ปีการศึกษา 2533

เครื่องวัดความสูงด้วย ULTRASONIC

โดย

ประสิทธิ์ ครองกิจศิริ 323410

ระพี จันทร์กระจ่าง 323416

อาจารย์ที่ปรึกษา

ผศ. นิกร สุขุตมตันติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชา "เทคโนโลยีอุตสาหกรรม"

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง "เครื่องวัดความสูงด้วย ULTRASONIC"

ผู้จัดทำ

1. ประสิทธิ์ ครองกิจศิริ 323410
2. รชพี จันทร์กระจ่าง 323416

อาจารย์ที่ปรึกษา

(ผศ.นิกร สุขุตมตันติ)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดความสูงด้วย ULTRASONIC

ประสิทธิ์ ครองกิจศิริ
ระพี จันทร์กระจ่าง
นิกร สุขุตมตันติ อาจารย์ที่ปรึกษา
ปีการศึกษา 2533

บทคัดย่อ

โครงงานนี้มีชื่อว่าเครื่องวัดความสูงอุลต้าโซนิค เป็นการนำเอาMicroprocessor มาประยุกต์ใช้งานโดยใช้หลักการสะท้อนของคลื่นเสียง โดยนำไมโครโปรเซสเซอร์มาทำการ INTERFACE กับ วงจรรับและส่ง ULTRASONIC การทำงาน CPUจะทำการ ON เครื่องส่ง และ ทำการนับพร้อมทั้งเช็ค INPUT ว่าคลื่นสะท้อนกลับมาหรือยังค่าที่ได้จะเป็นระยะทาง ซึ่งจะเก็บไว้เป็น REFERENCE เมื่อจะทำการวัดคน จะนำค่าที่นับได้มาทำการลบกับค่า REFERENCE ก็จะได้เป็นค่าความสูง ซึ่งสามารถเก็บไว้และเรียกดูได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ULTRASONIC HEIGHT MEASURAMENT

PRASIT KHONGKITSIRI
RAPEE CHANKRACHANG
NIKORN SUKUTAMATANTI ADVISOR

ACADEMIC YEAR 1990

Abstract

This project is named Ultrasonic height measurament, That take the Z-80 Microprocessor interfece with transmitter-receiver Ultrasonic circuit. Method operation, CPU will turn on transmitter and count Timmer while check input it the wave reflect back. The count valne is the distant and reference. When measure the height, take the reference minus by counter value that the height of men. It can record and recall as well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทนำ 1

บทที่ 1 ทรานสดิวเซอร์ในภาครับและภาคส่ง 2
1.1 อุลตร้าโซนิคทรานสดิวเซอร์ตัวรับและตัวส่ง 3
1.2 วงจรภาคส่ง 9
1.3 วงจรภาครับ 16

บทที่ 2 การใช้ ROM กับ Z-80 27
2.1 ROM คืออะไร 27
2.2 คุณสมบัติที่สำคัญในการทำงานของ ROM 29
2.3 ขั้นตอนในการอ่านข้อมูลจาก ROM 31
2.4 การต่อ ROM กับ Z-80 บัส 32
2.5 ผังแอดเดรส (ADDRESS MAPPING) 33
2.6 การสร้างสัญญาณ CHIP SELECT สำหรับ ROM. 34
2.7 การสร้างสัญญาณ สำหรับ หน่วยความจำ 36
2.8 การต่อ CHIP SELECT LINES 37
2.9 วิธีการใช้ ROM หลายๆ ตัว 39
2.10 ผังหน่วยความจำของ ROM ชนิดความจุมาก 40
2.11 การเพิ่มบัฟเฟอร์ให้กับแอดเดรสบัส 41
2.12 การเพิ่มบัฟเฟอร์ให้กับบัสข้อมูล 43
2.13 ตัวอย่างของระบบที่สมบูรณ์ที่ใช้กับ ROM 43

บทที่ 3 การใช้ STATIC RAM กับ Z-80 47
3.1 ลักษณะการทำงานของ STATIC RAM. 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนที่เกิดขึ้นเมื่อทำการอ่าน RAM. 49
3.3 ขั้นตอนที่เกิดขึ้นเมื่อทำการเขียนข้อมูลเข้าไปใน RAM..... 50
3.4 อุปกรณ์หน่วยความจำที่ใช้งานจริง 51
3.5 ลำดับขั้นตอนการเขียนข้อมูลเข้าไปในตัว 2114 54
3.6 ลำดับขั้นตอนการอ่านข้อมูลออกจาก 2114 55
3.7 การต่อ ADDRESS LINE เข้ากับ Z-80 56
3.8 การต่อบัสข้อมูลโดยไม่ใช้บัฟเฟอร์ 59
3.9 การสร้าง CONTROL LINE สำหรับอ่านและเขียนหน่วยความจำ..... 60
3.10 การใช้บัฟเฟอร์บนบัสข้อมูลกับ STATIC RAM. 62
3.11 ระบบ 4K x 8 - BIT STATIC RAM ที่สมบูรณ์ 63
3.12 หน่วยความจำ STATIC RAM เบอร์ 6116 63

บทที่ 4 เครื่องวัดความสูงด้วยอุลตร้าโซนิค 69
4.1 คุณสมบัติของเครื่องวัดความสูงด้วยอุลตร้าโซนิค 70
4.2 BLOCK DIAGRAM 71
4.3 การวางแผนการสร้าง 72
4.4 วงจรการทำงาน 75
4.5 การทำงานของเครื่อง 78
4.6 การทดลอง 84

สรุป 85

ภาคผนวก

กิตติกรรมประกาศ 86
หนังสืออ้างอิง 87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทนำ

โครงงานนี้ให้ชื่อว่า "เครื่องวัดความสูงด้วย ULTRASONIC " โดยแนวความคิดของเราคือต้องการมีเครื่องวัดความสูงที่สามารถเก็บ DATA ได้ด้วยซึ่ง DATA จะเก็บเรียงลำดับตามกันไป สามารถเลือกเก็บ DATA เป็นช่วงได้

เราจะใช้ ULTRASONIC ที่ความถี่ 40 KHz โดย CPU จะเป็นตัวส่งสัญญาณ TRIG ออกไปเพื่อให้ตัวส่งเริ่มส่งสัญญาณออกไปแล้ว CPU จะเริ่มทำการ COUNT ตัวนั้นขึ้นเรื่อย ๆ จนกว่าตัวรับสัญญาณเข้ามาก็จะนำค่าที่ COUNT ได้นั้นมาทำการหาระยะทาง ซึ่งจะต้องทดลองก่อน เพื่อหาค่าอัตราส่วนที่ถูกต้องซึ่งระยะทางที่ได้ จะเป็นระยะทางจากเครื่องวัดถึงศีรษะของผู้วัด เพราะฉะนั้นความสูงจริง ๆ ของคนวัดคือ ความสูงของเครื่องวัดจากพื้นดินลบด้วยค่าระยะทางทางที่วัดได้ ก็จะเป็นความสูงจริงของคน ๆ นั้น

การนำ CPU มาใช้นั้นทำให้ระบบลดความยุ่งยากลงไปมากเพราะระบบของ CPU ใช้ IC น้อยตัว และราคาก็ถูกด้วย

การทำงาน FUNCTION ต่าง ๆ ก็ขึ้นอยู่กับ SOFT WARE ที่เราเขียนเข้าไป การที่เราเลือก Z-80 เพราะเป็น CPU ที่เราคุ้นเคย INTRODUCTION SET ของมันอยู่แล้วทำให้คล่องในการเขียน PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ทรานสดิวเซอร์ในภาครับและภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 อุลตร้าโซนิคทรานสดิวเซอร์

สาเหตุที่มีการนำเอาคลื่นย่านอุลตร้าโซนิคมาใช้เพราะว่า เป็นคลื่นที่มี ทิศทาง ทำให้เราสามารถเล็งเสียงไปยังเป้าหมายที่ต้องการได้โดยเจาะจง เป็นคุณสมบัติอย่างหนึ่งของคลื่น ยิ่งคลื่นมีความถี่สูงขึ้น ความยาวคลื่นก็จะยิ่งสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิด (ที่ให้เสียงนั้นออกมา) ของตัวที่ห้กำเนิดเสียงความถี่นั้น เช่น คลื่นความถี่ 300 Hz ในอากาศ จะมีความยาวถึงประมาณ 1 เมตรเศษ ๆ ซึ่งจะยาวกว่าช่วงที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียงโดยทั่วไปมากมาย คลื่นจะหักเบนที่ขอบด้านนอกของตัวกำเนิดเสียงทำให้เกิดการกระจายทิศทางคลื่น แต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอุลตร้าโซนิค เช่น 40 KHz จะมีความยาวคลื่นในอากาศเพียงประมาณ 8 มม เท้านั้น ซึ่งเล็กกว่ารูเปิดของตัวที่ให้กำเนิดเสียงความถี่นี้มาก ๆ คลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบ จึงพุ่งออกมาเป็นลำเคบ ที่เราเรียกว่ามีทิศทาง

การมีทิศทางของคลื่นเสียงย่านอุลตร้าโซนิคทำให้เรานำไปใช้งานได้หลายอย่าง เช่นนำไปใช้ในเครื่องควบคุมระยะไกล (ULTRASONIC REMOTE CONTROL) เครื่องล้างอุปกรณ์ (ULTRASONIC CLEANER)

อุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไปมา ซึ่งทำให้เกิดคลื่นเสียงอุตร้าโซนิคกระจายไปในอากาศได้ หรือแปลงพลังงานในรูปอื่นได้ มีชื่อตามเทคนิคว่า อุลตร้าโซนิคทรานสดิวเซอร์ (ULTRASONIC TRANSDUSER)

แบบของอุลตร้าโซนิคทรานสดิวเซอร์

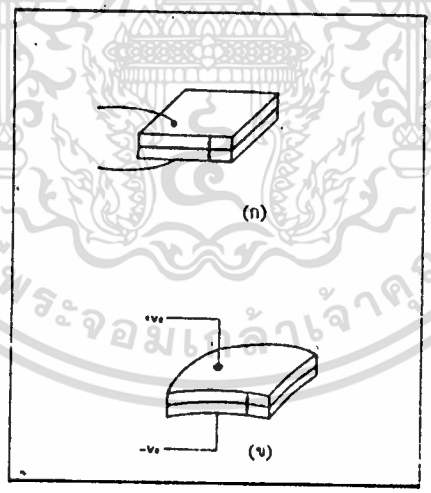
- เปียโซอิเลกทริก (PIEZO ELECTRIC TRANSDUCER) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าและพลังงานกลโดยมีความถี่รีโซแนนซ์ที่อยู่ค่าหนึ่ง

- แมกนีโตสตริคทีฟ (MAGNETOSTRICTRICE TRANSDUCER) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่

- อิเลคโตรสตริคทีฟ (ELECTROSTRICTIVE TRANSDUCER) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล

ทรานสดิวเซอร์แบบเปียโซอิเลคทริค

ภายในตัวอุลตร้าโซนิคทานสดิวเซอร์แบบเปียโซอิเลคทริค แบบที่มีใช้กันในปัจจุบัน จะประกอบด้วยชิ้นสารเซรามิคสี่เหลี่ยมซึ่งมีผิวโลหะเงินฉาบอยู่ทั้ง 2 หน้า เพื่อให้ต่อสายไฟออกมาเป็น 2 ขา ชิ้นสารเซรามิคนี้ประกอบขึ้นจากสารเซรามิค 2 ชิ้นประกอบกันอยู่โดยวางให้ขั้วไดโพลทางไฟฟ้าในอะตอมของมันมีทิศทางตรงกันข้ามกันดังแสดงเป็นลูกศรในรูปที่ 1



รูปที่ 1 ก. โครงสร้างภายในตัวอุลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคทริคที่ใช้สารเซรามิค (ข) เมื่อป้อนแรงดันให้กับตัวมัน จะทำให้ชิ้นสารเซรามิคโก่งงอไปมาทำให้เกิดคลื่นเสียงอุลตร้าโซนิคกระจายไปในอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ชิ้นสารเซรามิคถูกยึดติดภายในตัวถังอย่างดีเพื่อไม่ให้เกิดการสั่นขณะที่มันทำงานอยู่ได้รับผลกระทบกระเทือนจากภายนอกตัวถังจะเป็นรูปทรงกระบอกที่มีเส้นผ่าศูนย์กลางและมีความยาวประมาณ 1 ถึง 2.5 ซม ด้านหน้าทำเป็นช่องเปิดมีตระแกรงติดอยู่เพื่อทำให้คลื่นอุลตร้าโซนิคเข้ามาหรือออกไป

เมื่อมีสัญญาณแรงดันมาตกคร่อมขั้วทั้งสองของชิ้นสาเซรามิค ดังในรูป 1(ข) จะทำให้ชิ้นสารโก่งงอมากหรือน้อย หรือในทิศทางใดตามขนาดและทิศทางการเปลี่ยนแปลงขนาดของสัญญาณนั้น ๆ ทำให้เกิดการกดอัดอากาศโดยรอบเกิดเป็นคลื่นเสียงที่มีความถี่เดียวกับสัญญาณนั้นออกไป โดยทั่ว ๆไปกำลังเอาท์พุทที่ออกมาจะตกประมาณ 10 % ของกำลังไฟฟ้าที่ที่ป้อนเข้าไป แต่กำลังเอาท์พุทจะสูงสุดที่ค่าประมาณนี้ต่อเมื่อความถี่ของสัญญาณตรงกับความถี่เรโซแนนซ์

ในทำนองกลับกันเมื่อมีคลื่นเสียงที่มีความถี่รีโซแนนซ์ของชิ้นสารเซรามิคเข้ามาจะทำให้ชิ้นสารโก่งงอไปมาและเกิดสัญญาณแรงดันซึ่งมีขนาดเล็กขึ้นมาคร่อมขั้วทั้งสองของตัวมันเอง

<u>ตัวส่งและตัวรับ</u>

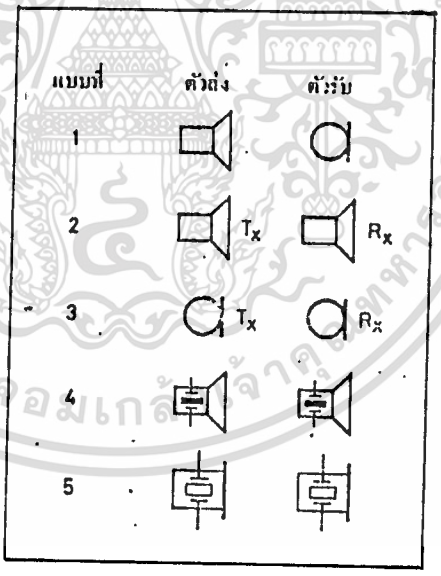
ทรานสดิวเซอร์แบบเปียโซอิเลคทริคที่ใช้สารเซรามิค จะมีอยู่ 2 อย่าง คือ ตัวส่ง (TRANSMITTER และ ตัวรับ (RECEIVER)

ตัวส่ง ก็คืออุลตร้าโซนิคทรานสดิวเซอร์ที่ถูกออกแบบเจาะจงมาให้แปลงสัญญาณไฟฟ้าที่ให้ตัวมัน ให้ออกมาเป็นคลื่นเสียงอุลตร้าโซนิค หน้าที่ของตัวส่งจึงคล้าย ๆ กับเป็นลำโพง ส่วนตัวรับก็คืออุลตร้าโซนิคทรานสดิวเซอร์ที่ถูกออกแบบเจาะจงมาให้แปลงคลื่นเสียงย่านอุลตร้าโซนิคที่มาตกกระทบตัวมันให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจึงคล้าย ๆ กับเป็นไมโครโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุนี้เวลาเขียนสัญญาลักษณะของอุลตร้าโวนิคทรานสดิวเซอร์จึงนิยมเขียนตามหน้าที่ของมันคือถ้าเป็นตัวส่งก็เขียนสัญญาลักษณ์เป็นลำโพง ถ้าเป็นตัวรับก็เขียนสัญญาลักษณ์เป็นไมโครโฟน ดังแบบที่ 1 ในรูปที่ 2 แต่ก็มีหนังสือบางเล่มเขียนสัญญาลักษณ์ทั้งตัวรับตัวส่ง เป็นไมโครโฟนหรือลำโพงอย่างใดอย่างหนึ่งดังในรูปที่ 2 และ 3 แต่เขียนตัวอักษรย่อว่า Tx(TRANSMITTER) และ Rx (RECEEIVER) กำกับอยู่ด้วย

อุลตร้าโซนิคทรานสดิวเซอร์แบบเซรามิคที่มีอยู่ในตลาดจะมีความถี่รีโซแนนซ์ให้เลือกตั้งแต่ 23 KHz ขึ้นไปจนถึง 40 KHz โดยความถี่ 40 KHz เป็นที่นิยมมากที่สุด



รูปที่ 2 แสดงตัวอย่างการเขียนสัญญลักษณ์ของอุลตร้าโซแบบต่าง ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

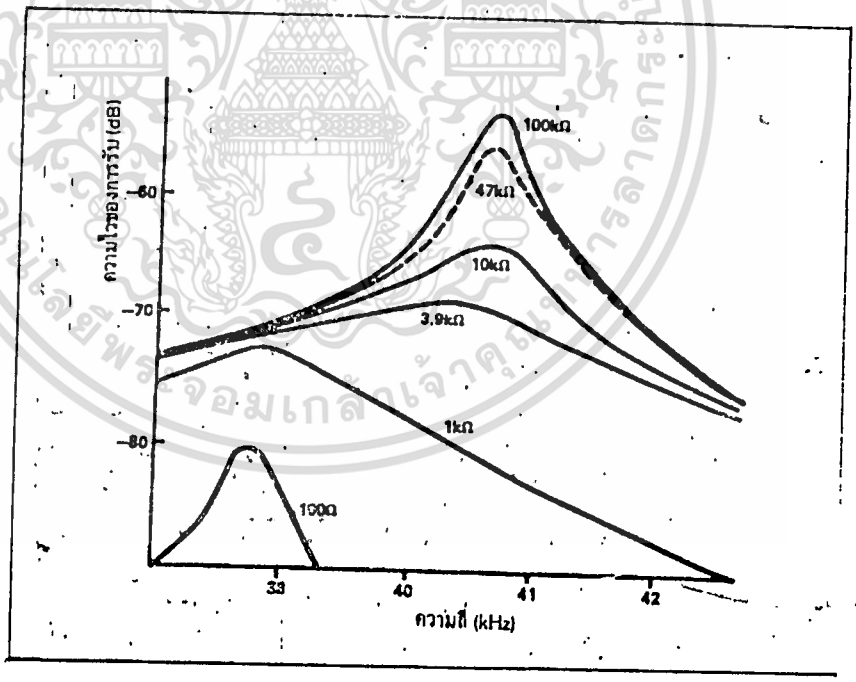
ข้อควรรู้ในการใช้งานตัวส่งและตัวรับ

-ไม่ควรทำให้ตัวทรานสดิวเซอร์ได้รับการกระแทก เพื่อป้องกันโครงสร้างภายในมิให้เสียหาย

-ทรานสดิวเซอร์ที่มีใช้กันทั่วไปจะทนแรงดันคร่อมตัวมัน สูงสุดได้ไม่เกิน 20 Vrms ขนาดของสัญญาณที่ป้อนให้กับตัวทรานสดิวเซอร์ควรไม่เกินนี้

-ความถี่รีโซแนนซ์ (ความถี่ที่ตัวมันทำงานได้ประสิทธิภาพสูงสุด) ของทรานสดิวเซอร์ 40 KHz ที่มีทั่วไปจะผิดพลาดไปไม่เกิน บวกลบ 1 KHz และมีแถบความถี่ (BRANWIDTH) ประมาณ 4.5 KHz สำหรับตัวส่ง และมีแถบความถี่ประมาณ 5.0 KHz สำหรับตัวรับ

-อุณหภูมิใช้งานของทราสดิวเซอร์ควรอยู่ภายในช่วง -20 °C ถึง +60 °C



รูปที่ 3 แสดงผลการทดสอบตัวรับตัวหนึ่งโดยลองเปลี่ยนโหลดเป็นค่าต่าง ๆ กัน แล้วป้อนคลื่นเสียงความถี่ต่าง ๆ กันเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ที่ตำแหน่งเบนจากแกนของตัวส่งไปประมาณ 30 องศา ความแรงของคลื่นเสียงที่ส่งออกไปจะลดลงจากแนนแกนประมาณ 10 db ในทำนองเดียวกันถ้าคลื่นเสียงที่พุ่งเข้ามา ในแนวที่เบี่ยงเบนไปจากแแนวของตัวรับไปประมาณ 30 องศา ความไวหรือขนาดแรงดันที่ออกมาก็จะลดลงประมาณ 10db ในการใช้งานที่เป็นการควบคุมระยะไกลในที่โล่งแจ้ง จึงควรพยายามให้ทั้งตัวรับและตัวส่งอยู่ในแนวที่พุ่งเข้าหากันมากที่สุด

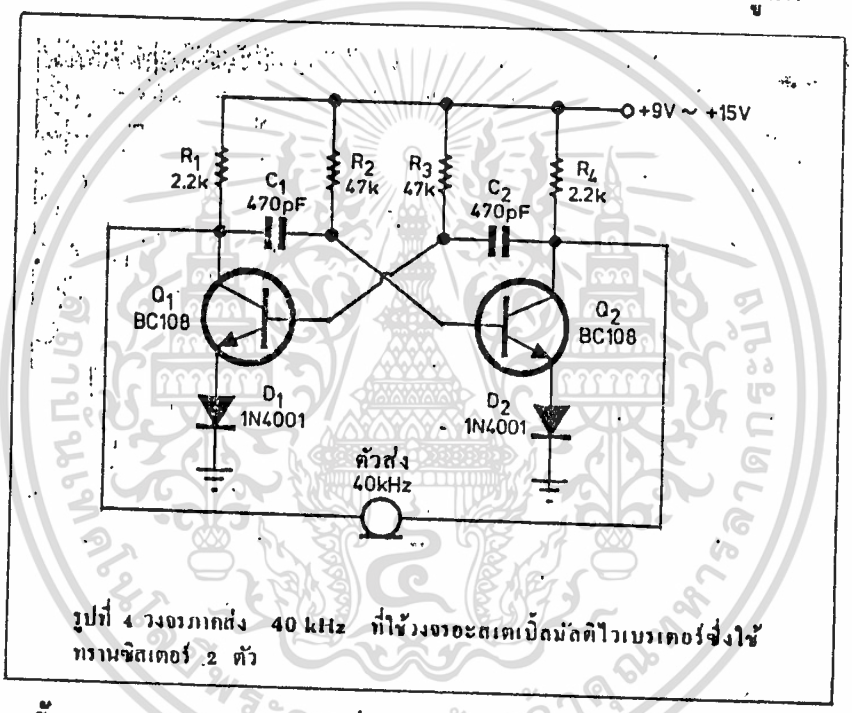
-ในกรณีที่ใช้งานตัวรับจะต้องมีความต้านทานต่อขนานกับตัวรับเพื่อทำหน้าที่เป็นโหลด ความต้านทานมีค่าอยู่ในช่วงจาก 10 k ~ 100 k จากการทดลองพบว่าถ้าเปลี่ยนโหลดจาก 100 k มาเป็น 10 kความไวจะลดลงประมาณ 10 ถึง 12 db แต่แถบความถี่จะกว้างขึ้น ถ้าใช้ความต้านทานต่ำกว่านี้ ความถี่เรโซแนนซ์จะลดลงไปจากที่ระบุ

-โดยปกติสามารถนำเอาตัวส่งตัวรับมาใช้งานแทนกันได้ เพียงแต่ให้มีความถีรีโซแนนซ์เดียวกันเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วงจรภาคส่ง

วิธีการสร้างเครื่องส่งเสียงอุลตร้าโซนิคนั้นไม่มีอะไรยาก ยงแต่หาวงจรกำเนิดสัญญาณรูปสี่เหลี่ยมหรือซายน์หรือเป็นรูปพัลซ์ก็ได้ ที่สามารถสร้างความถี่ ที่ตรงกับความถี่เรโซแนนซ์ ของอุลตร้าโซนิคทรานสดิวเซอร์ตัวส่งที่จะใช้แล้วต่อตัวส่งคร่อมขั้วเอาท์พุทเพื่อทำหน้าที่เป็นโหลดทำนองเดียวกับการต่อลำโพงเท่านั้นเอง วงจรภาคส่งอาจจะเป็นวงจรออสซิลเลเอตร์ ที่ใช้ทรานซิสเตอร์ตัวเดียว โดยมีขดลวดทำหน้าที่ป้อนกลับให้เกิดการออสซิลเลทก็ได้ แต่การที่ต้องใช้ขดลวดเฉพาะอาจทำให้หลายท่านยุ่งยากใจอยู่บ้าง ดังนั้นจึงอาจเลี่ยงมาเริ่มต้นที่วงจรอะสเตเบิ้ลมัลติไวเบรเตอร์ดังแสดงในรูปที่ ๔



รูปที่ 4 วงจรภาคส่ง 40 kHz ที่ใช้วงจรอะสเตเบิ้ลมัลติไวเบรเตอร์ซึ่งใช้ทรานซิสเตอร์ 2 ตัว

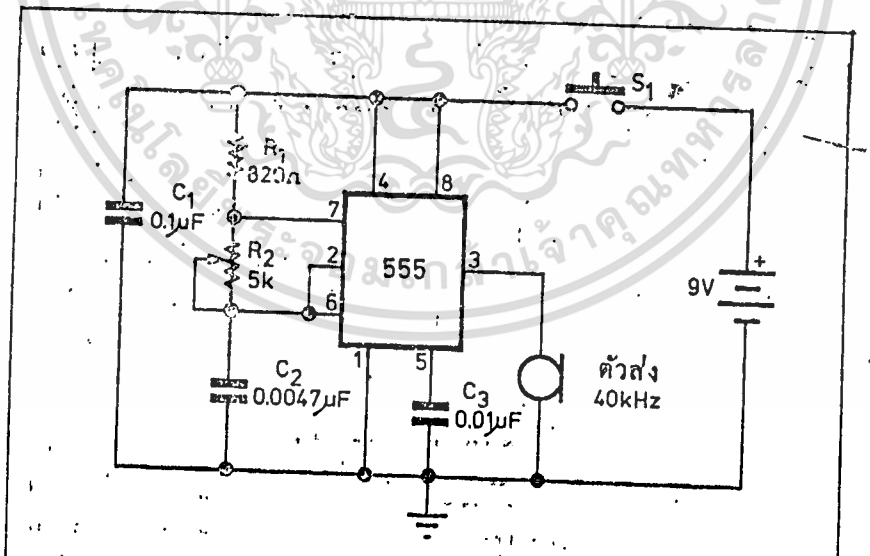
วงจรหน้าตาอย่างนี้ นักอีเลคทรอนิคส์รู้จักกันเป็นอย่างดีแล้ว จึงจะไม่อธิบายอะไรกันมากมายเพียงแต่ขอให้สังเกตว่าเขาต่อทรานสดิวเซอร์เข้าระหว่าง ขาคอลเลคเตอร์ทั้งสอง ทั้งนี้เพราะตัวทรานสดิวเซอร์มีค่าความต้านทานไฟตรงสูงมาก ขืนเอาไปต่อทำนองเดียวกับ R1 หรือ R4 จะไม่ทำงานนอกจากนั้น การต่อเช่นนี้ทำให้ขนาดสัญญาณสี่เหลี่ยมที่เกิดขึ้นคร่อมตัว ทรานสดิวเซอร์สามารถสูง (จากยอดคลื่นถึงยอดคลื่น) ขึ้นไปได้ถึงประมาณเกือบ 2 เท่าของแรงดันไฟเลี้ยง เพื่อให้ขนาดสัญญาณอุลตร้าโซนิคออกมาได้แรงมากขึ้นกว่าปกติ ค่าอุปกรณ์ต่าง ๆ ที่ให้ไว้นั้นบริษัท ฟิลิปส์-มูลลาร์ด เจ้าของ

028737
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเขาออกแบบมาให้ใช้กับตัวส่ง 40 kHz โดยค่าความถี่ของวงจรถูกกำหนดโดยค่าของ C1, C2, R2 และ R3 ซึ่งตามค่าเหล่านี้จะสร้างความถี่ต่ำกว่าที่ต้องการเล็กน้อย เมื่อตัวส่งเริ่มต้นทำงานโดยสั่นขึ้นมาแล้ว จะเกิดแรงดันขึ้นมากระตุ้นทรานซิสเตอร์ตัวใดที่หยุดนำกระแสให้นำกระแสก่อนกำหนด เป็นเหตุให้ความถี่ของวงจรถูกบังคับให้มีค่าสูงขึ้นเป็นการชิงโครไนซ์ ให้ความถี่ของวงจร เท่ากับ ความถี่เรโซแนนซ์ของตัวส่งโดยอัตโนมัติโดยที่ไม่ต้องมีการปรับแต่งความถี่แต่อย่างใด

D1 และ D2 ที่ขาอิมิตเตอร์ของทรานซิสเตอร์ ทำหน้าที่ช่วยจำกัดขนาดแรงดันย้อนกลับที่เกิดขึ้นระหว่าง ขาเบสและขาอิมิตเตอร์ในบางขณะ ไม่ให้สูงเกินกว่าอัตราทนแรงดันย้อนกลับ VEB (max) ของทรานซิสเตอร์ที่ใช้ ในการณีที่ใช้กับไฟเลี้ยง 9 Vdc วงจรจะดึงกระแสประมาณ 5 mA สำหรับท่านที่ต้องการจะดัดแปลงวงจรนี้ให้ไปใช้กับตัวส่ง 25 kHz ให้เปลี่ยนค่า C1 และ C2 เป็น 750 pF (โดยการใช้ 560 pF และ 180 pF มาขนานกัน)

แต่ถ้าท่านสนใจวงจรภาคส่งที่ใช้ไอซีขอให้ดูรูปที่ 5

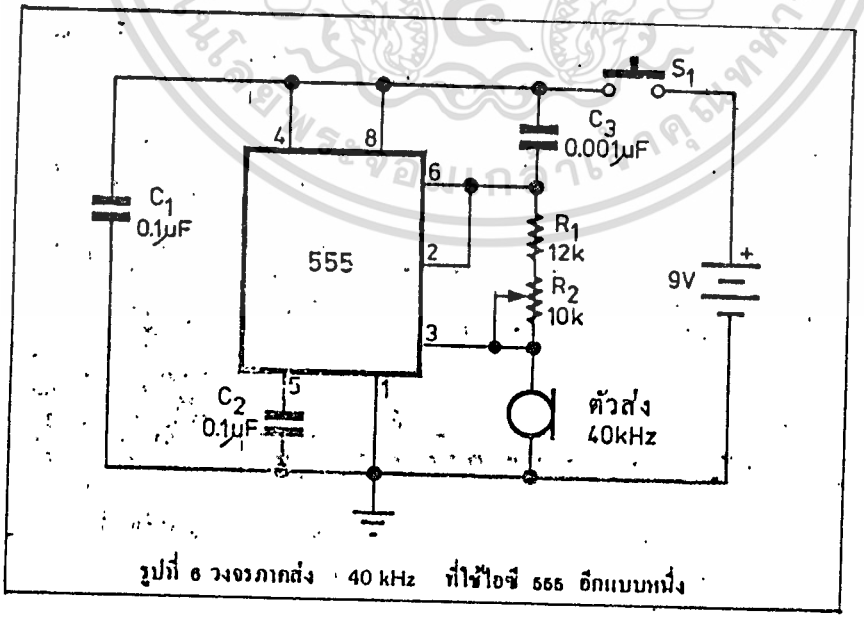


รูปที่ 5 วงจรภาคส่ง 40 kHz ที่ใช้ไอซี 555 มาทำหน้าที่เป็นวงจรอะสเตเบิลมัลติไวเบรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ซึ่งแสดงการนำเอไอซีตั้งเวลาเบอร์ 555 มาต่อเป็นวงจรอษสเตเบิ้ลมัลติไวเบรเตอร์ ซึ่งให้สัญญาณสี่เหลี่ยมออกมาป้อนแก่ทรานสดิวเซอร์ตัวส่งความถี่ 40 kHz ซึ่งต่อเป็นโหลดที่ขาเอาท์พุทของไอซี วงจรจะส่งเสียงอุลตร้าโซนิค ออกมาทุกครั้งที่กดสวิทช์ S1 ให้ไฟเลี้ยงเข้าวงจร R2 มีไว้เพื่อใช้ปรับแต่งความถี่ของการออสซิลเลทของวงจรให้เท่ากับความถี่เรโซแนนซ์ของตัวส่งที่ใช้เพื่อให้ได้กำลังเสียงส่งออกสูงสุด วิธีการปรับแต่งความถี่ทำได้ง่ายโดยต่อมิลลิแอมป์มิเตอร์อนุกรมกับสายไฟเลี้ยงวงจร แล้วปรับค่า R2 จนกระทั่งกระแสที่วงจรดึงไปจากแหล่งจ่ายไฟมีค่าสูงสุด (อย่างไรก็ตามเนื่องจากตัวส่ง และตัวรับอาจจะมีความถี่เรโซแนนซ์แตกต่างกันอยู่บ้าง ดังนั้นการปรับแต่งที่ตรงเป้าหมายที่สุดคือ ให้ผลการทำงานที่ดีที่สุดนั้น ควรจะวางภาคส่งและภาครับให้ตัวรับและตัวส่งพุ่งเข้าหากัน แล้วปรับ R2 ให้ขนาดสัญญาณที่ออกจากวงจรขยายของวงจรภาครับมีขนาดสัญญาณแรงที่สุด) ถ้าต้องการนำไปใช้กับตัวส่ง 25 kHz ให้เพิ่มค่า C2 ขึ้นเป็น 0.01uF หรือไม่ก็เปลี่ยนค่า R2 เป็น 10 k

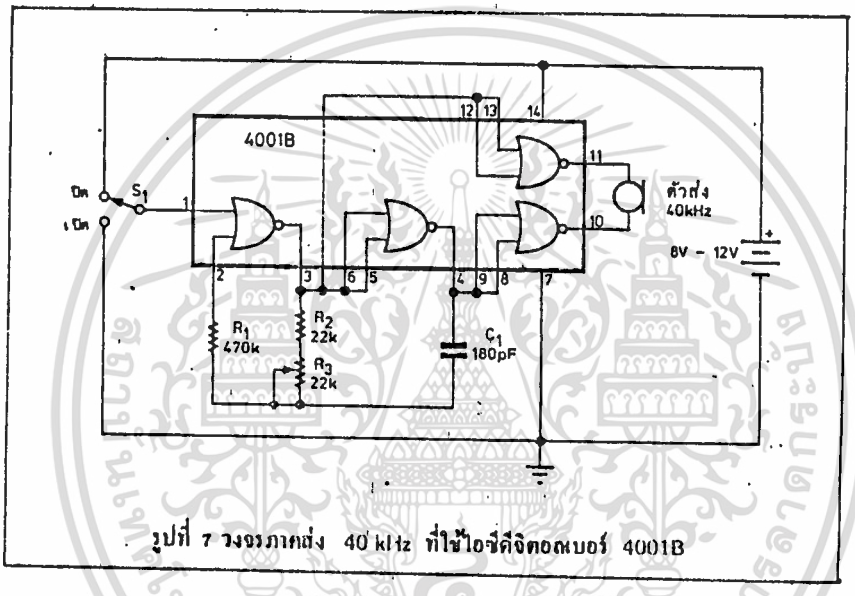
ในรูปที่ 6 แสดงการใช้งาน 555 เป็นวงจรอษสเตเบิ้ลมัลติไวเบรเตอร์อีกแบบหนึ่งที่ดัดแปลงมาจากรูปที่ 5 โดยให้เอาท์พุทขา 3 มาอยู่ในส่วนที่เกี่ยวกับการสร้างรูปคลื่นโดยตรงเลย



รูปที่ 6 วงจรภาคส่ง 40 kHz ที่ใช้ไอซี 555 อีกแบบหนึ่ง

ลักษณะการจัดวงจรเช่นนี้ทำให้ช่วงเวลาทำงาน(duty cycle)ของสัญญาณสี่เหลี่ยมเป็น 1:1 คงที่ตลอดไม่ว่าจะปรับ R2 ไปอย่างใดก็ตาม ในกรณีที่ใช้กับตัวส่ง 25 kHz ให้เพิ่มค่า C3 เป็น 0.0015 uFหรือไม่ก็เพิ่มค่า R1 เป็น 18 k

วงจรภาคส่งที่ง่ายอีกวงจรหนึ่งคือการใช้ไอซีดิจิตอลพวก CMOS เบอร์ 4001B ซึ่งภายในประกอบด้วยนอร์เกท (NOR gate) 4 ตัว ดังแสดงในรูปที่ 7



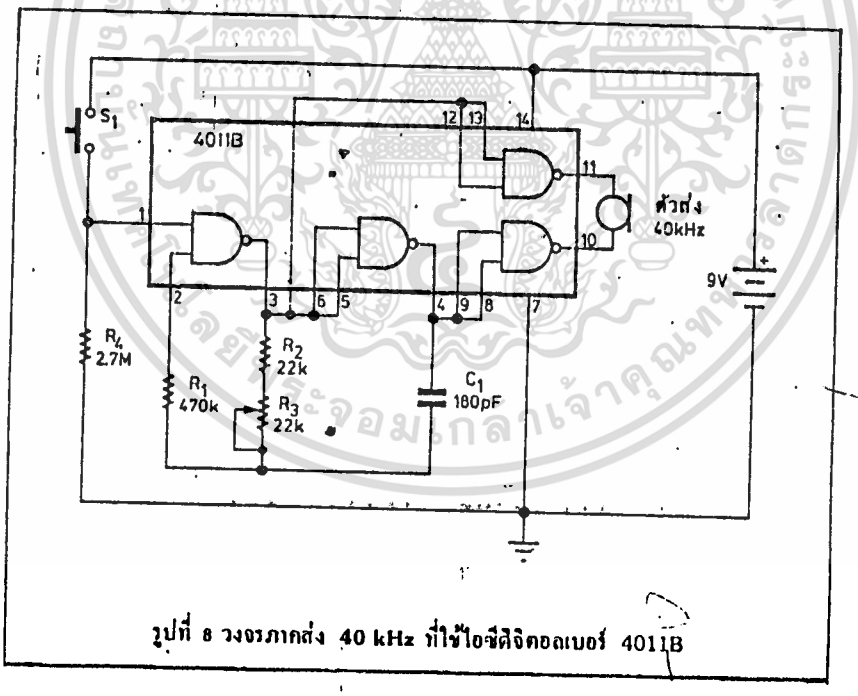
รูปที่ 7 วงจรภาคส่ง 40 kHz ที่ใช้ไอซีดิจิตอลเบอร์ 4001B

ในวงจรนี้เอาเกท 2 ตัวมาทำเป็นวงจรออสซิลเลเตอร์สร้างคลื่นรูปสี่เหลี่ยมขึ้นมา เกทอีก 2 ตัวที่เหลือต่อเป็นภาคบัฟเฟอร์เพื่อกันโหลดไม่ให้มามีผลต่อวงจรออสซิลเลเตอร์ บัฟเฟอร์แต่ละตัวจะนำเอาสัญญาณสี่เหลี่ยม2 สัญญาณซึ่งมีเฟสต่างกันมาป้อนให้แก่แต่ละขาของตัวส่งในลักษณะของพุช-พูล ในทำนองเดียวกับวงจรรูปที่ 4เพื่อเพิ่มขนาดสัญญาณคร่อมตัวส่งให้สูงขึ้น เป็นประมาณ 2 เท่าเมื่อเทียบกับการต่อตัวส่งเข้า กับเอาท์พุทของบัฟเฟอร์เพียงตัวเดียวความถี่ของการออสซิลเลทที่เหมาะสมปรับได้โดยการปรับแต่งค่า R3 ตามวิธีการดังที่เคยกล่าวมาแล้ว วงจรนี้จะทำงานให้คลื่นเสียงอุลตร้าโซนิคออกมาต่อเมื่อโยกสวิทช์ S1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

มาที่ตำแหน่ง "เปิด"เพื่อให้ขา 1 มีลอจิกเป็น 0 โดยการต่อลงกราวนด์ อย่างไรก็ตามในภาวะ "ปิด" จะมีกระแสไหลผ่านตัวไอซีอยู่บ้างแต่ก็น้อยมากเหลือเกินจนปล่อยทิ้งไว้อย่างนั้นได้ แต่ถ้าท่านต้องการให้หยุดทำงานสนิทจริง ๆ ก็อาจตัดแปลงวงจรเล็กน้อยโดยการตัดสวิทช์ S1 ออก ต่ออินพุทขา 1 เข้ากับอินพุทขา 2 แล้วเพิ่มสวิทช์กดติดปล่อยดับเข้าระหว่างขา 14 กับไฟเลี้ยงวงจร เมื่อต้องการให้วงจรทำงานก็เพียงแต่กดสวิทช์ในทำนองเดียวกันกับสวิทช์ในวงจรรูปที่ 5 และ 6 ในกรณีที่ต้องการดัดแปลงไปใช้กับตัวส่ง 25 kHz ให้เพิ่มค่า C1 เป็น 270 pF

ถ้าท่านหาไอซีเบอร์ 4001B ไม่ได้ อาจใช้ไอซีดิจิตอลเบอร์ 4011B ซึ่งภายในประกอบด้วย NAND gate 4 ตัวแทนก็ได้ แต่สวิทช์ S1 จะต้องต่อกลับกันดังแสดงในรูปที่ 8

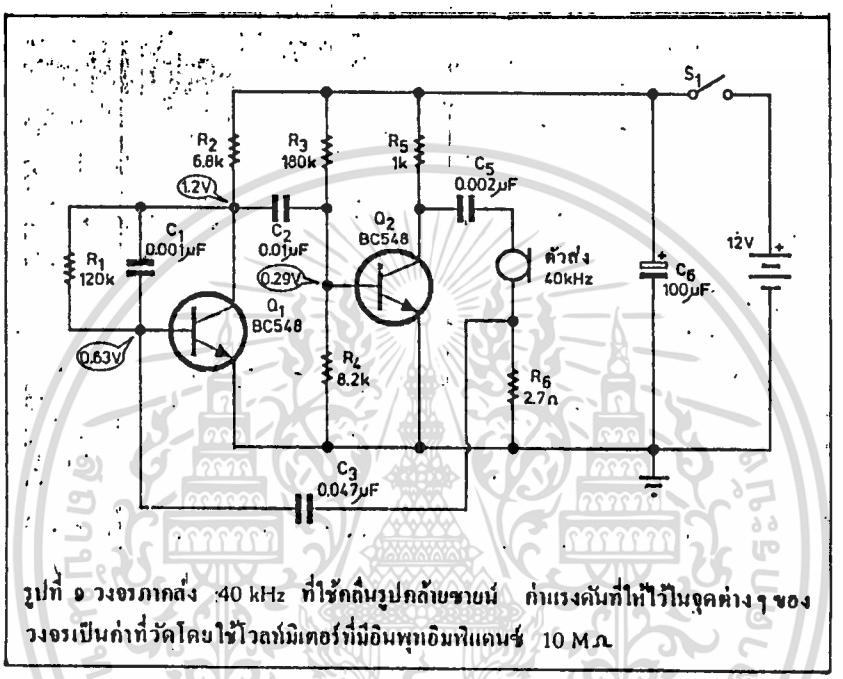


รูปที่ 8 วงจรภาคส่ง 40 kHz ที่ใช้ไอซีดิจิตอลเบอร์ 4011B

ซึ่งในที่นี้ได้เปลี่ยนสวิทช์โยกเป็นสวิทช์กดติดปล่อยดับ ดังนั้นจึงต้องเพิ่มตัวต้านทาน R4 เข้ามาเพื่อเป็นทางผ่านให้ขา 1 ลงกราวนด์ในขณะที่ไม่ได้กดสวิทช์ S1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรภาคส่งตั้งแต่รูปที่ 4 จนถึงรูปที่ 8 นั้นล้วนแล้วแต่เป็นวงจรที่จรจ้างคลื่นรูปสี่เหลี่ยมให้แก่ตัวส่งทั้งสิ้น คราวนี้ลองมาดูตัวอย่างวงจรภาคส่งที่ไม่ใช้คลื่นรูปสี่เหลี่ยมกันบ้าง ดังแสดงในรูปที่ 9



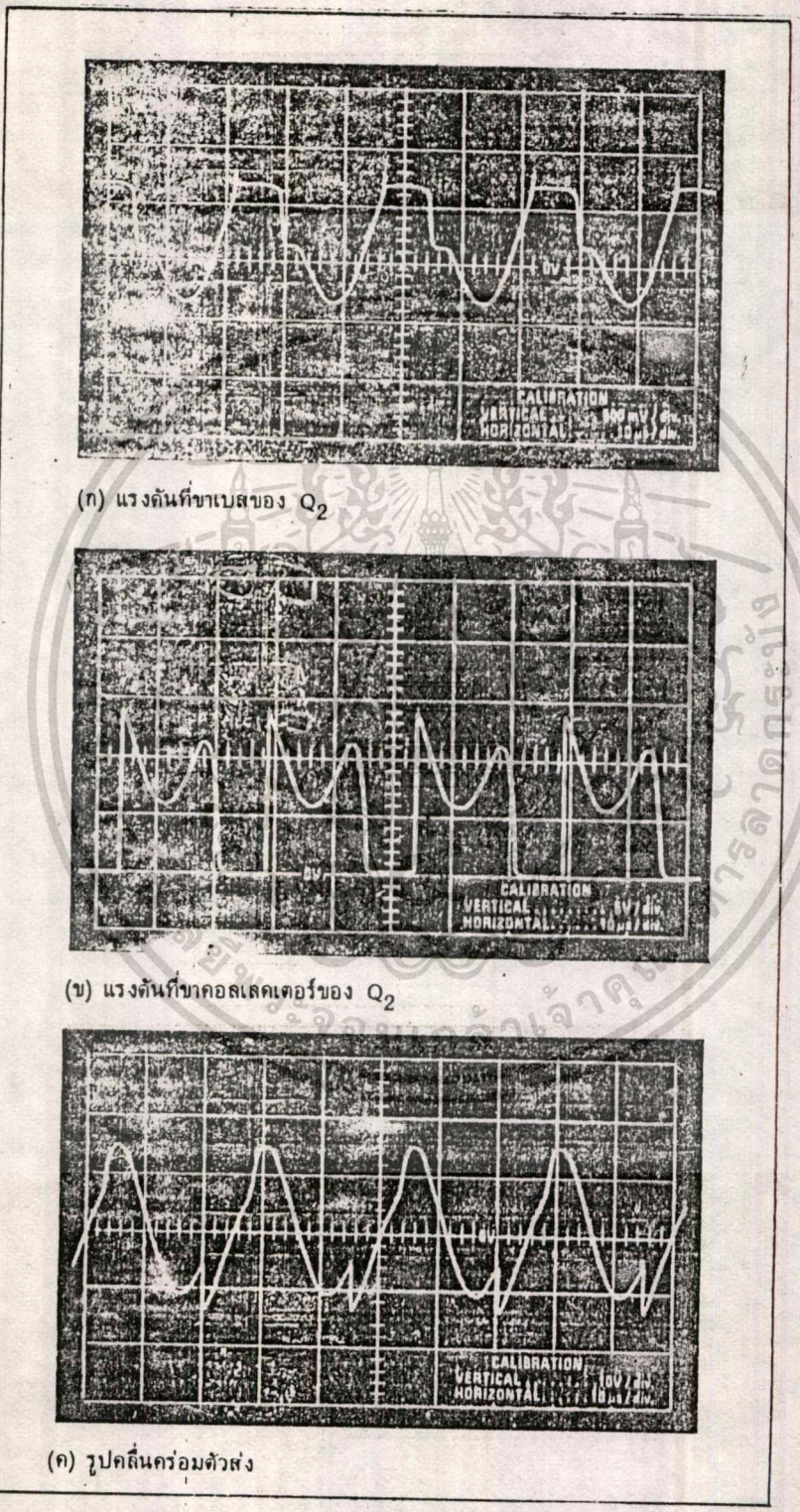
รูปที่ 9 วงจรภาคส่ง 40 kHz ที่ใช้คลื่นรูปคลื่นฉายน์ ค่าแรงดันที่ให้ไว้ในจุดต่าง ๆ ของวงจรเป็นค่าที่วัดโดยใช้โวลท์มิเตอร์ที่มีอินพุทอิมพีแดนซ์ 10 MΩ

ในวงจรนี้ใช้ทรานซิสเตอร์ 2 ตัวทำหน้าที่เป็นวงจรขยายที่ไม่กลับเฟส และใช้การป้อนกลับแบบบวกโดยผ่านตัวส่งและ C3 ซึ่งที่ความถี่เรโซแนนซ์ของทรานสดิวเซอร์จะทำให้การป้อนกลับแบบบวกมากพอที่จะทำให้เกิดการออสซิลเลทขึ้นมาได้ ตัวเก็บประจุ C1 และ C4 ใส่ไว้เพื่อป้องกันไม่ให้วงจรเกิดการออสซิลเลทที่ อาร์โมนิคที่สามของความถี่เรโซแนนซ์ ส่วน C5 ทำหน้าที่ช่วยเลื่อนจุดที่เรโซแนนซ์ให้สูงขึ้นมาอีกประมาณ500 Hz เพื่อความถี่ที่ออกมาเหมาะสมกับภาครับมากยิ่งขึ้นสำหรับวงจรนี้ใช้ได้กับไฟเลี้ยงตั้งแต่ 8 v จนถึง 20 v โดยดึงกระแสประมาณ 5 mA

ผลการทำงานของวงจรภาคส่งทั้งหมดที่ยกตัวอย่างมาให้ดูนี้ใกล้เคียงกันมาก และก็ให้ผลตรง ๆ คือ ส่งคลื่นเสียงอุลตร้าโซนิคแพร่ออกมาในอากาศได้ สำหรับวงจรภาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ส่งที่ยุ่งยากขึ้น อาจจะมีวงจรสร้างคลื่นความถี่ต่ำ เช่น 300 Hz ผสมเข้าไปกับความถี่ 40 kHz เพื่อป้องกันสัญญาณรบกวนความถี่สูงที่อาจจะเข้าไปกวนภาครับโดยให้ภาครับรับรู้คำสั่งที่เป็นคลื่นเสียงอุลตร้าโซนิคที่มีความถี่ต่ำเช่น 300 Hz ขี่มาด้วยเท่านั้น



(ก) แรงดันที่ขาเบสของ Q_2

(ข) แรงดันที่ขาคอลเลคเตอร์ของ Q_2

(ค) รูปคลื่นคร่อมตัวส่ง

รูปที่ 10 แสดงรูปคลื่นแรงดันที่จุดต่าง ๆ ของวงจรในรูปที่ 9 จะเห็นว่าสัญญาณที่เกิดขึ้นคร่อมตัวส่งจะมีรูปคลื่นคล้ายซายน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วงจรภาครับ

ตามที่ได้อธิบายมาแล้วว่า เมื่อมีคลื่นอุลตร้าโซนิคมาตกกระทบทรานสดิวเซอร์ตัวรับ จะเกิดแรงดันขนาดเล็กขึ้นมาคร่อมตัวทรานซิสเตอร์ แรงดันนี้จะมีรูปคลื่นออกมาเป็นซายน์ที่มีความถี่เดียวกับคลื่นที่มาตกกระทบตัวมัน และจะมีขนาดสูงสุดเมื่อความถี่ของคลื่นเสียงที่เข้ามาตรงกับความถี่เรโซแนนซ์ของตัวรับ ส่วนที่ความถี่อื่น ๆ ขนาดสัญญาณจะออกมาน้อยมาก บางท่านอาจสงสัยว่าทำไมสร้างคลื่นออกมาเป็นสี่เหลี่ยมความถี่ 40 kHz แต่คลื่นที่ออกมาจากตัวรับกลับเป็นรูปซายน์ ที่เป็นเช่นนี้เพราะคลื่นรูปสี่เหลี่ยมประกอบมาจากคลื่นรูปซายน์หลาย ๆ ความถี่มาผสมกัน คลื่นรูปซายน์ที่มีขนาดสูงสุดก็คือคลื่นที่มีความถี่ 40 kHz ส่วนคลื่นซายน์ความถี่อื่นๆ จะมีความถี่สูงกว่า 40 kHz เช่นอาจจะเป็น 80 kHz,120 kHz, เป็นต้น ส่วนตัวรับจะยอมให้ผ่านไปเพียงความถี่เดียวคือ 40 kHz ความถี่อื่น ๆ จะถูกตัดทิ้งไป ดังนั้นสัญญาณที่ออกมาจากตัวรับจึงเป็นคลื่นรูปซายน์ ความถี่ 40 kHz ซึ่งจริง ๆ แล้วแม้วงจรภาคส่งจะสร้างคลื่นสี่เหลี่ยมแต่คลื่นที่ออกกระจายมาในอากาศก็จะเป็นรูปซายน์ในทำนองเดียวกันกับหลักการที่อธิบายมานี้เมื่อเอาตัวรับและตัวส่งมาวางหันหน้าชนกันขนาดแรงดันคร่อมตัวรับโดยทั่วไปจะน้อยกว่า 1 โวลท์ ยิ่งระยะระหว่าง ตัวรับและตัวส่งห่างกันมากขึ้นขนาดแรงดันก็จะยิ่งลดลง อย่างเช่นที่ระยะห่างประมาณ 30 เมตร แรงดันที่เกิดขึ้นคร่อมตัวรับจะตกอยู่ในราวไม่กี่สิบไมโครโวลท์เท่านั้น ถ้ายิ่งห่างกันกว่านี้ไปอีกก็อาจจะถูกสัญญาแรบกวนกลบไปหมดดังนั้นในวงจรภาครับจึงต้องมีภาคขยาย เพื่อทำหน้าที่ขยายสัญญาณที่ออกจากตัวรับให้มีขนาดสูงพอ และก็ตามด้วยภาคเอาท์พุท ซึ่งจะทำหน้าที่แปลงสัญญาณที่ได้นี้ไปควบคุม หรือสั่งงานอุปกรณ์อื่นหรือเครื่องใช้ไฟฟ้าให้ทำงานตามต้องการอีกต่อหนึ่ง ที่ควรตระหนักเกี่ยวกับวงจรภาครับคือ วงจรขยายในภาครับมักจะมีอัตราขยายสูงมาก ไฟเลี้ยงที่จ่ายเข้ามาจึงควรใช้วงจรเรกูเลเตอร์มาช่วยทำให้แรงดันไฟเลี้ยงเรียบ และคงที่พอสมควร เพื่อให้เสถียรภาพการทำงานของวงจรดี ไม่ออสซิลเลท และไม่ทำงานผิดพลาดไปเมื่อแรงดันไฟเลี้ยงเปลี่ยนแปลงไปบ้าง

วงจรในรูปที่ 11 เป็นตัวอย่างการนำเอาทรานซิสเตอร์มาต่อกันเป็น วงจรภาครับ ซึ่งประกอบด้วยภาคขยายและภาคเอาท์พุท Q1 และ Q2 ทำหน้าที่เป็นภาคขยายชุดแรกมี Q3 และ Q4 ทำหน้าที่เป็นวงจรขยายภาคที่สอง และมี Q5 ทำหน้าที่เป็นภาคเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ขึ้นประมาณ 3,000 เท่า (70dB)R1 และ R3 ทำหน้าที่ให้ไบแอสแก่ขาเบสของ Q1 โดยมี C2 ลัดวงจรไบแอสส่วนนั้นในขณะที่มีสัญญาณอุลตร้าโซนิคเข้ามา เพื่อช่วยเพิ่มอัตราขยายให้สูงขึ้น ถ้าสัญญาณอุลตร้าโซนิคที่เข้ามาแรงมากวงจรขยายก็จะขลิบสัญญาณเอาท์พุท ให้ออกมาเป็นรูปสี่เหลี่ยมที่สวิงไปมาระหว่างไฟเลี้ยง และกราวนด์เอาท์พุทของ Q2จะถูกเรคติฟายโดย D1 และมี C3 กรองให้เป็นไฟตรงที่เรียบขึ้นแล้วมาป้อนให้แก่อินพุทขา 2 ของIC 1 IC 1 ทำหน้าที่เป็นวงจรเปรียบเทียบซึ่งจะคอยสังเกตดูระดับแรงดันที่ขา 2 เทียบกับขา 3 ที่มีขนาดแรงดันคงที่อยู่ค่าหนึ่งซึ่งกำหนดโดยการตั้งขากลางของ R7 ให้เหมาะสม ถ้าแรงดันที่ขา 2 ต่ำกว่าแรงดันที่ขา 3 เนื่องจากมีคลื่นเสียงอุลตร้าโซนิค เข้ามากระทบตัวรับ เอาท์พุทของ IC 1 จะเปลี่ยนจากแรงดัน "ต่ำ" (ประมาณ1.3v) ขึ้นไปเป็นแรงดัน "สูง" (ประมาณ10.5v)ซึ่งจะทำให้ทรานซิสเตอร์ Q3 นำกระแส และทำให้รีเลย์ทำงานไปด้วยแต่เพื่อไม่ให้หน้าสัมผัสของรีเลย์เด้งไปเด้งมาเนื่องจากขนาดสัญญาณแรงไม่เพียงพอ จึงได้เพิ่ม R10 เข้าไปช่วยให้เกิดการป้อนกลับแบบบวก คือถ้าแรงดันเอาท์พุทสูงขึ้นกว่าเดิมเนื่องจากมีสัญญาณอุลตร้าโซนิคเข้ามา การเปลี่ยนแปลงแรงดันเอาท์พุทนี้จะผ่าน R10 ไปทำให้แรงดันที่ขา 3 สูงเพิ่มขึ้นกว่าเดิมเล็กน้อย ซึ่งก็คือยิ่งทำให้แรงดันที่ขา 2 มีค่าต่ำกว่าแรงดันที่ขา 3 IC1 ก็จะเปลี่ยนขนาดแรงดันเอาท์พุทไปเป็น "สูง" ได้เร็วขึ้นแต่ถ้าต่อ C4 เข้าไปแทน R10 จะทำให้ IC1 ทำหน้าที่เป็นวงจรโมโนสเตเบิ้ลมัลติไวเบรเตอร์ไปด้วยกล่าวคือ C4 จะคอยหน่วงเวลาให้วงจรทำงานค้างอยู่ได้นานประมาณ 1 วินาที หลังจากที่คลื่นเสียงอุลตร้าโซนิคขาดหายไปแล้วเวลาใช้งานจริงให้ปรับแต่ง R7 ให้รีเลย์ไม่นำกระแสในสภาวะแวดล้อมปกติของการใช้งาน และให้ทำงานต่อเมื่อมีคลื่นเสียงอุลตร้าโซนิค จากภาคส่งที่อยู่ห่างในระยะใช้งานปกติเท่านั้นสำหรับท่านที่ชอบใช้ไอซีเป็นภาคขยายมากกว่าใช้ทรานซิสเตอร์ย่อย ขอให้ดูวงจรรูปที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ภาคแรก เพียงแต่เหลี่ยนแปลงค่าอุปกรณ์บางตัว ให้สามารถลดสัญญาณรบกวนที่ไม่ต้องการ และลดอัตราขยายที่ความถี่ต่ำลงเพื่อลดปัญหาในการทำงานลงเท่านั้น

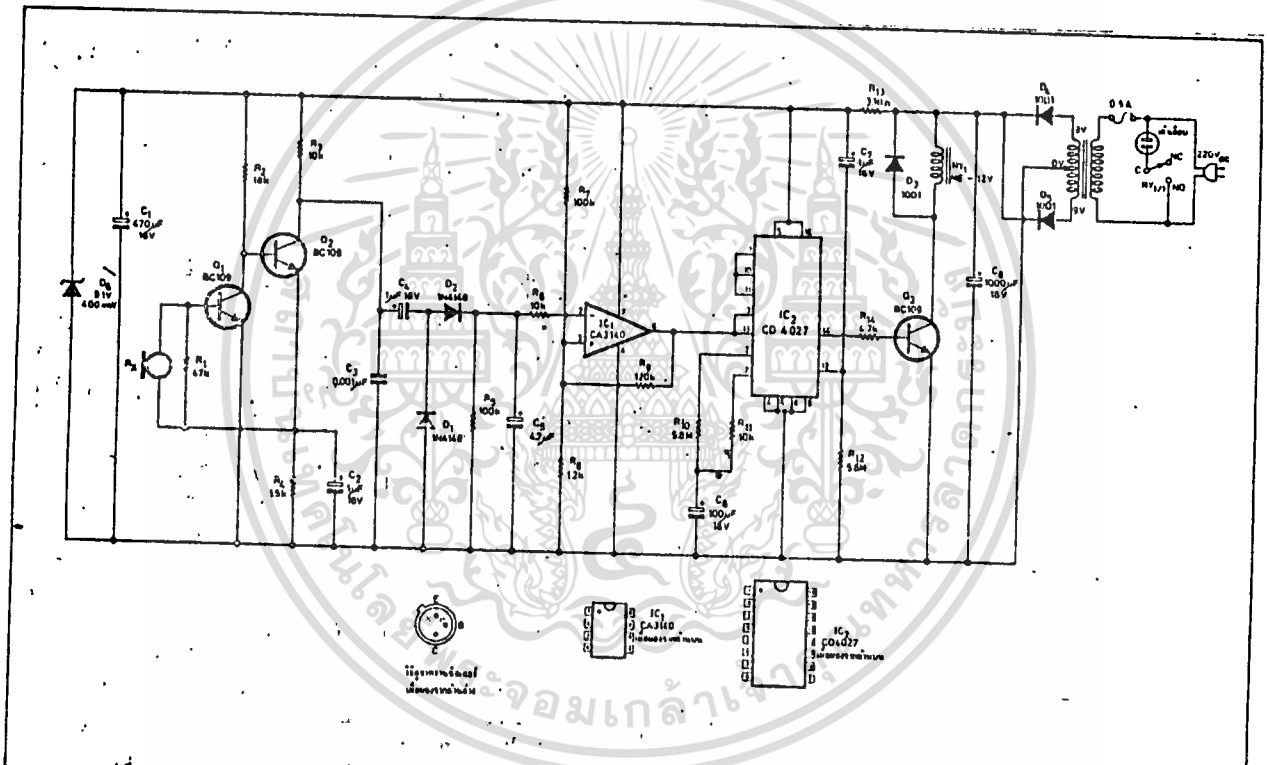
เอาท์พุทจากขา 13 จะป้อนผ่าน C7 เข้ามายังภาคไดโอดปั๊ม (diode pump circuit) ซึ่งจะทำหน้าที่แปลงรูปคลื่นอุลตร้าโซนิคให้ออกมาเป็นแรงดันไฟตรงที่ราบเรียบ คร่อม C8 C8 จะคายประจุผ่าน R9 และขาเบสของ Q1 ทำให้ Q1 และ Q2 ทำงาน มีกระแสไหลผ่านไปยังรีเลย์ ลักษณะการต่อ Q1 และ Q2 เช่นนี้ทำให้วงจรนี้สามารถจ่ายกระแสไปให้รีเลย์ได้มาก อาจจะได้ถึงมากกว่า 150 mA รีเลย์จึงไม่จำเป็นต้องเป็นชนิดไว ซึ่งรีเลย์ขนาดเหล่านี้มักจะมีหน้าสัมผัสใหญ่ พอที่จะรับกระแสของเครื่องใช้ไฟฟ้าได้สูง (เช่น10A ที่ 220 Vac) รีเลย์จะทำงานทุกครั้งที่มีคลื่นเสียงอุลตร้าโซนิค มากกระทบตัวรับ เวลาใช้งานให้ลองเปลี่ยนค่า R8 จนกว่าจะได้อัตราขยายที่พอใจ แต่ถ้าตั้งอัตราขยายสูงเกินไป (โดยปรับให้ R8 มีค่าสูง) สัญญาณรบกวนภายนอก เช่น สัญญาณกระตุ้งของโทรศัพท์แม้จะห่างไกลถึง 8 เมตร ก็อาจทำให้รีเลย์ทำงานได้

วงจรภาครับทั้งสามแบบที่กล่าวมาแล้ว ให้ภาคเอาท์พุทเหมือนกัน คือรีเลย์จะทำงานต่อเมื่อมีคลื่นอุลตร้าพุทโซนิคเข้ามาเท่านั้น แต่วงจรในรูปที่ 14 นั้นได้เพิ่มภาคโมโนสเตเบิ้ลและภาคฟลิปฟลอปเข้ามาเพื่อให้การทำงานของรีเลย์ แตกต่างไปจากเดิม กล่าวคือ เมื่อได้รับคลื่นอุลตร้าโซนิคครั้งหนึ่งรีเลย์จะทำงานค้าง ต่อเมื่อได้รับคลื่นอีกครั้งหนึ่งรีเลย์จึงจะหยุดทำงาน ตามวงจรในรูปที่ 14 นี้ Q_1 และ Q_2 จะทำการขยายสัญญาคลื่นอุลตร้าโซนิครูปซายน์ขนาดเล็กๆ ที่ออกมาจากตัวรับให้โตขึ้น แล้วนำมาเรดติฟายด้วย D_1 โดยมี C_3 และ R_6 แปลงให้สัญญาณออกมาเป็นพัลซ์ลบมาป้อนให้แก่ขา 2 ของ IC_1 อินพุทขา 2 นี้จะเปรียบเทียบยอดสูงสุดของพัลซ์นี้กับแรงดันที่ขา 3 ซึ่งถูกกำหนดโดยการแบ่งแรงดันของ R_7 และ R_8 โดยมี R_8 ทำหน้าที่เป็นตัวปรับแต่งความไวให้เหมาะสม เมื่อไม่มีสัญญาณปรากฏที่ขาคอลเลคเตอร์ของ Q_2 ขา 2 จะมีแรงดันเป็น "สูง" ทำให้แรงดันเอาท์พุทที่ขา 6 ของ IC_1 เป็นศูนย์โวลท์ แต่เมื่อมีสัญญาณพัลซ์ลบเข้ามาที่ขา 2 จะทำให้แรงดันที่ขา 2 ต่ำกว่าที่ขา 3 แรงดันเอาท์พุทจะเปลี่ยนไปเป็น "สูง" ทันที โดยมี LED_1 แสดงให้ผู้ใช้รู้ว่ามีสัญญาณอุลตร้าโซนิคเข้ามาอยู่ในขณะนั้น ส่วนตัวเก็บประจุ C_9 ทำหน้าที่ช่วยลดการทำงานผิดพลาดของออปแอมป์เนื่องจากสัญญาณรบกวน โดยสัญญาณพัลซ์จะต้องเข้ามาเป็นปริมาณหนึ่งวงจรจึงจะรับรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากที่สุด) LED$_1$ จะสว่างให้หมุน R_8 จนกระทั่ง LED$_1$ เพิ่งจะดับ นี่ก็จะเป็นตำแหน่งที่ความไวสูงสุด แต่ถ้ารู้สึกว่าจะไวเกินไปเพราะแม้แต่เสียงอุลตร้าโซนิคจากวัตถุโลหะกระทบกันค่อยๆ หรือจากเสียงดีดนิ้วหรือเสียงการทำงานของหน้าสัมผัสรีเลย์ที่ทำให้วงจรทำงานได้เองแล้ว ก็ควรลดความไวลงมาอีก เมื่อมีคลื่นอุลตร้าโซนิคเข้ามา LED$_1$ ก็ควรลดความไวลงมาอีกเมื่อมีคลื่นอุลตร้าโซนิคเข้ามา LED$_1$ ก็ควรจะสว่างให้เห็น

ถ้าท่านต้องการเพียงให้ทำงานปิด/เปิด เพียงอย่างเดียว ไม่ต้องการลักษณะการทำงานแบบชั่วคราวก็อาจดัดแปลงวงจรรูปที่ 14 ออกมาดังรูปที่ 15 วงจรในรูปที่ 15



รูปที่ 15 แสดงวงจรภาครับหลังในอุลตร้าโซนิคซึ่งให้หลักการทำงานเดียวกับรูปที่ 14 แต่ตัดลักษณะการทำงานชั่วคราวออกไป

นี้ค่อนข้างสมบูรณ์หน่อยเพราะแสดงวงจรไฟเลี้ยงมาให้ด้วย ตัวรับเสียงอุลตร้าโซนิคจะเปลี่ยนเสียงที่ได้รับมาเป็นแรงดันไฟฟ้ารูปซายน์ไปเข้าวงจรขยายที่ใช้ Q_1 Q_2 ออกมาทาง C4 ไปเข้าวงจรไดโอดปั๊ม ซึ่งประกอบด้วย D1, D2, R5 และ C5 ส่งผ่าน R6 ไปเข้าอินพุทขา 2 ของ IC1 ซึ่งเป็นออปแอมป์เบอร์ CA3140 C3 ที่ต่อเข้ามาด้วยก็

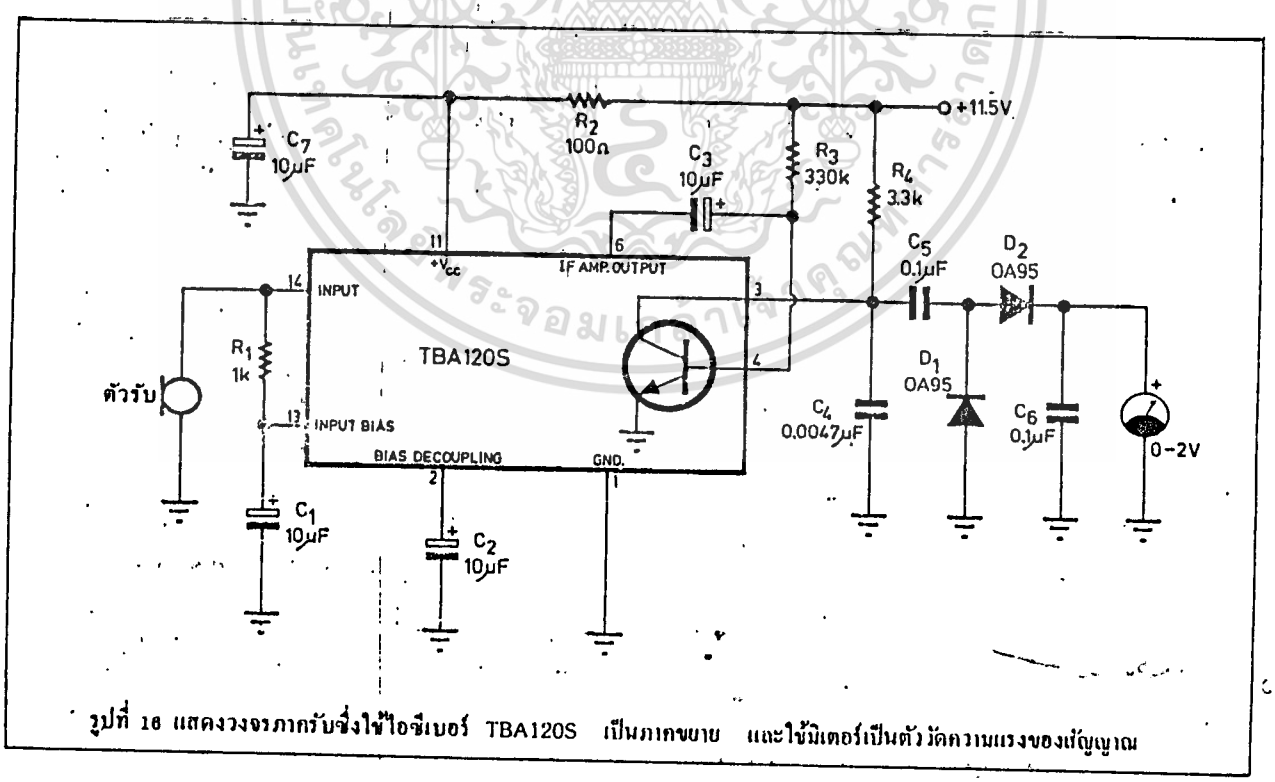
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เพื่อลัดวงจรสัญญาณความถี่สูงๆ ซึ่งอาจจะหลงเข้ามาไม่ให้ผ่านไปได้ R7 และ R8 แบ่งแรงดันเกือบเป็น 0 โวลท์ไปให้ขา 3 ของ IC1 เพื่อให้มีค่ามากกว่าแรงดันที่ขา 2 เล็กน้อย ดังนั้นขณะที่ไม่มีสัญญาณเป็นแรงดัน "สูง" เมื่อมีสัญญาณเข้ามาแรงดันที่ขา 2 จะมีค่าสูงกว่าขา 3 เอาท์พุท ก็จะเปลี่ยนไปเป็นแรงดัน "ต่ำ" เป็นสัญญาณส่งไปให้ IC2 เบอร์ CD4027 R6 ทำหน้าที่ป้อนกลับทางบวกโดยเมื่อไม่มีสัญญาณเข้ามา แรงดันสูงที่ขา 6 จะถูกแบ่งไปเล็กน้อยไปตกคร่อม R8 ซึ่ง R8 ก็มีแรงดันที่ได้มาจากการแบ่งไฟเลี้ยงอยู่แล้ว พอมีสัญญาณเข้ามาและแรงดันที่เอาท์พุทตกเป็น 0 โวลท์ แรงดันคร่อม R8 ส่วนที่แบ่งมาจากเอาท์พุทก็หายไปแรงดันที่ขา 3 จึงมีค่าลดลงทำให้แรงดันที่ขา 2 มีค่ามากขึ้น เอาท์พุทจึงเป็น 0 โวลท์อย่างรวดเร็ว การป้อนกลับนี้จะมีประโยชน์ในกรณีที่สัญญาณเข้ามาอ่อนมากพอๆ กับแรงดันที่ขา 3 ถ้าไม่ลดแรงดันของขา 3 ลงไปเอาท์พุทจะสั่นขึ้นลงตลอดเวลา

ภายใน IC2 เป็น IC2 เป็น JK ฟลิปฟลอป 2 ตัว ฟลิปฟลอปตัวแรกนั้นทำหน้าที่ส่งผ่านสัญญาณไปให้ฟลิปฟลอป 2 ฟลิปฟลอปตัวที่ 2 ก็จะให้สัญญาณแรงดันสูงไปขับ Q3 ให้รีเลย์ทำงานในจังหวะที่สัญญาณอุลตร้าโซนิค ที่เข้ามาหยุดหายไป และถ้าสัญญาณอุลตร้าโซนิคเข้ามาแล้วหายไปอีกครั้งหนึ่ง IC2 ก็จะควบคุมให้รีเลย์หยุดทำงาน ฟลิปฟลอปตัวแรกนั้นไม่ได้มีไว้เฉยๆ ขา 2 ซึ่งเป็นเอาท์พุทอันหนึ่งจะประจุ C6 โดยผ่าน R10 ในขณะที่มีสัญญาณเข้ามา เมื่อสัญญาณหยุดไปแล้ว ขา 7 ยังจะได้รับสัญญาณแรงดันสูงจาก C6 อยู่ สัญญาณนี้จะควบคุมไม่ให้รับสัญญาณอีกพักหนึ่งจนกว่า C6 จะคายประจุหมด ที่ต้องทำดังนี้ก็เพราะเสียงการทำงานของรีเลย์อาจจะไปรบกวนตัวรับ ทำให้วงจรได้รับสัญญาณเข้ามาทั้งๆ ที่เราไม่ได้ส่งให้ และรีเลย์ก็จะติสลับไปสลับมาอยู่อย่างนั้น จึงต้องหยุดการทำงานให้พ้นจากช่วงการทำงานของรีเลย์ R12 และ C7 จะทำให้ขา 12 ได้รับสัญญาแรงดันสูงตอนเริ่มต้นจ่ายไฟเข้าเครื่องเพื่อไปควบคุม IC2 ให้ควบคุมให้รีเลย์ทำงานในทันทีอีกต่อหนึ่งจากนั้นเวลาเราส่งสัญญาณอุลตร้าโซนิคเข้าไปวงจรก็จะทำงานตามปกติ ทางด้านวงจรจ่ายไฟ D4, D5 และ C8 จะแปลงไฟสลับให้เป็นไปตรง 12 โวลท์มาจ่ายให้รีเลย์และมี R13, D6 ช่วยลดแรงดันลงเหลือ 9.1 โวลท์จ่ายไฟที่คงที่ให้แก่วงจรส่วนที่เหลือ เพื่อไม่ให้สัญญาณรบกวนจากแหล่งจ่ายไฟไปรบกวนการทำงานแล้วยังมี C1 ช่วยกรองไฟให้เรียบขึ้นอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

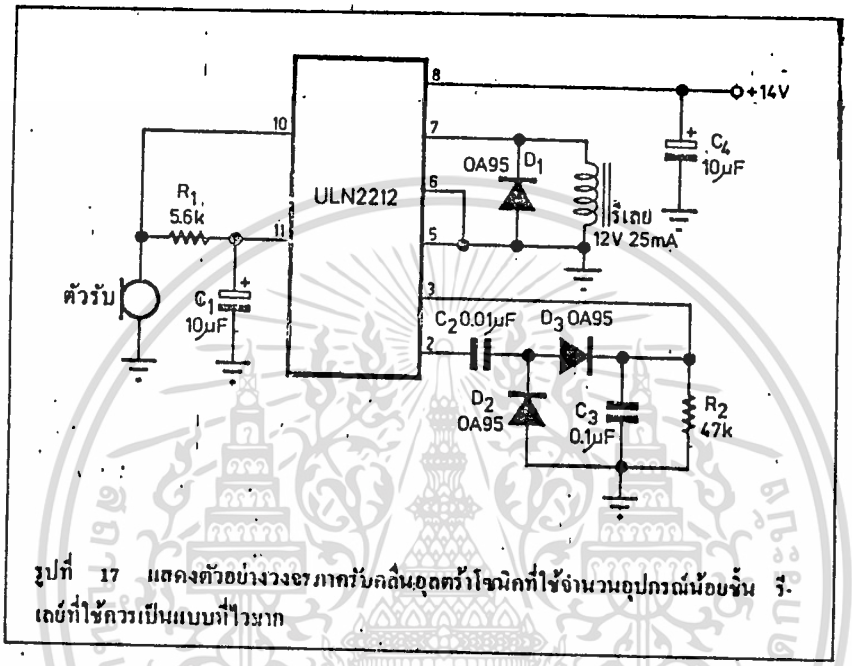
ถ้าใครมีไอซีที่ถูกออกแบบมาเจาะ จงให้ใช้ขยายสัญญาณ IF ในเครื่องรับวิทย FM และเครื่องรับโทรทัศน์ละก็อาจนำมาดัดแปลงเป็นภาคขยายสัญญาณอุลตร้าโซนิคได้เป็นอย่างดี ดังตัวอย่างวงจรในรูป 16 ซึ่งแสดงตัวอย่างการนำเอาไอซีเบอร์ TBA1205 ซึ่งทำหน้าที่เป็นภาคขยาย IF และดีเทคเตอร์มาสร้างเป็นภาคขยาย ไอซีเบอร์นี้มีผู้ผลิตกันหลายบริษัท อาทิเช่น ฟิลิปส์, NS สัญญาณเอาท์พุทอาจนำออกจากขา 6 (หรือขา10 ก็ได้) ผ่านตัวเก็บประจุ 10 uF มาให้แก่ขาเบสของทรานซิสเตอร์ซึ่งอยู่ภายในตัวไอซี เพื่อให้ช่วยขยายสัญญาณให้โตขึ้น R4 ทำหน้าที่เป็นโหลดให้แก่ตัวทรานซิสเตอร์ และมี C4ช่วยป้องกันวงจรออสซิลเลทเนื่องจากสัญญาณรบกวน จากนั้นสัญญาณจึงถูกส่งผ่านมาให้วงจรไดโอดปั๊ม เพื่อทำให้เป็นแรงดันไฟตรงมาป้อนให้แก่มิเตอร์ 0-2 Vdcแต่ถ้าไม่ต้องการใช้มิเตอร์ก็อาจต่อทรานซิสเตอร์ 2 ตัวเข้ามาขับรีเลย์ในทำนองเดียวกับวงจรที่กล่าวมาในตอนต้นๆ ก็ได้ มีข้อน่าสังเกตอยู่อย่างหนึ่งว่า เขาใช้ไดโอดชนิดเยอรมันเนียมในวงจรไดโอดปั๊มแทนที่จะใช้ซิลิกอนทั้งนี้เพราะแรงดันตกคร่อมตัวมันน้อยกว่า จึงได้ขนาดสัญญาณไปถึงตัวมิเตอร์สูงขึ้น



รูปที่ 16 แสดงวงจรภาครับซึ่งใช้ไอซีเบอร์ TBA120S เป็นภาคขยาย และใช้มิเตอร์เป็นตัววัดความแรงของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 17 เป็นตัวอย่างวงจรภาครับที่ง่ายมากอีกวงจรหนึ่ง ใช้ไอซีเบอร์ ULN-2212 ซึ่งเป็นของยี่ห้อ Sprague ทำหน้าที่เป็นภาคขยาย IF ของสัญญาณเสียงใน



รูปที่ 17 แสดงตัวอย่างวงจรภาครับคลื่นอุลตร้าโซนิคที่ใช้จำนวนอุปกรณ์น้อยชิ้น รีเลย์ที่ใช้ควรเป็นแบบที่ไวมาก

เครื่องรับโทรทัศน์ และมีภาคขยายเสียงอยู่ในตัวด้วยบางท่านอาจนำเอาเบอร์ LM1808 ของยี่ห้อ NS ซึ่งมีหน้าที่คล้ายๆ กันมาดัดแปลงใช้ในทำนองเดียวกันก็ได้ใน วงจรนี้ภาคขยาย IF ภายในตัวไอซีถูกนำมาใช้ขยายสัญญาณอุลตร้าโซนิคจากตัวรับ โดยมี R1 คอยให้ไบแอสแก่อินพุทที่ขา 10 สัญญาณเอาท์พุทจากขา 2 จะมาต่อเข้าวงจรไดโอดปั๊ม ทำให้เป็นแรงดันไฟตรง แล้วป้อนเข้าภาคเพาเวอร์แอมป์ทางขา 3 เพื่อให้ช่วยขยายกระแสให้มากพอที่จะไปสั่งงานรีเลย์ซึ่งต่ออยู่ที่เขาเอาท์พุท (ขา 7) แม้ว่าวงจรในรูปที่ 17 นี้จะง่ายมาก แต่ความไวก็น้อยกว่าวงจรก่อน ๆ

วงจรภาครับทั้งหมดที่กล่าวมาในหัวข้อนี้ สามารถขยายสัญญาณอุลตร้าโซนิคได้ดีตั้งแต่ช่วงความถี่ระหว่าง 20 -60kHz ดังนั้นจึงสามารถนำมาใช้ร่วมกับทรานสดิวเซอร์ ตัวรับที่มีความถี่เรโซแนนซ์ อยู่ในช่วงความถี่เหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การใช้ ROM กับ Z-80

ในบทนี้ เราจะมาศึกษากันว่าจะใช้งานไมโครโปรเซสเซอร์ Z80 ร่วมกับหน่วยความจำชนิด ROM ได้อย่างไร โดยจะกล่าวถึงชนิดของ ROM และการใช้งานก่อน ต่อจากนั้นจะกล่าวถึงการนำ ROM มาต่อร่วมกับไมโครโปรเซสเซอร์ Z80 เมื่อจบบทนี้ผู้อ่านจะทราบว่า Z80 สามารถทำงานร่วมกับ ROM ได้อย่างไร และสามารถนำเอาความรู้นี้ไปดัดแปลงใช้งานอื่นๆ ได้อีกมาก.

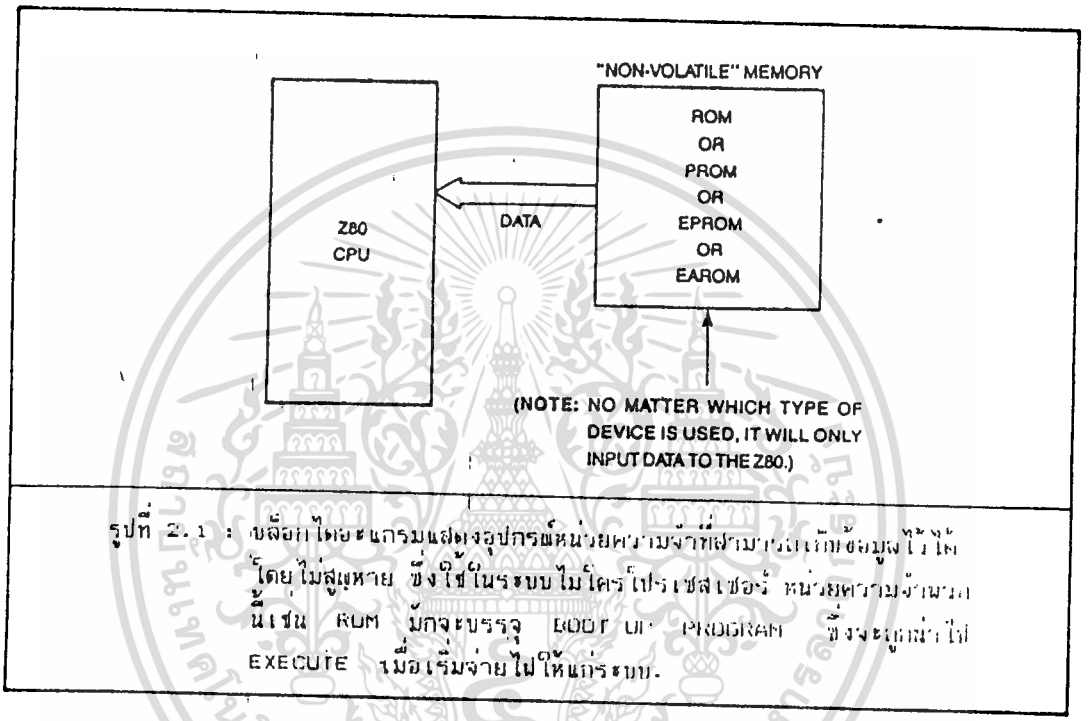
2.1 ROM คืออะไร?.

ในระบบไมโครโปรเซสเซอร์ จะมีหน่วยความจำชนิดหนึ่งที่ใช้สำหรับเก็บข้อมูลโดยที่ข้อมูลเหล่านี้ไม่สูญหายเมื่อไม่มีการจ่ายไฟเลี้ยงแก่ระบบ หน่วยความจำชนิดนี้ก็คือ ROM (READ ONLY MEMORY) นั่นเอง ข้อมูลที่เก็บอยู่ใน ROM สามารถอ่านออกมาได้ แต่ไม่สามารถเขียนข้อมูลเข้าไปใน ROM ได้อีก ประโยชน์ของ ROM ในระบบไมโครโปรเซสเซอร์คือ ทำให้ CPU ของระบบสามารถที่จะเริ่มทำงานโดยการกำหนดสภาวะของอุปกรณ์ฮาร์ดแวร์ต่างๆ (เป็นที่เก็บ MONITOR PROGRAM) ให้อยู่ในสภาวะที่พร้อมจะทำงานได้ เมื่อเราเริ่มจ่ายไฟเลี้ยงให้แก่ระบบ.

หน่วยความจำที่เก็บข้อมูลไว้ได้โดยไม่สูญหาย ที่ใช้ในระบบไมโครโปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซสเซอร์มีหลายชนิด เช่น READ-ONLY MEMORY (ROM) , PROGRAMMABLE READ-ONLY MEMORY (PROM) , ERASABLE PROGRAMMABLE READ-ONLY MEMORY (EPROM), ELECTRICALLY ALTERABLE READ-ONLY MEMORY (EAROM) (รูปที่ 2.1) ในระบบไมโครโปรเซสเซอร์ Z80 โดยทั่วๆไป จะใช้หน่วยความจำเหล่านี้ชนิดใดชนิดหนึ่งเท่านั้น ขึ้นอยู่กับความสะดวกและราคาในการใช้งานของระบบนั้นๆ ซึ่งจะกล่าวแยกแต่ละชนิดต่อไป.



รูปที่ 2.1 : บล็อกไดอะแกรมแสดงอุปกรณ์หน่วยความจำที่สามารถเก็บข้อมูลไว้ได้โดยไม่สูญหาย ซึ่งใช้ในระบบไมโครโปรเซสเซอร์ หน่วยความจำพวกนี้เช่น ROM มักจะบรรจุ BOOT UP PROGRAM ซึ่งจะถูกนำไป EXECUTE เมื่อเริ่มจ่ายไฟให้แก่ระบบ.

ROM (READ-ONLY MEMORY) : ข้อมูลทั้งหมดที่อยู่ใน ROM จะถูกโปรแกรมโดยผู้ผลิต (โปรแกรมมาจากโรงงาน) เราจะใช้ ROM เมื่อมีข้อมูลที่ไม่เปลี่ยนแปลงและมีความต้องการใช้งานเป็นจำนวนมาก.

PROM (PROGRAMMABLE READ-ONLY MEMORY) : ข้อมูลที่ต้องการโปรแกรมจะถูกโปรแกรมโดยผู้ใช้เอง โดยป้อนพัลส์ที่มีแรงดันสูง(HIGH VOLTAGE PULSED) ทำให้ METAL STRIPS หรือ POLYCRYSTALINE SILICON ที่อยู่ในตัว IC ขาดออกจากกัน ทำให้เกิดเป็นลอจิก "1" หรือ "0" ตามตำแหน่งที่กำหนดในหน่วยความจำนั้นๆ เมื่อ PROM ถูกโปรแกรมแล้วข้อมูลภายในจะไม่สามารถเปลี่ยนแปลงได้อีกหน่วยความจำชนิดนี้จะใช้เมื่อต้องการความเร็วในการทำงานสูง ซึ่งความเร็วจะสูงกว่าหน่วยความจำชนิดโปรแกรมได้ชนิดอื่นๆที่จะกล่าวต่อๆไป.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

EPROM (ERASABLE PROGRAMMABLE ROM) : ข้อมูลจะถูกโปรแกรมโดยผู้ใช้โดยการให้สัญญาณที่มีแรงดันสูง(HIGH VOLTAGE SIGNAL) ผ่านเข้าไปในตัว EPROM ซึ่งเป็นวิธีเดียวกับที่ใช้ใน PROM แต่ข้อมูลที่อยู่ใน EPROM เปลี่ยนแปลงได้ โดยการลบข้อมูลเดิมที่อยู่ใน EPROM ออกก่อนแล้วค่อยโปรแกรมเข้าไปใหม่ การลบข้อมูลนั้นทำได้ด้วยการฉายแสงอุลตร้าไวโอเลตเข้าไปในตัว IC โดยผ่านทางกระจกใสที่อยู่บนตัว IC เมื่อฉายแสงครู่หนึ่งข้อมูลที่อยู่ภายในก็จะถูกลบทิ้ง ซึ่งช่วงเวลาที่ฉายแสงนี้สามารถดูได้จากข้อมูลที่กำหนด (DATA SHEET) มากับตัว EPROM และ มีความเหมาะสมที่จะใช้เมื่องานของระบบมีโอกาสที่จะปรับปรุงแก้ไขข้อมูลใหม่.

EAROM (ELECTRICALLY ALTERABLE ROM) : ข้อมูลจะถูกโปรแกรมโดยผู้ใช้เหมือนใน EPROM แต่สิ่งที่แตกต่างก็คือข้อมูลของ EAROM สามารถลบได้โดยทางไฟฟ้าไม่ใช่โดยการฉายแสงแบบ EPROM.

หน่วยความจำที่ได้กล่าวมาแล้วนั้นเป็นหน่วยความจำที่ข้อมูลไม่สูญหายเมื่อไม่มีไฟเลี้ยง และสามารถอ่านข้อมูลมาใช้ได้เพียงอย่างเดียวเท่านั้น ซึ่งคุณสมบัติการใช้งานจะเหมือนกันขึ้นอยู่กับระบบว่าเหมาะสมจะใช้กับชนิดไหน ในหัวข้อต่อจากนี้ไปจะขอแทนหน่วยความจำเหล่านี้ด้วย ROM.

2.2 คุณสมบัติที่สำคัญในการทำงานของ ROM.

จากที่เคยทราบมาแล้วว่า CPU สามารถอ่านข้อมูลใน ROM ได้เพียงอย่างเดียว และข้อมูลที่อยู่ใน ROM จะถูกเฟทช์ (FETCH) โดย CPU เมื่อมีสัญญาณบ่งตำแหน่งที่ได้จากบัสแอดเดรส(ADDRESS BUS)ส่งเข้ามา จำนวนของบัสแอดเดรสนั้นจะขึ้นอยู่กับลักษณะโครงสร้างภายในของข้อมูลในหน่วยความจำ.

สิ่งที่จะบอกลักษณะโครงสร้างภายในของข้อมูลในหน่วยความจำ สามารถดูได้จาก DATA SHEET สำหรับ ROM ตัวนั้น เช่น ROM ที่บอกมาเป็น 1024x8 , 2048x8 , หรือ 4096x8 เหล่านี้เป็นตัวอย่างของการบอกโครงสร้างของหน่วยความจำ ตัวเลขชุดแรก (1024 , 2048 หรือ 4096) จะเป็นตัวบ่งบอกให้ทราบถึงจำนวนตำแหน่งที่จะใช้เก็บข้อมูลภายใน ROM ตัวนี้ส่วนตัวเลขชุดที่สอง (8) เป็นตัวที่บอกให้ทราบว่าจำนวนบิทของข้อมูลแบบขนานที่จะอ่านจาก ROM ว่ามีจำนวนเท่าไร จากตัวอย่างนี้เป็นแบบ 8 บิท.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

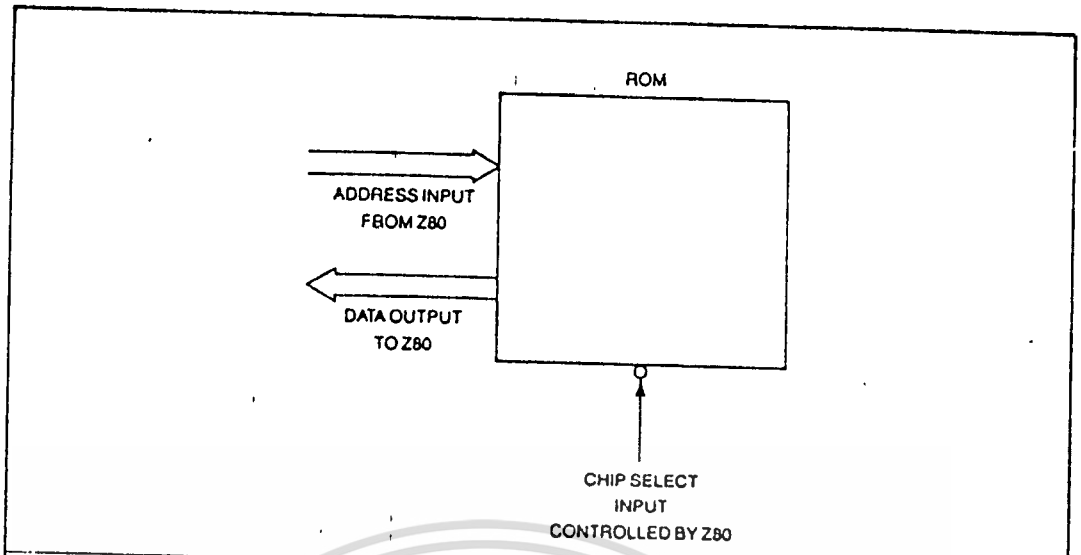
ในการกำหนดจำนวนเส้นของบัสแอดเดรสที่ใช้กับ ROM เราจะดูจากสิ่งที่บอกมากับตัวมัน เช่น 2048x8 เราได้รู้มาแล้วว่า ROM ตัวนี้จะมีตำแหน่งที่ใช้เก็บข้อมูลขนาด 8 บิท ได้จำนวน 2048 ตำแหน่ง ถ้าให้เลข 2048 นี้เท่ากับ 2^x ในที่นี้จะได้ X=11 ซึ่งก็คือจำนวนเส้นของบัสแอดเดรส ที่จำเป็นต้องใช้กับ ROM ตัวนี้ ฉะนั้นถ้ามีหน่วยความจำที่บอกมาเป็น 4096x8 ก็สามารถหาจำนวนเส้นของบัสแอดเดรสได้โดยให้ $2^x = 4096$ จะได้ X=12 จะเห็นว่าสมการที่กล่าวมาแล้วนั้นทำให้สามารถหาจำนวนเส้นของบัสแอดเดรสที่ต้องการใช้ใน ROM ตัวนั้นได้สะดวกขึ้น.

ในการใช้ ROM จะพบว่าจำนวนตำแหน่งในหน่วยความจำมักจะบอกมาเป็น K BYTES(กิโลไบท์) ซึ่ง 1K จะมีค่าเท่ากับ $2^{10} = 1024$ ฉะนั้น ROM ที่มีขนาด 1024x8 ก็คือ ROM ที่มีความจุ 1K BYTES (ในหนึ่งตำแหน่งของหน่วยความจำเก็บได้ 8 BITS ซึ่ง 8 BITS มีค่าเท่ากับ 1 BYTE) และ ROM ที่มีขนาด 4096x8 คือ ROM ที่มีความจุ 4K เป็นต้น.

สิ่งที่สำคัญต่อมาก็คือจะต้องทำความเข้าใจลักษณะการทำงานของ ROM ว่าข้อมูลจะถูกส่งออกจาก ROM เป็นแบบขนาน (IN A PARALLEL FASHION) หรือในอีกแง่หนึ่งก็คือ ขณะที่ข้อมูลทั้ง 8 บิท ถูกส่งออกมาจาก ROM เข้าสู่บัสข้อมูลของระบบ ในเวลาเดียวกันนั้น ไมโครโปรเซสเซอร์จะนำข้อมูลทั้ง 8 บิท นี้เข้ามาเก็บไว้ในรีจิสเตอร์ในเวลาเดียวกัน.

จากรูป 2.2 เป็นการแสดงบล็อกไดอะแกรม การทำงานของ ROMในรูปนี้จะเห็นว่ามีเส้นอินพุทเส้นหนึ่งที่เรียกว่า CHIP SELECT INPUT ซึ่งเรายังไม่เคยกล่าวถึงมาก่อน ฟังก์ชันการทำงานของเส้นอินพุทนี้จะเป็นตัวปิดหรือเปิด DATA OUTPUT LINE ของ ROM เมื่อ CHIP SELECT INPUT LINE ถูกทำให้ทำงาน DATA OUTPUT LINE ก็จะส่งข้อมูลออกจาก ROM ได้ โดยสถานะทางลอจิกของ CHIP SELECT LINE จะเป็นลอจิก "1" หรือ "0" นั้นขึ้นอยู่กับว่า ROM ตัวนี้ได้ถูกออกแบบมาให้ทำงานที่ลอจิกใด เมื่อ CHIP SELECT LINE ไม่ถูก ENABLE, DATA OUTPUT LINE จะอยู่ในสถานะ TRI-STATE (HIGH IMPEDANCE) กล่าวคือ CHIP SELECT LINE จะ ENABLE หรือ DISABLE เอาท์พุทของ ROM, ซึ่งในท้ายบทนี้เราจะมาดูกันว่าจะใช้งานขานี้ได้อย่างไร.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 2.2 : บล็อกไดอะแกรมแสดงการใช้งานอุปกรณ์จำพวก ROM ค่าแอดเดรส จะถูกป้อนเข้าไป เพื่อใช้ในการกำหนดตำแหน่งข้อมูลในหน่วยความจำ ซึ่งข้อมูลนี้จะถูกอ่านออกมา เมื่ออุปกรณ์เหล่านี้ตัวใดตัวหนึ่งถูกเลือกใช้งาน โดยสัญญาณ CHIP SELECT.

2.3 ขั้นตอนในการอ่านข้อมูลจาก ROM.

ลำดับขั้นตอนที่เกิดขึ้นในการอ่านข้อมูลแต่ละครั้งออกจาก ROM ดังแสดงไว้ข้างล่างนี้ เป็นลำดับขั้นตอนการทำงานทั่วๆ ไป โดยไม่คำนึงถึงชนิดของ CPU ที่ใช้ (ซึ่งจะกล่าวถึง Z80 ว่าจะทำตามขั้นตอนเหล่านี้อย่างไรที่หลัง)

1. ค่าแอดเดรสจะถูกป้อนเข้าไปยัง ROM (โดย CPU) ค่าแอดเดรสนี้จะกำหนดตำแหน่งของข้อมูลที่ต้องการอ่าน โดยข้อมูลจะถูกอ่านออกมาเพียงครั้งละ 1 ไบท์เท่านั้น.

2. CPU จะคอยอยู่ช่วงเวลาหนึ่ง(WAIT STATE) เรียกว่า ACCESS TIME ประมาณ 100-300 nanoseconds ขึ้นอยู่กับชนิดของ ROM ซึ่งเป็นเวลาที่ ROM ใช้ในการถอดรหัสของแอดเดรส (DECODE ADDRESS) ของข้อมูลที่ต้องการจะอ่านออกมาที่เอาท์พุทของ ROM.

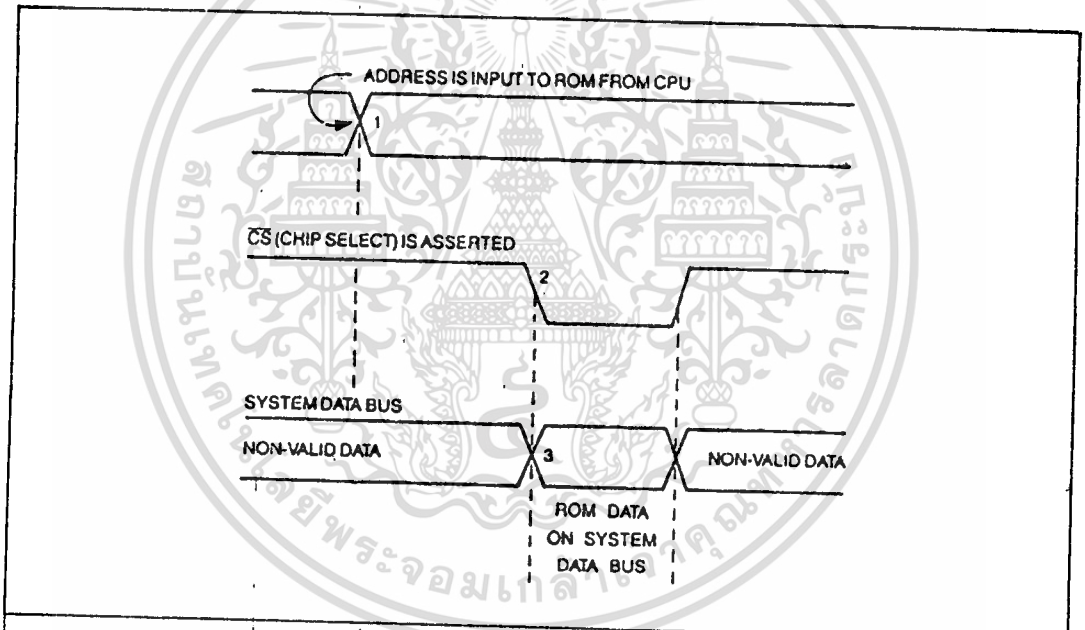
3. CHIP SELECT LINE จะถูกทำให้แอคทีฟ (ACTIVE) เพื่อให้ข้อมูลออกมาที่บัสข้อมูลของระบบได้ ต่อจากนั้น CPU จะสโตรป(STROBE)ข้อมูลเข้าไปเก็บไว้ในรีจิสเตอร์ภายใน CPU.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. CHIP SELECT LINE จะถูกสั่งให้เลิกทำงาน(สถานะ INACTIVE) เพื่อทำให้ข้อมูลที่อยู่บนบัสข้อมูลของระบบหายไป.

ลำดับการทำงานทั่วๆไปที่กล่าวมานี้ จะเกิดขึ้นทุกครั้งที่ CPU อ่านข้อมูลจาก ROM ในรูป 2.3 แสดงไดอะแกรมเวลา(TIMING DIAGRAM)ทั่วๆไปของลำดับการทำงานนี้.

CPU ถูกออกแบบให้สามารถทำตามขั้นตอนดังที่กล่าวมาแล้วข้างต้น ผู้ใช้ส่วนมากมักจะไม่คำนึงถึงขั้นตอนการทำงานเหล่านี้ เพราะในการใช้ไมโครโปรเซสเซอร์จะมีวงจรเวลาภายในตัว (INTERNAL TIMING) ช่วยอยู่แล้ว อย่างไรก็ตามความเข้าใจเกี่ยวกับลำดับการทำงานนี้ก็เป็นสิ่งสำคัญ เพราะจะได้รู้ถึงวิธีการนำไปประยุกต์ใช้งานได้อย่างมีประสิทธิภาพ.



รูปที่ 2.3 : แสดงไดอะแกรมเวลาของสัญญาณที่ใช้ในการอ่านข้อมูลจากหน่วยความจำประเภท ROM

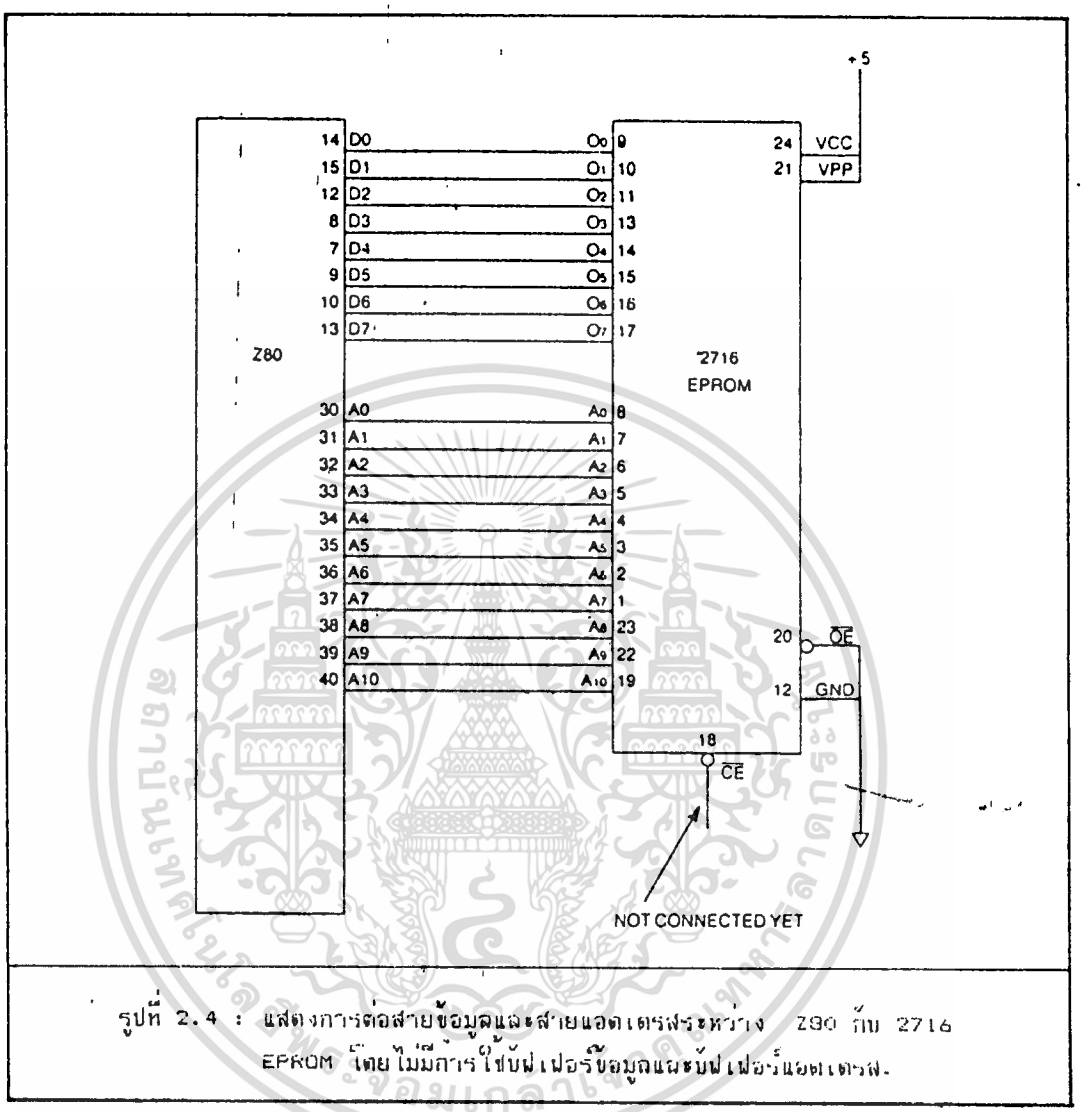
1. CPU ส่งค่าแอดเดรสให้กับ ROM
2. สัญญาณ CHIP SELECT ถูกส่งให้กับ ROM
3. ข้อมูลจาก ROM ถูกส่งออกมาบนบัสข้อมูล

2.4 การต่อ ROM เข้ากับ Z80 บัส.

ต่อไปนี้เราจะทำการต่อ ROM เข้ากับ Z80 ดังรูป 2.4 ซึ่งแสดงถึงการต่อบัสแอดเดรสและบัสข้อมูลของ Z80 เข้ากับ ROM โดยตรง จะสังเกตได้ว่าเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังไม่ได้ต่อ CHIP SELECT LINE ของ EPROM เบอร์ 2716 เข้ากับระบบ ทั้งนี้ เพราะเราจะต้องพิจารณาผังแอดเดรส (ADDRESS MAPPING) เสียก่อน.



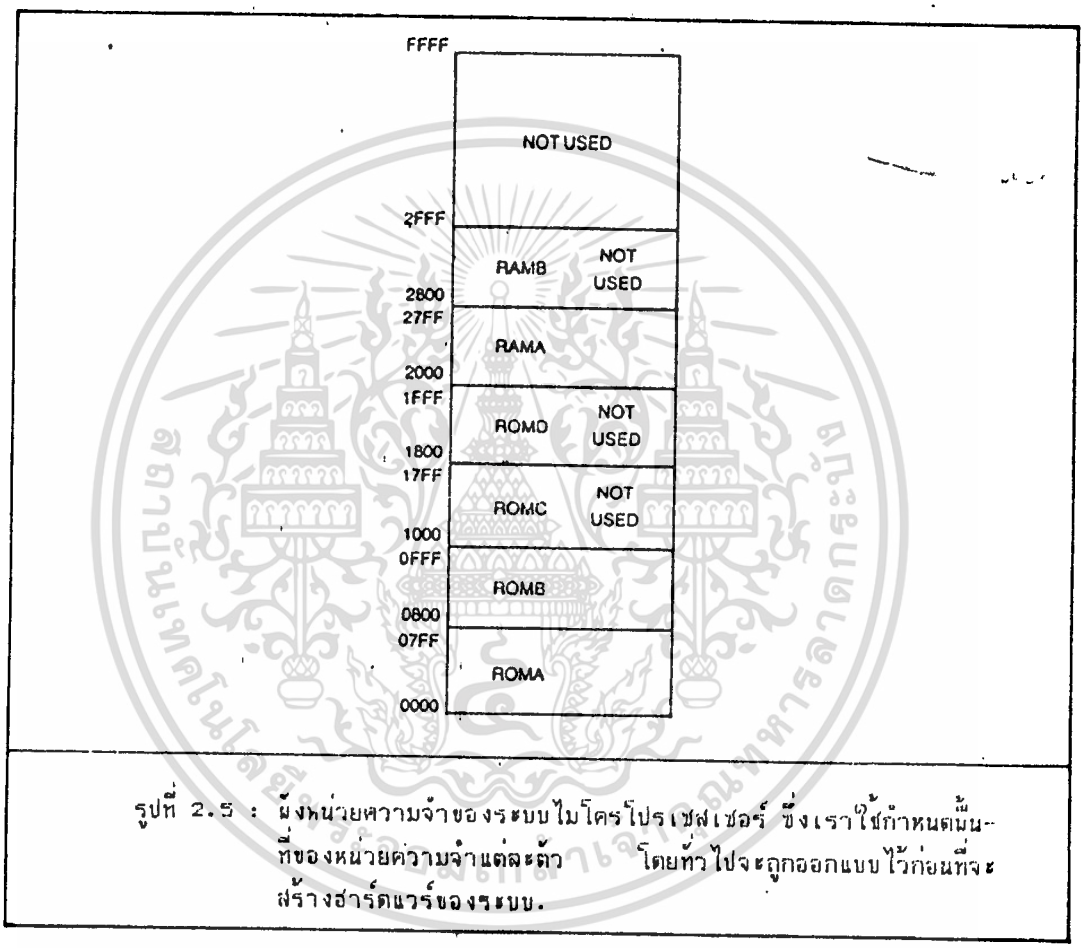
รูปที่ 2.4 : แสดงการต่อสายข้อมูลและสายแอดเดรสระหว่าง Z80 กับ 2716 EPROM โดยไม่มีการใช้บัฟเฟอร์ข้อมูลและบัฟเฟอร์แอดเดรส.

2.5 ผังแอดเดรส (ADDRESS MAPPING).

Z80 เป็นไมโครโปรเซสเซอร์ที่มีบัสแอดเดรส 16 เส้น (A0-A15) นั่นหมายถึง Z80 สามารถติดต่อกับแอดเดรสต่างๆ โดยตรงได้ถึง 2^{16} หรือเท่ากับ 65,536 ตำแหน่ง สำหรับในตัวอย่างนี้เราใช้ EPROM เบอร์ 2716 ซึ่งมีจำนวนตำแหน่งเพียง 2048 ตำแหน่ง ดังนั้นจึงจำเป็นที่จะต้องมีวงจรที่ใช้สำหรับเลือกตำแหน่งเหล่านี้จากทั้งหมด 65,536 หรือ 64K ตำแหน่ง หรือกล่าวได้ว่าถ้าเราแบ่ง 65,536 ออกเป็นกลุ่ม กลุ่มละ 2048 ตำแหน่งได้ 32 กลุ่ม และเลือกกลุ่มใดกลุ่มหนึ่งจาก 32 กลุ่มนี้.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะทำการสร้างระบบทางฮาร์ดแวร์ผู้ออกแบบจำเป็นต้องกำหนดโครงสร้างของผังหน่วยความจำ (MEMORY MAP) เสียก่อน เนื่องจากผังหน่วยความจำเป็นสิ่งที่บอกให้ทราบว่าตำแหน่งของ ROM , RAM และอุปกรณ์ I/O อยู่ที่ไหน ในรูป 2.5 ได้แสดงถึงผังหน่วยความจำ จะเห็นว่าในรูปนี้เราแบ่งตำแหน่งใน 65,536 ตำแหน่งออกเป็นกลุ่มตามหน้าที่ของการทำงานและในกลุ่มของ ROM และ RAM ที่แสดงไว้นี้ ได้เผื่อไว้สำหรับหน้าที่เฉพาะที่อาจจำเป็นต้องใช้ด้วย.



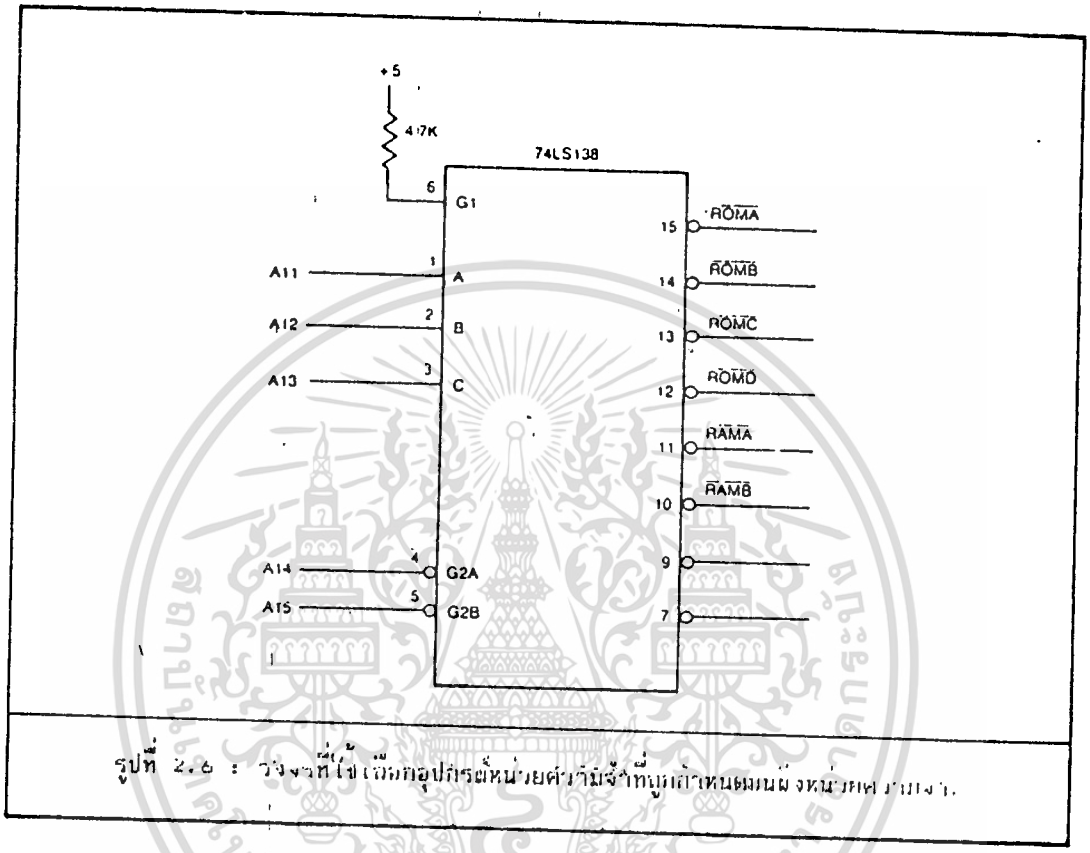
รูปที่ 2.5 : ผังหน่วยความจำของระบบไมโครโปรเซสเซอร์ ซึ่งเราใช้กำหนดพื้นที่ของหน่วยความจำแต่ละตัว โดยทั่วไปจะถูกออกแบบไว้ก่อนที่จะสร้างฮาร์ดแวร์ของระบบ.

หมายเหตุ ในรูปที่ 2.5 ค่าแอดเดรส 0000H จะใช้เป็นแอดเดรสเริ่มต้นของ ROM เมื่อ Z80 ถูกรีเซ็ทจากภายนอกหรือเป็นการรีเซ็ทที่เกิดจากการจ่ายไฟเลี้ยงให้แก่ระบบในครั้งแรก Z80 จะกระโดด(JUMP)ไปทำงานที่แอดเดรส 0000H.

2.6 การสร้างสัญญาณ CHIP SELECT สำหรับ ROM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 จะเห็นได้ว่า ROM A จะถูกเลือกเมื่อ Z80 มีเอาท์พุทแอดเดรส อยู่ระหว่าง 0000H กับ 07FFH ฉะนั้นระบบของเราจึงจำเป็นที่จะต้องมีสัญญาณที่ใช้เป็นตัวเลือกแอดเดรสให้อยู่ในช่วงที่เราต้องการ.



รูปที่ 2.6 : วงจรที่ใช้เลือกอุปกรณ์หน่วยความจำที่มีกำหนดขนาดหน่วยความจำ.

จากรูป 2.6 ได้แสดงวงจรที่ใช้เลือกแอดเดรสของหน่วยความจำ ซึ่ง เมื่อ A11, A12, A13, A14 และ A15 มีลอจิก "0" จะทำให้ขา 15 ของ IC 74LS138 มีลอจิกเป็น "0" (ROM A ถูกเลือก) และขา 15 จะเป็น "0" ภายในช่วงของแอดเดรสที่กำหนดไว้เท่านั้น ในตารางรูปที่ 2.7 ได้แสดงถึงการตอบสนองของขาเอาท์พุทของ IC 74LS138 เมื่อมีค่าแอดเดรสบนบัสแอดเดรสในช่วงต่างๆกัน.

เราจะเห็นได้ว่าในรูป 2.7 นั้น เมื่อสัญญาณที่ขา A11 มีลอจิกเป็น "1" ขา 1 ของ IC 74LS138 จะมีลอจิกเป็น "1" ด้วย และขา 14 จะมีลอจิกเป็น "0" ถ้าหากค่าแอดเดรสจาก Z80 อยู่ในช่วง 0000-FFFFH คือ สัญญาณที่ขา A11 มีลอจิกเป็น "0" ทำให้ขา 1 ของ 74LS138 มีลอจิกเป็น "0" ในทำนองเดียวกัน เราสามารถเลือกเอาท์พุทอื่นๆได้ โดยพิจารณาจากตารางรูป 2.7

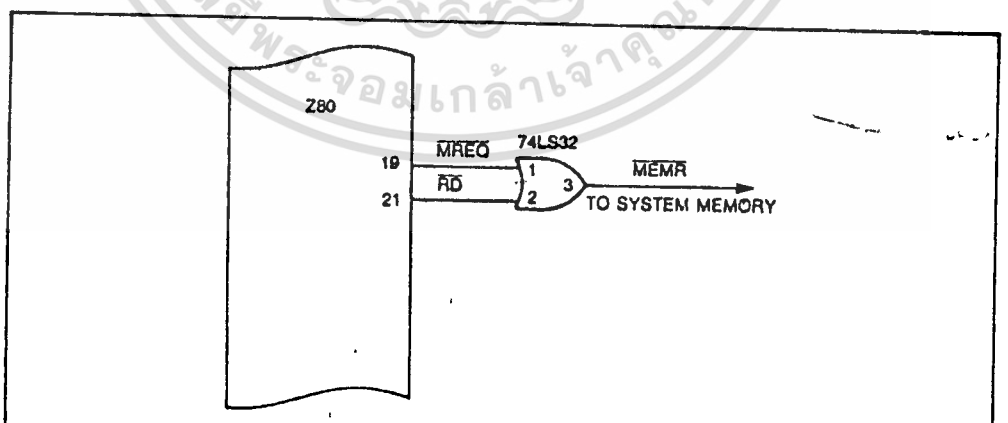
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A15	A14	A13	A12	A11	A10 ——— A0	HEX	PIN # = 0
0	0	0	0	0	0 ---------- 0	0000	15
0	0	0	0	0	1 ---------- 1	07FF	15
0	0	0	0	1	0 ---------- 0	0800	14
0	0	0	0	1	1 ---------- 1	0FFF	14
0	0	0	1	0	0 ---------- 0	1000	13
0	0	0	1	0	1 ---------- 1	17FF	13
0	0	0	1	1	0 ---------- 0	1800	12
0	0	0	1	1	1 ---------- 1	1FFF	12
0	0	1	0	0	0 ---------- 0	2000	11
0	0	1	0	0	1 ---------- 1	27FF	11
0	0	1	0	1	0 ---------- 0	2800	10
0	0	1	0	1	1 ---------- 1	2FFF	10

รูปที่ 2.7 : ผังหน่วยความจำของรูป 2.6.

2.7 การสร้างสัญญาณสำหรับอ่านหน่วยความจำ.

ในส่วนนี้เราจะมาดูวิธีการที่จะต่อขา CHIP SELECT INPUT เข้ากับ Z80 ในขั้นแรกจะต้องศึกษาเกี่ยวกับสัญญาณเวลาที่ใช้ในการควบคุมที่ส่งจาก Z80 เพื่อใช้ในการเริ่มและหยุดการส่งข้อมูล และเรียกสัญญาณนี้ว่า MEMORY READ ($\overline{MEMR}$) สัญญาณ $\overline{MEMR}$ และสัญญาณ MEMORY SELECT จะทำงานร่วมกันเพื่อที่จะนำเอาข้อมูลที่อยู่ใน ROM ออกไปที่บัสข้อมูลในช่วงเวลาที่ถูกต้อง ในรูป 2.8 เป็นวิธีหนึ่งที่ใช้ในการสร้างสัญญาณ $\overline{MEMR}$.



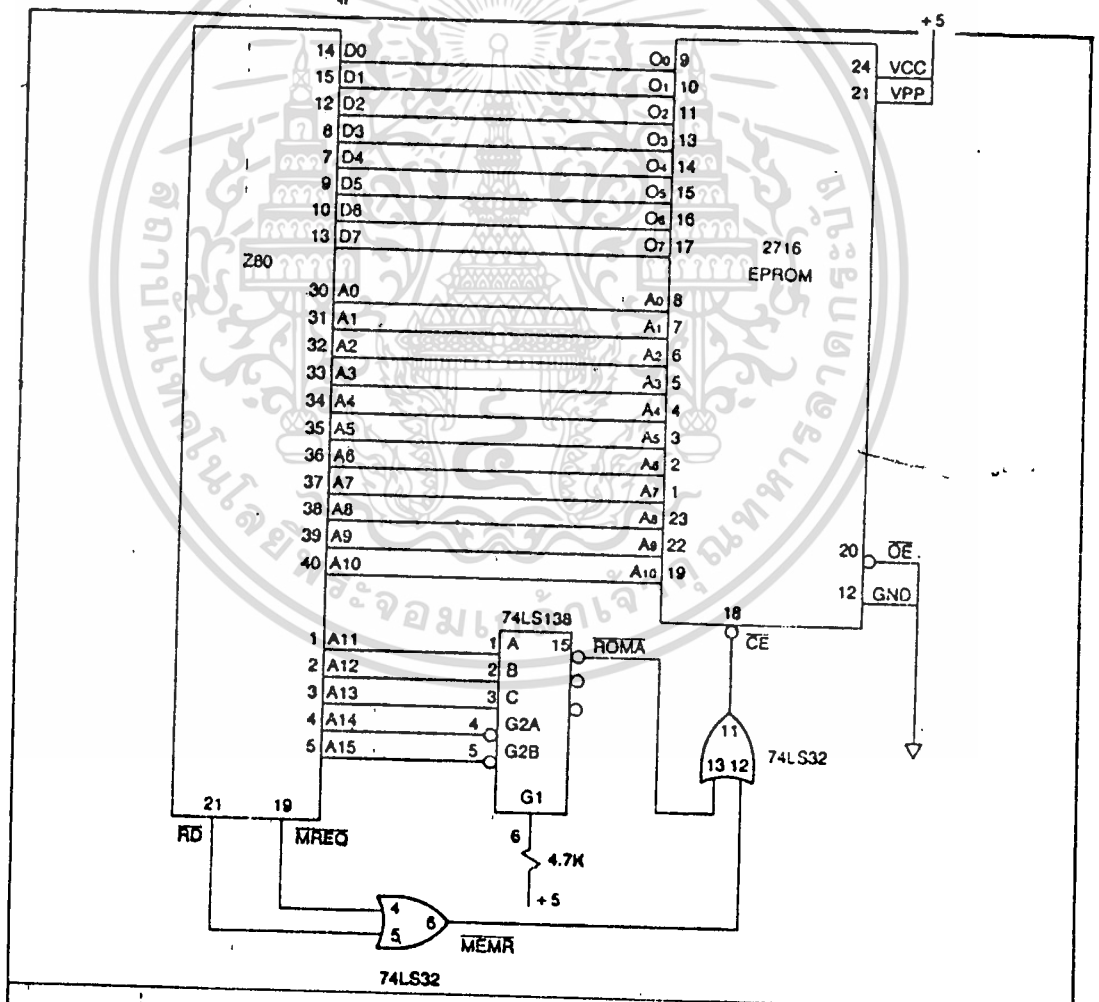
รูปที่ 2.8 : แสดงวิธีการสร้างสัญญาณควบคุมในขบวนการการอ่านข้อมูลจากหน่วยความจำ โดยได้มาจากการนำสัญญาณ $\overline{MREQ}$ และ $\overline{RD}$ มา OR กัน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Z80 ต้องการติดต่อกับหน่วยความจำ สัญญาณ $\overline{MREQ}$ จาก Z80จะถูกทำให้เป็นลอจิก"0" และ $\overline{RD}$ ซึ่งเป็นสัญญาณเวลาที่ใช้ควบคุมอีกสัญญาณหนึ่งที่ได้จาก Z80 เมื่อ Z80 ต้องการรับข้อมูลจากหน่วยความจำหรือจากอุปกรณ์ I/O, Z80ก็จะทำให้สัญญาณนี้มีลอจิกเป็น "0".

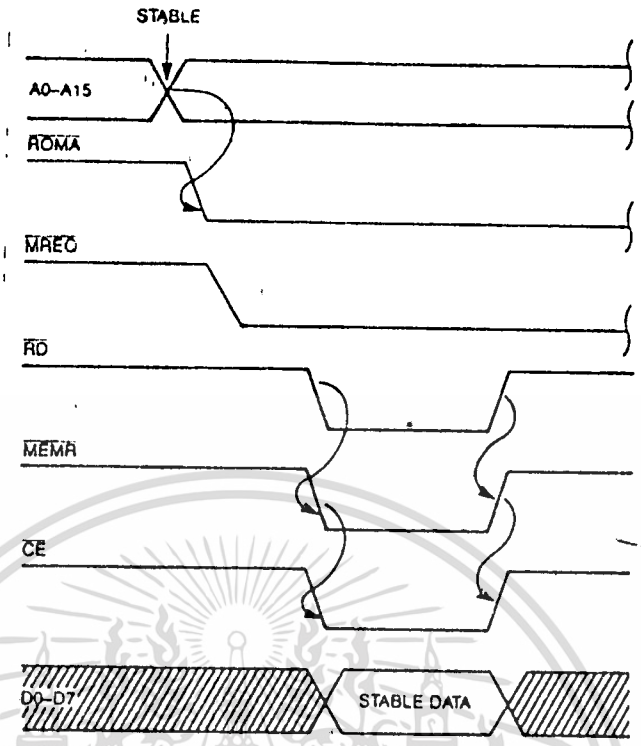
2.8 การต่อ CHIP SELECT LINES.

รูป 2.9 แสดงให้เห็นลักษณะการต่อ Z80 กับ ROM อย่างสมบูรณ์ เมื่อสัญญาณ MEMORY SELECT และ MEMORY READ (MEMR) เป็นลอจิก "0" ทั้งคู่ แล้วสัญญาณ CHIP SELECT ที่ต่อเข้ากับ EPROM เบอร์ 2716 จะทำงาน (มีลอจิกเป็น "0") รูป 2.10 แสดงไดอะแกรมเวลาของความสัมพันธ์ของสัญญาณที่สำคัญในระหว่างที่มีการอ่านข้อมูลจาก ROM.

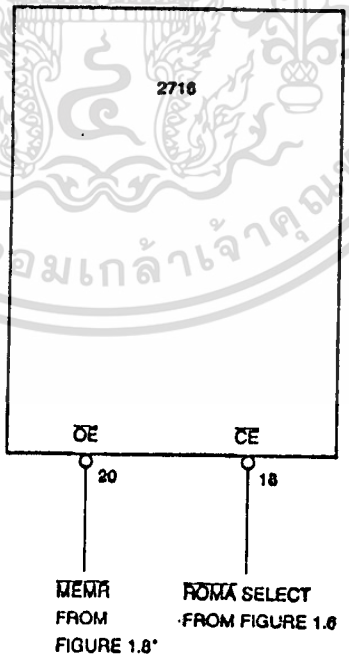


รูปที่ 2.9 : ผังวงจรสมบูรณ์แสดงการต่อ 2716 EPROM กับ Z80 ไมโครโปรเซสเซอร์ สัญญาณลอจิกที่ใช้เลือกอุปกรณ์หน่วยความจำ ถูกออกแบบไว้ตามผังหน่วยความจำ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 : ไดอะแกรมเวลาแสดงความสัมพันธ์ของสัญญาณที่สำคัญ ซึ่งใช้ในการอ่านข้อมูลจากหน่วยความจำตามรูป 2.9.



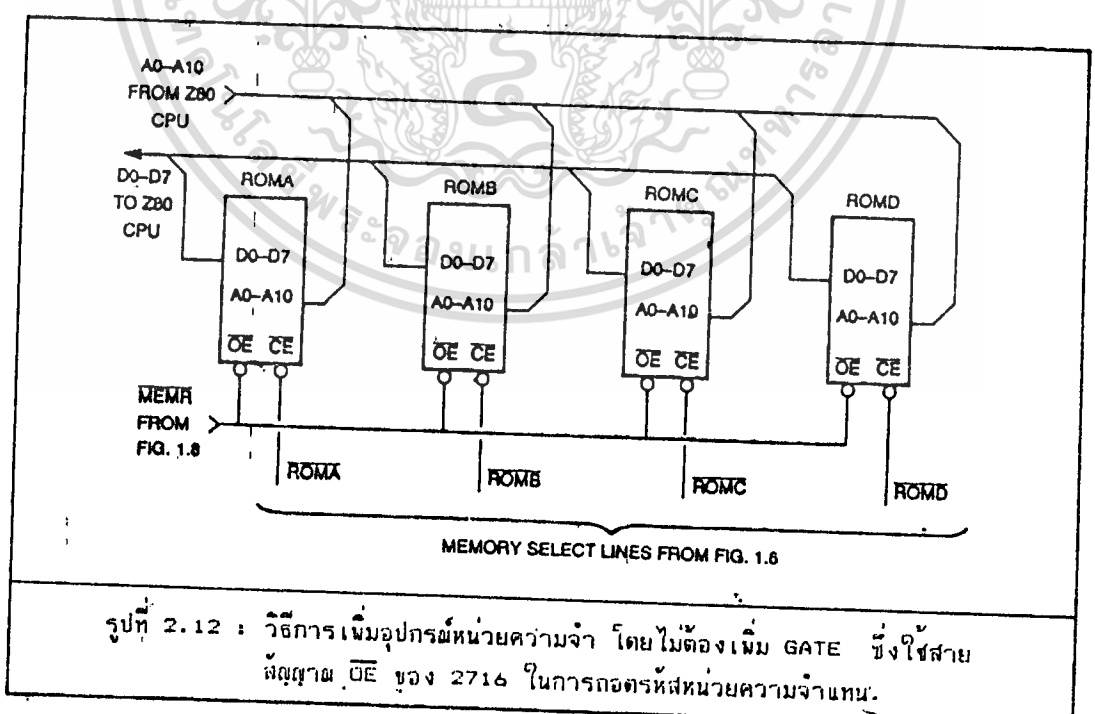
รูปที่ 2.11 : แสดงการใช้งานขา $\overline{OE}$ (OUTPUT ENABLE) และ $\overline{CE}$ (CHIP SELECT) ซึ่งทำให้ 2716 สามารถส่งข้อมูลออกมาได้.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ในรูปที่ 2.11 เป็นอีกวิธีหนึ่งที่สามารถนำมาใช้ในการเลือกข้อมูลจาก ROM ให้ออกมาบนบัสข้อมูลในเวลาที่ถูกต้อง วิธีนี้อาศัยความได้เปรียบของอุปกรณ์ถอดรหัสที่อยู่ภายในตัว 2716 เอง เพื่อที่จะทำให้ข้อมูลที่ต้องการถูกส่งออกมาบนบัสข้อมูล เราจำเป็นต้องทำให้ทั้ง $\overline{OE}$ (ขา 20) และ $\overline{CE}$ (ขา 18) มีลอจิกเป็น"0" พร้อมกันซึ่งจากวิธีเดิมเราต่อขา $\overline{OE}$(ขา 20) ลงกราวด์(ซึ่งเป็นสภาวะของการทำงานของขานี้).

2.9 วิธีการใช้ ROM หลายๆตัว.

เราสามารถนำ ROM หลายๆตัวมาใช้ในระบบได้ โดยอาศัยวิธีการเลือกที่เหมาะสม ดังแสดงในรูป 2.12 ซึ่งนำวิธีในรูป 2.11 มาใช้ เทคนิคนี้ทำให้เราสามารถเพิ่มจำนวน ROM ที่ใช้ในระบบได้มากขึ้นโดยไม่ต้องเพิ่มจำนวน GATE ขึ้นอีก (แต่วิธีการที่แสดงในรูป 2.10 จะต้องเพิ่มจำนวน GATE) แต่จะต้องจำไว้ว่าใน ROM แต่ละตัวจะมีกลุ่มแอดเดรสโดยเฉพาะสำหรับตัวมันเอง อย่างไรก็ตามเมื่อ Z80 ทำงานตามโปรแกรมจะไม่สามารถแยกออกว่า ข้อมูลเหล่านั้นมาจากที่ใด-เช่นในคำสั่งที่มี 3 ไบท์ ไบท์แรกอาจอยู่ในตำแหน่งสุดท้ายของ ROM A และ 2 ไบท์สุดท้ายอยู่ในตำแหน่งแรกๆของ ROM B เป็นต้น.



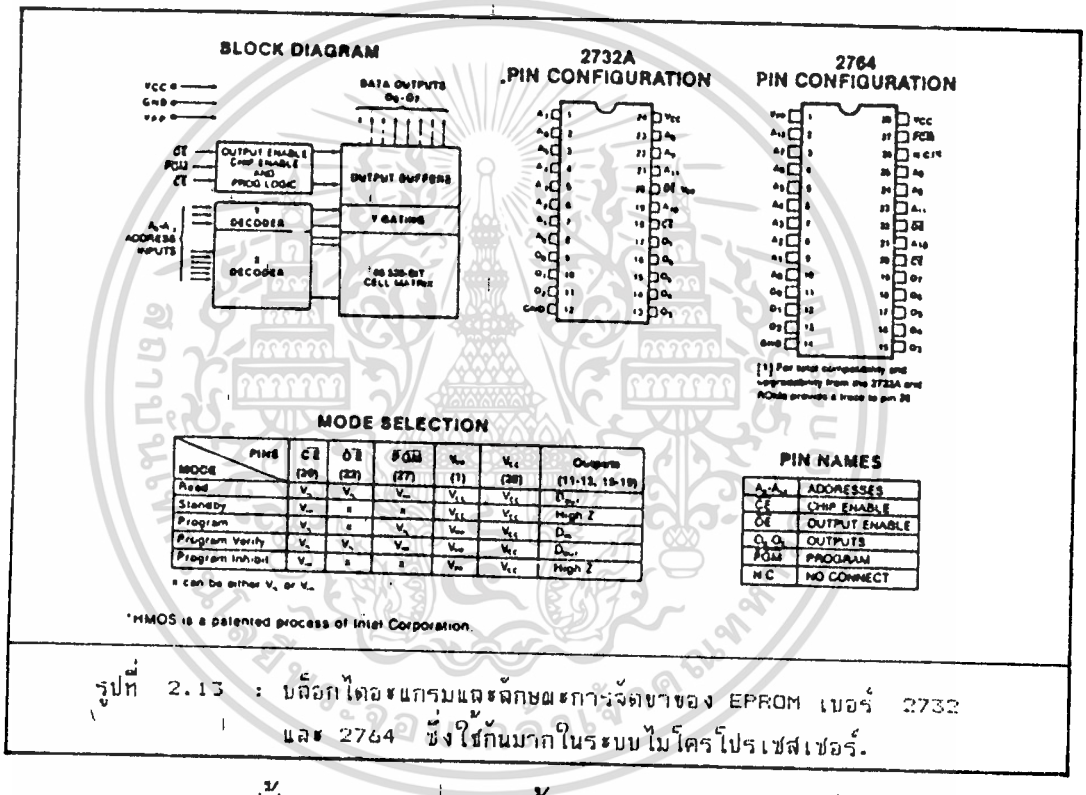
รูปที่ 2.12 : วิธีการเพิ่มอุปกรณ์หน่วยความจำ โดยไม่ต้องเพิ่ม GATE ซึ่งใช้สายสัญญาณ $\overline{OE}$ ของ 2716 ในการถอดรหัสหน่วยความจำแทน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ผังหน่วยความจำของ ROM ชนิดความจุมาก.

ROM ที่มีความจุมากๆที่นิยมใช้ในระบบอยู่ 2 ตัว คือ เบอร์ 2732 และ เบอร์ 2764 IC สองตัวนี้มีโครงสร้างเป็น 4096x8 และ 8192x8 ตามลำดับ ซึ่งได้แสดงตำแหน่งขาของทั้งสองตัวไว้ในรูป 2.13 .

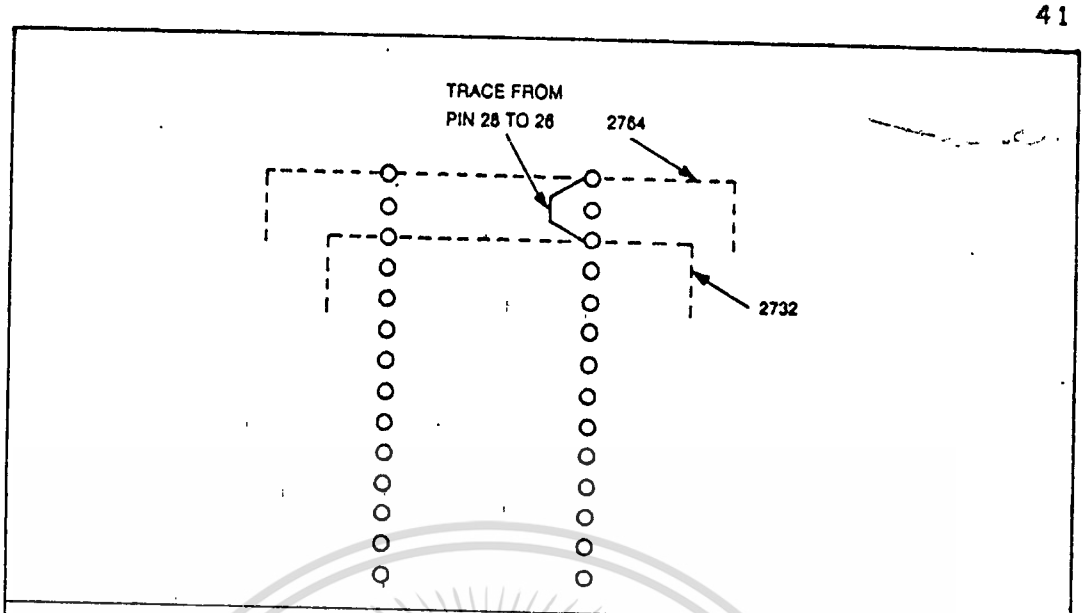
สิ่งที่น่าสนใจอีกสิ่งหนึ่งก็คือการจัดเรียงขาของ 2732 และ 2764 ทำให้ทั้งสองตัวนี้สามารถนำมาใช้แทนกันได้ แต่สิ่งที่แตกต่างกันก็คือจำนวนขาซึ่งสามารถแก้ไขได้โดยใช้ SOCKET ขนาด 28 ขาแล้วทำการต่อ Vcc เข้าที่ขา 28 และ 26 ของซอคเก็ท (ดังในรูปที่ 2.14).



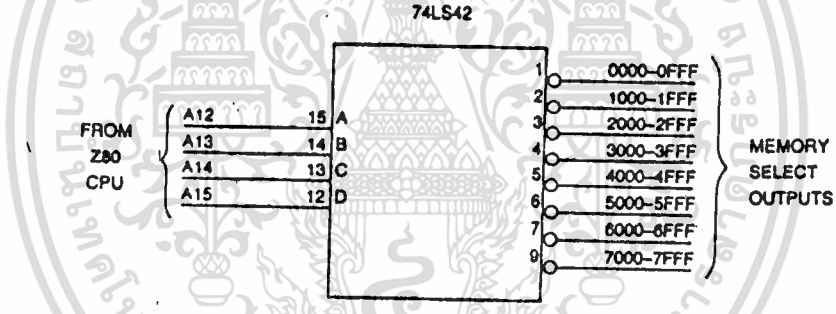
รูปที่ 2.13 : บล็อกไดอะแกรมและลักษณะการจัดขาของ EPROM เบอร์ 2732 และ 2764 ซึ่งใช้กันมากในระบบไมโครโปรเซสเซอร์.

IC คู่นี้ถูกออกแบบมาให้ใช้แทนกันได้ ในกรณีที่ผู้ออกแบบระบบเริ่มแรกต้องการใช้เบอร์ 2732 ก็เพียงพอ แต่เมื่อต้องการขยายระบบ ให้มีหน่วยความจำมากขึ้นก็สามารถเปลี่ยนมาใช้เบอร์ 2764 ได้ โดยไม่จำเป็นต้องทำการขยายบอร์ดให้ยุ่งยากอีกในกรณีนี้ผังหน่วยความจำจะต้องออกแบบเผื่อไว้แล้ว หรือเปลี่ยนจากรูป 2.5 ไปเป็นดังรูป 2.15 ดังนั้น ROM 2732 หนึ่งตัว จะถูกใช้สำหรับแอดเดรส 0000--0FFFH(ซึ่งในรูป 2.5 จำเป็นต้องใช้ 2716 ถึงสองตัว) ส่วนอุปกรณ์ฮาร์ดแวร์ที่ใช้ใน MEMORY SELECT LINE จะเหมือนกับวงจรที่ใช้กับ EPROM เบอร์ 2716 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 2.14 : บล็อกไดอะแกรมแสดง SOCKET ขนาด 28 ขา ซึ่งใช้ได้ทั้งกับ EPROM เบอร์ 2732 และ 2764.



รูปที่ 2.15 : แสดงผังฮาร์ดแวร์ของการถอดรหัสสายแอดเดรส เพื่อใช้ในการเลือกตำแหน่งของหน่วยความจำ โดยใช้ 74LS42 (BCD TO DECIMAL DECODER).

2.11 การเพิ่มบัฟเฟอร์ให้กับแอดเดรสบัส.

ระบบหน่วยความจำที่เราได้พิจารณามาแล้วนั้นเป็นแบบพื้นฐานที่ง่ายๆ แต่อย่างไรก็เป็นที่นิยมใช้กันอย่างกว้างขวางในอุตสาหกรรมต่างๆ จะสังเกตได้ว่ามีสิ่งหนึ่งที่น่าสนใจในการออกแบบระบบนั้นคืออาจมีการทำงานผิดพลาดเกิดขึ้นบนบัสแอดเดรสของ Z80 เนื่องจาก Z80 ไม่สามารถที่จะจ่ายกระแสให้แก่โหลดได้เพียงพอ(เกิดกรณี LOADING EFFECT).

ถ้าเราพิจารณาข้อมูลต่างๆของ Z80 ดังแสดงในรูป 2.16 ซึ่งมีข้อกำหนดสำหรับ I_{OL} (กระแสเอาท์พุทเมื่อมีลอจิกเป็น "0") และ I_{OH} (กระแสเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

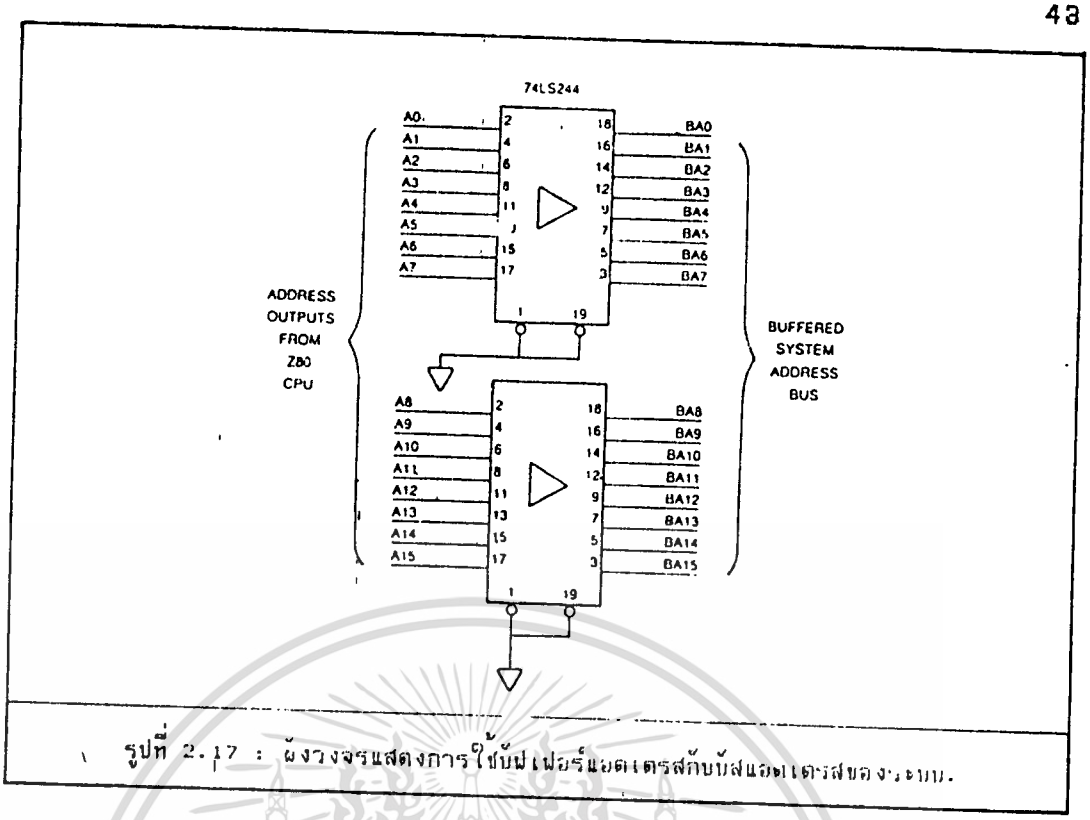
เมื่อมีลอจิกเป็น "1") สำหรับ ADDRESS OUTPUT LINE มีค่าเท่ากับ 1.8 mA และ -250 uA ตามลำดับ (เครื่องหมายลบเป็นการแสดงว่ากระแสไหลออกจากตัว Z80) จากข้อมูลนี้แสดงให้เห็นว่า ADDRESS OUTPUT LINE นี้สามารถที่จะมีโหลดที่เป็นแบบ TTL มาตรฐาน(STANDARD TTL) ได้เพียงตัวเดียว TTL จะต้องกระแสเข้าสู่ตัวมันประมาณ 1.6 mA เมื่อเป็นลอจิก "0" และจ่ายกระแสได้ 40 uA เมื่อเป็นลอจิก "1" ดังรูป 2.16 .

ในระบบหน่วยความจำที่เราได้ออกแบบไว้ก่อนหน้านี้นอุปกรณ์ TTL ที่เราใช้คือ IC 74LS138 LS ในที่นี้หมายถึง LOW POWER SCHOTTKY IC TTL ชนิดนี้จะมีคุณสมบัติทางกระแสและจ่ายกระแสเทียบเท่ากับ 1/3 ของ TTL มาตรฐาน และเราได้ใช้เป็นโหลดอยู่บน ADDRESS OUTPUT LINE A11-A15 ส่วน ADDRESS OUTPUT LINE เส้นอื่นทั้งหมดของ Z80 จะมีโหลดเป็นเพียงอินพุทแอดเดรสของหน่วยความจำเท่านั้น โดยทั่วๆไป อินพุทแอดเดรสของหน่วยความจำจะต้องการกระแส 10 uA ทั้งในลอจิก "0" และ "1" อย่างไรก็ตามบัสแอดเดรสอาจจะต้องต่อกับวงจรอื่นๆนอกเหนือจากหน่วยความจำ เช่น อุปกรณ์ I/O , CABLES หรือการเชื่อมโยงกับวงจรในบอร์ดอื่น หรือในการใช้งานอื่นๆบัสแอดเดรสมีโหลดที่เกินขีดจำกัดที่ Z80 จะรับได้ จึงจำเป็นต้องใช้บัฟเฟอร์แอดเดรส และควรจะใช้บัฟเฟอร์ที่ เมื่อบัสแอดเดรสใช้ในการขับสาย CABLE แม้ว่าโหลดทาง DC จะไม่เกินขีดจำกัด แต่จะมีค่า CAPACITIVE LOAD ของ CABLE แทนบนบัสแอดเดรส ในรูป 2.17 เป็นการเพิ่มแอดเดรสบัฟเฟอร์วงหนึ่งให้กับระบบ Z80 จะสังเกตได้ว่าระบบของ Z80 ไม่จำเป็นต้องใช้แอดเดรสบัฟเฟอร์เสมอไป ทั้งนี้ขึ้นอยู่กับลักษณะการใช้งานของระบบนั้น.

D.C. CHARACTERISTICS
T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITION
V_{ILC}	Clock Input Low Voltage	-0.3		0.8	V	
V_{IHC}	Clock Input High Voltage	Vcc-.8		Vcc+.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 1.8mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250 μA
I_{CC}	Power Supply Current			150*	mA	
I_{LI}	Input Leakage Current			±10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri State Output Leakage Current in Float			10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	0≤V_{IN}≤V_{CC}

รูปที่ 2.16 : ส่วนหนึ่งของ DATA SHEET แสดงค่า I_{OL} และ I_{OH} ของ CURRENT SINK และ CURRENT SOURCE ของ Z80.



รูปที่ 2.17 : ผังวงจรแสดงการใช้บัฟเฟอร์แอดเดรสกับบัสแอดเดรสของระบบ.

2.12 การเพิ่มบัฟเฟอร์ให้กับบัสข้อมูล.

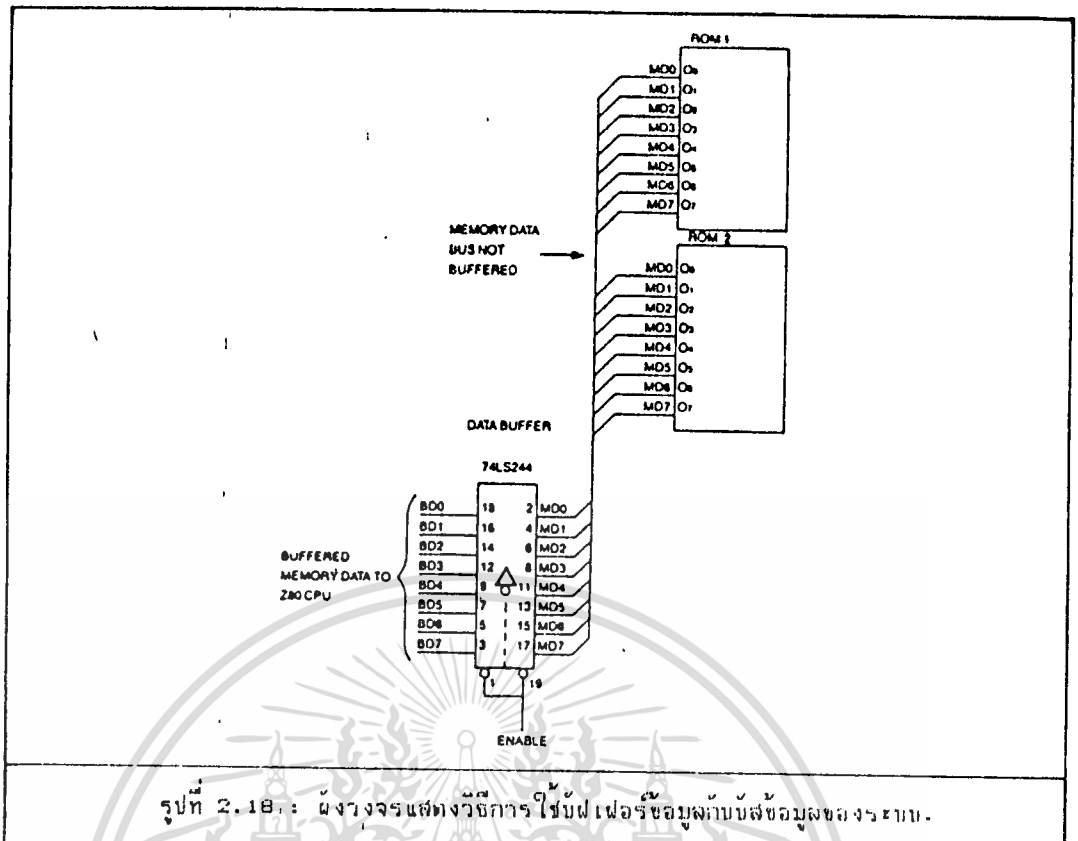
การใช้งานในบางระบบ, ROM อาจจะรับหรือจ่ายกระแสไม่เพียงพอสำหรับขับบัสข้อมูล ดังนั้นจึงจำเป็นที่จะต้องมีบัฟเฟอร์ให้กับบัสข้อมูล ซึ่งมีลักษณะคล้ายกับที่ใช้ในบัสแอดเดรสที่เคยกล่าวมาแล้ว.

บัฟเฟอร์ที่ใช้ในบัสข้อมูลจะเป็นแบบสองทิศทาง (BI-DIRECTIONAL) และมีลักษณะเป็น TRI-STATE ซึ่งเป็นสิ่งที่จำเป็นเพราะทำให้ระบบหน่วยความจำสามารถแยกตัวเองออกจากบัสข้อมูลได้ เมื่อ Z80 ไม่ต้องการติดต่อด้วย รูปที่ 2.18 แสดงการใช้บัฟเฟอร์ข้อมูลกับระบบ ROM จะเห็นได้ว่าข้อมูลที่ออกจาก ROM เหล่านี้มีหน้าที่เพียงขับอินพุทของบัฟเฟอร์ข้อมูลเท่านั้น ส่วนเอาท์พุทของบัฟเฟอร์ข้อมูลจะขับโหลดของบัสข้อมูลทั้งหมดของระบบ.

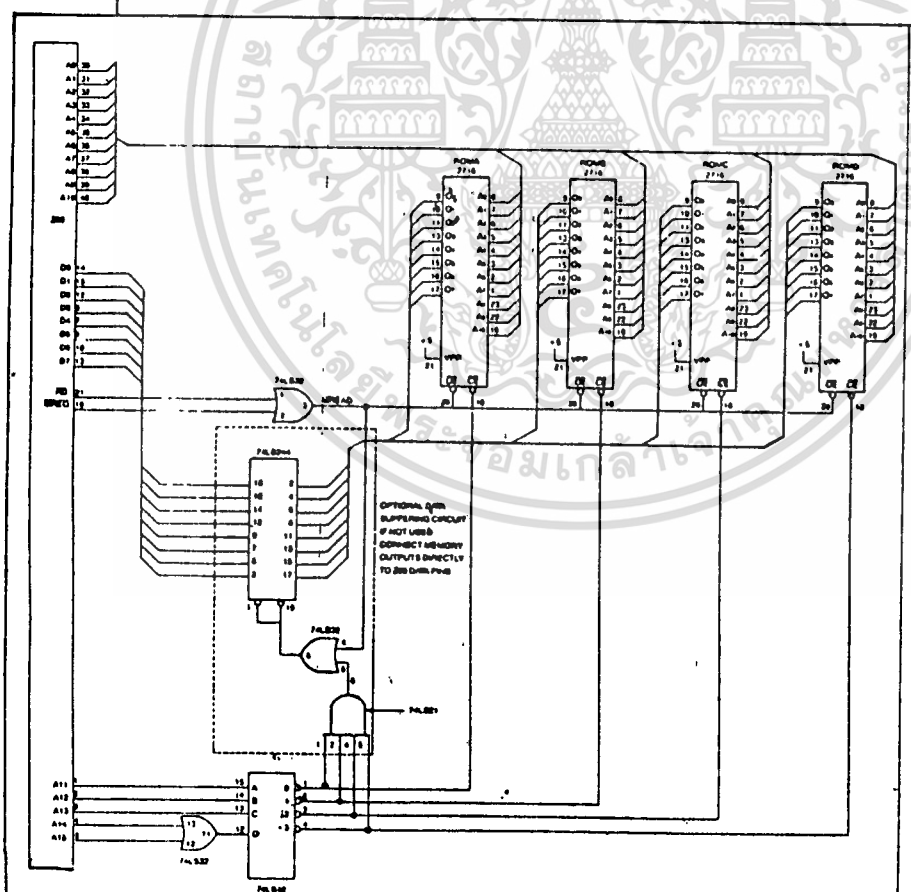
2.13 ตัวอย่างของระบบที่สมบูรณ์ที่ใช้กับ ROM.

รูปที่ 2.19 , 2.20 และ 2.21 แสดงระบบ ROM ที่สมบูรณ์ถูกสร้างขึ้นตามทฤษฎีที่ได้กล่าวมา ผังหน่วยความจำ (MEMORY MAP) ของระบบเหล่านี้เป็นดังรูป 2.22 ซึ่งอุปกรณ์ที่ใช้ในระบบที่แสดงนี้เป็นอุปกรณ์ที่หาได้ทั่วไปในปัจจุบัน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

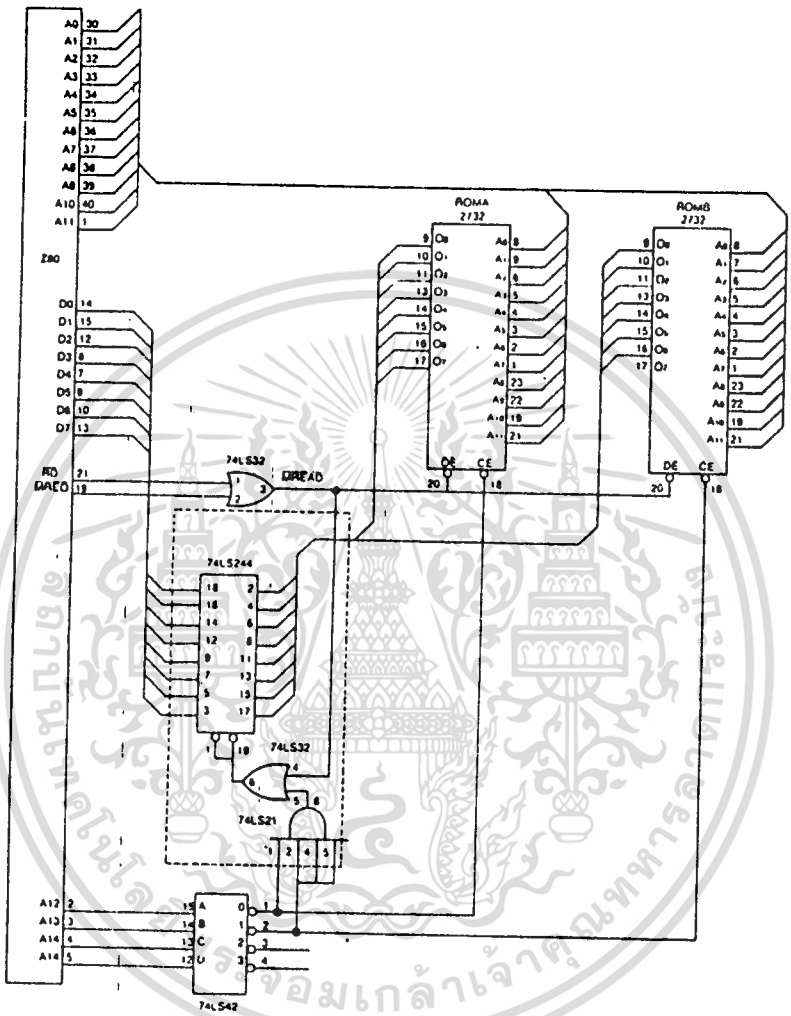


รูปที่ 2.18 : ผังวงจรแสดงวิธีการใช้บัฟเฟอร์ข้อมูลกับบัสข้อมูลของระบบ.



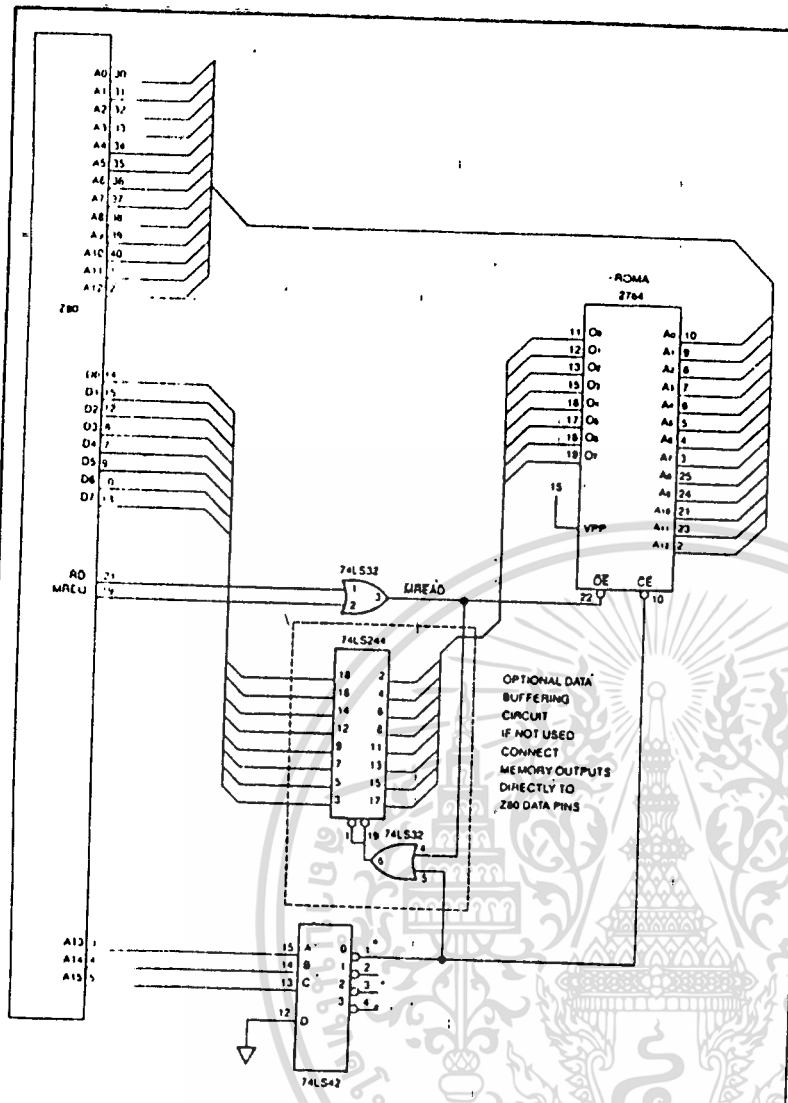
รูปที่ 2.19 : ระบบหน่วยความจำสมบูรณ์ที่ออกแบบตามผังหน่วยความจำของรูป 2.22 โดยใช้ 2716 เป็นอุปกรณ์หน่วยความจำ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

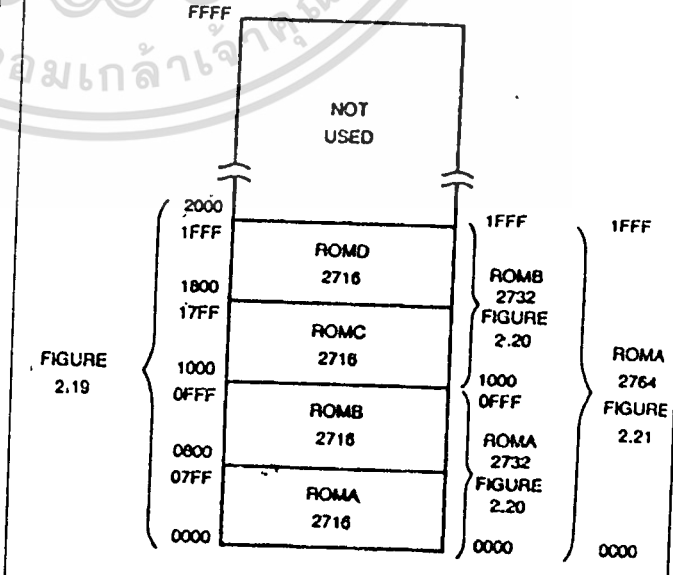


รูปที่ 2.20 : ระบบหน่วยความจำสมบูรณ์ที่ออกแบบตามผังหน่วยความจำของรูป 2.22 โดยใช้ 2732 เป็นหน่วยความจำ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 : วงจรหน่วยความจำสมบูรณ์ ที่ออกแบบตามผังหน่วยความจำของรูป 2.22 โดยใช้ 2764 เป็นหน่วยความจำ.



รูปที่ 2.22 : ผังหน่วยความจำ แสดงตำแหน่งของหน่วยความจำชนิดต่างๆ เมื่อใช้ อุปกรณ์หน่วยความจำ ผังเกตดูว่าผังนี้ต้องใช้ 2716 4 ตัว, 2732 2 ตัว หรือ 2764 1 ตัว.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การใช้ STATIC RAM กับ Z-80

ในบทนี้จะกล่าวถึงการนำ RAM (RANDOM ACCESS MEMORY) มาเชื่อมต่อ(INTERFACE)กับไมโครโปรเซสเซอร์ Z80 RAM ถูกใช้เป็นตัวเก็บโปรแกรม, ข้อมูล หรือ ตัวแปร เป็นการชั่วคราว (ซึ่งแตกต่างกับ ROM ที่กล่าวมาแล้วในบทที่ 2) และข้อมูลที่เก็บไว้ใน RAM จะสูญหายไปเมื่อไม่มีการจ่ายไฟเลี้ยงให้.

การเชื่อมต่อระหว่าง Z80 กับ STATIC RAM ไม่ได้เป็นเรื่องที่ยุ่งยาก ในบทนี้จะอธิบายถึงรายละเอียดของ STATIC RAM ที่ใช้ในโรงงานอุตสาหกรรม โดยใช้ STATIC RAM เบอร์ 2114 (1K/4บิท) และ เบอร์ 6116 (1K/8บิท) เป็นหลัก.

3.1 ลักษณะการทำงานของ STATIC RAM.

โดยเริ่มจากการพิจารณา การทำงานของหน่วยความจำชนิดสารกึ่งตัวนำที่สามารถอ่านและเขียนข้อมูลได้ เมื่อเข้าใจขั้นตอนนี้แล้ว จะพบว่าเป็นการง่ายที่จะเข้าใจวิธีการติดต่อระหว่าง Z80 กับ RAM ว่าเป็นอย่างไร แต่ถ้าผู้อ่านมีความเข้าใจเกี่ยวกับขั้นตอนเหล่านี้ดีแล้ว ผู้อ่านอาจจะข้ามเนื้อหาตอนแรกๆของบทนี้ไปได้.

RAM เป็นหน่วยความจำที่ใช้ในระบบไมโครโปรเซสเซอร์ซึ่งสามารถทั้งเขียนข้อมูลเข้าไปเก็บในตัวมันและอ่านออกมาได้, โดย CPU ต้องมีสัญญาณทางไฟฟ้าที่มีลักษณะเฉพาะสำหรับขบวนการเหล่านี้ ในรูป 3.1 แสดงบล็อกไดอะแกรมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ในตอนแรกนี้มาดูวิธีการต่อไฟเลี้ยงกันก่อน RAM จะต้องมีไฟเลี้ยงจ่ายให้ตลอดเวลา ซึ่งโดยทั่วๆไปใช้ศักดา +5 VOLTS กับ GROUND (เพื่อความถูกต้องขอให้ดูรายละเอียดจาก DATA SHEET ที่ได้จากผู้ผลิต) ที่จะกล่าวต่อไปนี้ขอสมมุติว่า RAM ทั้งหมดใช้ศักดา +5 VOLTS กับ GROUND.

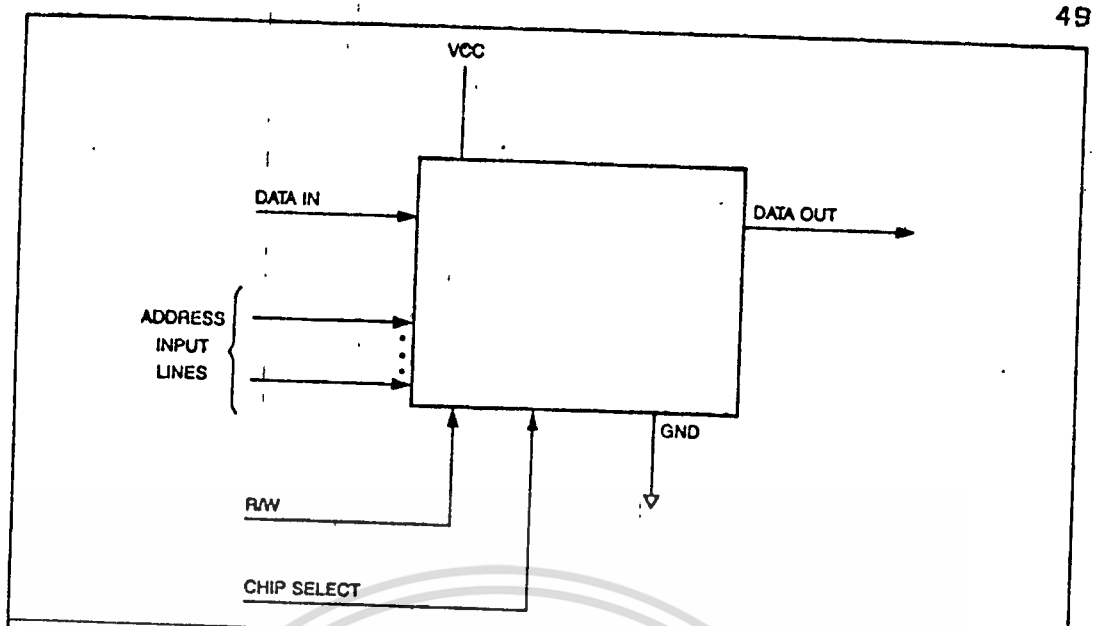
ในรูป 3.1 "DATA IN" เป็นสายที่ใช้นำข้อมูลที่ต้องการเขียนเข้ามาใน RAM ส่วน "DATA OUT" ใช้สำหรับอ่านข้อมูลที่อยู่ใน RAM ออกไป ใน RAM ตัวหนึ่งๆอย่างน้อยที่สุดจะต้องมี DATA IN หนึ่งสายและ DATA OUT หนึ่งสาย ซึ่งจำนวนที่แน่นอนของสายเหล่านี้ขึ้นอยู่กับโครงสร้างภายในของ RAM.

ในบทที่ 2 เราได้อธิบายถึงโครงสร้างภายในของ ROM มาแล้ว ซึ่งส่วนมากสามารถนำมาใช้อธิบาย RAM ได้ การบอกรายละเอียดโครงสร้างภายในของ RAM จะเป็นเช่นเดียวกับ ROM เช่น RAM ที่บอกมาเป็น 256x1 , 256x4 หรือ 1024x8 และอื่นๆ.

ตัวเลขตัวหลังของแต่ละชุด (1,4,8,...) จะบอกให้ทราบถึงจำนวนเส้นของ DATA IN และ DATA OUT ที่มีอยู่ หรือกล่าวได้ว่าทุกๆครั้งที่หน่วยความจำมีการเขียนข้อมูลเข้าไปหรืออ่านออกมา ตัวเลขตัวหลังจะกำหนดถึงจำนวนบิทของข้อมูลที่ถูกเขียนหรืออ่าน.

ต่อไป มาดู ADDRESS INPUT LINE ในรูป 3.1 (เราได้รู้กันมาบ้างแล้วในเรื่องของ ROM) ADDRESS LINE ของSTATIC RAM มีหน้าที่สำหรับเลือกตำแหน่งภายใน RAM เพื่อนำข้อมูลเข้าไปเขียนหรืออ่านออกมาในตำแหน่งนั้นๆ การกำหนดจำนวนตำแหน่งใน RAM สามารถใช้วิธีง่ายๆ คือการนับจำนวน ADDRESS LINE ซึ่งถ้าแปลงกลับก็จะได้จำนวนตำแหน่งจริง (เช่นมี ADDRESS LINE 10 เส้นก็จะมีจำนวนตำแหน่งที่แตกต่างกัน 2^{10}=1024 ตำแหน่ง) หรือสามารถกำหนดจำนวน ADDRESS LINE ของอุปกรณ์เมื่อรู้จำนวนตำแหน่งต่างๆที่มีอยู่ ตัวอย่างเช่น 1024x4 STATIC RAM จะมีความจุ 1024 หรือ 1K ตำแหน่งในแต่ละตำแหน่งจะถูกเขียนหรืออ่านพร้อมกันทีละ 4 บิท.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.1 : แสดงสัญญาณทางไฟฟ้าที่จำเป็นในการใช้อุปกรณ์จำพวก RAM.

เส้นสัญญาณสุดท้ายในรูปคือ (R/W) จะถูกใช้ทุกๆครั้งที่มีความต้องการเขียนหรืออ่านข้อมูลกับ RAM .

ต่อไปเราจะมาดูขั้นตอนที่เกิดขึ้นเป็นลำดับทุกๆครั้งที่ไมโครโปรเซสเซอร์ติดต่อกับ RAM จะมีการเขียนหรืออ่านข้อมูลเข้าออก ซึ่งในตอนแรกนี้จะกล่าวถึงขั้นตอนที่เกิดขึ้นในขบวนการอ่านก่อน.

3.2 ขั้นตอนที่เกิดขึ้นเมื่อทำการอ่าน RAM.

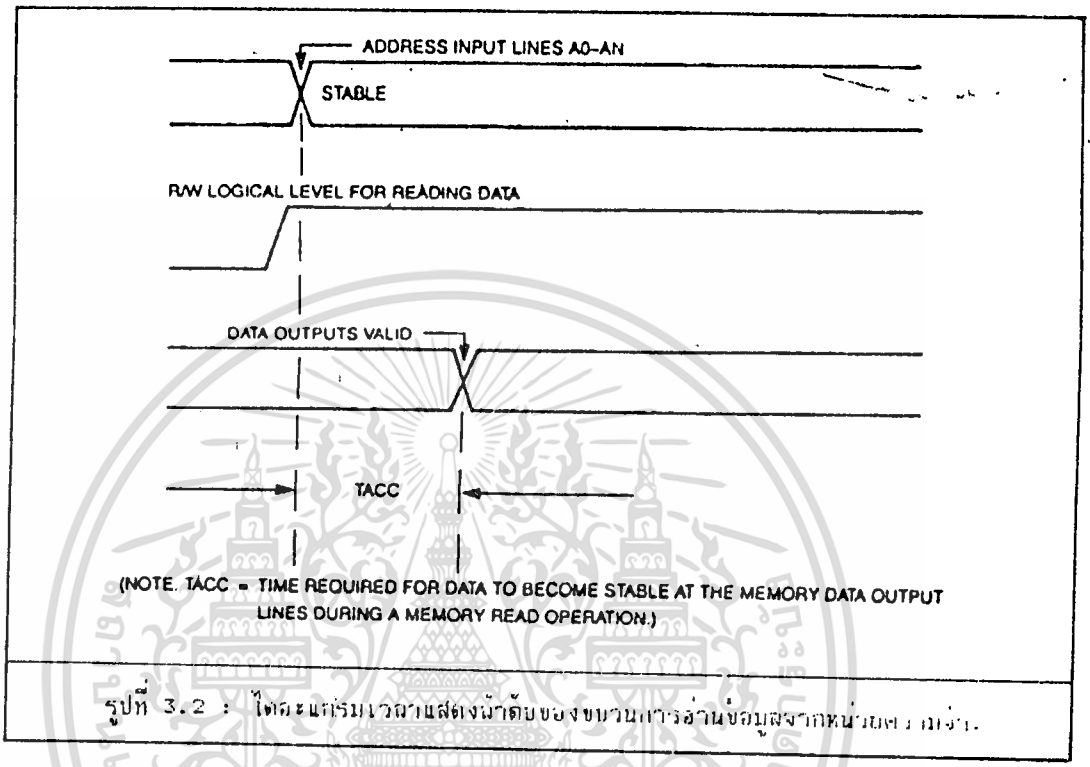
1. สัญญาณที่บ่งตำแหน่งแอดเดรสจะเข้ามาที่หน่วยความจำทางบัสแอดเดรส ในเวลาเดียวกัน RAM จะนำเอาสัญญาณนี้มาทำการถอดรหัสตำแหน่งภายใน ซึ่งเป็นตำแหน่งที่อยู่ของข้อมูลที่จะอ่าน.

2. สัญญาณบนเส้นควบคุม R/W จะถูกทำให้อยู่ในตำแหน่งลอจิกที่ถูกต้อง ซึ่งจะเป็นลอจิก "1" หรือ "0" ขึ้นอยู่กับหน่วยความจำนั้น (สามารถดูได้จาก DATA SHEET จากผู้ผลิต).

3. ระบบจะคอยอยู่ช่วงเวลาหนึ่ง เรียกช่วงเวลานี้ว่า READ ACCESS TIME เพื่อให้วงจรภายใน MEMORY CHIP ถอดรหัสตำแหน่งและเลือกข้อมูลในตำแหน่งนั้นๆ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากที่คอยอยู่ช่วงเวลาหนึ่งแล้ว ข้อมูลจะถูกส่งออกมาบน DATA OUT LINE และถูกอ่านโดยไมโครโปรเซสเซอร์ของระบบ ถ้าหากไมโครโปรเซสเซอร์อ่านข้อมูลทันที โดยไม่มี ACCESS TIME ข้อมูลที่ไม่ถูกต้องอาจจะถูกอ่านเข้าไปได้.



รูปที่ 3.2 : ไดอะแกรมเวลาแสดงจำดับของขบวนการอ่านข้อมูลจากหน่วยความจำ.

ขั้นตอนเหล่านี้แสดงในรูปที่ 3.2 และเราจะกลับมาดูกันอีกครั้งหนึ่งเมื่อกล่าวถึงการติดต่อระหว่าง ZBO กับ RAM ซึ่งขั้นตอนในการอ่านข้อมูลของ RAM เหมือนกับ ROM ที่จริงแล้วระบบไมโครโปรเซสเซอร์ไม่สามารถแยก(ทางไฟฟ้า)ออกได้ว่า ข้อมูลที่อ่านมานั้นมาจาก ROM หรือ RAM.

ต่อไปนี้เราจะมาดูขั้นตอน การทำงานระหว่างการเขียนข้อมูลเข้า RAM.

3.3 ขั้นตอนที่เกิดขึ้นเมื่อทำการเขียนข้อมูลเข้าไปใน RAM.

1. บน ADDRESS INPUT LINE จะมีสัญญาณออกมาบ่งตำแหน่งที่อยู่ภายในหน่วยความจำ ซึ่งเป็นตัวแหน่งที่จะนำข้อมูลเข้าไปเขียน.

2. ข้อมูลที่ต้องการเขียนจะปรากฏอยู่บน DATA INPUT LINE.

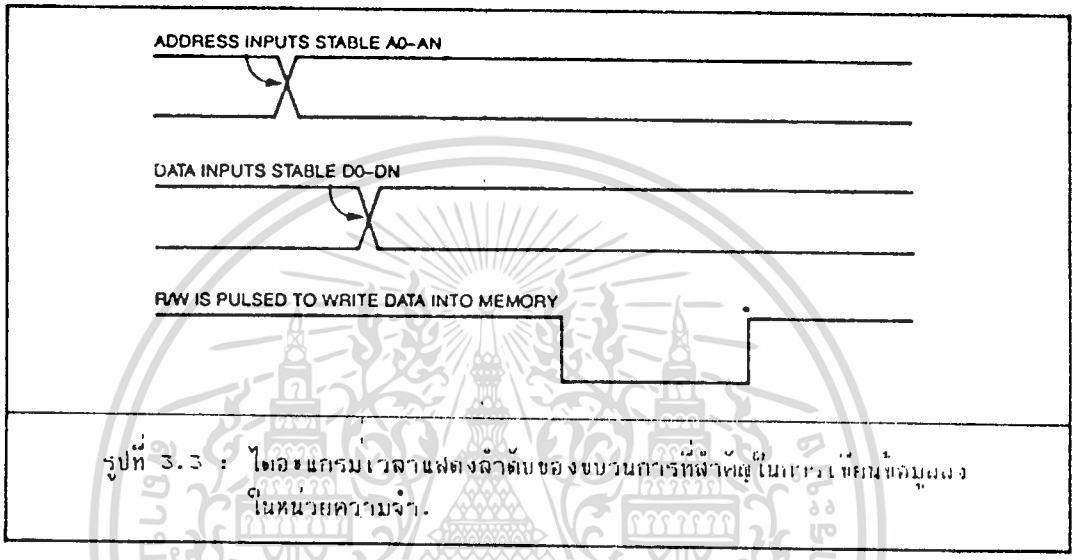
3. ระบบจะคอยอยู่ช่วงเวลาหนึ่ง ช่วงเวลานี้เรียกว่า WRITE ACCESS TIME (ซึ่งใช้เวลาประมาณ 200-300 nanoseconds) เพื่อที่จะทำให้วงจรถอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสภายใน อยู่ในสภาวะคงที่เสียก่อน.

4. หลังจากคอยอยู่ช่วงเวลาหนึ่งแล้ว เส้นควบคุม R/W ของหน่วยความจำก็จะทำให้เป็นระดับของสัญญาณที่ทำให้เกิดการเขียนเกิดขึ้น หรือเป็นพัลส์(PULSE) เพื่อที่จะทำให้ข้อมูลที่ DATA INPUT LINE ถูกเขียนเข้าไปใน RAM.

ในรูป 3.3 ได้แสดงไดอะแกรมเวลาของขั้นตอนที่เกิดขึ้นเหล่านี้แล้ว ซึ่งอุปกรณ์ฮาร์ดแวร์ที่ต้องการให้เป็นไปตามขั้นตอนนี้ ขึ้นอยู่กับไมโครโปรเซสเซอร์ที่ใช้.



รูปที่ 3.3 : ไดอะแกรมเวลาแสดงลำดับของขบวนการที่สำคัญในการเขียนข้อมูลลงในหน่วยความจำ.

3.4 อุปกรณ์หน่วยความจำที่ใช้งานจริง.

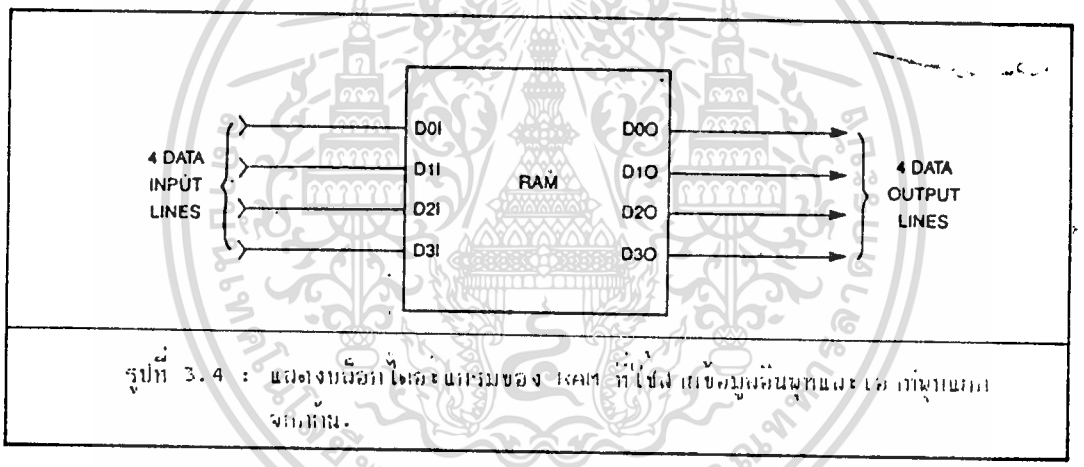
ต่อไปเราจะมาดูหน่วยความจำชนิด RAM ที่ใช้กันอย่างแพร่หลายในวงการอุตสาหกรรม จะเริ่มด้วยการพิจารณาคุณสมบัติที่สำคัญของตัวอุปกรณ์ก่อน แล้วต่อไปก็จะแสดงให้เห็นว่ามันสามารถนำข้อมูลเข้าไปเขียนและอ่านออกมาได้อย่างไร สิ่งที่จำเป็นอย่างยิ่งสำหรับการใช้ RAM ก็คือจะต้องทำความเข้าใจคุณสมบัติเฉพาะทางไฟฟ้าของอุปกรณ์ อย่างไรก็ตามเมื่อเข้าใจการทำงานของ STATIC RAM เบอร์ใดเบอร์หนึ่งแล้ว ก็เป็นการง่ายที่จะทำความเข้าใจ RAM เบอร์อื่นๆและนำไปใช้งานกับไมโครโปรเซสเซอร์เบอร์อื่นๆได้.

RAM ที่เราเลือกคือเบอร์ 2114 ซึ่งมีโครงสร้างเป็น STATIC, COMMON I/O ขนาด 1024 × 4 RAM ตัวหนึ่งๆจะมีลักษณะเป็น SEPARATE หรือ COMMON I/O อย่างใดอย่างหนึ่งเท่านั้น. SEPARATE I/O หมายถึง DATA INPUT LINE และ DATA OUTPUT LINE อยู่แยกออกจากกัน(ดังแสดงในรูปที่ 3.4), ส่วน COMMON I/O หมายถึง RAM ที่มี DATA INPUT LINE และ DATA OUTPUT

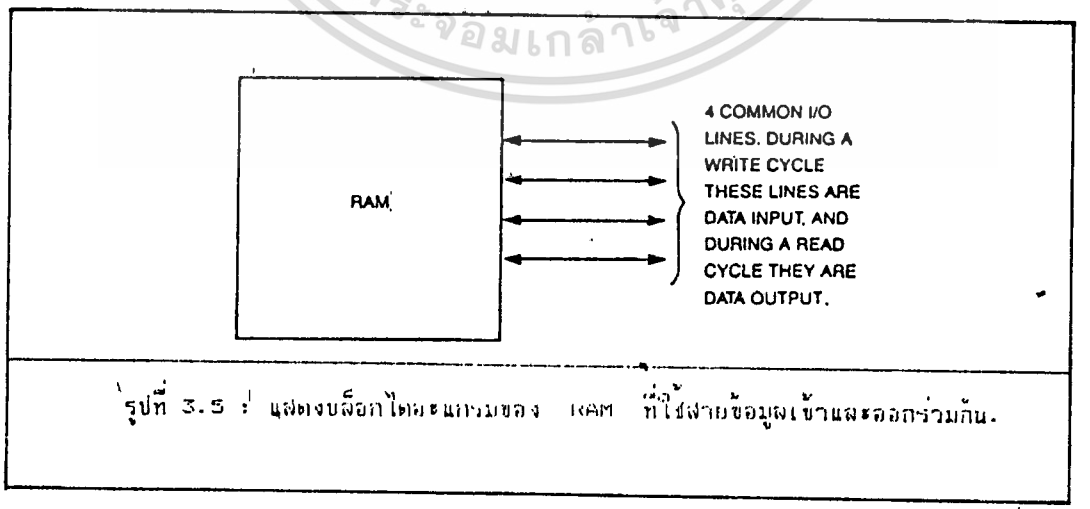
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LINE อยู่ร่วมกันในขาเดียวกัน. (ดังรูป 3.5).

เนื่องจาก RAM มีหน้าที่เพียงถูกเขียนและอ่านข้อมูลเท่านั้น ฉะนั้นขาดไม่ได้คือ DATA INPUT และ DATA OUTPUT LINE นอกจากนั้นการทำงานจะอ่านและเขียนในเวลาเดียวกันไม่ได้ ดังนั้น DATA INPUT LINE และ DATA OUTPUT LINE จะไม่ถูกใช้ในเวลาเดียวกัน จากความจริงที่กล่าวนี้ทำให้สามารถลดจำนวนขาของอุปกรณ์ สำรองไว้สำหรับในการเขียนและการอ่านใน RAM ได้ และก็เป็นหลักการที่ใช้ใน COMMON I/O การใช้ TIME-MULTIPLEXED ใน DATA INPUT และ DATA OUTPUT LINE หมายถึงเมื่อหน่วยความจำทำการอ่าน. DATA I/O LINE จะถูกใช้เป็นเส้นทางที่นำข้อมูลออกจาก RAM เมื่อทำการเขียนจะเป็นเส้นทางที่นำข้อมูลเข้าสู่ RAM สภาวะลอจิกของเส้นควบคุม R/W เป็นตัวกำหนดสภาวะของขา DATA I/O ของหน่วยความจำ ในรูป 3.6 ได้แสดงขาและบล็อกไดอะแกรม สำหรับอุปกรณ์หน่วยความจำเบอร์ 2114.



รูปที่ 3.4 : แสดงบล็อกไดอะแกรมของ RAM ที่ใช้งานเขียนข้อมูลอันหนึ่งและอ่านอีกอันหนึ่งแยกจากกัน.



รูปที่ 3.5 : แสดงบล็อกไดอะแกรมของ RAM ที่ใช้งานข้อมูลเข้าและออกร่วมกัน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2114A 1024 X 4 BIT STATIC RAM

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Max. Access Time (ns)	100	120	150	200	200	250
Max. Current (mA)	40	40	40	40	70	70

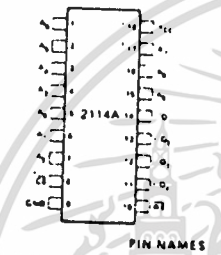
- HMOS Technology
- Low Power, High Speed
- Identical Cycle and Access Times
- Single +5V Supply ±10%
- High Density 18 Pin Package
- Completely Static Memory - No Clock or Timing Strobe Required
- Directly TTL Compatible: All Inputs and Outputs
- Common Data Input and Output Using Three-State Outputs
- 2114 Upgrade

The Intel® 2114A is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using HMOS a high performance MOS technology. It uses fully DC stable (static) circuitry throughout in both the array and the decoding therefore it requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

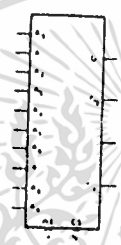
The 2114A is designed for memory applications where the high performance and high reliability of HMOS, low cost large bit storage, and simple interfacing are important design objectives. The 2114A is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select (CS) lead allows easy selection of an individual package when outputs are or-tied.

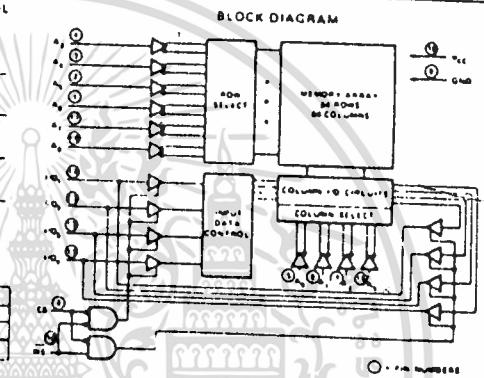
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0-A_9	ADDRESS INPUTS	V_{CC}	POWER +5V
$\overline{WE}$	WRITE ENABLE	GND	GROUND
$\overline{CS}$	CHIP SELECT		
I/O_1-I/O_4	DATA INPUT/OUTPUT		

รูปที่ 3.6 : บล็อกไดอะแกรม และ DATA SHEET บางส่วนของ 2114, 1K4, COMMON I/O STATIC RAM.

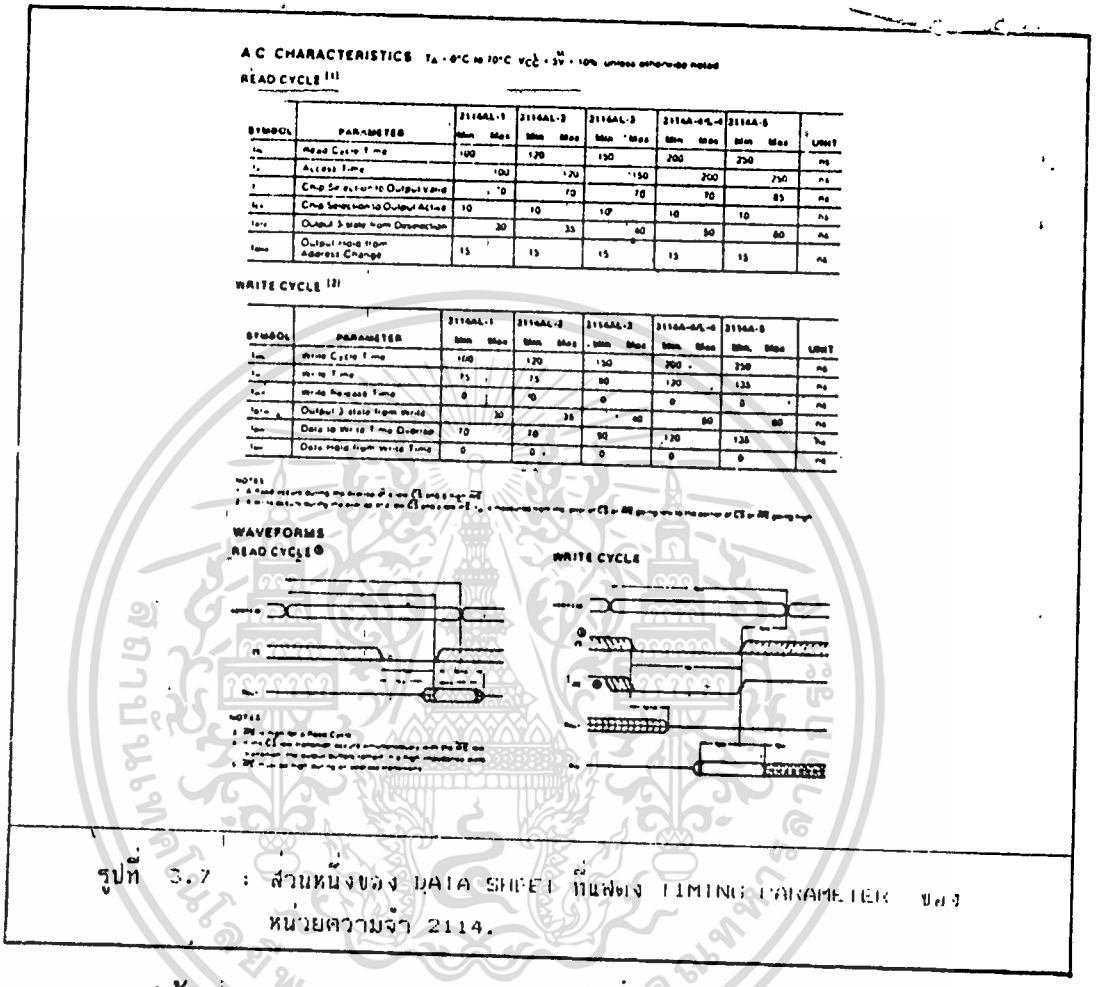
ต่อไปนี้เราจะกล่าวถึงสิ่งที่สำคัญของหน่วยความจำเบอร์นี้ (เรื่องที่อยู่ในย่อหน้าต่อไปนี้ อาจนำไปใช้งานกับหน่วยความจำสารกึ่งตัวนำส่วนมากได้ แม้ว่าหน่วยความจำนั้นไม่ได้เป็นแบบ COMMON I/O).

สำหรับตัวอย่างนี้ เราจะพิจารณาเฉพาะเบอร์ 2114 โดยสมมุติว่า MEMORY ACCESS TIME ของ 2114 มีค่าน้อยเมื่อเทียบกับคล็อกของระบบ และไม่มีความจำเป็นที่จะต้องสร้างสภาวะการคอย (WAIT STATES) ซึ่งสภาวะการคอยนี้จะใช้กับหน่วยความจำที่ทำงานช้า .

ในรูป 3.7 ได้แสดงส่วนหนึ่งของ DATA SHEET และไดอะแกรมเวลาสำหรับใช้งานกับหน่วยความจำเบอร์ 2114 ขั้นแรกมาดูการเขียนข้อมูลเข้าไปใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำก่อน ซึ่งเหตุการณ์ที่เกิดขึ้นเหล่านี้เป็นลำดับขั้นตอนทั่วๆไปที่ได้อธิบายไปแล้วในหัวข้อ 3.3 ของบทนี้ อย่างไรก็ตามในที่นี้เราจะอธิบายขบวนการที่นอกเหนือขึ้นไปอีก รวมทั้ง SPECIFICATION ของหน่วยความจำ.



A.C. CHARACTERISTICS

READ CYCLE [1]

SYMBOL	PARAMETER	2114AL-1 Min	2114AL-1 Max	2114AL-2 Min	2114AL-2 Max	2114AL-3 Min	2114AL-3 Max	2114A-4/L-4 Min	2114A-4/L-4 Max	2114A-6 Min	2114A-6 Max	UNIT
t_{RC}	Read Cycle Time	100		120		150		200		250		ns
t_A	Access Time		100		120		150		200		250	ns
t_{CO}	Chip Selection to Output Valid		70		70		70		70		85	ns
t_{CX}	Chip Selection to Output Active	10		10		10		10		10		ns
t_{OTD}	Output 3-state from Deselection		30		35		40		50		60	ns
t_{OHA}	Output Hold from Address Change	15		15		15		15		15		ns

WRITE CYCLE [2]

SYMBOL	PARAMETER	2114AL-1 Min	2114AL-1 Max	2114AL-2 Min	2114AL-2 Max	2114AL-3 Min	2114AL-3 Max	2114A-4/L-4 Min	2114A-4/L-4 Max	2114A-6 Min	2114A-6 Max	UNIT
t_{WC}	Write Cycle Time	100		120		150		200		250		ns
t_W	Write Time	75		75		90		120		135		ns
t_{WR}	Write Recovery Time	0		10		0		0		0		ns
t_{OTW}	Output 3-state from Write		30		35		40		50		60	ns
t_{DW}	Data to Write Time Overlap	70		70		90		120		135		ns
t_{DH}	Data Hold from Write Time	0		0		0		0		0		ns

WAVEFORMS

READ CYCLE

WRITE CYCLE

รูปที่ 3.7 : ส่วนหนึ่งของ DATA SHEET ที่แสดง TIMING PARAMETER ของหน่วยความจำ 2114.

3.5 ลำดับขั้นตอนการเขียนข้อมูลเข้าไปในตัว 2114.

1. สัญญาณของ ADDRESS INPUT LINE ต่างๆจะถูกกำหนดให้เป็นค่าของตำแหน่งที่ต้องการนำข้อมูลเข้าไปเขียน.

2. ข้อมูลที่ถูกส่งไปบน DATA I/O LINE ของ 2114 นั้น DATA I/O LINE จะมีลักษณะเป็นแบบ TRI-STATE ซึ่งถ้าไม่เป็นเช่นนี้จะเกิดการขัดกันระหว่างข้อมูลที่ป้อนให้กับข้อมูลที่เก็บไว้ภายในหน่วยความจำ ดังแสดงในรูป 3.8.

3. ต่อมาที่ขาอินพุท $\overline{WE}$ และ $\overline{CS}$ จะถูกทำให้ทำงานในเวลาเดียวกันโดยประมาณ เมื่อขาอินพุท $\overline{WE}$ เป็น สัญญาณ "0" สาย I/O จะทำหน้าที่เป็นสายนำข้อมูลเข้า ณ นั้นตัวหน่วยความจำจะไม่ถูกเขียนจนกว่าจะมีสัญญาณ "0" ป้อนเข้าที่ขาอินพุท $\overline{WE}$ ดังรูป 3.9 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

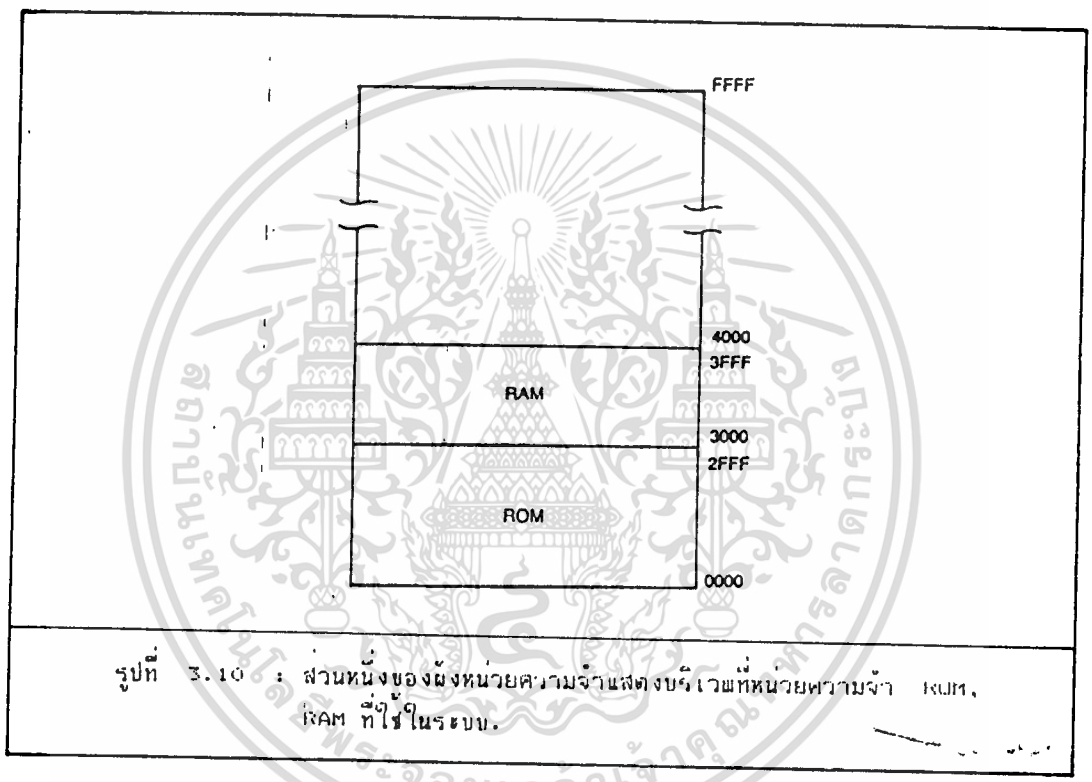
จะถูกใช้เป็นทางออกของข้อมูล.

3. DATA I/O LINE จะนำข้อมูลที่มีอยู่ในตำแหน่งนั้นออกมา.

ซึ่งจะเห็นว่าการอ่านข้อมูลจาก STATIC RAM มีการทำงานเหมือนกับการอ่านข้อมูลจาก ROM .

ต่อไปเราจะมาทำความเข้าใจว่า 2114 ทำงานอย่างไร เมื่อต่อเข้ากับไมโครโปรเซสเซอร์ Z80.

3.7 การต่อ ADDRESS LINE เข้ากับ Z80.



รูปที่ 3.10 : ส่วนหนึ่งของผังหน่วยความจำแสดงขอบเขตที่หน่วยความจำ ROM, RAM ที่ใช้ในระบบ.

เพื่ออธิบายว่า Z80 ติดต่อกับ RAM ได้อย่างไร ในที่นี้เราจะยกตัวอย่างระบบ RAM ที่ออกแบบมาอย่างสมบูรณ์แล้วสักระบบหนึ่ง เพื่อศึกษาระบบนี้ เราจะพิจารณาจากผังหน่วยความจำ (MEMORY MAP) เพื่อกำหนดตำแหน่งของข้อมูลใน RAM ว่าควรจะอยู่ส่วนไหนในช่วง 0000-FFFFH(0-64K) รูป 3.10 แสดงให้เห็นว่าหน่วยความจำชนิด RAM อยู่ในตำแหน่ง 3000 ถึง 3FFFH ในเลขฐานสิบหก ซึ่งจำนวนหน่วยความจำ RAM ในช่วงนี้มีขนาด 4096 หรือ 4Kx8 ตำแหน่ง.

สำหรับตัวอย่างนี้เราจะทำการสร้าง หน่วยความจำ 4Kx8 RAM เพื่อใช้

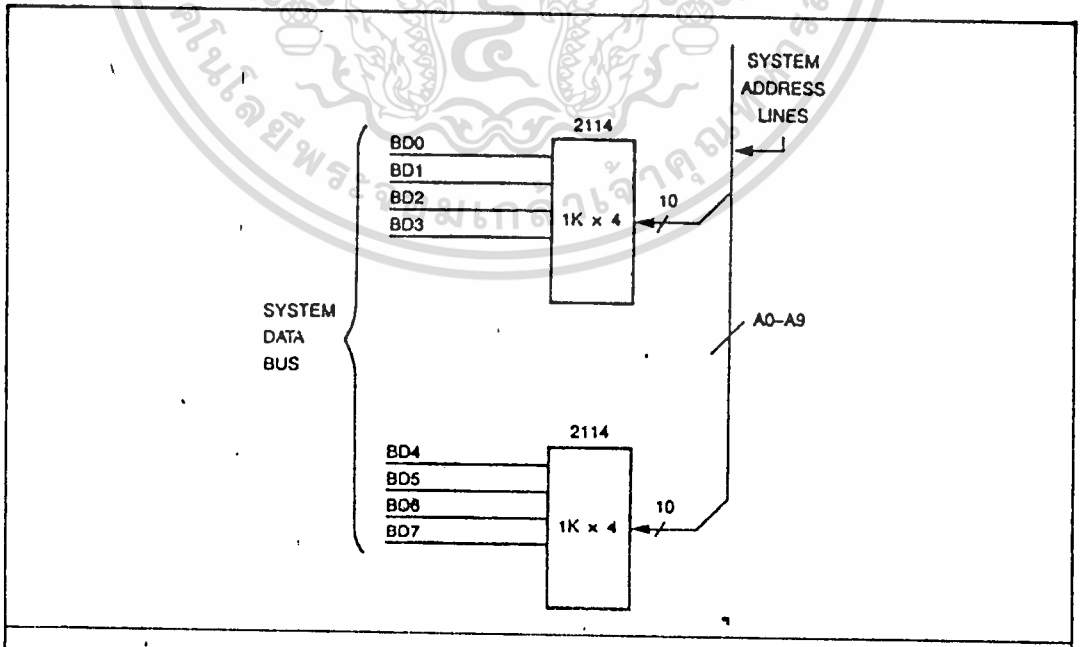
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ Z80 ถ้าดูจาก DATA SHEET ของ RAM 2114 ในรูป 3.6 แล้ว จะเห็นได้ว่าอุปกรณ์แต่ละตัวนี้เป็นแบบ 1Kx4 นั่นคือในการบ่งบอกตำแหน่งแต่ละครั้งจะได้ข้อมูลออกมาเพียง 4 บิท แต่ Z80 ใช้บัสข้อมูล 8 เส้นขนานกัน ฉะนั้นเมื่อเราใช้ RAM 2114 จึงจำเป็นต้องใช้สองตัวขนานกัน เพื่อที่จะสามารถส่งข้อมูลได้ครบ 8 บิท.

สิ่งแรกที่ต้องทำคือกำหนดจำนวนหน่วยความจำที่เป็น RAM ทั้งหมดเสียก่อน เพื่อนำไปคำนวณจำนวน RAM 2114 ที่ต้องใช้ ซึ่งสามารถคำนวณได้ดังนี้ เมื่อให้จำนวนหน่วยความจำทั้งหมดที่ใช้ RAM เท่ากับ 4096x8 หรือ 4096 ไบท์ หน่วยความจำเบอร์ 2114 แต่ละตัวเก็บได้ 1024x4 หรือ 1024 นิบเบิล (NIBBLES) ต้องใช้หน่วยความจำ 2114 จำนวน 2 ตัว สำหรับเก็บแต่ละ 1024 ไบท์ ดังนั้นเพื่อที่จะให้ได้ 4096 ไบท์ เราต้องใช้หน่วยความจำเบอร์ 2114 ถึง 8 ตัว ซึ่งแน่นอนเราไม่จำเป็นต้องการใช้จำนวนหน่วยความจำที่ออกแบบไว้ทั้งหมด.

สำหรับตัวอย่างนี้ ระบบ Z80 ของเราต้องการใช้ RAM แค่ 1K ไบท์ ของ RAM ทั้งหมด 4K ไบท์เท่านั้น.

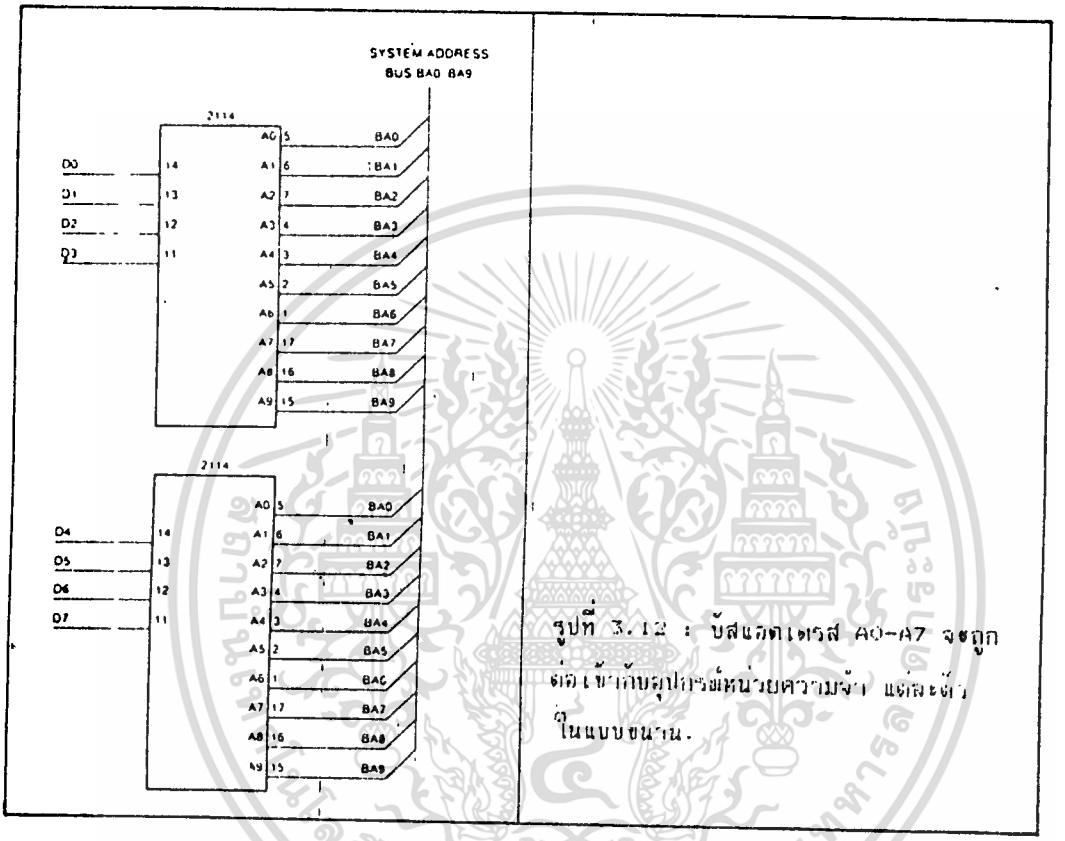
หมายเหตุ ทางที่ดีเราควรออกแบบระบบของเราให้มีตำแหน่งของหน่วยความจำให้มากกว่าที่เราจะใช้งาน เพื่อทำให้ง่ายสำหรับการขยายระบบในอนาคตเมื่อต้องการใช้ แต่ถ้าไม่ได้ทำดังนี้จะเป็นการยากอย่างยิ่งที่จะเพิ่มหน่วยความจำเข้าไปอีก.



รูปที่ 3.11 : อุปกรณ์หน่วยความจำ 2114 2 ตัวที่ต่อขนานกัน เพื่อให้ได้หน่วยความจำที่มีขนาด 8 บิท.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ขณะนี้เรารู้ว่าระบบ RAM ของเราต้องการหน่วยความจำเบอร์ 2114 จำนวนทั้งหมด 8 ตัว 2114 แต่ละตัวมี ADDRESS INPUT LINE 10 เส้น คือ A0-A9 และทั้งหมดต่อในลักษณะขนานกันไป นั่นคือ A0 ของ 2114 ทั้ง 8 ตัวจะต่อเข้าด้วยกัน, A1 ของ 2114 ทั้ง 8 ตัวก็ต่อเข้าด้วยกัน และเส้นอื่นๆก็เช่นเดียวกัน ดังรูป 3.12 .



รูปที่ 3.12 : บัสแอดเดรส A0-A7 จะถูกต่อเข้ากับอุปกรณ์หน่วยความจำ แต่ละตัวในแบบขนาน.

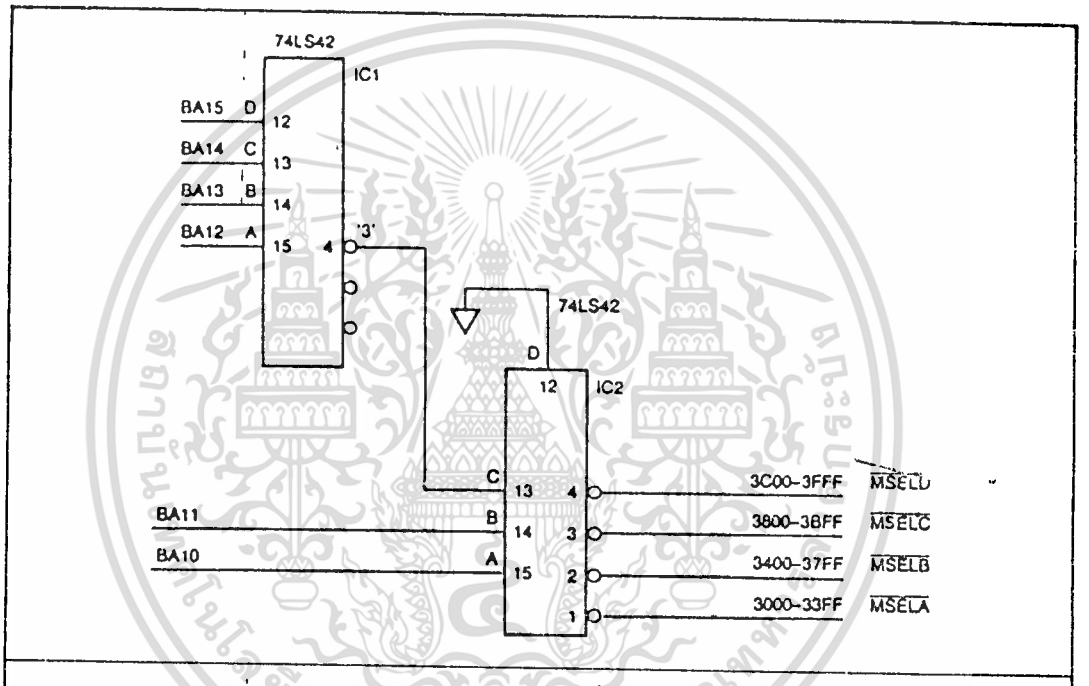
ส่วนบัสแอดเดรส A10-A15 จะถูกนำไปถอดรหัสเพื่อเลือกหน่วยความจำที่เป็น RAM นี้จากระบบหน่วยความจำอื่นๆ การถอดรหัสนี้จะมี 2 ระดับ คือสายแอดเดรส A10 และ A11 จะใช้ในการเลือกคู่ของ RAM 2114 ส่วน A12-A15 จะใช้เลือก หน่วยความจำนี้ RAM จาก C000-D000H ออกจากระบบหน่วยความจำอื่นๆทั้งหมด.

การถอดรหัสสายแอดเดรสนี้ ยังมีเทคนิคอีกหลายอย่างที่แตกต่างกันออกไป สำหรับวงจรที่แสดงในรูป 3.13 ก็เป็นอีกวิธีหนึ่งจากหลายๆวิธี ซึ่งทำให้เกิดแนวความคิดที่จะหาวิธีในการเลือกหน่วยความจำให้ได้ตามความต้องการ ถึงแม้ว่าเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำวงจรในรูป 3.13 ไปใช้งานได้ แต่อาจจะต้องนำไปปรับปรุงอีกซึ่งขึ้นอยู่กับระบบที่ใช้ โดยทั่วๆไปแล้วขึ้นกับความเร็วของระบบ , การโหลดของบัส (BUS LOADING EFFECT) และผังหน่วยความจำ ซึ่งในรูป 3.13 นี้ได้แสดงให้เห็นถึงอุปกรณ์ทางฮาร์ดแวร์ที่ใช้สำหรับการถอดรหัสแบบ 2 ระดับของหน่วยความจำชนิด RAM.

หมายเหตุ ในที่นี้อาจจะใช้บัฟเฟอร์สำหรับบัสแอดเดรสหรือไม่ก็ได้ การใช้บัฟเฟอร์ในบัสแอดเดรสได้อธิบายไปแล้วในบทที่ 2 ซึ่งสามารถใช้ได้ทั้ง RAM และ ROM.

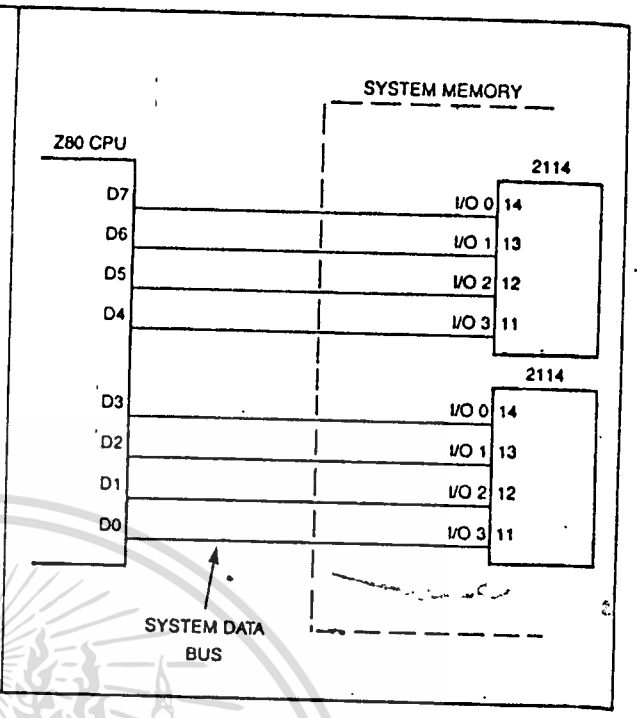


รูปที่ 3.13 : แสดงวงจรถอดรหัสที่ใช้ในการเลือกหน่วยความจำ.

3.8 การต่อบัสข้อมูลโดยไม่ใช้บัฟเฟอร์.

ต่อไปจะแสดงให้เห็นถึงการต่อบัสข้อมูลของหน่วยความจำเข้ากับบัสข้อมูลของ Z80 สำหรับในตัวอย่างนี้ เราจะไม่ใช้บัฟเฟอร์ โดยสมมุติว่าไม่มีการโหลดซึ่งกันและกันบนบัสข้อมูลของ Z80 และสามารถต่อ DATA I/O LINE ของหน่วยความจำ(เบอร์2114) กับบัสข้อมูลของ Z80 ได้โดยตรงโดยใช้การต่อแบบขนานดังรูป 3.14 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 :
แสดงการต่อบัสข้อมูลระหว่าง 2114 กับ Z80 CPU.

แต่สิ่งที่สำคัญก็คือ DATA I/O LINE ของหน่วยความจำเบอร์ 2114 จะทำงานได้ก็ต่อเมื่อมีสัญญาณควบคุมเข้ามาที่ขา $\overline{WE}$ และ $\overline{CS}$.

3.9 การสร้าง CONTROL LINE สำหรับอ่านและเขียนหน่วยความจำ.

เมื่อไมโครโปรเซสเซอร์ Z80 ทำการติดต่อกับหน่วยความจำ จะต้องมีสัญญาณจากสายควบคุมการอ่านคือ $\overline{MEMR}$ หรือ สัญญาณจากสายควบคุมการเขียนคือ $\overline{MEMW}$ ส่งออกไป โดยสัญญาณของสายควบคุมทั้งสอง ($\overline{MEMR}$ และ $\overline{MEMW}$) สามารถสร้างได้จากสัญญาณที่ได้จากสายควบคุมของ Z80 ($\overline{MREQ}$, $\overline{RD}$ และ $\overline{WR}$ ของ Z80. รูป 3.15 ได้แสดงวงจรที่ใช้สร้างสัญญาณ $\overline{MEMR}$ และ $\overline{MEMW}$ ซึ่งเป็นวิธีหนึ่งในการสร้างสัญญาณอ่านและเขียนหน่วยความจำ ขึ้นอยู่กับลอจิกที่ได้จากสายควบคุมทั้งสามของ Z80.

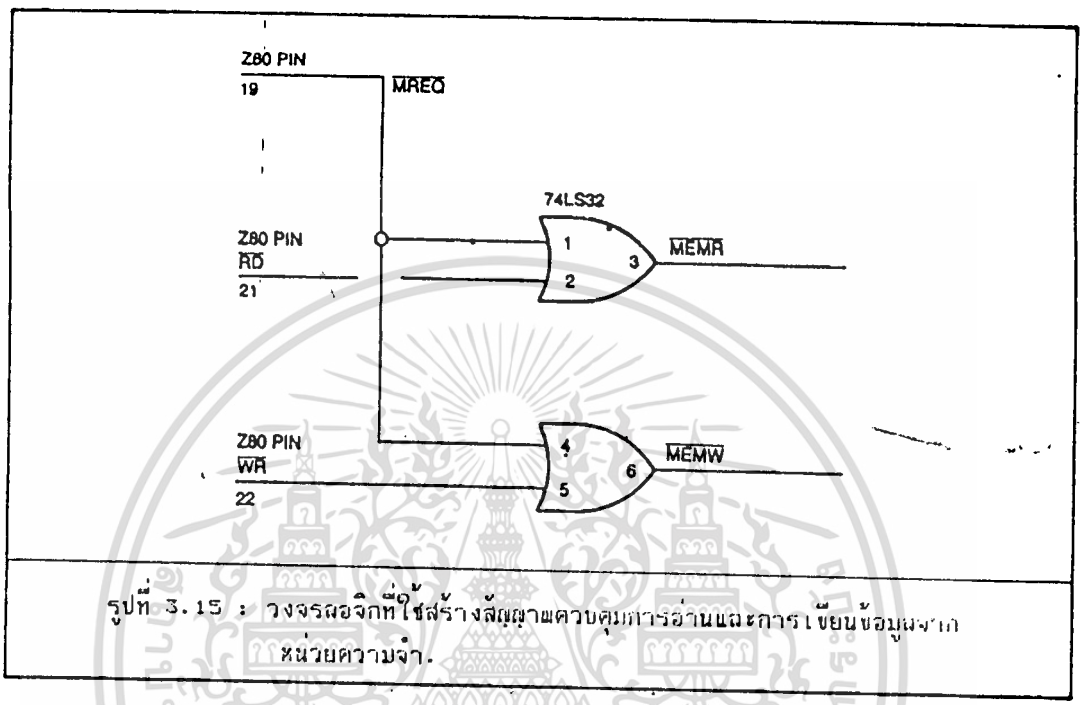
ต่อไปเราจะนำสายควบคุม $\overline{MEMR}$ กับ $\overline{MEMW}$ ที่ได้จากรูป 3.15 ไปใช้กับวงจรถอดรหัสแอดเดรสของหน่วยความจำ รูป 3.13 เพื่อที่จะนำไปควบคุมขา $\overline{WE}$ และ $\overline{CS}$ ของระบบ RAM.

หมายเหตุ ที่ได้อธิบายมานี้ไม่เพียงแต่จะใช้กับ หน่วยความจำเบอร์ 2114 เท่านั้น แต่สามารถใช้ได้กับหน่วยความจำชนิด STATIC เบอร์อื่นๆ ได้อีกด้วย.

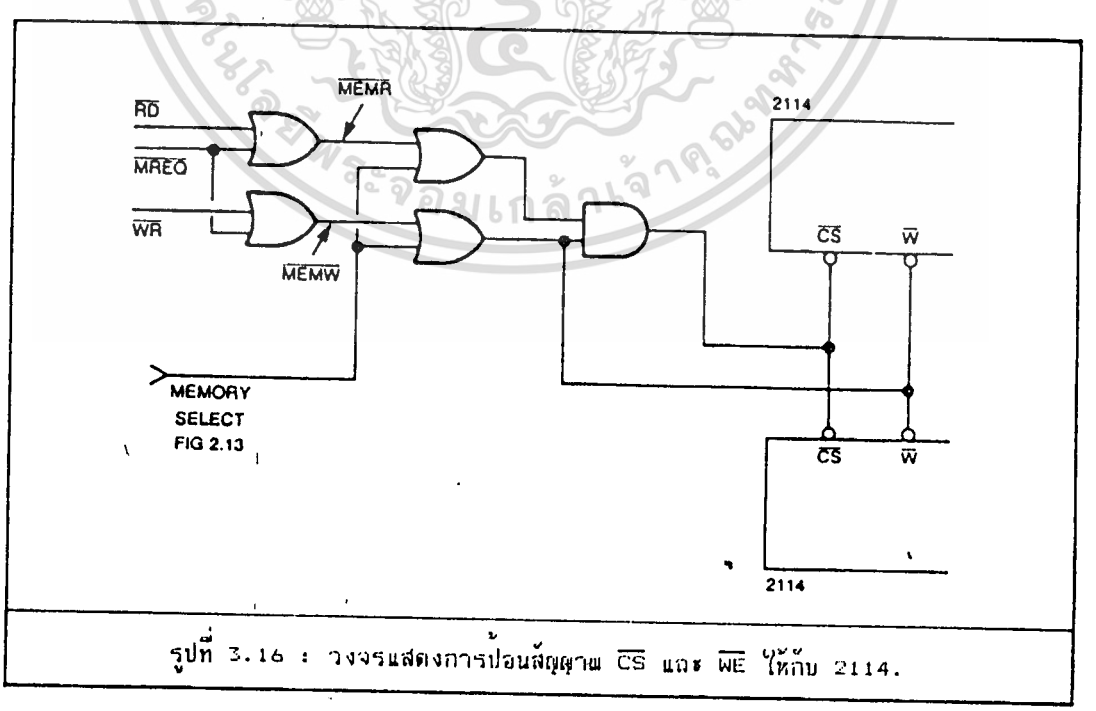
ในรูปที่ 3.16 แสดงถึงวิธีการต่อสายควบคุม $\overline{MEMR}$ และ $\overline{MEMW}$ กับ ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

อินพุท $\overline{WE}$ และ $\overline{CS}$ ของหน่วยความจำ 2114 ในการอ่านหน่วยความจำ วงจรนี้จะทำให้ขาอินพุท $\overline{WE}$ และ $\overline{CS}$ ทำงานในเวลาเดียวกัน(โดยประมาณ) ($\overline{WE}$ และ $\overline{CS}$ เป็นลอจิก "0") ส่วนในการเขียนนั้นขาอินพุท $\overline{CS}$ ของ 2114 จะทำงานเท่านั้น ($\overline{CS}$ มีลอจิก "0" $\overline{WE}$ มีลอจิก "1").



รูปที่ 3.15 : วงจรลอจิกที่ใช้สร้างสัญญาณควบคุมการอ่านและการเขียนข้อมูลจากหน่วยความจำ.



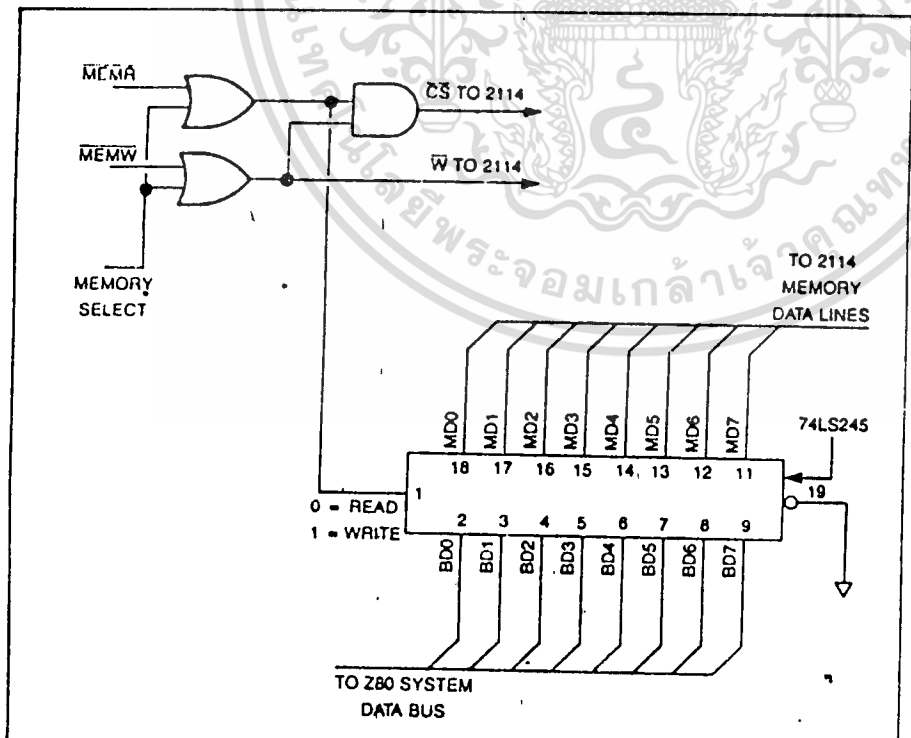
รูปที่ 3.16 : วงจรแสดงการป้อนสัญญาณ $\overline{CS}$ และ $\overline{WE}$ ให้กับ 2114.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การใช้บัฟเฟอร์บนบัสข้อมูลกับ STATIC RAM.

เราได้กล่าวถึงการต่อ DATA I/O LINE ของหน่วยความจำ 2114 กับบัสข้อมูล ของ Z80 โดยตรงมาบ้างแล้ว ซึ่งเป็นการต่อโดยไม่ใช้บัฟเฟอร์ ต่อไปจะมาดูกันว่าถ้าระบบของเราต้องการบัฟเฟอร์ การต่อจะเป็นอย่างไรโดยสมมุติว่าการใช้งานนี้ เราใช้บัฟเฟอร์ข้อมูลแบบ 2 ทิศทาง (BI-DIRECTIONAL DATA BUFFER) บนบัสข้อมูลของ Z80 ซึ่งก็คืออยู่บน DATA I/O LINE ของหน่วยความจำด้วย ในระหว่างที่หน่วยความจำถูกอ่านโดย Z80 นั้น ข้อมูลที่ได้ออกมาจากหน่วยความจำจะต้องขับโหลดที่อยู่บนบัสข้อมูลทั้งหมด อีกทั้งยังขับอินพุทของบัฟเฟอร์ข้อมูลแบบ 2 ทิศทางนั่นเอง.

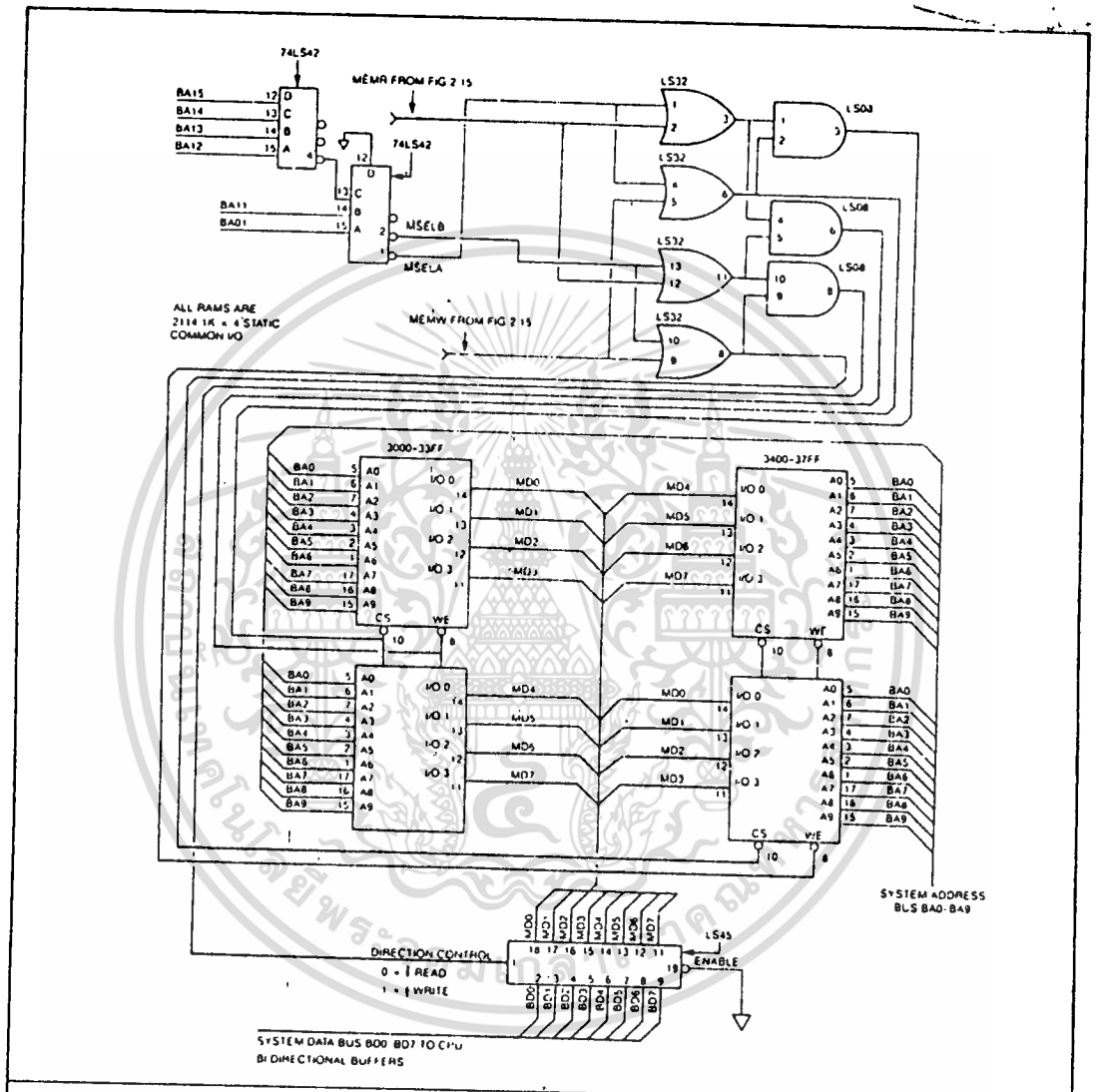
รูปที่ 3.17 แสดงให้เห็นวงจรหนึ่งในการต่อบัฟเฟอร์เข้ากับระบบ STATIC RAM จะสังเกตได้ว่า DATA I/O LINE ของหน่วยความจำ 2114 จะต่อเหมือนกับที่เราเคยรู้มาก่อน เว้นเสียแต่ไม่ได้ต่อกับ Z80 โดยตรง แต่จะต่อกับข้างหนึ่งของบัฟเฟอร์แบบสองทิศทางแทน การควบคุมทิศทางของบัฟเฟอร์นี้ขึ้นอยู่กับสถานะลอจิกของสัญญาณ $\overline{MEMR}$ เมื่อสัญญาณนี้มีลอจิก "0" และระบบ RAM นี้ถูกเลือกใช้งาน บัฟเฟอร์ข้อมูลจะให้ข้อมูลจาก RAM ออกไปสู่บัสข้อมูลของระบบ.



รูปที่ 3.17 : วงจรบางส่วนแสดงการใช้บัฟเฟอร์เบอร์ 74LS245 ซึ่งเป็นบัฟเฟอร์แบบสองทาง (BI-DIRECTIONAL BUFFER) เข้ากับ 2114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.18 นี้ เป็นระบบของ STATIC RAM ที่สมบูรณ์ที่จะใช้ร่วมกับไมโครโปรเซสเซอร์ Z80 จะสังเกตได้ว่าระบบนี้ไม่รวมเอาวงจรที่เพื่อรีเฟรชข้อมูลของหน่วยความจำเข้าไปด้วย ถ้าไม่ต้องการรีเฟรชข้อมูลก็สามารถต่อสาย MD ต่างๆ เข้ากับบัสข้อมูลของระบบได้โดยตรงเลย.



รูปที่ 3.18 : แสดงวงจรการต่อหน่วยความจำขนาด 2Kx8 เข้ากับ Z80 CPU.

3.12 หน่วยความจำ STATIC RAM เบอร์ 6116 .

ยังมีหน่วยความจำ STATIC RAM อื่นๆอีกที่สามารถนำมาใช้ได้โดยง่ายกับระบบการควบคุมของ Z80 ที่น่าสนใจก็คือ เบอร์ 6116 ซึ่งเป็นแบบ 2Kx8,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

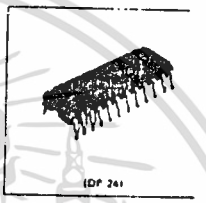
COMMON I/O STATIC RAM ในรูป 3.19 ซึ่งเป็นข้อมูลจำเพาะของ STATIC RAM เบอร์ 6116 เป็น STATIC RAM ที่มีคุณสมบัติในการทำงานคล้ายกันกับ 2114.

รูปที่ 3.19 : DATA SHEET ของ 6116 (ชนิด COMMON I/O).

HM6116P-2, HM6116P-3, HM6116P-4

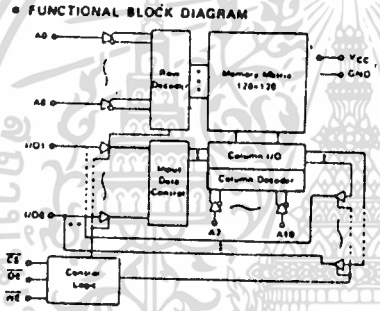
2048-word × 8-bit High Speed Static CMOS RAM

- FEATURES
  - Single 5V Supply and High Density 24 pin Package
  - High Speed: Fast Access Time 120ns/150ns/200ns (max.)
  - Low Power Standby and Low Power Operation: Standby 100μW (typ.), Operation 180mW (typ.)
  - Completely Static RAM: No clock or Timing Strobe Required
  - Directly TTL Compatible: All Input and Outputs
  - Pin Out Compatible with Standard 16K EPROM/MASK ROM
  - Equal Access and Cycle Time



DIP 24

- FUNCTIONAL BLOCK DIAGRAM



- PIN ARRANGEMENT

(Top View)

- ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Rating	Unit
Voltage on Any Pin Relative to GND	V_{IN}	−0.5 to +7.0	V
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Temperature Under Bias	T_{bias}	−10 to +85	°C
Power Dissipation	P_T	1.0	W

- TRUTH TABLE

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	Mode	V_{CC} Current	I/O Pin	Ref. Cycle
H	X	X	Not Selected	I_{SB}, I_{SB1}	High Z	
L	L	H	Read	I_{CC}	Dout	Read Cycle (1)~(3)
L	H	L	Write	I_{CC}	Din	Write Cycle (1)
L	L	L	Write	I_{CC}	Din	Write Cycle (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 DATA SHEET ของ 6116(ต่อ).

• RECOMMENDED DC OPERATING CONDITIONS (T_a = 0 to +70°C)

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}	4.5	5.0	5.5	V
	GND	0	0	0	V
Input Voltage	V_{IH}	2.2	3.5	6.0	V
	V_{IL}	-1.0*	-	0.8	V

* Pulse Width 50 ns, DC V_{IL} min = -0.3V

• DC AND OPERATING CHARACTERISTICS (V_{CC} = 5V ±10% GND = 0V T_a = 0 to +70°C)

Item	Symbol	Test Conditions	HM6116P-2			HM6116P-3/4			Unit
			min	typ*	max	min	typ*	max	
Input Leakage Current	I_{LI}	V_{CC} = 5.5V, V_{IN} = GND to V_{CC}	-	-	10	-	-	10	μA
Output Leakage Current	I_{LO}	$\overline{CS}$ = V_{IH} or $\overline{OE}$ = V_{IH}, $V_{I/O}$ = GND to V_{CC}	-	-	10	-	-	10	μA
Operating Power Supply Current	I_{CC}	$\overline{CS}$ = V_{IL}, $I_{I/O}$ = 0mA	-	40	80	-	35	70	mA
	I_{CC1}**	V_{IN} = 3.5V, V_{IL} = 0.6V, $I_{I/O}$ = 0mA	-	-	35	-	30	-	mA
Average Operating Current	I_{CC2}	Min cycle, duty = 100%	-	40	80	-	35	70	mA
Standby Power Supply Current	I_{SB}	$\overline{CS}$ = V_{IH}	-	5	15	-	5	15	mA
	I_{SB1}	$\overline{CS}$ ≥ V_{CC} -0.2V, V_{IN} ≥ V_{CC} -0.2V or V_{IN} ≤ 0.2V	-	0.02	2	-	0.02	2	mA
Output Voltage	V_{OL}	I_{OL} = 4mA	-	-	0.4	-	-	0.4	V
		I_{OL} = 2.1mA	-	-	-	-	-		
	V_{OH}	I_{OH} = -1.0mA	2.4	-	-	2.4	-	-	V

* V_{CC} = 5V T_a = 25°C
** Reference Only

• AC CHARACTERISTICS (V_{CC} = 5V ±10% T_a = 0 to +70°C)

• AC TEST CONDITIONS

Input Pulse Levels 0.8 to 2.4V
Input Rise and Fall Times: 10 ns
Input and Output Timing Reference Levels: 1.5V
Output Load: 1TTL Gate and C_L = 100pF* (*Including scope and Jig)

• READ CYCLE

Item	Symbol	HM6116P-2		HM6116P-3		HM6116P-4		Unit
		min	max	min	max	min	max	
Read Cycle Time	t_{RC}	120	-	150	-	200	-	ns
Address Access Time	t_{AA}	-	120	-	150	-	200	ns
Chip Select Access Time	t_{ACS}	-	120	-	150	-	200	ns
Chip Selection to Output in Low Z	t_{CLZ}	10	-	15	-	15	-	ns
Output Enable to Output Valid	t_{OE}	-	80	-	100	-	120	ns
Output Enable to Output in Low Z	t_{OLZ}	10	-	15	-	15	-	ns
Chip deselection to Output in High Z	t_{CHZ}	0	40	0	50	0	60	ns
Chip Disable to Output in High Z	t_{OHZ}	0	40	0	50	0	60	ns
Output Hold from Address Change	t_{OH}	10	-	15	-	15	-	ns

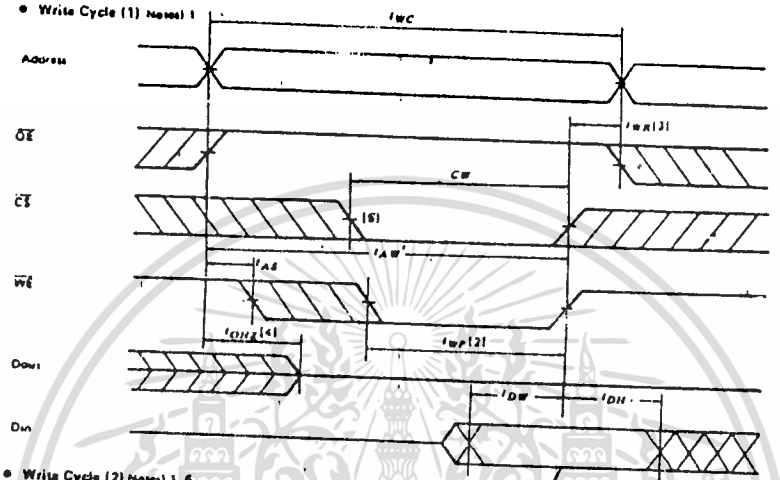
• WRITE CYCLE

Item	Symbol	HM6116P-2		HM6116P-3		HM6116P-4		Unit
		min	typ	min	max	min	max	
Write Cycle Time	t_{WC}	120	-	150	-	200	-	ns
Chip Selection to End of Write	t_{CW}	70	-	90	-	120	-	ns
Address Valid to End of Write	t_{AW}	100*	-	120	-	140	-	ns
Address Set Up Time	t_{AS}	20	-	20	-	20	-	ns
Write Pulse Width	t_{WP}	70	-	90	-	120	-	ns
Write Recovery Time	t_{WR}	5	-	10	-	10	-	ns
Output Disable to Output in High Z	t_{OHZ}	0	40	0	50	0	60	ns
Write to Output in High Z	t_{WHZ}	0	50	0	60	0	80	ns
Data to Write Time Overlap	t_{DW}	35	-	40	-	50	-	ns
Data Hold from Write Time	t_{DH}	5	-	10	-	10	-	ns
Output Active from End of Write	t_{OW}	5	-	10	-	10	-	ns

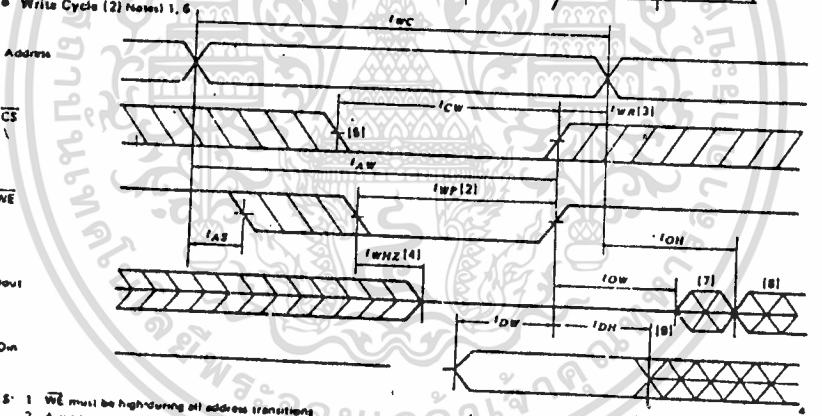
รูปที่ 3.19 DATA SHEET ของ 6116(ต่อ).

● TIMING WAVEFORM

● Write Cycle (1) Notes) 1



● Write Cycle (2) Notes) 1, 6



NOTES: 1. $\overline{WE}$ must be high during all address transitions.
2. A write occurs during the overlap (t_{WP}) of a low $\overline{CS}$ and a low $\overline{WE}$.
3. t_{WR} is measured from the earlier of $\overline{CS}$ or $\overline{WE}$ going high to the end of write cycle.
4. During this period, I/O pins are in the output state so that the input signals of opposite phase to the outputs must not be applied.
5. If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transitions or after the $\overline{WE}$ transition, output remain in a high impedance state.
6. $\overline{OE}$ is continuously low ($\overline{OE}$ = V_{IL}).
7. D_{out} is the same phase of write data of this write cycle.
8. D_{out} is the read data of next address.
9. If $\overline{CS}$ is Low during this period, I/O pins are in the output state. Then the data input signals of opposite phase to the outputs must not be applied to them.

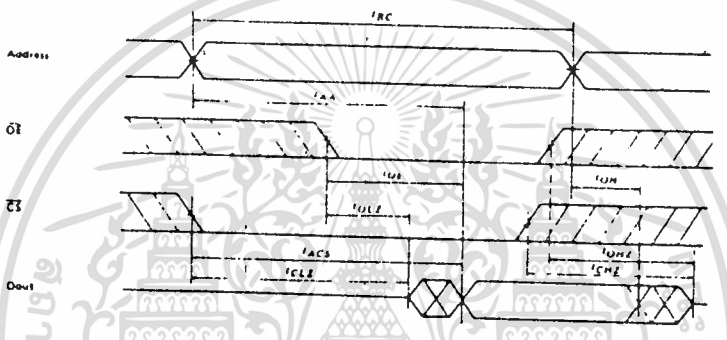
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 DATA SHEET ของ 6116(ต่อ).

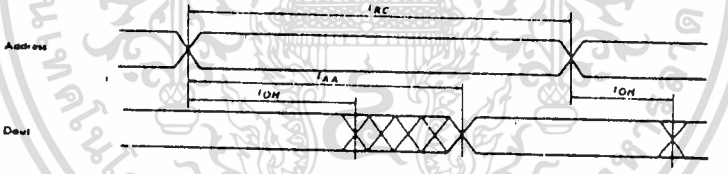
■ CAPACITANCE V = 1MHz T_a = 25°C

Item	Symbol	Test Conditions	typ	max	Unit
Input Capacitance	C_{in}	V_{in} = 0V	3	5	pF
Input/Output Capacitance	$C_{I/O}$	$V_{I/O}$ = 0V	5	7	pF

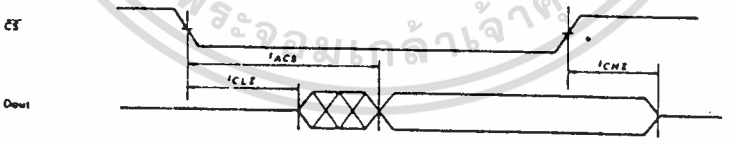
■ Read Cycle (1) Notes 1, 5



■ Read Cycle (2) Notes 1, 2, 4, 5



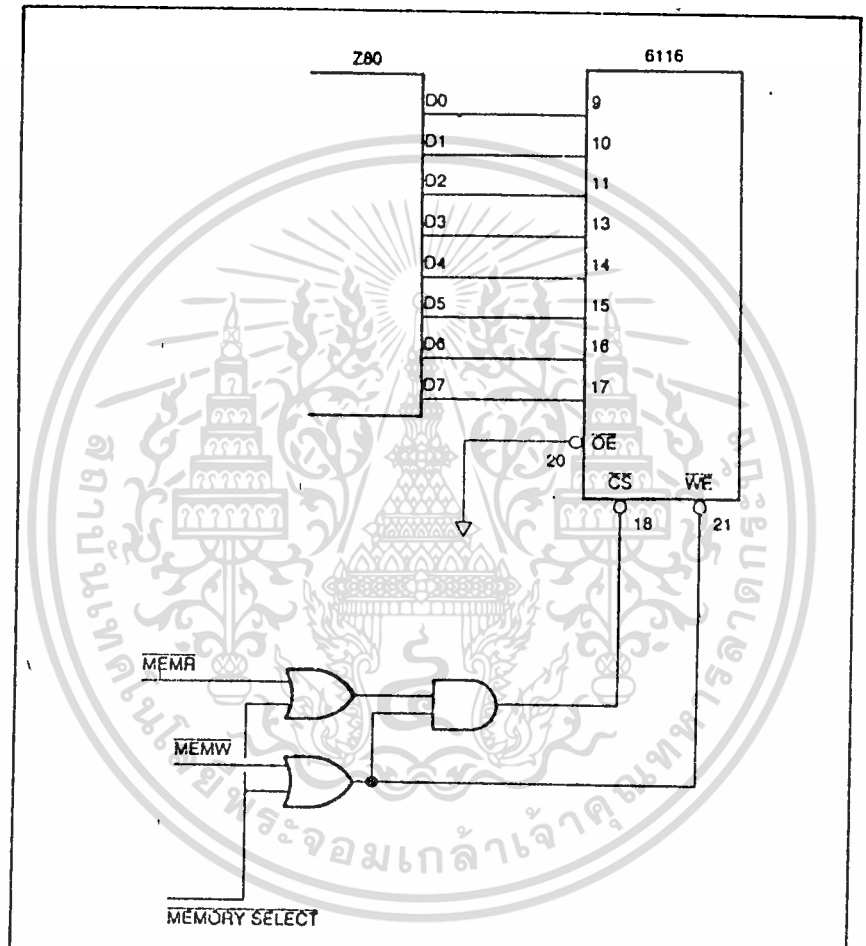
■ Read Cycle (3) Notes 1, 3, 4, 5



NOTES 1 $\overline{WE}$ is High for Read Cycle
2 Device is continuously selected $\overline{CS}$ = V_{IL}
3 Address Valid prior to or coincident with $\overline{CS}$ transition Low
4 $\overline{OE}$ = V_{IL}
5 When $\overline{CS}$ is Low, the address input must not be in the high impedance state

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.20 เป็นวงจรหนึ่งที่ใช้ในการต่อ 6116 RAM เข้ากับระบบควบคุมของ Z80 ซึ่งใช้ 6116 เพียงตัวเดียวก็สามารถทำให้เป็นระบบ RAM ที่สมบูรณ์ได้ (2114 ต้องใช้ถึง 2 ตัว) เนื่องจากมีโครงสร้างเป็นแบบ 2Kx8 จึงทำให้ 6116 RAM เหมาะสมกับการใช้งานกว่าไม่ว่าจะเป็นระบบใหญ่หรือเล็ก.



รูปที่ 3.20 : แผนผังแสดงการต่อ 6116 เข้ากับ Z80 CPU.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 4
เครื่องวัดความสูงด้วยอุลตร้าโซนิค

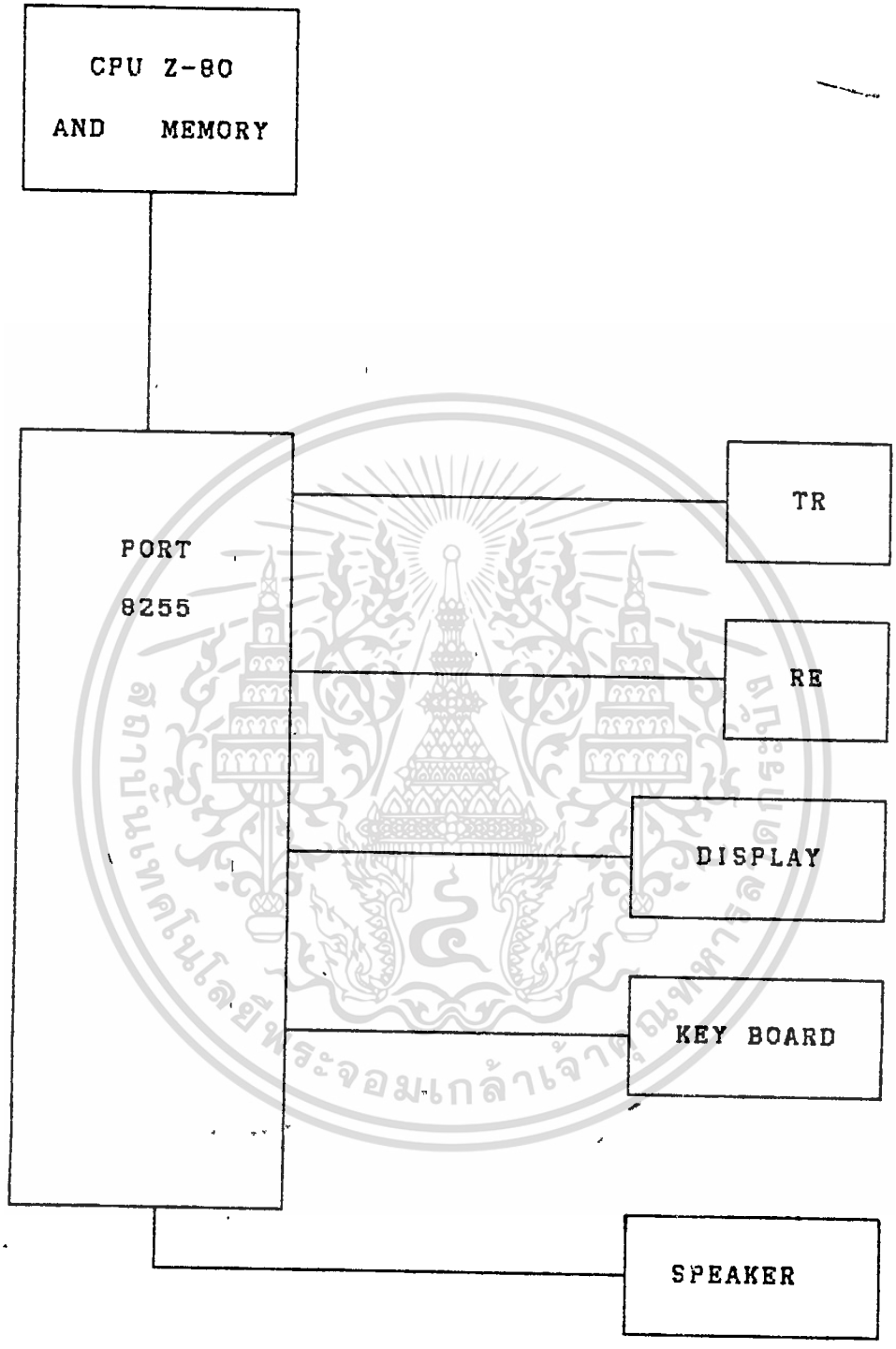
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ** คุณสมบัติและความสามารถของเครื่อง **

1. สามารถที่จะหา REFFERANCE ในการติดตั้งเครื่องได้เอง
2. วัดความสูงของคนทั่วไป
3. สามารถเก็บเป็นห้องได้ 31 ห้อง ห้องละ 98 คน รวม 3038 คน
4. สามารถเรียกดูความสูงของแต่ละคนในแต่ละห้องได้
5. สามารถบอกจำนวนคนที่สูงกว่าค่าที่กำหนดไว้ได้
6. สามารถทีจะ DETECT ความสูงที่ต้องการได้ อีกทั้งตั้ง LIMIT ความสูงไว้ได้ โดยใช้เสียงเป็นตัวบอก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

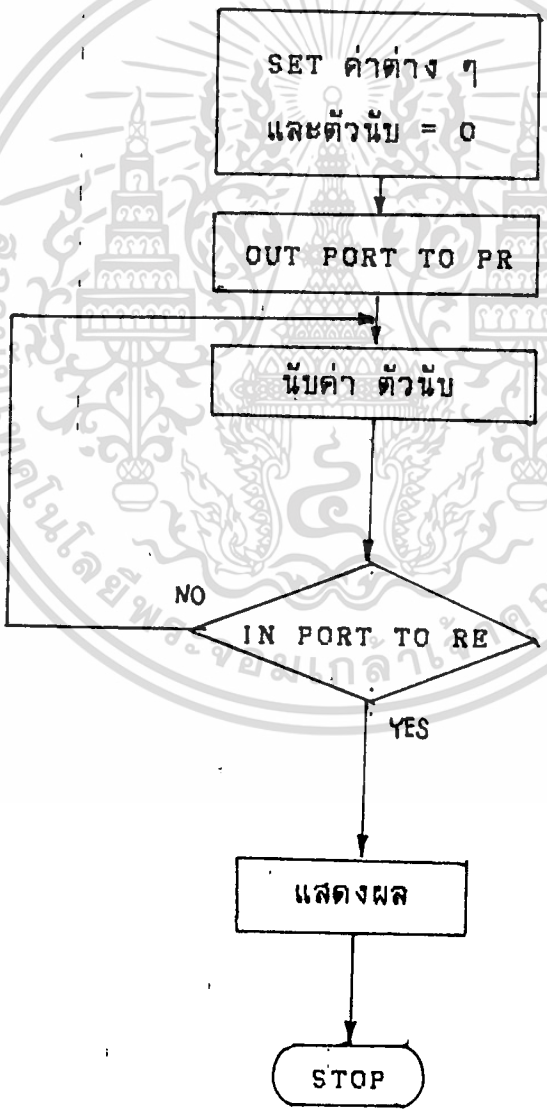


4.2 บล๊อกไดอะแกรมการทำงานของเครื่องวัดความสูงอุลตร้าโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวางแผนในการสร้าง

ในการสร้างเครื่องวัดความสูงด้วย ULTRASONIC ใช้ MICRO PROCESSOR เป็นตัวควบคุมการทำงานทั้งหมด ดังนั้นเราจำเป็นต้องเขียนโปรแกรมให้เครื่องปฏิบัติตามคำสั่ง เริ่มด้วยเกี่ยวกับคลื่น ULTTRASONIC ว่ามันสามารถได้จริงหรือไม่ โดยการรับและส่งโดบเครื่อง ULTRASONIC ทดลอง เมื่อแน่ใจว่าทำงานได้ก็นำมาต่อเข้ากับชุด SINGLE BOARD (ET-BOARD, VERSION 3.0) มาทำการทดสอบ โดยสร้างชุดรับและชุดส่ง ULTRASONIC ขึ้นมาและทำการ INTERFACE เข้าด้วยกัน โดยเขียน FLOW CHART ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก FLOW CHART จะเห็นได้ว่าให้ MICRO PROCESSOR ส่งออกไปควบคุมที่ PORT OUT TR และคอยรับ PORT IN เข้ามาในระหว่างที่รอก็ทำการนับจำนวนตัวนับไปด้วย เมื่อคลื่น ULTRASONIC ส่งออกไปและสะท้อนกลับมาที่ตัวรับ (RE) และส่งเข้า PORT IN , MICRO PROCESSOR จะทำการหยุดนับค่าและออกมาจาก LOOP ให้แสดงผลออกมา ผลที่แสดงนั้นเป็นเลขฐาน 16 ที่ไม่สามารถจะรู้ค่าจริง ๆ ได้ ซึ่งเป็นการทดสอบความเป็นไปได้ของเครื่องวัดความสูงนี้

เมื่อมีความเป็นไปได้เช่นนั้นแล้ว ก็ลงมือสร้างชุดควบคุมโดยใช้ MICRO PROCESSOR มาควบคุมได้เลย โดยที่เราจำเป็นต้องเขียนโปรแกรมการควบคุมทั้งหมดก็เนื่องมาจากชุดเราสร้างขึ้นไม่มี SOFT WARE ที่มาจัดการกับชุดของเครื่องนี้ โดยอาศัยชุด SINGLE BOARD มาเขียนโปรแกรมใส่ลงไปใน RAM PACK (เป็น RAM ธรรมดาที่นำเอา BATTERY มาติดเพื่อไม่ให้โปรแกรมหายไปขณะที่ถอดออกจาก SINGLE BOARD ไปใส่ในชุดควบคุมที่เราสร้างขึ้น) โดยเราจะเขียนโปรแกรมเข้าไปที่ละส่วน ๆ และทำการทดสอบว่าทำงานได้จริงหรือไม่ (มีวิธีการอีกหลายวิธีการที่จะเขียนโปรแกรม เช่น เขียนโปรแกรมลงบนเครื่อง PC และทำการทอสอบและแก้ไขบนเครื่อง PC เมื่อแน่ใจก็ทำการโหลดข้อมูลลงใน RAM PACK หรือ EPROM EMOLATOR นำไปใส่ในเครื่องควบคุมก็ได้เช่นกัน) โดยการทำในลักษณะนี้ไปเรื่อย ๆ จนเสร็จทุกอย่างที่วางแผนได้ก็เสร็จสิ้น

ในการหาค่าระยะทางจริง ๆ เราทำการหาค่าหลาย ๆ ค่าต่อ 10 CM ว่ามีค่าเท่าไร โดยใช้เครื่องเป็นตัวทดสอบแล้วนำมาหาค่าหลาย ๆ ค่ามาทำการหาค่าเฉลี่ยออกมาค่าหนึ่ง เมื่อได้ค่าเฉลี่ยแล้วในระยะ 10 CM เราก็หาค่าระยะ 1 CM ว่ามีค่าเท่าไหร่ โดยใช้วิธีการคำนวณทางคณิตศาสตร์ธรรมดา ค่าระยะ 1 CM นี้จะมีค่าจากการนับอยู่ค่าหนึ่งในที่นี้ก็คือการเพิ่มค่า REGISTOR ขึ้นไปนั่นเอง (REGISTOR HL)

สมมุติว่าจากการนับ REGISTOR ไปเรื่อย ๆ ขณะที่ส่งไป เมื่อคลื่น ULTRASONIC สะท้อนกลับมาถึงตัวรับ ก็จะถูก CHECK โดย MICRO PROCESSOR ตลอดเวลา เมื่อมันสะท้อนกลับมา MICRO PROCESSOR จะทำการหยุดนับและนำค่า REGISTOR ที่นั่นมาทำการคำนวณว่ามีค่าเท่าไหร่ แล้วนำไปเปรียบเทียบกับค่าที่นั้นได้ต่อ 1 CM ก็จะได้ค่าออกมา จากนั้นก็ทำการแปลงเป็นเลขฐานสิบแสดงผลให้เราทราบได้

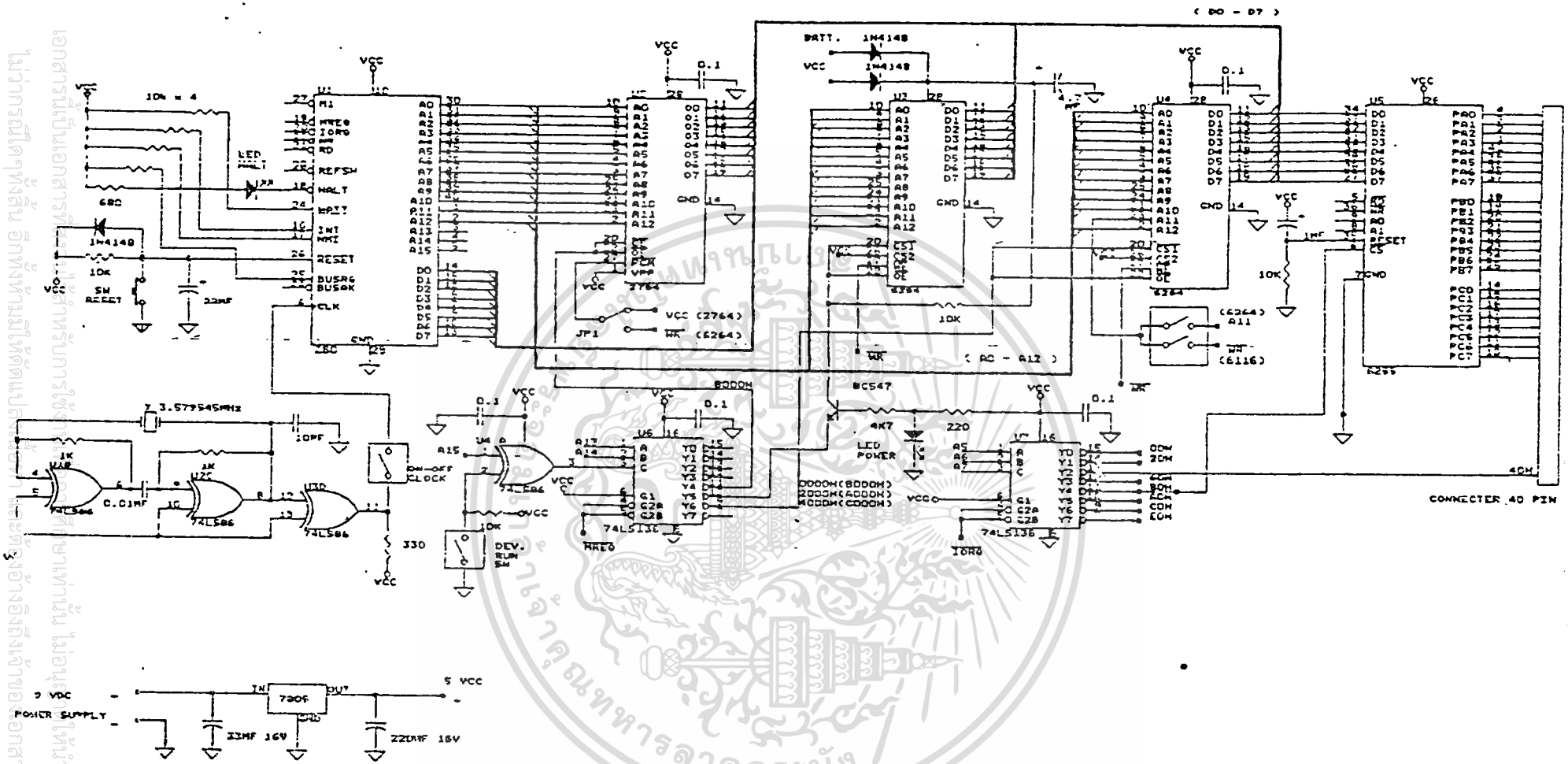
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTOR จะนับได้ไม่เท่ากันก็เนื่องมาจากระยะทางที่คลื่น ULTRASONIC สะท้อนไปและสะท้อนกลับจะมีระยะทางไม่เท่ากัน ซึ่งเป็นพื้นฐานในการสร้างชุดนี้ขึ้นมา

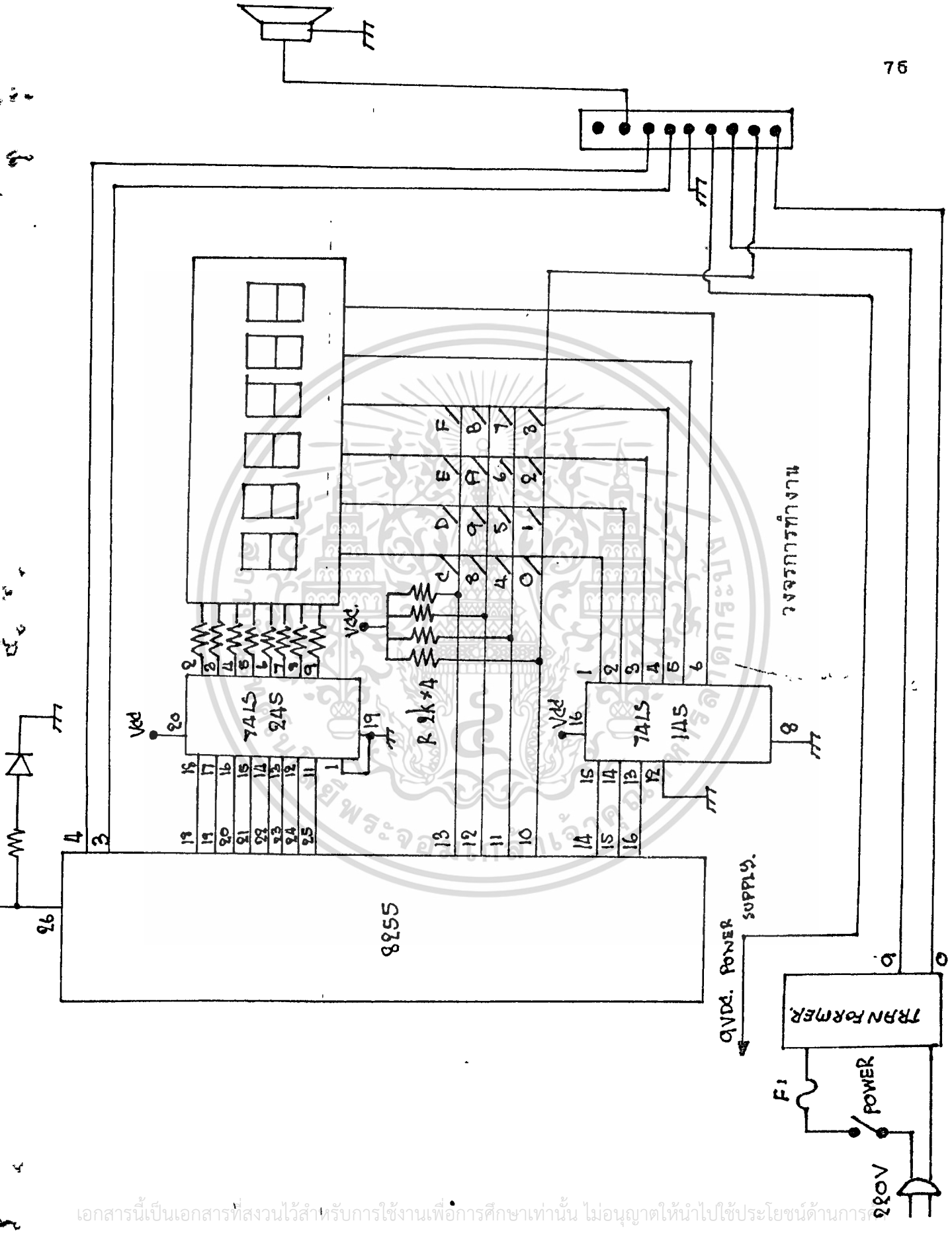
* หมายเหตุ CYTAL ที่ใช้สร้างเป็น CLOCK ให้กับ CPU นั้น จะมีผลกับการนับมาก เกี่ยวกับความละเอียด ถ้าใช้ CLOCK ที่ความถี่ต่ำ ๆ แล้วจะมีความละเอียดในการวัดผิดพลาดมากกว่าการใช้ CLOCK ที่มีความถี่สูงกว่า และถ้านำโปรแกรมที่เขียนแล้วชุดเดียวกันไปใช้กับอีกเครื่องหนึ่ง ที่มี CYTAL ไม่เท่ากัน ก็จะผิดพลาดได้ง่ายเช่นกัน เมื่อเราเขียนโปรแกรมและจัดเครื่องต่าง ๆ เรียบร้อยแล้ว และทำการรันหาความผิดพลาดจนไม่พบสิ่งผิดพลาดแล้ว โดยที่โปรแกรมของเราทั้งหมดจะอยู่ในส่วนของ RAM PACK ทั้งหมด จากนั้นเราจะทำการเอาข้อมูลที่อยู่ใน RAM PACK มาทำโปรแกรมลงใน EPROM และนำ EPROM ไปใช้กับเครื่องของเราได้เลย โดยไม่อาศัย RAM PACK อีกต่อไป

ในการเขียนโปรแกรม เราก็จะใช้ SINGLE BOARD มาเขียน โดยเขียนไปทีละลำดับก่อนว่ามีอะไรบ้าง ซึ่งจะดูได้ในส่วนของโปรแกรมที่เขียนขึ้น ขอแนะนำให้เขียนไว้ให้เป็นส่วนย่อยไว้ก่อน แล้วใช้วิธีการเรียกไป (CALL) ซึ่งจะสะดวกกว่าที่จะเขียนโปรแกรมรวมไปเลย

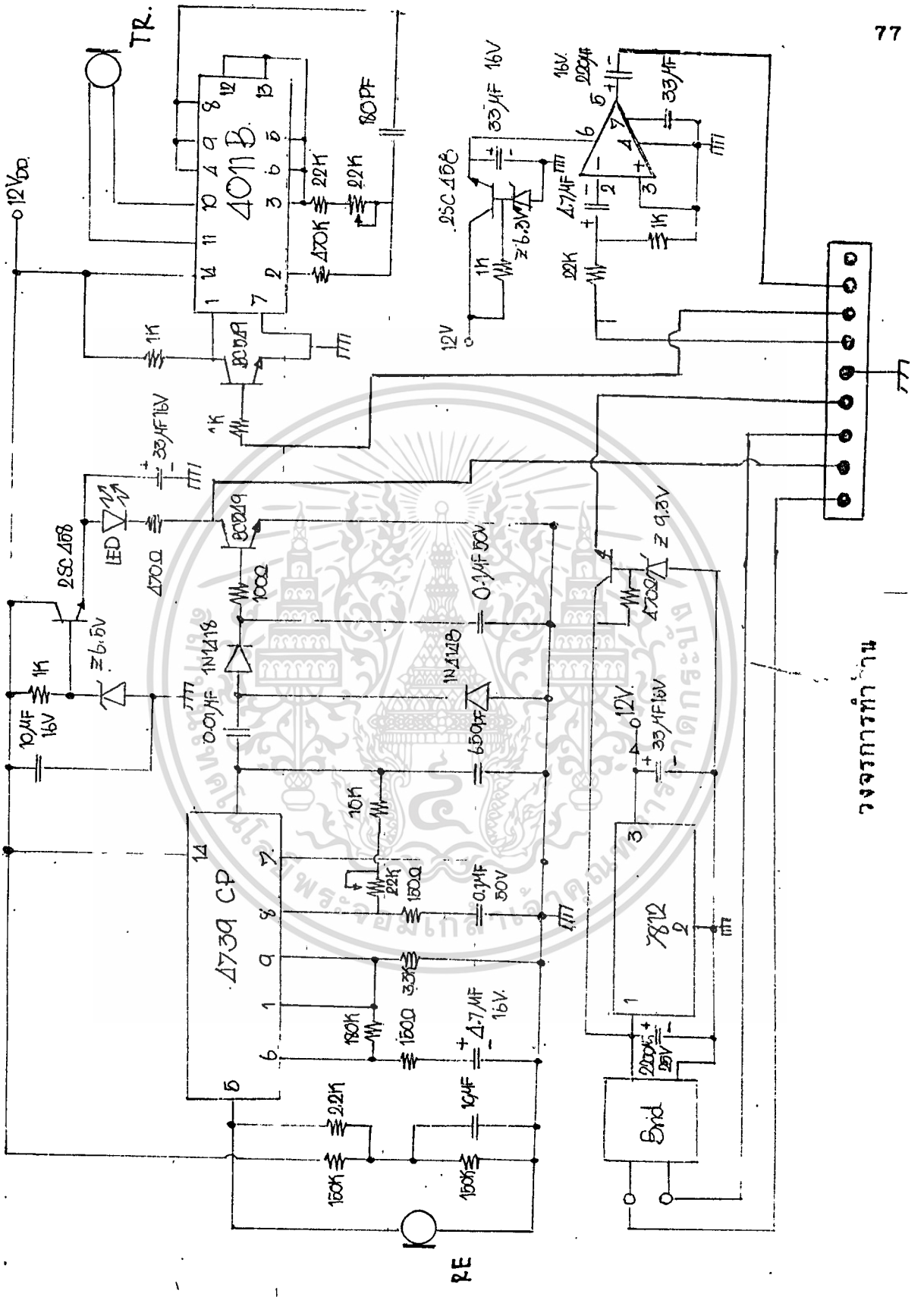
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4.4 วงจรการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทำงานของเครื่องวัดความสูงด้วยอุลตร้าโซนิค

เริ่มด้วยนำเครื่องไปติดตั้งกำแพง ให้สูงจากพื้นประมาณ 185 - 195 cm เมื่อติดตั้งเรียบร้อยแล้วก็ทำการ ON SWITCH POWER เครื่องจะแสดงคำว่า 2L.2533 ซึ่งแสดงรุ่นของนักศึกษาและปีการศึกษาตามลำดับ จากนั้นจะแสดงอยู่ครู่หนึ่ง ต่อไปจะแสดงคำว่า " SET REF" (SET REFFERANCE) ซึ่งจะพร้อมที่จะทำการวัดเพื่อหาจุดอ้างอิง เมื่อเรานำไปติดตั้ง ค่า REFFERANCE นี้เป็นค่าที่เก็บในเครื่องตลอดไป จนกว่าจะ OFF POWER ของเครื่องเท่านั้น

จากนั้นก็กด KEY "C" ซึ่งเป็น ENTER เพื่อเก็บค่า REFFERACE ไว้ เมื่อกดแล้ว เครื่องจะทำการหาค่าเอง โดยจะ DISPLAY ให้เห็น ตัวเลขที่แสดงอาจผิดพลาดไปเนื่องมาจากตัวแปรหลายอย่าง เช่น ทิศทางของคลื่น ULTRASONIC ไม่สามารถจะสะท้อนกลับมาได้ เสียงรบกวนจากภายนอก การติดตั้งที่ผิดตำแหน่งไปเมื่อทำการวัดถูกต้องแล้วเครื่องจะแสดงตัวเลขอยู่กับที่ให้เห็นเป็นระยะทางจากเครื่องถึงพื้น ต่อไปทำการกด KEY " D " เมื่อทำการกดแล้วเครื่องก็จะเก็บค่าที่วัดได้ในเครื่องเป็นที่เรียบร้อย และแสดงคำว่า " READY " ต่อไปทำการเลือกห้องว่าจะอยู่ห้องใด โดยการกด KEY " A " เครื่องจะแสดงคำว่า" ROOM.XX "(ROOM หมายถึง ห้อง, XX หมายถึง หมายเลขห้อง) จากนั้นเราก็กดหมายเลขห้องที่ต้องการ โดยมีตั้งแต่ห้อง 0 - 31 ห้อง

ทดลองกดเลข "0" ลงไป ขณะนี้เครื่องจะอยู่ที่ห้องหมายเลข "0" (ในห้องนี้จะแตกต่างจากห้องอื่นตรงที่ว่าไม่สามารถที่จะ RECORD ได้) จากนั้นให้นำคนที่จะวัดมาทำการวัดโดยยืนให้ตรงกับตำแหน่งเครื่องวัด และทำการวัดโดยการกด KEY "C" เครื่องจะแสดงคำว่า "READY" และให้ทำการกด KEY "C" อีกครั้ง ก็จะได้ส่วนสูงของคน ๆ นั้นออกมา ให้นำคนต่อไปมาทำการวัด แล้วกด KEY "C" อีก ก็จะได้ส่วนสูงของคนที่สอง โดยทำในลักษณะเดียวกันไปเรื่อย ๆ

ทดลองกด KEY "A" ใหม่ เลือกห้องที่ "1" กด KEY "C" สองครั้งก็สามารถที่จะวัดได้เช่นกัน คราวนี้ทดลองเลือกห้องที่ "32" เมื่อกด KEY "C" เครื่องจะแสดงคำว่า " ERROR " ให้เห็น แสดงว่าเกินห้องที่มีอยู่คือ 31 ห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่ทดลองผ่านมาเป็นการวัดส่วนสูงโดยไม่มีการ RECORD ไว้ ไม่ว่าจะเลือกห้องไหนตั้งแต่ห้อง 0 ถึงห้องที่ 31 ก็จะมีลักษณะเดียวกันหมด ต่อไปเราจะทำการวัดโดยเก็บ DATA ไว้ด้วย เริ่มด้วยการกด KEY "A" เพื่อทำการเลือกห้อง (ห้องที่ 0 ไม่สามารถที่เก็บ DATA ได้)

ทดลองเลือกห้อง "0" ดู

```
"A"     =   ROOM.XX
00      =   ROOM.00  <---   เลือกห้องที่ "00"
"C"     =   READY
"D"     =   REC
"C"     =   ERROR
```

เครื่องจะแสดงคำว่า "ERROR" ให้เห็นแสดงว่าไม่สามารถเก็บได้

ทดลองเลือกห้องที่ "1" ดูบ้าง

```
"A"     =   ROOM.XX
01      =   ROOM.01   เลือกห้องที่ "1"
"C"     =   READY
"D"     =   REC
"C"     =   01    <---   คนที่หนึ่งจะทำการวัด
"C"     =   ____  <---   แสดงค่าที่วัดได้
"C"     =   02    <---   คนที่สองจะทำการวัด
"C"     =   ____  <---   แสดงค่าที่วัดได้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด "C" เรื่อย ๆ จนครบ 98 คน ก็จะเต็มหนึ่งห้องเมื่อเรากดต่อไปจะแสดงคำว่า " FUUL " ให้เห็นและแสดงคำว่า "READY" ซึ่งพร้อมที่ทำงานต่อไป ถ้าต้องการวัดห้องอื่นก็ทำในลักษณะเดียวกันโดยการกด "A" เพื่อทำการเลือกห้อง

ต่อไปเราจะดูค่าที่เราวัดได้ KEY "B" เป็นการเรียกดูค่าที่เราวัดได้โดยตามขั้นตอนดั้งนี้

```
"A"     =   ROOM.XX    <--- เลือกดูห้องที่จะดู
01      =   ROOM.01    <--- ดูห้องที่ "1"
"C"     =   READY      <--- พร้อมที่จะดู
"B"     =   ST.NO.01   <--- คนที่ "1"ของห้องที่ "1"
"C"     =   01-XXX     <--- ค่าที่เรียกดูคนที่ "1"
"C"     =   02-XXX     <--- ค่าที่เรียกดูคนที่ "2"
 .
 .
 .
"C"     =   98-XXX     <--- ค่าที่เรียกดูคนที่ "98"
"C"     =   01-XXX     <--- จะกลับมาคนที่ "1" ใหม่
```

จากส่วนที่ผ่านมาเป็นการเรียกดูตามลำดับต่อ ๆ กันมา เราสามารถที่จะเรียกดูคนที่เท่าไรก็ได้ โดยขั้นตอนดั้งนี้

```
"A"     =   ROOM.XX
01      =   ROOM.01
"C"     =   READY
"B"     =   ST.NO.XX
15      =   ST.NO.15     <--- เรียกดูคนที่ "15"
"C"     =   15-XXX       <--- แสดงคนที่ "15"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทำนองเดียวกันก็สามารถที่จะเรียกดูคนที่เท่าไรก็ได้ตั้งแต่คนที่ 1 ถึงคนที่ 98 ถ้ากดเกินกว่านี้เครื่องจะแสดงคำว่า "ERROR" ออกมา ในการเรียกดูนี้ถ้าห้องใดที่ยังไม่ได้เก็บข้อมูล เมื่อเราเรียกดูเครื่องจะแสดงคำว่า " NO.DATA " ซึ่งไม่มีข้อมูลในห้องนี้

ในการค้นหาข้อมูลที่มีค่ามากกว่าค่าที่เราตั้งไว้ ว่ามีจำนวนเท่าไรเราสามารถให้เครื่องแสดงให้เราทราบได้ ด้วยการใช้ KEY "E" ซึ่งเป็น FUNCTION หาข้อมูลที่เก็บที่เก็บไว้ไวนเครื่อง เมื่อเราเก็บข้อมูลไว้ในเครื่องเรียบร้อยแล้ว เราอยากทราบว่าในข้อมูลมีความสูงที่มากกว่าค่าที่เราตั้งไว้มีกี่คนเราทำได้ดังนี้ โดยที่เราจะเรียกดูในห้องที่ "1" สมมุติให้ห้องที่ "1" มีข้อมูล ตั้งแต่คนที่ 1 ถึง 6 มีข้อมูลดังนี้

คนที่	ข้อมูล
1	5
2	12
3	7
4	12
5	13
6	6

```
"READY"

"A"     =   ROOM.XX
 1      =   ROOM.01
"C"     =   READY
"E"     =   [ 000
"E"     =   ] 000
 5      =   ] 005        <--- ตั้งค่าสูงกว่า 5 มีเท่าไร
"C"     =   ST.NO.06     <--- มีจำนวนมากกว่า 6 คน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"E"     =   J 000
10      =   J 010       <--- ตั้งค่าสูงกว่า 10 มีเท่าไร
"C"     =   ST.NO.03    <--- มีจำนวนมากกว่า 3 คน
"E"     =   J 000
20      =   J 020       <--- ตั้งค่าสูงกว่า 20 มีเท่าไร
"C"     =   ST.NO.00    <--- มีจำนวนมากกว่า 0 คน
```

ในทำนองเดียวกันถ้าต้องการจะเปรียบเทียบกับห้องอื่นแล้วก็ทำเหมือนกันโดยการเลือกห้องก่อน

เมื่อเราต้องจะทดสอบว่าบุคคลที่จะมาวัดมีความสูงตามที่เราหรือไม่เช่นต้องการสูงกว่า 160 cm หรือต่ำกว่า 150 cm ในเครื่องนี้สามารถทดสอบได้โดยจะแสดงผลโดยใช้เสียงบอกให้เราทราบได้ KEY "F" เป็น FUNCTION ที่ทำหน้าที่นี้

โดยสามารถที่จะตั้งค่าให้มากหรือน้อยกว่า เริ่มต้นด้วยการ SET REFFERANCE ก่อน (เหมือนกับการหาค่าความสูง)

สมมุติให้ทดสอบหาค่าที่น้อยกว่า 10

```
" READY "
          "F"     =   [ 000     เป็นการตั้งค่าที่น้อยกว่า
          10      =   010
          "C"     =   READY
          "C"     =   004       เสียงจะเป็นเสียง ติ๊ก ต่อก ออกมา
```

แสดงว่ามีค่าน้อยกว่าค่าที่เรากำหนดไว้

```
          "C"     =   002       เสียงเหมือนเดิม
          "C"     =   014       เสียงจะเป็นเสียง ปี๊ป ออกมาแสดง
```

ว่ามากกว่าที่กำหนด

```
          "C"     =   015       เสียงเหมือนเดิม
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมุติให้หาค่าที่มากกว่า 10 ก็ทำเช่นกันโดยกด KEY "F"

"F" =] 000

10 =] 010

"C" = READY

"C" = 014 เสียงจะเป็นเสียง ติ๊ก ต๋อก ออกมาที่ลำโพงแสดงว่าความสูงเกินกว่า 10 ตามที่กำหนด

"C" = 013 เสียงเหมือนเดิม

"C" = 003 เสียงจะเป็นเสียง ปิ๊ป ออกมาแสดงว่าต่ำกว่าที่กำหนด

"C" = 002 เสียงเหมือนเดิม

"C" = 000 เสียงเหมือนเดิม

** หมายเหตุ ** เสียง ติ๊ก ต๋อก หมายถึง การยอมรับตรงกับที่กำหนด

เสียง ปิ๊ป หมายถึง การไม่ยอมรับตรงกับที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดลอง

จากการทดลองโดยการใช้งานจริง ด้วยคลื่นอุลตร้าโซนิคมีการเคลื่อนที่เป็นเส้นตรง และสามารถสะท้อนได้ จึงได้นำเอามาประยุกต์สร้างเป็นเครื่องวัดความสูง และสามารถทำงานได้เป็นที่พอใจ แต่ก็มีอุปสรรคพอสมควรเกี่ยวกับการสะท้อนกลับของคลื่นอุลตร้าโซนิค เพราะว่าเมื่อมีระยะทางที่ไกลออกไปทำให้คลื่นอ่อนกำลังลง และบางครั้งไม่สามารถสะท้อนกลับมาสู่เครื่องเป็นผลให้เครื่องไม่สามารถทำงานได้

จากการทดลองระยะที่ดีที่สุดจะอยู่ประมาณ 180 - 190 CM โดยที่เราปรับอัตราการขยายของภาครับด้วย Vr 22 k จะมีผลที่ดีขึ้น แต่เมื่อปรับไปมากๆแล้วก็จะทำให้เกิดการผิดพลาดได้เช่นกันเพราะสัญญาณรบกวนภายนอกจะเข้าสู่ตัวรับอุลตร้าโซนิคได้มากขึ้น การทดลองของเครื่องในขณะที่เงียบไม่มีเสียงรบกวนจะดีกว่าในขณะที่มีเสียงรบกวน ทั้งนี้เพราะว่าจะมีสัญญาณรบกวนรอบๆเข้ามารบกวนต่อเครื่องวัด แม้กระทั่งการสั่นสะเทือนก็จะมีผลต่อเครื่องเช่นกัน(ย่อนกลับไปดูคุณสมบัติของทรานสดิวเซอร์)

ปัญหาอีกประการก็คือวัตถุที่สะท้อนกลับไม่ตั้งฉากกับเครื่องวัดเป็นเหตุให้คลื่นอุลตร้าโซนิคหักเหไปในทิศทางอื่นจึงจำเป็นต้องจัดให้เครื่องวัดและวัตถุที่สะท้อนตั้งฉากกันมากที่สุด ด้วยข้อเสียต่างๆที่เกิดขึ้นเราสามารถแก้ไขให้เครื่องดีขึ้นได้ อย่าเช่นเราจะให้ภาคส่งคลื่นอุลตร้าโซนิคให้มีกำลังมากขึ้นไป และในส่วนของภาครับมีความไวในการรับมากยิ่งขึ้น ก็สามารถที่จะแก้ปัญหาลงไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผล

จากแนวความคิดที่นำเอาคลื่นอุลตร้าโซนิคมาทำการประยุกต์ใช้ในการมาสร้างเครื่องวัดความสูง ผลที่ได้ออกมาทำให้เราทราบว่าเครื่องสามารถทำงานได้จริงคือสามารถวัดได้ตามต้องการ โดยในเครื่องวัดได้เอา MICRO PROCESSOR มาทำการควบคุมการทำงานทั้งหมด ทำให้ลดความยุ่งยากหลายๆอย่างลงไปได้และยังสามารถที่ให้ประสิทธิภาพได้ดีอีกด้วย ทั้งนี้ก็ขึ้นอยู่กับการที่เราจะเขียนโปรแกรมเข้าไปในเครื่อง เพื่อให้เครื่องปฏิบัติตามที่เรากำหนดได้

โดยทั้งหมดนี้ก็เกิดอุปสรรคต่างๆ แต่ก็เป็นสิ่งทำให้เราทราบว่าในการประยุกต์ใช้งานเราสามารถที่จะทำได้ ซึ่งทั้งหมดนี้เราก็ได้บรรลุถึงวัตถุประสงค์ตามที่เราได้ตั้งไว้ทุกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

หนังสือฉบับนี้เสร็จสมบูรณ์ได้ ด้วยความอนุเคราะห์จากอาจารย์ทุกท่านและเพื่อนๆทุกคน โดยเฉพาะอาจารย์ นิกร สุขุตมตันติ อาจารย์ที่ปรึกษา ที่ได้ให้ข่าวสารข้อมูลที่จำเป็น ตลอดจนคำแนะนำต่างๆ จึงขอขอบคุณทุกท่านไว้ ณ โอกาสนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ชูชัย,ดินัย,ทินกร,ธงชัย,ธานินทร์, "การใช้ Z-80 กับหน่วยความจำ", การใช้งาน Z-80 ,หน้า 9-50 , พิมพ์ครั้งแรก
2. บริษัทซีเอ็ดยูเคชั่น จำกัด " การใช้งานอุลตร้าโซนิค ", รวมโครงงานอิเลกทรอนิคส์, หน้า83-97, 2521
3. บริษัท ETT จำกัด " MANUAL " Z-80 ET-BOARD, หน้า 1-251,2532
4. บริษัทซีเอ็ดยูเคชั่น จำกัด "CHIP SUPPORT AMD MEMORY DATA BOOK " คู่มือไอซี, หน้าที่ 64, 2529
5. บริษัทซีเอ็ดยูเคชั่น จำกัด " MICROPROCESSOR DATA BOOK " คู่มือไอซี , หน้า 467, 2525

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROGRAM MONITOR

        ORG 00H
        -- MAIN --

        AF
RE:     3D
        20 FD       ,RE
        31 1F 3F
        3E 88
        D3 83
        3E 01
        D3 80
        21 00 20
        01 00 1F
G1:     3E 00
        77
        23
        0B
        78
        B1
        20 F7       ,G1
        21 3E 0F
        CD 00 05
        21 00 01
        CD 30 05
        21 26 3F
        3E 00
        06 0C
G2:     77
        23
        10 FC       ,G2
        3E 01
        32 34 3F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        21 4A 0F
        CD 00 05
G3:     CD 40 06
G4:     CD 10 05
        CD 40 05
        38 02          ,G5
        18 F6          ,G4
G5:     FE 0C
        20 EF          ,G3
        CD A4 05
G6:     CD B0 05
        22 2C 3F
        CD 00 06
        C5
        78
        0E 22
        CD B0 05
        C1
        79
        0E 24
        CD B0 05
        21 20 00
        CD 30 05
        CD 40 05
        30 DE          ,G6
        FE 0D
        20 DA          ,G6
        21 20 0F
        CD 00 05
        CD 40 06
        CD 10 05
        CD 40 05
        30 F8
        CD 60 06
        18 F0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    END

                SUBROUTINE LOAD-TABLE
                INPUT HL : START ADDRESS
0500                C5
                    D5
                    11 20 3F
                    01 60 00
                    ED B0
                    D1
                    C1
                    C9
                    END
                SUBROUTINE DISPLAY
0510                21 20 3F
                    0E 06
                    16 00
                T2: 7A
                    D3 82
                    7E
                    D3 81
                    AF
                T1: 3D
                    20 FD       ,T1
                    D3 81
                    14
                    23
                    0D
                    20  EF      ,T2
                    C9
                    END
                SUBROUTINE DISPLAY-DELAY
0530            T3: E5
                    CD 10 05
                    E1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                2B
                7D
                B4
                20 F6           ,T3
                C9
                END

        SUBROUTINE SCAN-KEY
        OUTPUT AF:KEY
        FLAG C : SET IF KEY PRESSED
0540            21 00 0F
                06 04
                16 00
        T4:     7A
                D3 82
                DB 82
                CB 7F
                28 16
                23
                CB 77
                28 11
                23
                CB 6F
                28 0C
                23
                CB 67
                28 07
                14
                23
                10 E4           ,T4
                37
                3F
                C9
                7E
                37
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                F5
                01 10 80
                CD C0 06
                F1
                C9
                END
        SUBROUTINE BCD-7SEGMENT
        INPUT C:DIGIT DISPLAY
              A:BCD
0580            06 3F
                F5
                CB 3F
                CB 3F
                CB 3F
                CB 3F
                21 10 0F
                E5
                16 00
                5F
                19
                7E
                02
                03
                E1
                F1
                E6 0F
                5F
                19
                7E
                02
                3E 00
                32 22 3F
                C9
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          SUBROUITNE CLEAR-DISPLAY

05A4                21 20 3F
                    06 06
                    AF
            S0:     77
                    23
                    10 FC           ,S0
                    C9
                    END

          SUBROUTINE COUNT
          OUTPUT HL

05B0                21 00 00
                    3E 00
                    D3 80
            S1:     23
                    DB 82
                    CB 7F
                    20 F9           ,S1
                    3E 01
                    D3 80
                    C9
                    END
           DIVIDE COUNT TO CENTIMETERS
           INPUT HL
           OUTPUT BC
0600                01 00 00
          ONE:     11 07 00
                    3E 00
          S2:       F5
                    7C
                    FE 01
                    30 07           ,TWO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                7D
                FE 06
                30 02 E1 C9
        TWO:    37
                3F
                ED 52
                CD 2A 06
                F1
                3C
                FE 0A
                20 E6           ,S2
                CD 2A 06
                18 DF           ,ONE
                79
                3C
                27
                4F
                3E 00
                88
                27
                47
                C9
                END
        SUBROUTINE  DEBOUNCE KEY
0640            06 40
        A0:     21 40 00
                CD 30 05
                06 00
        S3:     78
                D3 82
                DB 82
                E6 F0
                FE F0
                20 ED           ,A0
                04
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    78
                    FE 03
                    20 EF          ,S3
                    C9
                    END
          SUBROUTINE  FUNCTION KEY

0660                FE 0A
                    CA 30 08
                    FE 0B
                    CA 90 08
                    FE 0C
                    CA B0 06
                    FE 0D
                    CA E0 06
                    FE 0E
                    CA 10 0C
                    FE 0F
                    CA E0 0A
                    C9
                    END

      SUBROUTINE FUNCTION-KEY "ENTER"

0680                F1
                    CD B0 05
                    E5
                    C1
                    2A 2C 3F
                    37
                    3F
                    ED 42
                    CD 00 06
                    C5
                    CD A4 05
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                C9
                END
            SUBROUTINE    KEY RECORD
06E0            F1
                21 50 0F
                CD 00 05
        B3:     CD 40 06
        B0:     CD 10 05
                CD 40 05
                30 F8           ,B0
                FE 0C
                28 05           ,B1
                CD 60 06
                18 EC           ,B3
    B1,U2:      CD 40 07
                1A
                FE 99
                20 12           ,U1
                3E 98
                12
                21 44 0F
                CD 00 05
                21 F0 00
      U1:       CD 30 05
                C3 6B 00
                DD E5
                D5
                CD A0 07
                C5
                21 F0 00
                CD 30 05
                C1
                D1
                DD E1
                79
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DD 77 00
                DD 23
                78
                DD 77 00
                DD 23
                1A
                3C
                27
                12
                DD E5
                D5
                C5
                D1
                CD 60 0A
                D1
                DD E1
                18 B8          ,U2
                END

        SUBROUTINE  INSERT ROOM
0740            3A 2B 3F
                FE 00
                20 10          ,U3
                F1
                21 26 0F
                CD 00 05
                21 F0 00
                CD 30 05
                C3 6B 00
        U3:     DD 21 00 1F
                F5
                3E 00
                32 29 3F
                F1
                32 28 3F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CD 00 08
                3A 26 3F
                47
                0E 00
                DD 09
                3A 34 3F
                32 28 3F
                F5
                CD 00 08
                F1
                DD E5
                D1
                21 F0 00
                19
                E5
                D1
                12
                3A 26 3F
                4F
                06 00
                CB 21
                CB 10
                DD 09
                C9
                END
```

SUBROUTINE FIND MEMORY TO RECORD

```
0700            D5
                CD A4 05
                D1
                1A
                0E 20
                CD 80 05
                3E 40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          32 22 3F
C2:       CD 40 06
C0:       CD 10 05
          CD 40 05
          30 F8          ,C0
          FE 0C
          28 0F          ,C1
          FE 0A
          20 06          ,U4
          F1
          F1
          F1
          C3 6B 00
U4:       CD 60 06
          18 E2          ,C2
C1:       CD B0 05
          E5
          C1
          2A 2C 3F
          37
          3F
          ED 42
          CD 00 06
          C5
          78
          0E 22
          CD B0 05
          C1
          79
          C5
          0E 24
          CD B0 05
          3E 40
          32 22 3F
          C1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                C9
                END

        SUBROUTINE      BCD TO HEX
         INPUT  3F28-3F2A
         OUTPUT 3F26-3F27
0800            3E 00
                32 2A 3F
                0E 10
                06 03
                AF
                21 2A 3F
D2:             7E
                1F
                F5
                CB 7F
                28 02           ,D0
                D6 30
D0,U5           CB 5F
                28 02           ,D1
                D6 03
D1:             77
                2B
                F1
                10 EC           ,D2
                CB 1E
                2B
                CB 1E
                0D
                20 DE           ,U5
                C9
                END
```

SUBROUTINE FUNCTION KEY "INSERT ROOM NUMBER"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0830                F1
                    21 2C 0F
                    CD 00 05
                    CD 20 0A
                    3A 36 3F
                    FE 32
                    30 0B          ,D3
                    32 2B 3F
                    3E 01
                    32 34 3F
                    C3 6B 00
D3:                 21 26 0F
                    CD 00 05
                    21 F0 00
                    CD 30 05
                    C3 6B 00
                    END
  SUBROUTINE  INPUT 2 DIGID
0860      D5,U6:    CD 40 06
            D4:     CD 10 05
                    CD 40 05
                    30 F8          ,D4
                    FE 0C
                    C8
                    FE 0A
                    30 EE          ,D5
                    21 36 3F
                    ED 6F
                    3A 22 3F
                    F5
                    7E
                    0E 24
                    CD 80 05
                    F1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                32 22 3F
                18 D9           ,U6
                END
```

SUBROUTINE FUNCTION KEY " STUDENT NUMBER "

```
0890            F1
                21 32 0F
                CD 00 05
                CD 00 0A
                3A 36 3F
                FE 99
                38 0F           ,F0
                21 26 0F
                CD 00 05
                21 F0 00
                CD 30 05
                C3 6B 00
F0:             32 34 3F
                CD 20 09
                3A 34 3F
                47
                1A
                B8
                30 0F           ,F1
                21 38 0F
                CD 00 05
                21 F0 00
                CD 30 05
                C3 6B 00
F1:             DD 7E 00
                D5
                0E 24
                CD 80 05
                DD 23
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DD 7E 00
                0E 22
                CD 80 05
                3A 34 3F
                F5
                0E 20
                CD 80 05
                3E 40
                32 22 3F
                F1
                3C
                27
                47
                D1
                1A
                B8
                30 02       ,X2
                06 01
X2:             78
                32 34 2F
X3:             CD 40 06
X1:             CD 10 05
                CD 40 05
                30 F8       ,X1
                FE 0C
                CA B3 08
                CD 60 06
                18 EB
                END

SUBROUTINE FIND MEMORY TO READ

0920            3A 2B 3F
                FE 00
                20 10       ,X4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        F1
        21 26 0F
        CD 00 05
        21 F0 00
X4:     CD 30 05
        C3 6B 00
        DD 21 00 1F
        F5
        3E 00
        32 29 3F
        F1
        32 28 3F
        CD 00 08
        3A 26 3F
        47
        0E 00
        DD 09
        3A 34 3F
        32 28 3F
        F5
        CD 00 08
        F1
        DD E5
        D1
        21 F0 00
        19
        E5
        D1
        3A 26 3F
        4F
        06 00
        CB 21
        CB 10
        DD 09
        C9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    END
0A00                3A 34 3F
                    32 36 3F
                    0E 24
                    CD 80 05
                    3E 54
                    32 22 3F
                    CD 60 08
                    C9
                    END
0A20                3A 2B 3F
                    32 36 3F
                    0E 24
                    CD 80 05
                    3E 54
                    32 22 3F
                    CD 60 08
                    C9
                    END
                SUBROUTINE  COMPARE
0A60                3A 2F 3F
                    CB 47
                    28 07       ,X5
                    2A 20 3F
                    CD 90 0A
                    C9
            X5:     CB 67
                    C8
                    2A 39 3F
                    CD 90 0A
                    38 04
                    CD B0 0A
                    C9
                    CD B0 0A
                    C9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    END
0A90                CB 4F
                    28 07           ,X6
                    7A
                    BC
                    B8
                    C0
                    7B
                    BD
                    C9
X6:                 7A
                    BC
                    3F
                    D0
                    C0
                    7B
                    BD
                    3F
                    C9
                    END
0AB0                01 08 30
                    CD C0 06
                    21 20 00
                    CD 30 05
                    01 08 B0
                    CD C0 06
                    C9
                    END
0AD0                01 40 B0
                    CD C0 06
                    C9
                    END
```

SUBROUTINE FUNCTION KEY "F"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0AE0                F1
                    CD A4 05
                    AF
                    32 37 3F
                    32 38 3F
                    3A 38 3F
                    0E 20
                    CD 80 05
                    3A 37 3F
                    0E 24
                    CD 80 05
                    3A 2F 3F
                    CB 4F
                    20 07          ,X7
                    3E 39
                    32 21 3F
                    18 05
X7:                 3E 0F
                    32 21 3F
                    CD 40 06
X8:                 CD 10 05
                    CD 40 05
                    30 F8          ,X8
                    FE 0C
                    20 1B          ,X9
                    3A 38 3F
                    47
                    3A 37 3F
                    B0
                    28 0F
                    3A 2F 3F
                    F6 10
                    32 2F 3F
                    2A 37 3F
                    22 39 3F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                C3 6B 00
        X9:     FE 0F
                20 0B           ,X10
                3A 2F 3F
                EE 02
                32 2F 3F
                C3 F9 0A
        X10:    CD 60 06
                21 27 3F
                ED 6F
                23
                ED 6F
                C3 E9 0A
                END

        SUBROUTINE  FUNCTION KEY "E"
0C10            F1
                CD A4 05
                AF
                32 30 3F
                32 31 3F
                3A 31 3F
                0E 22
                CD 80 05
                3A 30 3F
                0E 24
                CD 80 05
                3A 2F 3F
                CB 4F
                20 07           ,Y0
                3E 39
                32 21 3F
                18 05           ,Y1
          Y0:   3E 0F
                32 21 3F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Y1:     CD 40 06
Y0:     CD 10 05
        CD 40 05
        30 F8           ,Z0
        FE 0C
        CA 80 0C
        FB 0E
        20 0B           ,Y2
        3A 2F 3F
        EE 02
        32 2F 3F
        C3 2B 0C
Y2:     CD 60 06
        21 30 3F
        ED 6F
        23
        ED 6F
        C3 1B 0C
        END
0C80    3A 2F 3F
        F6 01
        32 2F 3F
        CD 20 09
        06 00
        0E 00
        1A
        B9
        20 1B           ,Y3
        C5
        21 32 0F
        CD 00 05
        F1
        0E 24
        CD 80 05
        3E 54
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                32 22 3F
                21 FF 00
                CD 30 05
                C3 6B 00
Y3:             DD 5E 00
                DD 23
                DD 56 00
                DD 23
                DD E5
                C5
                CD 60 0A
                C1
                DD E1
                D1
                38 04       ,Y4
                78
                3C
                27
                47
Y4:             79
                3C
                27
                4F
                C3 8F 0C
                END
```

TABLE

0F00 HEXADECIMAL

00 04 08 0C 01 05 09 0D

02 06 0A 0E 03 07 0B 0F

0F10 SEVEN SEGMENT

3F 06 5B 4F 66 6D 7D 07

7F 6F 77 7C 39 5E 79 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0F20            " READY "
                00 50 79 77 5E 6E

0F26            " ERROR "
                00 79 50 50 54 50

0F2C            " RO.NO. " (ROOM NUMBER)
                50 DC 54 DC 00 00

0F32            " ST.NO. " (STUDENT NUMBER)
                6D F8 54 DC 00 00

0F38            " NOData "
                54 5C 5E 77 78 77

0F3E            " 2L.2533 "
                5B B8 5D 6D 4F 4F

0F44            " FULL "
                00 71 1C 38 38 00

0F4A            " SEtrEF. " (SET REFFERENT)
                6D 79 78 50 79 71

0F50            " REC " (RECORD DATA)
                50 79 3F 00 00 00
END TABLE

                * END OF PROGRAM *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 คำสั่งต่าง ๆ ของ Z80 แสดงทั้งรหัสภาษาเครื่อง, นีโมนิกและการทำงานของคำสั่ง

n คือ จำนวนถูกของสัญญาณนาฬิกา
M คือจำนวนแมชชีนไซเคิล

* หมายถึงคำสั่งที่ทำให้แฟลกเปลี่ยนแปลง

□: หมายถึงการใส่ค่าของข้อมูล (ตัวเลขหรือแอดเดรส) ฉ, อ-ว ไปในช่องนี้

▨: หมายถึงการใส่ค่า displacement (ด ไปในช่องนี้

	ภาษาเครื่อง	รหัสนีโมนิค	การทำงาน	M	T
*	8▨	ADC A, (HL)	A←A+(HL)+CY	2	7
*	DD8E▨	ADC A, (IX+ด)	A←A+(IX+d)+CY	5	19
*	FD8E▨	ADC A, (IY+ด)	A←A+(IY+d)+CY		
*	8F	ADC A, A	A←A+A+CY	1	4
*	88	ADC A, B	A←A+B+CY		
*	89	ADC A, C	A←A+C+CY		
*	8A	ADC A, D	A←A+D+CY		
*	8B	ADC A, E	A←A+E+CY		
*	8C	ADC A, H	A←A+H+CY		
*	8D	ADC A, L	A←A+L+CY		
*	CE□	ADC A, ฉ	A←A+ฉ+CY	2	7
*	ED4A	ADC HL, BC	HL←HL+BC+CY	4	15
*	ED5A	ADC HL, DE	HL←HL+DE+CY		
*	ED6A	ADC HL, HL	HL←HL+HL+CY		
*	ED7A	ADC HL, SP	HL←HL+SP+CY		
*	86	ADD A, (HL)	A←A+(HL)	2	7
*	DD86▨	ADD A, (IX+ด)	A←A+(IX+d)	5	19
*	FD86▨	ADD A, (IY+ด)	A←A+(IY+d)		
*	87	ADD A, A	A←A+A	1	4
*	80	ADD A, B	A←A+B		
*	81	ADD A, C	A←A+C		
*	82	ADD A, D	A←A+D		
*	83	ADD A, E	A←A+E		
*	84	ADD A, H	A←A+H		
*	85	ADD A, L	A←A+L		
*	C6□	ADD A, ฉ	A←A+ฉ	2	7
*	09	ADD HL, BC	HL←HL+BC	3	11
*	19	ADD HL, DE	HL←HL+DE		
*	29	ADD HL, HL	HL←HL+HL		
*	39	ADD HL, SP	HL←HL+SP		
*	DD09	ADD IX, BC	IX←IX+BC	4	15
*	DD19	ADD IX, DE	IX←IX+DE		
*	DD29	ADD IX, IX	IX←IX+IX		
*	DD39	ADD IX, SP	IX←IX+SP		
*	FD09	ADD IY, BC	IY←IY+BC	4	15
*	FD19	ADD IY, DE	IY←IY+DE		
*	FD29	ADD IY, IY	IY←IY+IY		
*	FD39	ADD IY, SP	IY←IY+SP		
*	A6	AND (HL)	A←A∧(HL)	2	7
*	DDA6▨	AND (IX+ด)	A←A∧(IX+d)	5	19
*	FDA6▨	AND (IY+ด)	A←A∧(IY+d)		
*	A7	AND A	A←A∧A	1	4
*	A0	AND B	A←A∧B		
*	A1	AND C	A←A∧C		
*	A2	AND D	A←A∧D		
*	A3	AND E	A←A∧E		
*	A4	AND H	A←A∧H		
*	A5	AND L	A←A∧L		
*	E6□	AND ฉ	A←A∧ฉ	2	7
*	CB46	BIT 0, (HL)		3	12
*	DDCB▨46	BIT 0, (IX+ด)		5	20
*	FDCB▨46	BIT 0, (IY+ด)			
*	CB47	BIT 0, A		2	8
*	CB40	BIT 0, B			
*	CB41	BIT 0, C			
*	CB42	BIT 0, D			
*	CB43	BIT 0, E			
*	CB44	BIT 0, H			
*	CB45	BIT 0, L			
*	CB4E	BIT 1, (HL)		3	12
*	DDCB▨4E	BIT 1, (IX+ด)		5	20
*	FDCB▨4E	BIT 1, (IY+ด)			
*	CB4F	BIT 1, A		2	8
*	CB48	BIT 1, B			
*	CB49	BIT 1, C			
*	CB4A	BIT 1, D			
*	CB4B	BIT 1, E			
*	CB4C	BIT 1, H			
*	CB4D	BIT 1, L			
*	CB56	BIT 2, (HL)		3	12
*	DDCB▨56	BIT 2, (IX+ด)		5	20
*	FDCB▨56	BIT 2, (IY+ด)			
*	CB57	BIT 2, A		2	8
*	CB50	BIT 2, B			
*	CB51	BIT 2, C			
*	CB52	BIT 2, D			
*	CB53	BIT 2, E			
*	CB54	BIT 2, H			
*	CB55	BIT 2, L			
*	CB5E	BIT 3, (HL)		3	12
*	DDCB▨5E	BIT 3, (IX+ด)		5	20
*	FDCB▨5E	BIT 3, (IY+ด)			
*	CB5F	BIT 3, A		2	8
*	CB58	BIT 3, B			
*	CB59	BIT 3, C			
*	CB5A	BIT 3, D			
*	CB5B	BIT 3, E			
*	CB5C	BIT 3, H			
*	CB5D	BIT 3, L			
*	CB66	BIT 4, (HL)		3	12
*	DDCB▨66	BIT 4, (IX+ด)		5	20
*	FDCB▨66	BIT 4, (IY+ด)			
*	CB67	BIT 4, A		2	8
*	CB60	BIT 4, B			
*	CB61	BIT 4, C			
*	CB62	BIT 4, D			
*	CB63	BIT 4, E			
*	CB64	BIT 4, H			
*	CB65	BIT 4, L			
*	CB6E	BIT 5, (HL)		3	12
*	DDCB▨6E	BIT 5, (IX+ด)		5	20
*	FDCB▨6E	BIT 5, (IY+ด)			
*	CB6F	BIT 5, A		2	8
*	CB68	BIT 5, B			
*	CB69	BIT 5, C			
*	CB6A	BIT 5, D			
*	CB6B	BIT 5, E			
*	CB6C	BIT 5, H			
*	CB6D	BIT 5, L			
*	CB76	BIT 6, (HL)		3	12
*	DDCB▨76	BIT 6, (IX+ด)		5	20
*	FDCB▨76	BIT 6, (IY+ด)			
*	CB77	BIT 6, A		2	8
*	CB70	BIT 6, B			
*	CB71	BIT 6, C			
*	CB72	BIT 6, D			
*	CB73	BIT 6, E			
*	CB74	BIT 6, H			
*	CB75	BIT 6, L			
*	CB7E	BIT 7, (HL)		3	12
*	DDCB▨7E	BIT 7, (IX+ด)		5	20
*	FDCB▨7E	BIT 7, (IY+ด)			
*	CB7F	BIT 7, A		2	8
*	CB78	BIT 7, B			
*	CB79	BIT 7, C			
*	CB7A	BIT 7, D			
*	CB7B	BIT 7, E			
*	CB7C	BIT 7, H			
*	CB7D	BIT 7, L			
*	DC□□	CALL C, ▨▨	เมื่อแฟลก C ถูกเซ็ต	5	17 (1)
*	FC□□	CALL M, ▨▨	เมื่อแฟลก S ถูกเซ็ต (ค่าลบ)		
*	D4□□	CALL NC, ▨▨	เมื่อแฟลก C ถูกรีเซ็ต		
*	C4□□	CALL NZ, ▨▨	เมื่อแฟลก Z ถูกรีเซ็ต		
*	F4□□	CALL P, ▨▨	เมื่อแฟลก S ถูกรีเซ็ต (ค่าบวก)		
*	EC□□	CALL PE, ▨▨	เมื่อแฟลก P/V ถูกเซ็ต (พาริตี้คู่)	3	10 (2)
*	E4□□	CALL PO, ▨▨	เมื่อแฟลก P/V ถูกรีเซ็ต (พาริตี้คี่)		
*	CC□□	CALL Z, ▨▨	เมื่อแฟลก Z ถูกเซ็ต		
*	CD□□	CALL ▨▨		5	17
*	3F	CCF	CY←CY'	1	4
*	BE	CP (HL)	A−(HL)	2	7
*	DDBE▨	CP (IX+ด)	A−(IX+d)	5	19
*	FDBE▨	CP (IY+ด)	A−(IY+d)		
*	BF	CP A	A−A	1	4
*	B8	CP B	A−B		
*	B9	CP C	A−C		
*	BA	CP D	A−D		
*	BB	CP E	A−E		
*	BC	CP H	A−H		
*	BD	CP L	A−L		
*	FE□	CP ฉ	A−ฉ	2	7
*	EDA9	CPD	A−(HL), HL←HL−1, BC←BC−1	4	16
*	EDB9	CPDR	A−(HL), HL←HL−1, BC←BC−1, ทำจนกระทั่ง BC = 0	5 (3)	21
				4	16
*	EDA1	CPI	A−(HL), HL←HL+1, BC←BC−1	4	16
*	EDB1	CPIR	A−(HL), HL←HL+1, BC←BC−1, ทำจนกระทั่ง BC = 0	5 (3)	21
				4	16
*	2F	CPL	A←Ā	1	4
*	27	DAA		1	4
*	35	DEC (HL)	(HL)←(HL)−1	3	11
*	DD35▨	DEC (IX+ด)	(IX+d)←(IX+d)−1	6	23
*	FD35▨	DEC (IY+ด)	(IY+d)←(IY+d)−1		
*	3D	DEC A	A←A−1	1	4
*	05	DEC B	B←B−1		
*	0B	DEC BC	BC←BC−1	1	6
*	0D	DEC C	C←C−1	1	4
*	15	DEC D	D←D−1		
*	1B	DEC DE	DE←DE−1	1	6
*	1D	DEC E	E←E−1	1	4
*	25	DEC H	H←H−1		
*	2B	DEC HL	HL←HL−1	1	6
*	DD2B	DEC IX	IX←IX−1	2	10
*	FD2B	DEC IY	IY←IY−1		
*	2D	DEC L	L←L−1	1	4
*	3B	DEC SP	SP←SP−1	1	6
*	F3	DI	IFF←0 ห้ามขัดจังหวะที่มาสเคอร์ไซท์	1	4
*	10□	DJNZ ฉ	B←B−1, ถ้า B−1 ≠ 0 แล้ว PC←PC+ฉ, ถ้า B−1 = 0 แล้ว PC←PC+2	3 2	13 8

ทำการเทสแต่ละบิตในแต่ละไบต์ ว ถึงบิต 7 ของข้อมูลในแต่ละรีจิสเตอร์ที่ระบุมาและทำให้แฟลก Z ถูกเซ็ตเป็น "1" ถ้าบิตในไบต์นั้นเป็น "0" ค่าของข้อมูลภายในรีจิสเตอร์และหน่วยความจำจะไม่มีการเปลี่ยนแปลง

การทำงานของแต่ละคำสั่งในกลุ่มนี้ จะทำการเทสบิตที่ระบุในไบต์ที่ระบุ ว่าเป็น "0" หรือ "1" แล้วเซ็ตแฟลก Z ตามผลที่ได้ ค่าของข้อมูลภายในรีจิสเตอร์และหน่วยความจำจะไม่มีการเปลี่ยนแปลง

การบวกของข้อมูล ว ถึงค่าที่มีในแต่ละคำสั่ง

การลบระหว่างข้อมูลภายในรีจิสเตอร์ A กับรีจิสเตอร์ต่าง ๆ หรือหน่วยความจำ

การบวกของข้อมูล ว ถึงค่าที่มีในแต่ละคำสั่ง

การ AND ทางตรรกะระหว่างรีจิสเตอร์ A กับไบต์ข้อมูลต่าง ๆ

การ OR ทางตรรกะระหว่างรีจิสเตอร์ A กับไบต์ข้อมูลต่าง ๆ

การ XOR ทางตรรกะระหว่างรีจิสเตอร์ A กับไบต์ข้อมูลต่าง ๆ

การเปรียบเทียบระหว่างข้อมูลภายในรีจิสเตอร์ A กับรีจิสเตอร์ต่าง ๆ

ลบ 1 ออกจากข้อมูลภายในแต่ละรีจิสเตอร์หรือหน่วยความจำ

(1) PC←NN เมื่อไม่เกิดการโอนไป
(2) PC←PC+2 เมื่อไม่เกิดการโอนไป
(3) ใช้ตรวจสอบเมื่อ BC ≠ 0 และ A ≠ (HL)
ใช้ตรวจสอบเมื่อ BC = 0 หรือ A = (HL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ภาษาเครื่อง	รหัสที่โปรแกรม	การทำงาน	M	T
	FB	EI	IFF←1 ให้อินเตอร์รัพท์ได้	1	4
	E3	EX (SP), HL	L↔(SP), H↔(SP+1)	5	19
	DDE3	EX (SP), IX	IXL↔(SP), IXH↔(SP+1) แลก	6	23
	FDE3	EX (SP), IY	IYL↔(SP), IYH↔(SP+1) เปลี่ยน		
	08	EX AF, AF'	AF↔AF'	1	4
	EB	EX DE, HL	DE↔HL		
	D9	EXX	BC↔BC', DE↔DE', HL↔HL'		
	76	HALT	ให้หยุดการทำงาน	1	4
	ED46	IM 0	กำหนดโหมดของการอินเตอร์รัพท์	2	8
	ED56	IM 1			
	ED5E	IM 2			
*	ED78	IN A, (C)	อ่านข้อมูลจากพอร์ตอินพุตที่กำหนดโดยตัวโลเปอร์แรนด์ C ไปยังรีจิสเตอร์ที่กำหนด	3	12
*	ED40	IN B, (C)			
*	ED48	IN C, (C)			
*	ED50	IN D, (C)			
*	ED58	IN E, (C)			
*	ED60	IN H, (C)			
*	ED68	IN L, (C)			
	DB□	IN A, (n)	อ่านข้อมูลจากพอร์ตอินพุตที่กำหนดโดยตัวโลเปอร์แรนด์	3	11
*	34	INC (HL)	(HL)←(HL)+1	3	11
*	DD34■■	INC (IX+d)	(IX+d)←(IX+d)+1, เพิ่มค่า 1 เข้าไปในข้อมูลของหน่วยความจำหรือรีจิสเตอร์	6	23
*	FD34■■	INC (IY+d)	(IY+d)←(IY+d)+1		
*	3C	INC A	A←A+1	1	4
*	04	INC B	B←B+1		
	03	INC BC	BC←BC+1	1	6
*	0C	INC C	C←C+1	1	4
*	14	INC D	D←D+1		
	13	INC DE	DE←DE+1	1	6
*	1C	INC E	E←E+1	1	4
*	24	INC H	H←H+1		
	23	INC HL	HL←HL+1	1	6
	DD23	INC IX	IX←IX+1	2	10
	FD23	INC IY	IY←IY+1		
*	2C	INC L	L←L+1	1	4
	33	INC SP	SP←SP+1	1	6
*	EDAA	IND	(HL)←(C), B←B−1, HL←HL−1	4	16
*	EDBA	INDR	(HL)←(C), B←B−1, HL←HL−1, กระทำซ้ำจนกระทั่ง B = 0	5	21 (4)
				4	16
*	EDA2	INI	(HL)←(C), B←B−1, HL←HL+1	4	16
*	EDB2	INIR	(HL)←(C), B←B−1, HL←HL+1, กระทำซ้ำจนกระทั่ง B =0	5	21 (4)
				4	16
	E9	JP (HL)	ภาคระโดดไปโดยไม่มีเงื่อนไข	1	4
	DDE9	JP (IX)		2	8
	FDE9	JP (IY)			
	C3□□	JP nn		3	10
	DA□□	JP C, nn	ภาคระโดดโดยมีเงื่อนไข เมื่อแฟล็ก C ถูกเซ็ต	3	10
	FA□□	JP M, nn	เมื่อแฟล็ก S ถูกเซ็ต (ค่าลบ)		
	D2□□	JP NC, nn	เมื่อแฟล็ก C ถูกเคลียร์		
	C2□□	JP NZ, nn	เมื่อแฟล็ก Z ถูกเคลียร์		
	F2□□	JP P, nn	เมื่อแฟล็ก S ถูกเคลียร์ (ค่าบวก)		
	EA□□	JP PE, nn	เมื่อแฟล็ก P/V ถูกเซ็ต (พาริตี้คู่)		
	E2□□	JP PO, nn	เมื่อแฟล็ก P/V ถูกเคลียร์ (พาริตี้คี่)		
	CA□□	JP Z, nn	เมื่อแฟล็ก Z ถูกเซ็ต		
	38 (e−2)	JR C, e	เมื่อแฟล็ก C ถูกเซ็ต	3	12 (5)
	30□	JR NC, e	เมื่อแฟล็ก C ถูกเคลียร์		
	20□	JR NZ, e	เมื่อแฟล็ก Z ถูกเคลียร์		
	28□	JR Z, e	เมื่อแฟล็ก Z ถูกเซ็ต	2	7 (6)
	18□	JR e	กระโดดโดยไม่มีเงื่อนไข	3	12
	02	LD (BC), A	(BC)←A	2	7
	12	LD (DE), A	(DE)←A		
	77	LD (HL), A	(HL)←A	2	7
	70	LD (HL), B	(HL)←B		
	71	LD (HL), C	(HL)←C		
	72	LD (HL), D	(HL)←D		
	73	LD (HL), E	(HL)←E		
	74	LD (HL), H	(HL)←H		
	75	LD (HL), L	(HL)←L		
	36□	LD (HL), n	(HL)←n	3	10
	DD77■	LD (IX+d), A	(IX+d)←A	5	19
	DD70■	LD (IX+d), B	(IX+d)←B		
	DD71■	LD (IX+d), C	(IX+d)←C		
	DD72■	LD (IX+d), D	(IX+d)←D		
	DD73■	LD (IX+d), E	(IX+d)←E		
	DD74■	LD (IX+d), H	(IX+d)←H		
	DD75■	LD (IX+d), L	(IX+d)←L		
	DD36■□	LD (IX+d), n	(IX+d)←n	5	19
	FD77■	LD (IY+d), A	(IY+d)←A	5	19
	FD70■	LD (IY+d), B	(IY+d)←B		
	FD71■	LD (IY+d), C	(IY+d)←C		
	FD72■	LD (IY+d), D	(IY+d)←D		
	FD73■	LD (IY+d), E	(IY+d)←E		
	FD74■	LD (IY+d), H	(IY+d)←H		
	FD75■	LD (IY+d), L	(IY+d)←L		
	FD36■□	LD (IY+d), n	(IY+d)←n	5	19
	32□□	LD (nn), A	(nn)←A	4	13
	ED43□□	LD (nn), BC	(nn)←C, (nn+1)←B	6	20
	ED53□□	LD (nn), DE	(nn)←E, (nn+1)←D		
	22□□	LD (nn), HL	(nn)←L, (nn+1)←H	5	16
	DD22□□	LD (nn), IX	(nn)←IXL, (nn+1)←IXH	6	20

(4) ใช้คำสั่งนี้เมื่อ B ≠ 0 ใช้คำสั่งข้างล่าง เมื่อ B = 0

(5) ใช้คำสั่งนี้เมื่อเป็นโปรแกรมโลเคชั่น PC←PC + e

(6) ใช้คำสั่งข้างล่างเมื่อไม่เป็นโปรแกรมโลเคชั่น PC←PC + 2

	ภาษาเครื่อง	รหัสที่โปรแกรม	การทำงาน	M	T
	FD22□□	LD (nn), IY	(nn)←IYL, (nn+1)←IYH	6	20
	ED73□□	LD (nn), SP	(nn)←SPL, (nn+1)←SPH		
	0A	LD A, (BC)	A←(BC)	2	7
	1A	LD A, (DE)	A←(DE)		
	7E	LD A, (HL)	A←(HL)		
	DD7E■	LD A, (IX+d)	A←(IX+d)	5	19
	FD7E■	LD A, (IY+d)	A←(IY+d)		
	3A□□	LD A, (nn)	A←(nn)	4	13
	7F	LD A, A	A←A	1	4
	78	LD A, B	A←B		
	79	LD A, C	A←C		
	7A	LD A, D	A←D		
	7B	LD A, E	A←E		
	7C	LD A, H	A←H		
*	ED57	LD A, I	A←I	2	9
	7D	LD A, L	A←L	1	4
	3E□	LD A, n	A←n	2	7
*	ED5F	LD A, R	A←R	2	9
	46	LD B, (HL)	B←(HL)	2	7
	DD46■	LD B, (IX+d)	B←(IX+d)	5	19
	FD46■	LD B, (IY+d)	B←(IY+d)		
	47	LD B, A	B←A	1	4
	40	LD B, B	B←B		
	41	LD B, C	B←C		
	42	LD B, D	B←D		
	43	LD B, E	B←E		
	44	LD B, H	B←H		
	45	LD B, L	B←L		
	06□	LD B, n	B←n	2	7
	ED4B□□	LD BC, (nn)	C←(nn), B←(nn+1)	6	20
	01□□	LD BC, nn	BC←nn	3	10
	4E	LD C, (HL)	C←(HL)	2	7
	DD4E■	LD C, (IX+d)	C←(IX+d)	5	19
	FD4E■	LD C, (IY+d)	C←(IY+d)		
	4F	LD C, A	C←A	1	4
	48	LD C, B	C←B		
	49	LD C, C	C←C		
	4A	LD C, D	C←D		
	4B	LD C, E	C←E		
	4C	LD C, H	C←H		
	4D	LD C, L	C←L		
	0E□	LD C, n	C←n	2	7
	56	LD D, (HL)	D←(HL)	2	7
	DD56■	LD D, (IX+d)	D←(IX+d)	5	19
	FD56■	LD D, (IY+d)	D←(IY+d)		
	57	LD D, A	D←A	1	4
	50	LD D, B	D←B		
	51	LD D, C	D←C		
	52	LD D, D	D←D		
	53	LD D, E	D←E		
	54	LD D, H	D←H		
	55	LD D, L	D←L		
	16□	LD D, n	D←n	2	7
	ED5B□□	LD DE, (nn)	E←(nn), D←(nn+1)	6	20
	11□□	LD DE, nn	DE←nn	3	10
	5E	LD E, (HL)	E←(HL)	2	7
	DD5E■	LD E, (IX+d)	E←(IX+d)	5	19
	FD5E■	LD E, (IY+d)	E←(IY+d)		
	5F	LD E, A	E←A	1	4
	58	LD E, B	E←B		
	59	LD E, C	E←C		
	5A	LD E, D	E←D		
	5B	LD E, E	E←E		
	5C	LD E, H	E←H		
	5D	LD E, L	E←L		
	1E□	LD E, n	E←n	2	7
	66	LD H, (HL)	H←(HL)	2	7
	DD66■	LD H, (IX+d)	H←(IX+d)	5	19
	FD66■	LD H, (IY+d)	H←(IY+d)		
	67	LD H, A	H←A	1	4
	60	LD H, B	H←B		
	61	LD H, C	H←C		
	62	LD H, D	H←D		
	63	LD H, E	H←E		
	64	LD H, H	H←H		
	65	LD H, L	H←L		
	26□	LD H, n	H←n	2	7
	2A□□	LD HL, (nn)	L←(nn), H←(nn+1)	5	16
	21□□	LD HL, nn	HL←nn	3	10
	ED47	LD I, A	I←A	2	9
	DD2A□□	LD IX, (nn)	IXL←(nn), IXH←(nn+1)	6	20
	DD21□□	LD IX, nn	IX←nn	4	14
	FD2A□□	LD IY, (nn)	IYL←(nn), IYH←(nn+1)	6	20
	FD21□□	LD IY, nn	IY←nn	4	14
	6E	LD L, (HL)	L←(HL)	2	7
	DD6E■	LD L, (IX+d)	L←(IX+d)	5	19
	FD6E■	LD L, (IY+d)	L←(IY+d)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 (ต่อ)

	ภาษาเครื่อง	รหัสนีโมนิค	การทำงาน	M	T
	6F	LD L, A	L←A	1	4
	68	LD L, B	L←B		
	69	LD L, C	L←C		
	6A	LD L, D	L←D		
	6B	LD L, E	L←E		
	6C	LD L, H	L←H		
	6D	LD L, L	L←L		
	2E□	LD L, n	L←n	2	7
	ED4F	LD R, A	R←A	2	9
	ED7B□□	LD SP, (nn)	SPp←(nn), SPr←(nn+1)	6	20
	F9	LD SP, HL	SP←HL	1	6
	DDF9	LD SP, IX	SP←IX	2	10
	FDF9	LD SP, IY	SP←IY		
	31□□	LD SP, nn	SP←nn	3	10
•	EDA8	LDD	(DE)←(HL), DE←DE−1, HL←HL−1, BC←BC−1	4	16
•	EDB8	LDDR	(DE)←(HL), DE←DE−1, HL←HL−1, BC←BC−1, ทำซ้ำจนกระทั่ง BC = 0	5	21 (6)
				4	16
•	EDA0	LDI	(DE)←(HL), DE←DE+1, HL←HL+1, BC←BC−1	4	16
•	EDB0	LDIR	(DE)←(HL), DE←DE+1, HL←HL+1, BC←BC−1, ทำซ้ำจนกระทั่ง BC = 0	5	21 (6)
				4	16
•	ED44	NEG	A←0−A หาค่าคอมพลีเมนต์ที่ 2 ของข้อมูลใน A	2	8
	00	NOP	เพิ่มค่า PC ไป 1 โดยไม่มีผลอย่างอื่น	1	4
•	B6	OR (HL)	A←A∨(HL)	2	7
•	DDB6■	OR (IX+d)	A←A∨(IX+d)	5	19
•	FDB6■	OR (IY+d)	A←A∨(IY+d)		
•	B7	OR A	A←A∨A	1	4
•	B0	OR B	A←A∨B		
•	B1	OR C	A←A∨C		
•	B2	OR D	A←A∨D		
•	B3	OR E	A←A∨E		
•	B4	OR H	A←A∨H		
•	B5	OR L	A←A∨L		
•	F6□	OR n	A←A∨n	2	7
•	EDBB	OTDR	(C)←(HL), B←B−1, HL←HL−1, ทำซ้ำจนกระทั่ง B = 0	5	21 (6)
				4	16
•	EDB3	OTIR	(C)←(HL), B←B−1, HL←HL+1, ทำซ้ำจนกระทั่ง B = 0	5	21 (6)
				4	16
	ED79	OUT (C), A	(C)←A	3	12
	ED41	OUT (C), B	(C)←B		
	ED49	OUT (C), C	(C)←C		
	ED51	OUT (C), D	(C)←D		
	ED59	OUT (C), E	(C)←E		
	ED61	OUT (C), H	(C)←H		
	ED69	OUT (C), L	(C)←L		
	D3□	OUT (n), A	(n)←A	3	11
•	EDAB	OUTD	(C)←(HL), B←B−1, HL←HL−1	4	16
•	EDA3	OUTI	(C)←(HL), B←B−1, HL←HL+1	4	16
	F1	POP AF	F←(SP), A←(SP+1)	3	10
	C1	POP BC	C←(SP), B←(SP+1)		
	D1	POP DE	E←(SP), D←(SP+1)		
	E1	POP HL	L←(SP), H←(SP+1)		
	DDE1	POP IX	IXr←(SP), IXp←(SP+1)	4	14
	FDE1	POP IY	IYr←(SP), IYp←(SP+1)		
	F5	PUSH AF	(SP−1)←A, (SP−2)←F	3	11
	C5	PUSH BC	(SP−1)←B, (SP−2)←C		
	D5	PUSH DE	(SP−1)←D, (SP−2)←E		
	E5	PUSH HL	(SP−1)←H, (SP−2)←L		
	DDE5	PUSH IX	(SP−1)←IXp, (SP−2)←IXr	4	15
	FDE5	PUSH IY	(SP−1)←IYp, (SP−2)←IYr		
	CB86	RES 0, (HL)		4	15
	DDCB■ 86	RES 0, (IX+d)		6	23
	FDCB■ 86	RES 0, (IY+d)			
	CB87	RES 0, A		2	8
	CB80	RES 0, B			
	CB81	RES 0, C			
	CB82	RES 0, D			
	CB83	RES 0, E			
	CB84	RES 0, H			
	CB85	RES 0, L			
	CB8E	RES 1, (HL)		4	15
	DDCB■ 8E	RES 1, (IX+d)		6	23
	FDCB■ 8E	RES 1, (IY+d)			
	CB8F	RES 1, A		2	8
	CB88	RES 1, B			
	CB89	RES 1, C			
	CB8A	RES 1, D			
	CB8B	RES 1, E			
	CB8C	RES 1, H			
	CB8D	RES 1, L			
	CB96	RES 2, (HL)		4	15
	DDCB■ 96	RES 2, (IX+d)		6	23
	FDCB■ 96	RES 2, (IY+d)			
	CB97	RES 2, A		2	8
	CB90	RES 2, B			
	CB91	RES 2, C			
	CB92	RES 2, D			
	CB93	RES 2, E			
	CB94	RES 2, H			
	CB95	RES 2, L			
	CB9E	RES 3, (HL)		4	15
	DDCB■ 9E	RES 3, (IX+d)		6	23

(6) ใช้คำสั่งนี้เมื่อเริ่ม BC / 0. ใช้คำสั่งนี้เมื่อเริ่ม BC = 0

	ภาษาเครื่อง	รหัสนีโมนิค	การทำงาน	M	T
	FDCB■ 9E	RES 3, (IY+d)		6	23
	CB9F	RES 3, A		2	8
	CB98	RES 3, B			
	CB99	RES 3, C			
	CB9A	RES 3, D			
	CB9B	RES 3, E			
	CB9C	RES 3, H			
	CB9D	RES 3, L			
	CBA6	RES 4, (HL)		4	15
	DDCB■ A6	RES 4, (IX+d)		6	23
	FDCB■ A6	RES 4, (IY+d)			
	CBA7	RES 4, A		2	8
	CBA0	RES 4, B			
	CBA1	RES 4, C			
	CBA2	RES 4, D			
	CBA3	RES 4, E			
	CBA4	RES 4, H			
	CBA5	RES 4, L			
	CBAE	RES 5, (HL)		4	15
	DDCB■ AE	RES 5, (IX+d)		6	23
	FDCB■ AE	RES 5, (IY+d)			
	CBAF	RES 5, A		2	8
	CBA8	RES 5, B			
	CBA9	RES 5, C			
	CBAA	RES 5, D			
	CBAB	RES 5, E			
	CBAC	RES 5, H			
	CBAD	RES 5, L			
	CBB6	RES 6, (HL)		4	15
	DDCB■ B6	RES 6, (IX+d)		6	23
	FDCB■ B6	RES 6, (IY+d)			
	CBB7	RES 6, A		2	8
	CBB0	RES 6, B			
	CBB1	RES 6, C			
	CBB2	RES 6, D			
	CBB3	RES 6, E			
	CBB4	RES 6, H			
	CBB5	RES 6, L			
	CBBE	RES 7, (HL)		4	15
	DDCB■ BE	RES 7, (IX+d)		6	23
	FDCB■ BE	RES 7, (IY+d)			
	CBBF	RES 7, A		2	8
	CBB8	RES 7, B			
	CBB9	RES 7, C			
	CBBA	RES 7, D			
	CBBB	RES 7, E			
	CBBC	RES 7, H			
	CBBD	RES 7, L			
	C9	RET	return PCr←(SP), PCp←(SP+1), SP←SP+1	3	10
	D8	RET C	เมื่อแฟลก C ถูกเซ็ต	1	5 (8)
	F8	RET M	เมื่อแฟลก S ถูกเซ็ต (ค่าลบ)	3	11 (7)
	D0	RET NC	เมื่อแฟลก C ถูกรีเซ็ต		
	C0	RET NZ	เมื่อแฟลก Z ถูกรีเซ็ต		
	F0	RET P	เมื่อแฟลก S ถูกรีเซ็ต (ค่าบวก)		
	E8	RET PE	เมื่อแฟลก P/V ถูกเซ็ต (พาริตี้คู่)		
	E0	RET PO	เมื่อแฟลก P/V ถูกรีเซ็ต (พาริตี้คี่)		
	C8	RET Z	เมื่อแฟลก Z ถูกเซ็ต		
	ED4D	RETI	สิ้นสุดการทำงานของโปรแกรมย่อยที่ใช้เพื่อร้องขอการขัดจังหวะ	4	14
	ED45	RETN	สิ้นสุดการทำงานของโปรแกรมย่อยที่ใช้เพื่อร้องขอการขัดจังหวะแบบไม่บดบังได้	4	14
•	CB16	RL (HL)		4	15
•	DDCB■ 16	RL (IX+d)		6	23
•	FDCB■ 16	RL (IY+d)			
•	CB17	RL A		2	8
•	CB10	RL B			
•	CB11	RL C			
•	CB12	RL D			
•	CB13	RL E			
•	CB14	RL H			
•	CB15	RL L			
•	17	RLA	หมุนข้อมูลใน A ไปทางซ้าย	1	4
•	CB06	RLC (HL)		4	15
•	DDCB■ 06	RLC (IX+d)		6	23
•	FDCB■ 06	RLC (IY+d)			
•	CB07	RLC A		2	8
•	CB00	RLC B			
•	CB01	RLC C			
•	CB02	RLC D			
•	CB03	RLC E			
•	CB04	RLC H			
•	CB05	RLC L			
•	07	RLCA	หมุนข้อมูลใน A ไปทางซ้าย	1	4
•	ED6F	RLD		5	18
•	CB1E	RR (HL)		4	15
•	DDCB■ 1E	RR (IX+d)		6	23
•	FDCB■ 1E	RR (IY+d)			
•	CB1F	RRA		2	8
•	CB18	RR B			

ที่ตำแหน่งไม่เป็น "0"

หมุนข้อมูลในหน่วยความจำที่กำหนดไปทางซ้าย

หมุนข้อมูลใน A ไปทางซ้าย

หมุนข้อมูลในหน่วยความจำไปทางซ้าย

หมุนข้อมูลในรีจิสเตอร์ไปทางซ้าย

หมุนข้อมูลใน (HL) ไปทางซ้าย

หมุนข้อมูลในหน่วยความจำไปทางขวา

(7) เมื่อเงื่อนไขถูกต้องจะมีการทำงานเพิ่มขึ้นไปอีกหนึ่งรอบ (8) เมื่อเงื่อนไขไม่ถูกต้องหรือไม่เป็นไปตามเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	การเขียนคำสั่ง	รหัสที่ใช้ในหน่วยความจำ	การทำงาน	M	T
•	CB19	RR C	หมุนข้อมูลในรีจิสเตอร์ไปทางขวา	2	8
•	CB1A	RR D			
•	CB1B	RR E			
•	CB1C	RR H			
•	CB1D	RR L			
•	1F	RRA	หมุนข้อมูลใน A ไปทางขวา	1	4
•	CB0E	RRC (HL)	หมุนข้อมูลในหน่วยความจำไปทางขวา	4	15
•	DDCB OE	RRC (IX+d)		6	23
•	FDCB OE	RRC (IY+d)			
•	CB0F	RRC A	หมุนข้อมูลในรีจิสเตอร์ไปทางขวา	2	8
•	CB08	RRC B			
•	CB09	RRC C			
•	CB0A	RRC D			
•	CB0B	RRC E			
•	CB0C	RRC H			
•	CB0D	RRC L			
•	0F	RRCA	หมุนข้อมูลใน A ไปทางขวา	1	4
•	ED67	RRD		5	18
	C7	RST 00H	ค่าเริ่มสภาพ (restart)	3	11
	CF	RST 08H			
	D7	RST 10H			
	DF	RST 18H			
	E7	RST 20H			
	EF	RST 28H			
	F7	RST 30H			
	FF	RST 38H			
•	9E	SBC A, (HL)	A←A−(HL)−CY	2	7
•	DD9E	SBC A, (IX+d)	A←A−(IX+d)−CY	5	19
•	FD9E	SBC A, (IY+d)	A←A−(IY+d)−CY		
•	9F	SBC A, A	A←A−A−CY	1	4
•	98	SBC A, B	A←A−B−CY		
•	99	SBC A, C	A←A−C−CY		
•	9A	SBC A, D	A←A−D−CY		
•	9B	SBC A, E	A←A−E−CY		
•	9C	SBC A, H	A←A−H−CY		
•	9D	SBC A, L	A←A−L−CY		
•	DE	SBC A, n	A←A−n−CY	2	7
•	ED42	SBC HL, BC	HL←HL−BC−CY	4	15
•	ED52	SBC HL, DE	HL←HL−DE−CY		
•	ED62	SBC HL, HL	HL←HL−HL−CY		
•	ED72	SBC HL, SP	HL←HL−SP−CY		
•	37	SCF	CY←1	1	4
	CBC6	SET 0, (HL)		4	15
	DDCB C6	SET 0, (IX+d)		6	23
	FDCB C6	SET 0, (IY+d)			
	CBC7	SET 0, A		2	8
	CBC0	SET 0, B			
	CBC1	SET 0, C			
	CBC2	SET 0, D			
	CBC3	SET 0, E			
	CBC4	SET 0, H			
	CBC5	SET 0, L			
	CBCE	SET 1, (HL)		4	15
	DDCB CE	SET 1, (IX+d)		6	23
	FDCB CE	SET 1, (IY+d)			
	CBCF	SET 1, A		2	8
	CBC8	SET 1, B			
	CBC9	SET 1, C			
	CBCA	SET 1, D			
	CBCB	SET 1, E			
	CBCC	SET 1, H			
	CBCD	SET 1, L			
	CBD6	SET 2, (HL)		4	15
	DDCB D6	SET 2, (IX+d)		6	23
	FDCB D6	SET 2, (IY+d)			
	CBD7	SET 2, A		2	8
	CBD0	SET 2, B			
	CBD1	SET 2, C			
	CBD2	SET 2, D			
	CBD3	SET 2, E			
	CBD4	SET 2, H			
	CBD5	SET 2, L			
	CBDE	SET 3, (HL)		4	15
	DDCB DE	SET 3, (IX+d)		6	23
	FDCB DE	SET 3, (IY+d)			
	CBDF	SET 3, A		2	8
	CBD8	SET 3, B			
	CBD9	SET 3, C			
	CBDA	SET 3, D			
	CBDB	SET 3, E			
	CBDC	SET 3, H			
	CBDD	SET 3, L			
	CBE6	SET 4, (HL)		4	15
	DDCB E6	SET 4, (IX+d)		6	23
	FDCB E6	SET 4, (IY+d)			
	CBE7	SET 4, A		2	8
	CBE0	SET 4, B			
	CBE1	SET 4, C			
	CBE2	SET 4, D			
	CBE3	SET 4, E			
	CBE4	SET 4, H			
	CBE5	SET 4, L			
	CBEE	SET 5, (HL)		4	15
	DDCB EE	SET 5, (IX+d)		6	23
	FDCB EE	SET 5, (IY+d)			
	CBEF	SET 5, A		2	8
	CBE8	SET 5, B			
	CBE9	SET 5, C			
	CBEA	SET 5, D			
	CBEB	SET 5, E			
	CBEC	SET 5, H			
	CBED	SET 5, L			
	CBF6	SET 6, (HL)		4	15
	DDCB F6	SET 6, (IX+d)		6	23
	FDCB F6	SET 6, (IY+d)			
	CBF7	SET 6, A		2	8
	CBF0	SET 6, B			
	CBF1	SET 6, C			
	CBF2	SET 6, D			
	CBF3	SET 6, E			
	CBF4	SET 6, H			
	CBF5	SET 6, L			
	CBFE	SET 7, (HL)		4	15
	DDCB FE	SET 7, (IX+d)		6	23
	FDCB FE	SET 7, (IY+d)			
	CBFF	SET 7, A		2	8
	CBF8	SET 7, B			
	CBF9	SET 7, C			
	CBFA	SET 7, D			
	CBFB	SET 7, E			
	CBFC	SET 7, H			
	CBFD	SET 7, L			
•	CB26	SLA (HL)		4	15
•	DDCB 26	SLA (IX+d)		6	23
•	FDCB 26	SLA (IY+d)			
•	CB27	SLA A		2	8
•	CB20	SLA B			
•	CB21	SLA C			
•	CB22	SLA D			
•	CB23	SLA E			
•	CB24	SLA H			
•	CB25	SLA L			
•	CB2E	SRA (HL)		4	15
•	DDCB 2E	SRA (IX+d)		6	23
•	FDCB 2E	SRA (IY+d)			
•	CB2F	SRA A		2	8
•	CB28	SRA B			
•	CB29	SRA C			
•	CB2A	SRA D			
•	CB2B	SRA E			
•	CB2C	SRA H			
•	CB2D	SRA L			
•	CB3E	SRL (HL)		4	15
•	DDCB 3E	SRL (IX+d)		6	23
•	FDCB 3E	SRL (IY+d)			
•	CB3F	SRL A		2	8
•	CB38	SRL B			
•	CB39	SRL C			
•	CB3A	SRL D			
•	CB3B	SRL E			
•	CB3C	SRL H			
•	CB3D	SRL L			
•	96	SUB (HL)	A←A−(HL)	2	7
•	DD96	SUB (IX+d)	A←A−(IX+d)	5	19
•	FD96	SUB (IY+d)	A←A−(IY+d)		
•	97	SUB A	A←A−A	1	4
•	90	SUB B	A←A−B		
•	91	SUB C	A←A−C		
•	92	SUB D	A←A−D		
•	93	SUB E	A←A−E		
•	94	SUB H	A←A−H		
•	95	SUB L	A←A−L		
•	D6	SUB n	A←A−n	2	7
•	AE	XOR (HL)	A←A⊕(HL)	2	7
•	DDAE	XOR (IX+d)	A←A⊕(IX+d)	5	19
•	FDAE	XOR (IY+d)	A←A⊕(IY+d)		
•	AF	XOR A	A←A⊕A	1	4
•	A8	XOR B	A←A⊕B		
•	A9	XOR C	A←A⊕C		
•	AA	XOR D	A←A⊕D		
•	AB	XOR E	A←A⊕E		
•	AC	XOR H	A←A⊕H		
•	AD	XOR L	A←A⊕L		
•	EE	XOR n	A←A⊕n	2	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้