

พี.ซี ควบคุม A/D, D/A 48 ช่อง

P.C CONTROL A/D, D/A 48 CHANNEL



ปริญาานิพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033279

หัวข้อปริญญาบัตร

คอมพิวเตอร์ควบคุม A/D และ D/A 48 แชนแนล

โดย

นายพิมลศักดิ์ ลิ้มปัสุทธิพงศ์

นายภักดี คงผล

อาจารย์ที่ปรึกษา

อาจารย์อุทัย ศรีธีระวิโรจน์

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2536

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับปริญญาบัตรฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญาบัตร

-ประธานกรรมการ
- ()
-กรรมการ
- ()
-กรรมการ
- ()
-กรรมการ
- ()
-กรรมการ
- ()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ควบคุมอนาลอกเป็นดิจิทัล, ดิจิตอลเป็นอนาลอก
PC CONTROL A/D, D/A

โดย	นายพิมลศักดิ์	ลิมป์สุทธิพงศ์
	นายภักดี	คงผล
อาจารย์ที่ปรึกษา	อาจารย์อุทัย	ศรีธีระวิโรจน์

บทคัดย่อ

ปัจจุบันนี้ได้มีการนำเอาคอมพิวเตอร์มาใช้งานอย่างแพร่หลาย ฟีซีควบคุม A/D, D/A เป็นตัวอย่างหนึ่งของงานด้านวิศวกรรมซึ่งมีลักษณะดังนี้

- การควบคุมโวลท์ที่เจดด้วยฮอนาล็อกเป็นการควบคุมง่าย แต่เมื่อเปลี่ยนโวลท์ต้องเปลี่ยน Hard Ware ด้วยทำให้เกิดความยุ่งยาก

- การควบคุมด้วยคอมพิวเตอร์ เป็นที่นิยมกันมากในปัจจุบันนี้เพราะการควบคุมด้วยคอมพิวเตอร์เราสามารถควบคุมทางด้าน Soft Ware ทำให้การเปลี่ยนแปลงรูปแบบการควบคุมเพียงแต่แก้ไขในส่วนของ Soft Ware เท่านั้น

- การควบคุมด้วยคอมพิวเตอร์สามารถควบคุมได้หลายชนแนลพร้อม ๆ กัน โดยที่ Hard Ware มีขนาดเล็กถ้าควบคุมด้วยฮอนาล็อก Hard Ware มีขนาดใหญ่ตามจำนวนชนแนล

Abstract

Today. Every work control by computer. Example , PC control Analog singnal convert to digital singnal and digital singnal convert to analog singnal. It have operated

- Analog singnal control is very easy. but when you change voltage you must change of hardware.

- present we have use computer control very much because computer control can change easy software

- computer control. It have small size , light and can use mutiple channal control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดีเพราะว่า ได้รับคำแนะนำข้อมูลเกี่ยวกับวงจร A/D, D/A จากอาจารย์อภัย ศรีธีรวิโรจน์ซึ่งเป็นอาจารย์ที่ปรึกษา ส่วนข้อมูลทางด้านเทคนิคได้รับคำแนะนำจากอาจารย์อรรถสิทธิ์ ซึ่งเป็นอาจารย์ในภาควิชาเทคนิคอุตสาหกรรมและขอขอบคุณ คุณชาญศักดิ์ มงคลบริลักษ์ ให้ค่าปรึกษาโปรแกรมกราฟิก, ขอขอบคุณ คุณนงลักษณ์ เพ็ญเพชร ที่ช่วยเหลือด้านกราฟิกและคุณกิตติศักดิ์ จิตเกื้อ ที่ให้ค่าปรึกษาทางด้านภาษาซี ตลอดจนเพื่อนๆทุกคนในภาควิชาที่ให้กำลังใจมาตลอดจนกระทั่งปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทนำ.....	1
-ความเป็นมาของโครงการ.....	1
-วัตถุประสงค์ของโครงการ.....	1
-ขอบเขตของโครงการ.....	1
-ประโยชน์.....	2
บทที่ 1 ทฤษฎีการทำงานของ 8255 เบื้องต้น.....	3
-ชาตต่างๆของ 8255.....	5
-Mode ต่างๆของ 8255.....	6
-การเชื่อมต่อกับ Z80.....	8
-รายละเอียดโหมด 8255.....	13
-Mode 0.....	13
-Mode 1.....	18
-Mode 2.....	22
บทที่ 2 การอินเตอร์เฟส.....	26
-การจัดแอดเดรสสำหรับอินพุตเอาต์พุตใน IBM PC.....	26
-เทคนิคการดีโค้ดแอดเดรส.....	28
-เทคนิคการดีโค้ดตามลักษณะพอร์ต.....	30
-อินเตอร์เฟสการ์ด.....	34
-การทดสอบและการทำงาน.....	35
-วงจรรีโค้ด.....	35
-การใช้งาน 8255 อินเตอร์เฟสการ์ด.....	36
บทที่ 3 วงจร A/D,D/A.....	40
-วงจร D/A แบบใช้ความต้านทานหลายค่า.....	40
-วงจร D/A แบบใช้ IC.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-การทำงานของวงจร D/A ที่ใช้ในโครงการ.....	43
-วงจร A/D แบบใช้วงจรเปรียบเทียบ.....	43
-การทำงานของวงจร A/D ที่ใช้ในโครงการ.....	45
บทที่ 4 วงจรดีโค้ดแชนแนล.....	49
บทที่ 5 Softwareและการใช้งาน.....	56
-การใช้งานโปรแกรม.....	56
-ฟังก์ชัน Auto.....	57
-ฟังก์ชัน Out.....	58
-ฟังก์ชัน D/A.....	59
-ฟังก์ชัน Save.....	61
-ฟังก์ชัน Load.....	61
-ฟังก์ชัน Clear.....	62
-ฟังก์ชัน A/D.....	63
-ฟังก์ชัน Exit.....	63
-ฟังก์ชัน Os-shell.....	63
-ฟังก์ชัน Help.....	63
-สรุปการใช้ Hot Key.....	64
บทที่ 6 บทสรุปและข้อเสนอแนะ.....	65
เอกสารอ้างอิง.....	67
ภาคผนวก.....	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รูปที่ 2.10 แสดงแผ่นปรีนและการวางอุปกรณ์.....	39
บทที่ 3	วงจร A/D, D/A	49
	รูปที่ 3.1 แสดงวงจร D/A แบบใช้ ค.ต.ท หลายค่า.....	41
	รูปที่ 3.2 แสดงวงจร ดิจูเอ MC1408.....	42
	รูปที่ 3.3 แสดงการต่อวงจร A/D แบบ"เฟรช".....	44
	รูปที่ 3.4 แสดงแผ่นปรีนและอุปกรณ์ของวงจร D/A.....	46
	รูปที่ 3.5 แสดงแผ่นปรีนและอุปกรณ์ของวงจร A/D.....	46
	รูปที่ 3.6 แสดงวงจร D/A.....	47
	รูปที่ 3.7 แสดงวงจร A/D.....	48
บทที่ 4	วงจร Decode channal.....	49
	รูปที่ 4.1 แสดงการติดตั้งอุปกรณ์บนแผ่นปรีนวงจรดีโค้ด.....	50
	รูปที่ 4.2 วงจรดีโค้ด 16 แชนแนล.....	51
	รูปที่ 4.3 วงจร A/D, D/A, ดีโค้ด บนปรีน.....	52
	รูปที่ 4.4 แสดงวงจรดีโค้ด 48 แชนแนล.....	53
	รูปที่ 4.5 แสดงการติดตั้งภายในกล่อง.....	54
	รูปที่ 4.6 แสดงด้านหน้ากล่อง.....	54
	รูปที่ 4.7 แสดงหลังกล่อง.....	55
บทที่ 5	Software และการนำไปใช้งาน.....	56
	รูปที่ 5.1 Demo of convert plus 4.2.....	56
	รูปที่ 5.2 Working area และbutton function.....	57
	รูปที่ 5.3 ไคอะลือกของฟังก์ชัน Auto.....	58
	รูปที่ 5.4 ไคอะลือกของฟังก์ชัน Out.....	59
	รูปที่ 5.5 ไคอะลือกของฟังก์ชัน D/A.....	60
	รูปที่ 5.6 แสดงการใช้ฟังก์ชัน D/A.....	61
	รูปที่ 5.7 ไคอะลือกของฟังก์ชัน Clear.....	62
	รูปที่ 5.8 ไคอะลือกของฟังก์ชัน Exit.....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
บทหน้า.....	1
บทที่ 1 ทฤษฎีการทำงานของ 8255 เบื้องต้น.....	3
ตารางที่ 1.1 แสดงผลของสัญญาณควบคุมต่างๆที่มีผลต่อ 8255.....	12
ตารางที่ 1.1 แสดงผลของสัญญาณควบคุมต่างๆที่มีผลต่อ 8255 (ต่อ).....	13
ตารางที่ 1.2 ตารางแสดงตำแหน่งขากรณีเป็นอินพุตเข้าที่พุก ของ 8255.....	19
ตารางที่ 1.3 ตารางแสดงขาของ Port C.....	24
บทที่ 2 การอินเตอร์เฟส.....	26
ตารางที่ 2.1 การใช้งานแอดเดรสสำหรับพอร์ทบน IBM PC.....	27
ตารางที่ 2.1 การใช้งานแอดเดรสสำหรับพอร์ทบน IBM PC (ต่อ).....	28
บทที่ 3 วงจร A/D, D/A	49
ตารางที่ 3.1 แสดงแรงดันอินพุตเข้าที่พุกของวงจร A/D แบบ "เฟรช".....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ความเป็นมาและความสำคัญของโครงการ

ในระบบสื่อสารในปัจจุบัน สิ่งที่เราสนใจอยู่เรื่องหนึ่งคือการแปลงสัญญาณอนาลอกเป็นดิจิทัล ไม่ว่าจะสัญญาณเสียงในระบบโทรศัพท์หรือระบบไมโครเวฟก็จำเป็นต้องเปลี่ยนจากอนาลอกเป็นดิจิทัลก่อนแล้วจึงนำไปมอดูเลตกับคลื่นไมโครเวฟ และด้านรับก็ต้องแปลงจากดิจิทัลมาเป็นอนาลอกอีกครั้ง นอกจากนี้ยังใช้สัญญาณออสซิลเลเตอร์ VCO (Voltage control oscillator) เพื่อผลิตความถี่ที่เราต้องการ , ใช้ควบคุมความสว่างของหลอดไฟโดยใช้งานร่วมกับ Dimmer, ใช้ควบคุมมอเตอร์ขนาดเล็ก และในการควบคุมที่กล่าวมาเป็นการควบคุมโดยการป้อนสัญญาณดิจิทัลเข้าทางอินพุตโดยผู้ใช้โดยตรง

ปัญหาใหญ่ของวงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกคือ ในการควบคุมด้วยการป้อนดิจิทัลอินพุตโดยตรงไม่มีความแน่นอน การควบคุมด้วยสัญญาณอนาลอกเมื่อต้องการเปลี่ยนแรงดันเอาต์พุตอาจจะต้องเปลี่ยน Hardware ด้วย และไม่สามารถควบคุมได้หลายขั้นแนลพร้อม ๆ กันนอกจากนี้แล้วการควบคุมด้วยอนาลอกไม่สามารถบันทึกค่าเอาต์พุตได้

วัตถุประสงค์ของโครงการ

เพื่อสร้างวงจร D TO A หลายขั้นแนลควบคุมด้วยคอมพิวเตอร์แทนการป้อนสัญญาณดิจิทัลอินพุตในแต่ละขั้นแนล สามารถบันทึกค่าเอาต์พุตไว้ในแผ่นดิสก์เก็ตได้ สามารถป้อนค่าอินพุตได้จากภายนอก และยังสามารถกำหนดค่า Voltage ของแต่ละขั้นแนลจากจอคอมพิวเตอร์ได้

ขอบเขตของโครงการ

1. วงจร D TO A สามารถจ่ายกระแสได้ประมาณ 25 มิลลิแอมป์
2. สามารถป้อนอินพุตจากภายนอกได้ที่แต่ละขั้นแนลเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ระดับของแรงดันแต่ละแขนจะแสดงบนมอโนเตอร์ของคอมพิวเตอร์
4. โปรแกรมจะติดต่อกับผู้ใช้ด้วย mouse และ keyboard เท่านั้น
5. ค่าผิดพลาดของ out put ประมาณ 5%

ประโยชน์

1. สามารถนำไปควบคุมหลอดไฟโดยต่อร่วมกับ DIMMER
 2. สามารถนำไปใช้เป็นอุปกรณ์ทดลองในห้องปฏิบัติการของภาคได้
 3. สามารถนำไปควบคุม VCO (VOLTAGE CONTROL OSCILLATOR)
 4. เป็นแนวทางในการออกแบบสร้างวงจรที่ควบคุมด้วย คอมพิวเตอร์
 5. สามารถเปลี่ยน OUT PUT VOLTAGE โดยปรับที่จอยคอมพิวเตอร์ได้
- โดยสะดวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

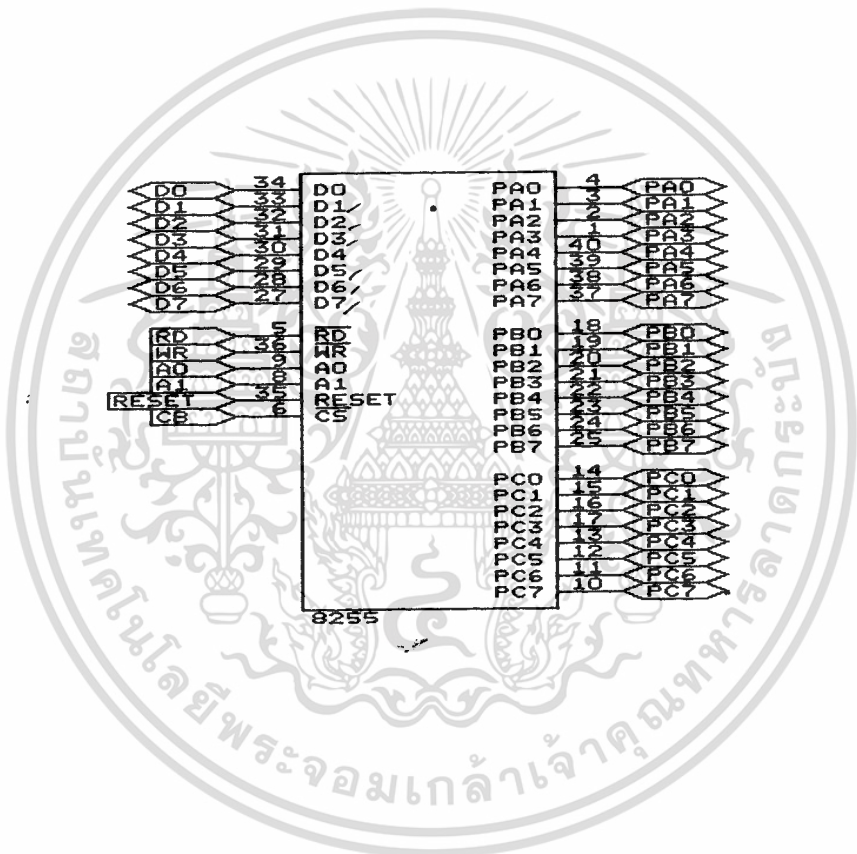
ทฤษฎีและการใช้งาน 8255 เบื้องต้น

ไมโครโปรเซสเซอร์นั้นนอกจากติดต่อกับหน่วยความจำโดยการนำข้อมูลไปเก็บไว้หรืออ่านข้อมูลใด ๆ ออกจากหน่วยความจำแล้วตัว CPU เองอาจจะต้องติดต่อกับส่วนประกอบภายนอกอื่น ๆ อีกด้วยเช่น การรับคีย์บอร์ด การแสดงผลหรือแม้แต่การนำ CPU ไปควบคุมอุปกรณ์ต่าง ๆ นั้น CPU ต้องติดต่อ (รับหรือส่งข้อมูล) โดยผ่านทางอินพุทหรือเอาต์พุทพอร์ท ซึ่งอาจสามารถใช้ไอซี TTL บางเบอร์มาใช้เป็นพอร์ทสำหรับ CPU ได้ แต่ทั้งนี้การใช้ไอซี TTL มีข้อจำกัดหลายอย่าง เช่น ในกรณีที่มีความจำเป็นจะต้องใช้พอร์ทหลาย ๆ พอร์ท เพราะต้องติดต่อกับอุปกรณ์ภายนอกหลายจุด จึงต้องใช้ไอซีเหล่านี้จำนวนหลายตัวและอาจทำให้ยากในการออกแบบวงจร อีกทั้งไม่สามารถจะเปลี่ยนแปลงลักษณะการทำงานให้แตกต่างไปจากเดิมที่ได้ออกแบบไว้แล้ว ดังนั้นผู้ผลิต CPU ในตระกูลต่าง ๆ จึงมักจะผลิตไอซีประเภท LSI ที่ทำหน้าที่เป็นพอร์ทมาเพื่อใช้งานร่วมกับ CPU เบอร์นั้น ๆ ได้สะดวกซึ่งจะทำให้การรับส่งข้อมูลมีความเชื่อถือได้สูง และยังสามารถเปลี่ยนแปลงชนิดของพอร์ท (จากอินพุทเป็นเอาต์พุทหรือจากเอาต์พุทเป็นอินพุท) ได้ง่ายโดยการควบคุมของ CPU เอง ในบทนี้จะกล่าวถึงไอซีที่ทำหน้าที่เป็นอินพุทและเอาต์พุทพอร์ท ซึ่งเป็นที่นิยมในการนำไปใช้งานมากที่สุดอีกทั้งยังมีราคาถูกหาซื้อได้ง่ายคือไปซีเบอร์ 8255 ของบริษัทอินเทลที่จริงแล้วไอซีเบอร์ 8255 นี้ได้ถูกออกแบบและผลิตขึ้นมาเพื่อใช้งานร่วมกับ CPU เบอร์ 8080 แต่ก็สามารถนำมาใช้กับ Z-80 หรือ CPU เบอร์อื่น ๆ ได้ โดยในบทนี้จะกล่าวถึงคุณสมบัติการทำงานรวมทั้งการนำ 8255 ไปใช้งานร่วมกับ Z-80 ในเบื้องต้นด้วยลักษณะเบื้องต้น

8255 นั้นเป็นไอซี LSI ขนาด 40 ขา จากรูปที่ 1a แสดงตำแหน่งของขาต่าง ๆ ทั้ง 40 ขา ส่วนรูป 1b แสดงแผนผังภายในของ 8255 ซึ่ง 8255 นี้มีพอร์ทสำหรับรับส่งข้อมูลอยู่ด้วยกัน 3 พอร์ทมีชื่อดังนี้คือ พอร์ท A, B และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

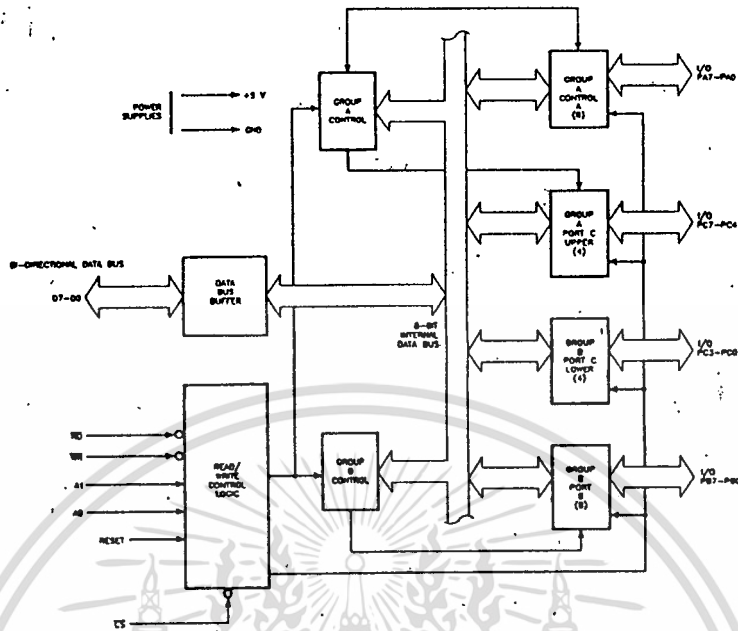
C โดยพอร์ท C นี้จะแบ่งออกเป็น 2 ส่วนคือ พอร์ท C ล่าง (CLO) กับพอร์ท C บน (CHI) นอกจากนี้แล้วยังมีพอร์ทอีกพอร์ทหนึ่ง ซึ่งทำหน้าที่ควบคุมการทำงานของพอร์ท A B และ C โดยการรับคำสั่งมาจาก CPU พอร์ทนี้เรียกว่า พอร์ทควบคุม (Control port) พอร์ทนี้จะใช้งานก็ต่อเมื่อ CPU ต้องการกำหนดลักษณะการทำงานของพอร์ท A, B และ C หรือต้องการเปลี่ยนแปลงพลังจากที่กำหนดไว้เดิม CPU จะส่งรหัสควบคุมมาทางดาต้าบัส (Data Bus) ให้แก่พอร์ทควบคุมนี้



รูปที่ 1.1 ตำแหน่งขาต่างๆของ 8255

การกำหนดรหัสที่ใช้ในการควบคุมพอร์ทต่าง ๆ นี้จะกล่าวในตอนต่อไปในทางปฏิบัติผู้ออกแบบระบบต้องนำรหัสควบคุมที่ได้มาตามข้อกำหนดของ 8255 นี้ไปใส่ในโปรแกรมเพื่อให้ CPU ทำการส่งรหัสควบคุมนี้มายังพอร์ทควบคุมเมื่อระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 แผนผังภายในของ 8255

นั้นเริ่มต้นทำงานหน้าที่ของขาต่างๆ ก่อนที่จะกล่าวถึงการนำ 8255 ไปใช้งาน ควรทราบถึงหน้าที่ของขาต่าง ๆ ของ 8255 ทั้ง 40 ขาเสียก่อน จะทำให้เข้าใจถึงวิธีการใช้งานได้ดียิ่งขึ้นขาต่าง ๆ ของ 8255 สามารถแบ่งออกได้ดังนี้

CS (Chip Select) ขานี้ใช้สำหรับรับสัญญาณจากภายนอกเพื่อใช้ในการเลือกว่าจะให้ 8255 ตัวนี้ทำงานหรือไม่โดยถ้าขานี้ได้รับลอจิก "0" จะทำให้ 8255 เชื่อมต่อเข้ากับระบบบัสต่าง ๆ ของ CPU (โดยการเป็น Hi-Z)

RD (Read Enable) เป็นขาอินพุตที่จะรับสัญญาณจาก CPU ถ้าขานี้ได้รับลอจิก "0" และขณะนั้นขา CS ต้องเป็นลอจิก "0" ด้วย 8255 จะทำการส่งข้อมูลจากพอร์ตที่ CPU ต้องการติดต่อด้วยนั้นให้แก่ CPU ทางดาต้าบัส

WR (Write Enable) มีหน้าที่การทำงานตรงข้ามกับขา RD คือ ถ้าขา WR นี้ได้รับลอจิก "0" (CS ต้องเป็น "0" ด้วยเช่นกัน) 8255 จะรับข้อมูลจากดาต้าบัสของ CPU ส่งออกไปยังพอร์ตที่ CPU กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\overline{RESET} คือ ขาที่ทำหน้าที่ Reset 8255 เมื่อใดที่ 8255 ได้รับสัญญาณ Reset มันจะกลับเข้าสู่โหมดอินพุตคือทุก ๆ พอร์ตจะเป็นอินพุตพอร์ต ขา \overline{RESET} นี้ใช้เมื่อต้องการเคลียร์สถานะต่าง ๆ ของ 8255

D_0-D_7 คือขาข้อมูลที่ใช้ในการติดต่อรับส่งข้อมูลกับ CPU โดยขา D_0-D_7 นี้จะต่อเข้ากับดาต้าบัสของ CPU เพื่อให้ CPU ส่งข้อมูลออกไปยังพอร์ตหรือรับข้อมูลจากพอร์ตส่งให้แก่ CPU ผ่านทาง D_0-D_7 นี้

A_0-A_1 คือขาแอดเดรสที่ใช้ในการเลือกพอร์ตที่ CPU ต้องการจะติดต่อด้วยซึ่งมีความเป็นไปได้ทั้งหมด 4 ค่า ดังนี้คือ

00	=	พอร์ต A
01	=	พอร์ต B
10	=	พอร์ต C
11	=	พอร์ตควบคุม

PA_0-PA_7 เป็นขาสัญญาณของพอร์ต A ใน 8255 ซึ่งจะถูเลือกโดยค่าของ A_0-A_1 และเมื่อพอร์ตนี้ถูกเลือกใช้ข้อมูลต่าง ๆ ก็จะถูกส่งผ่าน PA_0-PA_7 นี้ไปยัง D_0-D_7 (กรณีที่ให้พอร์ต A นี้เป็นอินพุตพอร์ต) หรือจาก D_0-D_7 (กรณีที่ เป็นเอาต์พุตพอร์ต)

PB_0-PB_7 เป็นขาสัญญาณของพอร์ต B ซึ่งจะถูเลือกโดยลอจิกที่ A_0-A_1 เช่นกันกับพอร์ต A และพอร์ต B นี้มีข้อจำกัดในการรับส่งข้อมูลที่ต่างจากพอร์ต A ในบางกรณี

PC_0-PC_7 ซึ่งเป็นสาขาสัญญาณของพอร์ต C ซึ่งจะแบ่งออกเป็น 2 กลุ่มคือ PC_0-PC_3 และ PC_4-PC_7 โดยแต่ละกลุ่มสามารถแยกกันทำงานได้โดยอิสระคือ กลุ่มหนึ่งอาจเป็นอินพุตพอร์ตในขณะที่อีกกลุ่มหนึ่งเป็นเอาต์พุตพอร์ตได้ แต่จะทำงานพร้อม ๆ กันโดยการเลือกด้วยลอจิกที่ A_0-A_1 การใช้งาน 8255 8255 นั้น แบ่งลักษณะการทำงานออกเป็น 3 โหมด (Mode) ด้วยกันคือ

-โหมด 0 เป็นโหมดอินพุตหรือเอาต์พุตพอร์ตอย่างใดอย่างหนึ่งซึ่งทั้ง 3 พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ A, B และ C สามารถทำงานในโหมดนี้ได้

- โหมด 1 เป็นโหมดอินพุทหรือเอาต์พุทพอร์ตอย่างใดอย่างหนึ่งเช่นกันแต่จะมีลักษณะการทำงานเป็นลักษณะของ Handshaking ซึ่งจะกล่าวรายละเอียดในภายหลังในโหมดนี้ทำงานได้เฉพาะพอร์ต A และ B

- โหมด 2 เป็นโหมด Bidirectional คือเป็นได้ทั้งอินพุทและเอาต์พุทพอร์ตในเวลาเดียวกันและทำงานแบบ handshaking เช่นเดียวกับโหมด 1 ในโหมดนี้ใช้ได้เฉพาะพอร์ต A เท่านั้น

การกำหนดโหมดการทำงานของ 8255 นั้นทำได้โดย CPU ทำการส่งรหัสควบคุมผ่านทางดาต้าบัสมายังพอร์ตควบคุม (Control port) ของ 8255 รหัสควบคุมนี้จะมีขนาด 1 ไบต์ เรียกว่า Control Byte และในแต่ละบิตของ Control Byte (1 Byte = 8 bit) นั้นจะมีความหมายเฉพาะของตัวเองดังแสดงในรูปที่ 2 ซึ่งจะอธิบายได้ดังนี้

บิต D_7 เป็นบิตที่แสดงว่า Byte นี้เป็นรหัสควบคุม (Control Byte) ที่จะมีผลต่อการกำหนดโหมดการทำงานของ 8255

บิต D_6 และ D_5 มีความหมายในการเลือกโหมดของพอร์ต A ซึ่งสามารถทำงานได้ทั้ง 3 โหมด โดยลจิกที่ D_6 และ D_5 จะมีความหมายดังนี้

00	=	โหมด 0
01	=	โหมด 1
10	=	โหมด 2
11	=	โหมด 2

บิต D_4 ถ้าเป็นลจิก "0" หมายถึงสั่งให้พอร์ต A ทำหน้าที่เป็นเอาต์พุทพอร์ต แต่ถ้าเป็นลจิก "1" พอร์ต A จะเป็นอินพุทพอร์ตบิตนี้จะมีความหมายเมื่อเราให้ 8255 ทำงานในโหมด 0 หรือโหมด 1 เท่านั้นเพราะในโหมดที่ 2 พอร์ต A จะเป็นทั้งอินพุทและเอาต์พุทพอร์ตในเวลาเดียวกันบิต D_3 เป็นบิตที่กำหนดการทำงานของพอร์ต C บน (PC_4-PC_7) ถ้าบิตนี้เป็นลจิก "0" พอร์ต C บนนี้จะ เป็นเอาต์พุทพอร์ต ถ้าเป็น "1" จะเป็นอินพุทพอร์ต

บิต D_2 เป็นบิตที่ใช้สำหรับกำหนดโหมดการทำงานของพอร์ท B ถ้าเป็นลอจิก "0" หมายถึงให้พอร์ท B ทำงานในโหมด 0 ถ้าเป็นลอจิก "1" จะทำงานในโหมด 1 บิต D_1 เป็นการกำหนดให้พอร์ท B เป็นอินพุทหรือเอาต์พุทพอร์ท ถ้า D_1 เป็นลอจิก "0" จะเป็นเอาต์พุทพอร์ท แต่ถ้าเป็นลอจิก "1" จะเป็นอินพุทพอร์ท

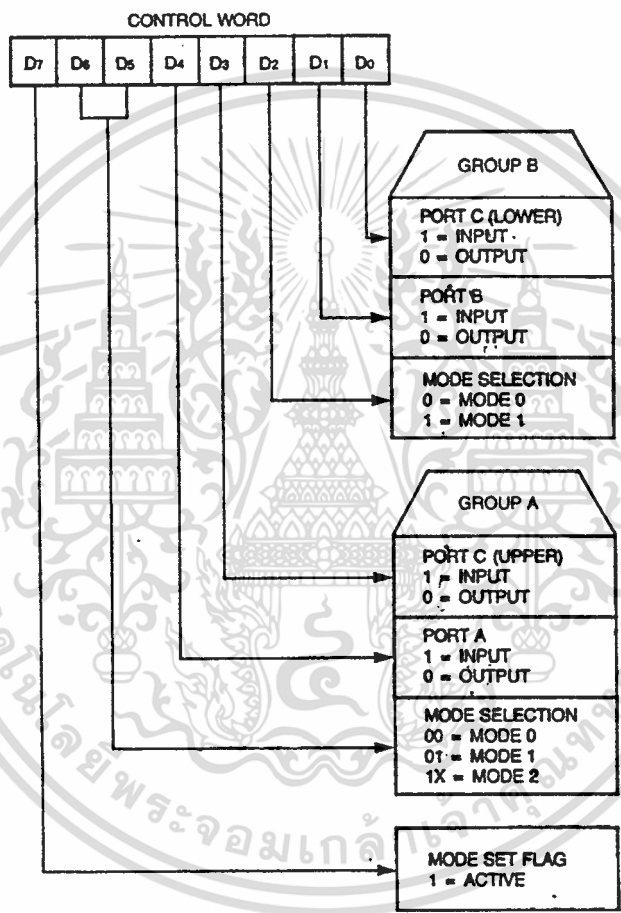
บิต D_0 เป็นบิตที่ใช้กำหนดการเป็นอินพุทหรือเอาต์พุทพอร์ทของพอร์ท C ล่าง (PC_0-PC_9) ถ้าบิตนี้เป็น "0" จะเป็นเอาต์พุทถ้าเป็น "1" จะเป็นอินพุท ตัวอย่างการกำหนดรหัสควบคุมที่จะส่งให้แก่พอร์ทควบคุมของ 8255 หรือที่เรียกกันว่า การโปรแกรม 8255 เช่น ถ้าเราต้องการให้พอร์ท A และพอร์ท C บนเป็นอินพุทพอร์ท ส่วนพอร์ท B กับพอร์ท C ล่างเป็นเอาต์พุทพอร์ท และทุกพอร์ททำงานในโหมด 0 จะสามารถกำหนดรหัสควบคุมได้ดังรูปที่ 5.3 จากรูปเราจะได้รับรหัสควบคุมหรือคอนโทรลไบท์นี้เป็น 10011000 หรือ 98H ในระบบเลขฐานสิบหก และนำมาเขียนเป็นคำสั่งสำหรับ Z-80 ได้ดังนี้

LD A, 98H ; กำหนดรหัสควบคุม
OUT (Control port), A ; ส่งรหัสควบคุมให้แก่พอร์ทควบคุม
การเชื่อมต่อพอร์ท 8255 เข้ากับ Z-80

จากลักษณะการทำงานของขาต่าง ๆ ของ 8255 ที่กล่าวมาแล้วนั้นจะเห็นว่าเราสามารถนำมาใช้กับ CPU เบอร์ Z-80 ได้ไม่ยากซึ่งการต่อ 8255 เข้ากับ Z-80 นั้น ขึ้นอยู่กับผู้ออกแบบระบบว่าต้องการกำหนดให้พอร์ททั้ง 4 พอร์ทของ 8255 เป็นพอร์ทหมายเลขใดของ Z-80 โดยการให้ขาแอดเดรสของ Z-80 เองเป็นตัวเลือกพอร์ทของ 8255 จากรูปที่ 1.4 แสดงตัวอย่างการต่อแอดเดรสของ Z-80 มาทำการเลือกพอร์ทโดยประกอบด้วย Decoder เบอร์ 744138 ซึ่งปกติในระบบไมโครคอมพิวเตอร์ทั่ว ๆ ไปก็มักใช้ Decoder ง่าย ๆ ในการเลือกพอร์ท จากการต่อสายแอดเดรสของ Z-80 เข้ากับ Decoder และ 8255 ดังรูปที่ 3.5 นี้จะทำให้ได้หมายเลขพอร์ทของ 8255 สำหรับให้ CPU ทำการเลือกได้ดังนี้

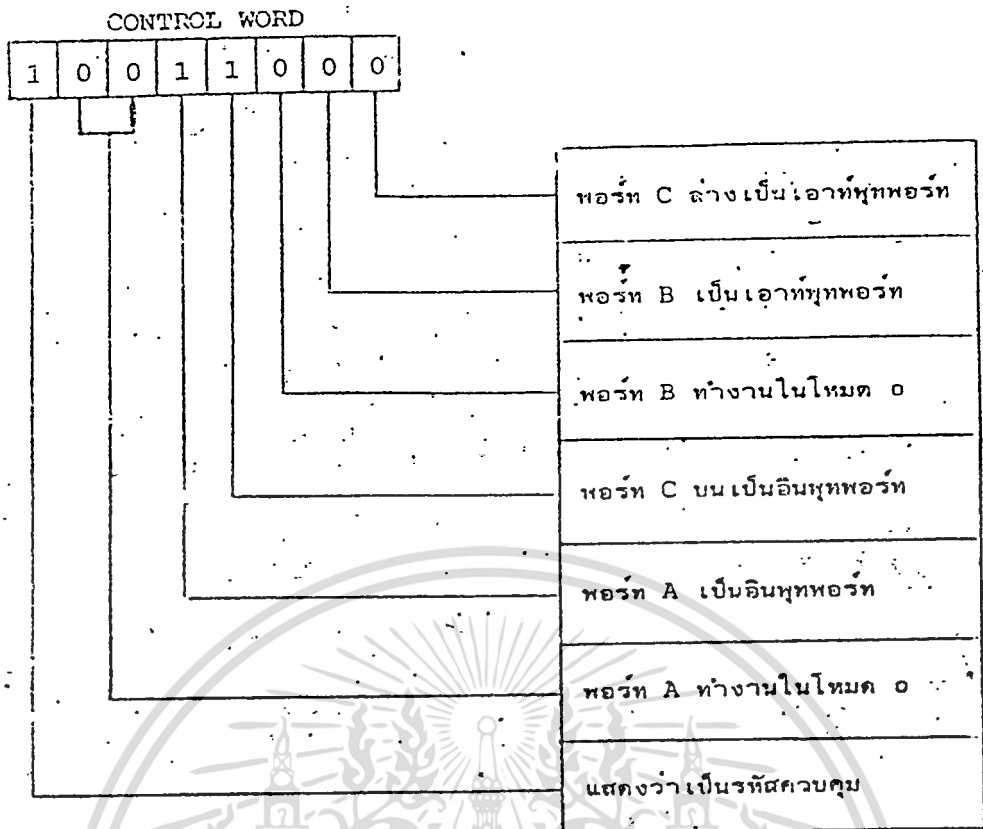
พอร์ทหมายเลข 00H = พอร์ท A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 แสดงความหมายของแต่ละบิตในรหัสควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



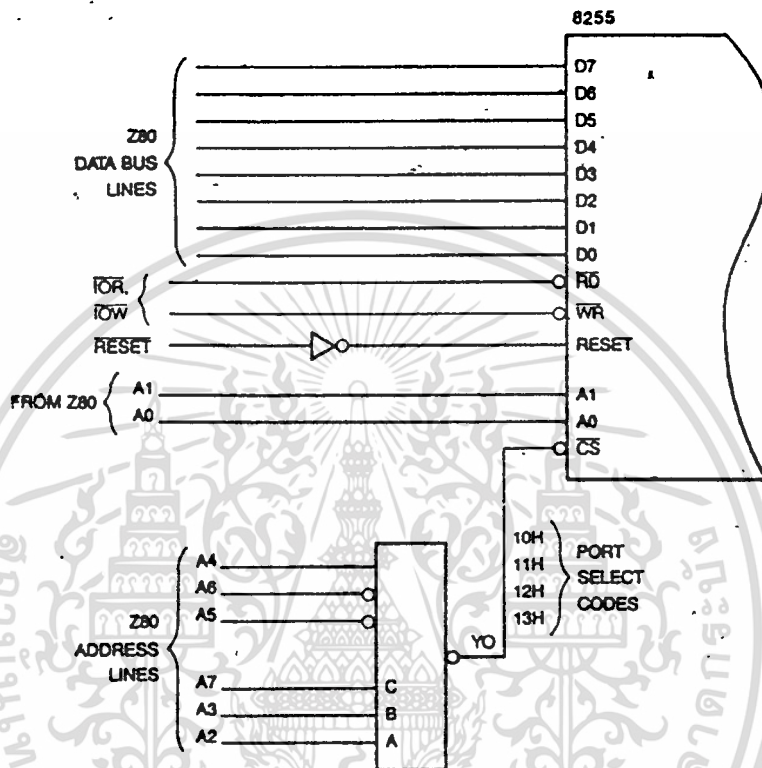
รูปที่ 1.4 ตัวอย่างการกำหนดรหัสควบคุม

พอร์ทหมายเลข 02H = พอร์ท C

พอร์ทหมายเลข 03H = พอร์ทควบคุม

สังเกตว่าที่ขา CS ของ 8255 จะได้รับลอจิก "0" เมื่อขา A_6 , A_5 และ A_7 มีลอจิกเป็น "0" ทั้งหมด (จากคุณสมบัติของ 74138) เป็นการเปิดให้ใช้งาน 8255 ตัวนี้แต่ A_0 และ A_1 ของ Z-80 นั้นต่อเข้าโดยตรงกับขา A_0 - A_1 ของ 8255 ส่วน A_2 , A_3 และ A_4 นั้นไม่มีผลในการกำหนดพอร์ทของ 8255 ตัวนี้ ถ้าเราให้ขาทั้งสามที่เหลือนี้มีลอจิกเป็น "0" ทั้งหมดก็จะได้หมายเลขพอร์ทต่าง ๆ ดังกล่าวตัวอย่างเช่นเมื่อต้องการเลือกพอร์ท B ของ 8255 ตัวนี้ ขา A_1 และ A_0 จะต้องมีลอจิกเป็น "01" ตามลำดับ และขา A_6 , A_5 , A_7 ต้องเป็น "0" ด้วย (เพื่อให้ขา CS เป็น "0") ส่วนขา A_2 , A_3 , A_4 ก็ให้เป็น "0" เมื่อรวมลอจิก A_0 - A_7 จะได้ลอจิกดังนี้คือ 00000001 หรือ 01H ในเลขฐาน 16 ซึ่งเป็นหมายเลขของพอร์ท B ดังที่กล่าวไปแล้ว ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.5 แสดงการต่อ 8255 เข้ากับ Z-80

จริงแล้วการกำหนดหมายเลขพอร์ทในลักษณะที่ไม่ได้ใช้ A_2 , A_3 , A_4 นี้ถ้าให้บิตใดบิตหนึ่งใน 3 บิตนี้มีลอจิกเป็น "1" ก็ยังสามารถเลือกพอร์ทของ 8255 นี้ได้ แต่หมายเลขพอร์ทจะเปลี่ยนไป เช่น ถ้าให้ A_3 เป็น "1" ถ้าต้องการเลือกพอร์ท B จะได้ลอจิก A_0 - A_7 เป็น 00001001 หรือ 09H เป็นชื่อของพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B สรุปได้ว่าในลักษณะการกำหนดหมายเลขพอร์ตโดยลอจิกที่ A_2 , A_0 และ A_4 ไม่มีความสำคัญในการเลือกนั้นจะทำให้พอร์ตแต่ละพอร์ตของ 8255 นั้น เลือกได้หลายหมายเลข เช่นพอร์ต B อาจเลือกได้โดยให้หมายเลข 01H หรือ 09H ก็ได้ แต่ปกติแล้วเราจะนิยมให้ขาแอดเดรสที่ไม่ได้ใช้ (A_2 , A_0 , A_4) มีลอจิกเป็น "0" ในที่นี้เราจึงเรียกชื่อพอร์ต A, B, C และพอร์ตควบคุมด้วยหมายเลขพอร์ต 00H-03H เท่านั้น นอกจากการต่อ A_0 - A_7 เพื่อใช้ในการเรียกหมายเลขพอร์ตแล้ว สัญญาณควบคุมที่จำเป็นสำหรับ 8255 อีกชุดหนึ่งก็คือ สัญญาณควบคุมการเขียนและการอ่าน (WR, RD ซึ่งมีความหมายถึงการส่งหรือรับ) จากรูปที่ 5.4 เช่นกันจะเห็นว่าเราต่อขา WR กับ RD ของ 8255 เข้ากับขา WR และ RD ของ CPU Z-80 โดยตรงได้เนื่องจากเป็นขาแอดคทีฟ "0" เช่นเดียวกัน ส่วนสัญญาณ IORQ ของ Z-80 นั้นเรานำไปควบคุมขาอินพุต (Enable) ของ 74138 คือ ขา G_{2A} และ G_{2B} ซึ่ง 74138 ตัวนี้จะมีชื่อว่าพอร์ตดีโค๊ดเดอร์ท่าหน้าที่เป็นตัวเลือกพอร์ตหลาย ๆ ชุดในกรณีที่มีพอร์ตหลาย ๆ ชุด (ในที่นี้ 8255 1 ตัว คือพอร์ต 1 ชุด) เมื่อใดที่ Z-80 ต้องการติดต่อกับอุปกรณ์ภายนอก (อินพุตหรือเอาต์พุต) มันก็จะส่งสัญญาณ IORQ ออกมาที่ขา G_{2A} และ G_{2B} ทำให้ 74138 ทำการดีโค๊ดตามเงื่อนไขอินพุต (A_2 , A_0 , A_4) แล้วจึงใช้เอาต์พุตของ 74138 นี้ไปควบคุมการแอดคทีฟของพอร์ตแต่ละชุดได้ในตารางที่ 1.1 เป็นตารางแสดงผลของสัญญาณควบคุมต่าง ๆ ที่มีผลต่อ 8255

CS	RD	WR	A1	A0	การทำงาน
1	1	0	0	0	ส่งข้อมูลไปที่พอร์ต A
1	0	1	0	0	รับข้อมูลจากพอร์ต A
1	1	0	0	1	ส่งข้อมูลไปที่พอร์ต B
1	0	1	0	1	รับข้อมูลจากพอร์ต B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CS	RD	WR	A1	A0	การทำงาน
1	1	0	1	0	ส่งข้อมูลไปที่พอร์ท C
1	0	1	1	0	รับข้อมูลจากพอร์ท C
1	1	0	1	1	ส่งรหัสควบคุมไปที่พอร์ทควบคุม
1	0	1	1	1	ไม่มีความหมาย
0	X	X	X	X	8255 ไม่ทำงานปลดสายอินพุตออกจากระบบ

ตารางที่ 1.1 เป็นตารางแสดงผลของสัญญาณควบคุมต่างๆที่มีผลต่อ 8255 (ต่อ)

เมื่อเราได้ทราบถึงการกำหนดหมายเลขพอร์ทต่าง ๆ ของ 8255 แล้ว จากตัวอย่างในรูปที่ 3 เราจึงสามารถเขียนโปรแกรมการควบคุม 8255 ได้ดังนี้

LD A, 98H ; กำหนดรหัสควบคุม

OUT (03H), A ; ส่งรหัสควบคุมไปยังพอร์ทควบคุม

เมื่อพอร์ทควบคุมของ 8255 ได้รับรหัสควบคุมนี้มันก็จะทำให้พอร์ทต่าง ๆ เป็นอินพุตหรือเอาต์พุตพอร์ทตามที่ต้องการ โดยจากตัวอย่างก็จะได้พอร์ท A และพอร์ท C บน เป็นอินพุตพอร์ท ส่วนพอร์ท B กับพอร์ท C ล่างเป็นเอาต์พุตพอร์ทตามต้องการ

รายละเอียดการทำงานในแต่ละโหมดของ 8255

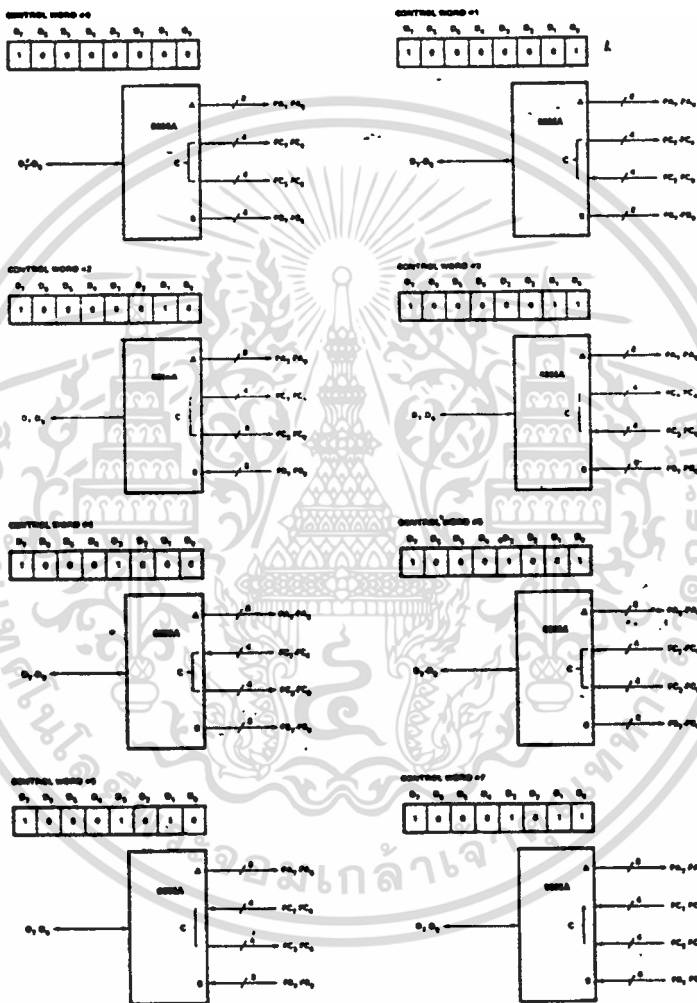
- โหมด 0

ในโหมด 0 นี้ เป็นการกำหนดให้พอร์ททุกพอร์ทของ 8255 นี้ เป็นอินพุตเอาต์พุตพอร์ทแบบพื้นฐานหรือที่เรียกว่า Simple I/O Port ซึ่งเป็นที่นิยมใช้กันมาก เมื่อนำมาเป็นพอร์ทของ Z-80 มีรูปแบบความเป็นไปได้ในการโปรแกรมให้พอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

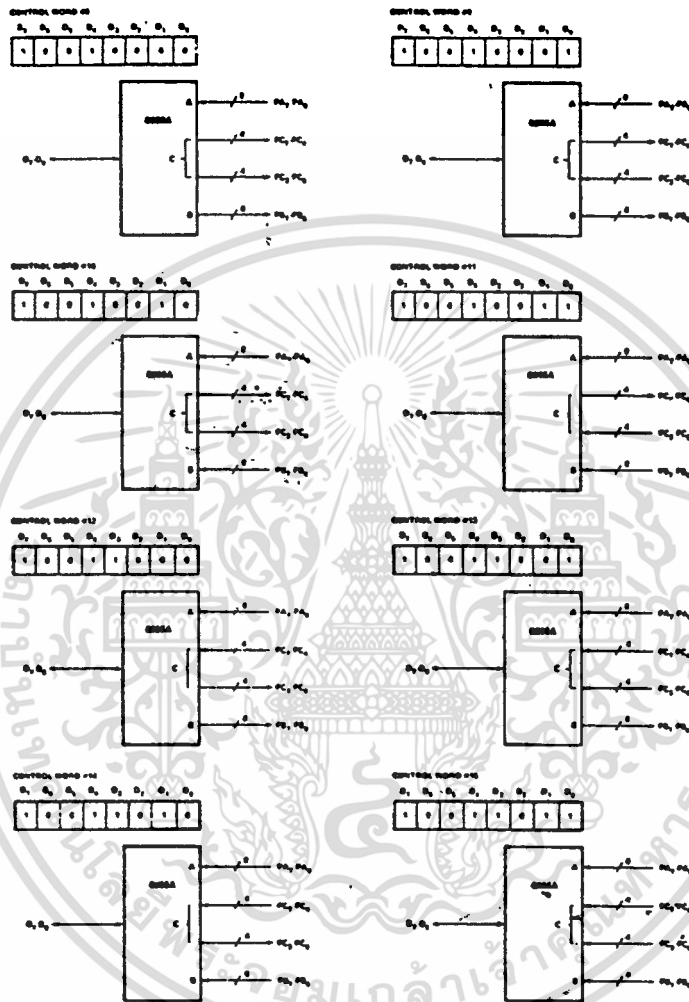
A, B และ C เป็นอินพุทหรือเอาต์พุทพอร์ตได้ทั้งหมด 16 รูปแบบ ดังรูปที่ 1.6

MODE 9 Configurations



รูปที่ 1.6 รหัสควบคุมของการทำงานในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.6 รหัสการทำงานในโหมด 0 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปในแต่ละช่องจะแสดงรูปแบบการทำงานของในโหมด 0 ของ 8255 หนึ่งรูปแบบ โดยจะแสดงถึงสถานะการทำงานของพอร์ท A, B และ C ด้วยลูกศรถ้าลูกศรที่ออกจากตัว 8255 หมายถึง พอร์ทนั้นเป็นเอาต์พุตพอร์ท แต่ถ้าลูกศรชี้เข้าหมายถึง เป็นอินพุตพอร์ท ส่วนพอร์ท C นั้นจะแบ่งเป็น 2 ส่วน คือ พอร์ท C บน และพอร์ท C ล่าง พอร์ทละ 4 เส้นส่วนด้านบนของแต่ละช่องแสดงค่าของรหัสควบคุม (Control Word) ที่จะต้องส่งให้แก่พอร์ทควบคุมในรูปแบบเลขฐาน 16 เพื่อโปรแกรมให้พอร์ทต่างๆ ทำงานเป็นอินพุต เอาต์พุตตามที่แสดงไว้ในช่องนั้น เช่นถ้าเราต้องการให้พอร์ท A และพอร์ท C บน เป็นอินพุตพอร์ท ส่วนพอร์ท B และพอร์ท C ล่างเป็นเอาต์พุตพอร์ทดังตัวอย่างที่เคยกล่าวมาแล้วถ้าดูจากรูปที่ 5 ก็จะตรงกับช่องของ Control Word # 14 มีลอจิกที่ D_7-D_0 ดังนี้คือ 10011010 หรือ 98H ในฐาน 16 เช่นเดียวกับที่เคยกล่าวมาแล้วนั่นเอง

จงเขียนโปรแกรมควบคุม 8255 ซึ่งมีหมายเลขพอร์ทต่าง ๆ ดังกล่าวข้างต้นให้ทำงานในโหมด 0 โดยให้พอร์ท A, B และพอร์ท C ล่างเป็นเอาต์พุตพอร์ท ส่วนพอร์ท C บน เป็นอินพุตพอร์ทหลังจากนั้นให้ส่งข้อมูลค่า A7H ออกไปยังพอร์ท A, B และ C ตามลำดับ

วิธีทำ

จากโจทย์ดังกล่าวเราสามารถหาค่าของ Control word ได้ จากรูปที่ 5 ซึ่งจะตรงกับ Control Word # 4 มีลอจิก D_7-D_0 เป็น 10001000 เท่ากับ 88H ในเลขฐาน 16 สามารถนำไปเขียนเป็นโปรแกรมได้ดังนี้

```
LD A, 88H      ; กำหนดรหัสควบคุม
OUT (03H), A   ; ส่งไปยังพอร์ทควบคุม
LD A, A7H     ; เตรียมค่าที่กำหนด (จากโจทย์)
OUT (00H), A  ; ส่งไปยังพอร์ท A
OUT (01H), A  ; ส่งไปยังพอร์ท B
OUT (02H), A  ; ส่งไปยังพอร์ท C
```

หลังจากทำโปรแกรมนี้อัปเดตแล้วที่พอร์ท A และ B จะมีค่าเป็น A7H และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่นี้ไว้ (LATCH) จนกว่า CPU จะส่งค่าใหม่มาแทน ส่วนพอร์ท C นั้นเนื่องจากเป็นเอาต์พุตพอร์ทเฉพาะพอร์ท C ล่าง (4 bit) ดังนั้นค่าที่ส่งมายังพอร์ท C ล่างก็คือค่าของ 4 bit ล่างของดาต้าบัส (D_0-D_7) ได้แก่ D_0-D_3 เท่านั้น ทำให้ได้ค่าจากพอร์ท C เป็น 7H (ถ้าเขียนให้เต็ม 8 bit จะได้เป็น 07H) และในการแบ่งพอร์ท C ออกเป็น 2 ส่วน เช่นนี้สิ่งที่ต้องคำนึงถึงคือ ถ้า CPU ต้องการรับหรืออ่านข้อมูลจากพอร์ท C ส่วนที่เป็นอินพุตพอร์ทเพียง 4 บิต แต่ค่าที่ได้รับเข้ามาได้จะเป็น 8 bit คือรวมค่าของ 4 bit ที่เป็นเอาต์พุตพอร์ทมาด้วย เช่นถ้าเราเขียนโปรแกรมต่อจากโปรแกรมข้างบนด้วยคำสั่งอ่านข้อมูลจากพอร์ท C เข้ามาไว้ที่รีจิสเตอร์ A ดังนี้

IN A, (02H) ; อ่านข้อมูลจากพอร์ท C บนมาไว้ที่ A

สมมติให้ข้อมูลที่พอร์ท C บน เป็น 0110 ข้อมูลที่อ่านได้จากพอร์ท C และนำมาไว้ที่ A จะเป็น 67H ค่า "6" คือ 0110 ซึ่งจะเป็นของบิต 4 ถึงบิต 7 ของพอร์ท C (PC_4-PC_7) ส่วนค่า "7" คือค่าของพอร์ท C ล่าง (PC_0-PC_3) ที่ได้ถูกส่งออกมาก่อนที่จะมีการอ่านค่าจากพอร์ท C บน แต่ถ้าเราต้องการเฉพาะข้อมูลของพอร์ท C บน ที่อ่านมาได้จะต้องนำไปตัดส่วนของ 4 bit ล่าง ออกด้วยคำสั่ง AND หรืออื่น ๆ ดังนี้

IN A, (02H) ; อ่านข้อมูลจากพอร์ท C บน มาไว้ที่ A

AND FOH ; ตัดส่วนของ 4 bit ล่างออก

จะได้ข้อมูลที่ A เป็น 60H คือ ข้อมูลที่อ่านมาได้เพื่อนำไปใช้งานต่อไปตามจุดประสงค์ของผู้เขียนโปรแกรม จากตัวอย่างที่ผ่านมาจะเห็นได้ว่าการใช้งาน 8255 ในโหมด 0 นี้ง่ายและสะดวกมากจึงนิยมใช้กันในไมโครคอมพิวเตอร์แผ่นพิมพ์เดี่ยว (Single Board) ในส่วนของระบบแสดงผล (LED 7-SEGMENT) การรับคีย์บอร์ด , การกำเนิดเสียง ฯลฯ


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมด 1

จากที่กล่าวมาแล้วในตอนต้นว่าสำหรับโหมด 1 นี้ เป็นการรับส่งข้อมูลในแบบ Handshaking ความหมายของ Handshaking ก็คือระหว่าง CPU, พอร์ต และ อุปกรณ์ภายนอกนั้นขณะที่รับส่งข้อมูลกันนั้น นอกจากการรับส่งข้อมูลกันแล้ว ยังต้องมีสัญญาณในการตอบรับในแต่ละครั้งของการรับส่งข้อมูล โดยผู้รับกับผู้ส่งนั้นจะต้องทำงานสัมพันธ์กันตลอดเวลา ซึ่งเป็นประโยชน์ในกรณีที่อุปกรณ์ภายนอกนั้นมีการทำงานที่ช้ากว่า CPU ทำให้ไม่สามารถทำงาน (รับหรือส่ง) ข้อมูลได้ทัน CPU จึงต้องใช้วิธีรับส่งข้อมูลแบบ Handshaking โดยอุปกรณ์ภายนอกจะเป็นตัวกำหนดจังหวะในการรับส่งข้อมูลเอง เช่น การส่งข้อมูลจากคอมพิวเตอร์ไปยัง เครื่องพิมพ์ซึ่งเครื่องพิมพ์นั้นทำงานได้ช้ากว่าคอมพิวเตอร์มากเมื่อคอมพิวเตอร์ส่งข้อมูลตัวอักษรตัวแรกให้แก่เครื่องพิมพ์ทำการพิมพ์ เครื่องพิมพ์ก็จะทำการประมวลผลต่าง ๆ และเมื่อพร้อมที่จะพิมพ์ตัวอักษรตัวนั้นแล้วก็จะส่งสัญญาณบอกคอมพิวเตอร์ให้ทำการส่งตัวอักษรตัวต่อมาได้ การติดต่อระหว่างเครื่องพิมพ์กับคอมพิวเตอร์จึงเป็นไปโดยไม่ผิดพลาด ส่วนสัญญาณที่ใช้ในการควบคุมการรับส่งข้อมูลนี้จะได้มาจากพอร์ต C เพราะในโหมดนี้พอร์ตที่ใช้ในการรับส่งข้อมูลได้คือ พอร์ต A และ B เท่านั้น ส่วนพอร์ต C จะเป็นตัวส่งและรับสัญญาณควบคุมกับอุปกรณ์ภายนอก และสัญญาณควบคุมในแต่ละบิตของพอร์ต C จะเป็นตามตารางที่ 2 ลักษณะของการรับส่งข้อมูลแบบ Handshaking นั้นแสดงในรูปที่ 1.6 ในกรณีเป็นอินพุทพอร์ต (A หรือ B ก็ตาม) เมื่ออุปกรณ์ภายนอกต้องการส่งข้อมูลให้แก่ CPU มันก็จะส่งข้อมูลเข้ามายังพอร์ตพร้อมกับส่งสัญญาณ STB (Strobe) มาบอกแก่ 8255 เมื่อ 8255 ได้รับสัญญาณ STB ก็จะรับข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายในก่อนแล้ว 8255 ก็จะส่งสัญญาณ IBF (Input Buffer Full) เป็นลอจิก "1" ไปบอกแก่อุปกรณ์ภายนอกว่าพอร์ตได้รับข้อมูลมาเก็บไว้ในบัฟเฟอร์ (รีจิสเตอร์) แล้วและขอ่าเพิ่งส่งมาอีกจนกว่า CPU จะรับข้อมูลนั้นไปจากพอร์ตแล้ว (โดยดูจากการที่ CPU ส่งสัญญาณ RD มายัง 8255) สัญญาณ IBF ก็จะกลับเป็น "0" เพื่อบอกให้อุปกรณ์ภายนอกส่งข้อมูลต่อไปได้ส่วนในกรณีที่เป็นการเอาต์พุทพอร์ตนั้น เมื่อ CPU ส่งข้อมูลออกมายังพอร์ตของ 8255 ตัว 8255 ก็จะรับข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแล้วส่งสัญญาณ OBF (Output Buffer Full) บอกไปยังอุปกรณ์ภายนอกว่ามีข้อมูลมารอที่บัฟเฟอร์ (รีจิสเตอร์) ของพอร์ทแล้วให้มารับไปได้ อุปกรณ์ภายนอกก็จะส่งสัญญาณตอบรับ ACK (Acknowledge) มายังพอร์ทเพื่อขอรับข้อมูลไป หลังจากนั้นสัญญาณ OBF ก็จะกลับเป็น "1" เพื่อรอให้ CPU ส่งข้อมูลใหม่ต่อไปในทางปฏิบัติแล้วการที่ CPU จะทราบได้อย่างไรว่าเมื่อไรที่อุปกรณ์ภายนอกได้ส่งข้อมูล



พอร์ท	การณินทร	การณินทร
PC ₀	INTR _B	INTR _B
PC ₁	IBF _B	OBF _B
PC ₂	STB _B	ACK _B
PC ₃	INTR _A	INTR _A
PC ₄	STB _A	I/O
PC ₅	IBF _A	I/O
PC ₆	I/O	ACK _A
PC ₇	I/O	OBF _A

ตารางที่ 1.2

เข้ามารอที่บัฟเฟอร์ของพอร์ทแล้ว ในกรณีที่ป็นอินพุทพอร์ท หรือเมื่อไรที่อุปกรณ์ภายนอกได้รับข้อมูลจากพอร์ทไปแล้วพร้อมที่จะให้ CPU ส่งข้อมูลต่อไปได้ในกรณีที่ป็นอินพุทพอร์ทนั้น มีวิธีที่จะให้ CPU ทราบได้ 2 ลักษณะ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเขียนโปรแกรมให้ CPU ทำการตรวจสอบสัญญาณควบคุมจาก 8255 เช่น ในกรณีเอาท์พุทพอร์ทก็จะให้ CPU คอยตรวจสอบบิตที่ 7 ของพอร์ท C (OBF) หลังจากทีส่งข้อมูลให้แก่พอร์ทไปแล้วถ้าบิต 7 นี้ยังเป็นลอจิก "0" อยู่แสดงว่า อุปกรณ์ภายนอกยังไม่ได้รับข้อมูลไปจากพอร์ทแต่ถ้าบิตที่ 7 เป็น "1" แสดงว่าอุปกรณ์ภายนอกได้รับข้อมูลไปแล้วจึงให้ CPU ทำการส่งข้อมูลชุดใหม่ต่อไปเช่นเดียวกัน ในกรณีของอินพุทพอร์ทก็ให้ CPU ตรวจสอบบิตที่ 1 ของพอร์ท C ได้ในทำนอง เดียวกัน แต่วิธีนี้ CPU จะต้องเสียเวลาตรวจสอบลอจิกของบิตดังกล่าวอยู่ตลอดเวลา ซึ่งจะทำให้เสียเวลาของ CPU ไปโดยเปล่าประโยชน์ จึงควรเลือก ใช้วิธีนี้ต่อเมื่อพิจารณาแล้วว่าการเสียเวลาของ CPU นั้นไม่มีผลเสียต่อระบบอีกวิธีหนึ่งนั้นคือการให้อุปกรณ์ภายนอกนั้นทำการขออินเทอร์พท์จากที่แสดงในตารางที่ 1.2 นั้นจะเห็นว่าที่บิต 0 และบิต 3 ของพอร์ท C นั้นจะเป็นสัญญาณอินเทอร์พท์ (INTR) ของพอร์ท B และ A ตามลำดับซึ่งสามารถใช้สัญญาณอินเทอร์พท์นี้ให้เป็น ประโยชน์ได้ ในการต่อใช้งานสัญญาณ INTR ไปที่ขา INT ของ Z-80 นี้จะ ต้องผ่าน Not gate (Inverter) เสียก่อนเนื่องจากการขออินเทอร์พท์ของ Z-80 นั้นเป็นแอกทีฟ "0" การรับส่งข้อมูลในแบบนี้แสดงในรูปที่ 1.7 และรูปที่ 1.6 ในรูปที่ 1.7 กรณีที่เป็นอินพุทพอร์ท เมื่ออุปกรณ์ภายนอกส่งข้อมูลให้แก่พอร์ท พร้อมทั้งส่งสัญญาณ STB มาให้พอร์ทรับข้อมูลนี้ไปเก็บในรีจิสเตอร์ เมื่อพอร์ทรับ ข้อมูลนี้ไปเก็บแล้วก็ส่งสัญญาณ IBF ตอบรับไปยังอุปกรณ์ภายนอกเพื่อให้หยุด การส่งข้อมูลขึ้น แล้วจึงทำการขออินเทอร์พท์โดยการส่งสัญญาณ INTR ผ่าน Inverterไปที่ขา INT ของ CPU เพื่อขอให้ CPU มารับข้อมูลนี้ไป หลังจาก CPU รับข้อมูลนี้ไปแล้ว (8255 ได้รับสัญญาณ RD พอร์ทก็จะของอินเทอร์พท์ ทำให้สัญญาณ IBF กลับเป็น "0" เป็นการบอกให้อุปกรณ์ภายนอกได้ทราบว่าพร้อม ที่จะรับข้อมูลไปทีต่อไปได้แล้ว ส่วน ในรูปที่ 1.8 เป็นกรณีของเอาท์พุทพอร์ท เมื่อ CPU ต้องการส่งข้อมูลให้อุปกรณ์ภายนอก โดยการส่งสัญญาณ WR แก่ 8255 ให้รับข้อมูลไปไว้ที่รีจิสเตอร์บัฟเฟอร์ โดยพอร์ทจะรับข้อมูลไว้แล้วส่ง สัญญาณ OBF ไปบอกอุปกรณ์ภายนอกให้มารับข้อมูลไป และหลังจากที่อุปกรณ์ภายนอกรับข้อมูลไปแล้วก็จะตอบรับโดยการส่งสัญญาณ ACK แก่ 8255 ทำให้สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณ OBF กลับเป็น "1" พร้อมกับทำให้สัญญาณ INTR เป็น "1" ด้วย จึงจะ
เป็นการขออินเทอร์พท์ให้ CPU ส่งข้อมูลใหม่ต่อไปในกรณีที่ต้องการส่งข้อมูลต่อ
เนื่องตามกรรมวิธีเดิม แต่ถ้า CPU ยังไม่ต้องการส่งข้อมูลต่อหรือยังไม่มีข้อมูลก็
สามารถส่งให้พอร์ทขออินเทอร์พท์ได้ (ทำให้ INTR เป็น "0") ด้วยการ
ส่งแอดเดรสไบท์ไปยังพอร์ทควบคุมเป็นการกำหนดให้บิตใด ๆ ของพอร์ท C เป็น
ลอจิก "0" หรือ "1" ตามต้องการ มีวิธีการกำหนดดังนี้

บิต 7 เป็น "0" เพื่อแสดงว่าเป็นแอดเดรสไบท์

บิต 6 ถึงบิต 4 ไม่ใช้จะให้เป็นลอจิกใดก็ได้ ปกติจะให้ เป็น "0"

บิต 3 ถึงบิต 1 เป็นการบอกตำแหน่งบิตของพอร์ท C ที่ต้องการอ้างอิงดังนี้

000 = PC_0 (บิต 0 ของ Port C)

001 = PC_1

010 = PC_2

011 = PC_3

100 = PC_4

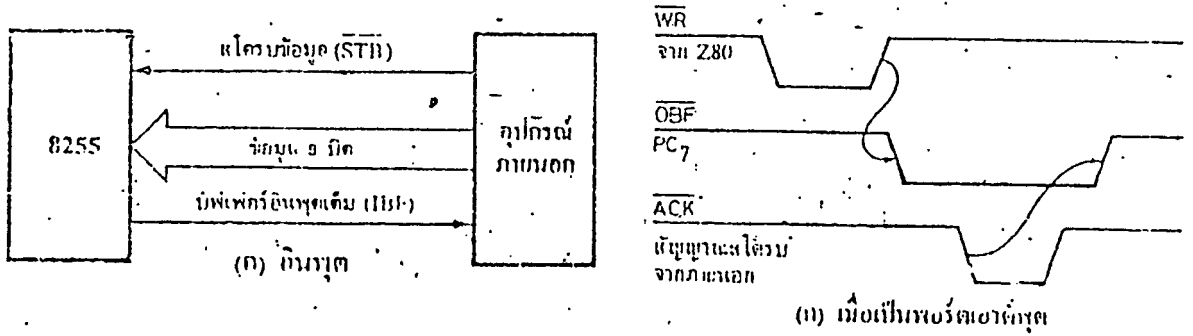
101 = PC_5

110 = PC_6

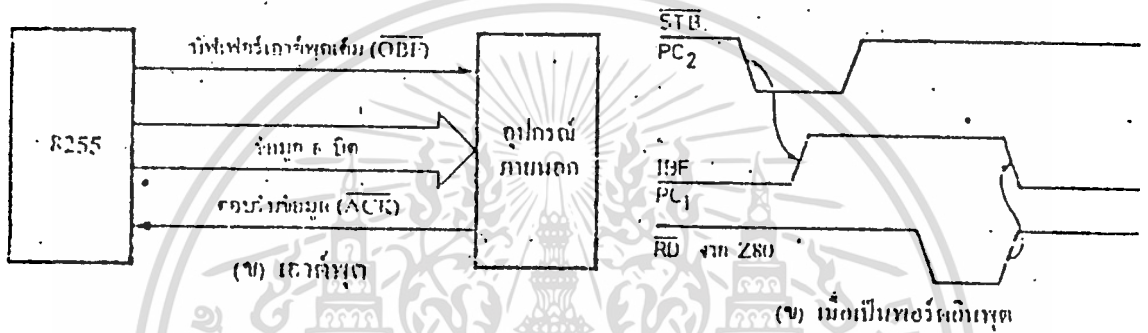
111 = PC_7

บิต 0 ให้บิตใด ๆ ของพอร์ท C ที่กำหนดไว้ (ด้วยบิต 3-บิต 1) นั้นมีลอจิกเป็น
"0" หรือ "1" เช่นถ้าต้องการให้พอร์ท B ขออินเทอร์พท์ ก็คือทำให้ขา
 PC_0 ซึ่งเป็นขา INTR ของพอร์ท B นั้นกลับเป็นลอจิก "0" จะได้แอดเดรสไบท์
เป็น 00000000 หรือ 00H ในฐาน 16 การรับส่งข้อมูลแบบ Handshaking
โดยกรรมวิธีขออินเทอร์พท์นี้ ถ้าในกรณีที่มี 8255 ทำงานร่วมกันหลายตัวจะทำให้
วิธีการขออินเทอร์พท์นี้ยุ่งยากมาก เนื่องจากต้องมีกรรมวิธีในการจัดลำดับความ
สำคัญเมื่อเกิดมีอุปกรณ์ที่ร้องขออินเทอร์พท์พร้อมกันหลาย ๆ ตัว ซึ่งมีเทคนิคหลาย
แบบทั้งนี้ผู้ที่จะใช้ 8255 ทำงานในโหมด 1 นี้ (หรือโหมด 2 ก็ตาม) จะต้องมี

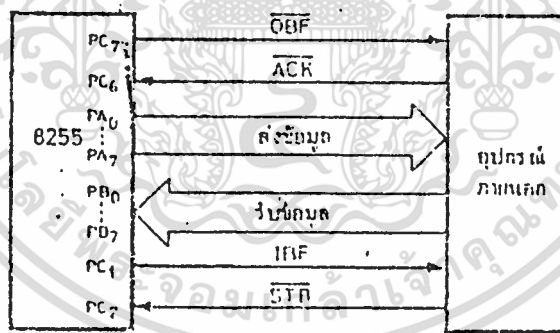
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กรณีอินพุตพอร์ท



กรณีเอาต์พุตพอร์ท



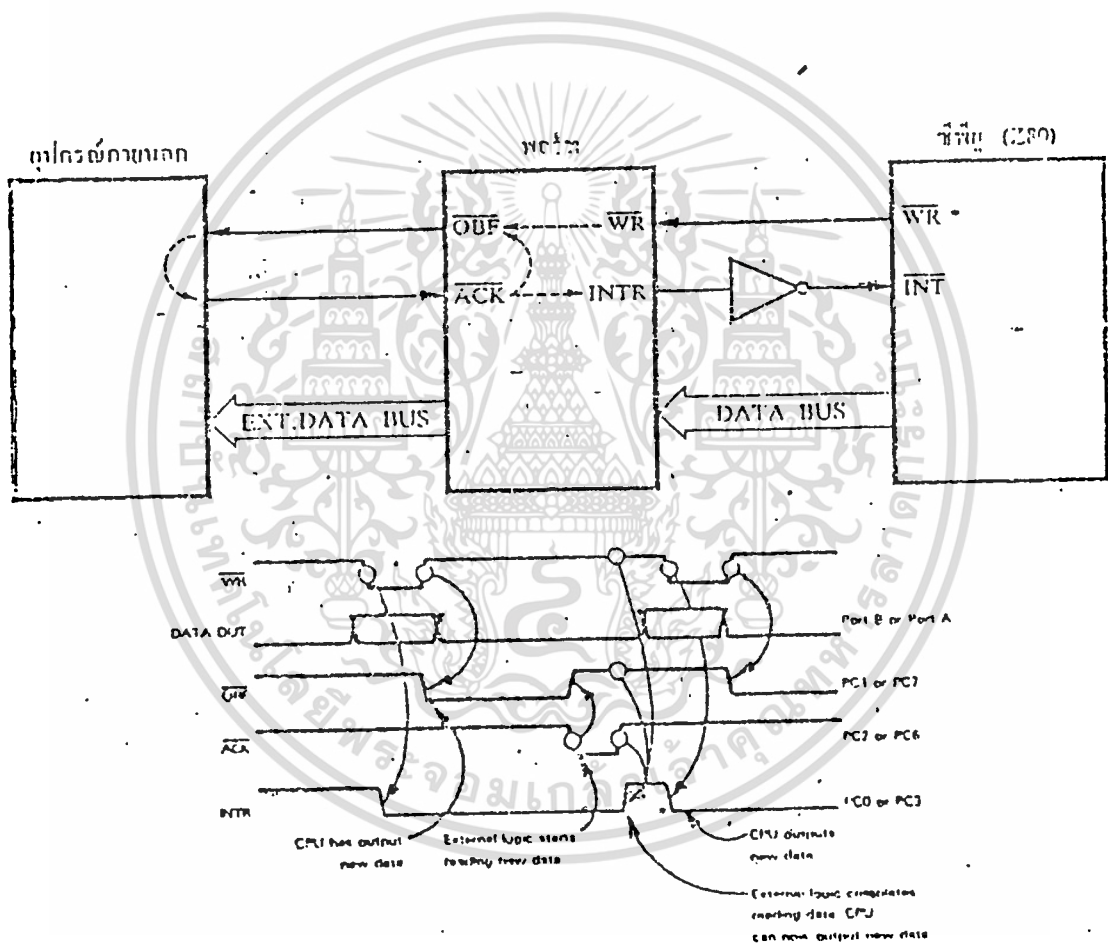
รูปที่ 1.7 การจัดสัญญาณการ HANDSHAKING

ความเข้าใจการทำงานรวมทั้งมีความรู้ในระบบไมโครคอมพิวเตอร์มากพอสมควร - โหมด 2

ในโหมดที่ 2 นี้เป็นการรับส่งข้อมูลในแบบ Handshaking เช่นเดียวกับโหมด 1 แต่พอร์ทที่จะใช้งานในโหมดนี้ได้นั้นมีเพียงเฉพาะพอร์ท A พอร์ทเดียวแต่พอร์ท

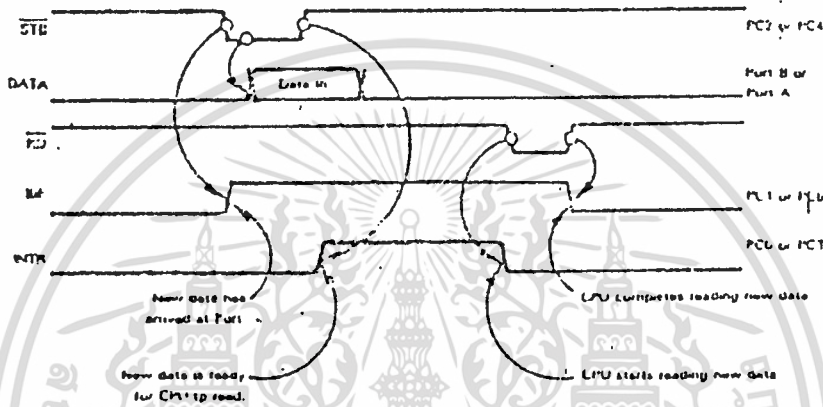
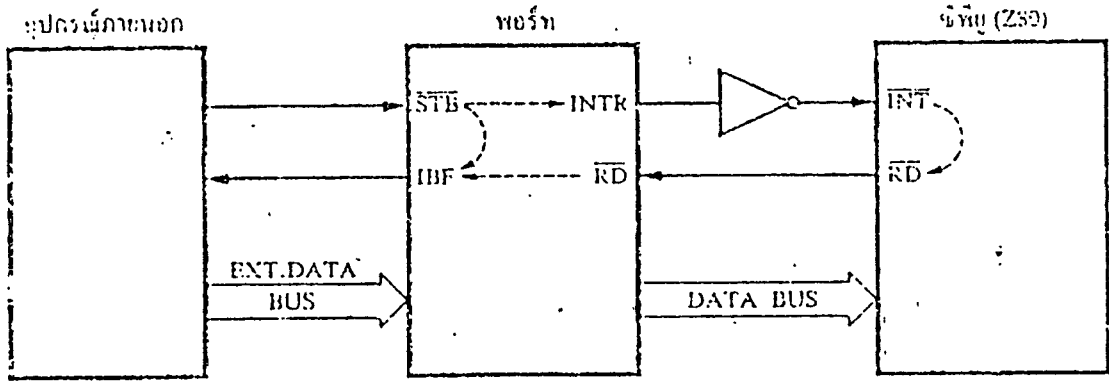
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A ในโหมดนี้สามารถเป็นได้ทั้งอินพุตและเอาต์พุตพอร์ตหรือที่เรียกว่าเป็นพอร์ตแบบสองทิศทาง (Bidirectional) ส่วนสัญญาณที่ใช้ในการทำ Handshaking คือ สัญญาณจากพอร์ต C ซึ่งมีลักษณะตามตารางที่ 3 ขณะที่โปรแกรมให้พอร์ต A ทำงานในโหมด 2 นี้ เรายังสามารถโปรแกรมให้พอร์ต B นั้นทำงานในโหมด 0 หรือโหมด 1 ก็ได้ซึ่งจะทำให้สามารถนำพอร์ต B ไปใช้งานอื่น ๆ ได้อีกโดยที่เป็นอิสระจากกัน



รูปที่ 1.8 การขออินเทอร์รัพท์กรณีเป็นอินพุตพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.9 การขออินเทอร์รัพท์กรณีเป็นเอาต์พุตพอร์ท

พอร์ท C	ความหมาย
PC ₀	I/O
PC ₁	I/O
PC ₂	I/O
PC ₃	INTR _A
PC ₄	STB _A
PC ₅	IBF _A
PC ₆	ACK _A
PC ₇	OBF _A

ตารางที่ 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทท้าย

จากที่กล่าวมาแล้วทั้งหมดนี้ จะเห็นได้ว่า 8255 สามารถใช้งานได้ง่ายและสะดวก เราจึงมักพบเห็น 8255 ในวงจรไมโครโปรเซสเซอร์ทั่ว ๆ ไป ซึ่งการศึกษาการทำงานของ 8255 ให้เข้าใจ จะทำให้สามารถนำไปประยุกต์ใช้ในการอินเตอร์เฟสไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกต่าง ๆ ได้โดยไม่ยากนัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การอินเตอร์เฟส

การอินเตอร์เฟสเป็นสิ่งสำคัญมาก ในการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกเช่นการเชื่อมต่อคอมพิวเตอร์กับ joystick เป็นต้นเนื่องจากการอินเตอร์เฟสใน IBM PC มีแอดเดรสให้เลือกหลายแอดเดรสขึ้นอยู่กับความสะดวกความเหมาะสม และต้องดูด้วยว่าแอดเดรสนั้นถูกใช้งานอยู่หรือไม่การอินเตอร์เฟสในแต่ละชุดแอดเดรสต้องต่างกัน

การจัดแอดเดรสสำหรับอินพุทเอาต์พุตพอร์ต ในไอบีเอ็ม พีซี

ในการควบคุมและตรวจสอบสภาวะการทำงาน รวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิปพอร์ตหรือการ์ดต่างๆ ที่ใช้ในระบบของ ไอบีเอ็ม พีซี นั้นจะกระทำโดยผ่านทางพอร์ตไอโอของระบบ ดังนั้นในการที่จะใช้งานหรือควบคุม การทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ตไอโอต่างของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ตเหล่านี้ต้องกระทำโดยการ อ้างถึงแอดเดรสของพอร์ตไอโอ เหล่านี้โดยตรงเราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสบนไอบีเอ็ม พีซี ด้วยการอ้างแอดเดรสบนไอบีเอ็ม พีซี เอที(ใช้ซีพียู 80286)ถูกออกแบบมาให้ใช้แอดเดรสของไอโอพอร์ตได้ 10 บิตคือ A0-A9 เท่านั้น จึงสามารถที่จะ อ้างแอดเดรสของไอโอพอร์ตสูงสุดเพียง 1024 พอร์ต ใน 1024 พอร์ตนี้ยังแบ่งออกเป็น "0" เราจะอ่านข้อมูลเฉพาะพอร์ตของชิปพอร์ตต่าง ๆ ที่อยู่บนเมนบอร์ดของไอบีเอ็ม พีซีแต่ถ้า A9 เป็น "1" ก็จะอ่านข้อมูลได้เฉพาะจากพอร์ตของการ์ดต่าง ๆ เท่านั้น

ดังนั้นถ้าเราต้องการสร้างแผ่นเชื่อมโยงกับไอบีเอ็ม พีซี จึงต้องเลือกแอดเดรสสำหรับพอร์ตที่อยู่บนการ์ดตามที่แสดงไว้ในตารางที่ 2.1 สำหรับโครงการปริณญาณพนธ์นี้จะเลือกแอดเดรสของพอร์ตช่วง 300-31FH ซึ่งใช้กับโปรโตไทป์การ์ด ทั้งนี้เพื่อความสะดวกของผู้ออกแบบและป้องกันการอ้างพอร์ตซ้ำกับการ์ดที่มีอยู่ แต่ก็สามารถใช้งานในเบอร์พอร์ตได้ถ้าไม่มีการ์ดนั้นติดตั้งอยู่ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การใช้งานแอดเดรสสำหรับพอร์ตบนไอบีเอ็ม พีซี

หมายเลขพอร์ต	ชื่ออุปกรณ์
000H-01FH	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 1,8237A-5
020H-03FH	อินเทอร์รัพต์คอนโทรลเลอร์หมายเลข 1,8259A
040H-05FH	ไทเมอร์ 8254-2
060H-06FH	8042 คีย์บอร์ด
070H-07FH	นาฬิกา และ NMI และทิมอสแรม
080H-09FH	DMA เพจรีจิสเตอร์
0A0H-0BFH	อินเทอร์รัพต์คอนโทรลเลอร์หมายเลข 2,8259A
0C0H-0DFH	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 2,8237A-5
0F0H	เคิลีวีโปรเซสเซอร์คณิตศาสตร์
0F1H	วีซีทีโปรเซสเซอร์คณิตศาสตร์
0F8H-0FFH	โปรเซสเซอร์คณิตศาสตร์
1F0H-1F8H	ฮาร์ดดิสก์
200H-207H	เกมไอโอ
278H-27FH	พอร์ตเครื่องพิมพ์หมายเลข 2
28FH-2FFH	พอร์ตอนุกรมหมายเลข 2
300H-31FH	โปรโตไทป์การ์ด
360H-36FH	สำรอง
378H-37FH	พอร์ตเครื่องพิมพ์หมายเลข 1
380H-38FH	SDLC, ไบพิงค์ 2
3A0H-3ARH	โมโนโครมและเครื่องพิมพ์
3C0H-3CFH	สำรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การใช้แอดเดรสสำหรับพอร์ตบน IBM PC (ต่อ)

หมายเลขพอร์ต	ชื่ออุปกรณ์
3F0H-3F7H	ควบคุมดิสดิสเกตต์
3F8H-3FFH	พอร์ตอนุกรมหมายเลข 1

เทคนิคการดีโค้ดแอดเดรส

วงจรดีโค้ดแอดเดรสถือว่าเป็นหัวใจสำคัญของการอินเตอร์เฟสระหว่าง เมนบอร์ดกับอุปกรณ์ไอโอพอร์ต จะอยู่ร่วมกันเพื่อทำหน้าที่สองอย่าง คือกำหนดให้เป็นพอร์ตอินพุตอ่านข้อมูล หรือให้พอร์ตเอาต์พุตเขียนข้อมูลบนอุปกรณ์เอาต์พุตหรืออินพุต สัญญาณที่จะต้องนำมาใช้เพื่อเป็นตัวเลือกกระบวนการที่เหมาะสมที่ CPU จะติดต่อกับอุปกรณ์ อินพุต/เอาต์พุต หรืออุปกรณ์อินพุต/เอาต์พุตติดต่อกับ CPU คือแอดเดรส ใช้ตั้งแต่ AO-A9 (10 บิต) สำหรับแอดเดรสพอร์ต IOR แอดที่ฟลลจิก "0" เพื่อบอกว่าขณะนี้เป็นการอ่านข้อมูลจากอินพุตพอร์ต IOW แอดที่ฟลลจิก "0" เพื่อบอกว่าขณะนี้เป็นการเขียนข้อมูลไปยังเอาต์พุตพอร์ต AEN แอดที่ฟลลจิก "1" เพื่อดิสเอเบิลวงจรอื่น ๆ ขณะเกิดขบวนการ DMA ในการดีโค้ดจะเป็นลลจิก "0"

รูปที่ 2.1 แสดงถึงพื้นฐานของการดีโค้ดซึ่งแยกออกเป็น 3 ส่วน ส่วนแรกเป็นการนำแอดเดรส AO-A9 มาทำการดีโค้ดสร้างสัญญาณ- ADDRESSD SEL และนำสัญญาณนี้ไปรวมกับ-AENได้สัญญาณ -I/O SELECT เพื่อแยกว่าเป็นอุปกรณ์ไอโอ ถ้าใช้สัญญาณนี้ร่วมกับ-IOR จะได้สัญญาณสำหรับเลือกอุปกรณ์อินพุต -IOW จะเป็นสัญญาณสำหรับเลือกอุปกรณ์อินพุต -IOW จะเป็นสัญญาณสำหรับเลือกอุปกรณ์เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคการดีโด้สามารถแบ่งได้ 2 กลุ่มดังนี้

2.1 เทคนิคในการดีโด้แอดเดรสแบ่งตามการใช้อุปกรณ์

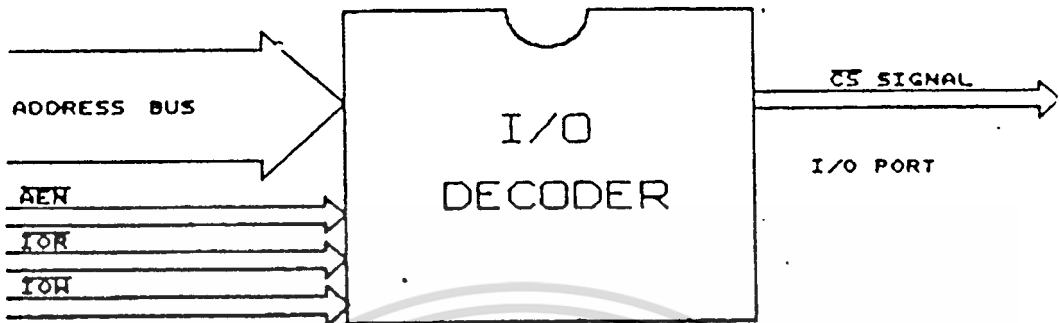
2.1.1 การดีโด้แบบใช้ไอซีลอจิกเกตธรรมดา เช่น AND, OR, NOR, NAND และ NOT GATE ในการดีโด้มักใช้เมื่อต้องการพอร์ตเพียงพอร์ตเดียว

2.1.2 การดีโด้แบบ Fixed ใช้ไอซีดีโด้เบอร์ต่าง ๆ เช่น 74LS138 เป็นต้น วิธีการดีโด้แบบนี้เป็นวิธีที่ง่ายใช้เมื่อต้องการพอร์ตตั้งแต่ 4 พอร์ตขึ้นไป

2.1.3 การดีโด้แบบใช้สวิตช์เลือกโดยใช้ไอซีเปรียบเทียบระหว่าง 2 อินพุต เช่น EXCLUSIVE-OR หรือไอซีเบอร์ 74LS688 (8 BIT COMPARATOR) การดีโด้ทั้งสองแบบแรกที่กำลังกล่าวมามีข้อเสาคือแอดเดรสที่เราเลือกใช้งานนั้นอาจจะซ้ำกับแอดเดรสของการ์ดอื่นที่เรานำมาเพิ่มเข้าไปในระบบภายหลังก็ได้ และอีกข้อหนึ่งคือไม่สามารถเปลี่ยนแปลงแอดเดรสปัญหาเช่นนี้เราสามารถแก้ไขได้โดยใช้วงจร์ดีโด้ที่สามารถเปลี่ยนแปลงค่าแอดเดรสได้โดยเพียงแต่เปลี่ยนตำแหน่งของสวิตช์ (DIP SW.)

2.1.4 การดีโด้โดยให้ PROM (Programmable Read Only Memory) มักจะให้พอร์ตที่เราต้องการมีแอดเดรสแยกกันอย่างอิสระ ต้องมีการโปรแกรมก่อนนำมาใช้งาน หรืออุปกรณ์ที่ทำการดีโด้ไม่มากเกินไปจะนำไปใช้ดีโด้ การทำงานที่อยู่บนการ์ดต่างๆ มารวมไว้บนการ์ดเดียวกัน

2.1.5 การดีโด้โดยการใช้ PAL (PROGRAMMABLE ARRAY LOGIC) และ PLA (PROGRAMMABLE LOGIC ARRAY) เทคนิคการดีโด้แบบนี้ ต้องเขียนอินพุตและเอาท์ให้อยู่ในรูปฟังก์ชันลอจิก และนำไปโปรแกรมลงบน PAL ชนิดของ PAL มีให้เลือกตามจำนวนอินพุตเอาท์พุทที่เหมาะสม วิธีการใช้งานนั้นสามารถศึกษาได้จากคู่มือ วิธีนี้สามารถลดจำนวนอุปกรณ์ในวงจร์ดีโด้ให้เหลือเพียง PAL เพียงอย่างเดียวน



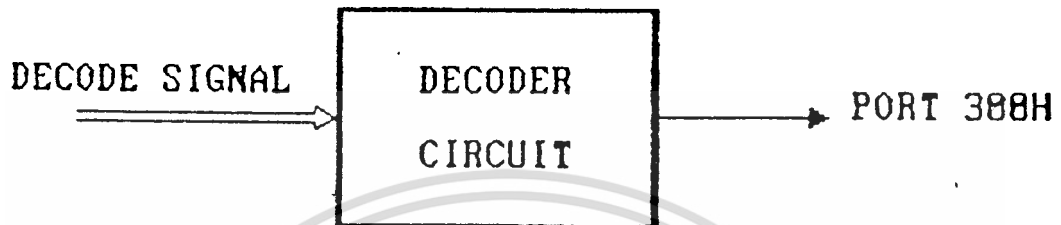
รูปที่ 2.1 แสดงถึงพื้นฐานของการดีโค้ด

เทคนิคการออกแบบวงจรดีโค้ดตามลักษณะพอร์ต

การพิจารณาเลือกเทคนิคใดมาใช้ในการดีโค้ด จึงขึ้นอยู่กับพอร์ตที่เราต้องการมีลักษณะอย่างไร บางครั้งก็จำเป็นต้องนำแต่ละเทคนิคมาใช้ร่วมกัน โครงการนี้สนใจเฉพาะ 3 แบบแรกเท่านั้นเพราะมีการใช้งานทั่วไป ส่วนการใช้ PROM และ PAL นั้นมีความยุ่งยากและใช้ในกรณีพิเศษเท่านั้นดังนั้นจึงจัดการออกแบบวงจรดีโค้ดแอดเดรสแบบต่างๆ เพื่อสร้างพอร์ตที่เราต้องการ ซึ่งพอจะแบ่งลักษณะของพอร์ตได้ 4 ลักษณะดังนี้

1. Single Port Single Address เป็นลักษณะวงจรดีโค้ดที่สร้างพอร์ตขึ้นมาเพียงพอร์ตเดียวและมีแอดเดรสที่จะติดต่อกับพอร์ตนี้เพียงพอร์ตเดียวมีโครงสร้างดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

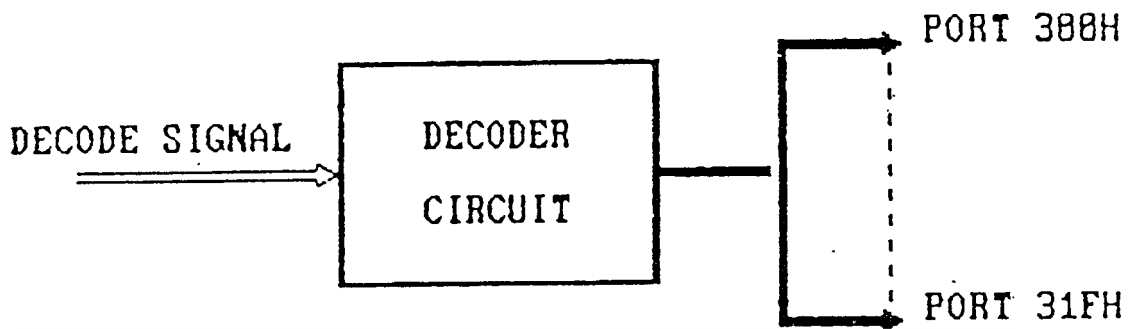


รูปที่ 2.2 โครงสร้างของ Single Port Single Address

2. Single Port Multiple Address เป็นลักษณะวงจรถัดที่สร้างพอร์ตขึ้นมาพอร์ตเดียวเช่นกัน แต่มีแอดเดรสที่จะติดต่อกับพอร์ตนี้ได้หลายเบอร์มีโครงสร้างดังรูป 2.3 วงจรนี้เราไม่สามารถนำสัญญาณแอดเดรสมาตีได้ทั้งหมดโดยจะละช่วงแอดเดรสไปบ้าง ตัวอย่างเช่น ต้องการตีโค้ดพอร์ตในช่วง 300H ถึง 31FH ดังนั้นสัญญาณลอจิกของแอดเดรสมีดังนี้

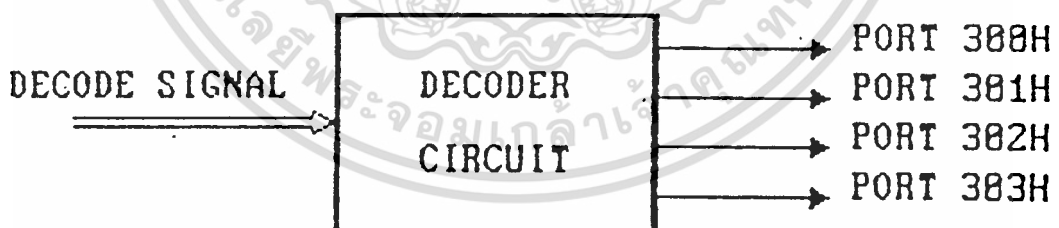
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	X	X	X	X	X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างของ Single Port Multiple Address

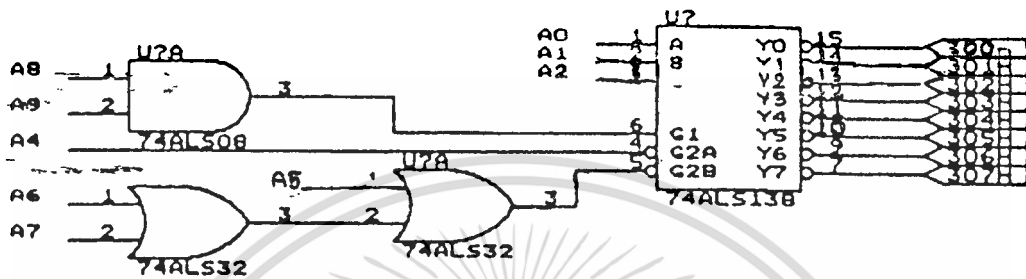
3. Multiple Port Single Address เป็นลักษณะวงจรถัดที่สร้างพอร์ตขึ้นมาหลายพอร์ต และมีแอดเดรสที่จะติดต่อกับพอร์ตนี้เพียงพอร์ตละหนึ่งแอดเดรสเท่านั้นมีโครงสร้างดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างของ Multiple Port Single Address

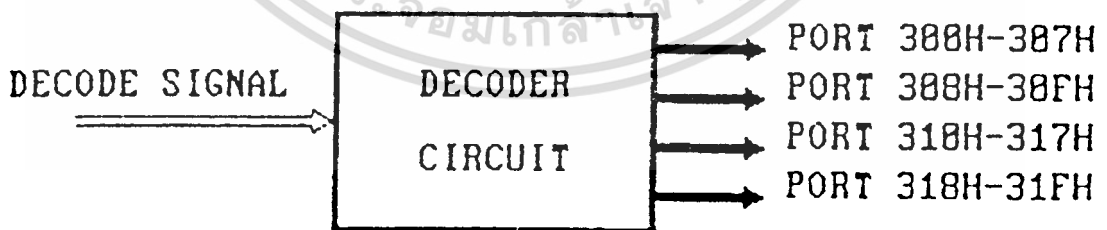
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรทั้ง 2 แบบเป็นการตีโค้ดแบบพอร์ตเดียว เรามักใช้ไอซีลอจิกเกตธรรมดาในการตีโค้ด แต่ถ้าเราต้องการใช้งานมากกว่า 1 พอร์ตเช่น 4 พอร์ตเราจะใช้ไอซีตีโค้ดเบอร์ 74LS138 และ 74LS139 เลือกช่วงในการตีโค้ด



รูปที่ 2.5 วงจร Multiple Port Single Address

4. Multiple Port Multiple Address เป็นลักษณะตีโค้ดที่สร้างพอร์ตขึ้นมาหลายพอร์ต และมีแอดเดรสที่จะติดต่อกับพอร์ตนี้ได้พอร์ตละหลายแอดเดรสเป็นช่วงแอดเดรสมีโครงสร้างดังรูปที่ 2.6

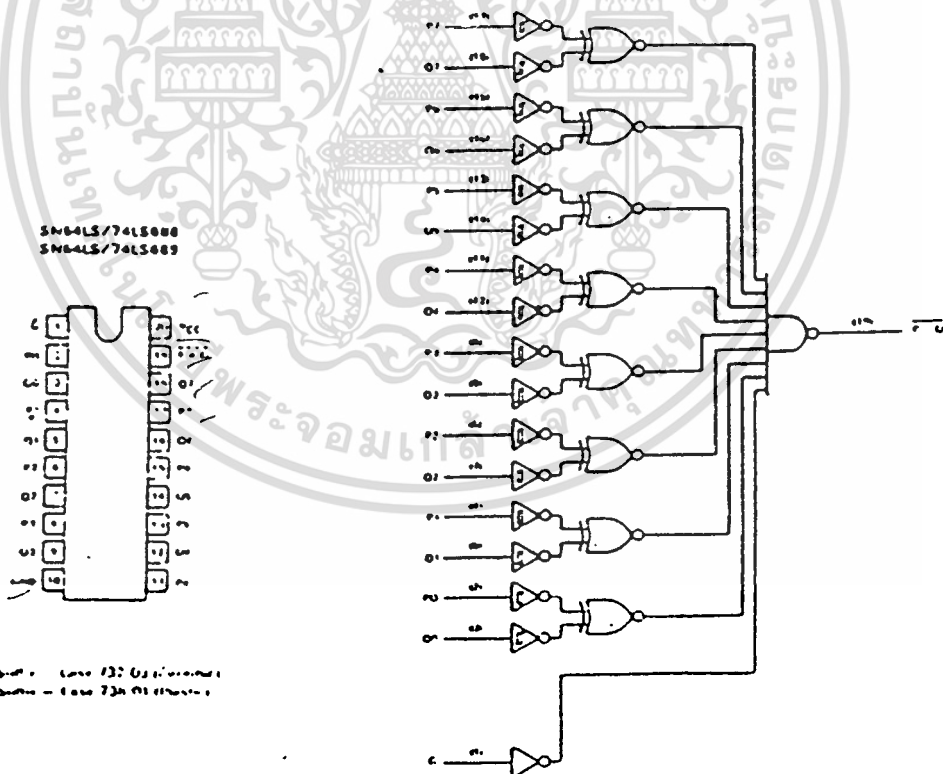


รูปที่ 2.6 โครงสร้างของ Multiple Port Multiple Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERFACE GARD

จากบทที่ 1 ได้กล่าวถึง 8255A และเทคนิคการตีโค้ดแบบต่าง ๆ ที่สามารถนำไปใช้งานได้แล้ว การที่จะต่อ 8255A เข้ากับคอมพิวเตอร์ IBM PC/AT นั้นก็สะดวกยิ่งขึ้นเพียงแต่สร้างวงจรชุดตีโค้ดแอดเดรสให้แก่ 8255A เท่านั้น การที่จะให้ใช้งานได้สะดวกจึงใช้การตีโค้ดแบบเปลี่ยนแอดเดรสได้จึงเลือกใช้อุปกรณ์การตีโค้ด 2 ชนิดเข้าช่วยคือ DIP SWITCH และ IC 74LS68 ไอซีตัวนี้ทำหน้าที่ในการเปรียบเทียบสัญญาณว่าเหมือนกันหรือไม่โดยสามารถเปรียบเทียบได้ 8 คู่ โดยอินพุตแต่ละคู่จะอยู่ที่ขา P และ G เมื่อใดก็ตามที่สัญญาณอินพุตตรงกัน จะให้เอาต์พุตที่ขา 19 เป็น "0" ขาที่ 19 ของ 74LS68 จะถูกต่อเข้ากับ ขา CS ของ 8255A และเมื่อสัญญาณที่อินพุตไม่เท่ากัน 74LS68 จะให้เอาต์พุตออกมาเป็น "1" นอกจากนี้ 74LS68 ยังถูกควบคุมการทำงานโดยคอมพิวเตอร์ โดยจะนำสัญญาณ AEN มาต่อเข้ากับขาที่ 1 ของ 74LS68 จะแอดคที่ฟัลลจิก "0"



รูปที่ 2.7 การจัดขาและตารางเอาต์พุตของ 74LS68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบและวงจรการทำงาน

เราศึกษาโครงสร้างและการทำงานของ 8255A ไปแล้วต่อไปนี้จะกล่าวถึง การนำ 8255A อินเทอร์เฟซกับ IBM/AT ซึ่งเป็นวงจรที่ใช้งานรวมถึงรายละเอียดของอุปกรณ์ที่ใช้และแผ่นลายวงจรในการอธิบายจะอธิบายเป็นส่วนๆดังนี้

1. วงจรดีโค้ด
2. ส่วนของ Ship ชิปพอร์ท

วงจรดีโค้ด

IC ที่เรานำมาดีโค้ดเป็น IC 74LS688 จะทำการดีโค้ดสัญญาณต่างๆจากสล๊อตของ IBM PC มาใช้เพื่อเลือกแอดเดรสให้กับ 8255A สัญญาณที่นำมาใช้ในการดีโค้ดมี A0-A9 แต่ 8255A ต้องการสัญญาณจาก A0 และ A1 ไปควบคุมรีจิสเตอร์ภายในตัวมันดังนั้นสัญญาณ แอดเดรสที่เรานำมาใช้จึงมีแต่ A3-A9 เท่านั้น ส่วน A0-A1 ถูกต่อเข้าไปที่ขาของ 8255A โดยตรง(ขา 9,8)

การดีโค้ดนั้นใน IBM PC นั้นมีด้วยกัน 2 ชนิด

1. การดีโค้ดหน่วยความจำ
2. การดีโค้ดพอร์ท

การดีโค้ดหน่วยความจำไม่ขอกล่าวถึง เพราะปริวิตานพจน์นี้ใช้การอินเทอร์เฟซพอร์ทเท่านั้น การดีโค้ดพอร์ทจะเลือกแอดเดรสที่จะใช้ก่อน การดีโค้ดนี้ใช้แอดเดรส 10 บิต(A0-A9)เข้ามาทำการดีโค้ดส่วนตำแหน่ง A10-A15 นั้นไม่นำมาใช้จากรูปที่ 2.9 เป็นการแสดงวงจรดีโค้ดของอินเทอร์เฟซ 8255A สังเกตได้ว่าเรานำสัญญาณ A0-A9, AEN, IOR, IOW, RESET มาใช้เท่านั้น ส่วนสัญญาณอื่นไม่นำมาใช้เช่น IOCS16, SBHE เพราะใช้ 8255A เพียงตัวเดียวส่งข้อมูล 8 บิต

การทำงานของ 74LS688 นั้นรายละเอียดของขาต่างๆได้อธิบายไว้แล้วในตอนต้น IC ตัวนี้จะทำการเปรียบเทียบสัญญาณอินพุต 2 ส่วนคือ P0-P7 และ Q0-Q7 จะให้เข้าที่พุกออกมาเป็น "0" เมื่อ P=Q สัญญาณ Q จะถูกต่ออยู่กับ Dip switch ในการเลือกแอดเดรสของการ์ดเลือกโดยเช็ทสวิทช์ ตัวอย่างเช่นในปริวิตานพจน์นี้จะใช้แอดเดรสช่วง 300-31FH เมื่อมีแอดเดรสค่า 300H มาเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อินพุทของ 74LS688 (P) ที่ขา 19 ของ 74LS688 ให้เอาท์พุท "0" ทำให้ CS ของ 8255A เป็น "0" 8255A ก็จะต่อคาตาบัสเข้ากับคาตาบัสของคอมพิวเตอร์ สำหรับแอดเดรสที่ใช้ในที่นี้มี 4 แอดเดรสคือ

- 300H ควบคุมการทำงานของพอร์ต A
- 301H ควบคุมการทำงานของพอร์ต B
- 302H ควบคุมการทำงานของพอร์ต C
- 303H ส่ง Control Word ให้กับ 8255A

วงจรการต่อชิปซีพพอร์ต 8255A

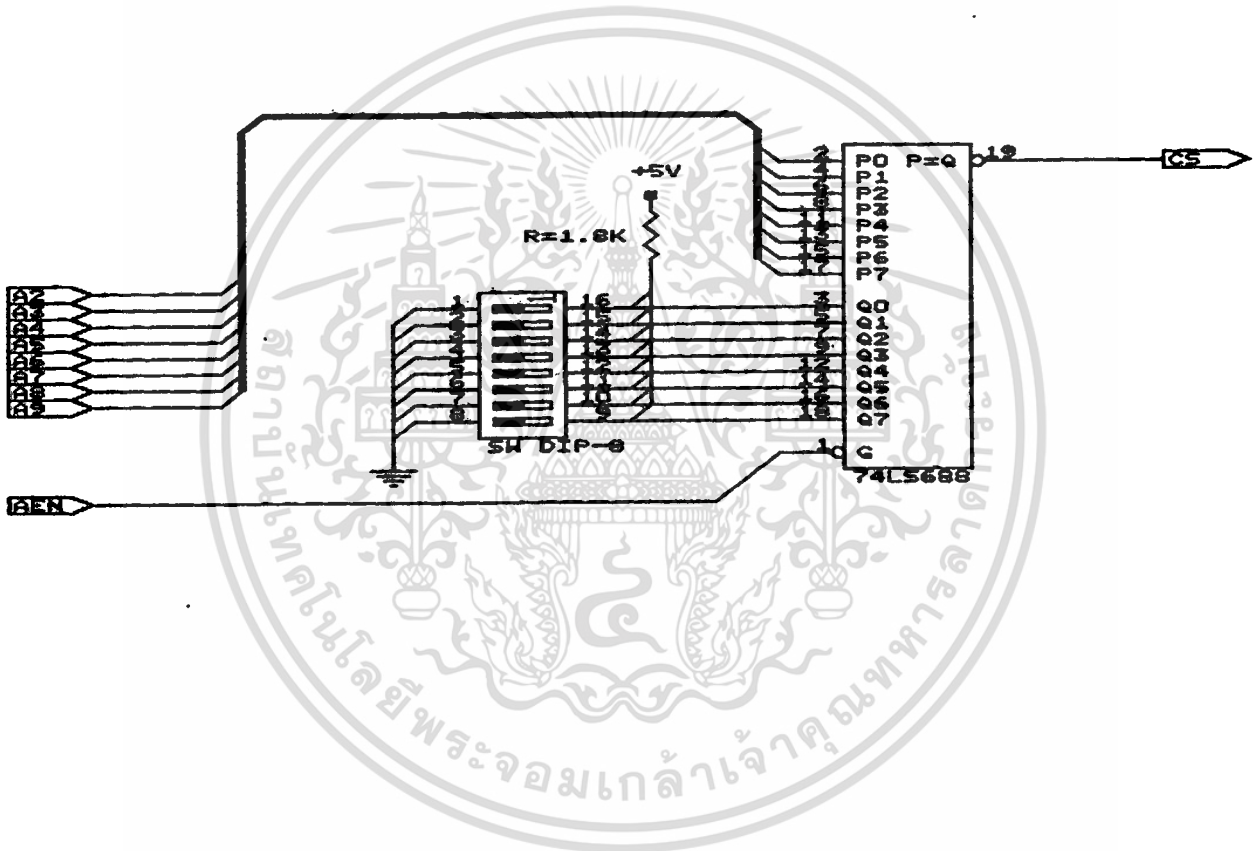
ทฤษฎีการทำงานของ 8255A ได้กล่าวมาแล้วในบทที่ 5 8255A จะต่อเข้ากับชุดดีโอดี เพื่อเลือกให้ 8255A ทำงาน เมื่อแอดเดรสตรงกับที่เซตไว้ 74LS88 จะให้เอาท์พุทเป็น "0" ทำให้ CS เป็น "0" ด้วย คาตาบัสของ IBM PC จะเชื่อมกับคาตาบัสของ 8255A และสามารถอ่านและเขียนข้อมูล บน 8255A ได้

การใช้งาน 8255A INTERFACE GARD

การตั้งแอดเดรสนั้นจะเห็นได้ว่าบิต A8 และ A9 ต้องเป็น "1" และจะนำ A0, A1 8255A นำไปใช้ในการ Set Controlword เช่น port A ทำงานเป็น "00" และ port B เป็น "01" port C เป็น "10" ส่ง Control word เป็น "11"

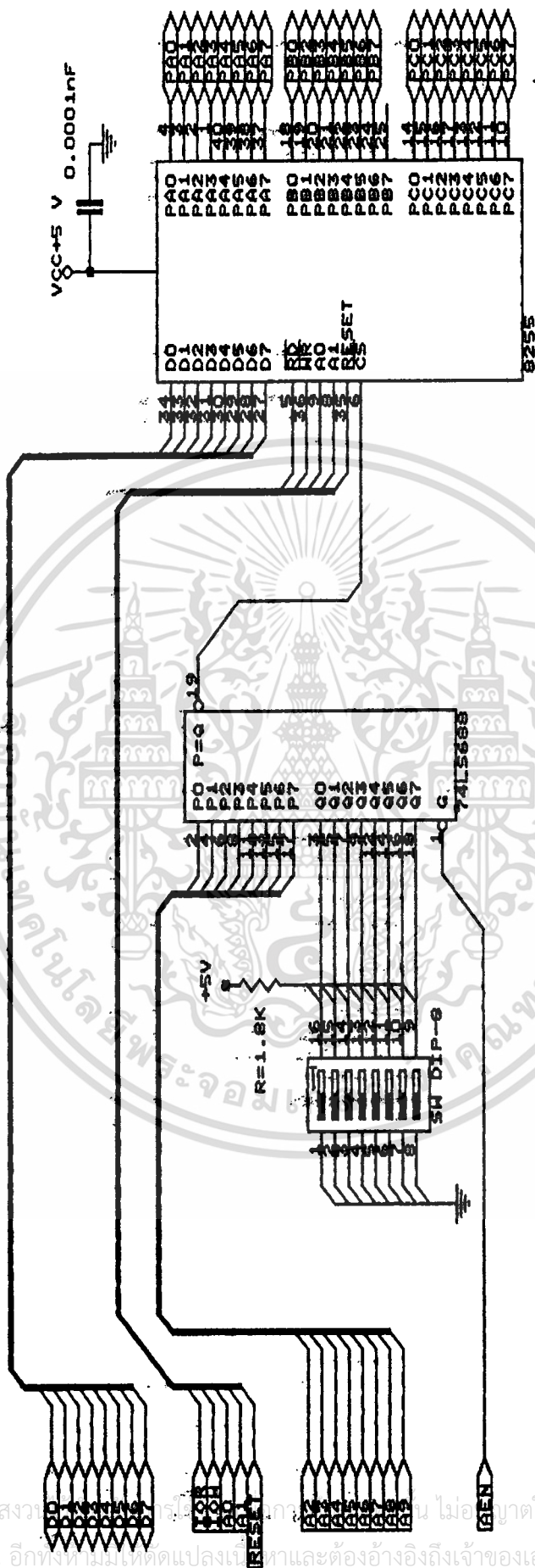
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	x	x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



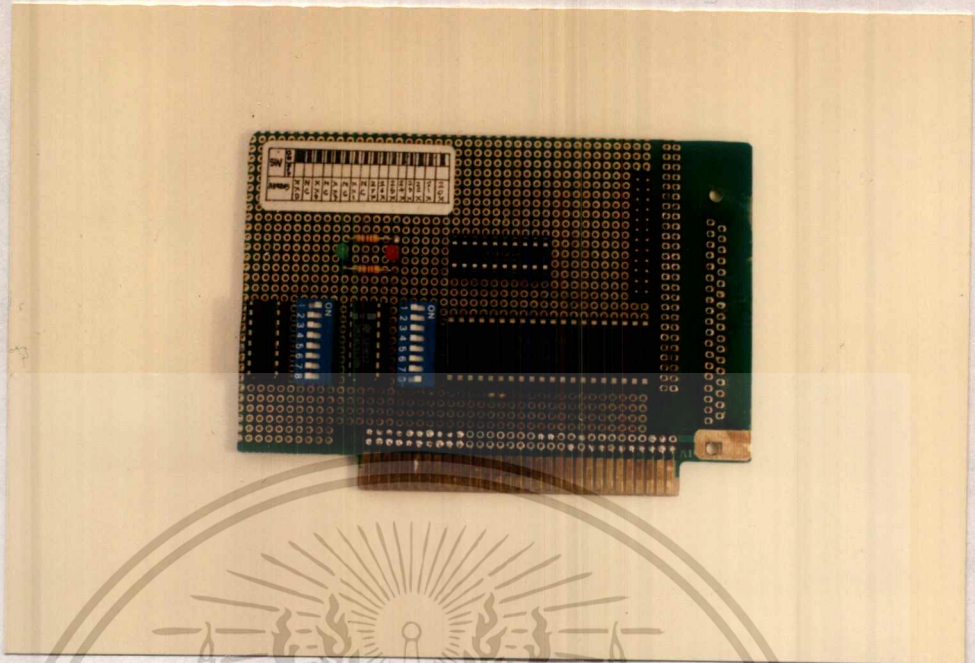
รูปที่ 2.8 แสดงการทำงานของวงจรถัดโค้ดแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงวงจรสมบูร์ณของวงจรอินเทอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงแผ่นปรีนของการ์ดที่ทำเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วงจร A/D และ D/A

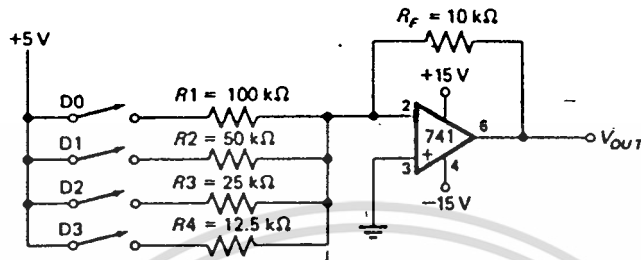
วงจร D/A แบบใช้ความต้านทานหลายค่า

วงจร D/A แบบนี้ใช้ตัวต้านทานและออปแอมป์เพื่อเปลี่ยนระดับสัญญาณลอจิก 2 ระดับเป็นแรงดันที่ได้สัดส่วนกันดังรูปที่ 3.1 แสดงวงจรเปลี่ยนสัญญาณดิจิทัลขนาด 4 บิตออปแอมป์ที่ใช้มีเกนสูงมาก มีความต้านทานเข้าที่พหุค่า ความต้านทานอินพุตสูงมาก สิ่งสำคัญที่สุดคือสัญญาณที่เข้าที่พหุถูกป้อนกลับมายังอินพุตแบบไม่กลับเฟส เข้าที่พหุของออปแอมป์จะเป็นตัวจ่ายหรือรับกระแสเพื่อให้แรงดันที่เปรียบเทียบนั้นมีค่าเดียว วงจรในรูปที่ 3.1 ต่อขาไม่กลับเฟสลงกราวด์ดังนั้นที่ขากลับเฟสก็จะมีแรงดัน 0 โวลต์ด้วยโดยไม่ได้ต่อลงกราวด์ตรงๆ เรียกว่า Virtual ground มาดูที่สวิทช์ D_0 ปิดตัวต้านทาน R_1 ค่า 100 K จะมีแรงดัน 5 โวลต์ ที่ปลายข้างหนึ่งอีกข้างหนึ่งเป็น 0 จากกฎของโอมจะมีแรงดันตกคร่อม 5 โวลต์ กระแสไหลผ่าน 0.05 มิลลิแอมป์กระแสนี้ไม่อาจเข้าไปยังอินพุตของ ออปแอมป์ ได้ดังนั้นจึงไหลผ่านไปยังเข้าที่พหุ โดยผ่านความต้านทานป้อนกลับ R_f เท่ากับ 10 K จะได้แรงดันเข้าที่พหุเท่ากับ $10K \times -0.05$ มิลลิแอมป์เท่ากับ -0.5 โวลต์

เมื่อเปิดวงจรที่สวิทช์ D_0 และปิดวงจรที่สวิทช์ D_1 (R_2 เป็นครึ่งหนึ่งของ R_1) กระแสจะเพิ่มเป็น 2 เท่าหรือ 0.5 มิลลิแอมป์ไหลผ่าน R_f และ R_2 ทำให้มีแรงดันที่เข้าที่พหุเป็น -1 โวลต์ต่อไปก็ปิดวงจร D_0, D_1 จะได้กระแส 0.05 มิลลิแอมป์ไหลผ่าน R_1 และ 0.1 มิลลิแอมป์ผ่าน R_2 รวมกระแสเป็น 0.15 มิลลิแอมป์ได้แรงดันเข้าที่พหุ -1.5 โวลต์

แรงดันเข้าที่พหุจะเพิ่มขึ้นเรื่อยๆ เหมือนขั้นบันไดดังนั้น 4 บิตได้ถึง 15 ระดับ แต่ละระดับต่างกัน -0.5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงวงจร D/A แบบใช้ความต้านหลายค่าแบบใช้ไอซี (Monolithic and hybrid D/A converters)

โมโนลิทิก หมายถึง "หินก้อนเดียว" เมื่อนำมาใช้ในวงจรรวมจะเป็นการชี้บอกว่า วงจรหนึ่งถูกบรรจุอยู่ในสารกึ่งตัวนำเพียงตัวเดียว

ไฮบริด บรรจุสารกึ่งตัวนำที่เรียกว่า ชิพ เพียงตัวเดียวหรือมากกว่า มีตัวต้านทานหรือตัวประกอบวงจรอื่น ๆ อยู่ในกรอบของไอซีตัวเดียว

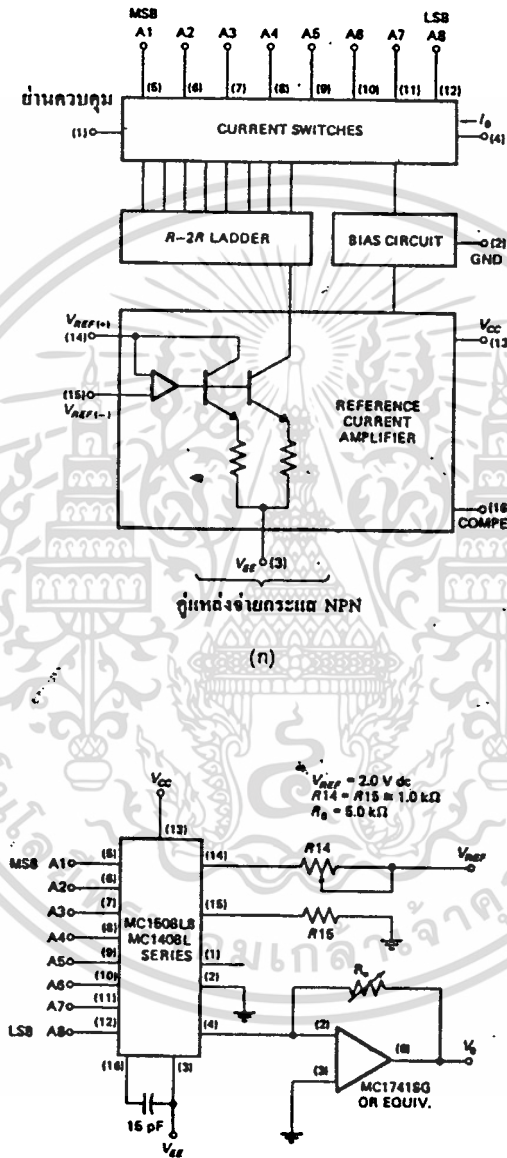
ตัวอย่างของไอซีเปลี่ยนสัญญาณดิจิทัลเป็นแอนะล็อกขนาด 8 บิตคือ MC 1408 มีผังการทำงานดังรูปที่ 3.2 1408L เป็น DIP (Dual In-line Package) 16ขา ให้ $V_{CC} +5V$ และ V_{EE} จาก $-5V$ ถึง $-15V$ ใน 1408L R/2R แลตเตอร์แบ่งกระแสที่ได้จากภาคขยายเป็น 8 ระดับขึ้นอยู่กับค่าทางเลขฐานสอง (binary)

ทรานซิสเตอร์แบบไบโพลาร์จะสวิตช์ให้กระแสที่สอดคล้องกับอินพุต A1 ถึง A8 การเรียงจากบิตที่มีนัยสำคัญต่ำสุดจะกลับกันกับของวงจรนับที่ทั่วไป แต่วงจรเปลี่ยนสัญญาณดิจิทัลเป็นแอนะล็อกบางตัวก็ไม่ได้เรียงอย่างนี้ ควรอ่านคู่มือให้ดีเสียก่อน MC 1408L มีกระแสเข้าที่พุกที่สามารถเปลี่ยนเป็นแรงดันได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบและตัวต้านทานดังในรูปที่ 3.2 แรงดันนี้คำนวณโดยใช้สูตร

$$V_{out} = V_{REF} / R_{14} * R_o (A1/2 + A2/4 + \dots + A6/64 + A7/128 + A8/256)$$



รูปที่ 3.2 แสดงวงจรีชอป MC1408 ของโมโตโรลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจร D/A ที่ใช้ในโครงงาน

เมื่อสัญญาณ D0 - D7 มาจาก port A ของ 8255A สัญญาณจะมารอที่ D0-D7 ของ 74LS273 เพื่อรอสัญญาณจาก port B ของ 8255A ขา B7 มาเข้าที่ขา 11 (CLK) ของ 74LS273 เมื่อ 74LS273 รับสัญญาณเข้ามาที่ขา 11 ซึ่งเป็น Clock ทำงานที่ขอบขาขึ้น หลังจากนั้นจะนำ D0- D7 ที่อินพุตออกที่เอาต์พุต Q1-Q8 ข้อมูลจะถูก Latch จนกว่าจะมีข้อมูลชุดใหม่มาและมีสัญญาณมาที่ขา 11 อีกครั้ง เอาต์พุตของ 74LS273 จะถูกต่อเข้าไปยังขาของไอซี MC1408 ที่ขา A1-A8 (MC1408) ทำหน้าที่แปลงสัญญาณอินพุตที่เป็นดิจิทัลเป็นสัญญาณอนาลอกดังรูปที่ 3.6

$$V_o = V_{ref} (A1/2 + A2/4 + \dots A8/256)$$

V_o ที่ได้จะมีค่าเป็นลบ (ขา4) เพื่อปรับเกนและปรับเป็นไฟบวกโดยใช้วงจรถ่ายกลับ Inverting จะได้ V_o ที่มีค่าอยู่ระหว่าง 0 - 10 โวลต์

วงจรถ่ายกลับโดยใช้วงจรถ่ายกลับเทียบขนาดหรือแบบ"เฟรช"

วงจรถ่ายกลับนี้ใช้หลักการง่าย ๆ และเป็นวิธีที่รวดเร็วที่สุดคือใช้วงจรถ่ายกลับเทียบที่ต่อขนานกันดังรูปที่ 3.3 ประกอบด้วยสลอปแอมป์ที่ต่อเป็นวงจรถ่ายกลับและตัวต้านทานต่อไว้เพื่อแบ่งแรงดันที่ขาอินพุตแบบกลับ (inverting) ให้มีขนาดต่างกัน

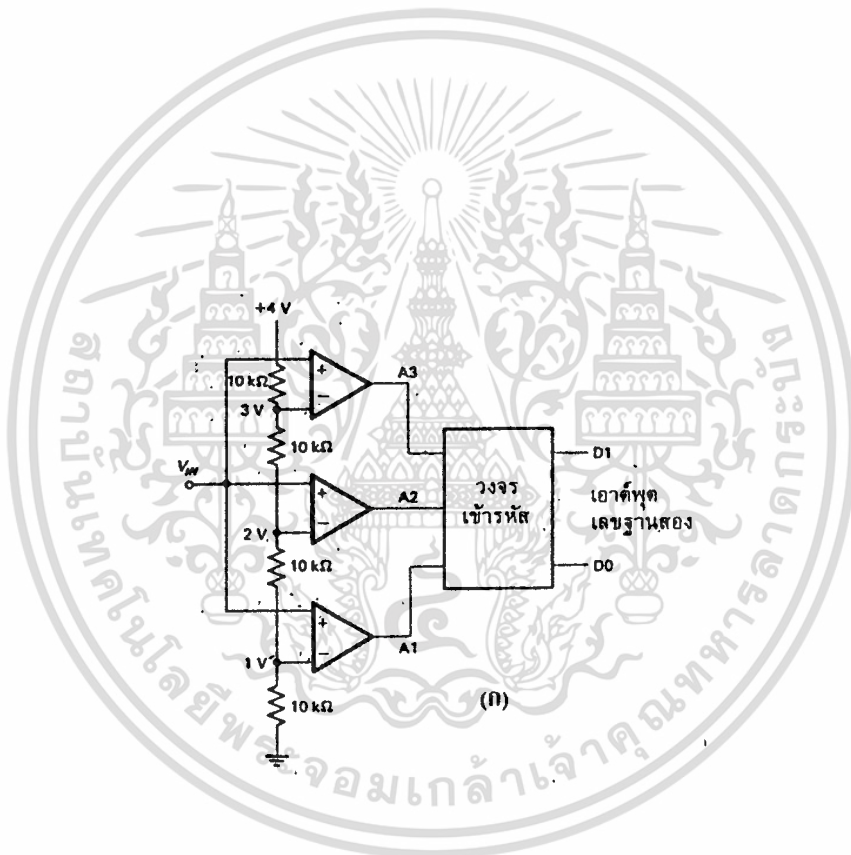
จากหลักการของวงจรถ่ายกลับทั่วไประหว่างขาอินพุตที่ขาอินพุตแบบไม่กลับ(noninverting) มีค่าสูงกว่าที่ขาอินพุตแบบกลับ เอาต์พุตจะได้แรงดันค่าสูงกว่าตารางที่ 3.1 จะเข้าใจยิ่งขึ้นว่าที่แรงดันค่าต่างๆมีผลต่อเอาต์พุตของวงจรถ่ายกลับแต่ละตัวอย่างไรเอาต์พุตที่ได้จากวงจรถ่ายกลับนี้จะนำไปหารที่สเป็นเลขฐานสองต่อไป

จำนวนของวงจรถ่ายกลับที่ต้องใช้ในวงจรถ่ายกลับขึ้นอยู่กับขนาดของสัญญาณอนาลอกที่อินพุต จากวงจรถ่ายกลับในรูปที่ 3.3 ถ้าแรงดันอินพุตมีค่า 1 โวลต์ไม่เพียงพอที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้วงจรเปรียบเทียบตัวใดให้เอาต์พุตเป็น "high" ที่แรงดัน 1 ถึง 2 โวลต์วงจรเปรียบเทียบที่มีระดับเทรชโฮลด์ต่ำสุดก็จะให้เอาต์พุตเป็น "high" แรงดัน 2-3 โวลต์วงจรเปรียบเทียบทั้ง A1 และ A2 ให้เอาต์พุตเป็น "high" ถ้าแรงดันอินพุตมากกว่า 3 โวลต์วงจรเปรียบเทียบก็จะให้เอาต์พุต "high" ทั้งหมด

เมื่อต้องการวงจรที่มีความละเอียดสูงขึ้น จำเป็นต้องใช้วงจรเปรียบเทียบเพิ่มขึ้นเช่น ถ้าต้องการความละเอียด 3 บิตต้องใช้วงจรเปรียบเทียบถึง 7 ตัว ความละเอียด 4 บิตใช้วงจรเปรียบเทียบ 15 ตัว



รูปที่ 3.3 แสดงการต่อวงจร A/D แบบ "เฟรช"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

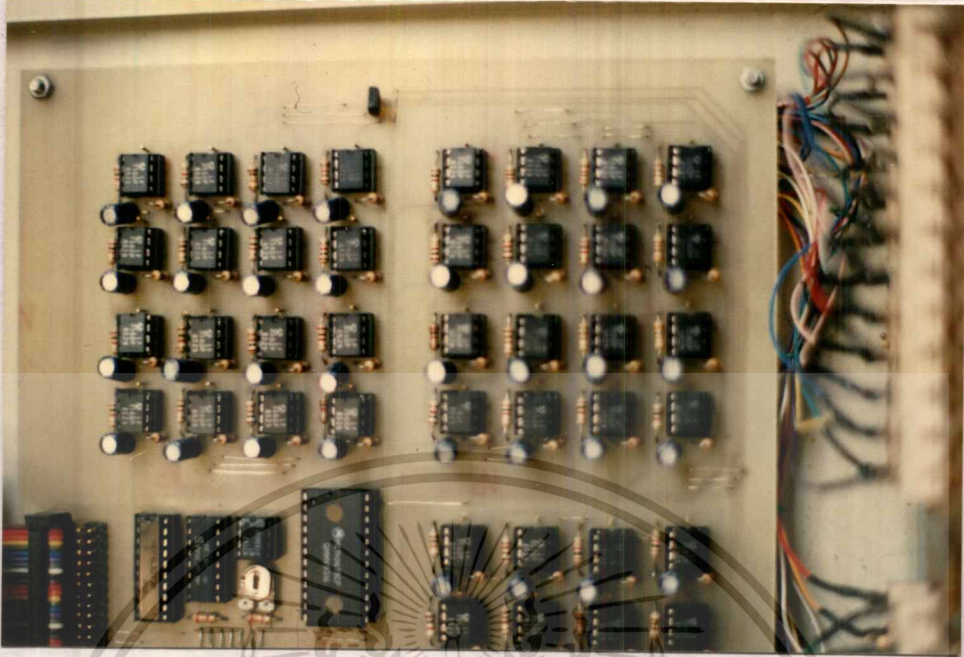
แรงแค้น อินพุต V_{in} (โวลต์)	เอาต์พุตของ วงจรเปรียบเทียบ			เอาต์พุต เลขฐานสอง	
	A1	A2	A3	D1	D0
0 - 1	0	0	0	0	0
1 - 2	1	0	0	0	1
2 - 3	1	1	0	1	0
3 - 4	1	1	1	1	1

ตารางที่ 3.1

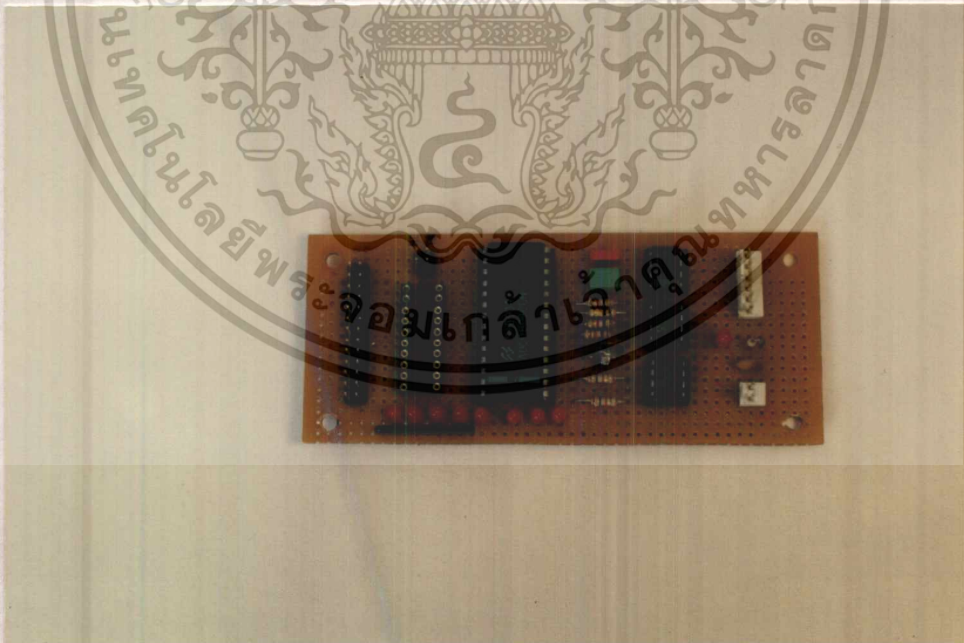
การทำงานของวงจร A/D ที่ใช้ในโครงงาน

วงจร A/D ใช้ไอซี ADC0809 เป็นตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลจากรูปที่ 3.7 เนื่องจาก ADC0809 รับอินพุตได้ 5 VDC แต่เราต้องการอินพุต 10 โวลต์ จึงใช้วงจร Inverting Amp ที่มีอัตราขยาย 0.5 เท่าจะทำให้เอาต์พุต 5 โวลต์เมื่อป้อนอินพุต 10 โวลต์ที่ขา 6 ของ opamp A1 เอาต์พุตที่ได้จะมีค่าเป็นลบ จึงใช้ Inverting ที่มีอัตราขยายเท่ากับ 1 เพื่อปรับไฟเป็นไฟบวก เอาต์พุตที่ออกมาจะป้อนเข้าสู่ขา 26 (INO) ADC0809 สำหรับ Clock ที่ป้อนให้ ADC0809 นั้น สร้างจากไอซี LM555 เอาต์พุตที่ได้จะออกมาจะมีความถี่ 500 KHZ (ขา3) ถูกลดเข้ากับขา 10 (CLK) ของ ADC0809 ส่วน V_{ref} ใช้บวก 5 โวลต์ ต่อที่ขา 12 และลบ V_{ref} ที่ขา 16 ต่อลง Ground สำหรับ A0 - A2 ต่อลง Ground ทั้งหมด ADC0809 จะทำงานเมื่อขา Start, ALE, OE เป็น "1" ADC0809 จะรับค่าสัญญาณอนาล็อกทางด้านอินพุตแปลงเป็นสัญญาณดิจิตอลออกทางด้านเอาต์พุตที่ขา D0 - D7 74LS245 เป็น Buffer ชั้นกลาง ระหว่าง Port A 8255A กับเอาต์พุตของ ADC0809 เมื่อสัญญาณ DIR เป็น "1" และ G เป็น "1" A1 - A8 และ B1 - B8 จะต่อถึงกันเอาต์พุตจะไปปรากฏที่ Port 8255 เพื่อรอให้ CPU อ่านค่าเข้าไปเก็บใน Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

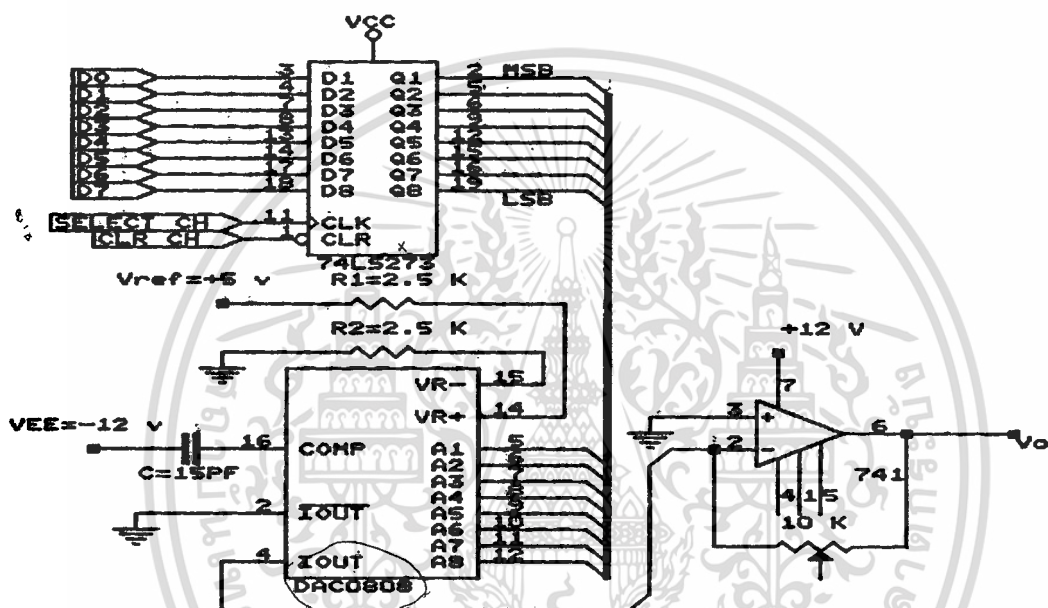


รูปที่ 3.4 แสดงแผ่นปริ้นและอุปกรณ์วงจร D/A



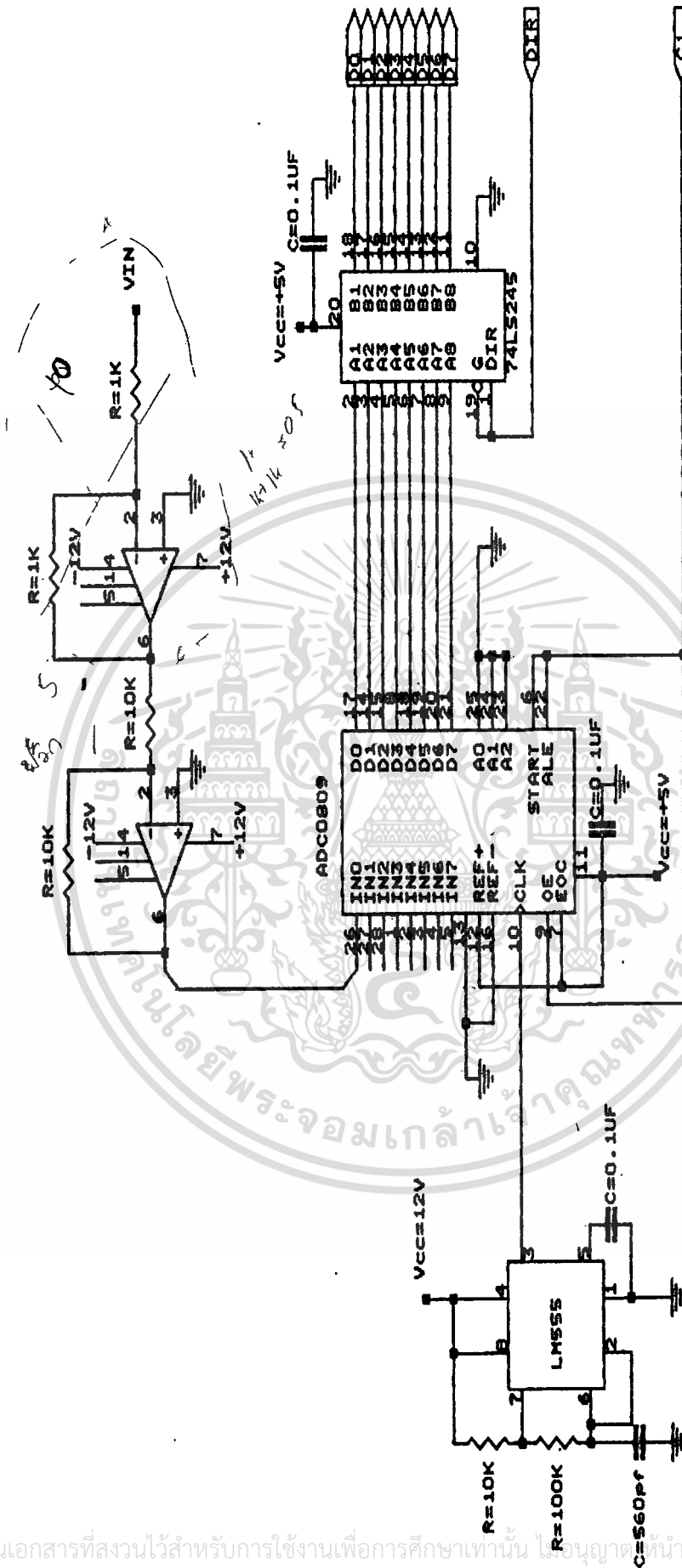
รูปที่ 3.5 แสดงแผ่นปริ้นและอุปกรณ์ของวงจร A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงวงจร D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงวงจร A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

วงจร Decode Channel

วงจรนี้ใช้ไอซีเบอร์ MC14067 ของ Motorola เมื่อสัญญาณดิจิตอลถูกเปลี่ยนเป็นสัญญาณอนาล็อก สัญญาณจะมารออยู่ที่ขา 1 ของ MC14067 เมื่อมีสัญญาณจาก Port B มาเข้าที่ขา S0 - S3 ก็จะทำกาการส่งอินพุตไปยังเอาต์พุต MC14067 มีเอาต์พุต 16 Channel ในปริภูมยานิพจน์นี้ใช้ทั้งหมด 48 Channel จึงต้องใช้ MC14067 ถึง 3 ตัว

ขา 1 เป็นสัญญาณอินพุตอนาล็อก

ขา 10 เป็น S0

ขา 11 เป็น S1

ขา 14 เป็น S2

ขา 13 เป็น S3

ขา 15 เป็น Enable ที่ลอจิก "0"

ขา 24 เป็น Vcc

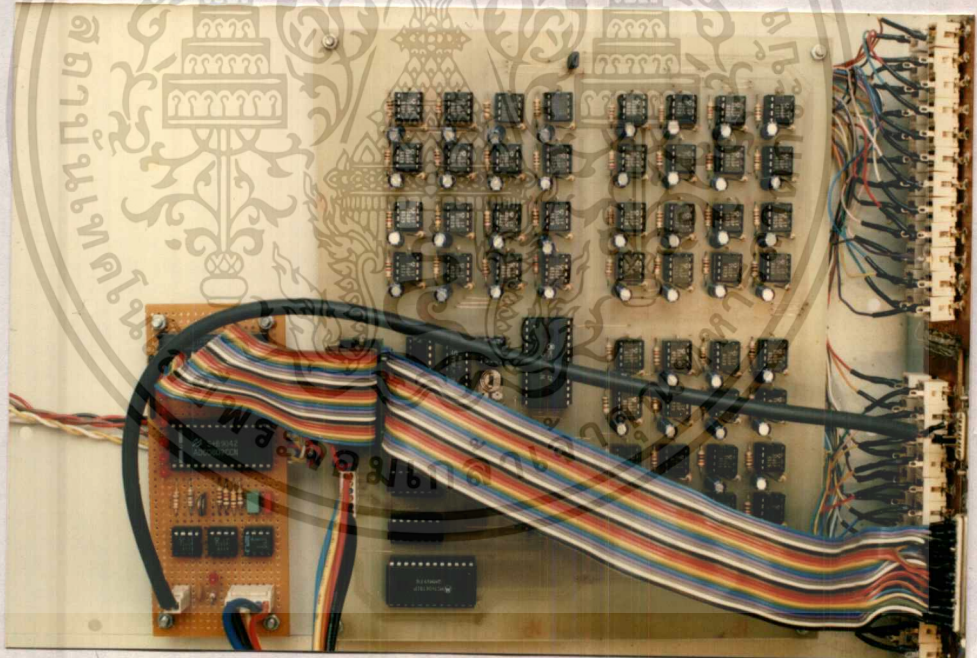
ขา 12 เป็น Ground

ที่เหลือจะเป็นเอาต์พุตทั้งหมด 16 Channel ในการทำงานขา E ต้องเป็น "0" ตลอด ถ้าเป็น "1" จะไม่ทำงาน

S0	S1	S1	S3	E	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A0	A1	A2	A3	A4	CHANNEL
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



รูปที่ 4.1 แสดงการติดตั้งอุปกรณ์บนแผ่นปริ้นวงจรตีโค้ด

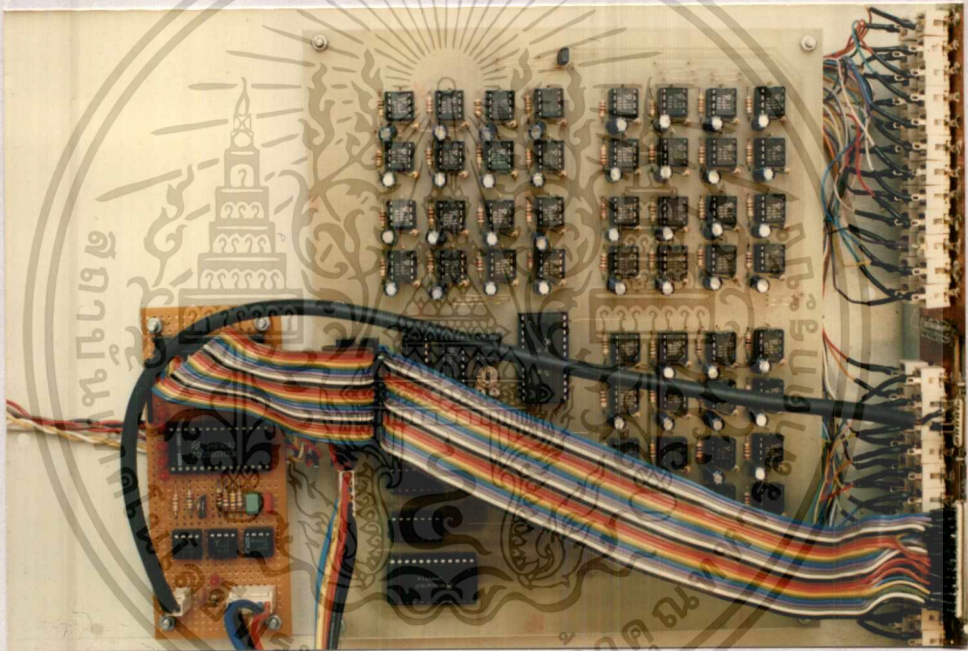
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 วงจร Decode Channel 16 Channel

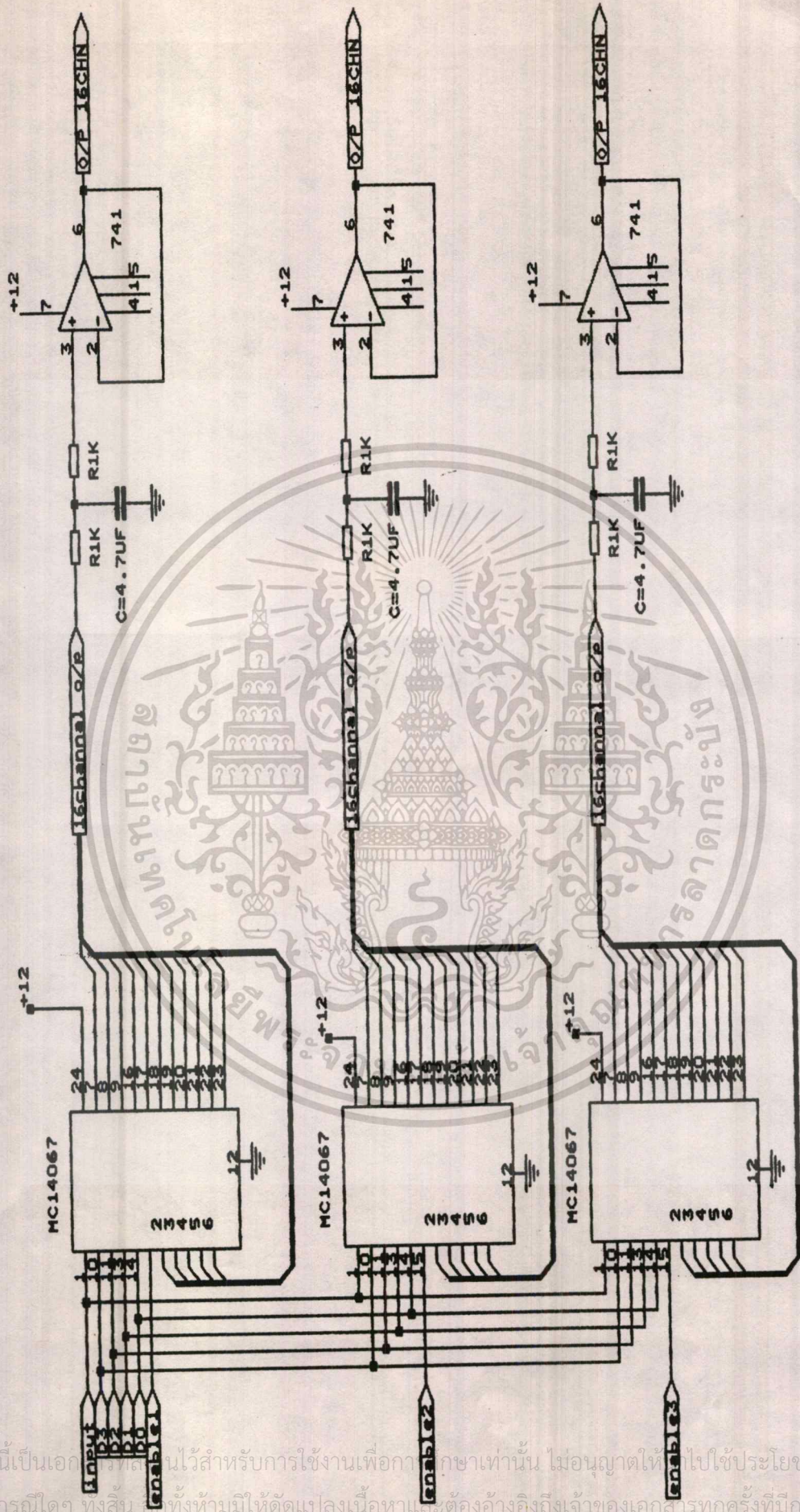
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทำงานของ 14067 จะทำงานทีละตัวโดยควบคุมที่ขา E (ขาที่ 15) ตัวที่ถูกเลือกที่ขา 15 จะมีสภาวะลอจิกเป็น "0" อีก 2 ตัว ตัวที่ขา 15 จะมีลอจิกเป็น "1" เอาท์พุทที่ได้จะนำไปเข้าวงจร RC เพื่อเก็บไว้ใน C และมีวงจร Follower เป็น Buffer เพื่อออกสู่ภายนอกนำไปใช้งาน



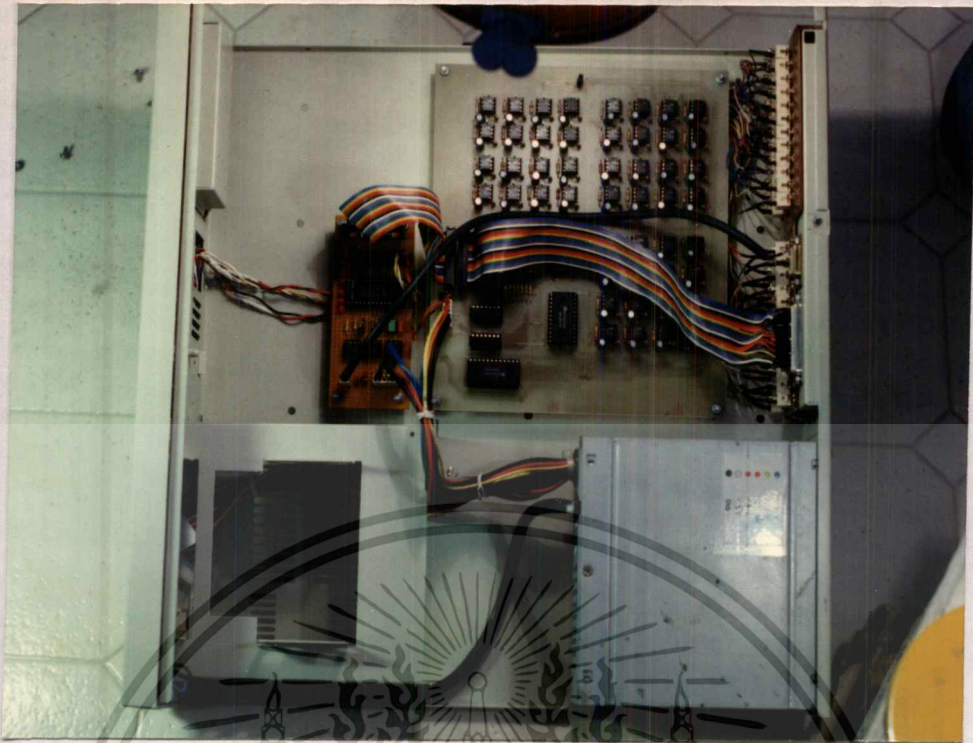
รูปที่ 4.3 แสดงวงจร D/A , A/D , Decode บนแผ่นปริ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

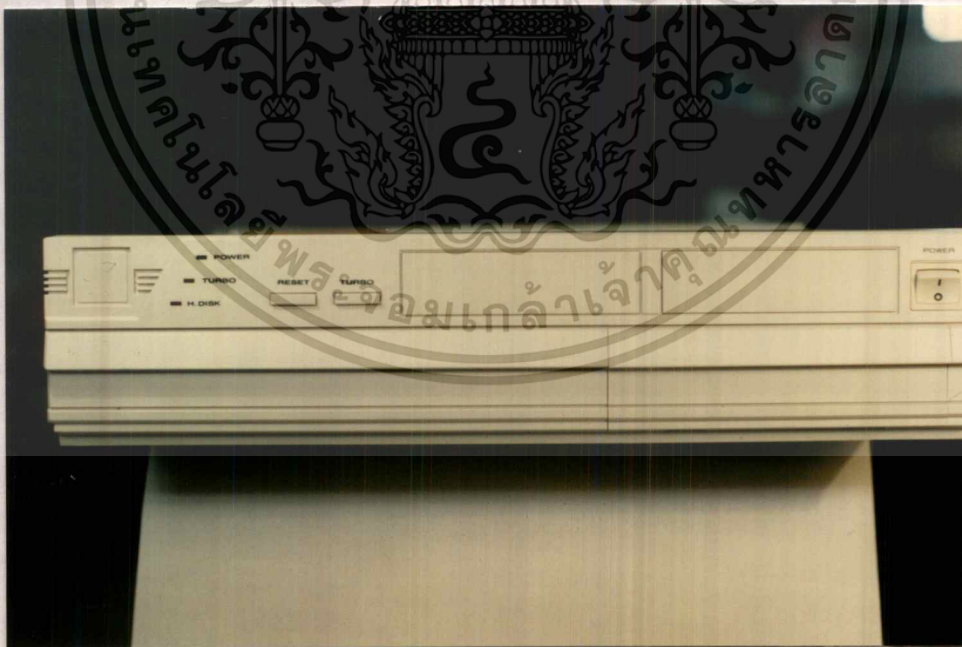


รูปที่ 4.4 แสดงวงจรที่ได้ตัด 48 แชนแนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีให้ตัดแปลงเนื้อหาจะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

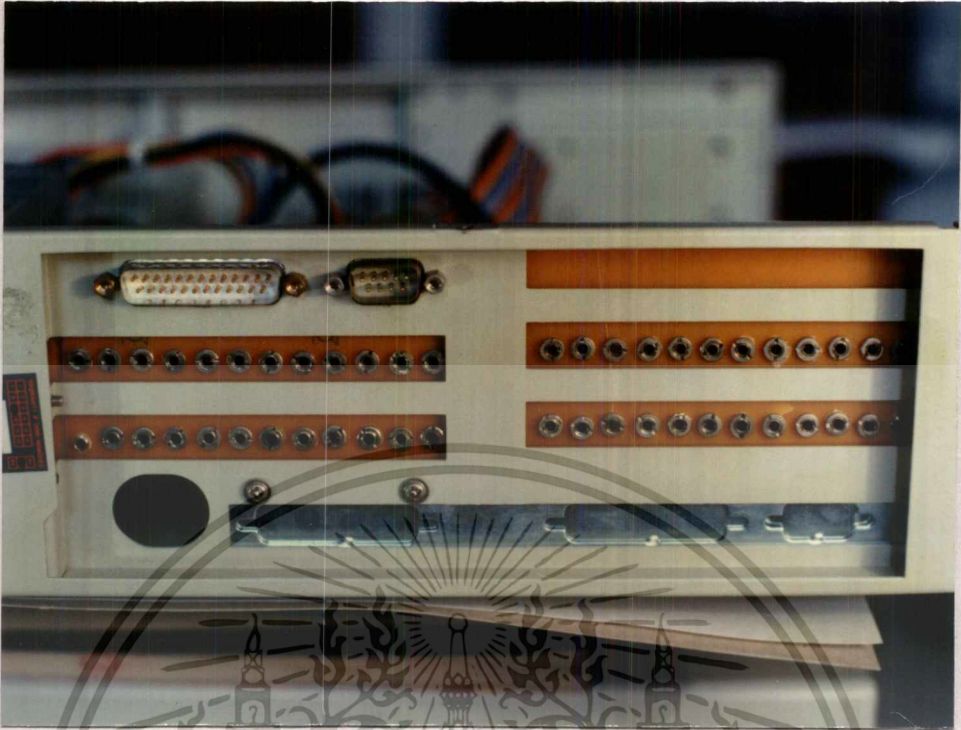


รูปที่ 4.5 แสดงการติดตั้งภายในกล่อง



รูปที่ 4.6 แสดงด้านหน้าของกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงด้านหลังของกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

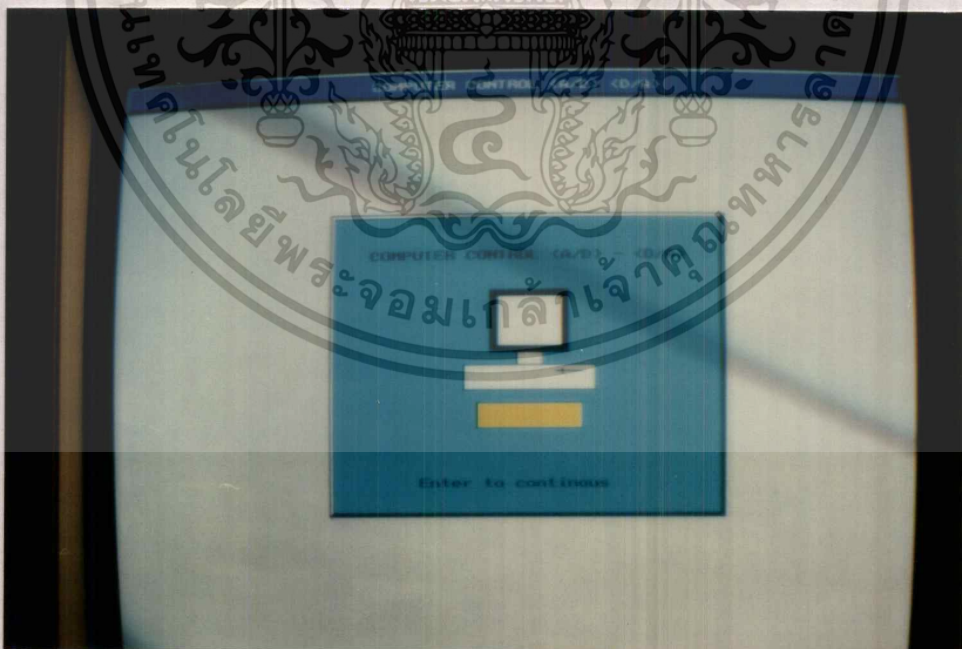
Software และการใช้งาน

Convert Plus Pro เป็น softwear ใช้ควบคุมการทำงานของเครื่อง Digital to Analog & Analog to Digital to Analog Converter พัฒนาขึ้นโดยใช้ภาษา C ผู้ออกแบบโปรแกรมได้ออกแบบให้มีการทำงานติดต่อกับผู้ใช้แบบ GUI (Graphics User Interface) ทำให้โปรแกรมมีความสวยงามและสะดวกในการใช้งาน

ตัวโปรแกรมแบ่งออกได้เป็นสองส่วน คือ ส่วนที่เป็น Graphic และส่วนที่เป็น Controller ในส่วนที่เป็น Graphics จะมีการแบ่งพื้นที่ออกเป็น 3 ส่วน คือ Background windows, working area windows, Button function

การใช้งานโปรแกรม

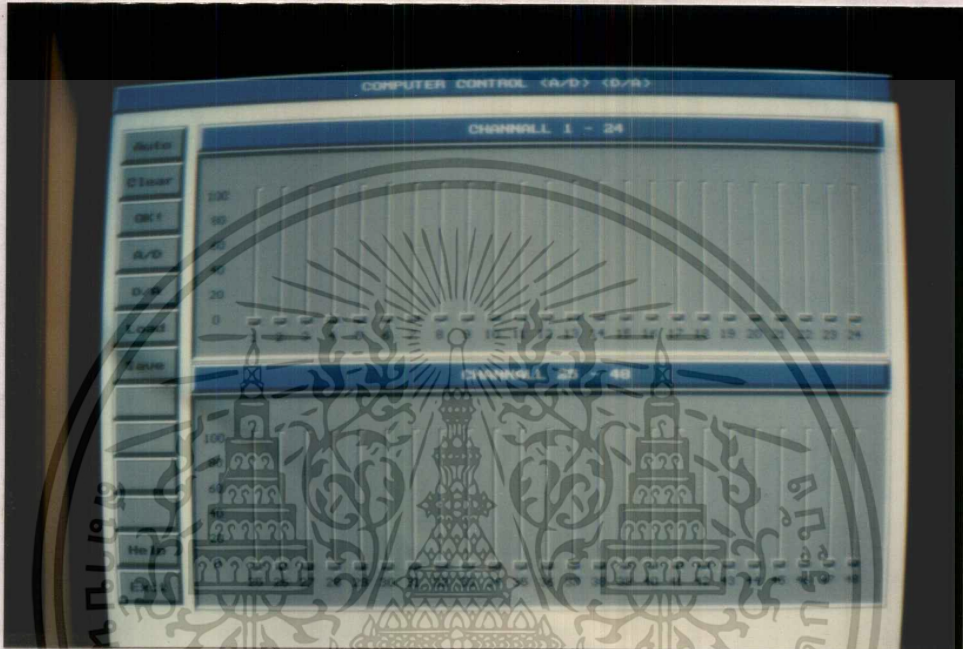
เมื่อ RUN โปรแกรมโดยพิมพ์ pro42 ที่ command line จะพบกับ Demo ของโปรแกรมดังรูป



รูปที่ 5.1 Demo Of Convert plus 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคาะ Enter เพื่อผ่านไปในส่วนการทำงาน Demo ทายไปและปรากฏ working aera และ button function ขึ้นมาดังรูปที่ 5.2 ขณะนี้โปรแกรมก็พร้อมจะรับการติดต่อสั่งงานจากผู้ใช้



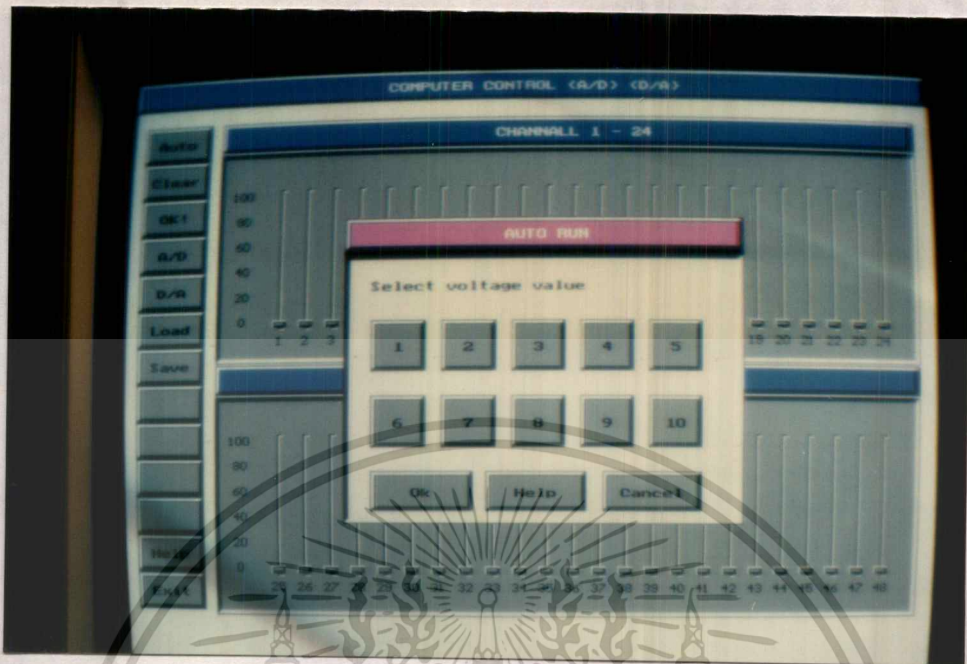
รูปที่ 5.2 working aera and button function

การเลือกใช้งาน function ทำได้โดยการ กด key ตัวอักษรตัวแรกหรือตัวที่สองของ founction จะปรากฏ ไดอะล็อกบ็อกขึ้นมา เพื่อรับข้อมูลที่ใช้งานของ function นั้น ผู้ใช้เพียงแต่กดปุ่มตอบคำถามของไดอะล็อกเท่านั้นนอกจากนี้ผู้ใช้สามารถเรียก Help ขึ้นมาดูได้ตลอดเวลา เมื่อเกิดข้อติดขัดบางอย่าง การใช้งานโดยการกด key 'H' และกด ESC เพื่อออกจาก Help

ฟังก์ชัน Auto

กด 'u' เพื่อเข้าสู่ฟังก์ชัน จะปรากฏไดอะล็อกบ็อกขึ้นมาดังรูป 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 ไดอะล็อกของฟังก์ชัน Auto

ฟังก์ชัน Auto เป็นการส่งค่าแรงดันเอาต์พุตค่าใดค่าหนึ่งจาก 0-10 volt เพียงค่าเดียวทุก Channel ทุกครั้งที่เรียกฟังก์ชันขึ้นมา มันจะทำการ Clear ค่าแรงดันเดิมให้เป็น 0 volt ทุกครั้ง การออกจากการทำงานของฟังก์ชัน โดยการกด ESC

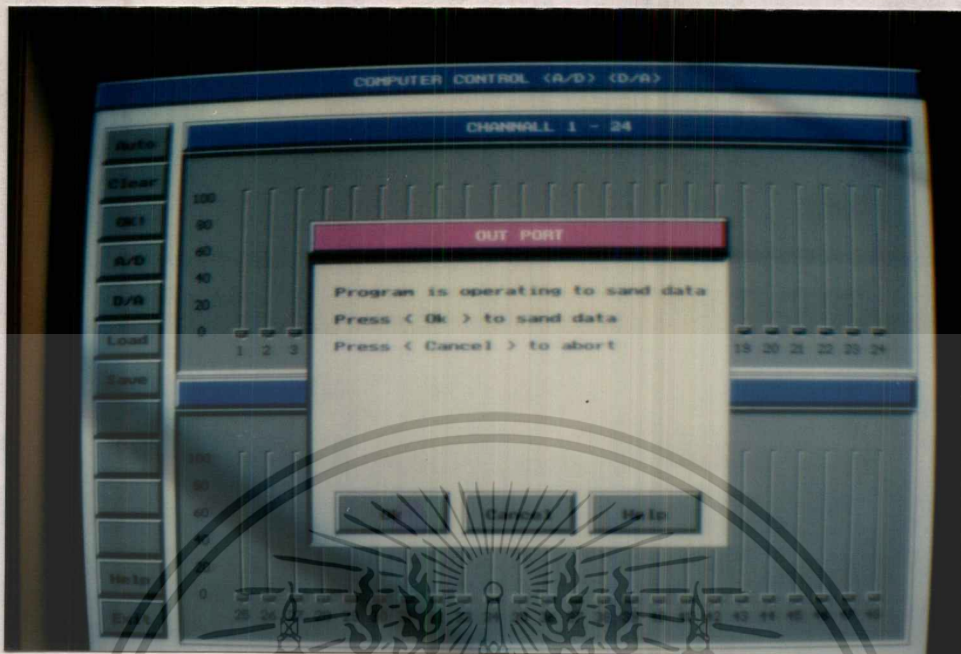
การทำงานกับฟังก์ชันนี้ เพียงแค่กดตัวเลขค่าแรงดันที่ต้องการเท่านั้น มันจะรับค่าไปเก็บไว้เพื่อส่งให้กับฟังก์ชันที่ทำการส่งข้อมูลออกไปสู่การ์ด interface

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' ไดอะล็อกบล็อกของฟังก์ชัน Auto จะหายไปจะปรากฏ Help windows ขึ้นมาทันทีและเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไปและจะปรากฏไดอะล็อกบล็อกของฟังก์ชัน Auto ขึ้นมาทันที

ฟังก์ชัน Out

กด 'o' เพื่อเข้าสู่ฟังก์ชัน จะปรากฏไดอะล็อกบล็อกขึ้นมาดังรูป 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



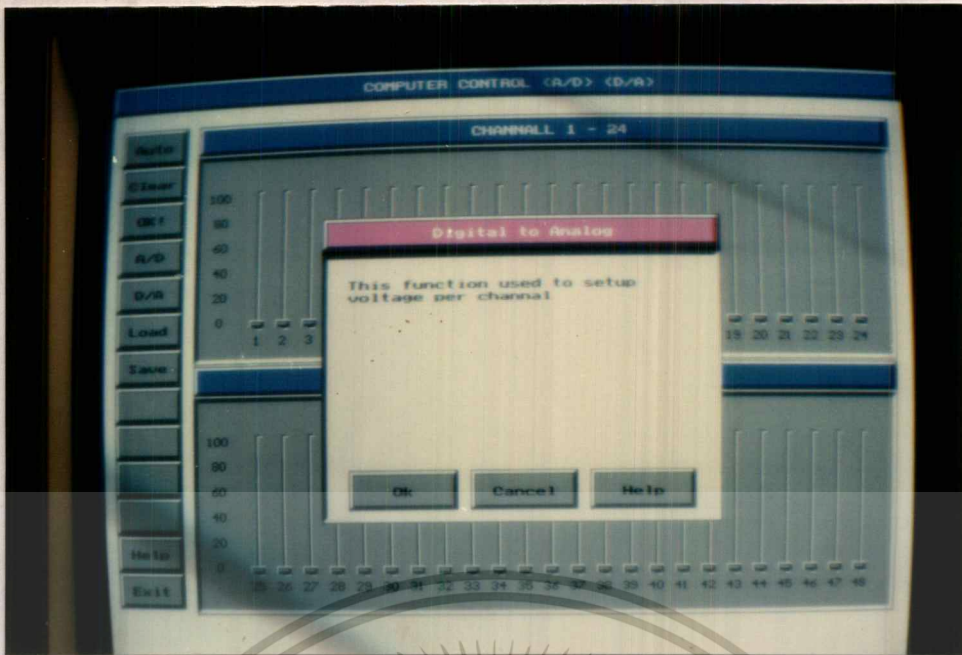
รูปที่ 5.4 ไดอะล็อกของฟังก์ชัน Out

ฟังก์ชัน Out เป็นการส่งค่าแรงดันเอาต์พุตออกไปสู่การ์ด interface ในรูปของเลขฐาน 16 ฟังก์ชัน out จะทำการส่งข้อมูลอย่างต่อเนื่อง เมื่อต้องการ Clear ข้อมูลหรือต้องการ set ข้อมูลใหม่ต้องออกจากฟังก์ชันเสียก่อนเราสามารถออกจากการทำงานของฟังก์ชันโดยการกด ESC

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' ไดอะล็อกบล็อกของฟังก์ชัน Out จะหายไปจะปรากฏ Help windows ขึ้นมาแทนและเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไปและจะปรากฏไดอะล็อกบล็อกของฟังก์ชัน Out ขึ้นมาแทน

ฟังก์ชัน D/A

กด 'd' เพื่อเข้าสู่ฟังก์ชัน จะปรากฏไดอะล็อกบล็อกขึ้นมาดังรูป 5.5



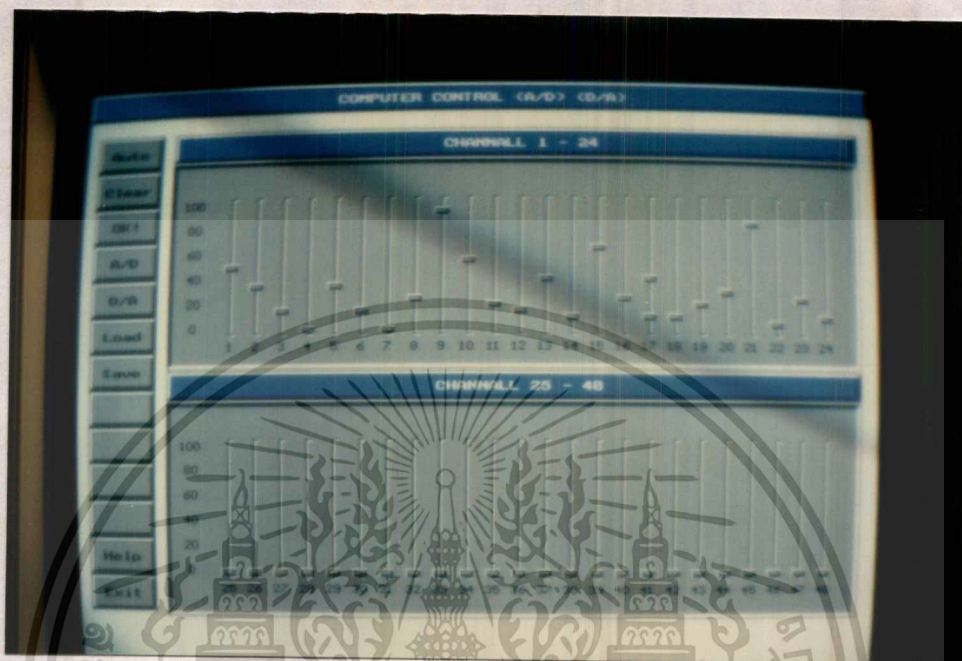
รูปที่ 5.5 ไดอะล็อกของฟังก์ชัน D/A

ฟังก์ชัน D/A เป็นการsetค่าแรงดันเอาต์พุตจาก 0-10 volt ของแต่ละ Channel โดยการใช้ arrow key ร่วมกับ PgUp, PgDn ทุกครั้งที่เรียกฟังก์ชันขึ้นมา มันจะทำการ Clear ค่าแรงดันเดิมให้เป็น 0 volt ทุกครั้งการออกจากการทำงานของฟังก์ชัน โดยการกด ESC

การทำงานกับฟังก์ชันนี้ จะใช้การกด arrow key up/down เพื่อเพิ่ม/ลดแรงดันเอาต์พุต การเปลี่ยน channel โดยการกด arrow key left/right ร่วมกับ key PgUp, PgDn

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' ไดอะล็อกบลิ๊คของฟังก์ชัน D/A จะหายไปจะปรากฏ Help windows ขึ้นมาแทน และเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไป และจะปรากฏไดอะล็อกบลิ๊คของฟังก์ชัน D/A ขึ้นมาแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงการใช้ฟังก์ชัน D/A

ฟังก์ชัน Save

กด 's' เพื่อเข้าสู่ฟังก์ชัน Save เป็นการบันทึกข้อมูลค่าแรงดันเอาต์พุตของแต่ละ channel รูปของเลขฐาน 16 ลงสู่ Disk ออกจากการทำงานของฟังก์ชันโดยการกด ESC

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' ไดอะล็อกบ็อกของฟังก์ชัน Save จะหายไปจะปรากฏ Help windows ขึ้นมาทันทีและเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไปและจะปรากฏไดอะล็อกบ็อกของฟังก์ชัน Save ขึ้นมาทันที

ฟังก์ชัน Load

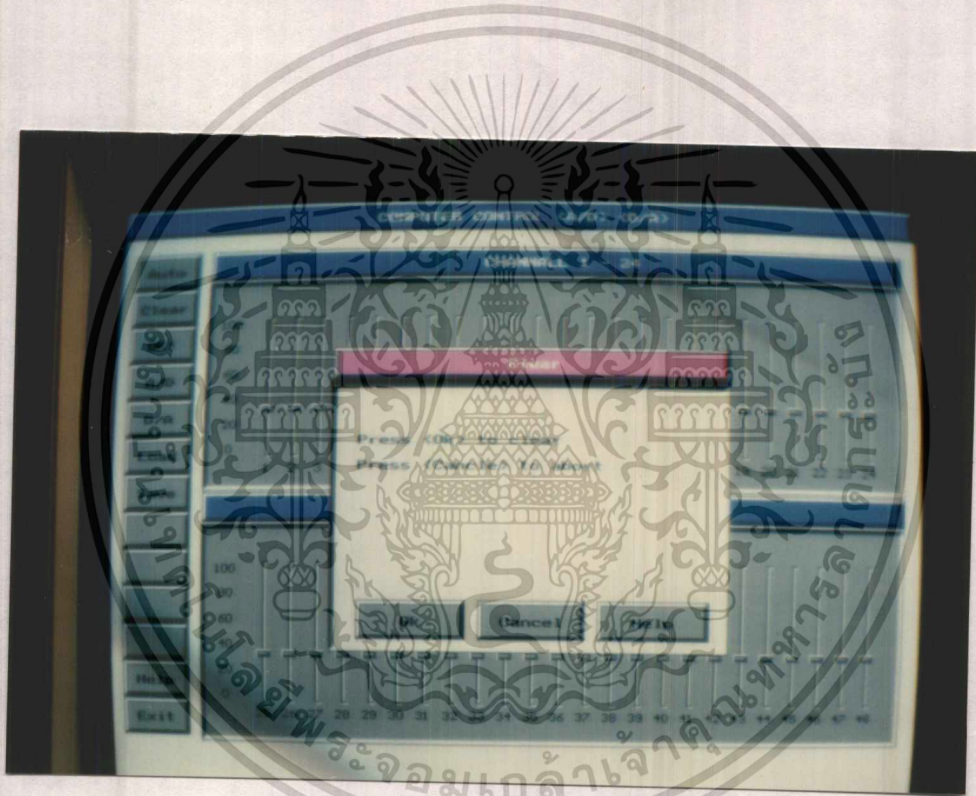
กด 'L' เพื่อเข้าสู่ฟังก์ชัน Load เป็นการอ่านข้อมูลค่าแรงดันเอาต์พุตของแต่ละ channel รูปของเลขฐาน 16 จาก Disk ออกจากการทำงานของฟังก์ชัน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการกด ESC

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'L' ไตอะลือกบล็อกของฟังก์ชัน Load จะหายไปจะปรากฏ Help windows ขึ้นมาทันทีและเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไปและจะปรากฏไตอะลือกบล็อกของฟังก์ชัน Load ขึ้นมาทันที

ฟังก์ชัน Clear

กด 'c' เพื่อเข้าสู่ฟังก์ชัน จะปรากฏไตอะลือกบล็อกขึ้นมามีรูปที่ 5.7



รูปที่ 5.7 ไตอะลือกของฟังก์ชัน Clear

ฟังก์ชัน Clear เป็นการ Clear ค่าแรงดันเอาต์พุตของทุก Channel มันจะทำการ Clear ค่าแรงดันเดิมให้เป็น 0 volt ทุกครั้งการออกจากการทำงานของฟังก์ชัน โดยการกด ESC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' โดยอะลือกบลิ็อกของฟังก์ชัน Clear หายไปจะปรากฏ Help windows ขึ้นมาแทนและเมื่อต้องการออกจาก Help ให้กด ESC Help windows หายไป และจะปรากฏโดยอะลือกบลิ็อกของฟังก์ชัน Clear ขึ้นมาแทน

ฟังก์ชัน A/D

กด 'a' เพื่อเข้าสู่ฟังก์ชัน A/D เป็นการรับค่าแรงดันอินพุตที่ป้อนเข้ามา ในรูปของสัญญาณ Analog โดยได้จากการแปลงสัญญาณของ Analog to digital converter Unit ในส่วนของ Hardware

การทำงานผู้ใช้กดปุ่ม Enter มันก็จะทำการเก็บค่าที่ส่งมาไปไว้ในหน่วยความจำและแสดงผลบนหน้าจอ monitor

การเรียกใช้ Help ในฟังก์ชันทำได้โดยการกด 'h' โดยอะลือกบลิ็อกของฟังก์ชัน A/D จะหายไปจะปรากฏ Help windows ขึ้นมาแทนและเมื่อต้องการออกจาก Help ให้กด ESC Help windows จะหายไปและจะปรากฏโดยอะลือกบลิ็อก

ฟังก์ชัน Exit

กด 'x' เพื่อเข้าสู่ฟังก์ชัน จะปรากฏโดยอะลือกบลิ็อกขึ้นมาดังรูปที่ 5.8

กดปุ่ม Enter เพื่อจบการทำงาน ถ้าไม่ต้องการกด ESC เพื่อทำงานต่อ

ฟังก์ชัน Os-Shell

กด 'e' เพื่อเข้าสู่ฟังก์ชัน

ฟังก์ชันนี้จะออกจากโปรแกรมเพื่อเข้าสู่ Dos ที่ชั่วคราว เมื่อต้องการกลับเข้ามาพิมพ์ EXIT ที่ command line

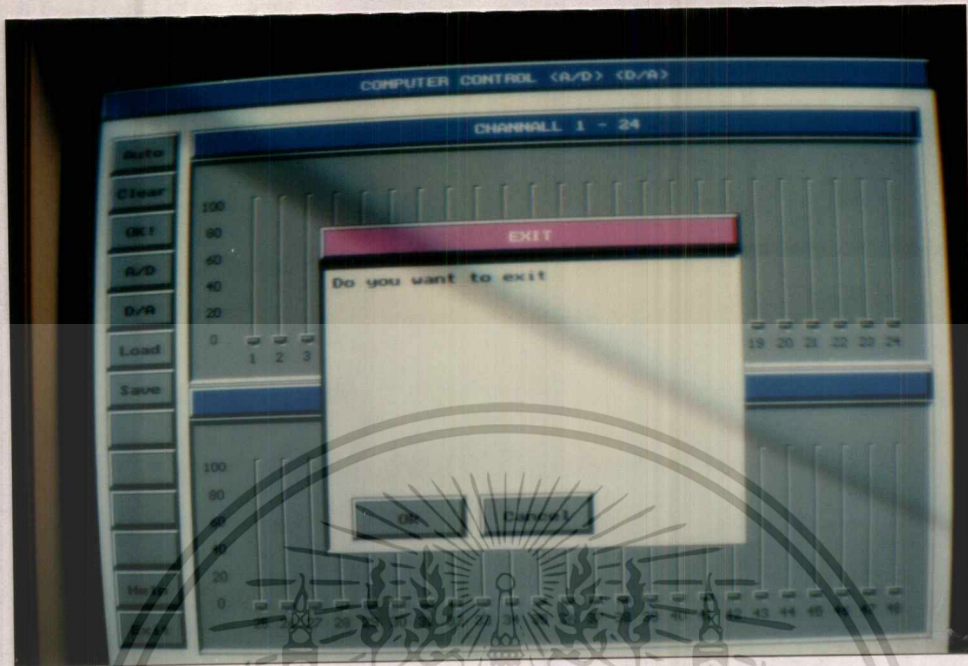
ฟังก์ชัน Help

กด 'h' เพื่อเข้าสู่ฟังก์ชัน

ฟังก์ชัน Help สามารถเรียกใช้ในทุกฟังก์ชัน ยกเว้นตัวมันเองและ os-shell

กด ESC เมื่อต้องการออกจาก Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 ไดอะล็อกของฟังก์ชัน EXIT

สรุป Hot key ที่ใช้งาน

<u>FUNCTION</u>	<u>KEY</u>
Auto	u
Out	o
Clear	c
D/A	d
A/D	a
Load	l
Save	s
Exit	x
Cancel	Esc
Ok	Enter

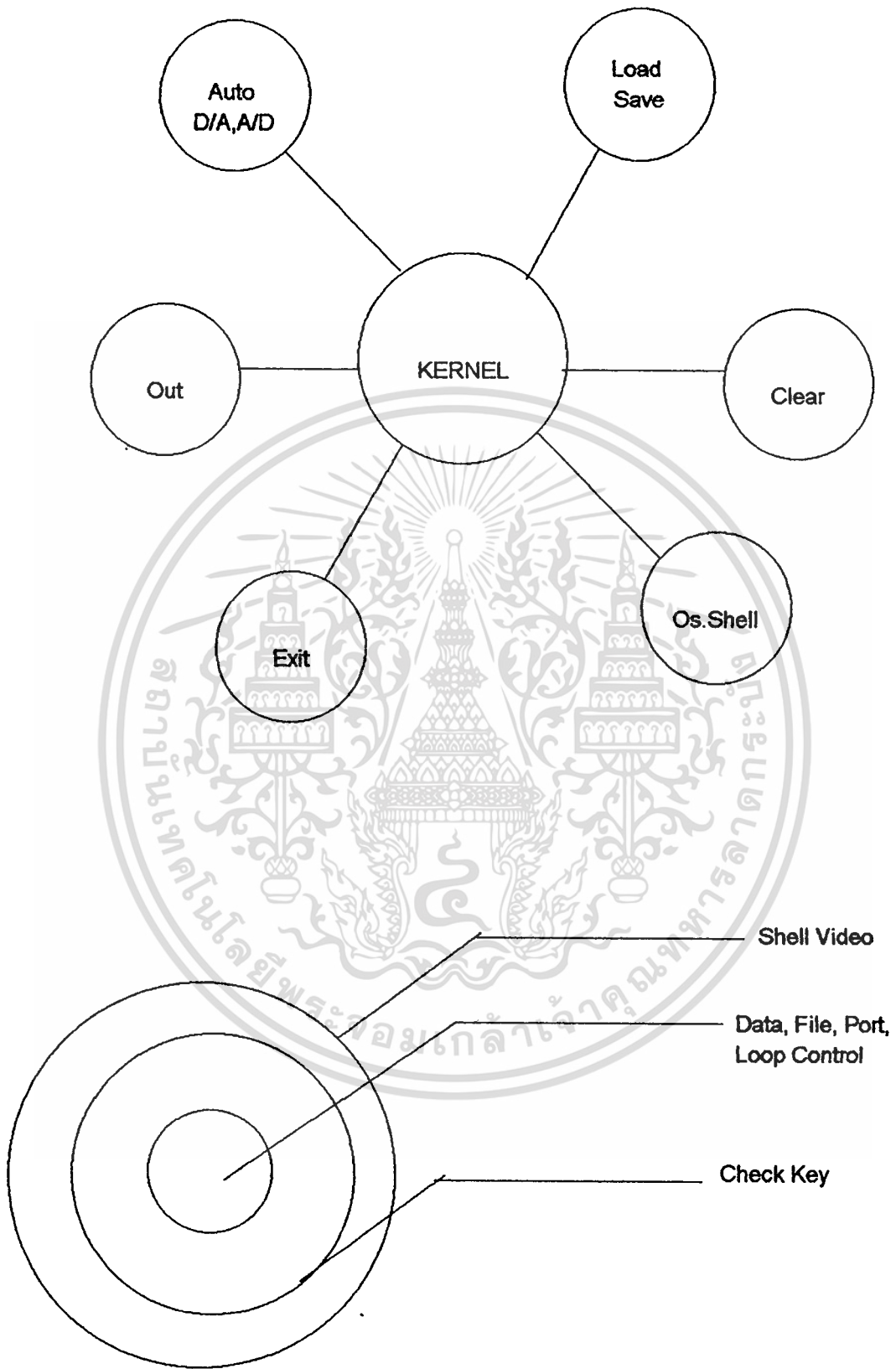
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>FUNCTION</u>	<u>KEY</u>
Os-Shell	e
Devolper	k

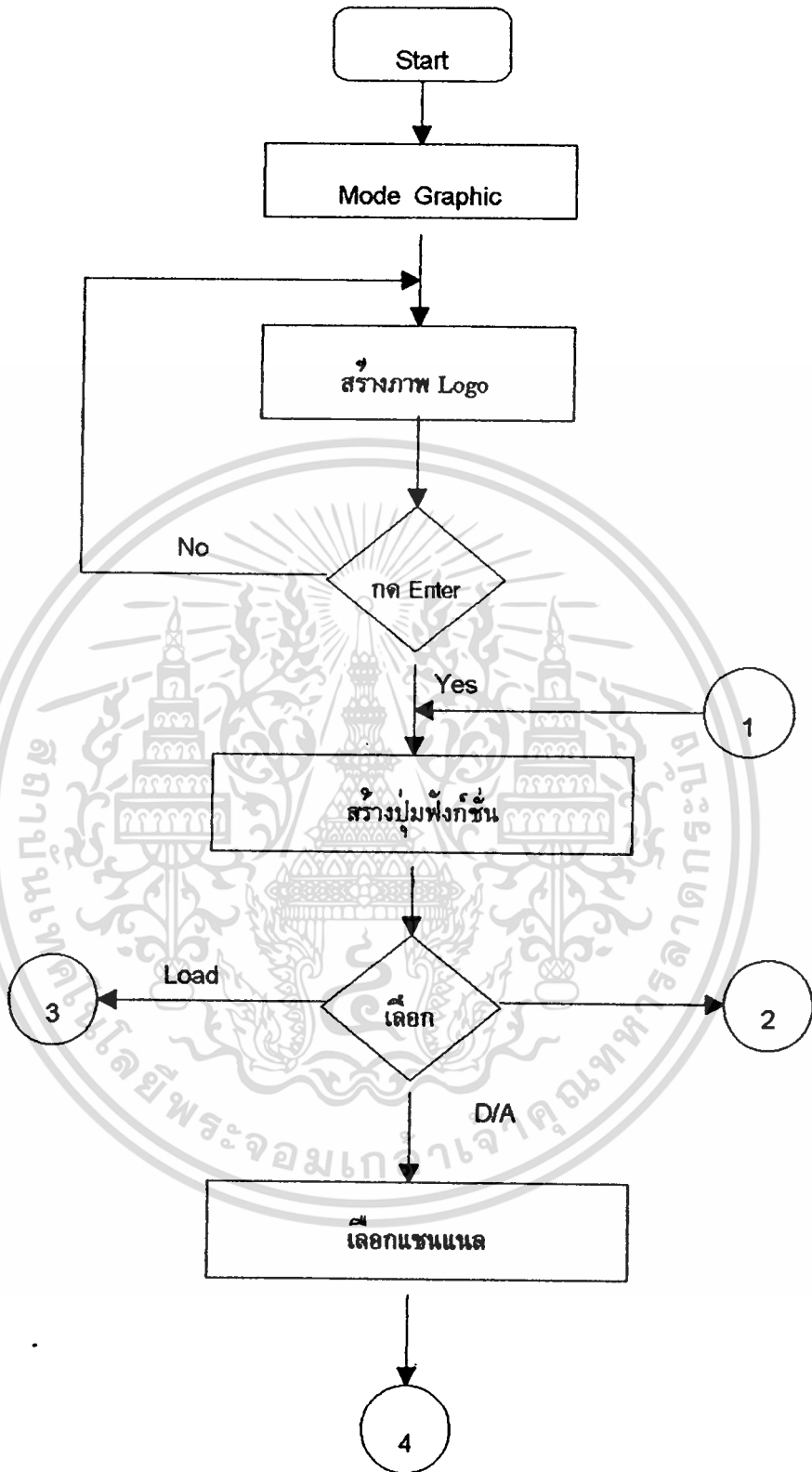
Convert Plus Pro ต้องการเครื่องที่ใช้ CPU 80286 ขึ้นไป มีหน่วยความจำ 1 MB Dos version 3.3 ขึ้นไปจอภาพVGA Monitor เนื่องจากมีการทำงานใน Garphics Mode จึงควรใช้ CPU 80386DX-33 ขึ้นไปจึงให้ผลในการทำงานในส่วนที่เป็น Graphic รวดเร็วขึ้น

แนวทางพัฒนาโปรแกรมนี้ผู้พัฒนาได้สร้าง Function tool kit ในส่วนของ Graphics ไว้มากพอที่จะนำมาใช้ออกแบบโปรแกรมให้มีความสามารถในการทำงานร่วมกันของ Function ในส่วนที่ไป Control Hardwear เช่นการให้แรงดันเ้าท์ทุกเป็น step by step การให้เ้าท์ทุกแบบ real time การตั้งเวลาของเ้าท์ทุกแต่ละ channal ให้จ่ายแรงดันออกมาในเวลาต่างๆกันเพื่อใช้ในการควบคุม start dc motor เป็นต้นที่สำคัญอย่างซึ่งควรมีการใช้เ้าท์กับโปรแกรมได้จะทำให้มีการใช้งานได้สะดวกยิ่งขึ้น

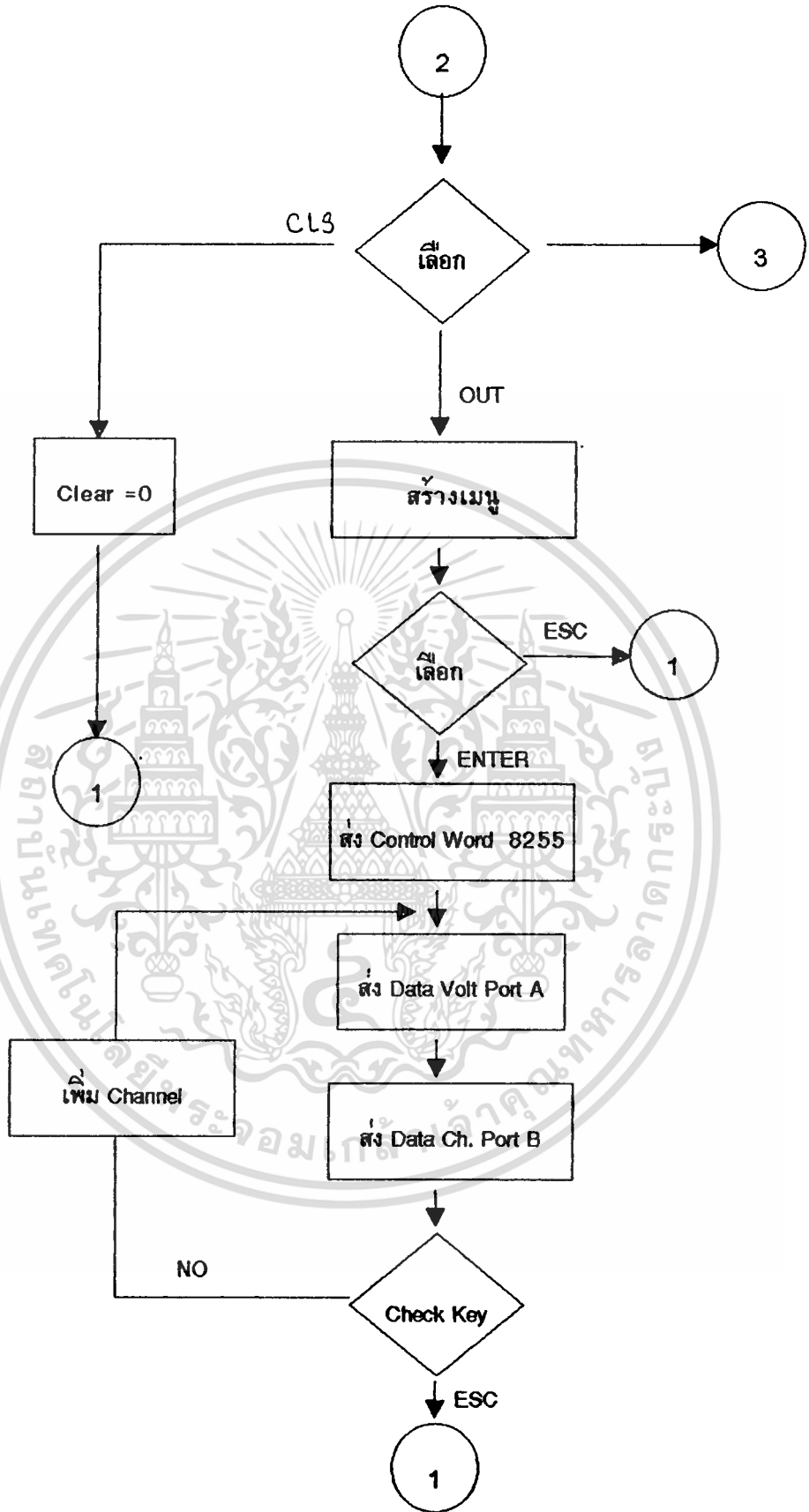
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



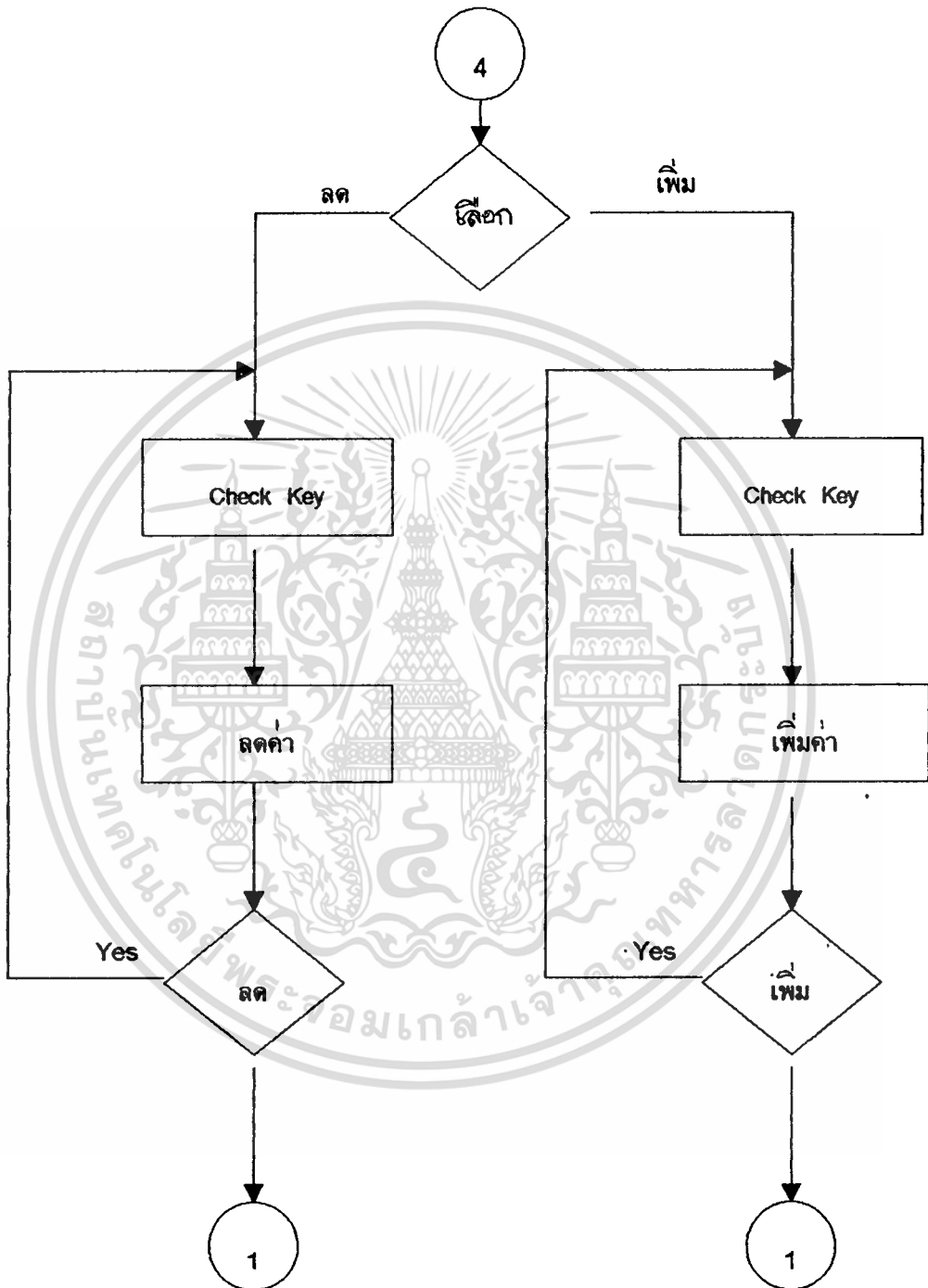
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



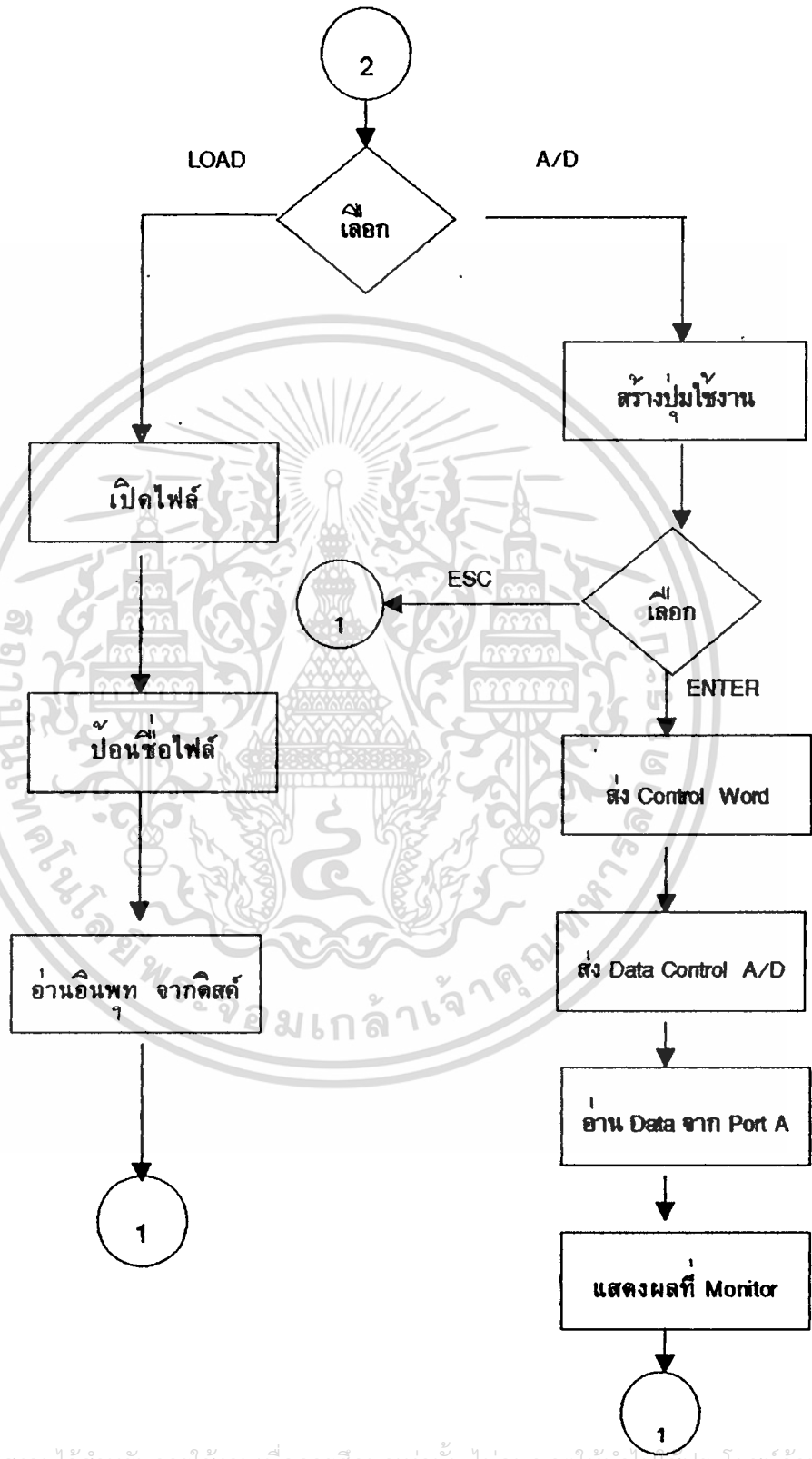
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



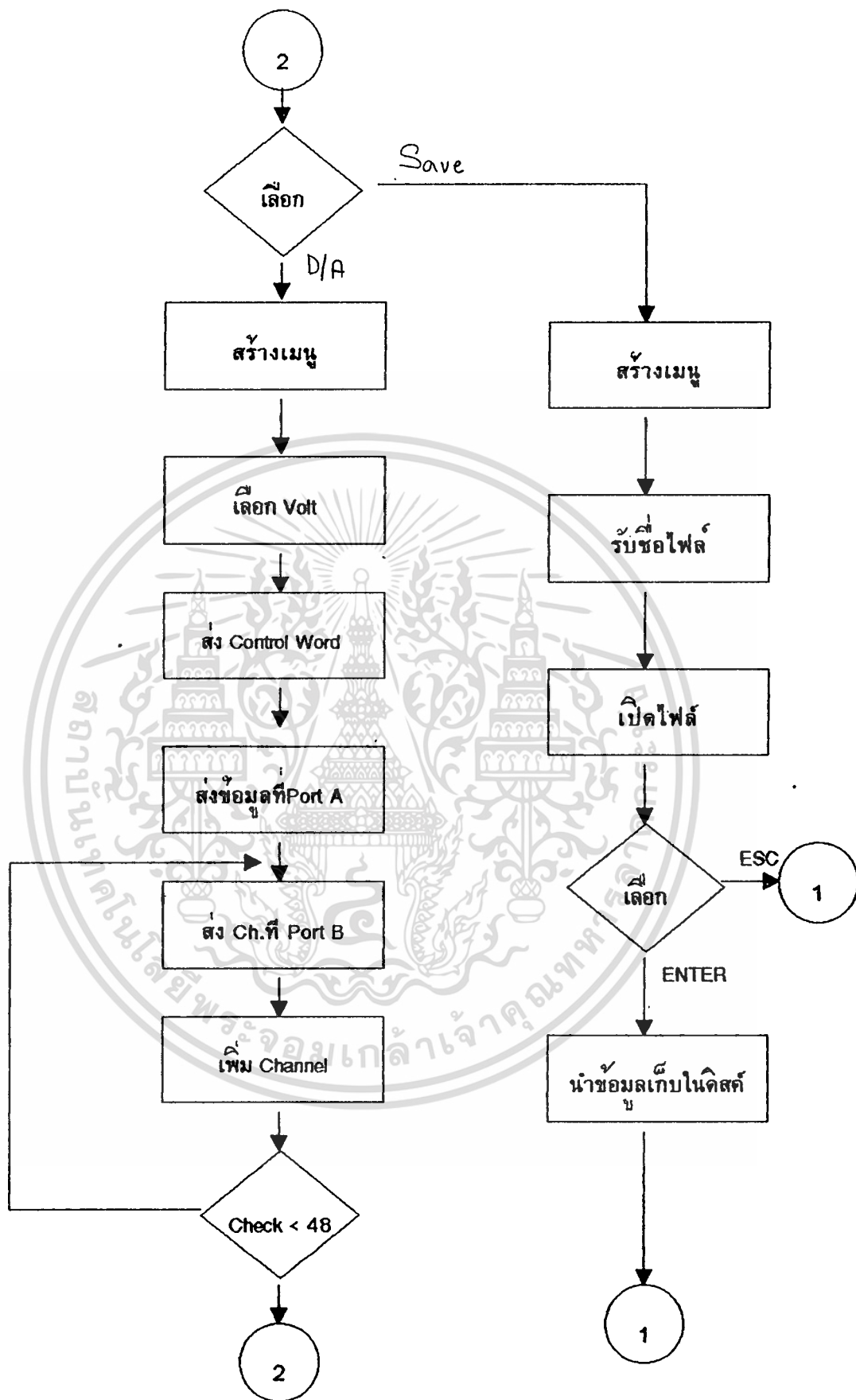
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



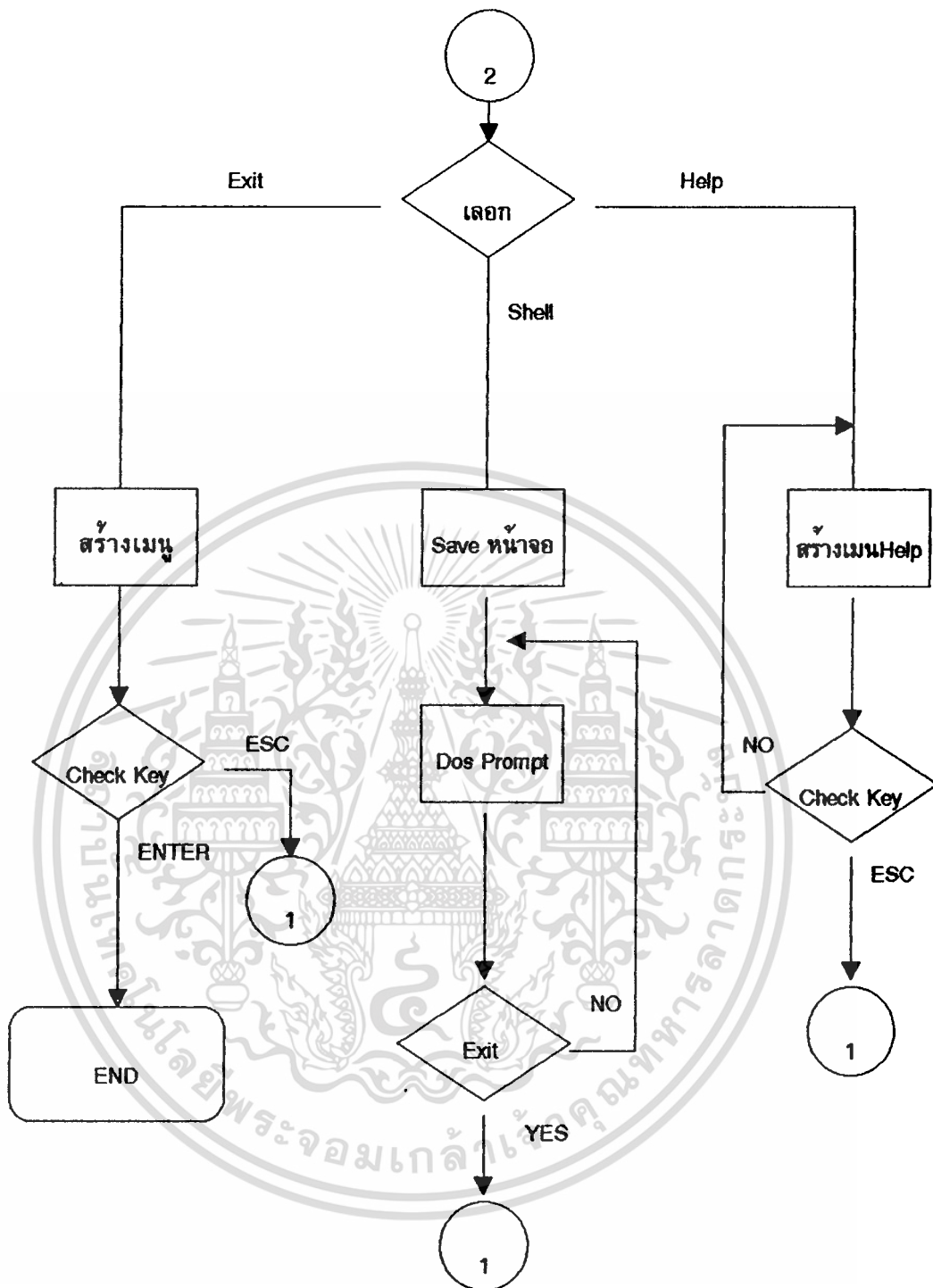
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและข้อเสนอแนะ

ในการจัดทำโครงการนี้ เพื่อเป็นการศึกษาถึงการควบคุมวงจร A/D, D/A ว่าเราสามารถควบคุมได้อย่างไร

ในปัจจุบันนี้มีการใช้งานวงจร A/D D/A มีใช้อย่างแพร่หลายในการศึกษา ครั้งนี้จะสามารถเป็นประโยชน์สำหรับผู้สนใจในภาชนะหน้าได้เป็นอย่างดี สำหรับคุณสมบัติของวงจร A/D D/A ที่ควรศึกษายังมีรายละเอียดมากกว่าในหนังสือเล่มนี้ จะขอยกตัวอย่างเช่น ลักษณะการป้องกันสัญญาณรบกวน, ความผิดพลาดที่เกิดขึ้นในวงจร D/A , ความผิดพลาดเนื่องจากเกน(gain error) , ความผิดพลาดที่วงจรไม่เป็นเชิงเส้น(linearity error) เป็นต้นซึ่งมิได้กล่าวเลยแต่สามารถศึกษาเพิ่มเติมได้จากเอกสารอ้างอิงตามที่แสดงไว้

ในส่วนของวงจรควบคุมใช้ IC 74LS273 เป็นตัวรับข้อมูลนับว่ามีความสะดวกและประหยัดมากกว่าการใช้ไมโครคอนโทรลเลอร์โดยรับการควบคุมจากคอมพิวเตอร์โดยตรง จากการทดลองด้วยโปรแกรมต่าง พบว่าการควบคุมแบบแสดงผลทางกราฟิกนั้นจะใช้งานได้ดีกว่าและแม่นยำมากกว่า แต่ในการควบคุมยังมีข้อผิดพลาดอยู่มาก เมื่อทำการวิเคราะห์สามารถเห็นข้อผิดพลาดดังนี้

การใช้งาน IC Digital To Analog # MC1408 ค่อนข้างจะเสียง่ายในการใช้งานจริงควรเปลี่ยนมาใช้วงจรแบบ R-2R Ladder เป็นชุด DAC แทนจะเหมาะสมกว่าสำหรับงานที่ไม่ต้องการความละเอียดมากนัก

สำหรับ IC Analog To Digital # 0809 เสียง่ายเช่นกันควรใช้วงจรแบบเปรียบเทียบชานหรือแบบ"flash"ให้หลักการง่ายแต่ก็มีข้อเสียคือเมื่อต้องการความละเอียดมากขึ้นต้องเพิ่มวงจรเปรียบเทียบแต่ก็มีข้อดีคือ มีความเร็วสูงมากเป็น นาโนวินาที

อย่างไรก็ตามโครงการนี้ก็มิวัตถุประสงค์เพื่อใช้ในการศึกษา ทดลอง เพื่อเป็นพื้นฐานแนวคิด ในการนำไปใช้งานและเพื่อพัฒนาต่อไปในอนาคต

เอกสารอ้างอิง

1. รศ. กฤษดา วิชาชีวานนท์, "ไอซีดีจีตอล", บริษัท ซีเอ็ดดูเคชั่น จำกัด, หน้า 204-208 พิมพ์ครั้งที่ 2 พ.ศ. 2531
2. ชานินทร์ ถาวรศาสนวงศ์, ทินกร ดูก, "การอินเตอร์เฟส IBM PC บริษัท ฟิสิกส์เซ็นเตอร์ การพิมพ์, หน้า 1-51, 191-196, พิมพ์ครั้งที่ 1
3. กนกศักดิ์, รัฐชน, "COMPUTER CONTROL DIMMER" วิทยานิพนธ์ภาควิชา เทคโนโลยีอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. "คู่มือเทียบเบอร์ไอซี TTL", บริษัท ซีเอ็ดดูเคชั่น จำกัด , 2522
5. เปรมจิตร วิสุมศิริ, "พื้นฐานวงจร เลขดี ดีทูเอ" เซมิคอนดักเตอร์, ฉบับที่ 102 ตุลาคม 2533 หน้า 272-279
6. เปรมจิตร วิสุมศิริ, "พื้นฐานวงจร เลขดี ดีทูเอ" เซมิคอนดักเตอร์, ฉบับที่ 103 ธันวาคม 2533 หน้า 302-309
7. "TURBO C", BORLAND INTERNATIONAL INC, FIRST PRINTING 1987 U.S.A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/**

FILE NAME : CONVERT-PLUS-PRO
Graphics Programmer : Pimonsak limsuttipong
Control Programmer : Pakdee kongpol
Create : 10 nov 1993
Update :
Purpose : Software control Hardware <D/A>,<A/D>
Comment : This can compile by TURBO C Compiler
only and can use only VGA Driver and
Monitor

Copyright (c) 1993 King Mongkut's Institute of Technology Ladkrabang
(KMIT'L)**/

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<dos.h>
#include<process.h>
#include<graphics.h>
#include<conio.h>
#include<string.h>
#include<bios.h>
#include<alloc.h>
#include"channal.i"
#include"data.i"
#include"subpro.i"
#include"help.i"
#include"voids.i"
#include"shell.i"
#include"scancode.i"
#include"read.i"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define port0    0x303
#define shipa    0x300
#define shipb    0x301
#define shipc    0x302

char message[30],data[30],mode[10],string[255];
int volt=199;
int oop=0;
int channel=130;
int out_data=0;
int flag_da=0;
int group = 0;
int ob_volt=0x00;
int dac=0,opp=0;
float data_volt[48];
init_graph()
{
    int g_driver=DETECT;
    int g_mode;
    char g_path;
    initgraph(&g_driver,&g_mode," ");
    readfont();
    return(0);
}

void *save_image(int left,int top,int right,int bottom)
{
    void far *image;
    image = farmalloc(imagesize(left,top,right,bottom));
    getimage(left,top,right,bottom,image);
    putimage(left,top,image,XOR_PUT);
    return(image);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void *save_image2(int left,int top,int right,int buttom)
{
void far *image;
image = farmalloc(imagesize(left,top,right,buttom));
getimage(left,top,right,buttom,image);
putimage(left,top,image,OR_PUT);
return(image);
}

```

```

int init_save()

```

```

{
dip6=save_image2(125,95,610,210);
putimage(125,95,dip6,COPY_PUT);
dip7=save_image2(125,290,610,400);
putimage(125,290,dip7,COPY_PUT);
return(0);
}

```

```

windows(left,top,right,buttom,message,data)

```

```

char *message,*data;

```

```

{
win = save_image(left,top,right,buttom);
frame_3d(left,top,right,buttom-215,message,WHITE,LIGHTMAGENTA);
windows_message(left,top+29,right,buttom,data);
return(0);
}

```

```

windows_help(left,top,right,buttom,message,data)

```

```

char *message,*data;

```

```

{
box = save_image(left,top,right,buttom);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frame_3d(left,top,right,bottom-215,message,BLACK,LIGHTGREEN);
windows_message(left,top+29,right,bottom,data);
return(0);
}
int demo()
{
    int key;
    int left=180,top=120,right=480,bottom=360;
    box = save_image2(left,top,right,bottom);
    frame_3d(left,top,right,bottom,"",BLACK,LIGHTCYAN);
    icon();
    demo_key();
    return(0);
}
int icon()
{
    setcolor(BLACK);
    rectangle(300,180,360,230);
    setfillstyle(1,WHITE);
    floodfill(303,184,BLACK);
    rectangle(305,185,355,225);
    setfillstyle(1,WHITE);
    floodfill(306,186,BLACK);
    rectangle(280,235,380,255);
    setfillstyle(1,WHITE);
    floodfill(290,240,BLACK);
    rectangle(290,260,370,280);
    setfillstyle(1,WHITE);
    floodfill(295,270,BLACK);
    rectangle(320,230,340,235);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setfillstyle(1,WHITE);
floodfill(325,232,BLACK);
setcolor(LIGHTBLUE);
settextstyle(2,0,1);
setusercharsize(120,170,120,170);
outtextxy(290,238,"IBM");
setcolor(BLACK);
settextstyle(2,0,1);
setusercharsize(110,100,110,100);
line(350,245,370,245);
line(355,243,355,247);
outtextxy(330,140,"WELCOME TO");
outtextxy(330,155,"COMPUTER CONTROL <A/D> - <D/A>");
outtextxy(315,320,"      Enter to continous");
settextstyle(0,0,1);
return(0);
}
void os_shell(void)
{
void far *ptr[10];
int s;
setviewport(0,0,639,479,1);
s=save_screen(ptr,0,0,639,479);
if(s>0)
{
closegraph();
printf("\t\tType 'EXIT' to return program...\n");
system("");
init_graph();
restore_screen(ptr,0,0,639,479);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
beep()
{
    union
    {
long divisor;
unsigned char c[2];
    } count;
    unsigned char p;
    unsigned i;
    count.divisor = 119382/420;
    outportb(67,182);
    outportb(66,count.c[0]);
    outportb(66,count.c[1]);
    p = inportb(97);
    outportb(97,p ! 3);
    for(i=0;i<4000;i++);
    outportb(97,p);
    return(0);
}
press(x1,y1,x2,y2,message)
char *message;
{
    beep();
    botton_inv(x1,y1,x2,y2,message);
    delay(250);
    botton(x1,y1,x2,y2,message);
    return(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int demo_key()
{
    int key=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 13 : /* Enter */
beep();
frame_3d_inv(180,120,480,360,"",BLACK,LIGHTCYAN);
delay(250);
frame_3d(180,120,480,360,"",BLACK,LIGHTCYAN);
delay(250);
putimage(180,120,box,COPY_PUT);
break;
        case 'k': /* Present*/
break;
        default : demo_key();
    }
    return(0);
}

void read_key_exit()
{
    int key=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 27 : press(290,320,370,350,"Cancel");
                putimage(180,120,win,COPY_PUT);
                break;
        case 13 : press(200,320,280,350,"Ok");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    close_system();
default : read_key_exit();
    }
}
int read_key_auto()
{
    int key=0,i;
    key = bioskey(0);
    switch(tolower(key))
    {
case '1': press(200,200,240,240,"1");
        volt=189;
        putimage(180,120,win,COPY_PUT);
        break;
case '2': press(252,200,292,240,"2");
        volt=179;
        putimage(180,120,win,COPY_PUT);
        break;
case '3': press(304,200,344,240,"3");
        volt=169;
        putimage(180,120,win,COPY_PUT);
        break;
case '4': press(356,200,396,240,"4");
        volt=159;
        putimage(180,120,win,COPY_PUT);
        break;
case '5': press(408,200,448,240,"5");
        volt=149;
        putimage(180,120,win,COPY_PUT);
        break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case '6': press(200,260,240,300,"6");
    volt=139;
    putimage(180,120,win,COPY_PUT);
    break;
case '7': press(252,260,292,300,"7");
    volt=129;
    putimage(180,120,win,COPY_PUT);
    break;
case '8': press(304,260,344,300,"8");
    volt=119;
    putimage(180,120,win,COPY_PUT);
    break;
case '9': press(356,260,396,300,"9");
    volt=109;
    putimage(180,120,win,COPY_PUT);
    break;
case '0': press(408,260,448,300,"10");
    volt=99;
    putimage(180,120,win,COPY_PUT);
    break;
case 'h': press(285,320,360,350,"Help");
    help_auto();
    volt=0;
    read_key_auto();
    break;
case 27 : press(373,320,448,350,"Cancel");
    volt=0;
    putimage(180,120,win,COPY_PUT);
    break;
/* case 13 : press(200,320,275,350,"Ok");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    volt=1;

        putimage(180,120,win,COPY_PUT);

    break;*/
default : read_key_auto();
    }
    return(volt);
}

void read_key_save()
{
    int key=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
break;
        case 13 : press(200,320,280,350,"Ok");
putimage(180,120,win,COPY_PUT);
break;
        case 'h' : press(380,320,460,350,"Help");
help_save();
read_key_save();
break;
        default : read_key_save();
    }
}

void read_key_load()
{
    int key=0;
    key = bioskey(0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(tolower(key))
{
    case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
break;

    case 13 : press(200,320,280,350,"Ok");
putimage(180,120,win,COPY_PUT);
break;

    case 'h': press(380,320,460,350,"Help");
help_save();
read_key_load();
break;

    default : read_key_load();
}
}

int read_key_out()
{
    int key=0,flag_out=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
flag_out=0;
break;

        case 13 : press(200,320,280,350,"Ok");
/*putimage(180,120,win,COPY_PUT);*/
flag_out=1;
break;

        case 'h': press(380,320,460,350,"Help");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

help_out();
flag_out=0;
read_key_out();
break;
    default : read_key_out();
    }
    return(flag_out);
}

int read_key_digital()
{
    int key=0,digit=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
digit=0;
break;
        case 13 :press(200,320,280,350,"Ok");
putimage(180,120,win,COPY_PUT);
digit=1;
break;
        case 'h' : press(380,320,460,350,"Help");
help_digital();
digit=0;
read_key_digital();
break;
        default : read_key_digital();
    }
    return(digit);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
int read_key_analog()
{
    int key=0,mag=0;
    key = bioskey(0);
    switch(tolower(key))
    {
        case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
mag=0;
break;
        case 13 :press(200,320,280,350,"Ok");
putimage(180,120,win,COPY_PUT);
mag=1;
break;
        case 'h' : press(380,320,460,350,"Help");
help_analog();
mag=0;
read_key_digital();
break;
        default : read_key_analog();
    }
    return(mag);
}
int read_key_clear()
{
    int key=0,clr=0;
    key = bioskey(0);
    switch(tolower(key))
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case 27 : press(290,320,370,350,"Cancel");
putimage(180,120,win,COPY_PUT);
clr=1;
break;

        case 13 : press(200,320,280,350,"Ok");
putimage(180,120,win,COPY_PUT);
clr=0;
break;

        case 'h': press(380,320,465,350,"Help");
help_clear();
clr=1;
read_key_clear();
break;

        default : read_key_clear();
    }
    return(clr);
}

int read_kb()
{
    union {
    char c[2];
    int i;
} key;
    key.i = bioskey(0);
    switch(tolower (key.c[0]))
    {
    case 'u' : /* Auto */
        auto_run();
        break;
    case 'c' : /* Clear */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clear_all();
break;
case 'o' : /* OUT */
out();
break;
case 'a' : /* A/D */
analog();
break;
case 'd' : /* D/A */
digital();
break;
case 'e' : /* Shell */
press(15,340,70,369,"Shell");
os_shell();
break;
case 'l' : /* Load */
load();
break;
case 's' : /* Save */
save();
break;
case 'h' : /* Help */
help();
break;
case 'x' : /* Exit */
exit_itim();
break;
}
read_kb();
return(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

int auto_run()
(
    press(15,40,70,69,"Auto");
    windows(180,120,480,360,"AUTO RUN","");
    outtextxy(200,170,"Select voltage value ");
    botton(200,200,240,240,"1");
    botton(252,200,292,240,"2");
    botton(304,200,344,240,"3");
    botton(356,200,396,240,"4");
    botton(408,200,448,240,"5");
    botton(200,260,240,300,"6");
    botton(252,260,292,300,"7");
    botton(304,260,344,300,"8");
    botton(356,260,396,300,"9");
    botton(408,260,448,300,"10");
    botton(200,320,275,350,"");
    botton(373,320,448,350,"Cancel");
    botton(285,320,360,350,"Help");
    read_key_auto();
    putimage(125,95,dip6,COPY_PUT);
    putimage(125,290,dip7,COPY_PUT);
    delay(250);
    if(volt!=0)
    {
        auto_adj_volt(volt);
        free(win);
    }
    opp=1;
    read_kb();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return(opp);
}
int clear_all()
{
    int rip=0,clear=0,i;
    static float data_volt[48];
    press(15,70,70,99,"Clear");
    windows(180,120,480,360,"Clear","");
    outtextxy(200,190,"Press <Ok> to clear");
    outtextxy(200,210,"Press <Cancle> to abort");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,465,350,"Help");
    rip=read_key_clear();
    if(rip==0)
    {
        putimage(125,95,dip6,COPY_PUT);
        putimage(125,290,dip7,COPY_PUT);
        out_port_clear();
        for(i=0;i<48;i++) data_volt[i]=0;
        clear=1;
    }
    if(rip==1)
    {
        free(win);
        clear=0;
    }
    channal=130;
    volt=199;
    free(win);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return(clear);
}
int out()
{
    int group_data,i;
    press(15,100,70,129,"OUT");
    windows(180,120,480,360,"OUT PORT","");
    outtextxy(200,180,"Program is operating to sand data");
    outtextxy(200,210,"Press < Ok > to sand data");
    outtextxy(200,230,"Press < Cancel > to abort");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,460,350,"Help");
    out_data=read_key_out();
    if(out_data==1) /* out port data of auto functi
    {
        if(opp==1)
        {
            group_data = table_auto(volt);
            out_port(group_data);
            out_data=read_key_out();
            out_port_clear();
        }
        if(dac==1) out_port_ch();
    }
    for(i=0;i<48;i++) data_volt[i]=0;
    putimage(180,120,win,COPY_PUT);
    free(win);
    return(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void analog()
{
    press(15,130,70,159,"A/D");
    windows(180,120,480,360,"Analog To Digital","");
    /*outtextxy(200,180,"Program is operating to sand data");
    outtextxy(200,210,"Press < Ok > to sand data");
    outtextxy(200,230,"Press < Cancel > to abort");*/
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,460,350,"Help");
    read_key_analog();
}

int digital()
{
    int i;
    putimage(125,95,dip6,COPY_PUT);
    putimage(125,290,dip7,COPY_PUT);
    for(i=0;i<48;i++) data_volt[i]=0;
    press(15,160,70,189,"D/A");
    windows(180,120,480,360,"Digital to Analog","");
    outtextxy(200,170,"This function used to setup ");
    outtextxy(200,180,"voltage per channal");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,460,350,"Help");
    dac=read_key_digital();
    if(dac==1)
    get_volt();
    if(dac==0)
    free(win);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return(0);
}
void load()
{
    press(15,190,70,219,"Load");
    windows(180,120,480,360,"Load"," ");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,460,350,"Help");
    setfillstyle(1,0);
    bar(250,220,400,250);
    outtextxy(325,200,"Enter file name...");
    readstr(35,15,11,string);
    read_file();
    read_key_load();
    free(win);
}
void save()
{
    press(15,220,70,249,"Save");
    windows(180,120,480,360,"Save As","");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    botton(380,320,460,350,"Help");
    setfillstyle(1,0);
    bar(250,220,400,250);
    outtextxy(325,200,"Enter file name...");
    readstr(35,15,11,string);
    save_as();
    read_key_save();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    free(win);
}
void exit_itim()
{
    press(15,400,70,429,"Exit");
    windows(180,120,480,360,"EXIT","Do you want to exit");
    botton(200,320,280,350,"Ok");
    botton(290,320,370,350,"Cancel");
    read_key_exit();
    free(win);
}

void close_system()
{
    delay(20);
    closegraph();
    exit(0);
}

int out_port_clear()
{
    int  ch[48]={c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12
        ,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22
        ,c23,c24,c25,c26,c27,c28,c29,c30,c31,c32
        ,c33,c34,c35,c36,c37,c38,c39,c40,c41,c42
        ,c43,c44,c45,c46,c47,c48 };
    int  data[11]={D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11};
    int  data_out[48];
    int  i,k;
    outportb(port0,0x80);
    outportb(shipa,0x00);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do
{
    for(i=0;i<=47;i++)  outportb(shipb,ch[i]);
}
while(!kbhit());
return(0);
}

int out_port(int ob_data)
{
    int  ch[48]={c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12
        ,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22
        ,c23,c24,c25,c26,c27,c28,c29,c30,c31,c32
        ,c33,c34,c35,c36,c37,c38,c39,c40,c41,c42
        ,c43,c44,c45,c46,c47,c48 };
    int data[11]={D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11};
    int data_out[48];
    int i;
    outportb(port0,0x80);
    outportb(shipa,ob_data);
    do
    {
        for(i=0;i<=47;i++)  outportb(shipb,ch[i]);
    }
    while(!kbhit());
    return(0);
}

int out_port_ch()
{
    int  ch[48]={c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12
        ,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ,c23,c24,c25,c26,c27,c28,c29,c30,c31,c32
    ,c33,c34,c35,c36,c37,c38,c39,c40,c41,c42
    ,c43,c44,c45,c46,c47,c48 }];

int  c[48]={c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12
    ,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22
    ,c23,c24,c25,c26,c27,c28,c29,c30,c31,c32
    ,c33,c34,c35,c36,c37,c38,c39,c40,c41,c42
    ,c43,c44,c45,c46,c47,c48 }];

int  k,j,cv[47];
outportb(port0,0x80);
do
{
    for(j=0;j<48;j++)
    {
cv[j]=qqq(data_volt[j]);
outportb(shipa,cv[j]);      /* data */
delay(50);
outportb(shipb,ch[j]); /* chanal */
delay(50);
    }
}
while(!kbhit());
return(0);
}

int qqq(float data[])
{
    int i,c[47];
    for(i=0;i<48;i++)
        c[i]=con_table(data_volt[i]);
    return(c[47]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
int con_table(float data)
{
    int g_volt=0x00;
    switch(data)
    {
        case 1 : g_volt=0x18;
break;
        case 2 : g_volt=0x32;
break;
        case 3 : g_volt=0x4C;
break;
        case 4 : g_volt=0x68;
break;
        case 5 : g_volt=0x80;
break;
        case 6 : g_volt=0x9A;
break;
        case 7 : g_volt=0xB2;
break;
        case 8 : g_volt=0xCB;
break;
        case 9 : g_volt=0xE4;
break;
        case 10: g_volt=0xFF;
break;
    }
    return(g_volt);
}
int table_auto(int volt)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
int con_table(float data)
{
    int g_volt=0x00;
    switch(data)
    {
        case 1 : g_volt=0x18;
        break;
        case 2 : g_volt=0x32;
        break;
        case 3 : g_volt=0x4C;
        break;
        case 4 : g_volt=0x68;
        break;
        case 5 : g_volt=0x80;
        break;
        case 6 : g_volt=0x9A;
        break;
        case 7 : g_volt=0xB2;
        break;
        case 8 : g_volt=0xCB;
        break;
        case 9 : g_volt=0xE4;
        break;
        case 10: g_volt=0xFF;
        break;
    }
    return(g_volt);
}
int table_auto(int volt)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int group_volt=0x00;
    switch(volt)
    {
        case 189 : group_volt=0x18;
    break;
        case 179 : group_volt=0x32;
    break;
        case 169 : group_volt=0x4C;
    break;
        case 159 : group_volt=0x68;
    break;
        case 149 : group_volt=0x80;
    break;
        case 139 : group_volt=0x9A;
    break;
        case 129 : group_volt=0xB2;
    break;
        case 119 : group_volt=0xCB;
    break;
        case 109 : group_volt=0xE4;
    break;
        case 99  : group_volt=0xFF;
    break;
    }
    return(group_volt);
}

int Table_volt(float data)
{
    int volt,offset;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data = data*2;
switch(data)
{
case 0: volt = 199;
ob_volt=0x00;
break;
case 1: volt = 194;
break;
case 2: volt = 189;
ob_volt=0x18;
break;
case 3: volt = 184;
break;
case 4: volt = 179;
ob_volt=0x32;
break;
case 5: volt = 174;
break;
case 6: volt = 169;
ob_volt=0x4C;
break;
case 7: volt = 164;
break;
case 8: volt = 159;
ob_volt=0x68;
break;
case 9: volt = 154;
break;
case 10: volt = 149;
ob_volt=0x80;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
case 11:volt = 144;
break;
case 12:volt = 139;
ob_volt=0x9A;
break;
case 13:volt = 134;
break;
case 14:volt = 129;
ob_volt=0xB2;
break;
case 15:volt = 124;
break;
case 16:volt = 119;
ob_volt=0xCB;
break;
case 17:volt = 114;
break;
case 18:volt = 109;
ob_volt=0xE4;
break;
case 19:volt = 104;
break;
case 20:volt = 99;
ob_volt=0xFF;
break;
}

if(group==0)
offset = 0;
else offset = 195;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return (volt+offset);
}
int Table_channal(int data)
{
    int index,offset;;
    switch(data)
    {
        case 130: index = 0;
break;
        case 150: index = 1;
break;
        case 170: index = 2;
break;
        case 190: index = 3;
break;
        case 210: index = 4;
break;
        case 230: index = 5;
break;
        case 250: index = 6;
break;
        case 270: index = 7;
break;
        case 290: index = 8;
break;
        case 310: index = 9;
break;
        case 330: index = 10;
break;
        case 350: index = 11;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
break;
    case 370: index = 12;
break;
    case 390: index = 13;
break;
    case 410: index = 14;
break;
    case 430: index = 15;
break;
    case 450: index = 16;
break;
    case 470: index = 17;
break;
    case 490: index = 18;
break;
    case 510: index = 19;
break;
    case 530: index = 20;
break;
    case 550: index = 21;
break;
    case 570: index = 22;
break;
    case 590: index = 23;
break;
}

    if (group==0)
        offset = 0;
    else offset = 24;
    return (index+offset);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
int get_volt()
{
    int index;
    float temp,b=0.0;
    union {
        char c[2];
        int i;
    } key;
    do
    {
        if(bioskey(1))
        {
            key.i = bioskey(0);
            if(!key.c[0])
            {
                index = Table_channal(channal);
                if(group==0)
                {
                    volt = Table_volt(data_volt[24-((610-channal)/20)]);
                    data_volt[index]=10.0-(volt-99.0)/10.0;
                }
            }
            else
            {
                volt = Table_volt(data_volt[index/*24/-((610-channal)/20)*/1]);
                data_volt[index]=10.0-(volt-294.0)/10.0;
            }
        }
        switch(key.c[1])
        {
            case 72 :/*UP*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

volt-=5;
if(volt==94 || volt==289)
{
    volt+=5;
    beep();
}
step_up_adj_volt(volt,channel);
break;
case 80 : /*DOWN*/
    volt+=5;
    if(volt==204 || volt==399)
    {
        volt-=5;
        beep();
    }
    step_down_adj_volt(volt,channel);
    break;
case 75 : /*LEFT*/
    channel-=20;
    if(channel<130)
    {
        beep();
        channel+=20;
    }
    index = Table_channel(channel);
    if(group==0)
    {
        volt = Table_volt(data_volt[24-((610-channel)/20)]);
        data_volt[index]=10.0-(volt-99.0)/10.0;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
volt = Table_volt(data_volt[index/*24-((610-channal)/20)*/]);
data_volt[index]=10.0-(volt-294.0)/10.0;
}
break;
case 77 : /*RIGHT*/
channal+=20;
if(channal==610)
{
beep();
channal-=20;
}
index = Table_channal(channal);
if(group==0)
{
volt = Table_volt(data_volt[24-((610-channal)/20)]);
data_volt[index]=10.0-(volt-99.0)/10.0;
}
else
{
volt = Table_volt(data_volt[index/*24-((610-channal)/20)*/]);
data_volt[index]=10.0-(volt-294.0)/10.0;
}
break;
case 73 : /*PgUp*/
group = 0;
index = Table_channal(channal);
volt = Table_volt(data_volt[24-((610-channal)/20)]);
data_volt[index]=10.0-(volt-99.0)/10.0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;
case 81 : /*PgUd*/
    if(group++>1) group = 1;
        index = Table_channel(channel);
        volt = Table_volt(data_volt[index/*24-((610-channel)/20)*/]);
        data_volt[index]=10.0-(volt-294.0)/10.0;
        break;
}
index = Table_channel(channel);
    if(group==0)
        data_volt[index]=10.0-(volt-99.0)/10.0;
else
    data_volt[index]=10.0-(volt-294.0)/10.0;
}
}
}
while(key.c[0] != 27 );
press(15,160,70,189,"D/A");
flag_da=1;
return(flag_da);
}
void init_button()
{
    int i,j;
    for(i=130;i<610;i+=20)
    {
        dip_button(i-5,199,LIGHTGRAY);
        dip_button(i-5,394,LIGHTGRAY);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

int auto_adj_volt(int volt)
{
    int i,j;
    for(i=130,j=0;i<610,j<24;i+=20,j++)
    {
        putimage(i-5,199,dip1[j],COPY_PUT);
        putimage(i-5,394,dip2[j],COPY_PUT);
        dip_button(i-5,volt,LIGHTGRAY);
        dip_button(i-5,volt+195,LIGHTGRAY);
    }
    return(0);
}

int step_up_adj_volt(int step_volt,int channel)
{
    dip_button(channel-5,step_volt,LIGHTGRAY);
    putimage(channel-5,step_volt+5,dip1[1],COPY_PUT);
    if(step_volt==99) beep();
    return(0);
}

int step_down_adj_volt(int step_volt,int channel)
{
    putimage(channel-5,step_volt-5,dip1[0],COPY_PUT);
    dip_button(channel-5,step_volt,LIGHTGRAY);
    if(step_volt==199) beep();
    return(0);
}

int save_button()
{
    int i,j;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=130,j=0;i<610,j<24;i+=20,j++)
{
    dip1[j]=save_image2(i-5,199,i+5,204);
    dip2[j]=save_image2(i-5,394,i+5,399);
}
return(0);
}

int score()
{
    botton(15,40,70,69,"Auto");
    botton(15,70,70,99,"Clear");
    botton(15,100,70,129,"OUT");
    botton(15,130,70,159,"A/D");
    botton(15,160,70,189,"D/A");
    botton(15,190,70,219,"Load");
    botton(15,220,70,249,"Save");
    botton(15,250,70,279,"");
    botton(15,280,70,309,"");
    botton(15,310,70,339,"");
    botton(15,340,70,369,"Shell");
    botton(15,370,70,399,"Help");
    botton(15,400,70,429,"Exit");
    return(0);
}

show_screen()
{
    frame_3d(80,40,620,63,"CHANNAL 1 - 24",WHITE,LIGHTBLUE);
    frame_3d(80,235,620,258,"CHANNAL 25 - 48",WHITE,LIGHTBLUE);
    frame_3d_inv(80,65,620,233,"",LIGHTGRAY,LIGHTGRAY);
    frame_3d_inv(80,260,620,433,"",LIGHTGRAY,LIGHTGRAY);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    botton(15,340,70,369,"");
    scal();
    return(0);
}
void main()
{
    init_graph();
    out_frame();
    demo(180,120,480,360,"","");
    out_port_clear();
    delay(50);
    score();
    delay(200);
    show_screen();
    save_buttton();
    init_buttton();
    init_save();
    read_kb(volt);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MOTOROLA

MC1408
MC1508

Specifications and Applications Information

EIGHT-BIT MULTIPLYING DIGITAL-TO-ANALOG CONVERTER

... designed for use where the output current is a linear product of an eight-bit digital word and an analog input voltage.

- Eight-Bit Accuracy Available in Both Temperature Ranges
Relative Accuracy: $\pm 0.1\%$ Error maximum
(MC1408L8, MC1408P8, MC1508L8)
- Seven and Six-Bit Accuracy Available with MC1408 Designated by 7 or 6 Suffix after Package Suffix
- Fast Settling Time - 300 ns typical
- Noninverting Digital Inputs are MTTL and CMOS Compatible
- Output Voltage Swing - +0.4 V to -5.0 V
- High-Speed Multiplying Input
Slew Rate 4.0 mA/ μ s
- Standard Supply Voltages: +5.0 V and -5.0 V to -15 V

EIGHT-BIT MULTIPLYING DIGITAL-TO-ANALOG CONVERTER

SILICON MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620-02



P SUFFIX
PLASTIC PACKAGE
CASE 648-05

D SUFFIX
PLASTIC PACKAGE
CASE 751B-01
SO-16*



*Surface Mount pin-out is different than standard package pin-out for this device. Please refer to Chapter 7 for details.

FIGURE 1 - D-to-A TRANSFER CHARACTERISTICS

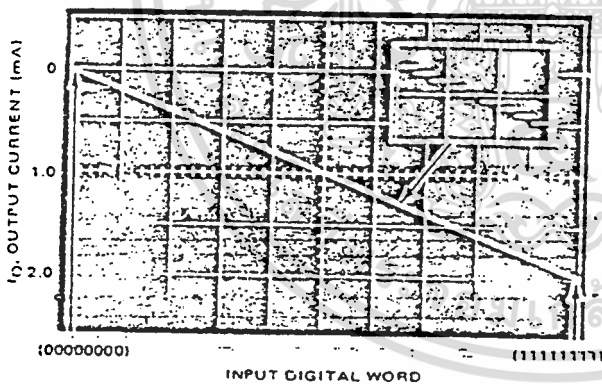
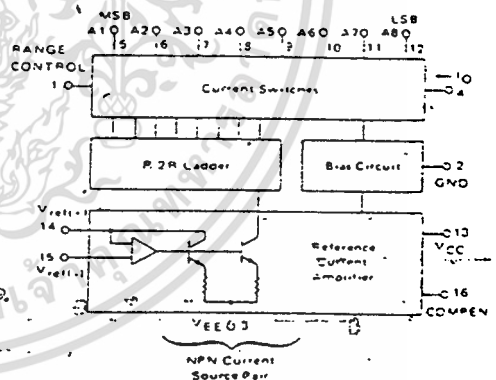


FIGURE 2 - BLOCK DIAGRAM



TYPICAL APPLICATIONS

- Tracking A-to-D Converters
- Successive Approximation A-to-D Converters
- 2 1/2 Digit Panel Meters and DVM's
- Waveform Synthesis
- Sample and Hold
- Peak Detector
- Programmable Gain and Attenuation
- CRT Character Generation
- Audio Digitizing and Decoding
- Programmable Power Supplies
- Analog-Digital Multiplication
- Digital-Digital Multiplication
- Analog-Digital Division
- Digital Addition and Subtraction
- Speech Compression and Expansion
- Stepping Motor Drive

MOTOROLA LINEAR/INTERFACE DEVICES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIMUM RATINGS (T_A = +25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V _{CC} V _{EE}	+5.5 -16.5	V _{dC}
Digital Input Voltage	V ₅ thru V ₁₂	0 to +5.5	V _{dC}
Applied Output Voltage	V _O	+0.5, -5.2	V _{dC}
Reference Current	I ₁₄	5.0	mA
Reference Amplifier Inputs	V ₁₄ , V ₁₅	V _{CC} , V _{EE}	V _{dC}
Operating Temperature Range	T _A	-55 to +125 0 to +75	°C
Storage Temperature Range	T _{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.0 V_{dC}, V_{EE} = -15 V_{dC}, $\frac{V_{ref}}{R_{14}} = 2.0$ mA, MC1508L8: T_A = -55°C to +125°C, MC1408L Series: T_A = 0 to +75°C unless otherwise noted. All digital inputs at high logic level.)

Characteristic	Figure	Symbol	Min	Typ	Max	Unit
Relative Accuracy (Error relative to full scale I _O) MC1508L8, MC1408L8, MC1408P8 MC1408P7, MC1408L7, See Note 1 MC1408P6, MC1408L6, See Note 1	4	E _r	-	-	±0.19 ±0.39 ±0.78	%
Settling Time to within ±1/2 LSB (includes t _{PLH}) (T _A = +25°C) See Note 2	5	t _S	-	300	-	ns
Propagation Delay Time T _A = +25°C	5	t _{PLH} , t _{PHL}	-	30	100	ns
Output Full Scale Current Drift		TC _{IO}	-	-20	-	PPM/°C
Digital Input Logic Levels (MSB) High Level, Logic "1" Low Level, Logic "0"	3	V _{IH} V _{IL}	2.0	-	0.8	V _{dC}
Digital Input Current (MSB) High Level, V _{IH} = 5.0 V Low Level, V _{IL} = 0.8 V	3	I _{IH} I _{IL}	-	0 -0.4	0.04 -0.8	mA
Reference Input Bias Current (Pin 15)	3	I ₁₅	-	-1.0	-5.0	μA
Output Current Range V _{EE} = -5.0 V V _{EE} = -15 V, T _A = 25°C	3	I _{OR}	0 0	2.0 2.0	2.1 4.2	mA
Output Current V _{ref} = 2.000 V, R ₁₄ = 1000 Ω	3	I _O	1.9	1.99	2.1	mA
Output Current (All bits low)	3	I _{O(min)}	-	0	4.0	μA
Output Voltage Compliance (E _r ≤ 0.19% at T _A = +25°C) Pin 1 grounded Pin 1 open, V _{EE} below -10 V	3	V _O	-	-	-0.55, +0.4 -5.0, -0.4	V _{dC}
Reference Current Slew Rate	6	SR I _{ref}	-	4.0	-	mA/μs
Output Current Power Supply Sensitivity		PSRR(-)	-	0.5	2.7	μA/V
Power Supply Current (All bits low)	3	I _{CC} I _{EE}	-	+13.5 -7.5	-22 -13.5	mA
Power Supply Voltage Range (T _A = +25°C)	3	V _{CCR} V _{VEER}	+4.5 -4.5	+5.0 -15	-5.5 -16.5	V _{dC}
Power Dissipation All bits low V _{EE} = -5.0 V _{dC} V _{EE} = -15 V _{dC} All bits high V _{EE} = -5.0 V _{dC} V _{EE} = -15 V _{dC}	3	P _D	-	105 190 90 160	170 305 -	mW

Note 1 All current switches are tested to guarantee at least 50% of rated output current.
Note 2 All bits switched.

MOTOROLA LINEAR/INTERFACE DEVICES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DAC0808, DAC0807, DAC0806



A to D, D to A

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

General Description

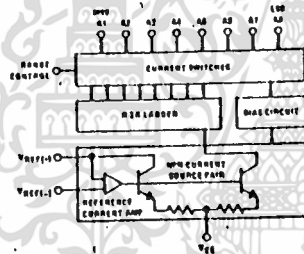
The DAC0808 series is an 8 bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of 255 $I_{REF}/256$. Relative accuracies of better than $\pm 0.19\%$ assure 8 bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8 bit zero accuracy for $I_{REF} \geq 2$ mA. 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 MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

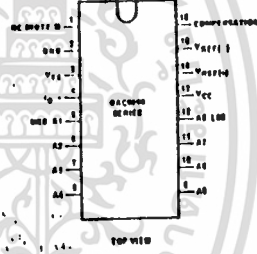
Features

- Relative accuracy: $\pm 0.19\%$ error max (DAC0808)
- Full scale current match: ± 1 LSB typ
- 7 and 8 bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and 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



Dual-In-Line Package



Typical Application

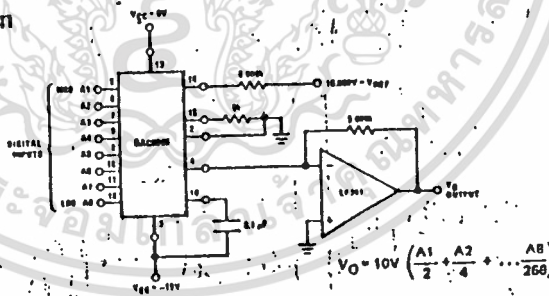


FIGURE 1. +10V Output Digital to Analog Converter

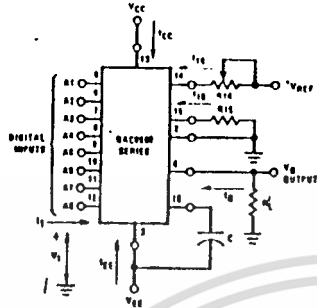
Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS*					
		D PACKAGE (D16C)		J PACKAGE (J16A)		N PACKAGE (N1)	
8 bit	$-55^{\circ}C \leq T_A \leq +125^{\circ}C$	DAC0808LD	MC1508LB	DAC0808LCJ	MC1408LB	DAC0808LCN	MC
8 bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$			DAC0807LCJ	MC1408L7	DAC0807LCN	MC
7 bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$			DAC0806LCJ	MC1408L6	DAC0806LCN	MC
8 bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$						

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits



V_I and I_1 apply to Inputs A1-A8.

The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right)$$

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

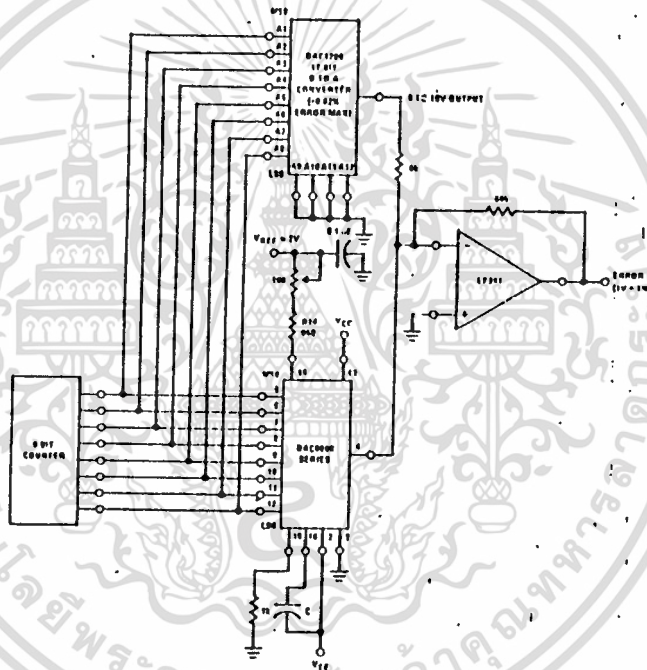


FIGURE 4. Relative Accuracy Test Circuit

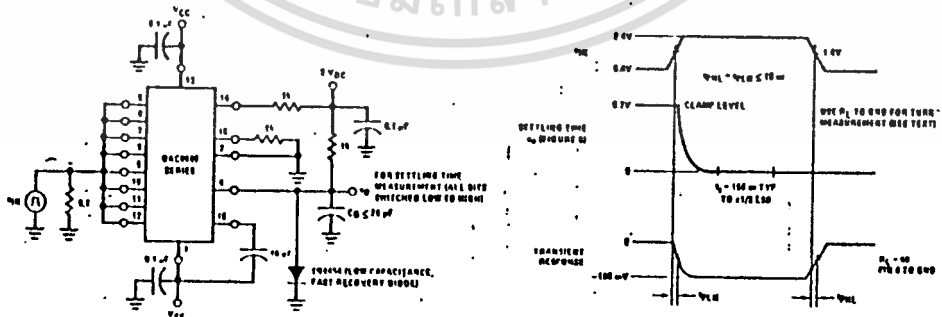
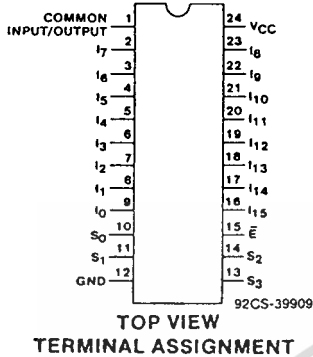


FIGURE 5. Transient Response and Settling Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



16-Channel Analog Multiplexer/Demultiplexer

Type Features:

- Wide analog input voltage range:
- Low "on" resistance:
70 Ω typ ($V_{CC} = 4.5V$)
60 Ω typ ($V_{CC} = 6V$)
- Fast switching and propagation speeds
- "Break-before-make" switching:
(6 ns typ @ 4.5V)

The RCA-CD54/74HC/HCT4067 are digitally controlled analog switches which utilize silicon-gate CMOS technology to achieve operating speeds similar to LSTTL with the low power consumption of standard CMOS integrated circuits.

These analog multiplexers/demultiplexers control analog voltages that may vary across the voltage supply range. They are bidirectional switches thus allowing any analog input to be used as an output and visa-versa. The switches have low "on" resistance and low "off" leakages. In addition, these devices have an enable control which when high will disable all switches to their "off" state.

The CD54HC/HCT4067 are supplied in 24-lead ceramic dual-in-line packages (F suffix). The CD74HC/HCT4067 are supplied in 24-lead dual-in-line plastic packages (E suffix).

Row spacing in the E and F suffix packages is 600 mils, not 300 mils as in most other packages. These CD74HC/HCT4067 devices are also supplied in 24-lead dual-in-line surface mount plastic packages (M suffix). All types are also available in chip form (H suffix).

Family Features:

- Fanout (Over Temperature Range):
Standard Outputs - 10 LSTTL Loads
Bus Driver Outputs - 15 LSTTL Loads
- Wide Operating Temperature Range:
CD74HC/HCT: -40 to +85°C
- Balanced Propagation Delay and Transition Times
- Significant Power Reduction Compared to LSTTL Logic ICs
- Alternate Source is Philips/Sigmetics
- CD54/74HC Types:
2 to 6 V Operation
High Noise Immunity:
 $N_{IL} = 30\%$, $N_{IH} = 30\%$ of V_{CC} @ $V_{CC} = 5V$
- CD54/74HCT Types:
4.5 to 5.5 V Operation
Direct LSTTL Input Logic Compatibility
 $V_{IL} = 0.8 V$ Max., $V_{IH} = 2 V$ Min.
CMOS Input Compatibility
 $I_i \leq 1 \mu A$ @ V_{OL} , V_{OH}

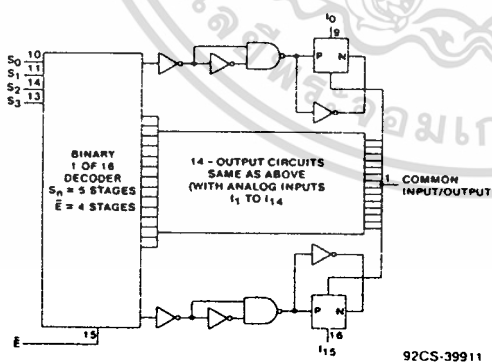


Fig. 1 - Functional diagram.

TRUTH TABLE

S0	S1	S2	S3	E	Selected Channel
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

1 = High Level
0 = Low Level
X = Don't Care

Trademark(s)® Registered
Marca(s) Registrada(s)
Printed in USA/1-86

Information furnished by RCA is believed to be accurate and reliable. However, no responsibility is assumed by RCA for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of RCA.

This document contains information on a new product. Specific and information contained herein are subject to change without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STATIC ELECTRICAL CHARACTERISTICS

CHARACTERISTIC	CD74HC/CD54HC4067										CD74HCT/CD54HCT4067										UNIT
	TEST CONDITIONS			74HC/54HC TYPES			74HC TYPE		54HC TYPE		TEST CONDITIONS			74HCT/54HCT TYPES			74HCT TYPE		54HCT TYPE		
	LOGIC V_i V	SWITCH V_{is} V	V_{cc} V	+25°C			-40/ +85°C		-55/ +125°C		LOGIC V_i V	SWITCH V_{is} V	+25°C			-40/ +85°C		-55/ +125°C			
				Min	Typ	Max	Min	Max	Min	Max			Min	Typ	Max	Min	Max	Min	Max		
High-Level Input Voltage V_{ih}			2	1.5	—	—	1.5	—	1.5	—			2	—	—	2	—	2	—		
			4.5	3.15	—	—	3.15	—	3.15	—											
			6	4.2	—	—	4.2	—	4.2	—											
Low-Level Input Voltage V_{il}			2	—	—	0.5	—	0.5	—	0.5					0.8	—	0.8	—	0.8		
			4.5	—	—	1.35	—	1.35	—	1.35											
			6	—	—	1.8	—	1.8	—	1.8											
Maximum "On" Resistance R_{on} $I_i = 1\text{mA}$	V_{cc} or Gnd	V_{cc} or Gnd	4.5	—	70	160	—	200	—	240	V_{cc} or Gnd	V_{cc} or Gnd	—	70	160	—	200	—	240		
			6	—	60	140	—	175	—	210											
	V_{cc} to Gnd	V_{cc} to Gnd	4.5	—	90	180	—	225	—	270	V_{cc} to Gnd	V_{cc} to Gnd	—	90	180	—	225	—	270		
			6	—	80	160	—	200	—	240											
Maximum "On" resistance between any two switches ΔR_{on}			—	—	10	—	—	—	—	—			—	—	10	—	—	—	—		
			—	—	—	8.5	—	—	—	—			—	—	—	—	—	—	—		
Switch "Off" Leakage Current I_{LZ} 16 Channels	$\bar{E} = V_i$	V_{cc} or Gnd	6	—	—	± 0.8	—	± 0.8	—	± 0.8	$\bar{E} = V_{cc}$	V_{cc} or Gnd	—	—	± 0.8	—	± 0.8	—	± 0.8		
Logic Input Leakage Current I_i	V_{cc} or Gnd	—	6	—	—	± 0.1	—	± 0.1	—	± 0.1			—	—	± 0.1	—	± 0.1	—	± 0.1		
Quiescent Device Current I_{cc} $I_i = 0\text{mA}$	V_{cc} or Gnd	—	6	—	—	8	—	80	—	160	V_{cc} or Gnd	—	—	—	8	—	80	—	160		
Additional Quiescent Device Current per input pin ΔI_{cc} 1 unit load	—	—	—	—	—	—	—	—	—	—	$V_{cc} - 2.1$	—	—	100	360	—	450	—	490		

*For dual-supply systems theoretical worst case ($V_i = 2.4\text{V}$, $V_{cc} = 5.5\text{V}$) specification is 1.8 mA
 **Any Voltage Between V_i & Gnd

HCT INPUT LOADING TABLE

INPUT	UNIT LOADS*
$S_0 - S_3$	0.5
\bar{E}	0.3

*Unit Load is ΔI_{cc} limit specified in Static Characteristic Chart, e.g. 360 μA max @ 25°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SWITCHING CHARACTERISTICS (V_{CC} = 5 V, T_A = 25°C, Input t_r, t_f = 6 ns)

CHARACTERISTIC	CL (pF)	TYPICAL		UNITS
		HC	HCT	
Propagation Delay Time: Switch In to Switch Out	15	4	4	ns
Switch Turn Off Ē to Out		15	23	
Sn to Out	15		21	
Switch Turn On Ē to Out		15	23	
Sn to Out	15		25	
Power Dissipation Capacitance*		C _{PD}	93	

C_{PD} is used to determine the dynamic power consumption, per package.
 $P_D = C_{PD} V_{CC}^2 f_i + \sum (C_L + C_S) V_{CC}^2 f_o$ where: f_i = input frequency f_o = output frequency
 C_L = load capacitance C_S = switch capacitance
 V_{CC} = supply voltage

SWITCHING CHARACTERISTICS (C_L = 50 pF, Input t_r, t_f = 6 ns)

CHARACTERISTIC	V _{CC} V	25°C				-40°C to +85°C				-55°C to +125°C				UNITS
		HC		HCT		74HC		74HCT		54HC		54HCT		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Propagation Delay Time Switch In to Out	t _{PLH}	2	60	—	—	—	75	—	—	—	90	—	—	ns
	t _{PHL}	4.5	12	—	12	—	15	—	15	—	18	—	18	
		6	10	—	—	—	13	—	—	—	15	—	—	
Switch Turn-On Ē to Out	t _{PZL}	2	275	—	—	—	345	—	—	—	415	—	—	
	t _{PZH}	4.5	55	—	60	—	69	—	75	—	83	—	90	
		6	47	—	—	—	59	—	—	—	71	—	—	
Sn to Out	t _{PZH}	2	300	—	—	—	375	—	—	—	450	—	—	
		4.5	60	—	60	—	75	—	75	—	90	—	90	
		6	51	—	—	—	64	—	—	—	76	—	—	
Switch Turn-Off Ē to Out	t _{PLZ}	2	275	—	—	—	345	—	—	—	415	—	—	
	t _{PHZ}	4.5	55	—	55	—	69	—	69	—	83	—	83	
		6	47	—	—	—	59	—	—	—	71	—	—	
Sn to Out	t _{PLZ}	2	250	—	—	—	315	—	—	—	375	—	—	
	t _{PHZ}	4.5	50	—	50	—	63	—	63	—	75	—	75	
		6	43	—	—	—	54	—	—	—	64	—	—	
Input (Control) Capacitance	C _I	—	10	—	10	—	10	—	10	—	10	—	10	pF

ANALOG CHANNEL CHARACTERISTICS — Typical Values at T_A = 25°C

CHARACTERISTICS	TEST CONDITION	V _{CC} V	HC/HCT	UNITS
Switch Frequency Response at -3 dB (Fig. 12)	Fig. 3 Notes 1 & 2	4.5	89	MHz
Sine Wave Distortion	Fig. 4	4.5	0.051	%
Feedthrough Noise Ē to Switch S to Switch	Fig. 5 Notes 2 & 3	4.5	TBE TBE	mV
Switch "OFF" Signal Feedthrough (Fig. 13)	Fig. 6	4.5	-75	dB
Switch Input Capacitance	C _S	—	5	pF
Common Capacitance	C _{COM}	—	50	

- NOTES: 1. Adjust input level for 0 dBm at output + 10 kHz.
 2. V_{IS} is centered at V_{CC}/2
 3. Adjust input for 0 dBm at V_I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

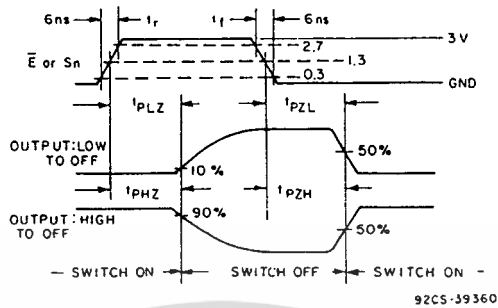


Fig. 9 - Switch turn-on and turn-off propagation delay times waveforms for HCT Types.

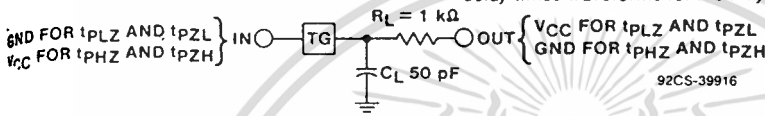


Fig 10 - Switch on/off propagation delay time test circuit.

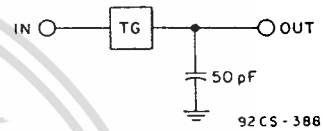


Fig. 11 - Switch In to Switch Out Propagation delay time test circuit.

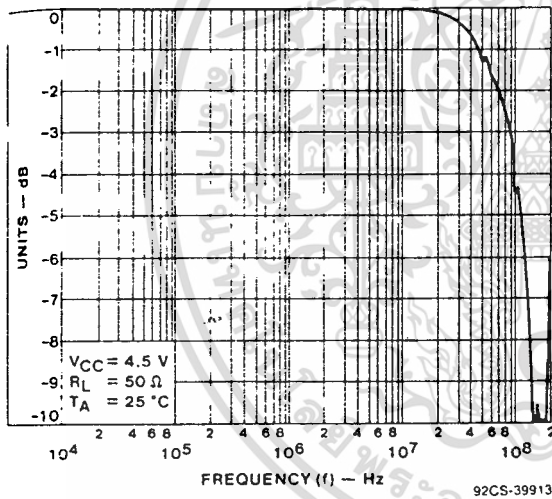


Fig 12 - Typical switch frequency response.

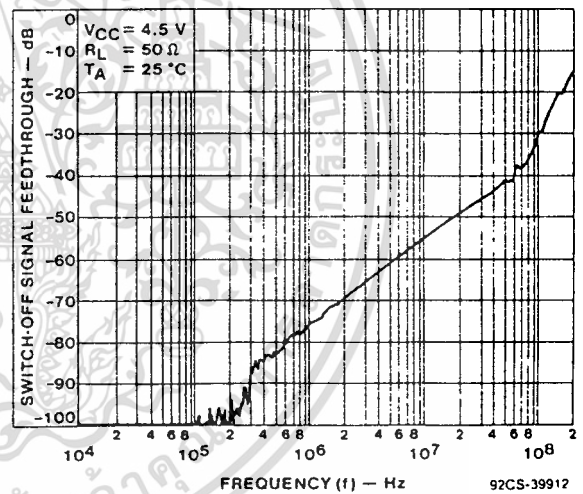


Fig 13 - Typical switch-off signal feed through vs. frequency.

OPERATING AND HANDLING CONSIDERATIONS

1. Handling

All inputs and outputs of RCA CMOS/QMOS devices have a network for electrostatic protection during handling. Recommended handling practices for CMOS/CMOS devices are similar to those described in ICAN-6525, "Guide to Better Handling and Operation of CMOS Integrated Circuits."

2. Operating

Operating Voltage

During operation near the maximum supply voltage limit, care should be taken to avoid or suppress power supply turn-on and turn-off transients, power supply ripple or ground noise, any of these conditions must not cause Vcc-Gnd to exceed the absolute maximum rating.

Input Signals

To prevent damage to the input protection circuit, input signals should never be greater than Vcc nor less than Gnd. Input currents must not exceed 20 mA even when the power supply is off.

Unused Inputs

A connection must be provided at every input terminal. All unused input terminals must be connected to either Vcc or Gnd, whichever is appropriate.

Output Short Circuits

Shorting of outputs to Vcc or Gnd may damage CMOS/QMOS devices by exceeding the maximum device dissipation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADC0808, ADC0809 8-Bit μ P Compatible A/D Converters With 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

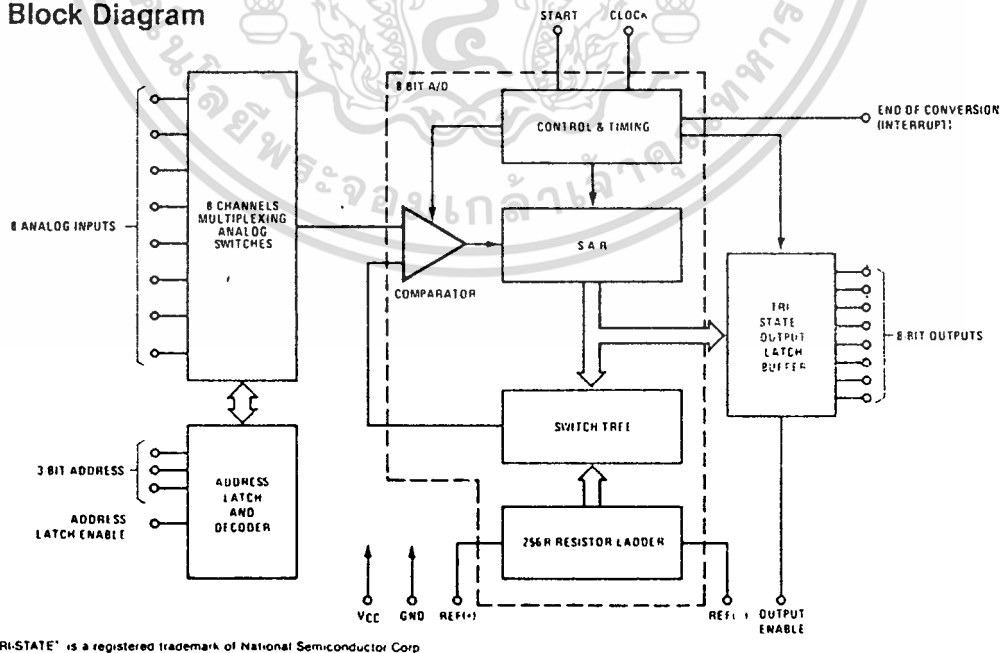
The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE[®] outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

Features

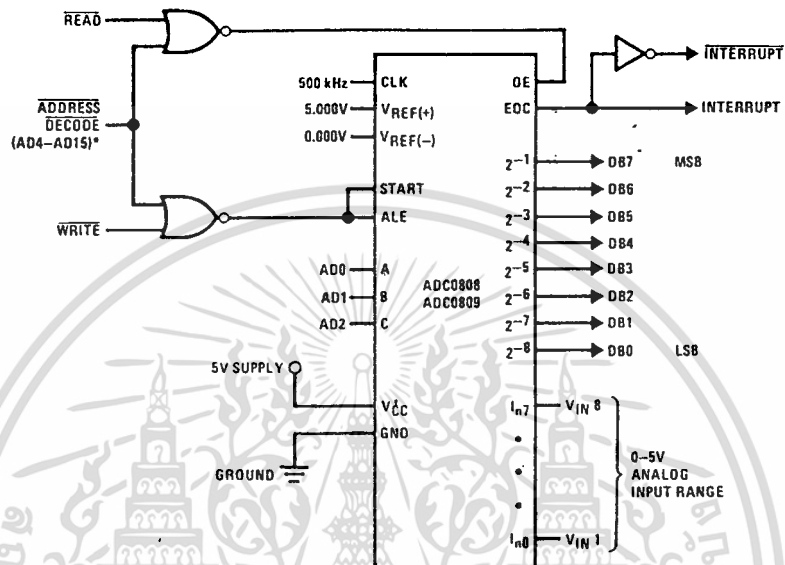
- Resolution — 8-bits
- Total unadjusted error — $\pm 1/2$ LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μ s
- Single supply — $5 V_{DC}$
- Operates ratiometrically or with $5 V_{DC}$ or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T²L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range — -40°C to $+85^{\circ}\text{C}$ or -55°C to $+125^{\circ}\text{C}$
- Low power consumption — 15 mW
- Latched TRI-STATE[®] output

Block Diagram



TRI-STATE[®] is a registered trademark of National Semiconductor Corp.

Typical Application



* Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

MICROPROCESSOR INTERFACE TABLE

PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	$\overline{\text{MEMR}}$	$\overline{\text{MEMW}}$	INTR (Thru RST Circuit)
8085	$\overline{\text{RD}}$	$\overline{\text{WR}}$	INTR (Thru RST Circuit)
Z-80	$\overline{\text{RD}}$	$\overline{\text{WR}}$	INT (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA \cdot 2-R/W	VMA \cdot 2-R/W	IROA or IRQB (Thru PIA)

Ordering Information

TEMPERATURE RANGE		-40°C to +85°C	-55°C to +125°C
Error	$\pm 1/2$ Bit Unadjusted	ADC0808CCN	ADC0808CCJ
	± 1 Bit Unadjusted	ADC0809CCN	
Package Outline		N28A Molded DIP	J28A Hermetic DIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CJ $4.5V \leq V_{CC} \leq 5.5V$, $-55^{\circ}C \leq T_A \leq +125^{\circ}C$ unless otherwise noted
 ADC0808CCJ, ADC0808CCN, and ADC0809CCN $4.75 \leq V_{CC} \leq 5.25V$, $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ unless otherwise noted

Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)					
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -360 \mu A$	$V_{CC}-0.4$	—	V
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = 1.6 \text{ mA}$		0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	$I_O = 1.2 \text{ mA}$		0.45	V
I_{OUT}	TRI-STATE® Output Current	$V_O = 5V$ $V_O = 0$		3	μA μA

Electrical Characteristics

Timing Specifications: $V_{CC} = V_{REF(+)} = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20 \text{ ns}$ and $T_A = 25^{\circ}C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t_s	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t_H	Minimum Address Hold Time	(Figure 5)		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_S = 0\Omega$ (Figure 5)		1	2.5	μs
t_{H1}, t_{H0}	OE Control to Q Logic State	$C_L = 50 \text{ pF}$, $R_L = 10k$ (Figure 8)		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	$C_L = 10 \text{ pF}$, $R_L = 10k$ (Figure 8)		125	250	ns
t_c	Conversion Time	$f_c = 640 \text{ kHz}$, (Figure 5) (Note 7)	90	100	116	μs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(Figure 5)	0		$8 + 2 \mu s$	Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE® Output Capacitance	At TRI-STATE® Outputs. (Note 12)		10	15	pF

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of $7 V_{DC}$

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute $0 V_{DC}$ to $5 V_{DC}$ input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See Figure 3. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See Figure 13.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure 6). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Pin	-0.3V to ($V_{CC} + 0.3V$)
Except Control Inputs	
Voltage at Control Inputs	-0.3V to +15V
(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A = 25^\circ\text{C}$	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1)	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC0808CJ	$-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$
ADC0808CCJ, ADC0808CCN, ADC0809CCN	$-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$
Range of V_{CC} (Note 1)	$4.5 V_{DC} \text{ to } 6.0 V_{DC}$

Electrical Characteristics

Converter Specifications: $V_{CC} = 5 V_{DC} = V_{REF(+)}$, $V_{REF(-)} = \text{GND}$, $T_{MIN} \leq T_A \leq T_{MAX}$ and $f_{CLK} = 640 \text{ kHz}$ unless otherwise stated.

Parameter	Conditions	Min	Typ	Max	Units
ADC0808					
Total Unadjusted Error (Note 5)	25°C $T_{MIN} \text{ to } T_{MAX}$			$\pm 1/2$ $\pm 3/4$	LSB LSB
ADC0809					
Total Unadjusted Error (Note 5)	$0^\circ\text{C} \text{ to } 70^\circ\text{C}$ $T_{MIN} \text{ to } T_{MAX}$			± 1 $\pm 1 1/4$	LSB LSB
Input Resistance	From Ref (+) to Ref (-)	1.0	2.5		k Ω
Analog Input Voltage Range	(Note 4) V (+) or V (-)	GND-0.10		$V_{CC} + 0.10$	V_{DC}
$V_{REF(+)}$ Voltage, Top of Ladder	Measured at Ref (+)		V_{CC}	$V_{CC} + 0.1$	V
$\frac{V_{REF(+)} + V_{REF(-)}}{2}$ Voltage, Center of Ladder		$V_{CC}/2 - 0.1$	$V_{CC}/2$	$V_{CC}/2 + 0.1$	V
$V_{REF(-)}$ Voltage, Bottom of Ladder	Measured at Ref (-)	-0.1	0		V
Comparator Input Current	$f_c = 640 \text{ kHz}$, (Note 6)	-2	± 0.5	2	μA

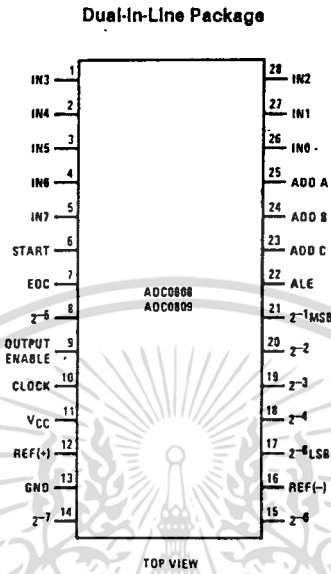
Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CJ $4.5V \leq V_{CC} \leq 5.5V$, $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ unless otherwise noted
ADC0808CCJ, ADC0808CCN, and ADC0809CCN $4.75 \leq V_{CC} \leq 5.25V$, $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ unless otherwise noted

Parameter	Conditions	Min	Typ	Max	Units
ANALOG MULTIPLEXER					
$I_{OFF(+)}$	OFF Channel Leakage Current $V_{CC} = 5V$, $V_{IN} = 5V$, $T_A = 25^\circ\text{C}$ $T_{MIN} \text{ to } T_{MAX}$		10	200 1.0	nA μA
$I_{OFF(-)}$	OFF Channel Leakage Current $V_{CC} = 5V$, $V_{IN} = 0$, $T_A = 25^\circ\text{C}$ $T_{MIN} \text{ to } T_{MAX}$	-200 -1.0	-10		nA μA
CONTROL INPUTS					
$V_{IN(1)}$	Logical "1" Input Voltage		$V_{CC} - 1.5$		V
$V_{IN(0)}$	Logical "0" Input Voltage			1.5	V
$I_{IN(1)}$	Logical "1" Input Current (The Control Inputs)	$V_{IN} = 15V$		1.0	μA
$I_{IN(0)}$	Logical "0" Input Current (The Control Inputs)	$V_{IN} = 0$	-1.0		μA
I_{CC}	Supply Current	$f_{CLK} = 640 \text{ kHz}$		0.3 3.0	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagram



Timing Diagram

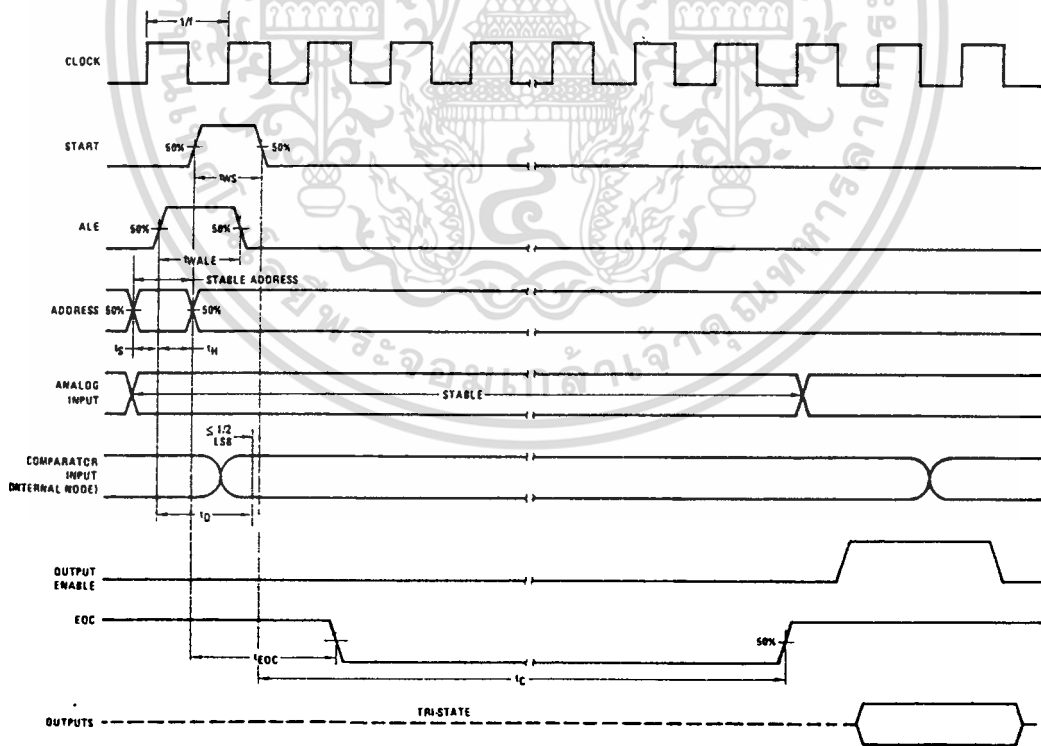
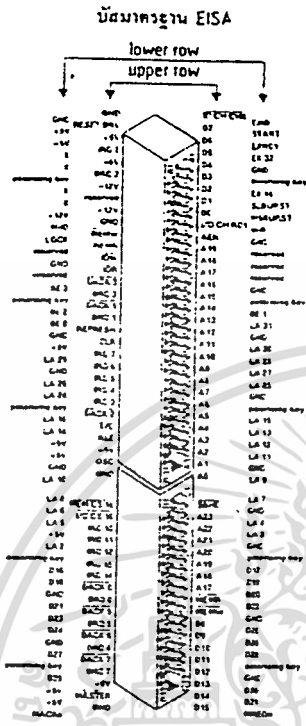
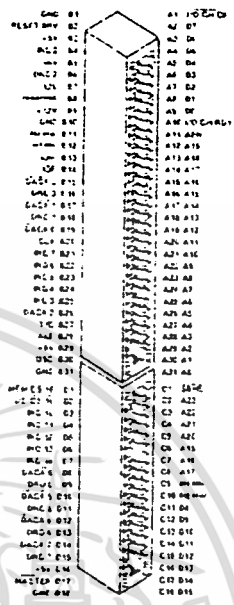


FIGURE 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผลิตภัณฑ์ระบบมาตรฐาน IBM



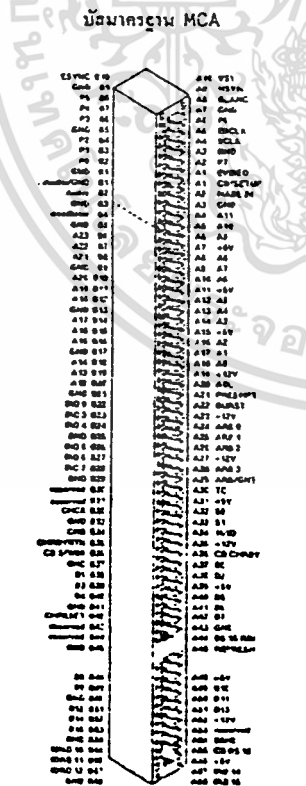
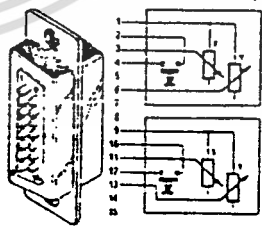
คอนเน็คเตอร์สำหรับบอร์ดรุ่น PS2 (6 ขา)



คอนเน็คเตอร์บอร์ดแบบ DIN (6 ขา)



คอนเน็คเตอร์สำหรับบอร์ดแบบ D (15 ขา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้