ปีการศึกษา 2533

เครื่องควบคุมระยะไกลเอนกประสงค์แบบโปรแกรมได้
PROGRAMMABLE UNIVERSAL REMOTE CONTROL

โดย

วีรพล พัวพันธ์

อาพล โล่ห์ตระกูลวัฒน์

อิษฎ์ ศิริสวัสดิ์

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว.28801

13 มี.ค. 2534

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชาอิเล็คทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบระยะไกลเอนกประสงค์แบบโปรแกรมได้

ผู้จัดทำ

1. นาย วีรพล หัวพันธ์ เลขประจำตัว 30.1263
2. นาย อำพล โสภัตระกูลวัฒน์ เลขประจำตัว 30.1363
3. นาย อิษฎ์ ศิริสวัสดิ์ เลขประจำตัว 30.1364

..... อาจารย์ที่ปรึกษา

(ผศ. พลผดุง ผดุงกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028801

เครื่องควบคุมระยะไกลเอนกประสงค์แบบโปรแกรมได้

(Programmable Universal Remote Control)

นาย วีรพล พัวพันธ์ 30.1263
นาย อำพล โสหัตระกูลวัฒน์ 30.1363
นาย อิษฎ์ ศิริสวัสดิ์ 30.1364

ผศ.พลผดุง ผดุงกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2533

บทคัดย่อ

เครื่องควบคุมระยะไกลแบบโปรแกรมได้ (Programmable Universal Remote Control) เป็นเครื่องควบคุมระยะไกลแบบที่ใช้สัญญาณอินฟาเรดในการควบคุมอุปกรณ์ต่างๆ โดยสามารถบันทึกสัญญาณควบคุมของเครื่องควบคุมระยะไกลที่ใช้สัญญาณอินฟาเรดเครื่องอื่นๆได้ 64 สัญญาณ โดยนำเข้าไปเก็บในหน่วยความจำ ซึ่งมีการแสดงผลการทำงานของสัญญาณผ่านทางจอแสดงผล LCD ทั้งนี้ก็เพื่ออำนวยความสะดวกให้กับผู้ใช้ โดยทำการลดจำนวนเครื่องควบคุมระยะไกลจากหลายๆ เครื่องให้เหลือเพียงเครื่องเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programmable Universal Remote Control

Mr.Weerapol Poorphan 301263
Mr.Umphol Lohtrakoolwat 311363
Mr.Itt Sirisawat 301364

Advisor :
Assist.Prof Polphadung Phadungkul
Year 1990

ABSTRACT

Programmable Universal Remote Control is remote control that uses infrared-signal to control any other equipments, that has its' remote control,by recording other infrared signals. It can be recorded up to 64 signals for 8 modes each contains 8 functions and stores all signals in its' memory. It uses LCD display to show what modes and functions it is operating at that time. Its' advantages is to reduce many remote controls to one.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ในปัจจุบันนี้ เราสามารถกล่าวได้ว่าคงไม่มีใครที่ไม่เคยได้ใช้เครื่องควบคุมระยะ-ไกล(Remote Control) ทั้งนี้เพราะว่าเครื่องใช้ไฟฟ้าชนิดและรุ่นต่างๆที่มีจำหน่ายทั่วไปในท้องตลาดต่างก็มีเครื่องควบคุมระยะไกลเป็นอุปกรณ์หลักที่จำเป็นต้องใช้งานด้วยเสมอ เพื่ออำนวยความสะดวกแก่ผู้ใช้ ซึ่งเครื่องควบคุมระยะไกล 1 เครื่อง สามารถใช้ควบคุมเครื่องใช้ไฟฟ้าได้เพียง เครื่องเดียวเท่านั้น ดังนั้นจะเห็นได้ว่าถ้าเรามีเครื่องใช้ไฟฟ้า 2 ชนิดๆละ 1 เครื่อง เราจะมีเครื่องควบคุมระยะไกลถึง 2 เครื่อง ซึ่งก็อาจจะไม่เกิดปัญหาหรือความยุ่งยากใดๆในการใช้เครื่องควบคุมระยะไกล 2 เครื่อง แต่ถ้าเรามีเครื่องใช้ไฟฟ้าถึง 6 เครื่อง ในบริเวณเดียวกันเราก็จำเป็นที่จะต้องใช้เครื่องควบคุมระยะไกลถึง 6 เครื่องด้วยกัน ซึ่งจะเป็นการไม่สะดวกไม่ว่าจะเป็นการใช้งาน ดังเช่น อาจจะมีการผิดพลาดในการหยิบเครื่องควบคุมระยะไกลสลับเครื่องไม่ตรงกับชนิดของเครื่องใช้ไฟฟ้าที่จะใช้งาน เป็นต้น

ทางคณะผู้จัดทำและอาจารย์ที่ปรึกษา จึงได้พยายามหาวิธีที่จะนำมาแก้ไขปัญหาดังที่กล่าวในข้างต้น ซึ่งพบว่าเราจะสามารถแก้ปัญหาได้โดย การจัดทำเครื่องควบคุมระยะไกลที่สามารถใช้ควบคุมอุปกรณ์ต่างๆที่มีตามท้องตลาดได้หลายชนิด โดยให้ชื่อว่า เครื่องควบคุมระยะ-ไกลเอนกประสงค์แบบโปรแกรมได้ (Programmable Universal Remote Control) ซึ่งเครื่องนี้สามารถที่จะเรียนรู้และบันทึกสัญญาณจากเครื่องควบคุมระยะไกลอื่นๆไว้ในหน่วยความจำภายในตัวเอง และพร้อมที่จะใช้งานได้ทันทีหลังจากที่ผู้ใช้ได้บันทึกสัญญาณควบคุมต่างๆ เข้าเก็บไว้แล้ว

สุดท้ายนี้ ทางคณะผู้จัดทำหวังว่าโครงงานนี้คงเป็นประโยชน์ต่อบุคคลทั่วไป หากมีข้อผิดพลาดประการใด ทางคณะผู้จัดทำขออภัยไว้ ณ ที่นี้.

นาย วีรพล พัวพันธ์
นาย อำพล โสฬศกุลวัฒน์
นาย อิษฎ์ ศิริสวัสดิ์
คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

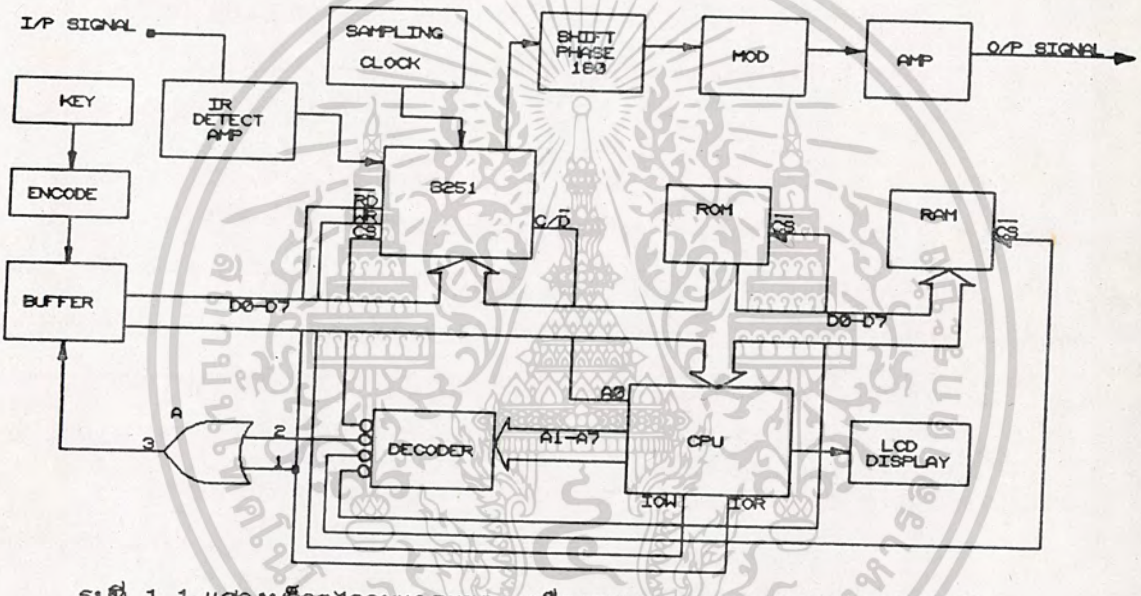
สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก-ข
คำนำ	ค
สารบัญ	ง
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการพื้นฐานของโครงงาน	3
บทที่ 3 การออกแบบและการสร้าง	28
บทที่ 4 การทดลองและผลการทดลอง	36
บทที่ 5 สรุปผลและวิจารณ์	40
ภาคผนวก	
-วงจรที่ใช้ในการทดลอง	42
กิตติกรรมประกาศ	43
หนังสืออ้างอิง	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เครื่องควบคุมระยะไกลแบบโปรแกรมได้ (Programmable Universal Remote Control)นี้เป็นเครื่องควบคุมระยะไกลที่ใช้ไมโครโปรเซสเซอร์เป็นส่วนควบคุม ซึ่งเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้สามารถทำงานได้โดยไม่ต้องมีการต่อเชื่อมกับอุปกรณ์อื่นภายนอกอีก (STAND-ALONE) หรือเหมือนกับเครื่องควบคุมระยะไกลที่ใช้กันอยู่ทั่วไป เราสามารถเขียนบล็อคไดอะแกรมแสดงโครงสร้างและการทำงานทั้งหมดของเครื่องควบคุมระยะไกลแบบโปรแกรมได้อย่างคร่าวๆ ดังรูปที่ 1



รูปที่ 1.1 แสดงบล็อคไดอะแกรมของเครื่องควบคุมระยะไกลแบบโปรแกรมได้

จากรูปที่ 1.1 จะเห็นได้ว่าโครงสร้างของเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้ประกอบไปด้วยส่วนต่างๆ ดังนี้:

1. <u>หน่วยประมวลผล</u>

-เป็นส่วนที่ควบคุมและประมวลผลการทำงานของระบบ ซึ่งได้ใช้ไมโครโปรเซสเซอร์ที่มีการประมวลผลแบบ 8 บิท

2. <u>หน่วยความจำในตัว</u>

-หน่วยความจำแบบอ่านได้อย่างเดียว(ROM) ใช้เป็นส่วนที่เก็บโปรแกรมควบคุมการทำงานทั้งหมดของระบบ

-หน่วยความจำแบบอ่านและเขียนได้ (RAM) ใช้เป็นส่วนที่เก็บข้อมูล

3. <u>หน่วยสร้างความถี่และสัญญาณนาฬิกา</u> ซึ่งประกอบด้วย;

-วงจรสร้างสัญญาณที่ใช้สำหรับสุ่มในการเก็บข้อมูล (Sampling Clock) ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

มีความถี่ (Sampling Rate) เท่ากับ 22 กิโลเฮิรตซ์(KHz)

-วงจรสร้างสัญญาณที่ใช้เป็นความถี่พาหะ (Carrier Frequency) ในการมอดดูเลท (Modulate) กับข้อมูลที่ส่งออกไป มีความถี่เท่ากับ 31 กิโลเฮิรตซ์(KHz)

4. <u>หน่วยติดต่อสื่อสารข้อมูลแบบอนุกรม</u>

-เป็นส่วนที่ทำหน้าที่ติดต่อสื่อสารข้อมูลแบบอนุกรมระหว่างหน่วยรับ-ส่งสัญญาณควบคุมกับส่วนประมวลผล ซึ่งจะใช้ความถี่ในการสุ่มข้อมูล (Sampling Rate) เท่ากับ 22 KHz

5. <u>หน่วยรับ-ส่งสัญญาณควบคุม</u>

-เป็นส่วนที่ทำหน้าที่ในการรับ-ส่งสัญญาณควบคุมในช่วงความยาวคลื่นแสงอินฟราเรด (ประมาณ 850-1000 นาโนเมตร)

6. <u>หน่วยแสดงผล</u>

-เป็นส่วนที่ทำหน้าที่แสดงฟังก์ชั่นการทำงานบนจอ LCD (โดยบอกว่าขณะนี้ระบบกำลังทำงานในฟังก์ชั่นใดอยู่)

สำหรับภาคจ่ายไฟที่ใช้ป้อนให้กับระบบต่างๆภายในเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้ จะใช้ไฟจากแบตเตอรี่ขนาดเล็ก (size AA) จำนวน 4 ก้อน (5โวลท์) เนื่องจากการใช้แบตเตอรี่เป็นภาคจ่ายไฟ ทำให้ต้องพยายามใช้อุปกรณ์ที่ใช้ไฟน้อย เพื่อเป็นการประหยัดพลังงานจากแบตเตอรี่นั่นเอง ดังนั้นอุปกรณ์ทุกตัวที่ใช้ในโครงงานนี้จึงเป็นแบบซีมอส (เนื่องจากอุปกรณ์ที่เป็นแบบซีมอสนี้จะใช้กระแสไฟน้อยกว่าแบบทีทีแอล)

ซึ่งในบทต่อๆไปจะกล่าวถึงการทำงานและวงจรในแต่ละส่วนของเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการพื้นฐานของโครงงาน

สำหรับบทที่ 2 นี้จะขอกล่าวถึงทฤษฎีและหลักการพื้นฐานที่สำคัญในส่วนต่างๆ ของโครงงาน ซึ่งเป็นสิ่งที่สำคัญที่ช่วยคณะผู้จัดทำในการออกแบบ และการคำนวณสร้างระบบการทำงานทั้งหมดของโครงงาน ทฤษฎีและหลักการพื้นฐานในส่วนต่างๆ ดังนี้

2.1 ความรู้พื้นฐานเกี่ยวสัญญาณอินฟราเรด

2.2 โครงสร้างและหลักการทำงานของ Z80 ไมโครโปรเซสเซอร์

2.3 โครงสร้างและหลักการทำงานภาครับส่งข้อมูลแบบอนุกรม

2.4 โครงสร้างและหลักการทำงานภาครับส่งข้อมูลแบบขนาน

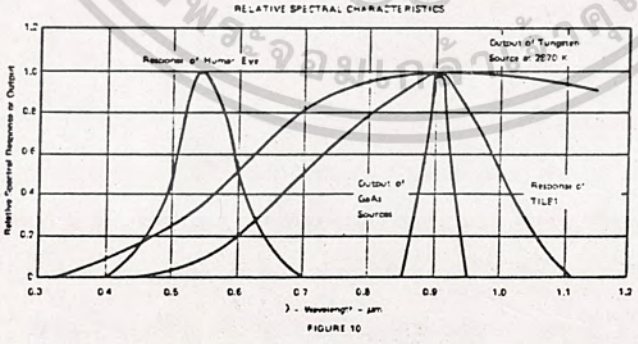
2.5 โครงสร้างและหลักการโปรแกรมหน่วยแสดงผล

2.6 โครงสร้างส่วนสร้างสัญญาณนาฬิกา

และจะขอกล่าวรายละเอียดเป็นลำดับดังต่อไปนี้

2.1 ความรู้พื้นฐานเกี่ยวกับสัญญาณอินฟราเรด (Introduction to Infrared Signal)

สัญญาณอินฟราเรดนั้นเป็นส่วนหนึ่งของสเปคตรัมแสง(Light Spectrum) ซึ่งจะอยู่ในย่านความยาวคลื่นของแสงที่ตามนุษย์ไม่สามารถที่จะมองเห็นได้ เนื่องจากสัญญาณอินฟราเรดนี้จะมีค่าความยาวคลื่นประมาณ 850 - 1000 นาโนเมตร ในขณะที่ตามนุษย์สามารถที่จะมองเห็นได้ที่แสงที่มีค่าความยาวคลื่นในช่วงประมาณ 400 - 700 นาโนเมตร ดังรูปที่ 2.1.1



รูปที่ 2.1.1 แสดงลักษณะคลื่นแสงที่ค่าความยาวคลื่นค่าต่างๆ

สำหรับสัญญาณในเครื่องควบคุมระยะไกล มีหลายแบบให้ดู ดังต่อไปนี้

1. เครื่องควบคุมระยะไกลที่ใช้สัญญาณอินฟราเรด
2. เครื่องควบคุมระยะไกลที่ใช้สัญญาณอุลตร้าโซนิก
3. เครื่องควบคุมระยะไกลที่ใช้สัญญาณคลื่นวิทยุ

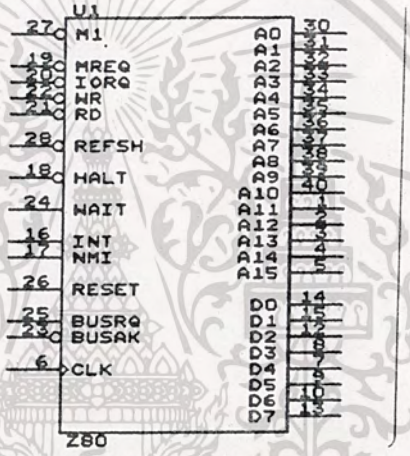
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในที่นี้ จะขอกล่าวแต่ในแบบแรกเท่านั้น จะเห็นได้ว่าถ้าเป็นเครื่องควบคุมระยะไกลที่ใช้สัญญาณอินพราเรดในการส่งข้อมูลนั้น อุปกรณ์ที่จะติดต่อสื่อสารกับเครื่องควบคุมระยะไกลดังกล่าวจะต้องอุปกรณ์ที่สามารถส่งหรือตรวจจับสัญญาณแบบอินพราเรดด้วยเช่นกัน ดังเช่น TIL38, TIL39, TIL100, MRD821 เป็นต้น

สัญญาณที่ได้จากเครื่องควบคุมระยะไกลที่ใช้สัญญาณอินพราเรดนั้น จะมีลักษณะเป็นพัลส์ที่มีความกว้างต่างๆกัน ส่งออกมาในรูปแบบอนุกรม(คือข้อมูลจะเรียงกันตามลำดับ)จะได้เป็นชุดข้อมูลกัน

2.2 โครงสร้างและหลักการทำงาน Z80 ไมโครโปรเซสเซอร์

2.2.1) โครงสร้างทางสถาปัตยกรรม



รูปที่ 2.2.1 แสดงลักษณะการจัดขาสัญญาณต่างๆของ Z80

รายละเอียดและหน้าที่ของขาสัญญาณต่างๆ มีดังนี้

A0-A15 เป็นสายสัญญาณที่ Z80 ใช้ในการอ้างตำแหน่งต่างๆ ในหน่วยความจำหรือตำแหน่งของอุปกรณ์อินพุตเอาต์พุต(I/O Device) มีทั้งสิ้น 16 สาย

D0-D7 เป็นสายสัญญาณที่ Z80 ใช้ในการเคลื่อนย้ายและรับส่งข้อมูลระหว่างหน่วยความจำหรืออุปกรณ์อินพุตเอาต์พุตกับ Z80 มีทั้งสิ้น 8 สาย

$\overline{M1}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อบอกให้ทราบว่ายังอยู่ในสภาวะการเฟทช์ (fetch)

$\overline{MREQ}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อขอติดต่อกับหน่วยความจำในการรับและส่งข้อมูล

$\overline{IORQ}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อขอติดต่อกับอุปกรณ์อินพุตเอาต์พุต ในการรับและส่งข้อมูล

$\overline{RD}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อขอรับข้อมูลทั้งจากหน่วยความจำหรืออุปกรณ์อินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น โดยไม่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้เขียน ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{WR}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อขอส่งข้อมูลทั้งจากหน่วยความจำหรืออุปกรณ์อินพุต เอาต์พุต

$\overline{RFSH}$ เป็นสายสัญญาณ Z80 ส่งออกมาเพื่อบอกให้ทราบว่า Z80 กำลังส่งข้อมูลที่ใช้ในการทำรีเฟรช (Refresh) ข้อมูลในหน่วยความจำชนิดไดนามิก (Dynamic)

$\overline{HALT}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาในขณะที่ Z80 หยุดการทำงาน หรือ Z80 กำลังทำคำสั่งฮอลท์ (HALT)

$\overline{WAIT}$ เป็นสายสัญญาณอินพุต และเป็นสัญญาณที่หน่วยความจำหรืออุปกรณ์อินพุต เอาต์พุตส่งออกมาบอก Z80 ว่ายังไม่พร้อมที่จะรับส่งข้อมูลกับ Z80 และจะรอจนกว่าสัญญาณนี้หมดสิ้นแล้วจึงทำการรับส่งข้อมูล

$\overline{INT}$ เป็นสายสัญญาณอินพุต และเป็นสัญญาณที่อุปกรณ์อินพุต เอาต์พุตส่งมาให้ Z80 เพื่อที่อุปกรณ์อินพุต เอาต์พุตขอทำการอินเตอร์รัพท์ (Interupt) แบบ "มาสเคเบิลอินเตอร์รัพท์" (Maskable Interupt)

RESET เป็นสายสัญญาณอินพุตที่ใช้ในการรีเซต (Reset) Z80 ให้เริ่มการทำงานใหม่ หรือใช้ทำให้โปรแกรมเคาน์เตอร์ (Program Counter) มีค่าเป็นศูนย์

$\overline{NMI}$ เป็นสายสัญญาณอินพุต และเป็นสัญญาณที่อุปกรณ์อินพุต เอาต์พุตส่งมาให้ Z80 เพื่อที่อุปกรณ์อินพุต เอาต์พุตขอทำการอินเตอร์รัพท์(Interupt) แบบ "นอนมาสเคเบิลอินเตอร์รัพท์" (Non-Maskable Interupt)

$\overline{BUSRQ}$ เป็นสายสัญญาณอินพุต และเป็นสัญญาณที่อุปกรณ์อินพุต เอาต์พุตส่งมาให้ Z80 เพื่อขอใช้บัส (Bus) ในการเคลื่อนย้ายข้อมูลกับหน่วยความจำโดยไม่ผ่าน Z80

$\overline{BUSAK}$ เป็นสายสัญญาณเอาต์พุต และเป็นสัญญาณที่ Z80 ส่งออกมาเพื่อบอกว่าในขณะนั้น Z80 ไม่ได้ใช้บัสแล้ว

2.2.2) <u>รีจิสเตอร์ต่างๆ ภายใน Z80</u>

ภายใน Z80 นั้นจะประกอบด้วยรีจิสเตอร์ภายในที่เป็นขนาด 8 บิท 18 ตัว รีจิสเตอร์ขนาด 16 บิท 4 ตัว รูปโครงของรีจิสเตอร์ภายใน Z80 ซึ่งสามารถเขียนเป็นแผนผัง ได้ดังรูปที่ 2.2.2

A′	F′
B′	C′
D′	E′
H′	L′

A	F
B	C
D	E
H	L
อินเด็กซ์รีจิสเตอร์ (IX)	
อินเด็กซ์รีจิสเตอร์ (IY)	
สแตกพอยน์เตอร์ (SP)	
โปรแกรมเคาน์เตอร์ (PC)	
I	
R	

รูปที่ 2.2.2 โครงสร้างรีจิสเตอร์ภายในของ Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรมเคาท์เตอร์ (Program Counter,PC)

เป็นรีจิสเตอร์สำหรับเก็บแอดเดรส มีหน้าที่ที่ชี้ตำแหน่งของคำสั่งในหน่วยความจำ เพื่อให้ซีพียูเฟทช์คำสั่งนั้นได้อย่างถูกต้อง หลังจากนั้นแล้วค่าในโปรแกรมเคาท์เตอร์จะเพิ่มขึ้นอย่างอัตโนมัติ

2. สแตคพอยท์เตอร์ (Stack Pointer,SP)

เป็นรีจิสเตอร์สำหรับซีพียูใช้เป็นตัวชี้ไปยังหน่วยความจำ เมื่อซีพียูมีการกระทำคำสั่ง PUSH หรือ POP ก็จะเป็นการเขียนหรืออ่านค่า จากตำแหน่งที่ สแตคพอยท์เตอร์ชี้

3. อินเดกซ์ รีจิสเตอร์ (Index Register,IX,IY)

เป็นรีจิสเตอร์อิสระ โดยปกติจะใช้เป็นฐานในการชี้ไปยังแอดเดรสของหน่วยความจำที่มีลักษณะแบบที่มีค่าออฟเซต (Offset) เพื่อการเขียนหรืออ่านข้อมูลลงในหน่วยความจำ

4. อินเทอร์รัพท์เพจแอดเดรส รีจิสเตอร์ (Interrupt Page Register,I)

เป็นรีจิสเตอร์ที่ใช้ในการกระโดดไปทำโปรแกรมอื่น ในขณะที่มีการอินเทอร์รัพท์

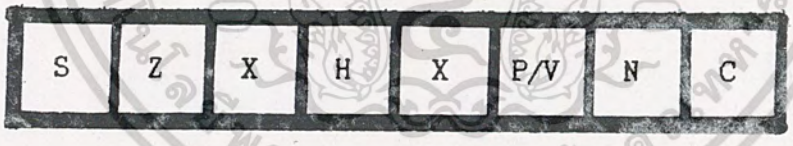
5. รีจิสเตอร์ รีเฟรชหน่วยความจำ (Refresh Memory Register,R)

เป็นรีจิสเตอร์ที่ใช้ในการ รีเฟรชหน่วยความจำ จากซีพียู

6. รีจิสเตอร์ ใช้งานทั่วไป

เป็นรีจิสเตอร์ที่ใช้งานทั่วไปได้แก่ A,BC,DE,HL

หมายเหตุ สำหรับรีจิสเตอร์ F ใช้แสดงสภาวะต่าง ๆ ของซีพียูดังนี้



C : การแสดงสภาวะของตัวทด หรือตัวยืม จากการคำนวณ

N : การแสดงสภาวะการบวก หรือลบ ของซีพียู

P/V : การแสดงสภาวะของพาริตี้(Parity) และ โอเวอร์ โฟลว์(Over Flow)

Z : การแสดงสภาวะการเป็น 0 ของผลลัพท์จากการคำนวณ

S : การแสดงสภาวะของ เครื่องหมายจากการคำนวณ

H : การแสดงสภาวะของตัวทดช่วย

X : เป็นแฟลกที่ไม่ได้ใช้งาน

2.2.3) อินพุต เอาท์พุตของซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติแล้วซีพียูจะมีการติดต่อกับอุปกรณ์ภายนอก (IO Device) นอกเหนือจากการติด

ต่อกับหน่วยความจำ ดังนั้นย่อมต้องมีการอ้างตำแหน่งของอุปกรณ์ภายนอก เช่นเดียวกับที่อ้างแอดเดรสในหน่วยความจำ โดยระบบของ Z80 จะใช้แอดเดรสในการอ้างอิงอุปกรณ์ภายนอก 8 เส้น (A0-A7) ซึ่งจะอ้างได้ทั้งหมด 256 ตำแหน่ง ซึ่งค่าการอ้างอิงแต่ละตำแหน่งนั้นจะได้จากการถอดรหัสจากวงจรถอดรหัส

สัญญาณที่ใช้ในการควบคุมอุปกรณ์ภายนอกได้แก่ IORQ, RD, WR

<u>ลำดับการเขียนข้อมูลจากพอร์ทเอาท์พุต</u>

ในการส่งข้อมูลจากซีพียูไปพอร์ทเอาท์พุต จะมีลำดับการทำงานดังนี้

1. Z80 ส่งหมายเลขพอร์ท เพื่อใช้เป็นตำแหน่งอ้างอิง
2. ซีพียูจะส่งข้อมูลที่ต้องการออกมาที่ คาต้าบัส (D0-D7)
3. การส่งสัญญาณ $\overline{\text{IORQ}}$ และ $\overline{\text{WR}}$ จาก Z80 เพื่อแสดงว่าซีพียูต้องการติดต่อกับอุปกรณ์ภายนอก
4. ข้อมูลก็จะถูกส่งไปยังที่พอร์ทเอาท์พุตได้ยังที่ต้องการ

<u>ลำดับการอ่านข้อมูลจากพอร์ตอินพุต</u>

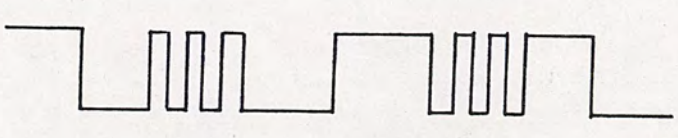
ในการอ่านข้อมูลจากพอร์ทอินพุตไปยังซีพียูจะมีลำดับการทำงานดังนี้

1. Z80 ส่งหมายเลขพอร์ท เพื่อใช้เป็นตำแหน่งอ้างอิง
2. การส่งสัญญาณ $\overline{\text{IORQ}}$ และ $\overline{\text{RD}}$ จาก Z80 เพื่อแสดงว่าซีพียูต้องการติดต่อกับอุปกรณ์ภายนอก
3. ข้อมูลจากพอร์ทอินพุตจะส่งมายังซีพียูได้ตามที่ต้องการ

2.3 โครงสร้างและหลักการทำงานสำหรับส่งข้อมูลแบบอนุกรม (Serial Communication)

<u>การรับส่งข้อมูลแบบอนุกรม</u>

สัญญาณข้อมูลที่ใช้กันทั่วๆไป ของเครื่องควบคุมระยะไกล ที่ใช้ความถี่ใต้แดงจะมีลักษณะเป็นพัลซ์ (pulse)ซึ่งจะมีลักษณะดังนี้



รูปที่ 2.3.1 แสดงลักษณะของสัญญาณควบคุมที่ใช้ความถี่ใต้แดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะเห็นได้ว่า เป็นลักษณะของข้อมูลแบบอนุกรมดังนั้นในการรับส่งสัญญาณควบคุมของเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้ จึงใช้ไอซีที่สำหรับติดต่อสื่อสารแบบอนุกรม ซึ่งมีอยู่

หลายเบอร์ เช่น Z-80 SIO, 8251A เป็นต้น แต่ในโครงงานนี้ได้ใช้ไอซีเบอร์ 8251A ดังนั้น จะขออธิบายเกี่ยวกับไอซีเบอร์นี้เท่านั้น

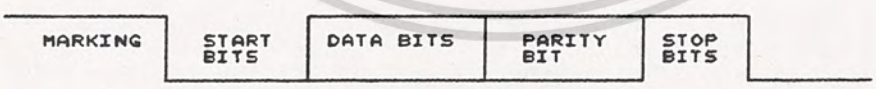
ฮาร์ดแวร์และการต่อขาต่างๆ ของ 8251

ไอซีเบอร์ 8251 นี้ เป็นไอซีที่ใช้ในการติดต่อสื่อสารข้อมูลแบบอนุกรม ซึ่งจะถูกควบคุมจากไมโครโปรเซสเซอร์ผ่านทางซอฟท์แวร์ ในการส่งข้อมูลไอซีตัวนี้จะรับข้อมูลแบบขนาน มาจากซีพียูและทำการเปลี่ยนข้อมูลให้เป็นแบบอนุกรมก่อนที่จะส่งออกไป ทำนองเดียวกันในการรับข้อมูล ไอซีตัวนี้จะทำการรับข้อมูลแบบอนุกรมเข้ามา และทำการเปลี่ยนข้อมูลให้เป็นแบบขนานก่อนที่จะส่งข้อมูลให้ตัวซีพียูนำไปประมวลผลต่อไป

การรับส่งข้อมูลแบบอนุกรมของ 8251 สามารถทำได้ 2 แบบ(mode)คือ

1) Asynchronous Mode

คือการรับส่งข้อมูลที่เวลาระหว่างข้อมูลแต่ละชุดมีค่าไม่แน่นอน CPU จะรับรู้ว่าข้อมูลช่วงไหนเป็นข้อมูลที่จะนำไปใช้งานได้ ก็ต่อเมื่อมีตัวบอกให้ CPU ทราบว่าขณะนี้ ถึงจุดเริ่มต้นของข้อมูลแล้วและขณะนี้ข้อมูลจบสิ้นแล้ว นั่นก็คือถึงข้อมูลแต่ละชุดที่ต้องการจะให้ CPU นำไปใช้งาน จะต้องมีบิตเริ่มต้นของข้อมูล(start bit) และบิตสุดท้ายหรือบิตสิ้นสุดของข้อมูล (stop bit) โดยปกติแล้ว บิตเริ่มต้นจะมีสถานะเป็น 0 (low) ขนาด 2 บิตติดต่อกัน (ทั้งนี้เนื่องจากในขณะที่ไม่มีการรับส่งข้อมูลขาที่ใช้รับข้อมูลของ 8251 จะมีสถานะเป็น (high) หรือ Marking ตลอด) ส่วนบิตสุดท้ายของข้อมูลจะมีสถานะเป็น 1 (high) ขนาด 1 หรือ 1.5 หรือ 2 บิต (ขึ้นอยู่กับการกำหนดของผู้ใช้ในการรับส่งข้อมูล) ลักษณะของข้อมูลที่ใช้ในการรับส่งแบบ Asynchronous สามารถแสดงได้ ดังนี้



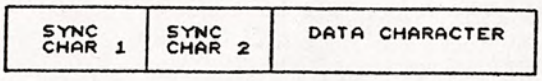
รูปที่ 2.3.2 แสดงลักษณะของข้อมูลที่ใช้ในการรับส่งแบบ Asynchronous

2) Synchronous Mode

คือการรับส่งข้อมูลที่เวลาระหว่างข้อมูลแต่ละชุดมีค่าแน่นอน ดังนั้นจึงไม่จำเป็นจะต้องมีบิตเริ่มต้นหรือ บิตสุดท้ายของข้อมูล เพื่อบอกให้ CPU รู้ช่วงของข้อมูลที่จะนำไปใช้งาน แต่อย่างไรก็ตาม เราจำเป็นจะต้องบอกให้ CPU ทราบถึงจุดเริ่มต้นของข้อมูล โดยจุดเริ่มต้นของข้อมูลในที่นี้ไม่ได้หมายถึงตัวบิตเริ่มต้นใน Asynchronous mode แต่หมายถึงจุดที่บอกการเริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูลโดยจะบอกในลักษณะ ที่เรียกว่า Sync charactor โดยจะมีลักษณะของข้อมูลที่ใช้ในการรับส่งดังรูป



รูปที่ 2.3 แสดงลักษณะของข้อมูลที่ใช้ในการรับส่งแบบ Synchronous

สำหรับโครงงานนี้ได้ใช้การรับส่งข้อมูลแบบอนุกรมแบบ Synchronous ทั้งนี้เนื่องจากเราไม่สามารถที่จะรู้ได้ว่า สัญญาณควบคุมที่เราจะรับเข้ามาเก็บไว้ในหน่วยความจำของเครื่องของมีการส่งแบบไหน และถ้ามีการส่งแบบ Asynchronous เราก็ไม่สามารถจะทราบได้ว่า บิตเริ่มต้นและบิตสุดท้ายของข้อมูลมีลักษณะเป็นอย่างไร และอยู่ตรงไหน ทั้งนี้เพื่อจะทำให้เครื่องควบคุมระยะไกลแบบโปรแกรมได้เครื่องนี้ สามารถที่จะรับสัญญาณควบคุมจากเครื่องคุมระยะไกลเครื่องอื่นๆ ได้ทุกแบบและทุกบริษัทที่ผลิต

เนื่องจากการรับส่งข้อมูลแบบ Synchronous นี้จะต้องมี Sync Character เป็นตัวบอกให้ CPU ทราบว่าถึงช่วงของข้อมูลที่จะต้องนำไปใช้งานแล้ว ซึ่ง Sync Character ของ 8251 มีอยู่ 2 แบบคือ 1. Internal Sync Character (OUTPUT)

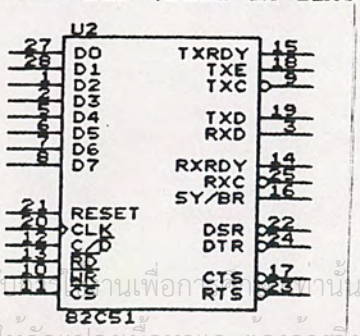
2. External Sync Character (INPUT)

ในโครงงานนี้จะใช้ Sync Charactor แบบ External Sync Character

เนื่องจากในโครงงานนี้ใช้การรับส่งข้อมูลแบบ Synchronous และมี External Sync Character ดังนั้นในส่วนที่เหลือของรายงานฉบับนี้จะอธิบายเฉพาะการรับส่งข้อมูลแบบนี้เท่านั้น

<u>สัญญาณและการต่อใช้งาน (Interface) 8251</u>

ไอซีเบอร์ 8251 นี้เป็นไอซีแบบ DIP (Dual in line Package)ซึ่งมีจำนวนขาเท่ากับ 28 ขา โดยมีลักษณะดังรูป

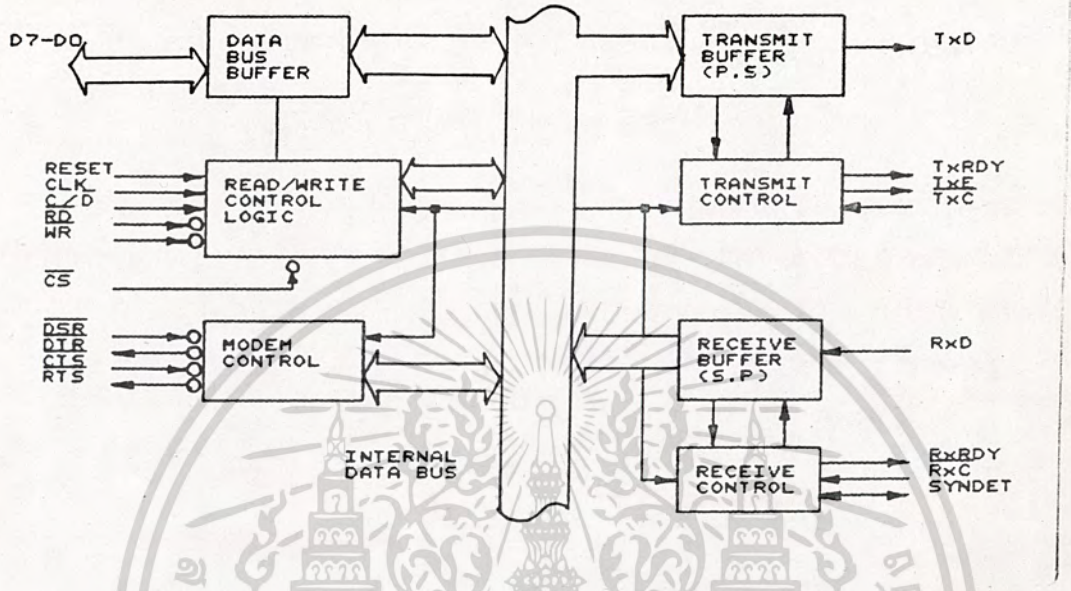


รูปที่ 2.3.4 แสดงรูปร่างและความหมายของแต่ละขาของ 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028801

และมีโครงสร้างการทำงานภายในตัว 8251 ซึ่งสามารถเขียนเป็น Block Diagram ได้ดังรูป



รูปที่ 2.3.5 แสดง Block Diagram ของโครงสร้าง 8251

ก่อนอื่นจะขออธิบายถึงการต่อใช้งาน 8251 เฉพาะที่ใช้ในโครงงานนี้ โดยจะอธิบายที่ละขาของ 8251 ดังนี้

1.ขา Reset:ขาที่ 21

เป็นขาที่ใช้ Reset 8251 โดยถ้าสัญญาณที่ขานี้มีสถานะเป็น 1(high) ก็จะทำให้ 8251 อยู่ในโหมดนี้เรียกว่า ไอเดิล (idle mode) และจะคงอยู่ในโหมดนี้ไปเรื่อยๆ จนกว่าจะมีคำสั่งควบคุม (control words) ชุดใหม่ส่งมาควบคุม(โปรแกรม) ตัว8251 ให้เริ่มทำงานใหม่

ในโครงงานนี้ได้ต่อขา Reset ของ8251เข้ากับขา Reset ของ CPU โดยผ่าน invertor 1 ตัว (สัญญาณ Reset ของCPU กับ 8251มีสถานะตรงข้ามกัน) ทั้งนี้เพราะต้องการให้ 8251 มีการ Resetพร้อมกับ CPU

2.ขา CLK (Clock):ขาที่ 20

เป็นขาอินพุทที่ใช้ป้อนสัญญาณนาฬิกาให้กับ8251 โดยสัญญาณนาฬิกาที่ป้อนให้กับ 8251 นี้จะต้องมีความถี่ไม่เกิน 3 MHz และจะต้องมากกว่าอัตราการรับส่งข้อมูลของ 8251 ประมาณ 30 เท่าหรือมากกว่านั้น

ในโครงงานนี้ได้ต่อขาCLKของ 8251 เข้ากับขาCLKของ CPUโดยตรง เพราะ CLK ที่ CPU ใช้อยู่มีความถี่ของสัญญาณนาฬิกาประมาณ 2 MHz เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ขา $\overline{WR}$ (Write):ขาที่ 10

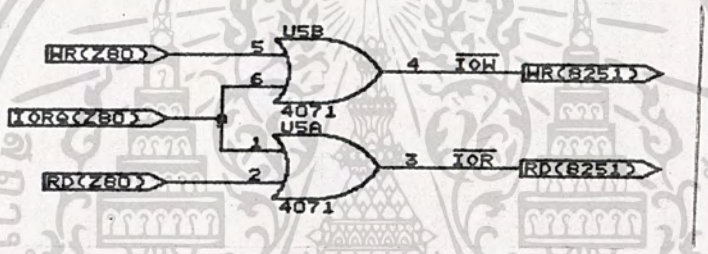
เมื่อขาอินพุทขานี้มีสถานะเป็น 0 (low) แสดงว่าในขณะนั้น CPU ต้องการที่จะส่งข้อมูลออกผ่านทาง8251

ในโครงงานนี้ ได้ต่อขา $\overline{WR}$ ของ8251เข้ากับสัญญาณ $\overline{IOW}$ ที่มาจาก CPU

4.ขา $\overline{RD}$(Read):ขาที่ 13

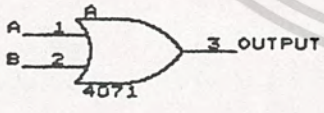
เมื่อขาอินพุทขานี้มีสถานะเป็น 0(low) แสดงว่าในขณะนั้น CPU ต้องการที่จะอ่านข้อมูลเข้าโดยผ่านทาง 8251 ดังนั้นจะเห็นได้ว่าขา $\overline{WR}$ และ $\overline{RD}$ จะมีสถานะเป็น 0 พร้อมกันไม่ได้

ในโครงงานนี้ได้ต่อขา $\overline{RD}$ ของ8251 เข้ากับสัญญาณ $\overline{IOR}$ ที่มาจาก CPU โดยการสร้างสัญญาณ $\overline{IOW}$ และ $\overline{IOR}$ ที่มาจาก CPU สามารถทำได้โดยใช้ OR GATE เข้าช่วยดังรูป



รูปที่ 2.3.6 แสดงการสร้างสัญญาณ $\overline{IOR}$, $\overline{IOW}$ ให้แก่ 8251

ทั้งนี้เนื่องจาก OR GATE มี Truth Table ดังรูป



I/P		
A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	1

ตารางที่ 2.3.1 แสดงค่าทางลอจิกของ OR-GATE

5.ขา $\overline{C/D}$ (Control/Data):ขาที่ 12

ขาอินพุทขานี้จะเป็นขาที่ทำงานร่วมกับขา $\overline{WR}$ และ $\overline{RD}$ เพื่อที่จะบอกให้ 8251 รู้ว่าข้อมูลนี้อยู่บนบัสข้อมูล ในขณะที่ CPU ทำการติดต่อกับ 8251 นี้เป็นข้อมูลที่จะนำไปใช้งาน (data charactor) คำสั่งควบคุม (control word) หรือตัวบอกสถานะของ 8251 ในขณะนั้น (st-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

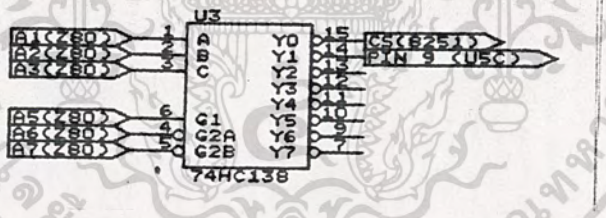
atus Information) โดยถ้าขาอินพุทขานี้มีสถานะเป็น 1(high) ก็แสดงว่าข้อมูลที่อยู่บนบัสข้อมูลในขณะนั้นเป็นคำสั่งควบคุมหรือตัวบอกสถานะ แต่ถ้าขานี้มีสถานะเป็น 0(low) ก็จะแสดงว่าข้อมูลที่อยู่บนบัสข้อมูล ในขณะนั้นเป็นข้อมูลที่จะนำไปใช้งาน

โดยปกติแล้ว จะต่อขา C/$\overline{D}$ ของ 8251 เข้ากับขาแอดเดรสที่ 0(A_0) ซึ่งในโครงงานนี้ก็ต่อเช่นนี้เหมือนกัน ซึ่งจะเห็นได้ว่าตัว 8251 ตัวเดียวจะมีหมายเลขพอร์ท (port NO.) ถึง 2 หมายเลข โดยจะเป็นพอร์ทควบคุม 1 หมายเลข และ พอร์ทข้อมูลอีก 1 หมายเลข

6.<u>ขา $\overline{CS}$ (Chip Select):ขาที่ 11</u>

เป็นขาอินพุทที่ใช้บอก 8251 ว่า ในขณะนี้ CPU ต้องการที่จะติดต่อใช้งานกับ 8251 โดยขานี้จะ active ที่สถานะ 0(low) ดังนั้นถ้าขานี้มีสถานะเป็น 1(high) ก็จะทำให้สัญญาณ RD หรือ WR ไม่มีผลต่อตัว 8251 (ไม่มีการรับส่งข้อมูลผ่าน 8251) และยังมีผลทำให้บัวข้อมูลมีสถานะลอยตัว(float state)อีกด้วย

ในโครงงานนี้ได้ใช้ไอซีเบอร์ 74HC138 เป็นตัวถอดรหัส(Decode)หมายเลขพอร์ท โดยที่ในการทำโครงงานนี้ได้เลือกพอร์ทหมายเลข 030H เป็นพอร์ทข้อมูลของ 8251 และพอร์ทหมายเลข 031H เป็นพอร์ทควบคุมของ 8251 ซึ่งจะมีลักษณะการต่อใช้งานดังนี้



รูปที่ 2.3.7 แสดงลักษณะการต่อใช้งานของ Decoder

จากรูป จะเห็นได้ว่าเราไม่ได้ต่อ A_0 เข้าที่ตัวถอดรหัส ทั้งนี้เพราะว่าเราใช้ A_0 เป็นส่วนที่เลือกว่าจะใช้พอร์ทข้อมูลหรือพอร์ทควบคุม โดยที่ต่อ A_0 เข้าที่ขา C/$\overline{D}$ ของ 8251 ดังนั้น เอ้าท์พุทของตัวถอดรหัสแต่ละเอ้าท์พุทจึงสามารถถอดรหัสหมายเลขพอร์ทได้ทีละ 2 หมายเลข คือ.

$\overline{Y_0}$	=	030H,031H	$\overline{Y_4}$	=	038H,039H
$\overline{Y_1}$	=	032H,033H	$\overline{Y_5}$	=	03AH,03BH
$\overline{Y_2}$	=	034H,035H	$\overline{Y_6}$	=	03CH,03DH
$\overline{Y_3}$	=	036H,037H	$\overline{Y_7}$	=	03EH,03FH

จากที่อธิบายถึงลักษณะการใช้งานของขา $\overline{RD}$,$\overline{WR}$,C/$\overline{D}$,$\overline{CS}$ มาแล้วนั้น จะเห็นได้ว่าโดยปกติแล้วที่ขาอินพุททั้ง 4 ขานี้จะมีการทำงานร่วมกัน โดยสามารถที่จะสรุปได้ดังตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{C/D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	ความหมาย
0	0	1	0	เป็นการอ่านข้อมูลจาก 8251 ลงสู่ data bus
0	1	0	0	เป็นการเขียนข้อมูลจาก data bus ลงสู่ 8251
1	0	1	0	เป็นการอ่านค่าในสเตตัสรีจิสเตอร์ของ 8251 ลงสู่ data bus
1	1	0	0	เป็นการอ่านข้อมูลจาก data bus มาใช้เป็น ข้อมูลที่ใช้ควบคุม
-	1	1	0	data bus ที่ 8251 จะมีสภาพเป็น ไตร-สเตท
-	-	-	1	data bus ที่ 8251 จะมีสภาพเป็น ไตร-สเตท

ตารางที่ 2.3.2 แสดงการทำงานร่วมกันของสัญญาณ C/D,RD,WR,CS

7. ขา $\overline{CTS}$ (Clear to Send):PIN 17

ถ้าอินพุทขานี้มีสถานะลอจิกเป็น "0" (low) ก็จะทำให้ 8251 สามารถส่งข้อมูลออกไปได้ แต่มีเงื่อนไขว่า บิท T_x Enable ในไบท์คำสั่ง(Command Byte)จะต้องถูกเซ็ทให้มีสถานะเป็น "1" (High) ด้วย ดังนั้นโครงงานลักษณะการต่อใช้งานของขา CTS นี้จะถูกต่อไว้ที่กราวนด์ เพื่อให้ 8251 สามารถส่งข้อมูลออกได้ตลอดเวลาที่ต้องการ (บิท T_x Enable ถูกเซ็ทให้มีสถานะ "1")

8. ขา TxC (Transmitter Clock):PIN 9

อินพุทขานี้เป็นขาที่ใช้ควบคุมอัตราการส่งข้อมูลที่ส่งออกจาก 8251 โดยในโหมดซิงโครนัสของโครงงานนี้จะมีอัตราโบด์(Baud Rate)เท่ากับความถี่ที่ป้อนเข้าทางขา RxC นี้

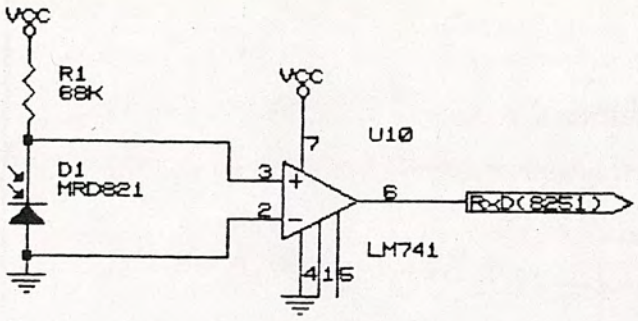
9. ขา RxC (Receiver Clock):PIN 25

อินพุทขานี้ทำหน้าที่ควบคุมอัตราการรับข้อมูลที่เข้ามายัง 8251 โดยในโหมดซิงโครนัสของโครงงานนี้จะมีอัตราโบด์(Baud Rate)เท่ากับความถี่ที่ป้อนเข้าทางขา RxC นี้

10. ขา RxD (Receive Data):PIN 3

อินพุทขานี้ใช้ทำหน้าที่รับข้อมูลเข้ามายัง 8251 ในโครงงานนี้ขา RxD จะถูกต่อเข้ากับเอ้าท์พุทของวงจรส่วนรับสัญญาณควบคุม ดังรูปที่ 2.3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3.8 แสดงวงจรที่ใช้ต่อเชื่อมกับขา RxD

11.ขา TxD (Transmit Data):PIN 19

เอ๊าท์พุทขานี้ทำหน้าที่ในการส่งข้อมูลออกจาก 8251 ในโครงงานนี้ได้ต่อขานี้เข้ากับภาคขยายสัญญาณและภาคมอดดูเลทความถี่พาหะก่อนที่ข้อมูลจะถูกส่งออกไปใช้งานภายนอก ดังรูป

รูปที่ 2.3.9 แสดงวงจรที่ใช้ต่อเชื่อมกับขา TxD

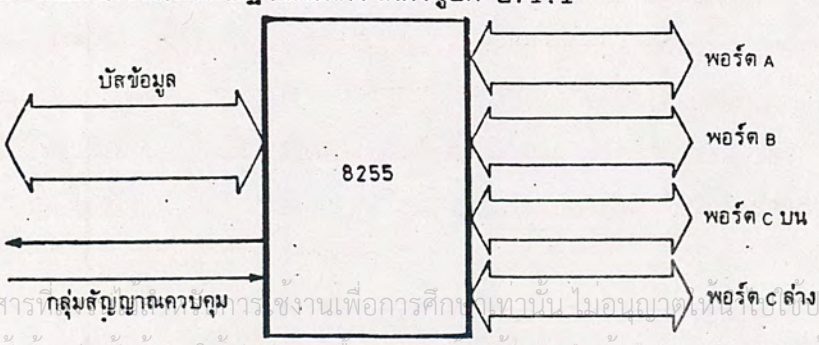
12.ขา D0-D7

เป็นบัสข้อมูล ซึ่งเราต่อเข้าโดยตรงกับบัสข้อมูลของซีพียู

2.4 โครงสร้างและหลักการทำงานของ 8255 พอร์ตข้อมูลแบบขนาน

2.4.1) โครงสร้างทางสถาปัตยกรรม

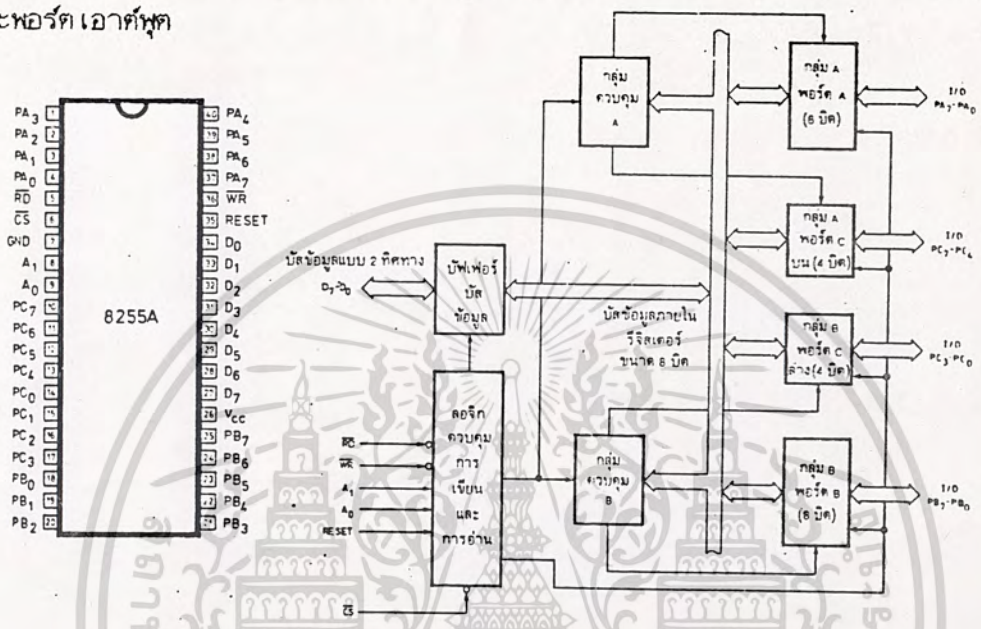
8255 เป็นไอซีที่มี 40 ขา ที่สามารถจะนำมาต่อเป็นพอร์ตให้กับไมโครโปรเซสเซอร์ได้ถึง 3 พอร์ต โดยมีโครงสร้างพื้นฐานที่แสดงได้ดังรูปที่ 2.4.1



รูป 2.4.1 แผนผังโครงสร้างของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเรียกพอร์ตของ8255นั้น เราสามารถเรียกพอร์ตต่างๆว่าเป็นพอร์ต A พอร์ต B และพอร์ต C โดยพอร์ต C แยกออกเป็น 2 ส่วนคือ พอร์ต C ล่างหรือตั้งแต่ PC_0-PC_3 มีจำนวน 4 บิต และพอร์ต C บนหรือตั้งแต่ PC_4-PC_7 ที่พิเศษคือ พอร์ตทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุต



รูป 2.4.2 แผนผังวงจรภายในและการจัดขาของ 8255

รูปที่ 2.4.2 เป็นแผนผังภายในและการจัดขาของ 8255 ซึ่งในการทำงานของวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงาน โดยไมโครโปรเซสเซอร์จะส่งคำสั่งมาโปรแกรมการทำงานหรือกำหนดรูปแบบของพอร์ตให้เป็นอินพุตหรือเอาต์พุตได้ตามที่ต้องการ

2.4.2) <u>รายละเอียดและหน้าที่ของขาสัญญาณต่างๆมีดังต่อไปนี้</u>

D_0-D_7 เป็นขาที่ข้อมูลอินพุตเอาต์พุตจะต้องผ่านเข้าออกทางขานี้ ดังนั้นจึงต่อขานี้เข้ากับบัสข้อมูลของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถที่จะอ่านหรือเขียนข้อมูลจากพอร์ตผ่านทางบัสนี้

$\overline{CS}$ (สัญญาณเลือกชิป) ขานี้เป็นขาอินพุตที่จะรับสัญญาณจากภายนอกเพื่อเลือกให้ 8255 ทำงาน โดยเมื่อขานี้เป็น "0" ก็จะมีผลทำให้ 8255 ถูกต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถที่จะอ่านหรือเขียนข้อมูลจากพอร์ตได้

$\overline{RD}$ (สัญญาณการอ่าน) เป็นสัญญาณอินพุตที่ต้องส่งมาจากไมโครโปรเซสเซอร์ โดยเมื่อสัญญาณที่ขานี้และสัญญาณ CS เป็น "0" พร้อมกัน ก็จะทำให้ไมโครโปรเซสเซอร์สามารถที่จะอ่านข้อมูลจากพอร์ตที่เป็นพอร์ตอินพุตของ 8255 ได้

$\overline{WR}$ (สัญญาณการเขียน) เป็นสัญญาณอินพุตที่ต้องส่งมาจากไมโครโปรเซสเซอร์ โดยเมื่อสัญญาณที่ขานี้และสัญญาณ $\overline{CS}$ เป็น "0" พร้อมกัน ก็จะทำให้ไมโครโปรเซสเซอร์สามารถที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

จะเขียนข้อมูลให้พอร์ตที่ต้องการได้

A_0-A_1 (สัญญาณแอดเดรส) ลอจิกของสัญญาณทั้งสองจะถอดรหัสออกเป็น 4 รหัส เพื่อที่จะทำการกำหนดรีจิสเตอร์ภายในที่เชื่อมต่อกับพอร์ตอินพุตเอาต์พุตของ 8255

RESET (สัญญาณรีเซต) เป็นสัญญาณที่ส่งเข้ามาจากภายนอกเพื่อที่จะทำการรีเซต 8255 เพื่อให้ 8255 ทำการเคลียร์สถานะต่างๆของตัวมันเอง โดยเมื่อ 8255 ถูกรีเซตก็จะทำให้มันกลับเข้าสู่โหมดอินพุต หรือทุกๆพอร์ตเป็นพอร์ตอินพุตนั่นเอง

PA_0-PA_7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต A ส่วนการเลือกพอร์ตนั้นก็เลือกจากการถอดรหัสขา A_0-A_1 นั่นเอง

PB_0-PB_7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต B ส่วนการเลือกพอร์ตนั้นก็เลือกจากการถอดรหัสขา A_0-A_1 นั่นเอง

PC_0-PC_7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต C ส่วนการเลือกพอร์ตนั้นก็เลือกจากการถอดรหัสขา A_0-A_1 นั่นเอง สำหรับพอร์ต C นี้แบ่งออกเป็น 2 กลุ่มคือ กลุ่ม PC_0-PC_3 และกลุ่ม PC_4-PC_7

2.4.3) <u>รีจิสเตอร์ภายในของ 8255</u>

8255 มีพอร์ตอยู่ 4 พอร์ตโดยที่แต่ละพอร์ตเสมือนเป็นรีจิสเตอร์ที่สามารถเขียนและอ่านได้ รีจิสเตอร์แต่ละตัวนี้จึงถูกกำหนดด้วยแอดเดรสตามที่เราตั้งเอาไว้ เช่น ตั้งให้มีค่าแอดเดรสเป็น 10H,11H,12H และ 13H เป็นต้น ซึ่งรีจิสเตอร์แต่ละตัวที่กล่าวมานี้จะได้รับการกำหนดการใช้งานควบคู่ไปกับสัญญาณ $\overline{RD}$ และ $\overline{WR}$ เพื่อแสดงความหมายของรีจิสเตอร์(พอร์ต)แต่ละตัว ดังตารางที่ 2.5.1 ข้างล่างนี้

$\overline{RD}$	$\overline{WR}$	A_0	A_1	ความหมาย
1	0	0	0	เขียนข้อมูลลงพอร์ต A
0	1	0	0	อ่านข้อมูลจากพอร์ต A
1	0	0	1	เขียนข้อมูลลงพอร์ต B
0	1	0	1	อ่านข้อมูลจากพอร์ต B
1	0	1	0	เขียนข้อมูลลงพอร์ต C
0	1	1	0	อ่านข้อมูลจากพอร์ต C
1	0	1	1	เขียนรหัสควบคุม 8255
0	1	1	1	อ่านข้อมูล (ไม่มีความหมาย)

ตารางที่ 2.4.1 สัญญาณควบคุมการทำงานของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งาน 8255 นั้นเราจะต้องส่งรหัสควบคุมเข้าไปยังพอร์ตข้อมูลควบคุมเพื่อควบคุมการทำงานของ 8255 โดยการเขียนรหัสควบคุม 8255 ทั้งนี้เพราะ 8255 มีการทำงานอยู่ 3 โหมด คือโหมด 0 โหมด 1 และโหมด 2 ซึ่งแต่ละโหมดก็มีการทำงานที่แตกต่างกันออกไปดังจะได้กล่าวในรายละเอียดต่อไป

2.4.4) การทำงานในโหมดต่างๆของ 8255

ในการที่จะกำหนดโหมดการทำงานให้กับ 8255 นั้น เราจะต้องส่งรหัสควบคุมเข้าไปยังพอร์ตควบคุมของ 8255 โดยแต่ละบิตของรหัสควบคุมมีความหมายดังต่อไปนี้

บิต D_7 เป็นบิตที่แสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น "1" หมายถึงรหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเซตโหมดต่างๆของ 8255

บิต D_6-D_5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมีอยู่ 3 โหมดคือ โหมด 0 โหมด 1 และโหมด 2 โดย ถ้า D_6และD_5 เป็น "0" ทั้งคู่ก็จะเป็นการเลือกโหมด 0

ถ้า D_6 เป็น "0" และ D_5 เป็น "1" ก็จะเป็นการเลือกโหมด 1

ถ้า D_6 เป็น "1" ก็จะเป็นการเลือกโหมด 2 ไม่ว่า D_5 จะเป็นอะไรก็ตาม

บิต D_4 ถ้ามีค่าเป็น "0" หมายถึงการกำหนดพอร์ต A เป็นเอาต์พุต ถ้ามีค่าเป็น "1" จะหมายถึงการกำหนดให้พอร์ต A เป็นอินพุต

บิต D_3 เป็นบิตที่บอกถึงการเซตของพอร์ต C บน ถ้าเป็น "0" จะทำให้พอร์ต C บนเป็นเอาต์-พุต ถ้าเป็น "1" จะทำให้พอร์ต C บนเป็นอินพุต

บิต D_2 เป็นบิตที่บอกถึงการเซตโหมดของพอร์ต B ถ้าเป็น "0" หมายถึง การเลือกพอร์ต B เป็นโหมด 0 และถ้าเป็น "1" หมายถึงการเลือกโหมด 1

บิต D_1 เป็นการกำหนด อินพุต เอาท์พุตของพอร์ต B ถ้าเป็น "0" หมายถึง เอาท์พุต ถ้าเป็น "1" หมายถึง อินพุต

บิต D_0 เป็นการกำหนด อินพุต เอาท์พุตของพอร์ต C ถ้าเป็น "0" หมายถึง เอาท์พุต ถ้าเป็น "1" หมายถึง อินพุต

การโปรแกรม 8255 จะเริ่มจากการเซตค่าที่ต้องการแล้วเอาท์พุตไปยังพอร์ตควบคุม เช่น ถ้าต้องการโปรแกรมให้ทั้ง พอร์ต A,B และ C เป็นพอร์ตเอาท์พุตหมด เราจะเลือก 8255 ให้อยู่ในโหมด 0 โดยมีรหัสควบคุมเป็น 10000000 หรือ 80H ดังนั้นจึงเขียนคำสั่งได้เป็น

```
LD   A,80H      หมายถึง กำหนดรหัสควบคุม
Out  (13H),A    หมายถึง ส่งไปยังพอร์ตควบคุม
```

หลังจากที่กระทำคำสั่ง OUT นี้ผ่านไปแล้ว พอร์ต A,B และ C จะเป็นพอร์ตเอาท์พุตหมดซึ่งก็จะส่งข้อมูลจากซีพียูไปยังพอร์ตต่าง ๆ ได้ เช่น ถ้าต้องการส่งข้อมูล 8AH ไปยังพอร์ต A ข้อมูล 41H ไปยังพอร์ต B และข้อมูล 25H ไปยัง พอร์ต C คำสั่งที่ใช้คือ

```
LD   A,8AH      หมายถึง เลือกค่า 8AH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUT (10H),A หมายถึง ส่งให้พอร์ต A

LD A,41H หมายถึง เลือกค่า 41H

OUT (11H),A หมายถึง ส่งให้พอร์ต B

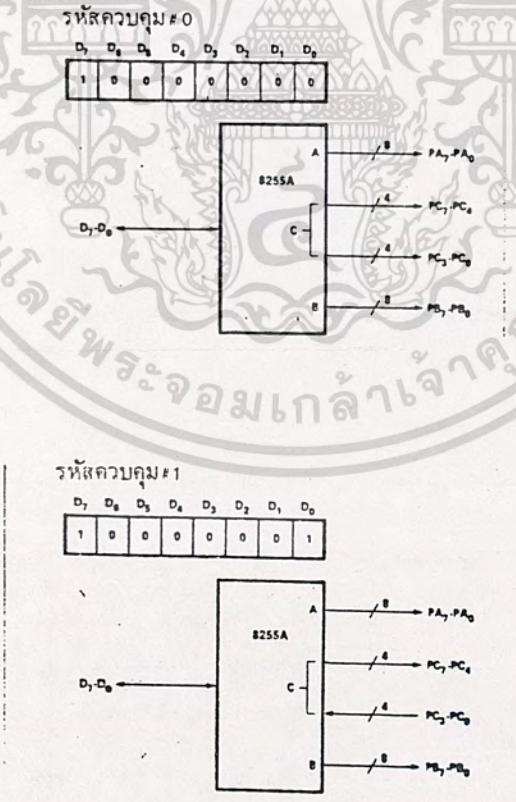
LD A,25H หมายถึง เลือกค่า 25H

OUT (12H),A หมายถึง ส่งให้พอร์ต C

เนื่องจากมีพอร์ตที่รับส่งข้อมูล 3 พอร์ต คือ พอร์ต A,พอร์ต B และ พอร์ต C ซึ่งพอร์ต C จะแยกออกเป็น สองส่วนคือ พอร์ต C ล่าง และ พอร์ต C บน เราสามารถโปรแกรมให้ทั้ง 4 พอร์ตนี้ เป็น อินพุต หรือ เอาท์พุต ก็ได้ เช่น ถ้าข้อมูลควบคุมเป็น 82H จะทำให้พอร์ต B เป็น อินพุต พอร์ต A และ พอร์ต C เป็นเอาท์พุต

2.4.4.1) การทำงานของ 8255 ในโหมด 0

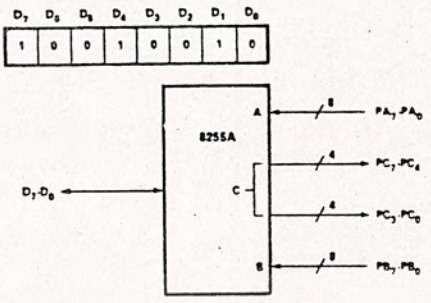
โหมด 0 เป็นโหมดที่กำหนดให้พอร์ตทุกพอร์ตบนตัว 8255 เป็นพอร์ตอินพุต เอาท์พุตแบบพื้นฐาน รูปแบบความเป็นไปได้ จึงมีทั้งสิ้น 16 รูปแบบตามลักษณะของพอร์ต A พอร์ต B พอร์ต C บน และ พอร์ต C ล่าง ลักษณะของรหัสควบคุมแต่ละแบบจะเป็นดังรูปที่ 2.4.3



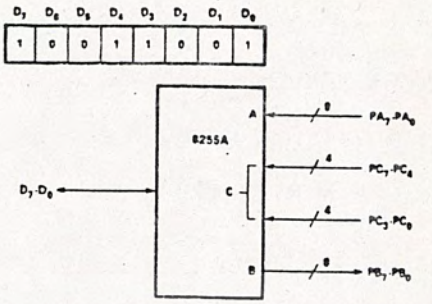
รูปที่ 2.4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

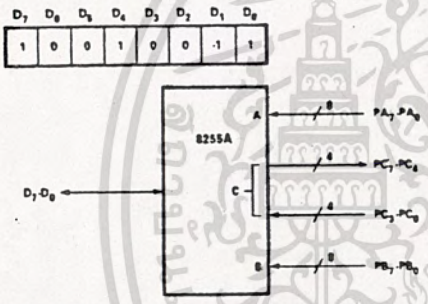
รหัสควบคุม # 10



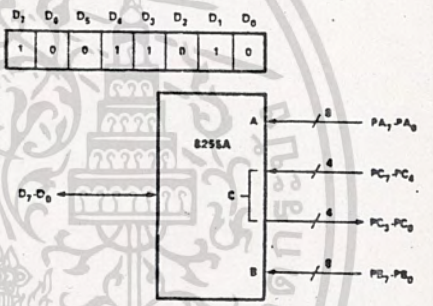
รหัสควบคุม # 13



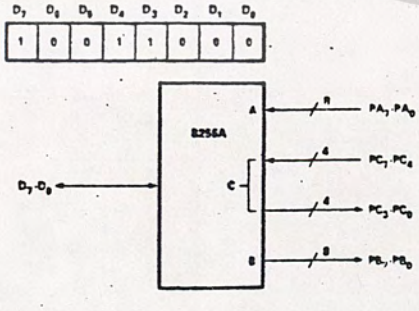
รหัสควบคุม # 11



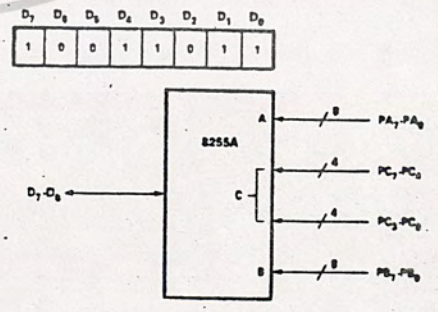
รหัสควบคุม # 14



รหัสควบคุม # 12



รหัสควบคุม # 15.

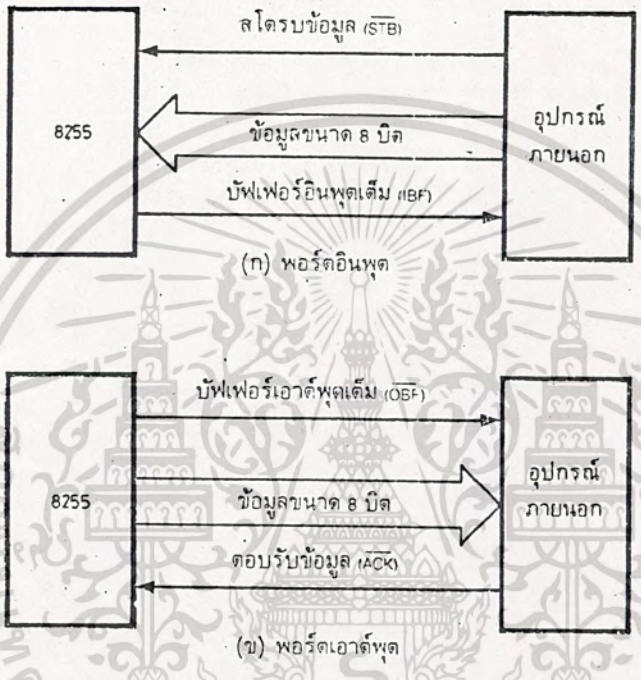


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4.3

2.4.4.2) การทำงานของ 8255 ในโหมด 1

การทำงานของ 8255 ในโหมด 1 เป็นโหมดที่ทำให้ อินพุต เอาท์พุตมีการตรวจสอบสัญญาณ (HandShaking) โดยใช้อินพุต เอาท์พุตของพอร์ต A และ พอร์ต B เป็นหลัก และใช้พอร์ต C บนเป็นตัวตรวจสอบสัญญาณ(handshake) ของพอร์ต A ส่วนพอร์ต C ล่าง เป็นตัวตรวจสอบ สัญญาณของพอร์ต B การจัดสัญญาณต่าง ๆ เหล่านี้ แสดงได้ดังรูป 2.4.4



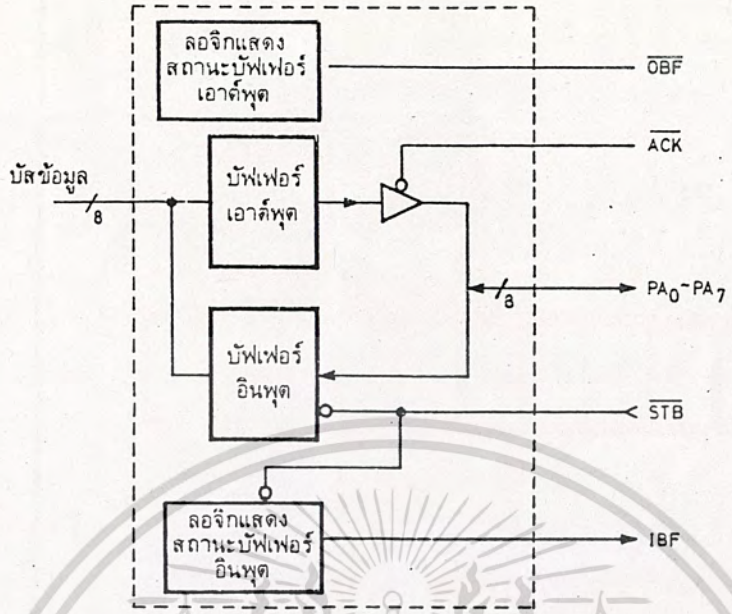
รูปที่ 2.4.4

2.4.4.3) การทำงานของ 8255 ในโหมด 2

8255 ยังมีโหมดการทำงานอีกโหมดหนึ่งคือโหมด 2 ซึ่งทำได้เฉพาะพอร์ต A ในโหมดนี้ 8255 จะใช้พอร์ต A ทำหน้าที่เป็นพอร์ตแบบ 2 ทิศทาง คือ สามารถเป็นได้ทั้งพอร์ตอินพุตและเอาท์พุต โดยโครงสร้างของพอร์ต A ทั้งอินพุต เอาท์พุตมีตัวตรวจสอบสัญญาณทั้งคู่ ส่วนพอร์ต C จะทำหน้าที่เป็นสัญญาณตรวจสอบ โดยมีสัญญาณแต่ละขาดังตารางที่ 2.4.2

พอร์ต C	ความหมาย
PC_0	I/O
PC_1	I/O
PC_2	I/O
PC_3	$INTR_A$
PC_4	$\overline{STB}_A$
PC_5	IBF_A
PC_6	$\overline{ACK}_A$
PC_7	$\overline{OBF}_A$

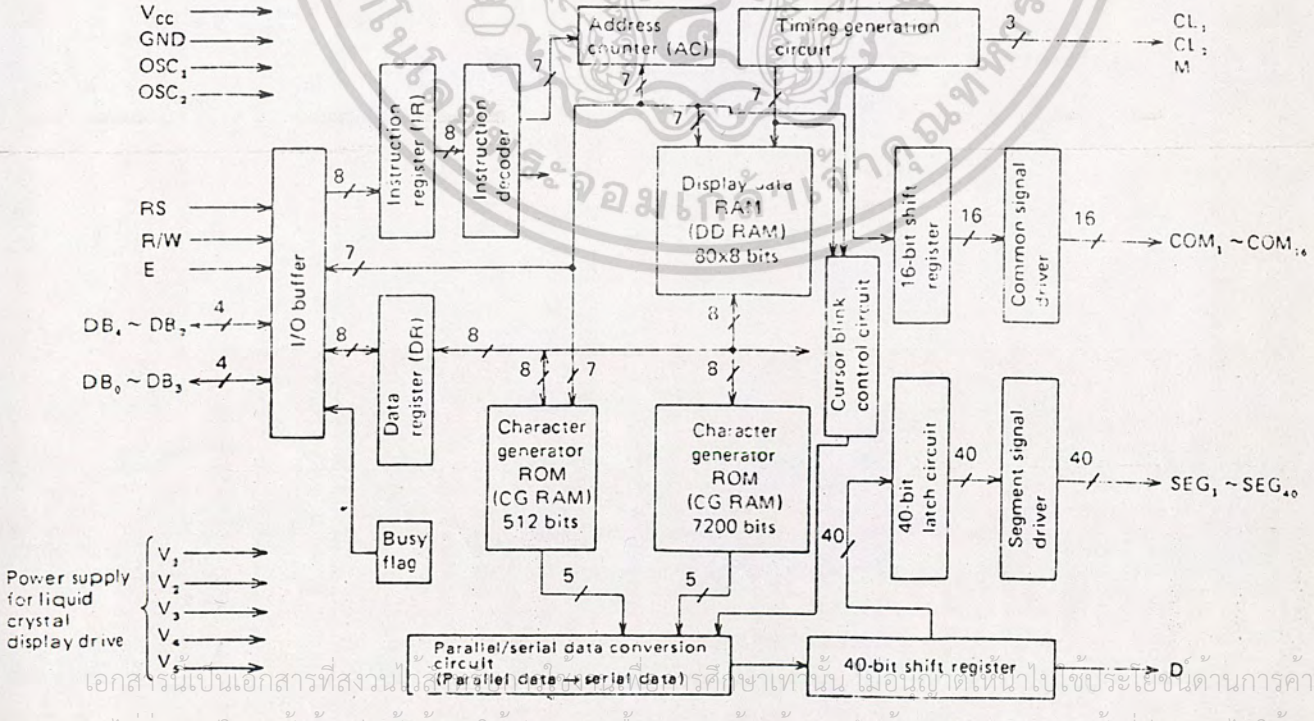
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.5 โครงสร้างพอร์ต A ที่ทำงานแบบพอร์ต 2 ทาง

2.5 โครงสร้างและหลักการโปรแกรมหน่วยแสดงผล

โครงสร้างภายในของวงจรควบคุมการแสดงผล LCD ดังรูปที่ 2.5.1

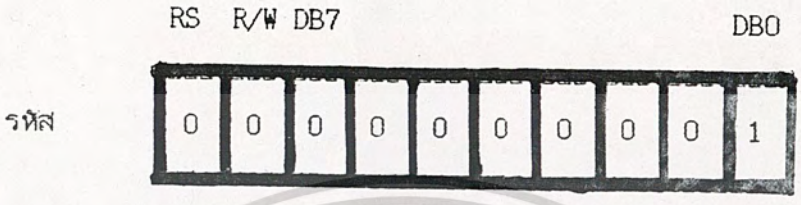


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5.1 แสดงโครงสร้างภายในของวงจรควบคุม

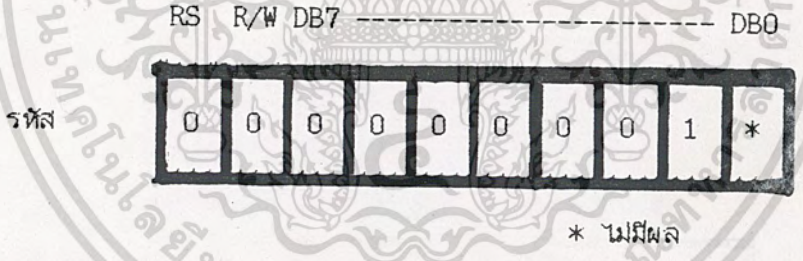
รายละเอียดคำสั่ง

1. คำสั่งเคลียร์หน้าจอ (Clear display)



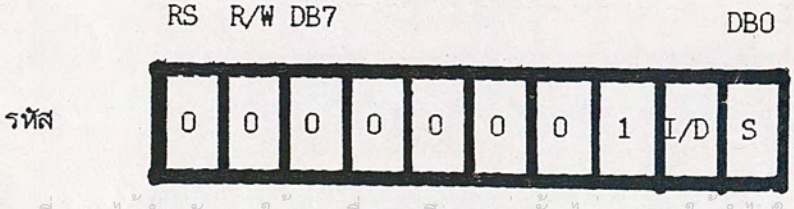
คำสั่งนี้จะเป็นการเขียนช่องว่าง (SPACE) นั่นคือรหัสเอสกี้ (ASCII) 20H เข้าไปในตำแหน่งบนหน้าจอหมดและทำการเซทค่าของตำแหน่งอ้างอิงบนหน้าจอ(DD Ram Address) ให้เป็นศูนย์ ตัวเคอร์เซอร์ (Cursor) จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ การเซท I/D = 1,S ไม่มีการเปลี่ยน

2. คำสั่งรีเทิร์นโฮม (Return home)



คำสั่งนี้จะทำการเซทค่าของตำแหน่งอ้างอิงบนหน้าจอให้เป็นศูนย์ ตัวเคอร์เซอร์จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ ข้อมูลในจอภาพไม่เปลี่ยนแปลง

3. คำสั่งเอ็นทรีโหมดเซท (Entry mode set)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้ว จะทำให้ตำแหน่งอ้างอิงบนหน้าจอ เพิ่มขึ้นหนึ่งหรือลดลงหนึ่ง โดยถ้าเป็น 1 = เพิ่มขึ้นหนึ่ง, 0 = ลดลงหนึ่ง

บิต S : เป็นตัวกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัวเคอร์เซอร์อยู่กับที่ ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่กับที่ ตัวเคอร์เซอร์จะถูกดันไปทางขวามือ

4. คำสั่งเปิดปิดหน้าจอ (Display On/Off Control)

	RS	R/W	DB7							DB0
รหัส	0	0	0	0	0	0	1	D	B	C

บิต D : ถ้า D เป็น 1 จะเป็นการเปิดหน้าจอ
ถ้า D เป็น 0 จะเป็นการปิดหน้าจอ

C : ถ้า C เป็น 1 จะเป็นการแสดงเคอร์เซอร์
ถ้า C เป็น 0 จะเป็นการไม่แสดงเคอร์เซอร์

B : ถ้า B เป็น 1 จะเป็นการกำหนดให้เคอร์เซอร์กระพริบ
ถ้า B เป็น 0 จะเป็นการกำหนดให้เคอร์เซอร์ไม่กระพริบ

5. คำสั่งเลื่อนเคอร์เซอร์หรือฮักชระ (Cursor Display Shift)

	RS	R/W	DB7							DB0
รหัส	0	0	0	0	0	1	S/C	R/L	*	*

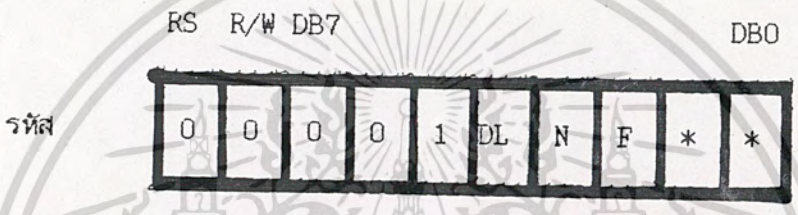
* ไม่มีผล

เป็นคำสั่งกำหนดให้ตำแหน่งเคอร์เซอร์ หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่านโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต S/C	R/L	
0	0	ทำการย้ายเคอร์เซอร์ไปจากตำแหน่งเดิมทางซ้าย 1 ตำแหน่ง
0	1	ทำการย้ายเคอร์เซอร์ไปจากตำแหน่งเดิมทางขวา 1 ตำแหน่ง
1	0	เป็นการดันตัวอักษรที่เกิดไปทางซ้าย 1 ตำแหน่ง
1	1	เป็นการดันตัวอักษรที่เกิดไปทางขวา 1 ตำแหน่ง

6. <u>คำสั่งการเซตฟังก์ชั่น</u> (Function Set)



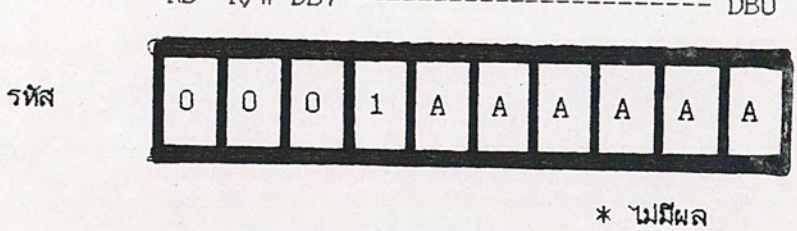
* ไม่มีผล

บิต DL: เป็นการเซตการติดต่อว่าจะให้เป็นแบบ 8 บิต หรือ 4 บิต โดยถ้าต้องการติดต่อแบบ 4 บิต บิตนี้จะเป็น 0 แต่ถ้าต้องการติดต่อแบบ 8 บิต บิตนี้จะเป็น 1

N : เป็นการเซตบรรทัดการแสดงผล โดยถ้า N = 0 จะแสดง 1 บรรทัด N = 1 จะแสดง 2 บรรทัด ในกรณีมากกว่า 2 บรรทัดก็ให้เซต N = 1

F : เป็นการเซตขนาดของจุดแสดงผลว่าจะให้เป็นแบบ 5x7 หรือ 5x10 โดยถ้า F = 0 จะเป็นแบบ 5x7 และ F = 1 จะเป็นแบบ 5x10

7. <u>คำสั่งที่ใช้กำหนดค่าแอดเดรสของตัวสร้างอักขระ</u> (Set CG RAM Address)

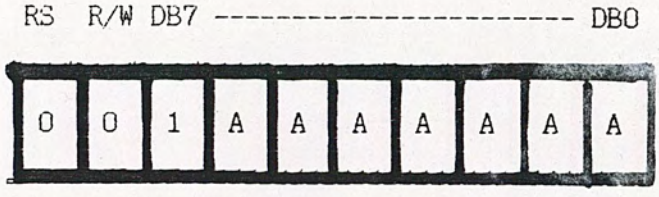


* ไม่มีผล

เป็นการเซตค่าแอดเดรสใน CG RAM โดยต้องทำการเซตแอดเดรสก่อนเขียนข้อมูลหรืออ่านข้อมูล จาก CG RAM ด้วย

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. <u>คำสั่งที่ใช้กำหนดค่าแอดเดรสบนหน้าจอ</u> (Set DD RAM Address)

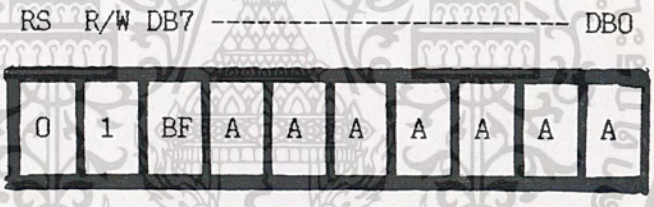


เป็นคำสั่งที่ใช้เซตค่าแอดเดรสใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM โดยจำนวนแอดเดรสที่เกิดขึ้นบนจอจะขึ้นอยู่กับการเซตค่า N โดย

ถ้า N = 0 (1 บรรทัด) แอดเดรสจะอยู่ที่ 00H - 4FH

ถ้า N = 1 (2 บรรทัด) แอดเดรสจะอยู่ที่ 00H - 27H สำหรับบรรทัดที่ 1 และ 40H - 67H สำหรับบรรทัดที่ 2

9. <u>คำสั่งอ่านค่าสถานะและแอดเดรส</u> (Read BUSY FLAG & Address)

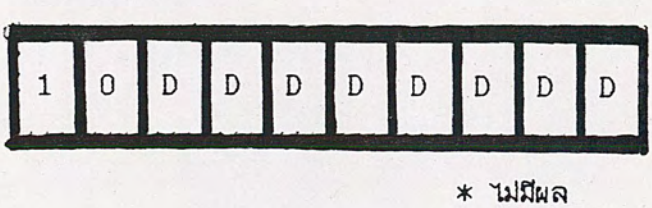


เป็นคำสั่งอ่านค่าสถานะว่าขณะนั้นจอแสดงผล LCD พร้อมที่จะรับข้อมูลหรือไม่

ถ้า BF = 1 แสดงว่าจอแสดงผลยังไม่พร้อมที่จะรับข้อมูลหรือคำสั่ง

ถ้า BF = 0 แสดงว่าจอแสดงผลพร้อมที่จะรับข้อมูลหรือคำสั่ง

10. <u>คำสั่งเขียนข้อมูลลงใน CG RAM หรือ DD RAM</u>



เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG RAM หรือ DD RAM โดยก่อนที่จะเขียนข้อมูลจะต้องกำหนดแอดเดรสเริ่มต้นก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. คำสั่งอ่านข้อมูลจาก CG RAM หรือ DD RAM

	RS	R/W	DB7							DB0
รหัส	1	1	D	D	D	D	D	D	D	D

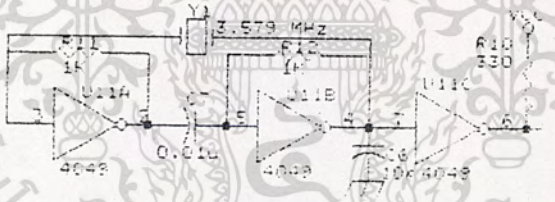
* ไม่มีผล

เป็นคำสั่งอ่านข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนที่จะอ่านข้อมูลจะต้องกำหนดแอดเดรสเริ่มต้นก่อน

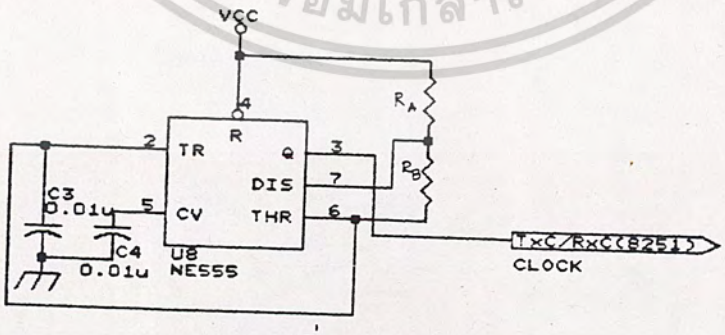
2.6 โครงสร้างส่วนสร้างสัญญาณนาฬิกา

สำหรับในโครงงานนี้ จะใช้วงจรสร้างสัญญาณนาฬิกา 2 แบบด้วยกัน คือ

1. วงจรสร้างสัญญาณนาฬิกาให้ส่วนประมวลผล ซึ่งจะมีวงจรดังรูปที่ 2.6.1
2. วงจรสร้างสัญญาณนาฬิกา ให้กับภาครับส่งข้อมูลแบบอนุกรม ซึ่งจะมีวงจรดังรูป 2.6.2



รูปที่ 2.6.1



รูปที่ 2.6.2

จากรูป 2.6.1 ความถี่ที่ได้จากวงจรนั้น จะมีความถี่ประมาณความถี่ของ คริสตอล ดังนั้น ความถี่ที่ต้องการจึงขึ้นอยู่กับ คริสตอลที่เราเลือกใช้

จากรูป 2.6.2 จะเห็นได้ว่าลักษณะเป็นวงจร อะสเตเบิล(ASTABLE) ซึ่งความถี่ที่สร้างจากวงจรนั้น สามารถคำนวณได้จากสมการต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f = (1.43)/((Ra+2Rb)*C) \quad \text{---------- (2.6.1)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

ในการออกแบบจะแบ่งการออกแบบเป็น 3 ส่วนใหญ่ ดังต่อไปนี้

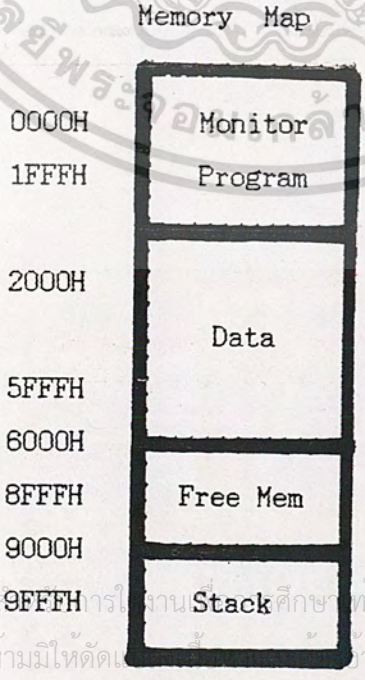
1. ส่วนประมวลผลกลาง
2. ส่วนรับส่งข้อมูลจาก เครื่องควบคุมระยะไกลภายนอก
3. ส่วนแสดงผล

3.1 ส่วนประมวลผลกลาง

ในส่วนนี้ใช้ ไมโครโปรเซสเซอร์ เบอร์ Z80 เป็นตัวประมวลผลกลาง และประกอบด้วยหน่วยความจำแบบรอม 8 กิโลไบท์,แรม 32 กิโลไบท์ โดยมีการจัดแบ่งแอดเดรสของหน่วยความจำแต่ละแบบดังนี้

3.1.1 หน่วยความจำแบบรอม ใช้ ไอซี เบอร์ 2764 ซึ่งมีขนาด 8K*8 บิท โดยที่ส่วนนี้จะใช้บันทึก มอนิเตอร์โปรแกรม ซึ่งอยู่ในช่วงแอดเดรส 0000H - 1FFFH

3.1.2 หน่วยความจำแบบแรม ใช้ ไอซี เบอร์ 62256 ซึ่งมีขนาด 32K*8 บิท โดยเนื้อที่ส่วนนี้ใช้เป็นที่บันทึกข้อมูลที่เก็บจากสัญญาณเครื่องควบคุมระยะไกล โดยแบ่งเป็น 2 ส่วน คือ ส่วนที่เก็บข้อมูล (ช่วงแอดเดรส 2000H - 5FFFH)และส่วนที่เป็นสแตค (ช่วงแอดเดรส 9F00H - 9FFFH)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนรับส่งข้อมูลจากเครื่องควบคุมระยะไกลภายนอก

ในระบบส่วนนี้ จะเน้นการสร้างและออกแบบเป็นส่วนย่อย ๆ ดังนี้

3.2.1 การเลือกและออกแบบการทำงานของ 8251

3.2.2 การออกแบบและคำนวณวงจรส่วนสัญญาณนาฬิกา

3.2.3 การออกแบบภาครับส่งสัญญาณอินฟราเรด

3.2.1 การเลือกและออกแบบการทำงานของ 8251

1. การเลือกโหมด

สำหรับโครงงานนี้ได้เลือกการทำงานของ 8251 ในโหมดซิงโครนัส ดังนั้นจะกล่าวถึงการทำงานเฉพาะในโหมดนี้เท่านั้น

รูปแบบของ MODE WORD ที่ใช้ในโครงงานนี้คือ

D7	D6	D5	D4	D3	D2	D1	D0
SCS	ESD	EP	EN	L2	L1	0	0

โดยแต่ละบิทมีความหมายดังนี้

-บิท D0,D1 จะมีค่าเป็น 0,0 ตามลำดับ ซึ่งเป็นการกำหนดการทำงานในโหมดซิงโครนัส ให้แก่ 8251

-บิท D2,D3 เป็นบิทที่ใช้กำหนดความยาวของข้อมูล โดยสามารถเลือกได้ตั้งแต่ 5-8 บิท ดังนี้

D3	D2	ความยาวข้อมูล
0	0	5 บิท
0	1	6 บิท
1	0	7 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D3	D2	ความยาวข้อมูล
1	1	8 บิท

***ในโครงงานนี้จะใช้ความยาวข้อมูล 8 บิท ดังนั้นบิท D3,D2 จะมีค่าเป็น 0,0

-บิท D4 (PEN : Parity Enable)

เป็นบิทที่กำหนดว่าจะให้มีค่าพาริตี้(Parity)หรือไม่ โดยถ้าบิทนี้เป็น "1" จะเป็นการกำหนดให้มีพาริตี้(Parity Enable) แต่ถ้าบิทนี้เป็น "0" จะไม่มีพาริตี้ (Parity Disable)

-บิท D6 (ESD : External Sync Character)

เป็นบิทที่กำหนดการตรวจจับสัญญาณซิงค์(Sync Detect)ว่าเป็นจะตรวจจับสัญญาณซิงค์ภายนอก(External)นั่นคือบิทนี้มีค่าเป็น "0" หรือสัญญาณซิงค์ภายใน (Internal)นั่นคือบิทนี้มีค่าเป็น "1"

-บิท D7 (SCS : Single Character Sync)

เป็นบิทที่ใช้กำหนดลักษณะของสัญญาณซิงค์ ว่าจะให้เป็นแบบสัญญาณซิงค์เดี่ยว (Single)นั่นคือให้บิทนี้มีค่าเป็น "1" หรือเป็นสัญญาณซิงค์แบบคู่(Double)นั่นคือให้บิทนี้มีค่าเป็น "0"

ในโครงงานนี้กำหนดให้ Mode Word มีค่าเท่ากับ 0CCH (11001100B)

2. การกำหนด Sync Character

ในโครงงานนี้กำหนดให้ Sync Character มีค่าเท่ากับ 055H

3. การกำหนดค่าสเตตัสต่างๆใน Command Instruction

รูปแบบของ Command Instruction ของ 8251 คือ

D7	D6	D5	D4	D3	D2	D1	D0
EH	IR	RTS	ER	SBRK	RxE	DTR	TxEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

โดยแต่ละบิทมีความหมายดังนี้

-บิท D_0 (TxEN = Transmit enable) เป็นบิตที่ใช้กำหนดว่าจะให้มีการส่งข้อมูลออกหรือไม่ บิตนี้เป็น 1 ก็คือ กำหนดให้มีการส่งข้อมูลออกได้ (Enable) แต่ถ้าบิตนี้เป็น 0 ก็คือกำหนดให้ไม่มีการส่งข้อมูลออก (Disable)

-บิท D_1 (DTR = Data Terminal Ready) ถ้าบิตนี้มีค่าเป็น 1 ก็จะมีผลทำให้ขา DRT มีสถานะเป็น 0

-บิท D_2 (RxE = Recieve Enable) เป็นบิทที่ใช้กำหนดว่าจะให้มีการรับเข้าหรือเปล่าโดยถ้าบิทเป็น 1 ก็คือกำหนดให้มีการรับข้อมูลเข้าได้ (Enable) แต่ถ้าบิทนี้เป็น 0 ก็คือกำหนดให้ไม่มีการรับข้อมูลเข้า (Disable)

-บิท D_3 (SBRK = Send Break Charactor) ถ้าบิทนี้เป็น 1 ก็จะมีผลทำให้ขา T*D มีสถานะเป็น 0 แต่ถ้าบิทนี้มีค่า 0 ก็จะเป็นการบอกให้ 8251 ทำงานตามปกติ

-บิท D_4 (ER = Error Reset) เป็นบิทที่ใช้กำหนดว่าจะให้มีการรีเซทความผิดพลาดที่เกิดขึ้น(Pairty Error,Overrun Error และ Eraming Error)โดยถ้าบิทนี้มีค่าเป็น 1 ก็คือกำหนดให้มีการรีเซทความผิดพลาดดังกล่าวโดยอัตโนมัติ เมื่อเกิดความผิดพลาดขึ้น

-บิท D_5 (RTS = Request to Send) ถ้าบิทนี้มีค่าเป็น 1 ก็จะมีผลทำให้ขา RTS มีสถานะเป็น 0

-บิท D_6 (IR = Internal Reset) ถ้าบิทนี้มีค่าเป็น 1 ก็จะมีผลทำให้ 8251 ถูกรีเซทและกลับไปอ่านค่าในโหมดคำสั่ง (Mode Instruction)ใหม่

-บิท D_7 (EH = Enable Hunt Mode) เป็นบิทที่ใช้กำหนดว่าจะให้ 8251 มีการค้นหา Sync Charactor หรือเปล่า ถ้าบิทนี้มีค่าเป็น 1 ก็จะมีผลทำให้ 8251 ค้นหา Sync Charactor ก่อนที่จะอ่านข้อมูลเพื่อนำไปใช้งาน

หลังจากที่เราทำการโปรแกรม 8251 เรียบร้อยแล้วขณะนี้ 8251 ก็พร้อมที่จะทำงานโดยในขณะที่ 8251 ทำงานนั้น ก็ยังจะต้องมีการติดต่อกับ CPU อยู่ตลอด ทั้งนี้ก่อนที่ CPU จะออกไปก็ต้องทำการตรวจสอบดูก่อนว่า 8251 พร้อมที่จะรับข้อมูลหรือยัง หรือก่อนที่ CPU จะทำการอ่านข้อมูลก็จะต้องทำการตรวจสอบดูก่อนว่า 8251 พร้อมที่จะส่งข้อมูลหรือยัง ซึ่ง CPU สามารถจะทำการตรวจสอบดังกล่าวได้จากตัว 8251เลย กล่าวคือ CPU สามารถจะรู้สถานะของ 8251 ในขณะนั้นได้โดยทำการอ่านข้อมูลเข้ามาจาก Status Word ซึ่งลักษณะของ Status Word มีดังนี้

D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแต่ละบิทมีความหมายดังนี้

-บิท D_0 (TxRDY) เป็นบิทที่ใช้บอกว่าในขณะนี้ 8251 พร้อมที่จะส่งข้อมูลหรือไม่ โดยถ้าบิทนี้มีค่าเป็น 1 ก็แสดงว่า 8251 พร้อมที่จะรับส่งข้อมูล แต่ถ้าบิทนี้มีค่าเป็น 0 ก็แสดงว่า 8251 ยังไม่พร้อมที่จะส่งข้อมูล

-บิท D_1 (RxRDY) เป็นบิทที่ใช้บอกว่าในขณะนี้ 8251 พร้อมที่จะรับข้อมูลหรือไม่ โดยถ้าบิทนี้มีค่าเป็น 1 ก็แสดงว่า 8251 พร้อมที่จะรับข้อมูล แต่ถ้าบิทนี้มีค่าเป็น 0 ก็แสดงว่า 8251 ยังไม่พร้อมที่จะรับข้อมูล

-บิท D_2 (TxEMPTY) บิทนี้จะมีค่าเป็น 1 เมื่อ 8251 ไม่มีข้อมูลที่จะส่งออกไป และจะมีค่าเป็น 0 เมื่อ 8251 ได้รับข้อมูลที่จะส่งออกจาก CPU

-บิท D_3 (PE =Parity Error) บิทนี้จะมีค่าเป็น 1 เมื่อเกิดความผิดพลาดทางพาริตี้ (Parity Error) และจะมีค่าเป็น 0 (Reset) เมื่อ ได้รับบิท ER ใน Command Instruction

-บิท D_4 (OE = Overrun Error) บิทนี้จะมีค่าเป็น 1 เมื่อเกิดความผิดพลาดแบบ โอเวอร์รัน (Overrun Error) และจะมีค่าเป็น 0 (Reset) เมื่อได้รับบิท ER ใน Command Instruction

-บิท D5 (FE = Framming Error) บิทนี้จะใช้ในกรณีของการใช้งาน 8251 ในโหมดอะซิงโครนัสเท่านั้น โดยที่บิทนี้จะมีค่าเป็น 1 เมื่อการตรวจเช็คข้อมูลในแต่ละชุดแล้วไม่พบบิทท้าย (Stop Bit) ของข้อมูล

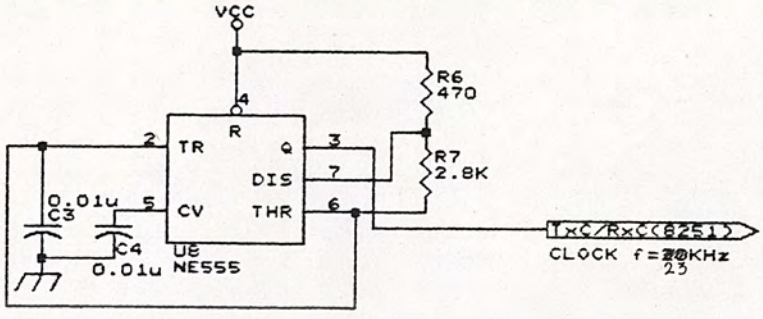
-บิท D6 (SYNDET) บิทนี้จะมีค่าเป็น 1 เมื่อ 8251 ถูกใช้งานในโหมดซิงโครนัสแล้วมีการตรวจเช็คอักขระที่ใช้ซิงค์ข้อมูลในการส่ง-รับพบหรือตรงกันทั้งสองฝ่าย

-บิท D7 (DSR = Data Set Ready) บิทนี้จะมีค่าเป็น 0 เมื่อชุดของข้อมูลถูกนำมาพร้อมที่จะกระทำการ (Operate) ใดๆก็ได้

3.2.2 การออกแบบและคำนวณวงจรส่วนสัญญาณนาฬิกา

ในโครงงานนี้ใช้ความถี่ของสัญญาณนาฬิกาในการส่งข้อมูลเท่ากับ 23.5 KHz ที่ต่อเข้าขา Txc,Rxc ของ 8251 โดยค่าความถี่ของสัญญาณนาฬิกาดังกล่าว สร้างขึ้นจากวงจรกำเนิดสัญญาณนาฬิกา (Astable Multivibrator) ของไอซีเบอร์ 555 (วงจรดังรูปที่ 3.2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.1(a) Sampling Clock circuit

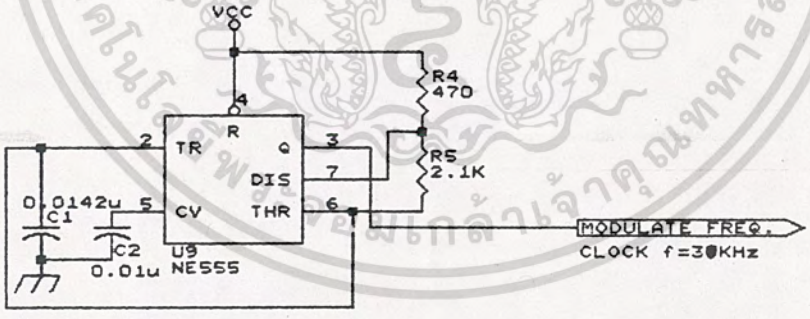
จะได้ว่าความถี่ในการรับส่งข้อมูล(Sampling Rate) จะหาได้จากสมการ 2.6.1

$$f = (1.43)/((270 + 2.8K)*0.01\mu F)$$

$$= 23.58\ KHz$$

ในวงจรส่วนนี้เราจะใช้ค่า R6 = 470 Ohm,R7 = 2.8 KOhm,C3 = 0.01 μF

ในส่วนที่ใช้ความถี่ของสัญญาณนาฬิกาในการมอดูเลทเพื่อใช้เป็นความถี่พาหะ ค่าความถี่ของสัญญาณนาฬิกาที่ใช้เท่ากับ 30 KHz (อาศัยวงจรกำเนิดสัญญาณนาฬิกาเดียวกัน)



รูปที่ 3.2.1(b) Carrier Frequency Circuit

จะได้ว่าความถี่พาหะในการมอดูเลท สามารถคำนวณได้จากสมการ 2.6.1

$$f = (1.43)/((270 + 2.1K)*0.01\mu F)$$

$$= 30.6\ KHz$$

ในวงจรส่วนนี้เราจะใช้ค่า R4 = 470 Ohm,R5 = 2.1 KOhm,C1 = 0.01 μF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนแสดงผล

สำหรับส่วนแสดงผลนี้ได้เลือกใช้ LCD เป็นหน่วยแสดงผล โดยมีการรับส่งข้อมูลระหว่าง CPU กับ LCD ผ่านทางพอร์ทของ 8255 (ไม่ว่าจะเป็นสัญญาณควบคุมจอและข้อมูลที่ CPU ส่งให้ LCD) โดยกำหนดการใช้พอร์ทต่าง ๆ ของ 8255 ให้อยู่ในโหมด 0 คือ พอร์ท A และ พอร์ท B (พอร์ท B ใช้ 3 บิทล่าง) ดังนั้นให้พอร์ท A เป็นพอร์ทข้อมูลและพอร์ท B เป็นพอร์ทควบคุม LCD

กำหนดพอร์ทข้อมูลหมายเลข	2CH
กำหนดพอร์ทควบคุม LCD หมายเลข	2EH
รหัสควบคุม 8255 คือ	80H
พอร์ทควบคุม 8255 คือ	2FH

3.4 การออกแบบวงจรภาครับส่งสัญญาณอินฟราเรด

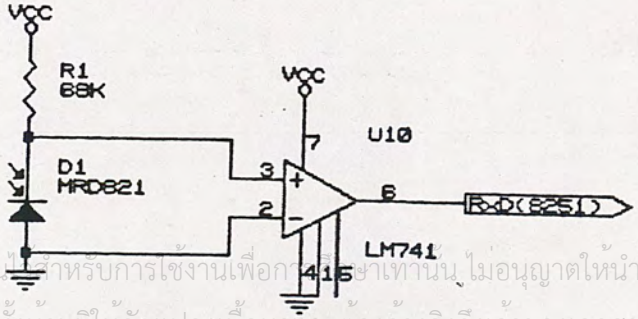
<u>เครื่องส่งและเครื่องรับ</u>

ในส่วนของเครื่องรับและเครื่องส่งในโครงงานนี้ ประกอบขึ้นจากวงจรง่ายๆ ที่ไม่มีความสลับซับซ้อนอะไรมากนัก โดยจะแยกอธิบายทีละส่วนดังต่อไปนี้

<u>วงจรของภาครับ</u>

ในส่วนของวงจรภาครับนี้เป็นส่วนของวงจรที่ใช้รับข้อมูลที่ส่งมาในรูปของสัญญาณอินฟราเรด โดยมีโฟโต้ไดโอด (Photo Diode) เป็นตัวรับสัญญาณและทำการแปลงสัญญาณที่ได้รับให้เป็นข้อมูลที่ใช้ส่งให้กับขา RxD ของ 8251

วงจรในภาครับสัญญาณอินฟราเรดของเครื่องควบคุมระยะไกลแบบโปรแกรมได้นี้สามารถเขียนแสดงได้ดังรูปที่ 3.3.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

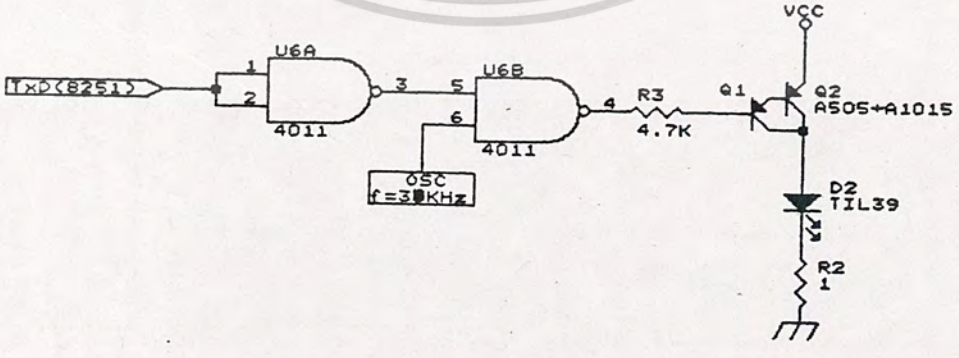
รูปที่ 3.3.1 แสดงวงจรของภาครับ

จากรูปจะเห็นได้ว่าลักษณะของสัญญาณที่มาตกกระทบโฟโต้ไดโอดกับสัญญาณเอ้าท์พุทที่ได้มีลักษณะกลับเฟสกัน ทั้งนี้เนื่องจากวงจรทางภาคส่งของเครื่องควบคุมระยะไกลแบบโปรแกรมได้ ใช้ทรานซิสเตอร์ชนิด PNP เป็นตัวขยายสัญญาณ

วงจรของเครื่องส่ง

ในส่วนของวงจรภาคส่งนี้เป็นส่วนของวงจรที่ใช้ในการส่งข้อมูลที่เก็บไว้ในหน่วยความจำของเครื่องควบคุมระยะไกลแบบโปรแกรมได้ ออกไปใช้ในการควบคุมอุปกรณ์ต่างๆ โดยมีตัวส่งสัญญาณอินฟราเรดทำหน้าที่เป็นตัวแปลงข้อมูลที่เก็บไว้ในหน่วยความจำให้เป็นสัญญาณอินฟราเรด โดยก่อนที่จะทำการแปลงข้อมูลที่เก็บไว้ในหน่วยความจำให้เป็นสัญญาณอินฟราเรดนั้นจะต้องนำเอาข้อมูลที่เก็บไว้ในหน่วยความจำ มาทำการมอดดูเลตกับสัญญาณพาหะที่มีความถี่ประมาณ 33 กิโลเฮิร์ต เสียก่อน แต่เนื่องจากข้อมูลที่เราเก็บไว้ในหน่วยความจำนั้นมีเฟสตรงกันข้ามกับสัญญาณจริงๆ (ที่ใช้ในการควบคุมอุปกรณ์) ดังนั้นก่อนที่จะทำการมอดดูเลตข้อมูลที่เก็บไว้ในหน่วยความจำนั้น เราจะต้องทำการกลับเฟสข้อมูลที่เก็บไว้ในหน่วยความจำให้เหมือนเดิมเสียก่อนแล้วจึงทำการมอดดูเลต โดยเราจะทำการมอดดูเลตสัญญาณควบคุมผ่านทาง แนน เกต (NAND GATE) ซึ่งสัญญาณเอ้าท์พุทหลังการมอดดูเลตที่ได้จาก NAND GATE ก็จะถูกกลับเฟสเรียบร้อยแล้ว หลังจากนั้นสัญญาณดังกล่าวก็จะถูกขยายโดยภาคขยาย (ซึ่งประกอบด้วยทรานซิสเตอร์แบบ PNP 2 ตัว ต่อแบบคาร์ลิงตันกัน) ต่อจากนั้นก็นำเอาสัญญาณที่ผ่านการขยายแล้วไปทำการขับ (Drive) ตัวส่งอินฟราเรดอีกทีหนึ่ง

วงจรของภาคส่งสามารถเขียนแสดงได้ดังรูปที่ 3.3.2



รูปที่ 3.3.2 แสดงวงจรของภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1) การทดลอง

ทำการทดลองโดยการบันทึกสัญญาณควบคุมเครื่องเล่นวีดีโอเทป และเครื่องรับโทรทัศน์รุ่นต่างๆ ดังนี้

4.1.1) เครื่องรับโทรทัศน์

มิตซูบิชิ	รุ่น ซาวด์นสเคว์ 14 นิ้ว
โซนี่	รุ่น ไตรนิตรอน 14,21 นิ้ว
ฟิลิปส์	รุ่น เทรินด์เซ็ต 21 นิ้ว
กรุนดิก	รุ่น 27 นิ้ว
โซนี่	(ของธุรการภาควิชาอิเล็คทรอนิคส์)

4.1.2) เครื่องเล่นวีดีโอ

มิตซูบิชิ	รุ่น E11
เนชันแนล	รุ่น NV20
เนชันแนล	(ของธุรการภาควิชาอิเล็คทรอนิคส์)

4.1.3) เครื่องเล่นสเตอริโอ

โซนี่	รุ่น FH 616R

4.2) ผลการทดลอง

4.2.1) สามารถติดต่อของสัญญาณควบคุมจากเครื่องควบคุมระยะไกลของเครื่องใช้ไฟฟ้าต่าง ๆ ดังที่กำหนดไว้ในการทดลอง

4.2.2) สามารถนำข้อมูลของสัญญาณควบคุมที่ได้บันทึกเอาไว้ในหน่วยความจำส่งเป็นสัญญาณควบคุมเครื่องใช้ไฟฟ้าต่าง ๆ ที่ใช้ในการทดลองได้ทุกเครื่องและหน้าที่การทำงานอย่างถูกต้อง

4.2.3) สามารถจัดเก็บสัญญาณจากเครื่องควบคุมระยะไกลภายนอกได้ 64 สัญญาณ

4.2.4) สามารถส่งสัญญาณควบคุมได้ในระยะทางไกล 10 - 12 เมตร

4.2.5) มีการแสดงผลสภาวะการทำงานทางจอแสดงผลชนิด LCD โดยจะแสดงว่าขณะนั้นเครื่องกำลังทำงานอยู่ในโหมดและหน้าที่การทำงานใด

4.2.6) เป็นระบบอิสระควบคุมการทำงานด้วยไมโครโปรเซสเซอร์ (stand

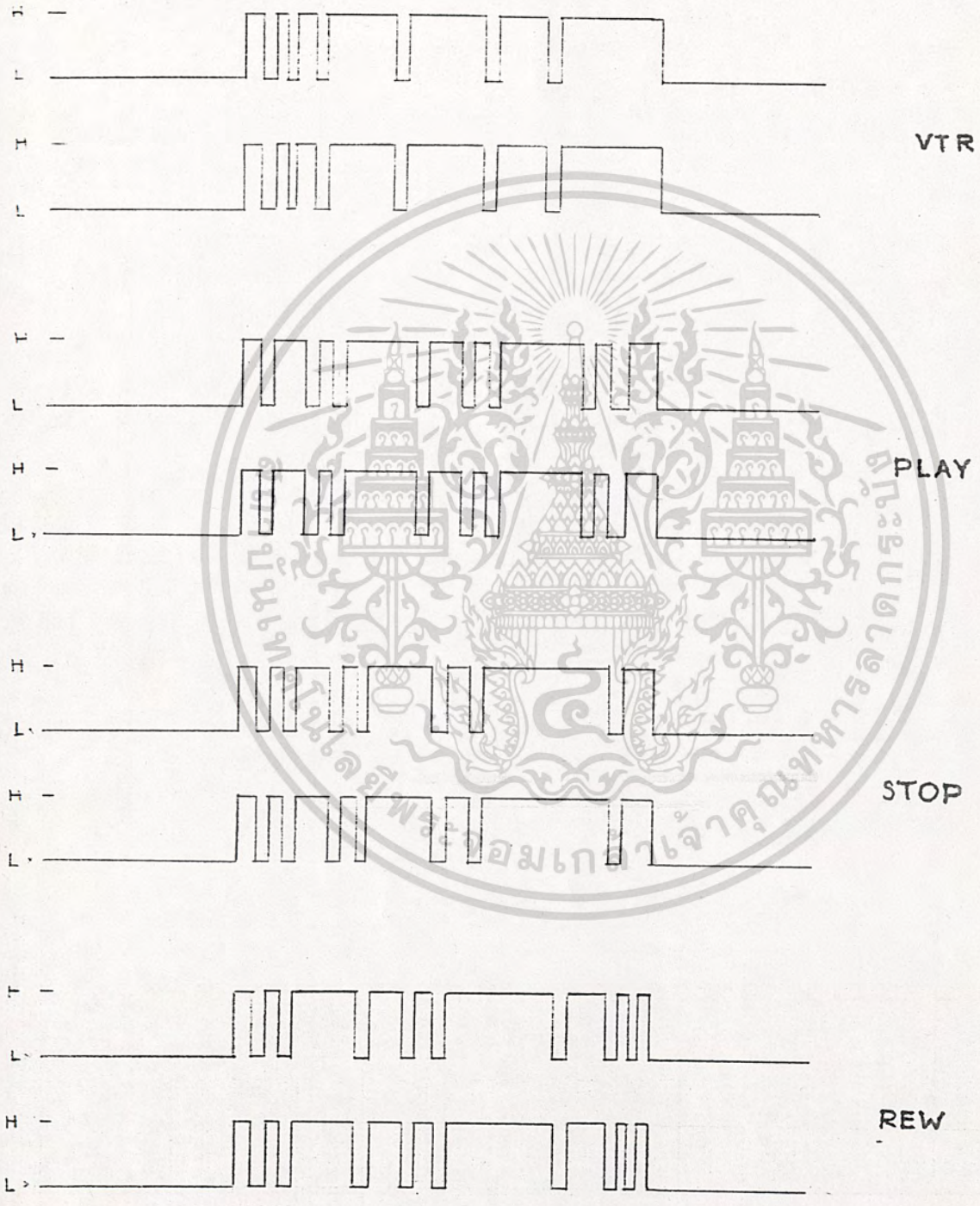
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

alone)

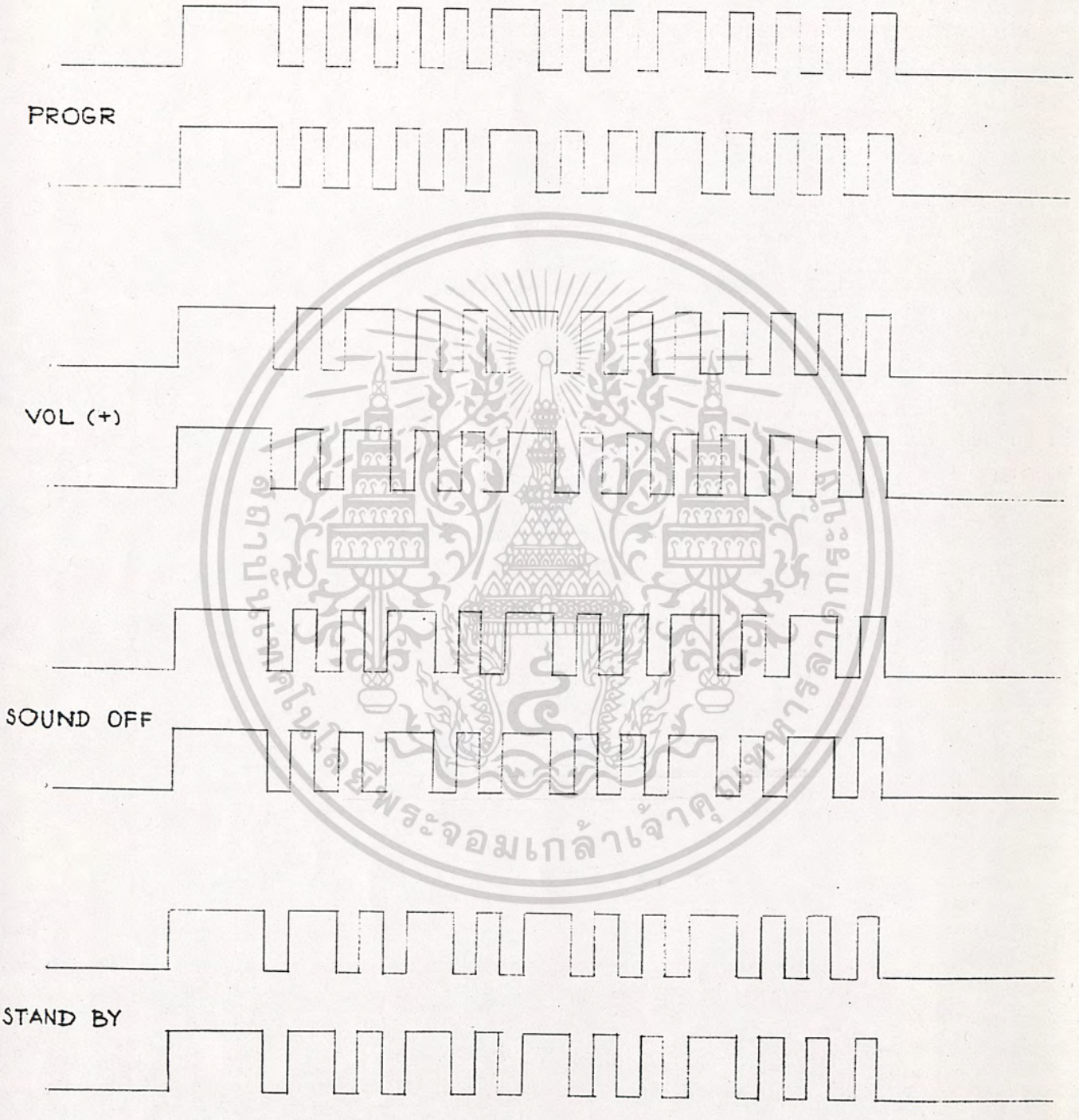
4.2.7) มีแหล่งจ่ายไฟกระแสตรงภายในตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างผลการทดลอง เปรียบเทียบสัญญาณขาออกกับเอาต์พุต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและวิจารณ์

โดยสรุปแล้วผลการทดลองอยู่ในระดับที่น่าพอใจมาก คือสามารถทำการควบคุมเครื่องใช้ไฟฟ้าต่าง ๆ ในการทดลองทุกชนิดได้อย่างถูกต้อง โดยมีการแสดงผลหรือสภาวะการทำงานของระบบด้วยจอแสดงผลชนิด LCD ทำให้ผู้ใช้สามารถใช้งานได้อย่างสะดวกและง่ายขึ้น

ในการออกแบบสร้างโครงงานนี้ประสบปัญหาบางประการซึ่งผู้จัดทำได้ทำการแก้ไขและพัฒนาปรับปรุงให้ดีขึ้น จึงมีความเห็นควรที่จะเสนอแนะไว้ ณ ที่นี้ เพื่อเป็นแนวทางในการพัฒนาแก่ผู้ที่สนใจหรือนำโครงงานนี้ไปใช้งาน ให้ได้เข้าใจถึงจุดที่เกิดปัญหาต่าง ๆ ดังนี้

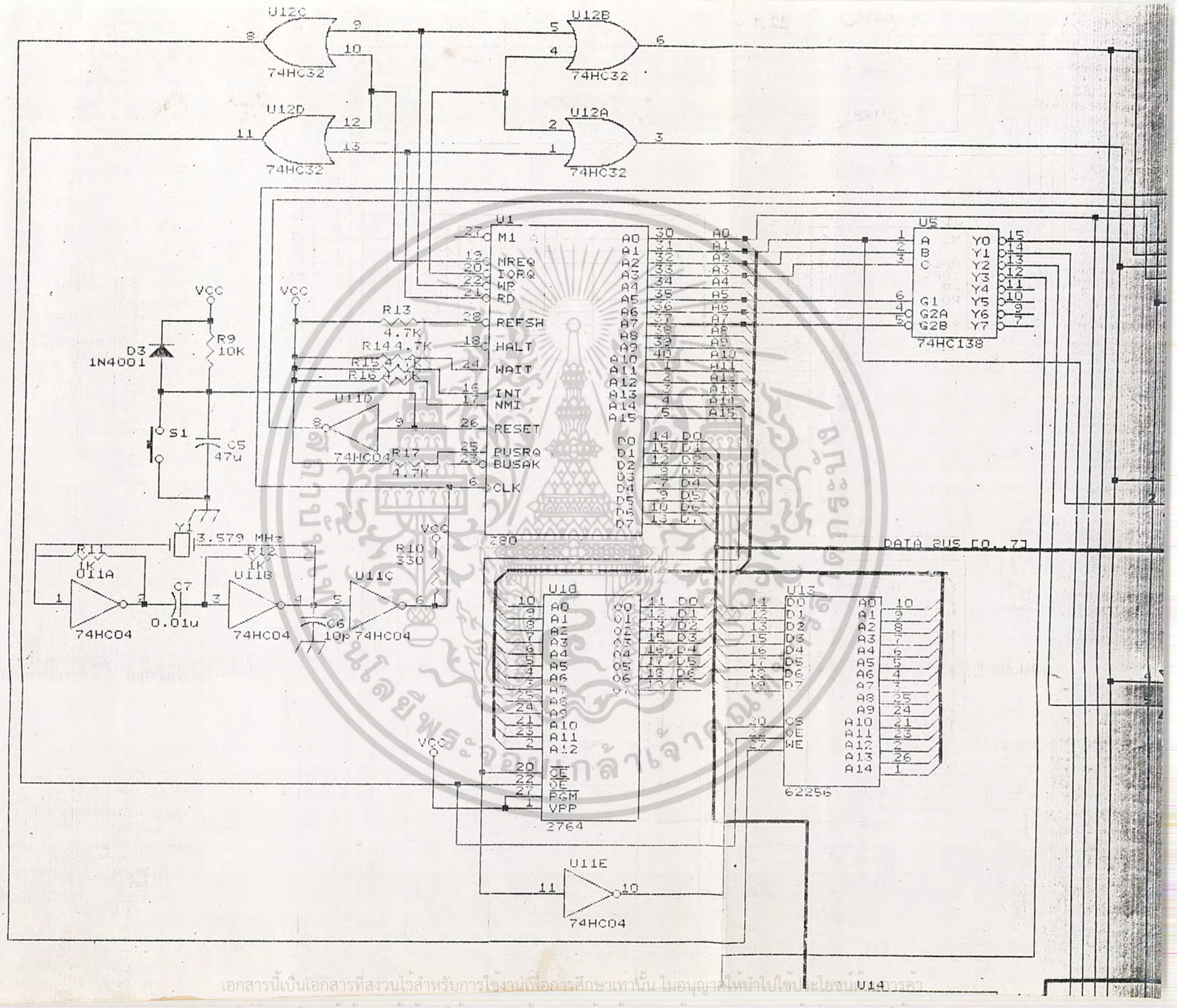
1) เนื่องจากขนาดโครงงานที่ทำนี้เป็นต้นแบบ (Prototype) จึงยังมีขนาดที่ค่อนข้างใหญ่อยู่ จึงยังไม่เหมาะสมและสะดวกในการนำไปใช้งานจริงนัก การพัฒนาในรูปแบบขั้นต่อไปจึงควรทำให้ขนาดโครงงานเล็กลง โดยอาจจะมีการนำไมโครโปรเซสเซอร์แบบชิงเกิลชิปมาใช้ในวงจร ซึ่งสามารถจะทำให้ขนาดของวงจรเล็กลงได้

2) ทางด้านซอฟต์แวร์ ควรที่จะได้มีการพัฒนาโปรแกรมภายในที่จะสามารถทำให้การใช้งานเป็นไปอย่างสะดวก และง่ายต่อการใช้งานยิ่งขึ้น เช่นรูปแบบการแสดงผล การบันทึกสัญญาณ และการใช้งานในโหมดต่าง ๆ

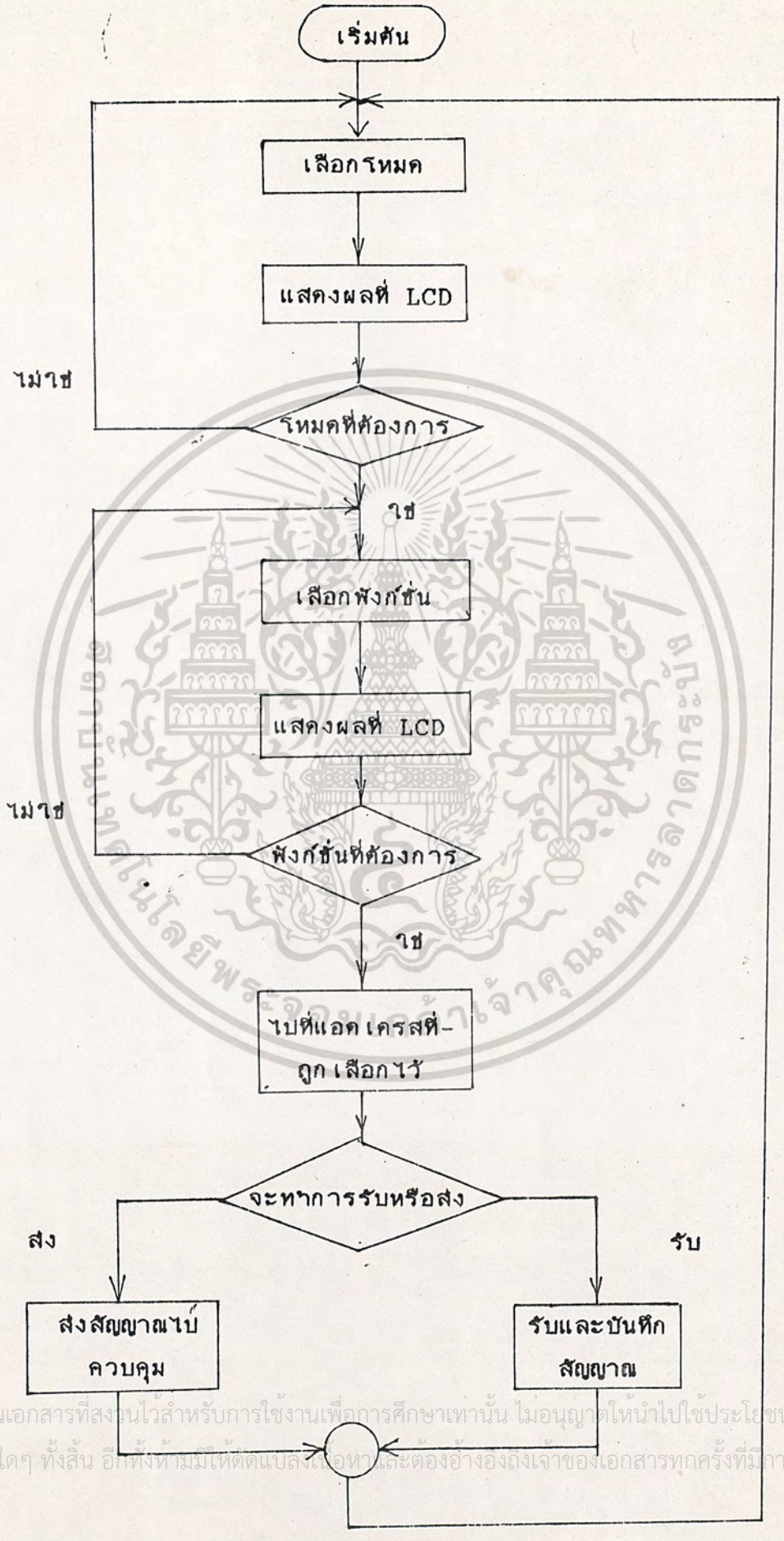
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จไปด้วยดีด้วยความช่วยเหลือของบุคคลหลาย ๆ ฝ่ายดังต่อไปนี้

ผศ. พลพริ้ง ผดุงกุล

อาจารย์ วิชา ศรีธัญญาพงศ์

พี่มือก พี่เล็ก พี่อ้อง

เพื่อน ๆ และน้อง ๆ ที่น่ารักทุก ๆ คน

จึงใคร่ขอขอบคุณมา ณ ที่นี้ด้วย

วีรพล พัวพันธ์

อำพล โสภัคระกูลวัฒน์

อิษฎ์ ศิริสวัสดิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

เอกสารอ้างอิงที่เป็นภาษาไทยและภาษาอังกฤษ

1). รณชัย จันทรานี, "อิเลคโทรนิคส์จูนเนอร์,อุลตร้าโซนิค-อินฟราเรด รีโมทคอนโทรล, ออโตเมติคจูนนิ่งซิสเต็ม", โรงเรียน อาร์.ที.ที.เอส เฉลิมกรุง

2). "คู่มือ ไอซี CMOS 4000 SERIES", บริษัท ซีเอ็ดยูเคชั่น จำกัด, พิมพ์ครั้งที่ 3, 1987

3). "คู่มือลูซาไอซี", บริษัท ซีเอ็ดยูเคชั่น จำกัด, พิมพ์ครั้งที่ 3, 1988

4). "Microcommunications Handbook", Intel Corporation, 1988

5). James w. Coffron, "Z80 APPLICATIONS", SYBEX Inc., 1983

6). "Optoelectronics Data Book", Texas Instruments Incorporated, 1983-84

7). Lance A. Leventhal, "Z80 Assembly Language Programming", Osborne/McGraw-Hill, 4th Printing, 1988

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้