สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์ป้องกันการละเมิดลิขสิทธิ์ทาง Computer

นายกิตติชัย ธัชเสถียร รหัส 38054106
นายบัลลังก์ ทัฬหกิจ รหัส 38054130
นายวรกุล เมืองสุวรรณ รหัส 38054149

เลขหมู่.....
เลขทะเบียน.....33861
วัน, เดือน, ปี.....17 ก.ย. 2542

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARDLOCK

Mr. Kittichai Tachsatien 38054106

Mr. Banlang Thanhakit 38054130

Mr. Woragoon Muangsuwan 38054149

A Special Project Submitted in Partial Fulfillment of the

Requirement for the Degree of Bachelor of Science

Department of Mathematics and Computer Science

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1998

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ อุปกรณ์ป้องกันการละเมิดลิขสิทธิ์ทาง Computer (Hardlock)

โดย นายกิตติชัย ธัชเสถียร รหัส 38054106
นายบัลลังก์ ทัฬหกิจ รหัส 38054130
นายวรกุล เมืองสุวรรณ รหัส 38054149

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์ธีรวัฒน์ ประกอบผล
ดร. นันทิกา เบญจเทพานันท์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2541

(รองศาสตราจารย์ภัคคินี ชิตสกุล)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

กรรมการการสอบปัญหาพิเศษ

(อาจารย์กรกช ประชุมรักษ์)
ประธานกรรมการการสอบปัญหาพิเศษ

(อาจารย์สิริลักษณ์ อนันต์สถิตย์สิน)
กรรมการการสอบปัญหาพิเศษ

(อาจารย์ธีรวัฒน์ ประกอบผล)
กรรมการและอาจารย์ที่ปรึกษาปัญหาพิเศษ

(ดร. นันทิกา เบญจเทพานันท์)
กรรมการและอาจารย์ที่ปรึกษาปัญหาพิเศษ

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ อุปกรณ์ป้องกันการละเมิดลิขสิทธิ์ทาง Computer (Hardlock)

โดย นายกิตติชัย ธัชเสถียร รหัส 38054106

นายบัลลังก์ ทัฬหกิจ รหัส 38054130

นายวรกุล เมืองสุวรรณ รหัส 38054149

อาจารย์ที่ปรึกษา อาจารย์ธีรวัฒน์ ประกอบผล

ดร. นันทิกา เบญจเทพานันท์

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

ปีการศึกษา 2541

บทคัดย่อ

Hardlock หลายคนอาจคิดว่าการเขียนโปรแกรมให้ใช้ Hardlock นั้นยาก สำหรับโปรแกรมเมอร์ที่ไม่มีความรู้เรื่องการติดต่อกับพอร์ตต่างๆนั้นก็อาจจะจริง ในที่นี้จึงได้เกิดความคิดที่จะอำนวยความสะดวกให้โปรแกรมเมอร์ที่คิดจะใช้ Hardlock นำไปประยุกต์กับโปรแกรมของตนเอง เพื่อที่จะป้องกันการละเมิดลิขสิทธิ์ สามารถนำไปใช้ได้อย่างง่ายดาย โดยนำโปรแกรมที่เรียกว่า ActiveX DLL ไปใช้ ซึ่งได้มาจากการนำความรู้จากการทดลองและการเขียนโปรแกรมติดต่อกับพอร์ตต่างๆมาประยุกต์เขียนเป็นโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title HARDLOCK

Name Mr. Kittichai Tachsatien 38054106
Mr. Banlang Thanhakit 38054130
Mr. Woragoon Muangsuwan 38054149

Advisor Mr. Teerawat Prakobphon
Dr. Nantika Benjatepanan

Department Mathematics and Computer Science

Year 1998

Abstract

Hardlock seems to be a difficult program for many people and it is true for a programmer who knows nothing about port connection. As a result, in order to protect the copyright and to please the user a program called ActiveX DLL is chosen to be used. From the experiment of port connection of ActiveX DLL, it is quit easy for users or programmers to apply hardlock with their own programmers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปัญหาพิเศษนี้สำเร็จลงได้ คณะผู้จัดทำต้องขอขอบพระคุณท่านอาจารย์ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์และภาควิชาฟิสิกส์ประยุกต์ทุกๆท่าน โดยเฉพาะอาจารย์ธีรวัฒน์ ประกอบผลและดร.นันทิกา เบญจเทพานันท์ ที่ได้ประสิทธิประสาทความรู้ ตลอดจนให้คำปรึกษาและแนะนำแนวทางในการทำปัญหาพิเศษจนสำเร็จลุล่วงด้วยดีและขอบคุณเพื่อนนักศึกษาทุกๆคน ที่ได้ให้ความช่วยเหลือมาตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
หน้าอนุมัติ	i
บทคัดย่อปัญหาพิเศษภาษาไทย	ii
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ	iii
กิตติประกาศ	iv
บทที่ 1 บทนำ	1
- ความสำคัญและที่มาของปัญหาพิเศษ	1
- วัตถุประสงค์ของการทำปัญหาพิเศษ	1
- ขอบเขตของปัญหาพิเศษ	1
- ขั้นตอนในการดำเนินงาน	2
- ประโยชน์ที่คาดว่าจะได้รับ	2
- อุปกรณ์ที่ใช้ทำปัญหาพิเศษ	2
บทที่ 2 ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง	3
- เทคนิคการเชื่อมต่อ	3
- การเชื่อมต่อไมโครคอมพิวเตอร์	7
- ไอซีลอจิกเกตแบบ TTL	18
- การเชื่อมต่อ 8255	24
- ตัวถอดรหัส	28
- การเขียนโปรแกรมติดต่อกับ PORT และ ActiveX DLL	30
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน	34
- เครื่องมือและอุปกรณ์	32
- โปรแกรมที่ใช้	40
- ขั้นตอนและวิธีการดำเนินงาน	41
บทที่ 4 สรุปผลปัญหาพิเศษและข้อเสนอแนะ	46
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
หัวต่อเครื่องพิมพ์ 25 ขา	3
วงจรพรินเตอร์พอร์ต	4
สัญญาณต่างๆบน System Bus	7
ระบบที่รับข้อมูลจากสวิตซ์และแสดงข้อมูลทาง LED	9
ระบบเชื่อมต่อกับพอร์ต 2 พอร์ต	9
สัญญาณเวลาการติดต่อกับพอร์ตของคอมพิวเตอร์	10
ระบบติดต่อกับพอร์ต 2 พอร์ต	11
วงจรถอดรหัสอ่านข้อมูลหมายเลข 2BFH	12
การทำงานของ IC 74LS244	13
พอร์ตส่งข้อมูลหมายเลข 2BEH	14
การต่อหลอด LED กับพอร์ต	15
วงจรเชื่อมต่อคอมพิวเตอร์	16
วงจรถอดรหัสโดยใช้ดิปสวิตซ์เลือกหมายเลขพอร์ต	17
ลักษณะไอซีแบบ Dual in-line package(DIP)	18
โครงสร้างของไอซีดิจิตอลเบอร์ 7408	19
ลอจิกโพรบ	19
หน่วยความจำ ROM โดยสร้างจากไดโอด	20
หน่วยความจำ ROM โดยสร้างจากทรานซิสเตอร์แบบ MOS	21
โครงสร้างภายใน EPROM	22
โครงสร้างของไอซี 8255	24
การถอดรหัสให้ seven-segment แสดงผล	28
สัญลักษณ์ทางลอจิกของไอซี Decoder TTL	29
โครงสร้างภายในของไอซีเบอร์ 7447A	30
หน้าจอการเลือกรูปแบบ Project ใน VB	31
หน้าจอของ Project Properties	32
หน้าจอการเขียน ActiveX DLL	33
หน้าจอการทำเป็นไฟล์ DLL	33
วงจร NAND Gate	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
วงจร Hex Inverter	35
วงจร Hex Inverter Buffer/Driver	35
วงจร 8-Input NAND Gate	36
วงจร Quad 2-Input OR Gate	36
วงจร BCD-to-7 Segment Decoder/Driver	37
วงจร Dual 2 to 4 Line Decoder	38
วงจร Octal State Bus Driver	39
วงจร Octal D Type F-F 3 State	40
Hardlock ที่ผ่านทางพรินเตอร์พอร์ต	41
Hardlock ที่ผ่านทางพอร์ตขนาน	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางแสดงตำแหน่งของพอร์ตต่างๆ	11
ตารางแสดงการเลือกพอร์ตของ 8255	26
ตารางแสดงความหมายของบิตควบคุม	27
ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ NAND Gate	34
ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ Hex Inverter	35
ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ 8-Input NAND Gate	36
ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ Quad 2-Input OR Gate	37
ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ Dual 2 to 4 Line Decoder	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1. ความสำคัญและที่มาของปัญหาพิเศษ

ในปัจจุบัน Software ต่างๆนั้นถูกละเมิดลิขสิทธิ์ได้อย่างง่ายดาย เพียงแค่เอาแผ่นต้นฉบับมาทำการคัดลอก ก็จะมีความสามารถเท่ากับตัวต้นฉบับแล้ว จึงเป็นการสูญเสียทางปัญญาอย่างมาก แล้วยังมีผลกระทบต่อเศรษฐกิจโดยรวม ยิ่งทางประเทศของเราเป็นประเทศที่ไม่ใช่ประเทศที่ผลิต Software ขึ้นมาใช้เอง เราต้องมีการนำเข้ามาจากต่างประเทศ แต่ทางเรากลับนำมาทำการคัดลอกโดยผิดกฎหมาย ทำให้คนไทยเองก็ยิ่งไม่มีทางที่จะพัฒนา Software ขึ้นมาใช้เอง เพราะตระหนักว่าถึงแม้จะพัฒนา Software ขึ้นมาก็ไม่สามารถนำไปจำหน่าย เพื่อหากำไรมาพัฒนา Software ให้มีประสิทธิภาพมากขึ้นได้ จึงอยากให้มีการพัฒนาอุปกรณ์ที่จะสามารถจะป้องกัน Software ไม่ให้ถูกคัดลอกได้โดยง่ายนัก

2. วัตถุประสงค์ของการทำปัญหาพิเศษ

- เพื่อป้องกันการลักลอบและละเมิดลิขสิทธิ์ของ software
- เพื่อให้เข้าใจหลักในการสร้าง Hard lock โดยใช้การติดต่อ input และ output ทาง Printer port และ ทาง System Bus (ISA port)

3. ขอบเขตของปัญหาพิเศษ

- สามารถติดต่อกับ Hard lock โดยผ่านทาง Printer port และ ISA port และสามารถควบคุม Input ว่าจะถูกกำหนดให้มีค่าเท่าใด เพื่อให้สอดคล้องกับโปรแกรมที่เราต้องการจะ Run
- สามารถสร้าง Hard Lock ที่ใช้ผ่านทาง Printer port แบบเข้ารหัสตายตัวได้
- สามารถสร้าง Hard Lock ที่ใช้ผ่านทาง ISA port แบบผ่านตัวถอดรหัสได้
- สามารถเขียนโปรแกรมเพื่อนำไปให้ผู้อื่นที่ไม่มีความรู้ด้านการติดต่อกับพอร์ตต่างๆ นำไปประยุกต์กับโปรแกรมต่างๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขั้นตอนในการดำเนินงาน

1. ศึกษาพื้นฐานความรู้เบื้องต้นทางด้านดิจิตอล
2. ศึกษาพื้นฐานความรู้เกี่ยวกับ Printer port
3. ทำการเลือกภาษาเพื่อใช้ในการติดต่อทาง Printer port
4. ศึกษาเรื่อง Port input/output
5. ศึกษาเรื่อง ROM
6. ศึกษาเรื่อง ไอซีเบอร์ต่างๆ
7. เขียนโปรแกรมโดยใช้ภาษาที่เลือก
8. ทดสอบการทำงานและปรับปรุงแก้ไข

5. ประโยชน์ที่คาดว่าจะได้รับ

1. ป้องกันการละเมิดลิขสิทธิ์ของ software ต่างๆที่มีการเรียกใช้ Hard lock
2. ทำให้เข้าใจถึงระบบการติดต่อทาง Printer port และ ISA port
3. เป็นแนวทางของรูปแบบปัญหาเพื่อจะใช้ในปีต่อไป

6. อุปกรณ์ที่ใช้ทำในปัญหาพิเศษ

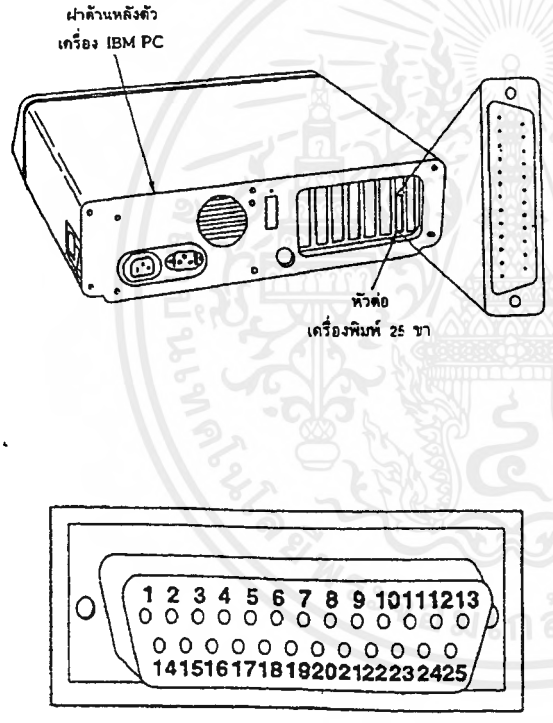
1. คอมพิวเตอร์
2. Printer port
3. อุปกรณ์ในการเชื่อมต่อวงจรอีเลกทรอนิกส์
4. IC ต่างๆ
5. ISA board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง

2.1 เทคนิคการเชื่อมต่อวงจร DAC 8 ช่องเข้ากับพรินเตอร์พอร์ต IBM PC

พรินเตอร์พอร์ต IBM PC เราสามารถประยุกต์นำมาใช้งานเป็นทางเข้าของข้อมูล (อินพุต พอร์ต) และทางส่งข้อมูลออก (เอาต์พุต พอร์ต) ได้ โดยไม่จำเป็นต้องสร้างฮาร์ดแวร์อินพุต/เอาต์พุตพอร์ตขึ้นมาใหม่อีกบน IBM PC รายละเอียดของหัวต่อเครื่องพิมพ์ 25 ขา และวงจรพรินเตอร์พอร์ต แสดงดังรูปที่ 2.1 และรูปที่ 2.2 ตามลำดับ

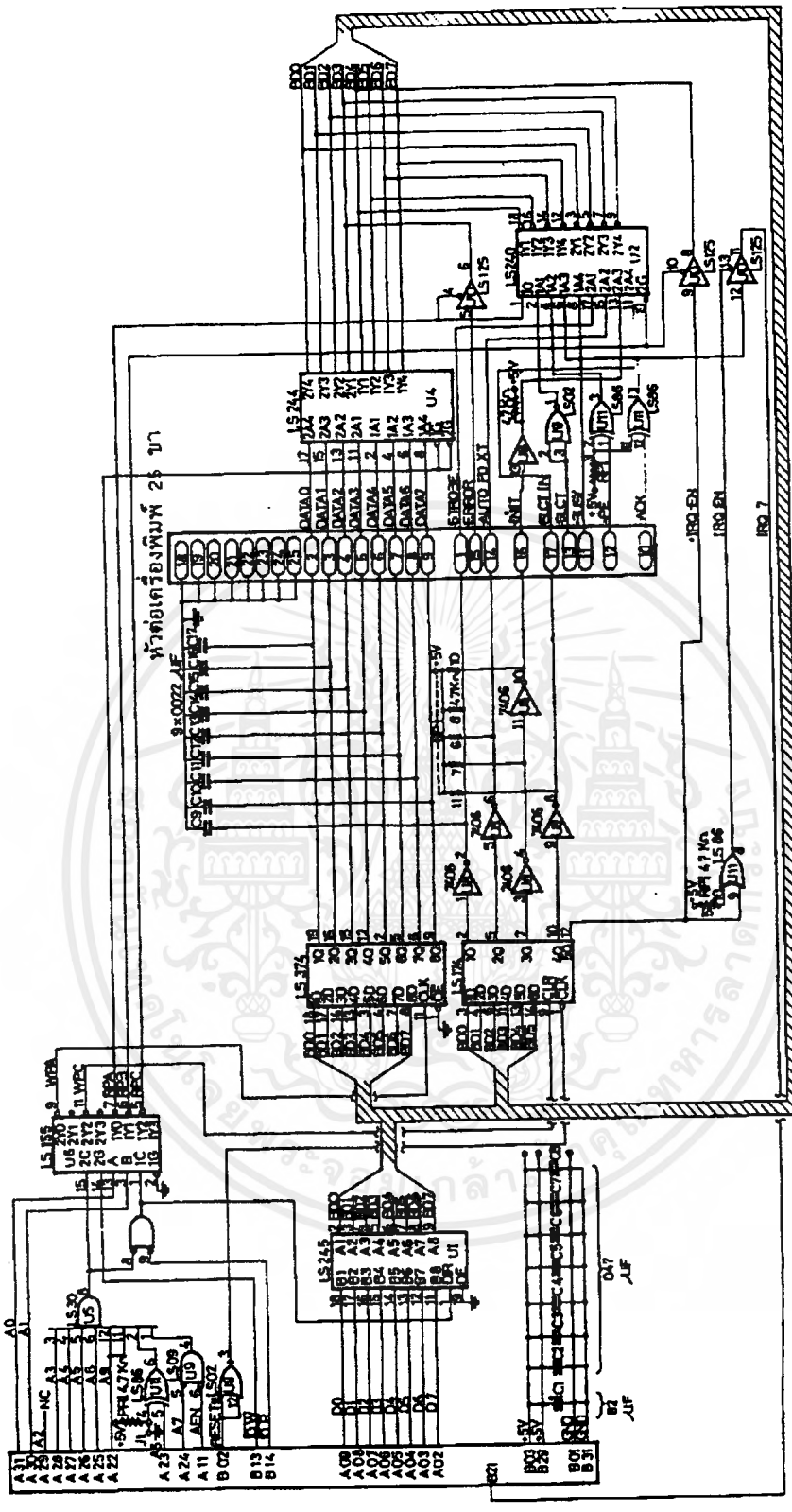


Printer Pins

Pin	Description
1	Strobe
2	Data bit 0
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7
10	Acknowledge
11	Busy
12	Out of paper
13	Select
14	Auto feed
15	Error
16	Initialize printer
17	Select input
18	Ground
19	Ground
20	Ground
21	Ground
22	Ground
23	Ground
24	Ground
25	Ground

รูปที่ 2.1 หัวต่อเครื่องพิมพ์ 25 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 วงจรพรินเตอร์พอร์ตบนเครื่อง IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรข้างต้นในรูปที่ 2.2 จะพบว่าข้อมูลขนาด 8 บิตจะถูกแลตช์ผ่านโดย IC 74LS374 ทางขา 2, 3, 4, 5, 6, 7, 8 และ 9 ของหัวต่อ 25 ขา ตามลำดับ ซึ่งสามารถอ่านสถานะบัฟเฟอร์ข้อมูลนั้นได้โดย IC 74LS244 แอดเดรสที่ถูกดีโค้ดไว้ในที่นี้ สำหรับบอร์ดอแดปเตอร์เครื่องพิมพ์คือหมายเลข 378H (สำหรับบอร์ดอแดปเตอร์โมโนโครมคือหมายเลข 3BCH)

ส่วนข้อมูลเข้าขนาด 5 บิต(XXXD3D4D5D6D7)นำเข้าได้ทางขา 10(Ack), 11 (Busy), 12 (PE), 13 (Select), 15 (Error) โดย IC 74LS240 และ 74LS125 แอดเดรสที่ถูกดีโค้ดไว้สำหรับบอร์ดอแดปเตอร์เครื่องพิมพ์คือ หมายเลข 389H (สำหรับบอร์ดอแดปเตอร์โมโนโครมคือหมายเลข 3BDH)

นอกจากนี้ยังมีขา 10(Strobe) และขา 16 (Reset) ที่สามารถใช้งานในการ Strobe date และreset data ได้อีก เป็นส่วนที่เพิ่มเติมที่นอกเหนือจาก เอาต์พุต data ขนาด 8 บิตดังกล่าว

การเชื่อมต่อวงจร DAC 8 ช่อง เข้ากับพรินเตอร์พอร์ต IBM PC แสดงรายละเอียดในการเชื่อมต่อวงจรแลตช์ข้อมูล 8 ช่อง เข้ากับพรินเตอร์พอร์ต และการเชื่อมต่อวงจรแลตช์ข้อมูลเข้ากับ IC เบอร์ DAC 0808

จะพบว่า ขาที่ 11, 12 ซึ่งเป็นขาสัญญาณ Busy และ PE ถูกต่อลงกราวด์ (low) และขา 13, 15 เป็นสัญญาณ Error และ Select ถูกต่อให้สัญญาณเป็น high โดยที่เอาต์พุตข้อมูลขา 2 ถึงขา 9 จะใช้เป็นทางส่งข้อมูลออกผ่านทางบัฟเฟอร์ข้อมูล IC 74LS244 ส่วนขา 16 ซึ่งเป็นสัญญาณ Reset จะต่อเข้ากับขา 4 (PR1) และ (CLR2) (ผ่าน inverter gate IC 74LS14) ของ IC 74LS74 2-flip flop เพื่อใช้ในการเคลียร์และส่งสัญญาณอีนาเบิลขา 1 (OE) ของ IC 74LS383 ตามลำดับ

ส่วนขา 1 ซึ่งเป็นสัญญาณสโตรบ (ผ่านทาง Inverter gate IC 74LS14) เข้าขา 11 (CLK) ของ IC 74LS374 เพื่อใช้ในการแลตช์ข้อมูลที่ขาขึ้น ซึ่งเอาต์พุตของ IC 74LS374 จะแลตช์ข้อมูลแก่ IC6-IC13 (74LS373) การที่จะเลือก IC ช่องใดนั้นจะขึ้นอยู่กับ IC5 (IC เบอร์ 74LS373 เช่นกัน) นอกจากสัญญาณ Strobe จะต่อเข้ากับขา 11 (CLK) ของ IC2 ดังกล่าวแล้วยังต่อเข้ากับขา 3 (CLK2), ขา 11 (CLK1) ของ IC3 (74LS74) และขา 9 (A2)
ขา 2 (B1) ของ IC4 (74LS221) ซึ่งทำหน้าที่เป็น monostable multivibrator คู่ เพื่อผลิตสัญญาณทริกแคบ ๆ ออกทางขา 13 ความกว้าง 1 μsec (สู่ขา 10 (Ack) ของพรินเตอร์พอร์ตเพื่อเป็นการใช้เช็คเมื่อทำงานเสร็จ) และขา 5 ความกว้าง 100 μsec (สู่ขา 11 (Latch) ของ IC 74LS373 เพื่อเป็นการเลือกช่องแลคช์ข้อมูลสู่วงจร DAC) ตามลำดับ

IC DAC 0808 ซึ่งได้ขยายสัญญาณอนาลอกเอาต์พุตได้โดย Op-Amp IC LF353 (high speed อินพุต impedance) และค่า R ปรับได้ (R10) ในส่วนแรงดันอ้างอิงนั้นได้ใช้ LM329

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

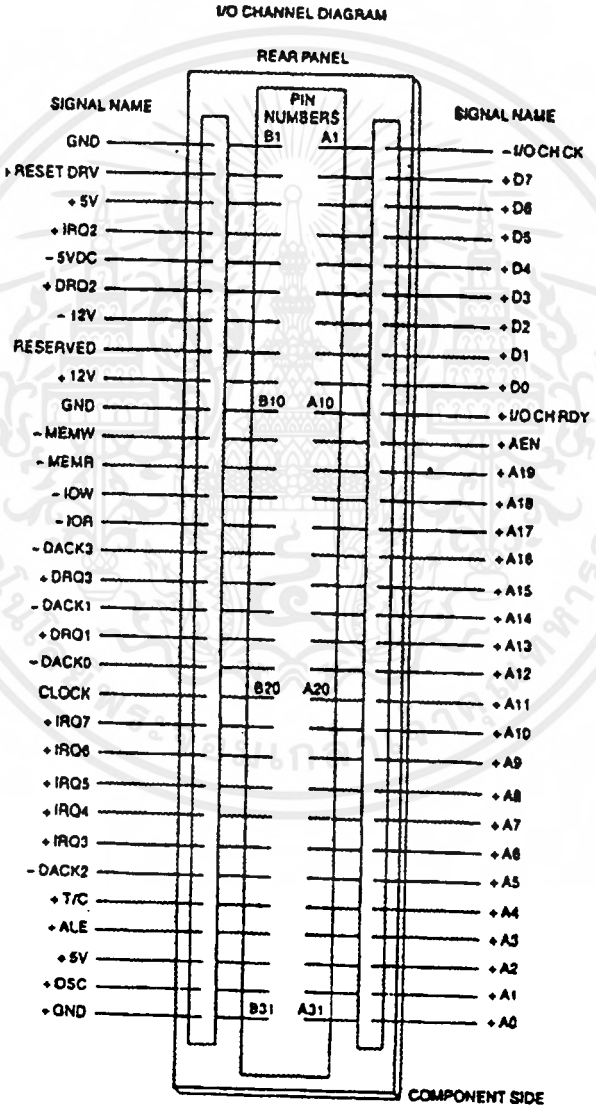
ขนาด 6.9 โวลต์ ต่อเข้ากับขา $+V_{EF}$ ผ่าน R8 (3.3K) มี clamping diode D1 ต่อคร่อมระหว่าง $+V_{EF}$ กับกราวด์อีกทอดหนึ่ง

การเขียนโปรแกรมใช้งานของ DAC 8 ช่อง สามารถกระทำได้โดยขั้นแรกส่งสัญญาณ low เข้าขา Reset เพื่อทำการเคลียร์ (clear) ข้อมูลเก่า ขั้นที่สองจึงส่งข้อมูลที่จะเลือกทำงานกับ DAC ช่องใดๆ ก็ให้สัญญาณเป็น high กับช่องนั้น จากนั้นจึงส่งสัญญาณ low เข้าขา Strobe เพื่ออีนาเบิลการเลือกช่องของ DAC ขันที่สามทำการส่งข้อมูลดิจิตอล สำหรับช่องนั้นๆ ออกทางขา D0-D7 อีก แล้วจึงส่งสัญญาณ low เข้าขา Strobe อีกครั้ง สัญญาณที่จะเช็คว่าการทำงานทั้งสามขั้นตอนสิ้นสุดลง เพื่อที่จะได้ทำงานส่งข้อมูลออกในรอบใหม่ สามารถทำได้โดยการตรวจสอบสัญญาณที่เข้าทางขา Ack ถ้าสัญญาณเปลี่ยนจาก low เป็น high เมื่อไร? ก็สามารถทำงานตามขั้นตอนแรกได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การเชื่อมต่อไมโครคอมพิวเตอร์

เครื่องคอมพิวเตอร์ IBM PC ที่เราใช้กันในปัจจุบันนอกจากใช้ในการเขียนโปรแกรมเพื่อประมวลผลต่างๆ แล้วยังสามารถใช้ในการควบคุมอุปกรณ์ต่างๆ ได้อีกด้วย โดยติดต่อทางพอร์ต (พอร์ต) ของเครื่องคอมพิวเตอร์ ข้อมูลที่จะติดต่อกับคอมพิวเตอร์จะเป็นสัญญาณดิจิตอล ส่งเข้าทางบัสดาต้า (Data Bus) ของคอมพิวเตอร์ ในที่นี้จะกล่าวถึงการสร้างพอร์ตขนาน สำหรับรับข้อมูลและส่งข้อมูล (parallel อินพุต/เอาต์พุต (I/O) พอร์ตs)



รูปที่ 2.3 แสดงสัญญาณต่าง ๆบน System Bus ของ IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 PROTOTYPE BOARDS

การสร้างพอร์ตสำหรับเชื่อมต่ออุปกรณ์ภายนอกเข้ากับเครื่องไมโครคอมพิวเตอร์ จะต้องสร้างแผงวงจรสำหรับติดต่อกับสัญญาณต่างๆ ที่จำเป็นบน IBM PC System Bus ในที่นี้จะกล่าวถึงระบบบัสที่ใช้กับ CPU เบอร์ 8088 เป็นหลัก ถ้าออกแบบวงจรให้สามารถใช้กับระบบนี้ได้แล้วก็สามารถที่จะใช้กับเครื่องคอมพิวเตอร์ที่ใช้ CPU เบอร์สูงกว่านี้ได้ พอร์ตที่สร้างขึ้นจะประกอบด้วยวงจรบนแผง PCB ที่สามารถเสียบลงบน system bus ได้ เรียกว่า Prototype Board สัญญาณต่างๆบน system bus แสดงได้ดังรูปที่ 2.3 ซึ่งสัญญาณจะมีค่าอยู่ในระดับ transistor-transistor logic (TTL) นอกจากนี้ยังมีขาที่จ่ายพลังงาน และ ground รวมอยู่ด้วย

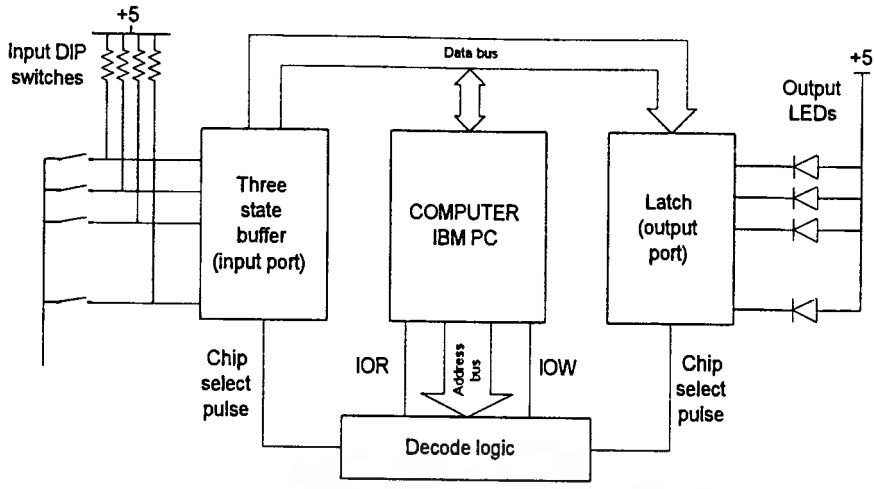
2.2.2 สัญญาณต่าง ๆที่สำคัญในการสร้างพอร์ตมีดังนี้

1) A0 – A19 มี 20 ขาเป็นขาสำหรับอ้างตำแหน่งของหน่วยความจำ (memory) และพอร์ตอินพุตเอาต์พุต
2) D0 – D7 เป็นขาข้อมูล (DATA) ขนาด 8 บิต สำหรับส่งและรับข้อมูลเข้าบัสข้อมูลของ CPU โดยจะส่งข้อมูลไปที่เอาต์พุตพอร์ต ถ้าสัญญาณ IOW ทำงาน หรือเขียนข้อมูลไปที่หน่วยความจำถ้าสัญญาณ MEMW ทำงาน และจะรับข้อมูลจากอินพุตพอร์ต ถ้าสัญญาณ IOR ทำงานหรืออ่านข้อมูลจากหน่วยความจำถ้าสัญญาณ MEMR ทำงาน
3) MEMR , MEMW , IOR , IOW เป็นสัญญาณที่ แอคทีฟ – low ใช้ควบคุมการอ่านหรือเขียนข้อมูลลงหน่วยความจำ และ พอร์ต I/O
4) AEN (แอดเดรส enable) จะแอคทีฟลอจิก 1 เครื่อง IBM PC จะใช้สัญญาณนี้เป็นตัวแบ่งว่าสัญญาณแอดเดรสที่เกิดขึ้นมาจากที่ใด ถ้าเป็นลอจิก “1” แสดงว่า เกิดการทำ DMA (ย่อมาจาก Direct Memory Access) ซึ่งเป็นการรับส่งข้อมูลโดยตรงระหว่างอุปกรณ์ อินพุต/เอาต์พุต ของเครื่องคอมพิวเตอร์ กับหน่วยความจำโดยไม่ผ่าน CPU ถ้าเป็นลอจิก “0” แสดงว่าพอร์ต I/O มีค่าแอดเดรสตรงกับบัสแอดเดรส

2.2.3 การออกแบบสร้างพอร์ตรับส่งข้อมูลแบบขนาน(PARALLEL I/O พอร์ต DESIGN)

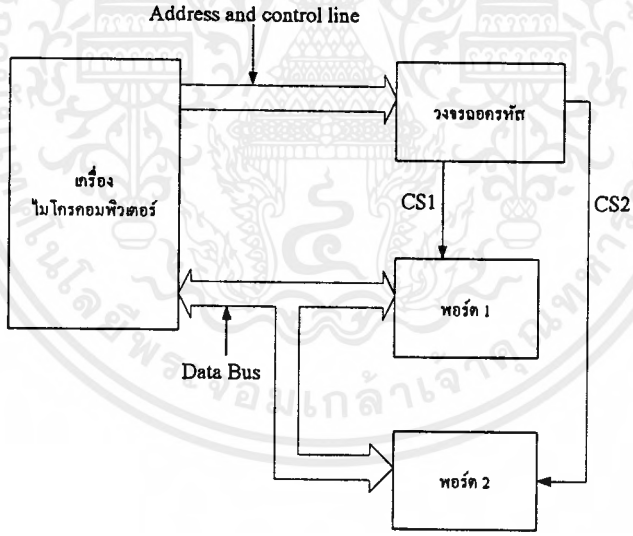
ในที่นี้จะยกตัวอย่างว่า ระบบที่จะออกแบบคือเครื่องคอมพิวเตอร์ IBM PC จะอ่านข้อมูลจากพอร์ตอินพุต คือสวิตช์ จากนั้นจะเอาข้อมูลที่อ่านได้ส่งไปยังพอร์ตเอาต์พุต โดยแสดงผลทางหลอด LED ซึ่งระบบแสดงได้ดังรูปที่ 2.4

ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงระบบที่รับข้อมูลจากสวิตซ์ และแสดงข้อมูลทาง LED

เนื่องจากคอมพิวเตอร์สามารถมีพอร์ต ต่ออยู่ได้เป็นจำนวนมาก ดังนั้นการสร้างพอร์ต ต่อเข้ากับไมโครคอมพิวเตอร์ จะถอดรหัสเสียก่อน ว่าระบบที่สร้างขึ้นต้องการติดต่อกับพอร์ตใด โดยสร้างเป็นวงจรถอดรหัสขึ้น จากขาแอดเดรสและสัญญาณควบคุมของไมโครคอมพิวเตอร์



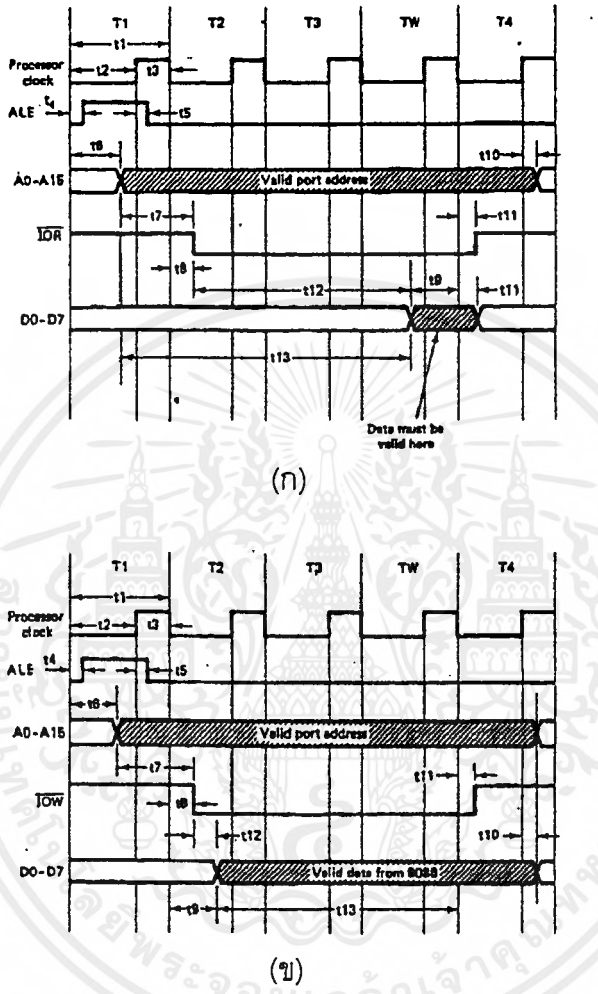
รูป 2.5 แสดงระบบเชื่อมต่อกับพอร์ต 2 พอร์ต

จากรูปที่ 2.5 ถ้าออกแบบระบบให้พอร์ต1 เป็นพอร์ตอินพุตและพอร์ต2 เป็นพอร์ตเอาต์พุต โดยพอร์ตแต่ละพอร์ต จะมีเลขประจำตัวอยู่ที่เรียกว่า memory-mapped I/O ถ้าคอมพิวเตอร์ต้องการติดต่อกับพอร์ตใด ก็ส่งสัญญาณแอดเดรสออกมาถ้าวงจรถอดรหัสถอดรหัสสัญญาณแอดเดรส พบว่าตรงกับพอร์ต 1 ก็จะส่งสัญญาณ CS1 ให้แอคทีฟ เพื่อให้พอร์ต 1 ทำงาน แต่ถ้า วงจรถอดรหัสพบว่าสัญญาณ แอดเดรส ตรงกับ พอร์ต 2 ก็จะทำสัญญาณ CS2 แอคทีฟ เพื่อให้ พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ทำงาน แต่ถ้าสัญญาณ แอดเดรส ที่ส่งออกมาไม่ตรงกับ พอร์ต ใดเลย พอร์ต ทั้งสองนี้ก็จะไม่ทำงาน

การที่คอมพิวเตอร์ต้องการติดต่อกับอุปกรณ์ I/O หรือพอร์ต คอมพิวเตอร์จะส่งสัญญาณต่างๆออกมา ดังรูปที่ 2.6

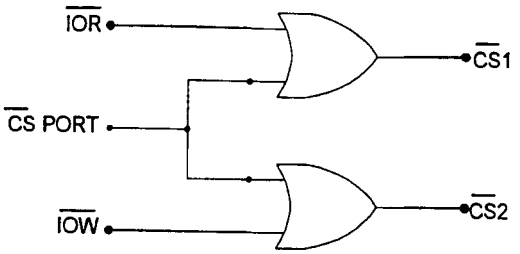


รูปที่ 2.6 แสดงสัญญาณเวลาการติดต่อกับพอร์ตของคอมพิวเตอร์

จากรูป 2.6 ก) คือ cycle timing ในการอ่าน พอร์ต อินพุต จะเห็นว่าในช่วงแรกของการอ่าน พอร์ต CPU จะส่งสัญญาณ แอดเดรส ออกมา จากนั้นสัญญาณ IOR จะ แอคทีฟ เพื่อบอกว่าต้องการจะอ่านข้อมูล เวลาต่อมา Data Bus ก็จะรับข้อมูลเข้า CPU สำหรับรูป 2.6 ข) เป็น cycle timing ในการส่งข้อมูลออกไปที่ พอร์ต ช่วงเวลาแรก CPU จะส่งสัญญาณ แอดเดรส ออกมา จากนั้นสัญญาณ IOW จะ แอคทีฟ เพื่อที่จะเขียนข้อมูลจาก Data Bus ไปยัง พอร์ต สัญญาณต่างๆเหล่านี้เกิดได้จากการเขียนโปรแกรม

จากรูป 2.5 เราสามารถออกแบบให้ พอร์ตอินพุต และ พอร์ต เอาต์พุต มีหมายเลขเดียวกันได้ แต่ให้ทำงานในเวลาต่างกัน วงจรถอดรหัส.สำหรับ CS1 และ CS2 อาจออกแบบได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงระบบติดต่อกับกับพอร์ต 2 พอร์ต

โดยสัญญาณ CS นำมาจากวงจรถอดรหัสหมายเลข พอร์ต เช่น ถอดรหัสไว้หมายเลข 300 ถ้าคอมพิวเตอร์ต้องการส่งข้อมูลออกไปยัง พอร์ต หมายเลข 300 สัญญาณ CS จะ แอคทีฟ ขณะเดียวกันจากรูป 2.6 ก) สัญญาณ IOR จะ แอคทีฟ ด้วย ทำให้ CS1 แอคทีฟ จึงอ่านข้อมูลเข้ามาได้

การรับส่งข้อมูลทางพอร์ตนั้น จะต้องมีการจัดตำแหน่งพอร์ตที่ไม่ซ้ำซ้อนกัน แต่ละพอร์ตจะต้องมีแอดเดรสหรือหมายเลขพอร์ตที่แน่นอนซึ่งใน IBM PC / XT จะใช้สัญญาณแอดเดรส A0 - A9 ในการติดต่อกับพอร์ต สำหรับการ์ด (card) และพอร์ตต่างๆที่เชื่อมต่อกับเครื่องคอมพิวเตอร์จะมีหมายเลขพอร์ตดังตารางต่อไปนี้

รหัสพอร์ต (HEX)	พอร์ตที่ใช้
000-00F	DMA chip 8253A – 5
020-021	Interrupt 8259A
040-043	Timer 8253 – 5
060-063	PPI 8255A – 5
080-083	DMA page registers
0Ax	NMI mask register
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
380-38C	SDLC communications
380-389	Binary synchronous communications (2)
3A0-3A9	Binary synchronous communications (1)
3B0-3BF	IBM monochrome display / printer
3C0-3CF	Reserved
3D0-3DF	Colour / graphic
3E0-3F7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous communications (1)

ตารางที่ 1 แสดงตำแหน่งของพอร์ตต่าง ๆที่ใช้ในเครื่อง IBM

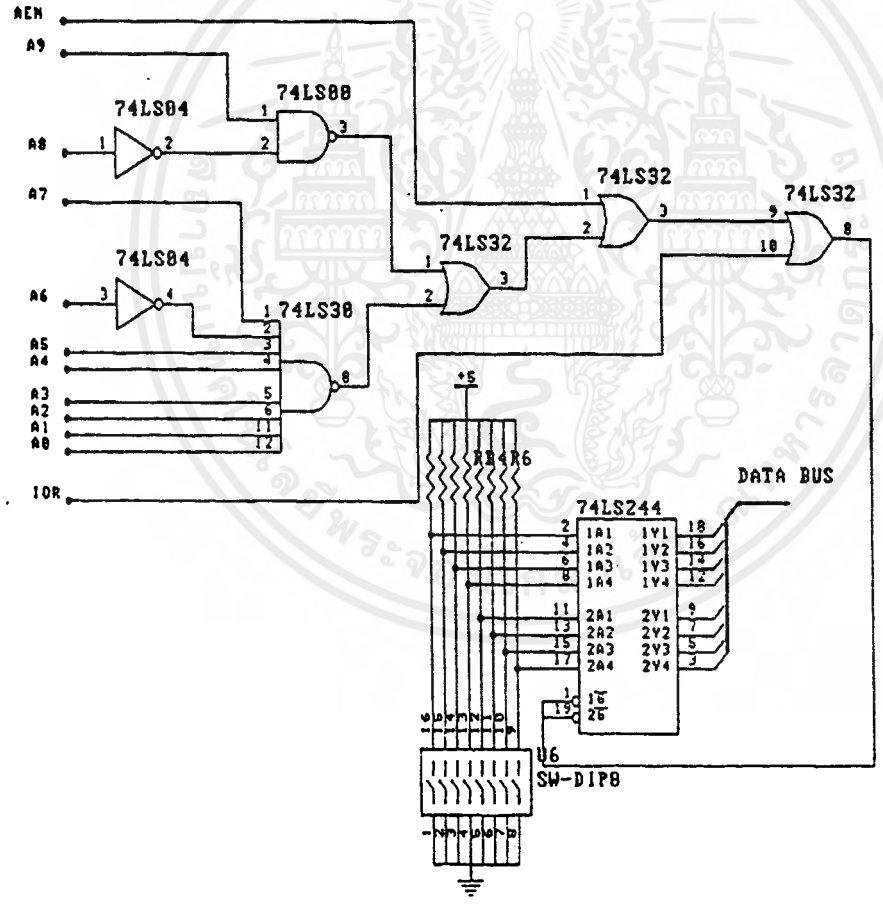
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นว่าถ้า A9 เป็น 0 (พอร์ตหมายเลข 000H – 1FFH)จะเป็นการติดต่อกับพอร์ตบนเมนบอร์ด ดังนั้นถ้าจะออกแบบสร้างการ์ดจะต้องเลือกแอดเดรสของพอร์ตให้ A9 เป็น 1

2.2.4 การสร้างพอร์ตรับข้อมูลแบบขนาน

ก่อนอื่นต้องออกแบบวงจรถอดรหัสหมายเลขพอร์ตก่อน ซึ่งจะต้องเป็นหมายเลขพอร์ตที่ว่าง สมมุติว่าเลือกพอร์ตหมายเลข 2BF ดังนั้น สัญญาณ A0 – A9 จะต้องเป็นดังนี้

A3	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	1	1	1
2		B				F			



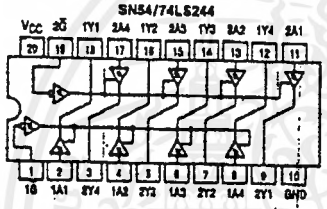
รูปที่ 2.8 วงจรถอดรหัสอ่านข้อมูลหมายเลข 2BFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะเดียวกันถ้า IBM PC ต้องการติดต่อกับพอร์ต I/O สัญญาณ AEN จะต้องเป็น 0 ดังนั้นวงจรถอดรหัสอาจออกแบบได้ดังรูปที่ 2.8 จาก cycle timing ในการอ่านข้อมูลจากพอร์ตจะเห็นว่า ถ้า IBM PC ต้องการอ่านข้อมูลจากพอร์ต สัญญาณ IOR จะต้องเป็น 0 ดังนั้นจะต้องเอาสัญญาณ IOR มาร่วมถอดรหัสด้วย

สัญญาณที่ได้จากจุด A อาจเรียกได้ว่า สัญญาณอีนาเบิ้ล (enable) สำหรับไปอีนาเบิ้ลอุปกรณ์ที่ใช้เป็นอินพุตพอร์ตต่อไป

อุปกรณ์ที่จะสร้างเป็นอินพุตพอร์ตได้จะต้องใช้ IC ที่เป็นบัปเฟอร์ มีลอจิก 3 สถานะ โดยสัญญาณควบคุมพอร์ตอินพุตจะเป็นตัวเปิดเกต เพื่อนำข้อมูลเข้าสู่บัสข้อมูล และ CPU จะอ่านเข้าไป ถ้าคอมพิวเตอร์ยังไม่พร้อมที่จะรับข้อมูลเข้า เอาต์พุตของ IC จะต้องอยู่ในสถานะ Hi – Z ตัว IC ที่นิยมใช้เป็นพอร์ตรับข้อมูลได้แก่ เบอร์ 74LS244 ซึ่งมีลักษณะการทำงานดังนี้



SN54/74LS244

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

GUARANTEED OPERATING RANGES

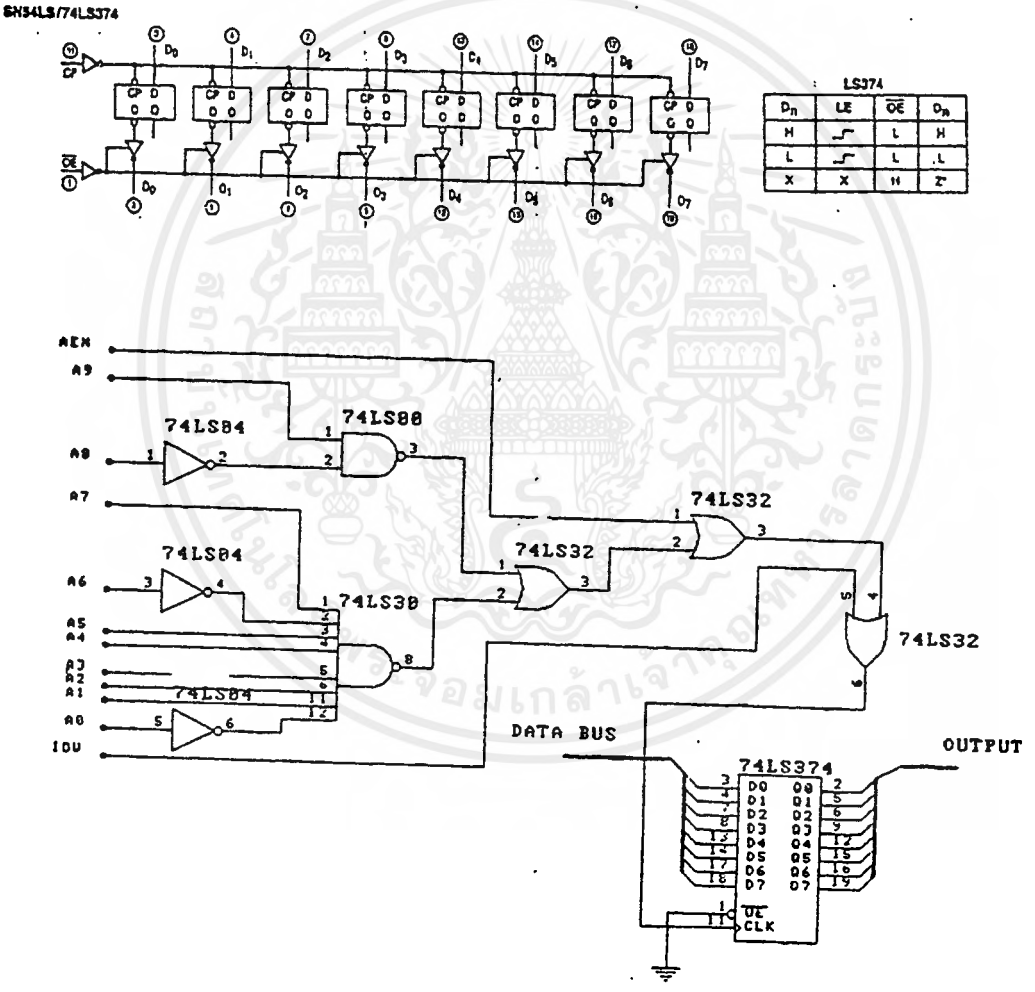
Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

ถ้านำสัญญาณอีนาเบิ้ลมาต่อกับขา 1 และ ขา 19 ถ้าคอมพิวเตอร์ต้องการติดต่อกับพอร์ต 2BF สัญญาณอีนาเบิ้ลจะเป็น “0” ข้อมูลจากอินพุตจะถูกส่งเข้าบัสข้อมูลได้ แต่ถ้าเป็นลอจิก “1” (ไม่ต้องการติดต่อกับพอร์ต 2BF) ขาเอาต์พุตของ 74LS244 ซึ่งต่ออยู่กับบัสข้อมูลจะเป็น Hi – Z วงจรแสดงได้ดังรูป ซึ่งจะรับข้อมูลจากดิปสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 การสร้างพอร์ตขนานสำหรับส่งข้อมูลออก

จะคล้ายกับการสร้างพอร์ตรับข้อมูล คือจะต้องมีวงจรถอดรหัสหมายเลขพอร์ต จากนั้นมาถอดรหัสร่วมกับสัญญาณ IOW เพื่อเป็นวงจรอีนาเบิ้ลต่อไป สำหรับการเลือกใช้ IC สำหรับเอาต์พุตพอร์ต จะต้องใช้ IC ที่สามารถ LATCH ข้อมูลได้ เนื่องจากคอมพิวเตอร์ทำงานเร็วมาก เอาต์พุตจะต้องคงค่าข้อมูลไว้ชั่วขณะให้อุปกรณ์ภายนอกสามารถรับข้อมูลได้ทัน IC ที่นิยมใช้ได้แก่เบอร์ 74LS373 , 74LS374 เป็นต้น ซึ่งเป็น IC D – TYPE FLIP-FLOPS 8 ตัวในชิพเดียว การทำงานและขาของ 74LS374 เป็นดังนี้



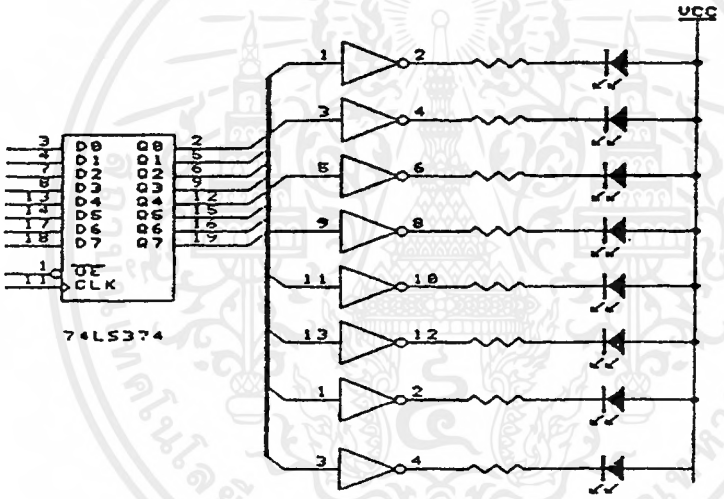
รูปที่ 2.9 แสดงพอร์ตส่งข้อมูลหมายเลข 2BEH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้จะถอดรหัสสำหรับเอาต์พุตพอร์ตเป็นหมายเลข 2BE ซึ่งสัญญาณ A0 – A9 จะต้องเป็นดังตารางข้างล่างนี้ วงจรสำหรับเอาต์พุตพอร์ตหมายเลข 2BE อาจเขียนได้ดังรูป 2.9

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

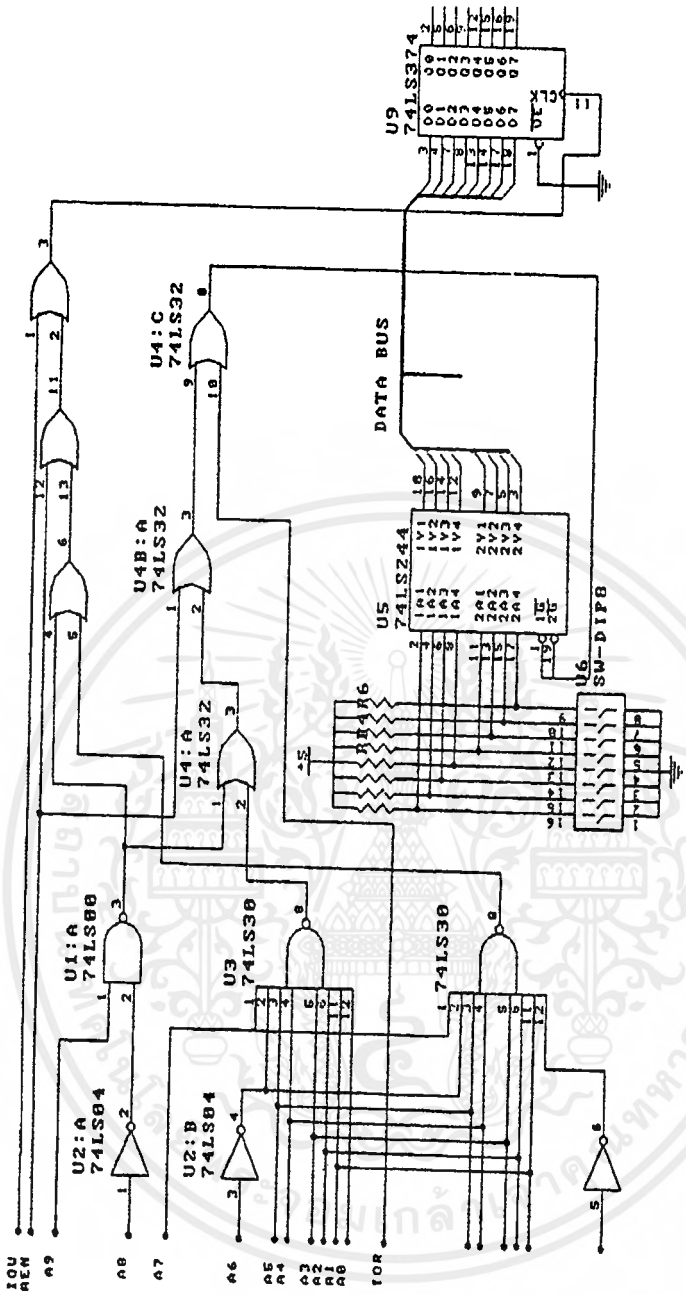
ถ้าต้องการส่งข้อมูลมาแสดงผลที่ LED สามารถต่อจาก 74LS374 ได้ดังรูปที่ 2.10 เนื่องจาก LED ต้องการกระแสไหลผ่านตัวมันประมาณ 20 mA จึงต้องใช้ IC 74LS06 มารับค่ากระแส ซึ่ง IC ตัวนี้จะมีเอาต์พุตเป็นแบบ OC ซึ่งสามารถรับกระแสเข้าได้มาก



รูปที่ 2.10 แสดงการต่อหลอด LED กับพอร์ต

จากรูปที่ 2.10 เราสมมุติให้ IBM PC รับข้อมูลจากอินพุตพอร์ตซึ่งเป็นดิปสวิตช์ และส่งข้อมูลออกทางเอาต์พุตพอร์ตคือ LED จะสามารถเขียนวงจรได้ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

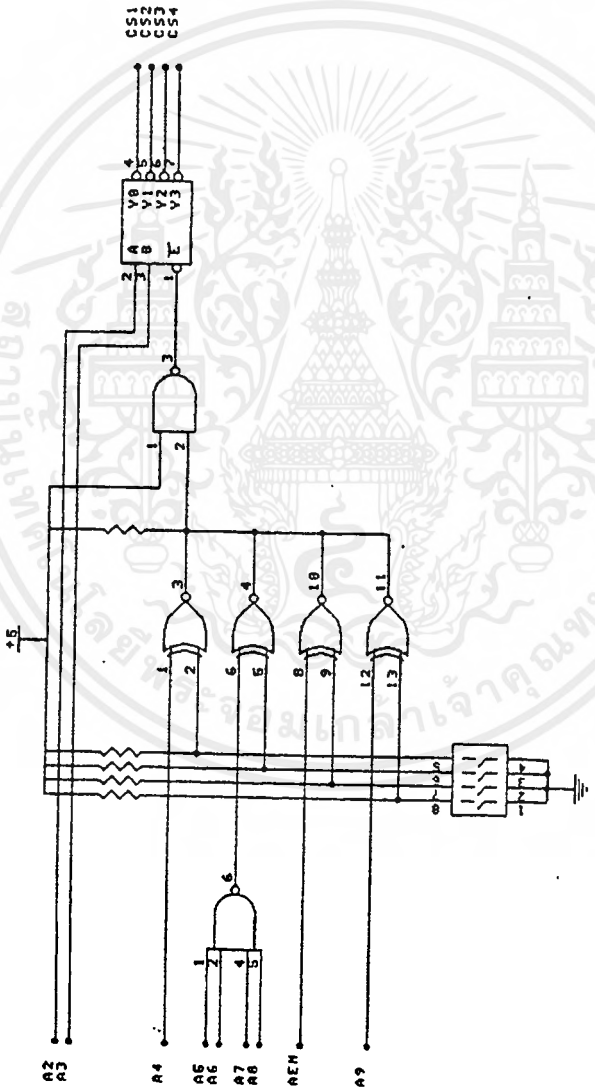


รูปที่ 2.11 แสดงวงจรเชื่อมต่อคอมพิวเตอร์ประกอบด้วยพอร์ตรับและส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การถอดรหัสพอร์ตแบบอื่น ๆ

จากที่ผ่านมาจะเห็นว่าถ้าเราถอดรหัสเบอร์ 2BF หรือ 2BE ถือว่าเป็นการถอดรหัสแบบ Fixed ซึ่งไม่สามารถเปลี่ยนหมายเลขพอร์ตเป็นหมายเลขอื่นได้ ถ้าในคอมพิวเตอร์มีแผงวงจรอื่นที่สร้างไว้เลือกใช้พอร์ตเบอร์ 2BF ด้วยจะทำให้เกิดความผิดพลาดขึ้น เพราะหมายเลขพอร์ตซ้ำกัน ดังนั้นการออกแบบวงจรถอดรหัสที่ดีจะต้องสามารถเลือกได้ว่าต้องการใช้หมายเลขพอร์ตหมายเลขใด โดยทั่วไปนิยมใช้ดิปสวิตซ์ (Dip - Switch) เป็นตัวเลือกและถ้าต้องการได้จำนวนพอร์ตเพิ่มจะใช้ IC ถอดรหัสร่วมด้วยซึ่งอาจออกแบบวงจรได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

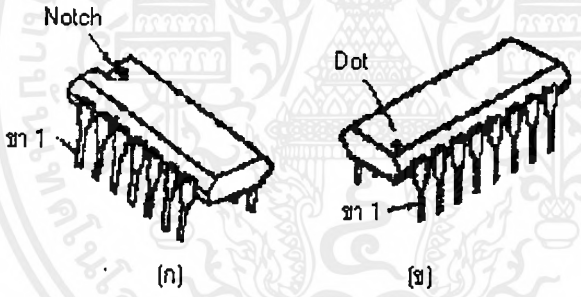
33861

2.3 ไอซีลอจิกเกตแบบ TTL

ไอซีที่ใช้ในงานดิจิตอลโดยทั่วไปจะหาได้ง่ายและราคาไม่แพง โดยจะสร้างออกมาเป็นตระกูลของไอซี โดยในแต่ละตระกูลของไอซีจะใช้เชื่อมต่อกันได้

ไอซีดิจิตอลกลุ่มแรก จะสร้างจากเทคโนโลยีแบบไบโพลา โดยภายในไอซีจะประกอบด้วย .ทรานซิสเตอร์, ไดโอด และตัวต้านทาน ไอซีดิจิตอลอีกกลุ่มหนึ่งจะเป็นตระกูลที่ใช้เทคโนโลยีแบบ Metal oxide semiconductor (MOS) ในห้องทดลองจะต้องได้ใช้ไอซีทั้งสองตระกูลนี้ ซึ่งเรียกว่า ไอซีตระกูล TTL และไอซีตระกูล CMOS โดยตระกูล CMOS จะกินพลังงานต่ำ และใช้ไฟเลี้ยงได้ในช่วงกว้าง โครงสร้างภายในจะสอดคล้องกับ insulated – gate field – effect transistors (IGFFTs)

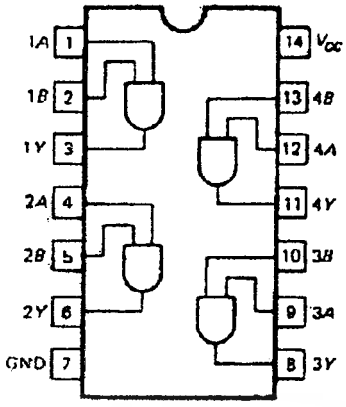
ลักษณะของไอซีดิจิตอลทั่วไป จะมีลักษณะดังรูปที่ 2.12 ก) โดยโครงสร้างแบบนี้เรียกว่า Dual inline package (DIP) ไอซีในลักษณะที่เห็นนี้เรียกว่า 14-pin DIP IC. สำหรับการกำหนดเลขขาจะอ่านแบบทวนเข็มนาฬิกา โดยเริ่มนับตั้งแต่ 1 ถึง 14 โดยมองจากด้านบนลงมา ดังรูปที่ 2.12 ก) โดยใกล้ๆตำแหน่งของขาที่ 1 จะมีจุดให้สังเกตบนด้านบนของตัวไอซี ดังรูปที่ 2.12 ข)



รูปที่ 2.12 ลักษณะไอซีแบบ Dual in-line package (DIP)
ก) ตำแหน่งของขาที่ 1 ข) เครื่องหมายจุดบอกขาที่ 1

ไอซีดิจิตอลที่สร้างขึ้น เราจะต้องรู้ขาแต่ละขาคืออะไร พิจารณารูปที่ 2.13 จะเป็นไอซีดิจิตอลที่ภายในประกอบด้วย AND เกต 2 อินพุต 4 ตัว ซึ่งเรียกว่า quadruple two-อินพุต AND gate เบอร์นี้คือ เบอร์ 7408 ซึ่งเป็นเบอร์หนึ่งในตระกูล 7400 ของไอซี TTL ไฟเลี้ยงที่จ่ายให้กับไอซีทำได้โดยต่อ GND กับขา 7 และ Vcc ต่อกับขา 14 และจะเห็นว่าขาที่เหลือทุกขาจะใช้เป็นอินพุตและเอาต์พุตของ AND เกต 4 ตัวนี้

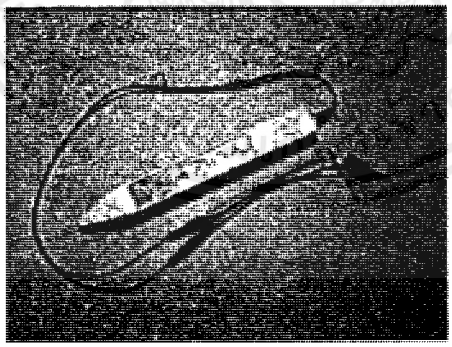
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 **โครงสร้างของไอซีดิจิตอลเบอร์** 7408

2.4 การตรวจสอบข้อผิดพลาดของวงจรเกต

เครื่องมือพื้นฐานที่ใช้หาจุดบกพร่องของวงจรคือ ลอจิกโพรบ (Logic probe) ซึ่งมีลักษณะดังรูป 3.3 ตัวลอจิกโพรบนี้จะใช้วัดไอซีได้ทั้งประเภท CMOS และ TTL โดยจะมีสวิตช์สำหรับเลือกชนิดของไอซีที่จะวัด การใช้เครื่องมือนี้จะต้องจ่ายไฟเลี้ยง ซึ่งสามารถใช้ไฟเลี้ยงของวงจรที่จะวัดได้เลย โดยจ่ายไฟบวกให้กับสายสีแดงและจ่ายไฟลบให้กับสายสีดำ การแสดงผลของลอจิกโพรบจะแสดงผลทางหลอด LED ว่าระดับลอจิกที่วัดนั้นเป็น High หรือ Low



รูปที่ 2.14 **ลอจิกโพรบ**

2.5 ROM, EPROM และ EAROMS

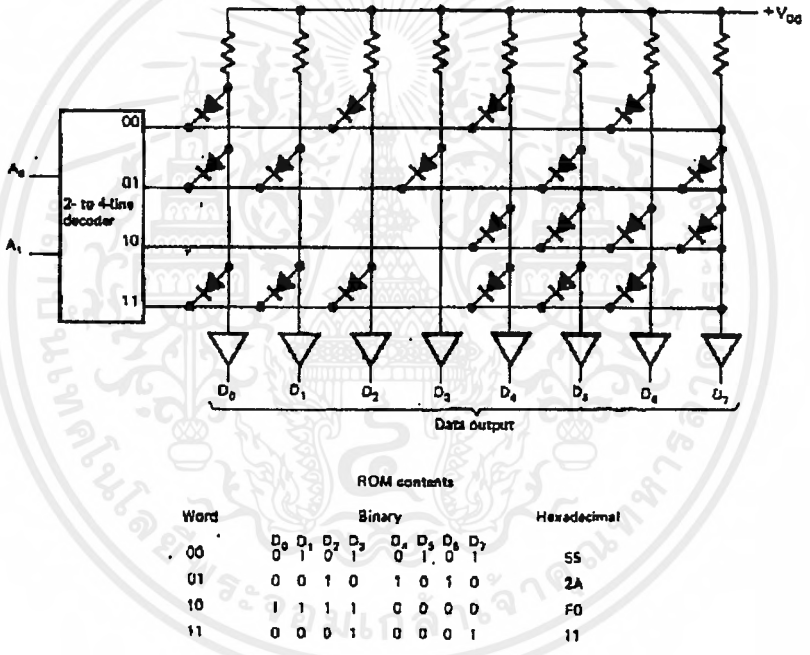
หน่วยความจำ ROM (Read-Only Memory) เป็นหน่วยความจำที่สำคัญมากในระบบไมโครโปรเซสเซอร์ หน่วยความจำชนิดนี้เราไม่สามารถเปลี่ยนแปลงข้อมูลที่บันทึกไว้ได้ ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์จะนิยมใช้หน่วยความจำ ROM สาเหตุที่เก็บโปรแกรมใน ROM ก็เพื่อไม่ให้โปรแกรมสูญหาย ระบบคอมพิวเตอร์ทุกระบบจะต้องมีหน่วยความจำ ROM เพื่อเก็บโปรแกรมที่ใช้สำหรับเริ่มต้นการทำงาน โปรแกรมประเภทนี้จะใช้ในการโหลดโปรแกรมที่เก็บอยู่ในหน่วยความจำสำรอง เช่น เทปหรือดิสก์ เข้าสู่หน่วยความจำ RAM ของระบบคอมพิวเตอร์

หน่วยความจำประเภท ROM จะใช้การเข้าถึงข้อมูลแบบสุ่มเช่นเดียวกับหน่วยความจำ RAM หน่วยความจำ ROM ที่นิยมกันมี 4 ชนิดดังต่อไปนี้

ชนิดแรกเป็น ROM แบบง่ายๆ ซึ่งเป็นไอซีที่ถูกโปรแกรมมาจากโรงงานแล้ว บางครั้งจะเรียกว่า mark-programmed ROM โรงงานผู้ผลิต ROM จะโปรแกรมข้อมูลลงไปตามที่ลูกค้าสั่ง ซึ่งจะต้องสั่งเป็นจำนวนมากเพราะราคาในการโปรแกรมมีราคาสูง หลังจากที่โปรแกรมแล้วจะไม่สามารถเปลี่ยนแปลงข้อมูลใน ROM ได้อีก



รูปที่ 2.15 หน่วยความจำ ROM ขนาด 4 เวิร์ด (4 ไบต์) โดยสร้างจากไดโอด เก็บข้อมูลได้ทั้งหมด 32 บิต บิตไดมีไดโอดต่ออยู่จะเป็นลอจิก 0 ถ้าไม่มีไดโอดจะเป็นลอจิก 1

หน่วยความจำ ROM แบบที่สองเรียกว่า EPROM (esasable programmable read-only memory) หน่วยความจำประเภทนี้ผู้ใช้สามารถโปรแกรมซ้ำได้หรือลบข้อมูลได้ แม้ว่าหน่วยความจำชนิดนี้จะมีราคาแพงกว่า ROM แต่ผู้ใช้จะสะดวกในการแก้ไขข้อมูลได้ตามต้องการ การลบหน่วยความจำ EPROM จะทำได้โดยการฉายแสงที่มีความถี่เหมาะสมเข้าไปในซิฟไอซี ปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

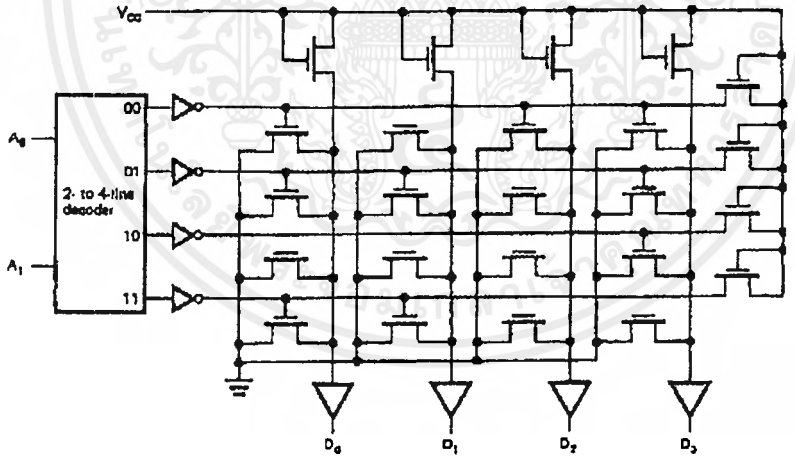
ระบบไมโครโพรเซสเซอร์ที่ผู้ใช้สามารถกำหนดรูปแบบการทำงานได้ จะใช้หน่วยความจำแบบ EPROM ผู้ผลิตสามารถแก้ไขโปรแกรมสำหรับสินค้าต่างๆได้ง่าย

หน่วยความจำ ROM แบบที่สามเรียกว่า EAROM (electrically alterable read-only memory) หน่วยความจำประเภทนี้สามารถลบข้อมูลหรือแก้ไขข้อมูลได้โดยใช้ไฟฟ้าซึ่งต่อกับ EPROM ซึ่งการแก้ไขข้อมูลจะต้องใช้อุปกรณ์ภายนอกช่วย

ในรูปที่ 2.15 แสดง ROM อย่างง่ายๆซึ่งสร้างมาจากไดโอดและไอซีถอดรหัสแบบ TTL หน่วยความจำนี้จะเก็บข้อมูลได้ 4 เวิร์ดมีขนาด 8 บิต หรือเรามองว่าเป็น ROM ขนาด 32 บิตได้ จากรูปถ้าบิตใดมีไดโอดต่ออยู่จะเป็นลอจิก 0

ตัวถอดรหัสจะมีสัญญาณเข้าทางอินพุต 2 เส้น เมื่อถูกถอดรหัสแล้วจะมีสัญญาณออก 4 เส้น ถ้าแถวใดถูกเลือกเอาต์พุตจะเป็น 0 ในแถวนั้นๆสำหรับเอาต์พุตค่าข้อมูลจะถูกส่งลงมาตามหลัก

ปัจจุบันเทคโนโลยีในการสร้าง ROM จะเป็นแบบ MOS โดยโครงสร้างจะเป็นทั้งแบบ NMOS และ CMOS ซึ่งกินพลังงานไฟต่ำ สำหรับเทคโนโลยีแบบโพลาร์จะใช้ในกรณีต้องการความเร็วในการทำงานสูงๆ ROM ชนิด NMOS จะใช้ MOS ทรานซิสเตอร์แทนไดโอด จากรูปที่ 2.16 จะเป็น ROM อย่างง่ายที่แต่ละเวิร์ดมี 4 บิต ROM ตัวนี้อาจเรียกอีกอย่างหนึ่งว่า 16 บิต ROM



ROM contents

Word	Binary	Hexadecimal
	D_0 D_1 D_2 D_3	
00	0 1 0 0	4
10	0 0 1 0	2
01	1 1 1 0	E

รูปที่ 2.16 หน่วยความจำ ROM ขนาด 4 เวิร์ด แต่ละเวิร์ดเก็บได้ 4 บิต โดยสร้างจากทรานซิสเตอร์แบบ MOS ถ้าขาเกตตัวใดต่อกับแถวบิตนั้นจะเป็นลอจิก 0

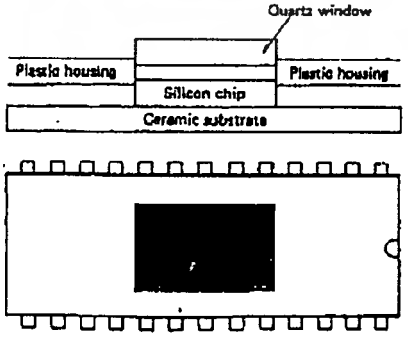
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าแต่ละหลักที่ส่งข้อมูลของ ROM จะใช้ทรานซิสเตอร์แบบ MOS สี่ตัว ต่อกันอยู่แบบพูลอัพทรานซิสเตอร์แบบ MOS บางตัว ขาเกตจะต่อกับสัญญาณเลือกแถว เมื่อแถวใดถูกเลือกแถวนั้นจะเป็นลอจิกสูง ถ้าขาเกตของทรานซิสเตอร์ต่ออยู่ทรานซิสเตอร์ตัวนั้นจะนำกระแสทำให้มีไฟไหลลงกราวด์จะเป็นลอจิก 0 ถ้าขาเกตไม่ต่ออยู่กับแถว หลักนี้จะเป็นลอจิก 1 และส่งข้อมูลออกทางเอาต์พุต

หน่วยความจำ ROM ถูกสร้างโดยขั้นตอนที่เรียกว่า mark-programming ลงบนซิฟเล็กๆ และเชื่อมต่อกันด้วยโลหะที่บางมากเป็นชั้นๆ ชั้นบนสุดบนไอซีจะเป็นชั้นที่ใช้ในวงจรโปรแกรมข้อมูลลงไป สารกึ่งตัวนำในชั้นนี้จะเชื่อมต่อขาเกตของทรานซิสเตอร์กับเส้นแถว ถ้าทรานซิสเตอร์ตัวใดไม่เชื่อมต่อกับเส้นแถวที่ถูกล็อก บิตของทรานซิสเตอร์ตัวนั้นจะมีลอจิกเป็น 1 ทางเอาต์พุต การโปรแกรมลงไปบน ROM จะทำได้โดยใช้แรงดันไฟฟ้าสูงๆผลิตภัณฑ์หลายๆอย่างนิยมใช้ ROM เช่นของเล่นที่ใช้ไมโครโพรเซสเซอร์ควบคุม วีดีโอเกม คอมพิวเตอร์ เป็นต้น

สำหรับหน่วยความจำประเภท EPROM โครงสร้างจะคล้ายกับ mask-programmed ROM คือใช้ทรานซิสเตอร์ประเภท MOS ต่อเป็นโครงสร้างคล้ายกับรูป 2.16 แต่ข้อแตกต่างระหว่าง EPROM กับ ROM มีหลายประการ ประการแรกคือ ทรานซิสเตอร์ทุกตัวใน EPROM จะต่อกับเส้นแถว ประการที่สองทรานซิสเตอร์จะออกแบบมาให้รับความต่างศักย์สูงๆที่ขาเกตได้ เมื่อความต่างศักย์ค่าสูงเข้ามาทรานซิสเตอร์จะได้รับประจุ และอยู่ในภาวะความต้านทานสูงซึ่งเป็นการเขียนข้อมูล ประการที่สามเราสามารถลบข้อมูลได้โดยใช้แสงอัลตราไวโอเลต (UV) ฉายลงบนทรานซิสเตอร์

EPROM สามารถโปรแกรมได้โดยใช้อุปกรณ์ที่สร้างพัลส์ที่มีความต่างศักย์สูงสร้างสัญญาณข้อมูล และสร้างแอดเดรสให้กับ EPROM ข้อมูลนี้จะคงอยู่จนกว่าเราจะป้อนแสงอัลตราไวโอเลตความเข้มสูง ลงบนหน้าต่างควอตซ์ของไอซี ข้อมูลเหล่านี้จะถูกลบหายไปโดยค่าแต่ละบิตใน EPROM จะเป็น 1 ในรูป 2.17 จะแสดงโครงสร้างของไอซี EPROM



รูปที่ 2.17 โครงสร้างภายในของ EPROM จะมีหน้าต่างควอตซ์สำหรับฉายแสงอัลตราไวโอเลตเพื่อลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

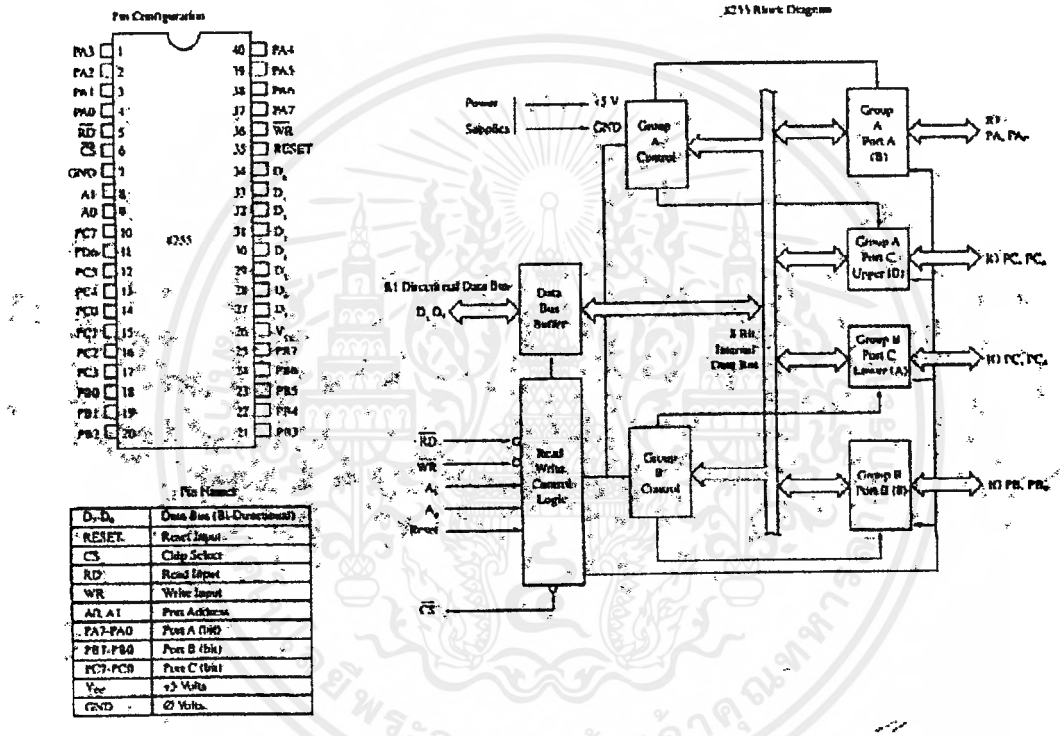
หน่วยความจำ ROM แบบที่ 3 จะนิยมใช้กันในบางโอกาสเรียกว่า EAROM (electrically alterable ROM) ข้อดีของ EAROM ก็คือข้อมูลภายในสามารถเปลี่ยนแปลงได้โดยการป้อนสัญญาณไฟฟ้าพิเศษเข้าไปในช่วงเวลาหนึ่งโดยปกติแล้วจะไม่กี่มิลลิวินาที ดังนั้นขบวนการเขียนข้อมูลจะช้ากว่ากระบวนการอ่านข้อมูล หน่วยความจำ EAROM นี้ข้อมูลภายในอาจเปลี่ยนแปลงได้แม้ว่าจะถูกติดตั้งในวงจรไปแล้ว หน่วยความจำประเภทนี้นิยมใช้ในบางระบบ ที่อาจมีการเปลี่ยนแปลงโปรแกรม

ถึงตรงนี้เราได้รู้จักหน่วยความจำ ROM สามชนิดที่แตกต่างกันแล้ว ROM (mask programmed RAM) เป็นหน่วยความจำที่มีความจุข้อมูลสูงและราคาถูก สำหรับ EPROM มีความจุข้อมูลต่ำกว่า ROM แต่เราสามารถโปรแกรมใหม่ได้ หน่วยความจำประเภทสุดท้ายคือ EAROM จะมีราคาแพงที่สุด ความจุข้อมูลต่ำ แต่เราสามารถลบข้อมูลได้โดยใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเชื่อมต่อ 8255 กับ MCS–51

ก่อนหน้านี้เราสามารถเลือกได้ว่าถ้าจะสร้างพอร์ตเอาต์พุตหรือพอร์ตอินพุตควรเลือกใช้ไอซีลักษณะใด ต่อมาจะแนะนำให้รู้จักกับไอซีที่เป็นพอร์ตขนานที่สามารถโปรแกรมให้ทำงานเป็นพอร์ทอินพุตหรือเอาต์พุตได้คือเบอร์ 8255 ซึ่งเป็นไอซี 40 ขา ซึ่งได้รับการออกแบบมาให้มีสัญญาณเชื่อมโยงกับไมโครโพรเซสเซอร์เบอร์ 8080 แต่ก็เหมาะที่ใช้กับ MCS–51 เช่นเดียวกัน ไอซีตัวนี้เป็นไอซีที่ต่อเป็นพอร์ตให้ไมโครโพรเซสเซอร์ได้ 3 พอร์ต



รูปที่ 2.18 แสดงโครงสร้างของไอซี 8255

การเรียกพอร์ตของ 8255 จะเรียกว่าพอร์ต A, B และ C โดยแต่ละพอร์ตจะมีขนาด 8 บิต และพอร์ต C จะแยกออกเป็นสองส่วนคือ PC0 – PC3 เรียกว่า พอร์ต C ล่าง จำนวน 4 บิต และพอร์ต C บน คือ PC4 – PC7 ที่พิเศษคือพอร์ตทุกพอร์ตเป็นได้ทั้งอินพุตและเอาต์พุต ลักษณะของ 8255 แสดงได้ดังรูป 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ขาต่าง ๆของ 8255

D0-D7 เป็นขาข้อมูลอินพุตและเอาต์พุตที่จะต้องผ่านเข้าออกจากส่วนนี้ D0-D7 จึงเป็นส่วนที่จะต่อเข้ากับระบบบัสของไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์สามารถอ่านหรือเขียนข้อมูลจากพอร์ตผ่านทางบัสนี้

CS ขานี้เป็นขาอินพุตที่จะรับสัญญาณจากภายนอกเพื่อเลือกชิพ 8255 โดยเมื่อขานี้เป็นลอจิก "0" จะทำให้ 8255 ต่อเข้ากับระบบบัสของไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนโทรลเลอร์เขียนหรืออ่านข้อมูลจากพอร์ตได้

RD ขาสัญญาณการอ่าน เป็นสัญญาณอินพุตที่ส่งมาจาก CPU เมื่อสัญญาณนี้เป็น "0" และ CS เป็น "0" ด้วย ตัว 8255 จะทำให้ตัว CPU อ่านข้อมูลจากบัสในขณะที่เป็นพอร์ตอินพุต

WR ขาสัญญาณการเขียน จะแอคทีฟเมื่อสัญญาณ WR เป็น "0" และ CS เป็น "0" สัญญาณนี้มาจาก CPU เมื่อต้องการเขียนข้อมูลลงบนพอร์ตที่กำหนด

A0-A1 ขาแอดเดรส ลอจิกของทั้งสองขานี้จะถอดรหัสออกมาได้ 4 ค่าเพื่อกำหนดค่ารีจีสเตอร์ภายในที่เชื่อมต่อเข้ากับพอร์ตอินพุตและเอาต์พุตของ 8255 เพื่อเลือกใช้พอร์ต A, B และ C

RESET ขารีเซต เป็นขาสัญญาณที่ส่งมาจากภายนอกเพื่อทำการรีเซต 8255 เพื่อเคลียร์สถานะต่างๆของ 8255 เมื่อ 8255 ได้รับการรีเซต มันจะกลับเข้าสู่โหมดอินพุตหรือทุกพอร์ตเป็นพอร์ตอินพุต

PA0-PA7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ชื่อพอร์ต A การเลือกพอร์ตจะเลือกโดยขา A0-A1

PB0-PB7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ชื่อพอร์ต B การเลือกพอร์ตจะเลือกโดยขา A0-A1

PC0-PC7 เป็นสายสัญญาณที่พอร์ตของ 8255 การกำหนดพอร์ตนี้จะได้รับการกำหนดโดยขาแอดเดรส A0-A1 พอร์ต C นี้แบ่งออกได้สองกลุ่มคือกลุ่ม PC0-PC3 และกลุ่ม PC4-PC7

ถ้าต้องการให้ 8255 ทำงานจะต้องทำให้ขา CS แอคทีฟ จากนั้นเลือกพอร์ตที่จะติดต่อโดย A0-A1 รีจีสเตอร์แต่ละตัวใน 8255 จะได้รับการกำหนดควบคุมคู่กับสัญญาณ RD และ WR เพื่อแสดงการทำงานต่างๆ ดังนั้นสัญญาณต่างๆของขาควบคุมที่นำมาประกอบกันจะมีความหมายดังตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RD	WR	A0	A1	ความหมาย
1	0	0	0	ส่งข้อมูลไปที่พอร์ต A
0	1	0	0	อ่านข้อมูลจากพอร์ต A
1	0	0	1	ส่งข้อมูลไปที่พอร์ต B
0	1	0	1	อ่านข้อมูลจากพอร์ต B
1	0	1	0	ส่งข้อมูลไปที่พอร์ต C
0	1	1	0	อ่านข้อมูลจากพอร์ต C
1	0	1	1	เขียนข้อมูลซึ่งเป็นรหัสควบคุม
0	1	1	1	ไม่ใช้

ตารางที่ 2 แสดงการเลือกพอร์ตของ 8255

การทำงานของไอซีเบอร์ 8255 จะทำงานได้ 3 โหมดคือ

1. Mode 0: Basic I/O
2. Mode 1: Strobed I/O
3. Mode 2: Bidirectional Bus

การให้ 8255 ทำงานแต่ละโหมดจะเลือกได้โดยการโปรแกรมให้กับ 8255 คำสั่งที่โปรแกรมจะมี 8 บิต แต่ละบิตมีความหมายดังนี้

บิต D7 เป็นบิตแสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น "1" จะหมายถึงรหัสควบคุมนี้มีผลต่อการเปลี่ยนการเซตโหมดต่างๆของ 8255

บิต D6 และ D5 เป็นโหมดของพอร์ต A ซึ่งมีการทำงาน 3 โหมด คือ โหมด 0,1 และ 2

บิต D4 ถ้ามีค่าเท่ากับ "0" หมายถึงการกำหนดพอร์ต A เป็นเอาต์พุต ถ้ามีค่าเป็น "1" จะกำหนดให้พอร์ต A เป็นพอร์ตอินพุต

บิต D3 เป็นบิตที่บอกการเซตพอร์ต C บน ถ้าเป็น "0" จะทำให้พอร์ต C บนเป็นเอาต์พุต

บิต D2 เป็นบิตที่บอกการเซตโหมดของพอร์ต B ถ้าเป็น "0" หมายถึงเลือกพอร์ต เป็นโหมด 0 และถ้าเป็น "1" คือการเลือกโหมด 1

บิต D1 เป็นการกำหนดอินพุตและเอาต์พุตของพอร์ต B ถ้าเป็น "0" คือพอร์ตเอาต์พุต และถ้าเป็น "1" คือพอร์ตอินพุตบิต

D0 เป็นบิตที่บอกการเซตพอร์ต C ล่าง ถ้าเป็น "0" จะทำให้พอร์ต C ล่างเป็นเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่	กลุ่ม	ความหมาย
D0	B	พอร์ต C ล่าง 1 = อินพุต 0 = เอาต์พุต
D1	B	พอร์ต B 1 = อินพุต 0 = เอาต์พุต
D2	B	เลือกโหมด 1 = โหมด 0 0 = โหมด 1
D3	A	พอร์ต C บน 1 = อินพุต 0 = เอาต์พุต
D4	A	พอร์ต A 1 = อินพุต 0 = เอาต์พุต
D5 D6	A	เลือกโหมด 00 = เลือกโหมด 0 01 = โหมด 1 1X = โหมด 2
D7		โหมดเซตแอคทีฟ 1 = แอคทีฟ

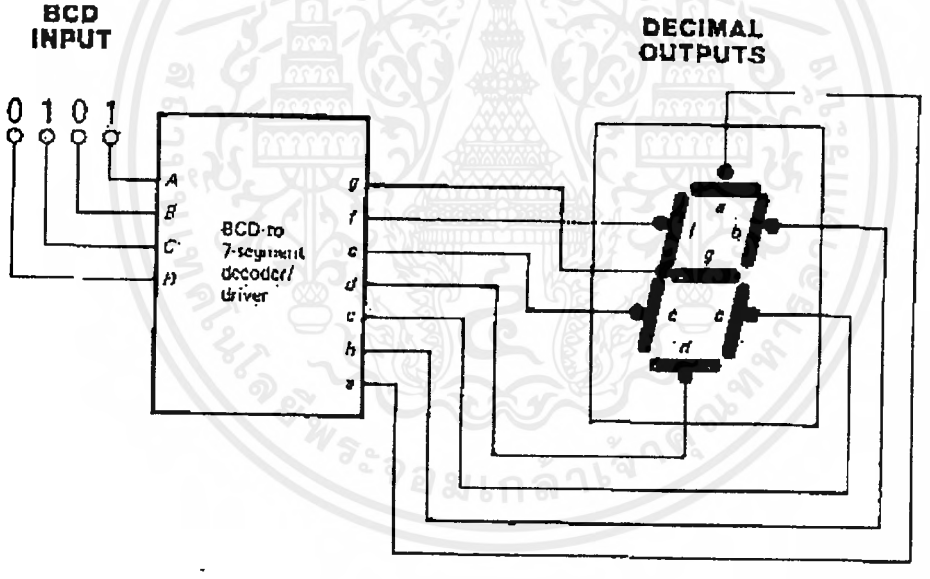
ตารางที่ 3 แสดงความหมายของบิตควบคุม

การทำงานในโหมด 0 จะเป็นการทำงานแบบอินพุตและเอาต์พุตพื้นฐาน ซึ่งจะประกอบด้วยพอร์ต 8 บิต 2 พอร์ตคือ พอร์ต A และพอร์ต B พอร์ตขนาด 4 บิต 2 พอร์ตคือ พอร์ต C บน และพอร์ต C ล่าง ทุกๆพอร์ตสามารถโปรแกรมให้เป็นอินพุตและเอาต์พุตได้ซึ่งมีทั้งหมด 16 รูปแบบด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ตัวถอดรหัส

ตัวถอดรหัส (Decoder) จะตรงข้ามกับตัวเข้ารหัส (encoder) จากรูปที่ 2.19 จะเห็นว่ามีตัวถอดรหัสสองตัวประกอบอยู่ในระบบ ซึ่งจะทำหน้าที่เปลี่ยนรหัส 8421 BCD ให้สามารถแสดงผลเป็นเลขฐานสิบที่สอดคล้องกันบนหลอดแสดงผลแบบ seven-segment ได้โดยตัวถอดรหัสจะทำหน้าที่แปลงรหัส 8421 BCD ออกมาและจะไปขับหลอด LED แต่ละเซกเมนต์ ในรูปที่ 2.19 จะแสดงตัวเลข B CD ค่า 0101 ซึ่งเป็นอินพุตให้กับ BCD-to- seven-segment decoder / driver โดยเอาต์พุตที่ได้ออกมาจะทำให้เซกเมนต์ a, c, d, f และ g สว่าง ดังรูปที่ 2.19 ซึ่งจะทำให้ seven-segment แสดงผลเป็นเลข 5



รูปที่ 2.19 การถอดรหัสให้ seven-segment แสดงผล

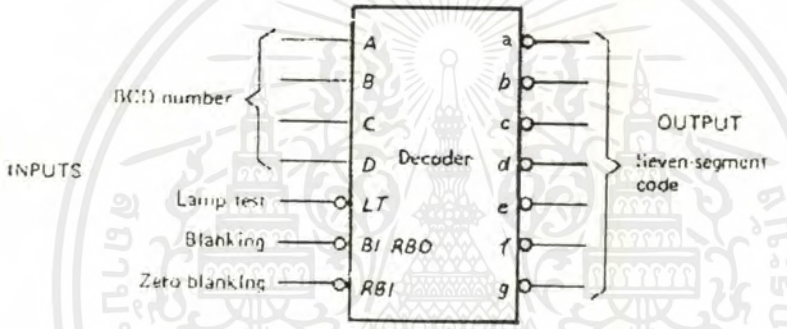
นอกจากการถอดรหัสออกมาเป็นฐานสิบแล้ว ตัวถอดรหัสแบบอื่นๆอาจมีอีกเช่นใช้แปลง BCD-to-binary, 4-to-16 line decoder และ 2-to-4-line decoder ภายในวงจรถอดรหัสจะประกอบด้วยวงจรลอจิกมากมาย ตัวถอดรหัสที่นิยมใช้มักอยู่ในรูปของไอซี ซึ่งภายในประกอบด้วยเกตตั้งแต่ 20 ถึง 50 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 BCD-TO-SEVEN-SEGMENT DECODER/DRIVER

ไอซีที่ทำหน้าที่เป็น BCD-TO-SEVEN-SEGMENT DECODER/DRIVER ได้แก่เบอร์ 7447 ซึ่งมีสัญลักษณ์แสดงได้ดังรูปที่ 2.20 ก) โดยอินพุตจะเป็นตัวเลข BCD ซึ่งเขียนเอาไว้เป็น A, B, C และ D สำหรับเอาต์พุตจะแอคทีฟ Low ถ้าให้อินพุตขา lamp-test (LT) เป็น Low เอาต์พุตทุกตัวจะเป็น Low หมด (a ถึง g) ถ้าให้ขาอินพุต Blanking (BI) Low High ripple-blanking เอาต์พุต (RBO) Low LED "blanking"

เอาต์พุตที่ออกมาจากไอซี 7447 สามารถเขียนเป็นตารางความจริงได้ดังรูปที่ 2.20 ข) จะสังเกตเห็นว่าอินพุตที่เข้ามาถ้ามีค่าสอดคล้องกับเลขฐานสิบคือ 10,11,12,13,14 และ 15 จะเป็นค่าที่ไม่ใช้ แต่ยังไงก็ตามค่าดังกล่าวจะไม่มีในรหัส BCD อยู่แล้ว เพราะรหัส BCD มีได้เพียงค่า 0 ถึง 9



(ก)

Decimal or function	INPUTS						BI RBO	OUTPUTS							Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	1
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

(ข)

รูปที่ 2.20 ก) สัญลักษณ์ทางลอจิกของไอซี Decoder TTL 7447A ข) ตารางความจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การเขียนโปรแกรมติดต่อกับพอร์ต

การเขียนโปรแกรมติดต่อกับพอร์ต ด้วย VISUAL BASIC นั้นจะจำเป็นต้องมีไฟล์ Inpout32.DLL ก่อน ซึ่งสามารถหาได้จาก Internet โดยนำไปเก็บไว้ที่ไดเรกตอรี่ windows/system เสร็จแล้วเมื่อต้องการนำมาใช้งานต้องมีการประกาศในโปรแกรมก่อนคือ

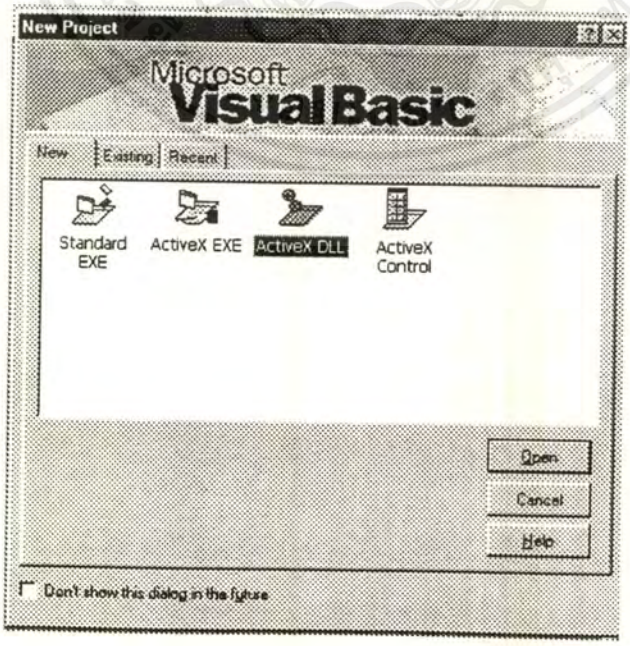
```
Private Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal Port Address As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal Port Address As Integer, ByVal Value As Integer)
```

เราสามารถติดต่อกับ Port โดยใช้คำสั่ง Inp และ Out แล้วตามด้วย Port ที่เราต้องการใช้ โดยใส่เป็นเลขฐาน 16 เช่น Inp &H3BD

เมื่อเราสามารถติดต่อกับพอร์ต ได้แล้วสิ่งต่อไปก็คือการเขียน ActiveX DLL ซึ่งมีไว้เพื่อให้ผู้เขียนโปรแกรมต่างๆ สามารถนำ Hardlock ไปใช้ได้โดยไม่ต้องมีความรู้เรื่องการติดต่อกับพอร์ต

2.11 ขั้นตอนการสร้าง ActiveX DLL

2.11.1. เริ่มสร้าง Project ใหม่เลือก ActiveX DLL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 กด F4 เพื่อเปิด Properties เปลี่ยนชื่อ Class1

Name = SimpleClass ← สำคัญเพราะจะต้องอ้างตอนใช้งาน

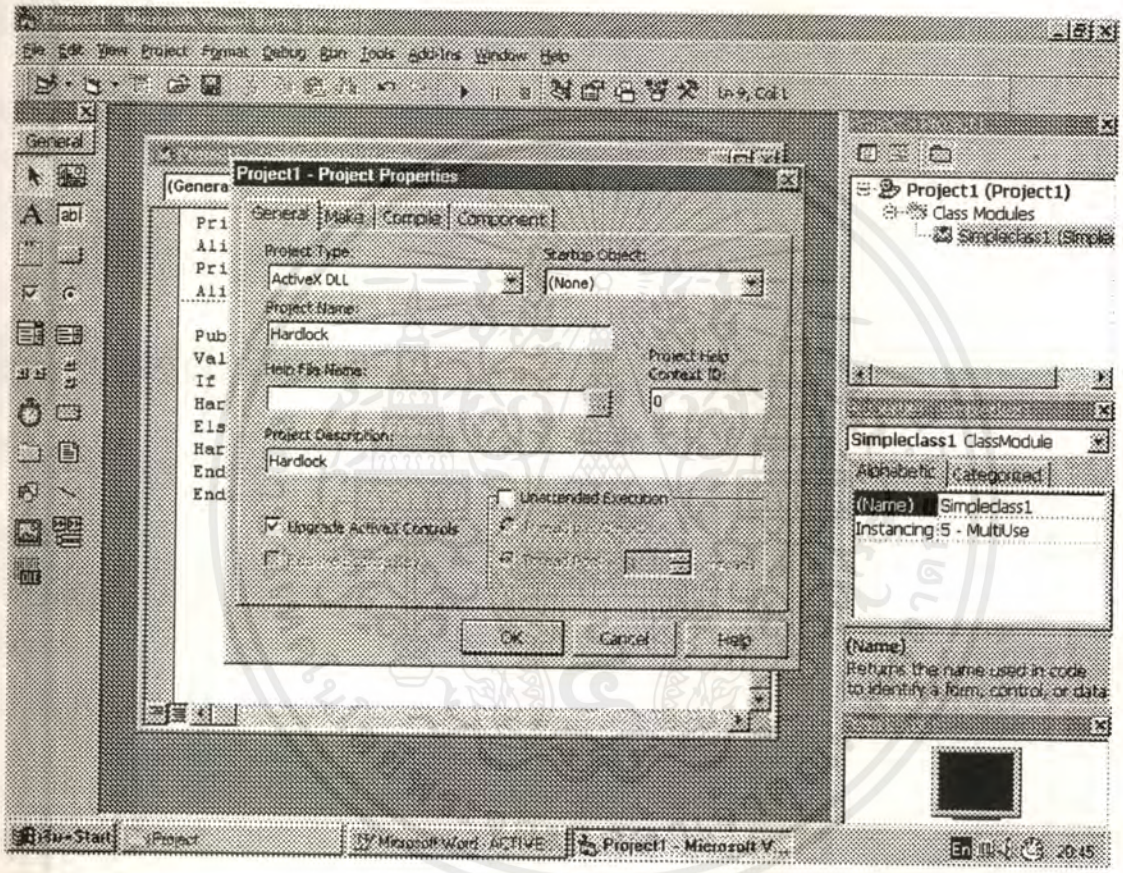
2.11.3 ไปที่เมนู Project เมาส์ Click Project1 Properties เลือกแถบคั่น General

Project Type = ActiveX DLL

Startup Object = None

Project Name = HardLock ← สำคัญเพราะจะต้องอ้างตอนใช้งาน

Project Description = Hard Lock

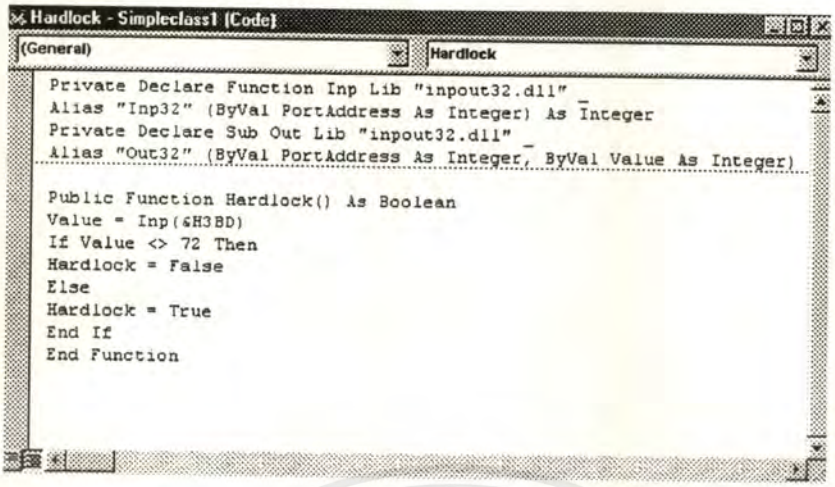


2.11.4 เขียน Code ใน SimpleClass

```
Public Function Hardlock() As Boolean
Value = Inp(&H3BD) ← ในที่นี้ใช้ PORT Printer เป็นตัว Test
If Value <> 72 Then ← ค่า 72 เป็นค่าที่ Set ไว้ใน Hard Lock
Hardlock = False
Else
Hardlock = True
End If
```

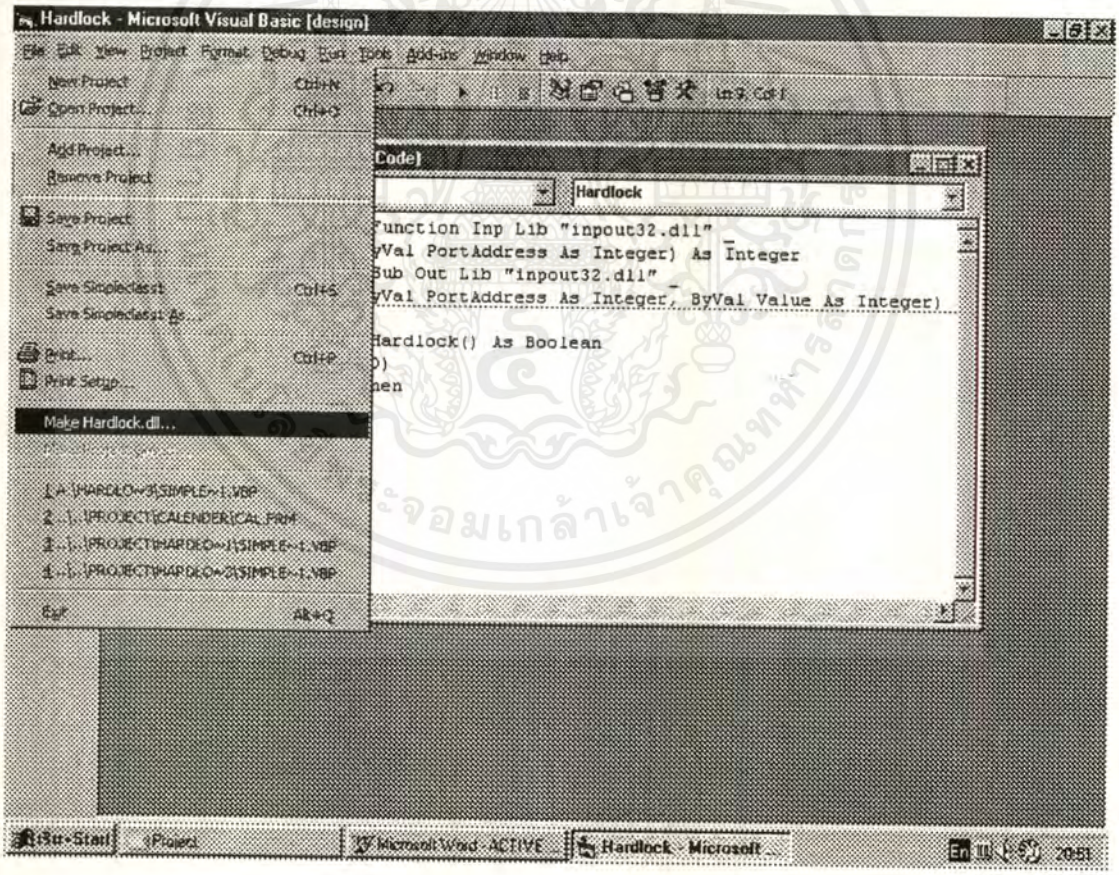
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Function



2.11.5 Save Project

2.11.6 สร้าง DLL ไปที่ Menu File → Make Hardlock.DLL

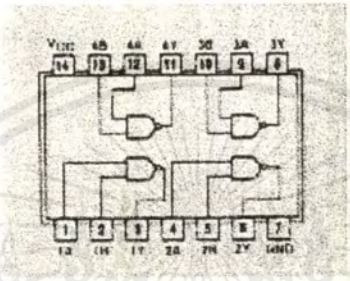


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนและการดำเนินงาน

3.1 เครื่องมือและอุปกรณ์ในการทำปัญหาพิเศษ

- เครื่องคอมพิวเตอร์ 80486 ขึ้นไป
- หัวต่อเครื่องพิมพ์ 25 ขา
- IC 74LS00 Quad 2-Input NAND Gate



รูปที่ 3.1.1 NAND Gate

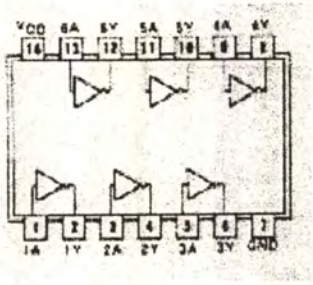
เป็น IC ที่ประกอบ NAND Gate ทั้งหมด 4 ตัว ที่จะรับอินพุตข้อมูล 2 ขา และเอาต์พุตข้อมูล 1 ขา ขา Vcc (ไฟเลี้ยง) ขาที่ 14 ขากราวด์ ขาที่ 7 ซึ่งผลลัพธ์ของการอินพุตและเอาต์พุตแสดงได้ดังตารางนี้

อินพุตที่1	อินพุตที่2	เอาต์พุต
0	0	1
0	1	1
1	0	1
1	1	0

ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ NAND Gate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IC 74LS04 Hex Inverter



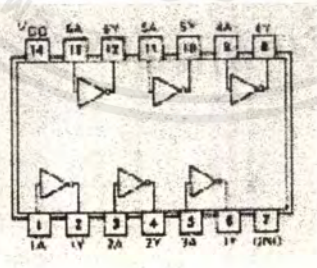
รูปที่ 3.1.2 Hex Inverter

เป็น IC ที่ประกอบ Hex Inverter ทั้งหมด 6 ตัว ที่จะรับอินพุตข้อมูล 1 ขา และเอาต์พุตข้อมูล 1 ขา ขา Vcc (ไฟเลี้ยง) ขาที่ 14 ขากราวด์ ขาที่ 7 ซึ่งผลลัพธ์ของการอินพุตและเอาต์พุตแสดงได้ดังตารางนี้

อินพุต	เอาต์พุต
0	1
1	0

ตารางแสดงผลลัพธ์อินพุตและเอาต์พุตของ Hex Inverter

- IC 74LS06 Hex Inverter Buffer/Driver

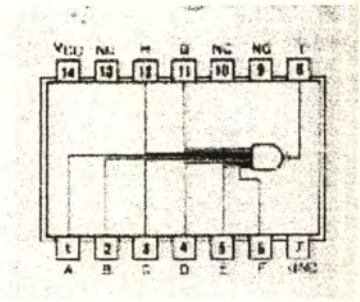


รูปที่ 3.1.3 Hex Inverter Buffer/Driver

คล้ายกับ IC 74LS04 แต่สามารถขับกระแสได้ประมาณ 0.02 mA และยังสามารถเป็น Buffer เก็บข้อมูลได้ชั่วขณะอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IC 74LS30 8-Input NAND Gate



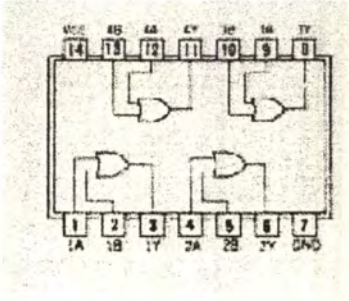
รูปที่ 3.1.4 8-Input NAND Gate

ประกอบด้วย NAND Gate ตัวที่มีอินพุตข้อมูล 8 ขาคือขาที่1,2,3,4,5,6,10,11,12และขาเอาต์พุต 1 ขา คือขาที่ 8 ส่วนขา Vcc (ไฟเลี้ยง) ขาที่ 14 และกราวด์ ขาที่ 7 ซึ่งผลลัพธ์ของการอินพุตและเอาต์พุตแสดงได้ดังตารางนี้

อินพุตที่1	อินพุตที่2	เอาต์พุต
0	0	1
0	1	1
1	0	1
1	1	0

ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ 8-input NAND Gate

- IC 74LS32 Quad 2-Input OR Gate



รูปที่ 3.1.5 Quad 2-input OR Gate

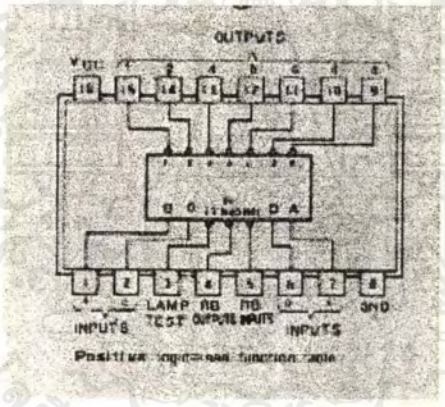
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น IC ที่ประกอบ OR Gate ทั้งหมด 4 ตัว ที่จะรับอินพุตข้อมูล 2 ขา และเอาต์พุตข้อมูล 1 ขา ขา Vcc (ไฟเลี้ยง) ขาที่ 14 ขากราวด์ ขาที่ 7 ซึ่งผลลัพธ์ของการอินพุตและเอาต์พุตแสดงได้ดังตารางนี้

อินพุตที่1	อินพุตที่2	เอาต์พุต
0	0	1
0	1	1
1	0	1
1	1	0

ตารางแสดงผลลัพธ์ของการอินพุตและเอาต์พุตของ Quad 2-input OR Gate

- IC 74LS47 BCD-to-7 Segment Decoder/Driver

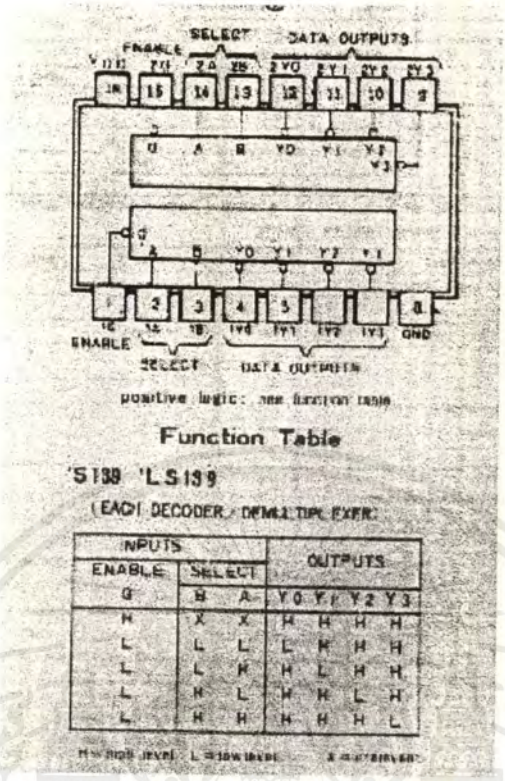


รูปที่ 3.1.6 BCD-to-7 Segment Decoder/Driver

เป็น IC ที่ทำหน้าที่ Decode ข้อมูลที่เป็นเลขฐานสองให้เป็นข้อมูลที่ถูกแสดงในเลขฐานสิบ ซึ่งผลที่ได้จะนำไปแสดงที่ 7 Segment ซึ่งจะอธิบายในบทถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IC 74LS139 Dual 2 to 4 Line Decoder



รูปที่ 3.1.7 Dual 2 to 4 Line Decoder

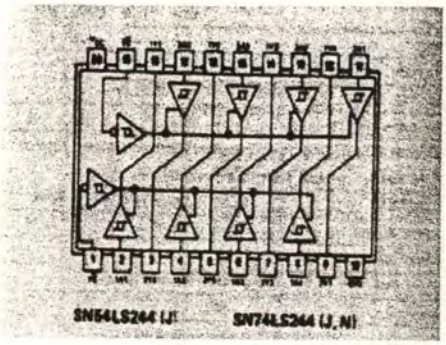
เป็น IC ที่มีขาที่อินพุตข้อมูล 2 ขาและขาที่เป็นเอาต์พุตข้อมูล 4 ขาโดยข้อมูลที่อินพุตมานั้นจะสามารถไปกำหนดเอาต์พุตทั้ง 4 ว่าเอาต์พุตขาไหนจะเป็นตัวทำงานโดยจะส่งสัญญาณ Enable ออกไป สัญญาณที่ส่งออกไปจะเป็นบิท 0 คืออุปกรณ์ตัวที่จะมารับข้อมูลจะต้องทำงานที่บิต 0 ด้วย (Active Low) ซึ่งผลลัพธ์ของการอินพุตและเอาต์พุตแสดงได้ดังตารางนี้

อินพุตที่ 1	อินพุตที่ 2	เอาต์พุต
0	0	Y0
1	1	Y1
1	0	Y2
1	1	Y3

ตารางแสดงผลลัพธ์อินพุตเอาต์พุตของ Dual 2 to 4 Line Decoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IC 74LS244 Octal 3 State Bus Driver

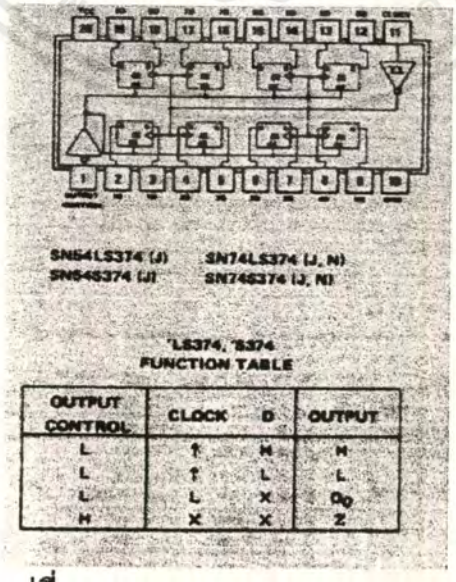


รูปที่ 3.1.8 Octal State Bus Driver

เป็น IC ที่ทำหน้าที่เป็น Buffer ที่สามารถขับกระแสได้โดยที่จะมีขาอินพุตข้อมูล 8 ขาและขาเอาต์พุตข้อมูล 8 ขา ขาที่ 1 และ 19 จะเป็นขาที่รับสัญญาณ Enable จากอุปกรณ์อื่นถ้าขณะนั้นต้องการที่จะส่งข้อมูลออก สัญญาณที่ว่านี้จะถูกส่งมาเป็นบิท 0 (Active Low)

- IC 74LS374 Octal D Type F-F 3 State

เป็น IC ที่ทำหน้าที่เป็น Buffer ในการ Latch ข้อมูลจากอุปกรณ์ 2 ชนิดที่มีความเร็วในการรับส่งข้อมูลไม่เท่ากัน โดยที่จะมีขาอินพุตข้อมูล 8 ขาและขาเอาต์พุตข้อมูล 8 ขา และมีขาที่ 11 เป็นขารับสัญญาณ Enable ที่ส่งมาจากอุปกรณ์อื่น เพื่อข้อมูลจะถูก Latch ออกไป สัญญาณที่ส่งมาจะเป็นสัญญาณนาฬิกา (Clock) ซึ่งจะทำงานที่ขอบขาขึ้นดูรูปที่ 3.1.9 ประกอบ



รูปที่ 3.1.9 Octal D Type F-F 3 State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IC 74LS688 8-Bit Equal to Comparator

  เป็น IC ที่ทำหน้าที่เปรียบเทียบข้อมูลที่เป็นอินพุต 2 ชุดชุดละ 8 บิท คือถ้าข้อมูลที่อินพุตเข้ามาทั้ง 2 ชุด มีค่าเดียวกันก็จะส่งสัญญาณ Enable ที่เป็นบิท 0 ออกไป(Active Low)

- IC UM82C55A

  ได้อธิบายในบทที่ 2 แล้ว

- หลอด LED
- ตัวต้านทาน 300 Ω , 10K , 1K
- DIP SWITCH

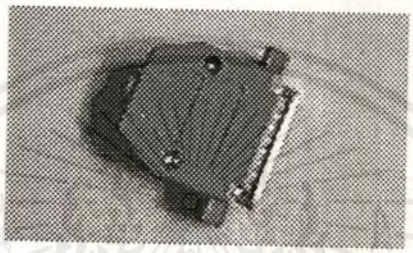
3.2 โปรแกรมที่ใช้ในการปัญหาพิเศษ

- Turbo C++
- Visual Basic 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนและวิธีการดำเนินงาน

3.3.1 ศึกษาการรับข้อมูลจากพอร์ตพรินท์เตอร์โดยนำมาประยุกต์ทำเป็น Hardlock ทางพรินท์เตอร์พอร์ต โดยจะนำเอา Dip Switch มาต่อกับขาที่ 10, 11, 12, 13, 15 ของพรินท์เตอร์พอร์ตแล้วนำต่อกราวด์แล้วนำการตั้งค่า Dip Switch ในการทดลองตั้งค่าไว้มีค่า 72 เพื่อว่าในส่วนของโปรแกรมเราจะเช็คค่าที่อินพุตเข้ามานั้นเป็นค่า 72 หรือไม่ ถ้าอ่านค่าเข้ามาแล้วไม่ใช่โปรแกรมที่ป้องกันด้วย Hardlock จะไม่ทำงาน



รูปที่ 3.3.1 Hardlock ที่ผ่านทางพรินท์เตอร์พอร์ต

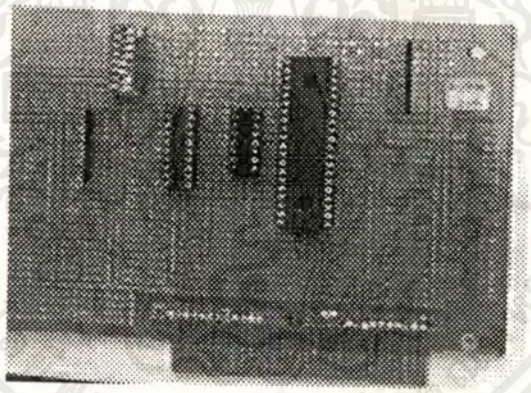
3.3.2 เปลี่ยนจากการอินพุตทางพอร์ตพรินท์เตอร์มาใช้พอร์ตขนานคือเราจะออกแบบวงจรและทำการทดลองสร้างบนการ์ด (PCB) แล้วเสียบลงใน Expansion Slot ในคอมพิวเตอร์เช่นเดียวกับการติดตั้ง Sound Card หรือ VGA Card ทั่วๆไปโดยเราจะออกแบบอินพุตและเอาต์พุต จากทฤษฎีที่เราศึกษามาโดยเราจะเลือกใช้แอดเดรสพอร์ตเบอร์ 2BF เมื่อเราออกแบบวงจร Decode ให้ทำงานที่แอดเดรสนี้แล้วจะต้องส่งสัญญาณให้ไป Enable IC ที่เป็น Buffer โดย Buffer นี้จะรับข้อมูลที่อินพุตเข้ามาโดย Dip Switch และส่งข้อมูลไปตามData Busในส่วนเอาต์พุตเราจะเลือกแอดเดรสเบอร์ 2BE แล้วก็ทำการออกแบบวงจร Decode เพื่อให้ไป Enable IC ที่เป็น Buffer เพื่อที่จะส่งข้อมูลออกไปแสดงผลที่หลอด LED โดย Buffer ตัวนี้จะรับข้อมูลจาก Data Bus

3.3.3 นำเอา Card ในส่วนของอินพุตมาประยุกต์ใช้รวมกับ IC เบอร์ 7447 เพื่อที่จะเป็นตัว Encode ข้อมูลให้ได้ข้อมูลชุดหนึ่งขึ้นมาโดยที่ผู้ใช้ทั่วไปจะไม่สามารถรู้ข้อมูลที่ถูก Decode ได้ถ้าเราป้องกันไม่ให้รู้เบอร์ IC ที่ Decode. ในการประยุกต์ใช้กับ IC 7447 นี้ทำให้เราเข้าใจการทำงานในส่วนอินพุตมากขึ้นและเราสามารถเอาข้อมูลที่ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Decode นี้มาเป็นตัวกำหนดแอดเดรสที่จะอ้างตำแหน่งข้อมูลบน ROM เพื่อนำเอาข้อมูลไปใช้เพื่อสร้างความซับซ้อนให้กับการทำงานของ Hardlock ได้

3.3.4 จะใช้ IC เบอร์ 8255 เข้ามาใช้เป็นพอร์ตอินพุตและเอาต์พุตเนื่องจากใน 2 หัวข้อที่ผ่านมาเป็นการกำหนดแอดเดรสพอร์ตคงที่คือ 2BF หรือ 2BE ในกรณีที่แอดเดรสพอร์ตนั้นไม่ว่างเราก็ไม่สามารถใช้ Card นั้นได้แต่ถ้าเราใช้ IC เบอร์ 8255 นี้ร่วมด้วยจะสามารถแก้ปัญหาที่กล่าวมาได้เพราะว่า IC ชนิดนี้มีพอร์ตถึง 3 พอร์ตให้ใช้งานและยังเป็นได้ทั้งอินพุตและเอาต์พุตพอร์ต เมื่อเราออกแบบวงจรเราก็สามารถเปลี่ยนแอดเดรสพอร์ตได้เมื่อมีการใช้แอดเดรสพอร์ตซ้ำกัน โดยเราจะส่งข้อมูลออกทางพอร์ต A ของ IC เบอร์ 8255 แล้วส่งเป็นอินพุตให้กับ IC เบอร์ 7447 เพื่อ Decode ให้ได้ข้อมูลชุดหนึ่งแล้วส่งข้อมูลที่ได้กลับไปเป็นอินพุตให้กลับ IC เบอร์ 8255 ทางพอร์ต B แล้วนำข้อมูลนี้ไปใช้เป็นค่ากำหนด Password ในการทดลองเราได้ส่งข้อมูลไปทางพอร์ต A เป็นค่า 9 จะได้ข้อมูลที่จะไปกำหนดการทำงานของโปรแกรมที่ควบคุมด้วย Hardlock ประเภทนี้

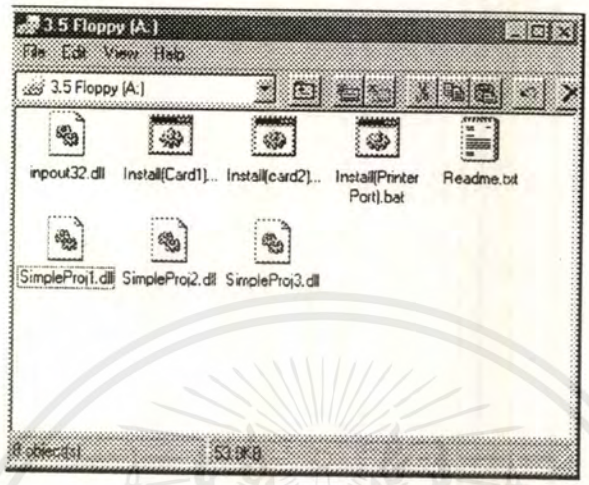


รูปที่ 3.3.2 Hardlock ที่ผ่านทางพอร์ตขนาน

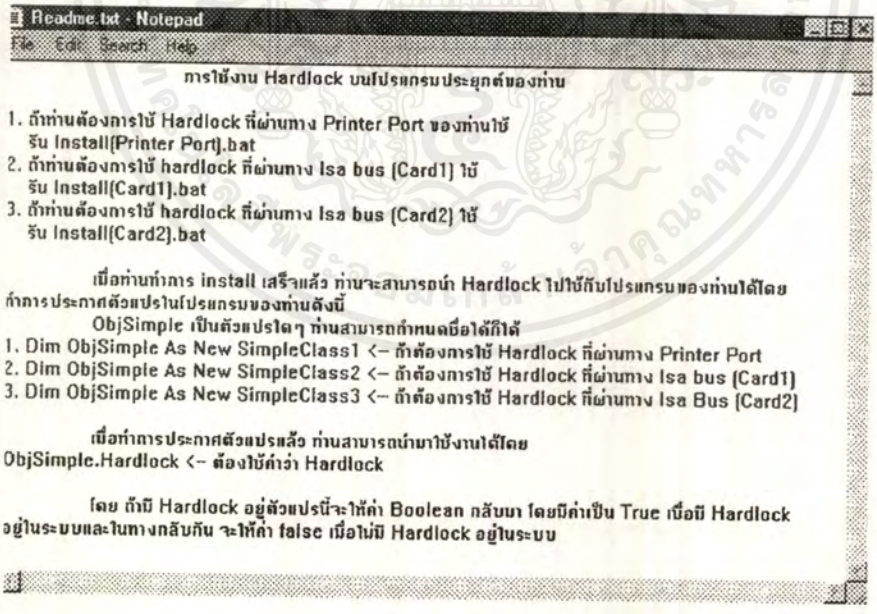
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การใช้งาน Hardlock ในโปรแกรม Visual Basic

3.4.1 Install ส่วนที่สำคัญจากแผ่น Driver เมื่อต้องการใช้งาน Hardlock โดยในแผ่นจะประกอบไปด้วย file ดังนี้

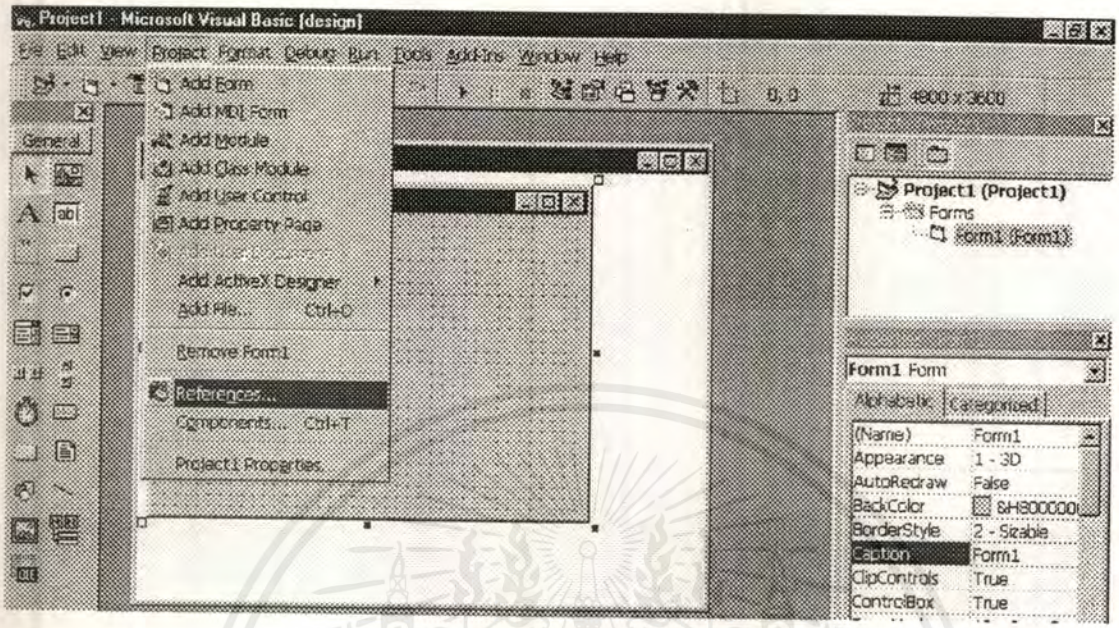


3.4.2 วิธีการใช้งาน Install สามารถอ่านได้จาก file Readme.txt ดังนี้

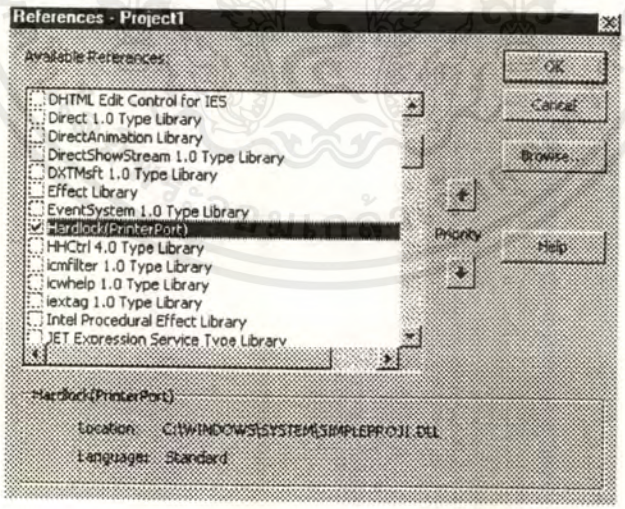


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 เมื่อทำการ Install เสร็จแล้ว ใน Visual Basic จะสามารถใช้งาน Hardlock ได้ โดย ไปที่ Preferences ของ Project นั้นๆ ดังรูป

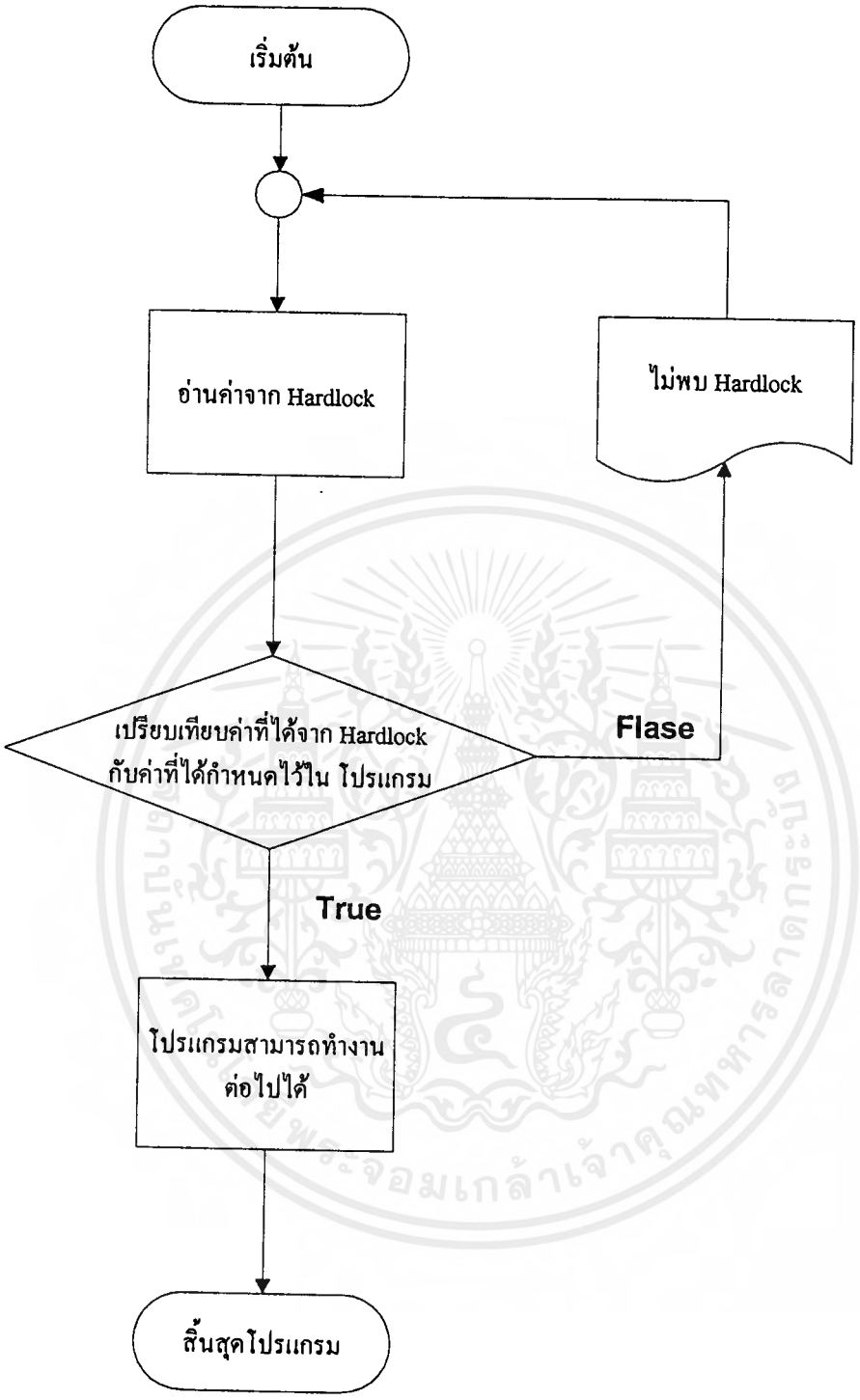


3.4.4 เมื่อไปที่ Reference แล้ว ทำเครื่องหมายที่หน้าข้อความ Hardlock (Printer Port) แล้วแต่ที่ได้ทำการ Install ว่าใช้ Hardlock แบบใด



3.4.5 เมื่อทำการเลือก Reference Hardlock แล้ว ก็สามารถใช้งาน Hardlock ได้ดังที่กล่าวไว้ใน file Readme.txt ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOWCHAT การทำงานของ HARDLOCK ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปผลปัญหาพิเศษและข้อเสนอแนะ

4.1 การติดต่อกับ hard Lock

4.1.1 ติดต่อแบบค่า Address คงที่

- เราสามารถติดต่อกับ Printer Port ผ่าน Address ของคอมพิวเตอร์แต่ละเครื่อง ซึ่งAddress ที่ใช้ในการติดต่อจะมีค่าตายตัว ซึ่งถูกกำหนดเป็นมาตรฐาน คือ ในการนำข้อมูลออกใช้378H, 3BCH ในการนำข้อมูลเข้าใช้ 389H, 3BDH

- ติดต่อแบบพอร์ตรับส่งข้อมูลแบบขนาน (Isa Port) ซึ่งเรากำหนดให้ใช้ Address ในการนำข้อมูลออกใช้ 2BEH ส่วนการนำข้อมูลเข้าใช้ 2BFH

4.1.2 การติดต่อกับ Hard Lock แบบกำหนด Address ได้หลายค่า

- เป็นการติดต่อแบบพอร์ตรับส่งข้อมูลแบบขนาน แต่นำ Dip Switch มาช่วยในการเลือก Address เองได้

4.2 การกำหนดรหัสใน Hard Lock

4.2.1 ในการใช้กับ Printer Port เราออกแบบวงจรให้สร้างรหัสอย่างง่ายขึ้น โดยค่าที่ได้จะมีค่าๆ เดียว

4.2.2 ในการติดต่อกับพอร์ตแบบขนาน เราสามารถเพิ่มความซับซ้อนให้มากขึ้น โดยผ่านเข้ารหัส ซึ่งทำให้ค่าที่ได้แตกต่างกันออกไป

4.3 การนำ Hard Lock ไปใช้งาน

ในที่นี้ทางผู้จัดทำได้สร้างโปรแกรมที่ติดต่อกับตัว Hard Lock ไว้ โดยทางผู้ใช้สามารถนำไปประยุกต์ใช้กับ โปรแกรม Visual Basic หรือ C++ ได้อย่างสะดวก โดยที่ไม่ต้องมีความรู้เกี่ยวกับการติดต่อกับ Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลของปัญหาพิเศษ

เมื่อนำ Hardlock ไปประยุกต์กับโปรแกรมต่างๆ ตามที่ผู้ใช้งานต้องการ ตัวโปรแกรมสามารถที่จะป้องกันการถูกลักลอบนำไปใช้งาน โดยที่ยังไม่ได้รับอนุญาตไม่ได้ ถ้าขาดตัว Hardlock

4.5 ข้อเสนอแนะ

4.5.1 ในส่วนของ Hard Lock ที่ติดต่อทาง Printer Port นั้น สามารถถูกถอดรหัสได้โดยง่าย เพราะเป็นการกำหนดค่าอย่างง่าย วงจรไม่มีความซับซ้อน

4.5.2 ในส่วนของ Hard Lock ที่ติดต่อทางพอร์ตขนาน เราสามารเพิ่มความซับซ้อนเข้าไปอีกโดยใช้ ROM ที่สามารถโปรแกรมเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. อ.ธีรวัฒน์ ประกอบผล,หนังสือระบบงานคอมพิวเตอร์ (COMPUTER ORGANIZATION), โครงการตำราเรียน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541
2. อ.ธีรวัฒน์ ประกอบผล,หนังสือเทคนิคการเชื่อมต่อ IBM PC ,Mc Graw Hill , 2541
3. อ.ธีรวัฒน์ ประกอบผล,สถาปัตยกรรมคอมพิวเตอร์,ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2541
4. อ.ธีรวัฒน์ ประกอบผล, ดิจิตอลอิเล็กทรอนิกส์,โครงการตำราเรียนสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น),2541
5. Michael Halvorson, Step by Step Microsoft Visual Basic5 , บริษัท เอช. เอ็น. กรุ๊ป จำกัด, 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้