

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

COMPUTER CONTROL DIMMER 64 CHANNEL

คอมพิวเตอร์ควบคุมการหรี่ไฟ 64 ช่อง



นาย ยงเกียรติ สันทะสอศิษฐ์
นาย วัฒนชัย ธรรมสาโรช
นาย สันทัศน์ อุดมฤกษ์ชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....

เลขทะเบียน.....34002

วัน, เดือน, ปี 27. 0. 2542

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

หากมีการเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

โดย

อาจารย์ที่ปรึกษา

ภาควิชา

ปีการศึกษา

คอมพิวเตอร์ควบคุมการหรีไฟ 64 ช่อง

นาย ยงเกียรติ สมนเทศอดิษฐ์

นาย วัฒนชัย ธรรมสาโรช

นาย สันทัศน์ อุดมฤกษ์ชัย

ผศ. อุทัย ศรีธีระวิโรจน์

อาจารย์ มนชนก ศรีเสือขาม

เทคนิคอุตสาหกรรม

2541

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับ
ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ

()

.....กรรมการ

()

.....กรรมการ

()

.....กรรมการ

()

.....กรรมการ

()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอรืควบคุมการหรีไฟ 64 ช่อง

นาย ขงเกียรติ สานเทศอดิศจัย

นาย วัฒนชัย ธรรมสาโรช

นาย สันทัศน์ อุดมฤกษ์ชัย

อาจารย์ที่ปรึกษา

ผศ. อุทัย ศรีธีระวิโรจน์

อ. มนชนก ศรีเลื้อขาม

บทคัดย่อ

ปริญญานิพนธ์นี้ได้มีการนำเอาคอมพิวเตอรืมาใช้งานเพื่อควบคุมการหรีไฟการควบคุมด้วยคอมพิวเตอรืนั้นเป็นที่นิยมกันมากในปัจจุบัน การควบคุมการหรีไฟด้วยคอมพิวเตอรืนั้นสามารถที่จะควบคุมด้วยโปรแกรม ถ้าต้องการที่จะแก้ไขก็จะแก้ไขเพียงบางส่วนที่เป็นโปรแกรมเท่านั้น และสามารถที่จะควบคุมได้ครั้งละหลายๆช่อง

ปริญญานิพนธ์นี้จัดทำชุดการคอินเตอรืเฟสซึ่งในการคอินเตอรืเฟสนั้นจะประกอบด้วยชุดคิโค้ดแอดเครต และวงจรแปลงสัญญาณคิจิตอลเป็นอนาล็อก และหลังจากที่แปลงสัญญาณแล้วจะส่งค่าสัญญาณอนาล็อกนั้นไปเพื่อมัลติเพล็กซ์ช่องสัญญาณจาก 1 ช่องให้เป็น 16 ช่อง ดังนั้นจะประกอบด้วยชุดมัลติเพล็กซ์ทั้งหมด 4 ชุดรวมจะได้ทั้งหมด 64 ช่อง โดยแต่ละช่องนั้นสามารถที่ปรับและเปลี่ยนค่าได้ตั้งแต่ 0-10 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPUTER CONTROL DIMMER 64 CHANNEL

MR. YONGKIAT SONTESADISAI

MR. WATCHAI THAMASAROJ

MR. SANTAD UDOMLERGCHAI

ADIVISOR

ASST. PROF. UTHAI SRITEERAWIROJ

MISS MONCHANOK SRISUAKAM

ABSTRACT

Computer system was used in preparation of this thesis to control dimmer. Nowadays controlled by computer system was very popular it could control by using this program. If it need any change it could be done at the program and program can control several channels at a time.

This thesis made interphase card which decode address and had digital to analog convert circuit. After that analog signal would be sent for multiplex a channel to be 16 channel. So there were four multiplex set totalling 64 channels and each channel could adapt its value from 0 to 10 volt.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความช่วยเหลือจากท่านอาจารย์ที่
 ปรึกษา และ อาจารย์ประจำภาควิชาเทคนิคอุตสาหกรรมทุกท่าน และได้รับข้อมูลต่างๆทางด้าน
 เทคนิคจาก ผศ. อุทัย ศรีธีระวิโรจน์ซึ่งเป็นอาจารย์ที่ปรึกษา ขอขอบคุณ คุณ เอกชัย ที่ให้คำปรึกษา
 ทางด้านฮาร์ดแวร์และซอฟต์แวร์ ขอขอบคุณ คุณ รัชฎา น้อยเคลือบ ที่ช่วยเหลือทางด้านการพิมพ์ และ
 ขอขอบคุณเพื่อนๆทุกคนที่ทำให้กำลังใจตลอดจนปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	V
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีไมโครคอมพิวเตอร์และการอินเทอร์เน็ต	4
2.1 ทฤษฎีไมโครคอมพิวเตอร์	4
2.2 การจัดแอดเดรสสำหรับพอร์ทใน IBM PC	9
2.3 เทคนิคการดีโค้ดแอดเดรส	11
2.4 การ์ดอินเทอร์เน็ต	13
บทที่ 3 วงจรแปลงสัญญาณ ดิจิตอลเป็นอนาล็อก	17
3.1 วงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อก	17
3.2 การคำนวณเอาต์พุตของ D/A	21
3.3 ลักษณะสมบัติของดีทูเอ	22
3.4 ลักษณะวงจรดีทูเอที่นำไปใช้งานจริง	23
บทที่ 4 วงจรมัลติเพล็กซ์ช่องสัญญาณ	25
4.1 หลักการทำงานของวงจรมัลติเพล็กซ์	25
4.2 วงจรเก็บระดับแรงไฟ	27
บทที่ 5 โปรแกรมควบคุมการทำงาน	29
5.1 หลักการใช้งานของโปรแกรม	29
5.2 โครงสร้างโปรแกรม (Flow chat)	31
5.3 ส่วนต่างๆ ของโปรแกรม	37
บทที่ 6 สรุปผลการทดลองและวิจารณ์	61
6.1 สรุปผลการทดลอง	61
6.2 แนวทางการพัฒนา	61
บรรณานุกรม	62
ภาคผนวก	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปภาพ	หน้า
บทที่ 1 บทนำ	
รูปที่ 1.3 BLOCK DIAGRAM แสดงการทำงาน	3
บทที่ 2 ทฤษฎีไมโครคอมพิวเตอร์และการอินเตอร์เฟส	
รูปที่ 2.1 โครงสร้าง 8088 ระบบบัสของ CPU	4
รูปที่ 2.2 แสดงสายสัญญาณเชื่อมต่อสล็อต	5
รูปที่ 2.3 แสดงพื้นฐานการตีโค้ด	10
รูปที่ 2.4 โครงสร้าง Single Port Single Address	12
รูปที่ 2.5 โครงสร้าง Single Port Multiple Address	12
รูปที่ 2.6 วงจร Multiple Port Single Address	13
รูปที่ 2.7 โครงสร้างของ Multiple Port Multiple Address	13
รูปที่ 2.8 ลักษณะวงจรภายในของ IC 74LS688	14
รูปที่ 2.9 แสดงวงจรอินเตอร์เฟส	16
บทที่ 3 วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก	
รูปที่ 3.1 วงจรคิทูเอแบบใช้ตัวต้านทานหลายค่า	18
รูปที่ 3.2 วงจรเปลี่ยนสัญญาณแบบ R/2R แลคเดอร์	19
รูปที่ 3.3 โครงสร้างภายใน MC 1408L	20
รูปที่ 3.4 บล็อกไดอะแกรมและการต่อใช้งานของ MC 1408L	21
รูปที่ 3.5 ความผิดพลาดที่เกิดขึ้นในการเปลี่ยนสัญญาณ	23
รูปที่ 3.6 แสดงวงจรคิทูเอที่ใช้งานจริง	24
บทที่ 4 วงจรมัลติเพล็กซ์ของสัญญาณ	
รูปที่ 4.1 วงจรภายในของไอซี CD 4067 หรือ MC 14067	25
รูปที่ 4.2 แสดงวงจรมัลติเพล็กซ์ 16 ช่อง	27
บทที่ 5 โปรแกรมควบคุมการทำงาน	
รูปที่ 5.1 รูปแสดงเมนูหลักทั้งหมด	52

สารบัญตาราง

ตาราง	หน้า
บทที่ 2 ทฤษฎีไมโครคอมพิวเตอร์และการอินเตอร์เฟส	
ตารางที่ 2.1 การใช้งานแอดเดรสสำหรับพอร์ทบน IBM PC	9
บทที่ 4 วงจรมัลติเพล็กซ์ช่องสัญญาณ	
ตารางที่ 4.1 แสดงการทำงานของ MC 14067	26



บทที่ 1

บทนำ

ในปัจจุบัน ความเจริญก้าวหน้าทางเทคโนโลยี ได้พัฒนาให้เครื่องมือและเครื่องใช้ต่างๆ เพื่ออำนวยความสะดวกแก่ผู้ใช้เป็นอย่างมาก ให้ความรวดเร็วในการใช้งานเป็นอย่างสูง โดยการใช้คอมพิวเตอร์ในการควบคุมการทำงาน เพื่อเอื้ออำนวยความสะดวก

DIMMER คือ อุปกรณ์ทางอิเล็กทรอนิกส์ชนิดหนึ่ง ซึ่งทำหน้าที่ปรับระดับแรงดันไฟฟ้าให้ แก่หลอดไฟเพื่อให้ได้ความสว่างตามต้องการแก่ผู้ใช้งาน

COMPUTER CONTROL DIMMER ก็เช่นกันนำเอาความก้าวหน้าทางเทคโนโลยีมาประยุกต์ ใช้งานในการควบคุมการหรี่ไฟของหลอดไฟก็จะทำการควบคุมปริมาณความสว่างของหลอดไฟให้ได้ ความสว่างตามต้องการ ซึ่งจะอำนวยความสะดวกในการใช้งานเป็นอย่างมาก

ปริญญาานิพนธ์ฉบับนี้ได้จัดทำการส่งข้อมูลออกจากพอร์ทคอมพิวเตอร์เพื่อแปลงสัญญาณจาก คิจิตอล เป็นอนาล็อกเพื่อที่จะเป็นการปรับแรงดันไฟฟ้าตั้งแต่ 0-10 V และ สามารถที่จะนำความรู้ไปปรับ ประยุกต์และนำไปประยุกต์ใช้งานให้เกิดประโยชน์มากยิ่งขึ้น

ในปริญญาานิพนธ์ฉบับนี้ ได้กล่าวเนื้อหาต่างๆดังต่อไปนี้

บทที่ 1 ได้กล่าวนำถึงความเป็นมาและอธิบายเนื้อหาคร่าวๆ ในแต่ละบทของปริญญาานิพนธ์ฉบับ นี้ รวมถึงประโยชน์และหลักการทำงานเบื้องต้น

บทที่ 2 ทฤษฎีของ MICROCOMPUTER 8088 เป็นการนำทฤษฎีและหลักการในการที่จะนำ ไปประยุกต์ใช้ในปริญญาานิพนธ์นี้ และ ทฤษฎีการอินเตอร์เฟส โดยจะกล่าวถึงการอินเตอร์เฟสเบื้องต้นและ วงจรอินเตอร์เฟสในปริญญาานิพนธ์นี้ โดยใช้ IC 74LS688 ในการ ดีโค้ด

บทที่ 3 วงจรแปลงสัญญาณ คิจิตอลเป็นอนาล็อก โดยจะกล่าวถึงหลักการเบื้องต้นของวงจร แปลงสัญญาณ คิจิตอลเป็นอนาล็อก และวงจรที่ใช้ในปริญญาานิพนธ์ฉบับนี้ โดยจะใช้ IC MC1408 หรือ IC DAC0800

บทที่ 4 การ Multiplex โดยจะกล่าวถึงการ Multiplex จาก อินพุต 1 ให้เอาท์พุต เป็น 16 โดย ใช้ IC NC4067 ในการ Multiplex

บทที่ 5 การใช้งาน และ ซอฟแวร์ จะเป็นการกล่าวถึงการใช้งานของ ซอฟแวร์ และ โปรแกรม ของปริญญาานิพนธ์นี้ เป็นการอธิบายถึงการใช้งานและ Flowchat ของโปรแกรมทั้งหมดเพื่อที่จะให้เข้าใจเกี่ยวกับโปรแกรม

บทที่ 6 บทสรุปและแนวทางการในการพัฒนา จะเป็นการสรุปผลการทดลองและ ปัญหาที่เกิดขึ้นรวมถึงแนวทางเพื่อที่จะพัฒนาของโครงการนี้

ในภาคผนวก ได้มีรายละเอียดของรายการอุปกรณ์ที่ใช้ในโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการนำไปใช้งาน

จากระบบการควบคุมสามารถควบคุมไฟได้หลายช่อง โดยใช้ที่จะใช้ จำนวนสายนั้นน้อยมาก และสามารถจะส่งได้ระยะทางไกล เพื่อที่จะเป็นการประหยัดสายโดยเราจะไม่ต้องใช้สายจำนวนมากตามจำนวนช่องและยังสามารถแสดงผลทางจอโมนิเตอร์ ซึ่งสามารถควบคุมได้ตามความพอใจ

ประโยชน์และการนำไปใช้งาน

1. เสริมสร้างบรรยากาศ และสามารถที่จะปรับความสว่างได้ตามความพอใจ

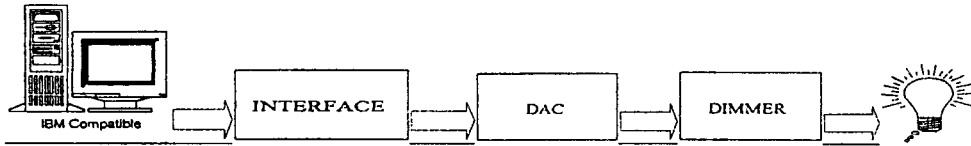
2. ใช้ในห้องประชุมเพื่อปรับแสงสว่างเพื่อให้เหมาะกับงานที่จะทำ เช่นในกรณีสัมมนา จะเปิดไฟเต็มที่ ใช้ในการ lecture หรือ กรณีต้องการฉายสไลด์ที่จะต้องลดความสว่างลง แต่ในบางส่วน ก็ให้มีแสงสว่างพอที่จะ lecture ได้ในขณะที่ กำลังฉายสไลด์ เช่น ข้างหน้าจะมีค้างหลังจะพอดีแสงได้ โดยที่เราสามารถปรับความสว่างได้ตั้งแต่ 0-10v

3. ในงานเวทีจะเน้นไปในทางการสร้างรูปแบบของแสงจะแตกต่างจากการแสดงละครในโรงงาน จะใช้ในรูปแบบการประหยัดพลังงาน ช่วยลดค่าแอร์ ได้มาก

4. ในโรงภาพยนตร์นั้นเวลาที่มีการฉาย ภาพยนตร์อยู่นั้นเราก็สามารถที่จะควบคุมไฟให้มีแสงสว่างน้อยหรืออาจจะใช้ดับไฟเลยก็ได้

5. สามารถที่จะนำไปควบคุมไฟ คิสโก้ได้โดยใช้หลักการไฟกระพริบ

6. สามารถที่จะ memory ได้เพื่อที่สะดวกในการใช้งาน



รูปที่ 1.1 BLOCK DIAGRAM การทำงาน

หลักการทำงานเบื้องต้นของคอมพิวเตอร์ควบคุมไฟ

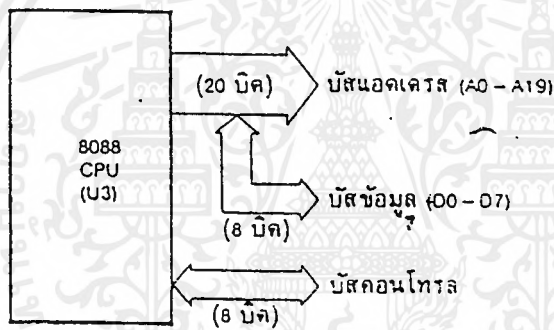
จาก Block Diagram Computer จะเป็นตัว Run Program โดยใช้โปรแกรมภาษาเคลฟเป็นโปรแกรมที่ใช้ในระบบวินโดวส์ ในการเขียนโปรแกรม Software ควบคุมทาง Hardware โดยตัวโปรแกรมจะเป็นส่วนกราฟฟิกที่จะใช้ควบคุมไฟโดยที่จะเป็น สไลด์เลื่อนตั้งแต่ 0-10 v โดยจะมีทั้งหมด 64 ช่องโดยจะสามารถที่จะเล่นได้ทั้งกำหนดรูปแบบเองและสามารถ memory ได้ซึ่งรูปแบบนั้นจะสามารถควบคุมโดยใช้เลื่อนครั้งละ 16 ช่อง ซึ่งจะส่งข้อมูลเป็น Digital ออกไปยังวงจรแปลงสัญญาณดิจิตอล เป็นอนาล็อก

บทที่ 2

ทฤษฎีไมโครคอมพิวเตอร์และการอินเตอร์เฟส

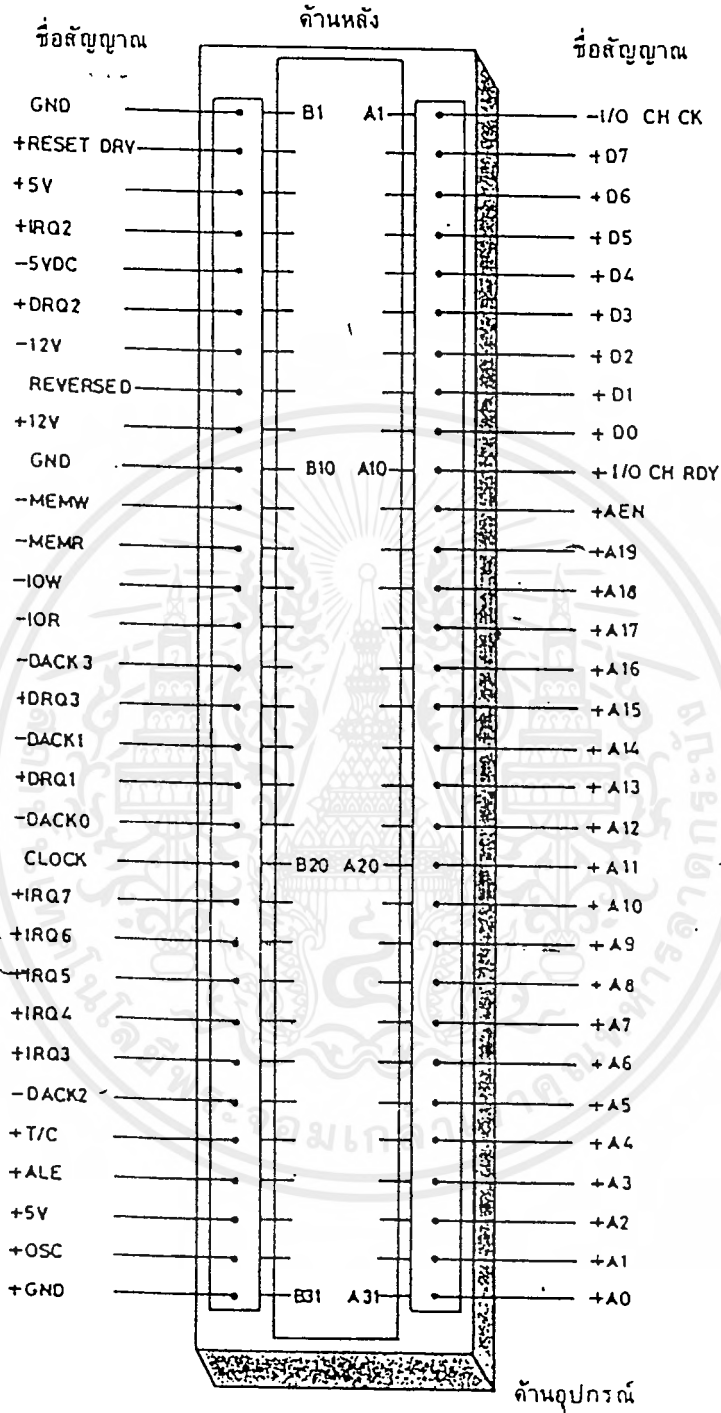
2.1 ทฤษฎีไมโครคอมพิวเตอร์

8088 เป็นซีพียูที่ได้รับการพัฒนาให้มีการรับส่งข้อมูลผ่านบัสข้อมูลเพียง 8 bit โครงสร้างของบัสที่ออกจาก CPU จะแยกออกเป็น 3 กลุ่ม คือ บัสข้อมูล (Data bus) บัสแอดเดรส และบัสควบคุม ข้อมูลแยกออกมาจากแอดเดรสโดยมีขนาดเพียง 8 bit ส่วนบัสควบคุมก็มี 8 bit เช่นกัน แต่สำหรับแอดเดรสจะมีทั้งหมด 20 bit



รูปที่ 2.1 โครงสร้าง 8088 ระบบบัสของ CPU

ระบบบัสทั้งหมดของ IBM PC ทุกเครื่องจะมีการผ่านบัฟเฟอร์อีกครั้ง โครงสร้างของบัสเมื่อเขียนเป็นโคแธแกรมได้ดังรูปที่ 2.1 โครงสร้างของบัสจะเชื่อมต่อเพื่อให้ผู้ใช้สามารถขยายเพิ่มเติมได้ โดยมีการต่อเชื่อมกับสล็อต โดยมีบัสข้อมูลสองทิศทาง 8 bit แอดเดรส 20 bit มีอินเทอร์รัพต์ 6 ระดับ มีส่วนสัญญาณควบคุมสำหรับการเขียนอ่าน input/output และหน่วยความจำ มีสัญญาณนาฬิกา และสายสัญญาณกำหนดเวลา มีเซนเนลการกำหนด DMA มีสัญญาณการควบคุมการรีเฟรชหน่วยความจำ มีสายตรวจสอบสล็อต สายไฟเลี้ยงขนาดของสล็อตมีขนาด 62 ขา โดยมีการจัดเรียงตำแหน่งขา ดังรูปที่ 2.2



รูปที่ 2.2 แสดงสายสัญญาณเชื่อมต่อที่สล๊อต

รายละเอียดเกี่ยวกับสัญญาณต่างๆ

OSC (Oscillator ; ขา B 30)

ขานี้เป็นเอาต์พุตที่เชื่อมต่อกับสัญญาณคล็อกที่มีค่าความถี่สูงสุดบนเมนบอร์ด คือ 14.31818 MHz

CLK (Clock ; ขา B 20)

ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งต่อกับสัญญาณคล็อกที่ถูกสร้างขึ้นโดยการหารสัญญาณ osc ด้วย 3 ทำให้ได้ความถี่ประมาณ 4.77 MHz (14.31818 MHz / 3)

RESET DRV (ขา B 2)

ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งจะแอคทีฟ (ลอจิก “1”) ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอคทีฟไปจนกว่าระบบต่างๆภายใน IBM / PC จะพร้อมที่จะทำงานได้ จากนั้น สัญญาณนี้ก็จะเปลี่ยนกลับเป็นลอจิก “0”

AO-A19 (Address Bus ; ขา A31-A12)

ขาสัญญาณทั้ง 20 ขาเป็นเอาต์พุต ใช้กำหนดแอดเดรสของหน่วยความจำการอ้างแอดเดรสของพอร์ท I/O นั้นจะใช้เส้นแอดเดรสเพียง 16 เส้น คือ AO-A15 ซึ่งจะทำให้อ้างแอดเดรสได้ 64K พอร์ท คือ A16-A19 นั้นจะไม่ถูกใช้งาน ใช้แอดเดรสในการอ้างแอดเดรสของพอร์ทเพียง 10 เส้น คือจาก A0 - A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

DO-D7 (Data Bus ; ขา A9-A2)

ขาสัญญาณนี้จะเป็นแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ท I/O โดยบิต DO จะมีนัยสำคัญต่ำสุดและบิต D7 จะมีนัยสำคัญสูงสุด

ALE (Address Latch Enable ; ขา B28)

ขาสัญญาณนี้เป็นสัญญาณเอาต์พุตที่ 8088 Bus Controller สร้างขึ้นเพื่อใช้สำหรับแสดงการเริ่มต้นของบัสไซเคิล และแสดงให้อุปกรณ์ภายนอกทราบว่าแอดเดรสที่ 8088 ต้องการจะติดต่อด้วยนั้นถูกส่งออกมาบนบัสแอดเดรสแล้ว

I/O CHCK (I/O Channel Check ; ขา A1)

ขาสัญญาณนี้เป็นอินพุตที่ใช้ในการแสดงความผิดพลาดเกี่ยวกับพาริตี ที่เกิดขึ้นในการทำงานของวงจรอินเตอร์เฟสหรืออุปกรณ์ I/O เมื่อขาสัญญาณนี้ได้รับลอจิก “0” จะทำให้ 8088 ถูกอินเตอร์รัพท์แบบ Non-Maskable (NMI)

Enable : ใช้ คำสั่ง OUT ส่งข้อมูล 80H ไปยังพอร์ท 00A0H

Disable : ใช้ คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

I/O CHRDY (I/O Channel Ready ; ขา A10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาสัญญาณนี้เป็นอินพุตที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อุปกรณ์ I/O หรือหน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น

IRQ2-IRQ7 (Interrupt Request 2 Through 7 ; ขา B4 และ B25-B21)

ขาสัญญาณทั้ง 6 นี้เป็นขาอินพุตที่ใช้สำหรับการขออินเตอร์รัพท์จาก 8088 โดยสัญญาณเหล่านี้จะต่อเข้ากับ 8259A บนเมนบอร์ดโดยตรง

IOR (I/O Read ; ขา B14)

ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ท

IOW (I/O Write ; ขา B13)

ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่งสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็น บัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น

MEMW (Memory Write ; ขา B11)

ขานี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่ง 8288 Bus Controller สร้างขึ้นในระหว่างบัสไซเคิลในการเขียนข้อมูลลงในหน่วยความจำของ 8088

MEMR (Memory Read ; ขา B12)

ขานี้เป็นเอาต์พุตจาก 8288 ซึ่งสัญญาณนี้จะแอกทีฟ (ลอจิก “0”) การอ่านข้อมูลจากหน่วยความจำของ 8088 เพื่อให้หน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น

DRQ1-DRQ3 (DMA Request 1-3 ; ขา B18 , B6 , และ ขา B16)

ขาสัญญาณทั้งสามนี้เป็นสัญญาณอินพุตแอกทีฟที่ลอจิก “1” ซึ่งอุปกรณ์ภายนอกสามารถใช้ในการขอ DMA จากระบบ โดยการป้อนระดับสัญญาณลอจิก “1” ให้กับขา DRQ ขาใดขาหนึ่ง (ขา DRQ ทั้งสามนี้จะต่อเข้ากับ DRQ1-DRQ3 ของ 8237A-5)

DACK0-DACK3 (DMA Acknowledge 0-3 ; ขา B19 , B17 , B26 และ B15)

สัญญาณทั้ง 4 นี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้วงจรภายนอกที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์ I/O ที่ขอ DMA

AEN (Address Enable ; ขา A11)

สัญญาณนี้เป็นเอาต์พุตที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก "1") นั้นเป็นบัสไซเคิลของขบวนการ DMA

T/C (Terminal Count ; ขา B27)

สัญญาณนี้ถูกสร้างขึ้นจากการนำเอาสัญญาณเอาต์พุตที่ขา EOP ของ 8237A-5 มากลับลอจิก (โดยใช้เกต Inverter) ทำให้สัญญาณ T/C นี้แอคทีฟที่ลอจิก "1"

บัสของแหล่งจ่ายไฟของระบบ

+5Vdc (ขา B3 และ B29)

ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) +5% คืออยู่ในช่วง +4.75 ถึง +5.25 Vdc

+12Vdc (ขา B9)

ขานี้จะต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) + 5% คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

-5Vdc (ขา B5)

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) +10% คืออยู่ในช่วง -5.5 ถึง -4.5 Vdc

-12Vdc (ขา B7)

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) +10% คืออยู่ในช่วง -13.2 ถึง -10.8 Vdc

GND (ขา B1, B10 และ B31)

ขาทั้งสามนี้จะต่อเข้ากับกราวด์ (Ground) ของระบบ

2.2 การจัดแอดเดรสสำหรับพอร์ตใน IBM PC

ในการควบคุมและตรวจสอบสถานะการทำงาน รวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็น ชิปขับพอร์ต หรือการ์ดต่าง ๆ ที่ใช้ในระบบของ IBM PC นั้นจะกระทำโดยผ่านทางพอร์ตไอโอของระบบ ดังนั้นในการที่จะใช้งาน หรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ตไอโอต่าง ๆ ของระบบด้วย และเนื่องจากการควบคุม หรือติดต่อกับพอร์ตเหล่านี้ ต้องกระทำโดยการอ้างถึงแอดเดรสของพอร์ตไอโอ เหล่านั้นโดยตรงเราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสบน IBM PC ด้วย IBM PC ถูกออกแบบมาให้ใช้แอดเดรสของไอโอพอร์ตได้ 10 บิต คือ A0-A9 เท่านั้น จึงสามารถที่จะอ้างแอดเดรสของไอโอพอร์ตสูงสุดถึง 1024 พอร์ต ใน 1024 พอร์ตนี้ยังแบ่งออกเป็น “ 0 ” เราจะอ่านข้อมูลเฉพาะพอร์ตของชิปขับพอร์ตต่างๆ ที่อยู่บนเมนบอร์ดของ IBM PC แต่ถ้า A9 เป็น “ 1 ” ก็จะอ่านข้อมูลได้เฉพาะจากพอร์ตของการ์ดต่าง ๆ เท่านั้น

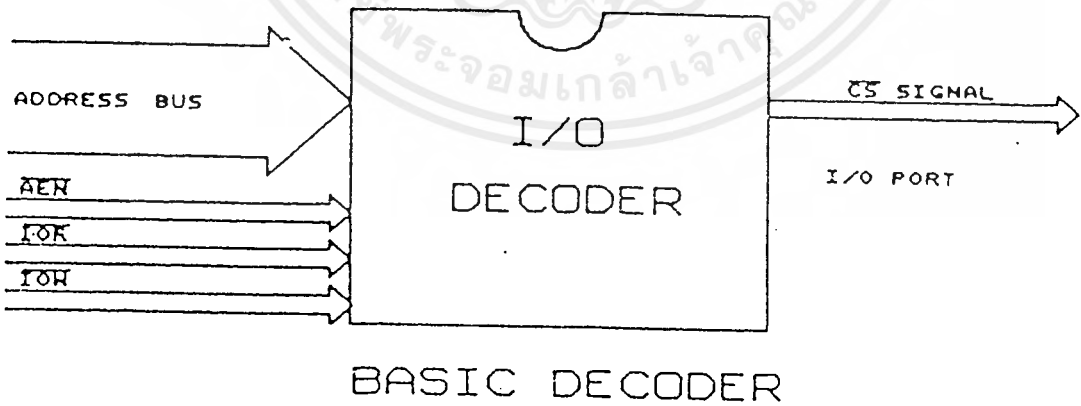
ดังนั้นเราต้องการสร้างแผนเชื่อมโยงกับ IBM/PC เลือกแอดเดรสสำหรับพอร์ตที่อยู่บนการ์ดตามที่แสดงไว้ในตารางที่ 2.1 สำหรับโครงการปริญญาโทฉบับนี้เราจะเลือกแอดเดรสของพอร์ตช่วง 300-31FH ซึ่งใช้ กับโปรโตไทป์การ์ด ทั้งนี้เพื่อความสะดวกของผู้ออกแบบ และป้องกันการอ้างพอร์ตซ้ำกับการ์ดที่มีอยู่ แต่ก็สามารถใช้งานในเบอร์พอร์ตได้ถ้าไม่มีการดัดแปลงติดตั้งอยู่ในระบบ

ตารางที่ 2.1 การใช้งานแอดเดรสสำหรับพอร์ตบน IBM PC

หมายเลขพอร์ต	ชื่ออุปกรณ์
000H-01FH	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 1 , 8237A-5
020H-03FH	อินเทอร์รัพต์คอนโทรลเลอร์หมายเลข 1 , 8259A
040H-05FH	ไทมเมอร์ 8254-2
060H-06FH	8042 คีย์บอร์ด
070H-07FH	นาฬิกา และ NMI และซิมอสแรม
080H-09FH	DMA เพจรีจิสเตอร์
0A0H-0BFH	อินเทอร์รัพต์คอนโทรลเลอร์หมายเลข 2 , 8259A
0C0H-0DFH	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 2 , 8237A-5
0F0H	เคลียร์โปรเซสเซอร์คณิตศาสตร์
0F1H	รีเซทโปรเซสเซอร์คณิตศาสตร์
0F8H-0FFH	โปรเซสเซอร์คณิตศาสตร์
1F0H-1F8H	ฮาร์ดดิสก์

ตาราง 2.1 การใช้แอดเดรสสำหรับพอร์ตบน IBM PC (ต่อ)

หมายเลขพอร์ต	ชื่ออุปกรณ์
200H-207H	เกมไอโอ
278H-27FH	พอร์ตเครื่องพิมพ์หมายเลข 2
28FH-2FFH	พอร์ตอนุกรมหมายเลข 2
300H-31FH	โปรโตไทป์การ์ด
360H-36FH	ลำรอง
378H-37FH	พอร์ตเครื่องพิมพ์หมายเลข 1
380H-38FH	SDLC, ไบซิงค์ 2
3A0H-3ARH	โมโนโครม และเครื่องพิมพ์
3C0H-3CFH	ลำรอง
3F0H-3F7H	ควบคุมดิส,ดิสเกตต์
3F8H-3FFH	พอร์ตอนุกรมหมายเลข 1



2.3 เทคนิคการดีโค้ดแอดเดรส

จากรูปที่ 2.3 แสดงถึงพื้นฐานของการดีโค้ดซึ่งแยกออกเป็น 3 ส่วน ส่วนแรกเป็นการนำแอดเดรส A0-A9 มาทำการดีโค้ดสร้างสัญญาณ ADDRESSD SEL และนำสัญญาณนี้ไปรวมกับ AEN ได้สัญญาณ I/O SELECT เพื่อแยกว่าเป็นอุปกรณ์ไอโอถ้าใช้สัญญาณนี้ร่วมกับ IOR จะได้สัญญาณสำหรับเลือกอุปกรณ์อินพุต IOW จะเป็นสัญญาณสำหรับเลือกอุปกรณ์อินพุต IOW จะเป็นสัญญาณสำหรับเลือกอุปกรณ์เอาต์พุต เทคนิคโดยทั่วไปในการดีโค้ดมีดังนี้

2.3.1 เทคนิคในการดีโค้ดแอดเดรสแบ่งตามการใช้อุปกรณ์

-การดีโค้ดแบบใช้ไอซีลอจิกเกตธรรมดา เช่น AND, OR, NOR, NAND และ NOT GATE ในการดีโค้ดมักใช้เมื่อต้องการพอร์ตเพียงพอร์ตเดียว

-การดีโค้ดแบบ FIXED ใช้ไอซีดีโค้ดเบอร์ต่าง ๆ เช่น 74LS138 เป็นต้น วิธีการดีโค้ดแบบนี้เป็นวิธีที่ง่ายใช้เมื่อต้องการพอร์ตตั้งแต่ 4 พอร์ตขึ้นไป

-การดีโค้ดแบบใช้สวิตช์เลือกโดยใช้ไอซีเปรียบเทียบระหว่าง 2 อินพุต เช่น EXCLUSIVE-OR หรือ ไอซีเบอร์ 74LS688 (8 BIT COMPARATOR) การดีโค้ดทั้งสองแบบแรกที่กำลังกล่าวมามีข้อเสียคือ แอดเดรสที่เราเลือกใช้งานนั้นอาจจะซ้ำกับแอดเดรสของการ์ดอื่นที่เรานำมาเพิ่มเข้าไปในระบบภายหลังก็ได้ และอีกข้อหนึ่งคือ ไม่สามารถเปลี่ยนแปลงแอดเดรส ปัญหาเช่นนี้เราสามารถแก้ไขได้โดยใช้วงจรดีโค้ดที่สามารถเปลี่ยนแปลงค่าแอดเดรสได้โดยเพียงแค่เปลี่ยนตำแหน่งของสวิตช์ (DIP SW.)

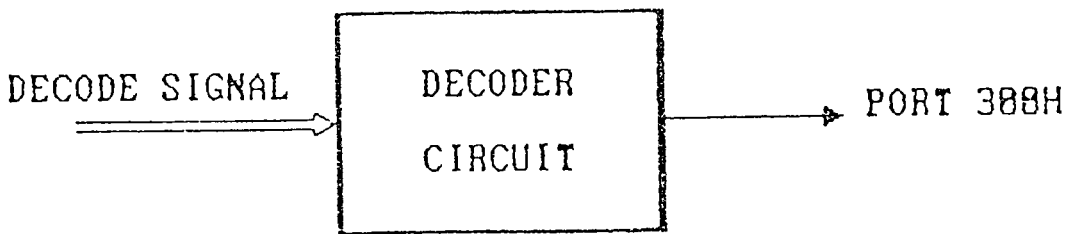
-การดีโค้ดโดยใช้ PROM (Programmable Read Only Memory) มักจะใช้พอร์ตที่เราต้องการมีแอดเดรสแยกกันอย่างอิสระ ต้องมีการโปรแกรมก่อนนำมาใช้งาน หรืออุปกรณ์ที่ทำการดีโค้ดไม่มากเกินไปจะนำไปใช้ดีโค้ด การทำงานที่อยู่บนการ์ดต่าง ๆ มารวมไว้บนการ์ดเดียวกัน

-การดีโค้ดโดยใช้ PAL (Programmable Array Logic) และ PLA (Programmable Logic Array) เทคนิคการดีโค้ดแบบนี้ต้องเขียนอินพุต และเอาต์พุตให้อยู่ในรูปฟังก์ชันลอจิก และนำไปโปรแกรมลงบน PAL ชนิดของ PAL มีให้เลือกตามจำนวนอินพุตเอาต์พุตที่เหมาะสม วิธีการใช้งานสามารถศึกษาได้จากคู่มือ วิธีนี้สามารถลดจำนวนอุปกรณ์วงจรดีโค้ดเหลือเพียง PAL เพียงอย่างเดียว

2.3.2 เทคนิคในการดีโค้ดแบบเลือกพอร์ต

การพิจารณาเลือกเทคนิคใดมาใช้ในการดีโค้ด จึงขึ้นอยู่กับพอร์ตที่เราต้องการมีลักษณะอย่างไร บางครั้งก็จำเป็นต้องนำแต่ละเทคนิคมาใช้ร่วมกัน เพื่อสร้างพอร์ตที่เราต้องการ ซึ่งพอจะแบ่งลักษณะของพอร์ตได้ 4 ลักษณะดังนี้

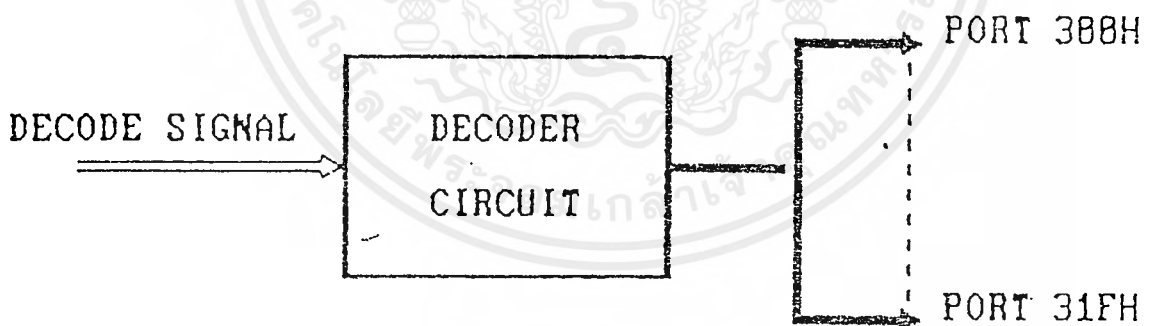
-Single Port Single Address เป็นลักษณะวงจรดีโค้ดที่สร้างพอร์ตขึ้นมาเพียงพอร์ตเดียว และมีแอดเดรสที่จะติดต่อกับพอร์ตนี้เพียงพอร์ตเดียวมีโครงสร้างดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้าง Single Port Single Address

-Single Port Multiple Address เป็นลักษณะวงจรดีโค้ดที่สร้างพอร์ทขึ้นมาพอร์ทเดียวเช่นกัน แต่มีแอดเดรสที่จะติดต่อกับพอร์ทนี้ได้หลายเบอร์ มีโครงสร้างดังรูป 2.5 วงจรนี้เราไม่สามารถนำสัญญาณแอดเดรสมาดีโค้ดทั้งหมดโดยจะกะช่วงแอดเดรสบิตล่างไว้ตัวอย่างเช่น ต้องการดีโค้ดพอร์ทในช่วง 300H ถึง 31FH ดังนั้นสัญญาณลอจิกของแอดเดรสมีดังนี้

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	X	X	X	X	X

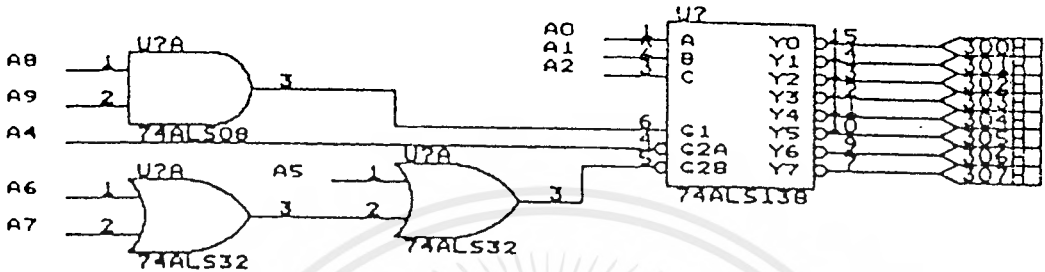


รูปที่ 2.5 โครงสร้าง Single Port Multiple Address

-Multiple Port Single Address เป็นลักษณะวงจรดีโค้ดที่สร้างพอร์ทขึ้นมาหลายพอร์ท และมีแอดเดรสที่จะติดต่อกับพอร์ทนี้เพียงพอร์ทละหนึ่งแอดเดรสเท่านั้น มีโครงสร้างดังรูปที่ 2.5 จากวงจรทั้ง 2 แบบเป็นการดีโค้ดแบบพอร์ทเดียว เรามักใช้ลอจิกเกตธรรมดาในการดีโค้ด แต่ถ้าเราต้องการ

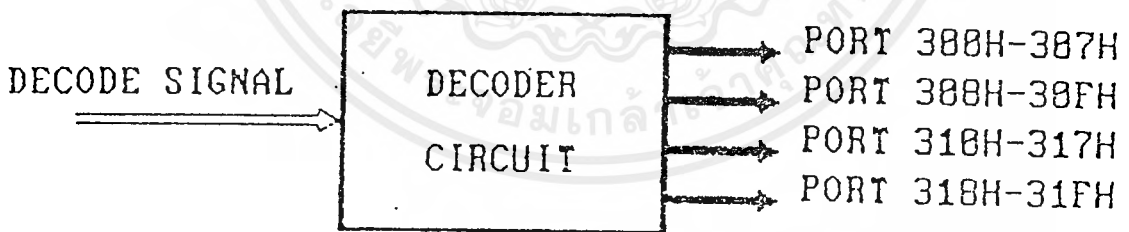
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้งานมากกว่า 1 พอร์ต เช่น 4 พอร์ต เราจะใช้ไอซีดีโค้ดเบอร์ 74LS138 และ 74LS139 เลือกว่าใน การดีโค้ด



รูปที่ 2.6 วงจร Multiple Port Single Address

-Multiple Port Multiple Address เป็นลักษณะดีโค้ดที่สร้างพอร์ตขึ้นมาหลายพอร์ต และมี แอคเครสที่จะติดต่อกับพอร์ตนี้ได้พอร์ตหลายแอสแอสเป็นช่วงแอสแอสมีโครงสร้างดังรูปที่ 2.7

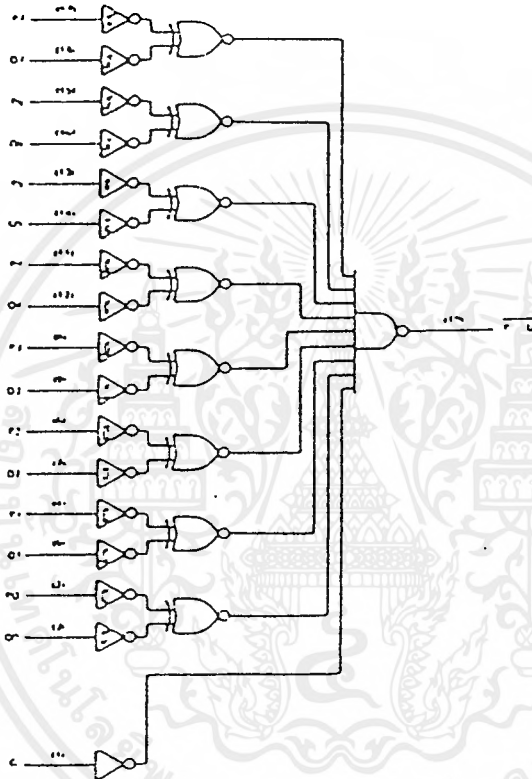


รูปที่ 2.7 โครงสร้างของ Multiple Port Multiple Address

2.4 การ์ดอินเตอร์เฟส

การที่จะดีโค้ดให้ใช้งานได้สะดวกจะต้องใช้การดีโค้ดแบบเปลี่ยนแอสแอส โดยเลือกใช้ อุปกรณ์การดีโค้ด 2 ชนิดเข้าช่วยคือ DIP SWITCH และ IC 74LS688 ไอซีตัวนี้ทำหน้าที่ในการ เปรียบเทียบสัญญาณว่าเหมือนกัน หรือไม่โดยสามารถเปรียบเทียบได้ 8 คู่ โดยอินพุตแต่ละคู่จะอยู่ที่ขา P เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ G เมื่อใดก็ตามที่สัญญาณอินพุตตรงกัน จะให้เอาต์พุตที่ขา 19 เป็น “ 0 ” ละเมื่อสัญญาณที่อินพุตไม่เท่ากัน 74LS688 จะให้เอาต์พุตออกมาเป็น “ 1 ” นอกจากนี้ 74LS688 ยังถูกควบคุมการทำงานโดยคอมพิวเตอร์ โดยจะนำสัญญาณ AEN มาต่อเข้ากับขาที่ 1 ของ 74LS688 จะแอสคิทที่ลอจิก “ 0 ”



รูป 2.8 ลักษณะวงจรภายในของ IC 74LS688

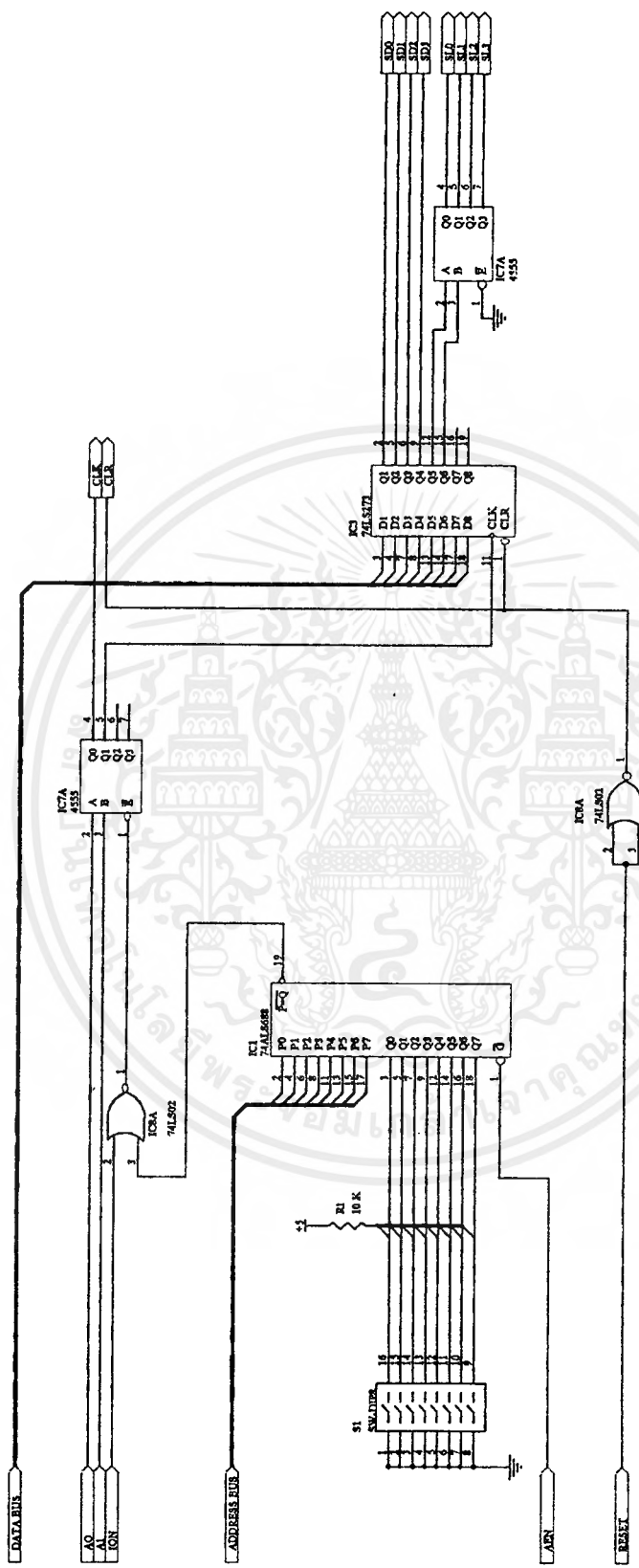
การดีโค้ดพอร์ตจะเลือกแอสคิทที่จะใช้ก่อน การดีโค้ดนี้ใช้แอสคิท 10 บิต (A0-A9) เข้ามาทำการดีโค้ด ส่วนตำแหน่ง A10-A15 นั้นไม่นำมาใช้งานของ 74LS688 นั้นรายละเอียดของขาต่าง ๆ ได้อธิบายไว้แล้วในตอนต้น IC 74LS688 ตัวนี้จะทำการเปรียบเทียบสัญญาณระหว่างอินพุต 2 ส่วนคือ P0 - P7 และ Q0 - Q7 จะให้เอาต์พุตออกมาเป็น “ 0 ” เมื่อ $P = Q$ สัญญาณ Q จะถูกค้อยู่กับ Dip Switch ในการเลือกแอสคิทของการ์ดเลือกโดยเช้ทสวิทซ์ เช่น ในปริณูณานิพนธ์นี้จะใช้แอสคิทช่วง 300-31FH เมื่อมีแอสคิทค่า 300H มาเข้าที่อินพุตของ 74LS688 ที่ขา 19 ของ 74LS688 ให้เอาต์พุต “ 0 ” จะนำไป NOR กับขาของสัญญาณ IOW เพื่อที่จะนำสัญญาณไปที่ IC 4555 ซึ่งทำหน้าที่ ดีโค้ด 4 เอาต์พุต เพื่อที่จะนำสัญญาณจาก Q0 และ Q1 ของ IC 4555 ไปควบคุม IC 74LS273 ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74LS273 ซึ่งทำหน้าที่เป็นบัฟเฟอร์หรือ ดี-ฟลิปฟล็อปก่อนที่จะส่งข้อมูลนั้นไปยังภาควงจรแปลง สัญญาณดิจิตอลเป็นอนาล็อก ส่วนที่ใช้ในการเลือกพอร์ทคือ IC 4555 คือในปริยญาณิพนธ์นี้ใช้ พอร์ท300H

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	0	0

จากแอดเดรสข้างบนนี้จะให้เอาทพุดพอร์ทเป็น 300H คือจะเปลี่ยนที่ A0 และ A1 เท่านั้น ส่วนขาสัญญาณ รีเซ็ต จะไปควบคุมขา CLR ของ IC 74LS273 และ IC74LS688 นั้นขา 1 จะต้องใช้ สัญญาณที่เป็น AEN สภาวะเป็น “0”





รูปที่ 2.9 แสดงวงจรในการอินเตอร์เฟส

บทที่ 3

วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก

3.1 วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก

วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก (Digital to analog) สามารถแบ่งออกเป็น ลักษณะการทำงานและการใช้งาน ที่ใช้กันอยู่ทั่ว ๆ ไป ในปัจจุบัน แบ่งออกได้เป็นแบบต่าง ๆ ดังนี้

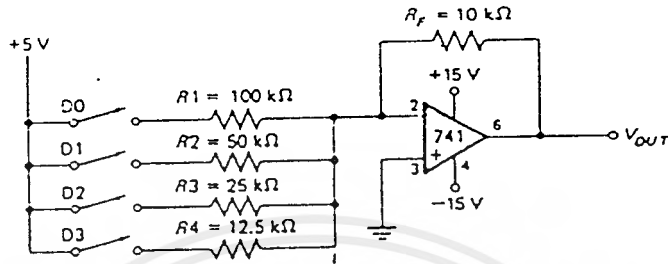
3.1.1 ตัวต้านทานหลายค่า (Binary Weighted Resistor D/A converter)

วงจรเปลี่ยนสัญญาณดิจิทัล (DIGITAL TO ANALOG) ชนิดนี้โดย ใช้ตัวต้านทานต่าง ๆ และออปแอมป์ เพื่อเปลี่ยนระดับสัญญาณลอจิก 2 ระดับ เป็นแรงดันที่ได้สัดส่วนกัน รูปที่ 3.1 แสดงวงจรเปลี่ยนสัญญาณดิจิทัลขนาด 4 บิต ออปแอมป์ที่ใช้มีอัตราขยายสูงมาก (โดยทั่วไปจะสูงกว่า 100,000 เท่า) มีความต้านทานด้านเอาต์พุตต่ำ ความต้านทานด้านอินพุตมีค่าสูงมาก สิ่งสำคัญที่สุดที่จะต้องตระหนักไว้ก็คือ สัญญาณที่เอาต์พุตถูกป้อนกลับมายัง อินพุตแบบกลับเฟส (การป้อนกลับแบบลบ) เพื่อเปรียบเทียบกับสัญญาณที่ขาอินพุตแบบไม่กลับเฟส เอาต์พุตของออปแอมป์จะเป็นตัวจ่ายหรือรับกระแส (Source or sink) เพื่อให้แรงดันที่เปรียบเทียบกับนั้นมีค่าเดียวกัน วงจรในรูปที่ 3.1 ต่อขาไม่กลับเฟสลงกราวด์ ดังนั้นที่ขากลับเฟส ก็จะมีแรงดัน 0 โวลต์ด้วย การที่อินพุตที่สายกลับเฟสเป็น 0 โวลต์ ด้วยโดยไม่ได้อัปเดตกราวด์โดยตรง จึงถูกเรียกว่ากราวด์เทียม (Virtual Ground)

ตอนที่สวิตช์ D0 ปิด ตัวต้านทาน R1 ค่า 100 กิโลโอห์ม จะมีแรงดัน 5 โวลต์ที่ปลายข้างหนึ่ง อีกข้างหนึ่งเป็น 0 โวลต์ (กราวด์เทียม) จากกฎของโอห์มจะมีแรงดันตกคร่อม 5 โวลต์ ซึ่งให้กระแสไหลผ่าน 0.05 มิลลิแอมป์ กระแสนี้ไม่อาจเข้าไปยังอินพุตของออปแอมป์ได้ เนื่องจากออปแอมป์ มีความต้านทานอินพุตสูงมาก และไม่สามารถส่งหรือรับกระแสมาก ๆ ได้ ดังนั้น กระแส 0.05 มิลลิแอมป์ จึงต้องไหลผ่านไปยังเอาต์พุต โดยผ่านตัวต้านทานป้อนกลับ R_F 10 กิโลโอห์ม จะได้แรงดันเอาต์พุต เท่ากับ $(10 \text{ กิโลโอห์ม}) \times (-0.05 \text{ มิลลิแอมป์}) = -0.5 \text{ โวลต์}$ เพื่อรับกระแสผ่านสวิตช์ D0 และรักษาสะถียรภาพกราวด์เทียมไว้ แต่ถ้าหากยังสงสัยในสภาวะกราวด์เทียม ก็ลองวาดเป็นวงจรดิไวเดอร์ที่มีแรงดันข้างหนึ่ง +5 โวลต์ ตรงกลางเป็น 0 โวลต์ และอีกปลายข้างหนึ่งมีค่า -0.5 โวลต์

เมื่อเปิดวงจรที่สวิตช์ D0 และปิดวงจรที่สวิตช์ D1 (ขณะที่ R2 มีค่าเป็นครึ่งหนึ่งของ R1) กระแสเพื่อเป็น 2 เท่าหรือ 0.1 มิลลิแอมป์ ไหลผ่าน R_F , กราวด์เทียม และ R2

ทำให้มีแรงดันเอาต์พุต -1 โวลต์ ต่อไปที่ปิดวงจรทั้งที่ D_0, D_1 จะได้กระแส 0.05 มิลลิแอมป์ไหลผ่าน R_1 และ 0.1 มิลลิแอมป์ ผ่าน R_2 รวมกระแส 0.15 mA ได้แรงดันเอาต์พุต -1.5 โวลต์



รูปที่ 3.1 วงจรดิจิทัลแบบใช้ตัวต้านทานหลายค่า

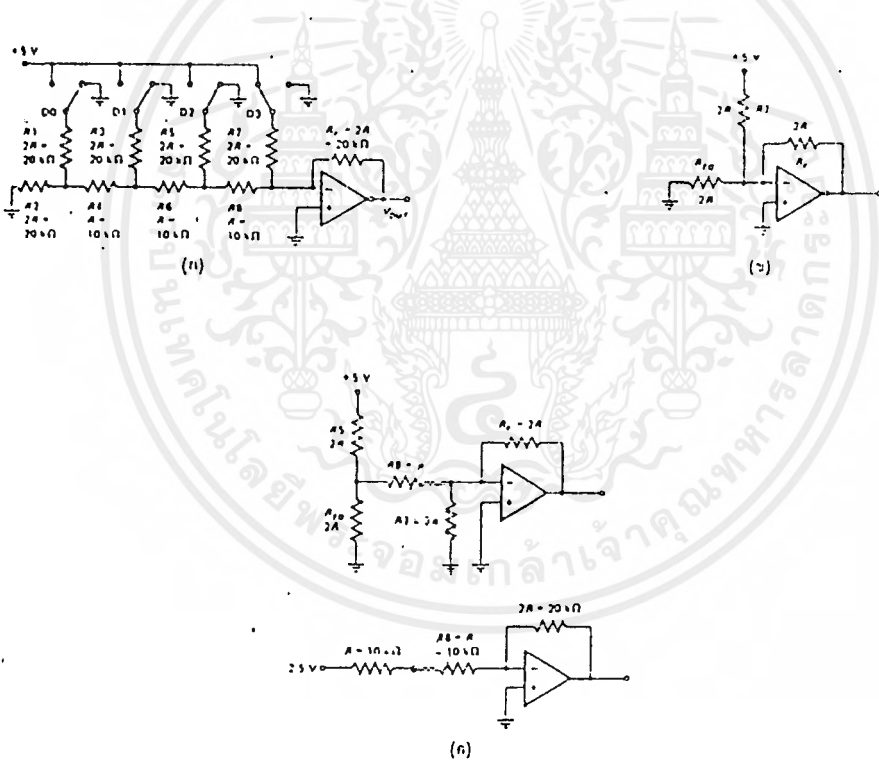
เมื่อเปลี่ยนการปิดเปิดสวิตช์ไปเรื่อย ๆ จะได้แรงดันเอาต์พุตค่าต่าง ๆ กัน กระแสที่ผ่านสวิตช์แต่ละตัว จะถูกรวมกันที่จุดกราวด์เทียม แล้วเปลี่ยนเป็นแรงดันที่เอาต์พุต โดยตัวต้านทานในป้อนกลับ R_F แรงดันเอาต์พุต จะเพิ่มขึ้นเป็นระดับ ๆ เหมือนขั้นบันได ดังนั้น 4 บิต จึงได้ 15 ระดับ แต่ละระดับต่างกัน -0.5 โวลต์ อาจกำหนดระยะห่างของแต่ละระดับได้ โดยเปลี่ยนขนาดของ R_F แต่ถ้า R_F มีค่ามากเกินไประดับบนสุดจะขับออปแอมป์ถึงจุดอิ่มตัว (ที่แรงดัน -14 โวลต์)

3.1.2 แบบใช้ตัวต้านทาน 2 ค่า ($R/2R$ Ladder D/A Converter)

เมื่อวงจรดิจิทัล มีขนาดมากกว่า 4 บิต วงจรตามรูปที่ 3.1 จะเกิดปัญหา เนื่องจากต้องการค่าความต้านทานที่มีช่วงกว้างมาก วิธีที่ใช้หลักการไบনারีเวทเหมือนกัน แต่ใช้ความต้านทานเพียง 2 ค่า แสดงในรูปที่ 3.2 (ก) ซึ่งกระแสจะถูกเปลี่ยนค่าแรงดันโดยออปแอมป์และตัวต้านทานป้อนกลับ R_F เหมือนวงจรในรูปที่ 3.1 วิธีนี้เรียกว่าการใช้ความต้านทาน 2 ค่า

สังเกตให้ดี หลักการความต้านทาน 2 ค่า ดูไปก็คล้ายกับกฎของเคียร์โฮฟ เพียงแต่ค่าความต้านทานที่ใช้เป็นอัตราส่วนที่ทำให้คำนวณได้ง่าย แรกเลยสมมติว่า สวิตช์ D_0 ซึ่งเป็นสวิตช์ในบิตที่มีนัยสำคัญสูงสุดนั้น ต่อกับแรงดันอ้างอิง 5 โวลต์ ในขณะที่สวิตช์ตัวอื่นปิดลงกราวด์ ดังนั้น R_1 และ R_2 จึงต่อขนานกันลงกราวด์ สังเกตตัวต้านทาน $2R$ ต่อขนานกับ $2R$ อีกตัวหนึ่งจึงมีค่าเท่ากับ R ค่า R นี้จะถูกบวกกับ R_4 กลายเป็นค่า $2R$ แล้วขนาน

กันกับ R3 ลงกราวด์ การรวมของ R3 และตัวต้านทานก่อนหน้า จึงทำให้เหลือเพียงค่า R ต่ออนุกรมกับ R6 พิจารณาเช่นเดียวกันกับวงจรส่วนที่เหลือก็จะได้เป็นวงจรง่ายขึ้น ดังรูปที่ 3.2(ข) โดยเหตุที่กราวด์เทียมของออปแอมป์มีแรงดัน 0 โวลต์ ทำให้ไม่มีกระแสไหลผ่านค่าความต้านทานเหล่านี้ลงกราวด์ จึงไม่ต้องสนใจในส่วนนี้ ดังนั้น แรงดัน 5 โวลต์ที่ปลายข้างหนึ่งของ R7 ค่า 20 กิโลโอห์ม ทำให้มีกระแส 0.25 มิลลิแอมป์ ผ่านที่จุดต่อและผ่าน RF 20 กิโลโอห์ม แรงดันเอาต์พุตที่ได้จากบิตที่มีนัยสำคัญสูงสุด จึงมีค่า -5 โวลต์ แรงดันที่ได้จากบิตที่มีนัยสำคัญรองลงมา ก็หาได้โดยปิดสวิตช์ D2 ไปยัง +5 โวลต์ และ D3 ลงกราวด์ ตัวต้านทานทั้งหมดที่อยู่ทางซ้ายของ R5 ในรูปที่ 3.2 (ก) ลดรูปเหลือเพียง 2R ต่อลงกราวด์ การวิเคราะห์วงจรสามารถนำทฤษฎีของเทวินินมาใช้ได้โดยมีการแบ่งแรงดันระหว่าง R5 และ 2R ที่ต่อลงกราวด์



รูปที่ 3.2 วงจรเปลี่ยนสัญญาณแบบ R/2R แลคเคอร์

ก) วงจรสมบูรณ์

ข) วงจรเสมือนในขณะที่สวิตช์ของบิตที่มีนัยสำคัญสูงสุดปิด

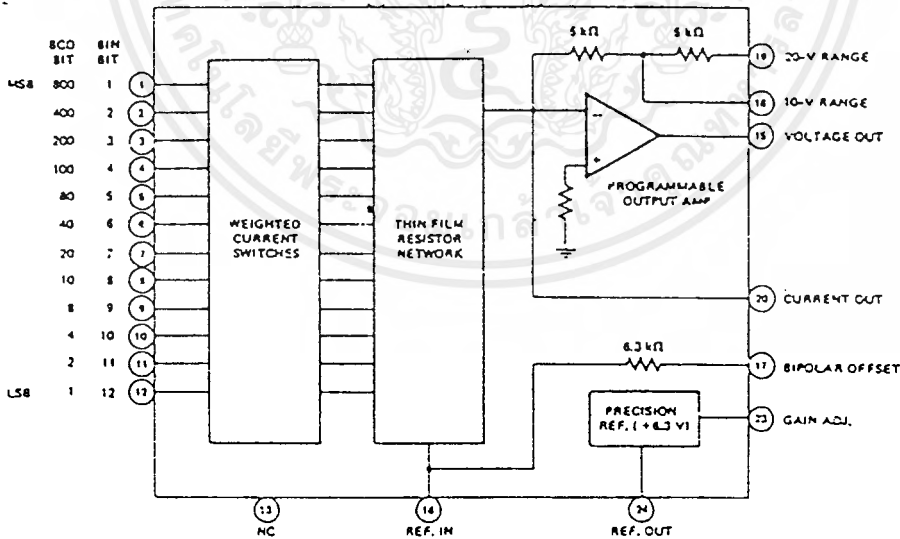
ค) วงจรเสมือนในขณะที่สวิตช์ของบิตที่มีนัยสำคัญรองลงมาปิด

ด้วยการวิเคราะห์ในทำนองเดียวกันนี้ สามารถหาแรงดันเอาต์พุตที่บิตต่ำลงมาอีก 1.25 โวลต์ และที่ค่าดิจิตอลต่ำสุดได้ 0.625 โวลต์ ในขณะที่ค่าดิจิตอลสูงสุด (สวิทช์ทุกตัวต่อไปที่ +5 โวลต์) ได้เอาต์พุตเต็มสเกลคือ 9.375 โวลต์

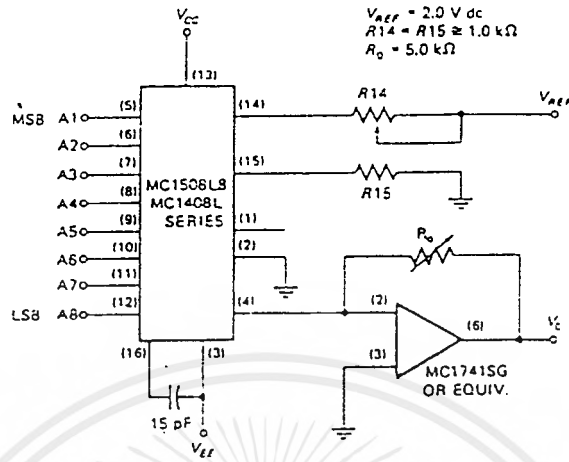
แม้ว่าคิพูเอ คอนเวอร์เตอร์ แบบ R/2R แลคเคอร์ จะวิเคราะห์ยากกว่าแบบใช้ตัวต้านทานหลายค่า (Weighted resistor) แต่จะง่ายกว่าสำหรับการต่อวงจรให้ถูกต้อง เพราะใช้ค่าความต้านทานเพียง 2 ค่าเท่านั้น จำนวนบิตก็เพิ่มได้โดยเพิ่มส่วนของ R/2R ลงไป วงจรนับ 4 บิตที่เป็น TTL หรือ CMOS อาจนำมาต่อแทนตำแหน่งของสวิทช์ในวงจรรูป 3.2 (ก) เพื่อให้แรงดันเอาต์พุตเป็นขั้นบันไดได้

3.1.3 แบบใช้ไอซี (Monolithic and Hybrid D/A Converter)

แบบนี้ใช้ค่าความต้านทานที่สร้างขึ้นมาพร้อมกัน ในขบวนการผลิตทาง IC ทำให้เรโซลต์ของความต้านทาน (R/2R) มีความแม่นยำมากกว่า IC ที่ผลิตมาใช้เป็น D/A มีอยู่หลายเบอร์ เช่น MC1408 , DAC0800 ซึ่งจำเป็นที่ต้องใช้แหล่งจ่ายไฟถึง 2 แหล่ง โครงสร้างของ MC1408 ดังแสดงในรูปที่ 2.20 และ การต่อใช้งานของ MC1408 ดังแสดงในรูป 3.4



รูปที่ 3.3 โครงสร้างภายใน MC 1408L



Theoretical V_0
$V_0 = \frac{V_{ref}}{R_{14}} (R_0) \left\{ \frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right\}$
ADJUST V_{ref} , R_{14} OR R_0 SO THAT V_0 WITH ALL DIGITAL INPUTS AT HIGH LEVEL IS EQUAL TO 9.961 V
$V_0 = \frac{2 \text{ V}}{1 \text{ k}\Omega} (5 \text{ k}\Omega) \left\{ \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} \right\}$
$= 10 \text{ V} \left\{ \frac{255}{256} \right\} = 9.961 \text{ V}$

รูปที่ 3.4 บล็อกไดอะแกรมและการต่อใช้งานของ MC 1408L

3.2 การคำนวณเอาต์พุตของ D/A

เราสามารถคำนวณค่าเอาต์พุตที่ V_0 ได้ดังนี้

$$V_0 = \frac{V_r}{R_1} (R_0) \left\{ \frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right\}$$

เช่น ถ้าทุกบิตเป็นศูนย์ จะได้

$$\begin{aligned}
 V_0 &= \frac{2 \text{ V}}{1 \text{ k}\Omega} (5 \text{ k}\Omega) \left\{ \frac{0}{2} + \frac{0}{4} + \frac{0}{8} + \frac{0}{16} + \frac{0}{32} + \frac{0}{64} + \frac{0}{128} + \frac{0}{256} \right\} \\
 &= 1 \text{ V} \left\{ \frac{0}{256} \right\} = 0 \text{ V}
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าทุกบิตเป็นหนึ่ง จะได้

$$V_o = \frac{2v}{1k\Omega}(5k\Omega) \left\{ \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} \right\}$$

$$= 1v \left\{ \frac{255}{256} \right\} = 9.96 \text{ V}$$

3.3 ลักษณะสมบัติของดีพูเอ

ลักษณะสมบัติอันแรกของการแปลงสัญญาณดิจิทัลเป็นอนาล็อก ที่จะพูดถึงก็คือ ความละเอียด (resolution) ซึ่งขึ้นกับจำนวนของบิตทางด้านอินพุต ตัวอย่างเช่น วงจรเปลี่ยนสัญญาณ 8 บิต มีระดับเอาต์พุต 2^8 หรือ 256 ระดับ ดังนั้น ความละเอียดคือ 1 ใน 256 วงจรเปลี่ยนสัญญาณ 12 บิต มีความละเอียด 2^{12} หรือ 4096 ความละเอียดบางครั้งจะคิดเป็นเปอร์เซ็นต์ คือ $1/4096 = 0.024\%$

ลักษณะสมบัติข้อต่อมาก็คือ ความถูกต้อง (accuracy) ของดีพูเอ ความถูกต้องจากการเปรียบเทียบระหว่างเอาต์พุตที่ปรากฏ โดยคิดที่เต็มสเกล ถ้าวงจรเปลี่ยนสัญญาณมีเอาต์พุตเต็มสเกล 10 โวลต์ มีความถูกต้อง $\pm 0.2\%$ ดังนั้นความผิดพลาดสูงสุดคือ 0.002×10 โวลต์ หรือ 20 มิลลิโวลต์ ในทางทฤษฎีแล้ว ความถูกต้องของวงจรเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อกไม่ควรต่ำกว่า ± 2 ของค่าที่ LSB (บิตที่มีนัยสำคัญต่ำสุด)

วงจรเปลี่ยนสัญญาณ 10 บิตมีความละเอียด $1/1024$ หรือประมาณ 0.1% ความถูกต้องควรมีค่า $\pm 0.05\%$

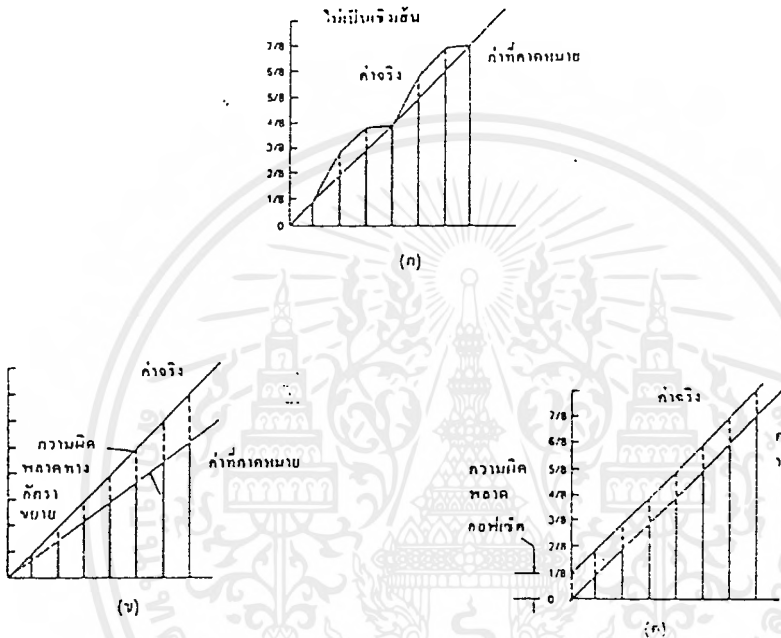
ความผิดพลาดอาจเกิดได้หลายประการ ในรูปที่ 3.5 แสดงไว้ 3 แบบดังนี้คือ

ความผิดพลาดเชิงเส้น (Linearity errors) ค่าจริงที่ได้จากเอาต์พุตจริง ต่างจากเอาต์พุตตามทฤษฎีที่ควรจะเป็นเส้นตรง ความผิดพลาดนี้มักจะมาจากความผิดพลาดจากแหล่งจ่ายกระแสหรือค่าความต้านทาน

ความผิดพลาดแบบที่สองคือ ความผิดพลาดทางอัตราขยาย (gain error) ความผิดพลาดนี้ มักเกิดจากความผิดพลาดของตัวต้านทานป้อนกลับของออปแอมป์ ที่ทำหน้าที่เปลี่ยนจากกระแสเป็นแรงดัน ความผิดพลาดออฟเซต หรือ offset error คือ เมื่ออินพุตทุกตัวเป็นศูนย์แล้ว เอาต์พุตไม่เป็นศูนย์ ทำให้เอาต์พุตมีค่าแรงดันผิดพลาดค่าหนึ่งบวกกับค่าจริงอยู่ตลอดเวลา ดังในรูปที่ 3.5(ค) ความผิดพลาดนี้เกิดจากความผิดพลาดของการขยายของออป

แอมป์ และกระแสไหลที่การสวิตช์ ลักษณะสมบัติต่อมาคือ โมโนโทนิก (monotonicity) จะเรียกว่าเป็น โมโนโทนิกก็ต่อเมื่อไม่มีการกระโดดข้ามขึ้น ตลอดย่านการใช้งาน

ส่วนเวลาเซตเอาต์พุต (output setting time) เป็นเวลาที่เอาต์พุตของวงจรเปลี่ยน สัญญาณ ใช้ในการเพิ่มขึ้นถึง $\pm$ ของ LSB หลังจากมีการเปลี่ยนแปลงทางอินพุต ถ้าวงจร เปลี่ยนสัญญาณถูกใช้งานย่านความถี่สูง อาจทำให้มีการเพิ่มแรงดันขึ้นได้อีกประการหนึ่ง



รูปที่ 3.5 ความผิดพลาดที่เกิดขึ้นในการเปลี่ยนสัญญาณ

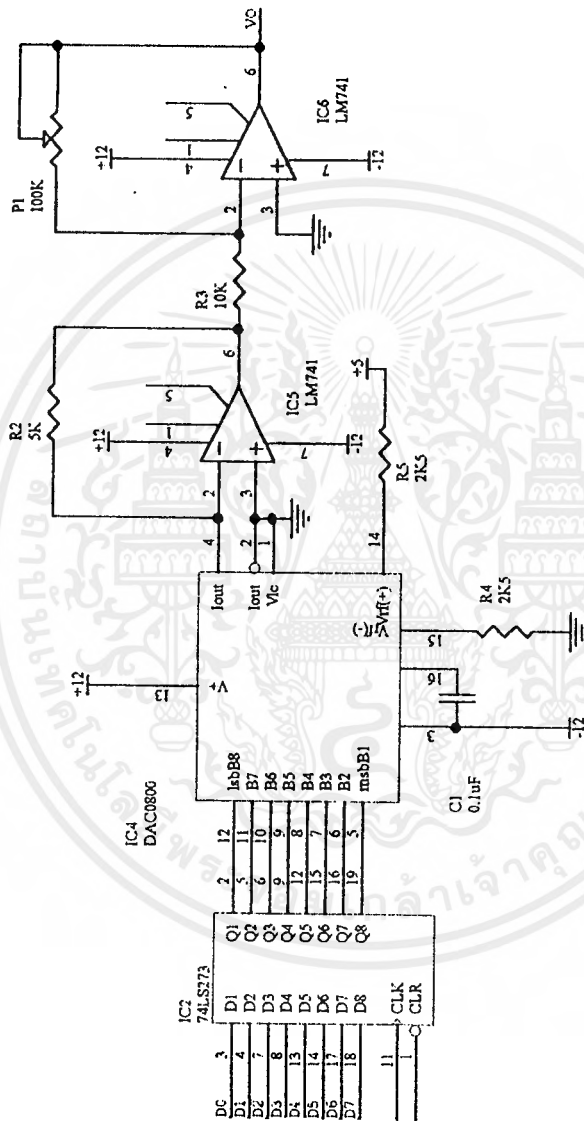
(ก) เชิงเส้น

(ข) อัตราขยาย

(ค) ออฟเซต

3.4 ลักษณะวงจรคิทูเอทีน่าไปใช้งานจริง

ในปริณูณานิพนธ์นี้ จะใช้ไอซีเบอร์ DAC 0800 ในการทดลองเพราะง่ายต่อการต่อใช้งานและสะดวก รูปวงจร 3.6 เป็นการต่อสัญญาณดิจิทัลมาจาก IC 74LS273 ซึ่งทำหน้าที่เป็น บัฟเฟอร์ เมื่อมีสัญญาณมาที่ขา CLK IC 74LS273 ก็จะส่งข้อมูลไปยัง IC DAC 0800 และเอาต์พุตของคิทูเอทีน่า 4 ก็จะผ่านออปแอมป์เพื่อที่จะแปลงสัญญาณจากกระแสเป็น โวลต์เตจ และทำการขยายสัญญาณให้แรงขึ้นโดยจะปรับที่ R100Kเพื่อให้ได้เอาต์พุต 0-10V ไอซีเบอร์ DAC 0800 จะมีหลักการทำงานเหมือนกับ MC1408 หรือ DAC 0800 คือจะมีอินพุตขนาด 8 บิต



รูป 3.6 แสดงวงจรดีพูเอทีใช้งานจริง

บทที่ 4

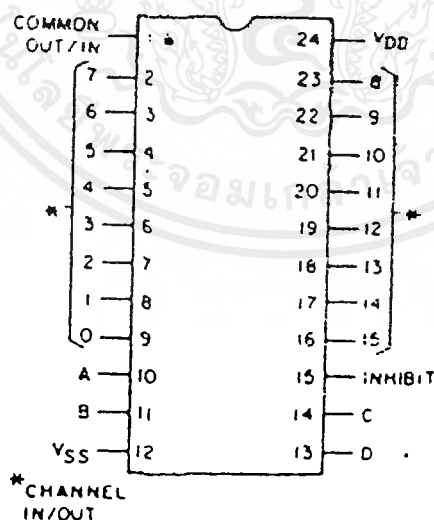
วงจรมัลติเพล็กซ์ช่องสัญญาณ

4.1 หลักการทำงานของวงจรมัลติเพล็กซ์

วงจรนี้ใช้ไอซีเบอร์ CD4067 หรือ MC 14067 เมื่อสัญญาณดิจิตอลถูกเปลี่ยนเป็นสัญญาณอนาล็อก จะนำสัญญาณที่ได้นั้นทำการแปลงสัญญาณดิจิตอลเพื่อเป็นอนาล็อกอีกครั้งหนึ่งเนื่องจากค่าโวลท์ที่ได้นั้นจะไม่สามารถเก็บแรงดันได้เมื่อทำการเปลี่ยนช่องสัญญาณเป็นช่องอื่นๆ และขาอินพุทของ IC CD4067 จะป้อนแรงดัน 5 โวลท์เพื่อให้เป็นสัญญาณเลือกช่อง เมื่อมีสัญญาณจาก IC 74LS273 ซึ่งเป็นสัญญาณที่ควบคุมมาจากโปรแกรม ออกพอร์ท 301H มาเข้าที่ขา SD0 - SD3 ก็จะทำให้การส่งอินพุทไปยังเอาต์พุท CD4067 ที่ขาสัญญาณที่กำหนดโดยขา SD0-SD3 นั้นเอง มีเอาต์พุท 16 ช่องซึ่งเป็นสัญญาณที่เลือกมาจากขา SD0 - SD3 ในปริยญาณิพนธ์นี้ใช้ทั้งหมด 64 ช่อง จึงต้องใช้ CD 4067 ถึง 4 ตัวเพื่อที่จะให้เอาต์พุตถึง 64 ช่องสัญญาณ

ส่วนที่เหลือจะเป็นเอาต์พุททั้งหมด 16 ช่อง ในการทำงานขา E หรือขาอินาเบิลต้องเป็น "0" ตลอด ไอซีนี้จึงจะทำงานได้ ถ้าเป็น "1" จะไม่ทำงาน

16-Channel Multiplexer/Demultiplexer,
V_{DD} = 3 V to 15 V

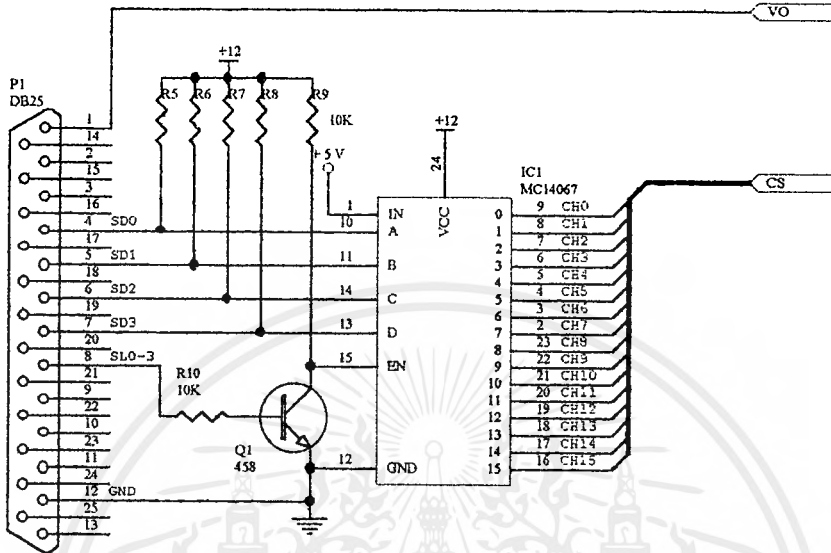


รูปที่ 4.1 วงจรภายในของไอซี CD 4067 หรือ MC 14067

SD0	SD1	SD2	SD3	ENABLE	CHANNEL
0	0	0	0	0	1
1	0	0	0	0	2
0	1	0	0	0	3
1	1	0	0	0	4
0	0	0	1	0	5
1	0	0	1	0	6
0	1	0	1	0	7
1	1	0	1	0	8
0	0	1	0	0	9
1	0	1	0	0	10
0	1	1	0	0	11
1	1	1	0	0	12
0	0	1	1	0	13
1	0	1	1	0	14
0	1	1	1	0	15
1	1	1	1	0	16

ตารางที่ 4.1 แสดงตารางการทำงานของ MC 14067

ในการทำงานของ CD4067 จะทำงานทีละตัวโดยควบคุมที่ขา E (ขาที่ 15) ตัวที่ถูกเลือกที่ขา 15 จะมีสถานะลอจิกเป็น “ 0 ” อีก 2 ตัว ตัวที่ขา 15 จะมีลอจิกเป็น “ 1 ” เอาท์พุทที่ได้จะนำไปส่งให้ IC 74LS373 เพื่อให้ IC ทำงานเปรียบเสมือนเป็นสัญญาณอินพุต



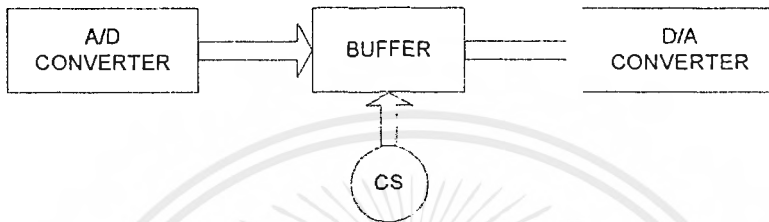
รูปที่ 4.2 แสดงวงจรมัลติเพล็กซ์ 16 ช่อง

จากรูปขาที่ 1 จาก DB 25 PIN จะเป็นแรงดันที่ได้มาจากการแปลงสัญญาณดิจิทัลเป็นอนาล็อกเราจะนำสัญญาณนี้ไปทำการแปลงกับไปเป็นสัญญาณดิจิทัลจะอธิบายในหัวข้อต่อไป และขา 1 ของไอซีจะป้อนไฟตรง 5 โวลต์ เพื่อไปทำการเลือกหรือมัลติเพล็กซ์ช่องจาก 1 ให้เป็น 16 ช่อง และขา 10,11,13,14 เป็นขา มัลติเพล็กซ์และขา 15 เป็นขาอินพุตของไอซี จะต้องเป็นลอจิก “0” จึงจะทำงาน แต่สัญญาณที่มาจากขา 8 ของ DB 25 PIN เป็นลอจิก “1” ดังนั้นต้องเพิ่มอินเวอร์เตอร์โดยใช้ทรานซิสเตอร์ C458 เพื่อให้เป็นลอจิก “0” และสัญญาณที่เอาพุทจะมี 16 ช่องนำไปเป็นชีพชีเล็กให้แก่ ไอซี 74LS373

4.2 วงจรเก็บระดับแรงไฟ

ในการที่นำแรงดันไฟไปใช้งานนั้นเมื่อมีการเปลี่ยนช่องสัญญาณจะทำให้ค่าที่ช่องเดิมนั้นตกเหลือ 0 โวลต์ เราจึงต้องเก็บแรงดันนั้นไว้เพื่อให้นำไปใช้งาน ในปริญญานิพนธ์นี้จะนำแรงดันไฟนั้นแปลงเป็นสัญญาณดิจิทัลแล้วจึงใช้บัฟเฟอร์ทำการเก็บสัญญาณนั้นเป็นดิจิทัลแล้วทำการ

แปลงสัญญาณดิจิตอลนั้นให้กับไปเป็นสัญญาณอนาล็อกในแต่ละช่องนั้นต้องใช้วงจรนี้ทุกช่อง เพื่อให้ได้ค่าตามต้องการ



รูป 4.3 รูปแสดงบล็อกไดอะแกรมของวงจรเก็บแรงดัน

จากบล็อกไดอะแกรมจะแสดงการทำงานทั้งหมดของภาคเก็บแรงดันไฟโดยสัญญาณไฟก่อนเข้าวงจร A/D converter นั้นจะมีค่าตั้งแต่ 0-10 โวลต์ เมื่อแปลงสัญญาณไฟเป็นสัญญาณดิจิตอลแล้วนั้น ส่วนของ BUFFER จะทำการเก็บข้อมูลเป็นระบบดิจิตอลเมื่อมีสัญญาณจากชุดมัลติเพล็กซ์หรือชุดเลือกช่องนั้นจะทำการส่งข้อมูลที่เป็นดิจิตอลไปยังชุดวงจร D/A converter เพื่อทำการแปลงสัญญาณเป็นอนาล็อกซึ่งมีค่า 0-10 โวลต์ และเมื่อทำการเปลี่ยนช่องสัญญาณ ส่วนของ BUFFER ก็ยังคงเก็บค่าอยู่ส่งผลให้วงจร D/A converter ก็ยังคงเก็บค่าแรงไฟเช่นเดิม

บทที่ 5

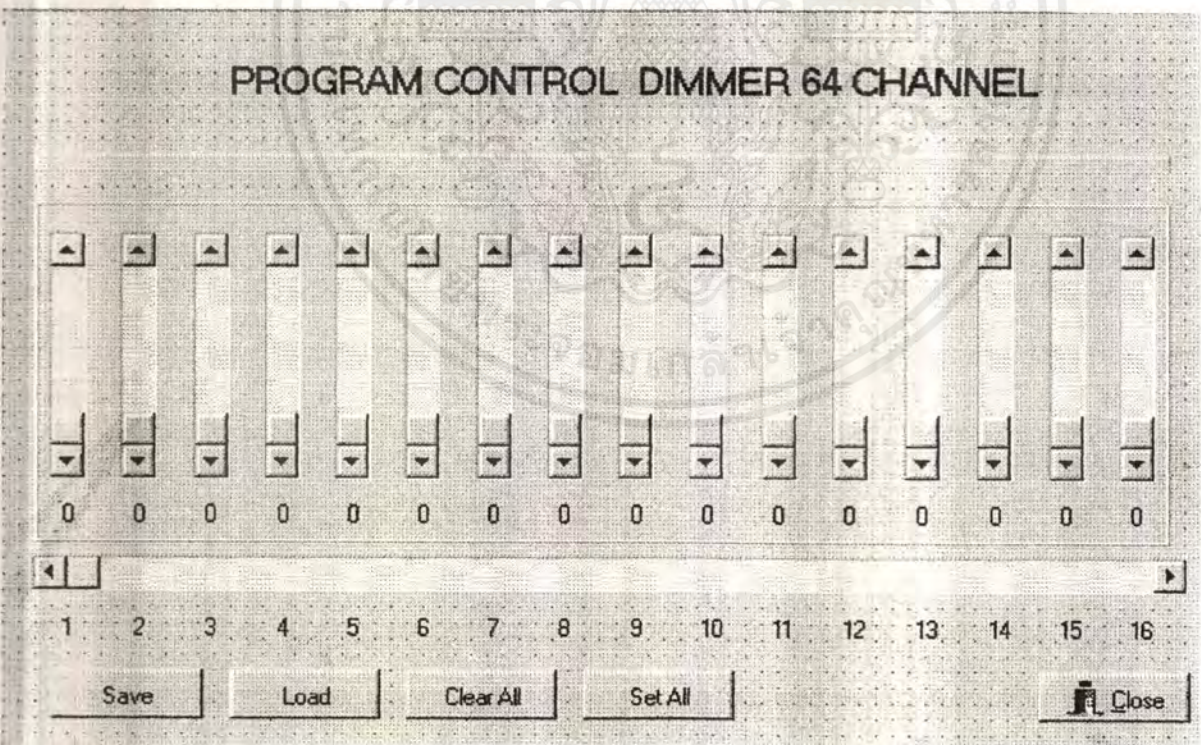
โปรแกรมควบคุมการทำงาน

โปรแกรม คือ ส่วนที่ใช้ควบคุมระบบฮาร์ดแวร์ทั้งหมดโดยจะใช้การส่งค่าออกพอร์ท ใน ส่วนของปริณิญาณิพนธ์นี้จะใช้โปรแกรมภาษาเคลไฟในการเขียนโปรแกรมเพราะจะทำให้สะดวก ต่อการใช้งานในด้านกราฟฟิกและการส่งข้อมูลออกพอร์ท โดยส่วนต่างๆ จะอธิบายได้ดังนี้

5.1 หลักการใช้งานของโปรแกรม

โปรแกรมที่ใช้นั้นจะมีฟังก์ชันการทำงานต่างๆ โดยจะควบคุมโดยใช้โปรแกรม Borland Delphi เวอร์ชัน 3.0 ซึ่งจะมีการกำหนดคอมโพเนนต์ได้ง่ายและสะดวกต่อการใช้งานทางด้านการ ส่งค่าออกทางด้านเอาท์พุทซึ่งจะมีฟังก์ชันการทำงานต่างๆ ดังนี้

เมนูหลัก (MAIN MENU) ก็จะมีลักษณะตามรูปด้านล่างนี้โดยจะประกอบด้วยส่วนที่เป็นสไลด์ ส่วนที่เป็นปุ่มคลิกที่มีฟังก์ชันการใช้งาน เช่น LOAD,SAVE เป็นต้นรายละเอียดจะกล่าวในช่วงต่อไป

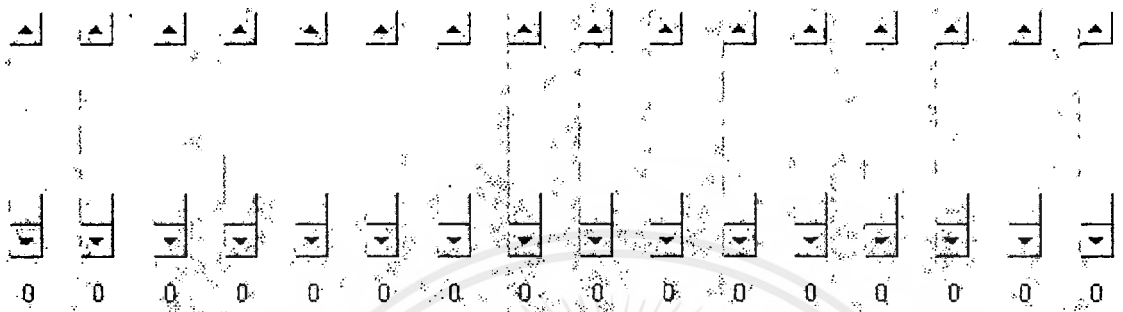


รูป 5.1 รูปแสดงเมนูหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 สไลด์แนวดิ่ง

มีไว้เพื่อเลื่อนปรับระดับแรงดันไฟ ซึ่งสามารถปรับได้ตั้งแต่ 0 โวลต์โดยจะเลื่อนในส่วนที่เป็นสไลด์ก็ได้หรือส่วนที่เป็นลูกศรด้านบนและด้านล่างของสไลด์ก็ได้ใน 1 เฟรม มี 16 ช่องเมื่อปรับระดับแรงดันแล้วจะมีตัวเลขบอกเพื่อให้ผู้ใช้ทราบว่าตอนนี้มีค่ากี่โวลต์ในแต่ละช่อง



5.1.2 สไลด์แนวนอน

มีไว้เพื่อเลื่อนช่องให้เพิ่มขึ้นครั้งละ 16 ช่องเนื่องจากในแต่ละเฟรมนั้นมีเพียง 16 ช่องเท่านั้น เพราะฉะนั้นจะมีทั้งหมด 4 เฟรมคือเฟรมแรกจะมีช่อง 1-16 เฟรมที่ 2 มี ช่อง 17-32 เฟรมที่ 3 มี ช่อง 33-48 เฟรมที่ 4 มี ช่อง 49-64



5.1.3 ปุ่ม SAVE

เมื่อคลิกที่ปุ่ม SAVE แล้วจะมีเมนูที่ใช้เพื่อทำการ SAVE ซึ่งสามารถ SAVE ได้ถึง 10 memory ก่อนที่จะ SAVE ต้องตั้งค่า memory ในจำนวน 64 ช่องซึ่งสามารถที่จะเก็บรายละเอียดได้ครบทุกช่อง



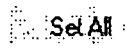
5.1.4 ปุ่ม LOAD

เมื่อคลิกที่ปุ่ม LOAD แล้วจะมีเมนูที่ใช้เพื่อทำการ LOAD ข้อมูลที่ได้ SAVE ไว้ใน 10 memory นั้น เมื่อทำการ LOAD ข้อมูลนั้นออกมาแล้วนั้น ค่าเอาต์พุตก็จะออกตามค่าที่ได้ LOAD ออกมาทันที



5.1.5 ปุ่ม SET ALL

ปุ่มนี้จะทำการเซตค่าทุกช่องในจำนวน 64 ช่องให้อาห์พุทออกค่า 10 โวลต์หรือ ค่า Maximum นั้นเอง



5.1.6 ปุ่ม CLEAR ALL

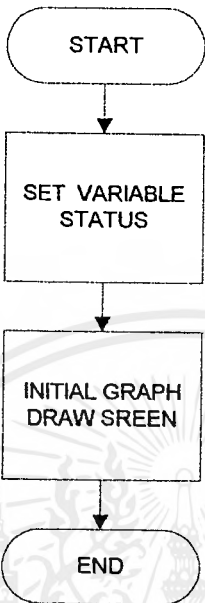
ปุ่ม Reset นั้นเองซึ่งจะทำการReset ให้ทุกช่องเป็น 0 โวลต์นั้นเอง



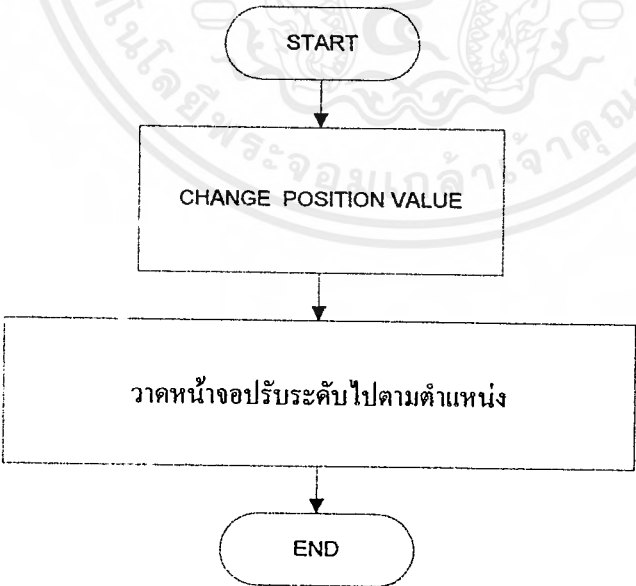
5.2 โครงสร้างโปรแกรม (FLOW CHAT)

โครงสร้างของโปรแกรมนั้นจะอธิบายเป็นแผนภูมิการไหล (FLOW CHAT) โดยจะประกอบด้วยส่วนหลักของโปรแกรม ส่วนที่เป็น SAVE ส่วนที่เป็น LOAD ส่วนที่เป็น SET ALL 1 ส่วนที่เป็น CLEAR ALL

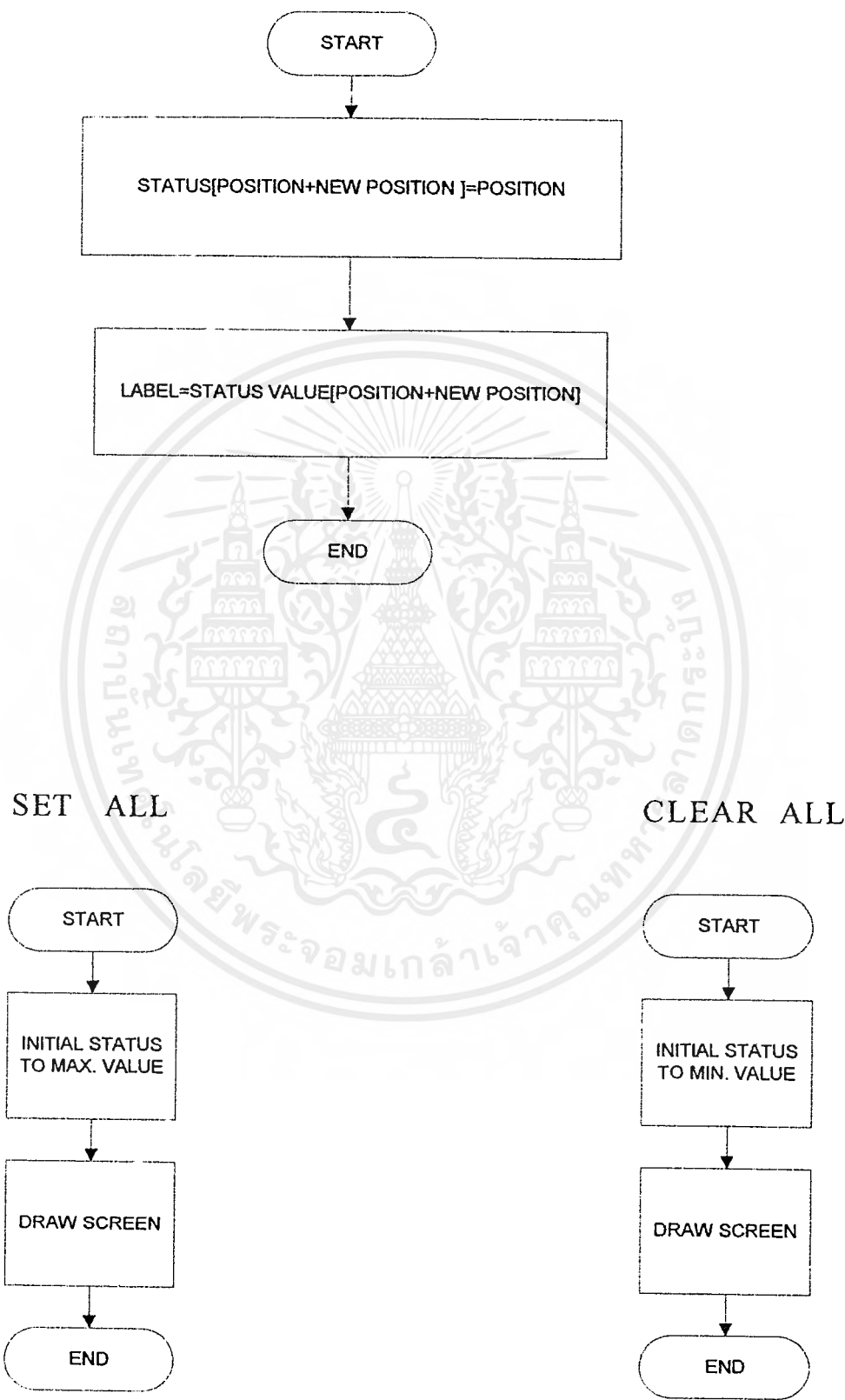
MAIN MENU



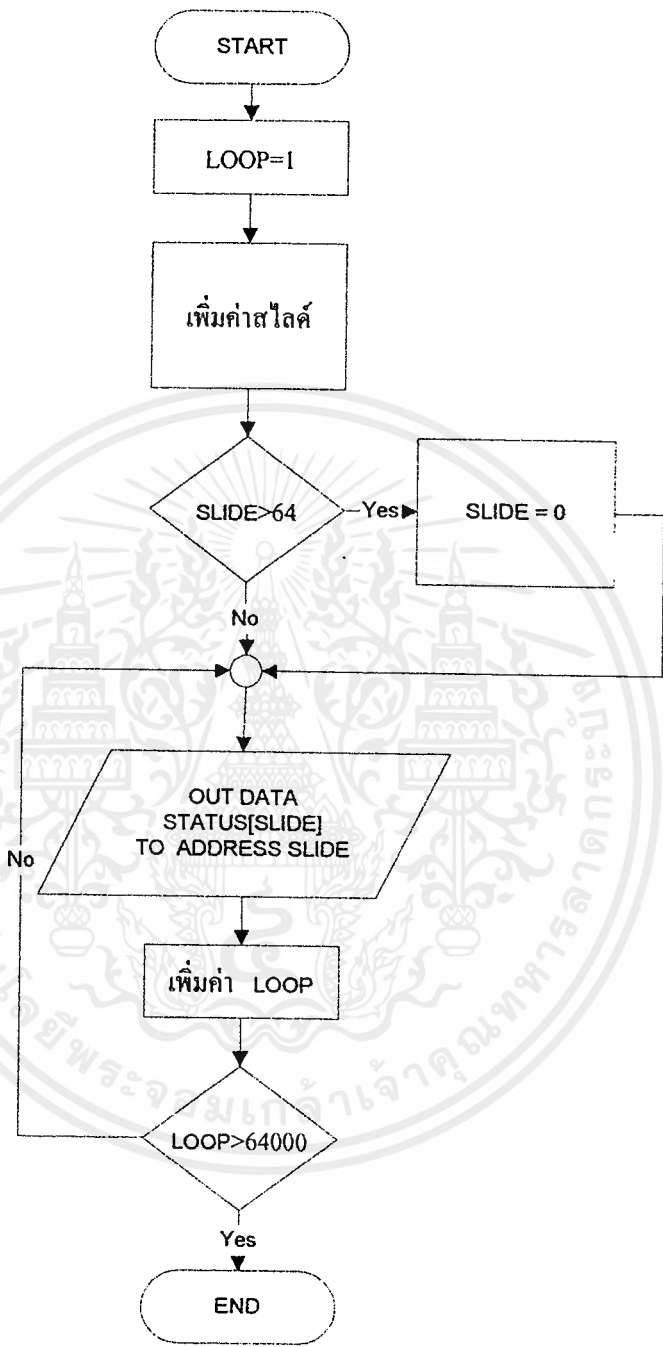
SCROLLBAR POSITION



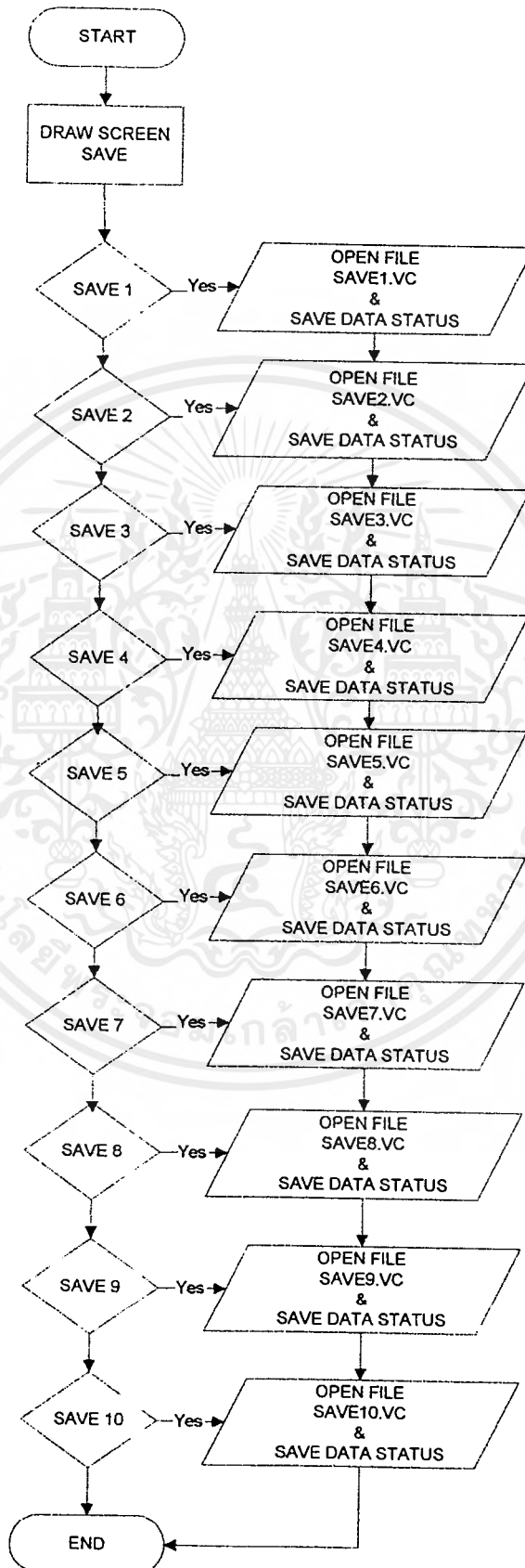
SCROLLBAR CHANGE POSITION



SCAN CHANNEL

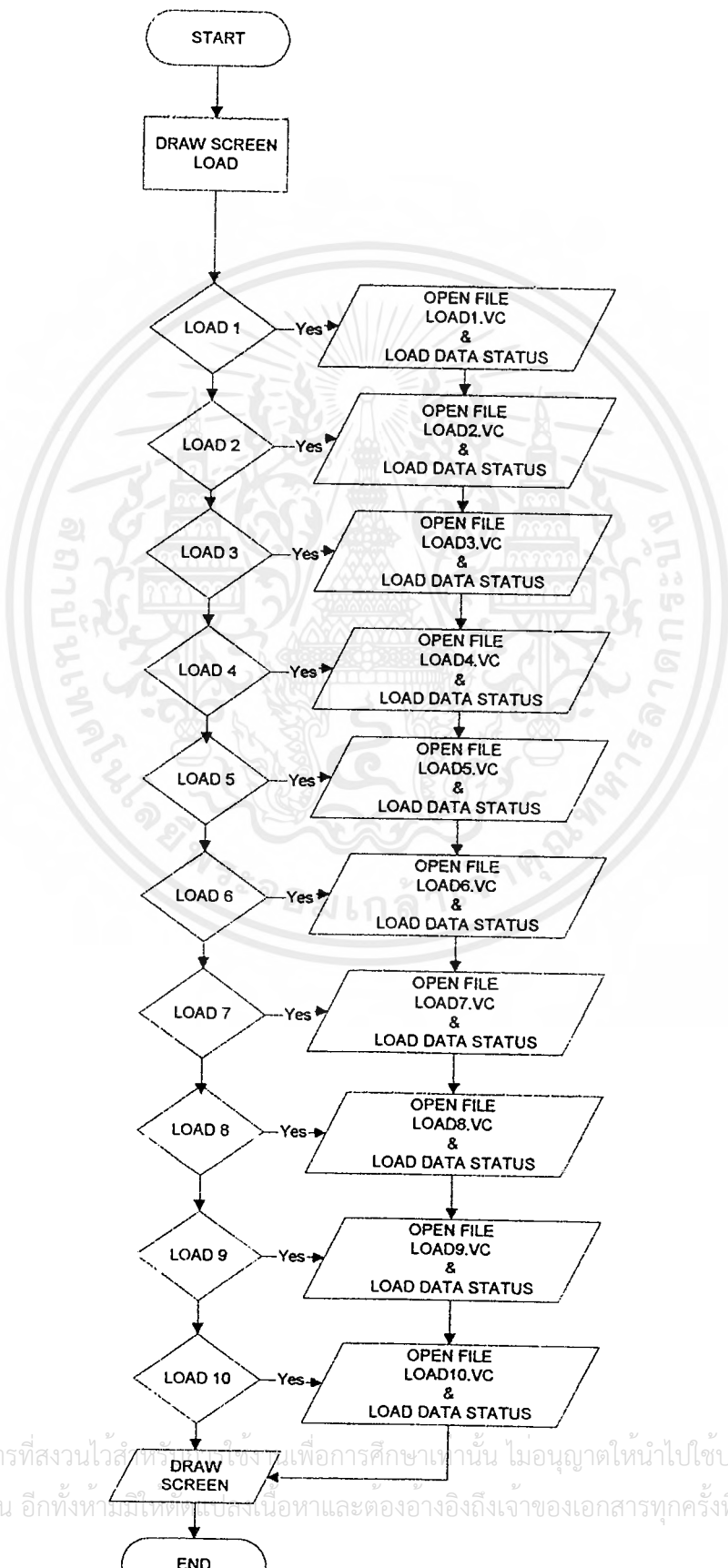


SAVE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOAD



5.3 ส่วนต่างของโปรแกรม

ส่วนต่างๆของโปรแกรม โดยจะมีส่วนที่เป็นเมนูหลักและส่วนที่เป็นเมนูเซฟและ โหลดซึ่งจะแสดงดังนี้คือ

```
unit Unit1;

interface

uses

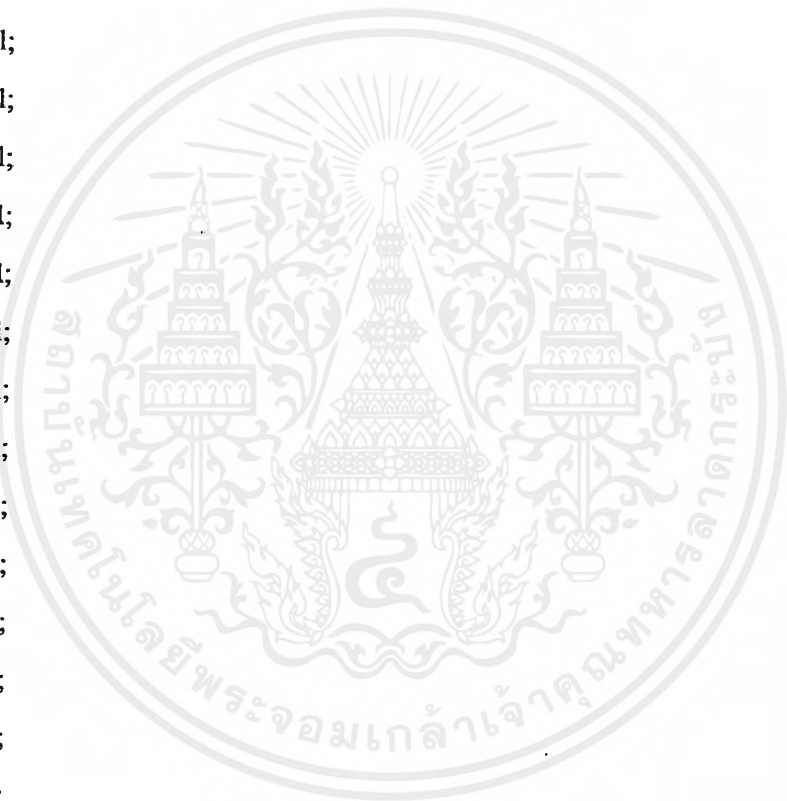
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Buttons;

type

  TForm1 = class(TForm)
    ScrollBar17: TScrollBar;
    Label1: TLabel;
    Panel1: TPanel;
    Panel2: TPanel;
    ScrollBar1: TScrollBar;
    ScrollBar2: TScrollBar;
    ScrollBar3: TScrollBar;
    ScrollBar4: TScrollBar;
    ScrollBar5: TScrollBar;
    ScrollBar6: TScrollBar;
    ScrollBar7: TScrollBar;
    ScrollBar8: TScrollBar;
    ScrollBar9: TScrollBar;
    ScrollBar10: TScrollBar;
    ScrollBar11: TScrollBar;
    ScrollBar12: TScrollBar;
    ScrollBar13: TScrollBar;
    ScrollBar14: TScrollBar;
    ScrollBar15: TScrollBar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ScrollBar16: TScrollBar;
 Label3: TLabel;
 Label4: TLabel;
 Label5: TLabel;
 Label6: TLabel;
 Label7: TLabel;
 Label8: TLabel;
 Label9: TLabel;
 Label10: TLabel;
 Label11: TLabel;
 Label12: TLabel;
 Label13: TLabel;
 Label14: TLabel;
 Label15: TLabel;
 Label16: TLabel;
 Label17: TLabel;
 Label18: TLabel;
 Label19: TLabel;
 Label20: TLabel;
 Label21: TLabel;
 Label22: TLabel;
 Label23: TLabel;
 Label24: TLabel;
 Label25: TLabel;
 Label26: TLabel;
 Label27: TLabel;
 Label28: TLabel;
 Label29: TLabel;
 Label30: TLabel;
 Label31: TLabel;



Label32: TLabel;

Label33: TLabel;

Label34: TLabel;

BitBtn1: TBitBtn;

BitBtn2: TBitBtn;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

BitBtn5: TBitBtn;

Timer1: TTimer;

procedure ScrollBar1Change(Sender: TObject);

procedure ScrollBar2Change(Sender: TObject);

procedure ScrollBar3Change(Sender: TObject);

procedure ScrollBar4Change(Sender: TObject);

procedure ScrollBar5Change(Sender: TObject);

procedure ScrollBar6Change(Sender: TObject);

procedure ScrollBar7Change(Sender: TObject);

procedure ScrollBar8Change(Sender: TObject);

procedure ScrollBar9Change(Sender: TObject);

procedure ScrollBar10Change(Sender: TObject);

procedure ScrollBar11Change(Sender: TObject);

procedure ScrollBar12Change(Sender: TObject);

procedure ScrollBar13Change(Sender: TObject);

procedure ScrollBar14Change(Sender: TObject);

procedure ScrollBar15Change(Sender: TObject);

procedure ScrollBar16Change(Sender: TObject);

procedure ScrollBar17Change(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

```

procedure BitBtn5Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
const
  max = 255;
var
  Form1: TForm1;
  status : array[1..64] of integer;
  loadsave,d : integer;
  volt: byte;
  ports : byte;
implementation
uses Unit2;
{$R *.DFM}
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
  label3.Caption := inttostr((max - scrollbar1.Position)div 25);
  status[1+(scrollbar17.Position*16)] := max-scrollbar1.Position;
  Ports := (scrollbar17.Position*16);
  asm mov dx,$301; mov al,[ports]; out dx,al; end;
  volt:= status[1+(scrollbar17.Position*16)];
  asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;
procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
  label4.Caption := inttostr((max-scrollbar2.Position)div 25);

```

```

status[2+(scrollbar17.Position*16)] := max-scrollbar2.Position;
Ports := 1+(scrollbar17.Position*16);
asm mov dx,$301; mov al,[ports]; out dx,al; end;
volt:= status[2+(scrollbar17.Position*16)];
asm mov dx,$300; mov al , [volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar3Change(Sender: TObject);
begin
    label5.Caption := inttostr((max-scrollbar3.Position)div 25);
    status[3+(scrollbar17.Position*16)] := max-scrollbar3.Position;
    Ports := 2+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[3+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar4Change(Sender: TObject);
begin
    label6.Caption := inttostr((max-scrollbar4.Position)div 25);
    status[4+(scrollbar17.Position*16)] := max-scrollbar4.Position;
    Ports := 3+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt:= status[4+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar5Change(Sender: TObject);
begin
    label7.Caption := inttostr((max-scrollbar5.Position)div 25);
    status[5+(scrollbar17.Position*16)] := max-scrollbar5.Position;
    Ports := 4+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;

```

```

    volt:= status[5+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar6Change(Sender: TObject);
begin
    label8.Caption := inttostr((max-scrollbar6.Position)div 25);
    status[6+(scrollbar17.Position*16)] := max-scrollbar6.Position;
    Ports := 5+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[6+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar7Change(Sender: TObject);
begin
    label9.Caption := inttostr((max-scrollbar7.Position)div 25);
    status[7+(scrollbar17.Position*16)] := max-scrollbar7.Position;
    Ports := 6+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[7+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

procedure TForm1.ScrollBar8Change(Sender: TObject);
begin
    label10.Caption := inttostr((max-scrollbar8.Position)div 25);
    status[8+(scrollbar17.Position*16)] := max-scrollbar8.Position;
    Ports := 7+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt:= status[8+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

```

```

procedure TForm1.ScrollBar9Change(Sender: TObject);
begin
    label11.Caption := inttostr((max-scrollbar9.Position)div 25);
    status[9+(scrollbar17.Position*16)] := max-scrollbar9.Position;
    Ports := 8+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt:= status[9+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

```

```

procedure TForm1.ScrollBar10Change(Sender: TObject);
begin
    label12.Caption := inttostr((max-scrollbar10.Position)div 25);
    status[10+(scrollbar17.Position*16)] := max-scrollbar10.Position;
    Ports := 9+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[10+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

```

```

procedure TForm1.ScrollBar11Change(Sender: TObject);
begin
    label13.Caption := inttostr((max-scrollbar11.Position)div 25);
    status[11+(scrollbar17.Position*16)] := max-scrollbar11.Position;
    Ports := 10+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[11+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;

```

```

procedure TForm1.ScrollBar12Change(Sender: TObject);
begin

```

```

    label14.Caption := inttostr((max-scrollbar12.Position)div 25);

```

```

status[12+(scrollbar17.Position*16)] := max-scrollbar12.Position;
Ports := 11+(scrollbar17.Position*16);
asm mov dx,$301; mov al,[ports]; out dx,al; end;
volt:= status[12+(scrollbar17.Position*16)];
asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;
procedure TForm1.ScrollBar13Change(Sender: TObject);
begin
    label15.Caption := inttostr((max-scrollbar13.Position)div 25);
    status[13+(scrollbar17.Position*16)] := max-scrollbar13.Position;
    Ports := 12+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[13+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;
procedure TForm1.ScrollBar14Change(Sender: TObject);
begin
    label16.Caption := inttostr((max-scrollbar14.Position)div 25);
    status[14+(scrollbar17.Position*16)] := max-scrollbar14.Position;
    Ports := 13+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    volt := status[14+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;
end;
procedure TForm1.ScrollBar15Change(Sender: TObject);
begin
    label17.Caption := inttostr((max-scrollbar15.Position)div 25);
    status[15+(scrollbar17.Position*16)] := max-scrollbar15.Position;
    Ports := 14+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;

```

```

voltage := status[15+(scrollbar17.Position*16)];
asm mov dx,$300; mov al,[voltage]; out dx,al; end;
end;

procedure TForm1.ScrollBar16Change(Sender: TObject);
begin
    label18.Caption := inttostr((max-scrollbar16.Position)div 25);
    status[16+(scrollbar17.Position*16)] := max-scrollbar16.Position;
    Ports := 15+(scrollbar17.Position*16);
    asm mov dx,$301; mov al,[ports]; out dx,al; end;
    voltage := status[16+(scrollbar17.Position*16)];
    asm mov dx,$300; mov al,[voltage]; out dx,al; end;
end;

procedure TForm1.ScrollBar17Change(Sender: TObject);
begin
    label19.Caption := inttostr(1+(scrollbar17.Position*16));
    label20.Caption := inttostr(2+(scrollbar17.Position*16));
    label21.Caption := inttostr(3+(scrollbar17.Position*16));
    label22.Caption := inttostr(4+(scrollbar17.Position*16));
    label23.Caption := inttostr(5+(scrollbar17.Position*16));
    label24.Caption := inttostr(6+(scrollbar17.Position*16));
    label25.Caption := inttostr(7+(scrollbar17.Position*16));
    label26.Caption := inttostr(8+(scrollbar17.Position*16));
    label27.Caption := inttostr(9+(scrollbar17.Position*16));
    label28.Caption := inttostr(10+(scrollbar17.Position*16));
    label29.Caption := inttostr(11+(scrollbar17.Position*16));
    label30.Caption := inttostr(12+(scrollbar17.Position*16));
    label31.Caption := inttostr(13+(scrollbar17.Position*16));
    label32.Caption := inttostr(14+(scrollbar17.Position*16));
    label33.Caption := inttostr(15+(scrollbar17.Position*16));
    label34.Caption := inttostr(16+(scrollbar17.Position*16));

```

```

scrollbar1.Position := max-status[1+(scrollbar17.Position*16)];
scrollbar2.Position := max-status[2+(scrollbar17.Position*16)];
scrollbar3.Position := max-status[3+(scrollbar17.Position*16)];
scrollbar4.Position := max-status[4+(scrollbar17.Position*16)];
scrollbar5.Position := max-status[5+(scrollbar17.Position*16)];
scrollbar6.Position := max-status[6+(scrollbar17.Position*16)];
scrollbar7.Position := max-status[7+(scrollbar17.Position*16)];
scrollbar8.Position := max-status[8+(scrollbar17.Position*16)];
scrollbar9.Position := max-status[9+(scrollbar17.Position*16)];
scrollbar10.Position := max-status[10+(scrollbar17.Position*16)];
scrollbar11.Position := max-status[11+(scrollbar17.Position*16)];
scrollbar12.Position := max-status[12+(scrollbar17.Position*16)];
scrollbar13.Position := max-status[13+(scrollbar17.Position*16)];
scrollbar14.Position := max-status[14+(scrollbar17.Position*16)];
scrollbar15.Position := max-status[15+(scrollbar17.Position*16)];
scrollbar16.Position := max-status[16+(scrollbar17.Position*16)];

```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
var i : integer;
```

```
begin
```

```
  for i := 1 to 64 do
```

```
    status[i] := 0;
```

```
  d := -1;
```

```
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
  loadsave := 1;
```

```
  form2.show;
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.BitBtn2Click(Sender: TObject);
```

```
begin
```

```
    loadsave := 2;
```

```
    form2.show;
```

```
end;
```

```
procedure TForm1.BitBtn4Click(Sender: TObject);
```

```
var i : integer;
```

```
begin
```

```
    for i := 1 to 64 do
```

```
        status[i] := 0;
```

```
        scrollbar1.Position := max-status[1+(scrollbar17.Position*16)];
```

```
        scrollbar2.Position := max-status[2+(scrollbar17.Position*16)];
```

```
        scrollbar3.Position := max-status[3+(scrollbar17.Position*16)];
```

```
        scrollbar4.Position := max-status[4+(scrollbar17.Position*16)];
```

```
        scrollbar5.Position := max-status[5+(scrollbar17.Position*16)];
```

```
        scrollbar6.Position := max-status[6+(scrollbar17.Position*16)];
```

```
        scrollbar7.Position := max-status[7+(scrollbar17.Position*16)];
```

```
        scrollbar8.Position := max-status[8+(scrollbar17.Position*16)];
```

```
        scrollbar9.Position := max-status[9+(scrollbar17.Position*16)];
```

```
        scrollbar10.Position := max-status[10+(scrollbar17.Position*16)];
```

```
        scrollbar11.Position := max-status[11+(scrollbar17.Position*16)];
```

```
        scrollbar12.Position := max-status[12+(scrollbar17.Position*16)];
```

```
        scrollbar13.Position := max-status[13+(scrollbar17.Position*16)];
```

```
        scrollbar14.Position := max-status[14+(scrollbar17.Position*16)];
```

```
        scrollbar15.Position := max-status[15+(scrollbar17.Position*16)];
```

```
        scrollbar16.Position := max-status[16+(scrollbar17.Position*16)];
```

```
end;
```

```
procedure TForm1.BitBtn5Click(Sender: TObject);
```

```
var i : integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```
  for i := 1 to 64 do
```

```
    status[i] := 255;
```

```
    scrollbar1.Position := max-status[1+(scrollbar17.Position*16)];
```

```
    scrollbar2.Position := max-status[2+(scrollbar17.Position*16)];
```

```
    scrollbar3.Position := max-status[3+(scrollbar17.Position*16)];
```

```
    scrollbar4.Position := max-status[4+(scrollbar17.Position*16)];
```

```
    scrollbar5.Position := max-status[5+(scrollbar17.Position*16)];
```

```
    scrollbar6.Position := max-status[6+(scrollbar17.Position*16)];
```

```
    scrollbar7.Position := max-status[7+(scrollbar17.Position*16)];
```

```
    scrollbar8.Position := max-status[8+(scrollbar17.Position*16)];
```

```
    scrollbar9.Position := max-status[9+(scrollbar17.Position*16)];
```

```
    scrollbar10.Position := max-status[10+(scrollbar17.Position*16)];
```

```
    scrollbar11.Position := max-status[11+(scrollbar17.Position*16)];
```

```
    scrollbar12.Position := max-status[12+(scrollbar17.Position*16)];
```

```
    scrollbar13.Position := max-status[13+(scrollbar17.Position*16)];
```

```
    scrollbar14.Position := max-status[14+(scrollbar17.Position*16)];
```

```
    scrollbar15.Position := max-status[15+(scrollbar17.Position*16)];
```

```
    scrollbar16.Position := max-status[16+(scrollbar17.Position*16)];
```

```
end;
```

```
procedure portout;
```

```
var i : integer;
```

```
  volt : byte;
```

```
begin
```

```
  { inc(d);
```

```
  if d > 64 then d := 0;
```

```
  for i := 0 to d do begin end;
```

```
  asm dec al; mov dx,$301; out dx,al; end;
```

```
  for i := 1 to status[d+1] do begin end;
```

```
  asm mov dx,$300; out dx,al; end;}
```

```

{   asm mov dx,$301; mov al,0; out dx,al; end;
    volt:= status[1];
    asm mov dx,$300; mov al,[volt]; out dx,al; end;}
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var i : integer;
begin
{   for i := 1 to 64000 do
        begin
            portout;
        end;}
end;
end.

unit Unit2;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, Buttons;
type
    TForm2 = class(TForm)
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        BitBtn3: TBitBtn;
        BitBtn4: TBitBtn;
        BitBtn5: TBitBtn;
        BitBtn6: TBitBtn;
        BitBtn7: TBitBtn;
        BitBtn8: TBitBtn;
        BitBtn9: TBitBtn;

```

```

BitBtn10: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form2: TForm2;
  s : string;
  f : textfile;
  index : integer;
implementation
uses unit1;
{$R *.DFM}
procedure TForm2.BitBtn1Click(Sender: TObject);
var index : integer;
begin
  AssignFile(F, 'save1.vc'); { File selected in dialog box }

```

```

if loadsave = 1 then
begin
    Rewrite(F);
    for index := 1 to 64 do
        writeln(f,inttostr(status[index]));
    CloseFile(F);
end else
if loadsave = 2 then
begin
    Reset(F);
    for index := 1 to 64 do
    begin
        readln(f,s);
        status[index] := strtoint(s);
    end;
    CloseFile(F);
end;
form2.close;
end;
procedure TForm2.BitBtn2Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save2.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else

```

```

if loadsave = 2 then
begin
    Reset(F);

    for index := 1 to 64 do
    begin
        readln(f,s);
        status[index] := strtoint(s);
    end;
    CloseFile(F);
end;
form2.close;
end;
procedure TForm2.BitBtn3Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save3.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else
    if loadsave = 2 then
    begin
        Reset(F);
        for index := 1 to 64 do
        begin
            readln(f,s);
            status[index] := strtoint(s);

```

```

end;
CloseFile(F);
end;
form2.close;
end;
procedure TForm2.BitBtn4Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save4.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        AssignFile(F, 'save4.vc'); { File selected in dialog box }
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else
    if loadsave = 2 then
    begin
        Reset(F);
        for index := 1 to 64 do
        begin
            readln(f,s);
            status[index] := strtoint(s);
        end;
        CloseFile(F);
    end;
    form2.close;
end;
procedure TForm2.BitBtn5Click(Sender: TObject);

```

```

var index : integer;
begin
    AssignFile(F, 'save5.vc'); { File selected in dialog box }
    if loadsave = 1 then
        begin
            Rewrite(F);
            for index := 1 to 64 do
                writeln(f,inttostr(status[index]));
            CloseFile(F);
        end else
            if loadsave = 2 then
                begin
                    Reset(F);
                    for index := 1 to 64 do
                        begin
                            readln(f,s);
                            status[index] := strtoint(s);
                        end;
                    CloseFile(F);
                end;
            form2.close;
        end;

procedure TForm2.BitBtn6Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save6.vc'); { File selected in dialog box }
    if loadsave = 1 then
        begin
            Rewrite(F);
            for index := 1 to 64 do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        writeln(f,inttostr(status[index]));
    CloseFile(F);
end else
if loadsave = 2 then
begin
    Reset(F);
    for index := 1 to 64 do
    begin
        readln(f,s);
        status[index] := strtoint(s);
    end;
    CloseFile(F);
end;
form2.close;
end;
procedure TForm2.BitBtn7Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save7.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else
    if loadsave = 2 then
    begin
        Reset(F);
        for index := 1 to 64 do

```

```

begin
    readln(f,s);
    status[index] := strtoint(s);
end;
CloseFile(F);
end;
form2.close;
end;
procedure TForm2.BitBtn8Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save8.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else
    if loadsave = 2 then
    begin
        Reset(F);
        for index := 1 to 64 do
        begin
            readln(f,s);
            status[index] := strtoint(s);
        end;
        CloseFile(F);
    end;
    form2.close;

```

```

end;

procedure TForm2.BitBtn9Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save9.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin
        Rewrite(F);
        for index := 1 to 64 do
            writeln(f,inttostr(status[index]));
        CloseFile(F);
    end else
    if loadsave = 2 then
    begin
        Reset(F);
        for index := 1 to 64 do
        begin
            readln(f,s);
            status[index] := strtoint(s);
        end;
        CloseFile(F);
    end;
    form2.close;
end;

procedure TForm2.BitBtn10Click(Sender: TObject);
var index : integer;
begin
    AssignFile(F, 'save10.vc'); { File selected in dialog box }
    if loadsave = 1 then
    begin

```

```

Rewrite(F);
for index := 1 to 64 do
    writeln(f,inttostr(status[index]));
CloseFile(F);
end else
if loadsave = 2 then
begin
    Reset(F);
    for index := 1 to 64 do
    begin
        readln(f,s);
        status[index] := strtoint(s);
    end;
    CloseFile(F);
end;
form2.close;
end;
procedure TForm2.FormShow(Sender: TObject);
begin
    if loadsave = 1 then
    begin
        BitBtn1.Caption := 'Save1';
        BitBtn2.Caption := 'Save2';
        BitBtn3.Caption := 'Save3';
        BitBtn4.Caption := 'Save4';
        BitBtn5.Caption := 'Save5';
        BitBtn6.Caption := 'Save6';
        BitBtn7.Caption := 'Save7';
        BitBtn8.Caption := 'Save8';
        BitBtn9.Caption := 'Save9';
    end;
end;

```

```

    BitBtn10.Caption := 'Save10';
end else
if loadsave = 2 then
begin
    BitBtn1.Caption := 'load1';
    BitBtn2.Caption := 'load2';
    BitBtn3.Caption := 'load3';
    BitBtn4.Caption := 'load4';
    BitBtn5.Caption := 'load5';
    BitBtn6.Caption := 'load6';
    BitBtn7.Caption := 'load7';
    BitBtn8.Caption := 'load8';
    BitBtn9.Caption := 'load9';
    BitBtn10.Caption := 'load10';
end;
end;
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    form1.scrollbar1.Position := max-status[1+(form1.scrollbar17.Position*16)];
    form1.scrollbar2.Position := max-status[2+(form1.scrollbar17.Position*16)];
    form1.scrollbar3.Position := max-status[3+(form1.scrollbar17.Position*16)];
    form1.scrollbar4.Position := max-status[4+(form1.scrollbar17.Position*16)];
    form1.scrollbar5.Position := max-status[5+(form1.scrollbar17.Position*16)];
    form1.scrollbar6.Position := max-status[6+(form1.scrollbar17.Position*16)];
    form1.scrollbar7.Position := max-status[7+(form1.scrollbar17.Position*16)];
    form1.scrollbar8.Position := max-status[8+(form1.scrollbar17.Position*16)];
    form1.scrollbar9.Position := max-status[9+(form1.scrollbar17.Position*16)];
    form1.scrollbar10.Position := max-status[10+(form1.scrollbar17.Position*16)];
    form1.scrollbar11.Position := max-status[11+(form1.scrollbar17.Position*16)];
    form1.scrollbar12.Position := max-status[12+(form1.scrollbar17.Position*16)];

```

```

form1.scrollbar13.Position := max-status[13+(form1.scrollbar17.Position*16)];
form1.scrollbar14.Position := max-status[14+(form1.scrollbar17.Position*16)];
form1.scrollbar15.Position := max-status[15+(form1.scrollbar17.Position*16)];
form1.scrollbar16.Position := max-status[16+(form1.scrollbar17.Position*16)];

end;

end.

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการทดลองและวิจารณ์

6.1 สรุปผลการทดลอง

จากวงจรอินเตอร์เฟสนั้นจะมีปัญหาตรงความไวในการเปลี่ยนแปลงของส่วนที่ควบคุม ฮาร์ดแวร์ เพราะจำนวนช่องมากจะทำให้ไม่สามารถที่จะหาการเปลี่ยนแปลงของการปรับแรงดัน คือ อาจจะทำให้ช้า และวงจรดีพูนนั้นการทดลองตอนแรกใช้เบอร์ 0808 จะมีปัญหาตรงค่อนข้างที่จะเสถียรจึงเปลี่ยนมาใช้ 0800 ซึ่งจะเหมือนกับ 0808 ทุกประการ และจะรวมไว้ในส่วนที่เป็น การ์ดอินเตอร์เฟส เพราะจะทำให้ประหยัดในการส่งข้อมูลออกไปยังส่วนที่ใช้แยกช่องสัญญาณ แต่จะเกิดปัญหาตรงที่ไม่สามารถเก็บแรงดันได้จึงต้องเพิ่มส่วนที่เก็บแรงดัน คือ ต้องแปลงสัญญาณที่เป็นอนาล็อกเป็นดิจิตอลก่อนเพื่อให้สามารถเก็บข้อมูลได้เนื่องจากระบบอนาล็อกไม่สามารถที่จะเก็บรักษาแรงดันเมื่ออินพุตไม่ป้อนค่าแรงดันได้ จึงต้องเก็บแบบดิจิตอลแล้วจึงแปลงสัญญาณกลับมาเป็นอนาล็อกโดยใช้ บัพเฟอร์ ถ้าเกิดว่าใช้การสแกนช่องเพื่อหาค่าแรงดัน นั้นจะมีปัญหาที่เวลาเนื่องจากความถี่หรือความไวต้องสูงมากเพราะจำนวนช่องมาก ดังนั้น ปริมาณนิพจน์ฉบับนี้จึงเปลี่ยนจากการสแกนมาเป็นการส่งค่าออกพอร์ทในแต่ละช่อง

6.2 แนวทางการพัฒนา

แนวทางที่จะพัฒนา คือ สามารถที่จะทำให้สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่างๆได้โดยอาจเพิ่มเติมส่วนที่เป็น ฮาร์ดแวร์ เช่น ควบคุมการเปิดปิดแอร์ เพื่อการประหยัดไฟฟ้า เป็นต้น

แต่อย่างไรก็ตามโครงการนี้มีวัตถุประสงค์เพื่อใช้ในการศึกษา ทดลอง เพื่อเป็นแนวความคิดที่จะพัฒนาปรับปรุงต่อไปในภายภาคหน้า

บรรณานุกรม

วิบูลย์ ชื่นแขก. ไมโครโปรเซสเซอร์. พิมพ์ครั้งที่ 2.

สำนักพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2532

จิตติ หนูแก้ว. ไมโครโปรเซสเซอร์และการออกแบบเบื้องต้น.

หจก.เอช-เอน การพิมพ์, 2536

ธานินทร์ ถาวรศาสนวงศ์.การอินเทอร์เฟส IBM/PC.

ฟิสิกเซ็นเตอร์.

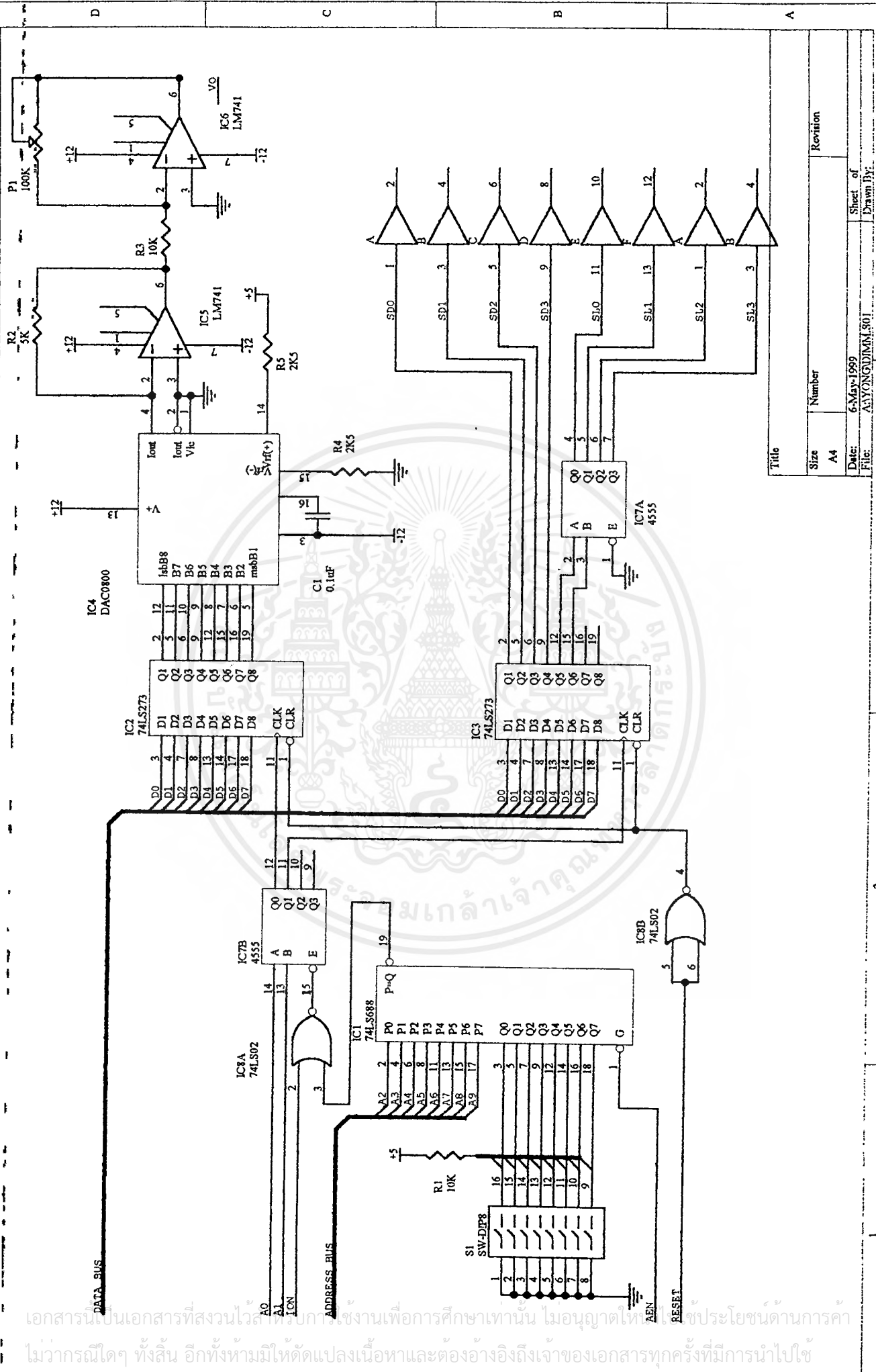
ปิยะ อำนวยพร.เรียนลัดจากตัวอย่างBORLAND DELPHI 3

บริษัท เฟิสท์ แปะซิฟิก มีเดีย (ไทยแลนด์) จำกัด, 2541

ALBERT PAVL PAUL MALVINO.DIGITAL COMPUTER ELECTRONICS AN
INTRODUCTION TO MICROCOMPUTER,Mc GRAW HILL,1995

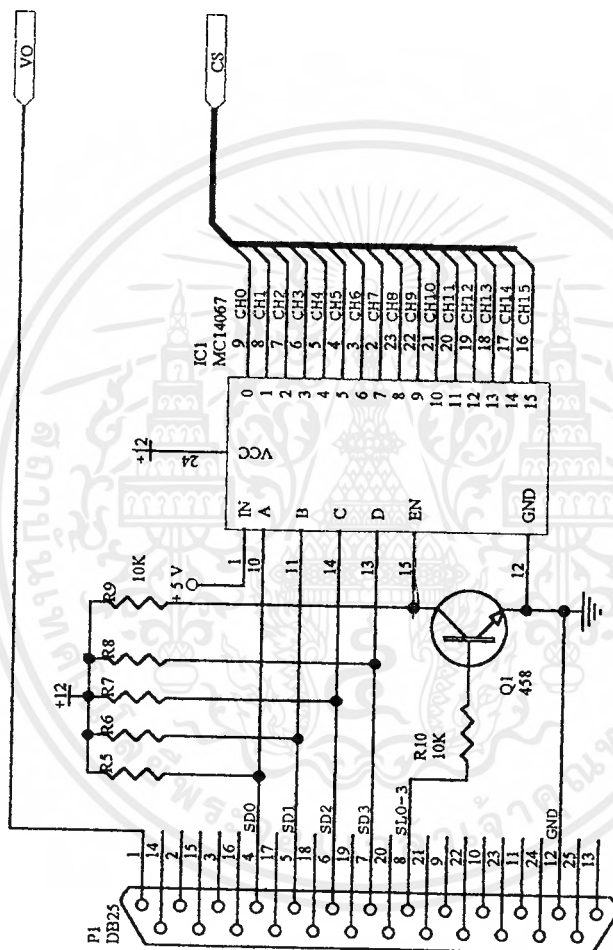


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



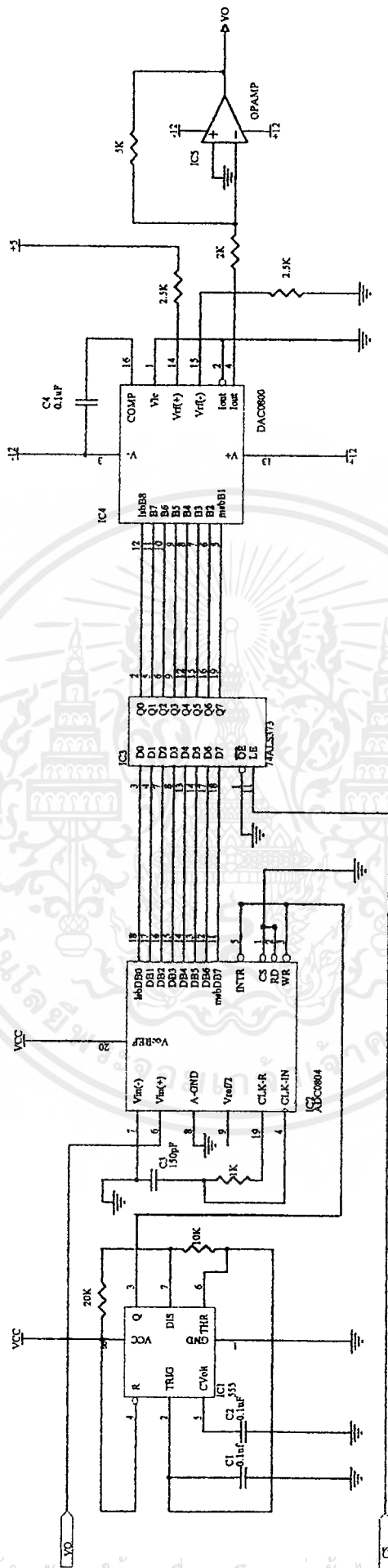
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title		Size	Number	Revision
		A4		
Date:	6 May 1999	Sheet of		
File:	ANYONGJIMM1801	Drawn by:		



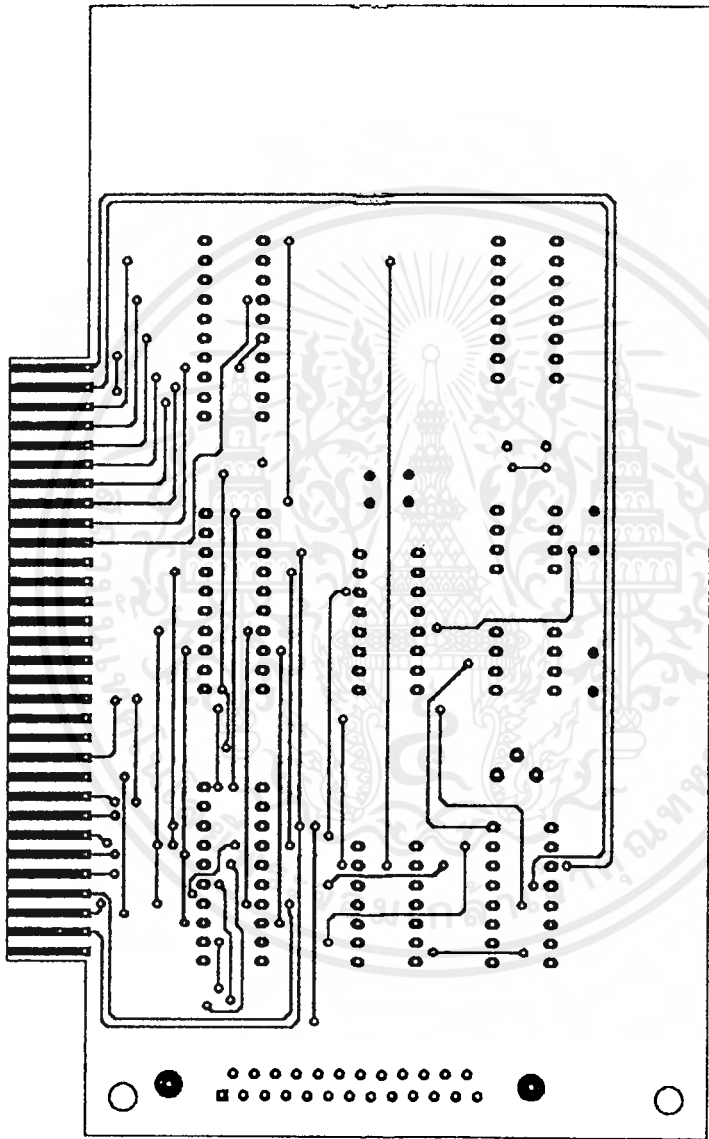
Title		Revision	
Size	Number	Revision	
A4			
Date:	6-May-1999	Sheet of	
File:	C:\DMM\16.SCH	Drawn By:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

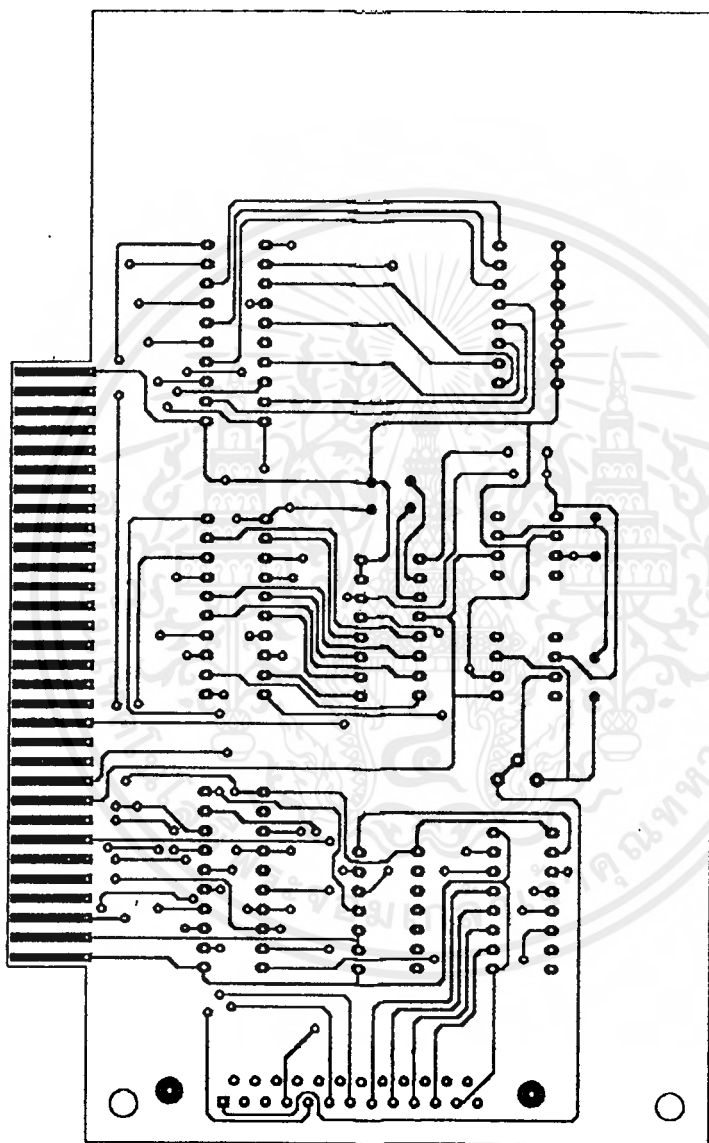


Title		Revision	
Size	Number	Revision	
D	D	Revision	
Drawn By	Drawn By	Revision	
Sheet 27	Sheet 27	Revision	
DATE	DATE	Revision	
FILE	FILE	Revision	

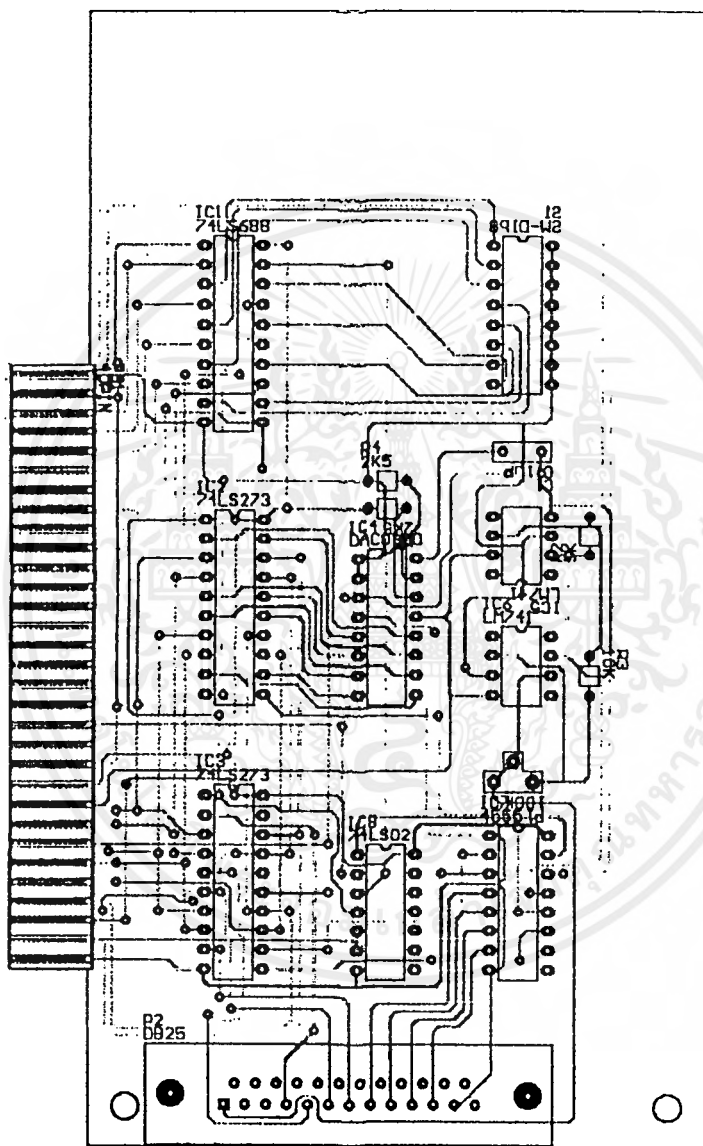
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



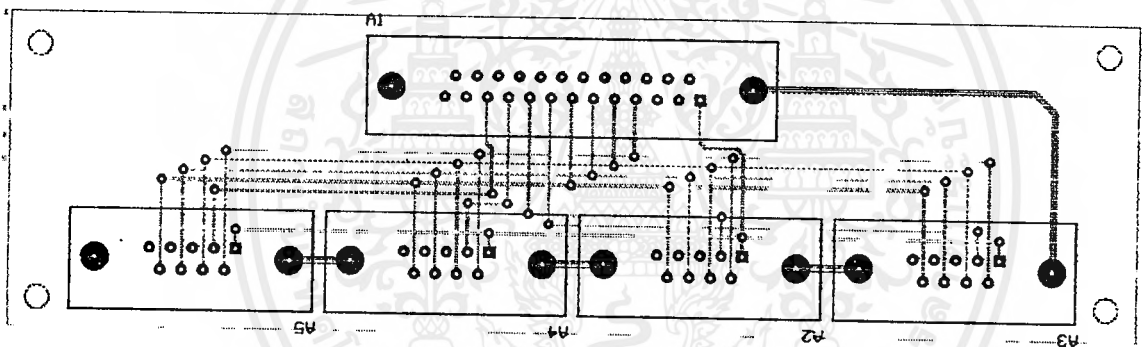
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



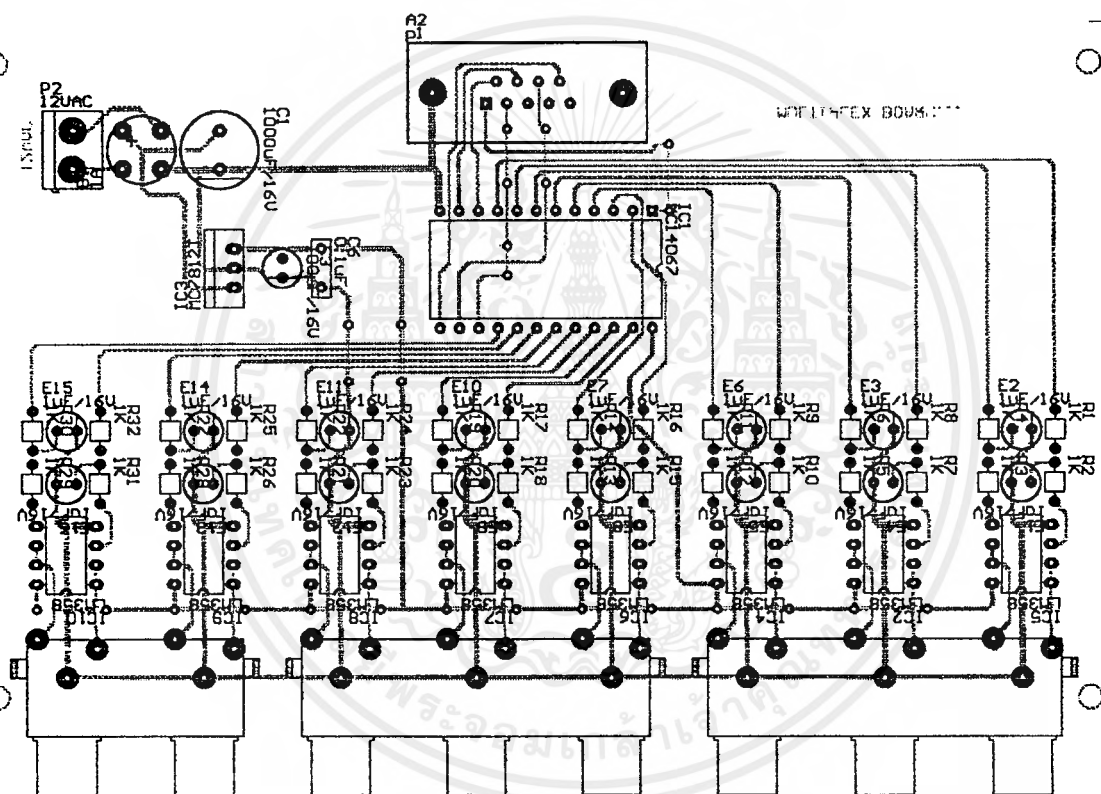
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



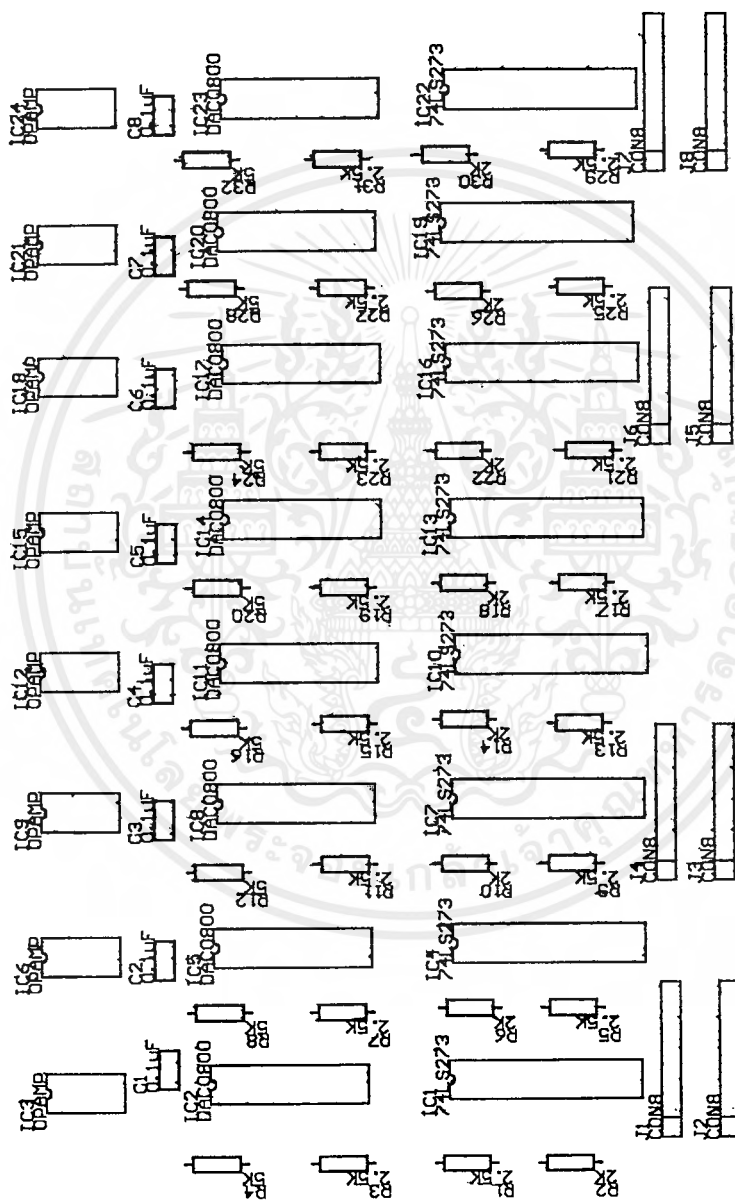
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

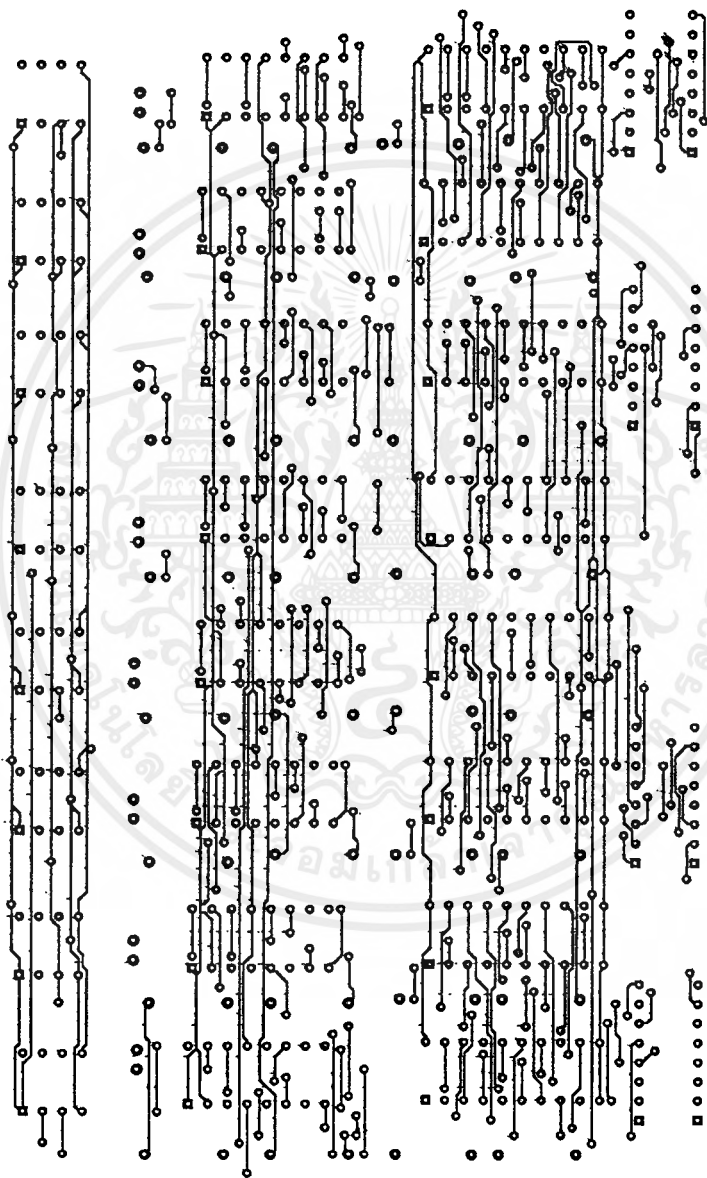


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

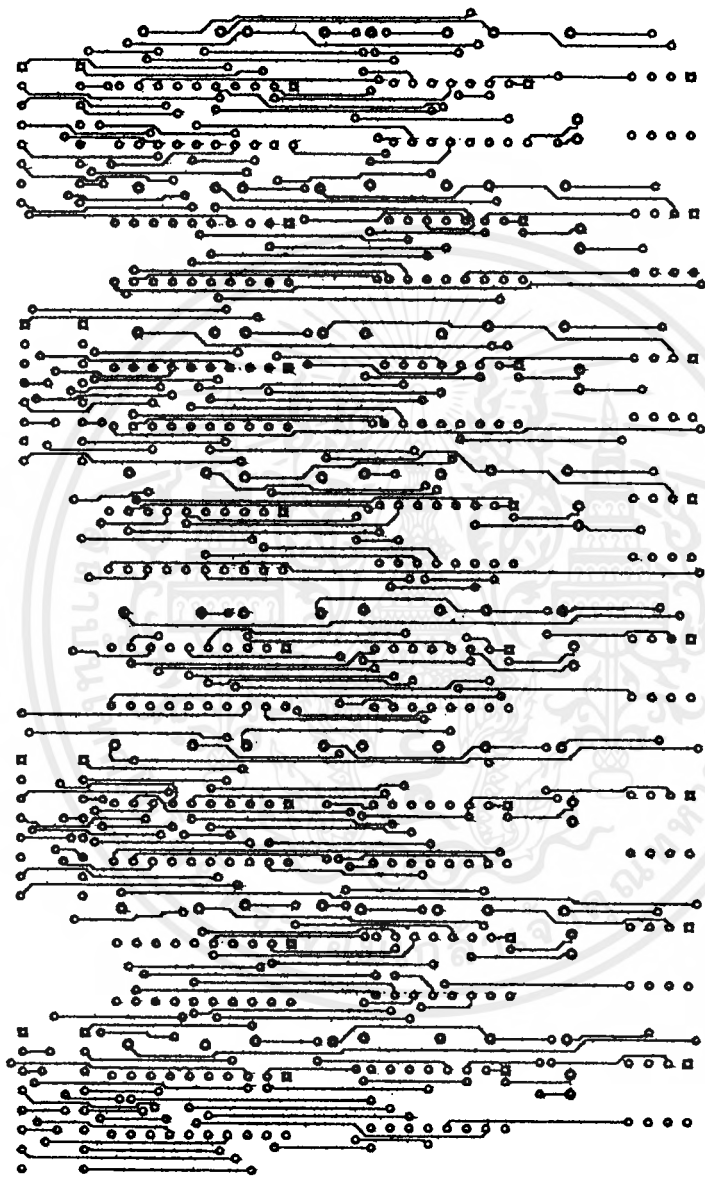


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



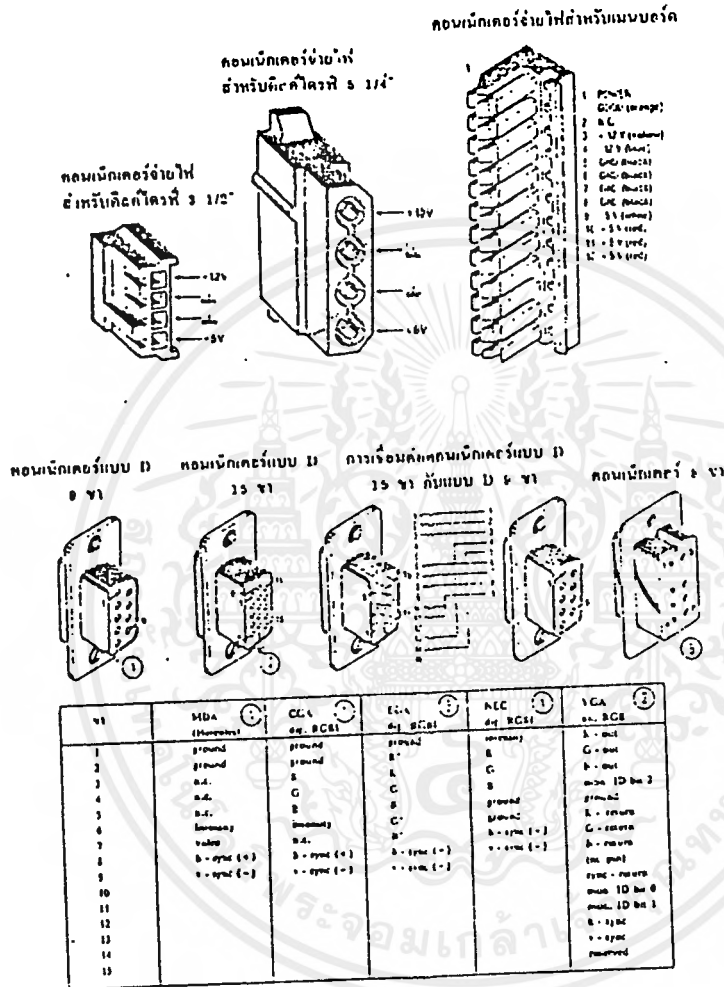


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



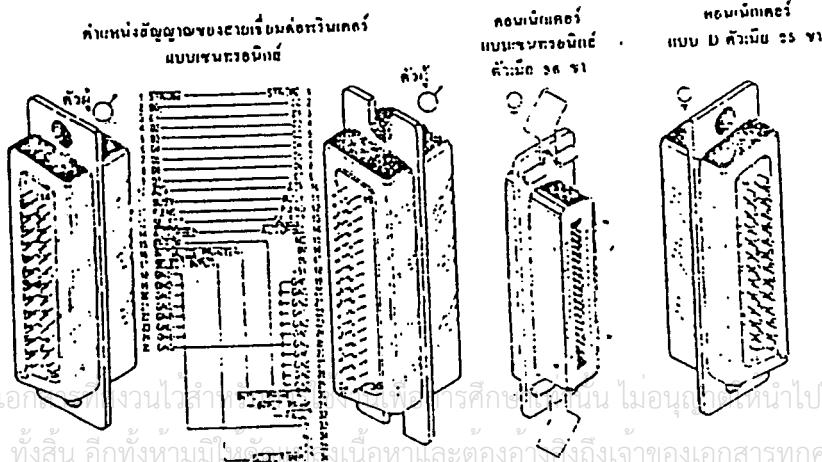
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONNECTOR กับตำแหน่งขาและสัญญาณ



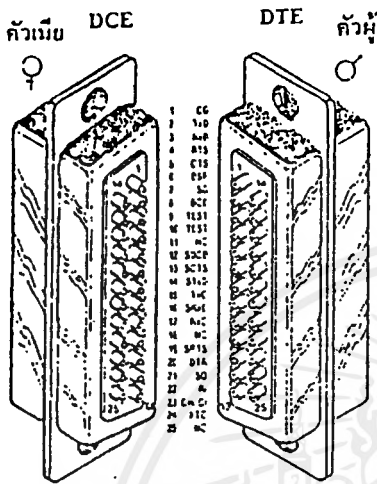
รูปที่ 1 คอนเนกเตอร์สำหรับภาคจ่ายไฟแบบทวีตซ์

รูปที่ 2 แสดงคอนเนกเตอร์และสัญญาณสำหรับจอภาพแบบต่าง ๆ

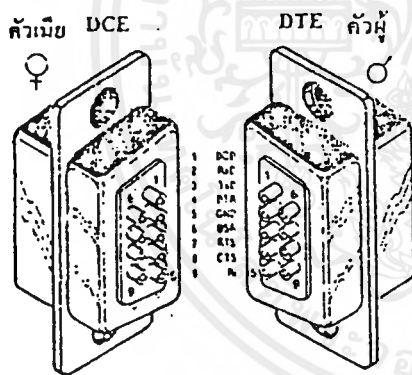


รูปที่ 3 สายเชื่อมต่อทรานซิสเตอร์

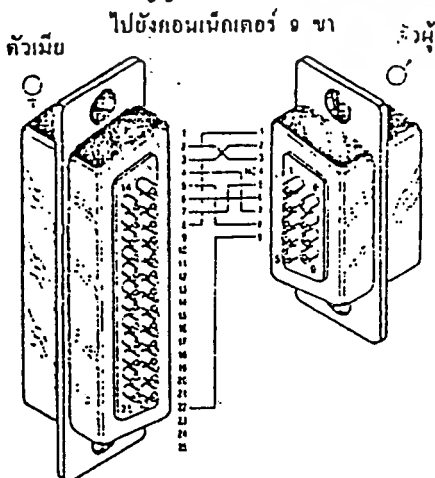
คอนเน็กเตอร์แบบ D 25 ขา



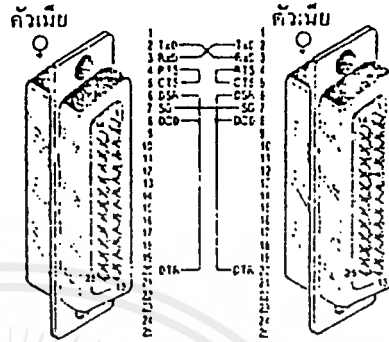
คอนเน็กเตอร์ ๑ ขา



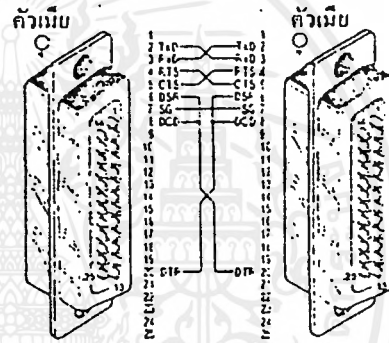
การเชื่อมต่อสัญญาณจากคอนเน็กเตอร์ 25 ขา



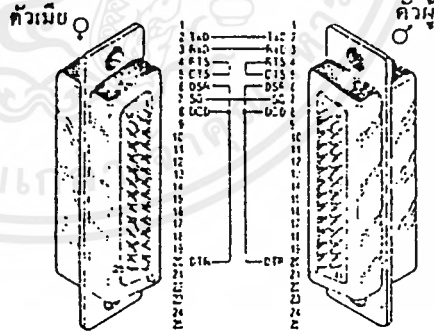
การเชื่อมต่อ DTE-DTE แบบ ๘ สาย



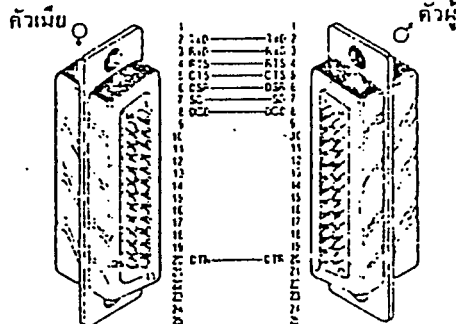
การเชื่อมต่อ DTE-DTE แบบ full connection



การเชื่อมต่อ DTE-DCE แบบ ๘ สาย



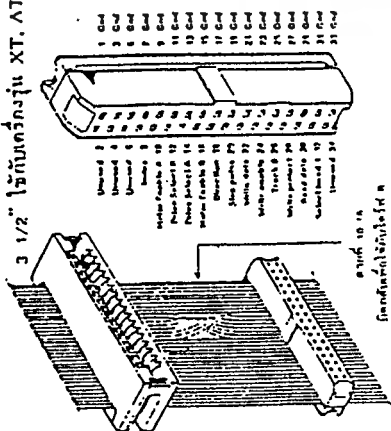
การเชื่อมต่อ DTE-DCE แบบ full connection



ฟลตเก็ดกัณศกัณศกัณศ

หวัคคัณศกัณศกัณศ

ว 1/2" ไรคัณศกัณศ XT, AT

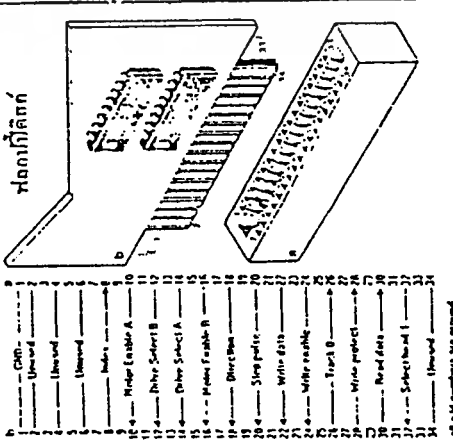


อณศ 10 14

กัณศกัณศกัณศ

คัณศกัณศกัณศกัณศ

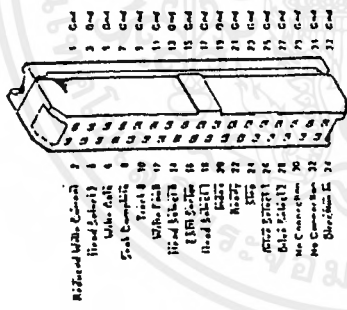
ฟลตเก็ดกัณศ



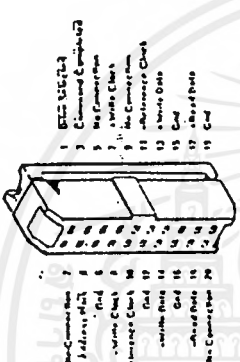
all pin numbers are ground

อณศกัณศกัณศกัณศ

คัณศกัณศ



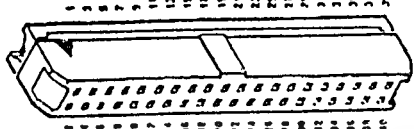
อณศ/คัณศ



คัณศกัณศกัณศกัณศ

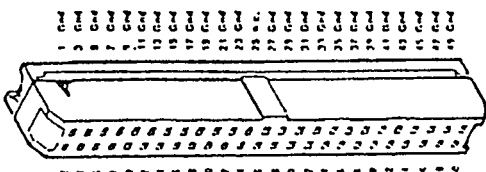


อณศกัณศกัณศกัณศ

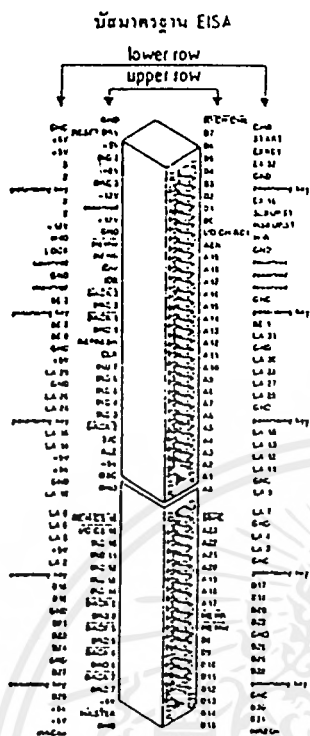


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CS0	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9	CS10	CS11	CS12	CS13	CS14	CS15	CS16	CS17	CS18	CS19	CS20	CS21	CS22	CS23	CS24	CS25	CS26	CS27	CS28	CS29	CS30
CS31	CS32	CS33	CS34	CS35	CS36	CS37	CS38	CS39	CS40	CS41	CS42	CS43	CS44	CS45	CS46	CS47	CS48	CS49	CS50	CS51	CS52	CS53	CS54	CS55	CS56	CS57	CS58	CS59	CS60	CS61
CS62	CS63	CS64	CS65	CS66	CS67	CS68	CS69	CS70	CS71	CS72	CS73	CS74	CS75	CS76	CS77	CS78	CS79	CS80	CS81	CS82	CS83	CS84	CS85	CS86	CS87	CS88	CS89	CS90	CS91	CS92
CS93	CS94	CS95	CS96	CS97	CS98	CS99	CS100	CS101	CS102	CS103	CS104	CS105	CS106	CS107	CS108	CS109	CS110	CS111	CS112	CS113	CS114	CS115	CS116	CS117	CS118	CS119	CS120	CS121	CS122	CS123

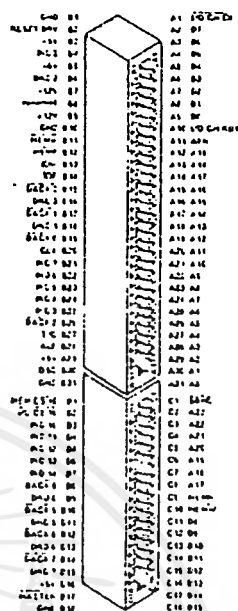
อณศกัณศกัณศกัณศ



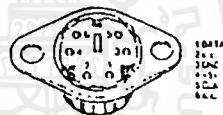
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CS0	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9	CS10	CS11	CS12	CS13	CS14	CS15	CS16	CS17	CS18	CS19	CS20	CS21	CS22	CS23	CS24	CS25	CS26	CS27	CS28	CS29	CS30
CS31	CS32	CS33	CS34	CS35	CS36	CS37	CS38	CS39	CS40	CS41	CS42	CS43	CS44	CS45	CS46	CS47	CS48	CS49	CS50	CS51	CS52	CS53	CS54	CS55	CS56	CS57	CS58	CS59	CS60	CS61
CS62	CS63	CS64	CS65	CS66	CS67	CS68	CS69	CS70	CS71	CS72	CS73	CS74	CS75	CS76	CS77	CS78	CS79	CS80	CS81	CS82	CS83	CS84	CS85	CS86	CS87	CS88	CS89	CS90	CS91	CS92
CS93	CS94	CS95	CS96	CS97	CS98	CS99	CS100	CS101	CS102	CS103	CS104	CS105	CS106	CS107	CS108	CS109	CS110	CS111	CS112	CS113	CS114	CS115	CS116	CS117	CS118	CS119	CS120	CS121	CS122	CS123



ต่อสายขั้วระบบมาตรฐาน IBM



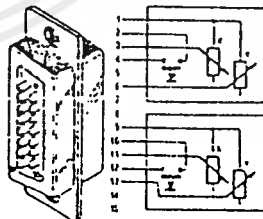
คอนเนกเตอร์สำหรับกับขั้วรุ่น PS2 (6 ขั้ว)



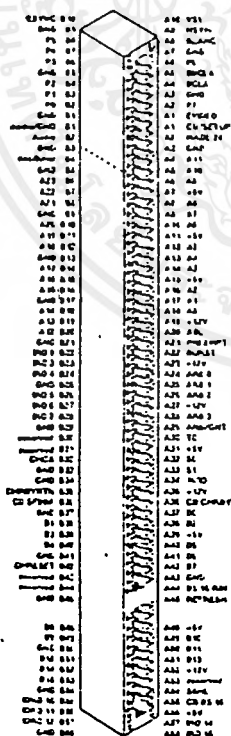
คอนเนกเตอร์ชนิดขั้วแบบ DIN (5 ขั้ว)



คอนเนกเตอร์สำหรับขั้วแบบ D (15 ขั้ว)



นิยามแถว MCA



54373/74373 Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power	
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package
T.I.	SN54LS373	14							SN74LS373	14								
FAIRCHILD	SN74LS373	14							SN74LS373	14								
MOTOROLA																		
N.S.C.																		
PHILIPS																		
SIGNETICS																		
SIEMENS																		
FUJITSU																		
HITACHI																		
mitsubishi																		
NEC																		
TOSHIBA																		

Electrical Characteristics SN54LS373 SN74LS373

absolute maximum ratings over operating free-air temperature range

Supply voltage V_{CC}	5V	Operating free-air temperature	SN54LS373: -55°C to +125°C SN74LS373: -40°C to +125°C
Input voltage	7V	Storage temperature range	-55°C to +150°C

recommended operating conditions

	SN54LS373	SN74LS373	UNIT
Supply voltage V_{CC}	4.5	5.0	V
High-level output current I_{OH}	1	2.6	mA
High-level output voltage V_{OH}	2.7	2.4	V
Pulse width t_p	15	15	ns
Setup time t_{SU}	10	10	ns
Hold time t_{H}	10	10	ns
Operating free-air temperature T_A	-55	0	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

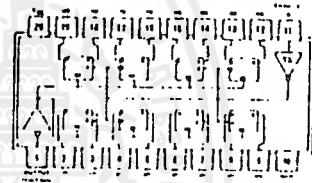
PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage			0.8		V
V_{IK} Input clamp voltage	V_{CC} MIN, I_I 10mA		1.5		V
V_{OH} High-level output voltage	V_{IL} V_{ILMAX} , I_{OH} MAX V_{CC} MIN	2.7	3.1		V
V_{OL} Low-level output voltage	V_{IH} 2V, I_{OL} 24mA		0.35	0.5	V
I_{OZH} Off-state output current, high-level voltage applied	V_{CC} MAX, V_{IH} 2V		20		µA
I_{OZL} Off-state output current, low-level voltage applied	V_{CC} MAX, V_{IL} 2V		20		µA
I_I maximum input current	V_{CC} MAX, V_I 2V		3.1		µA
I_{IH} High-level input current	V_{CC} MAX, V_I 2.7V		22		µA
I_{IL} Low-level input current	V_{CC} MAX, V_I 0.4V		2.8		µA
I_{OS} Short-circuit output current	V_{CC} MAX	30	132		mA
I_{CC} Supply current	Output control at 11V LS373		74	45	mA

switching characteristics, $V_{CC}=5V$, $T_A=25°C$

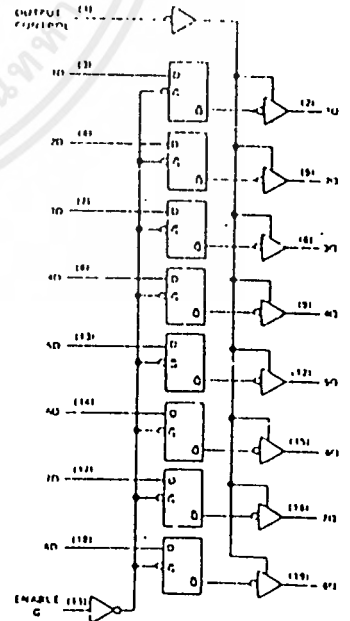
PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{max}					12	18	ns
t_{PLH}	Data	Any Q			12	18	ns
t_{PHL}	Any Q	Data			20	30	ns
t_{PLZ}	Output	Any Q	C_L 45pF, R_L 660Ω		15	28	ns
t_{PHZ}	Output	Any Q	See Notes 2 and 3		25	25	ns
t_{PLZ}	Output	Any Q	C_L 5pF, R_L 660Ω		12	20	ns
t_{PHZ}	Output	Any Q	See Note 3		15	23	ns

NOTE 1: For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
NOTE 2: All input values are at $V_{CC}=5V$, $T_A=25°C$.
NOTE 3: If the output is not in a valid state, the output current should be limited to 10mA and duration of the input current should not exceed 100ns.

Pin Assignments (Top View)



Pin	Function	Pin	Function
1	ENABLE	11	Q7
2	Q1	12	Q8
3	Q2	13	Q9
4	Q3	14	Q10
5	Q4		
6	Q5		
7	Q6		
8	Q1		
9	Q2		
10	Q3		
11	Q4		
12	Q5		
13	Q6		
14	Q7		



NOTE 2: Maximum clock frequency is tested with all outputs loaded.
NOTE 3: See test circuitry and waveforms on page 11.

NOTE 4: Maximum clock frequency
NOTE 5: Propagation delay time from input to output
NOTE 6: Output delay time from input to output
NOTE 7: Output delay time from input to output
NOTE 8: Output delay time from input to output
NOTE 9: Output delay time from input to output
NOTE 10: Output delay time from input to output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MM54HC688/MM74HC688
8-Bit Magnitude Comparator (Equality Detector)

General Description

This equality detector utilizes advanced silicon-gate CMOS technology to compare bit for bit two 8-bit words and indicates whether or not they are equal. The $P-U$ output indicates equality when it is low. A single active low enable is provided to facilitate cascading of several packages and enable comparison of words greater than 8 bits.

This device is useful in memory block decoding applications, where memory block enable signals must be generated from computer address information.

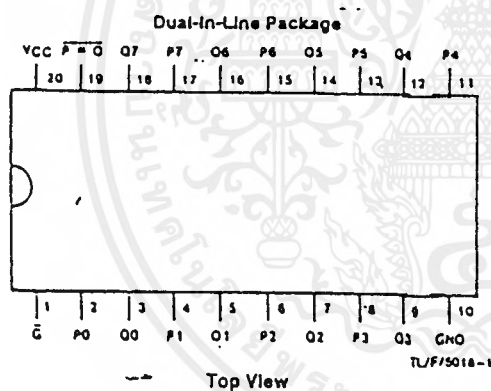
The comparator's output can drive 10 low power Schottky equivalent loads. This comparator is functionally and pin

compatible to the 54LS688/74LS688. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

Features

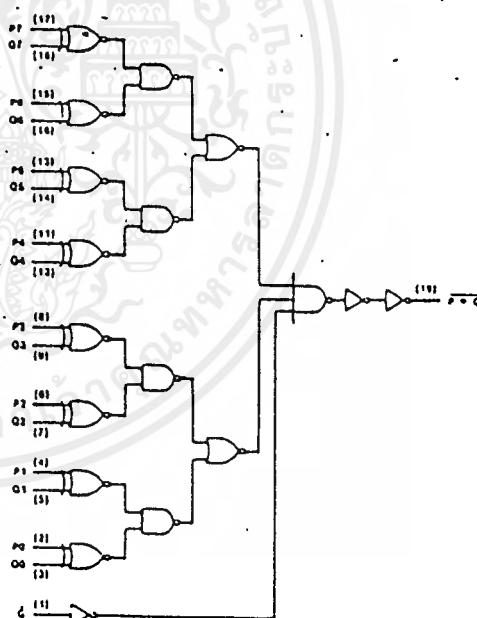
- Typical propagation delay: 20 ns
- Wide power supply range: 2-6V
- Low quiescent current: 80 μ A (74 Series)
- Large output current: 4 mA (74 Series)
- Same as 'HC521

Connection and Logic Diagrams



Order Number MM54HC688* or MM74HC688*

*Please look into Section 8, Appendix D for availability of various package types.



NW 15314-2

Truth Table

Inputs		$\overline{P=Q}$
Data	Enable $\overline{G}$	
P,Q		
$P = Q$	L	L
$P > Q$	L	H
$P < Q$	L	H
X	H	H



Digital-to-Analog Converters

DAC0808, DAC0807, DAC0806 8-Bit D/A Converters

general description

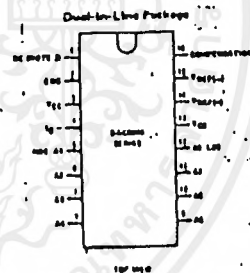
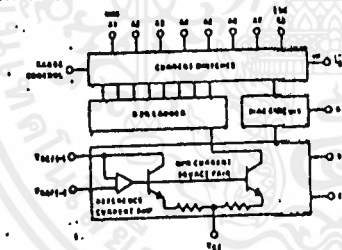
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 160 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of $256 I_{REF}/255$. Relative accuracies of better than $\pm 0.1\%$ assure 8-bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8-bit zero accuracy for $I_{REF} \geq 2 \mu A$. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1608/MC1408. For higher speed applications, see DAC0800 data sheet.

features

- Relative accuracy: $\pm 0.1\%$ error, maximum (DAC0808)
- Full scale current match: ± 1 LSB typ
- 7 and 8-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 160 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ μs
- Power supply voltage range: $\pm 4.5V$ to $\pm 18V$
- Low power consumption: 33 mW @ $\pm 5V$

block and connection diagrams



typical application

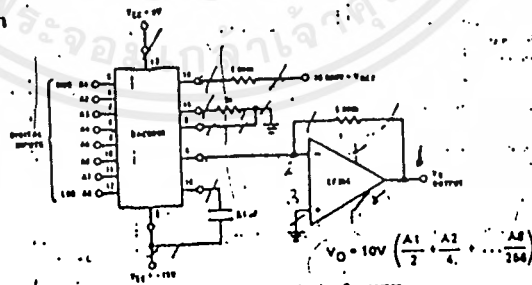


FIGURE 1. 10V Output Digital to Analog Converter

ordering information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS*			
		D PACKAGE (DIP16)	LM1608B	J PACKAGE (J16A)	N PACKAGE (N16A)
8-bit	$-55^{\circ}C \leq T_A \leq +125^{\circ}C$	DAC0808LD	LM1608B	DAC0808LJ	DAC0808LN
8-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$			DAC0808LJ	LM1408B-8
7-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$			DAC0807LJ	DAC0807LN
6-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$			DAC0806LJ	DAC0806LN

*Note: Devices may be ordered by using either order number.

DAC0808, DAC0807,
DAC0806

3

DAC0808, DAC0807,
DAC0806

absolute maximum ratings (T_A = 25°C unless otherwise noted)

Power Supply Voltage			Power Dissipation (Package Limitations)		
VCC		5.5 VDC	Cavity Package		1000 mW
VEE		-18.5 VDC	Derate above TA = 25°C		8.7 mW/°C
Digital Input Voltage, VI1-VI12	-10 VDC to +18 VDC		Operating Temperature Range		
Applied Output Voltage, VO	-11 VDC to +18 VDC		DAC0808L	-55°C ≤ TA ≤ +125°C	
Reference Current, IREF	8 mA		DAC0808LC Series	0 ≤ TA ≤ +75°C	
Reference Amplifier Inputs, VI4, VI5	VCC, VEE		Storage Temperature Range	-55°C to +150°C	

electrical characteristics

(VCC = 5V, VEE = -18 VDC, VREF/R14 = 2 mA, DAC0808L: T_A = -55°C to +125°C, DAC0808LC, DAC0807LC, DAC0806LC, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

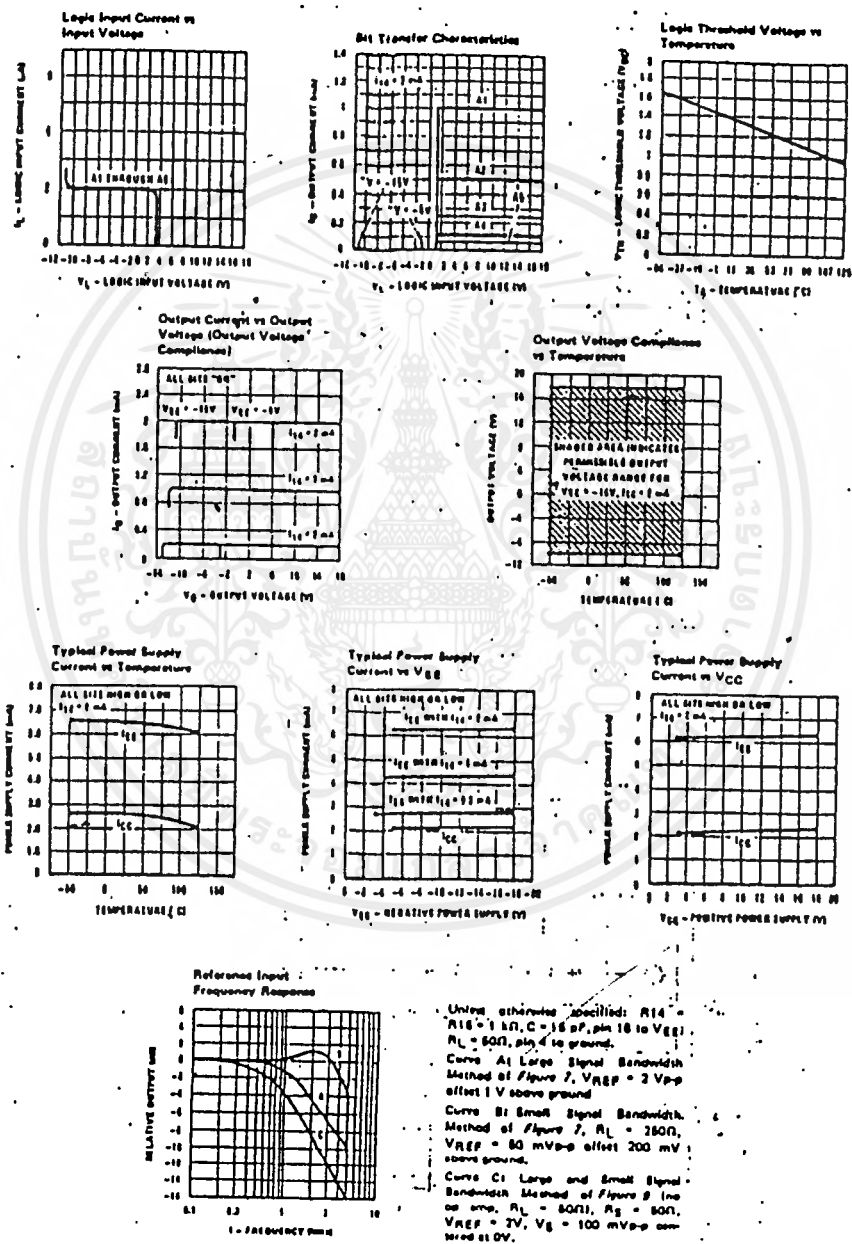
PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
E_1	Relative Accuracy (Error Relative to Full Scale I_O) (Figure 4) DAC0808L (LM1408 B) DAC0808LC (LM1408 B) DAC0807LC (LM1408 B), (Note 1) DAC0806LC (LM1408 B), (Note 1)			±0.10 ±0.20 ±0.20 ±0.20	%
	Settling Time to Within 1/2 LSB (Includes t_{PLH}) $T_A = 25^\circ\text{C}$ (Note 2), (Figure 5)		150		ns
t_{PLH} , t_{PML}	Propagation Delay Time $T_A = 25^\circ\text{C}$, (Figure 6)		30	100	ns
TCIQ	Output Full Scale Current Drift		120		ppm/°C
MSB	Digital Input Logic Levels (Figure 3)				
V_{IH}	High Level, Logic "1"	2			VDC
V_{IL}	Low Level, Logic "0"			0.8	VDC
MSB	Digital Input Current (Figure 3) High Level $V_{IH} = 8\text{V}$ Low Level $V_{IL} = 0.8\text{V}$		8 -0.003	0.040 -0.8	mA mA
I_{IS}	Reference Input Bias Current (Figure 3)		-1	-8	μA
	Output Current Range (Figure 3) $V_{EE} = -8\text{V}$ $V_{EE} = -18\text{V}$, $T_A = 25^\circ\text{C}$	0 8	2.8 2.8	2.1 4.2	mA mA
I_O	Output Current $V_{REF} = 2.000\text{V}$, $R_{14} = 1000\Omega$, (Figure 3)	1.8	1.80	2.1	mA
	Output Current, All Bits Low (Figure 3)		0	4	μA
	Output Voltage Compliance Pin 1 Grounded, V_{EE} Below -10V	$I_O \leq 0.10\text{A}$, $T_A = 25^\circ\text{C}$		-0.85, +0.4 -0.8, +0.4	VDC VDC
SA_{REF}	Reference Current Bias Rate (Figure 6)		8		mA/μs
	Output Current Power Supply Sensitivity $-8\text{V} \leq V_{EE} \leq -18.5\text{V}$		0.06	2.7	μA/V
	Power Supply Current (All Bits Low) (Figure 3)				
I_{CC}			2.3	22	mA
I_{EE}			-4.3	-13	mA
V_{CC} V_{EE}	Power Supply Voltage Range $T_A = 25^\circ\text{C}$, (Figure 3)	4.8 -4.8	5.0 -15	5.8 -18.8	VDC VDC
	Power Dissipation All Bits Low $V_{CC} = 5\text{V}$, $V_{EE} = -8\text{V}$ $V_{CC} = 5\text{V}$, $V_{EE} = -18\text{V}$ $V_{CC} = 18\text{V}$, $V_{EE} = -8\text{V}$ $V_{CC} = 18\text{V}$, $V_{EE} = -18\text{V}$		33 108 80 180	170 308	mW mW mW mW

Note 1: All current sources are tested to guarantee at least 80% of rated current.
 Note 2: All bits switched.
 Note 3: Range control is not required.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

typical performance characteristics

VCC = 5V, VEE = -15V, TA = 25°C, unless otherwise noted



Unless otherwise specified: $R_{14} = 1k\Omega$, $C = 15pF$, pin 16 to V_{EE} , $R_L = 50\Omega$, pin 4 to ground.
Curve A: Large Signal Bandwidth Method of Figure 7, $V_{REF} = 2V$ p-p offset 1V above ground.
Curve B: Small Signal Bandwidth Method of Figure 7, $R_L = 250\Omega$, $V_{REF} = 50mV$ p-p offset 200mV above ground.
Curve C: Large and Small Signal Bandwidth Method of Figure 8 (no on amp), $R_L = 50\Omega$, $R_2 = 50\Omega$, $V_{REF} = 2V$, $V_2 = 100mV$ p-p centered at 0V.

DAC0808, DAC0807,
DAC0806

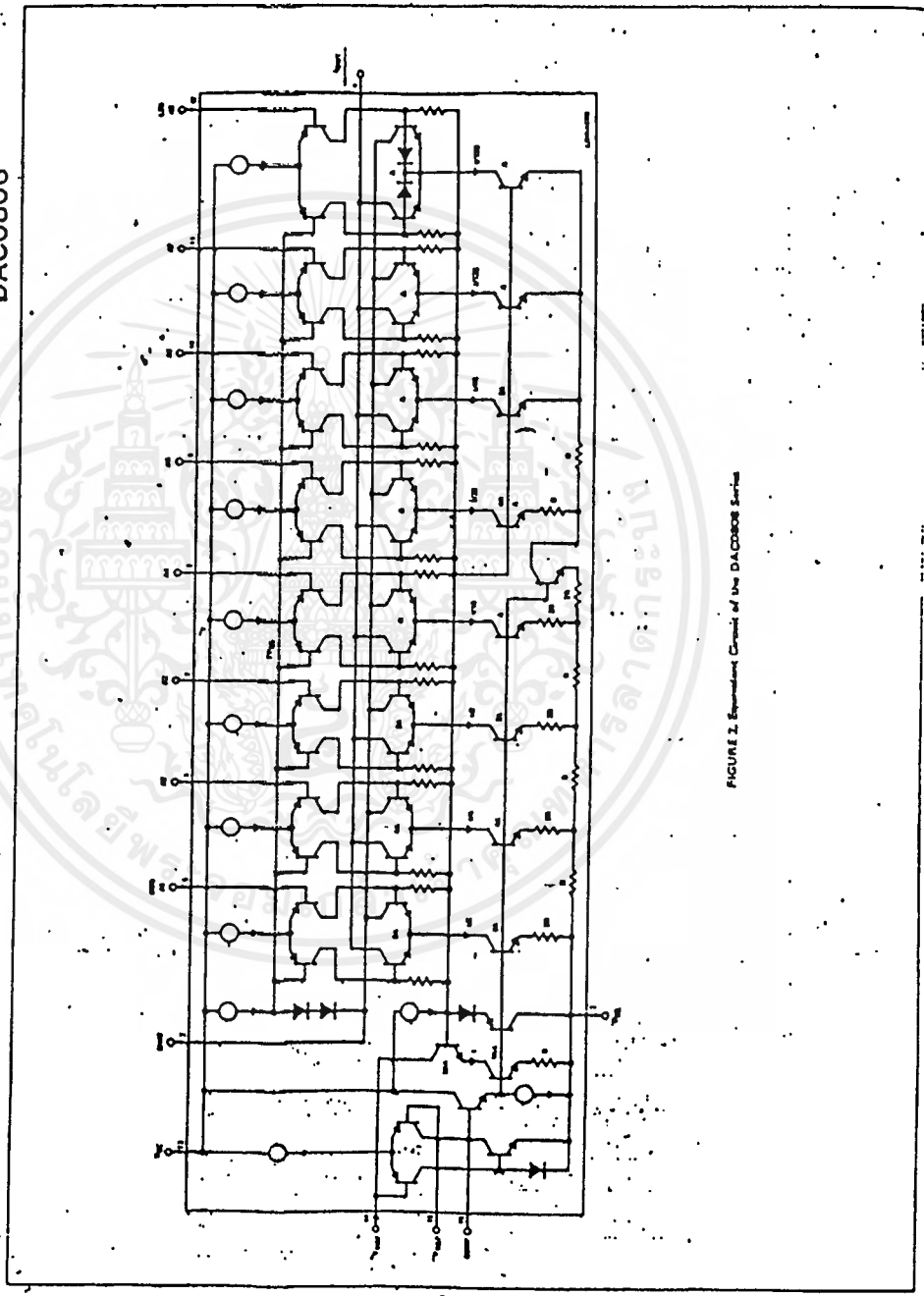
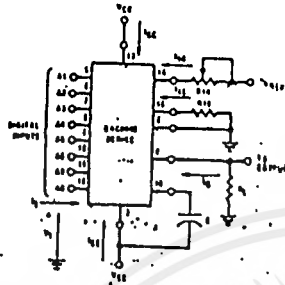


FIGURE 2. Schematic Circuit of the DAC0808 Series

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

test circuits



V_i and I_i apply to inputs A1-A8.

The resistor tied to pin 16 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_4}{16} + \frac{A_8}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right)$$

$$\text{where } K = \frac{V_{REF}}{R_{14}}$$

and $A_N = -1$ if A_N is at high level

$A_N = 0$ if A_N is at low level

FIGURE 3. Notation Definitions Test Circuit

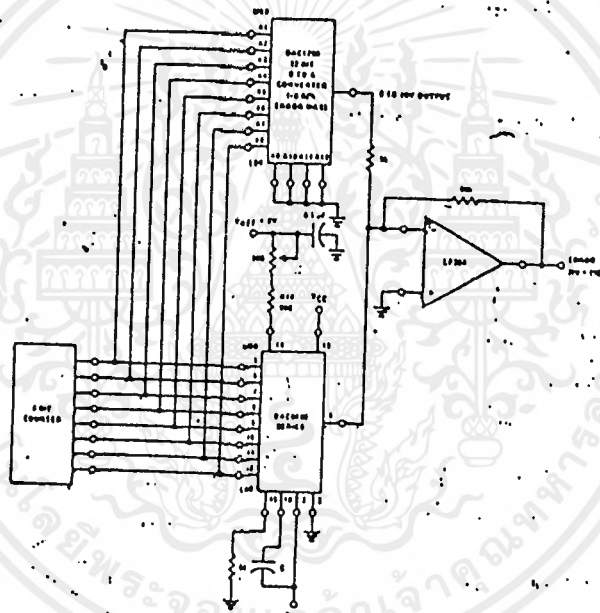


FIGURE 4. Relative Accuracy Test Circuit



FIGURE 5. Transient Response and Settling Time

test circuits (Continued)

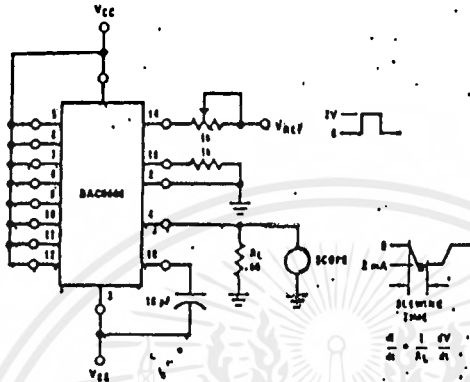


FIGURE 6. Reference Current Step Response Measurement

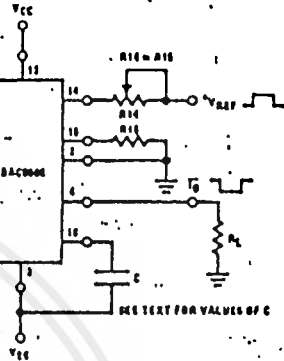


FIGURE 7. Positive V_{REF}

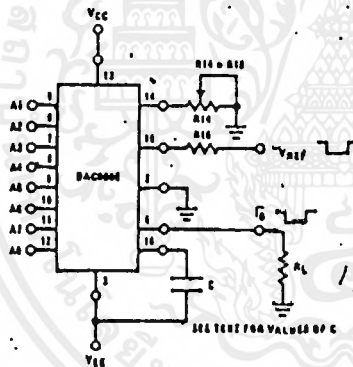


FIGURE 8. Negative V_{REF}

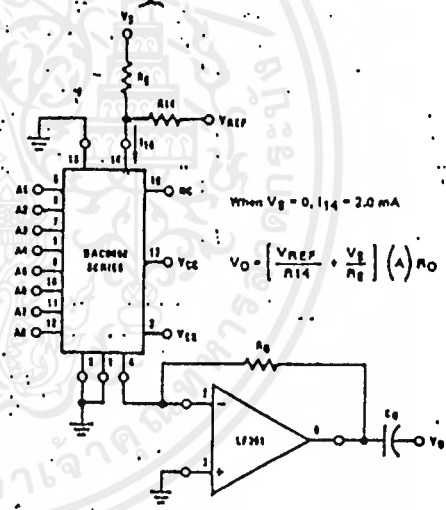


FIGURE 9. Programmable Gain Amplifier or Digital Attenuator Circuit

application hints

REFERENCE AMPLIFIER DRIVE AND COMPENSATION

The reference amplifier provides a voltage at pin 14 for converting the reference voltage to a current, and a turn-around circuit or current mirror for feeding the ladder. The reference amplifier input current, I_{14} , must always flow into pin 14, regardless of the set-up method or reference voltage polarity.

Connections for a positive voltage are shown in Figure 7. The reference voltage source supplies the full current

I_{14} . For bipolar reference signals, as in the multiplying mode, R_{15} can be tied to a negative voltage corresponding to the minimum input level. It is possible to eliminate R_{15} with only a small sacrifice in accuracy and temperature drift.

The compensation capacitor value must be increased with increases in R_{14} to maintain proper phase margin; for R_{14} values of 1, 2.5 and 5 k Ω , minimum capacitor values are 16, 37 and 76 pF. The capacitor may be tied to either VEE or ground, but using VEE increases negative supply rejection.

application hints (Continued)

A negative reference voltage may be used if R14 is grounded and the reference voltage is applied to R16 as shown in Figure 8. A high input impedance is the main advantage of this method. Compensation involves a capacitor to VEE on pin 16, using the values of the previous paragraph. The negative reference voltage must be at least 4V above the VEE supply. Bipolar input signals may be handled by connecting R14 to a positive reference voltage equal to the peak positive input level at pin 16.

When a DC reference voltage is used, capacitive bypass to ground is recommended. The 5V logic supply is not recommended as a reference voltage. If a well regulated 5V supply which drives logic is to be used as the reference, R14 should be decoupled by connecting it to 5V through another resistor and bypassing the junction of the 2 resistors with 0.1 μ F to ground. For reference voltages greater than 8V, a clamp diode is recommended between pin 14 and ground.

If pin 14 is driven by a high impedance such as a transistor current source, none of the above compensation methods apply and the amplifier must be heavily compensated, decreasing the overall bandwidth.

OUTPUT VOLTAGE RANGE

The voltage on pin 4 is restricted to a range of -0.8 to 0.5V when VEE = -5V due to the current switching methods employed in the DAC0808.

The negative output voltage compliance of the DAC0808 is extended to -5V where the negative supply voltage is more negative than -10V. Using a full-scale current of 1.992 mA and load resistor of 2.5 k Ω between pin 4 and ground will yield a voltage output of 256 levels between 0 and -4.980V. Floating pin 1 does not affect the converter speed or power dissipation. However, the value of the load resistor determines the settling time due to increased voltage swing. Values of R_L up to 500 Ω do not significantly affect performance, but a 2.5 k Ω load increases worst-case settling time to 1.2 μ s (when all bits are switched ON). Refer to the subsequent text section on Settling Time for more details on output loading.

OUTPUT CURRENT RANGE

The output current maximum rating of 4.2 mA may be used only for negative supply voltages more negative than -7V, due to the increased voltage drop across the resistors in the reference current amplifier.

ACCURACY

Absolute accuracy is the measure of each output current level with respect to its intended value, and is dependent upon relative accuracy and full-scale current drift. Relative accuracy is the measure of each output current level as a fraction of the full-scale current. The relative accuracy of the DAC0808 is essentially constant with temperature due to the excellent temperature tracking

of the monolithic resistor ladder. The reference current may drift with temperature, causing a change in the absolute accuracy of output current. However, the DAC0808 has a very low full-scale current drift with temperature.

The DAC0808 series is guaranteed accurate to within $\pm 1/2$ LSB at a full-scale output current of 1.992 mA. This corresponds to a reference amplifier output current drive to the ladder network of 2 mA, with the loss of 1 LSB (8 μ A) which is the ladder remainder shunted to ground. The input current to pin 14 has a guaranteed value of between 1.9 and 2.1 mA, allowing some mismatch in the NPN current source pair. The accuracy test circuit is shown in Figure 4. The 12-bit converter is calibrated for a full-scale output current of 1.992 mA. This is an optional step since the DAC0808 accuracy is essentially the same between 1.8 and 2.8 mA. Then the DAC0808 circuit's full-scale current is trimmed to the same value with R14 so that a zero value appears at the error amplifier output. The counter is activated and the error band may be displayed on an oscilloscope, detected by comparators, or stored in a peak detector.

Two 8-bit D-to-A converters may not be used to construct a 16-bit accuracy D-to-A converter. 16-bit accuracy implies a total error of $\pm 1/2$ of one part in 65,536, or $\pm 0.00078\%$, which is much more accurate than the $\pm 0.019\%$ specification provided by the DAC0808.

MULTIPLYING ACCURACY

The DAC0808 may be used in the multiplying mode with 8-bit accuracy when the reference current is varied over a range of 256:1. If the reference current in the multiplying mode ranges from 16 μ A to 4 mA, the additional error contributions are less than 1.8 μ A. This is well within 8-bit accuracy when referred to full-scale.

A monotonic converter is one which supplies an increase in current for each increment in the binary word. Typically, the DAC0808 is monotonic for all values of reference current above 0.5 mA. The recommended range for operation with a DC reference current is 0.5 to 4 mA.

SETTLING TIME

The worst-case switching condition occurs when all bits are switched ON, which corresponds to a low-to-high transition for all bits. This time is typically 150 ns for settling to within $\pm 1/2$ LSB, for 8-bit accuracy, and 100 ns to $1/2$ LSB for 7 and 6-bit accuracy. The turn OFF is typically under 100 ns. These times apply when $R_L \leq 500\Omega$ and $C_O \leq 25$ pF.

Extra care must be taken in board layout since this is usually the dominant factor in satisfactory test results when measuring settling time. Short leads, 100 pF supply bypassing for low frequencies, and minimum scope lead length are all mandatory.