สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วงจรสาธิตการแปลงสัญญาณอนาล็อกเป็นดิจิตอลและดิจิตอลเป็นอนาล็อก

A/D & D/A DEMONSTRATION

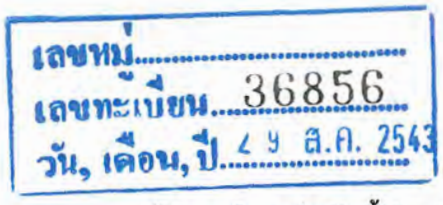
โดย

นางสาวชุตินธร สุกใส

นายบุญเสริม หงส์ทิพยฉัตร

อาจารย์ที่ปรึกษา

อาจารย์สว่าง เสิศธิรสุนทร



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง วงจรสาธิตการแปลงสัญญาณอนาล็อกเป็นดิจิตอลและดิจิตอลเป็นอนาล็อก

ผู้จัดทำ 1.นางสาว ชุตินธร สุกใส รหัส 39014135

2.นาย บุญเสริม หงส์ทิพยฉัตร รหัส 39014285

..... อาจารย์ที่ปรึกษา

(อาจารย์ สว่าง เลิศถิรสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสาธิตการแปลงสัญญาณอนาล็อกเป็นดิจิตอลและดิจิตอลเป็นอนาล็อก

นางสาว ชุตินธร สุกใส
นาย บุญเสริม หงส์ทิพยฉัตร

อาจารย์ที่ปรึกษา อ.สว่าง เลิศฤิรสุนทร
ปีการศึกษา 2542

บทคัดย่อ

ในปัจจุบันนี้อุปกรณ์ต่าง ๆ ภายในโรงงานอุตสาหกรรมที่ใช้สำหรับตรวจวัดปริมาณและควบคุมการทำงานของระบบต่าง ๆ ได้ถูกออกแบบมาให้สามารถใช้คอมพิวเตอร์เป็นตัวควบคุม เก็บข้อมูลและทำการประมวลผล แต่เนื่องจากคอมพิวเตอร์เป็นอุปกรณ์ที่รับส่งข้อมูลที่อยู่ในรูปแบบดิจิตอล ขณะที่อุปกรณ์ภายนอกทำงานและรับสัญญาณควบคุมในรูปแบบอนาล็อก ดังนั้นจึงจำเป็นจะต้องเชื่อมต่อคอมพิวเตอร์เข้ากับอุปกรณ์ภายนอกโดยการใช้ตัวแปลงสัญญาณ A/D และ D/A

ซึ่งในปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อเป็นสื่อการสอนให้เกิดความเข้าใจมากขึ้นในหลักการทำงานของ A/D และ D/A โดยได้ทำการออกแบบและทำการจัดสร้างการ์ดวงจรอินเตอร์เฟสกับคอมพิวเตอร์ โดยภายในประกอบด้วยส่วนวงจร A/D และ D/A ซึ่งจะทำการควบคุมและแสดงผลการทำงานของ A/D และ D/A โดยใช้โปรแกรมภาษา วิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A/D & D/A DEMONSTRATION

MISS. CHUTINTHORN SUGSAI
MR. BOONSERM HONGTIPPAYACHAT
MR. SAWANG LEARDTHIRASOONTHORN
ADVISOR

1999

Abstract

Today , almost industrial instruments used in control system that have been designed to be controlled , recorded and processed by computer. By the way , computer is processing in digital system but industrial instrument is working in analog system .They can work together by using A/D and D/A circuit.

This Thesis has been made to demonstrate how A/D and D/A can convert Analog to Digital and convert Digital to Analog . We have designed and made Interface Card with inside have A/D and D/A circuit. This Interface Card will be connected to computer and use Visual Basic language to control and show how A/D and D/A can convert Analog to Digital and convert Digital to Analog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมาและความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3ขอบเขตของโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการ	**4**
2.1 8255 พอร์ตข้อมูลแบบขนาน	4
2.2 การแปลงอนาล็อกเป็นดิจิตอล	14
2.3 การแปลงดิจิตอลเป็นอนาล็อก	19
2.4 การจัดแอดเดรสสำหรับพอร์ต I/O ใน IBM/PC	23
บทที่ 3 การคำนวณและการสร้าง	**34**
3.1 วงจรอินเตอร์เฟซกับเครื่องคอมพิวเตอร์	34
3.2 การเลือกไอซี แปลงสัญญาณอนาล็อกเป็นดิจิตอลมาใช้งานในวงจร	41
3.3 การเลือกไอซีแปลงสัญญาณดิจิตอลเป็นอนาล็อกมาใช้งานในวงจร	44
3.4 การพัฒนาทางด้านฮาร์ดแวร์	48
3.5สรุปการสร้างฮาร์ดแวร์	53
3.6การพัฒนาทางด้านซอฟทแวร์	55
บทที่ 4 การทดลองและผลการทดลอง	**61**
4.1 การออกแบบซ็อฟทแวร์	61
บทที่ 5 สรุปและวิจารณ์	**68**
5.1 สรุปผลการดำเนินงาน	68
5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข	68
5.3 แนวทางการพัฒนาในอนาคต	68
ภาคผนวก	
กิตติกรรมประกาศ	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 แสดงบล็อคไดอะแกรมขั้นตอนการทำงานเบื้องต้นของโครงงาน	2
รูปที่ 2.1 แผนผังโครงสร้างของไอซี 8255	4
รูปที่ 2.2 แผนผังวงจรภายในและการจัดขาของไอซี	4
รูปที่ 2.3 แสดงความหมายของบิทต่าง ๆในรหัสควบคุม	7
รูปที่ 2.4 ลักษณะของรหัสควบคุมแบบต่าง ๆในโหมด 0	8
รูปที่ 2.5 โครงสร้างตรวจสอบสัญญาณของพอร์ตอินพุทและพอร์ตเอาท์พุท	10
รูปที่ 2.6 วงจรการต่อ 8255ในโหมด I	11
รูปที่ 2.7 แผนผังเวลาการรับและส่งข้อมูลโดยใช้ตัวตรวจสอบสัญญาณ	12
รูปที่ 2.8 โครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง	13
รูปที่ 2.9 วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอลแบบสโลปเดี่ยว	14
รูปที่ 2.10 วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอลแบบสโลปคู่	15
รูปที่ 2.11 แสดงตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบวงจรนับและวงจรตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อกประกอบกัน	16
รูปที่ 2.12 แสดงการต่อวงจรแบบใช้วงจรเปรียบเทียบแบบขนาน	17
รูปที่ 2.13 แสดงการต่อวงจรแบบประมาณค่า	18
รูปที่ 2.14 แสดงส่วนประกอบพื้นฐานของการแปลงดิจิตอลเป็นอนาล็อก	20
รูปที่ 2.15 แสดงวงจรแปลง D/A แบบ Binary Weighted Resisteor Ladder	21
รูปที่ 2.16 แสดงวงจรD/A แบบ R-2R Ladders	22
รูปที่ 2.17 การใช้แอดเดรสบิทต่าง ๆ ในการอ้างแอดเดรสของพอร์ตใน IBM/PC	25
รูปที่ 2.18 ตัวอย่างวงจรดีโค้ดแอดเดรสแบบFixed	26
รูปที่ 2.19 ตัวอย่างวงจรดีโค้ดโดยใช้สวิตช์เลือก	28
รูปที่ 2.20 ตัวอย่างวงจรดีโค้ดโดยใช้ PROM	30
รูปที่ 3.1 ระบบบัสของ IBM/PC	35
รูปที่ 3.2 แสดงตำแหน่งขาไอซี 74LS 688	37
รูปที่ 3.3 วงจรดีโค้ดโดยใช้สวิตช์เลือก	38
รูปที่3.4 แสดงตำแหน่งขาของไอซี 74LS245	38
รูปที่ 3.5 แสดงการต่อวงจรพอร์ต 8255 ของอินเตอร์เฟสการ์ด	40
รูปที่ 3.6 แสดงลักษณะการต่อ ADC0805 เข้ากับไมโครโปรเซสเซอร์	41
รูปที่ 3.7 แสดงวงจรการนำไปใช้งานของ ADC0805	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 แสดงขาของวงจรไอซี LM 358 44
รูปที่ 3.9 แสดงวงจรการใช้งานของ DAC0832 44
รูปที่ 3.10 แสดงวงจรการนำไปใช้งานของ DAC0832 47
รูปที่ 3.11แสดงการต่อวงจรอินเตอร์เฟส 49
รูปที่ 3.12แสดงขาต่าง ๆ ของไอซี AD574A 48
รูปที่ 3.13แสดงวงจรการนำไปใช้งานของ AD574A 51
รูปที่3.14แสดงรูปขาของไอซี AD7837 51
รูปที่ 3.15แสดงวงจรการนำไปใช้งานของ AD7837 53
รูปที่ 3.16แสดงการ์ดวงจรที่ต่อเสร็จแล้วสำหรับ 8 บิท 54
รูปที่ 3.17แสดงการ์ดวงจรที่ต่อเสร็จแล้วสำหรับ 12 บิท 54
รูปที่ 3.18แสดงการทำงานของโปรแกรมส่วนหน้าจอ 56
รูปที่ 3.19แสดงการทำงานโปรแกรม 58
รูปที่ 3.20แสดงการทำงานของส่วนแสดงผลของกราฟ 60
รูปที่ 4.1 แสดงรูปการใส่ค่า Sampling Time และค่าเวลาทั้งหมดที่ต้องการ Sampling 62
รูปที่ 4.2 แลดง การเลือกช่วงค่าสัญญาณและเลือกกราฟ 63
รูปที่ 4.3แสดงหน้าจอกราฟการแซมปลิ้ง 64
รูปที่ 4.4แสดงกราฟการแซมปลิ้งและโฮลด 65
รูปที่ 4.5แสดง Hex Data Table 66
รูปที่ 4.6Continuous Graph 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 สัญญาณควบคุมการกระทำของ 8255	6
ตารางที่ 2.2 หน้าที่ของสัญญาณต่าง ๆของพอร์ต Cในโหมด 1	11
ตารางที่ 2.3 หน้าที่ของพอร์ตC ในโหมด 2	12
ตารางที่ 2.4 ตารางแสดงความสัมพันธ์ระหว่างอินพุทที่เป็นอนาล็อกกับเอาท์พุทที่เป็นดิจิตอล	17
ตารางที่ 2.5 แสดงแอดเดรสทั้ง 8 กลุ่ม	26
ตารางที่ 2.6 แสดงการจัดแอดเดรส	31
ตารางที่ 2.7 แสดงแอดเดรสอื่น ๆ	32
ตารางที่ 3.1 ตารางแสดงแอดเดรสพอร์ต I/O ของIBM/PC	34
ตารางที่ 3.2 แสดงผลการแปลงค่าของ ADC0805 เมื่อใส่อินพุท 0-5 โวลท์	42
ตารางที่ 3.3 แแสดงผลการแปลงค่าของ ADC0805 เมื่อใส่อินพุท –5 -+5 โวลท์	43
ตารางที่ 3.4 แสดงผลการแปลงค่าของ DAC0832	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

เนื่องมาจากในระบบงานควบคุมต่างๆจะต้องมีการนำเอาสัญญาณจากอุปกรณ์ตรวจจับ (Sensor) มาเข้ายังอุปกรณ์ควบคุม (Controller) เพื่อที่จะทำการวิเคราะห์สัญญาณและส่งต่อออกไปเพื่อทำการควบคุมระบบต่อไป แต่ในการส่งสัญญาณจากอุปกรณ์ตรวจจับเข้าสู่อุปกรณ์ควบคุมนั้น อุปกรณ์ตรวจจับจะทำการส่งสัญญาณออกมาเป็นสัญญาณอนาล็อก (Analog Signal) แต่สำหรับตัวอุปกรณ์ควบคุมนั้น ซึ่งในระบบงานปัจจุบันอาจเป็น ไมโครคอนโทรลเลอร์ หรือ คอมพิวเตอร์ เพราะใช้งานง่ายและมีความยืดหยุ่นสูง จะรับสัญญาณอินพุท (Input Signal) เป็นสัญญาณดิจิตอล (Digital Signal)และเช่นเดียวกันในการที่จะส่งสัญญาณจากอุปกรณ์ควบคุมออกไปเพื่อที่จะควบคุมระบบ(Plant)ใดระบบหนึ่งนั้น อุปกรณ์ควบคุมจะส่งสัญญาณออกไปควบคุมเป็นดิจิตอล แต่ระบบจะรับสัญญาณที่จะเข้ามาควบคุมเป็นอนาล็อก ดังนั้นจากสาเหตุดังกล่าวข้างต้น จึงทำให้จะต้องมีอุปกรณ์อย่างใดอย่างหนึ่งที่จะทำหน้าที่ แปลงสัญญาณทั้งระหว่างขาเข้าและขาออกจากอุปกรณ์ควบคุม นั่นคือจะต้องมี ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter : ADC) ทำหน้าที่แปลงสัญญาณด้านขาเข้าอุปกรณ์ควบคุมและมี ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก (Digital to Analog Converter : DAC) ทำหน้าที่แปลงสัญญาณด้านขาออก

สำหรับโครงงานนี้ จัดทำขึ้นเพื่อสาธิตการทำงานของ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลและตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก เนื่องจากการแปลงสัญญาณจากอนาล็อกเป็นดิจิตอลและการแปลงสัญญาณจากดิจิตอลเป็นอนาล็อกนั้นไม่สามารถดูและพิจารณาขั้นตอนของการเปลี่ยนแปลงสัญญาณของ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และ ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อกซึ่งจะกระทำอยู่ภายในไอซีได้ จะสามารถดูได้เฉพาะสัญญาณอินพุทและเอาท์พุทเท่านั้น โครงงานนี้จึงจัดทำขึ้นเพื่อแสดงรูปสัญญาณในระหว่างขั้นตอนการทำงานของตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และ ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก โดยใช้คอมพิวเตอร์ (Personal Computer) เป็นตัวแสดงผล

1.2 วัตถุประสงค์ของโครงงาน

1.เพื่อสาธิตหลักการทำงานขอตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และ ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก

2.เพื่อให้เกิดความเข้าใจมากขึ้น ในวิธีการ แซมปิ้ง และ โฮลด

3.เพื่อศึกษาหลักการติดต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.สามารถออกแบบการ์ดแปลงสัญญาณของตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก ที่ใช้ร่วมกับคอมพิวเตอร์ได้

5.เพื่อจะได้ศึกษาหลักการและแนวคิดในการเขียนโปรแกรม เพื่อใช้ในการควบคุมและแสดงผล และจะได้นำไปประยุกต์ใช้งานอื่น ๆ ต่อไปตามวัตถุประสงค์

6.เพื่อที่จะนำไปเป็นสื่อการเรียนการสอนให้กับผู้ที่สนใจเรื่องการติดต่อกับคอมพิวเตอร์และหลักการทำงานของ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และ ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก

7.เพื่อที่จะเรียนรู้และแก้ไขปัญหาที่เกิดขึ้นในขณะทำการทดลอง

1.3 ขอบเขตของโครงงาน

เพื่อทำการสร้างระบบจำลองการทำงานของ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก โดยที่จะแสดงผลออกมาทางจอ (Monitor) ของคอมพิวเตอร์ โดยมีขั้นตอนการทำงานดังนี้

1.รับสัญญาณอินพุทเป็นอนาล็อกจากภายนอกเป็นค่า –5 ถึง +5 Volt และ 0 ถึง 5 Volt

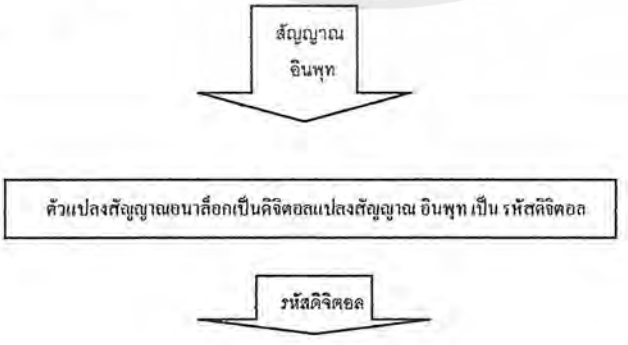
2.แสดงรูปสัญญาณอินพุทที่ถูกแซมปิ้ง โดย ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล (โดยมี ซ็อฟทแวร์ เป็นตัวคำนวณสร้างรูปกราฟขึ้นมา)

3.แสดงรูปสัญญาณอินพุทที่ถูกแซมปิ้งจากนั้นทำการโฮลด

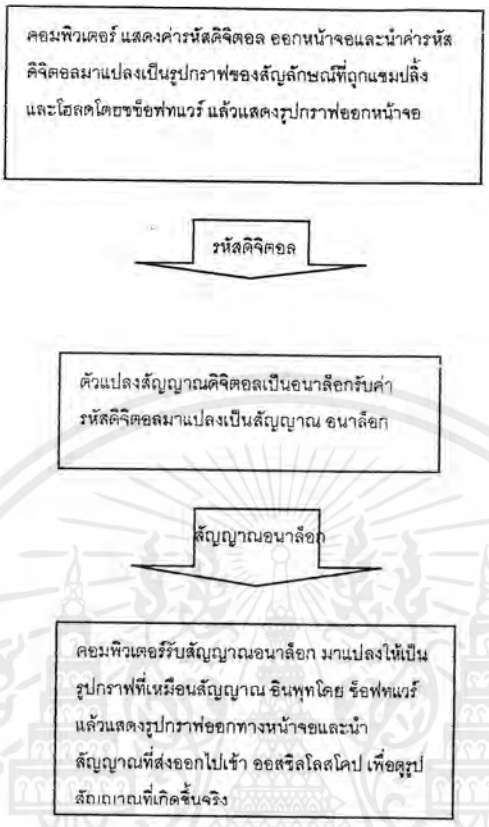
4.แสดงค่าของสัญญาณอินพุทที่ถูก ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล แปลงเป็นดิจิตอล ออกมาในรูปแบบตาราง

5.แสดงรูปสัญญาณเอาท์พุทที่ถูกส่งออกมาและทำการโฮลด

6.แสดงรูปสัญญาณเอาท์พุทที่ถูก ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก แปลงออกมาเป็นอนาล็อกโดยจะใช้ ออสซิลโลสโคป เป็นตัวแสดงผลโดยสามารถแสดงขั้นตอนเบื่องต้นเป็นบล็อกไดอะแกรม ได้ดังรูปที่1.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงบล็อคไดอะแกรมขั้นตอนการทำงานเบื้องต้นของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

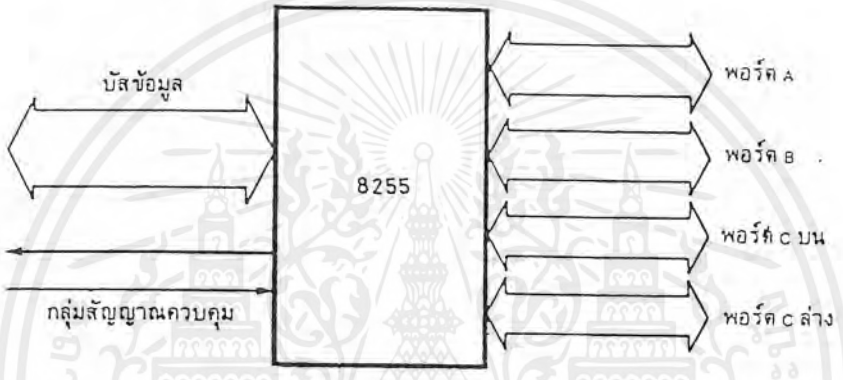
บทที่2

ทฤษฎี และ หลักการ

2.1 8255 พอร์ตข้อมูลแบบขนาน

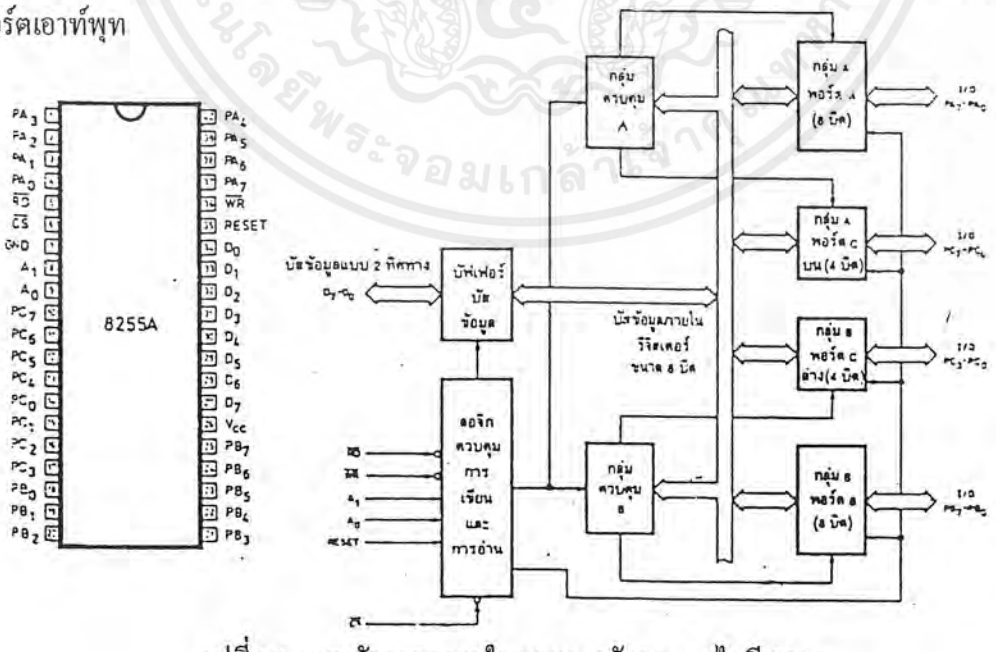
2.1.1 โครงสร้างพื้นฐานของ8255

8255 เป็นไอซีที่มี 40 ขา ที่ต่อเป็นพอร์ตให้ไมโครโปรเซสเซอร์ ได้ 3 พอร์ต โดยมีโครงสร้างพื้นฐานดังรูปที่ 2.1



รูปที่2.1 แผนผังโครงสร้างของไอซี 8255

การเรียก พอร์ตของ 8255 จะเรียกพอร์ตต่างๆ ว่า พอร์ตA, พอร์ตB , พอร์ตCโดยพอร์ต C แยกเป็น 2 ส่วน คือ พอร์ต C ล่าง และพอร์ต C บน โดยทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุท และพอร์ตเอาท์พุท



รูปที่2.2 แผนผังวงจรภายในและการจัดขาของไอซี 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 เป็นแผนผังภายในของไอซีและการจัดวางขาของไอซี 8255 การทำงานของวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงาน โดยไมโครโปรเซสเซอร์จะส่งคำสั่งมาโปรแกรมการทำงานหรือกำหนดรูปแบบของพอร์ตให้เป็นอินพุทหรือเอาท์พุทได้

2.1.2 ขาต่างๆ ของ 8255

ขาทั้ง 40 ขาของไอซีประกอบด้วย

1. D0 – D7 เป็นขาที่ข้อมูลอินพุทเอาท์พุทจะต้องผ่านเข้าออกจากส่วนนี้ D0 – D7 จึงต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถอ่านหรือเขียนข้อมูลจากพอร์ตผ่านทางบัสนี้
2. $\overline{CS}$ (สัญญาณเลือกชิพ) ขานี้เป็นขาอินพุทที่จะรับสัญญาณจากภายนอกเพื่อเลือกชิพ 8255 โดยเมื่อขานี้เป็น " 0" จะทำให้ 8255 ต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์เขียนหรืออ่านข้อมูลจากพอร์ตได้
3. $\overline{RD}$ (สัญญาณการอ่าน) เป็นสัญญาณอินพุทที่ต้องส่งมาจากซีพียูเมื่อสัญญาณที่ขานี้เป็น " 0" เป็นสัญญาณ $\overline{CS}$ เป็น "0" ด้วย ไอซี 8255 จะทำตัวให้ซีพียูอ่านข้อมูลจากบัสขณะที่เป็นพอร์ทอินพุท
4. $\overline{WR}$ เป็นสัญญาณการเขียน จะแอคทีฟเมื่อมีสัญญาณ $\overline{WR}$ และสัญญาณ $\overline{CS}$ เป็น "0" สัญญาณนี้จะมาจากซีพียูเมื่อต้องการเขียนข้อมูลลงบนพอร์ตอินพุทเอาท์พุทของ 8255
5. A0 – A1 (สัญญาณแอดแดรส) ลอจิกของสัญญาณทั้งสองจะถอดรหัสออกเป็น 4 รหัส เพื่อกำหนดรีจิสเตอร์ภายในที่เชื่อมต่อกับพอร์ตอินพุทเอาท์พุทของ 8255
6. RESET (สัญญาณรีเซต) เป็นสัญญาณที่ส่งจากภายนอกเข้ามาทำการรีเซต 8255 เพื่อเคลียสถานะต่างๆ ของ 8255 เมื่อ 8255 ได้รับการรีเซต ก็จะกลับเข้าสู่โหมดอินพุทหรือทุกพอร์ตที่เป็นพอร์ตอินพุท
7. PA0 – PA7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต A การเลือกพอร์ตจะเลือกโดยสัญญาณแอดเดรส A0 – A1
8. PB0 – PB7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต B การเลือกพอร์ตจะเลือกโดยสัญญาณแอดเดรส A0 – A1
9. PC0 – PC7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต C การเลือกพอร์ตจะเลือกโดยสัญญาณแอดเดรส A0 - A1 พอร์ต C นี้แบ่งเป็นสองกลุ่ม คือ กลุ่มPC0 -PC3 และกลุ่ม PC4 - PC7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 รีจิสเตอร์ภายในของ 8255

การใช้งาน 8255 จะต้องส่งรหัสควบคุม (control code) เข้าไปยังพอร์ตข้อมูลเพื่อควบคุมการทำงานของ 8255 โดยแต่ละพอร์ตจะเสมือนเป็นรีจิสเตอร์ ซึ่งสามารถอ่านและเขียนข้อมูลได้ สัญญาณของขาควบคุมที่ประกอบกันจะแสดงดังตารางที่2.1

ตารางที่2.1 สัญญาณควบคุมการกระทำของ 8255

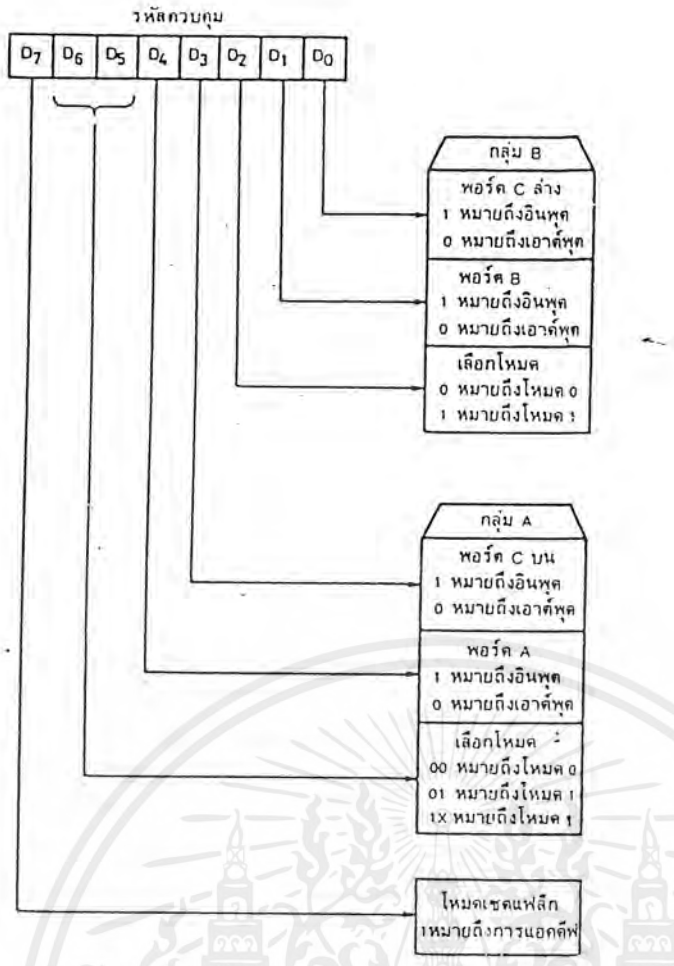
$\overline{RD}$	$\overline{WR}$	$\overline{A1}$	$\overline{A0}$	ความหมาย
1	0	0	0	เขียนพอร์ต A ซึ่งเป็นข้อมูล
0	1	0	0	อ่านพอร์ต A ซึ่งเป็นข้อมูล
1	0	0	1	เขียนพอร์ต B ซึ่งเป็นข้อมูล
0	1	0	1	อ่านพอร์ต B ซึ่งเป็นข้อมูล
1	0	1	0	เขียนพอร์ต Cซึ่งเป็นข้อมูลเ
0	1	1	0	อ่านพอร์ต Cซึ่งเป็นข้อมูล
1	0	1	1	เขียนข้อมูล ซึ่งเป็นรหัสควบคุม
0	1	1	1	อ่านเข้ามา ซึ่งไม่มีความหมายใด

ในการควบคุมการทำงานของ 8255 จะมีอยู่หลายโหมด แต่ละโหมดจะแตกต่างกันไปการโปรแกรมให้ 8255 ทำงานจะทำได้ 3 โหมดคือโหมด 0 โหมด 1 และโหมด 2

2.1.4 รหัสควบคุม

การกำหนดโหมดการทำงานของ 8255 นั้นจะต้องส่งข้อมูลคำสั่งเข้าไปโปรแกรมในพอร์ตควบคุมของ 8255 แต่ละบิทของข้อมูลที่ส่งไปจะมีความหมายในตัวเอง ลักษณะความหมายของแต่ละบิทในรหัสควบคุมแสดงได้ดังรูป 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.3 แสดงความหมายของบิทต่าง ๆ ในรหัสควบคุม

การโปรแกรม 8255 คือ การให้ค่ารหัสบิทต่าง ๆ เข้าไปในรหัสควบคุมแล้วส่งไปยังรีจิสเตอร์ของพอร์ตควบคุม ความหมายของบิทต่าง ๆ มีดังนี้

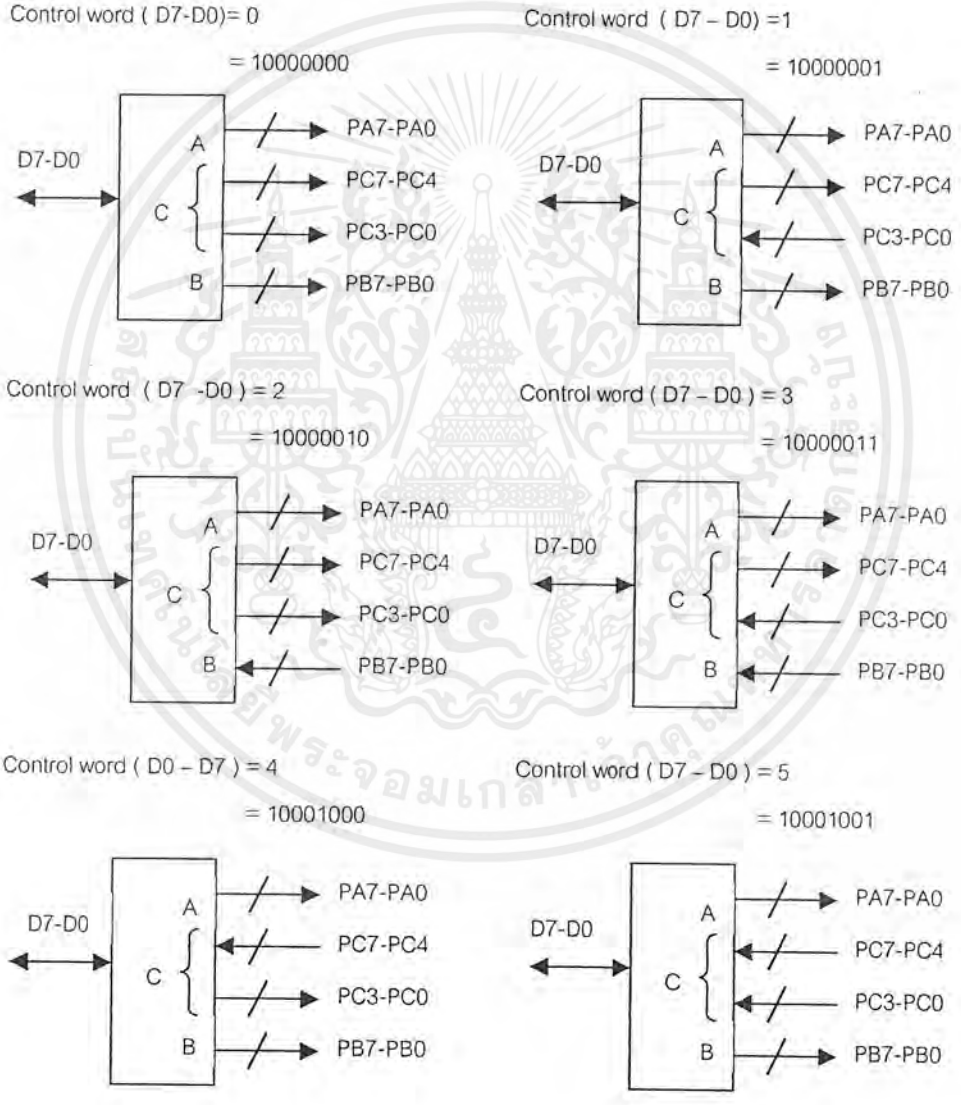
- บิท D7 เป็นบิทที่แสดงรหัสคำสั่งควบคุม ถ้าบิทนี้เป็น “1” หมายถึงรหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเซตโหมดต่าง ๆ ของ 8255
- บิท D6 และ D5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมี 3 โหมด คือโหมด 0 โหมด1 และ โหมด 2
- บิท D4 ถ้ามีค่าเป็น “0” หมายถึงการกำหนดพอร์ต A เป็นเอาท์พุต ถ้ามีค่าเป็น “1” จะหมายถึงการกำหนดให้พอร์ต A เป็นอินพุท
- บิท D3 เป็นบิทที่บอกถึงการเซตพอร์ต C บน ถ้าเป็น “0” จะทำให้พอร์ต C บนเป็นเอาท์พุต ถ้าเป็น “1” จะเป็นอินพุท
- บิท D2 เป็นบิทที่บอกถึงการเซตโหมดของพอร์ต B ถ้าเป็น “0” หมายถึง เลือกพอร์ต B เป็นโหมด 0 และถ้าเป็น “1” หมายถึงการเลือกโหมด 1
- บิท D1 ถ้ามีค่าเป็น “0” หมายถึงการกำหนดพอร์ต B เป็นเอาท์พุท ถ้ามีค่าเป็น “1” จะหมายถึงการกำหนดให้พอร์ต B เป็นอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิท D0 ถ้ามีค่าเป็น "0" หมายถึงการกำหนดพอร์ต C ล่างเป็นเอาท์พุท ถ้ามีค่าเป็น "1" จะหมายถึงการกำหนดให้พอร์ต C ล่างเป็นอินพุท

2.1.5 การทำงานในโหมด 0

โหมด 0 เป็นโหมดที่กำหนดให้พอร์ตทุกตัวบน 8255 เป็นพอร์ตอินพุทหรือเอาท์พุทแบบพื้นฐาน รูปแบบความเป็นไปได้จึงมีทั้งสิ้น 16 รูปแบบ ตามลักษณะของพอร์ต A พอร์ต B พอร์ต C บน และพอร์ต C ล่าง ลักษณะของรหัสควบคุมแต่ละแบบจะเป็นดังรูป2.4

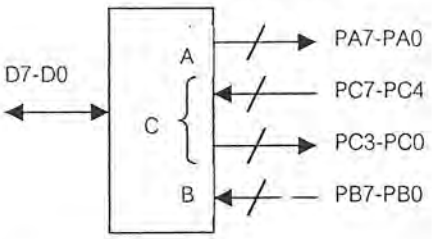


รูปที่ 2.4 ลักษณะของรหัสควบคุมแบบต่าง ๆ ในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

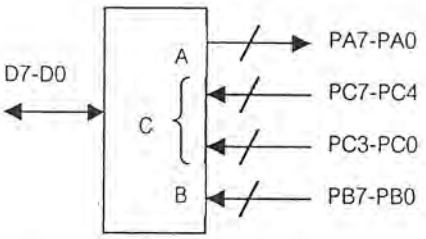
Control word (D7-D0) = 6

= 10001010



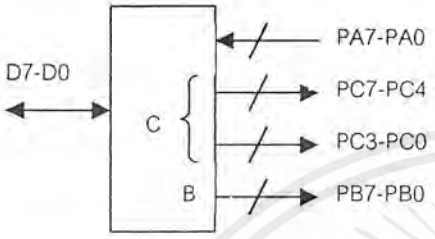
Control word (D7- D0) = 7

= 10001011



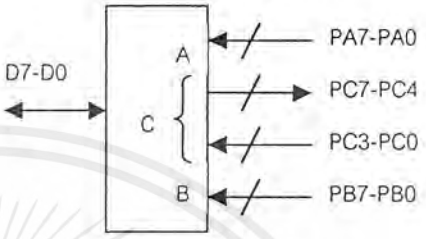
Control word (D7 – D0) = 8

= 10010000



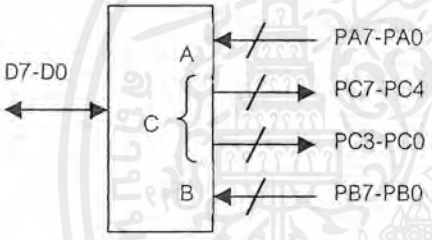
Control word (D7 – D0) =9

= 10010001



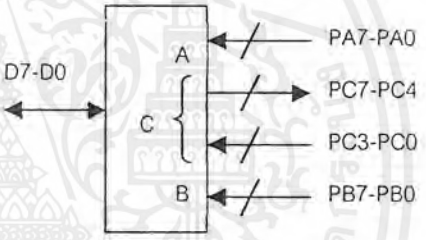
Control word (D7 -D0) = 10

= 10010010



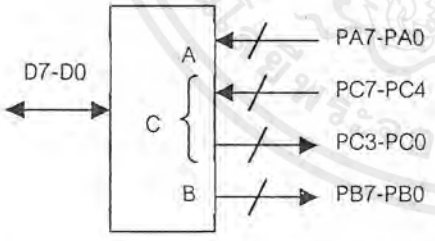
Control word (D7 – D0) = 11

= 10010011



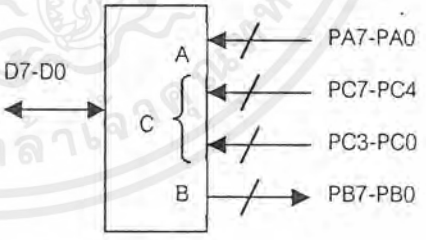
Control word (D0 – D7) = 12

= 10011000



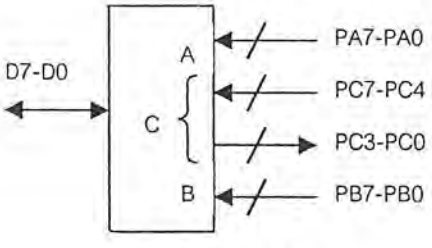
Control word (D7 – D0) = 13

= 10011001



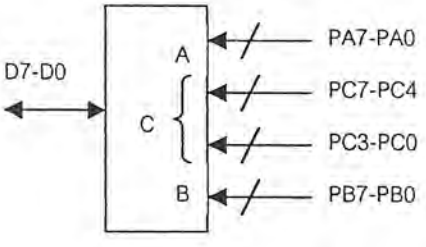
Control word (D7-D0) = 14

= 10011010



Control word (D7- D0) = 15

= 10011011

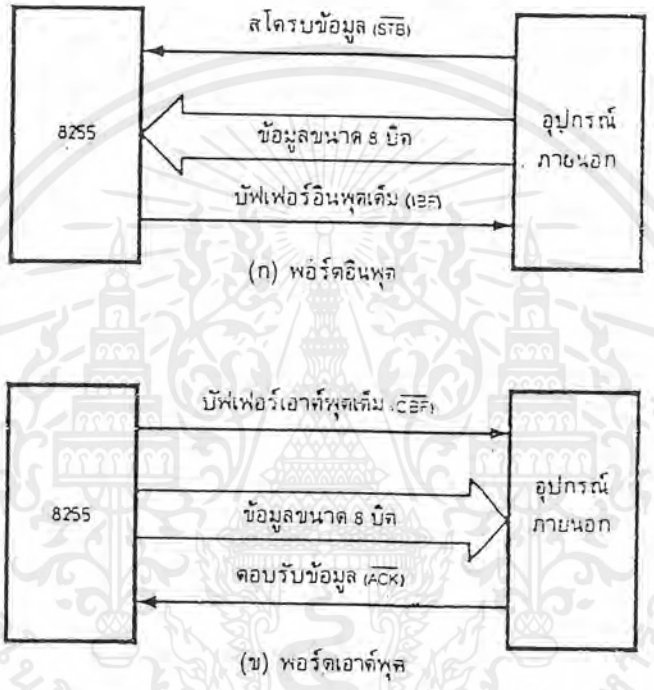


รูปที่ 2.4 (ต่อ) ลักษณะของรหัสควบคุมแบบต่าง ๆ ในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การทำงานในโหมด 1

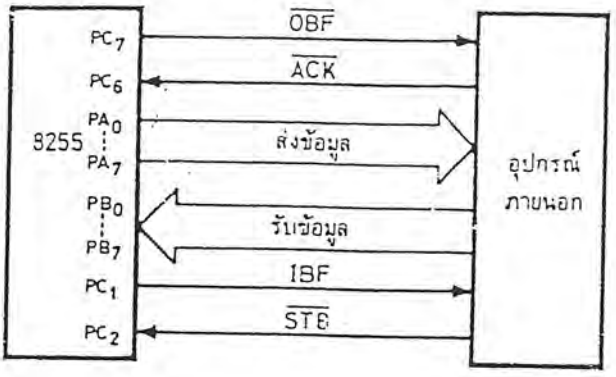
การทำงานของ 8255 ในโหมด 1 เป็นโหมดที่ทำให้อินพุทเอาท์พุทมีการตรวจสอบสัญญาณ (handshaking) โดยใช้อินพุทเอาท์พุทของพอร์ต A และพอร์ต B เป็นหลัก และใช้พอร์ต C บนเป็นตัวตรวจสอบสัญญาณ (handshake) ของพอร์ต A ส่วนพอร์ต C ล่างเป็นตัวตรวจสอบสัญญาณของพอร์ต B การจัดสัญญาณต่าง ๆ เหล่านี้แสดงดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้างตัวตรวจสอบสัญญาณของพอร์ตอินพุทและพอร์ตเอาท์พุท

แนวความคิดของการใช้พอร์ตอินพุทเอาท์พุท โดยมีตัวตรวจสอบสัญญาณก็เพื่อให้มีการซิงโครไนซ์ ระหว่างอุปกรณ์ภายนอกที่ทำงานได้ช้ากับการทำงานของคอมพิวเตอร์ที่ทำงานได้เร็วเช่น เครื่องพิมพ์ทำงานได้ช้า เมื่อคอมพิวเตอร์ส่งอักษรตัวแรกมา เครื่องพิมพ์รับตัวอักษรและกำลังจะพิมพ์ คอมพิวเตอร์ก็ส่งอักษรตัวที่ 2 ตัวที่ 3 ตามมา ทำให้การประมวลผลของอุปกรณ์เครื่องพิมพ์ทำงานไม่ทัน ซึ่งอาจจะทำให้ข้อมูลเกิดการสูญหาย ดังนั้นเครื่องพิมพ์จึงส่งสัญญาณบอกคอมพิวเตอร์ว่า " อย่าเพิ่งส่งเพราะยังไม่พร้อมจะรับ" ดังรูปที่ 2.5 จะใช้ PA0 – PA7 เป็นเอาท์พุทและ PB0 – PB7 เป็นอินพุท โดยมีพอร์ต C เป็นตัวตรวจสอบสัญญาณ ดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 วงจรการต่อ 8255 ในโหมด 1

เมื่อโปรแกรม 8255 เป็นโหมด 1 แล้ว ตัว 8255 จะให้พอร์ต C เป็นสัญญาณควบคุม โดยแต่ละบิทของพอร์ต C เป็นไปตามตารางที่ 2.2

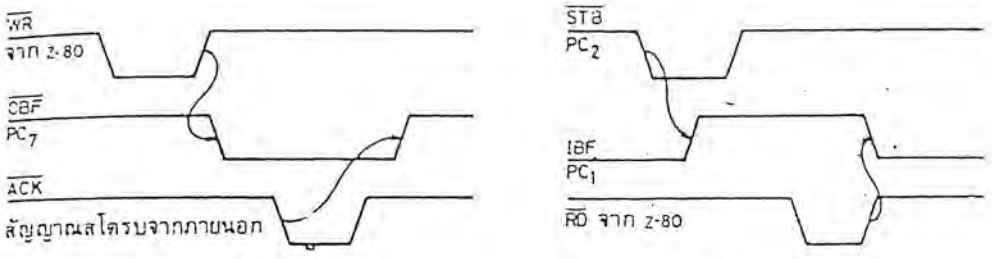
ตารางที่ 2.2 หน้าที่ของสัญญาณต่าง ๆ ของพอร์ต C ในการทำงานเป็นตัวตรวจสอบสัญญาณเมื่อ 8255 ทำงานในโหมด 1

ขา	กรณีอินพุท	กรณีเอาท์พุท
PC_0	$INTR_B$	$INTR_B$
PC_1	$\overline{IBF}_B$	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	I/O
PC_5	$\overline{IBF}_A$	I/O
PC_6	I/O	$\overline{ACK}_A$
PC_7	I/O	$\overline{OBF}_A$

โดยปกติ 8255 จะให้สัญญาณอินเตอร์รัพต์ไปบอก ซีพียู ด้วย สัญญาณอินเตอร์รัพต์ของ 8255 จะเกิดขึ้นที่ PC0 และ PC3 โดยที่บัฟเฟอร์พร้อมแล้วและต้องการให้ ซีพียู ส่งอินพุทหรือเอาท์พุทมาที่บัฟเฟอร์ สัญญาณอินเตอร์รัพต์จะเกิดขึ้น

โครงสร้างการตรวจสอบสัญญาณของ 8255 แสดงด้วยสัญญาณทางไฟฟ้าได้ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) เมื่อเป็นพอร์ตเอาท์พุต (ข) เมื่อเป็นพอร์ตอินพุต

รูปที่ 2.7 แผนผังเวลาการรับและส่งข้อมูลโดยใช้ตัวตรวจสอบสัญญาณ

สังเกตว่า การทำงานของ 8255 จะเกี่ยวข้องกับสัญญาณ $\overline{RD}$ และ $\overline{WR}$ ซึ่งจะทำให้สัญญาณควบคุมมีสถานะเปลี่ยนแปลงไป ทำให้การตรวจสอบสัญญาณซึ่งกันและกัน เป็นวิธีการรับส่งที่มีประสิทธิภาพ เช่น ในกรณีอินพุท เมื่ออุปกรณ์ภายนอกต้องการส่งข้อมูลให้ ซีพียู ก็จะส่งข้อมูลแบบขนานเข้ามาพร้อมทั้งสโตรป ($\overline{STB}$) บอก 8255 ซึ่ง 8255 จะนำข้อมูลนั้นไปเก็บไว้ที่รีจิสเตอร์ภายในก่อน แล้วส่งสัญญาณ (IBF) บอกว่าบัฟเฟอร์ยังเก็บค่าเอาไว้เต็มอยู่ไม่สามารถรับข้อมูลที่ส่งเข้ามาใหม่ได้ จนเมื่อ ซีพียู อ่านข้อมูลจากรีจิสเตอร์ไปแล้ว สัญญาณ (IBF) ก็จะเป็น "0" เพื่อบอกว่าส่งสัญญาณใหม่มาได้

ในทำนองเดียวกัน สำหรับพอร์ตเอาท์พุท เมื่อ CPU ส่งข้อมูลออกมาทางพอร์ตเอาท์พุทให้กับ 8255 ตัว 8255 ก็จะรับไว้ในรีจิสเตอร์ภายใน พร้อมทั้งส่งสัญญาณออกไปบอกอุปกรณ์ภายนอก ($\overline{OBF}$) ว่า มารับข้อมูลที่เอาท์พุทบัฟเฟอร์ อุปกรณ์ภายนอกเมื่อรับทราบและพร้อมจะอ่านก็จะส่งสัญญาณตอบ ($\overline{ACK}$) เพื่อแสดงว่าอ่านข้อมูลไปแล้ว

2.1.7 การทำงานในโหมด 2

การทำงานในโหมด 2 จะทำได้เฉพาะพอร์ต A โดยจะใช้พอร์ต A ทำหน้าที่แบบสองทิศทาง คือสามารถเป็นได้ทั้งพอร์ตอินพุทและเอาท์พุท โดยโครงสร้างของพอร์ต A ทั้งอินพุทเอาท์พุทมีตัวตรวจสอบสัญญาณทั้งคู่ ส่วนพอร์ต C จะทำหน้าที่เป็นสัญญาณตรวจสอบ โดยมีสัญญาณแต่ละขาดังตารางที่ 2.3

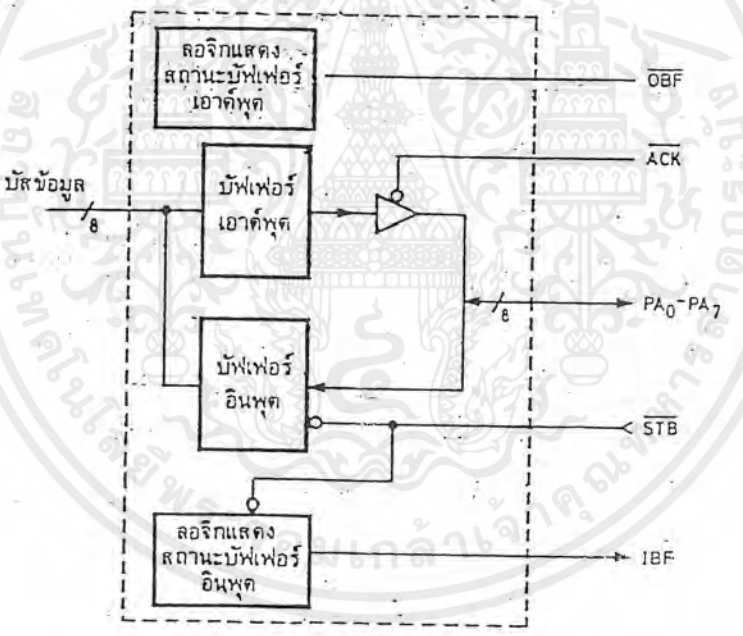
ตารางที่2.3 หน้าที่ของพอร์ต C ในโหมด 2

พอร์ต C	ความหมาย
PC_0	I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ส่วนโครงสร้างของพอร์ต A ที่ทำงานแบบ 2 ทิศทาง สามารถแสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง

สังเกตว่า เมื่อโปรแกรมพอร์ต A เป็นโหมด 2 แล้ว พอร์ต B จะต้องโปรแกรมเป็นโหมด0 หรือโหมด 1 ก็ได้ ซึ่งก็ทำงานแบบแยกอิสระอีก ในการใช้งานพอร์ตแบบ 2 ทิศทางนี้ใช้ได้กับงานบางประเภท เช่นใช้ในการรับส่งข้อมูลของพอร์ตมาตรฐานบางประเภทเช่น IEEE 488 หรือใช้เชื่อมโยงระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ในการรับส่งข้อมูลสลับกันไปและกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การแปลงอนาล็อกเป็นดิจิตอล (Analog to Digital Convertion)

2.2.1 แนะนำ

ในการที่จะให้ไมโครคอมพิวเตอร์รับรู้สัญญาณอนาล็อกได้จะต้องมีการแทนสัญญาณอนาล็อกระดับค่าหนึ่ง ๆ ด้วยค่าของตัวเลขไบนารี่ เราเรียกว่าการแปลงอนาล็อกเป็นดิจิตอล ในตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล บางชนิดจะมีการใช้ตัวแปลงเป็นดิจิตอล (D/A Conversion) อยู่ภายในตัวแปลงด้วย วิธีการใช้ในการแปลง อนาล็อกเป็นดิจิตอล มีหลายวิธี ที่มีใช้ในปัจจุบันอาจแยกได้ 4 วิธีดังนี้

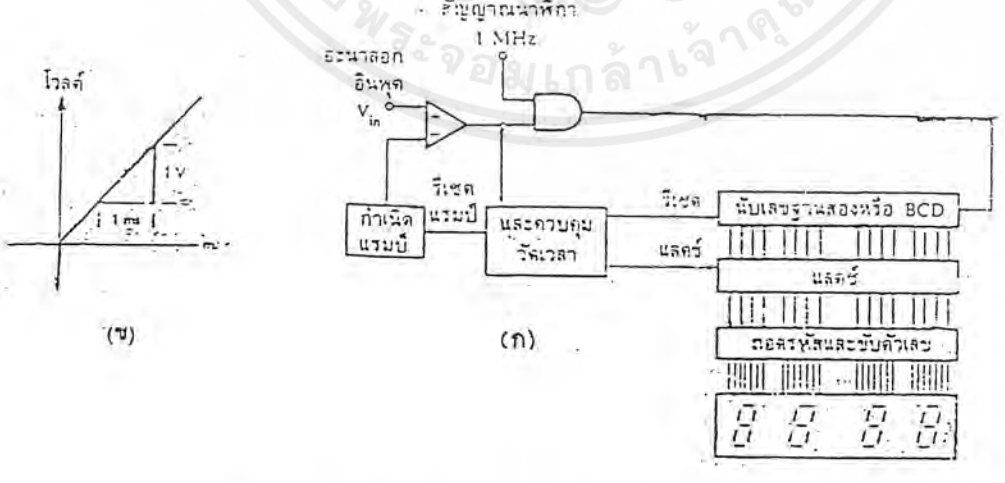
1. แบบอินทิเกรท (Integrating)
2. แบบขนาน (Flash)
3. แบบไบนารี่แรมพ์ (Binary Ramp)
4. แบบประมาณค่าอย่างต่อเนื่อง (Successive Approximation)

2.2.2 ตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลแบบใช้การอินทิเกรท

วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล ที่ใช้เทคนิคการอินทีเกรทสัญญาณมี 3 แบบ คือ

2.2.2.1 <u>แบบสโลปเดี่ยวหรือแบบแรมพ์</u> (Single Ramp หรือ Single Slope A/D Converter)

วงจร แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลแบบนี้แสดงไว้ในรูป 2.9 ประกอบด้วยวงจรกำเนิดสัญญาณแรมพ์ วงจรเปรียบเทียบ วงจรนับเลขฐานสอง(BCD)



รูปที่ 2.9 วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล แบบสโลปเดี่ยว

(ก) แสดงบล็อกไดอะแกรม (ข) ความชันของสัญญาณแรมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มทำการเปลี่ยนสัญญาณ สัญญาณแรมพ์และวงจรนับจะถูกรีเซตให้เป็น 0 แรงดันอนาล็อกถูกป้อนไปยังวงจรเปรียบเทียบทางขาอินพุทแบบไม่กลับ เมื่อแรงดันอินพุทที่ขานี้เป็นบวกมากกว่าที่ขาอินพุทแบบกลับ วงจรเปรียบเทียบก็จะให้เอาท์พุทระดับเป็น "สูง(High)" ทำให้แอนด์เกตปล่อยสัญญาณนาฬิกาผ่านไปยังวงจรนับได้และทำให้เริ่มเกิดสัญญาณแรมพ์

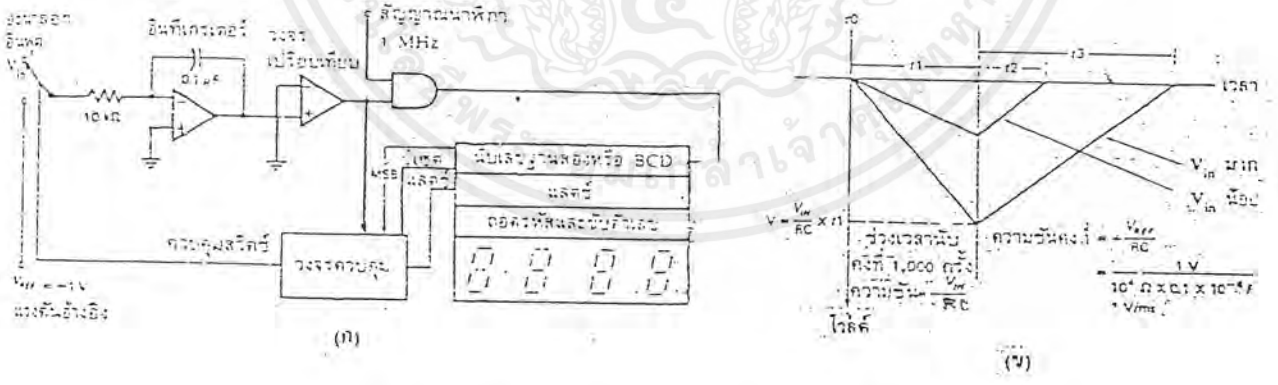
สัญญาณแรมพ์มีแรงดันเป็นบวกขึ้นเรื่อย ๆ จนมากกว่าระดับแรงดันอินพุท เอาท์พุทของวงจรเปรียบเทียบก็ตกลงมาเป็นระดับ "ต่ำ(Low)" ส่งผลให้แอนด์เกตไม่มีสัญญาณผ่านไปให้วงจรนับ

วงจรนับจะหยุดนับและเก็บค่าไว้ที่วงจรแลตช์ จากนั้นจึงทำการรีเซตวงจรนับและวงจรกำเนิดสัญญาณแรมพ์

ข้อเสียของวงจรแบบนี้คือไม่ควรใช้กับงานที่ต้องการความถูกต้องสูง เนื่องจากการเปลี่ยนแปลงในแหล่งกำเนิดสัญญาณแรมพ์ ขึ้นกับอุณหภูมิและผลตอบสนองต่อสัญญาณอินพุททำให้ไม่มีความคงที่ ดังนั้นจึงมีการปรับปรุงให้ดีขึ้นเป็นแบบสโลปคู่ (dual – slope)

2.2.2.2 <u>แบบสโลปคู่</u> (Dual – Slope A/D Converter)

ดังรูปที่ 2.10 แสดงบล็อกไดอะแกรมของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลแบบสโลปคู่ ซึ่งวงจรส่วนใหญ่คล้ายกับแบบสโลปเดี่ยว แต่มีสวิตช์ที่อินพุทเพิ่มขึ้นเพื่อทำการเลือกระหว่างแรงดันอินพุทกับแรงดันอ้างอิง (วงจรเปรียบเทียบต่อขาสัญญาณอินพุทกลับกันกับแบบสโลปเดี่ยว)



รูปที่ 2.10 วงจรเปลี่ยนสัญญาณ A/D แบบสโลปคู่

(ก) แสดงบล็อกไดอะแกรม (ข) เอาท์พุทของวงจรอินทีเกรเตอร์เมื่อเทียบกับเวลา

ส่วนประกอบของวงจรมี วงจรกำเนิดสัญญาณแรมพ์หรือวงจรอินทีเกรเตอร์ ซึ่งจะกำเนิดสัญญาณแรมพ์ที่เป็นเชิงเส้น แต่มีสโลปเป็นลบ ซึ่งวงจรเปรียบเทียบก็จะได้รับแรงดันเป็นลบเข้ามาแล้วให้เอาท์พุทเป็นบวกทำการเปิดแอนด์เกตให้สัญญาณนาฬิกาผ่านเข้าสู่วงจรนับ วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นับจะนับไปจนถึงค่าคงที่ที่กำหนดไว้ แล้วสับสวิตช์ต่อเข้ากับแรงดันอ้างอิง ซึ่งจะให้แรมพ์ที่สโลปเป็นบวกจนมีค่าเป็น 0 วงจรนับจะเก็บค่าที่เวลานั้นไว้ซึ่งค่าในวงจรนับจะเป็นสัดส่วนโดยตรงกับแรงดันอินพุท

ข้อดีของวงจรแบบนี้ คือ ความถูกต้องสูง ราคาสูง เสถียรภาพทางด้านอุณหภูมิดี

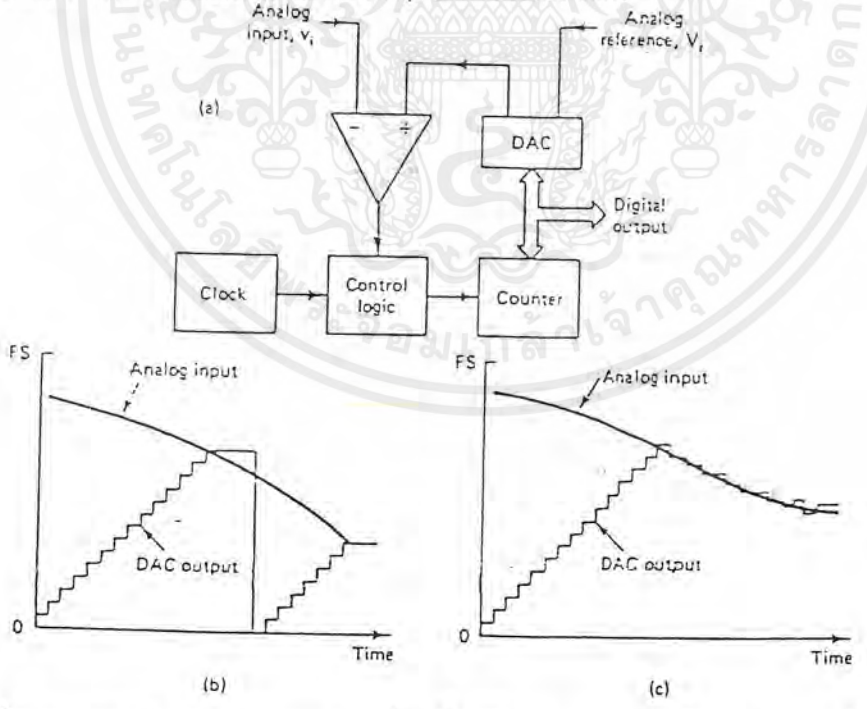
ข้อเสีย คือ ความเร็วต่ำ ในการเปลี่ยนสัญญาณ 1 ครั้งอาจใช้เวลาถึง 100 ms

2.2.2.3แบบชาร์จบาลานซ์(Change Balance A/D Converter)

วงจรแบบนี้จะคล้ายกับแบบสโลปคู่ แต่แทนที่จะให้อินพุท สวิตช์ไปมาระหว่างแรงดันที่ไม่รู้ค่ากับแรงดันอ้างอิง ก็ทำการแทรกพัลส์ของกระแสอ้างอิงมาตรง ๆ ที่จุดรวม (Summing Point) ของวงจรอินทีเกรเตอร์ในช่วงเวลาคงที่ โดยที่จำนวนพัลส์จะเป็นสัดส่วนโดยตรงกับแรงดันอินพุทที่ไม่รู้ค่า

2.2.3 ตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลแบบที่ใช้วงจรนับและวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกประกอบกัน

รูปที่ 2.11 แสดงบล็อกไดอะแกรมของวงจร โดยที่มีหลักการเช่นเดียวกันกับแบบสโลป เพียงแต่สโลปของวงจรนี้ถูกสร้างโดยวงจรนับและวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อก โดยแรมพ์นี้จะมีลักษณะเป็นขั้นบันไดเล็ก ๆ จำนวนมากนั่นเอง



รูปที่ 2.11 แสดงตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล แบบวงจรนับและวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกประกอบกัน

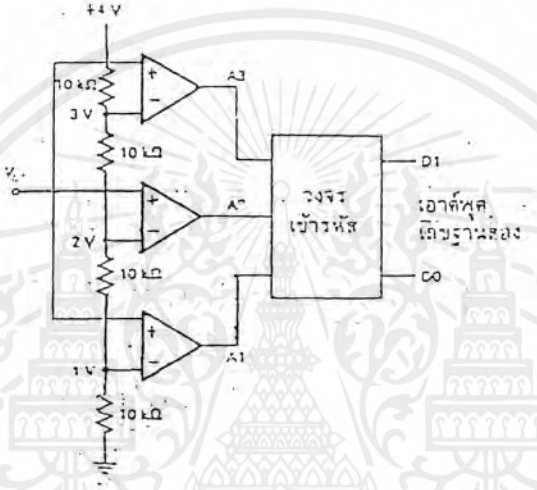
แสดงบล็อกไดอะแกรม (b) แสดงรูปคลื่นแบบวงจรนับ (c) แสดงรูปคลื่นแบบ tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.4 ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล แบบใช้วงจรเปรียบเทียบขนานหรือแบบ "แฟลช" (Parallel Comparator Simultaneous " Flash" A/D Converter)

วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล แบบนี้ใช้วงจรเปรียบเทียบที่ต่อขนานกัน ดังรูป 2.12 ประกอบด้วยออปแอมป์ที่ต่อเป็นวงจรเปรียบเทียบและตัวต้านทานต่อไว้เพื่อแบ่งแรงดันที่ขาอินพุทแบบกลับให้มีขนาดต่าง ๆ



รูปที่ 2.12 แสดงการต่อวงจร เปรียบเทียบแบบขนาน

จากหลักการของวงจรเปรียบเทียบทั่วไป เมื่อแรงดันอินพุทที่ขาอินพุทแบบไม่กลับ มีค่าสูงกว่าที่ขาอินพุทแบบกลับ เอาท์พุทจะได้ค่าสูง ดังตาราง 2.4 แสดงความสัมพันธ์ระหว่างแรงดันอินพุทที่เป็นอะนาล็อกกับเอาท์พุทที่เป็นดิจิตอล

ตารางที่ 2.4 ตารางแสดงความสัมพันธ์ระหว่างอินพุทที่เป็นอะนาล็อกกับเอาท์พุทที่เป็นดิจิตอล

แรงดันอินพุท Vin (โวลท์)	เอาพุทของวงจรเปรียบเทียบ			เอาท์พุทเลขฐานสอง	
	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36856

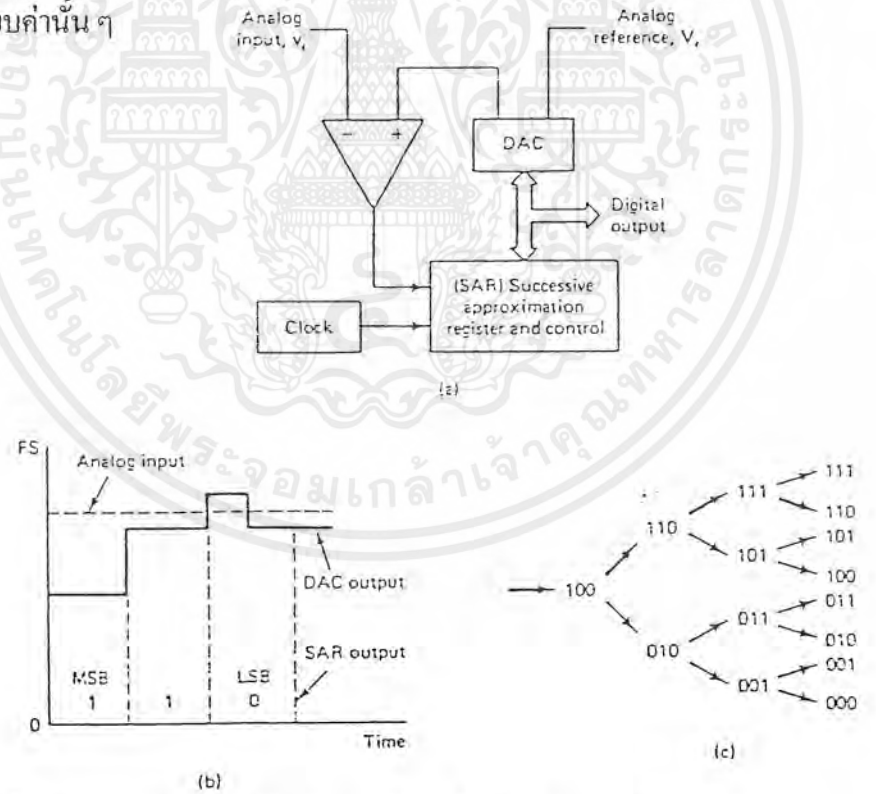
เมื่อต้องการวงจรที่มีความละเอียดสูงขึ้น จำเป็นต้องใช้วงจรเปรียบเทียบมากขึ้น เป็น $2^N - 1$ เมื่อ N แทนจำนวนบิทหรือความละเอียดที่ต้องการ

ข้อเสียของวงจรแบบนี้ คือจะใช้ตัวเปรียบเทียบมาก เช่น ใช้ 255 ตัว เมื่อต้องการความละเอียด 8 บิท

ข้อดีของวงจรแบบนี้ คือมีความเร็วสูงมาก คือใช้เวลาในการแปลงได้เร็วในระดับนาโนวินาที

2.2.5 ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบใช้การประมาณค่า (Successive Approximation A/D Converter)

ประกอบด้วยวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อก ซึ่งจะแปลงค่าแรงดัน (Voltage) เปรียบเทียบ มาเปรียบเทียบกับสัญญาณอินพุท โดยค่าที่เปรียบเทียบแต่ละครั้งจะเป็นเอาท์พุทของ ค่าดิจิตอลแต่ละบิท จากบิทสูงไปบิทต่ำ โดยจะมีพัลส์จากสัญญาณนาฬิกาคอยสั่งให้เปรียบเทียบทีละบิท ถ้า ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลมี 8 บิท ก็จะต้องการพัลส์ 8 ลูก เพื่อทำการเปรียบเทียบค่านั้น ๆ



รูปที่ 2.13 แสดงตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบประมาณค่า
(a)แสดงบล็อกไดอะแกรม (b)แสดงรูปคลื่นของวงจร (c)แสดง logic ของการทำงาน

จากรูปที่ 2.13 แสดงตัวอย่างการทำงานของตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 3 บิท โดยที่พัลส์ลูกแรกค่าจาก ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก จะมีค่าเท่ากับค่ากลางของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันอ้างอิง(Voltage Reference คือ 100) ซึ่ง น้อยกว่าอินพุทค่าในรีจิสเตอร์ภายในจึงเป็น "1" เมื่อมีพัลส์ลูกที่ 2 ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก จะเพิ่มค่าขึ้นเป็น 110 ซึ่งน้อยกว่า อินพุทอยู่อีก ค่าบิทที่ 2 จึงเป็น "1" เมื่อมีพัลส์ลูกที่ 3ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อกจะเพิ่มค่าเป็น 111 มากกว่าอินพุท ค่าบิทที่ 3 จึงเป็น "0" สังเกตว่า ความจริงสัญญาณอินพุทจะมีค่ามากกว่า 110 แต่น้อยกว่า 111 ซึ่ง ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อกA ขนาด 3 บิท มีความละเอียดไม่พอที่จะเปรียบเทียบได้ ต้องทำการเพิ่มขนาดบิทของตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก จึงจะได้เอาท์พุทที่มีค่าใกล้เคียงอินพุทขึ้น

ข้อดีของวงจรแบบนี้คือ ความละเอียดของวงจรขึ้นกับบิท ถ้าต้องการให้เอาท์พุทละเอียดมากก็เพิ่มขนาดของตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลและเวลาที่ใช้ในการแปลงคงที่ทุก ๆ ค่า

2.3 การแปลงดิจิตอลเป็นอนาล็อก (Digital to Analog Conversion)

2.3.1 แนะนำ

การแปลงดิจิตอลเป็นอนาล็อก จะใช้วงจรหรืออุปกรณ์ที่หน้าที่แปลงสัญญาณดิจิตอลซึ่งอาจเป็นแรงดันหรือกระแส ให้เป็นสัญญาณอะนาล็อกที่เป็นสัดส่วนกับสัญญาณดิจิตอลที่ป้อนเข้าไปเป็นอินพุทของวงจร เราสามารถเขียนสมการเอาท์พุทของการแปลงดิจิตอลเป็นอนาล็อก ได้ดังนี้

$$X = K \times A \times B$$

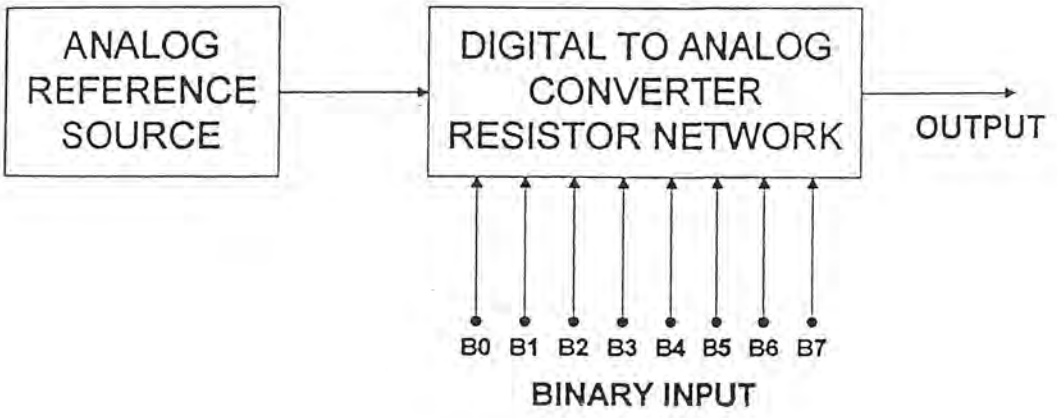
โดยที่ X = เป็นค่าแรงดันหรือกระแสทางด้านเอาท์พุท (อนาล็อก)

A = ค่าอ้างอิงอนาล็อก (เป็นแรงดันหรือกระแสก็ได้)

B = จำนวน (ค่า) ของตัวเลข Binary

K = ค่าคงที่จะมีค่าเป็น 1 เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงส่วนประกอบพื้นฐานของการแปลงดิจิตอลเป็นอนาล็อก

จากรูปที่ 2.14 แสดงส่วนประกอบต่าง ๆ ของระบบการแปลงดิจิตอลเป็นอนาล็อก โดยที่แหล่งจ่ายอ้างอิงของอนาล็อกจะมีค่าคงที่อยู่ค่าหนึ่ง อาจจะเป็นแหล่งจ่ายแรงดันหรือกระแสที่มีความเที่ยงตรงสูง จุดประสงค์เพื่อใช้เปรียบเทียบกับอินพุท เพื่อที่จะสร้างแรงดันหรือกระแสที่เอาท์พุท

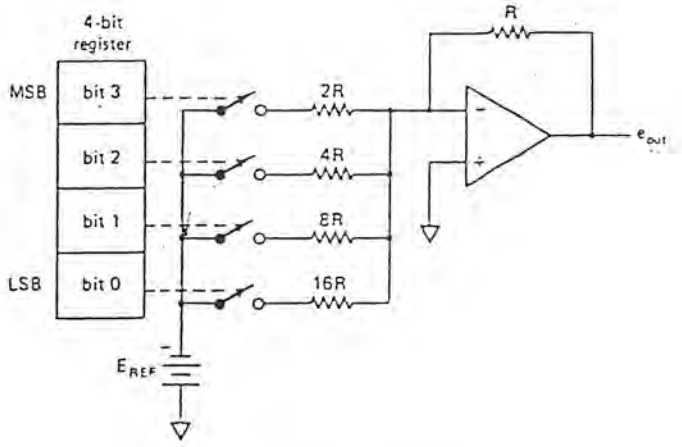
การแปลงดิจิตอลเป็นอนาล็อก จะแบ่งตามการใช้ตัวต้านทาน ซึ่งจะมีการใช้ตัวต้านทานโดยทั่ว ๆ ไปอยู่ 2 ลักษณะ คือ แบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Register Ladder) และแบบใช้ตัวต้านทานขั้นบันได (R – 2R Ladder)

2.3.2 การแปลงดิจิตอลเป็นอนาล็อกแบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Register Ladder)

การใช้ตัวต้านทานแบบนี้ จะเป็นการแปลงข้อมูลดิจิตอลให้เป็นอนาล็อกโดยตรง จะมีตัวต้านทานต่ออนุกรมอยู่กับแรงดันอ้างอิง (Vref) โดยจะมีอิเลคโทรนิคสวิทช์ใช้ในการ เปิด – ปิด สัญญาณ ตามสถานะของสัญญาณดิจิตอลซึ่งมีได้ 2 ระดับคือ ลอจิก 0 กับ 1 การต่อลักษณะนี้ค่าของตัวต้านทานจะมีค่าในแต่ละน้ำหนักที่คูณด้วย 2 ตลอดคือ จะมีตั้งแต่ R,2R,4R.... จนถึง nR หรือเขียนเป็นสมการได้ว่า $2^{(n-1)}R$ โดยที่ n คือ จำนวนบิทของเลขไบนารี่ ที่จะทำการแปลง

เมื่อตัวต้านทานถูกต่อลง กเรานด จะไม่มีกระแสไหลผ่านตัวต้านทานที่ต่ออยู่ แต่ถ้าตัวต้านทานตัวนั้นต่ออยู่กับแรงดันอ้างอิง (แทนด้วย ระดับลอจิกที่ตรงข้ามกับการต่อตัวต้านทานลงกเรานด) จะมีกระแสไหลผ่านตัวต้านทานตัวนั้น ดังนั้นถ้าหากตัวต้านทานค่า R และ 2R ถูกต่อกับแรงดันอ้างอิงพร้อมกัน ก็จะมีกระแสไหลผ่านตัวต้านทาน R มีค่าเท่ากับ Vref/R และกระแสไหลผ่าน 2R มีค่าเท่ากับ Vref/2R ค่ากระแสทั้ง 2 จะไหลมารวมกันที่จุดผสม (Summing Point) ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงวงจร D/A Converter แบบ Binary weighted Register Ladder

จากรูปที่ 2.15 สามารถเขียนสมการได้ดังนี้

$$I_{out} = \frac{Vref}{R}\left[\sum_{i=1}^{n}\frac{b_i}{2^{(i-1)}}\right]$$

โดยที่ I_{out} = กระแสเอาท์พุท (สัญญาณอะนาล็อก)มีหน่วยเป็น Ampere(A)

b_i = ค่าตัวเลขไบนารี่ อินพุท (0 หรือ 1)

R = ค่าตัวต้านทานตัวแรกที่มีค่าต่ำที่สุด (เป็นค่า R ของ บิทนัยสำคัญสูงสุด (MSB – Most Significant Bit)

จากรูปที่ 2.15 จะเห็นได้ว่าตัวต้านทานที่ทำให้กระแสไหลได้สูงสุด คือ R ดังนั้นตัวต้านทาน R จะเป็นตัวต้านทานของบิทนัยสำคัญสูงสุด (MSB – Most Significant Bit) และยังเห็นได้ว่าค่าของตัวต้านทานเพิ่มขึ้นที่ละมากๆ เช่น ถ้าหากมีการแปลงสัญญาณดิจิตอลขนาด 8 บิท แล้วถ้าใช้ค่า R = 10 kΩ แล้วตัวต้านทานตัวที่ 8 จะมีค่าเท่ากับ$2^{(8-1)}$ R = 128R = 128 x 10 kΩ = 1280kΩ หรือ 1.28 MΩ จะทำให้มีกระแสไหลผ่านตัวต้านทานตัวนี้น้อยมากและจากธรรมชาติที่ว่าตัวต้านทานค่ามากจะสร้างได้ยากและยังมีผลกระทบจากสิ่งแวดล้อมภายนอกอีก คือ ความร้อนจะทำให้ค่าความต้านทานเปลี่ยนไป (โดยเฉพาะตัวต้านทานที่มีค่ามาก ๆ) ทำให้ความละเอียดและความแม่นยำของการแปลง D/A แบบนี้ลดลง ดังนั้นจากข้อจำกัดที่ว่าเมื่อมีจำนวนบิทสูงขึ้นจะทำให้ต้องใช้ตัวต้านทานที่มีค่ามาก ๆ ซึ่งหาได้ยากและสร้างได้ยากรวมถึงยังมีผลกระทบต่อความแม่นยำในการแปลง เพราะฉะนั้นเมื่อมีจำนวนบิทสูงขึ้นจึงได้ทำการเปลี่ยนไปใช้แบบตัวต้านทานแบบขั้นบันได (R – 2R Ladder) ซึ่งจะมีการใช้ตัวต้านทานเพียง 2 ค่าเท่านั้น ซึ่งเป็นการขจัดปัญหาที่กล่าวมาข้างต้นได้ ดังจะกล่าวในหัวข้อถัดไป

2.3.3 การแปลงดิจิตอลเป็นอนาล็อกแบบตัวต้านทานขั้นบันได (R – 2R Ladder Converter)

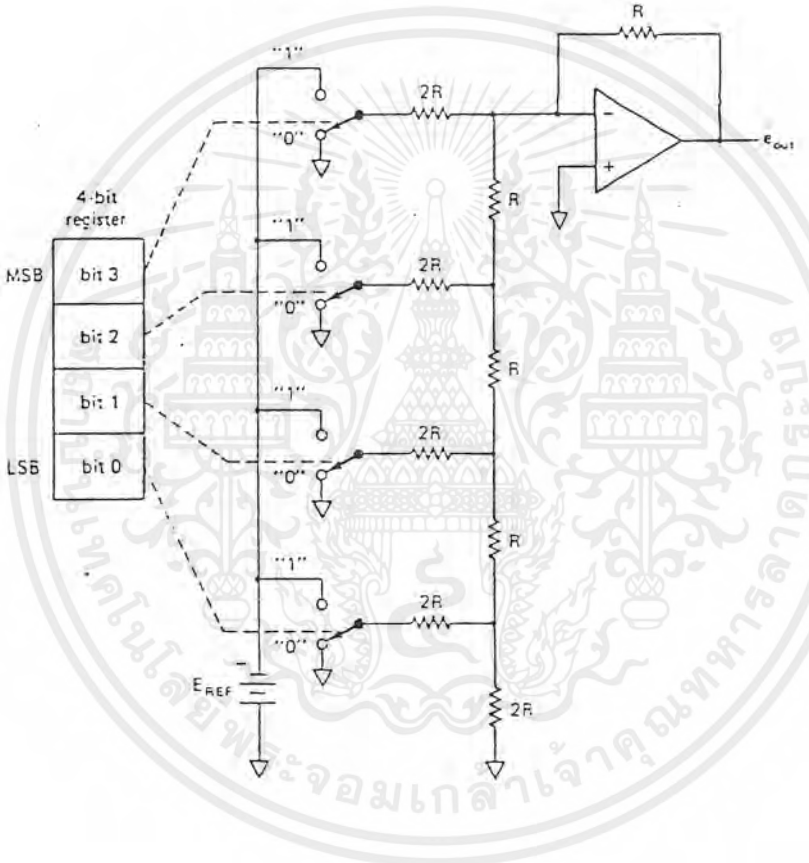
จะมีรูปวงจรดังแสดงในรูปที่ 2.16 โดยที่การใช้การแปลง D/A แบบ R – 2R Ladder นี้ใช้แก้ปัญหาของแบบแรก (Binary weighted Resistor Ladder) ที่ต้องใช้ความต้านทานหลายค่าซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจจะหาค่าได้ไม่ตรงกับที่ต้องการ มาใช้ตัวต้านทานเพียง 2 ค่าเท่านั้นในการแปลงแบบนี้ คือค่า R และ 2R ดังนั้นจากรูปที่ 2.16 วงจรสามารถเขียนสมการได้ดังนี้คือ

$$I_{out} = [\sum_{i=1}^{n} \frac{b_i}{2^{(i-1)}}]$$

ซึ่งเป็นสมการแบบเดียวกันกับแบบ Binary weighted Resistor Ladder



รูปที่ 2.16 แสดงวงจร D/A Converter แบบ R – 2R Ladder

2.3.4 คุณสมบัติและข้อกำหนดของตัวแปลง D/A (D/A Characteristics and Specification)

สำหรับตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก จะมีคุณสมบัติสำคัญดังต่อไปนี้

2.3.4.1 <u>ความละเอียด</u> (Resolution)

คุณสมบัติประการแรกของ D/A นี้คือความละเอียดของความสามารถในการเปลี่ยนแปลงพิจารณาบิทเลขฐานสองที่ป้อนเข้าที่อินพุทของตัวแปลง D/A ถ้าอินพุทเป็นเลขฐานสอง 8 ตัว จะแสดงว่ามีระดับของเอาท์พุทที่เป็นไปได้เท่ากับ $2^8 = 256$ ดังนั้นค่า Resolution ของ D/A ตัวนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 1 ใน 256 ในบางครั้งอาจจะกล่าวในรูปของเปอร์เซ็นต์ เช่น Resolutoin ของ D/A ขนาด 8 บิท คือประมาณ 0.39 %

2.3.4.2 ค่าเต็มพิกัดของแรงดันเอาท์พุท (Full Scale Output Voltage)

คุณสมบัติประการที่สองของ D/A คือค่าเต็มพิกัดของแรงดันเอาท์พุท คือค่าแรงดันสูงสุดที่ D/A สามารถให้ออกมาได้เมื่ออินพุทดิจิตอลมีค่าลอจิกเป็น "1" หมดทุกบิท

2.3.4.3 ความเที่ยงตรง (Accuracy)

ข้อกำหนดความเที่ยงตรงสำหรับ D/A คือการเปรียบเทียบระหว่าง เอาท์พุทที่เกิดขึ้นจริงกับเอาท์พุทที่คาดหวังไว้ (มาจากการคำนวณ) โดยที่จะมีการกำหนดค่าไว้เป็นเปอร์เซ็นต์ เช่น D/A มีแรงดันเต็มพิกัดของเอาท์พุท 10 V และความเที่ยงตรง 0.2 % ดังนั้นค่าผิดพลาดสูงสุดของเอาท์พุทใด ๆ เป็น 10V x 0002 = 20 mV สำหรับ D/A ในอุดมคตินั้นจะมีค่าความผิดพลาดสูงสุดไม่มากกว่า ½ ของบิทนัยสำคัญต่ำสุด (LSB – Least Significant Bit)

2.3.4.4 ความเป็นเชิงเส้น (Linearity)

คือการวัดค่าความเบี่ยงเบนของเอาท์พุทที่เป็นพิสัยจากเส้นตรงที่ตัวแปลงสัญญาณไม่มีการทำงานเลยไปจนถึงทำงานครบทุกบิท ซึ่งค่าเบี่ยงเบนในอุดมคติของเอาท์พุทจากเส้นตรง ควรมีไม่มากกว่า +½ ของค่า LSB

2.3.4.5 ค่าเวลาในการแปลง (Setting Time)

คือเมื่อเกิดการเปลี่ยนแปลงค่า Binary ที่ทำการป้อนกลับเป็นอินพุทเข้ามาในแต่ละบิทของ D/A เอาท์พุทจะเปลี่ยนแปลงเป็นสัญญาณอนาล็อกค่าใหม่ได้ถูกต้อง จะต้องใช้เวลาค่าหนึ่งซึ่งขึ้นกับข้อกำหนดของเวลาในการแปลงของตัว D/A ค่าเวลาที่เอาท์พุทใช้ไปภายใน ½ LSB ของค่าสุดท้ายจะเรียกว่า ค่าเวลา " Setting Time" สำหรับค่าการเปลี่ยนแปลงเอาท์พุทเต็มพิกัด คุณสมบัตินี้สำคัญเพราะถ้าตัวแปลงสัญญาณทำงานที่ความถี่สูง ๆ มันอาจจะไม่มีเวลาจะตั้งค่าก่อนที่มันจะสวิทช์ไปสู่อีกสถานะหนึ่ง

2.4 การจัดแอดเดรสสำหรับพอร์ตI/O ใน IBM/PC

ในหัวข้อนี้จะกล่าวถึงวิธีการอ้างและใช้งานในแอดเดรสต่าง ๆ ของพอร์ต I/O ที่ใช้งานอยู่ใน IBM/PC

2.4.1 การอ้างแอดเดรสของพอร์ต I/O

ในการควบคุมและตรวจสอบสภาวะการทำงานรวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพซัพพอร์ทหรือการ์ดต่าง ๆ ที่ใช้ในระบบ IBM/PC นั้นจะกระทำโดยผ่านทางพอร์ต I/O ของระบบ ดังนั้นในการที่จะใช้งานหรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นที่จะต้องศึกษาถึงวิธีการควบคุมพอร์ต I/O ต่าง ๆ ของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านี้ ต้องกระทำโดยการอ้างถึงแอดเดรสของพอร์ต I/O เหล่านั้นโดยตรง เราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสของ8088 ใน IBM/PC ด้วย

สำหรับแอดเดรสของพอร์ต I/O ต่าง ๆ นั้นจะเป็นแอดเดรสที่ถูกสร้างขึ้นโดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ต I/O เหล่านั้นโดยเฉพาะ คือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ส่วนการส่งข้อมูลให้กับพอร์ตเหล่านี้ จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้นไปยังแอดเดรสของพอร์ตที่ต้องการ และสำหรับการตรวจสอบหรือการอ่านข้อมูลจากพอร์ต ก็จะทำได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสของพอร์ตที่ต้องการเช่นกัน

ภายในไมโครโปรเซสเซอร์เบอร์ 8088 นี้จะมีแอดเดรสสำหรับใช้กับพอร์ต I/O อยู่ทั้งสิ้น65,536 หรือ 64K แอดเดรส (ในขณะที่มีแอดเดรสสำหรับหน่วยความจำอยู่ 1 MB) ซึ่งทำให้การอ้างแอดเดรสของพอร์ต I/O ที่ทำงานร่วมกับ 8088 นั้น ต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้น คือ A0 – A15 แต่สำหรับใน IBM/PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสในบัสแอดเดรสเฉพาะ 10 เส้นล่าง คือ A0 – A9 เท่านั้น ดังนั้นในการอ้างแอดเดรสของพอร์ตของอุปกรณ์หรือชิพซัพพอร์ทใด ๆ ที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วย โดยเส้นแอดเดรสที่เหลือคือ A10 – A15 นั้นจะไม่ถูกนำไปใช้งาน แต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ตที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วย เพียงแต่ไม่ได้ถูกนำมาดีโค้ดร่วมกับแอดเดรส A0 – A9 เท่านั้น ตัวอย่างเช่นในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ตที่ตรงกับแอดเดรส 0010H นั้น จะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ตที่ตรงกับแอดเดรส 0410H , 0810H, 0C10H ทั้งนี้เนื่องจากแอดเดรส 6 บิทบนไม่ได้ถูกใช้งานจึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10 – A15 นั้นไม่ทำให้เกิดความแตกต่างใด ๆ ทั้งสิ้น

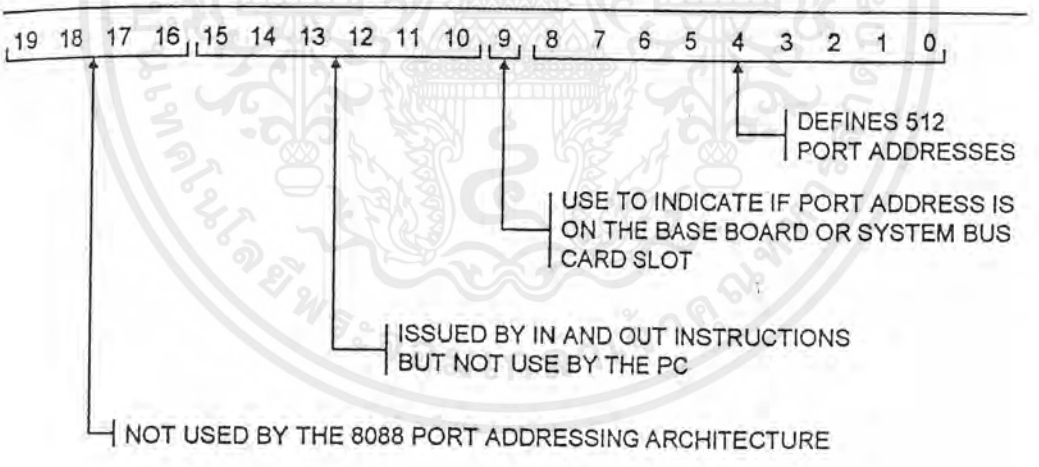
เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0 –A9) ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ตได้สูงสุดเพียง 1024 พอร์ต(จากจำนวน 64K พอร์ต) เท่านั้น นอกจากนี้ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ตของ IBM/PC ข้อมูลบิท A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ตทั้ง 1024 พอร์ตออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ต) อีกด้วย กล่าวคือถ้าข้อมูลในบิท A9 เป็น “0” แล้วเราจะทำการอ่านข้อมูลได้เฉพาะจากพอร์ตอุปกรณ์ หรือชิพซัพพอร์ตต่าง ๆ ที่อยู่บนเมนบอร์ด (Main Board) ของ IBM/PC เช่น 8253 – 5 , 8237 – 5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิท A9 นี้เป็น “1” ก็จะทำการอ่านข้อมูลได้เฉพาะจากพอร์ตที่อยู่บนการ์ดต่าง ๆ เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมานั้น จะสรุปได้ว่าพอร์ตบน IBM/PC ทั้ง 1024 พอร์ต ถูกแบ่งออกเป็น 2 กลุ่ม โดยที่กลุ่มแรกเป็นกลุ่มของพอร์ตที่อยู่บนเมนบอร์ด และกลุ่มที่สองเป็นกลุ่มที่จัดเตรียมไว้สำหรับพอร์ตที่อยู่บนการ์ดต่าง ๆ

สำหรับในกรณีของการส่งข้อมูลให้กับพอร์ตทั้ง 1024 พอร์ต เราสามารถที่จะเลือกส่งไปยังพอร์ตใด ๆ ใน IBM/PC ได้ ดังนั้นการเลือกแอดเดรสสำหรับพอร์ตที่อยู่บนการ์ดจึงสามารถทำได้โดยสะดวก แต่อย่างไรก็ตามสิ่งหนึ่งที่ต้องคำนึงถึงก็คือ ถ้าแอดเดรสที่เลือกให้กับพอร์ตที่อยู่ในตำแหน่งแอดเดรสนี้ก็เท่ากับเป็นการส่งข้อมูลให้กับทั้งพอร์ตที่อยู่บนเมนบอร์ด และบอร์ดที่อยู่บนการ์ดด้วย ซึ่งในกรณีเช่นนี้อาจก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรสให้กับพอร์ตที่ถูกสร้างขึ้นบนการ์ดต่าง ๆ จึงควรใช้ค่าแอดเดรสที่แอดเดรสบิท A9 มีค่าเป็น "1" คือแอดเดรส 0FE0H จนถึง 0FFFFH เท่านั้น (แอดเดรสบิท A10 - A15 ไม่ถูกใช้ในการดีโค้ด แต่เพื่อความสะดวกจึงกำหนดให้มีค่าเป็น "1" ในฐานสองทั้งหมด แต่ในกานใช้งานจริงอาจเปลี่ยนให้แอดเดรส A10 – A15 แต่ละบิทมีค่าเป็น "1"หรือ "0" ก็ได้)

สำหรับรูปที่ 2.17 นี้ จะแสดงถึงการใช้งานแอดเดรสบิทต่าง ๆ ในการอ้างแอดเดรสของพอร์ตใน IBM/PC



รูปที่ 2.17 การใช้แอดเดรสบิทต่าง ๆ ในการอ้างแอดเดรสของพอร์ตใน IBM/PC

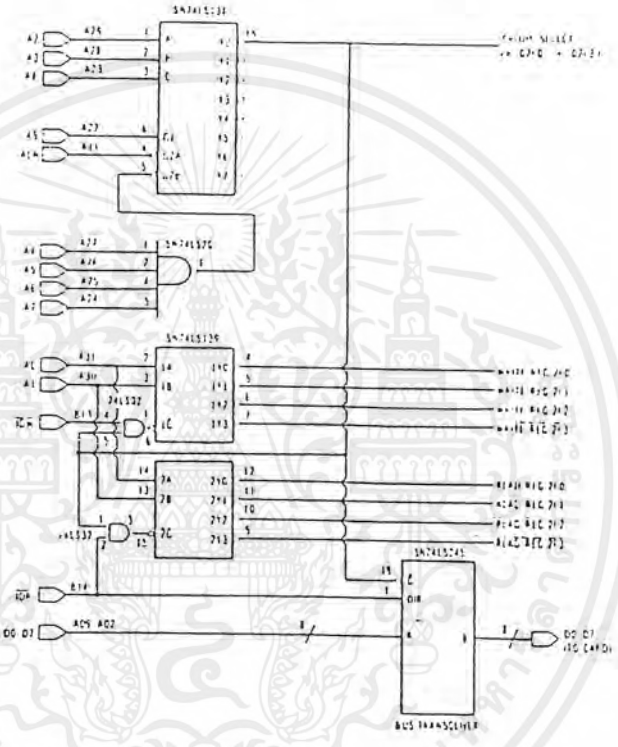
2.4.2 เทคนิคในการดีโค้ดแอดเดรสสำหรับพอร์ต I/O

ในหัวข้อต่าง ๆ ที่ผ่านมาข้างต้นนั้น ไดกล่าวถึงการอ้างแอดเดรสต่าง ๆ ของพอร์ต I/O ใน IBM/PC สำหรับในหัวข้อนี้จะกล่าวถึงวิธีต่าง ๆ ที่ใช้ในการดีโค้ดแอดเดรสต่าง ๆ ให้เป็นไปตามที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.1 การดีโค้ดแบบ Fixed

วิธีการดีโค้ดแบบนี้เป็นวิธีที่ง่ายและสะดวกในการดีโค้ดแอดเดรส หรือกลุ่มของแอดเดรสของพอร์ต I/O ซึ่งวิธีนี้เป็นการกำหนดจำนวนของแอดเดรสที่เราต้องกานใช้ จากนั้นจึงทำการเลือกบล็อกของแอดเดรสที่ยังไม่ถูกใช้งานโดยการ์ดหรือวงจรอินเทอร์เฟสอื่น ๆ (บล็อกของแอดเดรสที่เลือกต้องมีจำนวนแอดเดรสพอเพียงกับจำนวนแอดเดรสที่เราต้องกานใช้งาน) แล้วจึงออกแบบวงจรที่ทำการดีโค้ดแอดเดรสที่เราต้องการ สำหรับตัวอย่างวงจรที่ใช้ในการดีโค้ดแอดเดรสในแบบนี้จะแสดงดังรูปที่ 2.18



รูปที่ 2.18 ตัวอย่างวงจรดีโค้ดแอดเดรสแบบFixed

จากรูปจะเห็นได้ว่า วงจรที่ใช้นี้เป็นวงจรที่สามารถทำการดีโค้ดแอดเดรสได้ 8 กลุ่ม โดยแต่ละกลุ่มจะมีจำนวนแอดเดรส 4 แอดเดรส ซึ่งแอดเดรสทั้ง 8 กลุ่ม จะแสดงได้ดังตารางที่2.5ข้างล่าง

ตารางที่ 2.5 แสดงแอดเดรสทั้ง 8 กลุ่ม

กลุ่ม	แอดเดรส
0(Y0)	02F0H-02F3H
1(Y1)	02F4H-02F7H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2(Y2)	02F8H-02FBH
3(Y3)	02FCH-02FFH
4(Y4)	03F0H-03F3H
5(Y5)	03F4H-03F7H
6(Y6)	03F8H-03FBH
7(Y7)	03FCH-03FFH

สำหรับในตัวอย่างนี้จะเลือกใช้การดีโค้ดแอดเดรสในกลุ่ม 0 (เริ่มจากแอดเดรส 02F0H จนถึง 02F3H) คือใช้สัญญาณเอาท์พุท (สัญญาณ $\overline{GROUPSELECT}$) จากขา Y0 (ขา 15 ของ 74LS138) ไปทำการ OR กับสัญญาณ $\overline{IOR}$ และ $\overline{IOW}$ เพื่อสร้างเป็นสัญญาณอีนาเบิลวงจรดีโค้ด(74LS139) แอดเดรสอีก 4 แอดเดรส ซึ่งแบ่งเป็น 2 ชุด คือชุดที่เป็น $\overline{WRITEREG}$ ซึ่งจะแอกทีฟ(ลอจิก "0") เมื่อ CPU ต้องการจะส่งข้อมูลให้กับวงจรภายนอก (สัญญาณ $\overline{IOR}$ แอกทีฟ) และชุดที่เป็น $\overline{READREG}$ ซึ่งจะแอกทีฟเมื่อ CPU ต้องการจะอ่านข้อมูลจากวงจรภายนอก (สัญญาณ $\overline{IOR}$ จะแอคทีฟ) สัญญาณ $\overline{WRITEREG}$ และ $\overline{READREG}$ นี้ โดยทั่วไปจะนำไปเป็นสัญญาณสโตรป (Strobe) ให้กับวงขนภายนอกที่เกี่ยวข้อง เพื่อให้สามารถส่งหรือรับข้อมูลจาก CPU ได้ในช่วงเวลาที่เหมาะสม นอกจากนี้สัญญาณ $\overline{GROUPSELECT}$ ยังถูกนำไปใช้ในการอีนาเบิลบัฟเฟอร์ 74LS138 ด้วย เพื่อให้ CPU สามารถส่งหรือรับข้อมูลจากภายนอกได้เมื่อแอดเดรสในกลุ่มนี้ถูกเลือก สำหรับทิศทางของข้อมูลจะถูกควบคุมโดยสัญญาณ $\overline{IOR}$ ส่วนสัญญาณ AEN จะถูกนำมาใช้ในการดิสเอเบิลวงจรดีโค้ด โดยถ้าสัญญาณ AEN เป็น "1" ซึ่งเป็นช่วงเวลาของขบวนการ DMA นั้น 74LS138 จะถูกดิสเอเบิลทันที ทั้งนี้ก็เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นเนื่องจากการดีโค้ดแอดเดรสของพอร์ตในระหว่างขบวนการ DMA นั้นเอง (ในระหว่างนี้แอดเดรสบนบัสแอดเดรสจะเป็นแอดเดรสของหน่วยความจำ คือ สัญญาณ $\overline{MEMW}$ หรือ $\overline{MEMR}$ จะแอคทีฟ แต่ในขณะเดียวกันสัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ ก็จะแอคทีฟด้วย ดังนั้นถ้าไม่ดิสเอเบิลวงจรดีโค้ดไว้แล้ว อาจจะทำให้วงจรดีโค้ดคิดว่าแอดเดรสบนบัสแอดเดรสเป็นแอดเดรสของพอร์ต I/O ก็ได้)

ในการดีโค้ดแอดเดรสของพอร์ต I/O เราจะต้องคำนึงถึงช่วงเวลาของสัญญาณที่เกิดขึ้นในกระบวนการอ่าน หรือเขียนข้อมูลลงบนพอร์ต I/O ดังนี้

1. ในช่วงเริ่มต้นของบัสไซเคิลเกี่ยวกับพอร์ต I/O นั้น ถ้าสัญญาณจากวงจรดีโค้ดมีการหน่วงเวลา (Delay) มากเกินไป อาจจะทำให้สัญญาณดีโค้ดนี้เกิดขึ้นหลังจากที่สัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ แอคทีฟ และเนื่องจากค่าแอดเดรสบนบัสแอดเดรสนั้นเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นก่อนที่ค่าแอดเดรสที่ถูกต้องจะถูกส่งออกมาบนบัสแอดเดรสนั้น วงจรดีโค้ดจะได้รับค่าแอดเดรสอื่น ๆ อยู่ ซึ่งถ้าหากวงจรดีโค้ดมีการหน่วงเวลามากเกินไปแล้ว สัญญาณดีโค้ดแอดเดรสที่ไม่ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

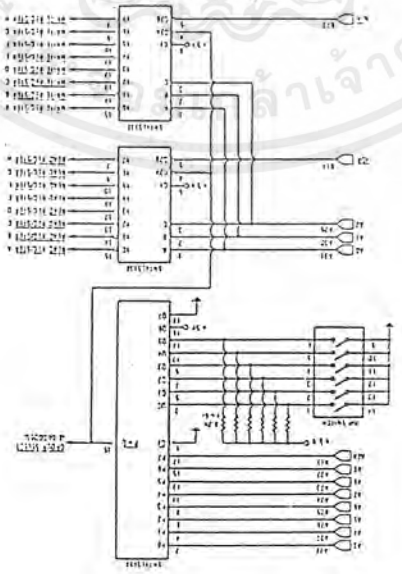
ต้องนี้อาจจะถูกหน่วงเวลาจนเกิดขึ้นในช่วงเวลาที่สัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ เกิดขึ้นแล้วก็ได้ ทำให้ข้อมูลบนบัสข้อมูลนั้นถูกส่งไปยังพอร์ตไม่ถูกต้อง สำหรับใน IBM/PC จะถูกออกแบบให้การหน่วงเวลาในวงจรดีโค้ดนั้นมีค่าไม่เกิน 92 nanosec

2. ในช่วงท้ายของบัสไซเคิลในการเขียนข้อมูลลงบนพอร์ต I/O นั้น ถ้าสัญญาณ $\overline{IOW}$ มีการหน่วงเวลาออกไป และวงจรดีโค้ดมีความเร็วในการทำงานสูงแล้ว อาจจะทำให้ข้อมูลในบัสไซเคิลนี้ถูกส่งให้กับพอร์ต I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสในบัสไซเคิดต่อไปถีได้สำหรับใน IBM/PC สัญญาณ $\overline{IOW}$ จะมีการหน่วงเวลาไปไม่เกิน 200 nanosec

อย่างไรก็ตามช่วงเวลาที่ต้องสนใจมากอีกช่วงเวลาหนึ่งคือ ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ $\overline{IOW}$ กับช่วงเวลาที่ข้อมูลที่ถูกต้องถูกส่งออกมาบนบัสข้อมูล ถ้าสัญญาณ $\overline{IOW}$ ถูกหน่วงเวลาไปเกินกว่า 120 nanosec แล้วอาจจะทำให้พอร์ต I/O ได้รับข้อมูลที่ไม่ถูกต้องก็ได้ และสำหรับสัญญาณ $\overline{IOR}$ นั้น ถ้ามีการหน่วงเวลาเกิดขึ้นแล้ว ก็จะทำให้ความเร็วในการอ่านข้อมูลถูกลดลง

2.4.2.2 การดีโค้ดโดยใช้สวิทช์เลือก

การดีโค้ดในแบบ Fixed ที่ได้กล่าวไว้ในหัวข้อที่กล่าวมานั้น มีข้อเสียอยู่บางประการ คือ แอดเดรสที่เราเลือกให้งานไว้นั้น อาจซ้ำกับแอดเดรสของการ์ดอื่นที่เรานำมาเพิ่มเข้าไปในระบบในภายหลังก็ได้ ซึ่งกรณีเช่นนี้เราต้องแก้ไขวงจร เพื่อหลีกเลี่ยงไปใช้แอดเดรสที่ยังว่างอยู่ และไม่ถูกใช้งานโดยการ์ดที่จะเพิ่มเข้าไปใหม่ ซึ่งยุ่งยากและต้องเสียเวลามากขึ้น ปัญหานี้เราสามารถแก้ไขได้โดยใช้วงจรดีโค้ดที่สามารถเปลี่ยนแปลงค่าแอดเดรสได้ โดยเพียงแต่เปลี่ยนตำแหน่งของสวิทช์ (ในที่นี้คือ DIP Switch) ที่เซ็ตไว้ในวงจรเท่านั้น ดังรูปที่ 2.19



รูปที่ 2.19 วงจรตัวอย่างแสดงวงจรดีโค้ดแบบสวิทช์เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นวงจรที่ทำการดีโค้ดกลุ่มแอดเดรสขนาด 8 แอดเดรส ซึ่งการเลือกกลุ่มแอดเดรสที่จะทำการดีโค้ดนี้ จะทำได้โดยการเซ็ต DIP Switch ที่ขา Q0 - Q5 ของ 74LS688

สำหรับหน้าที่ของ 74LS688 นี้ จะทำการเปรียบเทียบค่าอินพุท 2ชุด ที่ถูกส่งเข้ามาทางขา P0 –P7 และขา Q0 – Q7ถ้าอินพุททั้ง 2 ชุดนี้เท่ากันแล้ว เอาท์พุทที่ขา $\overline{P=Q}$ จะทำให้เอาท์พุทเป็นลอกจิก " 0" จากในวงจรขา P0 –P6 ของ 74LS688 ต่อกับแอดเดรสบิท A3 - A9 ในขณะที่ขา Q0-Q5

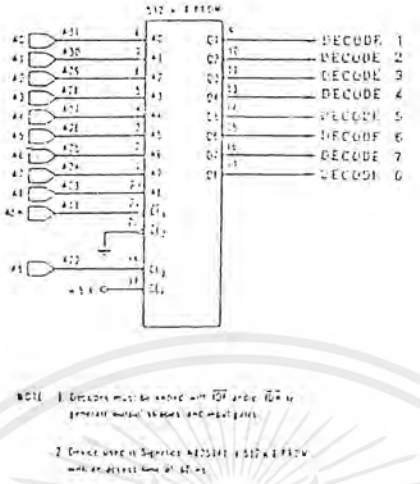
ต่อกับความต้านทานที่ทำหน้าที่เป็น Pull Up (รักษาระดับแรงดันให้เป็นลอจิก "0" ไว้ในกรณีที่ไม่มีอินพุทใด ๆ เข้ามา)และขา Q0 - Q5 นี้จะต่อกับปลายข้างหนึ่งของ DIP Switch ด้วยส่วนปลายอีกข้างหนึ่งของ DIP Switch จะต่อลง Ground (ลอจิก " 0") ไว้ ดังนั้นถ้าเราทำการ " ON" DIP Switch ที่ต่อกับขาใดขานั้นก็จะได้รับลอจิก "0" ในขณะที่ถ้า DIP Switch ที่ต่อกับขาใดถูก "OFF" ขานั้นก็จะได้รับลอจิก "1" และเนื่องจากอินพุทที่ขา P0-P5 นั้นเปลี่ยนแปลงตามไปด้วย จึงจะทำให้เอาท์พุทของ 74LS688 แอคทีฟได้ ทำให้เราสามารถเปลี่ยนค่าแอดเดรสที่จะทำการดีโค้ดได้ง่ายกว่าการดีโค้ดแบบFixed สำหรับขา Q6 นั้น จะต่อกับลอจิก "1" (+5V) และขา P6 ต่อกับแอดเดรสบิท A9 ในกรณีเช่นนี้จึงเท่ากับเป็นการบังคับให้แอดเดรสที่จะทำการดีโค้ดได้นั้น จะต้องมีแอดเดรสบิทA9 เป็น "1" เท่านั้น ส่วนขา P7 จะต่อกับสัญญาณ AEN โดยมีขา Q7 ต่อกับลอจิก "0" การต่อในลักษณะนี้ก็เพื่อป้องกันไม่ให้ 74LS688 ทำการดีโค้ดในระหว่างกระบวนการ DMA นั่นเอง เอาท์พุทจากขา $\overline{P=Q}$ ของ74LS688 นี้จะถูกนำไปใช้ในการอินาเบิล 74LS138 ทำหน้าที่ในการดีโค้ดแอดเดรสของกลุ่มแอดเดรสที่เราเลือก (โดยใช้ DIP Switch ดังที่ได้กล่าวในตอนต้น)

วงจรลักษณะนี้เราสามารถจะนำไปใช้เป็นการดีโค้ดในแบบ Fixed ได้โดยการนำเอา DIP Switch ออก จากนั้นถ้าอินพุทใดต้องการลอจิก "0" จึงจะใช้ตัวนำเชื่อมต่อระหว่างขั้วทั้งสองแทนการเซ็ต DIP Switch ให้ " ON" แต่ถ้าอินพุทใดต้องการให้ลอจิก "1" ก็ปล่อยขั้วทั้งสองนั้นไว้

2.4.2.3 การดีโค้ดโดยใช้ PROM

การดีโค้ดในแบบต่าง ๆ ที่กล่าวมาแล้วนั้น เป็นการดีโค้ดในลักษณะที่แอดเดรสของพอร์ตต่าง ๆ อยู่รวมกันเป็นกลุ่ม แต่ในบางกรณีพอร์ตที่เราใช้งานนั้นมีแอดเดรสแยกกันอย่างเป็นอิสระเช่นในการนำเอาหน้าที่การทำงานอยู่บนการ์ดต่าง ๆ มารวมไว้บนการ์ดเพียงการ์ดเดียว และมีความจำเป็นต้องคงค่าแอดเดรสของพอร์ตเดิม(ที่อยู่บนการ์ดเดิม) ไว้ด้วย ทำให้ไม่สามารถใช้การดีโค้ดในแบบต่าง ๆ ที่ผ่านมาได้ เนื่องจากการใช้วิธีการดีโค้ดแบบที่ผ่านมานั้นจะทำให้ต้องใช้อุปกรณ์ที่ทำการดีโค้มนั้นมากเกินไป ในกรณีเช่นนี้เราจำเป็นต้องใช้การดีโค้ดอีกแบบหนึ่ง ซึ่งจะได้กล่าวถึงในหัวข้อนี้ คือการดีโค้ดโดยใช้ PROM (Programmable Read Only Memory) ดังในรูปที่ 2.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 ตัวอย่างวงจรดีโค้ดโดยใช้ PROM

จากรูปข้างต้นเป็นวิธีการง่าย ๆ แบบหนึ่งในการดีโค้ดโดยใช้ PROM ซึ่งจะเห็นได้ว่าเราใช้เส้นแอดเดรส A0-A8 ของระบบต่อเข้ากับเส้นแอดเดรส A0-A8 ของ PROM และใช้บัสข้อมูลทั้ง 8 ของ PROM คือ Q1-Q8 เป็นเอาท์พุท สำหรับใช้เป็นสัญญาณดีโค้ดให้กับพอร์ตต่าง ๆ 8 พอร์ต อย่างไรก็ตามสัญญาณดีโค้ดทั้ง 8 เส้น คือ DECODE1-DECODE8 นี้ ยังคงต้องนำไป OR กับสัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ ก่อนจะนำไปอินาเบิลพอร์ตที่มีแอดเดรสตรงกับแอดเดรสที่ป้อนให้กับ PROM นั้น

จากที่ได้กล่าวมานั้นจะเห็นได้ว่าส่วนของวงจรดีโค้ดนั้นจะมี PROM เพียงตัวเดียวเท่านั้น ซึ่ง PROM ที่จะนำมรใช้งานนี้จะต้องถูกโปรแกรมมาก่อนแล้ว โดยข้อมูลที่โปรแกรมให้กับแอดเดรสต่าง ๆ ของ PROM นั้นจะต้องสัมพันธ์กับสัญญาณดีโค้ดที่เราต้องการ กล่าวคือเราต้องทราบเสียก่อนว่าค่าแอดเดรสของพอร์ตทั้ง 8 ที่เราต้องการจะดีโค้ดนั้นมีแอดเดรสใดบ้าง แล้วจึงกำหนดว่าพอร์ตใดจะใช้สัญญาณดีโค้ดเส้นใด จากนั้นจึงโปรแกรมข้อมูลให้กับ PROM โดยแอดเดรสใดต้องการให้สัญญาณดีโค้ดใดแอคทีฟ (ในที่นี้จะกำหนดให้สัญญาณดีโค้ดแอคทีฟที่ลอจิก “0”) ก็กำหนดให้ข้อมูลในบิทที่ตรงกับสัญญาณดีโค้ดนั้นเป็น “0” เช่นถ้าเรากำหนดให้แอดเดรสของพอร์ตที่เราต้องการจะดีโค้ดเป็น 0393Hและเลือกใช้สัญญาณ DECODE 5 เราก็จะต้องทำการโปรแกรมให้แอดเดรส 0193H ของ PROM (เหตุที่แอดเดรสของ PROM เป็น0193Hแทนที่จะเป็นแอดเดรส 0393H เหมือนกับแอดเดรสของพอร์ต ก็เพราะแอดเดรสของ PROM มีเพียง 9 บิท คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A0-A8 เท่านั้น ส่วนบิท A9 จะถูกต่อเข้ากับ PROM ภายหลังเพื่ออีนาเบิล PROM เมื่อข้อมูลในบิท A9 นี้เป็น " 1" เท่านั้น) มีข้อมูลในบิท Q5 (ถ้าเริ่มนับจากบิท D0 ก็คือบิท D4)เป็น "0" ส่วนบิทอื่น ๆ นั้นมีค่าเป็น "1"ทั้งหมด ดังนั้นการโปรแกรมแอดเอรส 01963H ของ PROM จึงต้องโปรแกรมด้วยข้อมูล 0EFH เป็นต้นสำหรับข้อมูลในแอดเดรสทั้ง 8 ที่กำหนดแล้ว จะต้องโปรแกรมให้ข้อมูลทุกบิทเป็น "1" ทั้งหมด ซึ่งก็คือโปรแกรมด้วยข้อมูล 0FFH นั่นเอง

ตัวอย่างเช่น

ถ้าแอดเดรสของพอร์ตทั้ง 8 ที่เราต้องการจะดีโค้ดเป็น 024A, 02B5 ,0317, 0361 , 0382, 03A8, 03C4 และ 03DB ในฐานสิบหกตามลำดับ โดยการกำหนดให้สัญญาณจากการดีโค้ดแอดเดรสเหล่านี้เป็นสัญญาณ DECODE1 จนถึง DECODE8 ตามลำดับแล้ว (เช่นสัญญาณจากการดีโค้ด 024AH ก็คือสัญญาณ DECODE1 และสัญญาณจากการดีโค้ด 02B5H ก็คือสัญญาณ DECODE2 เป็นต้น) เราจะต้องทำการโปรแกรม PROM ให้มีข้อมูลที่สัมพันธ์กับเอาท์พุทที่เราต้องการดังนี้

1.แอดเดรสของพอร์ตเป็นแอดเดรสที่บิท A9 ถูกใช้งานร่วมด้วย โดยในบิทนี้จะต้องมีข้อมูลเป็น "1" ในขณะที่แอดเดรสของ PROM จะมีเพียง 9 บิทก็คือ A0-A8 เท่านั้น เราจึงจัดแอดเดรสของ PROM เมื่อเทียบกับพอร์ตดังตารางที่ 2.6นี้

ตารางที่ 2.6แสดงการจัดแอดเดรส

แอสเดรสของพอร์ต (บิท A9 ถูกใช้งาน)	แอดเดรสของ PROM (เฉพาะบิท A0 - A9)
024A	04A
02B5	0B5
0317	117
0361	161
0382	182
03A8	1A8
03CA	1C4
03D4	1DB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ข้อมูลที่จะให้กับแอดเดรสทั้ง 8 ของ PROM จะต้องสัมพันธ์กับเอาท์พุทที่ต้องการ เช่น ถ้ามีการอ้างอิงถึงแอดเดรสของพอร์ต 02B5H แล้ว PROM จะต้องให้เอาท์พุทที่ลอจิก "0" ที่ขา Q2 (DECODE2) ส่วนเอาท์พุทที่ขาอื่นต้องเป็น "1" ดังนั้นจึงต้องมีการโปรแกรมให้แอดเดรส 00B5H ของ PROM มีข้อมูลเป็น 1111 1101 (ฐานสอง) หรือ 0FDH เป็นต้นสำหรับแอดเดรสอื่น ๆ นอกเหนือจากแอดเดรสทั้ง 8 นี้แล้วจะต้องถูกโปรแกรมให้มีข้อมูลเป็น 0FFH(ฐานสิบหก) ทั้งหมด ดังตารางที่ 2.7 นี้

ตารางที่ 2.7แสดงแอดเดรสอื่นๆ

แอดเดรส PROM(ฐานสิบหก)	ข้อมูล(ฐานสิบหก)
000-049	0FF
04A	0FE
04B-0B4	0FF
0B5	0FD
0B6-116	0FF
117	0FB
118-160	0FF
161	0F7
162-181	0FF
182	0EF
183-1A7	0FF
1A8	0DF
1A9-1C3	0FF
1C4	0BF
1C5-1DA	0FF
1DB	07F
1DC-1FF	0FF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามสิ่งสำคัญอีกสิ่งหนึ่งที่ต้องคำนึงถึงเสมอเมื่อใช้วิธีดีโค้ดแบบนี้คือ PROM ที่ใช้นั้นจะต้องใช้เวลาทำงานน้อยกว่า 92 nanosec ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การสร้างและการคำนวณ

3.1 วงจรอินเตอร์เฟสกับเครื่องคอมพิวเตอร์

3.1.1 หลักการติดต่อพอร์ต I/O ของ IBM PC

จากทฤษฎีในบทที่ 2 ได้กล่าวถึงการติดต่อพอร์ท I/O ของ IBM PC แล้วครั้งหนึ่งแต่เพื่อความเข้าใจมากขึ้นจึงขอกล่าวเพิ่มเติมในบทนี้ดังนี้ ในการที่จะทำการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก (I/O Device) นั้นเครื่องคอมพิวเตอร์จะทำโดยการติดต่อผ่านทางพอร์ต I/O ซึ่งพอร์ต I/O นั้นจะมีแอดเดรสโดยเฉพาะ คือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ภายในเครื่องคอมพิวเตอร์นั้นจะมีแอดเดรสใช้สำหรับพอร์ต I/O อยู่ทั้งหมด 64 K (65536) แอดเดรสโดยที่มีแอดเดรสบัสที่ใช้ติดต่อทั้งหมด 16 เส้น แต่ในเครื่อง IBM PC นั้นจะถูกออกแบบมาให้ใช้แอดเดรสบัสได้เฉพาะ 10 เส้นล่างเท่านั้น คือ A0 – A9 (A10 – A15 นั้นจะไม่ถูกนำไปใช้งาน) ดังนั้นจึงสามารถอ้างแอดเดรสได้สูงสุดเพียง 1024 แอดเดรส จาก 64 K ซึ่งพอร์ตทั้ง 1024 พอร์ตนี้จะถูกแบ่งออกเป็น 2 ส่วน ส่วนละ 512 พอร์ต ดังที่จะสามารถแสดงทั้ง 1024 ได้ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางแสดงแอดเดรสพอร์ท I/O ของIBM PC

000-00F	DMA chip 8237A-5	
020-021	Interrupt 8259A	
040-043	Timer 8253-5	
060-063	PPI 8255A-5	Assigned to
080-083	DMA page registers	system board
0Ax	NMI mark register	Components
0Cx	Reserved	
0Ex	Reserved	
100-1FF	Not usable	
200-20F	Game control	
210-217	Expansion unit	
220-24F	Reserved	
278-27F	Reserved	
2F0-2F7	Reserved	
2F8-2FF	Asynchronous communications (2)	
300-31F	Prototype card *	
320-32F	Fixed disk	
378-37F	Printer	Assigned to
380-38C	SDLC communications	feature card
380-389	Binary synchronous communications (2)	Ports
3A0-3A9	Binary synchronous communications (1)	
3B0-3BF	IBM monochrome display/printer	
3C0-3CF	Reserved	
3D0-3DF	Colorigraphics	
3E0-3EF	Reserved	
3F0-3F7	Diskette	
3F8-3FF	Asynchronous communications (1)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.1 นั้นจะเห็นได้ว่าพอร์ต 1024 พอร์ตถูกแบ่งเป็น 2 ส่วน นั้นคือ

- ส่วนที่ 1 แอดเดรสมีค่า 000H – 1FFH เป็นส่วนของพอร์ตที่อยู่บนเมนบอร์ด
- ส่วนที่ 2 แอดเดรสมีค่า 200H – 3FFH เป็นส่วนของพอร์ตที่จะใช้งานอยู่บนการ์ดที่ใช้เสียบใน ซล็อต ต่าง ๆ ของ IBM PC

ซึ่งทั้ง 2 ส่วนนี้จะถูกกำหนดการแบ่งด้วยข้อมูลในบิท A9 คือถ้า A9 มีข้อมูลเป็น "0" จะทำการติดต่อกับพอร์ตในส่วนที่ 1 (มีแอดเดรส 000H – 1FFH) เท่านั้น (ติดต่อกับ เมนบอร์ด เท่านั้น) แต่ถ้า A9 มีข้อมูลเป็น "1" ก็จะทำการติดต่อกับพอร์ตในส่วนที่ 2 (มีแอดเดรส 200H – 3FFh) เท่านั้น (ติดต่อกับการ์ดที่เสียบอยู่ในซล็อตเท่านั้น)

สำหรับในโครงงานนี้จะทำการเลือกใช้แอดเดรสพอร์ตช่วง 300H – 31FH ซึ่งจากตารางที่ 3.1 เป็นช่วงของ Prototype Card มาใช้เป็นแอดเดรสสำหรับที่จะใช้ในการอ้างถึง Interface Card ต่อไปและส่วนของ Slot ที่จะใช้สำหรับต่อกับ Card ต่าง ๆ นั้นสามารถแสดงได้ดังรูปที่ 3.1

รูปที่ 3.1 ระบบบัสของ IBM PC (ซล็อต ของ IBM PC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 จะขออธิบายส่วนประกอบของ สล๊อตเสริม(Expansion Slots) เฉพาะส่วนที่นำไปใช้งานเท่านั้น ว่าแต่ละขามีหน้าที่อย่างไรบ้างดังจะอธิบายได้ดังนี้

- RESET DRV(ขา B2)

ขาสัญญาณนี้เป็นเอาท์พุท ซึ่งจะแอคทีฟ (ลอจิก "1") ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอคทีฟไปจนกว่าระบบต่าง ๆ ภายใน IBM PC จะพร้อมที่จะทำงานได้ จากนั้นสัญญาณนี้ก็จะเปลี่ยนกลับเป็นลอจิก "0" นอกจากนี้ในระหว่างการทำงานของ IBM PC ถ้าระดับของแรงดันแหล่งจ่ายไฟตกลง สัญญาณนี้จะถูกทำให้แอคทีฟร่วมกัน

- A0 – A19 (Address Bus; ขา A31 – A12)

สัญญาณทั้ง 20 ขานี้เป็นเอาท์พุท ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำหรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อด้วย โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (LSB – Least Significant Bit) และ A19 จะมีนัยสำคัญสูงสุด (MSB – Most Significant Bit) สำหรับค่าแอดเดรสบนบัสแอดเดรส A0- A19 นี้จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่านเขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA – Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ)

- D0 – D7 (Data Bus ; ขา A9 –A2)

ขาสัญญาณนี้จะเป็นแบบ Bi – Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ต I/O กับ IBM/PC โดยบิท D0 จะมีนัยสำคัญต่ำสุด (LSB) และ บิท D7 จะมีนัยสำคัญสูงสุด (MSB)

- $\overline{\text{IOR}}$ (I/O Read; ขา B14)

ขาสัญญาณนี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ต I/O เพื่อให้พอร์ต I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอบขาขึ้นของสัญญาณ IOR ประมาณ 30 ns เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง

- $\overline{\text{IOW}}$ (I/O Write ; ขา B13)

ขาสัญญาณนี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0' ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ต I/O เพื่อให้พอร์ต I/O ที่มีแอดเรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น รับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- AEN (Address Enable; ขา 11)

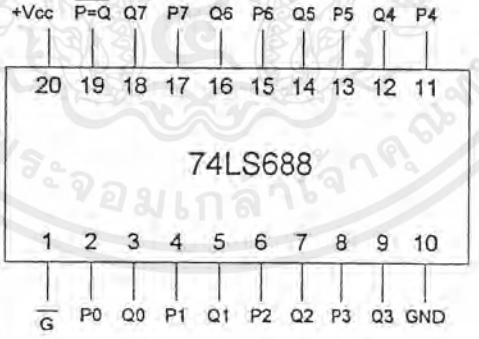
สัญญาณนี้เป็นเอาท์พุทที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณAEN แอคทีฟ (ลอจิก "1") นั้น เป็นบัสไซเคิลของขบวนการ DMA

- บัสของแหล่งจ่ายไฟของระบบ

แหล่งจ่ายไฟของระบบจะประกอบด้วยแหล่งจ่ายไฟ DC +5V,+12 V, -5 V , -12 V ซึ่งประกอบไปด้วยขา B3, B29, B9, B5 และ B7 ส่วนสัญญาณ GND จะประกอบไปด้วยขา B1, B10 และ B31 ซึ่งขาทั้งสามนี้จะต่อกับกราวด์(Ground) ของระบบ

3.1.2 เทคนิคการดีโค้ดแอดเดรสสำหรับพอร์ต I/O

จากทฤษฎีในบทที่ 2 นั้นได้กล่าวเทคนิคการดีโค้ดแอดเดรสสำหรับพอร์ต I/O ไว้หลายแบบด้วยกันได้แก่การดีโค้ดแบบ Fixed , แบบใช้สวิตช์เลือก และแบบใช้ PROM จะขอกล่าวถึงเฉพาะหลักการที่นำมาใช้ในการออกแบบในปริญญานิพนธ์นี้อีกครั้งเพื่อที่จะได้เกิดความเข้าใจมากขึ้นนั่นคือการดีโค้ดแบบใช้สวิตช์เลือกเทคนิคการดีโค้ดใช้สวิตช์เลือก จะเป็นการดีโค้ดแบบ Fixed ซึ่งอาจจะทำให้ได้แอดเดรสซ้ำกับ Card อื่นที่นำมาเสียบไว้ในระบบจะสามารถแก้ไขได้โดยการสับ DIP SW. เปลี่ยนแอดเดรสไปใช้ในตำแหน่งที่ว่างอยู่ ซึ่งในโครงงานนี้ตำแหน่งแอดเดรสของพอร์ตที่จะเลือกใช้จะอยู่ที่ 300H – 31FH การทำการดีโค้ดแบบนี้นั้นจะใช้ไอซีเปรียบเทียบระหว่างอินพุท 2 ชุดที่ถูกส่งเข้ามาโดยใช้ไอซี 74LS688 (8 bit comparator) ซึ่งเป็น Excusive-OR ดังที่จะสามารถแสดงตำแหน่งของขาต่างๆได้ดังรูปที่ 3.2

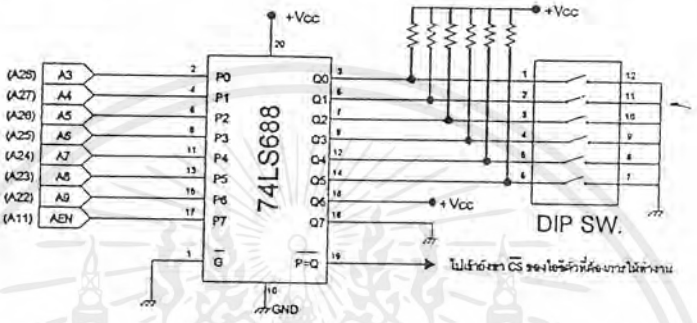


รูปที่ 3.2 แสดงตำแหน่งขาของไอซี 74LS688

มีหลักการทำงานดังนี้ พิจารณารูปที่ 3.3 ไอซี 74LS688 จะทำการเปรียบเทียบค่าอินพุท 2 ชุดที่ถูกส่งเข้ามาทางขา P0-P7 และขา Q0-Q7 ถ้าอินพุท 2 ชุดนี้เท่ากันแล้วเอาท์พุทที่ขา P=Q จะมีค่าเป็น logic "0" จากวงจรขา P0-P6 ต่อกับแอดเดรสบิท A3-A9 ในขณะที่ขา Q0-Q5 ต่อกับ R Pull up และต่ออยู่กับ DIP SW. ส่วนปลายอีกข้างหนึ่งของ DIP SW. ต่อลง GND (ลอจิก "0") ดังนั้นถ้า ON SW. ตัวใดขานั้นก็จะรับ ลอจิก "0" ในทางตรงกันข้ามถ้า OFF SW. ตัวใดขานั้นก็จะรับ ลอจิก "1" และเนื่องจากอินพุทที่ขา P0-P6 (A3-A9) ต้องมีค่าเท่ากับอินพุทที่ขา Q0-Q5 ดังนั้นถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

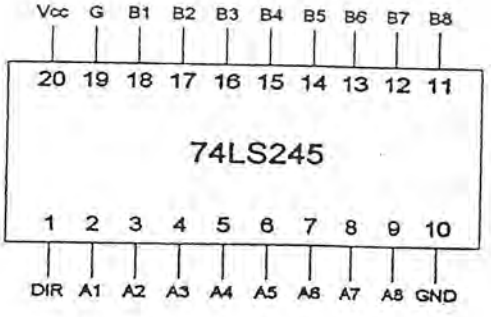
เปลี่ยนแปลงค่าที่ Set ไว้ที่ DIP SW. ก็จะทำให้แอดเดรส A3-A5 จะต้องเปลี่ยนแปลงตามไปด้วย เพื่อให้เอาท์พุทของ 74LS688 ทำงานได้ สำหรับขา Q6 จะต่อกับ ลอจิก "1" และขา P6 จะต้องต่อกับแอดเดรสบิท A9 เป็น "1" เท่านั้น และขา P7 จะต่ออยู่กับ AEN โดยที่ขา Q7 ต่ออยู่กับ logic "0" เพื่อที่จะเป็นการป้องกันไม่ให้เกิดการดีโค้ดในระหว่างขบวนการ DMA จากนั้นจะนำขาเอาท์พุทของ 74LS688 (ขา P=Q) ไปต่อเข้ากับสัญญาณ ชิพซีเล็กของอุปกรณ์ที่ต้องการให้ทำงานที่แอดเดรสนี้ ส่วนในการที่จะเปลี่ยนค่าแอดเดรสนั้นสามารถทำได้ง่าย โดยการเปลี่ยนตำแหน่งของสวิตช์ (DIP SW.)



รูปที่ 3.3 วงจรดีโค้ดโดยใช้สวิตช์เลือก

3.1.3 INTERFACE CARD

การที่จะต่อ 8255 เข้ากับคอมพิวเตอร์นั้นต้องใช้วงจรชุดดีโค้ดซึ่งในที่นี้เลือกใช้สวิตช์เลือก โดยมีส่วนประกอบคือ DIP SW. และไอซี 74LS688 โดยมีหลักการทำงานตามที่ได้กล่าวมาแล้วในหัวข้อ 3.1.2 ขาที่เอาท์พุทของ 74LS688 จะถูกต่อเข้ากับขา CS ของ 8255 นอกจากนี้ 74LS688 ยังถูกควบคุมการทำงานโดยคอมพิวเตอร์ โดยจะนำสัญญาณ AEN มาต่อที่ขา 1 ของ 74LS688 ซึ่งจะ Active low ซึ่งรูปวงจรของ INTERFACE CARD จะถูกแสดงไว้ในรูปที่ 3.4 โดยภายในวงจรจะมีไอซีเพิ่มขึ้นมา 1 ตัวนั่นคือ 74LS245 (Buffer 2 ทิศทาง) ซึ่งจะสามารถแสดงรูปของ 74LS245 ได้ดังรูปที่ 3.4



รูปที่ 3.4 แสดงตำแหน่งขของไอซี 74LS245

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 วงจรการทำงานของ Interface card

ภายในการ์ดจะประกอบด้วยวงจร 2 ส่วน คือ

1. วงจรดีโค้ด
2. ส่วนของ CHIP SUPPORT

วงจรดีโค้ด

จะใช้ไอซี เบอร์ 74LS688 จะทำการดีโค้ดสัญญาณแอดเดรสจากสล็อตของคอมพิวเตอร์มาเพื่อใช้แอดเดรสให้กับ 8255 สัญญาณที่จะนำมาใช้ในการดีโค้ดมี A0-A9 แต่ 8255 ต้องการสัญญาณจาก A0 และ A1 ไปควบคุมรีจิสเตอร์ภายในตัวมัน ดังนั้นสัญญาณที่สามารถนำมาใช้ได้จึงมีแค่ A2-A0 เท่านั้น ส่วน A0-A1 นั้นจะถูกต่อเข้ากับขา A0 และ A1 ของ 8255 โดยตรง

จากวงจรในรูปที่3.5จะสังเกตว่าจะนำแต่สัญญาณA0-9,AEN,IOR,IOW,RESET มาใช้เท่านั้น ส่วนสัญญาณอื่นๆ ไม่นำมาใช้

การทำงานของ 74LS688 ดังที่ได้กล่าวมาแล้วนั้น ไอซีนี้จะทำการเปรียบเทียบสัญญาณอินพุท 2 ส่วน คือ P0-P7 และ Q0-Q7 จะให้เอาท์พุทออกมาเป็น logic "0" เมื่อ P=Q' สัญญาณ Q จะถูกต่ออยู่กับ DIP SW. ในการเลือกแอดเดรสของการ์ดเลือกโดยตั้งสวิตซ์ ซึ่งในโครงงานนี้จะใช้แอดเดรสช่วง 300H – 31FHเมื่อมีแอดเดรสค่า 300H มาเข้าที่อินพุทของ 74LS668 ที่เอาท์พุทจะให้เอาท์พุทเป็น "0" ทำให้ CS ของ 8255 จะต่อ Data Bus ของตัวมันเองเข้ากับ Data Bus ของคอมพิวเตอร์ สำหรับแอดเดรสที่ใช้มี 4 ตำแหน่งคือ

300H ควบคุมการทำงานของพอร์ต A

301H ควบคุมการทำงานของพอร์ต B

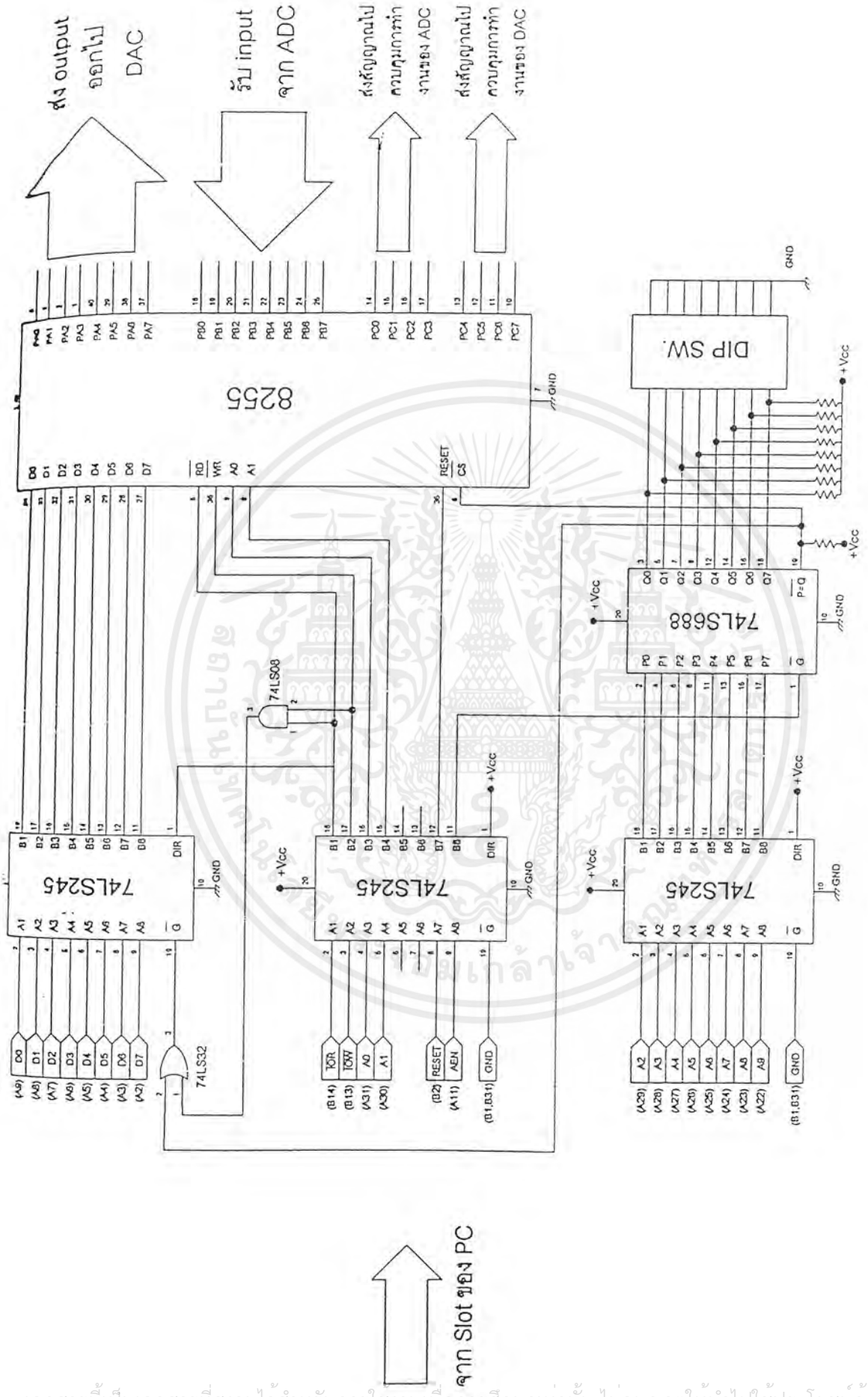
302H ควบคุมการทำงานของพอร์ต C

303H ส่ง Control word ให้กับ 8255

การใช้งาน 8255 Interface card จะต้องเริ่มต้นที่ตั้งแอดเดรสของการ์ดก่อนโดจที่จะทำการตั้งที่ DIP SW. ซึ่งการตั้งแอดเดรสนั้นต้องตั้งบิท A8 และ A9 เป็น "1" เพื่อให้ได้ค่าเป็น 300H สำหรับ A0,A1 จะนำไปใช้ในการตั้ง Control word ให้กับ 8255 โดยที่จะใช้โปรแกรมเป็นตัวตั้งค่า สำหรับค่าแอดเดรส A2-A9 จะทำการตั้งค่าดังต่อไปนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

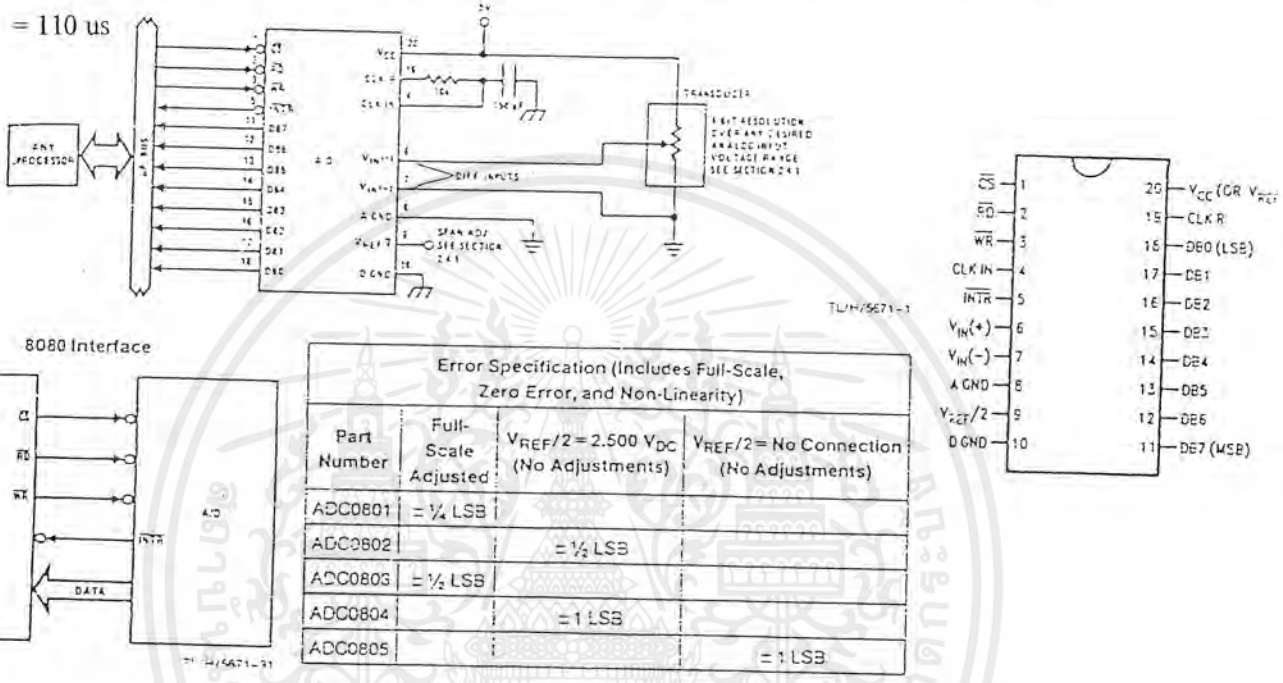


รูปที่ 3.5 แสดงการต่อวงจรพอร์ท 8255 ของ Interface card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเลือกใช้ไอซี ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลมาใช้งานในวงจร (เลือก ADC0805) และวงจรที่ใช้งาน

ADC0805 เป็นไอซีวงจร ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล 8 บิท แบบประมาณค่ามีวงจรแลตช์ภายในซึ่งเหมาะสำหรับต่อเข้ากับ Data Bus ของไมโครโปรเซสเซอร์ มี Conversion Time = 110 us



Part Number	Full-Scale Adjusted	Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity) $V_{REF}/2 = 2.500\ V_{DC}$ (No Adjustments)	$V_{REF}/2$ = No Connection (No Adjustments)
ADC0801	± ¼ LSB		
ADC0802		± ½ LSB	
ADC0803	± ½ LSB		
ADC0804		± 1 LSB	
ADC0805			± 1 LSB

รูปที่ 3.6 แสดงลักษณะการต่อ ADC0805 เข้ากับไมโครโปรเซสเซอร์

ความหมายของขาต่างๆของ ADC0805

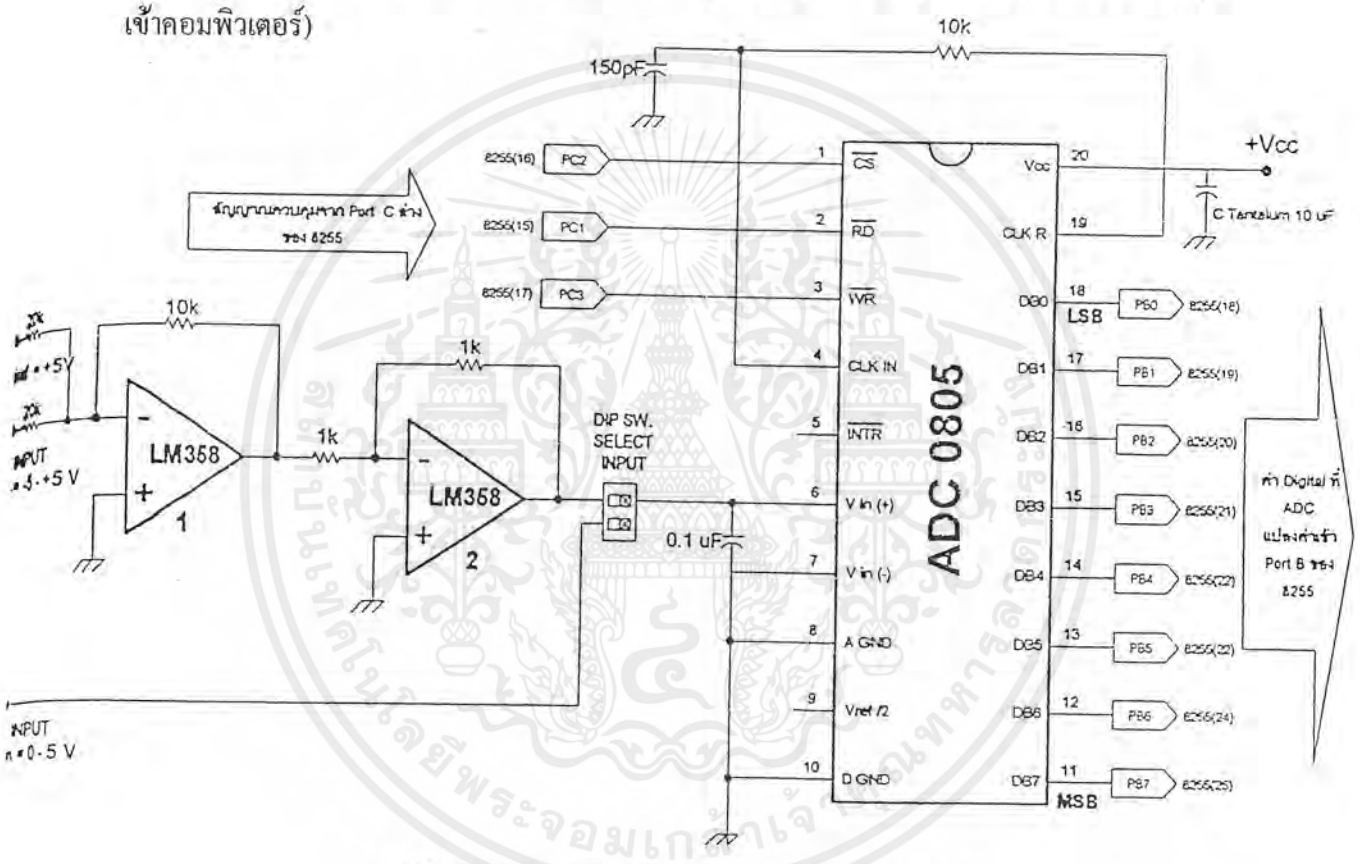
- $\overline{CS}$ Chip Select (Active Low) ไมโครโปรเซสเซอร์จะต้องให้ค่า 0 กับขานี้ เมื่อต้องการติดต่อกับ ADC0805
- $\overline{WR}$ Write (Active Low) ถ้าสัญญาณเป็น 0 ADC0805 จะทำการ Sampling ค่ามาเก็บไว้
- $\overline{RD}$ Read (Active Low) ถ้าสัญญาณเป็น 0 ADC0805 จะส่งค่าที่ทำการแปลงเป็นดิจิตอลแล้วให้กับไมโครเซสเซอร์
- V_{cc} Digital Supply Voltage เป็นขาที่ต่อกับแหล่งจ่ายไฟให้กับ A/D ซึ่งมีขนาด 5V
- $V_{ref/2}$ ต่อกับค่า $V_{ref/2}$ ซึ่ง A/D จะใช้เพื่อนำไปเปรียบเทียบกับอินพุทแต่สำหรับไอซี ADC0805 ไม่จำเป็นต้องต่อขานี้
- DB0-DB7 เป็นเอาท์พุทพอร์ตของข้อมูล ต่อเข้ากับบัสข้อมูลของไมโครโปรเซสเซอร์ โดย DB0 เป็น LSB (Least Significant Bit) DB7 เป็น MSB (Most Significant Bit)
- A GND และ D GND เป็น Ground ของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $V_{in}(+)$ และ $V_{in}(-)$ เป็นขาอินพุท
- CLK R และ CLK IN ต่อกับวงจรให้กำเนิดสัญญาณนาฬิกาของA/D

การต่อวงจรใช้งานของ ADC0805

จากรูปที่ 3.6 ได้นำ ADC0805 ไปทำการต่อใช้งานโดยที่จะทำการต่อลงอินเตอร์เฟสการ์ด ซึ่งจะติดต่อกับคอมพิวเตอร์โดยผ่านทางไอซี 8255 โดยจะต่อวงจรดังรูปที่ 3.8 และวงจรนี้เป็นส่วนหนึ่งของอินเตอร์เฟสการ์ดซึ่งจะมีส่วนไปติดต่อกับรูปที่ 3.4 (ส่วนสัญญาณควบคุม และส่วนข้อมูลเข้าคอมพิวเตอร์)



รูปที่ 3.7 แสดงวงจรการนำไปใช้งานของ ADC0805

จากวงจรข้างต้นนี้ได้มีการบันทึกข้อมูลเพื่อที่จะนำไปใช้ช่วยในการเขียนโปรแกรมดังตารางที่ 3.2 และ 3.3

ตารางที่ 3.2 แสดงผลการแปลงค่าของADC0805 เมื่อใส่อินพุท 0-5 โวลท์

Input Voltage	Digital Output from ADC 0805 MSB LSB
0.00	0000 0000
0.25	0000 1100
0.50	0001 1001
0.75	0010 0110
1.00	0011 0011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ต่อ แสดงผลการแปลงค่าของADC0805 เมื่อใส่อินพุท0-5 โวลท์

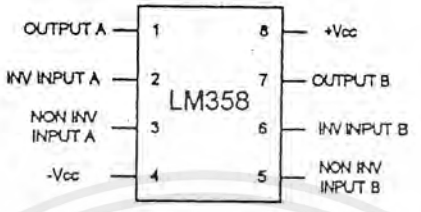
Input Voltage	Digital Output from ADC 0805 MSB LSB
1.25	0011 1111
1.50	0100 1100
1.75	0101 1001
2.00	0110 0101
2.25	0111 0010
2.50	0111 1111
2.75	1000 1011
3.00	1001 1000
3.25	1010 0101
3.50	1011 0010
3.75	1011 1110
4.00	1100 1011
4.25	1101 0111
4.50	1110 0100
4.75	1111 0001
5.00	1111 1111

ตารางที่ 3.3 แสดงการแปลงค่าของ ADC0805 เมื่อใส่อินพุท –5 -+5 โวลท์

Input Voltage	Digital Output from ADC 0805 MSB LSB
-5.05	0000 0000
-4.50	0000 1111
-4.00	0001 1011
-3.50	0010 1000
-3.00	0011 0101
-2.50	0100 0001
-2.00	0100 1110
-1.50	0101 1011
-0.50	0111 0100
0.00	1000 0000
0.50	1000 1101
1.00	1001 1000
1.50	1010 0110
2.00	1011 0010
2.50	1011 1111
3.00	1100 1100
3.50	1101 1000
4.00	1110 0101
4.50	1111 0001
5.00	1111 1111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

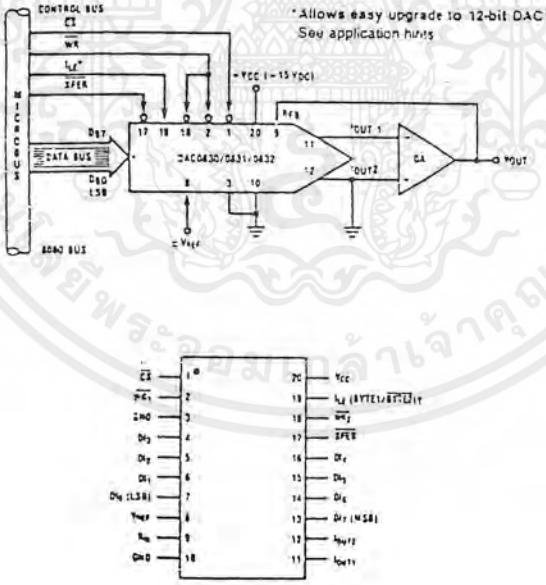
จากวงจรในรูปที่ 3.8 นั้นได้มีการใช้ไอซี LM358 มาเป็นวงจร Zero Span (จากรูปที่ 3.8 นั้นเป็นออปแอมป์ LM358 ตัวที่ 1 จะทำหน้าที่ปรับ Zero Span ส่วนตัวที่ 2 นั้นทำหน้าที่เป็นอินเวิร์ทเตอร์ทำขึ้นเพื่อที่จะทำการปรับค่าอินพุทให้สามารถรับค่าได้ทั้ง –5 และ +5 Volt และ 0-5 Volt ดังนั้นจึงได้แสดงรูปการจัดวางขาของไอซี LM358 ไว้ดังรูป 3.8



รูปที่ 3.8 แสดงขาของไอซี

3.3 การเลือกไอซี ตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก มาใช้งานในวงจร (เลือก DAC0823) และวงจรที่ใช้งาน

DAC0832 เป็นไอซีวงจร D/A 8 บิท ที่ใช้สำหรับต่อกับไมโครโปรเซสเซอร์แบบต่างๆ โดยที่จะต้องมี V_{ref} ต่อจากภายนอกและเป็น D/A แบบ R-R2 Ladder



รูปที่ 3.9แสดงวงจรการใช้งานของ DAC0832

ความหมายของขาต่างๆขอ DAC0832

- V_{ref} Reference Voltage Input ต่อกับแหล่งจ่ายไฟภายนอกเพื่อใช้อ้างอิงของวงจร R-R2 Ladder โดยมีค่าได้ระหว่าง +10 ถึง –10 Volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Vcc เป็นขาที่ต่อกับแหล่งจ่ายไฟ โดยเลือกค่าระหว่าง +5 ถึง –5 Volt
- I_{out1} และ I_{out2} เป็นขาเอาท์พุทของวงจร
- GND เป็นขากราวน์ของวงจร
- DI0-DI7 เป็นขาอินพุท 8 บิทของวงจร
- R_{fb} Feedback Resistor เป็น Shunt ของออปแอมพ์เพื่อทำให้มีค่า Voltage ออกมา
- $\overline{CS}$ Chip Select (Active Low) เป็นขาที่ใช้เลือกให้ชิปทำงาน
- ILE Input Latch Enable (Active High) ร่วมกับ $\overline{CS}$ เพื่อใช้ในการควบคุม WR1
- $\overline{WR1}$ ถ้ามีเป็น 0 DI จะถูกคงค่าไว้ในอินพุทแลทช์ ถ้ามีค่าเป็น 1 จะทำการแลตช์ค่าใหม่เข้ามา
- $\overline{WR2}$ (Active Low) ใช้ร่วมกับ $\overline{XFER}$ เพื่อให้ค่าที่อยู่ในอินพุทแลทช์ส่งไปที่ DAC Register
- $\overline{XFER}$ Tranfer Control Signal (Active Low) ใช้เพื่อควบคุม $\overline{WR2}$

การต่อวงจรใช้งานของ DAC0832

จากรูปที่ 3.9 ได้นำเอา DAC0832 ไปทำการต่อใช้งานโดยที่จะทำการต่อลงบนอินเตอร์เฟสการ์ด ซึ่งจะติดต่อกับคอมพิวเตอร์โดยผ่านทางไอซี 8255 โดยจะต่อวงจรดังรูปที่ 3.10 และวงจรนี้เป็นส่วนหนึ่งของอินเตอร์เฟสการ์ดซึ่งจะมีส่วนไปติดต่อกับรูปที่ 3.4 (ส่วนสัญญาณควบคุมและส่วนข้อมูลออกจากคอมพิวเตอร์)

จากวงจรรูปที่ 3.10 นี้ได้มีการบันทึกข้อมูบเพื่อที่จะนำไปใช้ช่วยในการเขียนโปรแกรมดังตาราง 3.4

ตารางที่ 3.4 แสดงผลการแปลงค่าของDAC0832

Binary Input B1 B2 B3 B4 B5 B6 B7 B8	Voltage output (0-5 Volt)	Voltage output (+5, -5 Volt)
1 1 1 1 1 1 1 1	5.00	5.03
1 1 1 1 1 1 1 0	4.99	5.02
1 1 1 1 1 1 0 0	4.99	4.95
1 1 1 1 1 0 0 0	4.92	4.80
1 1 1 1 0 0 0 0	4.76	4.48
1 1 1 0 1 0 0 0	4.60	4.16
1 1 1 0 0 0 0 0	4.44	3.83
1 1 0 0 1 0 0 0	3.96	2.87
1 1 0 0 0 0 0 0	3.80	2.55
1 0 1 0 1 0 0 0	3.33	1.58

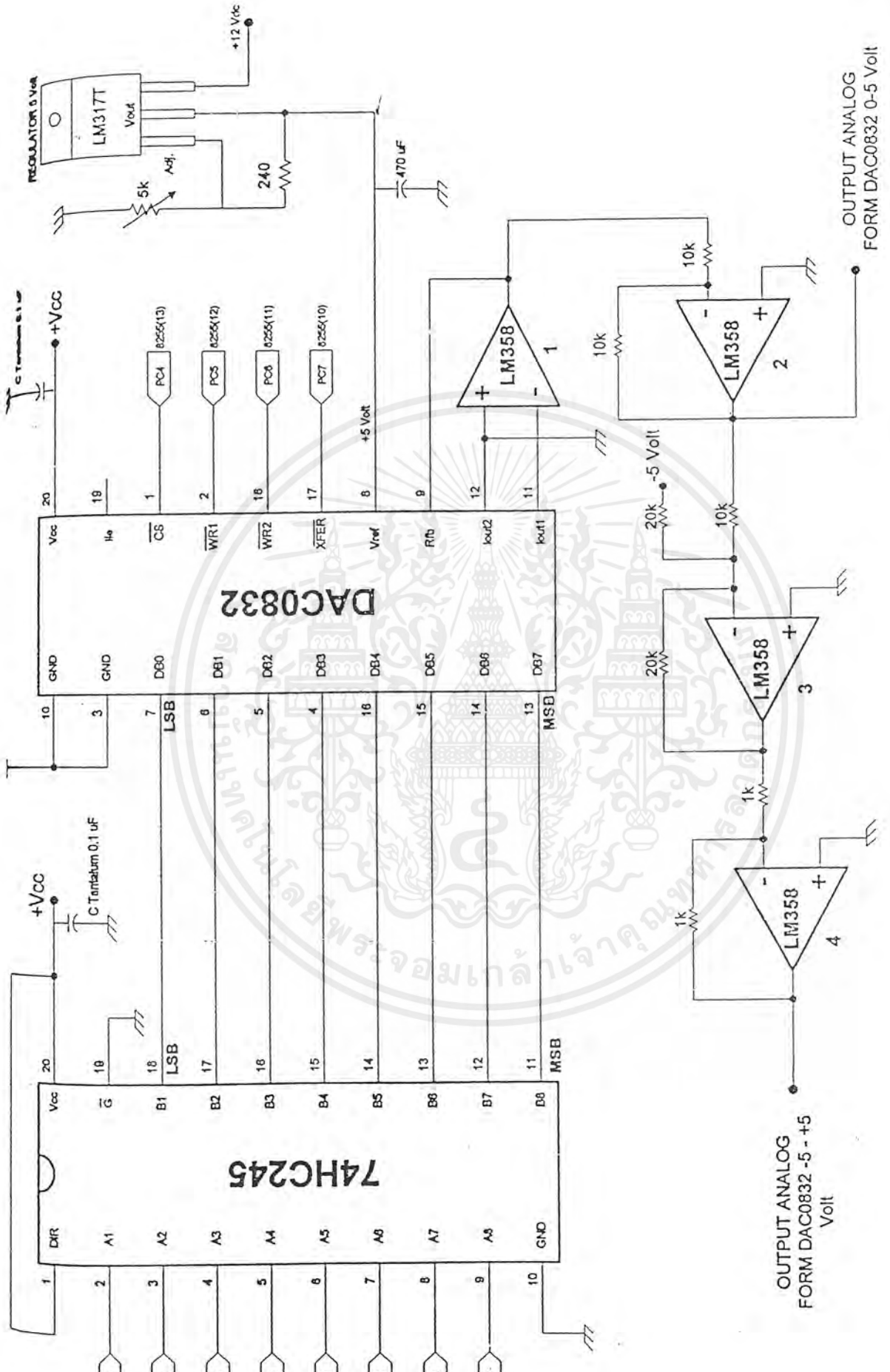
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4(ต่อ) แสดงผลการแปลงค่าของDAC0832

Binary Input B1 B2 B3 B4 B5 B6 B7 B8	Voltage output (0-5 Volt)	Voltage output (-5 ,+5 Volt)
1 0 1 0 0 0 0 0	3.17	1.26
1 0 0 0 0 0 0 0	2.53	-0.02
0 1 1 1 1 1 1 1	2.50	-0.09
0 1 0 0 1 1 1 1	1.56	-1.99
0 0 1 1 1 1 1 1	1.25	-2.62
0 0 1 1 0 0 0 0	0.95	-3.27
0 0 1 0 0 0 0 0	0.63	-3.88
0 0 0 1 1 1 1 1	0.63	-3.89
0 0 0 1 0 1 0 0	0.39	-4.38
0 0 0 1 0 0 0 0	0.31	-4.53
0 0 0 0 1 1 1 1	0.31	-4.56
0 0 0 0 1 0 0 0	0.15	-4.85
0 0 0 0 0 1 1 1	0.14	-4.89
0 0 0 0 0 0 1 1	0.05	-5.05
0 0 0 0 0 0 1 0	0.03	-5.09
0 0 0 0 0 0 0 1	0.02	-5.13
0 0 0 0 0 0 0 0	0.00	-5.17

จากรูปที่ 3.10 จะเห็นได้ว่าวงจรที่ใช้งานจริงนั้นจะมีส่วนของ Regulator ประกอบอยู่ด้วย โดยที่จะทำหน้าที่เป็นตัวจ่ายแรงดันให้แก่ DAC0832 ซึ่ง Regulator ที่นี้จะใช้ไอซีเบอร์ LM317T ซึ่งส่วนของวงจรนั้นสามารถดูได้จากรูปที่ 3.10 และในวงจรนี้ยังมีการใช้วงจร Zero Span โดยใช้ออปแอมป์เบอร์ LM358 F(โดยที่จากรูป 3.10 ออปแอมป์ LM358 ตัวที่ 3 จะทำหน้าที่เป็นตัวปรับ Zero Span และออปแอมป์ LM358 ตัวที่ 4 จะทำหน้าที่เป็นตัวอินเวิร์ตเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงวงจรการนำไปใช้งานของ DAC0832

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การพัฒนาทางด้านฮาร์ดแวร์

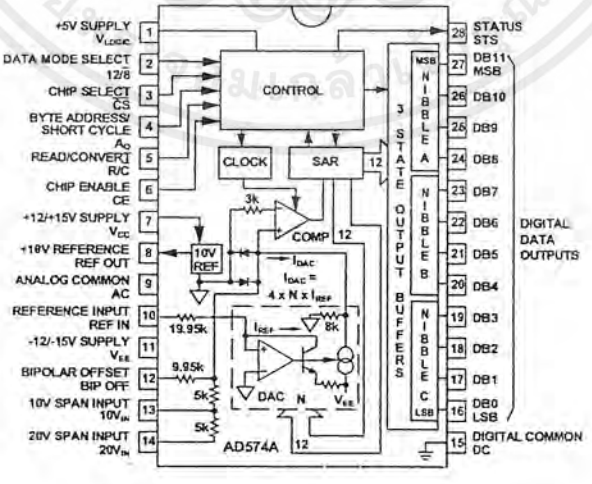
ในส่วนของฮาร์ดแวร์นี้เราสามารถพัฒนาให้ใช้A/D หรือ D/A ที่มีขนาดมากกว่า 8 บิทได้ เพื่อเพิ่มความละเอียดของการคำนวณและค่าที่ออกมาให้ตรงกับความเป็นจริงมากที่สุดโดยในที่นี้เราพัฒนาโดยใช้ไอซีขนาด 12 บิทเป็นตัวพัฒนาฮาร์ดแวร์ ซึ่งสามารถแสดงได้ดังนี้

3.4.1 การติดต่อกับคอมพิวเตอร์

การติดต่อกับคอมพิวเตอร์จะเหมือนกับในแบบ 8 บิทคือใช้ 8255 เป็นตัวขยายพอร์ตซึ่งในที่นี้จะใช้ 8255 จำนวน 2 ตัวในการส่งค่าเพื่อไปประมวลผล โดยในส่วนของการอินเตอร์เฟสจะเหมือนกันกับแบบ 8 บิทคือใช้แบบสวิตช์เลือกเหมือนกันโดยใช้ไอซี 74 LS688 เป็นตัวเปรียบเทียบค่าของอินพุท 2 ชุดเพื่อให้เกิดสัญญาณการเลือกใช้ 8255 ซึ่งในการเลือกใช้ตัว 8255 นี้เราจะส่งสัญญาณไปเลือกพร้อมกันทีเดียวทั้ง 2 ตัวเพื่อลดขนาดของวงจร โดยจะให้พอร์ต A เป็นเอาท์พุทพอร์ตเสมอเพื่อส่งค่าดิจิตอลให้กับตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก และให้พอร์ต B เป็นอินพุทพอร์ตเสมอเพื่อส่งสัญญาณดิจิตอลจากตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลให้กับคอมพิวเตอร์ ซึ่งสามารถแสดงได้ดังรูปที่ 3.12

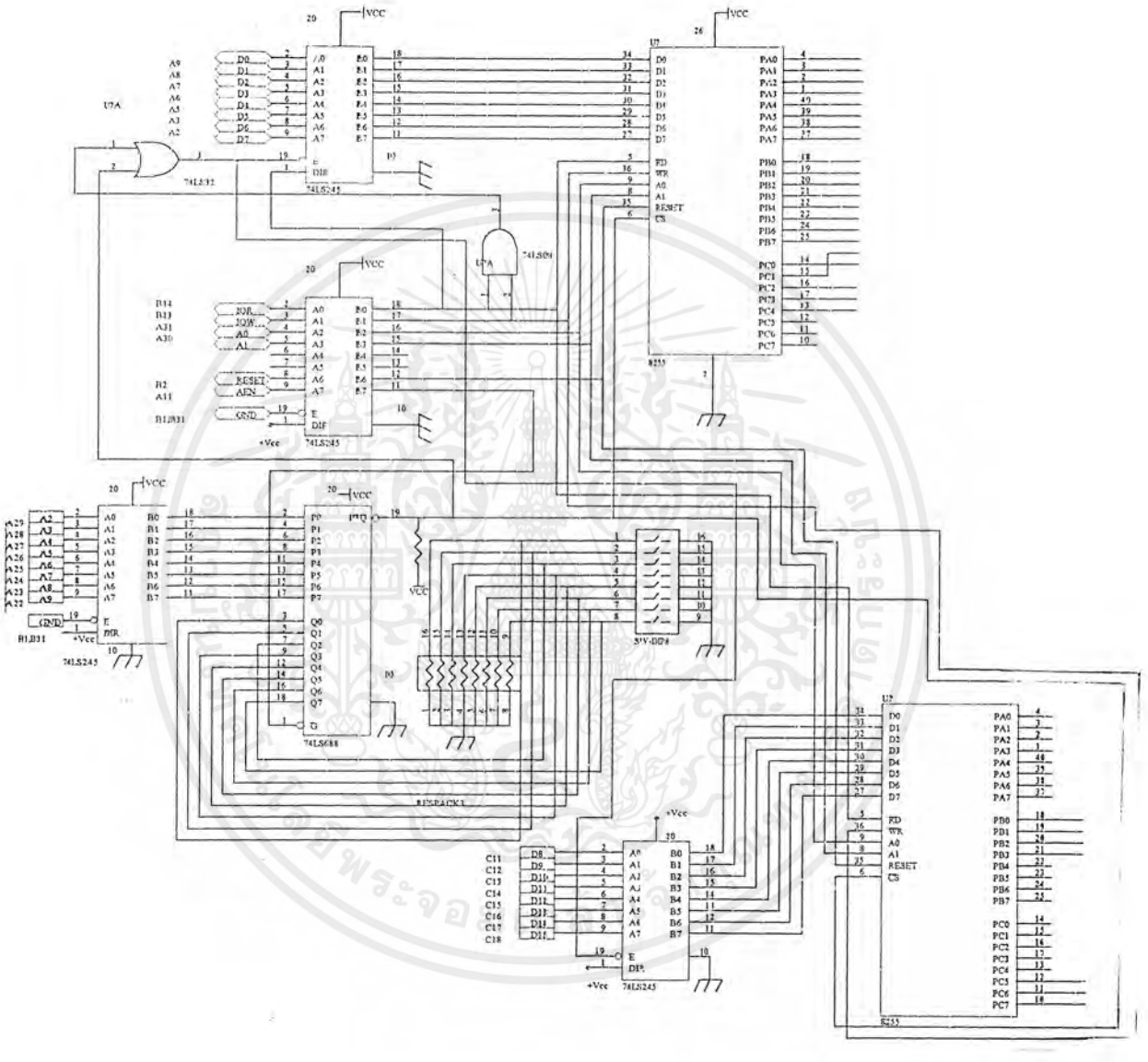
3.4.2 การเลือกตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลมาใช้ในวงจร(เลือกADC574A)

ตัวADC574A เป็นไอซีวงจรตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล 12 บิท สามารถต่อกับไมโครโปรเซสเซอร์โดยตรงได้ทั้ง 8 และ 12 บิท เป็นแบบประมาณค่าซึ่งมีแรงดันอ้างอิงและสัญญาณนาฬิกาภายใน ซึ่งสามารถแสดงได้ดังรูปที่ 3.11



รูปที่ 3.11 แสดงขาต่าง ๆ ของไอซี ADC574A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการต่อวงจรการอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

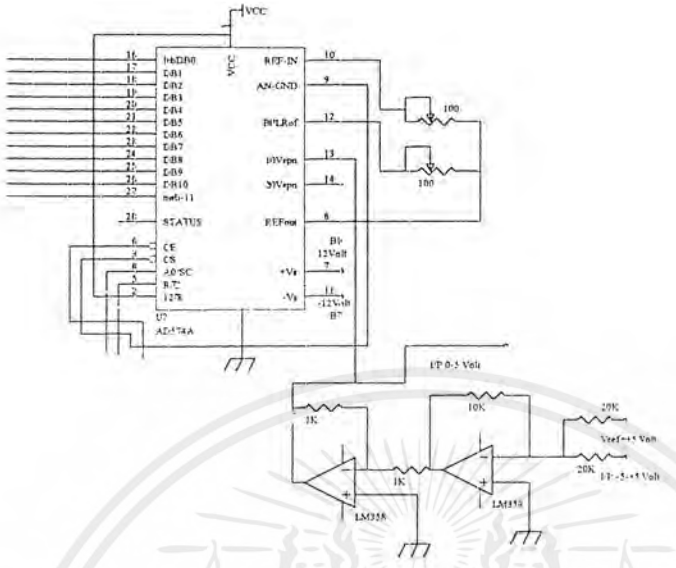
ความหมายของขาต่าง ๆ ของ ADC574A

- Vlogic ต่อกับแหล่งจ่ายไฟ 5 โวลท์
- 12/8 เลือกการทำงานของตัวแปลงสัญญาณ
- CS เลือกการทำงานของตัวแปลงสัญญาณ
- A_0 เลือกการทำงานของตัวแปลงสัญญาณ
- R/C เลือกการทำงานของตัวแปลงสัญญาณ
- CE เลือกการทำงานของตัวแปลงสัญญาณ
- Vcc ต่อกับแหล่งจ่ายไฟ +12/+15 โวลท์
- REF OUT ต่อกับแหล่งจ่ายไฟ 5 โวลท์สำหรับแบบunipolar
- AC ต่อเกราวนด
- REF IN ต่อแรงดันอ้างอิงจากภายนอกสำหรับแบบ unipolar
- V_{EE} ต่อกับแหล่งจ่ายไฟ –12/-15 โวลท์
- BIP OFF สำหรับแบบbipolar
- 10 Vin เลือกอินพุทช่วง 10 โวลท์
- 20 Vin เลือกอินพุทช่วง 20 โวลท์
- STS แสดงสถานะของตัวแปลงสัญญาณ
- DB0-DB11 เป็นขาเอาท์พุท 12 บิทของวงจร
- DC ต่อลงเกราวนด

การต่อวงจรการใช้งานของ ADC574A

จากรูป 3.12 ได้นำ ADC574A ไปทำการต่อใช้งานโดยที่จะทำการต่อลงการ์ดอินเตอร์เฟส ซึ่งจะติดต่อกับคอมพิวเตอร์โดยผ่านทางไอซี 8255 โดยจะต่อวงจรดังรูปที่ 3.13 และเป็นส่วนหนึ่งของอินเตอร์เฟสการ์ดซึ่งจะมีส่วนไปติดต่อกับรูปที่ 3. 11 ดังนี้

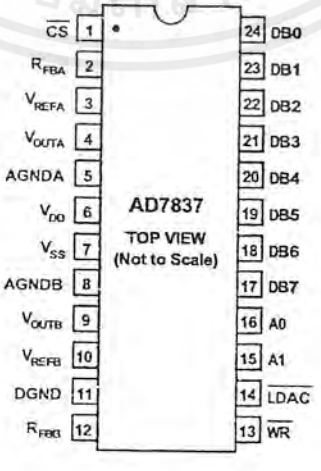
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงวงจรการนำไปใช้งานของ ADC574A

3.4.3 การเลือกไอซีตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อกมาใช้งานในวงจร(เลือกAD7837)

ตัว AD7837 เป็นไอซีวงจร D/A 12 บิท แต่จะรับค่าแค่ 8 บิท โดยตัวมันจะสร้าง 4 บิทขึ้นมาเพื่อคำนวณค่าที่เป็น 12 บิทออกไปใช้สำหรับต่อกับไมโครโปรเซสเซอร์แบบต่าง ๆ ซึ่งสามารถแสดงขาต่าง ๆ ได้ดังรูปที่ 3.14



รูปที่ 3.14 แสดงรูปขาของไอซีAD7837

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

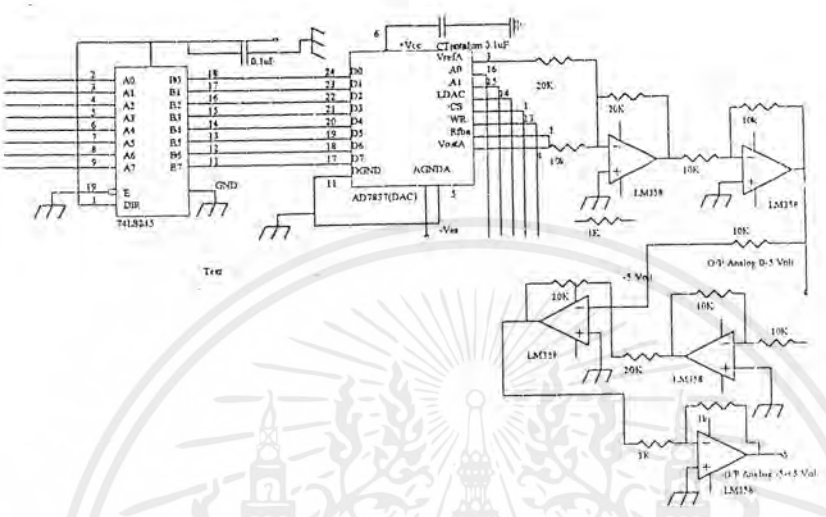
ความหมายของขาต่าง ๆ ของ AD7837

- CS ตัวแปลงสัญญาณจะถูกเลือกเมื่อactive low
- R_{FBA} ตัวความต้านทานของ DACA
- V_{REFA} แรงดันอ้างอิงสำหรับDAC
- V_{OUTA} เอาท์พุทของ DAC A
- AGNDA ต่อเกราวนด
- V_{DD} ต่อแหล่งจ่ายไฟบวก
- V_{SS} ต่อแหล่งจ่ายไฟลบ
- AGANDB ต่อเกรานวด
- V_{OUTB} เอาท์พุทของ DAC B
- V_{REFB} แรงดันอ้างอิวของDAC B
- DGAND ต่อเกราวนด
- R_{FBB} ตัวความต้านทานของDAC B
- WR ใช้เลือกการทำงานของตัวแปลงสัญญาณ
- LDAC ใช้เลือกการทำงานของตัวแปลงสัญญาณ
- A_1 ค่าแอดเดรสอินพุท สำหรับเลือกการทำงาน
- A_0 ค่าแอดเดรสอินพุท สำหรับเลือกการทำงาน
- DB7-DB4 บิทข้อมูล
- DB3-DB0 บิทข้อมูล

การต่อวงจรการใช้งานของAD7837

จากรูปที่ 3.14 ได้นำ AD7837 ไปต่อใช้งาน โดยที่จะทำการต่อลงบนอินเตอร์เฟสการ์ด ซึ่งจะติดต่อกับคอมพิวเตอร์ผ่านไอซี 8255 โดยต่อวงจรดังรูปที่ 3.15 และวงจรนี้จะเป็นส่วนหนึ่งของอินเตอร์เฟสการ์ดซึ่งจะมีส่วนไปติดต่อกับรูปที่ 3.11 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

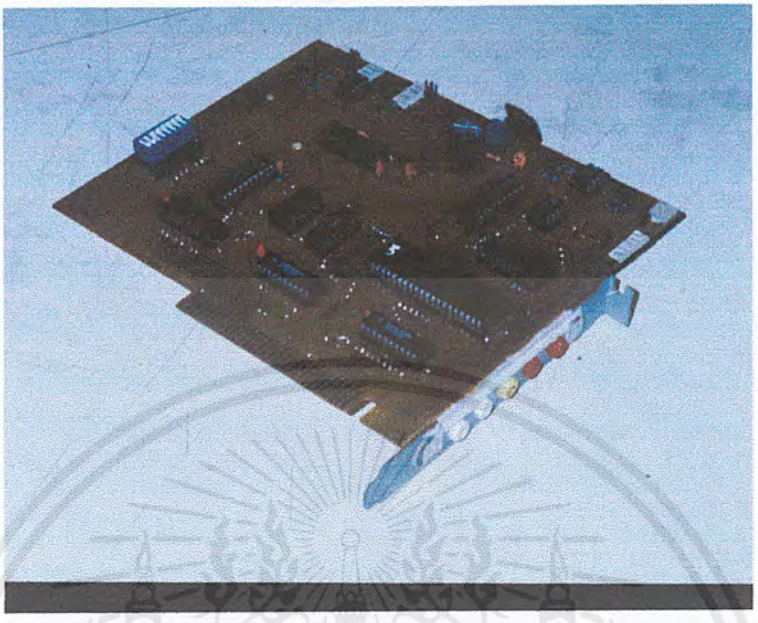


รูปที่ 3.15 แสดงวงจรการนำไปใช้งานของ AD7837

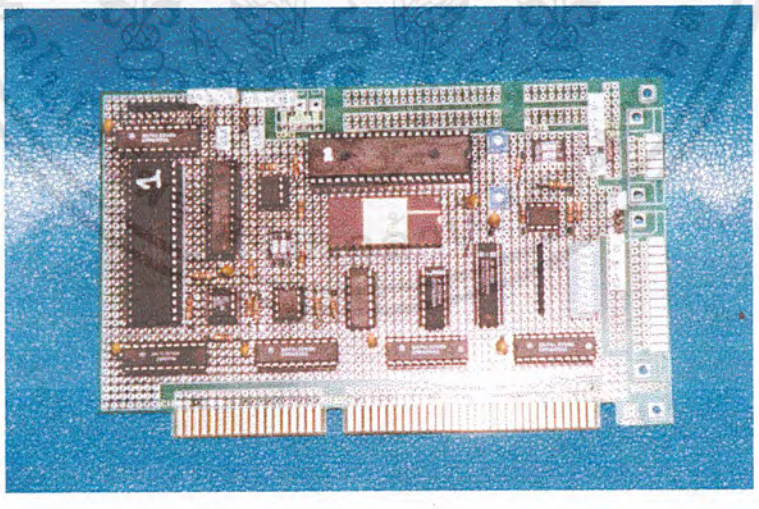
3.5 สรุปการสร้างฮาร์ดแวร์

สำหรับการสร้างฮาร์ดแวร์นั้นจะประกอบด้วยวงจร 3 ส่วนดังที่ได้กล่าวมาข้างต้น โดยเมื่อนำทั้ง 3 ส่วนนี้มาต่อรวมบนการ์ดเดียวกันจะสามารถแสดงภาพของวงจรที่ทำการต่อเรียบร้อยแล้วได้ดังรูปที่ 3.16 และ 3.17 โดยรูปที่ 3.16 แสดงวงจรดั้งเดิมและรูปที่ 3.17 แสดงวงจรใหม่ที่ต่อสำเร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงการ์ดวงจรที่ต่อเสร็จแล้ว สำหรับ 8 บิท



รูปที่ 3.17 แสดงการ์ดวงจรที่ต่อเสร็จแล้วสำหรับ 12 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6การพัฒนาด้านซอฟท์แวร์

จากการที่ระบบปฏิบัติการวินโดวส์ (Windows) ในคอมพิวเตอร์ เป็นระบบปฏิบัติการแบบกราฟฟิกที่อำนวยความสะดวกในการใช้งานได้เป็นอย่างมาก ทำให้คอมพิวเตอร์มีประสิทธิภาพสูงกว่าดอส ดังนั้นการเขียนโปรแกรมเพื่อรัน (Run) บนวินโดวส์จึงเป็นทางเดียวที่จะทำให้เกิดประโยชน์สูงสุดทั้งนี้เพราะ

1. โปรแกรมบนวินโดวส์จะนำความสามารถของวินโดวส์มาใช้อย่างเต็มที่
2. การเขียนโปรแกรมสำหรับวินโดวส์ทำได้ง่ายกว่าการเขียนโปรแกรมสำหรับดอส

โดยโปรแกรมหรือแอปพลิเคชั่นบนวินโดวส์มีวิธีเขียน 2 แบบ คือ

1. การเขียนแบบวิชวล หรือ วิชวลโปรแกรมมิง (Visual Programming)
2. การเขียนแบบนอนวิชวล หรือ นอนวิชวลโปรแกรมมิง (None Visual Programming)

วิชวล หมายถึง การมองเห็น การเขียนโปรแกรมแบบวิชวลจึงเป็นการเขียนโปรแกรมด้วยส่วนประกอบที่เรามองเห็นขณะกำลังเขียนโปรแกรม ซึ่งสามารถเขียนได้ง่ายกว่า โดยในการเลือกใช้คอมไพเลอร์ในการเขียนโปรแกรมแบบวิชวลนี้สามารถเลือกได้หลายแบบ เช่น วิชวล ซี พลัส พลัส (Visual C++) บอร์แลนด์ เดลไฟ (Borland Delphi) วิชวล เบสิก (Visual Basic) เป็นต้น

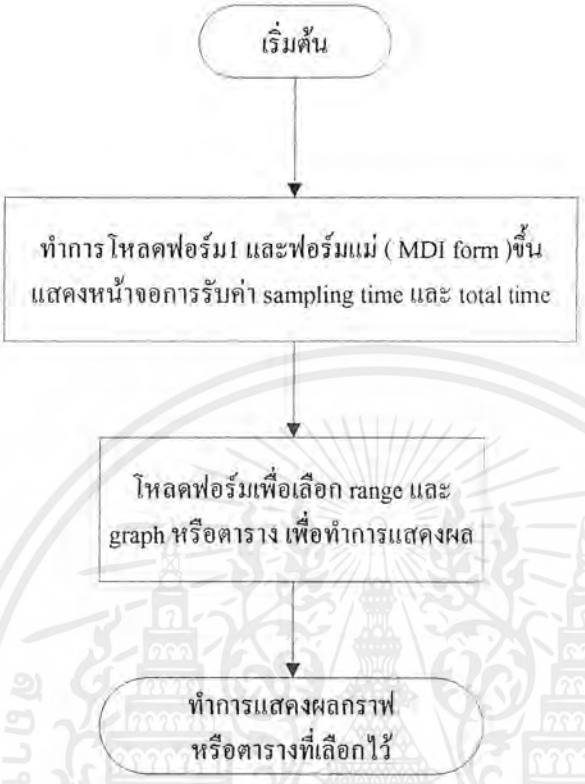
ในโครงงานนี้เลือกใช้ วิชวล เบสิก ในการเขียนโปรแกรม เนื่องจากวิชวล เบสิก เป็นภาษาที่ง่ายต่อการเขียน การทำความเข้าใจ เป็นที่นิยมอย่างแพร่หลาย และสามารถพัฒนาต่อได้ง่าย โดยในการออกแบบโปรแกรมได้แบ่งออกเป็น 2 ส่วน คือ

1.ส่วนติดต่อและควบคุมการ์ดอินเทอร์เฟส

2.ส่วนแสดงผลเป็นกราฟและตารางต่าง ๆ

โดยแสดงโฟลว์ชาร์ทการทำงานของโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงการทำงานของโปรแกรมส่วนหน้าจอ

3.6.1 ส่วนติดต่อและควบคุมการ์ดอินเทอร์เฟส

ส่วนติดต่อ ควบคุมการ์ดอินเทอร์เฟสนี้ เลือกใช้ บอร์แลนด์ เดลไฟ ในการสร้างฟังก์ชันดี แอล แอล (DLLs : Dynamic Link Lybraries) โดย ดี แอล แอล นี้เป็นไฟล์ที่รวบรวมคำสั่งต่าง ๆ ที่อาจถูกสร้างด้วยแอปพลิเคชั่นอื่น ๆ เช่น วิชวล ซีพลัสพลัส ,บอร์แลนด์ เดลไฟ แล้วนำมาใน วิชวล เบสิก หรือแอปพลิเคชั่นอื่น ๆ ได้ โดยไฟล์ ดี แ อล แอล นี้จะถูกเรียกในโหมด รันไทม์ (Run time) เนื่องจากเป็นไฟล์ที่ผ่านการคอมไพล์ (Compile) แล้ว

ไฟล์ ดี แอล แอล มีประโยชน์ดังนี้

1.เมื่อเราใช้ ดี แอล แอล ในการเก็บโค้ด (code) ที่ใช้ร่วมกันหลาย ๆ แอปพลิเคชั่นจะช่วยให้ประหยัดหน่วยความจำในการเก็บข้อมูล เนื่องจากมีการโหลด (load) ข้อมูลเพียงครั้งเดียว

2.ช่วยลดการเขียนโปรแกรม เนื่องจากสามารถใช้คำสั่งที่อยู่ใน ดี แอล แอล ที่ได้สร้างไว้แล้ว

3.ในกรณีที่เราได้แยกการทำงานของแต่ละส่วนของแอปพลิเคชั่นลงไปใน ดี แอล แอล เราสามารถทำการเปลี่ยนแปลงการทำงานของ ดี แอล แอล โดยที่ไม่ต้องแก้ไขแอปพลิเคชั่นทั้งหมด แล้วทำการแก้ไขเฉพาะ ดี แอล แอล ที่ทำการแก้ไขได้

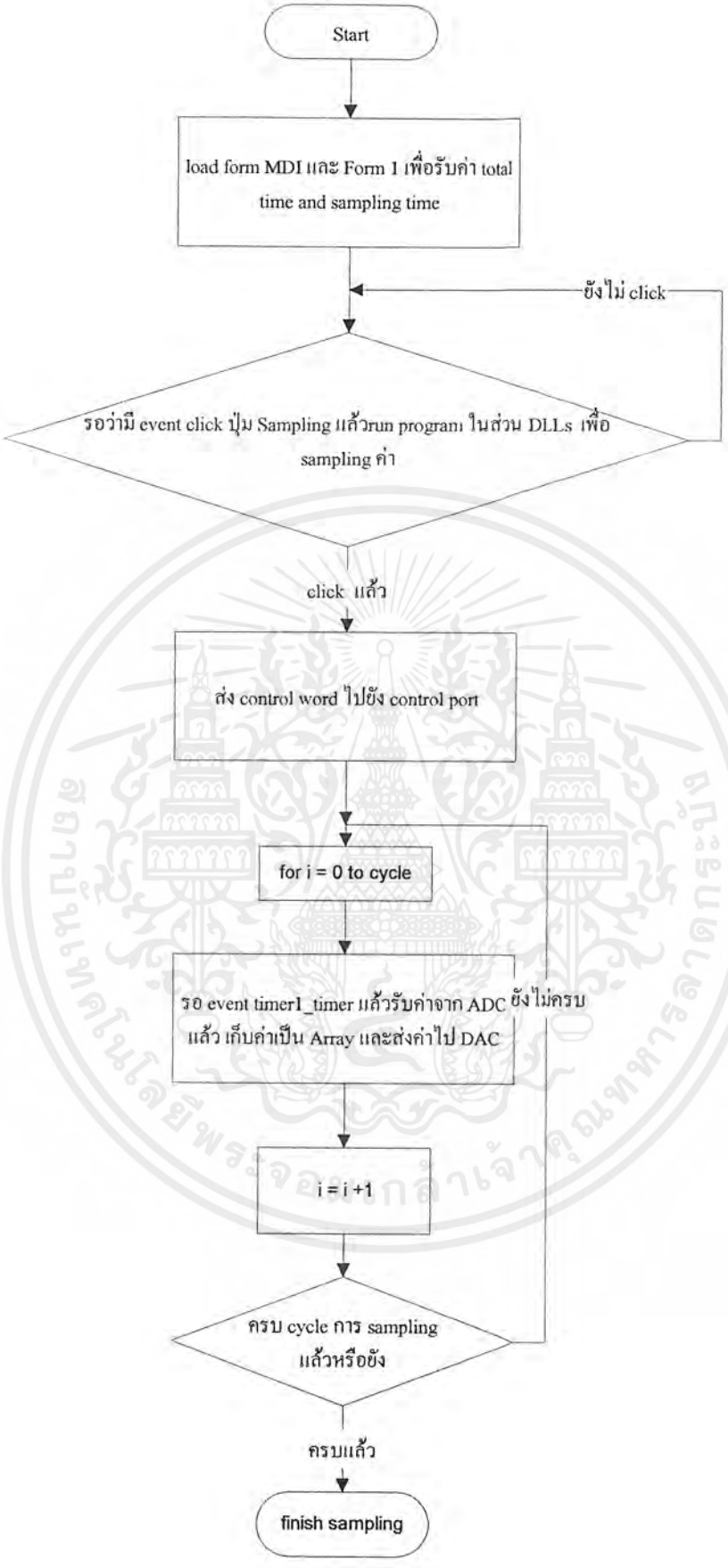
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.ในการทำงานร่วมกับวินโดวส์ เราต้องรู้จักการใช้ ดี แอล แอล เนื่องจากทุกๆ คำสั่งในระบบวินโดวส์จะถูกเก็บในรูปแบบ ดี แอล แอล (วินโดวส์ เอ พี ไอ : Windows API)

หลังจากคอมไพล์โค้ดที่สร้างจากบอร์แลนด์ เดลไฟ ได้แล้ว จะได้ไฟล์ดี แอล แอล มา และเพื่อให้ใช้งาน ดี แอล แอล ได้ ต้องลอกไฟล์ ดี แอล แอล ไปยัง ซับไดเรคทอรี (subdirectory) ซิสเต็ม (System) ของวินโดวส์ก่อน แล้วประกาศการใช้ ดี แอล แอล ในโปรแกรม วิชวล เบสิกที่เขียนขึ้น จึงจะใช้งานได้

โดยโฟลว์ชาร์ทการแซมปลิ้งสัญญาณ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงการทำงานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2ส่วนของการแสดงผลเป็นกราฟ และตารางต่าง ๆ

เมื่อรับข้อมูลมาจากส่วนการ์ดอินเทอร์เฟสแล้ว ส่วนการแสดงผลจะเขียนโปรแกรมด้วย วิชวล เบสิก โดยแสดงผลเป็น 4 รูปแบบดังนี้

1.Sampling Time graph

2.Sampling and Hold graph

3.Hex Data table

4.Continuous graph

3.6.2.1 Sampling Time graph

เป็นการแสดงผลการแซมปลิ้งค่าตามค่าเวลาที่ต้องการ โดยแสดงเป็นสัณลักษณ์กากบาท กับเส้นกราฟสีน้ำเงิน โดยในการออกแบบโปรแกรมในส่วนนี้ จะใช้คอนโทรล (Control) พินนาเคิล – บี พี เอส กราฟ คอนโทรล (Pinnacle-BPS Graph Control) หรือ คอนโทรลกราฟ ซึ่งจะสร้างกราฟตามข้อมูลที่เรากำหนดในพรอพเพอร์ตี (Property) กราฟดาตา (Graphdata) โดยมีพรอพเพอร์ตี ที่น่าสนใจดังนี้

BottomTitle ใช้แสดงข้อความใต้กราฟ

ColorData ใช้กำหนดสีของกราฟ

GraphStyle ใช้กำหนดรูปแบบในการแสดงผล เช่น แสดงในแนวนอน ในแนวตั้ง เป็นต้น

GraphTitle ใช้กำหนดข้อความเหนือรูปกราฟ

GraphType ใช้กำหนดรูปแบบของกราฟ เช่น กราฟแท่ง กราฟเส้น เป็นต้น

GridStyle ใช้กำหนดรูปแบบกริด (Grid)

NumPoints ใช้กำหนดจำนวนข้อมูลในแกน เอ็กซ์ (X)

NumSets ใช้กำหนดจำนวนข้อมูลในแต่ละกลุ่ม

3.6.2.2 Sampling and Hold graph

เป็นการแสดงผลคล้ายในรูปแบบแรก แต่มีการโฮลด์ (Hold) ค่าไว้ โดยจะใช้คอนโทรล พิคเจอร์บอคซ์ (Picture box) ซึ่งเป็นคอนโทรลมาตรฐานของวิชวล เบสิก

3.6.2.3 Hex Data table

เป็นการแสดงค่าในตารางในลักษณะ เป็นค่าเลขฐาน10 และเลขฐาน 16 โดยใช้คอนโทรล ไมโครซอฟท์ เฟล็กซ์กริด (Microsoft FlexGrid 6.0) ซึ่งเป็นคอนโทรลที่ใช้สร้างตารางเก็บข้อมูลต่าง ๆ โดยใช้พรอพเพอร์ตี เท็กซ์เมตริกซ์ (TextMatrix) ในการระบุค่าลงตาราง

3.5.2.4 Continuous graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการจำลองการทำงานของวงจรแปลงจากดิจิตอล เป็นอนาล็อก โดยจะใช้คอนโทรล กราฟเหมือนกับ Sampling Time graph

โดยมีโฟลว์ชาร์ทในส่วนการแสดงผลกราฟ และตารางต่างๆ ดังนี้

รูปที่ 3.20 แสดงการทำงานของส่วนแสดงผลของกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การออกแบบด้านซ็อฟท์แวร์

การออกแบบด้านซ็อฟท์แวร์นั้นได้ใช้ไมโครซอฟท์ วิชวล เบสิก ในการควบคุมการทำงานของเครื่องและการ์ด

โดยในการออกแบบซ็อฟท์แวร์ได้แบ่งโปรแกรมเป็น 2 ส่วนคือ

1.การติดต่อและควบคุมการ์ดอินเทอร์เฟส

2.การแสดงผลข้อมูลด้วยกราฟ และตารางต่าง ๆ

การติดต่อและควบคุมการ์ดอินเทอร์เฟส

เนื่องจากขการติดต่อพอร์ทอินพุต เอาต์พุตของวิชวล เบสิกในส่วนนี้ได้ใช้วิธีสร้างไฟล์ ดี แอล แอล ขึ้นมาใช้ติดต่อกับการ์ดอินเทอร์เฟส โดยใช้โปรแกรม บอร์แลนด์ เดลไฟ ซึ่งสามารถเขียน โค้ด แอสเซมบลี (Assembly) และคอมไพล์โค้ดนี้เป็นไฟล์ ดี แอล แอล ได้

ในการใช้งานไฟล์ ดี แอล แอล นี้เมื่อคอมไพล์เป็น ดี แอล แอล แล้วให้ คัดลอกไฟล์นี้ไปที่ซับไดเรคทอรี่ ซิสเต็ม ของวินโดวส์ แล้วอ้างในส่วนของวิชวล เบสิก ถึงไฟล์ที่จะใช้ด้วย

การแสดงผลข้อมูลด้วยกราฟต่าง ๆ

เมื่อรับข้อมูลมาจากส่วนการ์ดอินเทอร์เฟสแล้ว ส่วนการแสดงผลจะเขียนโปรแกรมด้วยวิชวลเบสิก โดยแสดงผลเป็น 4 รูปแบบดังนี้

1.Sampling Time graph

2.Sampling and Hold graph

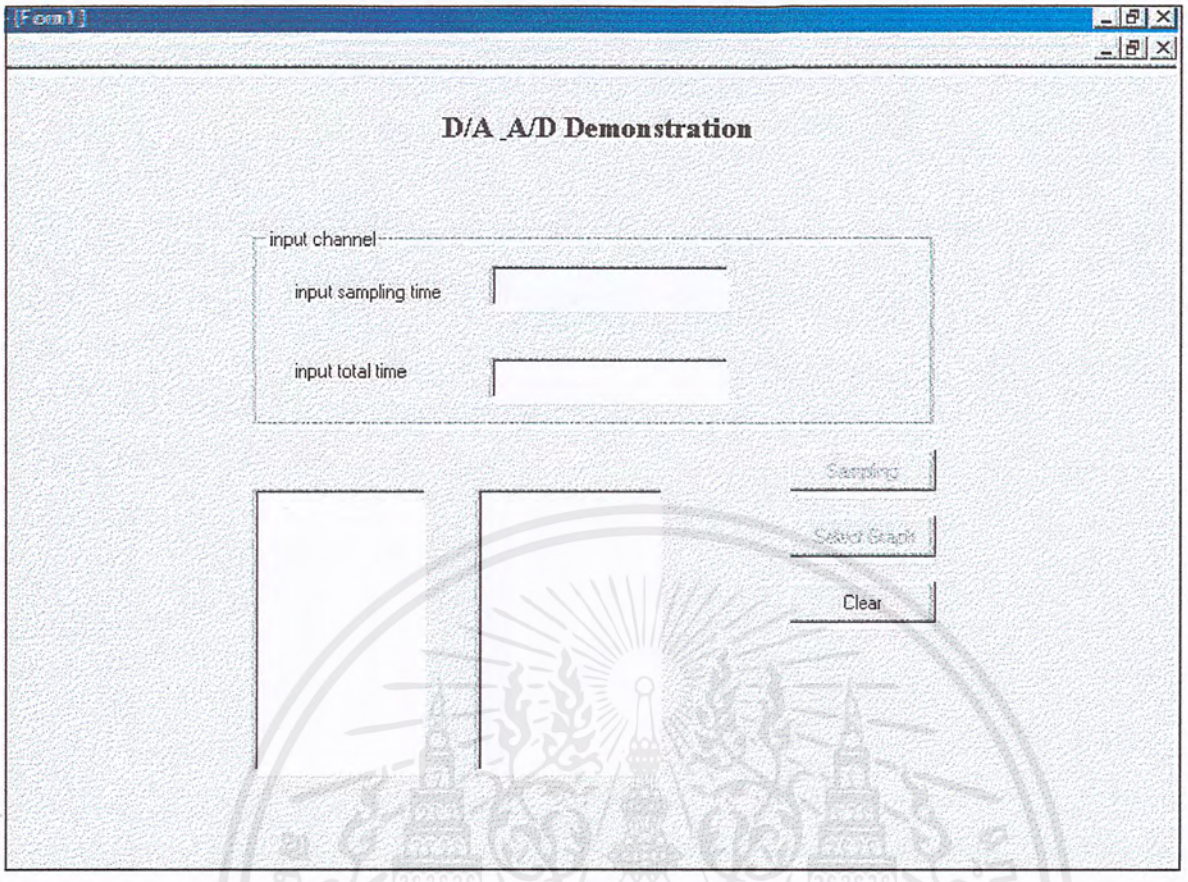
3.Hex Data table

4.Continuous graph

4.2 การใช้งานโปรแกรม

เมื่อทำการรันโปรแกรมแล้ว โปรแกรมจะแสดงหน้าจอแรก จะให้ผู้ใช้ พิมพ์ค่า Sampling Time และค่าเวลาทั้งหมดที่ต้องการ Sampling (Total Time) เพื่อที่จะนำค่าทั้ง 2 ค่า ไปใช้ในแซมปลิ้งสัญญาณ ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

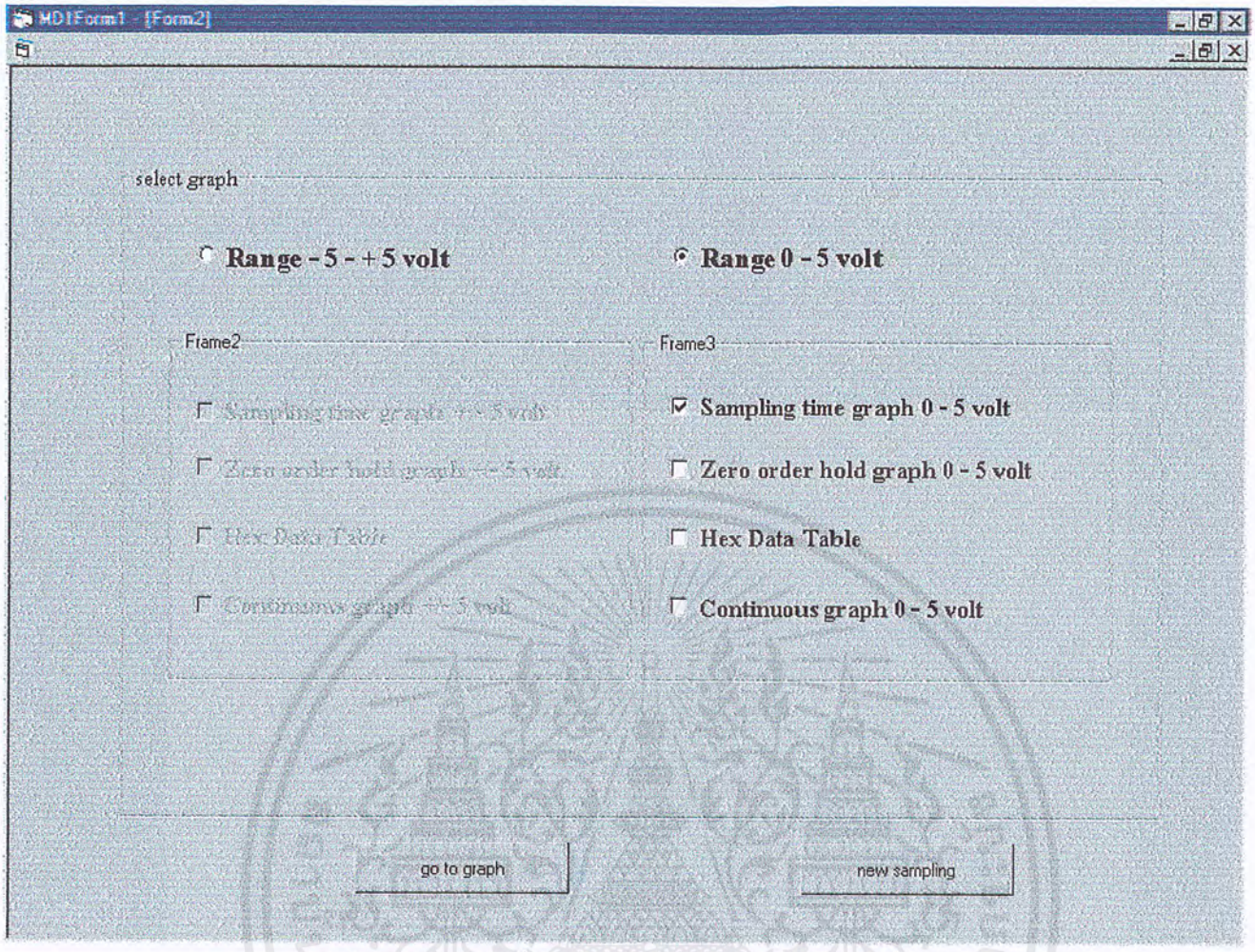


รูปที่ 4.1 แสดงรูปการใส่ค่า Sampling Time และค่าเวลาทั้งหมดที่ต้องการ Sampling

โดยโปรแกรมจะแอคทีฟ (Active) ที่เท็กซ์บอคซ์ (Text box) รับค่า Sampling time และจะแอคทีฟที่เท็กซ์บอคซ์ รับค่า Total time โดยถ้าไม่ใส่ค่าทั้งสองแล้วปุ่ม Sampling จะไมาสามารถใช้งานได้ และจะไม่แอคทีฟด้วย ซึ่งเมื่อใส่ค่าเสร็จแล้วให้กดปุ่ม Sampling เพื่อเลือกการแซมปลิ้ง และเมื่อทำการแซมปลิ้งเสร็จ ปุ่ม Select Graph จะแอคทีฟแล้วให้เลือกกราฟจากปุ่ม Select Graph ถ้าต้องการเริ่มใหม่ให้กดปุ่มเคลียร์

หน้าจอต่อมาเป็นหน้าจอสำหรับให้ผู้ใช้โปรแกรมเลือกช่วงของค่าสัญญาณอินพุทระหว่าง –5 ถึง +5 โวลท์ จากนั้นก็เลือกว่าต้องการดูกราฟแบบไหนมีทั้งหมด 4 ชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.2 แสดงการเลือก ช่วงค่าสัญญาณและเลือกกราฟ

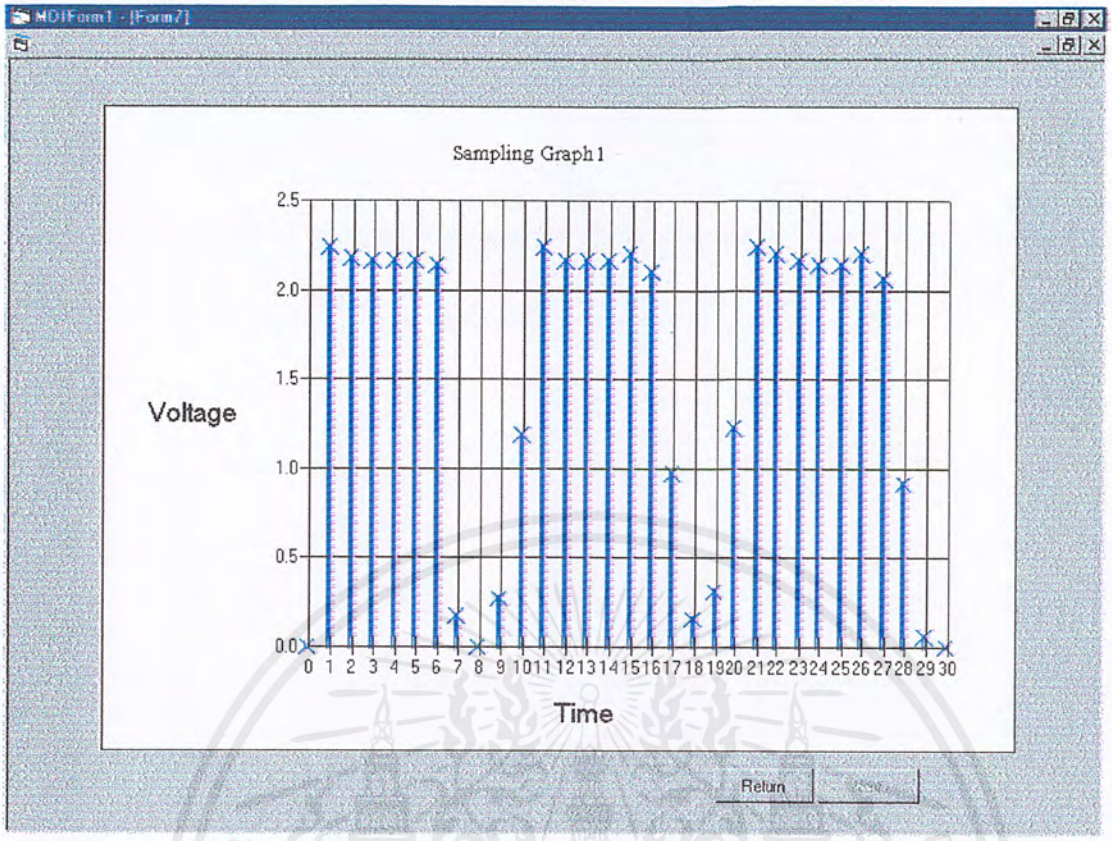
โดยผู้ใช้โปรแกรมสามารถเลือกได้โดยการเลือกที่เช็คบอคซ์ (Check box) เพื่อเลือกการดูกราฟ 4 แบบดังนี้

1. Sampling Graph
2. Samplig & Hold Graph
3. Hex Data Table
4. Continuous Graph

ซึ่งจะอธิบายลักษณะของกราฟต่างๆได้ดังนี้

ถ้าเราเลือก Sampling Graph ก็จะแสดงรูปออกมาได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

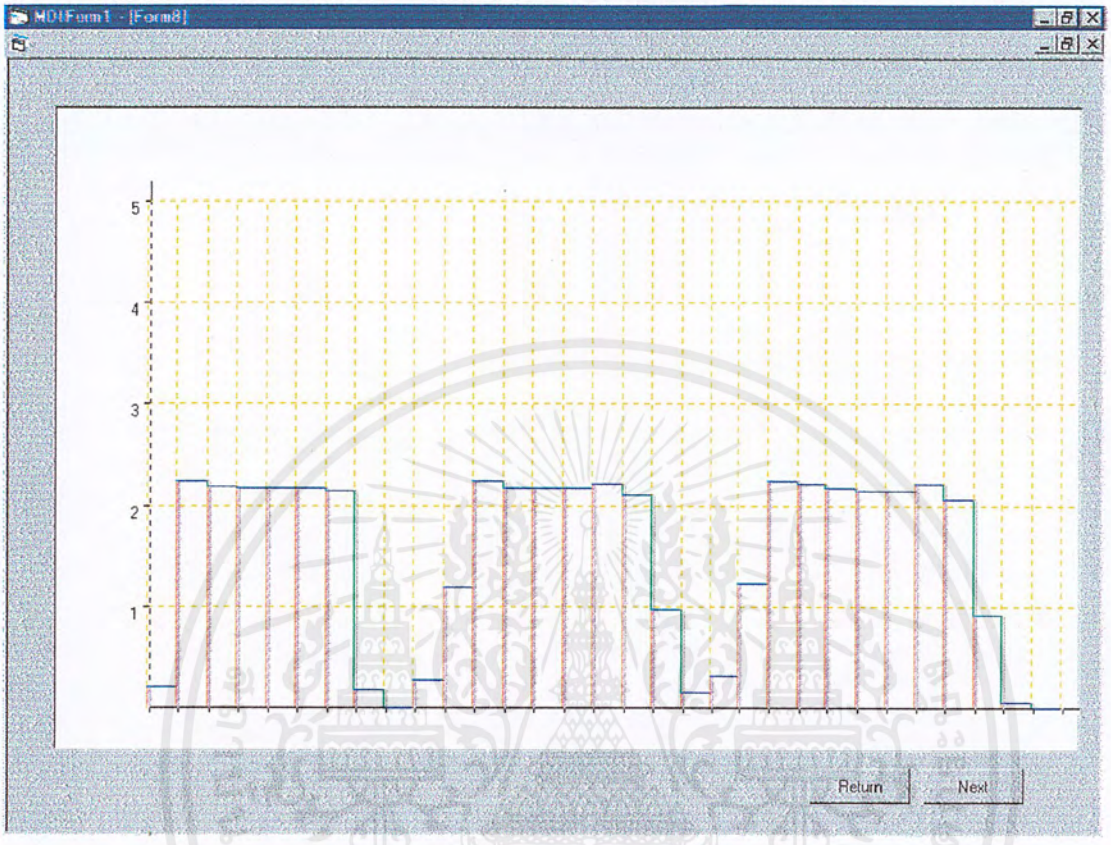


รูปที่ 4.3 แสดงหน้าจอการกราฟการแซมปลิ้ง

จากรูปเป็นกราฟที่วงจรแปลงอนาล็อกเป็นดิจิตอล ทำการแซมปลิ้งค่าสัญญาณไซน์ ตาม Sampling time ที่ผู้ใช้ป้อนให้ กราฟจะมีความละเอียดและคล้ายรูปสัญญาณจริงที่เกิดขึ้น ถ้าใช้ Sampling time น้อยลง

ถ้าเราเลือกรูปกราฟการแซมปลิ้งและโฮลด์ซึ่งจะทำแสดงการโฮลด์ค่าของวงจรแปลงอนาล็อกเป็นดิจิตอล กระบวนการนี้สำหรับวงจรแปลงอนาล็อกเป็นดิจิตอล จริง ๆ มีไว้เพื่อให้เวลาวงจรแปลงอนาล็อกเป็นดิจิตอลทำการแปลงค่าสัญญาณจากสัญญาณอนาล็อก มาเป็นสัญญาณดิจิตอล แสดงได้ดังรูปที่ 4.4

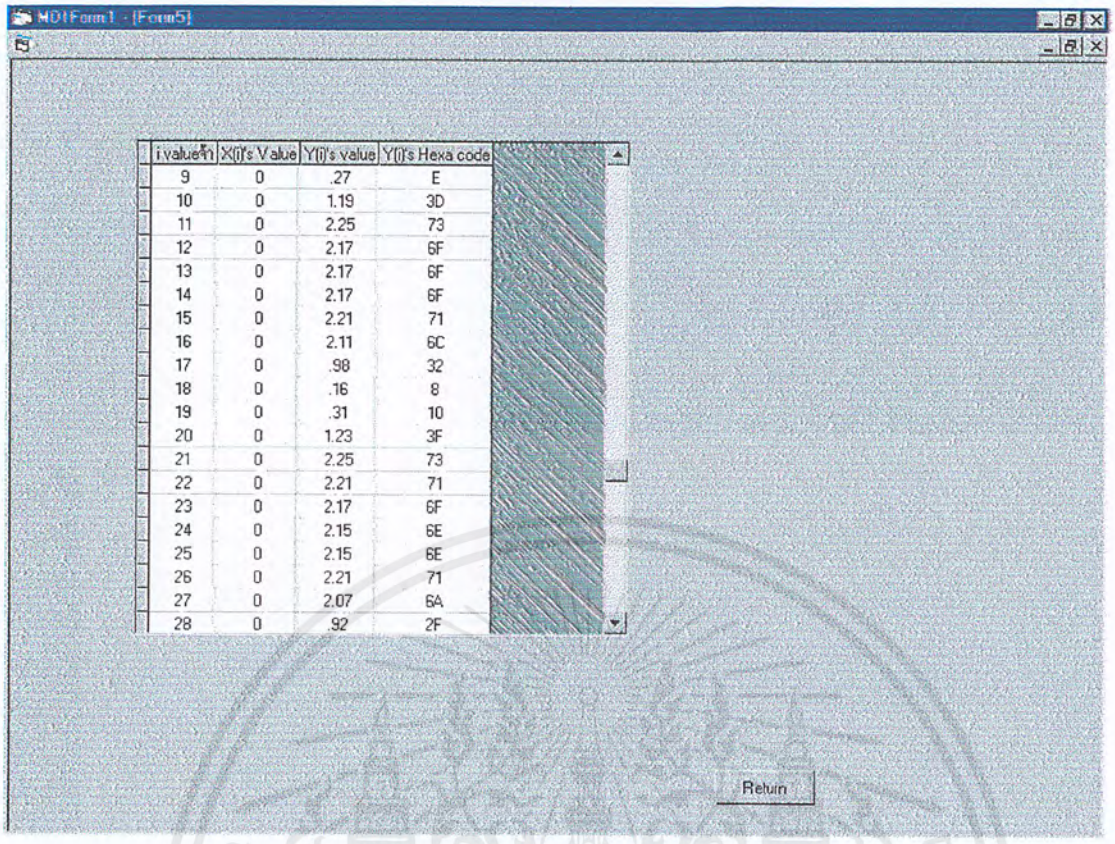
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.4 แสดงกราฟแซมปลิ้งและโฮลด

ต่อจากการโฮลด์ค่าอนาล็อกเป็นดิจิตอล จะทำการแปลงค่าอนาล็อกเป็นดิจิตอล ณ เวลา ที่แซมปลิ้ง นั้น ๆ ค่าดิจิตอล ที่ได้จากการโฮลด์ จะถูกแสดงใน Hex Data Table ซึ่งจะแสดงค่าดิจิตอล (เลขฐาน 16) ณ. เวลาแซมปลิ้ง นั้น ๆ ออกมา พร้อมทั้งแสดงค่าอนาล็อกของสัญญาณที่เข้ามาเป็นโวลท์เตจ (Voltage) อีกด้วย ดังรูป 4.5

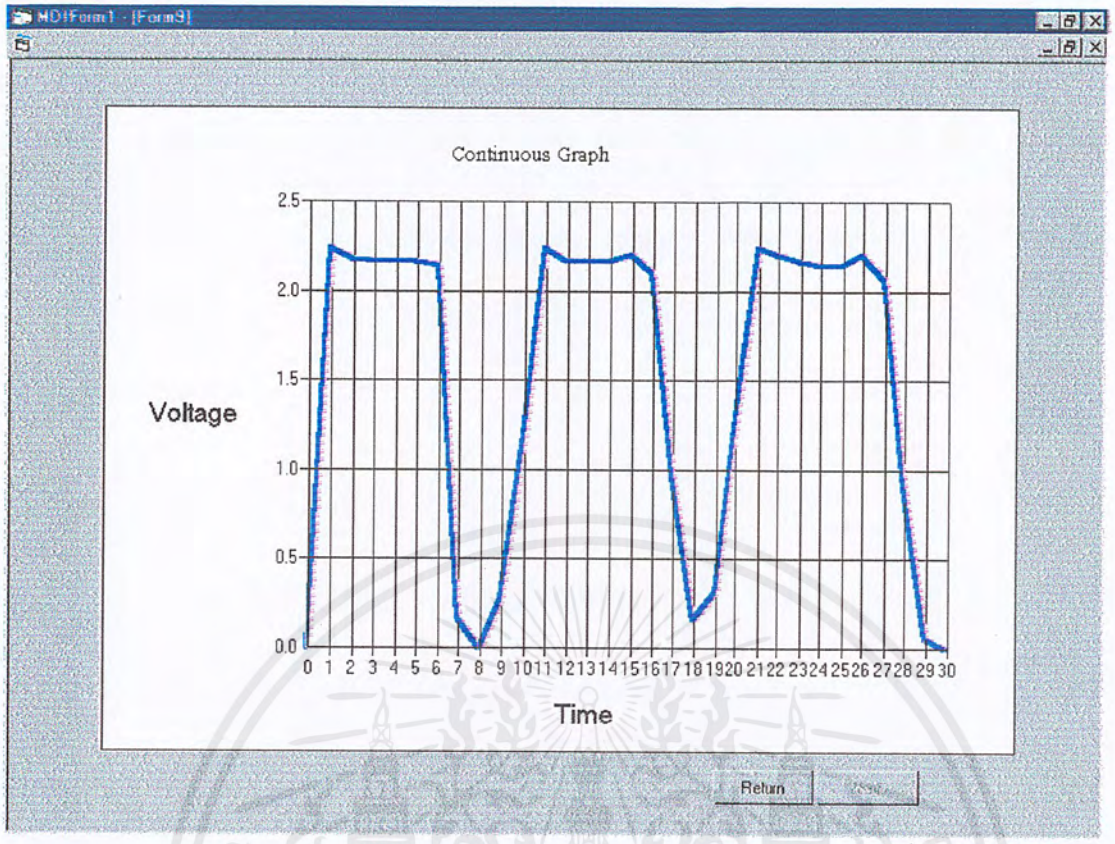
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดง Hex Data Table

ถ้าเลือกContinuous Graph ซึ่งหลังจากที่เราได้ค่าดิจิตอล มาแล้ว เราก็ทำการป้อนสัญญาณดิจิตอล นั้น เข้าวงจรแปลงดิจิตอลเป็นอนาล็อก เพื่อทำการจำลองรูปสัญญาณเดิมขึ้นมาใหม่ รูปสัญญาณที่ได้หลังจากผ่านวงจนแปลงดิจิตอลเป็นอนาล็อก แล้วจะมีลักษณะคล้ายกับรูปสัญญาณก่อนที่จะถูกแซมปลิ้ง ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.6 Continuous Graph

โดยเมื่อดูรูปกราฟแต่ละรูปเสร็จแล้ว โปรแกรมจะกลับมายังหน้าจอเลือกลักษณะรูปกราฟอีกครั้ง เพื่อให้ผู้ใช้สามารถเลือกดูลักษณะกราฟอื่น ๆ ของสัญญาณเดิมได้ โดยค่าของสัญญาณจะถูกเก็บไว้และสามารถเรียกดูจนกว่าจะทำการแซมปลิ้งใหม่จะได้รูปกราฟใหม่ส่วนสัญญาณ เอาท์พุท จากวงจรแปลงดิจิตอลเป็นอนาล็อก ที่จะถูกส่งออกมาจาก คอมพิวเตอร์ นั้น จะสามารถตรวจจับได้โดยการใช้ ออซซิลโลสโคป (Oscilloscope) ตรวจวัดออกมา ซึ่งถ้าต้องการจะกลับมาหน้าจอก่อนหน้าให้กดปุ่ม Return เรื่อย ๆ ก็จะกลับสู่หน้าจอเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

5.1สรุปผลการดำเนินงาน

ในการทำปริญญานิพนธ์นี้เป็นการทำชุดจำลองการทำงานของการแปลงสัญญาณอนาล็อกเป็นดิจิตอลและการแปลงสัญญาณดิจิตอลเป็นอนาล็อก โดยแสดงผลการจำลองการทำงานออกที่คอมพิวเตอร์ซึ่งในการทำงานครั้งนี้สิ่งที่ได้ทำประกอบไปด้วยส่วนประกอบ 2 ส่วนนั่นคือ ส่วนฮาร์ดแวร์และซ็อฟทแวร์

โดยในส่วนของฮาร์ดแวร์นั้นได้ทำการ์ด A/D D/Aเพื่อที่จะทำการรับส่งข้อมูลเข้าสู่คอมพิวเตอร์แล้วนำค่าที่ป้อนเข้าไปนั้นไปคำนวณค่าและแสดงผลออกมาทางคอมพิวเตอร์

โดยในส่วนของซ็อฟทแวร์นั้นได้ทำการเขียนโปรแกรมภาษา Visual Basicเพื่อใช้ในการควบคุมการทำงานของการ์ดและใช้สำหรับแสดงผลการจำลองการทำงานของการ์ดอินเทอร์เฟซ

5.2ปัญหาที่เกิดและแนวทางแก้ไข

สำหรับปัญหาของฮาร์ดแวร์ที่พบบ่อยคือสัญญาณรบกวนจากภายนอกขณะทำการแซมปลิ้งค่าซึ่งมีผลทำให้กราฟที่จะแสดงผลมีความคลาดเคลื่อนไปจากความเป็นจริงบ้าง

ในส่วนของซอฟท์แวร์นั้น ปัญหาที่พบบ่อยคือ มีสัญญาณรบกวนจนสัญญาณที่ Sampling ได้มีความคลาดเคลื่อนไปจากความเป็นจริง และเพื่อให้ได้รูปกราฟที่ดูสวยและใกล้เคียงกับค่าจริงที่สุดควรใช้ความถี่ Sampling สูง ๆ และป้อนสัญญาณอินพุทความถี่ต่ำในขณะที่ถ้าต้องการเห็นเอาท์พุทชัดเจนควรใช้ค่า Total time สูง ๆ คือมากกว่า 2000 มิลลิวินาทีขึ้นไป

5.3แนวทางพัฒนาในอนาคต

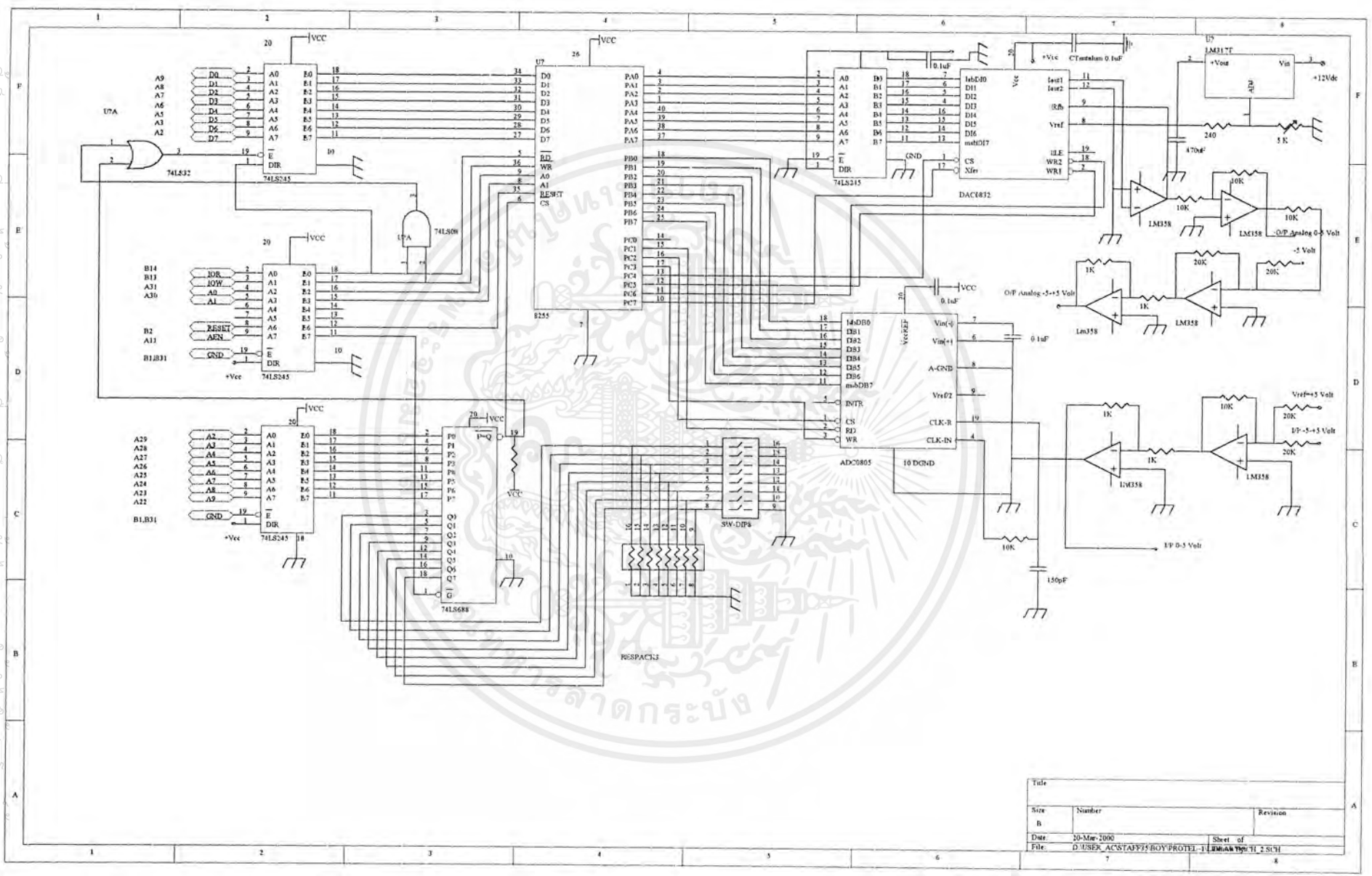
เนื่องจากวงจรในปัจจุบันที่ใช้ทดลองนี้เป็นแบบ 8 บิทซึ่งยังมีความละเอียดไม่มากดังนั้นถ้าต้องการให้มีความละเอียดมากกว่านี้ให้ทำการสร้างวงจรแบบ 12 บิท หรือ 16 บิทขึ้นมา

นอกจากนี้ควรพัฒนาให้สามารถรับสัญญาณที่ มากกว่า 1 กิโลเฮิร์ตซ ซึ่งสามารถทำได้โดยการใช้ตัวไอซีที่มีช่วงความถี่มากกว่านี้ และเนื่องจากในโครงงานนี้ เวลาต้องการใช้งานโปรแกรมต้องคัดลอกไฟล์ DLLs ไปซับไดเร็คทอรี่ System ของวินโดว์ ดังนั้นเพื่อให้สะดวกต่อการใช้งานจริงอาจใช้ Package and Development Wizard ทำเป็นไฟล์ Set up ไว้ Install โปรแกรมเลย จะได้ไม่ต้องคัดลอกไฟล์ลงซับไดเร็คทอรี่ System อีกครั้งหนึ่ง หรืออาจลองเปลี่ยนภาษาที่ใช้เขียนโปรแกรมมาเป็นเดลไฟ ซึ่งก็ง่ายต่อการใช้งาน หรือ Visual C++ ซึ่งมีความคล่องตัวและขีดความสามารถสูงก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงวงจรการ์ดอินเตอร์เฟสแบบ 8 บิท

ภาคผนวกแสดงโค้ด ดี แอล แอล

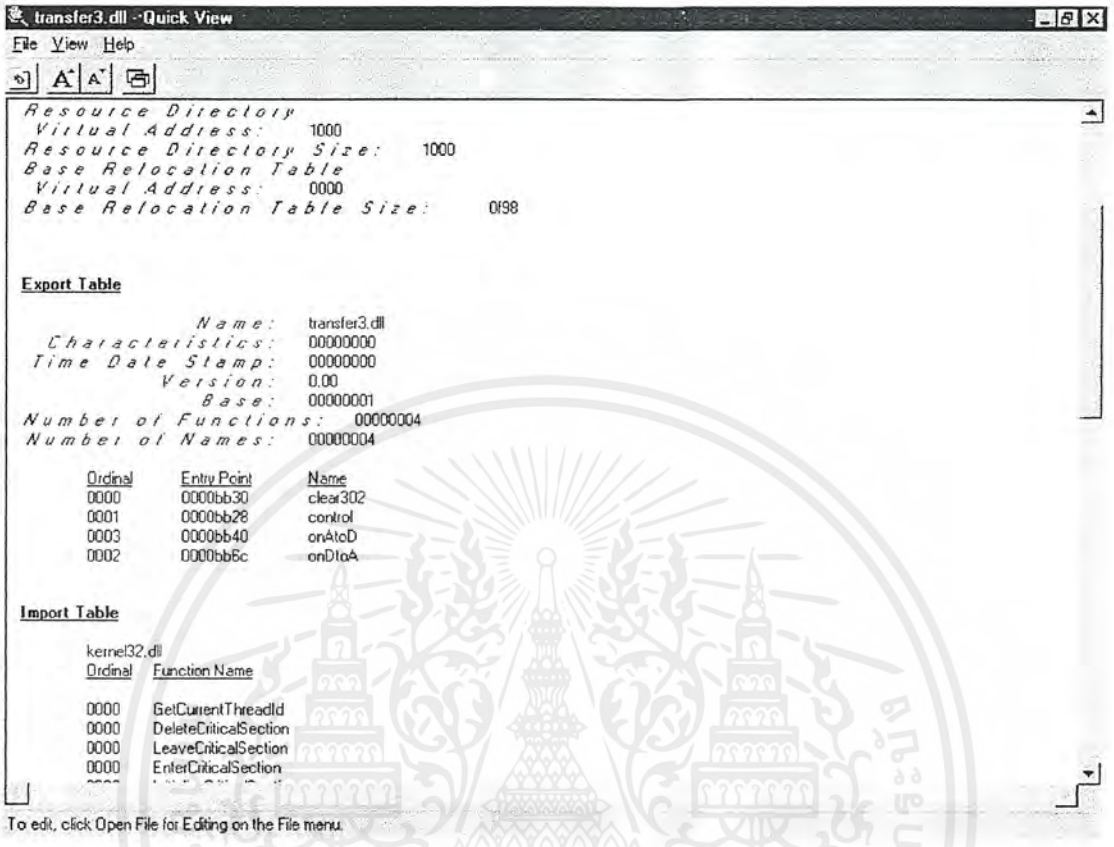
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library transfer3;

{ Important note about DLL memory management: ShareMem must be the
  first unit in your library's USES clause AND your project's (select
  Project-View Source) USES clause if your DLL exports any procedures or
  functions that pass strings as parameters or function results. This
  applies to all strings passed to and from your DLL--even those that
  are nested in records and classes. ShareMem is the interface unit to
  the BORLNDMM.DLL shared memory manager, which must be deployed along
  with your DLL. To avoid using BORLNDMM.DLL, pass string information
  using PChar or ShortString parameters. }

uses
  SysUtils,
  Classes;
        procedure  control();export;stdcall;
        { set control word to 8255 as outport(0x303 ,0x82)}
                asm
                        mov   AL,82H  // set control's word to acc
                        mov   DX,303H
                        out   DX,AL // mov control's word to control port
                end;
        procedure clear302();export;stdcall;
        { set  port 302 of 8255 to read from ADC as outport(0x302,0x00)and on only before
read data from ADC}
                begin
                        control();
                        asm
                              mov   AL,00H   // set const to acc
                              mov   DX,302H
                              out   DX,AL  //mov 00h to port 302
                end; { end asm}
        end; { end begin procedure clear302 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure onDtoA(data : byte);export;stdcall; {control to read from ADC port 301}
    var pass : byte;
    begin
       clear302();
       pass:=data;
    asm
      mov   AL,08H
      mov   DX,302H
      out   DX,AL{ on D/A to set ready for next step to send value
      from 'pass' that pass value from 'data' that pass value from
      array  again}
      mov   DX,300H
      mov   AL,data{pass}
      mov   AL,pass
      mov   DX,300H
      out   DX,AL{ mov value from array from sampling data and pass
      to data and to pass again and outport to 300h}
    end;
    end;
    function onAtoD():byte;export;stdcall;
    var  sampling :byte;
    begin
       control();
       clear302();
    asm
      mov   DX,301H
      in    AL,DX
      mov   sampling,AL
    end;
       onAtoD:=sampling;
    end;
  exports onAtoD,onDtoA,control,clear302;
begin
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



<u>ภาพผนวก</u> แสดงไฟล์ ดี แอล แอล ที่สร้างขึ้นโดยดูจากโปรแกรมควิควิว (Quick View)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ธานินทร์ ถาวรศาสนวงศ์ ,ทินกร ดุ๊ก ; "การอินเทอร์เฟส IBM PC"; หจก. สำนักพิมพ์ฟิสิกส์เซ็นเตอร์,2536

กมลมาศ กำจรกิจการ,. "คู่มือพัฒนาโปรแกรมด้วย Delphi4" ,PROVISION, 436 หน้า,2542

กนก กุสุมาลย์นุกูล,ไกรวุฒิ มั่นเสถียรสิน, "คู่มือการเขียนโปรแกรมDelphi4",Success Media Co.,Ltd.,422 หน้า

กิตติ ภักดีวัฒนะกุล,จำลอง ครูอุตสาหะ, "Visual Basic6 ฉบับโปรแกรมเมอร์" ,หจก.ไทยเจริญการพิมพ์,621 หน้า,2542

บัณฑิต จามรภูติ, "ทฤษฎี ไมโครโปรเซสเซอร์ 80X86" , บริษัท เอช. เอ็น. กรุ๊ป จำกัด,468 หน้า ,2539

ป่านคม พัฒนพีระเดช ,ภัทรชัย อุปริพุทธิกุล, " A/D & D/A Demonstration" ,54 หน้า, 2542

Michael Halvorson, "Microsoft Visual Basic 6.0 Profession Step by Step" , บริษัท เอช. เอ็น. กรุ๊ป จำกัด,686 หน้า,2542

John Clark Craig ,Jeff Webb, "การเขียนโปรแกรมบนวินโดวส์ด้วย Microsoft Visual Basic 6.0 ภาคปฏิบัติ" , บริษัท เอช. เอ็น. กรุ๊ป จำกัด, 864 หน้า ,2542

Willis J.Tompkins,John G.Webster , "Interfacing Sensors to the IBM PC",Prentice – Hall International ,Inc.,447 page ,1988

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้