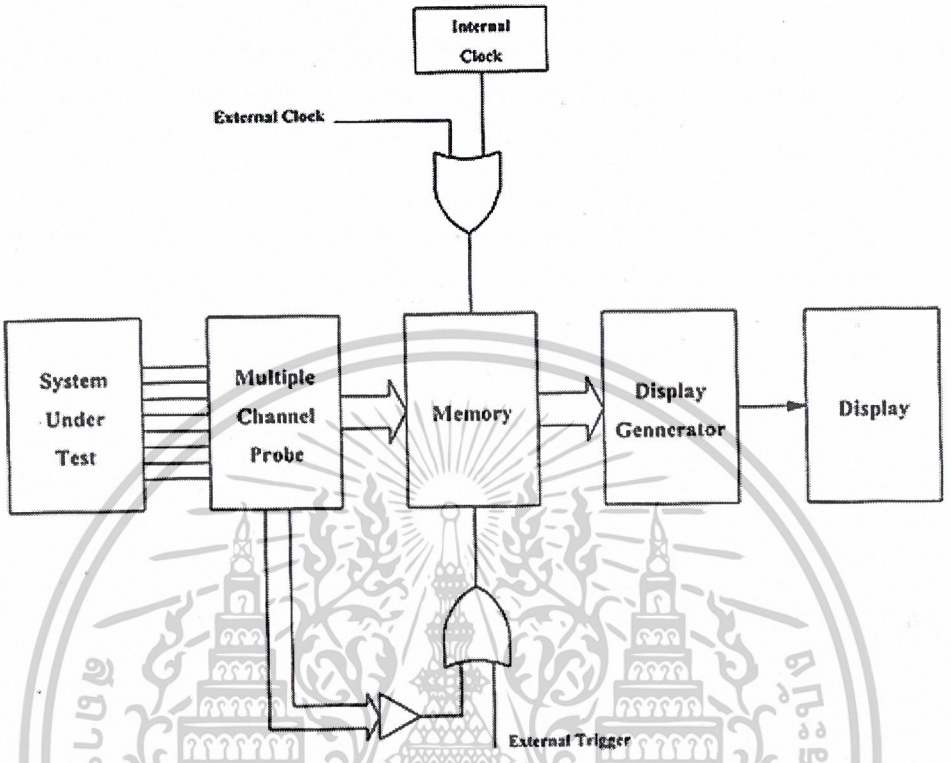
สัญญาณในการสุ่มจะต้องมีค่ามากกว่าสัญญาณอินพุตที่เข้ามาอย่างน้อย 2 เท่าหรือมากกว่า ถ้าสัญญาณสุ่มมีค่าน้อยกว่า 2 เท่าของความถี่อินพุต ข้อมูลที่ได้จะมีความผิดพลาดสูง ดังรูปที่2.3



รูปที่ 2.3 การสุ่มที่มีขนาดของสัญญาณสุ่มน้อยกว่า 2 เท่าของสัญญาณอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์



รูปที่ 2.4 บล็อกไดอะแกรมโดยทั่วไปของลอจิกอนาไลซ์เซอร์

จากคุณสมบัติของเครื่องลอจิกอนาไลซ์เซอร์ ที่สามารถรับสัญญาณดิจิตอล ได้หลายช่องสัญญาณในเวลาเดียวกัน และสัญญาณที่ออกมาจะเป็นในลักษณะข้อมูลดิจิตอลเป็นลำดับ (Sequential Digital Data) และสามารถที่จะแสดงรูปแบบของสัญญาณได้หลายแบบเช่น ไทม์มิ่ง (Timing), ฐานสอง (Binary), ฐานสิบหก (Hex.) ดังนั้น จากบล็อกไดอะแกรมสามารถอธิบายการทำงานได้ดังนี้

การทำงานของเครื่องลอจิกอนาไลซ์เซอร์

1. System Under Test คืออุปกรณ์ที่จะนำมาใช้ทำการวัดสัญญาณจากเครื่องลอจิกอนาไลซ์เซอร์
2. Multiple channel probe คือ ตัวตรวจจับสัญญาณจาก System Under Test อาจจะมี 8 Channel หรือ 16 Channel ตามแต่คุณสมบัติของเครื่องนั้นๆ
3. Memory คือ หน่วยความจำที่ใช้ในการเก็บข้อมูลที่ได้จากการวัดสัญญาณเข้ามา ซึ่งในการเก็บข้อมูลของเครื่องลอจิกอนาไลซ์เซอร์นี้ จะทำการเก็บแบบ FIFO(first in first out)
4. Display Generator คือ ส่วนที่กำเนิดสัญญาณที่จะแสดงทางหน้าจอ
5. Display คือ ตัวแสดงสัญญาณที่วัดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Inter-External clock คือ สัญญาณนาฬิกาที่ทำหน้าที่ให้สัญญาณที่วัดเกิดการพร้อมสัมพันธ์(Sync) กันระหว่างสัญญาณที่นำมาวัด(เข้ามา) และสัญญาณของเครื่องจะทำให้สัญญาณที่วัดได้มีความถูกต้องและเที่ยงตรงสูง
7. Trigger คือ ส่วนที่ทำการผลิตสัญญาณพัลซ์ขึ้นมา เพื่อที่จะทำให้เกิดสัญญาณการเริ่มสุ่มข้อมูลที่จะไปเก็บในหน่วยความจำ โดยทั่วไปการทริกมี 3 ประเภทดังนี้
   - การทริกภายนอก สัญญาณชนิดนี้จะส่งพัลซ์จากภายนอกเข้าไป เพื่อทำให้เกิดการเริ่มสุ่มข้อมูลจากอินพุต ดังนั้น การเก็บข้อมูลในหน่วยความจำ(Memory)ก็จะเริ่มขึ้น
   - การทริกภายใน จะมีการทริกด้วยสัญญาณ ทริกเกอร์เวอร์ด
   - การทริกด้วยมือ(Manual) เป็นการกระตุ้นที่ทำให้เกิดสัญญาณพัลซ์ด้วยมือ โดยผู้ใช้ทำการกดเองตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถาปัตยกรรมพีซีไอ

สถาปัตยกรรมพีซีไอ(Peripheral Component Interconnect; PCI) เป็นบัสที่กว้างมากและมีความเป็นอิสระจากโปรเซสเซอร์ที่ได้รับความนิยมเป็นอย่างมาก โดยเฉพาะในเครื่องคอมพิวเตอร์ส่วนบุคคลที่นำมาใช้เป็นบัสสำหรับอุปกรณ์ไอโอ บัสพีซีไอมีประสิทธิภาพสูงมากสำหรับอุปกรณ์ไอโอความเร็วสูงในยุคปัจจุบัน เช่น การ์ดสำหรับจอแสดงผล การ์ดสำหรับระบบเครือข่าย (LAN) หน่วยควบคุมดิสก์ และอื่นๆ มาตรฐานในปัจจุบัน กำหนดให้ใช้สายสัญญาณข้อมูล 64 เส้น ทำงานที่ความเร็ว 66 MHz ซึ่งจะทำให้ความเร็วในการถ่ายเทข้อมูลสูงถึง 528 ล้านไบต์ต่อวินาที หรือประมาณ 4.224 พันล้านบิตต่อวินาที นอกเหนือจากนี้ บัสพีซีไอยังได้รับการออกแบบมาเพื่อความประหยัดในการออกแบบอุปกรณ์ไอโอต่างๆ บัสพีซีไอใช้ชิพควบคุมการทำงานเพียงไม่กี่ตัว และยังสามารถสนับสนุนบัสอื่นได้อีกด้วย

บัสพีซีไอได้รับการพัฒนามาสนับสนุนการใช้งานไมโครโปรเซสเซอร์หลายชนิด โครงสร้างหลักของบัสพีซีไอประกอบด้วยกลุ่มฟังก์ชันใช้งานทั่วไป ใช้ระบบสัญญาณนาฬิกาแบบซิงโครนัสและใช้การควบคุมแบบบัสแบบศูนย์กลาง

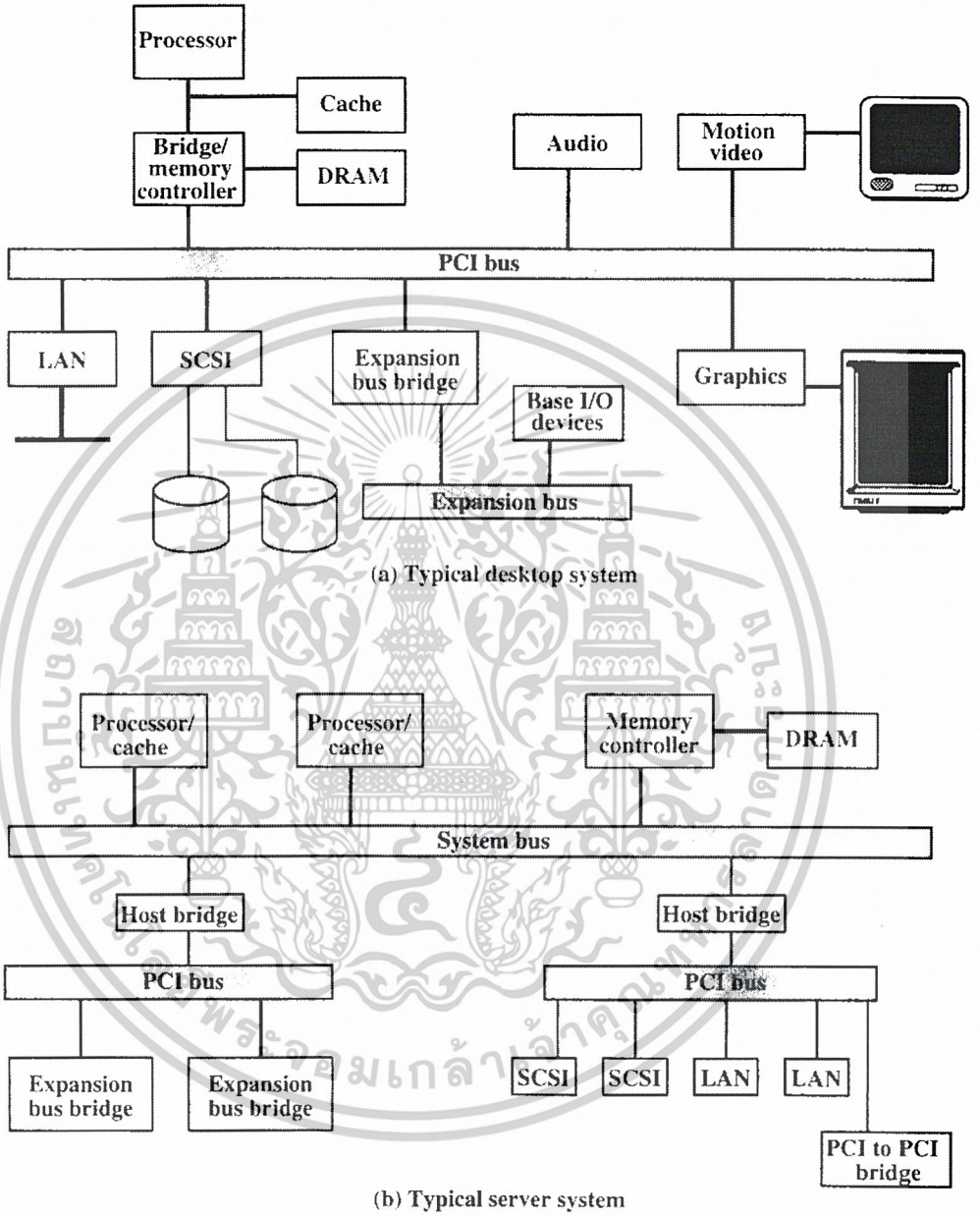
รูป 2.5 a แสดงให้เห็นถึงการใช้บัสพีซีไอในระบบที่มีโปรเซสเซอร์เดี่ยว การใช้หน่วยควบคุม DRAM ทำหน้าที่เป็นสะพานเชื่อม(bridge)กับบัสพีซีไอ ช่วยให้การทำงานร่วมกับโปรเซสเซอร์เป็นไปได้อย่างได้กระชับและมีความเร็วในการทำงานสูง หน่วยควบคุมฯ ทำหน้าที่เป็นบัฟเฟอร์ไปในตัว จึงช่วยให้ความเร็วในการทำงานของบัสพีซีไออาจแตกต่างไปจากความเร็วของซีพียูได้ ส่วนในระบบที่มีโปรเซสเซอร์หลายตัว ดังแสดงในรูป 2.5 b ระบบอาจประกอบด้วยบัสพีซีไอได้มากกว่าหนึ่งบัส ซึ่งจะเชื่อมต่อเข้ากับบัสหลักของระบบที่สนับสนุนการทำงานเฉพาะหน่วยความจำหลัก และสะพานเชื่อมต่อกับบัสพีซีไอ(host bridge) ในทำนองเดียวกัน การใช้สะพานเชื่อมช่วยทำหน้าที่เป็นบัฟเฟอร์ ทำให้โปรเซสเซอร์และบัสพีซีไอทำงานแยกจากกัน และช่วยทำให้การรับ-ส่งข้อมูลเป็นไปได้อย่างราบรื่น

2.3.1 โครงสร้างแบบบัส

บัสพีซีไออาจจัดให้ทำงานในระบบ 32 บิต หรือ 64 บิตก็ได้ ตาราง2.3 แสดงสัญญาณที่บังคับใช้งานจำนวน 49 สายสัญญาณสำหรับบัสพีซีไอ ซึ่งแบ่งออกตามหน้าที่เป็นกลุ่มดังนี้

- System pins: ประกอบด้วยสัญญาณนาฬิกา และสัญญาณรีเซ็ตระบบ(reset)
- Address and Data pins: ประกอบด้วยสายสัญญาณ 32 เส้น ซึ่งสลับกันทำหน้าที่เป็นสายสัญญาณสำหรับบอกตำแหน่งและสายข้อมูล สายอื่นๆ ในกลุ่มนี้จะทำหน้าที่ในการแปลความหมาย และตรวจสอบสายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างสถาปัตยกรรมพีซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Interface Control pins: ควบคุมจังหวะเวลาของรายการธุรกรรม(transaction) และจัดเตรียมการร่วมมือระหว่างตัวบอกสถานะและเป้าหมาย
- Arbitration pins: สายในกลุ่มนี้ไม่ใช่สายที่ใช้งานร่วม อุปกรณ์ไอโอแต่ละตัวที่สามารถทำหน้าที่เป็น master จะมีสายสั่งการเป็นของตนเองที่เชื่อมต่อกับส่วนควบคุมการใช้บัส (arbiter) ของหน่วยควบคุมพีซีไอ
- Error Reporting pins: ใช้ในการรายงานความผิดพลาดที่อาจเกิดขึ้น
- นอกเหนือจากข้อกำหนดของบัสพีซีไอข้างบนนี้ ยังได้มีข้อกำหนดเพิ่มเติมของสายสัญญาณอีก 51 เส้น ดังแสดงในตาราง 2.4 ซึ่งเป็นส่วนที่อาจมีหรือไม่มีใช้ก็ได้ ประกอบด้วยกลุ่มต่างๆดังนี้
- Interrupt pins: เป็นส่วนที่เตรียมไว้สำหรับอุปกรณ์ที่สามารถสร้างสัญญาณอินเทอร์รัพท์ได้ ซึ่งเป็นสายเฉพาะที่ไม่ได้ใช้งานร่วมเหมือนสาย arbitration pins นั่นคืออุปกรณ์แต่ละตัวจะมีสาย interrupt เป็นของตนเองที่เชื่อมต่อเข้ากับหน่วยควบคุมอินเทอร์รัพท์
- Cache Support pins: พินเหล่านี้สนับสนุนการทำงานของหน่วยความจำบนการ์ดอุปกรณ์พีซีไอที่ช่วยในการถ่ายเทข้อมูลระหว่างโปรเซสเซอร์กับอุปกรณ์ไอโอ หน่วยความจำนี้ทำงานในแบบที่เรียกว่า snoopy cache protocol
- 64-bit Bus Extension pins: ประกอบด้วยสายสัญญาณ 32 เส้น ที่ทำงานสลับระหว่างการส่งตำแหน่งที่อยู่กับข้อมูลที่รวมเข้ากับสายสัญญาณปกติอีก 32 เส้น กลายเป็นสายสัญญาณขนาด 64 บิต สายสัญญาณอื่นในกลุ่มนี้ใช้ในการแปลความหมาย และการตรวจสอบสัญญาณ และประกอบด้วยสายสัญญาณอีก 2 เส้นที่ใช้ในการตกลงอุปกรณ์พีซีไอ 2 ตัว ในการใช้สาย 64 เส้นในการสื่อสาร
- JTAG/Boundary Scan pins: สายสัญญาณนี้ สนับสนุนการทดสอบฟังก์ชันการทำงานการทำงานตามมาตรฐาน IEEE1149.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งพีซีไอ

กิจกรรมที่เกิดขึ้นอยู่ในรูปแบบของรายการธุรกรรม (transaction) ระหว่างผู้เริ่มการติดต่อ (Initiator or master) กับเป้าหมาย (target) เมื่อผู้เริ่มการติดต่อต้องการควบคุมบัส ประการแรกจะต้องกำหนดประเภทของรายการธุรกรรมที่กำลังจะเกิดขึ้น ในระหว่างการส่งตำแหน่งที่อยู่ของรายการธุรกรรม สายสื่อสาร C/BR จะถูกนำมาใช้ในการกำหนดประเภทนี้ คำสั่งที่ใช้คือ

Destination	Type	Description
System Pins		
CLK	in	Provides timing for all transactions and is sampled by all inputs on the rising edge. Clock rates up to 33 MHz are supported.
RST#	in	Forces all PCI-specifics, sequencers, and signals to an initialized state.
Address and Data Pins		
AD[31::0]	t/s	Multiplexed lines used for address and data.
C/BE[3::0]#	t/s	Multiplexed bus command and byte enable signals. During the data phase, the lines indicate which of the four byte lanes carry meaningful data.
PAR	t/s	Provides even parity across AD and C/BE lines one clock cycle later. The master drives PAR for address and write data phases; the target drives PAR for read data phases.
Interface Control Pins		
FRAME#	s/t/s	Driven by current master to indicate the start and duration of a transaction. It is asserted at the start and deasserted when the initiator is ready to begin the final data phases.
IRDY#	s/t/s	Initiator Ready. Driven by current bus master (initiator of transaction). During a read, indicates that the master is prepared to accept data; during a wire, indicates that valid data are present on AD
TRDY#	s/t/s	Target Ready. Driven by the target (selected device). During a read, indicates that valid data are present on AD; during a write, indicates that target is ready to accept data.
STOP#	s/t/s	Indicates that current target wishes the initiator to stop the current transaction.
IDSEL	in	Initialization Device Select. Used as a chip select during configuration read and write transactions.
DEVSEL	in	Device Select. Asserted by target when it has recognized its address. Indicates to current initiator whether any device has been selected.
Arbitration Pins		
REQ#	t/s	Indicates to the arbiter that this device requires use of the bus. This is a device-specific point-to-point line.
GNT#	t/s	Indicates to the device that the arbiter has granted bus access. This is a device-specific point-to-point line.
Error Reporting Pins		
PERR#	s/t/s	Parity Error.Indicates a data parity error is detected by a target during a write data phase or by an initiator during a reading data phase
SERR#	o/d	System Error. May be pulsed by any device to report address parity errors and critical errors other than parity.

ตารางที่ 2.3 สายสัญญาณที่สำคัญของสถาปัตยกรรมพีซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 สายสัญญาณส่วนที่เพิ่มเติมของสถาปัตยกรรมพีซีไอ

Designation	Type	Description
Interrupt Pins		
INTA#	o/d	Used to request an interrupt.
INTB#	o/d	Used to request an interrupt; only has meaning on a multifunction device.
INTC#	o/d	Used to request an interrupt; only has meaning on a multifunction device.
INTD#	o/d	Used to request an interrupt; only has meaning on a multifunction device.
Cache Support Pins		
SBO#	in/out	Snoop Backoff. Indicates a hit to a modified line.
SDONE#	in/out	Snoop Done. Indicates the status of snoop for the current accent. Asserted when snoop has been completed.
64-bit Bus Extension Pins		
AD[63::32]	t/s	Multiplexed lines used for address and data to extend bus to 64 bits.
C/BE[7::4]#	t/s	Multiplexed bus command and byte enable signals. During the address phase, the lines provide additional bus commands. During the data phase, the lines indicate which of the four extented byte lanes carry meaningful data.
REQ64#	s/t/s	Used to request 64-bit transfer.
ACK64#	s/t/s	Indicates target is willing to perform 64-bit transfer.
PAR64	t/s	Provides even parity across extended AD and C/BE lines one clock cycle later.
JTAG/Boundary Scan Pins		
TCK	in	Test Clock. Used to clock state information and test data into and out of the device during boundary scan.
TDI	in	Test Input. Used to serially shift test data and instructions into the device.
TDO	out	Test Output. Used to serially shift test data and instructions out of the device.
TMS	in	Test Mode Select. Used to control state of test access port controller.
TRST#	in	Test Reset Select. Used to initialize test access port controller.

in Input-only signal

out Output-only signal

t/s Bidirectional, Tri-state, I/O signal

s/t/s Sustained tri-state signal driven by only one owner at a time

o/d Open drain: allows multiple devices to share as a wire-OR

SignalOs active state occurs at low voltage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Interrupt acknowledge: การตอบรับอินเทอร์รัพท์ เป็นคำสั่งสำหรับการอ่านที่สร้างขึ้นมาสำหรับอุปกรณ์ที่ทำหน้าที่เป็นหน่วยควบคุมอินเทอร์รัพท์บนบัสพีซีไอ สายสัญญาณตำแหน่งข้อมูลไม่ถูกนำมาใช้ในขั้นตอนนี้
- Special cycle: คำสั่งจังหวะวงรอบพิเศษถูกนำมาใช้ โดยผู้ที่เริ่มต้นการสื่อสารบนบัสพีซีไอ ในการกระจายข่าวสารไปยังหลายเป้าหมายพร้อมๆกัน
- I/O Read and Write: การอ่านและบันทึกไอโอ ใช้ในการถ่ายโอนข้อมูลระหว่างผู้เริ่มการติดต่อและหน่วยควบคุมไอโอ อุปกรณ์ไอโอแต่ละตัวจะมีตำแหน่งที่เป็นอยู่เป็นของตัวเองและสายสัญญาณบอกตำแหน่งที่อยู่ ถูกนำมาใช้ระบุอุปกรณ์ไอโอที่ต้องการ และบอกข้อมูลที่ต้องการถ่ายโอนไปยัง หรือมาจากอุปกรณ์นั้นๆ
- Memory Read: การอ่านหน่วยความจำ Memory Read Line สายสัญญาณสำหรับการอ่านหน่วยความจำ และ Memory Read Multiple การอ่านหน่วยความจำหลายครั้ง ถูกนำมาใช้ในการกำหนดการถ่ายโอนข้อมูล ซึ่งเกิดขึ้นในช่วงเวลาหนึ่งวงรอบนาฬิกา การแปลความหมายของคำสั่งขึ้นอยู่กับว่า หน่วยควบคุมหน่วยความจำบนบัสพีซีไอ สนับสนุนการสื่อสารระหว่างหน่วยความจำหลัก กับหน่วยความจำ cache หรือไม่ ถ้ามีการสนับสนุน การถ่ายโอนข้อมูลเข้าสู่หรือออกจากหน่วยความจำจะกระทำผ่านสายสัญญาณ cache lines หรือเป็นลักษณะของบล็อก คำสั่งสำหรับการอ่านข้อมูลทั้งสามถูกอธิบายไว้ในตาราง 3.5 คำสั่ง Memory Write ใช้สำหรับการถ่ายโอนข้อมูลเข้าสู่หน่วยความจำในหนึ่งหน่วยวงรอบนาฬิกาหรือมากกว่านี้ คำสั่ง Memory Write and Invalidate บันทึกข้อมูลลงหน่วยความจำในหนึ่งวงรอบนาฬิกา หรือมากกว่านี้ และให้การรับประกันว่าจะมีข้อมูลในสาย cache line อย่างน้อยหนึ่งเส้นจะถูกบันทึกลงในหน่วยความจำหลัก ซึ่งเป็นการทำหน้าที่บันทึกข้อมูลจาก cache ลงหน่วยความจำหลัก
- Configuration Read: การอ่านสถานะระบบ และ Configuration Write การบันทึกสถานะระบบ ช่วยให้อุปกรณ์ที่ทำหน้าที่เป็น master อ่านและปรับปรุงข้อมูลที่นำมาใช้ในการควบคุมสถานการณ์ทำงาน (configuration parameters) ในอุปกรณ์ที่เชื่อมต่อกับบัสพีซีไอ ซึ่งอุปกรณ์พีซีไอแต่ละชิ้น อาจมีรีจิสเตอร์ควบคุมการทำงานมากถึง 256 ตัว ที่นำมาใช้ในระหว่างการเตรียมอุปกรณ์ให้พร้อมใช้งาน และการเปลี่ยนแปลงสถานการณ์ทำงานของอุปกรณ์นั้นๆ
- Dual Access Cycle: การใช้สายบัสคู่ ถูกใช้โดยผู้เริ่มต้นการสื่อสารในการระบุว่าการสื่อสารครั้งนี้ใช้สายทั้งสิ้น 64 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

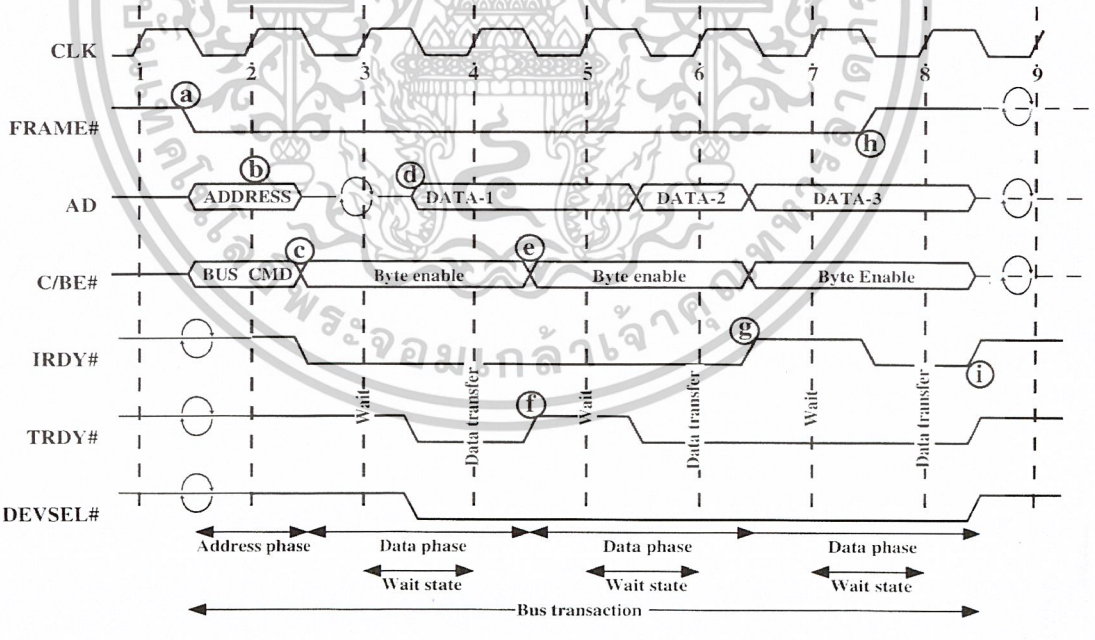
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตาราง2.5 การแปลความหมายคำสั่งอ่านข้อมูลในสถาปัตยกรรมพีซีไอ

Read Command Type	For Cachable Memory	For noncachable Memory
Memory Read	Bursting one-half or less of a cache line	Bursting 2 data transfer cycle or less
Memory Read Line	Bursting more than one-half a cache line to three cache lines	Bursting 3 to 12 data transfers
Memory Read Multiple	Bursting more than three cache lines	Bursting more than 12 data transfers

2.3.2 การเคลื่อนย้ายข้อมูล

การเคลื่อนย้ายข้อมูลแต่ละครั้งบนบัสพีซีไอ เรียกว่าเกิดการทำธุรกรรม (transaction) ครั้งหนึ่งซึ่งประกอบด้วย ขั้นตอนการกำหนดตำแหน่งที่อยู่ (address phase) หนึ่งครั้ง และขั้นตอนการเคลื่อนย้ายข้อมูล (data phase) อย่างน้อยหนึ่งครั้ง ต่อไปนี้จะอธิบายการอ่านข้อมูลที่เกิดขึ้นหนึ่งครั้ง ส่วนการบันทึกข้อมูลนั้นก็เกิดขึ้นในลักษณะเดียวกัน



รูปที่ 2.6 การอ่านข้อมูลในสถาปัตยกรรมพีซีไอ

58707

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.6 แสดงจังหวะเวลาที่เกิดขึ้นในการทำธุรกรรมอ่านข้อมูล เหตุการณ์ทั้งหมดจะเกิดขึ้นประมาณช่วงกลาง และเสร็จสิ้นในจังหวะเดียวกัน ซึ่งเป็นช่วงสุดท้ายของแต่ละวงรอบนาฬิกา อุปกรณ์ในบัสจะตรวจสอบสัญญาณในบัสตอนเริ่มต้นของวงรอบนาฬิกา ต่อไปนี้คือเหตุการณ์ที่เกิดขึ้นตามลำดับ

a) เมื่ออุปกรณ์ master หรือผู้เริ่มการสื่อสารได้เข้าควบคุมการใช้งานบัสแล้ว ก็จะเริ่มกระบวนการทำธุรกรรมด้วยการใส่สัญญาณ FRAME ซึ่งจะคงอยู่อย่างนี้จนกระทั่ง master ได้ส่งสัญญาณข้อมูลชุดสุดท้ายออกไปแล้ว ผู้เริ่มการสื่อสารจะใส่ตำแหน่งที่อยู่เป้าหมายเข้าไปที่สายสัญญาณบอกตำแหน่งที่อยู่ด้วย และใส่คำสั่งอ่านข้อมูลเข้าไปที่สายสัญญาณ C/BE ดังในรูป

b) ที่จุดเริ่มต้นของวงรอบนาฬิกาที่สอง อุปกรณ์เป้าหมายจะทราบตำแหน่งที่ถูกอ้างอิงถึงจากสาย AD

c) ผู้เริ่มการสื่อสารจะยืดจังหวะการส่งสัญญาณในสาย AD ซึ่งจะทำให้เกิด turnaround cycle (แสดงด้วยลูกศรวงเป็นวงกลม) ที่จะเกิดขึ้นกับสายสัญญาณทุกเส้น เพื่อเป็นการล้างสัญญาณที่อาจค้างอยู่จากคำสั่งก่อนหน้านี้ และเป็นการเตรียมบัสไว้สำหรับอุปกรณ์เป้าหมาย ผู้เริ่มทำการสื่อสารจะเปลี่ยนสัญญาณบนสาย C/BE เพื่อบอกให้ทราบว่าสาย AD เส้นใดที่ถูกนำมาใช้ในการถ่ายเทข้อมูล ผู้เริ่มการสื่อสารจะใส่สัญญาณ IRDY เข้าไปด้วย เพื่อบอกให้ทราบว่าตนเองพร้อมจะส่งข้อมูลชุดแรกแล้ว

d) เป้าหมายจะส่งสัญญาณ DEVSEL เพื่อบอกให้ทราบว่าตนเองได้เตรียมการรับข้อมูลพร้อมแล้ว จากนั้นจะส่งสัญญาณขอข้อมูลมาทางสาย AD และใส่สัญญาณ TRDY เข้ามาเพื่อยืนยันว่า ข้อมูลในสายข้อมูลนั้นเรียบร้อยดี

e) ผู้เริ่มทำการสื่อสารจะอ่านข้อมูลที่ตอนต้นของวงรอบนาฬิกาที่สี่ และเปลี่ยนสัญญาณในสาย Enable line ให้สอดคล้องกับการอ่านข้อมูลในชุดต่อไป

f) ในตัวอย่างนี้ อุปกรณ์เป้าหมายต้องการระยะเวลาระยะหนึ่ง เพื่อเตรียมการสำหรับการส่งข้อมูลในชุดที่สอง (และชุดต่อๆไป) ดังนั้นจึงหยุดส่งสัญญาณ TRDY เพื่อบอกให้ผู้เริ่มการสื่อสารทรายว่าจะไม่มีข้อมูลส่งมาในวงรอบนาฬิกาถัดไป ซึ่งผู้เริ่มการสื่อสารก็จะไม่อ่านข้อมูลในจังหวะนาฬิกาที่ห้า แต่ยังคงส่งสัญญาณ Byte Enable ไว้ตามเดิม ข้อมูลชุดที่สองจะถูกอ่านในวงรอบนาฬิกาที่หก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

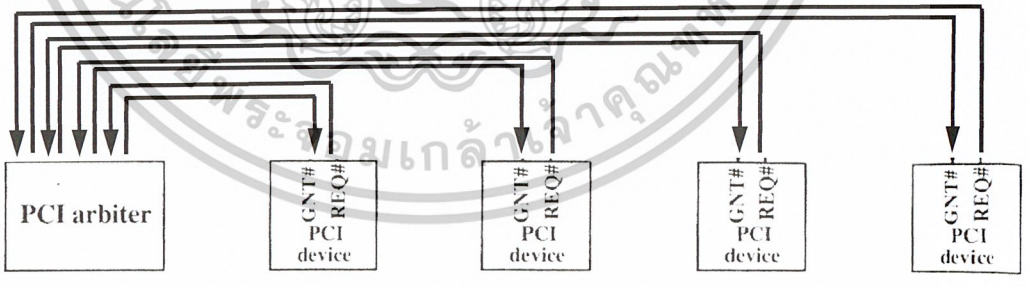
g) ในระหว่างวงรอบนาฬิกาที่หก อุปกรณ์เป้าหมายจะใส่ข้อมูลชุดที่สามเข้าไปในบัส แต่อย่างไรก็ตามในตัวอย่างนี้ ผู้เริ่มการสื่อสารยังไม่พร้อมที่จะรับข้อมูลชุดที่สาม (เช่น บัฟเฟอร์เต็ม) ดังนั้นจึงหยุดส่งสัญญาณ IRDY ซึ่งจะเป็นผลให้ข้อมูลชุดที่สามค้างอยู่ในบัสนานกว่าปกติหนึ่งวงรอบนาฬิกา

h) ผู้เริ่มการสื่อสารทราบว่าข้อมูลชุดที่สามเป็นชุดสุดท้าย จึงหยุดการส่งสัญญาณ FRAME เพื่อบอกให้อุปกรณ์เป้าหมายทราบว่า นี่คือการส่งสัญญาณข้อมูลชุดสุดท้าย ผู้เริ่มการสื่อสารจะต้องส่งสัญญาณ IRDY ออกไป เพื่อบอกให้ทราบว่าตนเองพร้อมที่จะรับข้อมูลชุดสุดท้าย

i) ผู้เริ่มการสื่อสารจะหยุดการส่งสัญญาณ IRDY เพื่อปล่อยให้บัสกลับสู่สถานะว่าง และอุปกรณ์เป้าหมายก็จะหยุดส่งสัญญาณ TRDY และ DEVSEL ด้วย

2.3.3 การตัดสินเลือกผู้ควบคุมบัส

บัสพีซีไอมีการทำงานแบบรวมศูนย์ และเป็นระบบซิงโครนัส สำหรับการตัดสินเลือกผู้ควบคุมการใช้บัส โดยการที่อุปกรณ์ Master แต่ละตัวจะมีสายความต้องการควบคุมบัส REQ และ GNT เป็นของตนเอง สายสัญญาณนี้จะต่อเข้ากับผู้ตัดสินการควบคุมบัสส่วนกลาง (Central Arbiter) ดังแสดงในรูป 2.7 การควบคุมบัสจึงเกิดขึ้นจากการใช้วิธีการง่ายๆ ในการแลกเปลี่ยนสัญญาณ request-grant หรือการร้องขอและการอนุมัตินั่นเอง



รูปที่ 2.7 PCI Arbiter ในสถาปัตยกรรมพีซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในข้อกำหนดของมาตรฐานบัสพีซีไอ ไม่ได้กำหนดวิธีการตายตัวสำหรับการคัดเลือกผู้ควบคุมบัสเอาไว้ทำให้ผู้ควบคุมส่วนกลางอาจใช้วิธีการมาก่อน – ได้ก่อน หรือวิธีการใดๆในการตัดสินก็ได้ อย่างไรก็ตามอุปกรณ์ master ในบัสพีซีไอจะต้องไดรับอนุญาตให้เป็นผู้ควบคุมบัสก่อนที่จะทำการใดๆเกี่ยวกับบัส (ซึ่งประกอบด้วยขั้นตอนการกำหนดที่อยู่ และขั้นตอนการส่งข้อมูลดังที่กล่าวมาแล้ว)

รูป 2.8 แสดงตัวอย่างที่อุปกรณ์ A และ B พยายามร้องขอการควบคุมบัสซึ่งมีลำดับเหตุการณ์ดังนี้

A. ณ ช่วงเวลาหนึ่งก่อนวงรอบนาฬิกาที่หนึ่ง อุปกรณ์ A ได้ส่งสัญญาณ REQ เข้ามาผู้ควบคุมส่วนกลาง (arbiter) สามารถรับสัญญาณนี้ได้ที่ตอนต้นของวงรอบนาฬิกาที่หนึ่ง

B. ในระหว่างวงรอบนาฬิกาที่หนึ่ง อุปกรณ์ B ร้องขอการใช้บัสด้วยการส่งสัญญาณ REQ ของตนเองเข้ามา

C. ในเวลาเดียวกัน ผู้ควบคุมส่วนกลางได้อนุญาตให้อุปกรณ์ A ควบคุมบัสได้โดยการส่งสัญญาณ GNT-A ไปยังอุปกรณ์ A

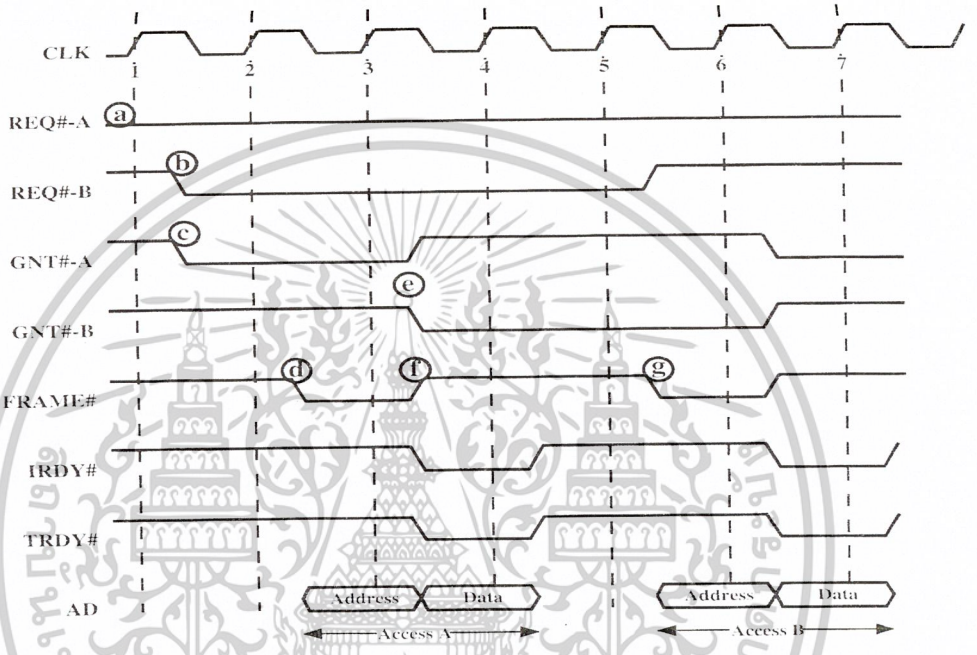
D. อุปกรณ์ A ตรวจพบสัญญาณ GNT-A ที่ตอนเริ่มต้นของวงรอบนาฬิกาที่สอง และในเวลาเดียวกันก็ตรวจไม่พบสัญญาณ IRDY และ TRDY ซึ่งแสดงว่าบัสนั้นว่างจากการใช้งานใดๆ อุปกรณ์ A จึงเริ่มกระบวนการอ่านข้อมูลด้วยการส่งสัญญาณ FRAME และใส่ตำแหน่งที่อยู่เข้าไปในสายสัญญาณบอกตำแหน่งที่อยู่ พร้อมกับส่งคำสั่งอ่านข้อมูลเข้าไปในสาย C/BE (ไม่ได้แสดงไว้- ดูตัวอย่างก่อนหน้านี้) นอกจากนี้ยังคงส่งสัญญาณ REQ-A เพราะต้องการส่งข้อมูลภายหลังจากที่เสร็จสิ้นการทำงานครั้งนี้

E. ผู้ควบคุมส่วนกลางตรวจสอบสัญญาณ REQ ทั้งหมดที่ตอนต้นของวงรอยนาฬิกาที่สามและทำการตัดสินใจ (ด้วยวิธีใดก็ตาม)มอบบัสให้แก่อุปกรณ์ B จึงส่งสัญญาณ GNT-B และหยุดส่งสัญญาณ GNT-A อุปกรณ์ B จะไม่สามารถใช้งานบัสได้จนกว่าบัสจะกลับมาสู่สถานะว่าง (idle)

F. อุปกรณ์ A หยุดส่งสัญญาณ FRAME เพื่อบอกให้อุปกรณ์เป้าหมายได้ทราบว่ากำลังส่งข้อมูลชุดสุดท้ายออกมา จากนั้นจึงใส่ข้อมูลเข้าไปที่สายข้อมูลและส่งสัญญาณ IRDY อุปกรณ์เป้าหมายอ่านข้อมูลที่ตอนต้นของวงรอบนาฬิกาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

G. ที่จุดเริ่มต้นของวงรอบนาฬิกาที่ห้า อุปกรณ์ B ตรวจพบว่าสัญญาณ IRDY และ FRAME ได้หยุดลงไปแล้ว ซึ่งเป็นการบอกว่าอุปกรณ์ B ได้เข้าควบคุมการใช้บัส แล้วจึงส่งสัญญาณ FRAME ของตนออกมาก พร้อมกับหยุดส่งสัญญาณ REQ เนื่องจากตนเองต้องการส่งข้อมูลเพียงรายการเดียว (จากนั้นจึงดำเนินการส่งข้อมูลต่อไป)



รูปที่ 2.8 การควบคุมบัสระหว่างมาสเตอร์ 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โมดูลไอโอ

2.4.1 โมดูลฟังก์ชัน

ฟังก์ชันการทำงานของโมดูลไอโอ สามารถแบ่งออกเป็นกลุ่มๆได้ดังนี้

- การควบคุมและการจับเวลา
- การสื่อสารระหว่างโปรเซส
- การสื่อสารระหว่างอุปกรณ์
- การจัดการบัฟเฟอร์
- การตรวจสอบข้อผิดพลาด

ฟังก์ชันไอโอหมายถึง ความต้องการในการควบคุมและการจับเวลา (Control and timing) เพื่อการประสานงานให้ข้อมูลสามารถไหลผ่านระหว่างทรัพยากรภายในกับอุปกรณ์ภายนอกได้ ตัวอย่างเช่น การควบคุมการถ่ายเทข้อมูลจากอุปกรณ์ภายนอกมายังโปรเซสเซอร์ อาจเกี่ยวพันกับการทำงานดังนี้

1. โปรเซสเซอร์ทำการติดต่อกับโมดูลไอโอ เพื่อตรวจสอบสถานะของอุปกรณ์ที่ต้องการใช้งานนั้น
2. โมดูลไอโอส่งข้อมูลแสดงสถานะกลับมาให้
3. ถ้าอุปกรณ์สามารถใช้งานได้ และพร้อมที่จะถ่ายเทข้อมูล โปรเซสเซอร์จะร้องขอการถ่ายเทข้อมูลผ่านการออกคำสั่งไปยังโมดูลไอโอ
4. โมดูลไอโอได้รับข้อมูลชุดแรก (8 บิต หรือ 16 บิต) มาจากอุปกรณ์ภายนอก
5. ข้อมูลถูกถ่ายทอดจากโมดูลไอโอต่อไปให้กับโปรเซสเซอร์

ถ้าระบบคอมพิวเตอร์มีบัสหลักใช้งาน การโต้ตอบระหว่างโปรเซสเซอร์และโมดูลไอโอจะเกี่ยวข้องกับการถือครองบัส โมดูลไอโอจะต้องสื่อสารกับโปรเซสเซอร์และสื่อสารกับอุปกรณ์ภายนอก การสื่อสารกับโปรเซสเซอร์เกี่ยวข้องกับสิ่งต่อไปนี้

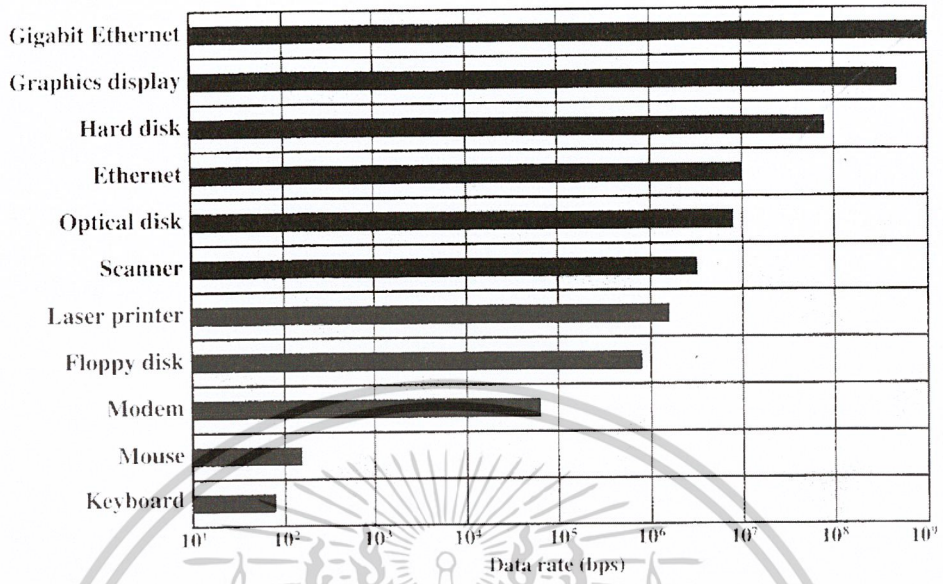
- **การถอดรหัสคำสั่ง** (command decoding): โมดูลไอโอรับคำสั่งจากโปรเซสเซอร์ ซึ่งโดยปกติจะส่งสัญญาณผ่านมาทางบัสควบคุมการทำงาน (control bus) ตัวอย่างเช่น โมดูลไอโอสำหรับตัวขับดิสก์จะรับคำสั่งเช่น READ SECTOR, WRITE SECTOR, SEEK track number, และ SCAN record ID คำสั่งสองคำสั่งหลังจำเป็นต้องมีการส่งผ่านตัวกำหนดค่ามาด้วย ซึ่งจะส่งมาทางบัสสำหรับถ่ายเทข้อมูล (data bus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ข้อมูล** (Data): เนื่องจากอุปกรณ์ประกอบ มักจะมีความเร็วต่ำมาก จึงมีความจำเป็นจะต้องทราบสถานการณ์ทำงานของอุปกรณ์ ตัวอย่างเช่น ถ้าโมดูลไอโอถูกสั่งมาให้โปรเซสเซอร์ (READ) อุปกรณ์ดังกล่าวอาจจะยังไม่ว่าง เนื่องจากกำลังให้บริการแก่คำสั่งที่มาถึงก่อนหน้านี้ ซึ่งโดยทั่วไปมีอยู่สองอย่างคือ BUSY (ไม่ว่าง) และREADY (ว่าง) นอกจากนี้อาจเป็นสัญญาณรายงานสถานะข้อผิดพลาดต่างๆ
- **การรับรู้ที่อยู่** (address recognition): อุปกรณ์ไอโอก็คล้ายกับหน่วยความจำ คือ แต่ละ word มีหมายเลขที่อยู่เฉพาะเป็นของตนเอง นั่นคืออุปกรณ์ไอโอแต่ละชิ้นจะต้องมีหมายเลขเฉพาะสำหรับอ้างอิงเป็นของตนเอง ดังนั้นโมดูลไอโอจะต้องสามารถรับรู้หมายเลขที่อยู่เฉพาะของอุปกรณ์แต่ละชนิดได้

งานที่มีความสำคัญมากที่สุดส่วนหนึ่งของโมดูลไอโอคือ การทำบัฟเฟอร์สำหรับข้อมูล (Data buffering) ความต้องการนี้จะสังเกตได้จากรูป 2.9 อัตราการถ่ายเทข้อมูลไปยังหรือมาจากหน่วยความจำหลัก หรือโปรเซสเซอร์นั้นสูงมาก แต่อัตราการถ่ายเทข้อมูลของอุปกรณ์ต่อพ่วงต่างๆ นั้นช้ามากกว่าหลายเท่าตัว และยังมีการถ่ายเทที่แตกต่างกันในช่วงที่กว้างมากด้วย ข้อมูลที่ถูกส่งมาจากหน่วยความจำหลัก ถูกส่งมายังโมดูลไอโอในลักษณะที่มีความเร็วสูงมากแต่เป็นช่วงสั้นๆ เรียกว่า เบิรซท์ (burst) ข้อมูลนั้นจะถูกเก็บไว้ในบัฟเฟอร์ของโมดูลไอโอ จากนั้นจึงถูกส่งไปยังอุปกรณ์ต่อพ่วงด้วยความเร็วต่ำเท่ากับอุปกรณ์นั้นๆ ซึ่งในด้านที่รับข้อมูลนั้นก็จะทำงานที่ความเร็วสูงมาก เพื่อไม่ให้หน่วยความจำจะต้องมาถูกบังคับให้ทำงานช้าลงเพราะอุปกรณ์ความเร็วต่ำ โมดูลจึงต้องสามารถทำงานได้เร็วเท่ากับหน่วยความจำโปรเซสเซอร์ และจะต้องสามารถทำได้ช้ามากเท่ากับความเร็วของอุปกรณ์ต่อพ่วงแต่ละตัวที่มีใช้งาน ในทางกลับกัน ถ้าอุปกรณ์ไอโอทำงานที่ความเร็วสูงมากกว่าที่หน่วยความจำจะทำงานตามได้ทันแล้ว โมดูลไอโอจะต้องใช้เทคนิคบัฟเฟอร์กับอุปกรณ์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 อัตราการถ่ายเทข้อมูลของอุปกรณ์ไอโอโดยทั่วไป

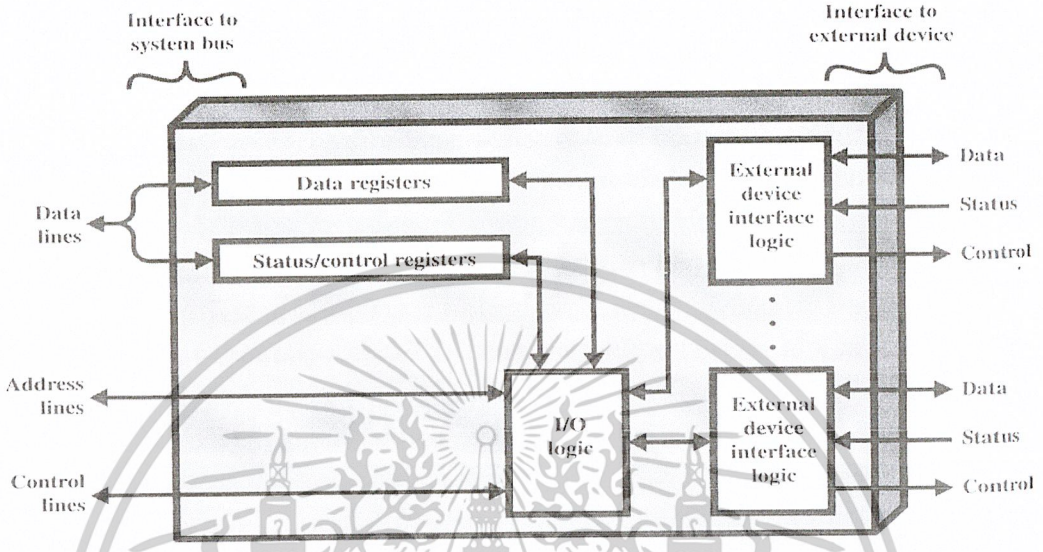
ท้ายที่สุด โมดูลไอโอมักจะมีความรับผิดชอบเกี่ยวกับการตรวจหาข้อผิดพลาดของข้อมูล (Error detection) และรับผิดชอบในการรายงานข้อผิดพลาดนั้นไปยังโปรเซสเซอร์

2.4.2 โครงสร้างโมดูลไอโอ

โมดูลไอโอมีความแตกต่างกันมากมาย ซึ่งขึ้นกับความซับซ้อนและจำนวนอุปกรณ์ไอโอภายนอกที่โมดูลนั้นสามารถควบคุมได้ รูปที่ 2.10 แสดงไดอะแกรมแบบบล็อกสำหรับโมดูลไอโอทั่วไป โมดูลจะเชื่อมต่อกับส่วนอื่นของเครื่องคอมพิวเตอร์ผ่านทางสายสัญญาณกลุ่มหนึ่ง เช่น ผ่านทางบัสหลักของเครื่องคอมพิวเตอร์ ข้อมูลที่ถูกส่งผ่านโมดูลจะถูกกระทำผ่านบัฟเฟอร์ที่เป็นรีจิสเตอร์ข้อมูล (Data register) อย่างน้อยหนึ่งตัว นอกจากนี้อาจมีรีจิสเตอร์แสดงสถานะ (status register) อีกหนึ่งหรือสองตัวใช้สำหรับการรายงานสถานการณ์ทำงานของอุปกรณ์ไอโอ รีจิสเตอร์แสดงสถานะยังทำหน้าที่เป็นรีจิสเตอร์สำหรับการควบคุม (control register) ในการรับคำสั่งควบคุมการทำงานมาจากโปรเซสเซอร์ การเชื่อมโยงภายในโมดูลไอโอจะติดต่อกับโปรเซสเซอร์ผ่านสายสัญญาณควบคุม โปรเซสเซอร์จะใช้สายควบคุมในการออกคำสั่งไปยังโมดูลไอโอ สายสัญญาณควบคุมบางเส้นอาจถูกใช้ด้วยโมดูลเอง เช่น การครอบครองบัสหรืออุปกรณ์ และสัญญาณแสดงสถานะ โมดูลไอโอยังจะต้องสมารถรับรู้และสร้างหมายเลขที่อยู่ที่เกี่ยวข้องกับอุปกรณ์ไอโอที่ควบคุมอยู่ โมดูลไอโอแต่ละหน่วยจะมีหมายเลขที่อยู่เฉพาะเป็นของตนเองอย่างน้อยหนึ่งหมายเลข และจะต้องมีหลายหมายเลขถ้าโมดูลนั้นควบคุมด้วยไอโอหลายตัว และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประการสุดท้าย โมดูลไอโอจะต้องมีตรรกะสำหรับการควบคุมการทำงานอุปกรณ์ไอโอที่ควบคุมอยู่ด้วย



รูปที่ 2.10 ไดอะแกรมแบบบล็อกสำหรับโมดูลไอโอ

โมดูลไอโอทำหน้าที่ในการอำนวยความสะดวก ให้โปรเซสเซอร์สามารถควบคุมการทำงานของอุปกรณ์ไอโอต่างๆได้อย่างง่ายดาย ในการนี้โมดูลไอโอจะต้องทำหน้าที่หลายอย่าง โมดูลไอโอจะต้องซ่อนรายละเอียดสำหรับการประสานจังหวะสัญญาณนาฬิกา การปรับรูปแบบข้อมูล และกลไกอิเล็กทรอนิกส์สำหรับอุปกรณ์ไอโอภายนอก ทั้งนี้เพื่อให้โปรเซสเซอร์สามารถทำงานได้ในรูปแบบคำสั่งของการอ่านหรือการบันทึกข้อมูล หรือการเปิดหรือปิดแฟ้มข้อมูลเท่านั้น ในรูปแบบที่ง่ายที่สุด โมดูลไอโออาจทำงานเฉพาะที่จำเป็นอย่างที่สุดเท่านั้น และเหลือการทำงานอื่นๆทั้งหมดไว้ให้กับโปรเซสเซอร์เป็นผู้สั่งการและควบคุมเอง (เช่น การสั่งม้วนเทปกลับตำแหน่งเดิมหลังเสร็จการใช้งาน)

โมดูลไอโอที่รับการทำงานส่วนใหญ่ไปทำเอง ซึ่งจะนำเสนอการเชื่อมต่อในระดับสูงให้กับโปรเซสเซอร์นั้นมักจะเรียกว่า ช่องสื่อสารไอโอ (I/O controller) หรือ ไอโอโปรเซสเซอร์ (I/O processor) ส่วนไอโอที่ทำงานน้อยและต้องอาศัยการควบคุมส่วนใหญ่จากโปรเซสเซอร์โดยตรงจะเรียกว่า หน่วยควบคุมไอโอ (device controller) ในเครื่องคอมพิวเตอร์ขนาดเล็กมักจะประกอบด้วยหน่วยควบคุมไอโอ ในขณะที่ไอโอโปรเซสเซอร์จะพบได้ในคอมพิวเตอร์เมนเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การควบคุมไอโอด้วยโปรแกรม

วิธีการควบคุมการทำงานของไอโอมีอยู่สามวิธี วิธีแรกคือการควบคุมไอโอด้วยโปรแกรม (Programmed I/O) ข้อมูลจะเกิดการแลกเปลี่ยนระหว่างโปรเซสเซอร์และโมดูลไอโอ โปรเซสเซอร์จะประมวลผลโปรแกรมที่เป็นการออกคำสั่งโดยตรงเกี่ยวกับการทำงานของไอโอรวมทั้งการตรวจสอบสถานะของอุปกรณ์ การสั่งการอ่านหรือบันทึกข้อมูล และการถ่ายเทข้อมูลเมื่อโปรเซสเซอร์ออกคำสั่งไปยังโมดูลไอโอ โปรเซสเซอร์จะต้องรอจนกว่าการตอบสนองต่อคำสั่งนั้นจะสิ้นสุดลง ถ้าโปรเซสเซอร์ทำงานได้เร็วกว่าโมดูลไอโอ การรอคอยนี้จะเป็นการเสียเวลาของโปรเซสเซอร์อย่างยิ่ง วิธีการที่สองเรียกว่าการควบคุมไอโอผ่านกระบวนการอินเทอร์รัพท์ (Interrupt-driven I/O) โปรเซสเซอร์จะออกคำสั่งไอโอ แล้วหันกลับไปประมวลผลคำสั่งอื่นต่อไป เมื่อโมดูลไอโอทำงานที่ได้รับมอบนั้นเสร็จเรียบร้อยแล้วก็จะส่งผ่านสัญญาณผ่านอินเทอร์รัพท์มาบอกให้ทราบ ทั้งสองวิธีนี้โปรเซสเซอร์จะต้องรับผิดชอบในการดึงข้อมูลออกมาจากหน่วยความจำหลักและส่งไปให้โมดูลไอโอ และจะต้องจัดการบันทึกข้อมูลที่ได้รับจากโมดูลเข้าไปในหน่วยความจำสำหรับการอ่านข้อมูล วิธีที่สามคือ ดีเอ็มเอ หรือการเข้าถึงหน่วยความจำหลักโดยตรง (Direct memory access; DMA) ด้วยวิธีการนี้ โมดูลไอโอและหน่วยความจำหลัก จะแลกเปลี่ยนข้อมูลกันโดยตรง โดยที่โปรเซสเซอร์จะมีส่วนร่วมในการทำงานน้อยมาก

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

ตารางที่ 2.6 เทคนิคไอโอแบบต่างๆ

ตาราง 2.6 แสดงให้เห็นความสัมพันธ์ระหว่าง วิธีการควบคุมการทำงานของโมดูลไอโอทั้งสามแบบ ลำดับต่อไปจะได้กล่าวถึงรายละเอียดการทำงานของวิธีการทั้งสามแบบนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 เรื่องทั่วไปเกี่ยวกับการควบคุมไอโอด้วยโปรแกรม

เมื่อโปรเซสเซอร์ประมวลผลโปรแกรม และพบคำสั่งที่เกี่ยวข้องกับไอโอ โปรเซสเซอร์จะประมวลผลคำสั่งนั้นด้วยการเรียกใช้ชุดคำสั่งที่เหมาะสมไปยังโมดูล ด้วยการควบคุมไอโอแบบใช้โปรแกรม โมดูลไอโอจะปฏิบัติตามคำสั่ง จากนั้นจะเปลี่ยนบิตที่เหมาะสมในรีจิสเตอร์บอกสถานะไอโอ (I/O status register) ดังแสดงในรูป 2.10 เมื่อปฏิบัติเสร็จเรียบร้อยโมดูลไอโอก็จะไม่ทำอะไรต่อไปอีก ไม่ว่าจะเป็นการบอกโปรเซสเซอร์ด้วยวิธีการใดๆ รวมทั้งการใช้อินเทอร์รัพท์ ดังนั้นจึงเป็นความรับผิดชอบของโปรเซสเซอร์ที่จะต้องคอยตรวจสอบดูสถานะของโมดูลไอโอ จนพบว่างานที่ได้สั่งในทำนั้นเสร็จเรียบร้อยแล้ว

เพื่อให้เข้าใจการทำงานของวิธีการควบคุมไอโอโดยการใช้โปรแกรม ในลำดับต่อไปจะกล่าวถึง การออกคำสั่งไอโอ (Commands) ที่โปรเซสเซอร์สั่งการไปยังโมดูลไอโอ จากนั้นจะกล่าวถึงคำสั่ง (instruction) ที่ถูกประมวลผลโดยโปรเซสเซอร์

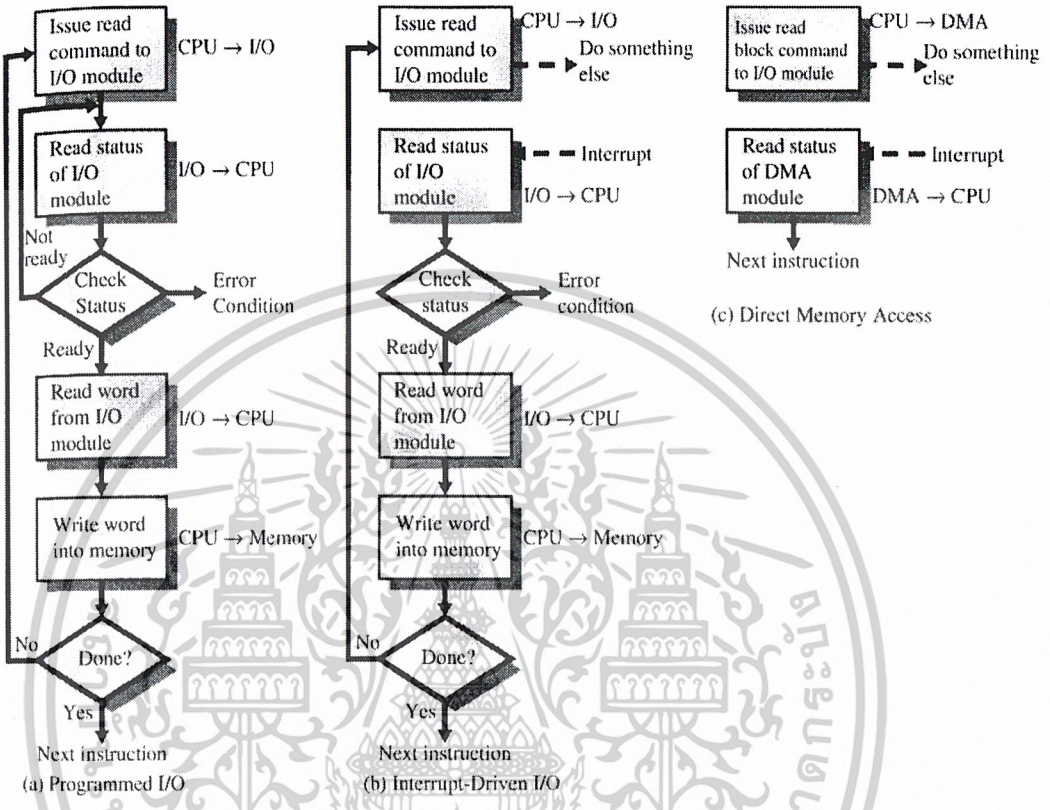
2.4.5 การออกคำสั่งไอโอ

ในการประมวลผลคำสั่งที่เกี่ยวข้องกับไอโอ โปรเซสเซอร์จะต้องกำหนดหมายเลขที่อยู่ (Address) ของโมดูลไอโอและอุปกรณ์ภายนอก และคำสั่งที่ต้องการให้ปฏิบัติ ซึ่งแบ่งออกได้เป็นสี่ประเภทคือ

- การควบคุม (Control): ใช้ในการกระตุ้นให้อุปกรณ์ไอโอพร้อมที่จะปฏิบัติงาน และบอกให้ทราบว่าจะต้องทำงานอะไร ตัวอย่างเช่น หน่วยเทปแม่เหล็กอาจได้รับคำสั่งให้ม้วนเทปกลับหรือหมุนเทปไปข้างหน้าหนึ่งระเบียน คำสั่งเหล่านี้จะได้รับการปรับแต่งให้มีความเหมาะสมกับอุปกรณ์แต่ละชนิด
- การทดสอบ (Test): ใช้สำหรับการทดสอบสถานการณ์ทำงานหลายอย่างของโมดูลไอโอและอุปกรณ์ โปรเซสเซอร์จะต้องการทราบว่าอุปกรณ์ที่ต้องการใช้งานนั้น มีความพร้อมใช้งานอยู่หรือไม่ หรือต้องการทราบว่าคำสั่งที่ให้ปฏิบัติก่อนหน้านี้ได้เสร็จเรียบร้อยหรือยังหรือว่ามีข้อผิดพลาดใดๆเกิดขึ้นหรือไม่
- การอ่านข้อมูล (Read): จะทำให้โมดูลไอโออ่านข้อมูลมาจากอุปกรณ์ที่ต่อพ่วงและใส่เข้าไว้ในบัฟเฟอร์ (แสดงให้เห็นเป็น data register ในรูป 2.10) จากนั้นโปรเซสเซอร์จะสามารถนำข้อมูลไปใช้ โดยออกคำสั่งให้โมดูลไอโอใส่ข้อมูลเข้าไปในบัสนำส่งข้อมูล (data bus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

– การบันทึกข้อมูล (Write): จะทำให้โมดูลไอโอนำข้อมูล (ไบต์ หรือ word) จากบัสนำส่งข้อมูล ไปส่งต่อให้กับอุปกรณ์ต่อพ่วงที่ต้องการ



รูปที่ 2.11 เทคนิค 3 แบบสำหรับการนำเข้าข้อมูล 1 บล็อก

รูป 2.11a แสดงตัวอย่างของการควบคุมไอโอโดยใช้โปรแกรมในการอ่านข้อมูลจากบล็อกข้อมูลในอุปกรณ์ต่อพ่วง (เช่น อ่านข้อมูลจากระเบียนในเทปแม่เหล็ก) เข้ามาในหน่วยความจำหลัก ข้อมูลจะถูกอ่านเข้ามาทีละ word (16 บิต) โปรเซสเซอร์จะต้องทำการวงรอบการตรวจสอบข้อมูลแต่ละ word ที่ถูกอ่านเข้ามา จนกว่าจะทราบว่าข้อมูลนั้นเข้ามาอยู่ในรีจิสเตอร์ข้อมูลของโมดูลไอโอเรียบร้อยแล้ว แผนผังแสดงการทำงานนี้ได้ชี้ให้เห็นจุดอ่อนของวิธีการควบคุมไอโอด้วยโปรแกรม ซึ่งเป็นวิธีที่เสียเวลามาก และทำให้โปรเซสเซอร์ต้องทำงาน (ตรวจสอบ) อยู่ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 คำสั่งไอโอที่ถูกประมวลผล

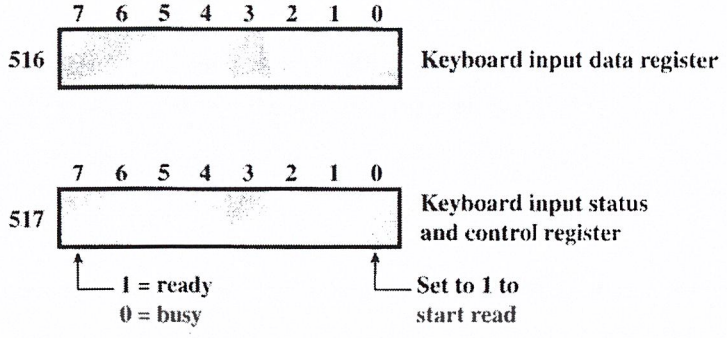
การควบคุมไอโอแบบใช้โปรแกรมนั้น ความสัมพันธ์ระหว่างคำสั่งที่เกี่ยวข้องกับไอโอที โปรเซสเซอร์ดึงขึ้นมาจากหน่วยความจำ (จากโปรแกรมของผู้ใช้) เพื่อนำมาประมวลผล กับคำสั่งไอโอที่โปรเซสเซอร์สั่งให้โมดูลไอโอทำการประมวลผลนั้นมีความเกี่ยวข้องกันอย่างใกล้ชิด นั่นคือ คำสั่งจากหน่วยความจำสามารถแปลงเป็นคำสั่งที่ส่งไปให้โมดูลไอโอได้ง่าย ปกติจะแปลงแบบคำสั่งต่อคำสั่ง รูปแบบของคำสั่งที่ถูกแปลงนั้นมักจะขึ้นอยู่กับอุปกรณ์ภายนอกที่ถูกอ้างอิงถึง

โดยทั่วไป อาจมีอุปกรณ์ไอโอจำนวนหนึ่งเชื่อมต่อเข้ากับโมดูลไอโอ อุปกรณ์แต่ละชนิดจะได้รับการกำหนดหมายเลขที่อยู่เฉพาะที่ไม่ซ้ำกัน เมื่อโปรเซสเซอร์ส่งคำสั่งไอโอมายังโมดูลไอโอ คำสั่งนั้นจะต้องระบุหมายเลขที่อยู่ของอุปกรณ์ไอโอที่ต้องการไว้ด้วยเสมอ ดังนั้นโมดูลไอโอจะต้องตรวจสอบสัญญาณที่มาตามสายว่าเป็นหมายเลขที่อยู่ของตัวเองหรือไม่

เมื่อโปรเซสเซอร์ หน่วยความจำหลัก และไอโอ ใช้บัสร่วมกัน จะมีวิธีการอ้างอิงหมายเลขที่อยู่ (Address) สองวิธีแรกคือ การทำแผนที่หน่วยความจำหลัก และแบบแยกออกจากกัน วิธีการทำแผนที่หน่วยความจำหลัก (memory-mapped I/O) จะแบ่งหน่วยความจำส่วนหนึ่ง (และมีเพียงส่วนเดียว) ไว้เป็นการเฉพาะสำหรับการเก็บข้อมูลต่างๆเกี่ยวกับอุปกรณ์ไอโอทั้งหมด รวมทั้งข้อมูลที่มีการส่งผ่านระหว่างกันด้วย โปรเซสเซอร์จะถือว่ารีจิสเตอร์ข้อมูลและรีจิสเตอร์แสดงสถานะของโมดูลไอโอเป็นหน่วยความจำส่วนหนึ่ง จึงใช้คำสั่งในการอ้างอิงหรือเข้าถึงหน่วยความจำและอุปกรณ์ไอโอเป็นคำสั่งอย่างเดียวกัน ดังนั้นถ้าสายสัญญาณไอโอมีจำนวน 10 เส้น จะมีปริมาณหน่วยความจำที่ต้องใช้เป็น 1024 (2_{10}=1024) ตำแหน่ง

การทำแผนที่หน่วยความจำหลัก มีความต้องการใช้สายสัญญาณในบัสสำหรับอ่านข้อมูลและสำหรับบันทึกข้อมูลอย่างละหนึ่งเส้นเท่านั้น วิธีการจัดไอโออีกแบบหนึ่งเรียกว่า การจัดการไอโอแบบแยกออกจากกัน (Isolated I/O) โดยบัสจะมีสายสัญญาณอ่านและบันทึกข้อมูลในหน่วยความจำ และสายคำสั่งนำเข้าและส่งออกข้อมูล สายคำสั่งฯ จะถูกนำมาใช้ในการกำหนดว่าหมายเลขที่อยู่นั้นเป็นการอ้างอิงถึงหน่วยความจำหรืออุปกรณ์ไอโอ ด้วยวิธีการนี้ทำให้โปรเซสเซอร์มีความสามารถในการอ้างอิงถึงตำแหน่งข้อมูล 1024 ตำแหน่งในหน่วยความจำ และอีก 1024 ตำแหน่ง สำหรับอุปกรณ์ไอโอที่มีสัญญาณจำนวน 10 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load AC	"1"	Load accumulator
	Store AC	517	Initiate keyboard read
202	Load AC	517	Get status byte
	Branch if Sign = 0	202	Loop until ready
	Load AC	516	Load data byte

(a) Memory-mapped I/O

ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load I/O	5	Initiate keyboard read
201	Test I/O	5	Check for completion
	Branch Not Ready	201	Loop until complete
	In	5	Load data byte

(b) Isolated I/O

รูปที่ 2.12 ข้อมูลในหน่วยความจำและข้อมูลไอโอ

รูป 2.12 แสดงความแตกต่างระหว่างเทคนิคการจัดการตำแหน่งที่อยู่ไอโอทั้งสองแบบ รูป 2.12a แสดงให้เห็นว่าโปรแกรมเมอร์จะมองเห็นส่วนติดต่ออุปกรณ์นำเข้าข้อมูล เช่น คีย์บอร์ดในลักษณะใด เมื่อใช้เทคนิคการทำแผนที่หน่วยความจำ สมมุติว่ามีสายสัญญาณตำแหน่งที่อยู่จำนวน 10 เส้น และมีหน่วยความจำขนาด 512 บิต (หมายเลข 0 ถึง 511) และหมายเลขที่อยู่ไอโออีก 512 ตำแหน่ง (หมายเลข 512 ถึง 1023) คีย์บอร์ดจะมีหมายเลขที่อยู่ 2 หมายเลขเป็นของตนเอง (516 และ 517) ที่ตำแหน่ง 516 หมายถึงรีจิสเตอร์ข้อมูล (data register) และ517 หมายถึงรีจิสเตอร์แสดงสถานะ (status register) ซึ่งทำหน้าที่เป็นรีจิสเตอร์สำหรับการควบคุม (control register) สำหรับการรับคำสั่งมาจากโปรเซสเซอร์ โปรแกรมที่แสดงในรูป 2.12 แสดงการอ่านข้อมูลขนาด 1 ไบต์จากคีย์บอร์ดเข้ามาไว้ในรีจิสเตอร์แอ็กคิวมูเลเตอร์ (accumulator register) ที่อยู่ภายในโปรเซสเซอร์ จะสังเกตว่าโปรเซสเซอร์จะทำงานซ้ำจนกว่าจะได้ข้อมูลที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.12 b แสดงเทคนิคการจัดการที่อยู่ไอโอแบบแยกออกจากกัน หมายเลขพอร์ต (port) ของไอโอที่ต้องการจะถูกอ้างอิงถึงได้ โดยการใช้คำสั่งไอโอเท่านั้น ซึ่งจะไปกระตุ้นการทำงานของสายสัญญาณไอโอที่อยู่ในบัส

สำหรับโปรเซสเซอร์ส่วนมากจะมีคำสั่งที่อ้างอิงหน่วยความจำอยู่เป็นจำนวนมาก ถ้านำเทคนิคการจัดการที่อยู่แบบแยกจากกันมาใช้จะทำให้มีคำสั่งไอโอเพียงจำนวนน้อย ดังนั้น เทคนิคการจัดการที่อยู่ไอโอแบบการทำแผนที่หน่วยความจำหลัก (Memory-mapped I/O) จึงได้เปรียบตรงที่จะมีคำสั่งใช้งานเกี่ยวกับไอโอได้มากมาย ทำให้มีทางเลือกในการเขียนโปรแกรมที่มีประสิทธิภาพได้หลายทาง แต่ในเวลาเดียวกัน วิธีการนี้ทำให้มีปริมาณหน่วยความจำหลักลดลง ในความเป็นจริงมีการนำเทคนิคทั้งสองอย่างมาใช้อย่างกว้างขวาง

2.4.7 ไอโอที่ใช้กลไกอินเทอร์รัพท์

หนทางเลือกในการแก้ปัญหาที่โปรเซสเซอร์จะต้องรอเป็นระยะเวลานานเมื่อควบคุมไอโอด้วยโปรแกรม ให้โปรเซสเซอร์ส่งคำสั่งไอโอออกไปยังโมดูลไอโอ แล้วไปทำงานอื่นที่มีประโยชน์มากกว่าการรอคอย โมดูลไอโอจะขัดจังหวะการทำงานของโปรเซสเซอร์เมื่อมีความพร้อมที่จะแลกเปลี่ยนข้อมูล โปรเซสเซอร์จะเป็นตัวจัดการการถ่ายเทข้อมูลเหมือนเดิม ซึ่งเมื่อทำงานเสร็จแล้วก็จะกลับไปทำงานที่เดิมก่อนที่จะถูกขัดจังหวะ

การทำงานโดยละเอียดทางด้านโมดูลไอโอเป็นดังนี้ สำหรับการอ่านข้อมูล เมื่อโมดูลไอโอได้รับสัญญาณ READ จากโปรเซสเซอร์ ก็จัดการติดต่อขอข้อมูลจากอุปกรณ์ไอโอที่เกี่ยวข้อง เมื่อข้อมูลที่ต้องการถูกส่งเข้ามาอยู่ในรีจิสเตอร์ข้อมูลเรียบร้อยแล้ว โมดูลไอโอจะส่งสัญญาณอินเทอร์รัพท์ หรือการขัดจังหวะการทำงานไปยังโปรเซสเซอร์ผ่านสายสัญญาณสำหรับการควบคุม จากนั้นก็รอจนกว่าจะได้รับสัญญาณร้องขอข้อมูลจากโปรเซสเซอร์ โมดูลไอโอจะส่งข้อมูลเข้าไปในบัสสัญญาณข้อมูล และพร้อมที่จะรับคำสั่งต่อไป

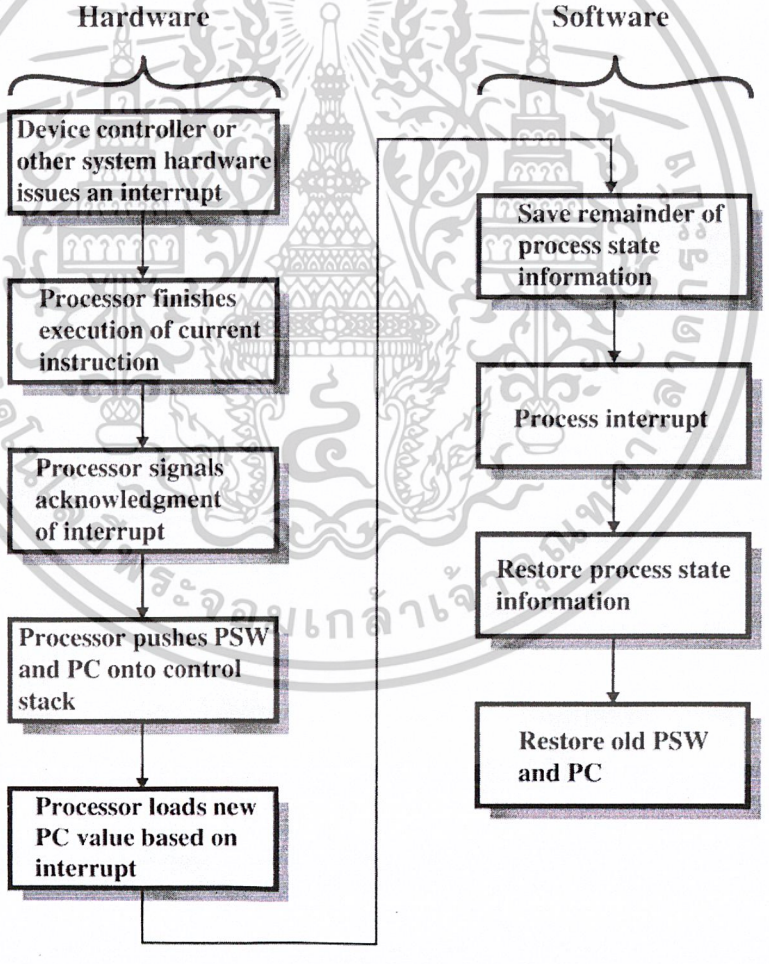
ส่วนการทำงานทางฝั่งโปรเซสเซอร์สำหรับการอ่านข้อมูลจะเป็นดังนี้ โปรเซสเซอร์เริ่มต้นด้วยการส่งคำสั่ง READ ไปยังโมดูลไอโอ เสร็จแล้วก็หันไปทำงานอื่น เช่น ในกรณีมราโปรเซสเซอร์กำลังทำการประมวลผลโปรแกรมหลายโปรแกรมพร้อมกัน ในทุกวงรอบการประมวลผลโปรเซสเซอร์จะตรวจสอบว่ามีอินเทอร์รัพท์ถูกส่งเข้ามาหรือไม่ ถ้าหากว่าไม่มีก็จะประมวลผลในวงรอบต่อไป เมื่อโมดูลไอโอส่งสัญญาณอินเทอร์รัพท์เข้ามา โปรเซสเซอร์จะทำการบันทึกข้อมูลและสถานการณ์ทำงานในขณะนั้นไว้ แล้วไปเรียกโปรแกรมสำหรับอินเทอร์รัพท์นั้นขึ้นมาประมวลผลแทน (เรียกว่าทำ context switch) ในกรณี โปรแกรมฯจะสั่งให้โปรเซสเซอร์ทำหารอ่านข้อมูลขนาดหนึ่ง word มาจากรีจิสเตอร์ข้อมูลของโมดูลไอโอ แล้วเก็บไว้ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลัก เมื่อทำงานแล้วก็จะอ่านข้อมูลและสถานะทั้งหมดของโปรแกรมที่ทำการประมวลผลอยู่ก่อนหน้านี้ (ทำ context switchอีกครั้งหนึ่ง) เพื่อทำการประมวลผลโปรแกรมนั้นต่อไป

รูป 2.11b แสดงการใช้อินเทอร์รัพท์สำหรับไอโอในการอ่านข้อมูลบล็อกหนึ่ง เมื่อเปรียบเทียบรูปนี้กับรูป 2.11a จะพบว่าการทำงานของไอโอที่ใช้กลไกอินเทอร์รัพท์นั้นมีประสิทธิภาพสูงกว่าการทำงานของไอโอที่ใช้โปรแกรมเป็นตัวควบคุมการทำงาน เนื่องจากโปรเซสเซอร์ไม่ต้องเสียเวลาในการรอคอย อย่างไรก็ตาม ไอโอที่ใช้กลไกอินเทอร์รัพท์ก็จะต้องใช้เวลาการทำงานของโปรเซสเซอร์เป็นอย่างมาก เนื่องจากข้อมูลที่ถูกอ่านจากหน่วยความจำไปเก็บไว้ในโมดูลไอโอหรือกลับกันนั้นจะต้องส่งผ่านตัวโปรเซสเซอร์เอง

2.4.8 กระบวนการอินเทอร์รัพท์



รูปที่ 2.13 กระบวนการอินเทอร์รัพท์อย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเกิดขึ้นของอินเทอร์รัพท์เป็นการกระตุ้นให้เกิดเหตุการณ์หลายอย่างติดตามมา ทั้งที่ตัวโปรเซสเซอร์และตัวโปรแกรม รูป 2.13 แสดงลำดับเหตุการณ์ที่เกิดขึ้นเป็นปกติสำหรับอินเทอร์รัพท์ทั่วไป เมื่อโมดูลไอโอจัดงานไอโอที่ได้รับคำสั่งให้ทำเสร็จสิ้นแล้ว จะเกิดเหตุการณ์ต่อไปนี้เกิดขึ้นกับฮาร์ดแวร์คอมพิวเตอร์

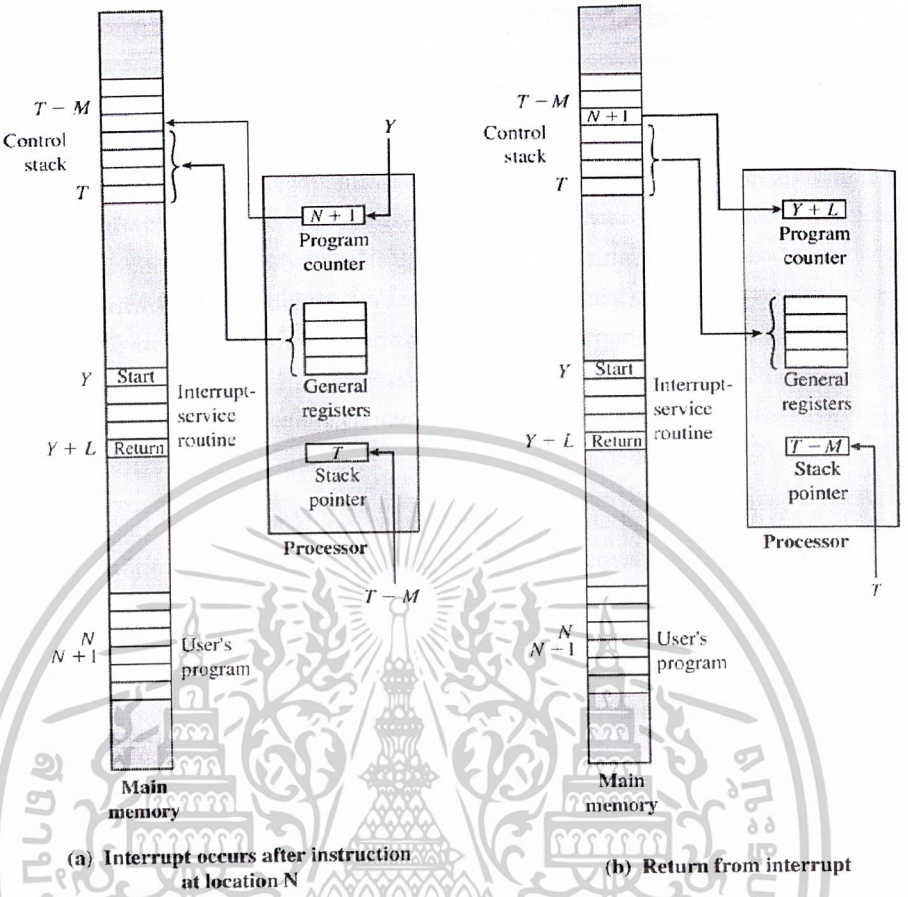
1. อุปกรณ์ไอโอจะส่งสัญญาณอินเทอร์รัพท์ไปยังโปรเซสเซอร์
2. โปรเซสเซอร์ทำการประมวลผลคำสั่งในโปรแกรมที่กำลังทำงานอยู่จนเสร็จเรียบร้อย (เฉพาะคำสั่งที่กำลังทำอยู่นั้น) ก่อนที่จะตอบสนองต่ออินเทอร์รัพท์
3. โปรเซสเซอร์จะทำการทดสอบอินเทอร์รัพท์ ตรวจสอบว่ามีเพียงอินเทอร์รัพท์เดียว และส่งสัญญาณตอบรับไปยังอุปกรณ์ที่ส่งสัญญาณอินเทอร์รัพท์นั้น การตอบรับจะทำให้อุปกรณ์นั้นหยุดส่งสัญญาณอินเทอร์รัพท์
4. โปรเซสเซอร์จะเตรียมการประมวลผลโปรแกรมสำหรับการอินเทอร์รัพท์นั้น (เรียกว่า interrupt routine) ซึ่งเริ่มต้นด้วยการบันทึกข้อมูลที่เกี่ยวข้องกับการประมวลผลโปรแกรมที่กำลังทำงานอยู่นั้น (ก่อนเกิดอินเทอร์รัพท์) ประกอบด้วย (1) สถานการณ์ทำงานของโปรเซสเซอร์ซึ่งเก็บอยู่ในรีจิสเตอร์ตัวหนึ่ง เรียกว่า Program status word (PSW) และ (2) ตำแหน่งของคำสั่งต่อไปที่จะถูกประมวลผล ซึ่งเก็บอยู่ในรีจิสเตอร์เรียกว่า Program counter (PC) ข้อมูลเหล่านี้อาจถูกเก็บอยู่ในสแต็กควบคุมการทำงานของระบบ (system control stack)
5. โปรเซสเซอร์จะดึงคำสั่งแรกของโปรแกรมสำหรับอินเทอร์รัพท์ที่เกิดขึ้นนั้น ขึ้นมาทำงานเพื่อตอบสนองต่ออินเทอร์รัพท์ โปรแกรมดังกล่าวอาจเป็นเพียงโปรแกรมเดียวสำหรับอินเทอร์รัพท์แต่ละอย่าง หรือโปรแกรมเดียวสำหรับอุปกรณ์แต่ละอย่าง ซึ่งขึ้นอยู่กับสถาปัตยกรรมของคอมพิวเตอร์หรือระบบปฏิบัติการที่เลือกใช้ ในกรณีที่มีโปรแกรมสำหรับอินเทอร์รัพท์มากกว่าหนึ่งโปรแกรม โปรเซสเซอร์จะต้องทำการตัดสินใจเลือกโปรแกรมหนึ่งขึ้นมาทำงาน ซึ่งข้อมูลสำหรับการตัดสินใจนี้อาจถูกส่งมาพร้อมกับสัญญาณอินเทอร์รัพท์แล้ว หรือโปรเซสเซอร์อาจจะต้องส่งสัญญาณไปถามอุปกรณ์ที่เป็นผู้ส่งสัญญาณอินเทอร์รัพท์เข้ามาจึงจะสามารถเลือกโปรแกรมที่ถูกต้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตำแหน่งคำสั่งของโปรแกรมสำหรับอินเทอร์รัพท์ ได้ถูกใส่เข้าไปในโปรแกรมเคาน์เตอร์เรียบร้อยแล้ว โปรเซสเซอร์ก็จะทำการดึงคำสั่งต่อไปขึ้นมาทำงาน เนื่องจากว่าคำสั่งที่ถูกดึงเข้าไปในโปรเซสเซอร์นั้นจำเป็นตัวกำหนดการทำงานของโปรแกรม ดังนั้นจึงเท่ากับว่า ในขณะนี้โปรเซสเซอร์ได้ทำการโอนการควบคุมไปให้กับโปรแกรมสำหรับอินเทอร์รัพท์เรียบร้อยแล้ว การประมวลผลโปรแกรมนี้จะทำให้เกิดเหตุการณ์ต่อไปนี้

6. ณ เวลานี้ โปรแกรมเคาน์เตอร์และ PSW ที่เกี่ยวข้องกับโปรแกรมที่ถูกขัดจังหวะการทำงานได้ถูกบันทึกไว้ในสแต็กของระบบเรียบร้อยแล้ว อย่างไรก็ตาม ยังคงมีข้อมูลอื่นที่ถือได้ว่าเป็นส่วนหนึ่งของสถานะของโปรแกรมที่กำลังถูกประมวลผลอยู่ นั่นคือข้อมูลที่เก็บอยู่ในรีจิสเตอร์ภายในโปรเซสเซอร์จะต้องถูกบันทึกไว้เช่นเดียวกัน เนื่องจากรีจิสเตอร์เหล่านี้อาจถูกใช้โดยโปรแกรมอินเทอร์รัพท์ ดังนั้นข้อมูลในรีจิสเตอร์เหล่านี้ (อาจหมายถึงข้อมูลแสดงสถานะอื่นๆ) จะต้องถูกบันทึกไว้ด้วย โดยทั่วไปโปรแกรมสำหรับอินเทอร์รัพท์จะเริ่มต้นจากการทำงานด้วยการบันทึกข้อมูลในรีจิสเตอร์ลงในสแต็ก รูป 2.14 a แสดงตัวอย่างง่ายๆ อันหนึ่ง ในกรณีนี้ โปรแกรมของผู้ใช้ถูกขัดจังหวะการทำงานภายหลังจากคำสั่งตำแหน่งที่ N ข้อมูลที่เก็บอยู่ในรีจิสเตอร์ทั้งหมดและตำแหน่งของคำสั่งที่จะถูกประมวลผลในลำดับต่อไป (N+1) จะถูกเก็บไว้ในสแต็ก ตัวชี้ตำแหน่งของสแต็กจะถูกปรับค่าใหม่ และโปรแกรมเคาน์เตอร์จะถูกบันทึกตำแหน่งของคำสั่งแรกของโปรแกรมสำหรับอินเทอร์รัพท์
7. โปรแกรมนี้ก็จะเริ่มทำงานต่อไป ด้วยการตรวจสอบสถานะที่เกี่ยวข้องกับการทำงานไอโอที่ต้องการ หรือเหตุการณ์อื่นที่ทำให้เกิดอินเทอร์รัพท์ การทำงานนี้อาจรวมถึงการส่งคำสั่งเพิ่มเติมหรือการส่งการตอบรับไปยังอุปกรณ์ไอโอ
8. เมื่อโปรแกรมอินเทอร์รัพท์ทำงานเสร็จแล้ว ก็จะปิดท้ายด้วยการคืนค่าให้แก่รีจิสเตอร์ในโปรเซสเซอร์ที่นำมาจากสแต็กของระบบ ดังที่แสดงในรูป 2.14 b
9. การทำงานส่วนสุดท้ายคือการคืนค่าให้แก่ PSW และโปรแกรมเคาน์เตอร์ที่นำมาจากสแต็กของระบบ ทำให้คำสั่งต่อไปที่จะถูกประมวลผลนั้นถูกอ่านมาจากคำสั่งในตำแหน่งต่อไปของโปรแกรมที่ถูกขัดจังหวะก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การเปลี่ยนแปลงของหน่วยความจำและรีจิสเตอร์สำหรับอินเทอร์รัพท์

โปรดสังเกตว่า การบันทึกข้อมูลแสดงสถานการณ์ทำงานทั้งหมดของโปรแกรมที่ถูกขัดจังหวะการทำงานนั้นเป็นเรื่องที่สำคัญมาก เพราะว่าค่าทั้งหมดจะต้องถูกคืนสภาพในภายหลัง ทั้งนี้เนื่องจากโปรแกรมสำหรับอินเทอร์รัพท์นั้นไม่ใช่การเรียกใช้ฟังก์ชันตามปกติ ยิ่งกว่านี้อินเทอร์รัพท์ยังสามารถเกิดขึ้นได้ในทุกเวลา และถือว่าเป็นเหตุการณ์ที่ไม่สามารถคาดได้ โดยข้อเท็จจริงแล้ว โปรแกรมที่ถูกขัดจังหวะและโปรแกรมสำหรับอินเทอร์รัพท์นั้นอาจไม่มีความเกี่ยวข้องใดๆระหว่างกันเลย และอาจเป็นโปรแกรมที่เกิดขึ้นจากผู้ใช้คนละคนกันก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการดำเนินงาน

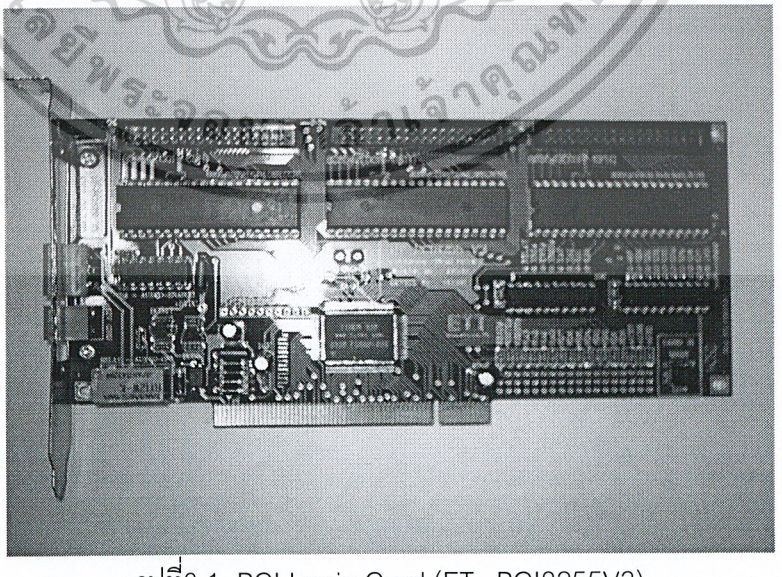
3.1 การใช้ PCI Logic Card สำหรับการวิเคราะห์สัญญาณ

โดยมีลำดับการปฏิบัติงานดังนี้

3.1.1 เริ่มศึกษาการใช้โปรแกรม Labview ในการเขียนโปรแกรมเพื่อการเชื่อมต่อ และรับส่งข้อมูลผ่านทาง I/O ของ PCI Logic Card

อุปกรณ์ที่ใช้มีดังนี้

- เครื่องคอมพิวเตอร์
  - CPU : Intel(R) Celeron(R) 2.40GHz.
  - Ram : 256MB
  - ระบบปฏิบัติการ :Microsofts Windows XP
- รายละเอียดของ ETT-8255 V.3 PCI Card
  - มี Input/Output 72 channel
  - ใช้ชิพTIGER 3200 ซึ่งทำงานโดยไม่จำเป็นต้องมี crystal
- I/O Board ของบริษัท ETT ซึ่งภายในประกอบไปด้วย ตัวแสดงผลทางเอาท์พุท(LED) 24 ตัว ,ตัวกำหนดค่าทางอินพุท (Dipswitch) 24 ตัว
- โปรแกรม Labview7.0



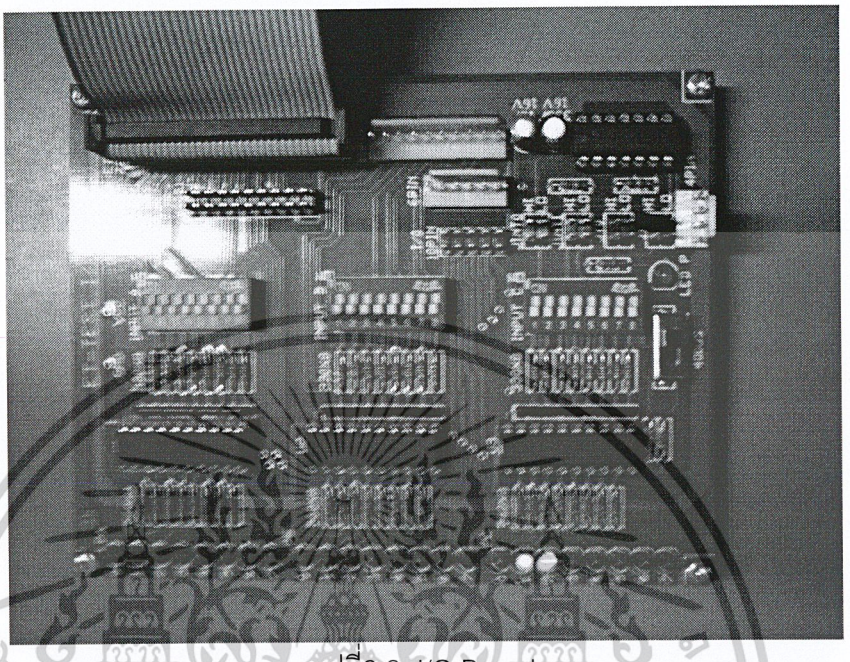
รูปที่3.1 PCI Logic Card (ET –PCI8255V3)

3.1.1.1 เขียนโปรแกรมสั่งให้ไฟวิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 เขียนโปรแกรมรับค่าจาก Dipswitch มาแสดงทางหน้าจอ

3.1.1.3 เขียนโปรแกรมสั่งให้แสดงกราฟทางหน้าจอ



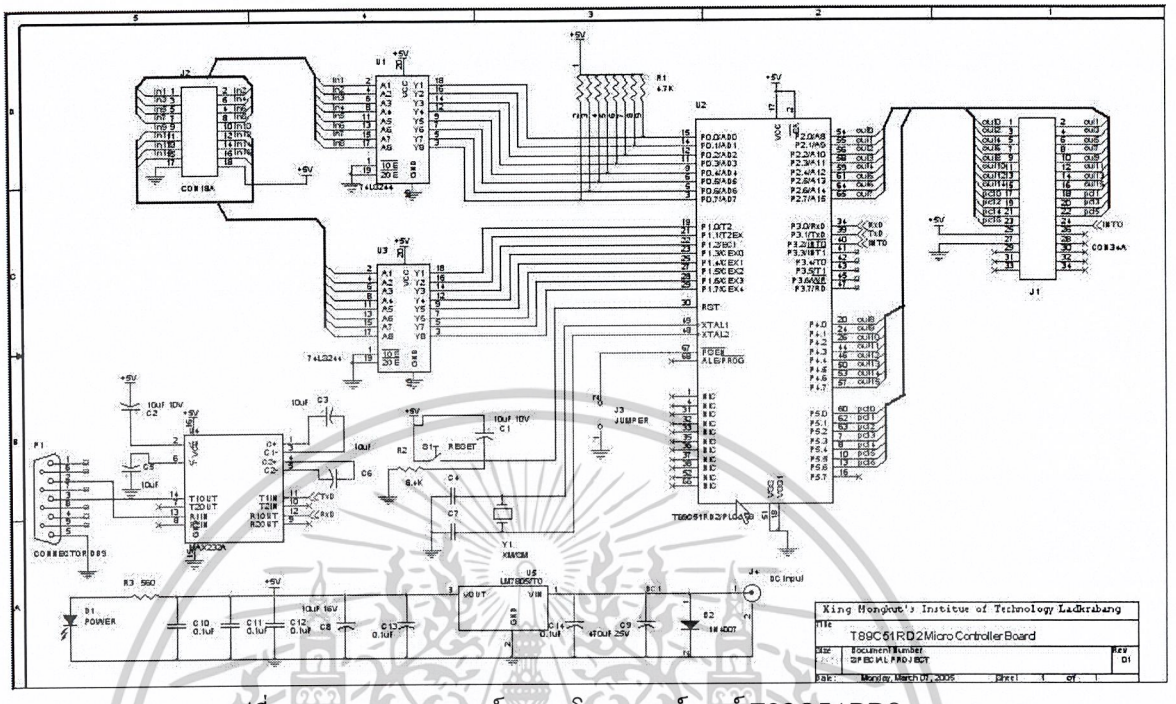
รูปที่3.2 I/O Board

3.1.2 ทำการการใช้โปรแกรม Labview ในการเขียนโปรแกรมเพื่อการรับค่าสัญญาณจาก Function Generator โดยผ่านบอร์ดไมโครคอนโทรลเลอร์เบอร์ T89C51RD2 ก่อนส่งค่า แล้วนำมาประมวลสัญญาณบนหน้าจอคอมพิวเตอร์

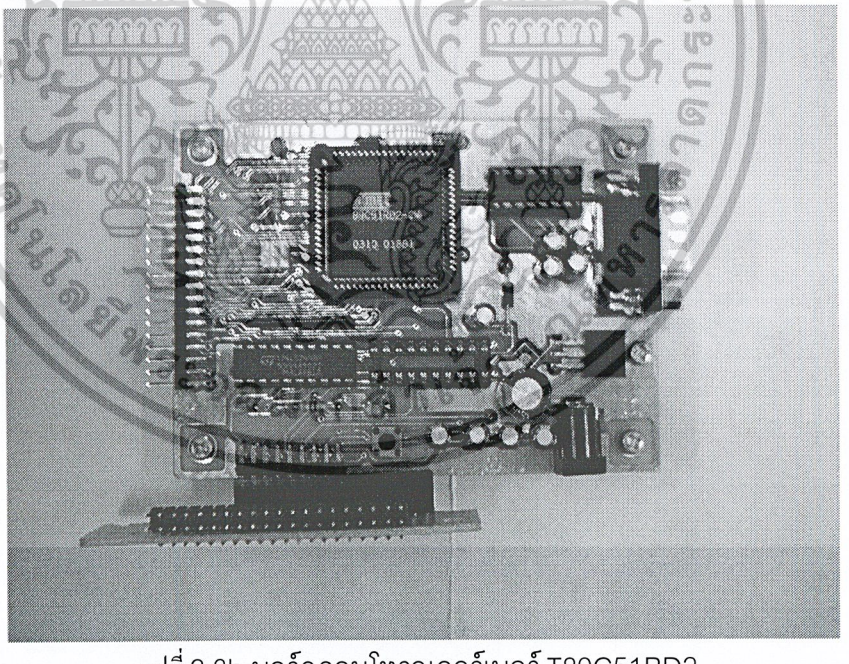
อุปกรณ์ที่ใช้มีดังนี้

- เครื่องคอมพิวเตอร์ ระบบปฏิบัติการ Intel(R) Celeron(R) CPU 2.40GHz. 256MB of Ram OS.MS.Windows XP
- PCI Logic Card (ET–PCI8255V3)
- Function Generator
- บอร์ดไมโครคอนโทรเลอร์ T89C51RD2
  - ใช้ชิพ T89C51RD2 หน่วยความจำภายใน 1024 byte เป็น CPU
  - สามารถใช้วัดสัญญาณด้าน Input ได้ถึง 16 channel
  - สามารถใช้ X2 mode เพื่อเพิ่มความเร็วจากการทำงานอิสระจาก CPU และอุปกรณ์เชื่อมต่อ Control ด้วย โปรแกรมภาษาassembly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3a แสดงวงจรบอร์ดคอนโทรลเลอร์เบอร์ T89C51RD2

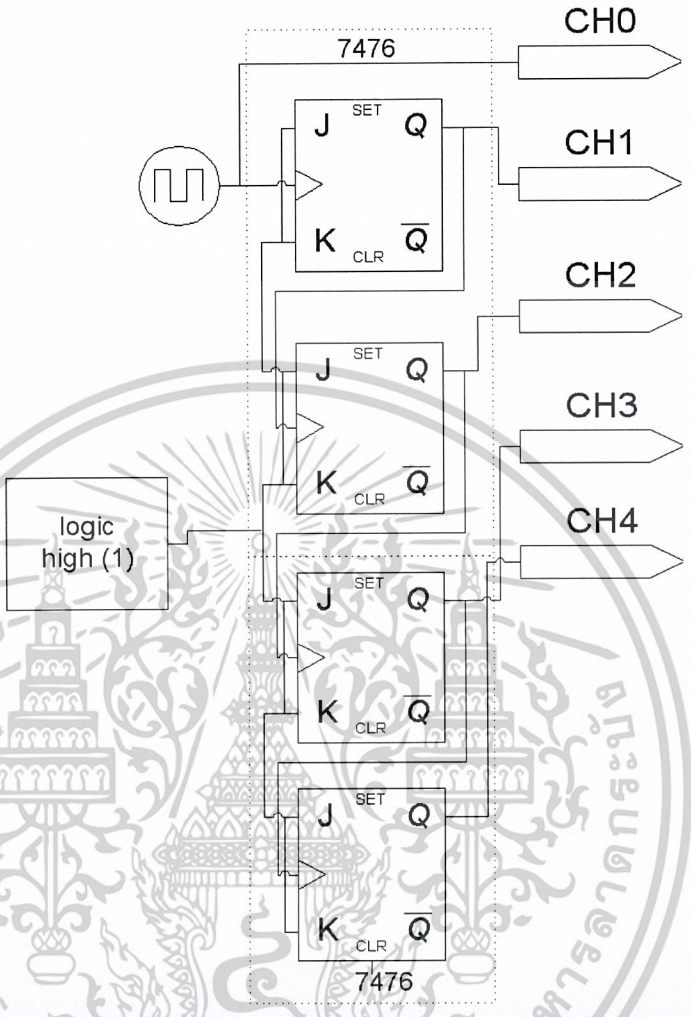


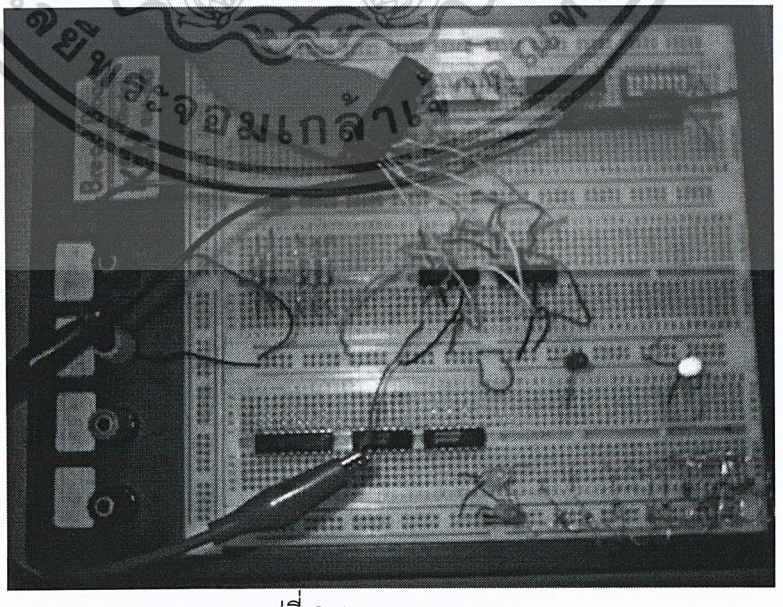
รูปที่ 3.3b บอร์ดคอนโทรลเลอร์เบอร์ T89C51RD2

- โปรแกรม Labview7.0
- โปรแกรม Assembly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรหาร

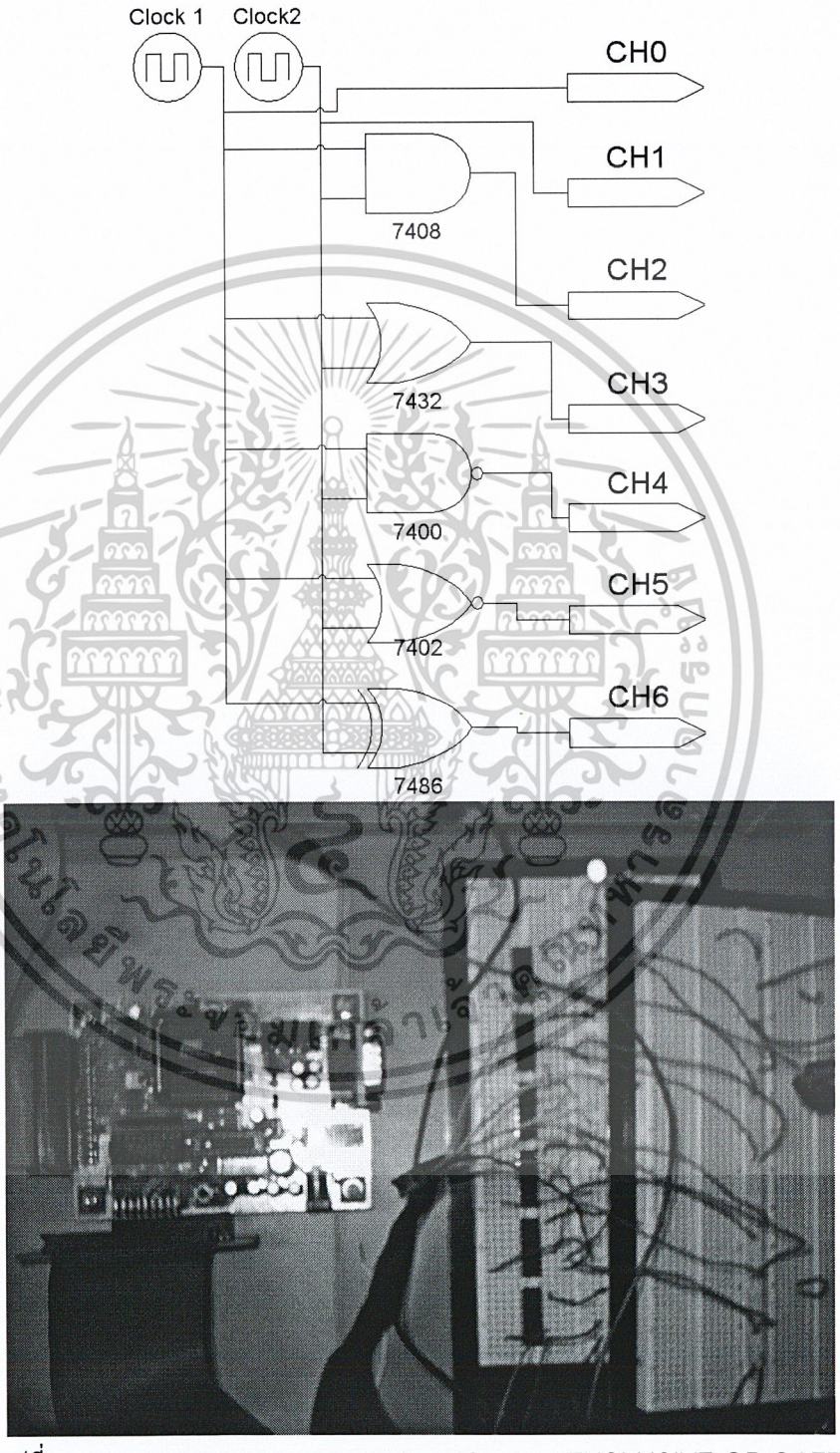




รูปที่ 3.4 แสดงวงจรหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรทางดิจิตอลโดยต่อกับเกทต่างๆ เช่น AND, OR, NAND, NOR และ EXCLUSIVE-OR GATE



รูปที่ 3.5 แสดงวงจร AND, OR, NAND, NOR และ EXCLUSIVE-OR GATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

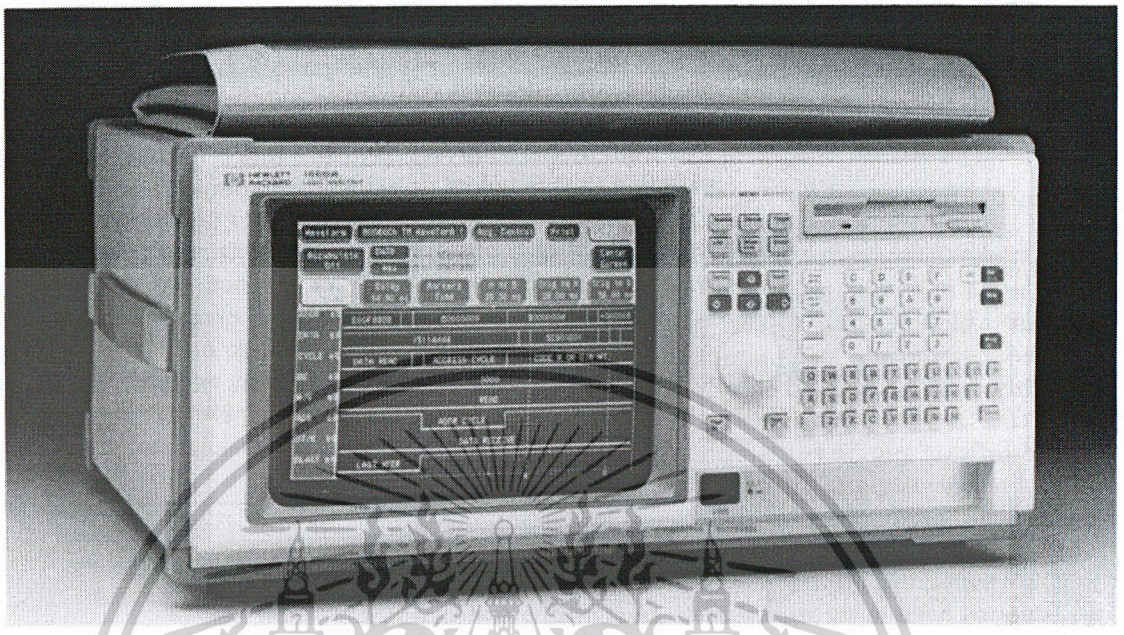
3.1.2.1 เขียนโปรแกรมรับค่าจาก Function Generator ที่ความถี่ต่างๆ โดยทำการวัดเพียงช่องสัญญาณเดียว เพื่อหาช่วงความถี่ของสัญญาณที่จะสามารถวัดได้แม่นยำ

3.1.2.2 เขียนโปรแกรมรับค่าผ่านวงจรหาร หลายช่องสัญญาณพร้อมๆกัน ในช่วงความถี่ที่หาได้จาก 3.1.2.1

3.1.2.3 เขียนโปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การใช้ Logic Analyzer สำหรับการวิเคราะห์สัญญาณ



รูปที่3.4 Logic Analyzer

โดยมีลำดับการปฏิบัติงานดังนี้

3.2.1 ทำการประมวลผลสัญญาณที่ได้จาก Function Generator โดยให้แสดงที่หน้าจอมอนิเตอร์ของเครื่องลอจิกอนาไลซ์เซอร์ โดยใช้ Probe ของเครื่องเป็นตัววัดสัญญาณ อุปกรณ์ที่ใช้มีดังนี้

- เครื่อง Logic Analyzer
- Function Generator

3.3 นำผลจากการประมวลสัญญาณที่ได้จาก PCI Logic Card และ Logic Analyzer มาเปรียบเทียบ

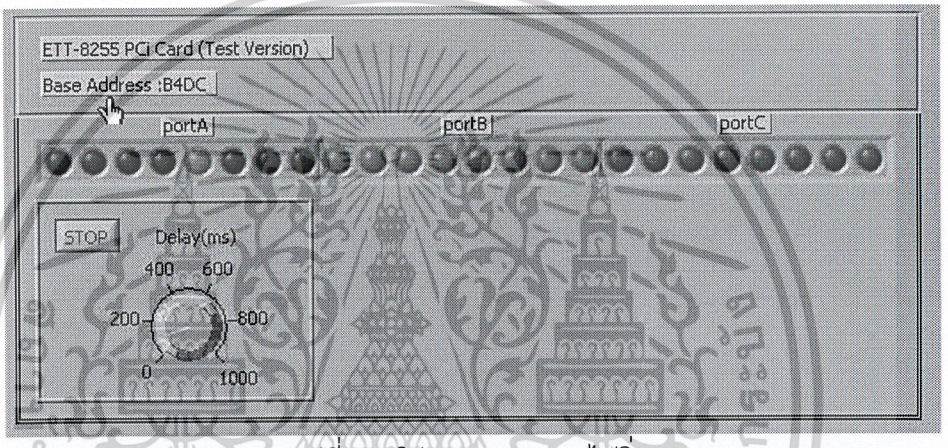
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและการอภิปรายผล

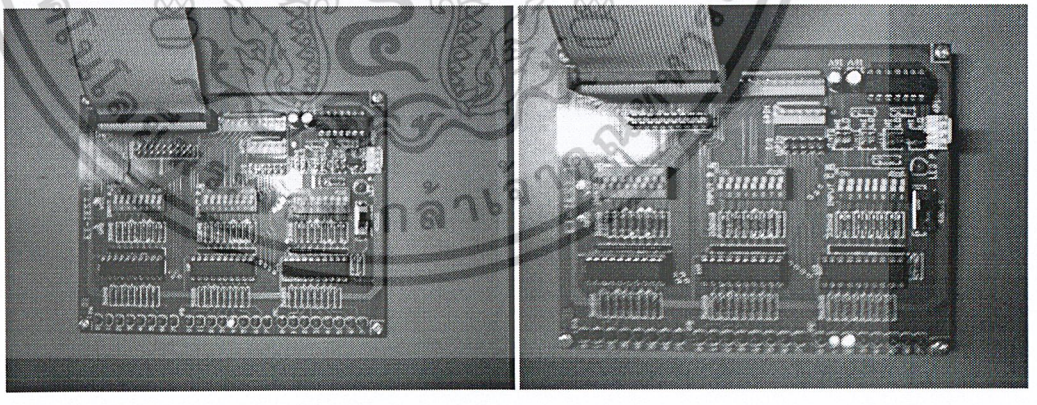
4.1 การใช้ PCI Logic Card สำหรับการวิเคราะห์สัญญาณ

4.1.1 การรับค่าผ่าน I/O

4.1.1.1 โปรแกรมไฟวิ่ง



รูปที่4.1 โปรแกรมควบคุมไฟวิ่ง

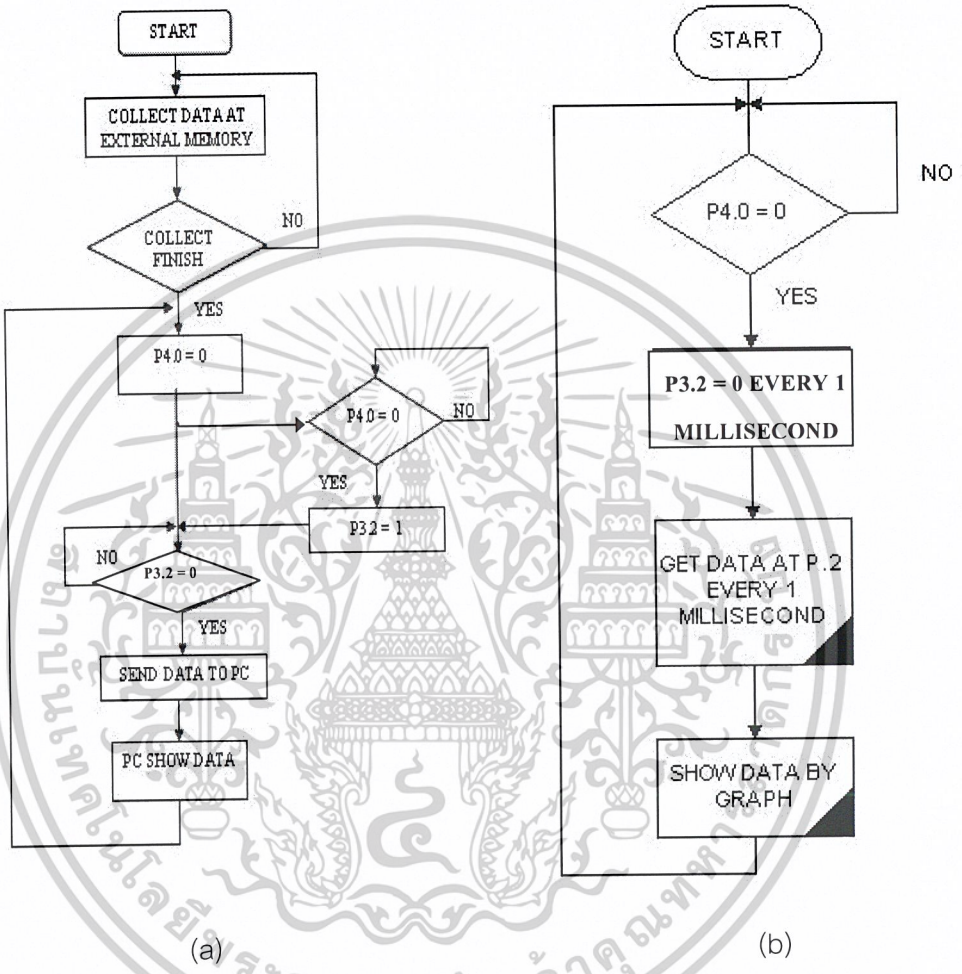


รูปที่4.2 ผลจากการควบคุมไฟวิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 รับค่าจาก Function Generator ผ่านบอร์ดไมโครคอนโทรเลอร์เบอร์ T89C51RD2

4.1.2.1 รับค่าจาก Function Generator ที่ความถี่ต่างๆโดยทำการวัดเพียงช่องสัญญาณเดียว เพื่อหาช่วงความถี่ของสัญญาณที่จะสามารถวัดได้แม่นยำ

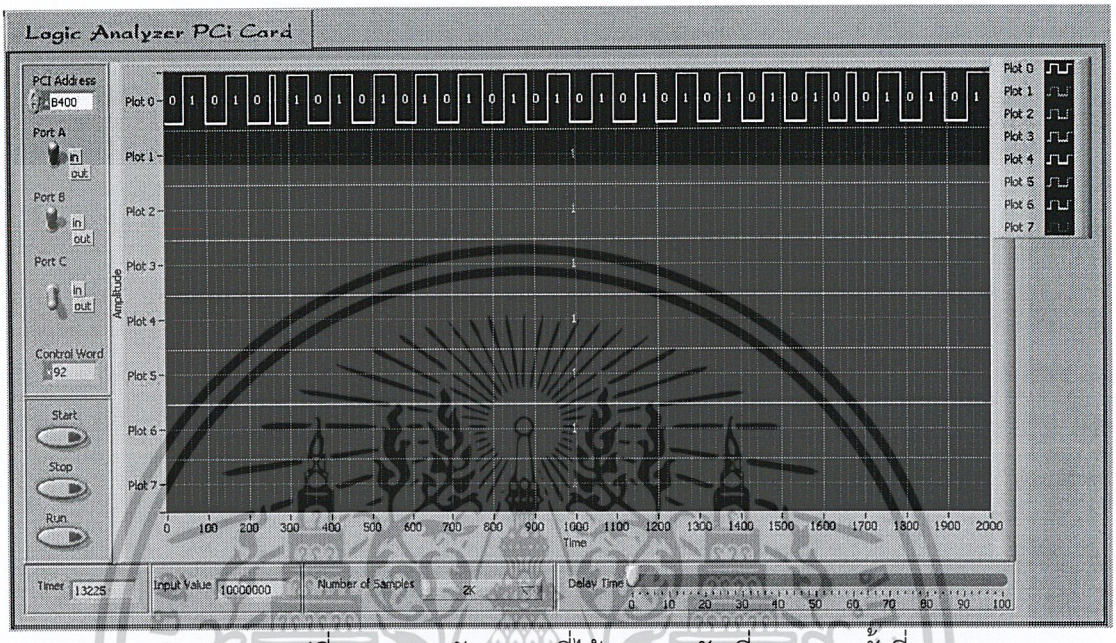


รูปที่4.3 (a) แสดงการทำงานของโปรแกรมแอสเซมบลีบนบอร์ดไมโครคอนโทรเลอร์

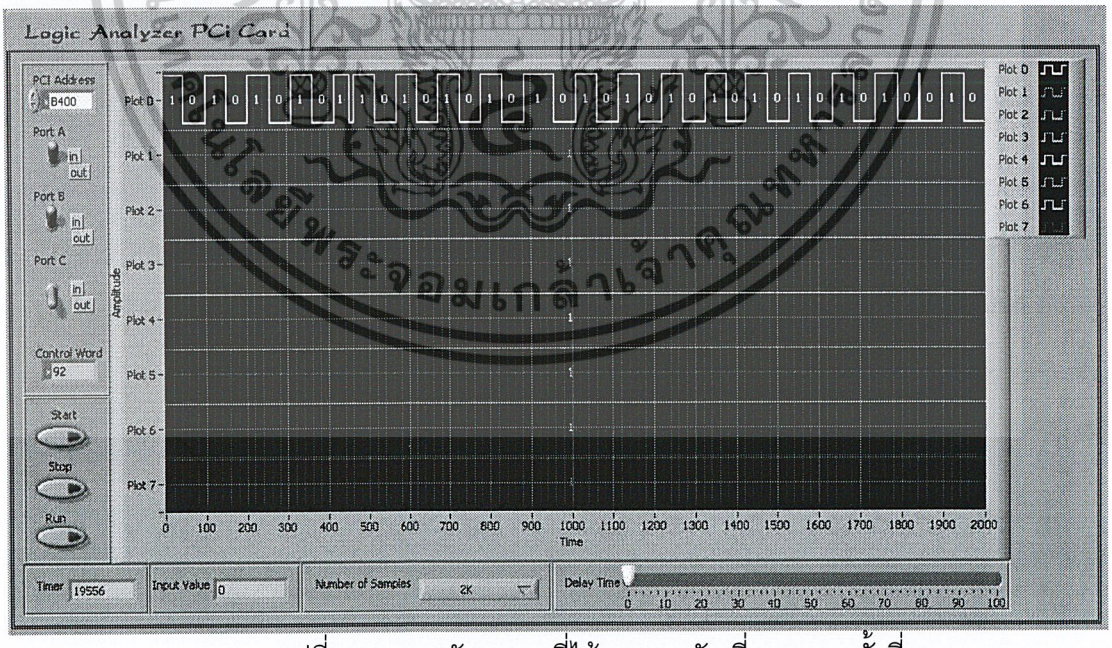
(b) โฟลวชาตแสดงการทำงานของโปรแกรม Labview บน PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 5 kHz.

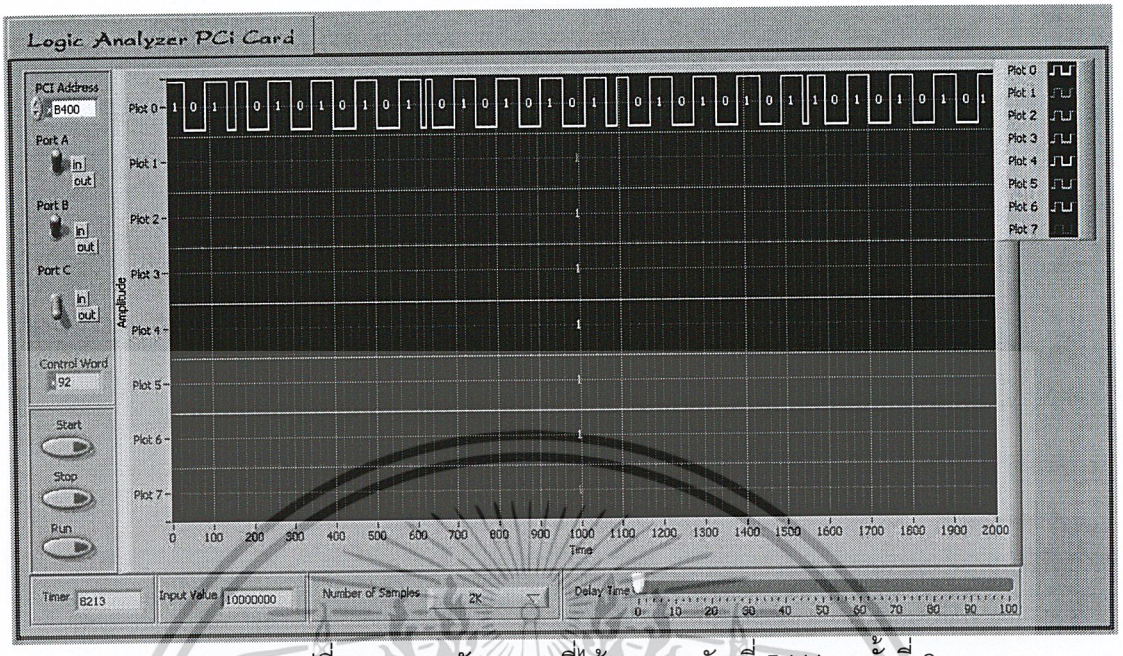


รูปที่4.4 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 1



รูปที่4.5 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

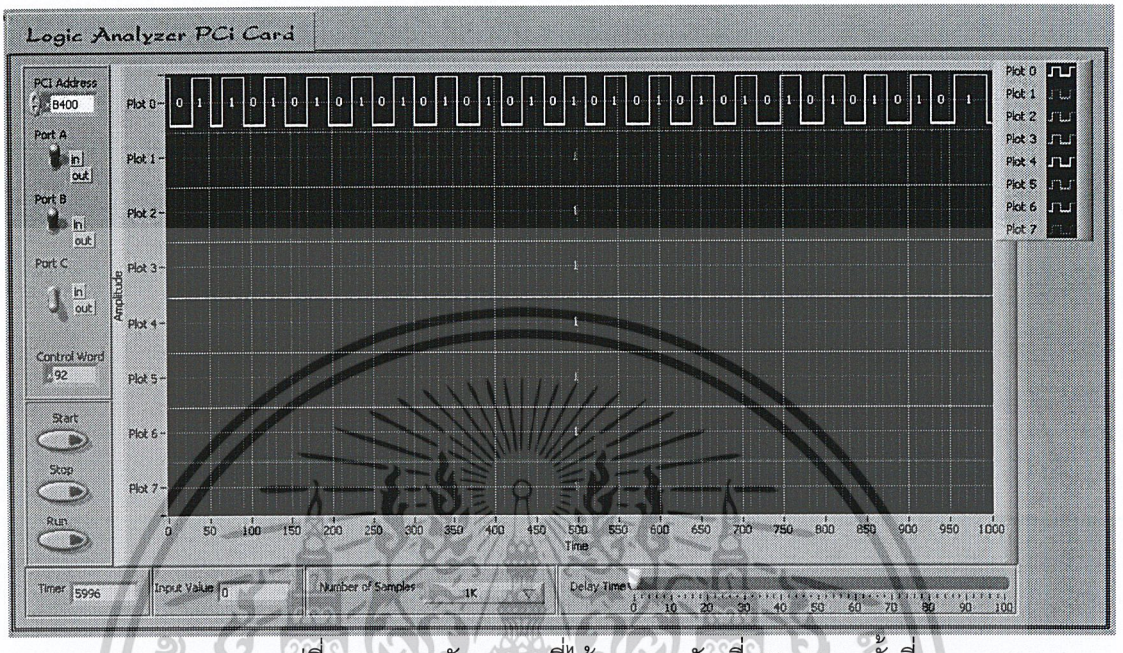


รูปที่4.6 แสดงสัญญาณที่ได้จากการวัด ที่ 5 kHz. ครั้งที่ 3

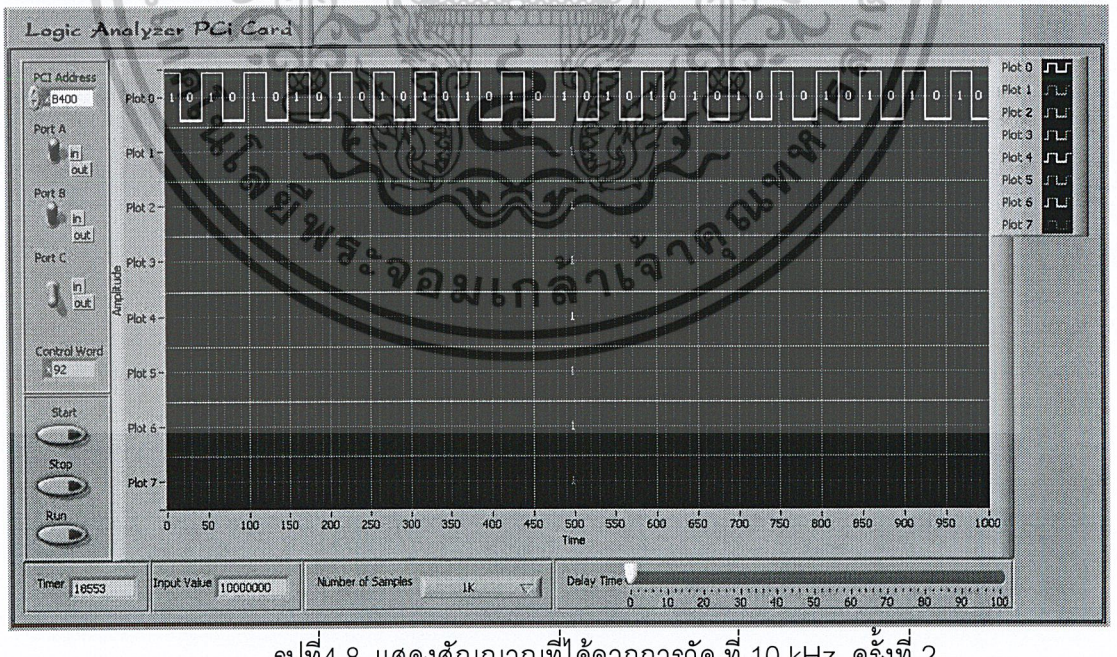
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 10 kHz.

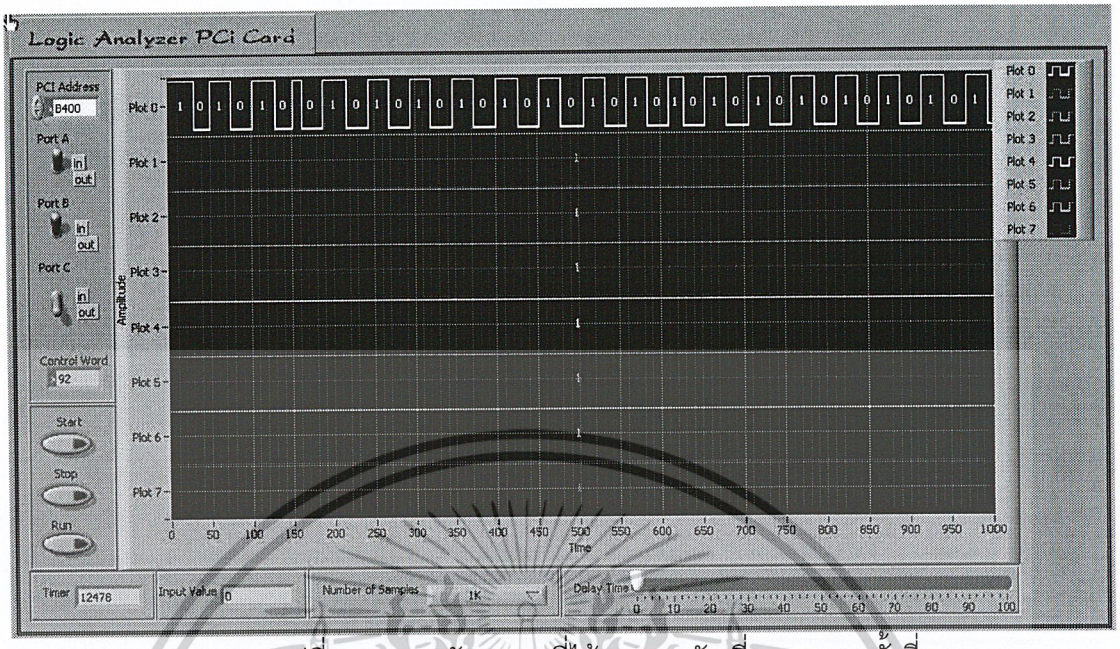


รูปที่4.7 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 1



รูปที่4.8 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 2

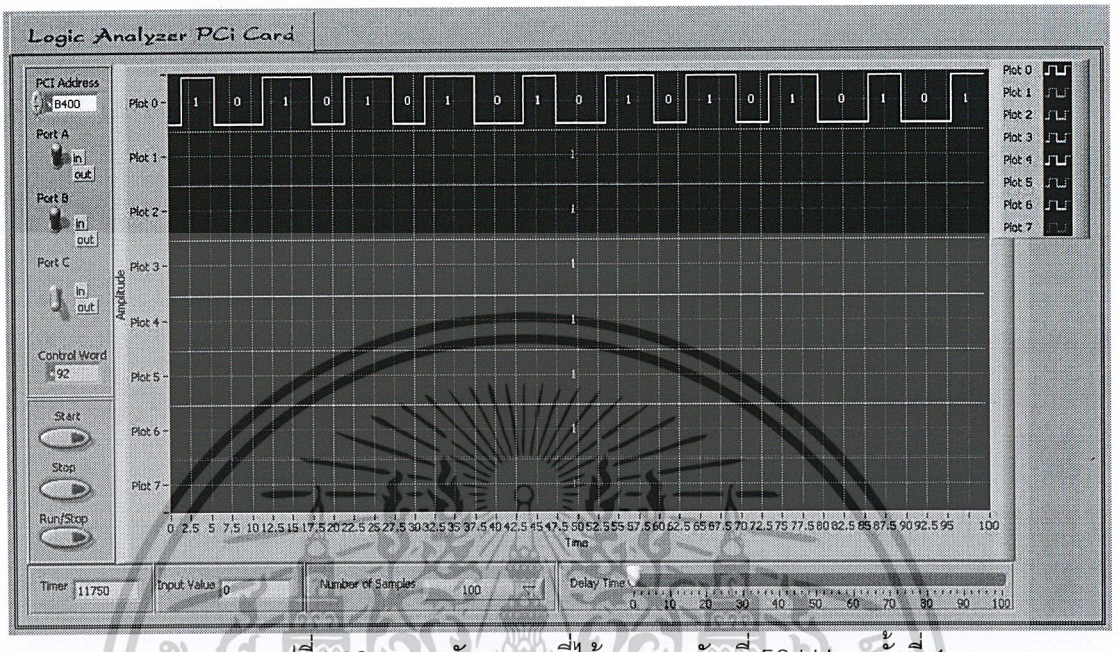
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



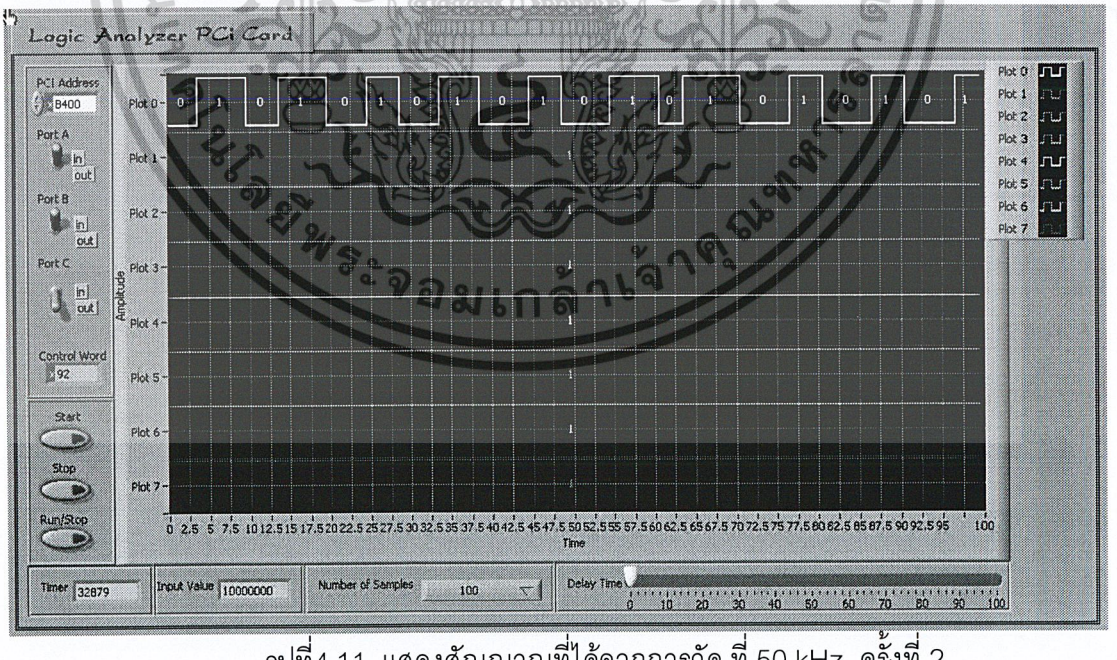
รูปที่4.9 แสดงสัญญาณที่ได้จากการวัด ที่ 10 kHz. ครั้งที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 50 kHz.

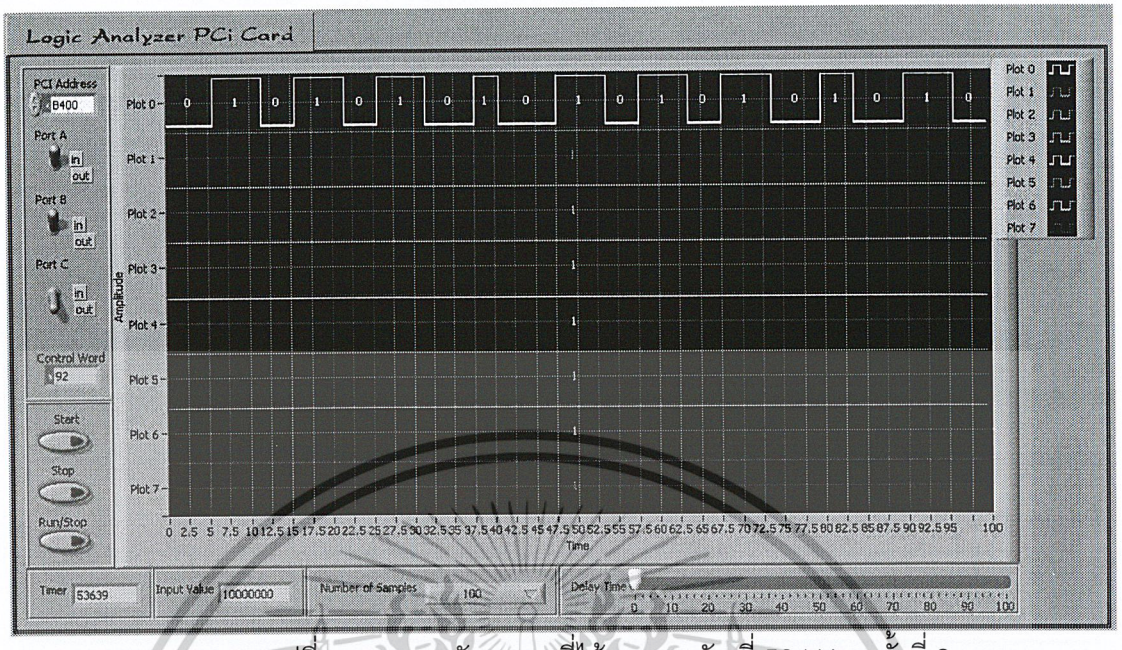


รูปที่4.10 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 1



รูปที่4.11 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 2

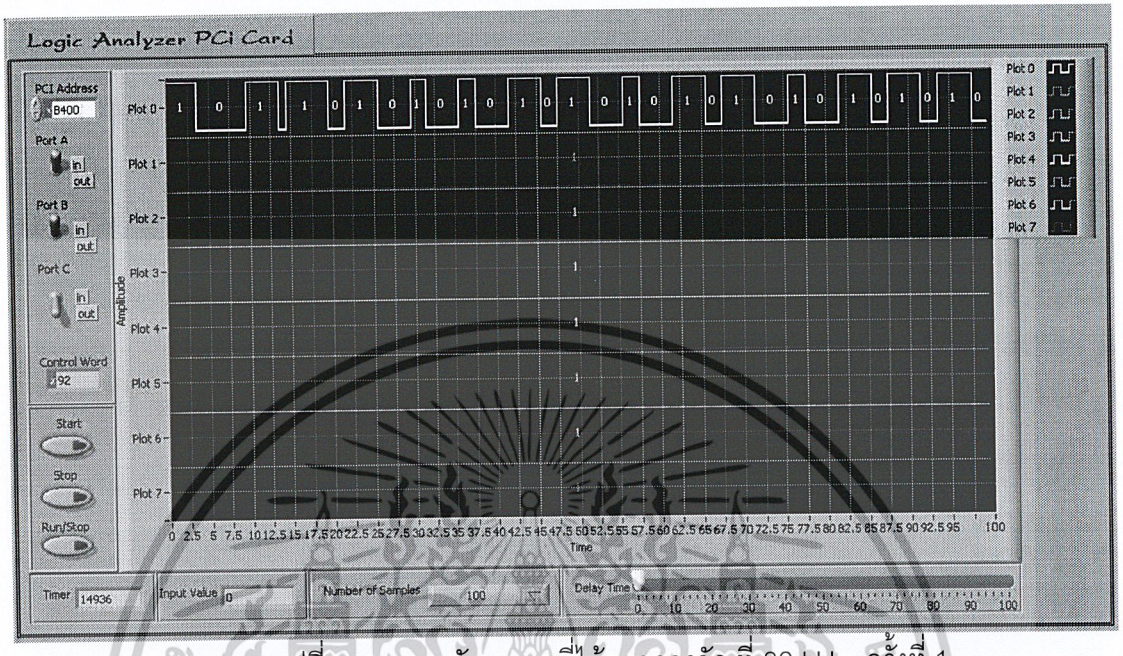
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



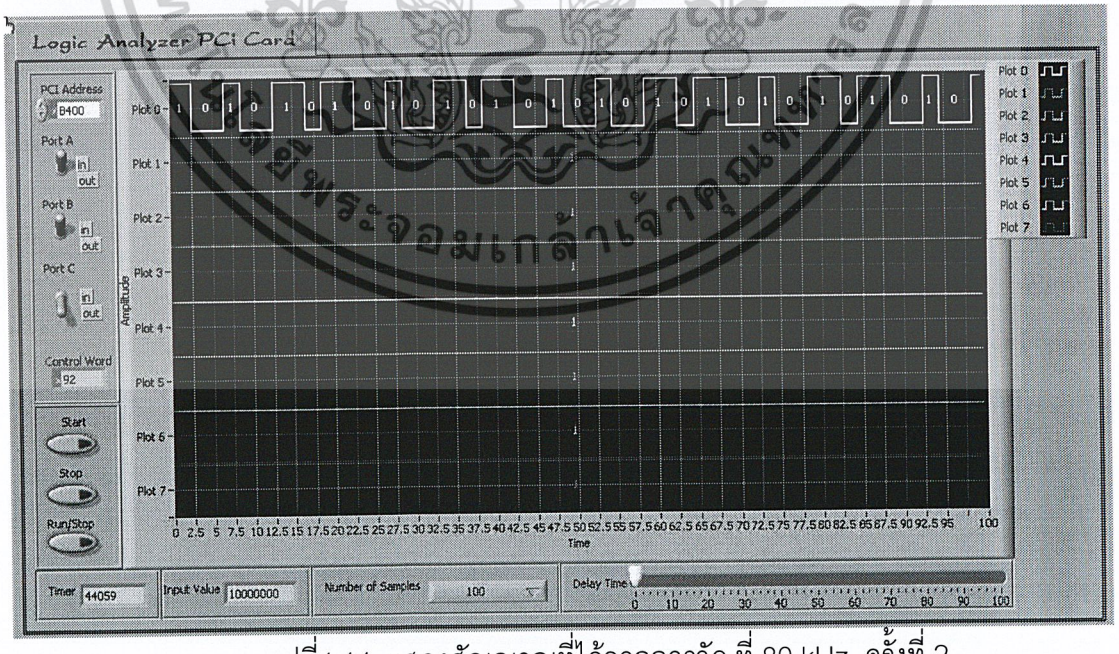
รูปที่4.12 แสดงสัญญาณที่ได้จากการวัด ที่ 50 kHz. ครั้งที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 80 kHz.

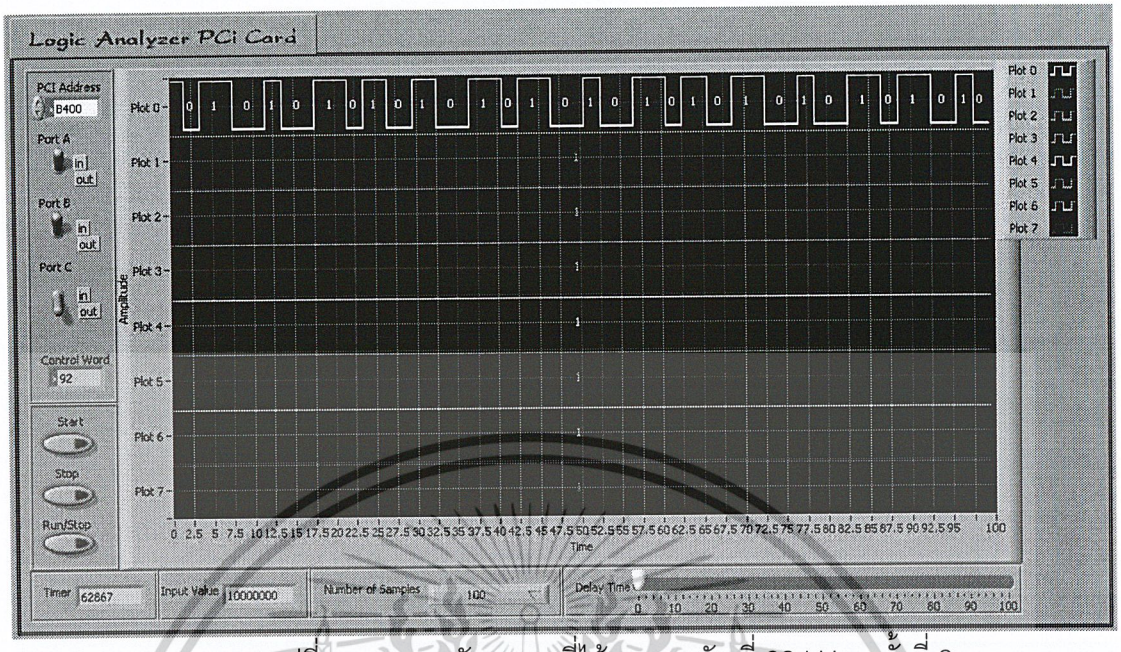


รูปที่4.13 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 1



รูปที่4.14 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

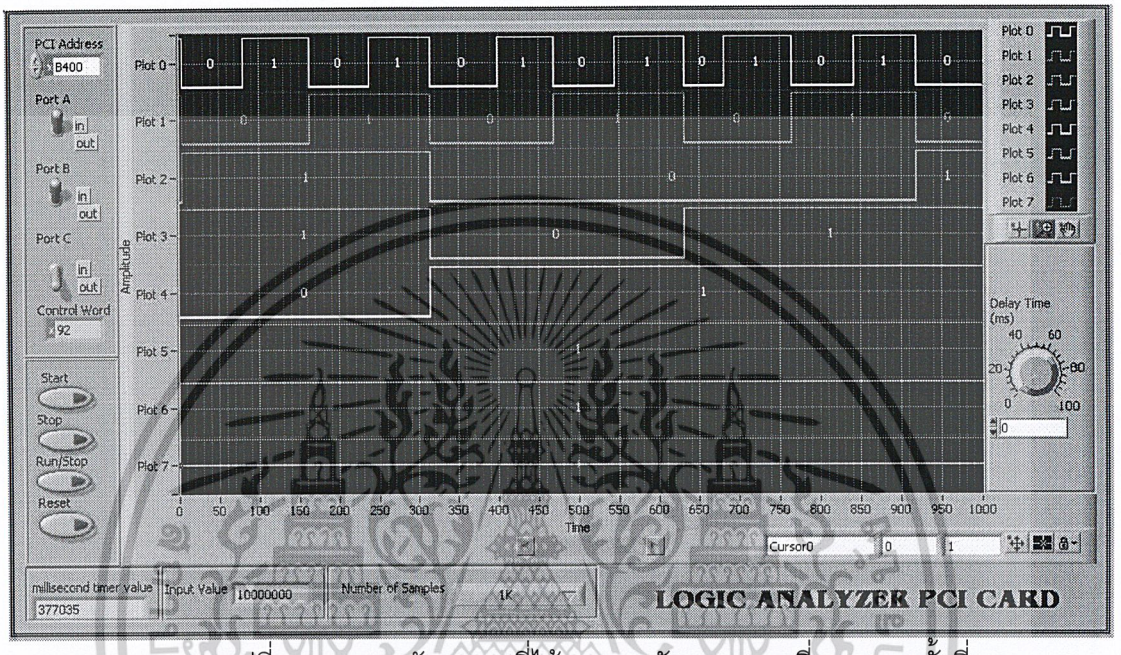


รูปที่4.15 แสดงสัญญาณที่ได้จากการวัด ที่ 80 kHz. ครั้งที่ 3

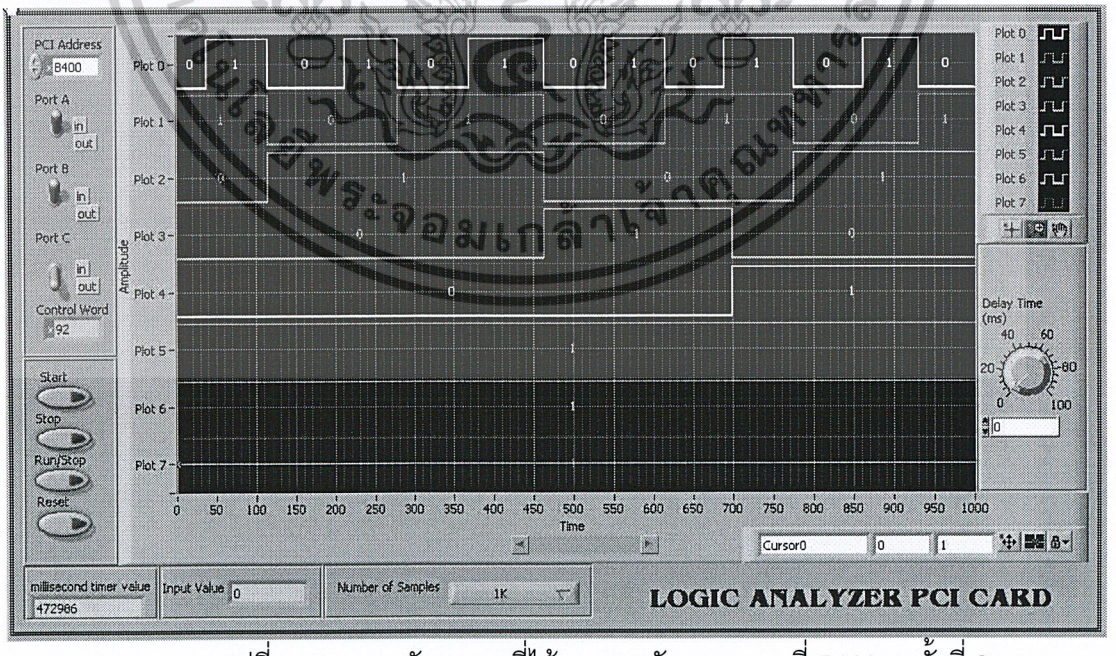
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2 รับค่าผ่านวงจรหารหลายช่องสัญญาณพร้อมๆกัน ในช่วงความถี่ที่หาได้

- ที่ความถี่ 5 kHz.

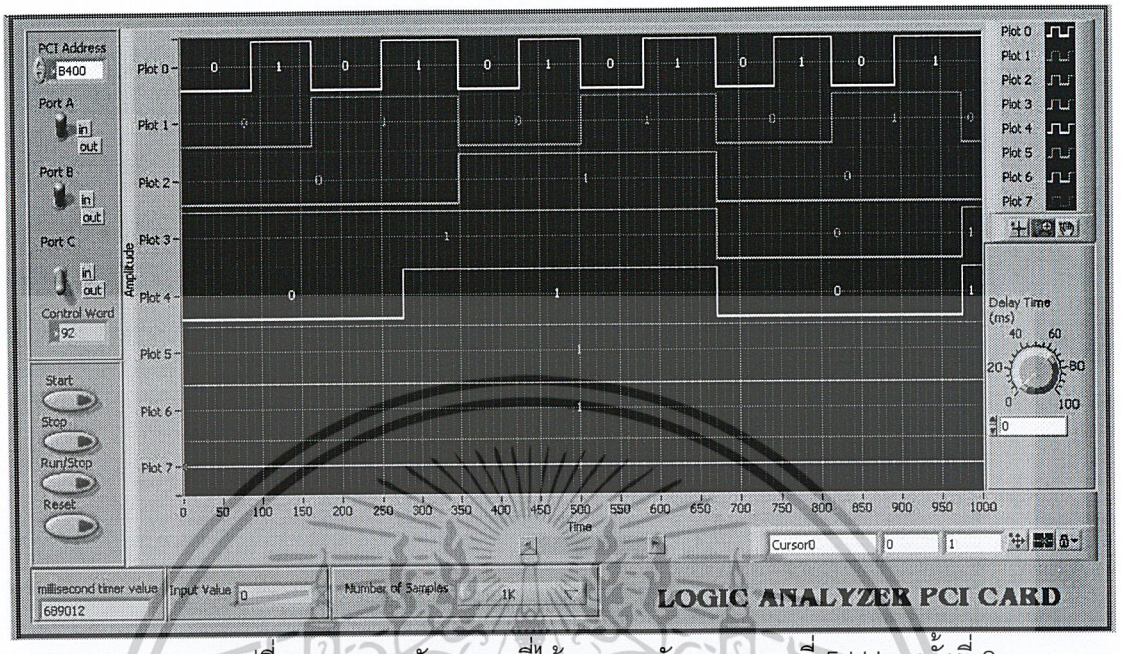


รูปที่4.16 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 1

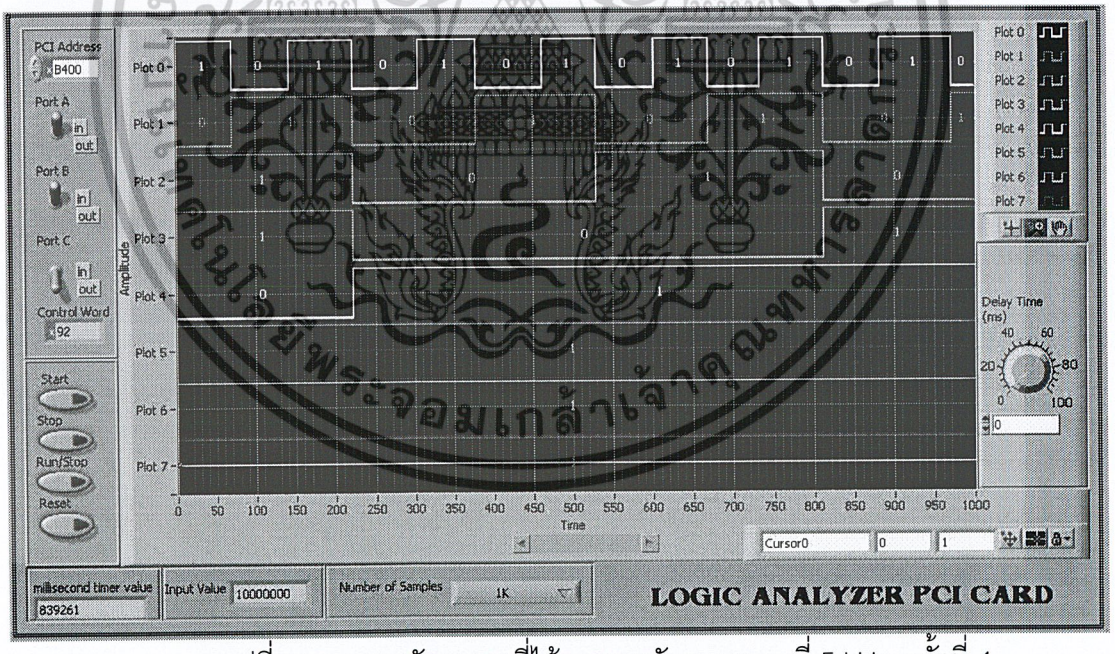


รูปที่4.17 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

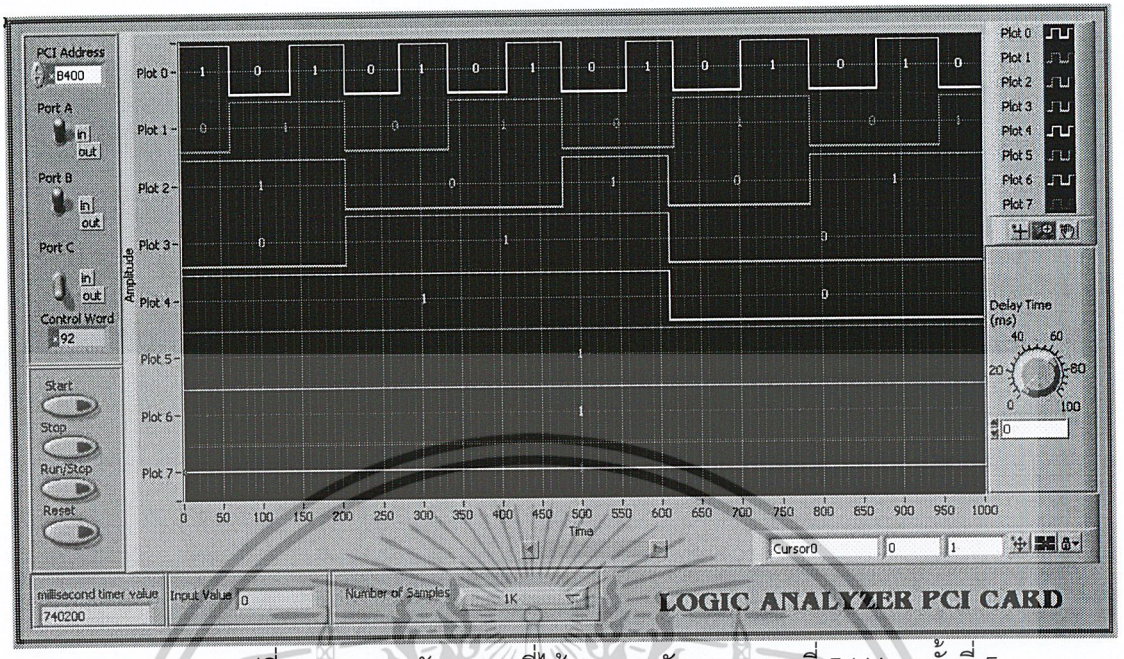


รูปที่4.18 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 3



รูปที่4.19 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 4

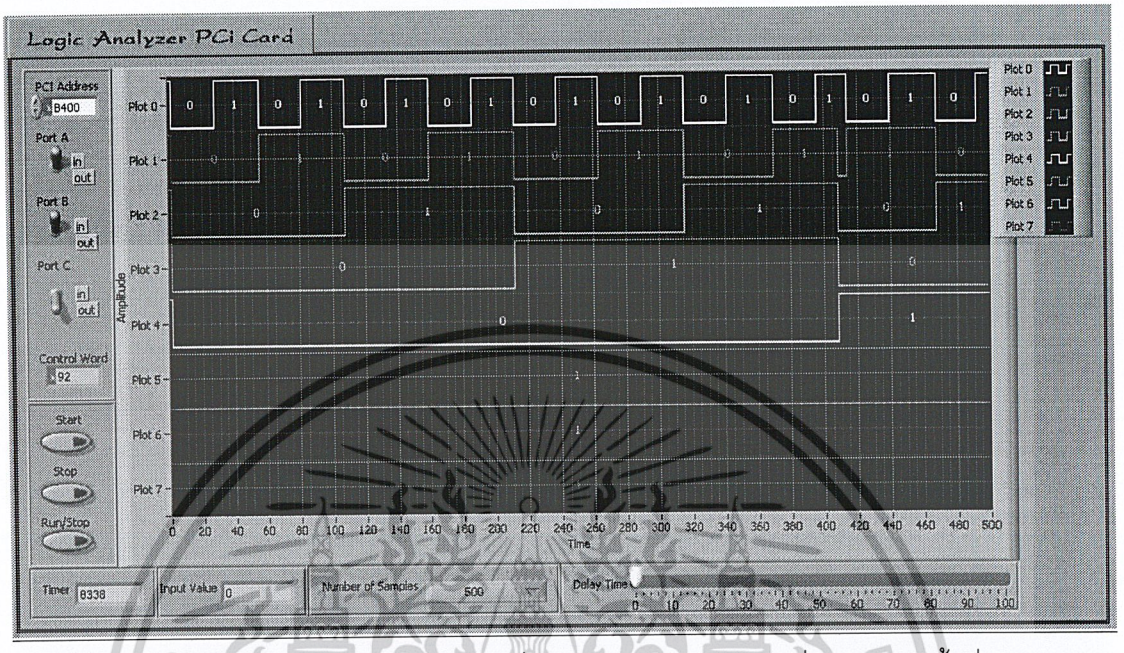
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



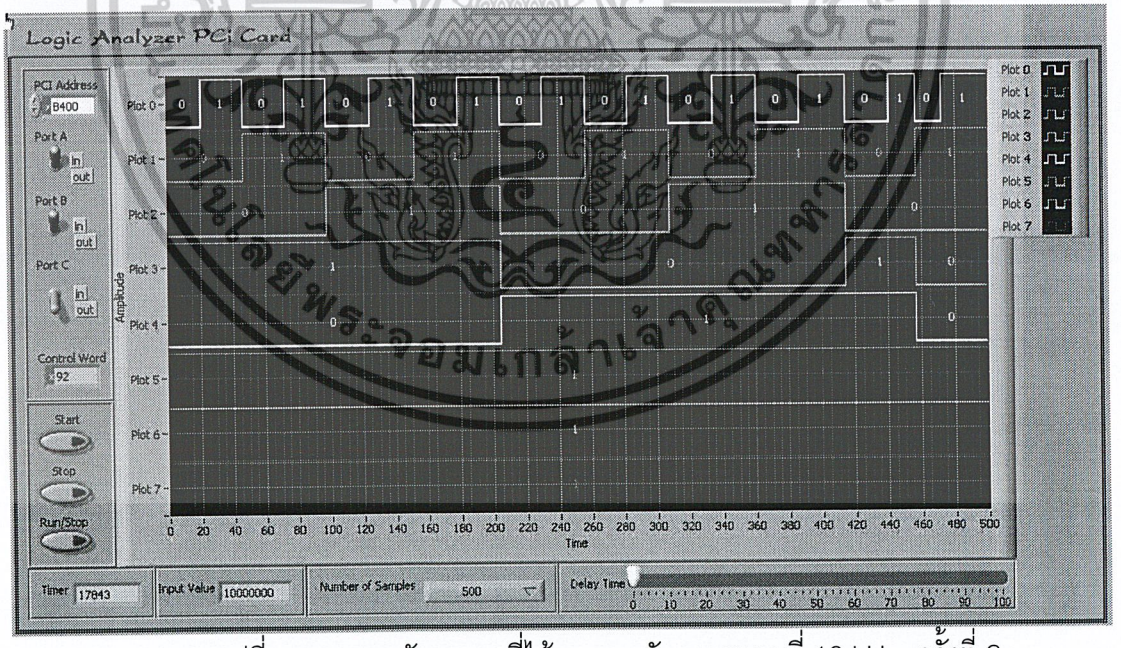
รูปที่4.20 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 5 kHz. ครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 10 kHz.

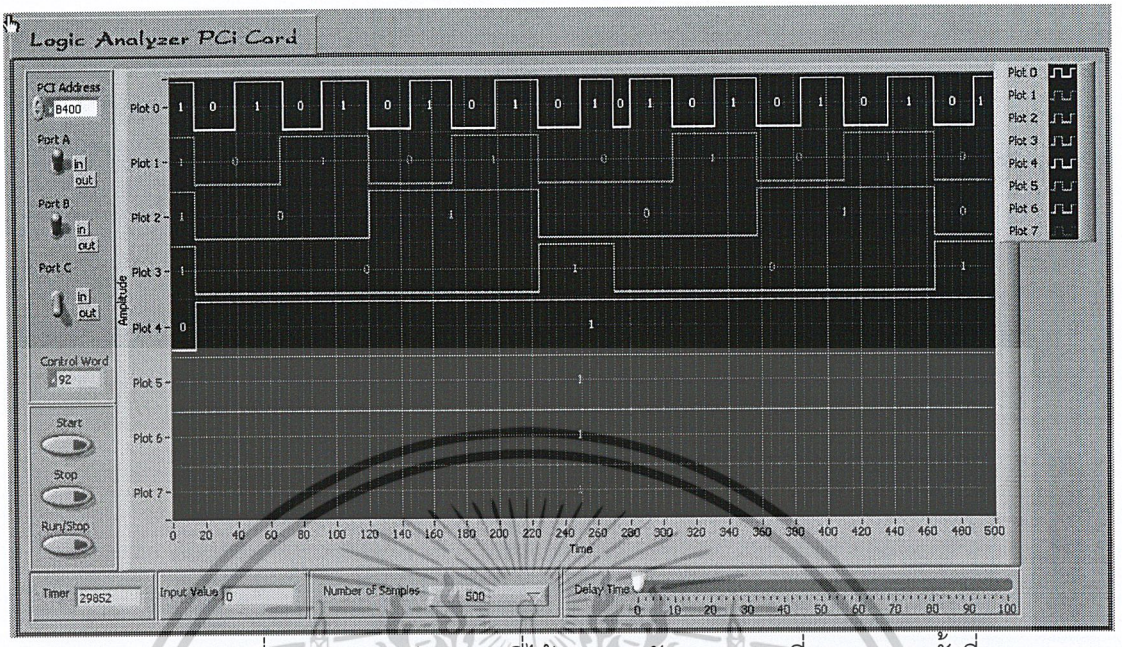


รูปที่4.21 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 1

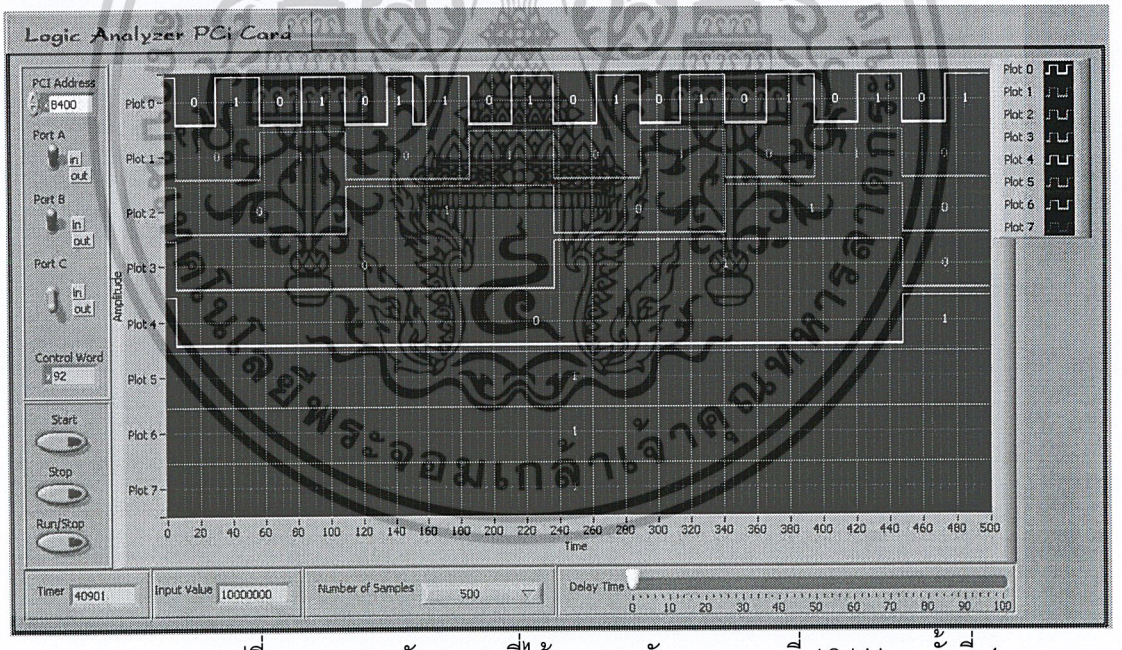


รูปที่4.22 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

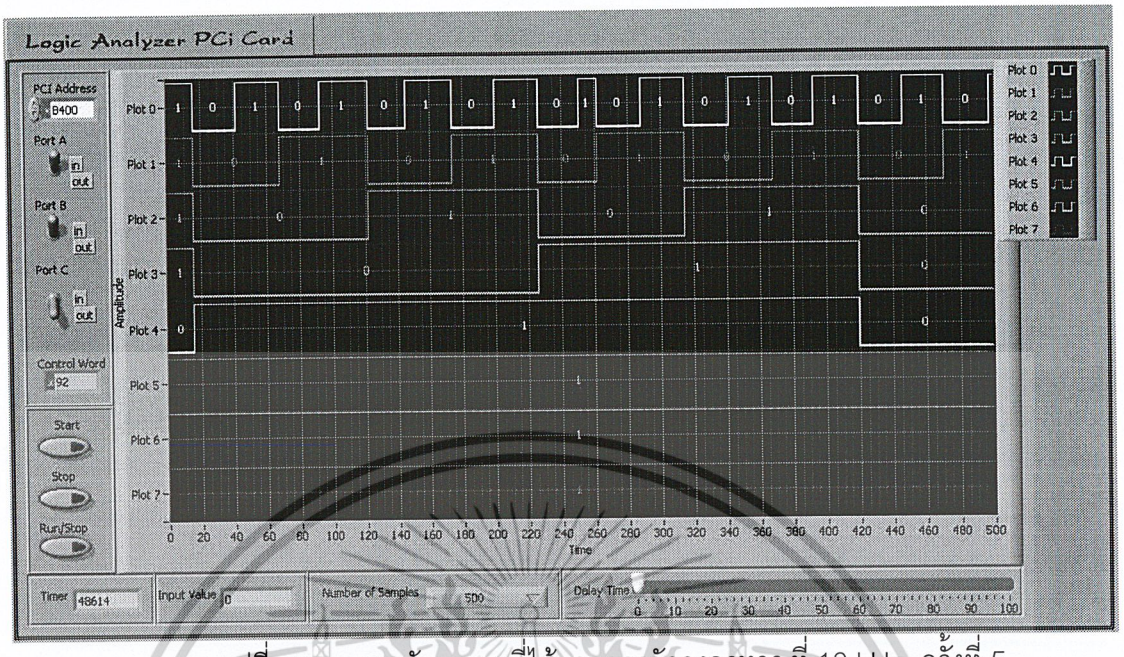


รูปที่4.23 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 3



รูปที่4.24 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 4

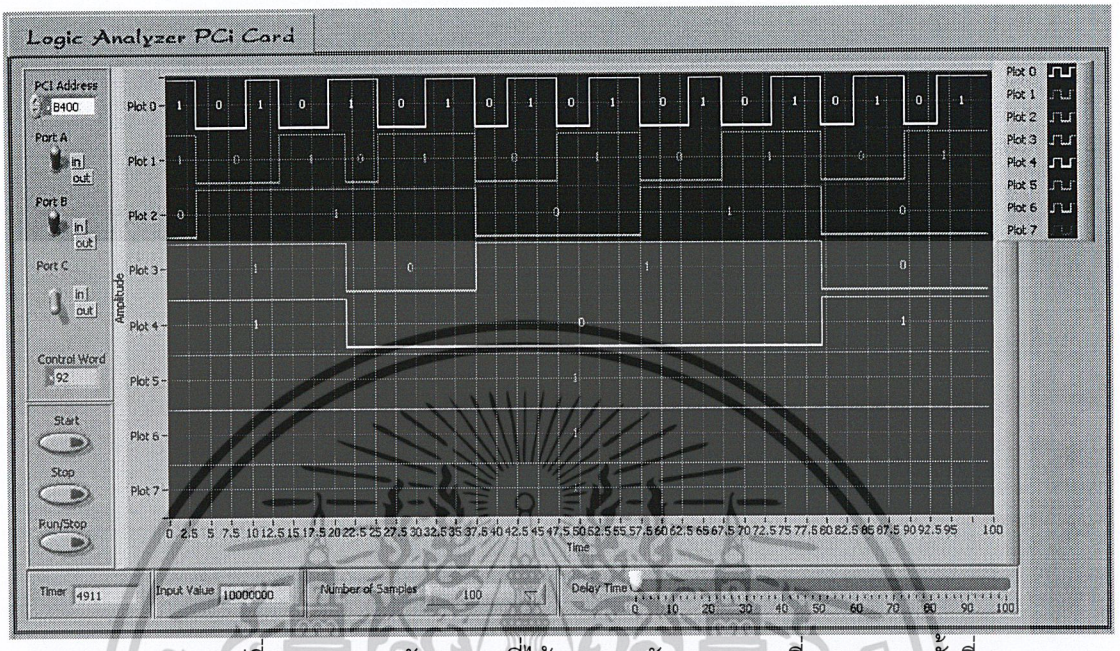
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



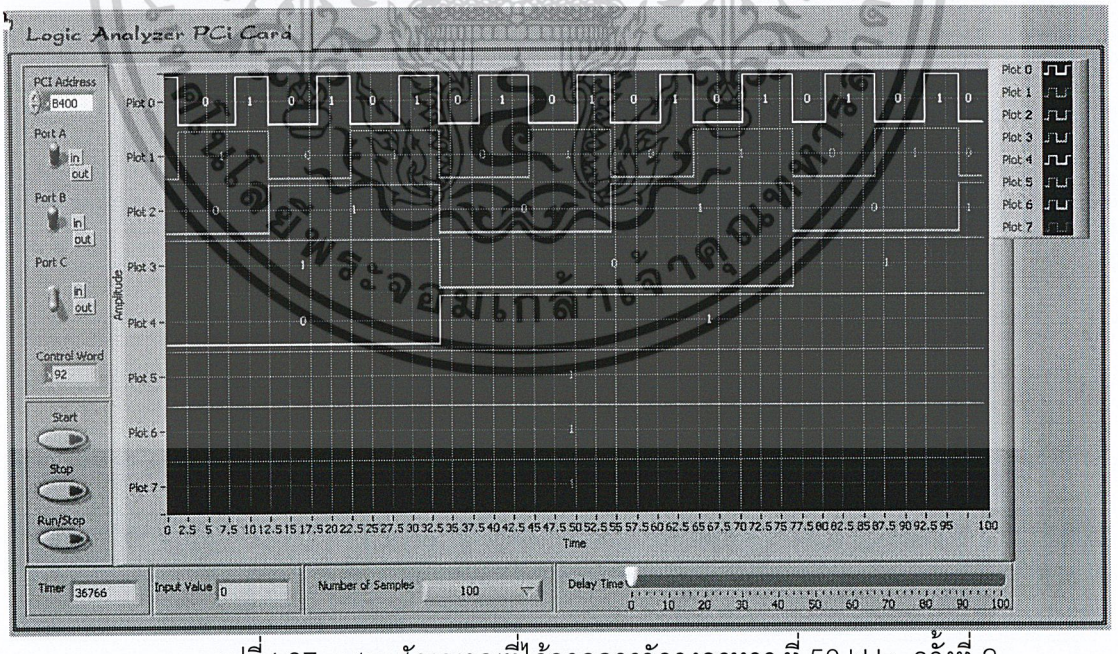
รูปที่4.25 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 10 kHz. ครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 50 kHz.

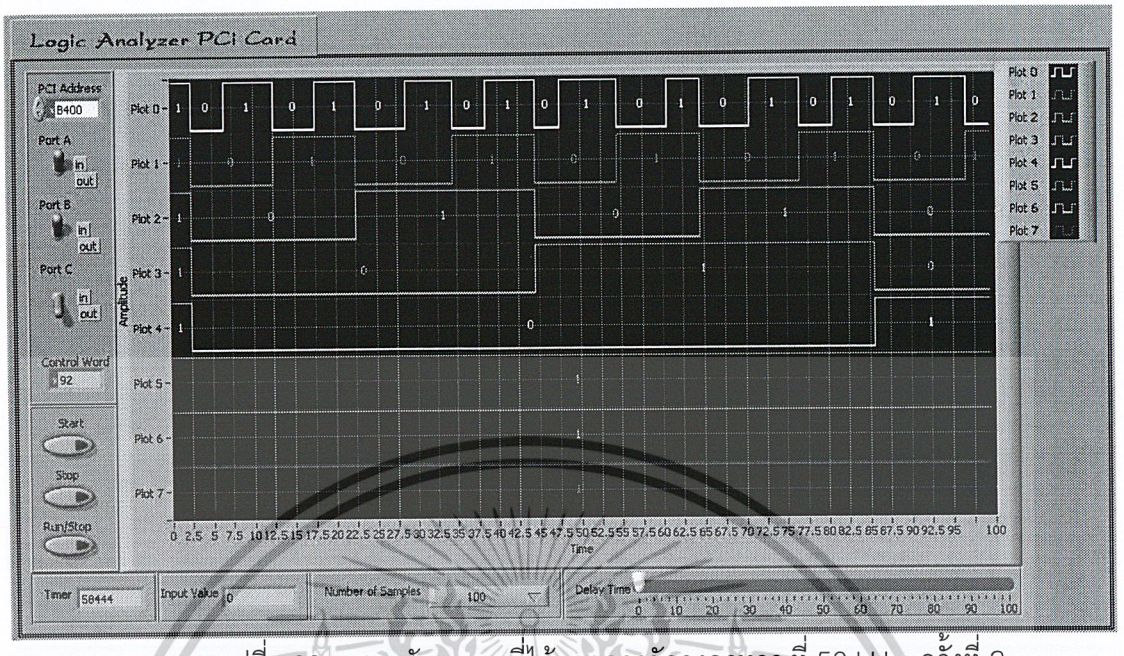


รูปที่4.26 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 1

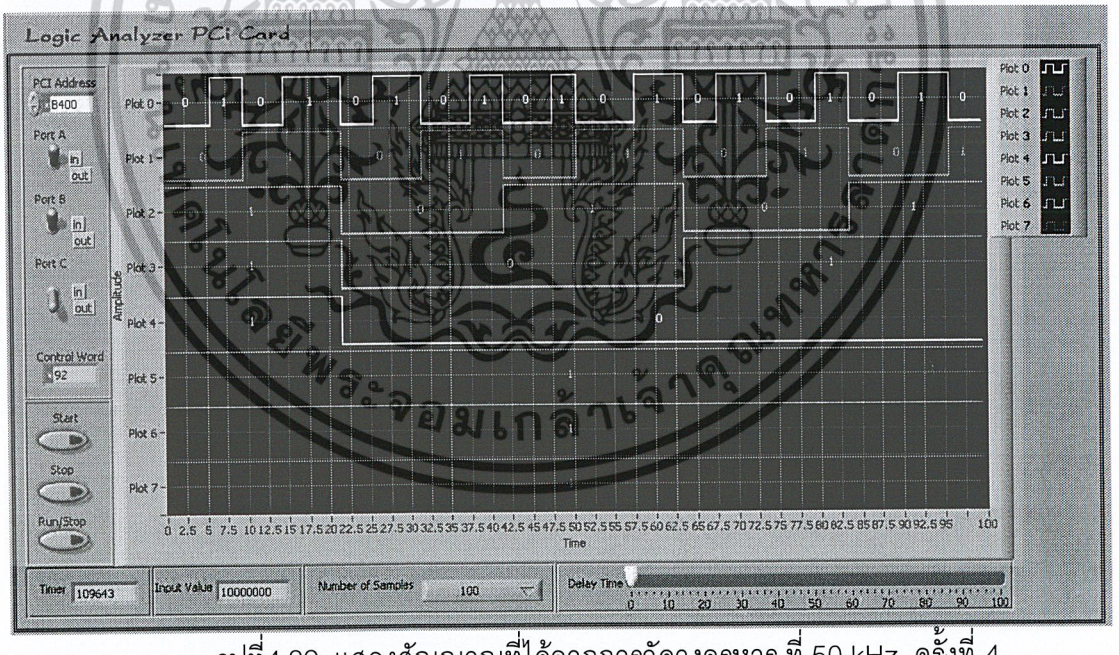


รูปที่4.27 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

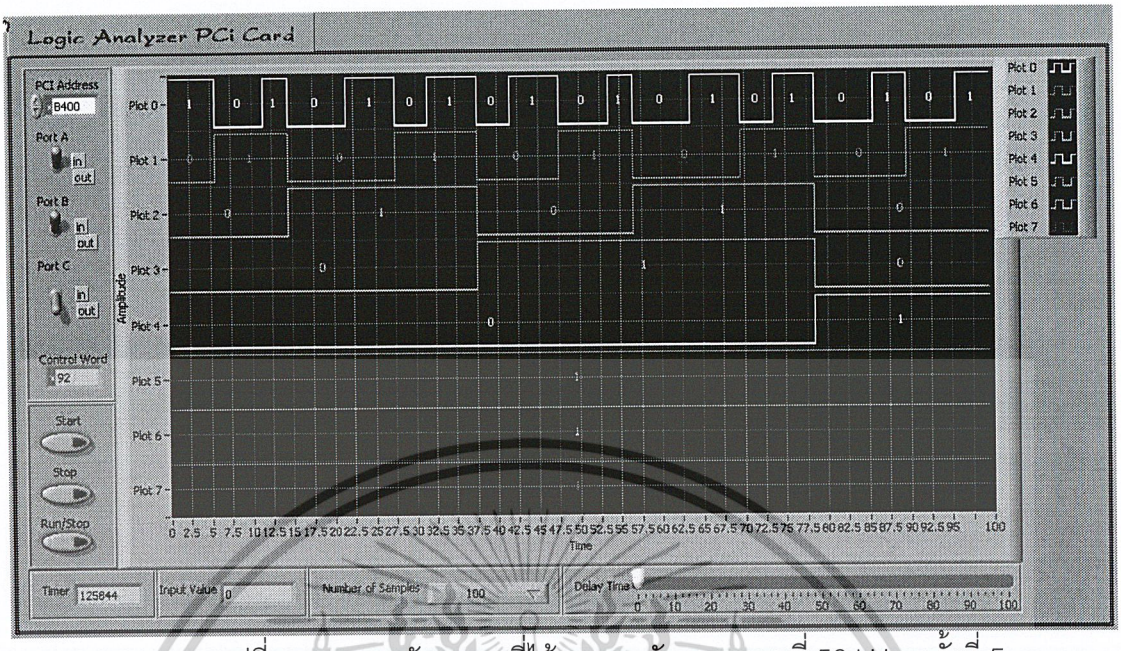


รูปที่4.28 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 3



รูปที่4.29 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

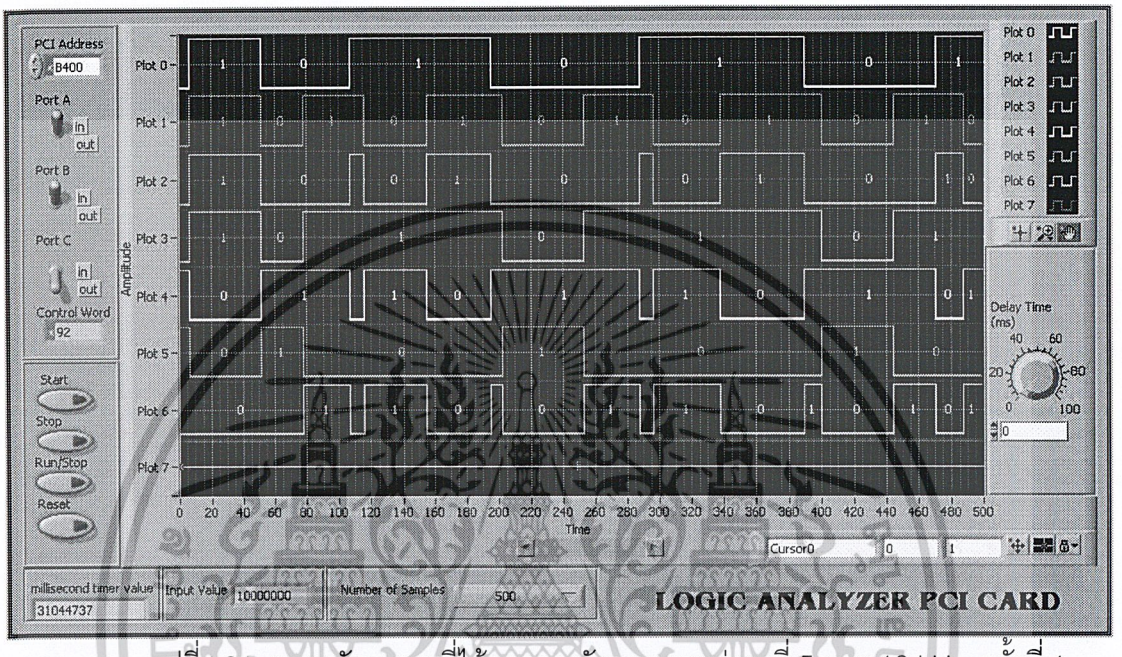


รูปที่4.30 แสดงสัญญาณที่ได้จากการวัดวงจรหาร ที่ 50 kHz. ครั้งที่ 5

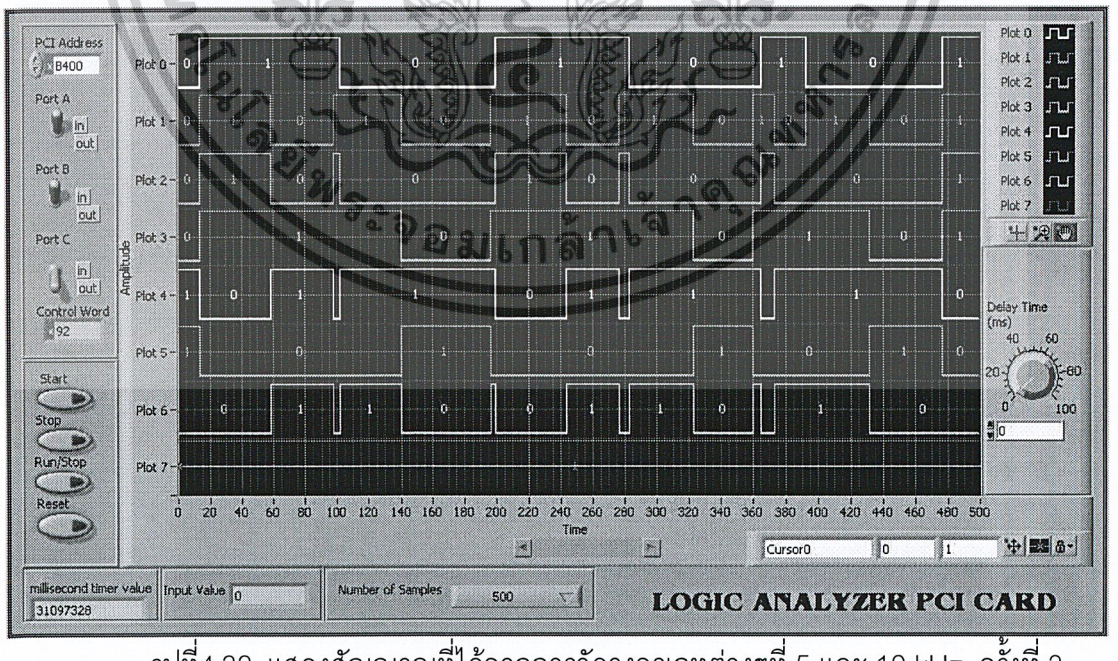
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.3 โปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่างๆ

- ที่ความถี่ 5 kHz.และ10 kHz.

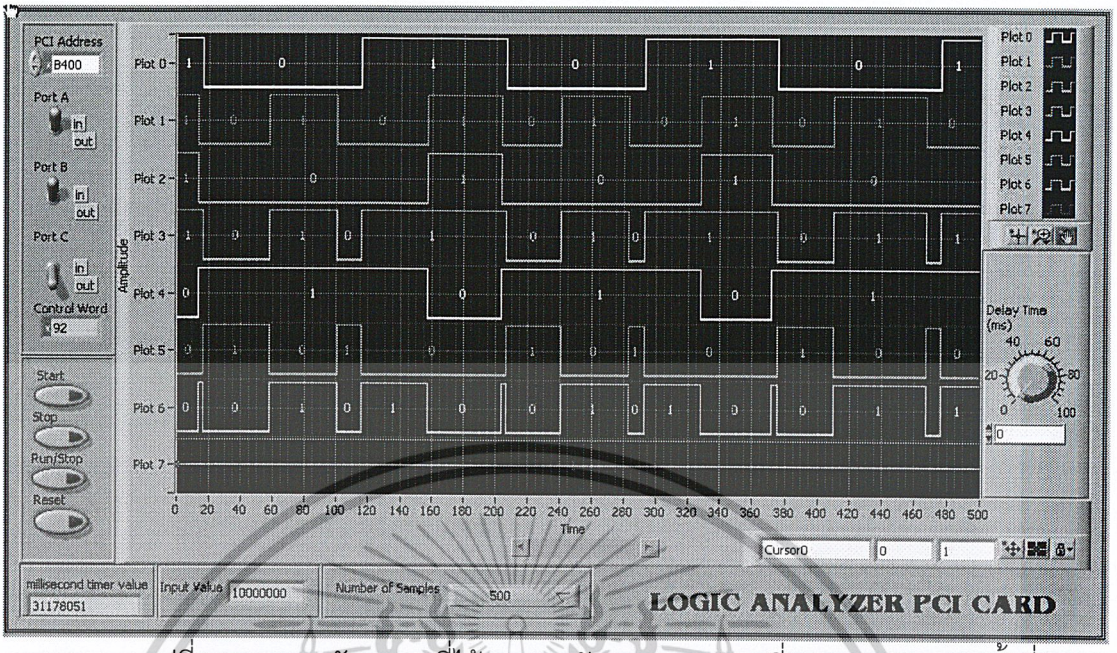


รูปที่4.31 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 1

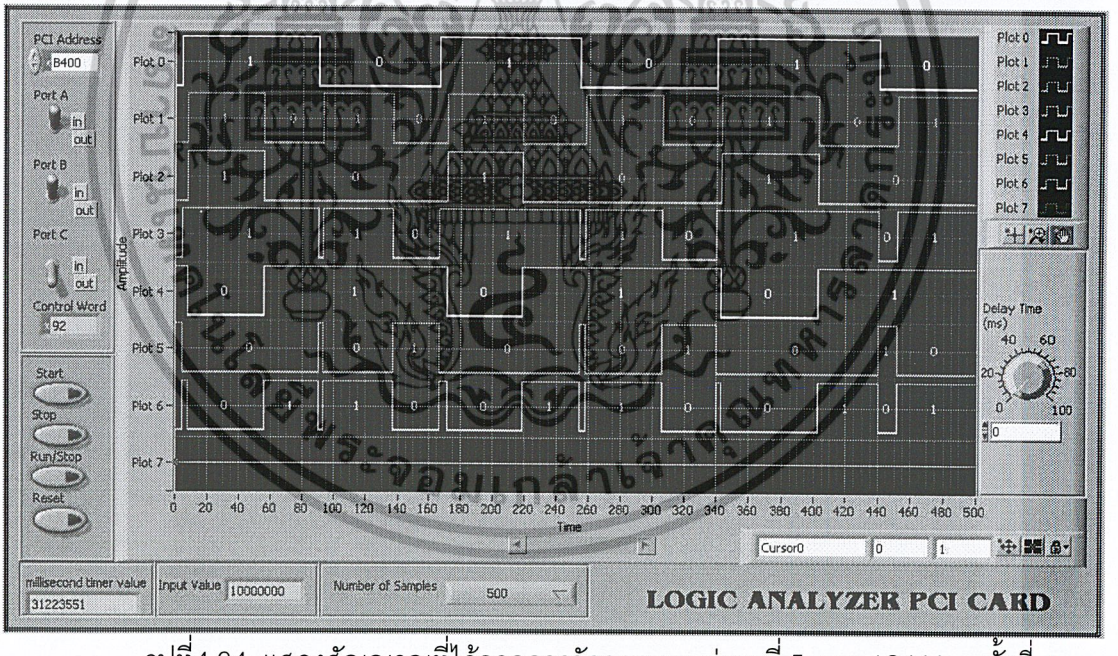


รูปที่4.32 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

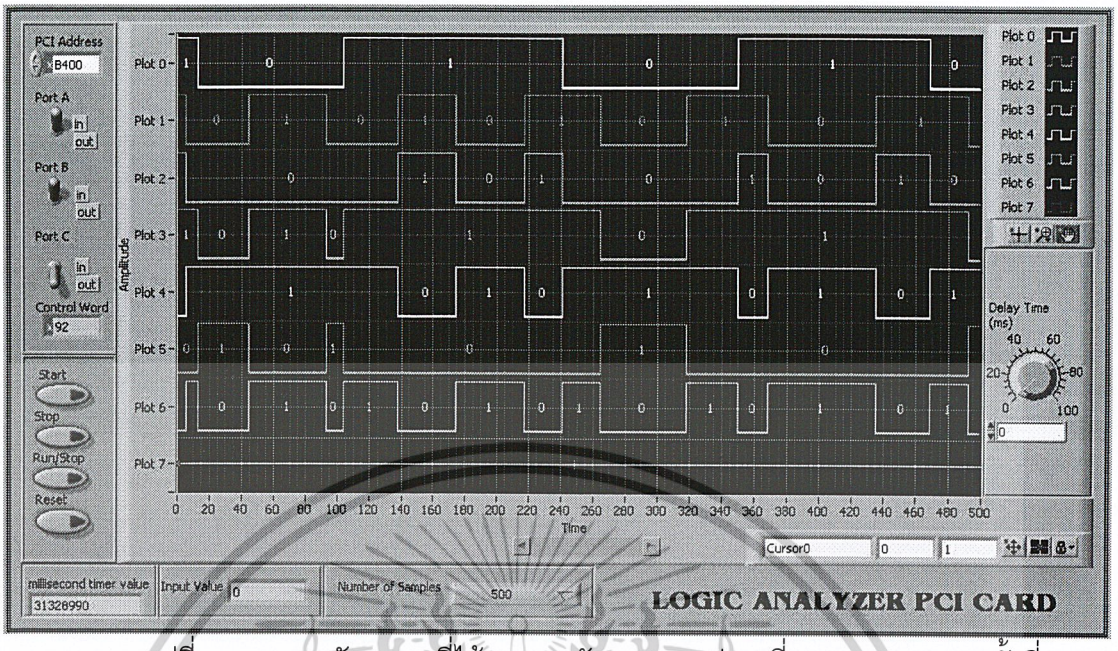


รูปที่4.33 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 3



รูปที่4.34 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 4

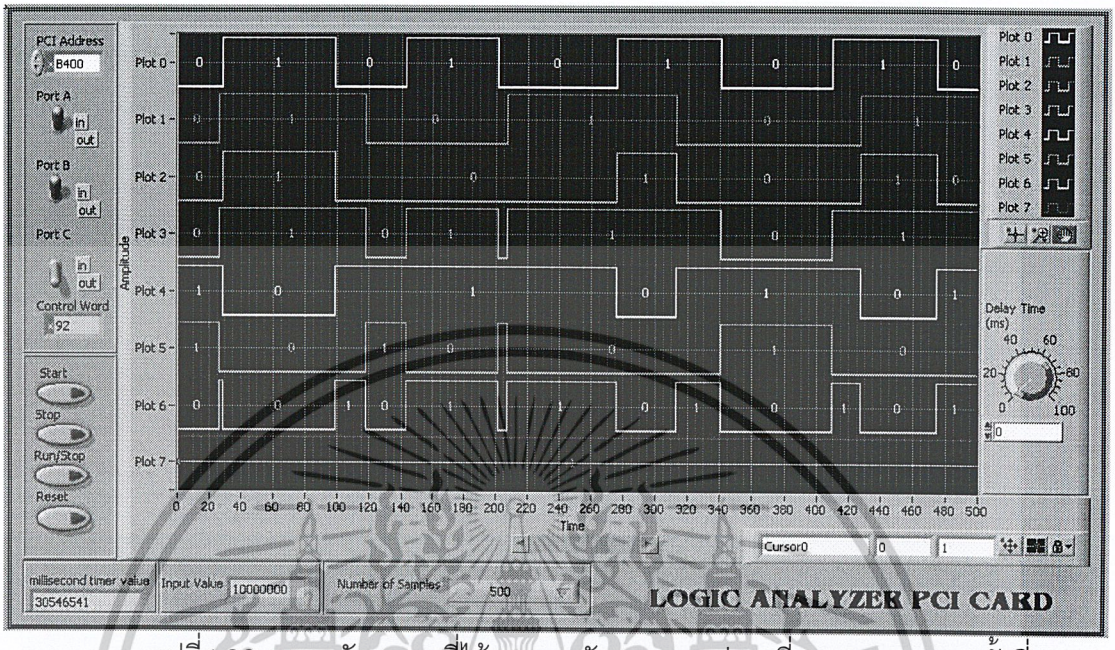
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



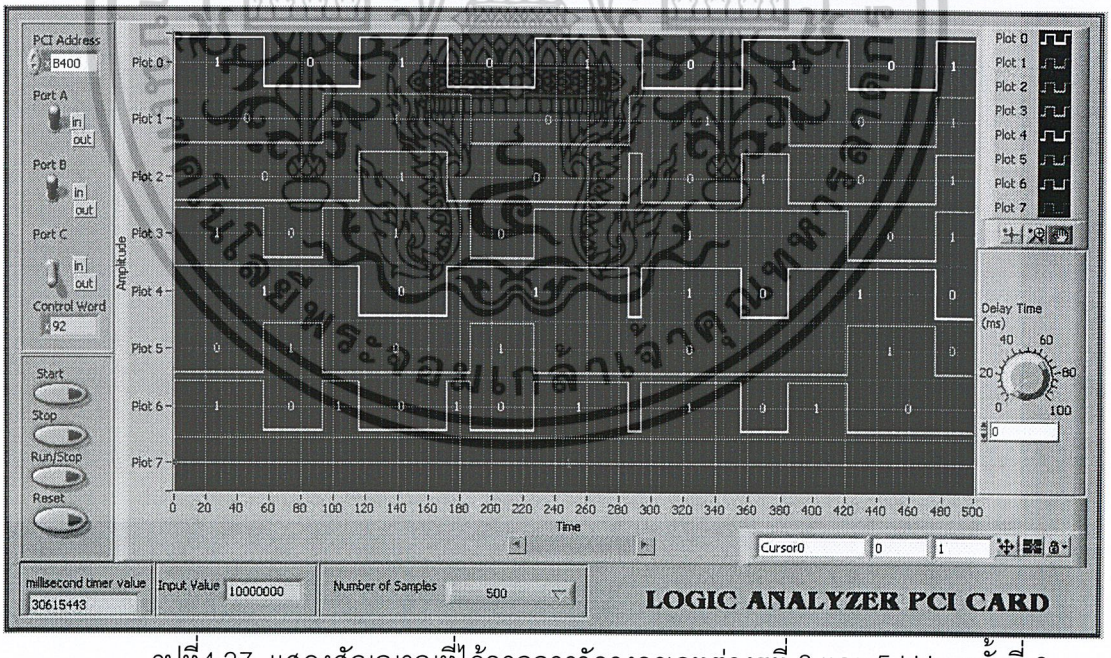
รูปที่4.35 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 5 และ 10 kHz. ครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ความถี่ 8 kHz.และ 5 kHz.

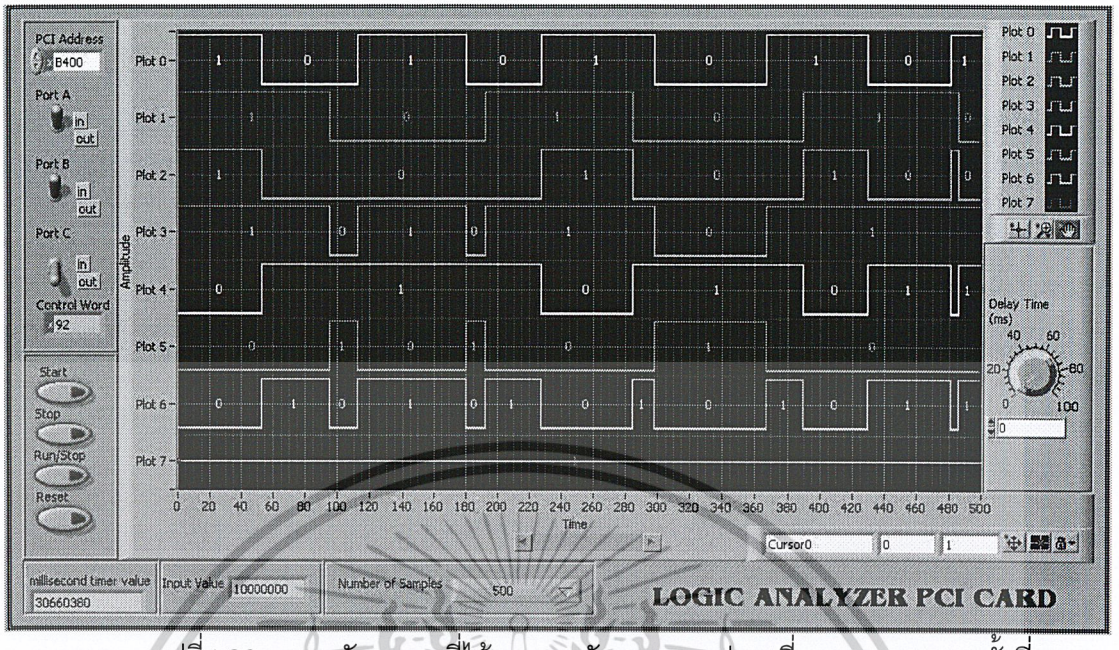


รูปที่4.36 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 1

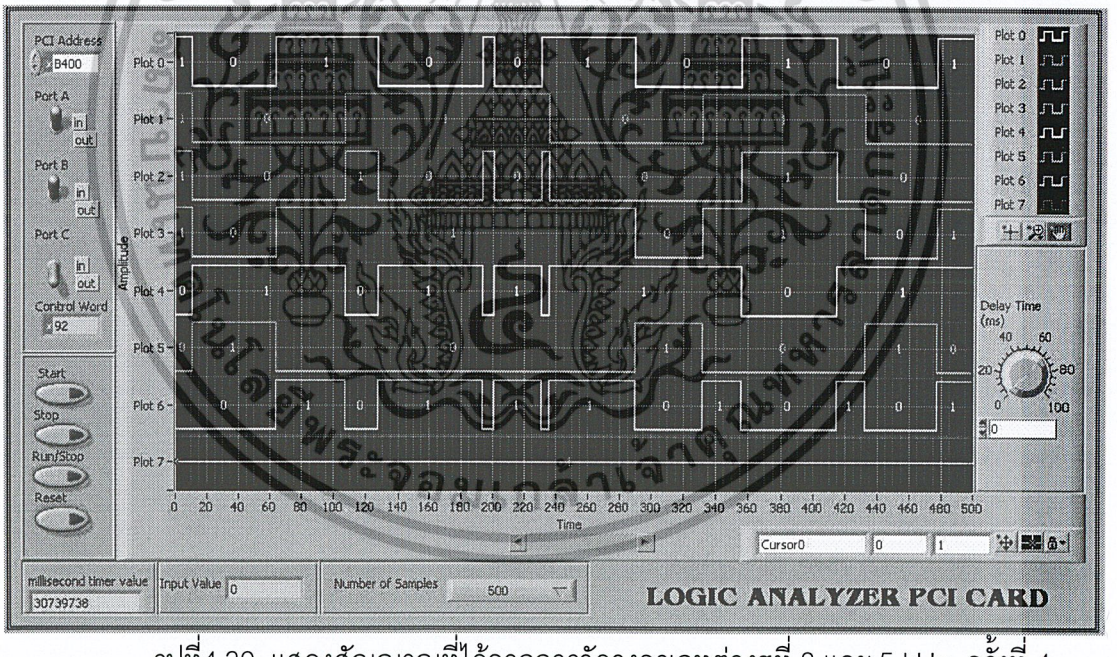


รูปที่4.37 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

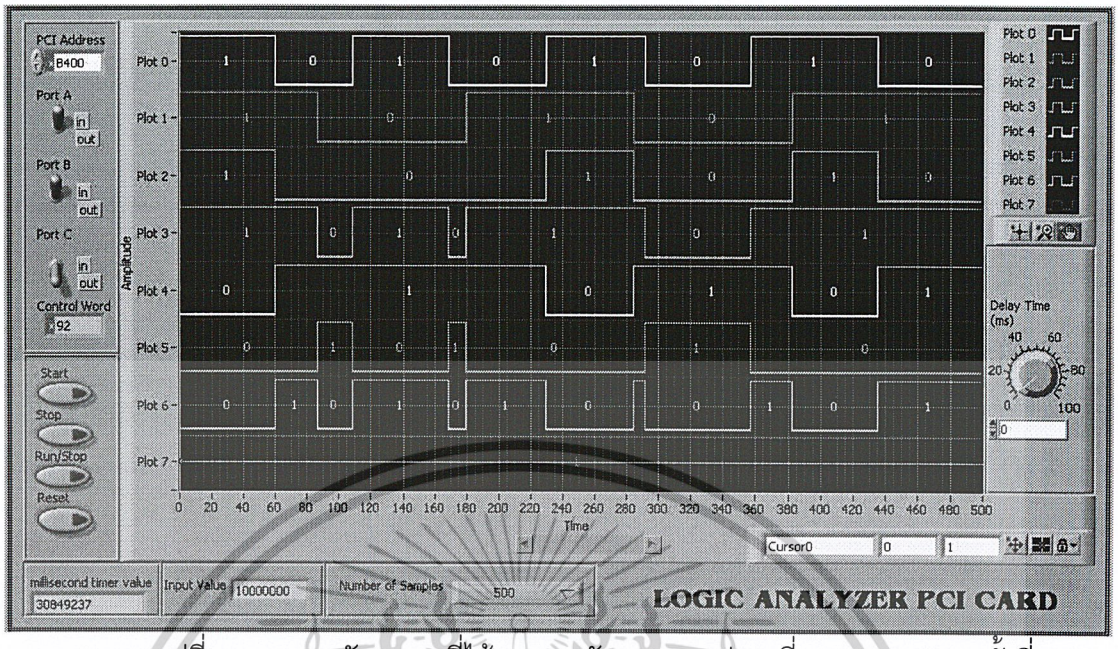


รูปที่4.38 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 3



รูปที่4.39 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



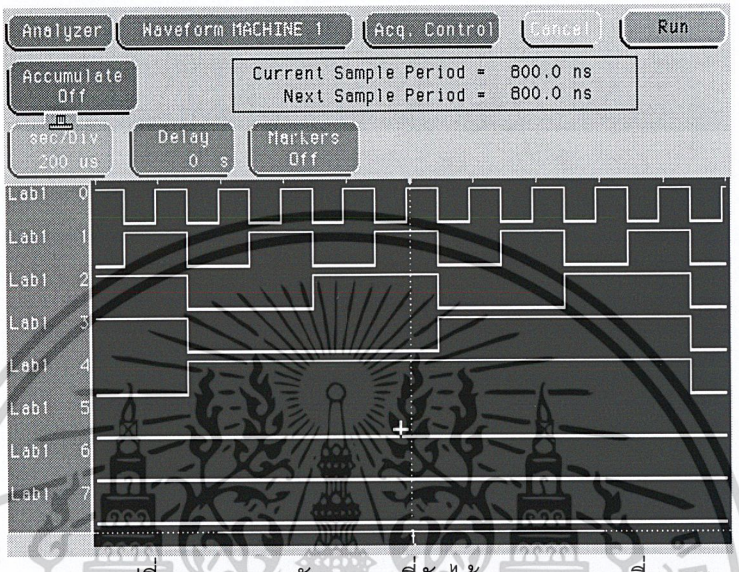
รูปที่4.40 แสดงสัญญาณที่ได้จากการวัดวงจรเกทต่างๆที่ 8 และ 5 kHz. ครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้ Logic Analyzer สำหรับการวิเคราะห์สัญญาณ

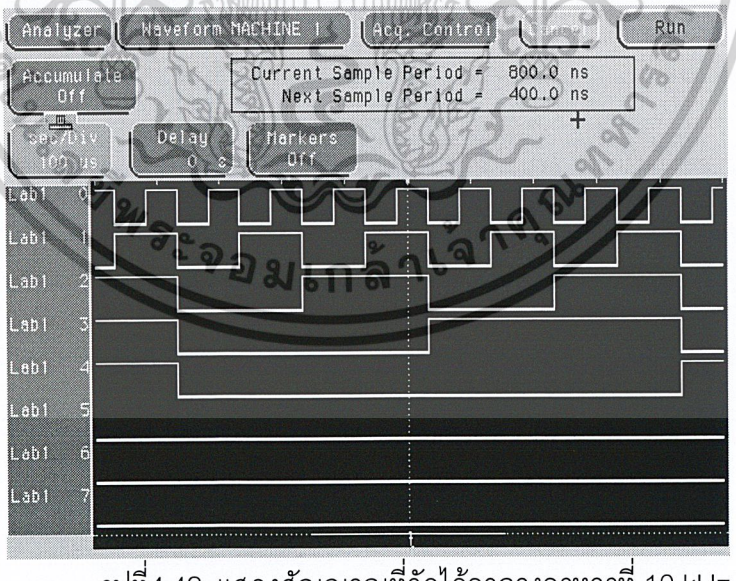
4.2.1 รับค่าผ่านวงจรหารหลายช่องสัญญาณ

- ที่ความถี่ 5 kHz.



รูปที่4.41 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 5 kHz

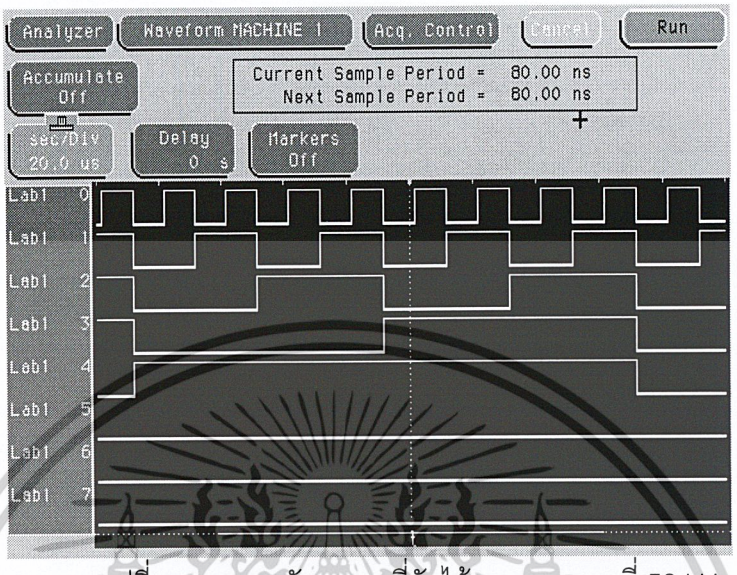
- ความถี่ 10 kHz.



รูปที่4.42 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ 50 kHz.

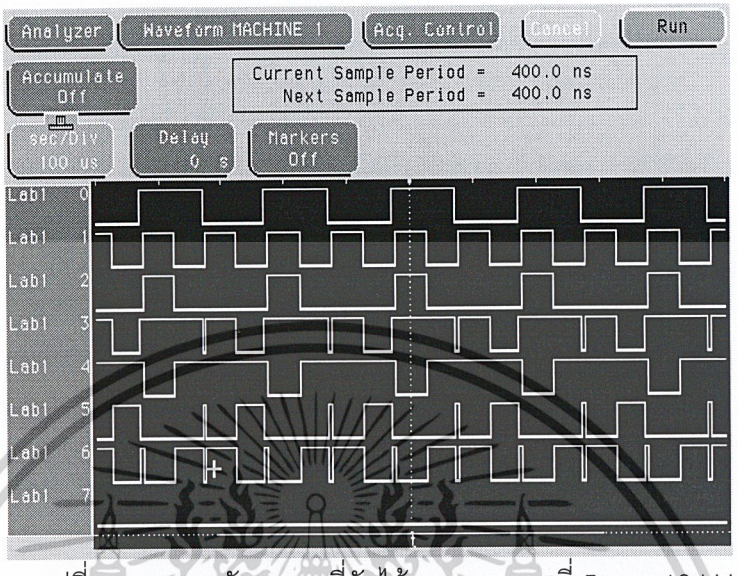


รูปที่4.43 แสดงสัญญาณที่วัดได้จากวงจรหารที่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

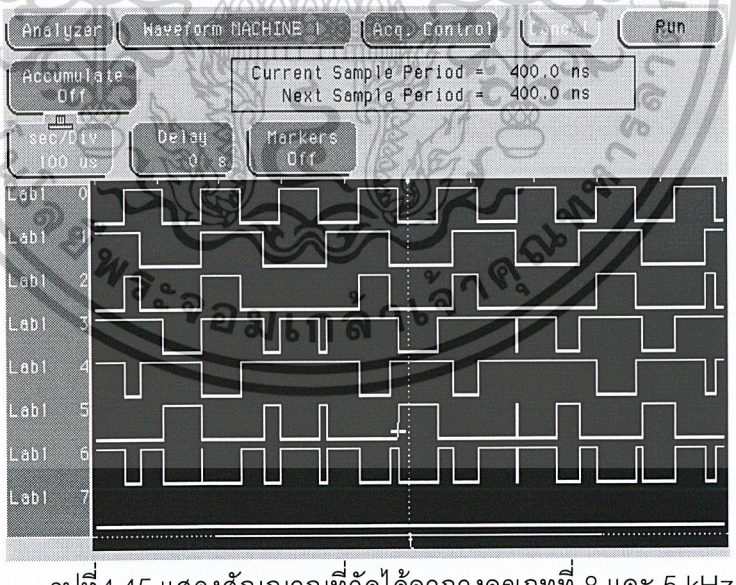
4.2.2 โปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่างๆ

- ความถี่ 5 kHz.และ10 kHz.



รูปที่4.44 แสดงสัญญาณที่วัดได้จากวงจรเกทที่ 5 และ 10 kHz

- ความถี่ 8 kHz. และ 5 kHz.



รูปที่4.45 แสดงสัญญาณที่วัดได้จากวงจรเกทที่ 8 และ 5 kHz

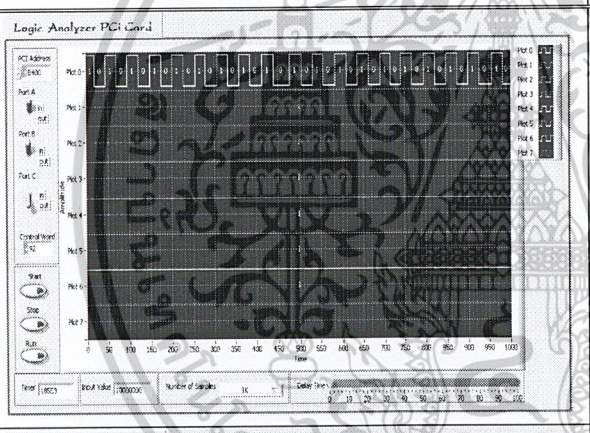
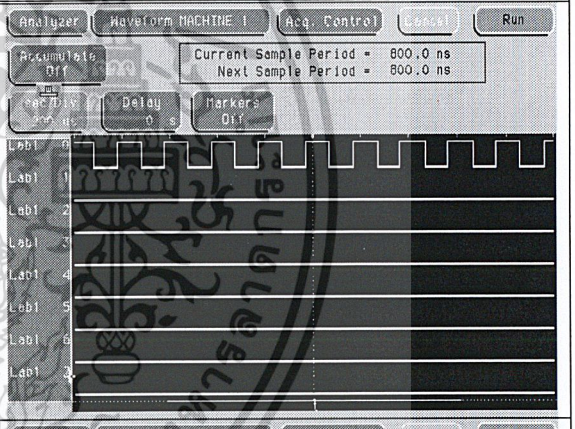
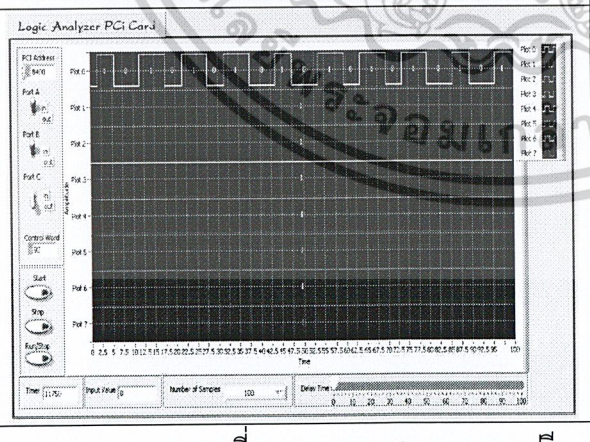
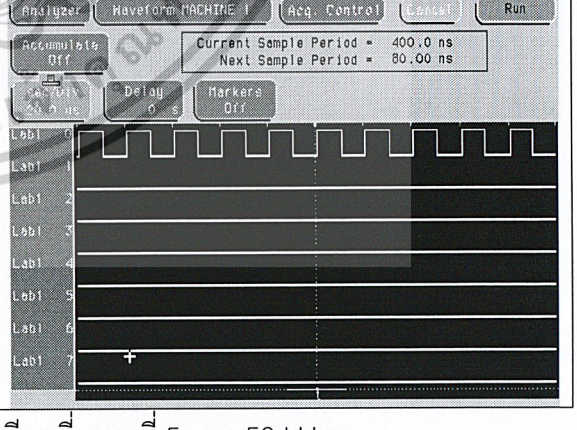
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

จากการทดลองเป็นการออกแบบโปรแกรมสำหรับให้มีการทำงานเหมือนเครื่องมือวิเคราะห์ทางตรรกะ ซึ่งเรียกกันว่า ลอจิกอนาไลซ์เซอร์ ทำการทดลองโดยใช้โปรแกรมมาเทียบกับการวัดโดยใช้เครื่องลอจิกอนาไลซ์เซอร์ จะสรุปได้เป็นขั้นตอนดังนี้

5.1 โปรแกรมรับค่าจาก Function Generator ที่ความถี่ต่าง ๆ

จะเห็นได้ว่าโปรแกรมจะสามารถวัดสัญญาณได้ใกล้เคียงกับเครื่องลอจิกกอนาไลซ์เซอร์ โดยใช้ความถี่ที่อยู่ในช่วง 5kHz. ถึง 50kHz.จะสามารถวัดได้ดีและค่อนข้างแม่นยำ

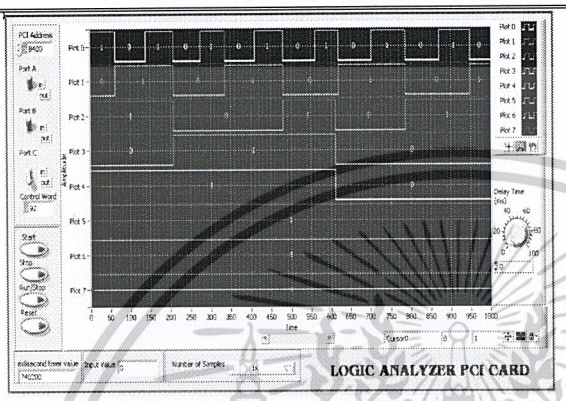
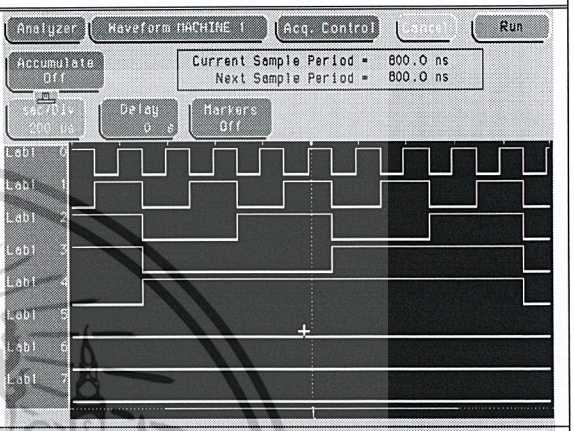
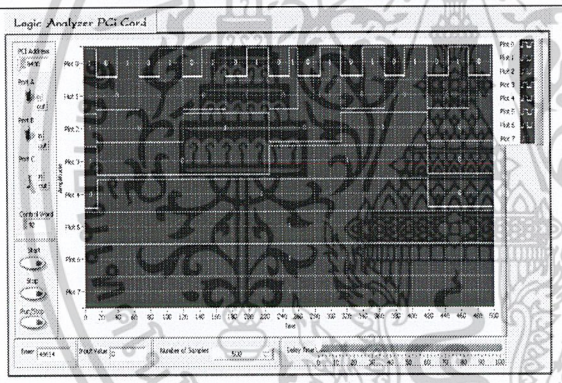
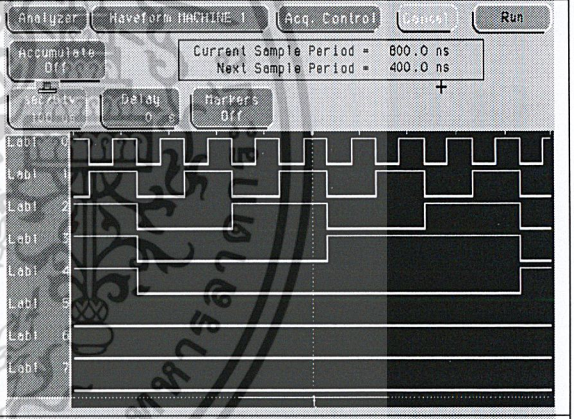
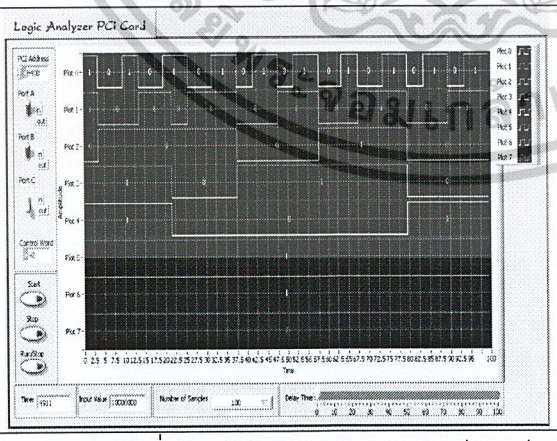
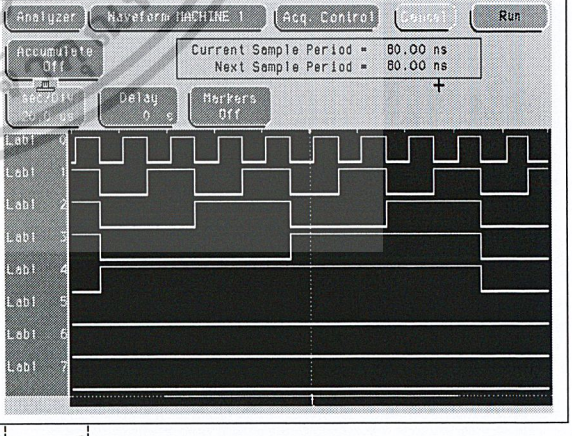
	Logic Analyzer PCI Card	Logic Analyzer
5k		
50k		

ตารางที่ 5.1 ตารางแสดงผลการเปรียบเทียบที่ความถี่ 5 และ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมรับค่าผ่านวงจรหารหลายช่องสัญญาณ

จากผลการทดลองที่ได้จะเห็นได้ว่า สัญญาณที่ได้ค่อนข้างตรงกับหลักการของวงจรหาร และเมื่อนำสัญญาณมาเทียบกับสัญญาณที่ได้จากเครื่องลอจิกอนาไลซ์เซอร์ จะได้ผลที่ใกล้เคียงกันมาก ซึ่งก็แสดงว่าโปรแกรมและวงจรหารนั้นมีความถูกต้อง

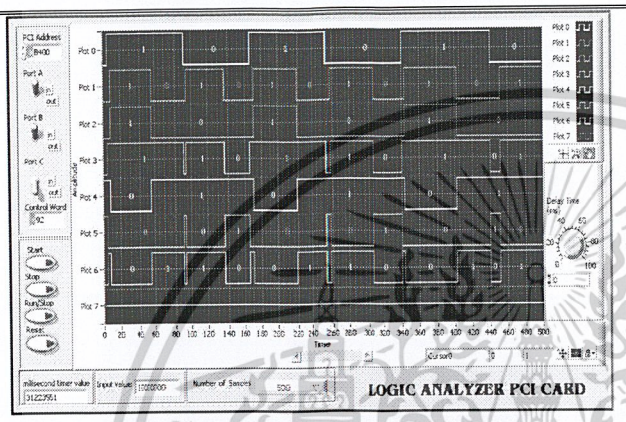
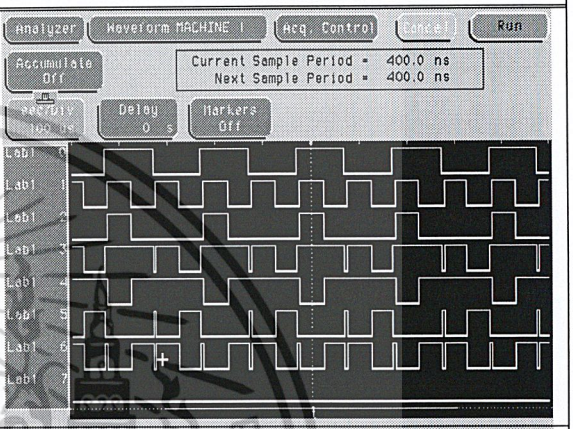
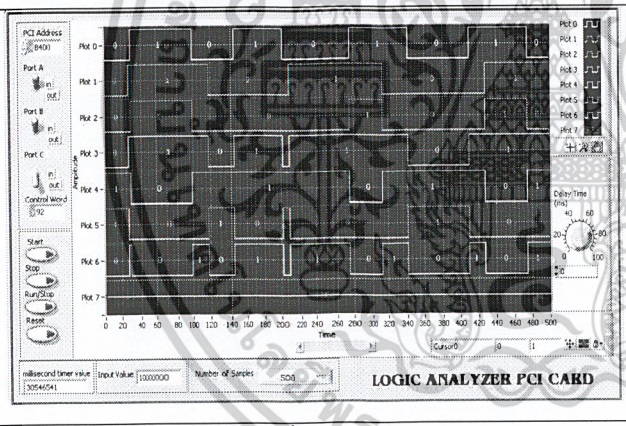
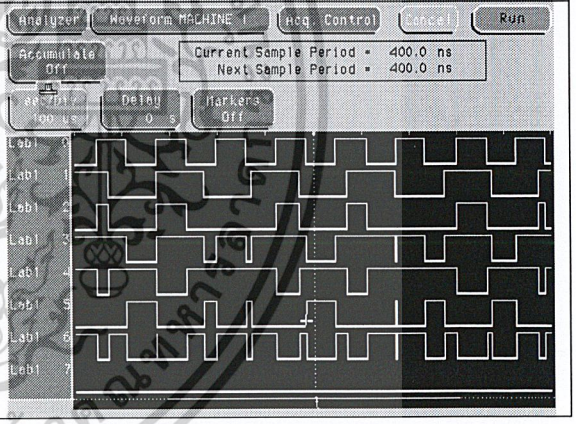
	Logic Analyzer PCI Card	Logic Analyzer
5k		
10k		
50k		

ตารางที่ 5.2 ตารางแสดงผลการเปรียบเทียบที่ความถี่ 5 ,10 และ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 โปรแกรมรับค่าผ่านวงจรทางดิจิตอลโดยใช้เกทต่าง ๆ

จากผลการทดลองที่ได้จะเห็นได้ว่า สัญญาณที่ได้ค่อนข้างตรงกับหลักการของวงจรเกท และเมื่อนำสัญญาณมาเทียบกับสัญญาณที่ได้จากเครื่องลอจิกอนาไลซ์เซอร์ จะได้ผลที่ใกล้เคียงกันมาก ซึ่งก็แสดงว่าโปรแกรมและวงจรเกทนั้นมีความถูกต้อง

	Logic Analyzer PCI Card	Logic Analyzer
5k และ 10k		
8k และ 5k		

ตารางที่ 5.3 ตารางแสดงผลการเปรียบเทียบจากการวัดวงจรเกท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 สรุปโดยรวม

จากผลการทดลองที่ได้จะเห็นได้ว่า สัญญาณที่ได้จากลอจิกอนาไลซ์เซอร์ พีซีไอการ์ด เมื่อนำมาเทียบกับสัญญาณที่ได้จากเครื่องลอจิกอนาไลซ์เซอร์ จะได้ผลที่ใกล้เคียงกันมาก ซึ่งก็แสดงว่าโปรแกรมและวงจรเกทนั้นมีความถูกต้องในช่วงความถี่ 5 ถึง 50 กิโลเฮิรตซ์ แต่ก็อาจยังมีบางช่วงสัญญาณที่ไม่สมบูรณ์ เนื่องมาจากการสุ่มสัญญาณด้วยตัวโปรแกรม

5.5 ข้อเสนอแนะและแนวทางการพัฒนา

เนื่องด้วยมาจากข้อจำกัดด้านซอฟท์แวร์ของการเส่งผ่านข้อมูลระหว่าง คอมพิวเตอร์พีซี และระบบที่ต้องการวัดสัญญาณผ่าน พีซีไอการ์ด จึงต้องมีบอร์ดไมโครคอนโทรเลอร์ขึ้นมาเพื่อเป็นหน่วยความจำสำหรับพักข้อมูล แต่ก็ยังมีวิธีที่น่าจะสามารถพัฒนาให้สามารถวัดสัญญาณในช่วงความถี่กว้างๆ และแม่นยำได้

- พัฒนาด้านซอฟท์แวร์ ไดร์เวอร์ ของพีซีไอการ์ด ให้สามารถนำข้อมูลที่ได้ไปเก็บในหน่วยความจำภายในโดยตรง (Direct Memory Access:DMA)
- พัฒนาด้านฮาร์ดแวร์ ที่เชื่อมต่อกับพีซีไอ ด้วยการเพิ่มวงจรหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

สัลยุทธ์ สว่างวรรณ, 2546.**สถาปัตยกรรมคอมพิวเตอร์**.ครั้งที่1.กรุงเทพฯ: เพียร์สัน เอ็ดดูเคชั่น อินโดไชน่า

เอกชัย หวายนำ, หิรัญ เฉียบแหลม.2543. **ลอจิกอนาไลซ์เซอร์**. ปริญญานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม สาขาวิศวกรรมการวัดคุม. บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ดร. สุเจตน์ จันทรังษ์, 2543 . **การออกแบบระบบดิจิตอล**. กรุงเทพฯ : แมคกรอ-ฮิล อินเตอร์เนชั่นแนล เอ็นเตอร์ไพร์ส,อิงค์.

รศ.วิชิต ศิริโชติ, 2543. **การเขียนโปรแกรมภาษาแอสแซมบลี** 89C52 8kB MicroController. กรุงเทพฯ : คณะวิทยาศาสตร์ ภาควิชาฟิสิกส์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บริษัท อีทีที จำกัด. ET-PCI8255 BUS INTERFACE. กรุงเทพฯ : บริษัท อีทีที จำกัด

เจริญ เพชรมุณี, 2547.**เรียนลัด** Labview.ครั้งที่1.กรุงเทพฯ: ซีเอ็ดยูเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก ซอร์ทโคดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
$mod52

dseg        at   0030h
count:      ds   2
sh:         ds   2
FLAG:       DS   1

cseg        at   0000h
            jmp  main
            org  0003h
            jmp  interrupt
            org  0100h
INIT_SER:
            MOV  SCON,#50H
            ORL  TMOD,#20H
            MOV  TH1,#0FDH
            SETB TR1
            SETB TI
            RET

main:
            CALL INIT_SER
            mov  dph,#00h
            mov  dpl,#00h
            mov  08eh,#00001100b
            mov  08fh,#00000001b
            mov  count,#1000
            mov  sh,#1000
            MOV  R0,#00
            setb ea
            setb ex0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
32          setb    IT0
33          SETB    P4.0
34
35    loop: setb    p4.1
36          mov     a,p0
37          clr     p4.1
38          movx    @dptr,a
39          inc     dptr
40          djnz    count,loop
41
42          clr     p4.0
43          mov     DPL,#00h
44          MOV     DPH,#00H
45
46    LOOP2:        CJNE    R0,#01,LOOP2
47          MOV     R0,#00
48          MOV     DPL,#00H
49          MOV     DPH,#00H
50          MOV     COUNT,#1000
51          MOV     SH,#1000
52          SETB    P4.0
53          JMP     LOOP
54
55    interrupt:
56          MOVX    A,@DPTR
57          call    cout
58          MOV     P2,A
59          INC     DPTR
60          DJNZ    SH,FIN
61          MOV     R0,#01
62    FIN:        RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
63  cout:   jnb  ti, cout
64          clr  ti          ;clr ti before the mov to sbuf!
65          mov sbuf, a
66          ret
67
68  end
```

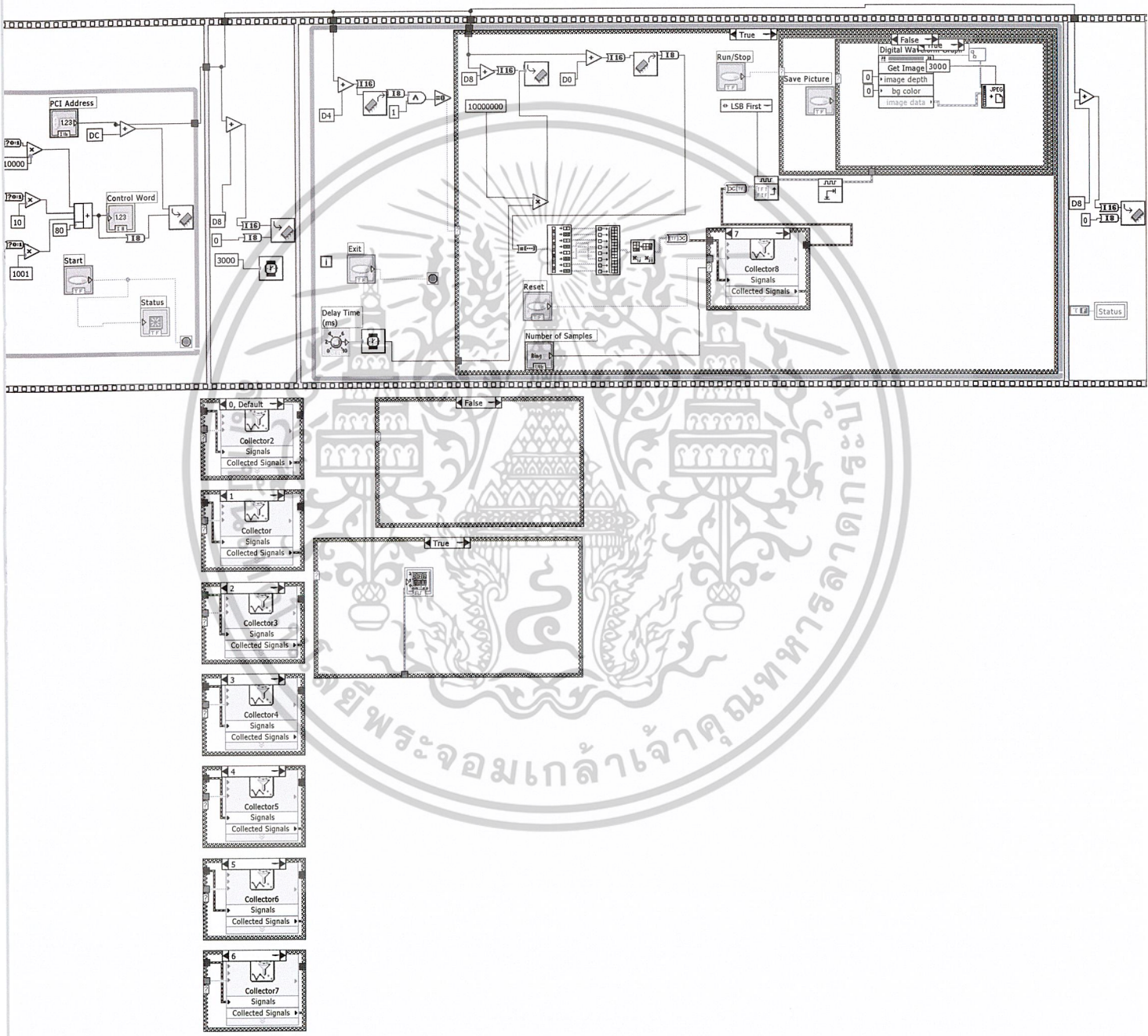

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.vi
ocuments and Settings\Administrator\Desktop\logic analyzer\Program\logic.vi
modified on 30/3/2548 at 17:50
ed on 31/3/2548 at 16:41

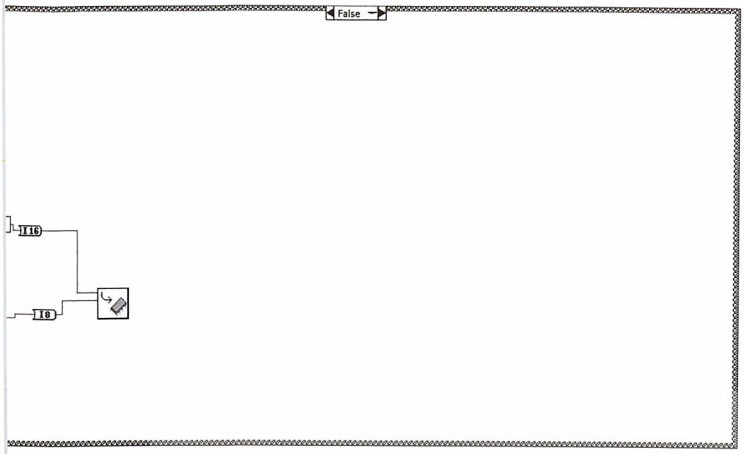
nector Pane

logic.vi

k Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ess VI Configuration Information

Collector2
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 100

Collector3
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 1000

Collector4
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Collector5
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 5000

Collector6
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 10000

Collector7
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 50000

Collector8
Collector
Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

This Express VI is configured as follows:

Number of Points to Collect: 100000

Convert to Dynamic Data
Convert to Dynamic Data

Convert from Dynamic Data
Convert from Dynamic Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Collector

Collector

Collects input signals and returns the most recent sample of collected data. This Express VI discards the oldest data point and adds the newest data point to the collected sample when the Express VI reaches the maximum number of data points.

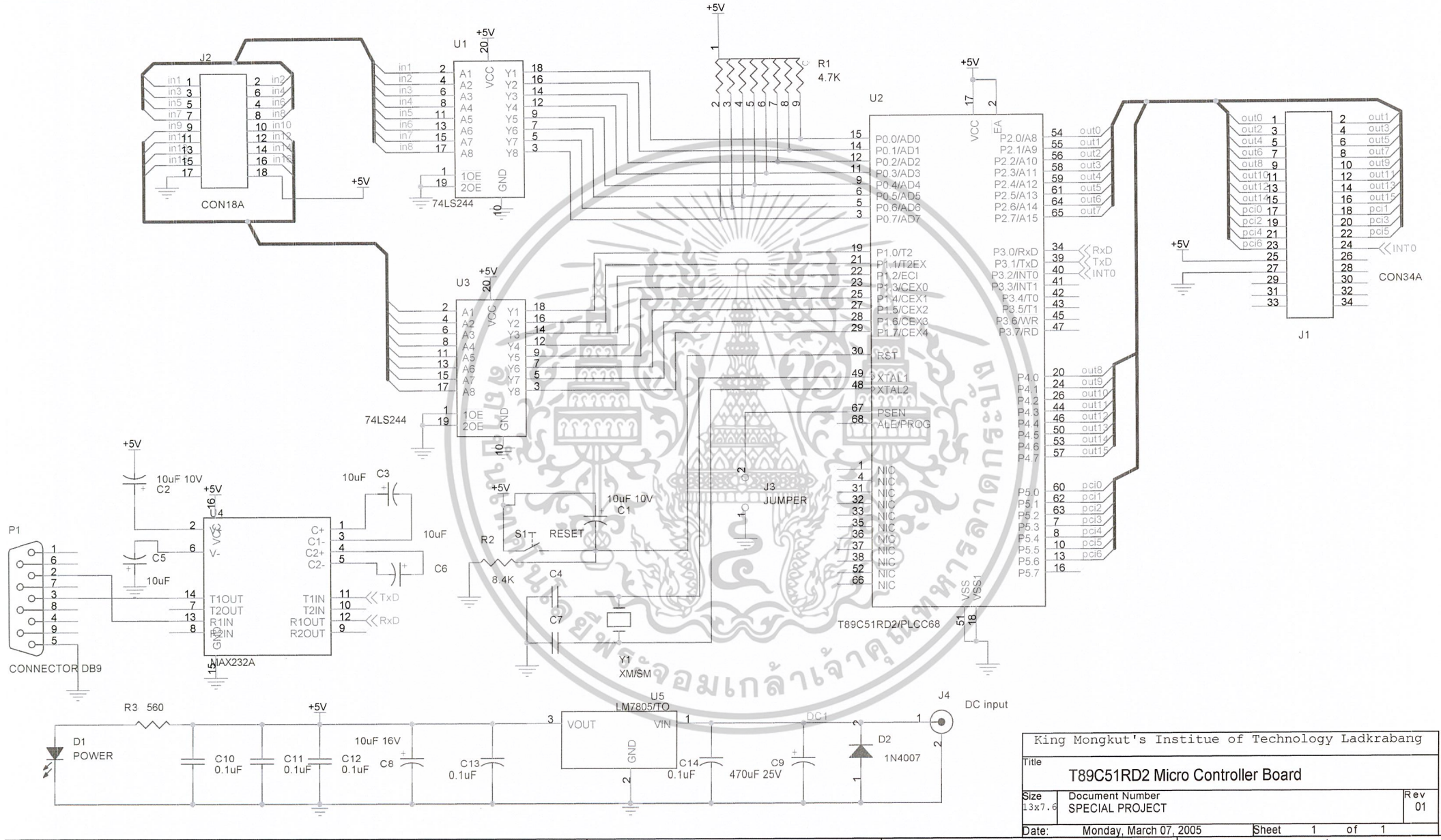
This Express VI is configured as follows:

Number of Points to Collect: 500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวค ข รายละเอียดอุปกรณ์ที่ใช้ในโครงงานพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


J2
CON18A
U1
74LS244
U3
74LS244
R1
4.7K
U2
T89C51RD2/PLCC68
J1
CON34A
J3
JUMPER
RESET
S1
R2
8.4K
C1
C2
C3
C4
C5
C6
C7
Y1
XM/SM
P1
CONNECTOR DB9
U4
MAX232A
U5
LM7805/TO
J4
DC input
D1
POWER
D2
1N4007
R3 560
C8
C9
470uF 25V
C10
C11
C12
C13
C14
10uF 16V
0.1uF
10uF 10V
10uF
+5V
King Mongkut's Institue of Technology Ladkrabang
T89C51RD2 Micro Controller Board
SPECIAL PROJECT
01
Monday, March 07, 2005
Sheet 1 of 1

Features

- 80C52 Compatible
  - 8051 Pin and Instruction Compatible
  - Four 8-bit I/O Ports (or 6 in 64/68 Pins Packages)
  - Three 16-bit Timer/Counters
  - 256 bytes Scratch Pad RAM
  - 7 Interrupt Sources With 4 Priority Levels
- ISP (In-System Programming) Using Standard V_{CC} Power Supply
- Boot Flash Contains Low Level Flash Programming Routines and a Default Serial Loader
- High-Speed Architecture
  - 40 MHz in Standard Mode
  - 20 MHz in X2 Mode (6 Clocks/Machine Cycle)
- 64K bytes On-chip Flash Program/Data Memory
  - Byte and Page (128 bytes) Erase and Write
  - 100K Write Cycles
- On-chip 1024 Bytes Expanded RAM (XRAM)
  - Software Selectable Size (0, 256, 512, 768, 1024 bytes)
  - 768 Bytes Selected at Reset for T87C51RD2 Compatibility
- Dual Data Pointer
  - Variable Length MOVX for Slow RAM/Peripherals
- Improved X2 Mode with Independant Selection for CPU and Each Peripheral
- 2K bytes EEPROM Block for Data Storage
  - 100K Write Cycle
- Programmable Counter Array with
  - High Speed Output
  - Compare/Capture
  - Pulse Width Modulator
    - Watchdog Timer Capabilities
    - Asynchronous Port Reset
    - Full-duplex Enhanced UART
    - Low EMI (Inhibit ALE)
  - Hardware Watchdog Timer (One-time Enabled with Reset-out)
- Power Control Modes:
  - Idle Mode
  - Power-down Mode
- Power Supply:
  - M version: Commercial and Industrial
    - 4.5V to 5.5V: 40 MHz (X1 Mode), 20 MHz (X2 Mode)
    - 3V to 5.5V: 33 MHz (X1 Mode), 16 MHz (X2 Mode)
  - L version: Commercial and industrial
    - 2.7V to 3.6V: 25 MHz (X1 Mode), 12 MHz (X2 Mode)
- Temperature Ranges: Commercial (0 to +70°C) and Industrial (-40 to +85°C)
- Packages: PDIL40, PLCC44, VQFP44, PLCC68, VQFP64

0 to 40 MHz Flash Programmable 8-bit Microcontroller

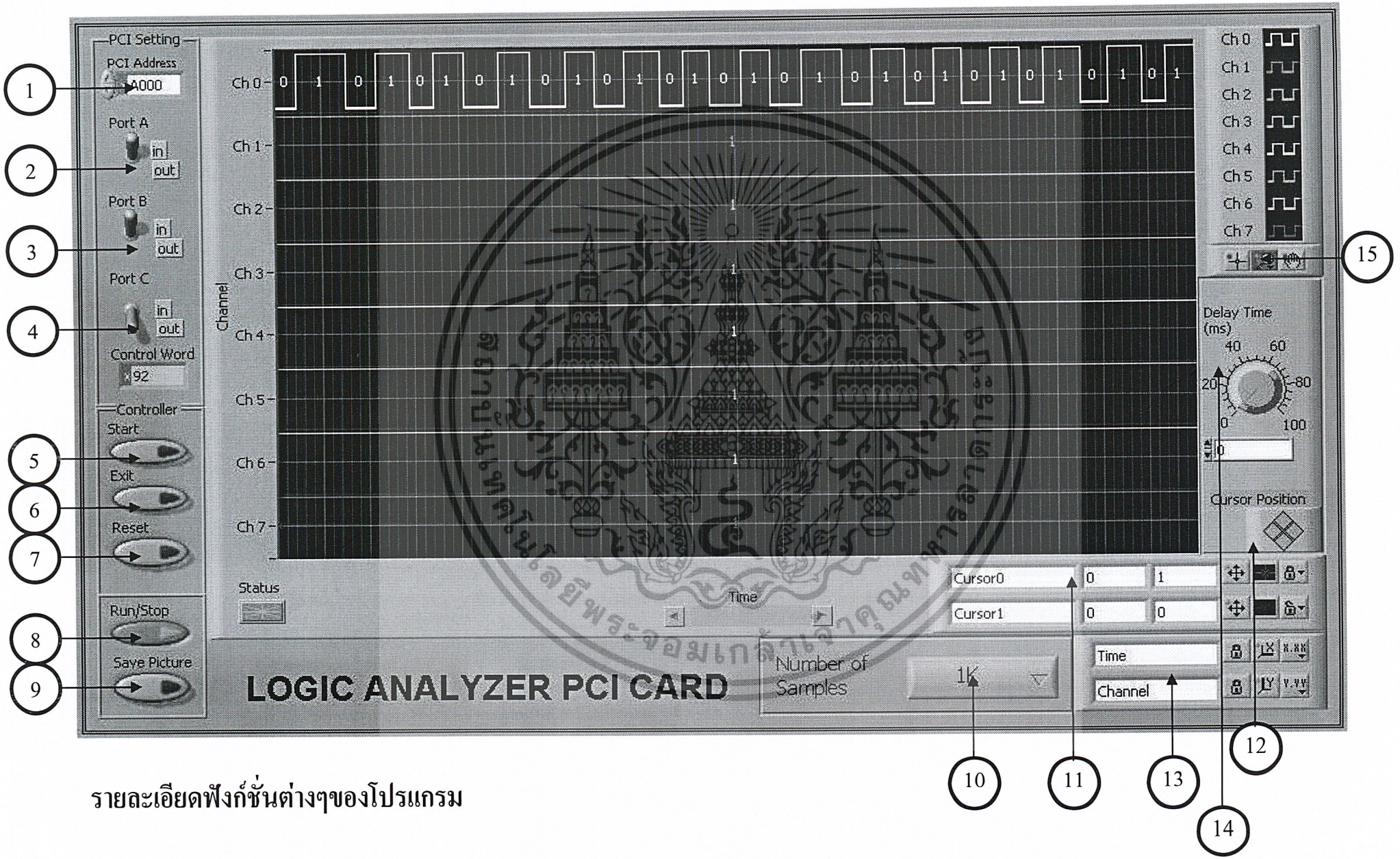
T89C51RD2

Rev. 4243G–8051–05/03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

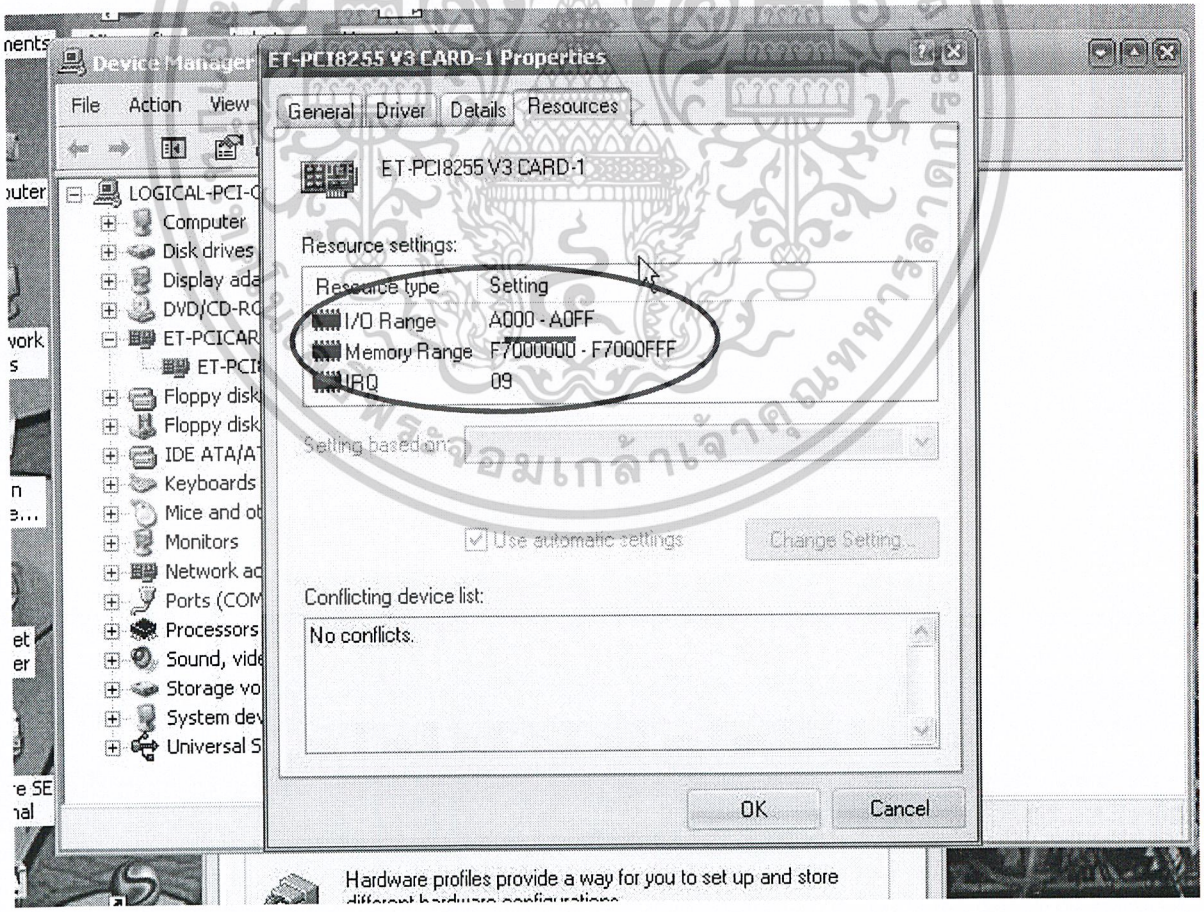
ภาคผนวก ค คู่มือการใช้โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รายละเอียดฟังก์ชั่นต่างๆของโปรแกรม

หมายเลข	รายละเอียด
1	ตั้งค่าของตำแหน่ง address ของ PCI Card ดูจากรูปด้านล่าง
2,3,4	ตั้งค่าการใช้งานผ่านชิพ 8255 ซึ่งมีทั้งหมด 3 พอร์ต คือ A , B และ C โดยเลือก In หรือ Out
5	ปุ่ม Start เพื่อเริ่มโปรแกรมหลังจากการตั้งค่าการทำงานของ PCI Card
6	ปุ่ม Exit เพื่อ ออกจากโปรแกรม
7	ปุ่ม Reset เพื่อเคลียร์หน้าจอระหว่างการทำงาน
8	ปุ่ม Run/Stop โดยที่ Run แสดงสัญญาณที่วัดได้และถ้าต้องการวิเคราะห์สัญญาณ ให้เลือก Stop
9	ปุ่ม Save digital signal เป็นไฟล์รูปภาพ หลังจากกดปุ่มที่ 7
10	ฟังก์ชันเลือกเก็บจำนวนตัวอย่างเพื่อแสดงออกทางหน้าจอ
11	ฟังก์ชันแสดง cursor เพื่อทำการวัดความต่างของแต่ละตำแหน่งcursor
12	ปุ่มเคลื่อนตำแหน่ง cursor
13	ฟังก์ชันเลือกแกนของ cursor ในแกน X(Timer) หรือ แกนY(Channel)
14	ปุ่มปรับ Delay Time เพื่อช่วงปรับเวลาในการรับค่า
15	ฟังก์ชันในการวิเคราะห์สัญญาณ เช่น ขยายรูปสัญญาณ เลื่อนรูปสัญญาณ ฯลฯ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้