ปีการศึกษา 2531

ลอจิก อนาไลเซอร์

(Logic Analyser)

โดย

นาย วรเทพ บุญธรรมจิต 28.1193

นางสาว วาสนา พงศานรากุล 28.1210

อาจารย์ที่ปรึกษา

อาจารย์ วันชัย ริ้วรุจา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023170

-9 ส.ค. 2532

ปริญญานิพนธ์ปีการศึกษา 2531

ภาควิชา ระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ลอจิก อนาไลเซอร์

ผู้จัดทำ

1.นาย วรเทพ บุญธรรมจิต 28.1193

2.นางสาว วาสนา พงศานรากุล 28.1210

...อาจารย์ที่ปรึกษา

(อาจารย์ วันชัย ริ้วรุจา)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023170

ลอจิก อนาไลเซอร์

วรเทพ บุญธรรมจิต
วาสนา พงศานรากุล
วันชัย ริ้วรุจา อาจารย์ที่ปรึกษา

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้เป็นการพัฒนาการออกแบบวงจรลอจิกอนาไลเซอร์เพื่อใช้วิเคราะห์สัญญาณการทำงานระบบดิจิตอลที่มีความถี่สูงถึง 20 MHz 8 ช่องโดยการสุ่ม (sampling) สัญญาณที่ต้องการวิเคราะห์แล้วนำค่าไปเก็บไว้ในหน่วยความจำความเร็วสูง (High Speed Ram) และนำไปแสดงผลออกทางจอภาพของไมโครคอมพิวเตอร์ โดยมีไมโครโปรเซสเซอร์ 8031 ชิงเกิลชิพ (Microprocessor 8031 Single Chip) เป็นตัวรับข้อมูลจากหน่วยความจำความเร็วสูงแล้วส่งข้อมูลไปยังไมโครคอมพิวเตอร์ และทำหน้าที่ควบคุมการทำงานของหน่วยความจำความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Analyser

Worathep Boonthamchit

Wassana Pongsanarakul

Wanchai Rewrucha Advisor

1988

Abstract

This thesis is a development and design of Logic Analyser , use for analize digital signal. It can sampling high frequency signal and keep these value into high speed RAM for display on microcomuter's monitor. By using MCS.51 , the single chip microcontroller , high speed RAM can transfer data to microcomputer. In the other hand MCS.51 is a buffer between high speed RAM and microcomputer. Our Logic Analyser can analize very high frequency signal about 20 MHz. correctly. Especially , it can simultaneously measure 8 signals that very useful for analizing system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันการพัฒนาทางเทคโนโลยีเป็นไปอย่างรวดเร็ว อุปกรณ์มีขนาดเล็กลงแต่มีประสิทธิภาพสูงขึ้นเครื่องมือพื้นฐานที่สามารถใช้วัดสัญญาณดิจิตอลได้ก็คือ ออสซิลโลสโคป (Oscillo Scope) แต่ออสซิลโลสโคปสามารถใช้วัดสัญญาณได้เพียงชั่วขณะเท่านั้น เมื่อเวลาผ่านไปก็ไม่สามารถย้อนกลับมาดูสัญญาณเก่าได้ จึงไม่สามารถที่จะวิเคราะห์สัญญาณระบบดิจิตอลได้ อีกทั้งออสซิลโลสโคปไม่สามารถวัดสัญญาณได้พร้อมๆกันหลายช่อง ซึ่งในบางครั้งการวิเคราะห์สัญญาณดิจิตอลจะต้องศึกษาและพิจารณาสัญญาณพร้อมๆกันหลายสัญญาณ ทำให้ไม่สามารถใช้ออสซิลโลสโคปช่วยในการวิเคราะห์ระบบดิจิตอลได้อย่างเต็มที่

เครื่องมือที่สามารถวัดสัญญาณดิจิตอลได้ และสามารถนำสัญญาณที่เกิดขึ้นแล้วมาแสดงผลได้อีก โดยแสดงผลพร้อมๆกันได้หลายช่อง ก็คือ ลอจิก อนาไลเซอร์ ซึ่งมีขายอยู่ในปัจจุบันแต่มีราคาแพงมาก ดังนั้นจึงได้พัฒนาโครงงานนี้ขึ้นมามีชื่อว่า " ลอจิก อนาไลเซอร์ " ซึ่งเป็นเครื่องมือวัดสัญญาณดิจิตอลที่สามารถวัดสัญญาณดิจิตอลได้พร้อมๆกันถึง 8 สัญญาณหรือ 8 ช่อง และมีหน่วยความจำที่ใช้สำหรับเก็บข้อมูล สามารถที่จะนำข้อมูลมาแสดงผลได้อีก หลังจากที่สัญญาณเกิดขึ้นแล้วเป็นเวลานานๆ นอกจากนี้ยังมีคุณสมบัติที่สำคัญคือ สามารถวัดสัญญาณที่มีความถี่สูงถึง 20 MHz ได้อย่างถูกต้องและมีราคาถูกอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 การสุ่มและข้อมูล

การสุ่มค่าสัญญาณให้ได้ค่าที่ถูกต้องหรือใกล้เคียงกับสัญญาณที่แท้จริงนั้น ความถี่ที่ใช้ในการสุ่มสัญญาณจะต้องเป็นอย่างน้อยสองเท่าของความถี่ของสัญญาณที่ถูกสุ่ม ถ้าความถี่ที่ใช้ในการสุ่มสัญญาณมีค่ามากกว่าความถี่ของสัญญาณที่สุ่มมากเท่าไร ความถูกต้องเหมือนสัญญาณจริงก็จะยิ่งมีมากขึ้น แต่ตัวที่จะนำมาใช้กำหนดความถี่ที่ใช้ในการสุ่มก็คือ ความเร็วในการทำงานของอุปกรณ์ส่วนเก็บข้อมูลโดยเฉพาะหน่วยความจำ หน่วยความจำทั่วๆไป จะมีความเร็วในการอ่านหรือเขียนข้อมูลประมาณ 6 MHz หรือแต่ละรอบการทำงานจะใช้เวลาประมาณ 150 ns ซึ่งจะเห็นได้ว่าจากทฤษฎีการสุ่มค่าสัญญาณจะมีการวัดค่าสัญญาณได้สูงสุดเพียง 3 MHz เท่านั้น ดังนั้นเราจึงนำหน่วยความจำธรรมดานี้จำนวน 8 ตัวมาต่อกัน โดยให้แต่ละตัวผลัดกันทำงาน เพราะฉะนั้นที่ความถี่ไม่เกิน 24 MHz หน่วยความจำแต่ละตัวจะทำงานได้ทันก่อนที่การทำงานครั้งต่อไปจะมาถึง ถ้าเรามองหน่วยความจำทั้ง 8 ตัวนี้ เป็นหน่วยความจำเพียงตัวเดียว จะเห็นว่าหน่วยความจำตัวนี้ทำงานได้เร็วขึ้นถึง 8 เท่า เราจึงเรียกว่า " หน่วยความจำความเร็วสูง "

การสร้างสัญญาณนาฬิกา (clock) ที่มีความถี่แน่นอนเพื่อทำการสุ่มค่าสัญญาณจะใช้วงจรออสซิลเลเตอร์ (Oscillator) สร้างสัญญาณนาฬิกาผ่านให้กับวงจรนับ(counter) เมื่อสัญญาณนาฬิกาผ่านเข้าวงจรนับ 1 ลูก สัญญาณวงจรนับก็จะนับเพิ่มขึ้นหนึ่งหมายถึง ทำการสุ่มค่าสัญญาณครั้งหนึ่ง หรือเขียนข้อมูลลงในหน่วยความจำค่าหนึ่ง วงจรนับจะนับจาก 0000 ไปจนถึงค่าสุดท้ายแล้วจะวนนับศูนย์ใหม่อีกครั้งหนึ่ง ซึ่งหมายความว่า ได้เขียนข้อมูลลงในหน่วยความจำเต็มแล้ว

2.2 การตรวจสอบสภาวะของสัญญาณ

จะต้องมีวงจรหนึ่งทำการตรวจสอบสภาวะของสัญญาณ แล้วไปควบคุมการทำงานของ ลอจิก อนาไลเซอร์ โดยสภาวะที่จะต้องตรวจสอบก็คือ

2.2.1 สัญญาณทริก (Trig) จากภายนอก ถ้าไม่มีสัญญาณทริกจากภายนอกมาทริก ลอจิก อนาไลเซอร์ จะยังไม่เริ่มสุ่มค่าสัญญาณ แต่ถ้ามีสัญญาณทริกจากภายนอกแล้ว วงจรนับก็จะเริ่มนับหรือเริ่มสุ่มค่าสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ตรวจสอบว่าหน่วยความจำเต็มหรือยัง การตรวจสอบสภาวะนี้มีความสำคัญมาก ถ้าหากว่าหน่วยความจำถูกเขียนข้อมูลเต็มแล้ว แต่ยังมีการสุ่มค่าสัญญาณอีก จะทำให้ข้อมูลใหม่ไปทับข้อมูลเก่า ดังนั้นจึงต้องมีการตรวจสอบสภาวะนี้ ถ้าข้อมูลเต็มแล้วก็ให้วงจรนับหยุดนับหรือหยุดสุ่มค่าสัญญาณ

2.3 การจัดการข้อมูล

เมื่อมีการสุ่มค่าสัญญาณ และเก็บค่าเหล่านั้นลงในหน่วยความจำจนเต็มแล้ว ก็จะเป็นการจัดการนำเอาข้อมูลที่มีอยู่ในหน่วยความจำ มาแสดงผลโดยมีไมโครโปรเซสเซอร์ 8031 (MCS 51) เป็นตัวจัดการข้อมูล

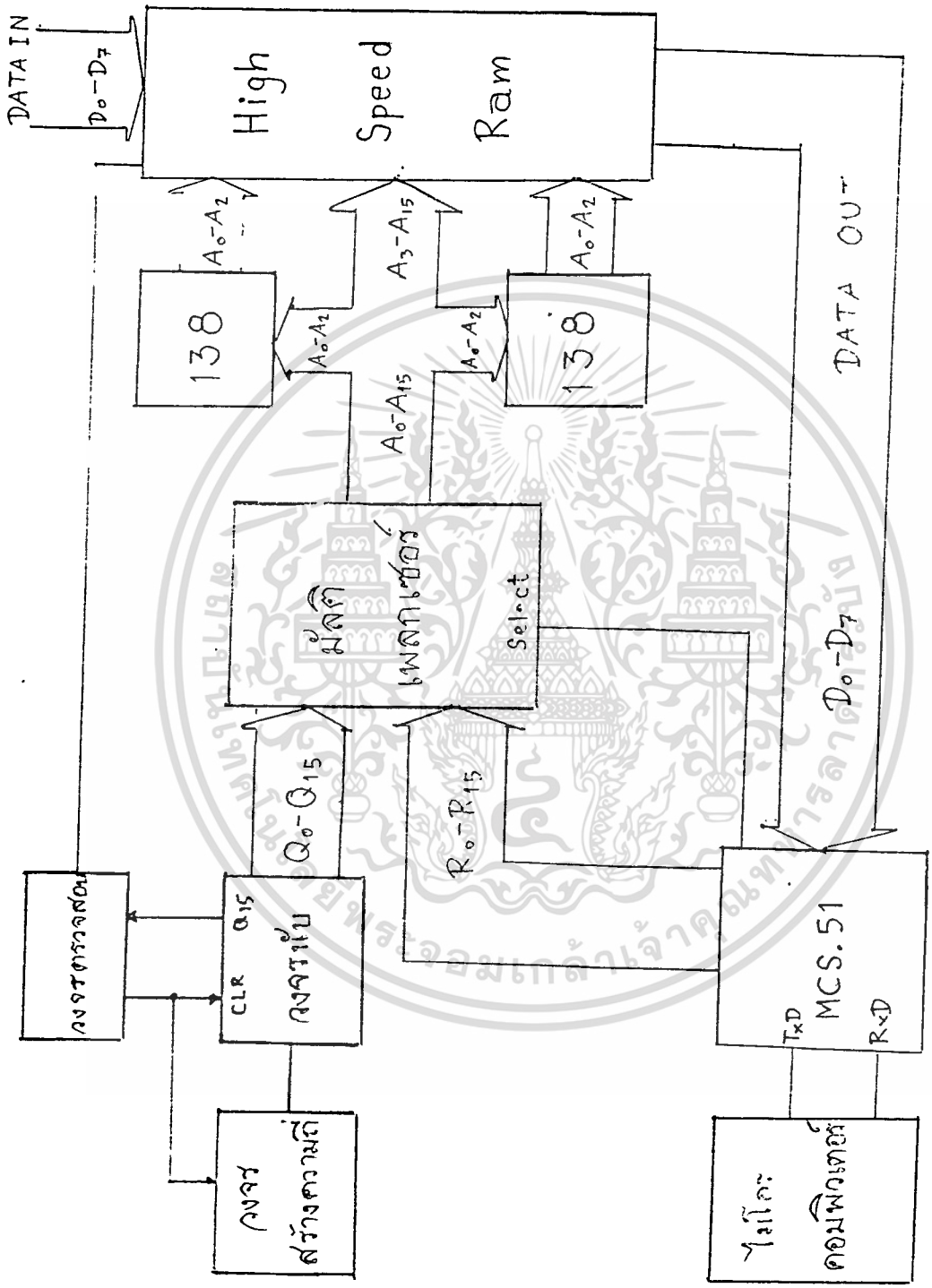
เมื่อต้องการแสดงผล ไมโครคอมพิวเตอร์จะส่งสัญญาณมาบอกให้ไมโครโปรเซสเซอร์ MCS 51 ก็จะอ่านข้อมูลจากหน่วยความจำความเร็วสูงที่ละตำแหน่ง แล้วส่งไปให้ไมโครคอมพิวเตอร์ โดยจะอ่านและส่งข้อมูลจากตำแหน่งศูนย์จนถึงตำแหน่งสุดท้าย การอ่านข้อมูลของ MCS 51 จากหน่วยความจำความเร็วสูงจะรับข้อมูลแบบขนาน และส่งไปให้ไมโครคอมพิวเตอร์แบบอนุกรม

MCS 51 จะเป็นตัวควบคุมการทำงานของหน่วยความจำความเร็วสูงโดยจะอ่านคำสั่งการทำงานจากอีพรอม (EPROM) แล้วทำงานตามคำสั่ง ถ้าต้องการจะเปลี่ยนแปลงหรือแก้ไขขั้นตอนการทำงานก็สามารถทำได้โดยการเปลี่ยนแปลงแก้ไขคำสั่งในอีพรอม

2.4 การแสดงผล

ในส่วนนี้จะเป็นซอฟท์แวร์ (SOFTWARE) เมื่อไมโครคอมพิวเตอร์รับข้อมูลจาก MCS 51 จนครบแล้วก็จะทำการประมวลข้อมูล เนื่องจากข้อมูลที่รับมามี 8 บิท แต่ละบิทคือค่าของสัญญาณแต่ละช่อง ดังนั้นจึงต้องนำค่าในบิทตรงกันมาต่อกัน (บิทที่หนึ่งต่อกับบิทที่หนึ่ง บิทที่สองต่อกับบิทที่สอง ไปจนถึงบิทที่แปดต่อกับบิทที่แปด) โดยนำข้อมูลจากตำแหน่งต่างๆมาต่อเรียงกันจากตำแหน่งต่ำสุดไปถึงตำแหน่งสูงสุด (จาก 0000 ถึง FFFF) ซึ่งก็จะได้ผลลัพธ์ออกเป็นเส้นไดอะแกรม (DIAGRAM) แสดงออกทางจอภาพของไมโครคอมพิวเตอร์ สามารถเขียนเป็นบล็อควงจรของลอจิก อนาไลเซอร์ได้ดังรูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 บล็อควงจรลอจิก อนาไลเซอร์

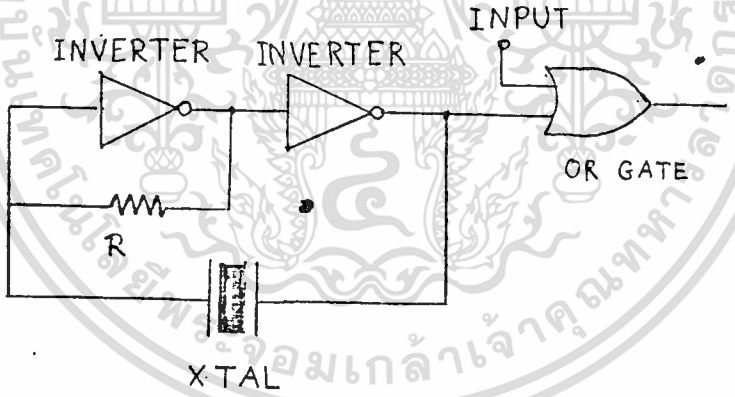
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

3.1 การสุ่มค่าสัญญาณ

หน่วยความจำที่ใช้ในโครงงานนี้ได้ใช้หน่วยความจำเบอร์ 6264 ซึ่งมีขนาด 8k * 8 จำนวน 8 ตัวมาต่อขนานกันเป็นหน่วยความจำความเร็วสูงที่มีขนาด 6 MHz * 8 หรือเท่ากับ 48 MHz (หน่วยความจำเบอร์ 6264 หนึ่งตัวมีความเร็วประมาณ 6 MHz) ดังนั้นความถี่ที่ใช้สุ่มสัญญาณจึงมีค่าสูงสุดประมาณ 48 MHz แต่ในโครงงานนี้ได้ทดลองใช้ความถี่สุ่มค่าสัญญาณ 37.75 MHz ซึ่งเราสามารถเปลี่ยนความถี่ที่ใช้สุ่มค่าสัญญาณได้ โดยเพียงเปลี่ยนคริสตอล (XTAL) เท่านั้น ดังนั้น ลอจิก อนาไลเซอร์ ชุดนี้จึงสามารถวัดสัญญาณที่มีความถี่สูงถึง 20 MHz ได้



วงจรรูปที่ 2 เป็นวงจรแสดงการสร้างสัญญาณนาฬิกาเพื่อกระตุ้นวงจรนับ โดยคริสตอลจะสร้างสัญญาณนาฬิกาเข้าออร์เกท (OR Gate) ถ้าขาอินพุท (input) มีค่าเป็นหนึ่ง (high) เอาท์พุท (output) ของออร์เกทก็จะมีค่าเป็นหนึ่งตลอด แต่ถ้าขาอินพุทมีค่าเป็นศูนย์ (low) แล้ว เอาท์พุทของออร์เกทก็จะมีค่าเป็นสัญญาณนาฬิกาที่มีความถี่เท่ากับความถี่ของคริสตอล สัญญาณเอาท์พุทของออร์เกทนี้จะต่อเข้ากับขา CLK ของวงจรนับ

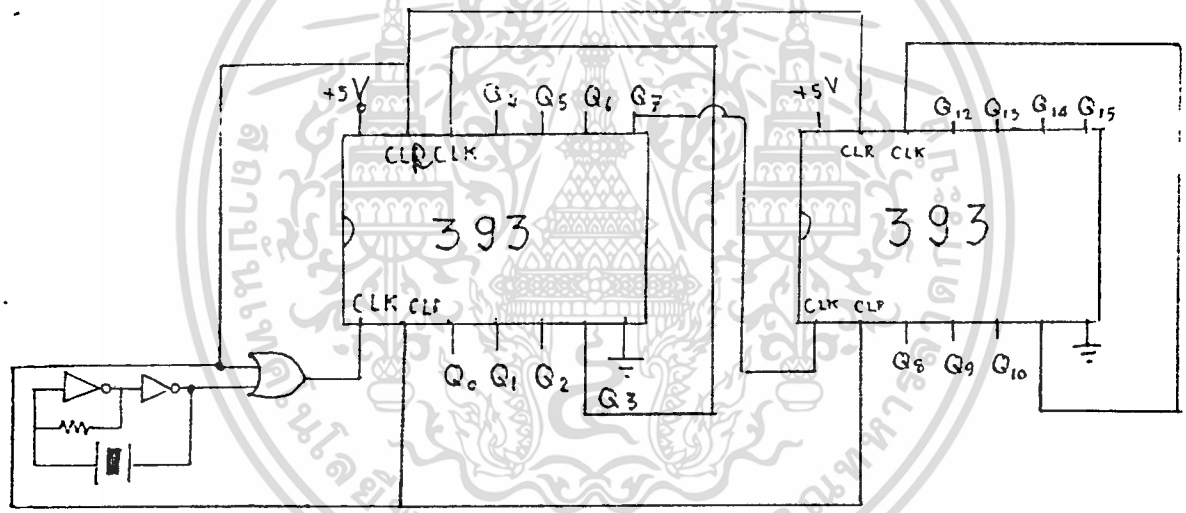
เนื่องจากหน่วยความจำความเร็วสูงมีขนาด 64 K * 8 และมีสายสัญญาณแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งถึง 16 เส้น คิดจาก

$$2^{16} = 64\ K$$

ดังนั้นวงจรนับจึงจะต้องนับถอยหลังจาก 0000 ถึง FFFF ซึ่งแสดงถึง 64 K ตำแหน่ง จึงต้องใช้ไอซีเบอร์ 74LS93 จำนวน 2 ตัว มีสายสัญญาณเอาท์พุทรวม 16 เส้น สายสัญญาณเอาท์พุทนี้จะต้องต่อกับสายสัญญาณแสดงตำแหน่งของหน่วยความจำความเร็วสูงโดยผ่านวงจนมัลติเพลกเซอร์ก่อน วงจรรูปที่ 3 ข้างล่างนี้เป็นวงจรนับที่ใช้ไอซีเบอร์ 74LS93 จำนวน 2 ตัว ต่ออนุกรมกัน



รูปที่ 3 วงจรนับ

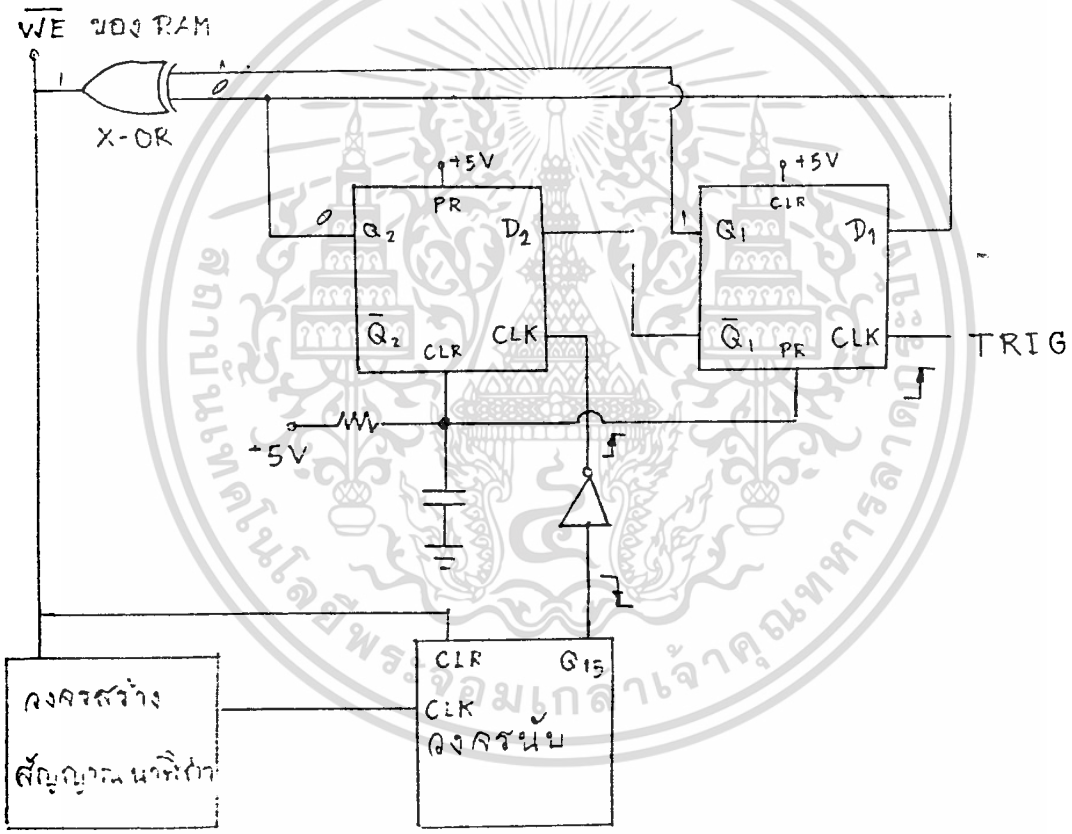
เมื่อมีสัญญาณนาฬิกาจากวงจรสร้างสัญญาณนาฬิกาเข้ามาที่ขา CLK ของวงจรนับและขา CLR เป็นศูนย์แล้ว วงจรนับก็จะเริ่มนับเพิ่มขึ้นเรื่อยๆจนถึง FFFF แล้วก็วนเริ่มใหม่ที่ 0000 เป็นเช่นนี้ไปเรื่อยๆ จนกว่าจะมีการเปลี่ยนแปลงสัญญาณนาฬิกาและสัญญาณ CLR ดังนั้นเราจึงสามารถควบคุมวงจรนับได้จากสัญญาณขา CLR กับขา CLK

3.2 การตรวจสอบสภาวะของสัญญาณ

วงจรตรวจสอบสภาวะการทำงานของ ลอจิก อนาไลเซอร์ นี้จะไปเคลียร์วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จรนับเพื่อไม่ให้นับ จนกว่าจะมีสัญญาณทริกจากภายนอกเข้ามากระตุ้นให้เริ่มเขียนข้อมูลลงไปในหน่วยความจำ ดังนั้น ลอจิก อนาไลเซอร์ จะเริ่มสุ่มค่าสัญญาณ ก็ต่อเมื่อมีสัญญาณจากภายนอกมาทริกให้ทำงาน และเมื่อทริกให้วงจรนับทำงานแล้ว วงจรตรวจสอบสภาวะการทำงานก็จะตรวจสอบว่านับจนถึง FFFF แล้วหรือยัง ถ้ายังนับไม่ถึง FFFF ก็ให้นับต่อไป แต่ถ้านับถึง FFFF แล้ว วงจรตรวจสอบนี้ก็จะให้สัญญาณมาควบคุมวงจรนับให้หยุดนับ ดังรูปที่ 4



รูปที่ 4 วงจรตรวจสอบสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มเปิดเครื่อง ดี ฟลิปฟลอป (D - Flip Flop) ทั้ง2 ตัวจะให้เอาท์พุททางขา Q ที่ต่างกัน คือ ฟลิปฟลอปตัวที่ขา PR ต่อกับแหล่งจ่ายไฟ +5 V. จะให้เอาท์พุทที่ขา Q เป็นศูนย์ ส่วนตัวที่ขา CLR ต่อกับแหล่งจ่ายไฟ +5 V. จะให้เอาท์พุทที่ขา Q เป็นหนึ่ง เพราะฉะนั้น เอาท์พุทของเอ็กซ์คลูซีฟออร์เกท (X-OR Gate) เมื่อเริ่มเปิดเครื่องก็จะเป็นหนึ่ง ทำให้หน่วยความจำไม่สามารถเขียนข้อมูลได้ และยังเคลียร์วงจรนับให้เป็น 0000 แต่ยังไม่มีการนับ

เมื่อมีสัญญาณทริกจากภายนอกเข้ามา ขา CLK จะแอ็กทีฟ (Active) ที่ขอบขาขึ้นของสัญญาณทริก ฟลิปฟลอปตัวที่หนึ่ง ก็จะให้เอาท์พุทขา Q เป็นศูนย์ ทำให้เอ็กซ์คลูซีฟออร์เกทให้เอาท์พุทเป็นศูนย์ วงจรนับก็จะเริ่มนับตั้งแต่ 0000 ไปจนถึง FFFF ก็จะวนกลับมานับ 0000 ใหม่อีกครั้ง ช่วงที่เปลี่ยนจาก FFFF ไปเป็น 0000 ขา Q_{16} ของวงจรนับจะเปลี่ยนสภาวะจาก "1" เป็น "0" เมื่อขา Q_{16} ให้สัญญาณขอบขาลง "↓" ผ่านอินเวอร์ทเตอร์ก็จะเป็นขอบขาขึ้น "↑" ทำให้ฟลิปฟลอปตัวที่ 2 ให้เอาท์พุทเป็น "1" เอาท์พุทของเอ็กซ์คลูซีฟออร์เกทก็จะเป็น "1" ทำให้วงจรนับหยุดนับ ดังสภาวะการทำงานตามตารางที่ 1

	สถานะของสัญญาณ							
	Q_1	$\bar{Q}_1$	Q_2	$\overline{WE}$	D_1	D_2	O/P ของ X-OR	CLR ของ CT.
ยังไม่ TRIG	1	0	0	1	0	0	1	1
ระหว่างนับ	0	1	0	0	0	1	0	0
นับครบ 64K	0	1	1	1	1	1	1	1

ตารางที่ 1

3.3 การติดต่อกับหน่วยความจำความเร็วสูง

การที่จะอ่านหรือเขียนข้อมูลลงในหน่วยความจำ จะต้องมีสายสัญญาณแสดงตำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

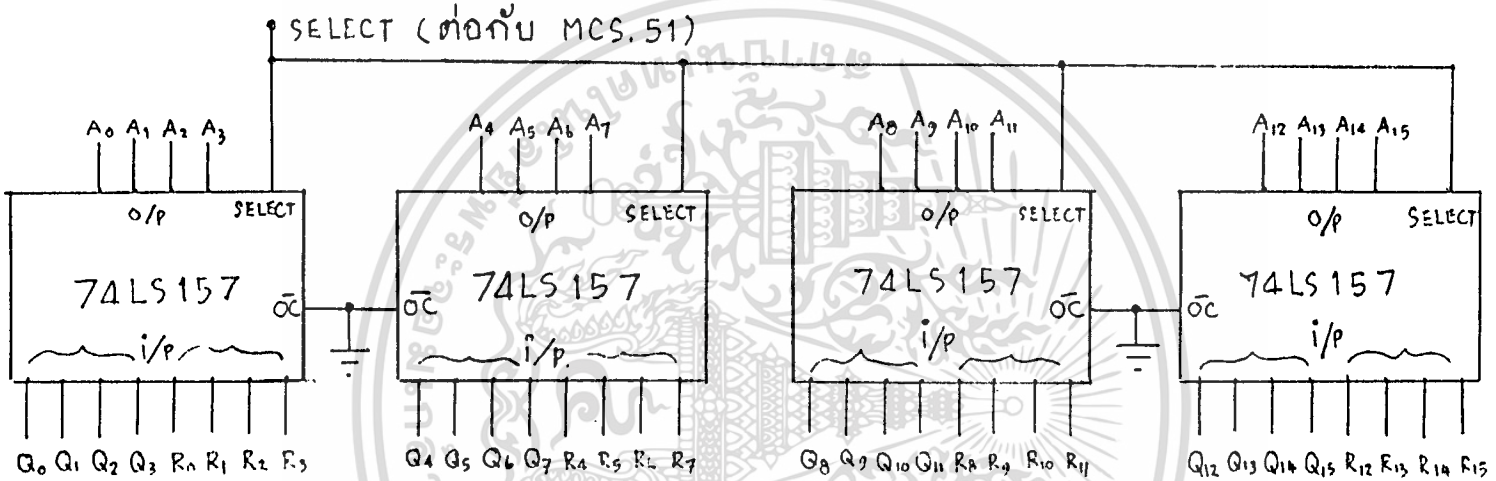
แหน่งเชื่อมติดต่อกัน โดยผ่านวงจรมัลติเพลกเซอร์ วงจรมัลติเพลกเซอร์นี้จะมีสัญญาณที่จะเลือกว่าให้สายแสดงตำแหน่งชุดใดผ่านออกไป ถ้าเป็นการเขียนข้อมูลก็จะให้สายแสดงตำแหน่งจากวงจรนับผ่านออกไป แต่ถ้าเป็นการอ่านข้อมูลก็จะให้สายแสดงตำแหน่งจาก MCS.51 ผ่านออกไป ดังรูปที่ 5

$Q_0 - Q_{15}$ คือ เอาท์พุทของวงจรนับ

$R_0 - R_{15}$ สายแสดงตำแหน่งของ MCS.51

$A_0 - A_{15}$ สายแสดงตำแหน่งเอาท์พุทของมัลติเพลกเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
023170

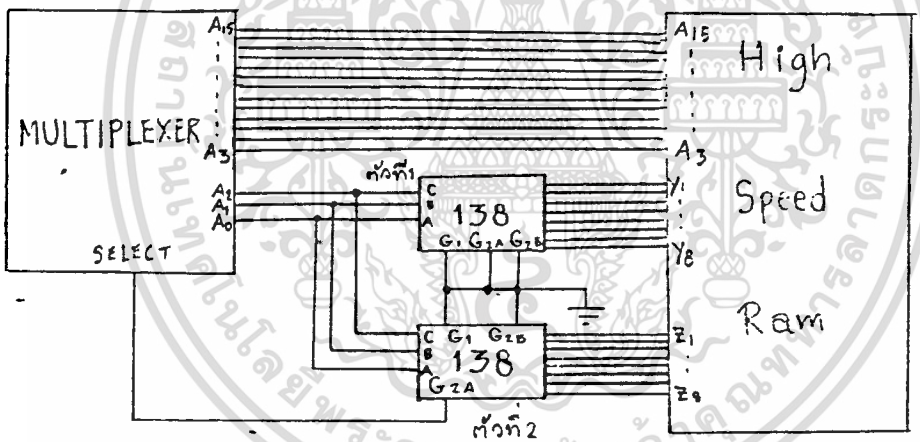


รูปที่ 5 วงจรมัลติเพลกเซอร์

IC เบอร์ 74LS157 มีอินพุท 2 ชุด ชุดละ 4 เส้น และมีเอาท์พุท 4 เส้น โดยมีสัญญาณเลือกว่าจะให้สายสัญญาณชุดใดผ่านออกไป ในที่นี้ มีสายแสดงตำแหน่ง 16 เส้น จึงต้องใช้ IC 74LS157 จำนวน 4 ตัว

สัญญาณ select มาจากขา P1.2 ของ MCS.51 ซึ่งจะเป็นสัญญาณควบคุมการอ่านและเขียนข้อมูล

A_0 - A_2 เอาท์พุทของมัลติเพลกเซอร์จะต่อเข้ากับ IC 74LS138 จำนวน 2 ตัวขนานกัน IC 74LS138 เป็นตัวถอดรหัส มีอินพุท 3 เส้น แต่ให้เอาท์พุท 8 เส้น เอาท์พุทของ IC 74LS138 ทั้งสองตัวจะต่อเข้ากับหน่วยความจำความเร็วสูง ดังรูปที่ 6



รูปที่ 6 วงจรถอดรหัส

IC 74LS138 ตัวหนึ่งจะเป็นตัวอีเนเบิ้ล (Enable) หน่วยความจำย่อย เมื่อต้องการเขียนข้อมูล ส่วนอีกตัวหนึ่งจะเป็นตัวอีเนเบิ้ลหน่วยความจำย่อย, เมื่อต้องการอ่านข้อมูล

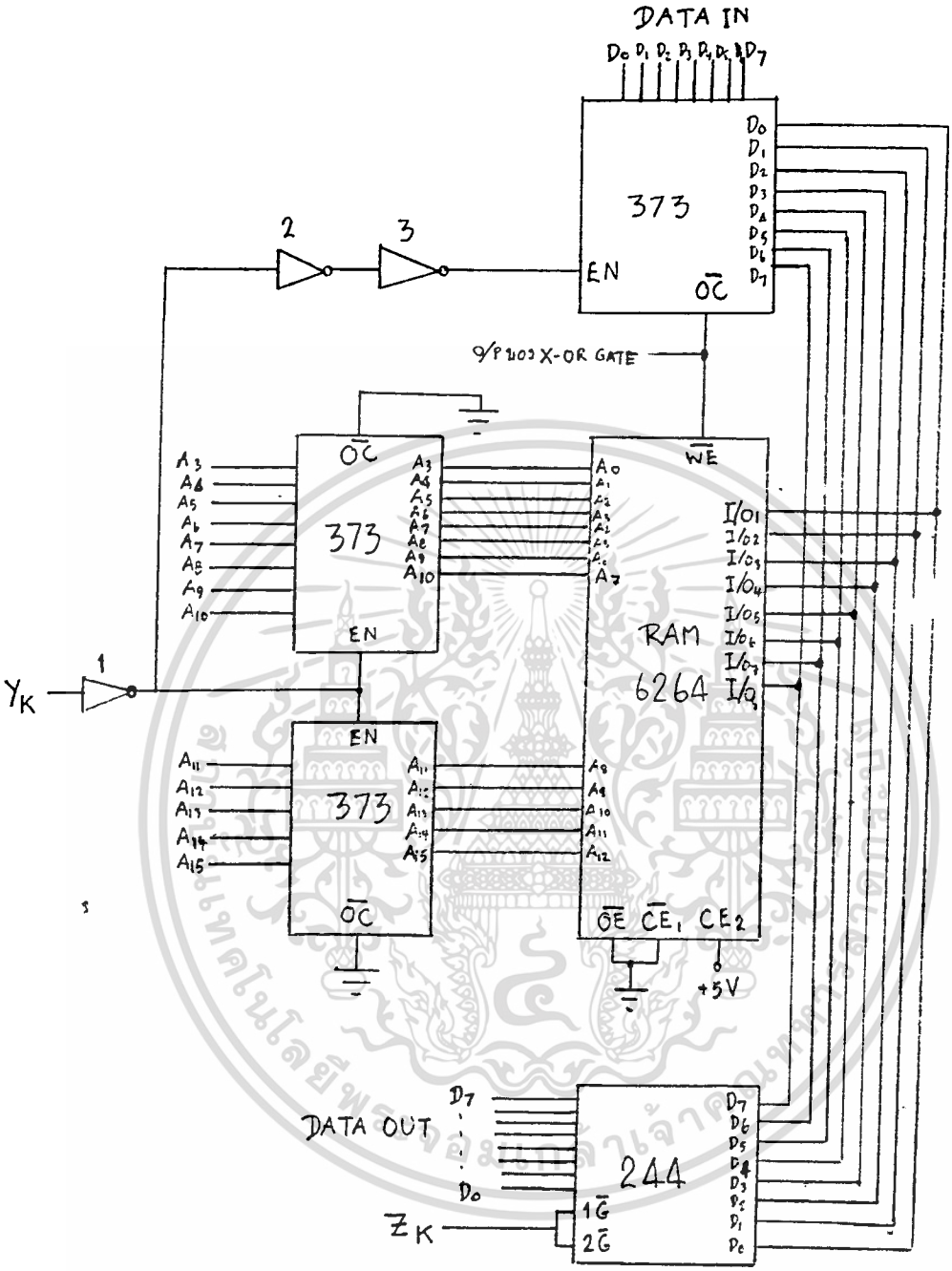
IC 74LS138 ตัวที่ถอดรหัสเมื่อต้องการเขียนข้อมูล จะถูกอีเนเบิ้ลตลอด ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวถอดรหัสเมื่อต้องการอ่านข้อมูล ขา G_2A จะต่อกับขาเซเล็ค (Select) ของมัลติเพลกเซอร์ ถ้าขาเซเล็คเป็น "1" ขา G_2A ก็จะไม่อีเนเบิ้ล แต่ถ้าขาเซเล็คเป็น "0" ขา G_2A ก็จะถูกอีเนเบิ้ล

ขา A_3 - A_{15} จะต่อเข้ากับ IC74LS373 เพื่อแลทช์แอดเดรส ขา A_3 - A_{15} จะต่อกับบอร์ดหน่วยความจำย่อยของหน่วยความจำสูงทั้ง 8 บอร์ดขนานกันหมด โดยมีขาเอาท์พุท Y_K ของ 74LS138 เป็นตัวอีเนเบิ้ลแอดเดรสแลทช์ Y_1 จะต่อกับบอร์ดหน่วยความจำอันดับที่ 1 Y_2 จะต่อกับบอร์ดที่ 2 Y_3 จะต่อกับบอร์ดที่ 3 ไปจน Y_8 ต่อกับบอร์ดที่ 8 ในทำนองเดียวกันเอาท์พุท Z ของ 74LS138 ตัวที่สองที่ถอดรหัสกรณีที่ต้องการอ่านข้อมูลก็จะต่อเข้ากับขา $\overline{OC}$(Output Control) ของไตรสเตท (Tri State) IC 74LS244 โดย Z_1ต่อกับขา $\overline{OC}$ ของ 74LS244 บอร์ดที่ 1 Z_2 ต่อกับบอร์ดที่ 2 ไปจน Z_8 ต่อกับบอร์ดที่ 8 ดังรูป 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 บอร์ดหน่วยความจำบอร์ดที่ K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตได้ว่า หน่วยความจำแต่ละบอร์ดจะทำงานไม่พร้อมกัน ขึ้นอยู่กับการถอดรหัสของ IC 74LS138 ทั้งสองตัวว่าจะเลือกให้บอร์ดใดทำงาน

จากรูปที่ 7 อินเวอร์ทเตอร์สองตัวที่ต่ออนุกรมกัน มีไว้เพื่อหน่วงเวลาให้สายแสดงตำแหน่งเข้าถึงหน่วยความจำก่อน แล้วข้อมูลจึงเข้าหน่วยความจำตามหลัง และเนื่องจากว่า 74LS373 ขาอีเนเบิ้ลจะแอ็กทีฟเมื่อเป็น "1" แต่เอาท์พุทของ 74LS373 จะแอ็กทีฟเมื่อเป็น "ø" ดังนั้น จึงต้องต่ออินเวอร์ทเตอร์เพื่อกลับค่า "ø" ให้เป็น "1"

ส่วนขา $\overline{OC}$ ของ 74LS373 เดต้าแลทช์จะต่อกับขา WE ของหน่วยความจำ 6264 โดยทั้งสองค่านี้จะต่อกับเอาท์พุทของเอ็กซ์คลูซีฟออร์เกท ในส่วนของวงจรตรวจสอบสภาวะ ในรูปที่ 4

3.4 การจัดการรับ-ส่งข้อมูล

ในส่วนนี้จะใช้ MCS.51 ควบคุมและจัดการข้อมูล โดย MCS.51 จะทำงานตามขั้นตอนที่ป้อนไว้แล้วในเบอร์ 2732 ซึ่งมีขนาด 4K * 8

เมื่อสุ่มค่าสัญญาณและเขียนค่าลงในหน่วยความจำจนเต็มแล้ว และไมโครคอมพิวเตอร์ต้องการที่จะแสดงผล ก็จะส่งสัญญาณมาบอก MCS.51 ให้ทำการติดต่อกับหน่วยความจำความเร็วสูงเพื่อส่งข้อมูลไปให้ไมโครคอมพิวเตอร์

เมื่อ MCS.51 ได้รับสัญญาณจากไมโครคอมพิวเตอร์ ก็จะส่งสัญญาณออกไปที่ขา P1.2 เพื่อให้มัลติเพลกเซอร์เลือกสายแสดงตำแหน่งของ MCS.51 ผ่านออกจากมัลติเพลกเซอร์ไปยังหน่วยความจำความเร็วสูง

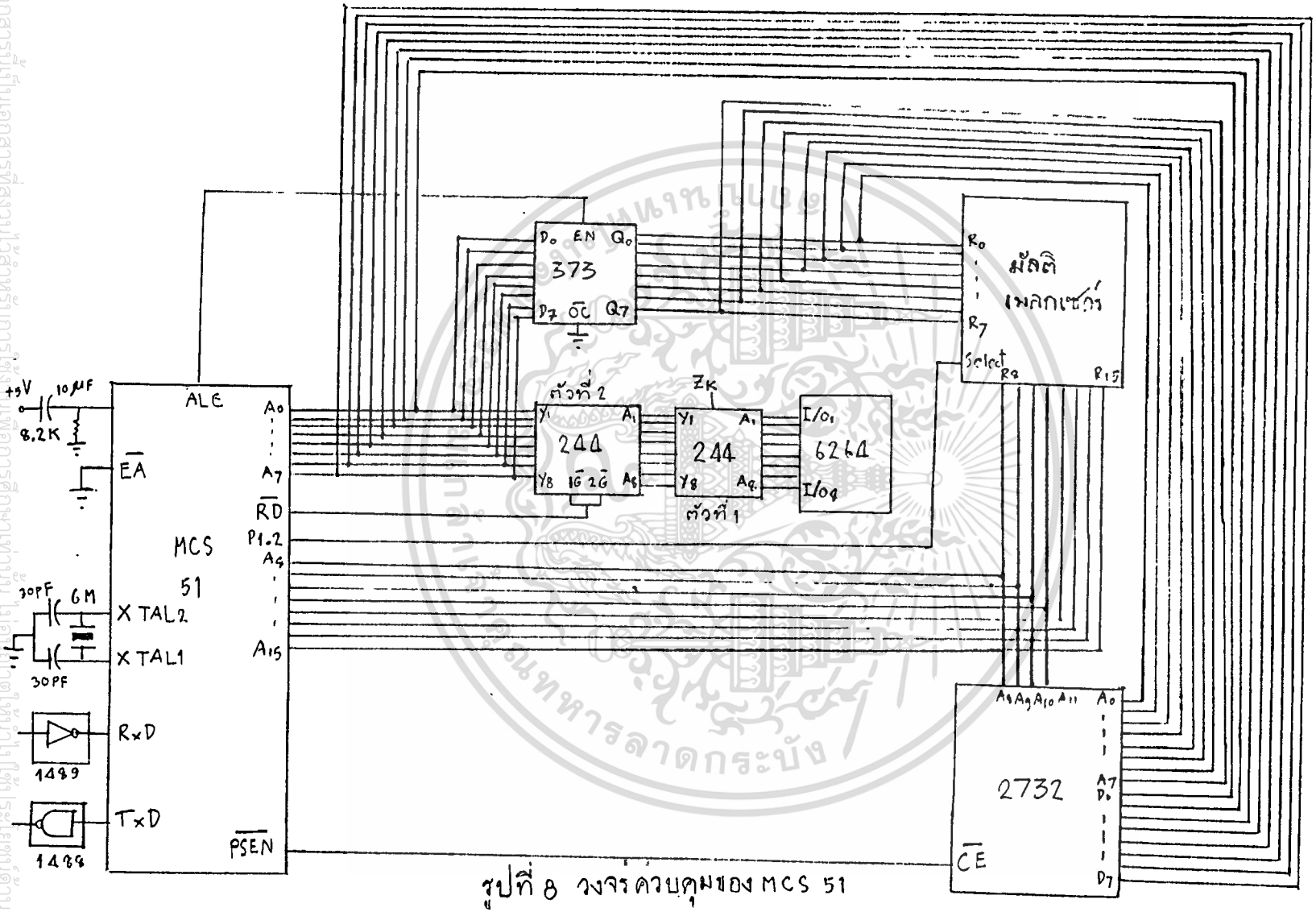
สัญญาณขา ALE ของ MCS.51 จะเป็นสัญญาณที่ไปอีเนเบิ้ล IC แลทช์เบอร์ 74LS373 ให้สายแสดงตำแหน่งผ่าน

สัญญาณขา PSEN เป็นสัญญาณที่จะไปอีเนเบิ้ลอีพรอม 2732

สัญญาณขา RD ของ MCS.51 จะไปอีเนเบิ้ลไตรสเตท 74LS244 ให้ข้อมูลผ่านออกมาได้

MCS.51 จะเริ่มต้นด้วยการส่งสัญญาณแสดงตำแหน่งไปยังหน่วยความจำถาวร แล้วอ่านคำสั่งจากอีพรอมผ่านทางขา $D_0 - D_7$ แล้วก็ทำงานตามคำสั่ง หลังจากทำงานจบหนึ่งคำสั่งแล้วก็ส่งสัญญาณแสดงตำแหน่งถัดไปไปยังอีพรอมอีก เพื่ออ่านคำสั่งต่อไป จะทำเช่นนี้ไปเรื่อย ๆ จนหมดคำสั่ง รูปวงจรควบคุมของ MCS.51 มีดังรูปที่ 8

เอกสารนี้ได้รับการเผยแพร่ผ่านระบบคลังปัญญาจุฬาฯ เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 วงจรควบคุมของ MCS 51

จากรูปไตรสเตท 74244 ตัวที่หนึ่งจะอยู่ในบอร์ดของหน่วยความจำย่อยทั้ง 8 บอร์ด ซึ่ง 74244 ของทุกบอร์ดจะต่อขนานถึงกันหมด ขาเอาท์พุทของ 74244 ตัวที่หนึ่ง จะต่อเข้าอินพุท 74244 ตัวที่ 2 ส่วน A_0 - A_{15} จะต่อเข้าขาอินพุทของมัลติเพลกเซอร์ (ดูในรูปที่ 5)

สายข้อมูล D_0 - D_7 กับสายแสดงตำแหน่ง A_0 - A_7 ของ MCS.51 จะใช้ร่วมกัน โดยมีขาสัญญาณ ALE , PSEN , RD เป็นสัญญาณควบคุมไม่ให้สัญญาณข้อมูลและสัญญาณแสดงตำแหน่งใช้งานพร้อมกัน จึงไม่เกิดการสับสนและผิดพลาด

คริสตอลที่ใช้ มีความถี่ 6 MHz เป็นความถี่ในการทำงานของ MCS.51

ขา RxD ของ MCS.51 ต่อเข้ากับเอาท์พุทของ 1489 เพื่อรับข้อมูลจากไมโครคอมพิวเตอร์

ขา TxD ของ MCS.51 ต่อเข้ากับอินพุทของ 1488 เพื่อส่งข้อมูลไปยังไมโครคอมพิวเตอร์

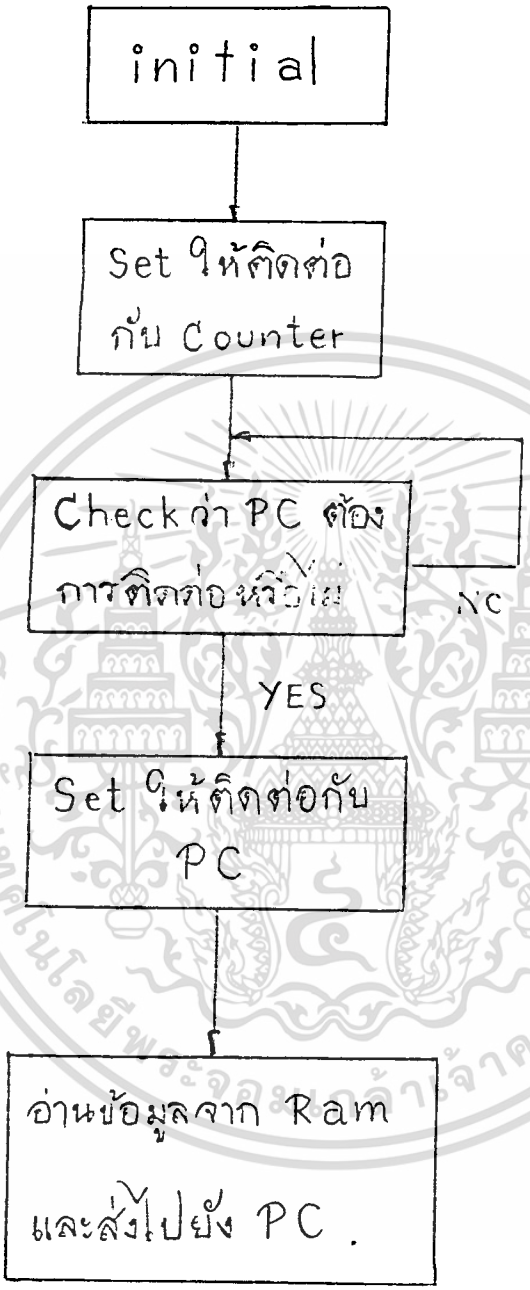
MC1488 เป็นตัวขับสัญญาณ (Driver) โดยเปลี่ยนแรงดันจาก +5 V ไปเป็น 12 V.

MC1489 เป็นตัวรับสัญญาณ (Receiver) ปรับแรงดันจาก 12 V. ไปเป็น 5V

ขาอินพุทของ 1489 และขาเอาท์พุทของ 1488 จะต่อกับ RS232 ผ่านเข้าไปในไมโครคอมพิวเตอร์ โดยรับค่าข้อมูลแบบอนุกรม

ขั้นตอนการทำงานของ 8031 ถูกป้อนลงในอีพรอม (EPROM) ซึ่งมีแผนผังแสดงขั้นตอน ดังรูปที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 ขั้นตอนการทำงานของ MCS.51

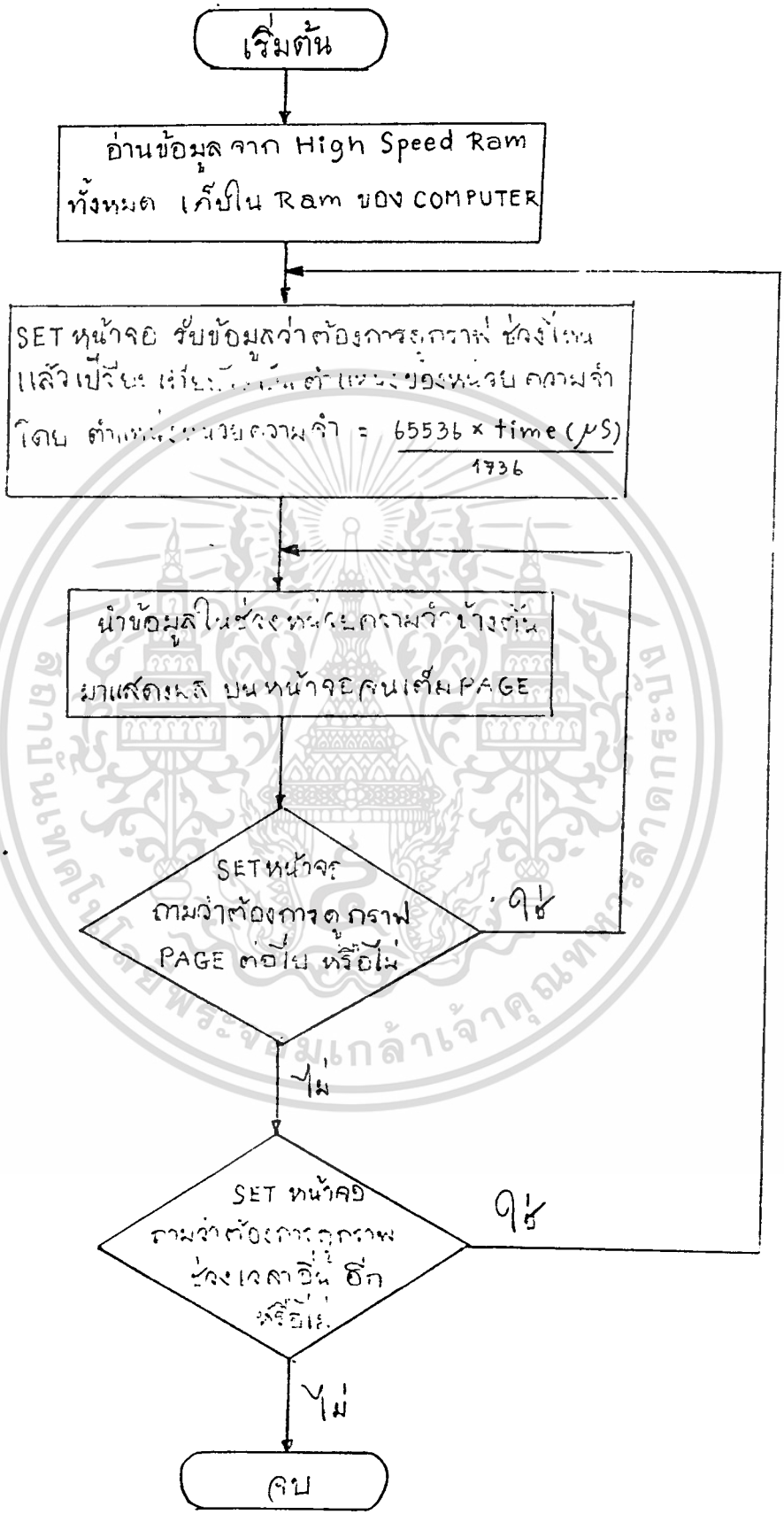
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ป้อนลงในอีพรอม โดยมีการเปลี่ยนแปลงออฟโค้ด (Opcode) ได้แสดงในภาคผนวก

3.5 การแสดงผล

ในส่วนแสดงผลเป็นซอฟท์แวร์ทั้งหมด มีเพียงการต่อเชื่อมกันระหว่าง MCS.51 กับไมโครคอมพิวเตอร์ด้วย RS232 ขั้นตอนการทำงานของส่วนแสดงผลมีแผนผังดังรูปที่ 10 และรายละเอียดของโปรแกรมแสดงไว้ในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 ขั้นตอนการทำงานของส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 วิธีทดลอง

ป้อนสัญญาณนาฬิกาที่มีความถี่คงที่ 1 MHz ให้กับขา CLK ของ 74LS393 ได้เอาท์พุท 8 สัญญาณ เป็นสัญญาณนาฬิกาโดย

สัญญาณที่ 1	มีความถี่คงที่	1	MHz
สัญญาณที่ 2	มีความถี่คงที่	500	KHz
สัญญาณที่ 3	มีความถี่คงที่	250	KHz
สัญญาณที่ 4	มีความถี่คงที่	125	KHz
สัญญาณที่ 5	มีความถี่คงที่	62.5	KHz
สัญญาณที่ 6	มีความถี่คงที่	31.25	KHz
สัญญาณที่ 7	มีความถี่คงที่	15.625	KHz
สัญญาณที่ 8	มีความถี่คงที่	7.8125	KHz

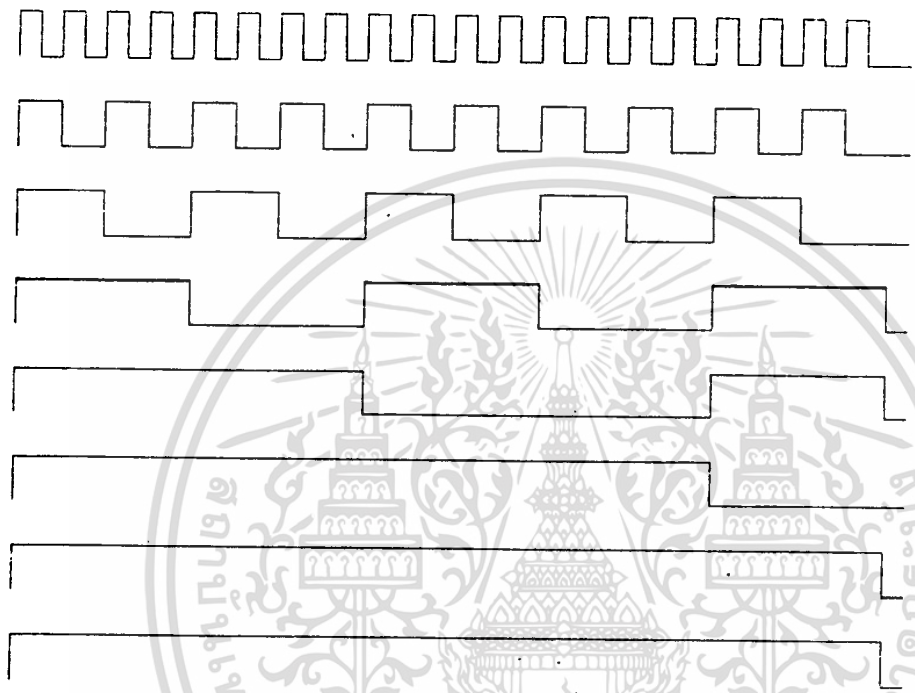
ใช้ลอจิกอนาไลเซอร์วัดสัญญาณทั้ง 8 โดยใช้ความถี่ในการสุ่มเท่ากับ 37.75 MHz แล้วแสดงผลออกทางจอภาพของไมโครคอมพิวเตอร์

4.2 ผลการทดลอง

ผลการทดลองได้แสดงบนจอภาพของไมโครคอมพิวเตอร์ ซึ่งมีสัญญาณดังรูปที่ 11

จากรูปที่ 11 สัญญาณที่แสดงบนจอภาพของไมโครคอมพิวเตอร์มีลักษณะเหมือนกับสัญญาณเอาท์พุทของวงจรนับ 74LS393 อีกทั้งยังมีความถี่ตรงกันอีกด้วย และจากการเปลี่ยนความถี่ของสัญญาณนาฬิกาที่ป้อนให้กับขา CLK ของ 74LS393 ได้ผลลัพธ์ถูกต้องตรงกับสัญญาณที่ถูกวัดทุกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11 ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

บทวิจารณ์และสรุป

จากผลการทดลองในบทที่ 4 เมื่อนำลอจิกอนาไลเซอร์วัดสัญญาณดิจิตอลใด ๆ ที่มีความถี่ไม่เกิน 20 MHz ลอจิกอนาไลเซอร์สามารถวัดและแสดงผลบนจอภาพไมโครคอมพิวเตอร์ได้อย่างถูกต้อง เหมือนสัญญาณที่แท้จริง ดังนั้น จึงสามารถที่จะนำลอจิกอนาไลเซอร์เครื่องนี้ไปใช้งานจริงได้

โครงงานนี้สามารถทำงานได้ตามหลักการและทฤษฎีดังได้กล่าวในบทที่ 2 ทุกประการ มีปัญหาอยู่ดพียงเล็กน้อยเท่านั้น อาทิเช่น ขนาดของสายไฟไม่เท่ากัน อุปกรณ์สนับสนุนไม่เพียงพอ ความปราณีตในการเชื่อมต่อสายไฟและอุปกรณ์มีน้อยกว่าโรงงานอุตสาหกรรม ซึ่งปัญหาเหล่านี้เป็นเพียงปัญหาเล็ก ๆ มีผลกระทบต่อการพัฒนาโครงงานเพียงเล็กน้อยเท่านั้น

โครงงานนี้สามารถพัฒนาให้สมบูรณ์มากขึ้นอีกได้โดยเพิ่มเติมในส่วนของซอฟท์แวร์ที่จะเพิ่มฟังก์ชัน (Function) บางอย่างที่ทำหน้าที่อำนวยความสะดวกในการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมภายในอีพรอม

```
        MOV     SCON , 50H          75  98  50
        MOV     PCON , 00H          75  87  00
        MOV     TCON , 40H          75  88  40
        MOV     TMOD , 20H          75  89  20
        MOV     TH1 , F3H           75  8D  F3
        MOV     IE , 90H            75  A8  90
        MOV     IP , 10H            75  B8  10
        MOV     A , FF              74  FF
        MOV     P1 , A              F5  90
        MOV     DPTR , 0000         90  00  00
        ANL     A8 , 7F             53  A8  7F
X1      JNB     RI , X1             30  98  FD
        CLR     RI                  C2  98
        MOV     A , 00              74  00
        MOV     P1 , A              F5  90
        MOV     R0 , FF             78  FF
X2      DJNZ    R0 , X2             D8  FE
        DJNZ    R0 , FF             78  FF
X3      DJNZ    R0 , X3             D8  FE
        MOV     R0 , FF             78  FF
X4      DJNZ    R0 , X4             D8  FE
        MOVX    A , DPTR            E0
        INC     DPTR                A3
        MOV     SBUF , A            F5  99
        ANL     A8 , 7F             53  A8  7F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
X5      JNB     TI , X5          30  99  FD
        CLR     TI               C2  99
        MOV     R0 , FF          78  FF
X6      DJNZ    R0 , X6          D8  FE
        MOV     R0 , FF          78  FF
X7      DJNZ    R0 , X7          D8  FE
        MOV     R0 , FF          78  FF
X8      DJNZ    R0 , X8          D8  FE
        LJMP    0018             02  00  18
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข

โปรแกรมส่วนแสดงผล

```
Program LogicAnalyser;
Uses  Graph,Crt,Dos;
Type  Data = Array[0..16383] of Byte;
Var   Block : Array[1..4] of ^Data;
      I,Q1,Q2,R,T,K,V,Z,U,N,J : Integer;
      P1,P2 : Real;
      Segment : Integer;
      Offset  : Integer;
      F,M : Byte;
      Keep : File Of Data;
{$R-}    {Range checking off}
{$B-}    {Boolean short circuiting off}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$M 65500,16384,655360} {Turbo 3 default stack and heap}

(*- MODULE ----- RS-232C Interface Routine -----*)

{--- OUT PORT-----------------------------------------------------}
{ $U+}

Procedure O232(var Dout:Byte);
Const  BUAD=$83; { 1200 Bit/Sec }
Var _Regs:Record Al,Ah:Byte; BX,CX,DX,BP,DI,SE,DS,ES,Flags: Integer end;
procedure _Int14(Cmd,Parm:Byte);
begin
    with _Regs do
    begin
       DX:=0; AH:=Cmd; AL:=Parm; Flags:=0; Intr($14,Dos.Registers(_Regs))
    end
end;
begin
   _Int14(0,BUAD);
   With _Regs do
   begin
      Repeat _Int14(1,Dout);
      Until (AH and $8E)=0;
    end;
end;

{--- IN PORT------------------------------------------------------}
{ $U+}
Procedure I232(var Dinp:Byte);
Const BUAD=$83; { 1200 Bit/Sec }
Var _Regs:Record Al,Ah:Byte; BX,CX,DX,BP,DI,SE,DS,ES,Flags: Integer end;
procedure _Int14(Cmd,Parm:Byte);
begin
   with _Regs do
   begin
      DX:=0; AH:=Cmd; AL:=Parm; Flags:=0;Intr($14,Dos.Registers(_Regs))
   end
end;
var Il:Byte;
begin
   _Int14(0,BUAD);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    With _Regs do
    begin
        Repeat _Int14(2,0); Until (Ah and $8E)=0; Il:=AL;
    end;
    DInp:=il
end;
    Procedure Memory;
    Begin
          ClrScr;
          Writeln('          LOGIC ANALYSER          ');
          Writeln('          **************          ');
          Writeln('                                  ');
          Writeln('Please Select Number Of Data (That you want to keep.)');
          Writeln('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
          Writeln(' 1. Keep 16K Byte                 ');
          Writeln(' 2. Keep 32K Byte                 ');
          Writeln(' 3. Keep 48K Byte                 ');
          Writeln(' 4. Keep 64K Byte                 ');
          Writeln('                                  ');
          Write('   Select No. : ');Readln(U);
          Assign(Keep,'Por');
          Rewrite(Keep);
          For I := 1 To U Do
          Begin
               GetMem(Block[I],16384);
               Segment := Seg(Block[I]^);
               Offset := 0;
               For Z := 0  to 16383 do
               Begin
                    M := $55;
                    O232(M);
                    I232(F);
                    Writeln(F);
                    Mem[Segment:Offset] := F;
                    Offset := Offset+1;
               End;
          End;
          Write('Press "0" To Save  Data : Otherwise Press "Any No." : ');
          Read(J);
          While (J = 0) Do
          Begin
            For I := 1 To U Do  Write(Keep,Block[i]^);
            Close(Keep);
          End;
    End;
    Procedure Calculate;
    Begin
          ClrScr;
          Writeln('       Please Select Range Of Time ');
          Writeln('       ***************************');
          Writeln('                                                  ');
          Writeln('Range No.1     0-434   microsec (Adress 00000-16383) ');
          Writeln('Range No.2   435-868   microsec (Adress 16384-32766) ');
          Writeln('Range No.3   869-1302  microsec (Adress 32767-49151) ');
          Writeln('Range No.4  1303-1736  microsec (Adress 49152-65535) ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Writeln('                                                          ');
        Write('              Select Range No.: '); Readln(I);
        Writeln('                                                          ');
        Writeln('                                                          ');
        Writeln('                                                          ');
        Writeln('Enter Subrange ');
        Writeln('~~~~~~~~~~~~~~ ');
        Write('Initial From : '); Readln(P1);
        Write('Until : '); Readln(P2);
        Q1 :=Round(16383*P1/434);
        Q2 :=Round (16383*P2/434);
        R := Q2-Q1;
        T := R Div 600;
End;
Procedure Scan;
Var   GraphDriver,GraphMode : Integer;
      ErrorCode : Word;
      M,W,X1,Y1,X2,Y2,N : Integer;
      L,G,Ch : Char;
      S1,S2,S3,S4,S5,S6 : String[7];
      FuncKey : Boolean;
Begin
      GraphDriver := Detect;
      InitGraph(GraphDriver,GraphMode,'  );
      ErrorCode := GraphResult;
      If ErrorCode <> grOk Then
      Begin
           Writeln('Graphic error: ', GraphErrorMsg(ErrorCode));
           Writeln('(You probably don''t have a graphic card!)');
           Writeln('Program aborted...');
           Halt(1);
      End;
      ClearViewPort;
      For W := 0 To T Do
      Begin
           ClearViewPort;
           M := 0;
           For Offset := (Q1+600*W) To (Q1+600*(W+1)) Do
           Begin
                X1 := 50+M;
                X2 := 50+(M+1);
                If (Mem[Segment:Offset-1] And $01) <> (Mem[Segment:Offset]
nd $01) Then
                   If Mem[Segment:Offset] And $01 <>0 Then
                      Begin Y1 := 40; Y2 := 20; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 20; Y2 := 20; End
                   Else
                      Begin Y1 := 20; Y2 := 40; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 40; Y2 := 40; End
                Else
                   If Mem[Segment:Offset] And $01 <>0 Then
                      Begin Y1 := 20; Y2 := 20; End
                   Else
                      Begin Y1 := 40; Y2 := 40; End;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MoveTo(0,0);
                Line(X1,Y1,X2,Y2);
nd $02) Then    If (Mem[Segment:Offset-1] And $02) <> (Mem[Segment:Offset] A

                   If Mem[Segment:Offset] And $02 <>0 Then
                      Begin Y1 := 80; Y2 := 60; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 60; Y2 := 60; End
                   Else
                      Begin Y1 := 60; Y2 := 80; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 80; Y2 := 80; End
                Else
                   If Mem[Segment:Offset] And $02 <>0 Then
                      Begin Y1 := 60; Y2 := 60; End
                   Else
                      Begin Y1 := 80; Y2 := 80; End;
                MoveTo(0,0);
                Line(X1,Y1,X2,Y2);
nd $04) Then    If (Mem[Segment:Offset-1] And $04) <> (Mem[Segment:Offset] A

                   If Mem[Segment:Offset] And $04 <>0 Then
                      Begin Y1 := 120; Y2 := 100; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 100; Y2 := 100; End
                   Else
                      Begin Y1 := 100; Y2 := 120; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 120; Y2 := 120; End
                Else
                   If Mem[Segment:Offset] And $04 <>0 Then
                      Begin Y1 := 100; Y2 := 100; End
                   Else
                      Begin Y1 := 120; Y2 := 120; End;
                MoveTo(0,0);
                Line(X1,Y1,X2,Y2);
nd $08) Then    If (Mem[Segment:Offset-1] And $08) <> (Mem[Segment:Offset] A

                   If Mem[Segment:Offset] And $08 <>0 Then
                      Begin Y1 := 160; Y2 := 140; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 140; Y2 := 140; End
                   Else
                      Begin Y1 := 140; Y2 := 160; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 160; Y2 := 160; End
                Else
                   If Mem[Segment:Offset] And $08 <>0 Then
                      Begin Y1 := 140; Y2 := 140; End
                   Else
                      Begin Y1 := 160; Y2 := 160; End;
                MoveTo(0,0);
                Line(X1,Y1,X2,Y2);
nd $10) Then    If (Mem[Segment:Offset-1] And $10) <> (Mem[Segment:Offset] A

                   If Mem[Segment:Offset] And $10 <>0 Then
                      Begin Y1 := 200; Y2 := 180; MoveTo(0,0);
                            Line(X1,Y1,X1,Y2); Y1 := 180; Y2 := 180; End
                   Else
                      Begin Y1 := 180; Y2 := 200; MoveTo(0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Line(X1,Y1,X1,Y2); Y1 := 200; Y2 := 200; End
Else
   If Mem[Segment:Offset] And $10 <>0 Then
      Begin Y1 := 180; Y2 := 180; End
   Else
      Begin Y1 := 200; Y2 := 200; End;
MoveTo(0,0);
Line(X1,Y1,X2,Y2);
If (Mem[Segment:Offset-1] And $20) <> (Mem[Segment:Offset]A
nd $20) Then

   If Mem[Segment:Offset] And $20 <>0 Then
      Begin Y1 := 240; Y2 := 220; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 220; Y2 := 220; End
   Else
      Begin Y1 := 220; Y2 := 240; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 240; Y2 := 240; End
Else
   If Mem[Segment:Offset] And $20 <>0 Then
      Begin Y1 := 220; Y2 := 220; End
   Else
      Begin Y1 := 240; Y2 := 240; End;
MoveTo(0,0);
Line(X1,Y1,X2,Y2);
If (Mem[Segment:Offset-1] And $40) <> (Mem[Segment:Offset]A
nd $40) Then

   If Mem[Segment:Offset] And $40 <>0 Then
      Begin Y1 := 280; Y2 := 260; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 260; Y2 := 260; End
   Else
      Begin Y1 := 260; Y2 := 280; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 280; Y2 := 280; End
Else
   If Mem[Segment:Offset] And $40 <>0 Then
      Begin Y1 := 260; Y2 := 260; End
   Else
      Begin Y1 := 280; Y2 := 280; End;
MoveTo(0,0);
Line(X1,Y1,X2,Y2);
If (Mem[Segment:Offset-1] And $80) <> (Mem[Segment:Offset]A
nd $80) Then

   If Mem[Segment:Offset] And $80 <>0 Then
      Begin Y1 := 320; Y2 := 300; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 300; Y2 := 300; End
   Else
      Begin Y1 := 300; Y2 := 320; MoveTo(0,0);
            Line(X1,Y1,X1,Y2); Y1 := 320; Y2 := 320; End
Else
   If Mem[Segment:Offset] And $80 <>0 Then
      Begin Y1 := 300; Y2 := 300 End
   Else
      Begin Y1 := 320; Y2 := 320 End;
MoveTo(0,0);
Line (X1,Y1,X2,Y2);
Case M Of
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    0 : Begin Str((P1+(13.25*W)+(434*(I-1))):7:2,S1);Outtextx
y(50,325,S1); End;
                  100 : Begin Str((P1+2.65+(13.25*W)+(434*(I-1))):7:2,S2);Out
textxy(170,325,S2); End;
                  200 : Begin Str((P1+5.3+(13.25*W)+(434*(I-1))):7:2,S3);Outt
extxy(290,325,S3); End;
                  300 : Begin Str((P1+7.95+(13.25*W)+(434*(I-1))):7:2,S4);Out
textxy(410,325,S4); End;
                  400 : Begin Str((P1+10.6+(13.25*W)+(434*(I-1))):7:2,S5);Out
textxy(530,325,S5); End;
                  500 : Begin Str((P1+13.25+(13.25*W)+(434*(I-1))):7:2,S6);Ou
ttextxy(650,325,S6); End;
                 Else
                 End;
                 M := M+1;
            End;
            Outtextxy(280,335,'Press Any Key To Continue: ');
            Ch := ReadKey;
            If Ch <> #0 Then FuncKey := False Else
               Begin FuncKey := True; Ch := Readkey; End;
        End;
        ClearViewPort;
        CloseGraph;
    End;
    Begin
        V := 0;
        While (V = 0) Do
        Begin
            Memory;
            K := 0;
            While (K = 0) Do
            Begin
                Calculate;
                Scan;
                Writeln('Press "0" To See Another Range : Otherwise Press "
Any No." : ');
                Read(K);
            End;
            Writeln('Press "0" To Insert New Data : Or Press "Any No." To Qu
it : ');
            Read(V);
            For I := 1 To U Do FreeMem(Block[i],16384);
        End;
    End.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ทั้งนี้ก็ด้วยความช่วยเหลือจากบุคคลหลายๆฝ่าย ซึ่งต้องขอขอบคุณ อาจารย์ วันชัย ริ้วรุจา ที่ให้คำแนะนำปรึกษา ให้การสนับสนุนทางด้านอุปกรณ์ มาโดยตลอด

ขอขอบคุณคุณสุเจตน์ จันทรางค์ ที่ให้ความช่วยเหลือด้านแอสเซมเบลอร์ของ 8031 และขอขอบคุณทางภาควิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนโครงงาน ในด้านเครื่องมือและอุปกรณ์ ในการทดลอง ที่มีส่วนทำให้โครงงานนี้สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Handbook MCS 51 , Intel Corporation , 1985

2. Logic Databook Volume 2 National Semiconductor , 1984

3. Interface Bipolar LSI Bipolar Memory Prgrammable Logic Databook National Semiconductor , 1983

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้