สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องทดสอบไอซี ทีทีแอล

TTL TESTER

โดย

นายบัณฑิต อินทร์จักร

นายอดิศักดิ์ สรสมุทร

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50330
วัน,เดือน,ปี 29 พ.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องทดสอบไอซี ทีทีแอล

(TTL Tester)

โดย

นาย บัณฑิต อินทร์จักร์ รหัส 43515918

นาย อดิศักดิ์ สรสมุทร รหัส 43515943

อาจารย์ที่ปรึกษา

รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2545

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์บัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องทดสอบไอซี ทีทีแอล

ผู้จัดทำ

1. นาย บัณฑิต อินทร์จักร์
2. นาย อดิศักดิ์ สรสมุทร

.....อาจารย์ที่ปรึกษา

(รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเรื่อง เครื่องตรวจสอบไอซี ทีทีแอล

TTL Tester

จัดทำโดย นาย บัณฑิต อินทร์จักร์ 43515918

นาย อดิศักดิ์ สรสมุทร 43515943

โครงงานนี้ ได้ผ่านการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

ลงชื่อ ..... อาจารย์ที่ปรึกษา

(รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องตรวจสอบไอซี ทีทีแอล

บัณฑิต อินทร์จักร์
อดิศักดิ์ สรสมุทร
รศ.ดร. สุรพันธุ์ เอี๋ยไพบูลย์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

เครื่องทดสอบไอซีทีทีแอล ในโปรเจ็คท์นี้ได้รวมเอาฟังก์ชันการทดสอบไอซี ทีทีแอล ที่มีจำนวนขาไม่เกิน 24 ขา อาทิเช่น ไอซีเกทธรรมดา , ไอซีเข้ารหัส ถอดรหัส , ไอซีมัลติเพล็กซ์ ดีมัลติเพล็กซ์ , ไอซีคำนวณทางคณิตศาสตร์ , ไอซีประเภท Open Collector รวมถึงไอซี ทีทีแอล ประเภทที่ใช้สัญญาณนาฬิกา ได้แก่ ฟลิป-ฟลอบ, ไอซีวงจรนับ เป็นต้น การทำงานของเครื่องตรวจสอบ จะทำงานร่วมกันระหว่าง ไมโครคอนโทรลเลอร์ กับ ไมโครคอมพิวเตอร์ ซึ่งจะทำการสื่อสารผ่านพอร์ตอนุกรม โดย ไมโครคอมพิวเตอร์จะเป็นตัวหลักในการสั่งงานให้ ไมโครคอนโทรลเลอร์ทำการติดต่อตรวจสอบไอซี และส่งข้อมูลกลับมาให้ไมโครคอมพิวเตอร์ ทำการประมวลผล และแสดงผลในรูปแบบของกราฟฟิก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTL Tester

Bandit Injak
Adisak Sornsamut
Dr.Surapan Airphaiboon (Advisor)
Academic Year 2001

Abstract

TTL Tester in this project can test ICs under 24 pin TTL, namely Basic gate, Decoder, Encoder, multiplex, Demultiplex, TTLs with Open Collector and TTLs that operate with clock such as Flip-Flop, Counter etc.

The operation of TTL Tester is a cooperation between microcontroller and microcomputer along with serial port communication. The microcomputer order the Test Code to microcontroller to operate in testing, after that microcontroller sends the results of testing back to microcomputer. The processing and display of results are the function of microcomputer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

เครื่องตรวจสอบไอซี ทีทีแอล ในโครงงานฉบับนี้ สำเร็จลุล่วงไปได้ซึ่งต้องขอขอบคุณ รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์ เป็นที่ปรึกษาคอยให้คำแนะนำ แนวทางแก้ปัญหาต่างๆที่พบในโครงงานตลอดจนเครื่องมืออุปกรณ์ต่างๆที่ใช้ในโครงงานขอบคุณอาจารย์ เทอดศักดิ์ ลิ่วหาทอง ที่อนุเคราะห์เครื่องโปรแกรมไมโครคอนโทรลเลอร์ และท้ายสุด ขอขอบพระคุณท่านคณาจารย์ทุกท่านที่ได้ถ่ายทอดความรู้ทางวิชาการต่างๆ ที่เป็นประโยชน์ต่อโครงงานนี้ ซึ่งทางผู้จัดทำโครงงานก็ขอขอบพระคุณทุกท่านอีกครั้งไว้ ณ ที่นี้ด้วย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บัณฑิต อินทร์จักร์

.....

(นาย บัณฑิต อินทร์จักร์)

อดิศักดิ์ สรสมุทร

.....

(นาย อดิศักดิ์ สรสมุทร)

วันที่ 8 / ม.ค. / 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III

สารบัญ

สารบัญตาราง

สารบัญรูป

บทที่ 1. บทนำ (Introduction) 1

บทที่ 2. IC TTL 2

2.1 ตระกูลของไอซีทีทีแอล 2

2.2 คุณสมบัติ และลักษณะของไอซีทีทีแอล 5

2.3 การทำงานของไอซีทีทีแอลประเภทต่างๆ 6

2.4 ค่าพารามิเตอร์ที่สำคัญของไอซีทีทีแอล 12

บทที่ 3. Serial Port 16

3.1 การติดต่อแบบอินเตอร์รัพท์ 16

3.2 การติดต่อแบบโพลลิ่ง 16

3.3 การใช้ Visual Basic ติดต่อผ่าน Serial Port 16

3.4 การใช้ไมโครคอนโทรลเลอร์ติดต่อกับ Serial Port 19

3.5 การเขียนโปรแกรมติดต่อกับ Serial Port 21

บทที่ 4. ทฤษฎี และการใช้งาน 8255 เบื้องต้น 23

4.1 ลักษณะเบื้องต้น 23

4.2 หน้าที่ของขาต่างๆ 25

4.3 การใช้งาน 8255 26

4.4 การต่อ 8255 เข้ากับ CPU 28

4.5 โหมดการทำงานของ 8255 30

บทที่ 5. โครงสร้างทางฮาร์ดแวร์ และ ซอฟต์แวร์ 35

5.1 โครงสร้างทาง ฮาร์ดแวร์ 35

5.1.1 Programmable Supply 35

5.1.2 ภาคป้อนและควบคุมข้อมูล 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
5.2 โครงสร้างทางซอฟต์แวร์	40
5.2.1 การติดต่อสื่อสารระหว่างไมโครคอนคอมพิวเตอร์ กับไมโครคอนโทรลเลอร์	40
5.2.2 การแสดงผล	44
5.2.3 FLOW CHART แสดงการทำงานในส่วนของไมโครคอมพิวเตอร์	45
5.2.4 FLOW CHART แสดงการทำงานในส่วนของไมโครคอนโทรลเลอร์	46
บทที่ 6 ผลการทดลอง	47
6.1 การติดต่อกับผู้ใช้งานของโปรแกรม ในไมโครคอมพิวเตอร์	47
6.2 การทดสอบไอซี BASIC GATE	48
6.3 การทดสอบไอซี ประเภทเข้ารหัส ถอดรหัส	52
6.4 การทดสอบไอซีประเภทที่ใช้สัญญาณนาฬิกา	53
6.5 การทดสอบไอซีประเภท Open Collector	54
บทที่ 7 สรุป	56
7.1 หลักการของเครื่องทดสอบไอซีทีทีแอล	56
7.2 ผลการทำงานของเครื่องตรวจสอบไอซี ทีทีแอล	57
ภาคผนวก	58
เอกสารอ้างอิง	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงRegister SCON (Serial Port Control Register)	20
3.2 แสดงรูปแบบการแสดงการเลือกโหมดการทำงานของพอร์ตอนุกรม	20
3.3 แสดงรูปการเขียนโปรแกรมตั้งค่าทำงานของ Timer	22
4.1 แสดงผลของสัญญาณควบคุมต่างๆที่มีผลต่อ 8255	29
4.2 แสดงขาสัญญาณในโหมด 1 แบบ Handshaking	31
4.3 แสดงสัญญาณ Handshaking ในโหมด 2	34
5.1 ตารางความจริงของ 74LS138	36
5.2 แสดงตารางความจริงของไอซีเบอร์ 7400 ที่ใช้ในโปรแกรม	42
5.3 แสดงตัวอย่างตารางที่ใช้เก็บข้อมูลของ Control Word	43
5.4 แสดงตัวอย่างตารางที่ใช้เก็บข้อมูลของขา Clock ของไอซีที่ใช้ Clock ในการทำงาน	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 วงจรไอซีทีทีแอล ตระกูลมาตรฐาน SN 54/74	2
2.2 วงจรไอซีทีทีแอล ตระกูลที่กินไฟน้อย SN 54L / 74L	3
2.3 วงจรไอซีทีทีแอล ตระกูลที่มีความเร็วสูง SN 54H / 74H	3
2.4 วงจรไอซีทีทีแอล ตระกูลช็อตกี้ SN 54S / 74S	4
2.5 Schottky Barrier Transistor	4
2.6 แสดงแพ็กเกจแบบต่าง ๆ ของทีทีแอล	5
2.7 แสดงลักษณะวงจรและสัญญาณทางลอจิกของไอซีทีทีแอล NAND GATE	7
2.8 แสดงลักษณะสัญญาณอินพุท – เอาท์พุทของวงจรไอซีทีทีแอล NAND GATE	8
2.9 วงจรลอจิก NOR เกท	8
2.10 โครงสร้างของ Open Collector ทีทีแอลเกท	10
2.11 เปรียบเทียบวงจรลอจิก ทีทีแอลเกทแบบ Open Collector กับ Totem Pole	10
2.12 แสดงลักษณะสัญญาณอินพุทเมื่อผ่านวงจร Schmitt Trigger NAND เกท	11
2.13 ลักษณะอินพุทและเอาท์พุท ของ Schmitt Trigger NAND เกท	11
4.1 แสดงตำแหน่งขาต่างๆของ 8255	24
4.2 แผนผังภายในของ 8255	24
4.3 แสดงความหมายแต่ละบิต ของรหัสควบคุม	27
4.4 แสดงการต่อ 8255 เข้ากับ CPU	28
4.5 การจัดสัญญาณในแบบ Handshaking	31
4.6 การของอินเตอร์รัพท์ กรณีเป็นอินพุทพอร์ต	32
4.7 การของอินเตอร์รัพท์ กรณีเป็นเอ้าท์พุทพอร์ต	32
5.1 Programmable Supply	35
5.2 บล็อกไดอะแกรมแสดงการทำงานของส่วนควบคุมข้อมูลทดสอบไอซี	37
5.3 แสดงตัวอย่างส่วนควบคุม (Latch Control Pin In/Out) , ส่วนส่งถ่ายข้อมูล (Latch Data) และส่วน ส่งผลลัพธ์กลับสู่ระบบ Microcontroller (Input Relult)	38
5.4 แสดงส่วนควบคุม R PULL - UP (R PULL - UP)	39
5.5 รูปแบบการต่อวงจรจาก 8255 เข้าสู่ขาของไอซีที่ทำการทดสอบ	41
5.6 วงจรเกทภายในไอซีเบอร์ 7400	41
5.7 แสดงรูปแบบของการแสดงผล มีเกท 1 เกทที่เสีย	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
5.8 แสดง FLOW CHART การทำงานในส่วนของ ไมโครคอมพิวเตอร์	45
5.9 แสดง FLOW CHART การทำงานในส่วนของ ไมโครคอนโทรลเลอร์	46
6.1 แสดงหน้าจอการใช้งานของโปรแกรมทดสอบไอซี ทีทีแอล	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ (Introduction)

ในการประกอบวงจร หรือทำการทดลองวงจรที่ต้องใช้ไอซีประเภท TTL บางครั้งอาจเจอเหตุบกพร่องในวงจร ซึ่งอาจจะเกิดขึ้นเนื่องจากหลายสาเหตุ รวมไปถึงการทำงานที่ไม่สมบูรณ์ของไอซีทีทีแอลที่ใช้ในวงจร โดยทั่วไปแล้วภายในแพ็กเก็จของไอซีทีทีแอล จะประกอบไปด้วยวงจรที่ทำหน้าที่ตามเบอร์ของไอซีนั้นๆหลายวงจร เช่น 7400 ประกอบไปด้วย Not Gate 6 ตัว และเป็นไปได้สูงที่จะมี Gate เพียงตัวใดตัวหนึ่งภายในแพ็กเก็จที่เสีย ซึ่งตัวอื่นยังสามารถใช้งานได้

เครื่องตรวจสอบไอซีทีทีแอล ที่ทำงานร่วมกับ ไมโครคอมพิวเตอร์ สามารถที่จะแสดงรายละเอียด ผลการตรวจสอบในรูปแบบที่เข้าใจง่าย ของทุกๆฟังก์ชันที่มีในแพ็กเก็จ ทำให้สามารถเลือกใช้งานเฉพาะฟังก์ชันที่ยังสมบูรณ์อยู่ และการที่มี ไมโครคอมพิวเตอร์ทำงานร่วมกับเครื่องตรวจสอบไอซีทีทีแอล ทำให้ประหยัดทรัพยากรทางด้าน Memory ที่ใช้ในเครื่องตรวจสอบ ซึ่งไมโครคอมพิวเตอร์ ในปัจจุบันก็ได้มีการนำมาใช้งานกันอย่างกว้างขวางอยู่แล้ว

เครื่องตรวจสอบไอซีทีทีแอลนี้ จะสำเร็จได้ ต้องใช้ความรู้ในหลายๆด้านประกอบกัน ไม่ว่าจะเป็นความรู้ทางด้านวงจร และการประกอบวงจร , ความรู้ทางด้านการโปรแกรมไมโครคอนโทรลเลอร์ ตลอดจนความรู้ทางด้านฐานข้อมูล และการโปรแกรมโดยใช้ Visual Basic ทำงานร่วมกับฐานข้อมูลที่สร้างขึ้นมา

ขอบเขตการทำงานของ เครื่องตรวจสอบไอซีทีทีแอลนี้ คือสามารถทำการตรวจสอบไอซีทีทีแอลที่มีขาไม่เกิน 24 ขา ประเภทของไอซีที่ตรวจสอบได้คือ ไอซีที่เป็นแบบ Basic gate และรวมถึงไอซีประเภทที่ใช้สัญญาณนาฬิกา ที่ใช้งานโดยทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

IC TTL

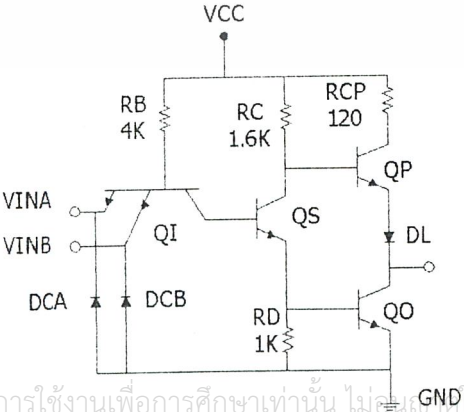
จุดมุ่งหมายของการผลิตวงจรไอซีทีทีแอลมาตรฐาน การออกแบบและการสร้าง เพื่อให้มีขนาดเล็ก กินไฟน้อย (Low Power Consumption) มีความเร็วสูง และมีความเชื่อถือที่ไว้ใจได้สูง ได้แบ่งวงจรทีทีแอลเป็น 4 ตระกูล คือ

- วงจรไอซีทีทีแอล ตระกูลมาตรฐาน (Standard SN 54 / 74)
- วงจรไอซีทีทีแอล ตระกูลที่กินไฟน้อย (Lower power SN 54L / 74L)
- วงจรไอซีทีทีแอล ตระกูลที่มีความเร็วสูง (High Speed SN 54H / 74H)
- วงจรไอซีทีทีแอล ตระกูลชื่อตกี้ (Schottky Diode clamps SN 54S / 74S)

2.1 ตระกูลของไอซีทีทีแอล

2.1.1 ไอซีทีทีแอลตระกูลมาตรฐาน (SN 54/74)

จากวงจรรูปที่ 2.1 แสดงวงจรไอซีทีทีแอลตระกูลมาตรฐาน จะเห็นว่าอินพุทเป็นแบบ Multiple Emitter Input ส่วนทางด้านเอาท์พุทใช้ Q_4 เป็น Active Pull Up ข้อดีของ Q_1 ที่เป็นแบบ Multiple Emitter คือเป็นทรานซิสเตอร์ขนาดเล็กมากและทำให้เกิดความเร็วสูง เอาท์พุทของ Q_4 ทำให้เกิดเอาท์พุทอิมพิแดนซ์ต่ำ ข้อดีคือ Improve Noise Immunity และทำให้ความเร็วสูง สรุปก็คือ ไอซีทีทีแอลตระกูล 54/74 เป็นวงจรที่มีความเร็วสูงพอสมควร และกินไฟไม่มาก ค่า Delay time ต่อเกท(เวลาที่ใช้ไป เมื่ออินพุทเข้าสู่เกท และไปปรากฏที่เอาท์พุทของเกท) ประมาณ 10 nS เหมาะที่จะนำไปใช้ในงานด้านต่าง ๆ เช่น ใช้ในวงจร Shift Register, Counter, Decoder

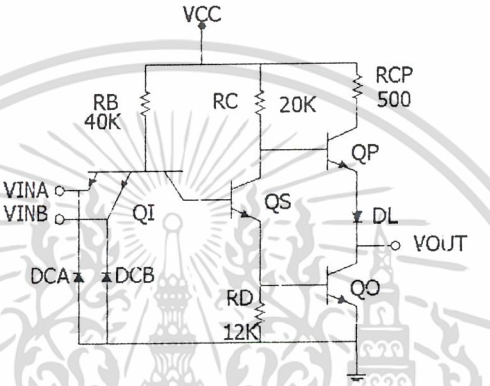


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 วงจรไอซีทีทีแอล ตระกูลมาตรฐาน SN 54/74

2.1.2 ไอซีทีทีแอลตระกูลที่กินไฟน้อย (SN 54L / 74L)

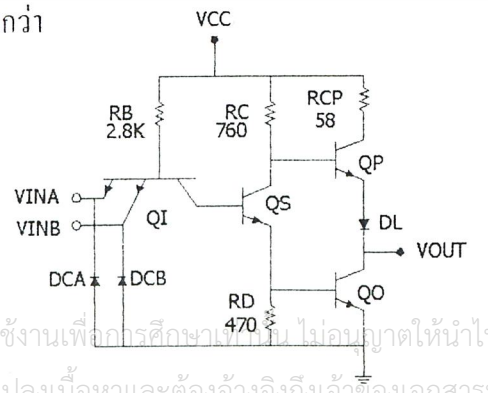
จากวงจรรูปที่ 2.2 นี้จะคล้ายกับวงจรในรูปที่ 2.1 แต่จะต่างกันตรงที่ความต้านทาน ซึ่งในวงจรรูปที่ 2.2 จะมีค่าความต้านทานมากกว่า ซึ่งการเพิ่มค่าความต้านทานนี้ทำให้วงจรกินกระแสน้อยลง แต่มีความเร็วต่ำ คือ มีความเร็วประมาณ 1 ใน 10 ของไอซีทีทีแอลตระกูล มาตรฐานเท่านั้น แต่ก็กินกระแสน้อยกว่าเกือบ 10 เท่า



รูปที่ 2.2 วงจรไอซีทีทีแอล ตระกูลที่กินไฟน้อย SN 54L / 74L

2.1.3 ไอซีทีทีแอล ตระกูลที่มีความเร็วสูง (SN 54H / 74H)

จากวงจรรูปที่ 2.3 เป็นวงจรไอซีทีทีแอลที่มีความเร็วสูง SN 54H / 74H จะเห็นว่าส่วนใหญ่จะคล้ายกับวงจรในรูปที่ 2.1 แต่จะต่างกันตรงที่ จะใช้ค่าความต้านทานที่มีค่าต่ำกว่า และใช้ไดโอดต่อแบบ Clamping กับอินพุททุกอินพุท เพื่อป้องกันและลดสัญญาณที่มาจากสายส่งสัญญาณสูงกว่าตระกูล SN 54 / 74 ประมาณ 6 nS ต่อเกท เพราะเกิดอินพุทอิมพิแดนซ์ทางด้านเอาท์พุท เข้าสู่อินพุทของวงจร เพราะสัญญาณรบกวนจะมีผลกระทบต่อวงจรที่มีค่า Rise time และค่า Fall time ที่ค่อนข้างเร็ว เมื่อมองที่เอาท์พุทจะพบว่า Q_3 และ Q_4 ต่อกันแบบ Darlington ทำให้ความเร็วที่อยู่ในช่วงที่วงจรอยู่ในสภาวะสเตเบิ้ล ซึ่งเอาท์พุทอิมพิแดนซ์มีค่าประมาณ 10 ถึง 100 โอห์ม ข้อเสียคือ ถึงแม้จะมีความเร็วที่สูงแต่ก็กินไฟมากกว่า



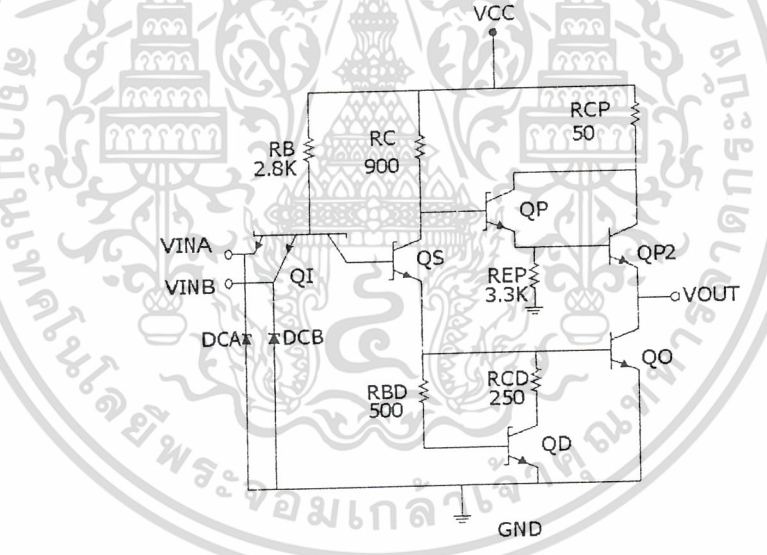
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 วงจรไอซีทีทีแอล ตระกูลที่มีความเร็วสูง SN 54H / 74H

2.1.4 ไอซีทีทีแอล ตระกูลช็อตกี้ (SN 54S / 74S)

จากวงจรรูปที่ 2.4 แสดงวงจรไอซีทีทีแอล ตระกูลช็อตกี้ SN 54S / 74S จะมีความเร็วสูง เพราะวงจรจะประกอบด้วย Schottky Barrier Transistor คือ ทรานซิสเตอร์ธรรมดาที่ทำงานร่วมกับ Schottky Diode การต่อเข้าด้วยกันเพื่อไม่ให้ทรานซิสเตอร์ทำงานที่จุดอิ่มตัว (Saturation Region)

ข้อดีของ Schottky Diode ที่มีเหนือกว่าไดโอดธรรมดาก็คือไม่มี Minority Carrier หรือถ้ามีก็น้อยมาก ดังนั้นจึงไม่มีประจุสะสมอยู่ที่บริเวณ Junction หรือบริเวณใกล้เคียง นอกจากนี้ Schottky Diode ยังทำงานที่ Forward Voltage drop ที่ต่ำกว่าไดโอดธรรมดา เมื่อใช้ Schottky Diode ต่อ Clamp เข้ากับอินพุทของ Q_1 เลยบังคับให้ Q_1 ทำงานที่จุดอิ่มตัว และอีกประการหนึ่ง เมื่อไม่มีประจุไฟฟ้าสะสมอยู่หรือค้างอยู่ ทั้งใน Schottky Diode และทรานซิสเตอร์หรือมีค่าน้อย ดังนั้นการเปลี่ยนแปลงสถานะของทรานซิสเตอร์จึงเป็นไปอย่างรวดเร็ว



รูปที่ 2.4 วงจรไอซีทีทีแอล ตระกูลช็อตกี้ SN 54S / 74S

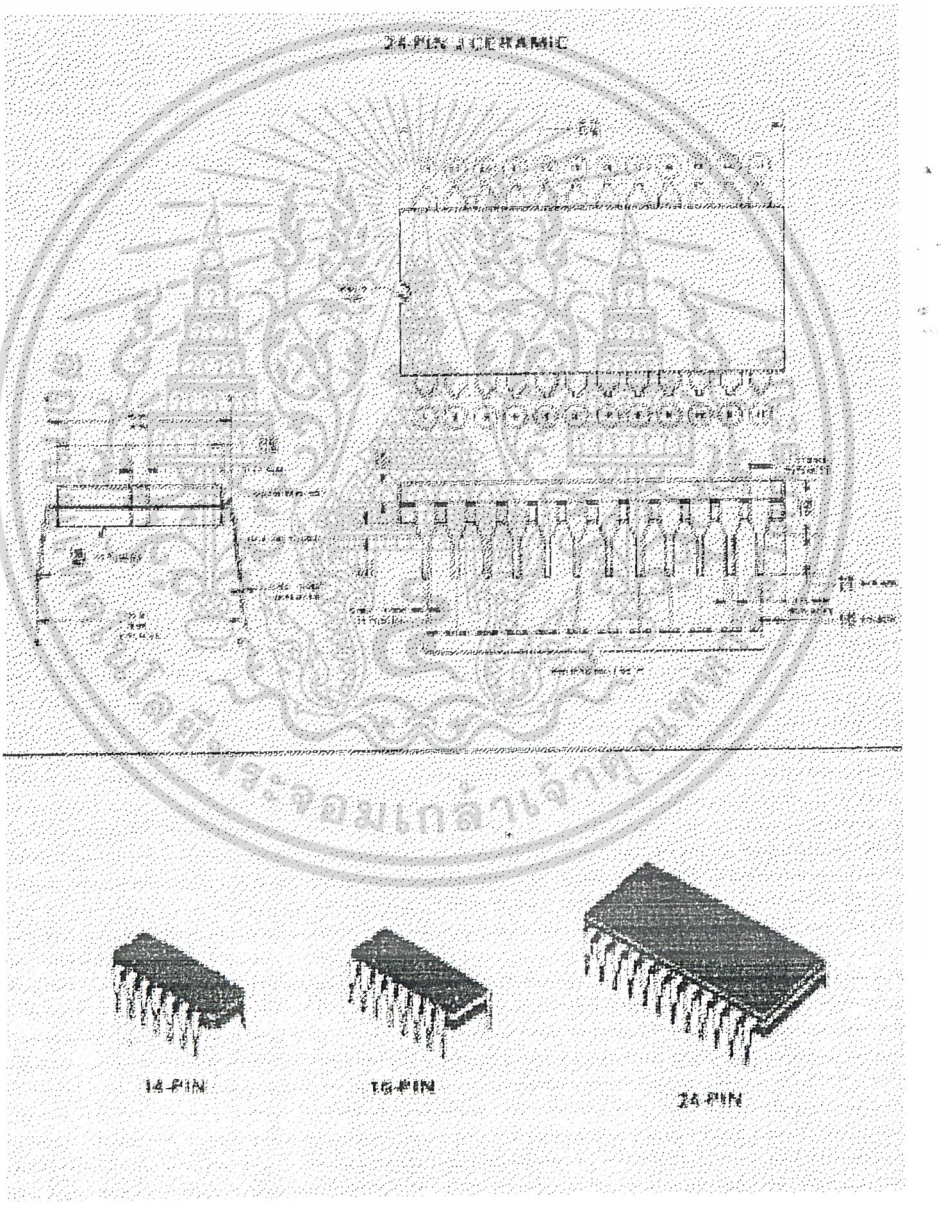
รูปที่ 2.5 Schottky Barrier Transistor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณสมบัติและลักษณะของไอซีทีทีแอล

2.2.1 ลักษณะและแพ็กเกจของไอซีทีทีแอล

วงจรทีทีแอลจะถูกแพ็กอยู่ในแพ็กเกจที่ทำด้วยพลาสติกหรือเซรามิค โดยทั่วไปเป็นรูปสี่เหลี่ยมผืนผ้า ขาของลอจิกจะถูกต่อออกมาอย่างเป็นระเบียบ โดยทั่วไปจะมี 14, 16, 20 และ 24 ขา ดังรูปที่ 2.6



รูปที่ 2.6 แสดงแพ็กเกจแบบต่าง ๆ ของทีทีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ระบบไฟจ่ายวงจรให้กับทีทีแอล

คุณสมบัติที่สำคัญซึ่งควรจะรู้จักของวงจรไอซีทีทีแอล ทั้ง 4 ตระกูล คือ ไฟจ่ายวงจร 5 โวลท์ ลอจิก "0" ทางด้านเอาท์พุท เท่ากับ 0.2 โวลท์ ลอจิก "1" ทางด้านเอาท์พุท เท่ากับ 3.0 โวลท์ Noise Immunity (ค่าความแตกต่างระหว่างอินพุทที่เป็นลอจิก "1" กับ ลอจิก "0" เท่ากับ 1.0 โวลท์)

การเลือกระบบจ่ายไฟให้แก่วงจรควรพิจารณาถึงแรงดันไฟฟ้าที่ออกมาคงที่ โดยมี Regulator ช่วยให้มีเอาท์พุทอิมพิแดนซ์ต่ำ มีระบบป้องกันไม่ให้มีกระแสทรานเซี้ยนสูง โดยเฉพาะในช่วงการเปลี่ยนสเตทของเอาท์พุททางทีทีแอล คือ ต้องมีการ Decouping วงจรจ่ายไฟในที่ที่เหมาะสม โดยทั่วไปแล้วต้องการไฟจ่ายให้แก่วงจร 5 โวลท์ และสามารถทำงานได้ตามปกติ ที่จ่ายไฟวงจรต่ำถึง ±250 มิลลิโวลท์ วิธีเช็คหาจำนวนกระแสไฟที่ต้องการ คือ หากระแสที่ไหลในวงจรลอจิกเกทแต่ละตัวแล้วบวกกัน หรือจากในหนังสือคู่มือทีทีแอล ทำให้ทราบว่าลอจิกเกทกินไฟฟ้ากี่มิลลิวัตต์ต่อเกทแล้วนำมารวมกัน เมื่อเอาค่าไฟจ่ายวงจรไปหารจะได้จำนวนกระแสไฟฟ้าตามต้องการ

2.3 การทำงานของไอซีทีทีแอลประเภทต่าง ๆ

2.3.1 ไอซีลอจิกเกท พื้นฐานทั่วไป

ทีทีแอลเกทแบบ NAND GATE

จากรูปที่ 2.7 จะเห็นว่าถ้าแต่ละสัญญาณอินพุทเป็นลอจิก "1" หมด จะได้สัญญาณเอาท์พุทเป็นลอจิก "0" แต่ถ้าตัวใดตัวหนึ่งหรือมากกว่าเป็นลอจิก "0" จะได้เอาท์พุทเป็นลอจิก "1" จากรูป Q1 เป็นอินพุท Q_2, Q_3 เป็น Driver ส่วน Q_4 เป็น Active Pull up ซึ่งเป็นเอาท์พุทของวงจร และจากรูปที่ 2.8 เป็นกราฟแสดงสัญญาณอินพุทต่อสัญญาณเอาท์พุท สมมุติ Vin เป็นสัญญาณอินพุท ถ้าอินพุทมีค่าน้อยกว่า Va แล้ว Q1 จะทำงานในช่วงไม่อิ่มตัว ทำให้ Q_2 ไม่ทำงานหรือไม่นำกระแส ทำให้ Q_3 ไม่ทำงานด้วย ทำให้ Q_4 ทำงานนำกระแสเต็มที่ ดังนั้นเอาท์พุทจะเท่ากับไฟที่จ่ายให้วงจร VCC สมการของเอาท์พุทเขียนได้ดังนี้

$$V_{OH} = V_{CC} - V_{CE\,(Q4)} - V_F$$

โดยที่ V_{OH} คือเอาท์พุทโวลท์เตจที่เป็นลอจิก "1"

V_F คือ Forward Voltage ที่ตกคร่อมไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

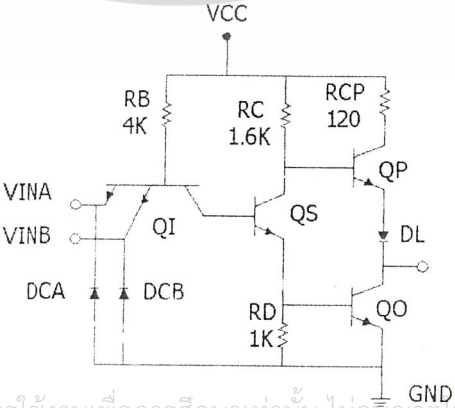
ในช่วงที่มีอินพุท Vin น้อยกว่า V_2 กระแสไฟฟ้าจำนวนหนึ่งจะไหลออกมาจาก Emitter ของ Q_1 คือ $I_{E1} = V_{CC}/R1$ ขณะที่ Q_4 เป็น Emitter Follower ทำให้เอาท์พุทอิมพิแดนซ์มีค่าต่ำ ถ้า V_1 มีค่ามากกว่า V_C จะทำให้ Q_1 ไม่ทำงานไม่นำกระแสเลย ส่วน Q_2 และ Q_3 จะนำกระแส ช่วงนี้เอาท์พุทโวลท์เตจจะมีค่าค่อนข้างต่ำใกล้ศูนย์ ทำให้เอาท์พุทเป็นลอจิก "0" จะเห็นว่าค่าสัญญาณเอาท์พุทโวลท์เตจ คือ ค่าแรงดัน $V_{CE\,(sat)}$ ของ Q_3 นั่นเอง วงจรรูปที่ 2.7 ต่อกับโหลดภายนอก จะมีกระแสไฟฟ้าจากภายนอกไหลเข้าสู่ Q_3 กระแสไฟฟ้าดังกล่าวเรียกว่า "Isink" ช่วงนี้จะได้สมการเอาท์พุทดังนี้

$$V_{OL} = V_{CE\,(sat)} = R_{CE\,(sat)}\,Isink$$

โดยที่ V_{OL} คือ เอาท์พุทโวลท์เตจที่เป็นลอจิก "0"

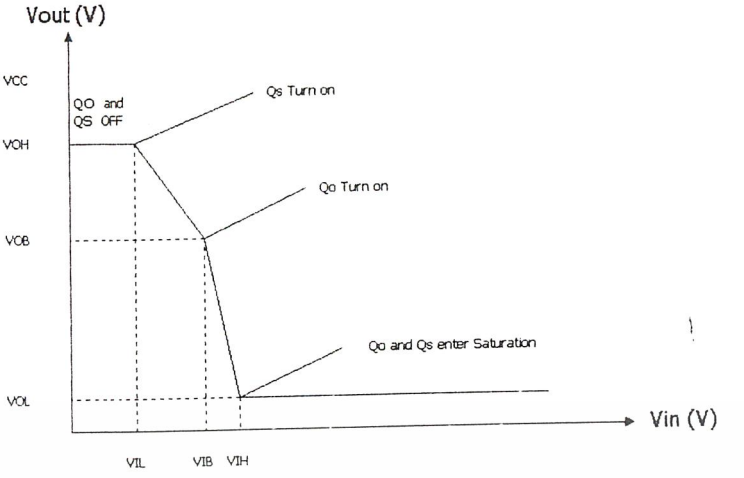
$R_{CE(sat)}$ คือ ความต้านทานระหว่างขา Collector กับขา Emitter ของทรานซิสเตอร์ Q_3

จะเห็นได้ว่าในช่วง Va < Vin < Vc วงจรจะเปลี่ยนสเตทจากลอจิก "0" ป้อนอินพุท Vin เข้าไปโดยเพิ่มค่าจาก 0 โวลท์ ขึ้นไปทางบวกเรื่อย ๆ ขณะที่เพิ่มค่าขึ้นนั้นกระแสไฟฟ้าที่ไหลจากขาเบสของทรานซิสเตอร์ Q_1 เข้าสู่ขา Emitter และจะเปลี่ยนทิศทางจากเบสของ Q_1 ไปขา Collector ของ Q_1 นั่นคือเข้าสู่ขาเบสของ Q_2 จนกระทั่ง Vin = Va =0.7 ทำให้ Q_2 นำกระแส ขณะนี้ Q_2 ทำงานในย่าน Linear ด้วยค่า Gain = R2 / R3 ช่วงนี้ Q_4 ยังนำกระแสอยู่และเอาท์พุทของ Q_2 จะมีค่าตาม Gain ของ Q_2 และเอาท์พุทของ Q_4 จะลดลงตาม Slope R2 / R3 นั่นคือลดจากจุด a ลงสู่จุด b นี้ อินพุท V_1 จะสูงพอที่สามารถทำให้ Q_3 นำกระแส ซึ่งเป็นเหตุให้ค่า The emitter impedance ของ Impedance ของ Q_2 ลดลง Gain ของ Q_2 จะเพิ่มขึ้นทันที จากรูปที่ 2.8 Slope a-b คือค่า Gain ของ Q_2 ($A_{Q2} = R2 \,//\, R3 \,//\, R_{BE(Q3)}$) และทรานซิสเตอร์ Q_4 จะไม่นำกระแส ที่อินพุท Vin เท่ากับ Vc ขณะนี้เอาท์พุทของวงจรเป็นลอจิก "0"



รูปที่ 2.7 แสดงลักษณะวงจรและสัญญาณทางลอจิกของไอซีทีทีแอล NAND GATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



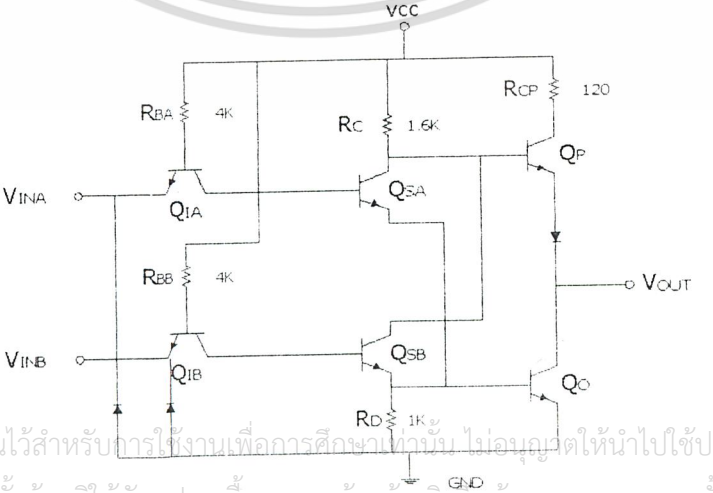
รูปที่ 2.8 แสดงลักษณะสัญญาณอินพุท – เอาท์พุทของวงจรไอซีทีทีแอล NAND GATE

ทีทีแอลแบบ AND GATE

วิธีสำหรับการสร้างลอจิก AND เกท โดยดัดแปลงจาก NAND เกท จากนั้นก็ต่อ Inverter เข้าที่เอาท์พุทของ NAND เกท ก็จะได้ AND เกท อีกวิธีที่จะทำให้ได้ AND เกทโดยใช้ NAND เกท ต่อกัน 2 ตัว ค่า Delay time ของ AND เกทจะมากขึ้นและกินไฟมากขึ้นด้วย แต่ถ้าเลือก NAND เกท ที่มีความเร็วสูงมาดัดแปลงจะทำให้ได้ AND เกทที่มีค่า Delay time สูงกว่า NAND เกท ประมาณ 1 nS และกินไฟมากกว่าไม่เกิน 5 mW ต่อเกท

ทีทีแอลแบบ NOR GATE

จากวงจรรูปที่ 2.9 เป็นวงจรลอจิก NOR เกท โดยถ้าอินพุทตัวใดตัวหนึ่งหรือทั้งสองเป็นลอจิก "1" Q_2 จะนำกระแสไฟฟ้า ทำให้ Q_3 นำกระแสด้วย ส่วน Q_4 จะยังไม่นำกระแส ดังนั้นด้านเอาท์พุทจะเป็นลอจิก "0" ($V_{OL} \leq 0.4$ V) แต่ถ้าอินพุทเป็นลอจิก "0" ทั้งสอง ($V_{IL} \leq 0.8$ V) ทำให้ Q_2 และ Q_3 ไม่นำกระแส ส่วน Q_4 จะนำกระแส ได้เอาท์พุทเป็นลอจิก "1" ($V_{OH} \geq 2.4$ V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9 วงจรลอจิก NOR เกท

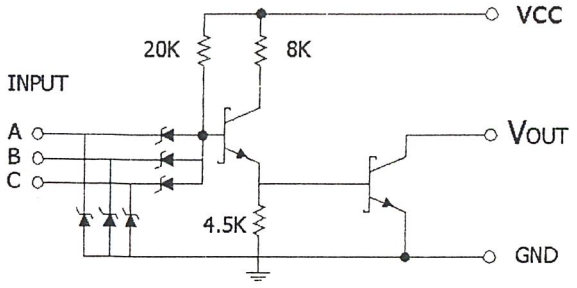
ทีทีแอลแบบ AND – OR – INVERTER (AOI) และ EXPANDER GATE

พิจารณาจากสมการบูลีน คือ $Y = \overline{AB} + \overline{CD} = \overline{AB + CD}$ จะเห็นว่าสามารถสร้างวงจรลอจิก ที่ให้ได้เอาท์พุท Y โดยใช้ทีทีแอล Open Collector NAND เกท ที่เอาท์พุทต่อเป็น Wire – AND คล้ายวงจรในรูปที่ 2.11 ก) ได้ อย่างไรก็ตามถ้าสมการ Y เป็นสมการที่ยาว คือ ประกอบด้วยเทอมหลาย ๆ เทอม (หมายความว่ามีจำนวนอินพุทหลาย ๆ อินพุทนั่นเอง) ถ้าใช้ทีทีแอล Open Collector NAND เกทนั้น จะให้ Fan out ต่ำ นอกจากนี้วงจรจะกินกระแสเพราะ RLที่ต่อมีค่าสูงเกินไป ทำให้เกิดปัญหาด้านความเร็วจะมีค่าต่ำลง ด้วยเหตุผลดังกล่าวการออกแบบวงจรลอจิกทีทีแอล AOI จึงมีจำนวน Fan out สูงสุดถึง 10 และค่า Delay time จะมีค่าสูงสุดประมาณ 15 nS เท่านั้น ที่จริงแล้ว AOI เกท เป็นวงจรเป็นไอซีคอมบิเนชั่นลอจิกที่ประกอบด้วย AND เกท (NAND + INVERTER) และ OR เกท (NOR +INVERTER)

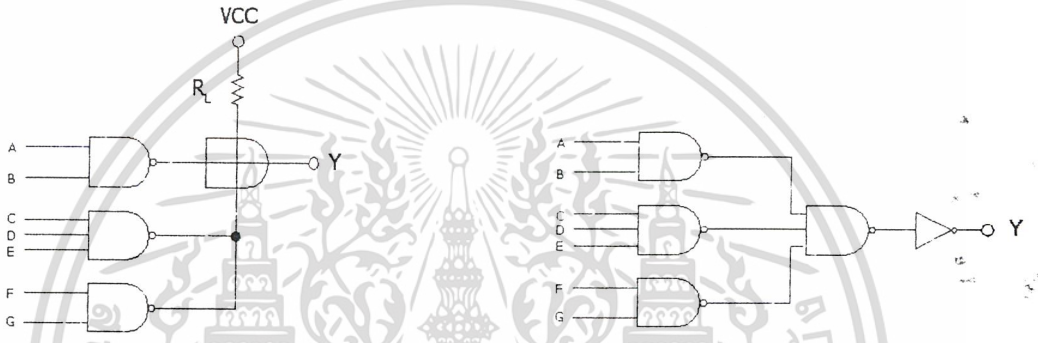
2.3.2 ไอซีลอจิกเกทแบบ Open Collector

ประโยชของทีทีแอลที่มีเอาท์พุทแบบ Totem Pole หรือ Active Pull up จะมีลักษณะของตัวจ่ายกระแสที่มีค่าอิมพิแดนซ์ที่ค่อนข้างต่ำ สำหรับโหลดที่เป็นแบบคาปาซิทีฟ ทำให้เกิดการอัดและคายประจุอย่างรวดเร็วกว่าวงจรที่เป็น RC ทั่ว ๆ ไป คือ เอาท์พุทอิมพิแดนซ์ต่ำ สามารถรับโหลดภายนอกได้สูง รวมทั้งสัญญาณรบกวนต่ำและมีความเร็วสูง ส่วนทีทีแอลที่มีเอาท์พุทเป็นแบบ Open Collector มีโครงสร้างดังรูปที่ 2.10 ข้อดีของทีทีแอลแบบ Open Collector ทีทีแอลเกทก็คือสามารถต่อเอาท์พุทกันได้โดยตรง ทำให้จำนวนเกทที่ใช้ลดลง จากรูปที่ 2.11 (ก) เป็นการต่อเอาท์พุทของ Open Collectorทีทีแอลเกททั้งหลายเข้าด้วยกัน ทำให้ได้เอาท์พุท Y ตามต้องการ และวงจรก็ทำหน้าที่เป็น AND เกท การต่อเอาท์พุทแบบนี้เรียกว่า Wire – AND ส่วนรูปที่ 2.11 (ข) จะมีหน้าที่เหมือนกับในรูปที่ 2.11 (ก) แต่ในรูปที่ 2.11 (ข) ใช้ Totem Pole ลอจิกเกทซึ่งไม่สามารถต่อเหมือนกับรูปที่ 2.11 (ก) ได้ จะเห็นว่าแม้เอาท์พุทเกทตัวใดเป็น “0” ก็จะไม่ทำให้วงจรเสียหาย เพราะมีความต้านทาน R_{EXT} จะเปลี่ยนแปลงตามจำนวนเกทที่ต่อร่วมกันอยู่ทางด้านเอาท์พุท ทั้งนี้ถ้าเป็นแบบ Totem Pole การต่อเอาท์พุทเข้าด้วยกันจะไม่สามารถทำได้เพราะในกรณีที่เอาท์พุทตัวใดตัวหนึ่งเป็น “0” จะมีกระแส I_{OL} ที่มีค่าสูงประมาณ 55 mA ซึ่งสูงพอที่จะทำลายเอาท์พุทตัวอื่นให้เสียหายได้ ขณะที่วงจรปกติมีค่ากระแส I_{OL} สูงประมาณ 16 mA เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 โครงสร้างของ Open Collector ที่ทีแอลเกท



ก) ต่อแบบ Wire AND โดยใช้ลอจิกเกท แบบ Open Collector

ข) ต่อแบบธรรมดาโดยใช้ลอจิก แบบ Totem Pole

รูปที่ 2.11 เปรียบเทียบวงจรลอจิก ทีทีแอลเกทแบบ Open Collector กับ Totem Pole

2.3.3 ไอซีลอจิกเกทประเภทที่ต้องการ Clock

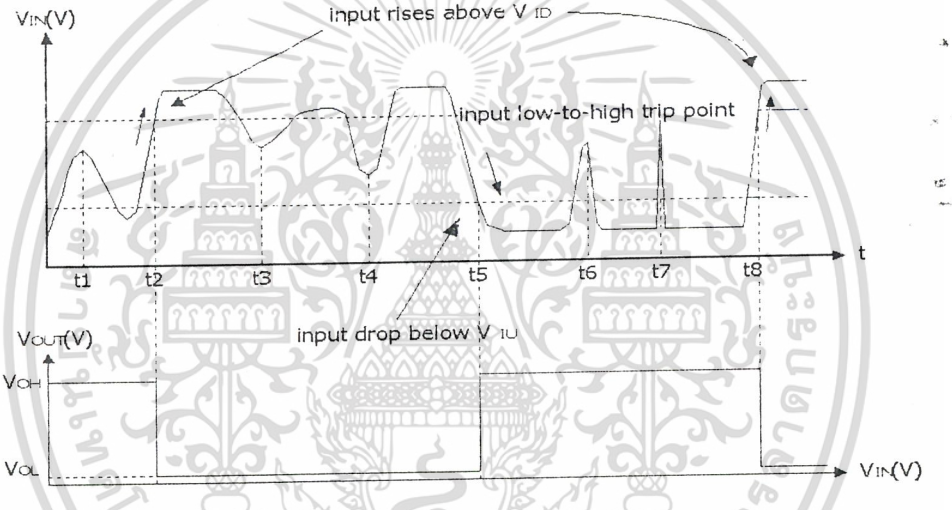
ไอซีทีทีแอลประเภทนี้ เมื่อจะให้ทำงาน นอกจากจะต้องป้อน VCC และ Ground แล้ว ยังมีอินพุทที่ต้องการ Clock อีกด้วย เมื่อจะให้ไอซีทำงานจะต้องป้อน Clock ไปพร้อมกับการป้อนลอจิกทางอินพุท ซึ่งได้แก่ ไอซีประเภท Flip Flop, Shift Register

2.3.4 ไอซี Monostable Mutivibrator

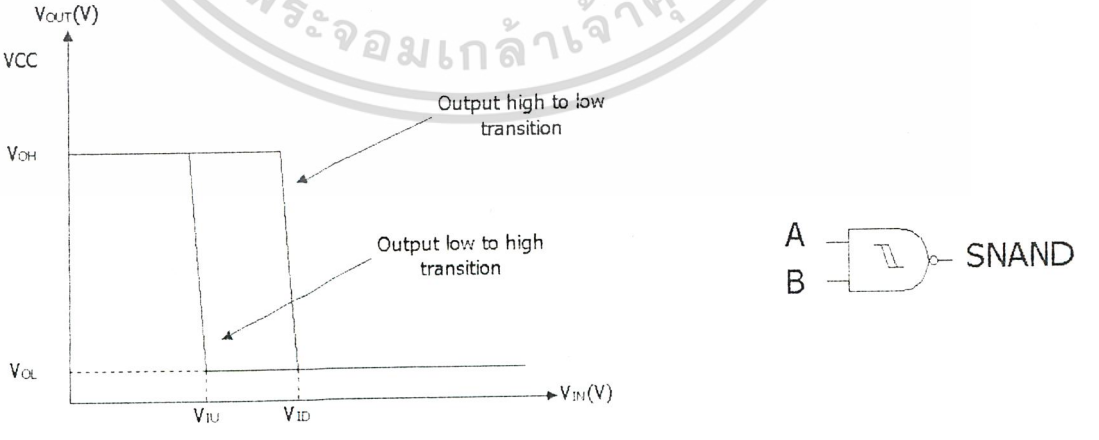
ถ้าสัญญาณที่ป้อนเข้าอินพุทของเกท เป็นลักษณะที่ไม่เป็นระเบียบ ดังรูปที่ 2.12 เช่น สัญญาณอนาลอก ซึ่งมีระดับสัญญาณสูงต่ำไม่สม่ำเสมอ หรือมีสัญญาณ Ripple ที่สูง ถ้าป้อนสัญญาณอินพุทของเกทโดยตรง อาจทำให้ลอจิกเกททำงานผิดพลาด นอกจากนี้การหาค่า Delay time ของเกท ก็จะมีปัญหา ดังนั้นจึงต้องมีวงจร Schmitt Trigger NAND เกท ทำหน้าที่เปลี่ยนสัญญาณที่ไม่เป็นระเบียบให้เป็นสัญญาณรูปสี่เหลี่ยมที่เป็นระเบียบก่อน แล้วจึงผ่านเข้าสู่อินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของลอจิกเกท อนึ่งการเปลี่ยนสัญญาณดังกล่าวนี้ มิได้ทำให้ข้อมูลเดิมของสัญญาณที่ไม่เป็นระเบียบเปลี่ยนแปลงแต่อย่างใด รูปที่ 2.12 (ก) แสดงลักษณะของสัญญาณก่อนและหลัง ทีทีแอล Schmitt Trigger NAND เกท ที่มีเอาท์พุทที่เป็นแบบ Totem Pole ส่วนรูปที่ (ข) เป็นการแสดงให้เห็นว่าถ้าป้อนอินพุทเข้าที่ลอจิกเกทโดยตรงโดยไม่ผ่าน Schmitt Trigger NAND เกทก่อน จะได้สัญญาณที่เป็นเส้นประ ส่วนเส้นกราฟที่เป็นเส้นทึบรูปสี่เหลี่ยมเป็นสัญญาณที่ออกจากเอาท์พุทของลอจิกเกท โดยมีการป้อน Vin เข้าที่ Schmitt Trigger NAND เกทก่อน สำหรับรูปที่ 2.13 แสดงลักษณะอินพุทและเอาท์พุทจะเห็นว่ามี Hysteresis Voltage เท่ากับ 800 mV



รูปที่ 2.12 แสดงลักษณะสัญญาณอินพุทเมื่อผ่านวงจร Schmitt Trigger NAND เกท



รูปที่ 2.13 ลักษณะอินพุทและเอาท์พุท ของ Schmitt Trigger NAND เกท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ค่าพารามิเตอร์ (Parameter) ที่สำคัญของไอซีทีทีแอลลอจิก

2.4.1 พารามิเตอร์ที่เป็นแรงดัน

V_{IH} **High – level input voltage**

ระดับแรงดันอินพุทที่อยู่ภายในค่าบวกที่มีค่ามากกว่า (หรือค่าที่เป็นลบน้อยกว่า) ของค่า 2 ช่วง ซึ่งใช้แทนระดับ High และ Low ค่าที่ระบุเป็นค่าที่ต่ำสุดของระดับ High ซึ่งยังคงทำให้ไอซีทำงานได้ตามเงื่อนไขที่ระบุ

V_{IL} **Low – level input voltage**

ระดับแรงดันอินพุทที่อยู่ภายในค่าบวกที่มีค่าน้อยกว่า (หรือค่าที่เป็นลบมากกว่า) ของค่า 2 ช่วง ซึ่งใช้แทนระดับ High และ Low ค่าที่ระบุเป็นค่าที่สูงสุดของระดับ Low ซึ่งยังคงทำให้ไอซีทำงานได้ตามเงื่อนไขที่ระบุ

V_{T+} **Positive – going Threshold voltage**

ระดับแรงดันที่อินพุทซึ่งทำให้อุปกรณ์ลอจิกนั้น ๆ ทำงานตามที่ระบุมาในข้อมูล ขณะที่แรงดันอินพุทเพิ่มขึ้นจากระดับที่ต่ำกว่า VT+

V_{T-} **Negative - going Threshold voltage**

ระดับแรงดันที่อินพุทซึ่งทำให้อุปกรณ์ลอจิกนั้น ๆ ทำงานตามที่ระบุมาในข้อมูล ขณะที่แรงดันอินพุทเพิ่มขึ้นจากระดับที่ต่ำกว่า VT-

V_{OH} **High – level output voltage**

แรงดันที่ขั้วเอาท์พุทในขณะที่ให้กระแสเอาท์พุท IOH ในภาวะที่อินพุทอยู่ในเงื่อนไขตามที่ระบุในข้อมูล ซึ่งจะให้ระดับแรงดันเป็น High ที่เอาท์พุท

V_{OL} **Low – level output voltage**

แรงดันที่ขั้วเอาท์พุทในขณะที่ให้กระแสเอาท์พุท IOH ในภาวะที่อินพุทอยู่ในเงื่อนไขตามที่ระบุในข้อมูล ซึ่งจะให้ระดับแรงดันเป็น Low ที่เอาท์พุท

$V_{O\ (on)}$ **On – State output voltage**

แรงดันที่ขั้วเอาท์พุทสำหรับค่ากระแสที่เอาท์พุท ที่ระบุในภาวะอินพุท อยู่ในเงื่อนไขตามที่ระบุในข้อมูล ซึ่งจะทำให้อุปกรณ์สวิทช์ที่เอาท์พุทอยู่ในสภาวะ On

$V_{O\ (off)}$ **Off - State output voltage**

แรงดันที่ขั้วเอาท์พุทสำหรับค่ากระแสที่เอาท์พุท ที่ระบุในภาวะอินพุท อยู่ในเงื่อนไขตามที่ระบุในข้อมูล ซึ่งจะทำให้อุปกรณ์สวิทช์ที่เอาท์พุทอยู่ในสภาวะ Off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : ลักษณะสมบัติข้อนี้มักจะใช้ระบุมาในกรณีที่ภาคเอาท์พุทไม่มีอุปกรณ์ Pull up อยู่ภายใน

2.4.2 พารามิเตอร์ที่เป็นกระแส

I_{IH} **High – level input current**

กระแสที่ไหลเข้าขั้วอินพุทเมื่อระดับแรงดัน High ที่ระบุ ถูกป้อนเข้ามาที่ขั้วเอาท์พุทนั้น

I_{IL} **Low - level input current**

กระแสที่ไหลเข้าขั้วอินพุทเมื่อระดับแรงดัน Low ที่ระบุ ถูกป้อนเข้ามาที่ขั้วเอาท์พุทนั้น

I_{OH} **High – level output current**

กระแสที่ไหลเข้าขั้วเอาท์พุทเมื่อได้รับแรงดันเอาท์พุทเป็น High

หมายเหตุ : ลักษณะสมบัติข้อนี้มักใช้ระบุ สำหรับภาคเอาท์พุทที่เป็นแบบ Open Collector ซึ่งมีเจตนาที่จะใช่ขับวงจรลอจิกอื่น ๆ

$I_{O\ (off)}$ **Off state output current**

กระแสที่ไหลเข้าขั้วเอาท์พุท เมื่อได้รับแรงดันเอาท์พุทและที่สภาวะของอินพุท ตามที่ระบุมาในข้อมูล ซึ่งจะทำให้อุปกรณ์สวิทช์ในภาคเอาท์พุทที่อยู่ในสภาวะ Off

หมายเหตุ : ลักษณะสมบัติข้อนี้มักใช้ระบุ สำหรับภาคเอาท์พุทที่เป็นแบบ Open Collector ซึ่งมีเจตนาที่จะใช่ขับวงจรลอจิกอื่น ๆ หรือสำหรับระบุภาคเอาท์พุทที่เป็น 3 สถานะ

I_{OS} **Short circuit output current**

กระแสที่ไหลเข้าที่ขั้วเอาท์พุท เมื่อขั้วเอาท์พุทถูกลัดวงจรลงกราวด์ (หรือแรงดันอื่นที่ระบุมา) ในภาวะที่อินพุทจะทำให้ ได้ระดับแรงดันที่เอาท์พุทห่างจากแรงดันที่กราวด์มากที่สุด

I_{CCH} **Supply current, output(s) High**

กระแสที่ไหลเข้าที่ขั้วรับไฟเลี้ยง VCC เมื่อเอาท์พุทที่อ้างอิงอยู่ที่ระดับแรงดัน High

I_{CCL} **Supply current, output(s) Low**

กระแสที่ไหลเข้าที่ขั้วรับไฟเลี้ยง VCC เมื่อเอาท์พุทที่อ้างอิงอยู่ที่ระดับแรงดัน Low

2.4.3 ลักษณะสมบัติที่เปลี่ยนแปลงสภาวะ (Dynamic Characteristics)

f_{max} **Max clock Frequency**

อัตราสูงสุดของสัญญาณนาฬิกาที่สามารถป้อนให้กับอินพุทของวงจรไบสเตเบิ้ล โดยยังคงทำให้การเปลี่ยนแปลงระดับแรงดันลอจิกที่เอาท์พุทมีเสถียรภาพ ขณะที่อินพุทได้รับเงื่อนไขซึ่งควรจะทำให้เกิดการเปลี่ยนแปลงภาวะที่เอาท์พุท ทุกครั้งที่ได้รับ Pulse ของสัญญาณนาฬิกา

t_{HZ} **Output disable time (of a three state output) from High level**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาระหว่างตำแหน่งที่ระบุบนรูปคลื่นของแรงดันอินพุทและเอาท์พุท โดยที่เอาท์พุทที่เป็น Tri - state กำลังจะเปลี่ยนจากระดับแรงดัน High ไปเป็นภาวะ high impedance (off)

t_{LZ} **Output disable time (of a three state output) from Low level**

เวลาระหว่างตำแหน่งที่ระบุบนรูปคลื่นของแรงดันอินพุทและเอาท์พุท โดยที่เอาท์พุทที่เป็น Tri - state กำลังจะเปลี่ยนจากระดับแรงดัน Low ไปเป็นภาวะ high impedance (off)

t_{PHL} **Propagation delay time , High – to – Low level output**

เวลาระหว่างตำแหน่งที่ระบุบนรูปคลื่นของแรงดันอินพุทและเอาท์พุท โดยที่เอาท์พุทกำลังจะเปลี่ยนจากระดับแรงดัน High ที่กำหนดไปเป็นระดับ Low ตามที่กำหนด

t_{TLH} **Transition time , Low – to – High level output**

เวลาระหว่างแรงดันระดับ Low ที่ระบุ และระดับ High ที่ระบุ บนรูปคลื่นที่กำลังเปลี่ยนจากระดับ Low ตามที่กำหนดไปเป็นระดับ High ตามที่กำหนด

t_{THL} **Transition time , High – to – Low level output**

เวลาระหว่างแรงดันระดับ High ที่ระบุ และระดับ Low ที่ระบุ บนรูปคลื่นที่กำลังเปลี่ยนจากระดับ High ตามที่กำหนดไปเป็นระดับ Low ตามที่กำหนด

t_W **Average pulse width**

เวลาระหว่างจุดที่มีขนาด 50 เปอร์เซ็นต์ (หรือที่จุดอื่น ๆ ที่ระบุ) บนขอบด้านขึ้นและด้านลงของ Pulse

t_{hold} **Hold time**

ช่วงเวลาที่สัญญาณพัลซ์ ยังคงมีที่ขั้วอินพุทที่ระบุ หลังจากที่มีการเปลี่ยนแปลงภาวะ Active เกิดขึ้นที่ขั้วเอาท์พุทอื่นที่ระบุ

$t_{Release}$ **Release time**

ช่วงเวลาระหว่างที่ปลดข้อมูล ที่ต้องการให้รับรู้ออกจากขั้วอินพุทที่อ้างอิง และการเกิดการเปลี่ยนแปลงภาวะ Active ที่ขั้วเอาท์พุทอื่นที่อ้างถึง

หมายเหตุ : ช่วงเวลาที่ระบุว่าเป็น Release time จะอยู่ภายในช่วงเวลา Set up และมีผลทำให้เกิดเป็น Negative hold time

$t_{Set\ up}$ **Set up time**

ช่วงเวลาซึ่งเมื่อป้อนสัญญาณและคงอยู่ที่ขั้วอินพุท ที่ระบุก่อนที่การเปลี่ยนแปลงในภาวะ Active จะเกิดขึ้นที่ขั้วอินพุทอื่นที่ระบุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

t_{ZH} **Output enable time (of a three state output) to High level**

เวลาระหว่างตำแหน่งที่ระบุบนรูปคลื่นของแรงดันอินพุทและเอาท์พุท โดยที่เอาท์พุทที่เป็น Tri - state กำลังจะเปลี่ยนจากภาวะ high impedance (off) ไปเป็นระดับแรงดัน High ตามที่กำหนด

t_{ZL} **Output enable time (of a three state output) to Low level**

เวลาระหว่างตำแหน่งที่ระบุบนรูปคลื่นของแรงดันอินพุทและเอาท์พุท โดยที่เอาท์พุทที่เป็น Tri - state กำลังจะเปลี่ยนจากภาวะ high impedance (off) ไปเป็นระดับแรงดัน Low ตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Serial Port

การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

3.1.การติดต่อแบบอินเตอร์รัพท์

ขบวนการอินเตอร์รัพท์ อุปกรณรอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสม็อ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพท์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

3.2.การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบต์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า"Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในVBเราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพท์ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ CommEvent เพียง ComEvReceive, ComEvSend

3.3 การใช้ Visual Basic ติดต่อผ่าน Serial port

หลักการทำงาน

เราจะทำฟอร์มสำหรับการตั้งค่าเพื่อติดต่อผ่าน Serial Portและควบคุมคำสั่งไปยังตัวไมโครคอนโทรลเลอร์ ทาง Serial Port เช่นกันคือเมื่อกดปุ่มควบคุมแต่ละปุ่มก็จะส่งคำสั่งไปยังตัวไมโครคอนโทรลเลอร์ เพื่อให้ ทำงานตามเงื่อนไขที่ได้กำหนดไว้ เช่น ส่งอักขระ "1" = ให้ไมโครคอนโทรลเลอร์ส่งค่าออกพอร์ตขนาน P1 เป็นต้น จากนั้นจะรอรับค่าจากไมโครคอนโทรลเลอร์ส่งกลับมาเพื่อยืนยันการทำงานว่าทำงานแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สร้างฟอร์ม

การออกแบบของเราจะใช้ VB Control ที่ชื่อว่า MSComm โดยที่คุณต้อง กำหนด Custom Control เข้าไปที่ เมนู Project--->Companents แล้วเลือกที่ช่อง MSComm ก็จะปรากฏเป็นรูปไอคอนโทร ศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม

องค์ประกอบในการใช้ MSComm

การตั้งค่าติดต่อกับพอร์ต

- **ComPort** คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)
- **Setting** คือ เราต้องกำหนดอัตรา Baud,Parity,Data,Stop เช่น 1200,n,8,1 เป็นต้น
- **HandShaking** คือ เราจะกำหนดได้ 4 แบบ 1.comNone 2.comXonXoff 3. comRTS 4.comTRSXonXoff

การใช้ Buffer ในการรับส่งข้อมูล

-InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา

-OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป

- **Rthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- **Sthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- **Inputlen** คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- **EOFEnable** คือ การที่บอกว่าสิ้นสุดของไฟล์(EOF) End of File
- **ParityReplace** คือ ค่าของคาเลกเตอร์ที่จะแทนในเมื่อเกิด Parity Error
- **NullDiscard** คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- **RTSEnable** คือ ทำให้มีสัญญาณ RTS (Request To Send)
- **DTSEnable** คือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

1. **CommPort** คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน

ตัวอย่าง MSComm1.CommPort=1

2. **Settings** คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings="Baudrate(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช้ใส่ N,Data(จำนวนบิตข้อมูล),BitStop(บิตสต๊อป)"

ตัวอย่าง MSComm1.Settings="1200,N,8,1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50330

3. **InputLen** คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัพเฟอร์

ตัวอย่าง MSComm1.InputLen=1

4. **PortOpen** คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False กัน

ตัวอย่าง MSComm1.PortOpen=True

5. **Rthreshold** คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล (Comport)มันทำให้เกิดCommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold =1

ลักษณะการใช้งานของคุณสมบัติต่าง ๆ ใน Visual Basic

```
Private Sub Form_Load()
    MSComm1.Settings="1200,N,8,1"
    MSComm1.CommPort=1
    MSComm1.InputLen=1
    MSComm1.PortOpen=True
    MSComm1.Rthreshold =1
End Sub
```

วิธีของการรับส่งข้อมูลจาก Serial Port

จากวิธีเขียนโค๊ดด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

Output =ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input =เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS232 นั่นเอง

การใช้ Even ใน OnComm() แสดงค่าต่างๆที่ใช้ใน MS Comm เราสามารถนำ Event ในตารางดังกล่าวมาใช้เขียนใน OnComm() โดยการใช้เงื่อนไขแบบ Select case ซึ่งสามารถเขียนได้ดังนี้

```
Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent
        Case comEvReceive
            Dim Buffer As Variant
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Buffer = MSComm1.Input
        ShowData txtRXTX, (StrConv((Buffer), vbUnicode))
    Case comEvSend : 'ส่วนนี้จะใส่เงื่อนไขให้ทำอะไรก็ได้ตามที่ได้ออกแบบไว้
    Case comEvCTS
    Case comEvDSR
    Case comEvCD
    Case comEvRing
    Case comEvEOF
    Case comBreak
    Case comCDTO
    Case comCTSTO
    Case comDCB
    Case comDSRTO
    Case comFrame
    Case comOverrun
    Case comRxOver
    Case comRxParity
    Case comTxFull
End Select
End Sub
```

3.4 การใช้ไมโครคอนโทรลเลอร์ติดต่อกับ Serial Port

รีจิสเตอร์ที่ใช้งานที่เกี่ยวข้อง

การใช้งานในการติดต่อสื่อสารแบบอนุกรมจะต้องนำรีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัพท์มาใช้ด้วยคือ TMOD และ TCON และในส่วนนี้มีรีจิสเตอร์ที่ใช้งานอีกคือ SCON และ SBUF

รีจิสเตอร์ SCON (Serial Port Control Register)

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการเลือกโหมดการทำงาน เก็บข้อมูลบิตที่ 9 (ปกติข้อมูลมี 8 บิต อยู่ในรีจิสเตอร์ SBUF ข้อมูลบิตที่ 9 อาจเป็น พาริตี้บิตก็ได้) มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	TR8	TI	RI

Register SCON (Serial Port Control Register)

SM0, SM1: (Serial port mode bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมมีอยู่ 4 โหมด

SM0	SM1	MODE	การทำงาน	อัตราการรับ-ส่ง
0	0	0	Shift register	Fosc/12
0	1	1	8 bit UART	Variable
1	0	2	9 bit UART	Fosc/32,fosc/64
1	1	3	9 bit UART	Variable

รูปแบบการแสดงการเลือกโหมดการทำงานของพอร์ตอนุกรม

SM2 : เป็นบิตที่ทำหน้าที่ควบคุมการทำงานในลักษณะการเชื่อมต่อไมโครคอนโทรลเลอร์หลายตัวเข้าด้วยกัน (Multi Processor Communication) สำหรับการใช้งานในโหมด 2 และ 3 เป็นดังนี้

SM2=1 จะทำให้แฟลกอินเตอร์รัพท์ทางด้านรับ (RI) ไม่ถูกเซตเมื่อรับข้อมูลเข้ามาแล้วเป็นบิตที่ 9 เป็น 0 สำหรับการทำงานในโหมด 1 ถ้าเซต SM2 แฟลกทางด้านรับจะไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มี Stop bit การใช้งานในโหมด 0 ต้องกำหนดให้ SM2=0

REN: (Enable Serial Reception) เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม

1 = ให้มีการรับข้อมูล

0 = ไม่ให้มีการรับข้อมูล

TB8: เป็นบิตของข้อมูลบิตที่ 9 ในการใช้ส่งข้อมูลใช้งานในโหมด 2 และ 3

RB8: เป็นบิตของข้อมูลบิตที่ 9 ในการใช้รับข้อมูลใช้งานในโหมด 2 และ 3 และหากใช้งานในโหมด 1 กำหนดให้ SM2=0 บิตนี้จะเป็นค่าของการ Stop bit ที่รับเข้ามาในโหมด 0 จะไม่ใช้บิตนี้

TI: (Transmit Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพท์การส่งข้อมูล จะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จเรียบร้อยลงในบิตที่ 8 และจะต้องเคลียร์บิตนี้ด้วยซอฟท์แวร์ทุกครั้งเมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ของการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RI: (Receive Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพท์การรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จเรียบร้อยลงในบิตที่ 8 และจะต้องเคลียร์บิตนี้ด้วยซอฟท์แวร์ทุกครั้งเมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ของการรับข้อมูล

รีจิสเตอร์ SBUF (Serial data buffer register) เป็นรีจิสเตอร์ขนาด 8 บิต หรือ 1 ไบต์มีแอดเดรสอยู่ที่ 99H ทำหน้าที่รับส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ การอ่านค่าก็จะอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะอ่านจากรีจิสเตอร์ SBUF ที่รับข้อมูลเข้ามาจากภายนอก ในทำนองเดียวกันการส่งข้อมูลก็จะนำค่าที่จะส่งออกไปไว้ที่ SBUF แล้วจึงทำการส่งออกไป

3.5 การเขียนโปรแกรมติดต่อกับ Serial Port

การเขียนโปรแกรมรับข้อมูล

```
JNB RI,$          ; เปรียบเทียบเงื่อนไขจนกว่าเป็นจริงเมื่อ Buffer Receive เรียบร้อยแล้ว
CLR RI            ; เคลียร์ค่าของ RI(Receive Interrupt) bit ใน SCON Register
MOV A,SBUF        ; เก็บค่าจาก Buffer ไปเก็บไว้ใน Register A
```

การเขียนโปรแกรมส่งข้อมูล

```
MOV SBUF,#30H     ; เอาค่าแอสกี้=30H ไปเก็บไว้ที่ Buffer ในการส่งข้อมูลผ่าน Serial Port
JNB TI,$          ; เปรียบเทียบเงื่อนไขเป็นจริงเมื่อ Buffer Send เรียบร้อยแล้ว
CLR TI            ; เคลียร์ค่าของ TI(Transmit Interrupt) bit ใน SCON Register
```

TCON (Time Control Register) และ TMOD (Time Mode Register)

เรานำเอาทั้งสอง มาทำหน้าที่สำหรับควบคุมการทำงานของTimerเพื่อจะนำมาใช้เป็นตัววนลูปจับเวลาของ อัตราการรับส่งข้อมูล(Baud Rate) คือ จะกำหนดค่าใน TMOD เช่น TMOD=00100000B ในที่นี้เราใช้รับส่งข้อมูล 11 bit คือจะมี 8 data bit, Start bit, Stop bit, Custom data bit ทั้งนี้เราจะกำหนดไม่ให้ Timer เกิด Interrupt โดยเราจะไปกำหนดที่ IE (Interrupt Enable Register)ตั้งค่าให้เป็นศูนย์ทุก bit ส่วน Baud rate ของการรับส่งข้อมูลนั้นจะอาศัยอัตราการเกิด Timer Overflow ร่วมกับค่าของ SMOD (Serial Mode Register) สำหรับค่าที่จะกำหนดให้ Timer เพื่อสร้างอัตรารับส่งของ Serial Port ในไมโครคอนโทรลเลอร์ สามารถใช้จากตารางด้านล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางกำหนดค่าให้ Timer เพื่อใช้เป็น Baud Rate

ความถี่(MHz)	อัตรารับส่ง Serial Port(bit/sec)	ค่าของ Timer ที่จะกำหนด
11.059	1200	E8H
11.059	2400	F4H
11.059	4800	FAH
11.059	9600	FDH

การเขียนโปรแกรมตั้งค่าทำงานของ Timer

อาจสรุปการเขียนโปรแกรมตั้งค่าของ Timer เช่น Baud Rate ของ Serial port=9600 bit/sec

```
MOV   TMOD,#00100000B      ; ตั้งค่าใน TMOD Register ให้เริ่มค่าใหม่อัตโนมัติ
MOV   IE,#00000000B        ;ตั้งค่าไม่ให้ Timer Interrupt
MOV   TL1,#FDH             ;ตั้งค่าอัตรารับส่งข้อมูล 9600 bit/sec
MOV   TH1,#FDH
SETB  TR1                  ; สั่ง Timer 1 ทำงาน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

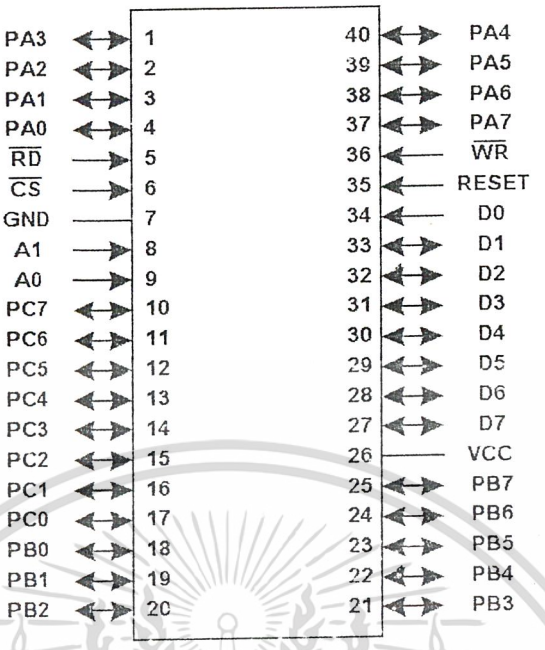
บทที่ 4
ทฤษฎีและการใช้งาน 8255 เบื้องต้น

ไมโครโปรเซสเซอร์นั้น นอกจากติดต่อกับหน่วยความจำ โดยการนำข้อมูลไปเก็บไว้ หรืออ่านข้อมูลใดๆ ออกจากหน่วยความจำแล้ว ตัว CPU เอง อาจจะต้องติดต่อกับส่วนประกอบภายนอกอื่นๆ อีกด้วย เช่น การรับคีย์ การแสดงผล หรือแม้แต่การนำเอา CPU ไปควบคุมอุปกรณ์ต่างๆนั้น CPU ต้องติดต่อโดยผ่านทางอินพุท หรือทางเอ้าท์พุทพอร์ต ซึ่งอาจสามารถใช้ไอซี ทีทีแอล บางเบอร์มาใช้เป็นพอร์ตสำหรับ CPU ได้ แต่ทั้งนี้การใช้ไอซีทีทีแอลมีข้อจำกัดหลายอย่างเช่น ในกรณีจำเป็นต้องใช้พอร์ตหลายๆพอร์ต เพราะต้องติดต่อกับอุปกรณ์ภายนอกหลายชุด จึงต้องใช้ไอซีเหล่านี้จำนวนหลายตัว และอาจทำให้ยากแก่การออกแบบวงจร อีกทั้งไม่สามารถจะเปลี่ยนแปลงลักษณะการทำงานให้แตกต่างไปจากเดิมโดยการโปรแกรมได้ ดังนั้นผู้ผลิต CPU ตระกูลต่างๆ จึงมักผลิตไอซีประเภท LSI ที่ทำหน้าที่เป็นพอร์ต เพื่อให้สามารถใช้ CPU นั้นๆ ได้สะดวก ซึ่งจะทำให้การรับส่งข้อมูลมีความน่าเชื่อถือและยังสามารถเปลี่ยนแปลงชนิดพอร์ต จากอินพุท เป็นเอ้าท์พุท เป็นต้นได้โดยง่าย โดยการควบคุมของ CPU เอง ในบทนี้จะกล่าวถึงไอซีที่ทำหน้าที่เป็น อินพุทและเอ้าท์พุทพอร์ต ซึ่งนิยมในการนำไปใช้งานมากที่สุด อีกทั้งมีราคาถูกหาซื้อได้ง่าย คือไอซีเบอร์ 8255 ของบริษัทอินเทล ที่จริงแล้วไอซีเบอร์ 8255 นี้จะถูกออกแบบและผลิตเพื่อใช้งานร่วมกับ CPU เบอร์ 8080 แต่ก็สามารถนำมาใช้งานกับ 8031 หรือเบอร์อื่นๆได้ โดยในบทนี้จะกล่าวถึงคุณสมบัติการทำงานรวมทั้งการนำ 8255 ไปใช้งานรวมกับ 8031 ในเบื้องต้นด้วย

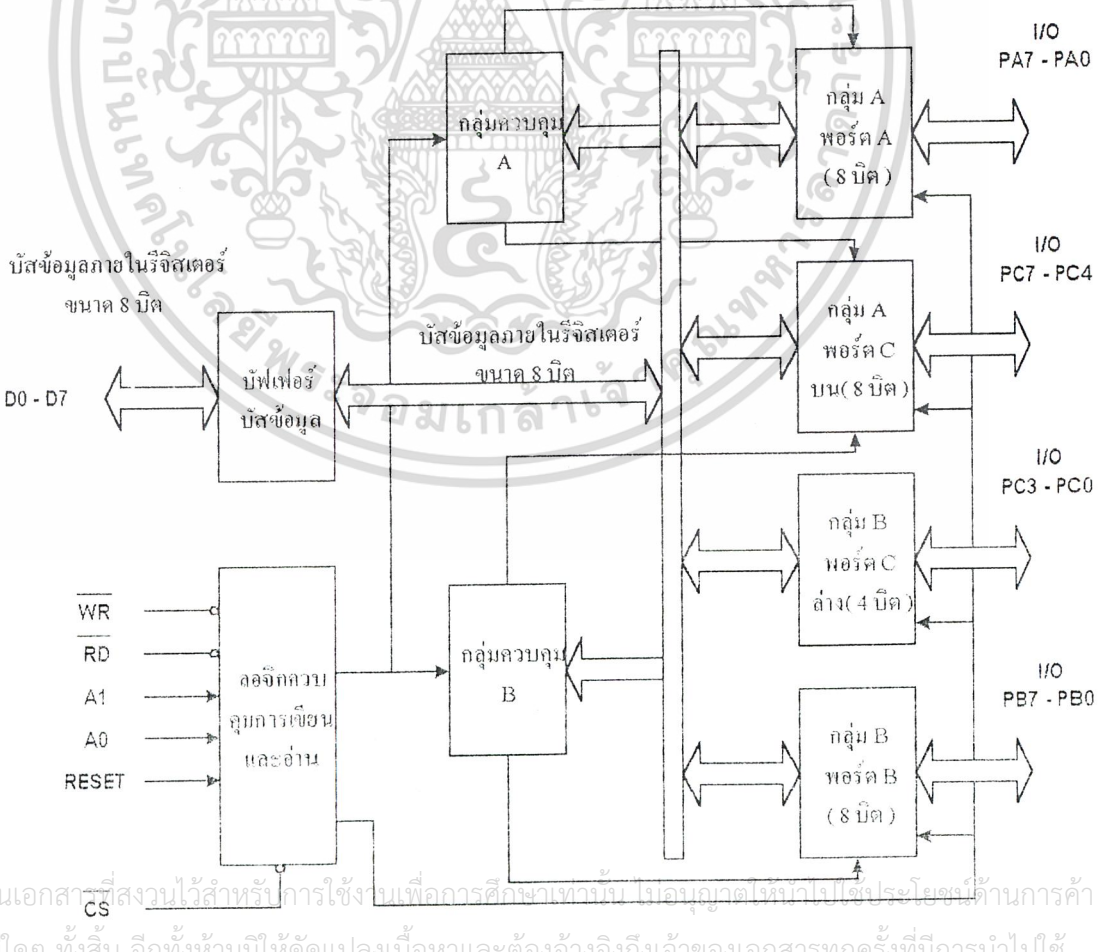
4.1 ลักษณะเบื้องต้น

ไอซี 8255 เป็นไอซี LSI ขนาด 40 ขา จากรูปที่ 4.1a แสดงตำแหน่งของขาต่างๆทั้ง 40 ขา ส่วนรูปที่ 4.1b แสดงแผนผังภายในของไอซี 8255 ซึ่งมีพอร์ตสำหรับส่งข้อมูลอยู่ด้วยกัน 3 พอร์ต มีชื่อดังนี้ A , B และ C โดยพอร์ต C จะแบ่งออกเป็น 2 ส่วน คือพอร์ต C ล่าง (CLO) กับ C บน (CHI) นอกจากนี้แล้ว ยังมีพอร์ตควบคุมอีกหนึ่งพอร์ต (Control Port) พอร์ตนี้จะใช้งานก็ต่อเมื่อ CPU ต้องการกำหนดลักษณะการทำงานของพอร์ต A , B และ C หรือต้องการเปลี่ยนแปลงหลังจากกำหนดไว้เดิม CPU จะส่งรหัสควบคุมมาทาง ดาต้าบัส (Data Bus) ให้แก่พอร์ตควบคุมนี้ ซึ่งการกำหนดรหัสที่ใช้ในการควบคุมพอร์ตต่างๆนี้ จะกล่าวในตอนต่อไป ในทางปฏิบัติผู้ออกแบบระบบต้องนำรหัสควบคุมที่ได้มาตามข้อกำหนดของ 8255 นี้ใส่ในโปรแกรมเพื่อให้ CPU ทำการส่งรหัสควบคุมนี้มายังพอร์ตควบคุม เมื่อระบบนั้นเริ่มต้นทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงตำแหน่งขาต่างๆของ 8255



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แผนผังภายในของ 8255

4.2 หน้าที่ของขาต่างๆ

ก่อนที่จะกล่าวถึงการนำ 8255 ไปใช้งาน ควรทราบถึงหน้าที่ของขาต่างๆ ของ 8255 ทั้ง 40 ขาเสียก่อน จะทำให้เข้าใจถึงวิธีการใช้งานได้ดียิ่งขึ้น ขาต่างๆของ 8255 สามารถแบ่งออกได้ดังนี้

- CS (Chip Select) ขานี้ใช้สำหรับสัญญาณจากภายนอกเพื่อใช้ในการเลือกว่าจะให้ 8255 ตัวนี้ทำงานหรือไม่ โดยที่ถ้าขานี้ได้รับลอจิก "0" จะทำให้ 8255 เชื่อมต่อเข้ากับระบบบัสต่างๆ ของ
- CPU และพร้อมที่จะติดต่อกับ CPU ได้ แต่ถ้าเป็นลอจิก "1" มันก็จะปลดตัวเองออกจากระบบบัสต่างๆ ของ CPU (High Impedance)
- $\overline{RD}$ (Read Enable) เป็นขาอินพุทที่จะรับสัญญาณจาก CPU ถ้าขานี้ได้รับลอจิก "0" และขณะนั้น ขา CS ต้องเป็น "0" ด้วย 8255 จะรับข้อมูลผ่านพอร์ตที่ CPU กำหนดไว้ โดยผ่านจากดาต้าบัสของ CPU
- $\overline{WR}$ (Write Enable) มีหน้าที่การทำงานตรงข้ามกับขา $\overline{RD}$ คือถ้าขา $\overline{WR}$ นี้ได้รับลอจิกเป็น "0" (โดยที่ขา CS ต้องเป็น "0" ด้วย) 8255 จะรับข้อมูลจากดาต้าบัสของ CPU ส่งออกไปยังพอร์ตที่ CPU กำหนดไว้
- RESET คือขาที่ทำหน้าที่ Reset 8255 เมื่อ 8255 ได้รับสัญญาณ Reset กลับเข้าสู่โหมด อินพุท คือทุกพอร์ตจะเป็น อินพุทพอร์ต ซึ่งขานี้ใช้เมื่อต้องการเคลียร์สถานะต่างๆของ 8255
- $D_0 - D_7$ คือขาข้อมูลที่ใช้ในการติดต่อรับส่งข้อมูลกับ CPU โดยขา $D_0 - D_7$ นี้จะต่อเข้ากับ ดาต้าบัสของ CPU เพื่อให้ CPU ส่งข้อมูลออกไปยังพอร์ต หรือรับข้อมูลจากพอร์ตส่งให้แก่ CPU ผ่านทาง $D_0 - D_7$ นี้
- $A_0 - A_1$ คือขาแอดเดรสที่ใช้ในการเลือกพอร์ตที่ CPU ต้องการติดต่อด้วย ซึ่งมีความเป็นไปได้ทั้งหมด 4 ค่าดังนี้ คือ

  00 = พอร์ต A

  01 = พอร์ต B

  10 = พอร์ต C

  11 = พอร์ตควบคุม

- $PA_0 - PA_7$ เป็นขาสัญญาณของพอร์ต A ใน 8255 ซึ่งถูกเลือกโดยค่าของ $A_0 - A_1$ และเมื่อพอร์ตนี้ถูกเลือกใช้ ข้อมูลต่างๆจะถูกส่งผ่าน $PA_0 - PA_7$ นี้ ไปยัง $D_0 - D_7$ (กรณีที่ให้พอร์ต A เป็นพอร์ต อินพุท) หรือจาก $D_0 - D_7$ มายัง $PA_0 - PA_7$ (กรณีพอร์ต A เป็นพอร์ตเอ้าท์พุท)
- $PB_0 - PB_7$ เป็นขาสัญญาณของพอร์ต B ใน 8255 ซึ่งถูกเลือกโดยค่าของ $A_0 - A_1$ เช่นกันกับพอร์ต A และ พอร์ต B นี้จะมีข้อจำกัดในการรับส่งข้อมูลที่ต่างจากพอร์ต A ในบางกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $PC_0 - PC_7$ เป็นขาสัญญาณของพอร์ต C ซึ่งแบ่งออกเป็น สองกลุ่มคือ $PC_0 - PC_3$ และ $PC_4 - PC_7$ โดยแต่ละกลุ่มสามารถแยกกันทำงานได้โดยอิสระ กลุ่มหนึ่งอาจเป็นเอ้าท์พุทพอร์ต ในขณะที่อีกกลุ่มหนึ่งเป็น อินพุทพอร์ตได้ แต่จะทำงานพร้อมกันโดยการเลือกด้วยลอจิกที่ $A_0 - A_1$

4.3 การใช้งาน 8255

8255 นั้นแบ่งการทำงานออกเป็น 3 โหมดด้วยกันคือ

โหมด0 เป็นโหมดอินพุท หรือเอ้าท์พุทอย่างใดอย่างหนึ่ง ซึ่งทั้งสามพอร์ต สามารถทำงานในโหมดนี้ได้

โหมด1 เป็นโหมดอินพุท หรือเอ้าท์พุทอย่างใดอย่างหนึ่งเช่นกัน แต่จะมีลักษณะการทำงานเป็นลักษณะของ Handshaking ซึ่งจะกล่าวรายละเอียดในภายหลัง ในโหมดนี้ทำงานใช้เฉพาะพอร์ต A และ B เท่านั้น

โหมด 2 เป็นโหมด Bi – directional คือเป็นทั้งอินพุท และเอ้าท์พุทพอร์ตในเวลาเดียวกัน และทำงานแบบ Handshaking เช่นเดียวกับโหมด 1 ในโหมดนี้ใช้ได้เฉพาะพอร์ต A เท่านั้น

การกำหนดโหมดการทำงานของ 8255 นั้น ทำได้โดย CPU ทำการส่งรหัสควบคุมผ่านทางดาต้าบัส มายังพอร์ตควบคุม (Control Port) ของ 8255 รหัสควบคุมจะมีขนาด หนึ่งไบท์ เรียกว่า Control byte และในแต่ละบิทของ Control byte นั้นจะมีความหมายเฉพาะของตัวมันเอง ดังแสดงในรูปที่ 4.3 ซึ่งสามารถอธิบายได้ดังนี้

บิท D_7 เป็นบิทที่แสดงว่า Byte นี้ เป็นรหัสควบคุม ที่จะมีผลต่อการกำหนดโหมดการทำงานของ 8255

บิท D_6 และ D_5 มีความหมายในการเลือกโหมดของพอร์ต A ซึ่งสามารถทำงานได้ทั้ง 3 โหมด โดยลอจิก D_6 และ D_5 จะมีความหมายดังนี้

00 = โหมด 0

01 = โหมด 1

10 = โหมด 2

11 = โหมด2

บิท D_4 ถ้าเป็นลอจิก "0" หมายถึงสั่งให้พอร์ต A ทำหน้าที่เป็นเอาท์พุทพอร์ต แต่ถ้าเป็นลอจิก "1" พอร์ต A จะเป็นอินพุทพอร์ต บิทนี้จะมีความหมายในกรณีที่เราให้ 8255 ทำงานในโหมด 0 หรือ 1 เท่านั้น เพราะในโหมดที่ 2 พอร์ต A จะเป็นอินพุทพอร์ต และ เอ้าท์พุทพอร์ตในเวลาเดียวกัน

บิท D_3 เป็นบิทที่กำหนดการทำงานของพอร์ต C บน ถ้าบิทนี้เป็นลอจิก "0" พอร์ต C บนนี้จะเป็นเอ้าท์พุทพอร์ต ถ้าเป็น "1" จะเป็น อินพุตพอร์ต

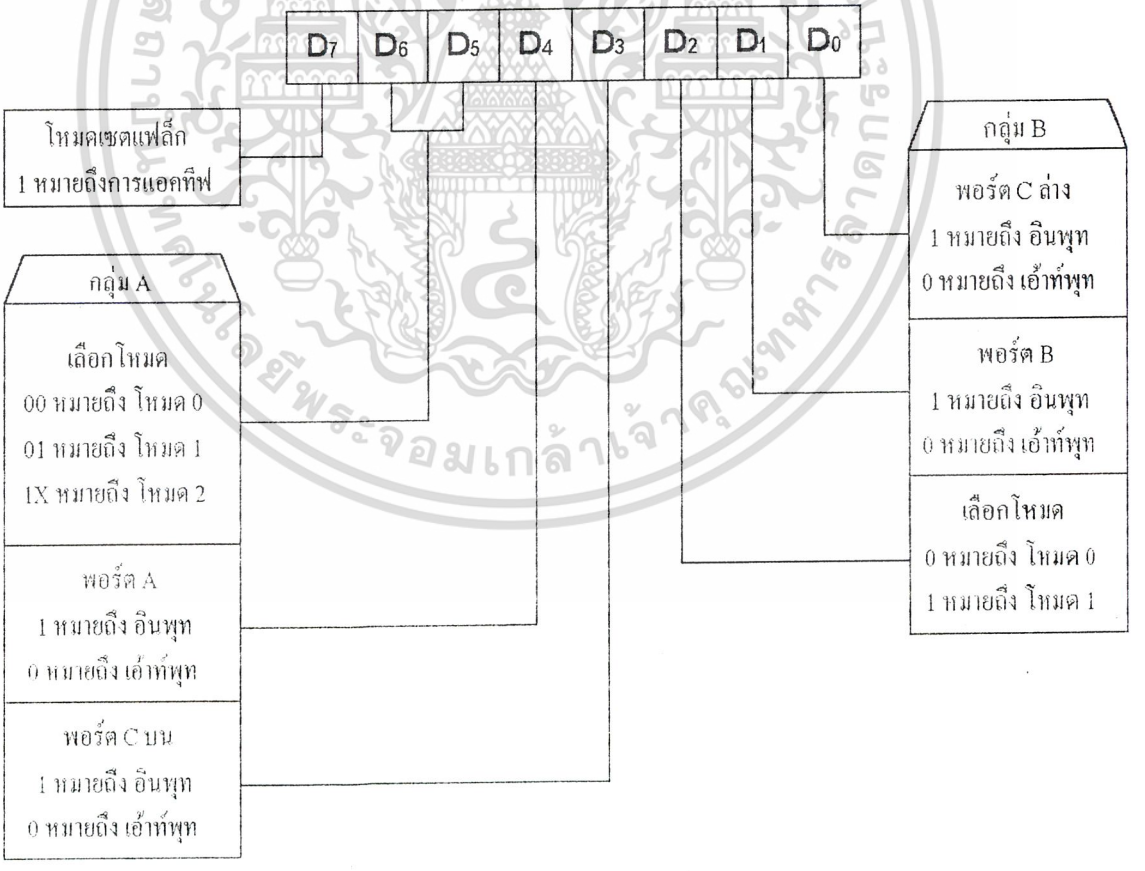
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิท D_2 เป็นบิทที่ใช้สำหรับการกำหนดโหมดการทำงานของพอร์ต B ถ้าเป็นลอจิก “0” จะหมายถึงให้พอร์ต B ทำงานในโหมด 0 ถ้าเป็นลอจิก “1” จะทำงานในโหมด 1

บิท D_1 เป็นการกำหนดให้พอร์ต B เป็นพอร์ต อินพุท หรือเอ้าท์พุท ถ้า D_1 เป็น “0” จะเป็นเอ้าท์พุทพอร์ต แต่ถ้าเป็นลอจิก “1” จะเป็นอินพุทพอร์ต

D_0 เป็นบิทที่กำหนดการทำงานของพอร์ต C ล่าง ถ้าบิตนี้เป็นลอจิก “0” พอร์ต C ล่างนี้จะเป็นเอ้าท์พุตพอร์ต ถ้าเป็น “1” จะเป็นอินพุทพอร์ต

ตัวอย่างการกำหนดรหัสควบคุมที่จะส่งให้แก่พอร์ทควบคุมของ 8255 หรือที่เรียกกันว่าการโปรแกรม 8255 เช่น ถ้าเราต้องการให้พอร์ต A และ พอร์ท C บน เป็นอินพุทพอร์ท ส่วนพอร์ท B กับ พอร์ต C ล่าง เป็นเอ้าท์พุตพอร์ต และทุกพอร์ต ทำงานในโหมด 0 จะสามารถกำหนดรหัสควบคุมได้ดังรูปที่ 4.3 จากรูป จะได้รหัสควบคุมหรือคอนโทรลไบท์นี้เป็น 10011000 หรือ 98H ในระบบเลขฐานสิบหก

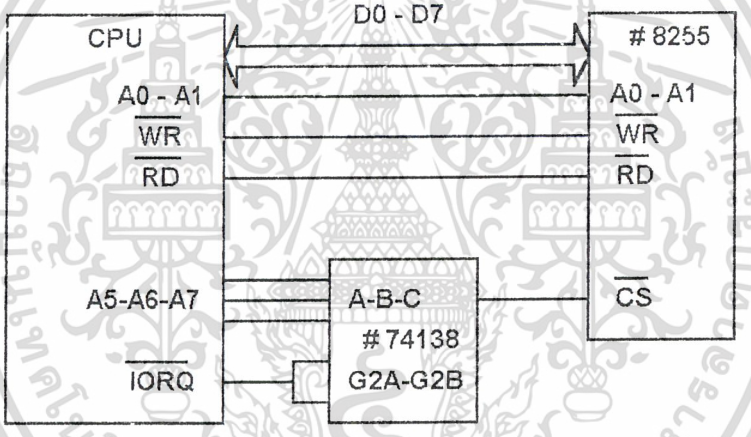


รูปที่ 4.3 แสดงความหมายแต่ละบิต ของรหัสควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การต่อ 8255 เข้ากับ CPU

จากลักษณะการทำงานของขาต่างๆของ 8255 ที่กล่าวมาแล้วนั้นจะเห็นว่าเราสามารถนำมาใช้กับคอนโทรลเลอร์ 8031 ได้ไม่ยาก ซึ่งการต่อ 8255 เข้ากับ 8031 นั้นขึ้นอยู่กับผู้ออกแบบว่าต้องการกำหนดให้พอร์ตทั้งสี่ ของ 8255 เป็นพอร์ตหมายเลขใด โดยการใช้ขาแอดเดรสของ 8031 เองเป็นตัวเลือกพอร์ตของ 8255 จากรูปที่ 4.4 แสดงตัวอย่างของการต่อขาแอดเดรสของ 8031 มาใช้ในการเลือกพอร์ตโดยประกอบกับ Decoder เบอร์ 74138 ซึ่งโดยปกติ คอนโทรลเลอร์ทั่วไปก็มักจะใช้ Decoder ช่วยทำหน้าที่ในการเลือกพอร์ต



รูปที่ 4.4 แสดงการต่อ 8255 เข้ากับ CPU

จากการต่อสายแอดเดรสของ 8031 เข้ากับ 8255 และ Decoder ดังรูปที่ 4.4 นี้จะทำให้ได้หมายเลขพอร์ตของ 8255 สำหรับ CPU ทำการเลือกได้ดังนี้

พอร์ตหมายเลข 00H = พอร์ต A

พอร์ตหมายเลข 01H = พอร์ต B

พอร์ตหมายเลข 10H = พอร์ต C

พอร์ตหมายเลข 11H = พอร์ตควบคุม

สังเกตว่าที่ขา CS ของ 8255 จะได้ลอจิก "0" เมื่อขา A_5 A_6 และ A_7 มีลอจิกเป็น "0" ทั้งหมด (จากคุณสมบัติของ 74138) เป็นการเปิดให้ใช้ 8255 ตัวนี้ แต่ขา A_0 A_1 ของ CPU นั้นต่อเข้าโดยตรงกับขา A_0 A_1 ของ 8255 ส่วน A_2 A_3 และ A_4 นั้นไม่มีผลในการกำหนดพอร์ตของ 8255 ตัวนี้ ถ้าเราให้ขาทั้งสามที่เหลือนี้ มีลอจิก "0" ทั้งหมดก็จะหมายความว่าได้หมายเลขพอร์ตคือ ตัวอย่างเช่นถ้าเราต้องการเลือกพอร์ต B ของ 8255 ตัวนี้ ขา A_0 A_1 ต้องมีลอจิกเป็น "10" ตามลำดับ และขา A_5 A_6 A_7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องเป็น “0” ด้วย (เพื่อให้ขา CS เป็น “0”) ส่วนขา A_2 A_3 และ A_4 ก็ให้เป็น “0” เมื่อรวมลอจิก A_0 – A_7 จะได้ลอจิกดังนี้คือ “00000001” หรือ 01H ในฐานสิบหก ซึ่งเป็นหมายเลขของพอร์ต B ดังที่กล่าวไปแล้ว ที่จริงแล้วการกำหนดเลขพอร์ตในลักษณะที่ไม่ได้ใช้ A_2 A_3 และ A_4 นี้ ถ้าให้บิทใดบิทหนึ่งในสามบิทนี้หรือทั้งหมดมีลอจิกเป็น “1” ก็ยังสามารถเลือกพอร์ตของ 8255 นี้ได้ แต่หมายเลขพอร์ตจะเปลี่ยนไป เช่นถ้าให้ A_3 เป็น “1” ถ้าต้องการเลือกพอร์ต B จะได้ลอจิกเป็น “00001001” หรือ 09H ซึ่งเป็นชื่อของพอร์ต B สรุปได้ว่าลักษณะการกำหนดหมายเลขพอร์ตโดยลอจิกที่ A_2 A_3 และ A_4 ไม่มีความสำคัญในการเลือกนั้นจะทำให้พอร์ตแต่ละพอร์ตของ 8255 สามารถเลือกได้หลายหมายเลข เช่นพอร์ต B อาจเลือกได้โดยใช้หมายเลขพอร์ต 01H หรือ 09H ก็ได้เป็นต้น แต่โดยปกติเรานิยมให้ขาแอดเดรสที่ไม่ได้ใช้มีลอจิกเป็น “0” ในที่นี้เราจึงเรียกพอร์ต A,B,C และพอร์ตควบคุมด้วยหมายเลขพอร์ต 00H – 03H เท่านั้น

นอกจากการต่อ A_0 – A_7 เพื่อใช้ในการเรียกหมายเลขพอร์ตแล้ว สัญญาณควบคุมที่จำเป็นสำหรับ 8255 อีกชุดหนึ่งคือ สัญญาณควบคุมการเขียนและอ่าน ($\overline{WR}$, $\overline{RD}$ ซึ่งมีความหมายถึงการส่ง หรือรับ) จากรูปที่ 4.4 เช่นเราต่อขา $\overline{WR}$ และ $\overline{RD}$ ของ 8255 เข้ากับขา $\overline{WR}$, $\overline{RD}$ ของ 8031 โดยตรงได้เนื่องจากเป็นขาแอคทีฟ “0” เช่นเดียวกัน ส่วน 74138 เราเรียกว่าพอร์ตโคดเดอร์ ทำหน้าที่เป็นตัวเลือกพอร์ตหลายชุด กรณีที่ต่อ 8255 หลายตัว

CS	$\overline{RD}$	$\overline{WR}$	A1	A0	การทำงาน
0	1	0	0	0	ส่งข้อมูลไปที่พอร์ต A
0	0	1	0	0	รับข้อมูลจากพอร์ต A
0	1	0	0	1	ส่งข้อมูลไปที่พอร์ต B
0	0	1	0	1	รับข้อมูลจากพอร์ต B
0	1	0	1	0	ส่งข้อมูลไปที่พอร์ต C
0	0	1	1	0	รับข้อมูลจากพอร์ต C
0	1	0	1	1	ส่งรหัสควบคุมไปที่พอร์ตควบคุม
0	0	1	1	1	ไม่มีความหมาย
1	X	X	X	X	8255 ไม่ทำงาน ปลดสายอินพุทออกจากระบบ

ตารางที่ 4.1 แสดงผลของสัญญาณควบคุมต่างๆที่มีผลต่อ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 โหมดการทำงานของ 8255

4.5.1 โหมด 0

ในโหมดนี้เป็นการกำหนดให้พอร์ตทุกพอร์ตของ 8255 เป็นอินพุท เอ้าท์พุทพอร์ตแบบพื้นฐาน หรือที่เรียกว่า Simple I/O Port ซึ่งนิยมใช้กันมาก เมื่อนำมาเป็นพอร์ตของ CPU ต่างๆ ซึ่งพอร์ตทั้งสามของ 8255 สามารถที่จะเป็นไปได้ทั้ง อินพุท หรือ เอ้าท์พุท ตามแต่ความต้องการใช้งาน

4.5.2 โหมด 1

เป็นการรับส่งข้อมูลแบบ Handshaking ความหมายของ Handshaking ก็คือ ระหว่าง CPU พอร์ต และอุปกรณ์ภายนอก ขณะที่รับส่งข้อมูลกันนั้น นอกจากรับส่งข้อมูลกันแล้ว ยังต้องมีการตอบรับในการรับส่งข้อมูลแต่ละครั้ง โดยที่ผู้รับกับผู้ส่งนั้นจะต้องทำงานสัมพันธ์กันตลอดเวลา ซึ่งเป็นประโยชน์ในกรณีที่อุปกรณ์ภายนอกมีการรับส่งข้อมูลที่ช้ากว่า CPU ทำให้ไม่สามารถทำงานได้ทัน CPU จึงต้องใช้วิธีรับส่งข้อมูลแบบ Handshaking โดยอุปกรณ์ภายนอกเป็นตัวกำหนดจังหวะในการรับส่งข้อมูลเอง เช่นการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ ซึ่งเครื่องพิมพ์ทำงานช้ากว่าคอมพิวเตอร์มาก เมื่อคอมพิวเตอร์ส่งตัวอักษรแรกให้กับเครื่องพิมพ์ เครื่องพิมพ์ก็จะทำการประมวลผลต่างๆ และเมื่อพร้อมที่จะพิมพ์ตัวอักษรตัวต่อไป ก็จะส่งสัญญาณไปบอกคอมพิวเตอร์ให้ส่งตัวอักษรตัวต่อไปมาได้ การติดต่อระหว่างเครื่องพิมพ์กับคอมพิวเตอร์จึงเป็นไปโดยไม่มีความผิดพลาด ส่วนสัญญาณที่ใช้ในการควบคุมการรับส่งนี้ จะมาจากพอร์ต C เพราะในโหมดนี้พอร์ตที่ใช้ในการรับส่งข้อมูลได้ คือพอร์ต A และ B เท่านั้น ส่วนพอร์ต C จะเป็นตัวส่งสัญญาณควบคุมกับอุปกรณ์ภายนอก และสัญญาณควบคุมในแต่ละบิทของพอร์ต C จะเป็นดังตารางที่ 4.2

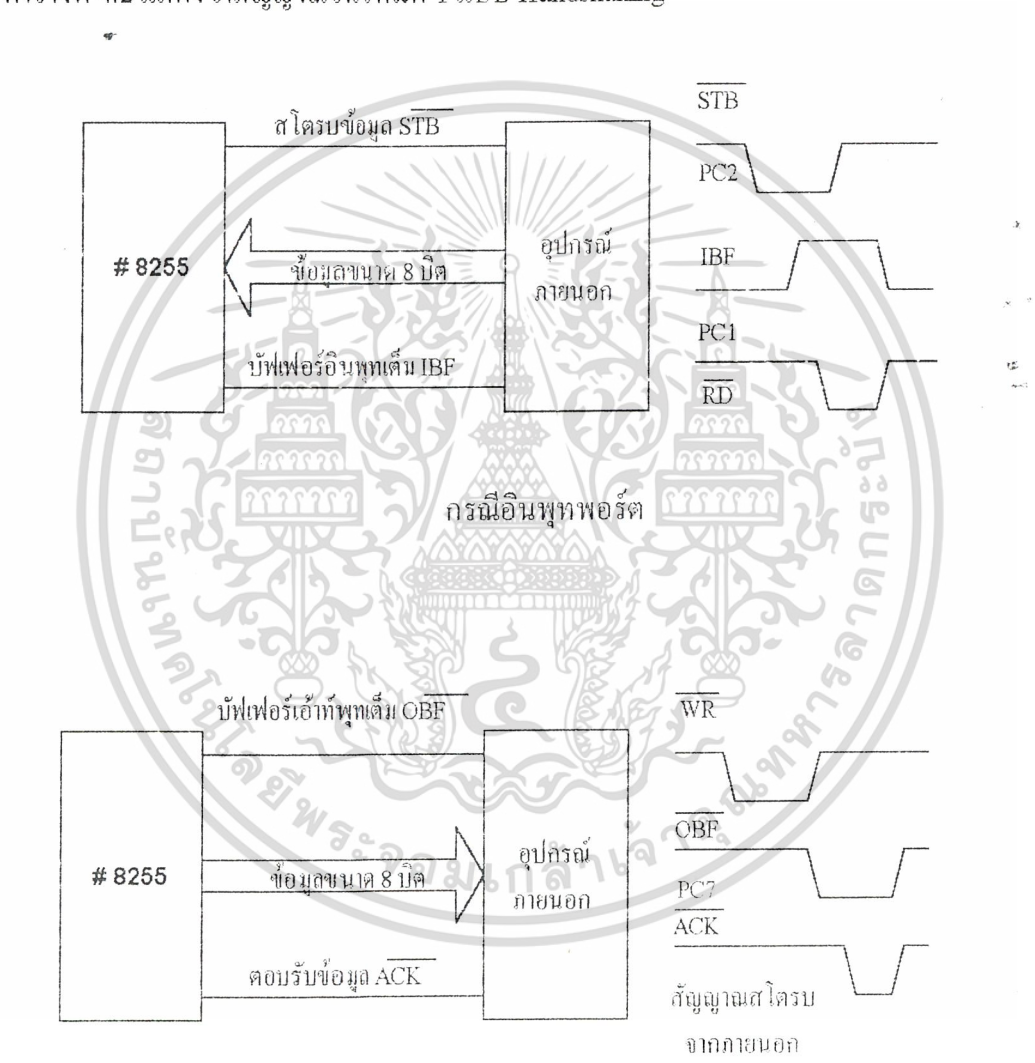
ส่วนกรณีที่เป็นเอ้าท์พอร์ตนั้น เมื่อ CPU ส่งข้อมูลออกมายังพอร์ทของ 8255 ก็จะรับข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายใน แล้วส่งสัญญาณ OBF (Output Buffer Full) บอกไปยังอุปกรณ์ภายนอกว่ามีข้อมูลมารอที่บัฟเฟอร์แล้ว (รีจิสเตอร์) ของพอร์ตแล้ว ให้มารับไปได้ อุปกรณ์ภายนอกก็จะส่งสัญญาณตอบรับ ACK (Acknowledge) มายังพอร์ตเพื่อรอรับข้อมูล หลังจากนั้นสัญญาณ OBF ก็จะกลายเป็น "1" เพื่อรอให้ CPU ส่งข้อมูลใหม่ต่อไป

ตำแหน่งขา	กรณีอินพุท	กรณีเอ้าท์พุท
PC0	$INTR_B$	$INTR_B$
PC1	IBF_B	$\overline{OBF_B}$
PC2	$\overline{STB_B}$	$\overline{ACK_B}$
PC3	$INTR_A$	$INTR_A$
PC4	$\overline{STB_A}$	I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC5	IBF_B	I/O
PC6	I/O	$\overline{ACK_A}$
PC7	I/O	$\overline{OBF_A}$

ตารางที่ 4.2 แสดงขาสัญญาณในโหมด 1 แบบ Handshaking



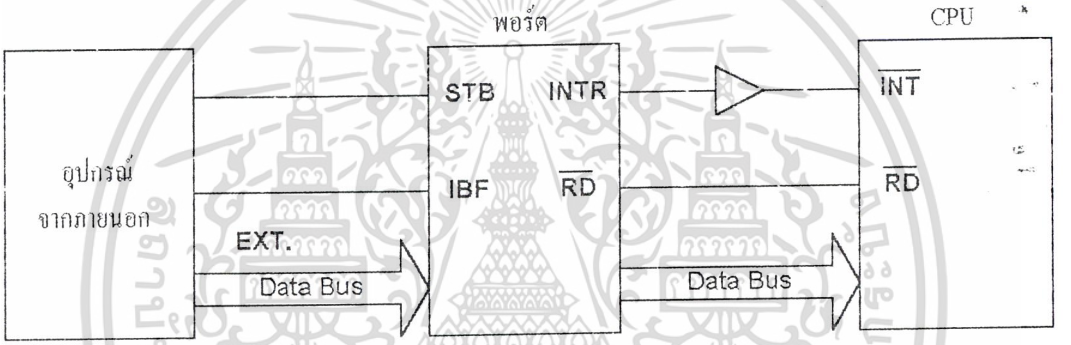
รูปที่ 4.5 การจัดสัญญาณในแบบ Handshaking

ในทางปฏิบัติแล้วการที่ CPU จะทราบได้อย่างไรว่าเมื่อไรที่อุปกรณ์ภายนอกได้ส่งข้อมูลเข้ามาที่บัฟเฟอร์ของพอร์ตแล้ว ในกรณีที่เป็นอินพุทพอร์ต หรือเมื่อไรที่อุปกรณ์ภายนอกได้รับข้อมูลจากพอร์ตไปแล้ว พร้อมที่จะให้ CPU ส่งข้อมูลต่อไปได้ กรณีที่เป็นเอ้าท์พุทนั้น มีวิธีจะให้ CPU ทราบได้สองลักษณะคือ ให้CPU ทำการ Polling แต่วิธีนี้ CPU จะต้องเสียเวลาตรวจสอบตลอดเวลา

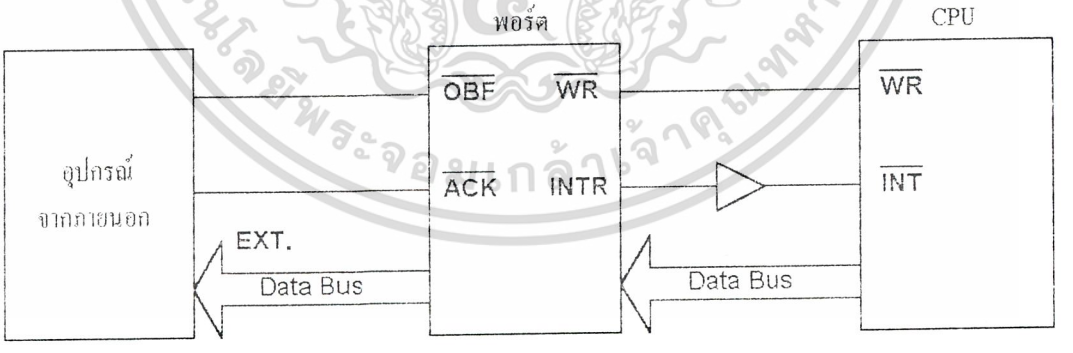
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทดังกล่าวอยู่ตลอดเวลา ซึ่งจะทำให้เสียเวลาของ CPU ไปโดยเปล่าประโยชน์ จึงควรเลือกใช้วิธีนี้ต่อเมื่อพิจารณาแล้วว่าการเสียเวลาของ CPU นั้น ไม่มีผลเสียต่อระบบ

อีกวิธีหนึ่งคือการให้อุปกรณ์ภายนอกนั้น ทำการขออินเตอร์รัพท์ จากที่แสดงในตารางที่ 2 นั้นจะเห็นว่า บิทที่ 0 และบิทที่ 3 ของพอร์ต C นั้นเป็นสัญญาณอินเตอร์รัพท์ (INTR) ของพอร์ต A และ B ตามลำดับ ซึ่งสามารถใช้สัญญาณอินเตอร์รัพท์นี้ให้เป็นประโยชน์ได้ ในการต่อใช้งานสัญญาณอินเตอร์รัพท์ไปที่ขา INT ของ Z-80 นี้จะต้องผ่าน NOT Gate (Inverter) เสียก่อน เนื่องจากการของอินเตอร์รัพท์ของ Z-80 นั้น แอคทีฟเป็น "0" การรับส่งข้อมูลในแบบนี้แสดงในรูปที่ 6 และรูปที่ 7



รูปที่ 4.6 การของอินเตอร์รัพท์ กรณีเป็นอินพุทพอร์ต



รูปที่ 4.7 การของอินเตอร์รัพท์ กรณีเป็นเอ้าท์พุทพอร์ต

ในรูปที่ 4.6 กรณีที่เป็นอินพุทพอร์ต เมื่ออุปกรณ์ภายนอกส่งข้อมูลให้แก่พอร์ต พร้อมทั้งส่งสัญญาณ STB มาให้พอร์ตรับข้อมูลนี้ไปเก็บในรีจิสเตอร์ เมื่อพอร์ตรับข้อมูลนี้ไปเก็บแล้วจะส่งสัญญาณ IBF ตอบรับไปยังอุปกรณ์ภายนอก เพื่อให้หยุดส่งข้อมูลชั่วคราว แล้วจึงทำการของอินเตอร์รัพท์ โดยการส่งสัญญาณ INTR ผ่าน Inverter ไปที่ขา INT ของ CPU เพื่อขอให้ CPU มารับข้อมูลนี้ไป หลังจาก CPU รับข้อมูลนี้ไปแล้ว (8255 ได้รับสัญญาณ *RD*) พอร์ตก็จะตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์รัพท์ ทำให้สัญญาณ IBF กลับเป็น “0” เป็นการบอกให้อุปกรณ์ภายนอกได้ทราบว่า พร้อมที่จะรับข้อมูลไบท์ต่อไปได้แล้ว

ส่วนในรูปที่ 4.7 เป็นกรณีของเอ้าท์พุทพอร์ตเมื่อ CPU ต้องการส่งข้อมูลให้อุปกรณ์ภายนอกโดยการส่งสัญญาณ $\overline{WR}$ แก่ 8255 ให้รับข้อมูลไปไว้ที่ รีจิสเตอร์บัฟเฟอร์ โดยพอร์ตจะรับข้อมูลไว้แล้วส่งสัญญาณ OBF ไปบอกแก่อุปกรณ์ภายนอกให้มารับข้อมูลไป และหลังจากที่อุปกรณ์ภายนอกมารับข้อมูลไปแล้วก็จะตอบรับโดยการส่งสัญญาณ ACK แก่ 8255 ทำให้สัญญาณ OBF กลับเป็น “1” พร้อมกับทำให้สัญญาณอินเตอร์รัพท์ เป็น “1” ด้วย ซึ่งจะเป็นการของอินเตอร์รัพท์ให้ CPU ส่งข้อมูลใหม่ต่อไป

ในกรณีที่ต้องการส่งข้อมูลต่อเนื่องจากวิธีเดิม แต่ถ้า CPU ยังไม่ต้องการส่งข้อมูลต่อ หรือยังไม่มีข้อมูลก็สามารถสั่งให้พอร์ตขอถอนอินเตอร์รัพท์ได้ (ทำให้ INTR เป็น “0”) ด้วยการส่งแอดเดรสไบท์ไปยังพอร์ตควบคุม เป็นการกำหนดให้บิทใดๆของพอร์ต C เป็นลอจิก “0” หรือ “1” ตามต้องการ มีวิธีการกำหนดดังนี้

บิท 7 เป็น “0” เพื่อแสดงว่าเป็นแอดเดรสไบท์

บิท 6 ถึงบิท 4 ไม่ใช้ คือจะเป็นลอจิกใดก็ได้ ปกติจะให้เป็นลอจิก “0”

บิท 3 ถึงบิท 1 เป็นการบอกตำแหน่งบิทของพอร์ต C ที่ต้องการอ้างถึงดังนี้

000 = PC0 (บิท 0 ของพอร์ต C)

001 = PC1

010 = PC2

011 = PC3

100 = PC4

101 = PC5

110 = PC6

111 = PC7

บิท 0 ให้บิทใดๆของพอร์ต C ที่กำหนดไว้ (ด้วยบิท 3 ถึงบิท 1) นั้นมีลอจิกเป็น “0” หรือ “1” เช่นถ้าต้องการให้พอร์ต B ขอถอนอินเตอร์ก็คือทำให้ขา PC0 ซึ่งเป็นขา INTR ของพอร์ต B นั้น กลับเป็นลอจิก “0” จะได้แอดเดรสไบท์เป็น 00000000 หรือ 00H ในฐาน 16

การรับส่งข้อมูลแบบ Handshaking โดยกรรมวิธีของอินเตอร์รัพท์นี้ ถ้าในกรณีที่มี 8255 ทำงานร่วมกันหลายตัว จะทำให้วิธีการขออินเตอร์รัพท์นี้ยุ่งยากมาก เนื่องจากต้องมีวิธีการจัดลำดับความสำคัญเมื่อเกิดมีการร้องขออินเตอร์รัพท์พร้อมกันหลายๆตัว ซึ่งมีหลายแบบ ทั้งนี้ผู้ที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ 8255 ทำงานในโหมด 1 นี้ (หรือ โหมด 2 ก็ตาม) จะต้องมีความเข้าใจในการทำงาน รวมทั้งมีความรู้ในระบบไมโครคอมพิวเตอร์มากพอสมควร

4.5.3 โหมด 2

ในโหมดที่ 2 นี้ เป็นการรับส่งข้อมูลในแบบ Handshaking เช่นเดียวกับโหมด 1 แต่พอร์ตที่จะใช้งานในโหมดนี้ได้นั้น มีเพียงเฉพาะพอร์ท A พอร์ทเดียว แต่พอร์ท A ในโหมดนี้สามารถเป็นได้ทั้งอินพุท และ เอ้าท์พุท หรือที่เรียกว่าเป็นพอร์ตแบบสองทิศทาง (Bi-directional) ส่วนสัญญาณที่ใช้ในการทำ Handshaking คือสัญญาณจาก พอร์ท C ซึ่งมีลักษณะตามตารางที่ 3

พอร์ต C	ความหมาย
PC0	I/O
PC1	I/O
PC2	I/O
PC3	$INTR_A$
PC4	STB_A
PC5	IBF_A
PC6	ACK_A
PC7	OBF_A

ตารางที่ 4.3 แสดงสัญญาณ Handshaking ในโหมด 2

ขณะที่โปรแกรมให้พอร์ต A ทำงานในโหมด 2 นี้ เรายังสามารถโปรแกรมให้พอร์ต B นั้นทำงานในโหมด 0 หรือ โหมด 1 ก็ได้ ซึ่งจะทำให้สามารถนำพอร์ท B ไปใช้งานอื่นๆได้อีก โดยที่เป็นอิสระจากกัน

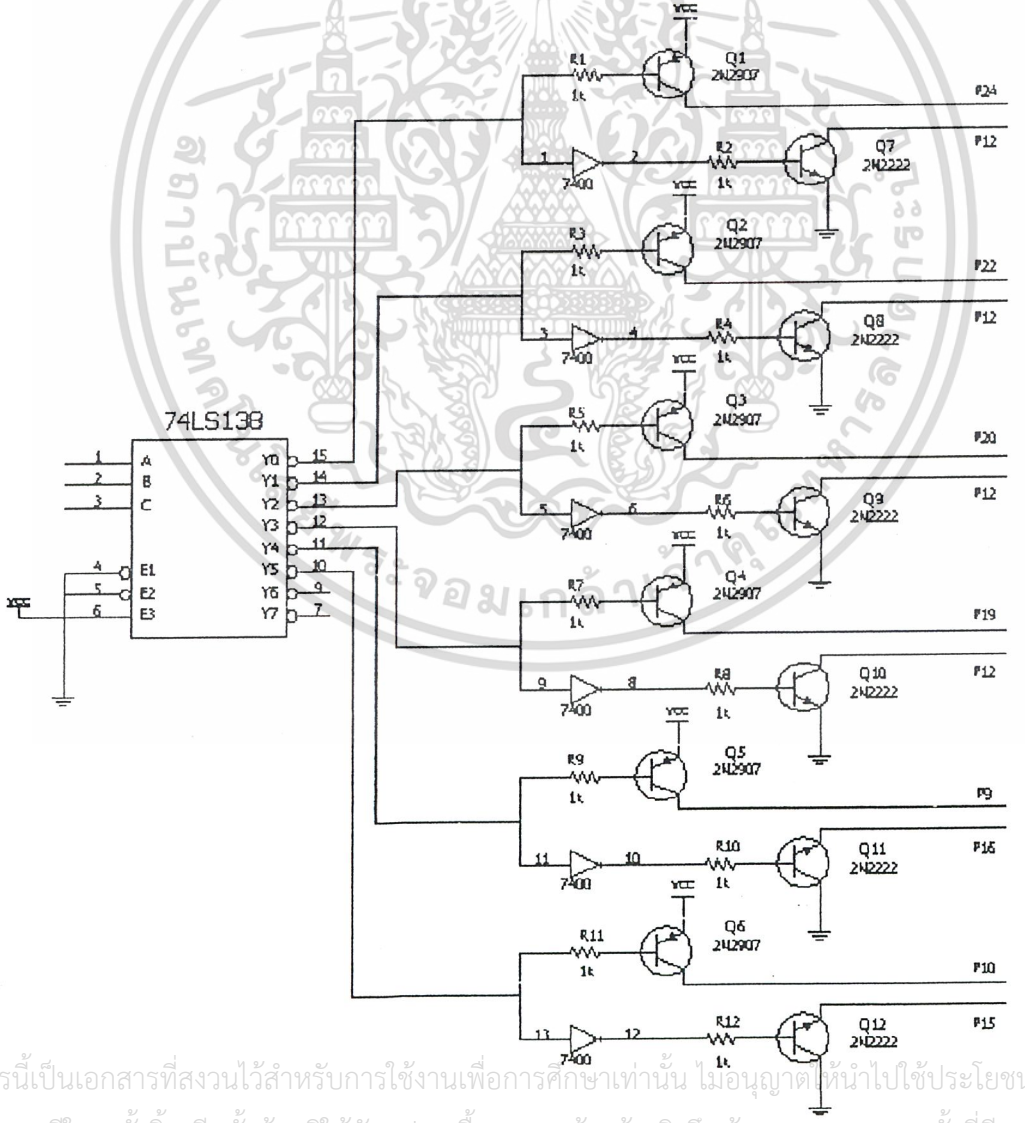
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โครงสร้างทางฮาร์ดแวร์ และ ซอฟท์แวร์

5.1 โครงสร้างทางฮาร์ดแวร์

5.1.1 Programmable Supply

ในการทดสอบไอซีนั้น เนื่องจากไอซีแต่ละตัวมีจำนวนขา และการจัดวางขาที่แตกต่างกันไป เพราะฉะนั้นขาไฟเลี้ยงและกราวน์ก็จะแตกต่างกันไปด้วย ดังนั้นการจะป้อนไฟเลี้ยงและกราวน์ให้กับไอซี ระหว่างการทดสอบแต่ละเบอร์นั้นจึงต้องใช้เทคนิคเพื่อให้สามารถป้อนไฟเลี้ยงได้ถูกต้องสำหรับไอซีแต่ละเบอร์ โดยเราจะใช้การ Decoder ร่วมกับการใช้ Transistor เป็นสวิตซ์ตัวเลือกช่องไฟเลี้ยงและกราวน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.1 Programmable Supply

จากรูปวงจร สามารถอธิบายการทำงานได้ดังนี้

ไอซีเบอร์ 74LS138 เป็นไอซี Decoder ชนิด TTL 3 To 8 Line หมายความว่ามีขาอินพุท 3 ขา คือ A,Bและ C และมีขาเอ้าท์พุท 8 ขาคือ Y0-Y7 ซึ่งไอซี 74LS138มีตารางความจริงดังนี้

INPUT					OUTPUT							
ENABLE		SELECT										
G1	G2	C	B	A	Y0	T1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

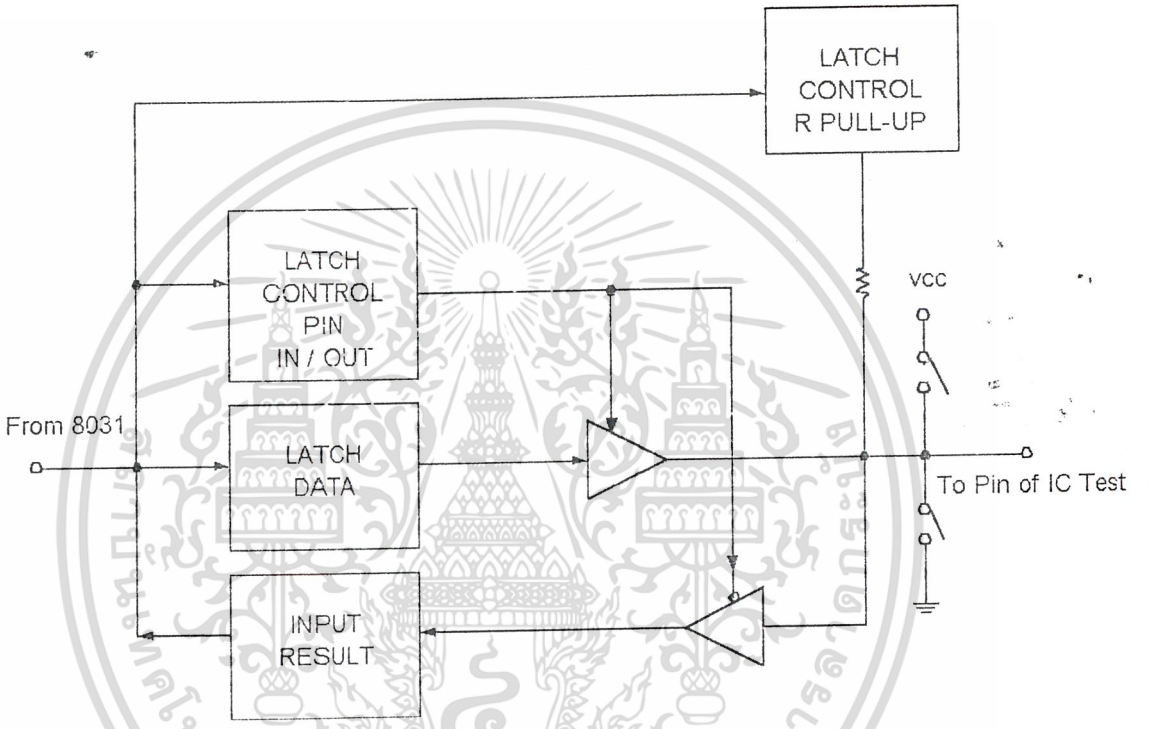
ตารางที่ 5.1 ตารางความจริงของ 74LS138

จากรูปขาอินพุททั้ง 3 ขา จะถูกต่ออยู่กับ PIA 8255 ซึ่งจะทำงานเป็น DATA BUFFER ของ MCS-51 Microcontroller ดังนั้นจะเห็นได้ว่าเราสามารถโปรแกรมให้ ไอซี 74LS138 Active ที่ขาใดก็ได้จากขาเอ้าท์พุททั้ง 8 ขา และเนื่องจากเอ้าท์พุทของ 74LS138 เมื่อ Active จะให้ลอจิกที่เป็น "0" (LOW) ดังนั้นการป้อนไฟเลี้ยง (VCC) จึงเลือกใช้ทรานซิสเตอร์ชนิด PNP เบอร์ 2N2907 เป็นสวิตช์ซึ่งทนกระแสได้สูงถึง 1 A และสำหรับตำแหน่งขากราวน์จะควบคุมได้เช่นเดียวกัน โดยต่อนอทเกทจากขาเอ้าท์พุทของ 74LS138 เพื่อนำไปขับทรานซิสเตอร์ชนิด NPN เบอร์ 2N2222 เนื่องจากมีการกำหนดตำแหน่งในการวางตัวไอซีไว้ที่จุดเดียวกันแล้วจะทำให้ขากราวน์เป็นจุดเดียวกันได้บ้าง สำหรับไอซีที่มีแพกเกจธรรมดา และแตกต่างกันตามแต่ละเบอร์ที่มีการวางขาที่เปลี่ยนไป ด้วยหากทำงานในลักษณะนี้ ทำให้สามารถที่จะต่อขาไฟบวกและกราวน์ ให้กับไอซีที่ทำการตรวจสอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 ภาคป้อนและควบคุมข้อมูล

ในภาคนี้ เป็นส่วนที่ติดต่อกับไอซีที่นำมาทดสอบโดยตรง จากบล็อกไดอะแกรมรูปที่ 5.2 วงจรในภาคนี้แบ่งออกเป็นส่วนหลักๆ คือ ส่วนควบคุม I/O , ส่วนส่งถ่ายข้อมูล , ส่วนบัฟเฟอร์ 3 สถานะ , ส่วนควบคุม R PULL-UP และส่วนส่งผลลัพธ์กลับสู่ระบบ Microcontroller



รูปที่ 5.2 บล็อกไดอะแกรมแสดงการทำงานของส่วนควบคุมข้อมูลทดสอบไอซี

ส่วนควบคุม I/O (LATCH CONTROL PIN INPUT)

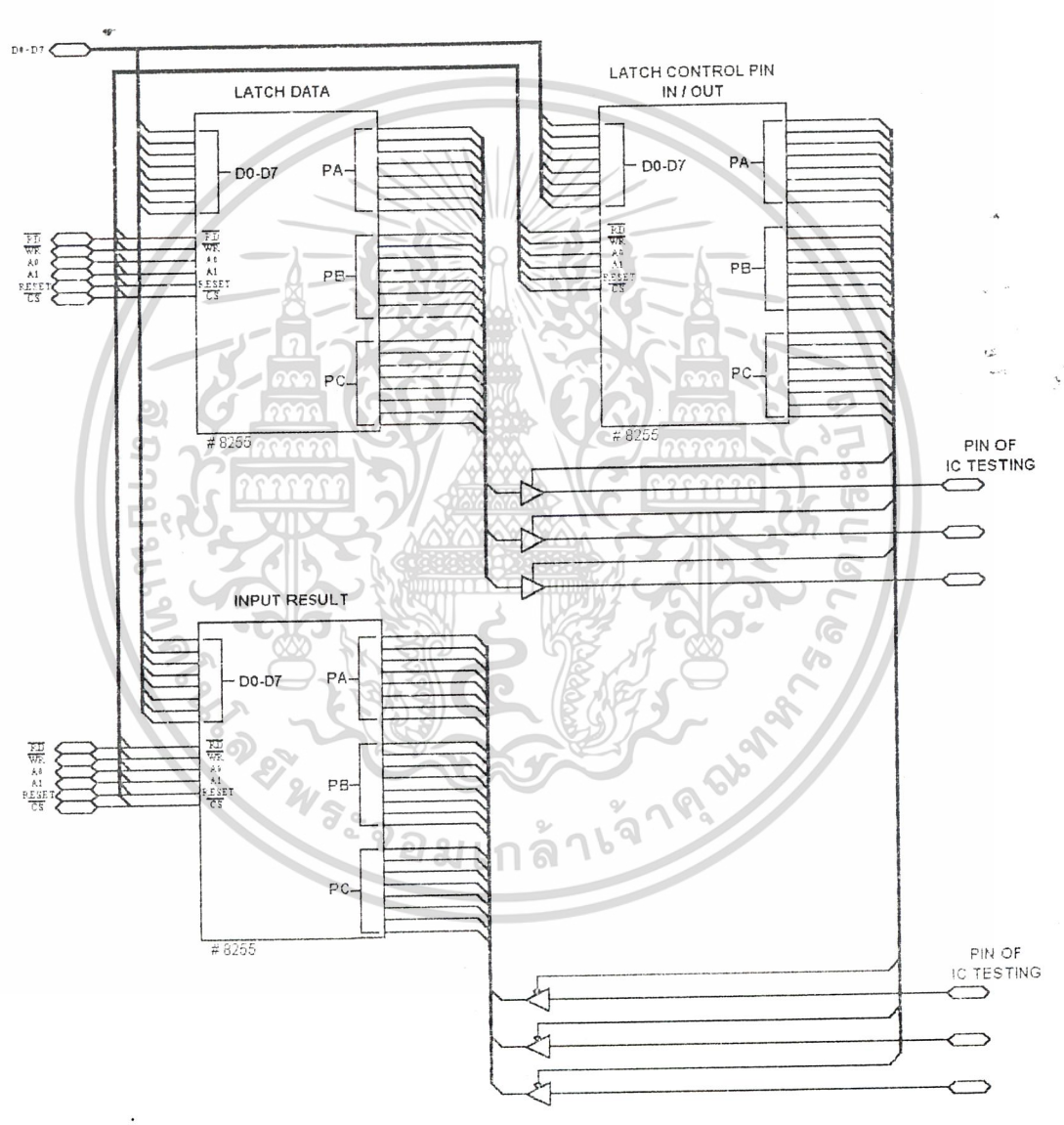
มีหน้าที่ควบคุมการปิดเปิดเส้นทางการส่งข้อมูล เข้าสู่ตัวไอซีที่ทดสอบ หรือจากตัวไอซีที่ทดสอบเข้าไปสู่ระบบ Microcontroller ถ้าขาใดของไอซีเป็นขาเอ้าท์พุท นั่นหมายความว่าไม่สามารถส่งข้อมูลเข้าสู่ขาดังกล่าวได้ ดังนั้นส่วนควบคุม I/O จะต้องตัดเส้นทางที่ข้อมูลจะเข้าส่งขาไอซีออกไป และเปิดเส้นทางที่สัญญาณจะเดินทางไปสู่ระบบ Microcontroller ได้

วงจรภายในจะใช้ PIA 8255 ทำหน้าที่เป็นตัวรับข้อมูลจากระบบ Microcontroller มาทำการ Latch ไว้เพื่อส่งต่อไปควบคุม Buffer 3 สถานะต่อไป จะเห็นได้ว่าใช้ Port ทั้ง 3 ของ 8255 ซึ่งมีทั้งหมด 24 เส้น สามารถที่จะคลอบคลุมการทดสอบไอซีที่มีจำนวนขาไม่เกิน 24 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนส่งถ่ายข้อมูล (LATCH DATA)

มีหน้าที่รับข้อมูลที่จะนำไปทดสอบไอซี จากคอนโทรลเลอร์มาเก็บไว้โดยผ่านทาง PIA 8255 เหมือนกับส่วนควบคุมอินพุท เอ้าท์พุททุกประการ ต่างกันเพียงหน้าที่ และชนิดของข้อมูลที่วงจรส่วนนี้จัดการเท่านั้น



รูปที่ 5.3 แสดงตัวอย่างส่วนควบคุม (Latch Control Pin In/Out) , ส่วนส่งถ่ายข้อมูล (Latch Data) และส่วน ส่งผลลัพธ์กลับสู่ระ บบ Microcontroller (Input Relult)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

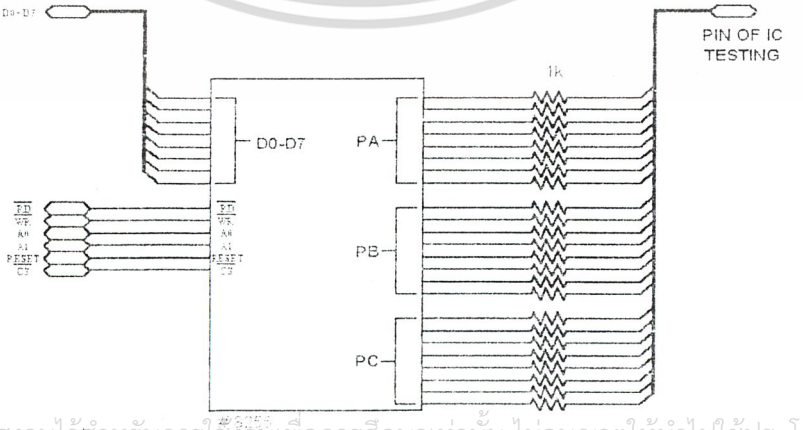
เมื่อ 8031 ได้ส่งข้อมูลมาเก็บไว้ในวงจรส่วนส่งถ่ายข้อมูลเรียบร้อยแล้ว ไอซีที่ทดสอบจะยังไม่ได้รับข้อมูลจากจุดนี้โดยตรง เพราะข้อมูลจะต้องผ่านส่วนสำคัญอีกส่วนนั่นคือ ส่วนบัฟเฟอร์ 3 สถานะดังรูปที่ 5.3 หน้าที่ของบัฟเฟอร์นี้ คล้ายกับประตูที่ทำการเปิดปิดให้ข้อมูลผ่านออกไป ดังที่กล่าวข้างต้นแล้วว่า ไอซีที่จะทำการทดสอบนั้น ถ้าขาใดเป็นขาเอ้าท์พุทแล้ว และยังจะมีการส่งข้อมูลออกไป จะทำให้เกิดความผิดพลาดขึ้นได้

ส่วนส่งผลลัพธ์กลับสู่ระบบ Microcontroller (INPUT RESULT)

มีหน้าที่ส่งข้อมูล ที่ได้จากเอ้าท์พุทของไอซีที่ทำการทดสอบ กลับเข้าสู่ระบบ Microcontroller การทำงานก็จะคล้ายๆกับส่วนส่งถ่ายข้อมูล ต่างกันเพียงทิศทางของการส่งถ่ายข้อมูล และลักษณะของบัฟเฟอร์เท่านั้น โดยบัฟเฟอร์ที่ใช้ในส่วนนี้จะไม่ทำงานพร้อมกับบัฟเฟอร์ที่อยู่ในส่วนส่งถ่ายข้อมูล ดังนั้นก็เป็นการกำหนดได้ว่า แต่ละขาของไอซีที่ทำการทดสอบ จะไม่เป็นขาอินพุท และเอ้าท์พุทพร้อมกัน

ไอซีเบอร์ 74HC126 และ 74HC125 เป็นบัฟเฟอร์ 3 สถานะ 4 ตัวบนชิฟเดียว ในการทดสอบไอซี 24 ขาจึงต้องใช้ไอซี อย่างละ 6 ตัว ซึ่ง 74HC126 มีขาคอนโทรลแอคทีฟ “1” และ 74HC125 มีขาคอนโทรลแอคทีฟ “0” และขาคอนโทรลของบัฟเฟอร์ทั้งสองเบอร์จะมาจากส่วนควบคุมเส้นเดียวกัน ดังนั้นถ้าขาของไอซีที่ทำการทดสอบขาใดเป็นขาเอ้าท์พุท ส่วนควบคุมจะต้องส่งลอจิก “0” ออกมา เพื่อตัดเส้นทางสัญญาณที่จะเข้าไปสู่ตัวไอซีที่ทำการทดสอบ และเพื่อเชื่อมเส้นทางของสัญญาณจากเอ้าท์พุทของไอซีที่ทำการทดสอบให้วิ่งไปสู่ส่วนของการส่งผลลัพธ์กลับสู่ระบบ Microcontroller ได้เป็นต้น

ส่วนควบคุม R PULL - UP (R PULL - UP)



รูปที่ 5.4 แสดงส่วนควบคุม R PULL - UP (R PULL - UP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีที่เป็นแบบ Open Collector ที่ขาเอ้าท์พุทจะต้องทำการต่อกับลอจิกสูง ผ่าน ตัวต้านทาน ดังนั้น จึงใช้ 8255 อีกตัวหนึ่งทำการควบคุมการตัดต่อ ตัวต้านทานเหล่านี้เข้ากับขาเอ้าท์พุทของไอซี เมื่อขาไหนที่เป็นเอ้าท์พุทแบบ Open Collector ก็ให้ขานั้นเป็นลอจิก "1" ส่วนขาที่ไม่ใช่เอ้าท์พุท ก็ให้เป็นลอจิก "0"

5.2 โครงสร้างทาง ซอฟท์แวร์

5.2.1 การติดต่อสื่อสารระหว่าง ไมโครคอนคอมพิวเตอร์ กับ ไมโครคอนโทรลเลอร์

ในการโปรแกรมให้การติดต่อระหว่างไมโครคอมพิวเตอร์ กับไมโครคอนโทรลเลอร์เป็นไปอย่างรวดเร็วและมีหลักการ จำเป็นที่จะต้องกำหนดรูปแบบของการติดต่อให้เป็นที่เข้าใจทั้งสองฝ่ายเสียก่อน

จากลักษณะของวงจร ที่มีไมโครคอนโทรลเลอร์คอยควบคุมสัญญาณเข้า-ออกของไอซีที่ทำการทดสอบ ส่วนการควบคุมว่าจะให้มีการเข้า-ออกอย่างไรอยู่ที่ไมโครคอมพิวเตอร์ ซึ่งจะติดต่อสั่งการไปถึงไมโครคอนโทรลเลอร์โดยผ่านทาง Serial Port ซึ่งขั้นตอนการติดต่ออย่างคร่าวๆ เรียงตามลำดับขั้นตอนเป็นดังนี้

หลังจากผู้ใช้ป้อนเบอร์ของไอซีที่จะทำการทดสอบแล้ว คอมพิวเตอร์ส่ง Control Word ไปให้ไมโครคอนโทรเลอร์ ซึ่ง Control Word นี้จะเป็นตัวบอกให้ ไมโครคอนโทรลเลอร์ว่า จะเซ็ทให้ขาไหนของไอซีที่ทำการทดสอบ เป็นอินพุท หรือว่า เอ้าท์พุท ตามเบอร์ของไอซีที่รับเข้ามา

หลังจากเซ็ทให้ขาของไอซีที่ทำการทดสอบเป็น อินพุท หรือ เอ้าท์พุทแล้ว ไมโครคอมพิวเตอร์ก็ส่งสัญญาณ Control Word อีกชุดหนึ่งไปบอกให้ ไมโครคอนโทรลเลอร์ว่าไอซีเบอร์นี้ มี Clock , VCC ,GND ที่ขาไหนบ้าง หรือ เป็นไอซีแบบ Open collector output หรือเปล่า

เมื่อไมโครคอนโทรลเลอร์ได้รับ Control Word เรียบร้อยแล้ว ก็ส่งสัญญาณตอบกลับ มาให้กับคอมพิวเตอร์ ถ้าเกิดการผิดพลาดขึ้นในการรับ-ส่งข้อมูล คอมพิวเตอร์ก็จะทำการส่ง Control Word ไปใหม่

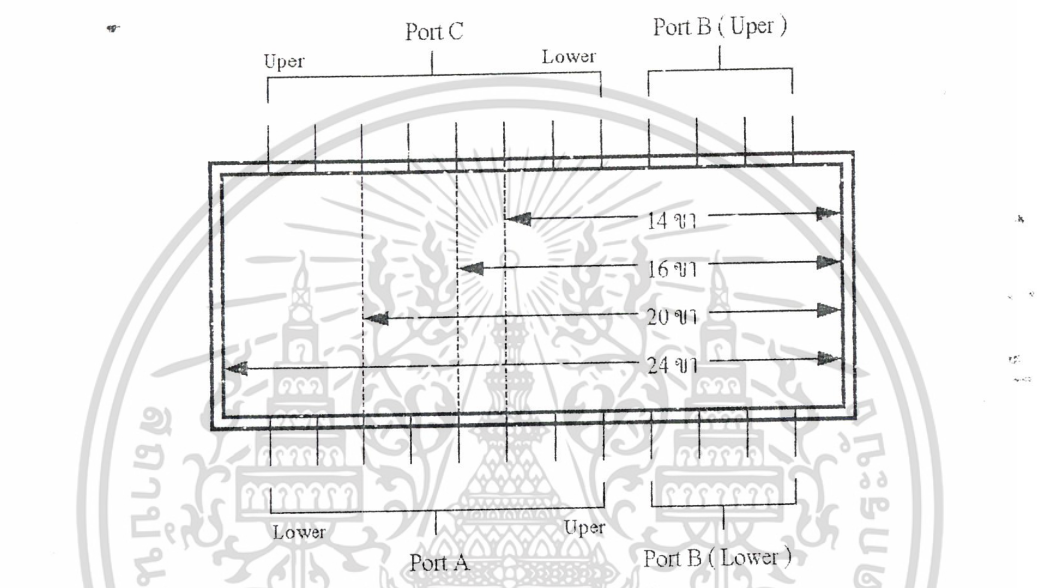
คอมพิวเตอร์ส่งข้อมูลที่จะทำการป้อนทดสอบไอซี โดยข้อมูลอินพุทที่ส่งไป จะเป็นไปตามตารางความจริงของไอซีเบอร์ที่ทำการทดสอบ

ไมโครคอนโทรลเลอร์ทำการส่งข้อมูลทดสอบ และเก็บผลลัพธ์ส่งกลับมาให้ ไมโครคอมพิวเตอร์

ไมโครคอมพิวเตอร์ ทำการประมวลผลว่า เอ้าท์พุทของไอซีที่ทำการทดสอบว่า ถูกต้องตามตารางความจริงหรือไม่ และทำการแสดงผลในรูปแบบของกราฟฟิก ให้ผู้ใช้งานทราบ

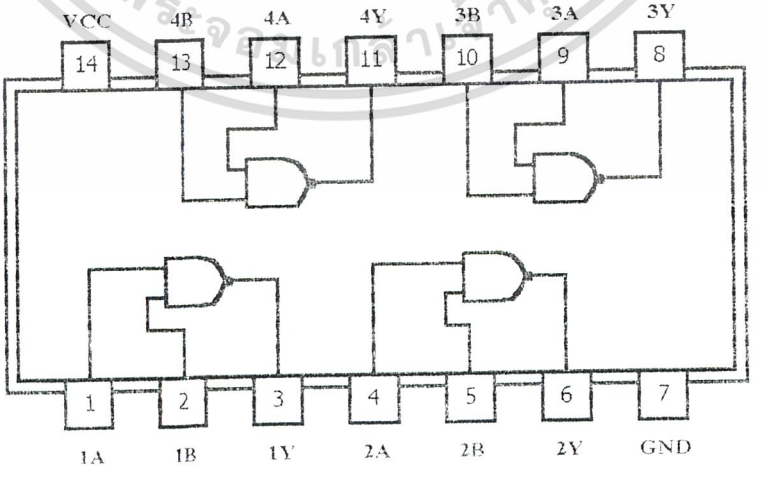
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลของ Control Word , ตารางความจริง ของไอซีแต่ละเบอร์จะทำการเก็บไว้ในรูปแบบของไฟล์ฐานข้อมูล ในการทำงาน Visual Basic จะทำการทำงานร่วมกับฐานข้อมูลเหล่านี้ ข้อมูลที่ทำการเก็บลงฐานข้อมูล มาจากรูปแบบการต่อวงจรเข้ากับแต่ละขาของไอซีที่ทำการทดสอบ ซึ่งลักษณะการต่อวงจรเข้ากับขาไอซีโดยรวม เป็นดังรูปต่อไปนี้



รูปที่ 5.5 รูปแบบการต่อวงจรจาก 8255 เข้าสู่ขาของไอซีที่ทำการทดสอบ

ยกตัวอย่างของไอซีเบอร์ 7400 ซึ่งมีวงจรภายในคือ



รูปที่ 5.6 วงจรภายในไอซีเบอร์ 7400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีเบอร์ 7400 นี้มี 14 ขา เทียบกับรูปที่ 5.5 จะเห็นว่าขาของไอซีที่จะเซ็ทให้เป็นขา อินพุท ลอจิกควบคุมบัฟเฟอร์จะต้องเป็น "1" ถ้าเป็นเอ้าท์พุทจะเป็น "0" ขา VCC , GND และขาที่ไม่ได้ใช้งาน จะให้ควบคุมด้วยลอจิก "0" ดังนั้นไอซี 7400 จะได้ Control Word เป็น

- Port A 0110 0000
- Port B 0110 0011
- Port C 0000 0011

หลังจากป้อน Control Word ควบคุมให้แต่ละขาของไอซีที่ทำการทดสอบ เป็น อินพุท หรือ เอ้าท์พุท และทำการเซ็ทคู่ของ VCC และ GND แล้ว ต่อไปก็ป้อนอินพุททดสอบตามตารางความจริง ซึ่งตัวอย่างตารางความจริงของไอซีเบอร์ 7400 เป็นดังนี้

ID	INPUT			OUTPUT
	PORT A	PORT B	PORT C	
1	0000 0000	0000 0000	0000 0000	0000 1111
2	0010 0000	0010 0001	0000 0010	0000 1111
3	0110 0000	0110 0011	0000 0110	0000 0000
4	0100 0000	0100 0010	0000 0100	0000 1111

ตารางที่ 5.2 แสดงตารางความจริงของไอซีเบอร์ 7400 ที่ใช้ในโปรแกรม

จากตารางความจริง ข้อมูลอินพุทที่ป้อน จะเรียงตามนี้คือ 00,01,11,10 แต่จะทำการป้อนไปพร้อมกันทุกๆเกท จึงได้ข้อมูลตามตารางที่ 5.1 ส่วนเอ้าท์พุท ในแต่ละครั้งที่ป้อนอินพุท จะทำการรวมไว้ในไบต์เดียว ซึ่ง 7400 มีเกททั้งหมด 4 ตัว บิทที่ใช้เก็บเอ้าท์พุทจึงใช้เพียง 4 บิทต่ำ

เมื่อได้เอ้าท์พุทแล้ว ก็ทำการประมวลผลว่ามีเกทตัวไหนบ้างที่เสีย แล้วทำการแสดงผล ส่วนไอซีเบอร์อื่นๆ ก็ใช้หลักการเดียวกัน เพียงแต่จะแตกต่างกันที่ Control Word และ ตารางความจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากตารางความจริงของไอซี (ตารางที่ 5.1) แล้ว ตารางของ Control Word และ ตารางที่แสดง ขาของไอซีที่มี Clock แสดงดังต่อไปนี้

IC_ID	Port A	Port B	Port C	Open collector
7400	0110 0000	0110 0011	0000 0011	NO
7404	0000 0101	0100 0101	0000 0010	NO
74xx	xxxx xxxx	xxxx xxxx	xxxx xxxx	x

ตารางที่ 5.3 แสดงตัวอย่างตารางที่ใช้เก็บข้อมูลของ Control Word

IC_ID	CLK1	CLK2
7473	1	6
7476	1	6
74xx	x	x

ตารางที่ 5.4 แสดงตัวอย่างตารางที่ใช้เก็บข้อมูลของขา Clock ของไอซีที่ใช้ Clock ในการทำงาน

รหัสที่ใช้ในการติดต่อสื่อสารระหว่างไมโครคอมพิวเตอร์ กับ ไมโครคอนโทรลเลอร์

เมื่อมีข้อมูลที่ต้องใช้ในการตรวจสอบไอซี และใช้สั่งการให้ไมโครคอนโทรลเลอร์ทำงานตามต้องการแล้ว ต่อไปก็มีความจำเป็นที่จะต้องกำหนดรหัสที่ใช้ในการติดต่อ เพื่อให้การติดต่อสื่อสารเป็นไปโดยไม่มีความผิดพลาด รหัสที่ใช้ก็จะมี

- ต้องการส่งข้อมูล (0FFH)
- พร้อมรับข้อมูล (0FEH)
- รับข้อมูลแล้ว (0FDH)
- จบการส่งข้อมูล (0FCH)

ตัวอย่างขั้นตอนในการส่ง Control Word

ไมโครคอมพิวเตอร์ -> ต้องการส่งข้อมูล

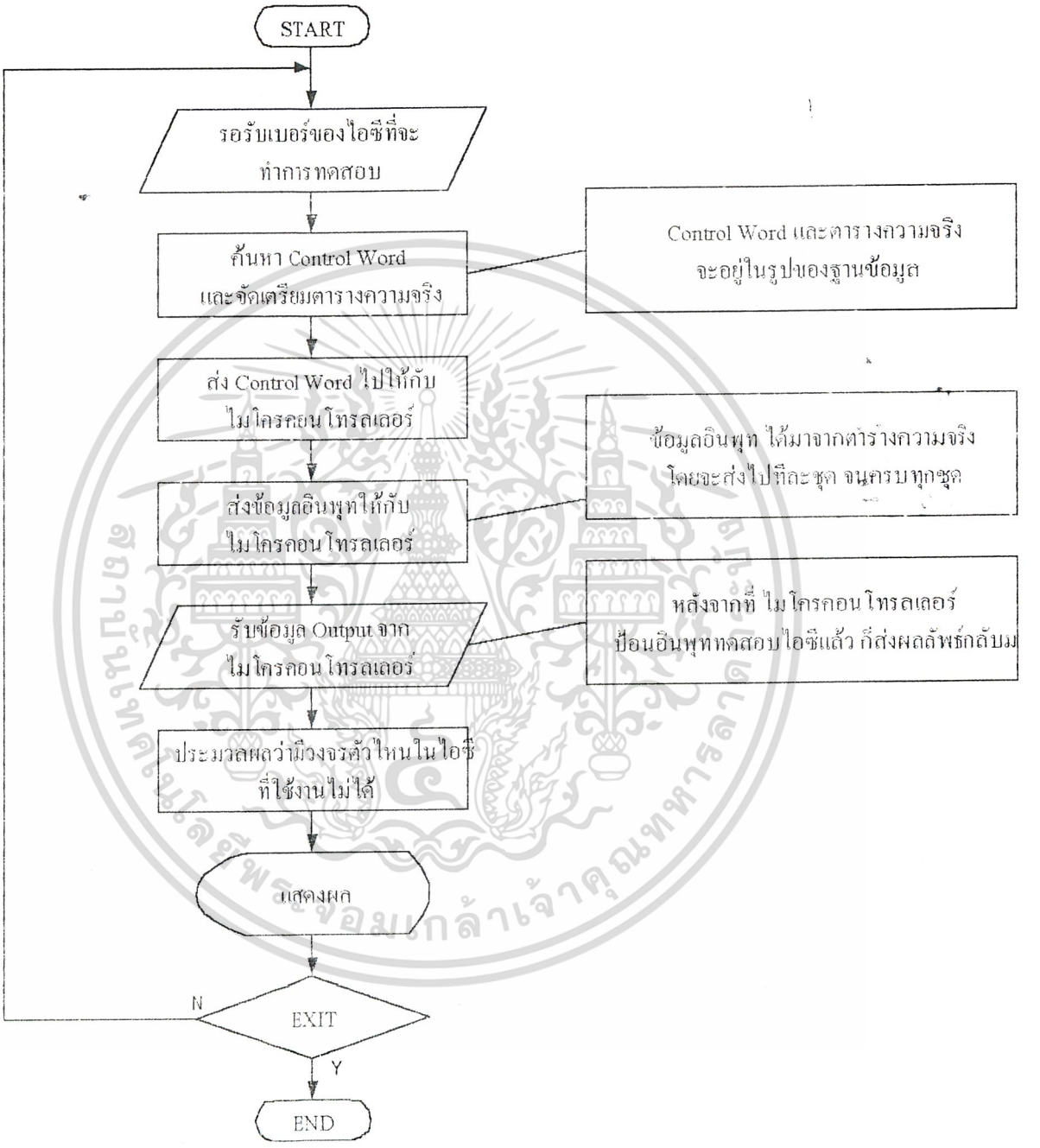
ไมโครคอนโทรลเลอร์ -> พร้อมรับข้อมูล

ไมโครคอมพิวเตอร์ -> ส่ง Control Word ชุดที่1

ไมโครคอนโทรลเลอร์ -> รับข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 FLOW CHART แสดงการทำงานในส่วนของ ไมโครคอมพิวเตอร์

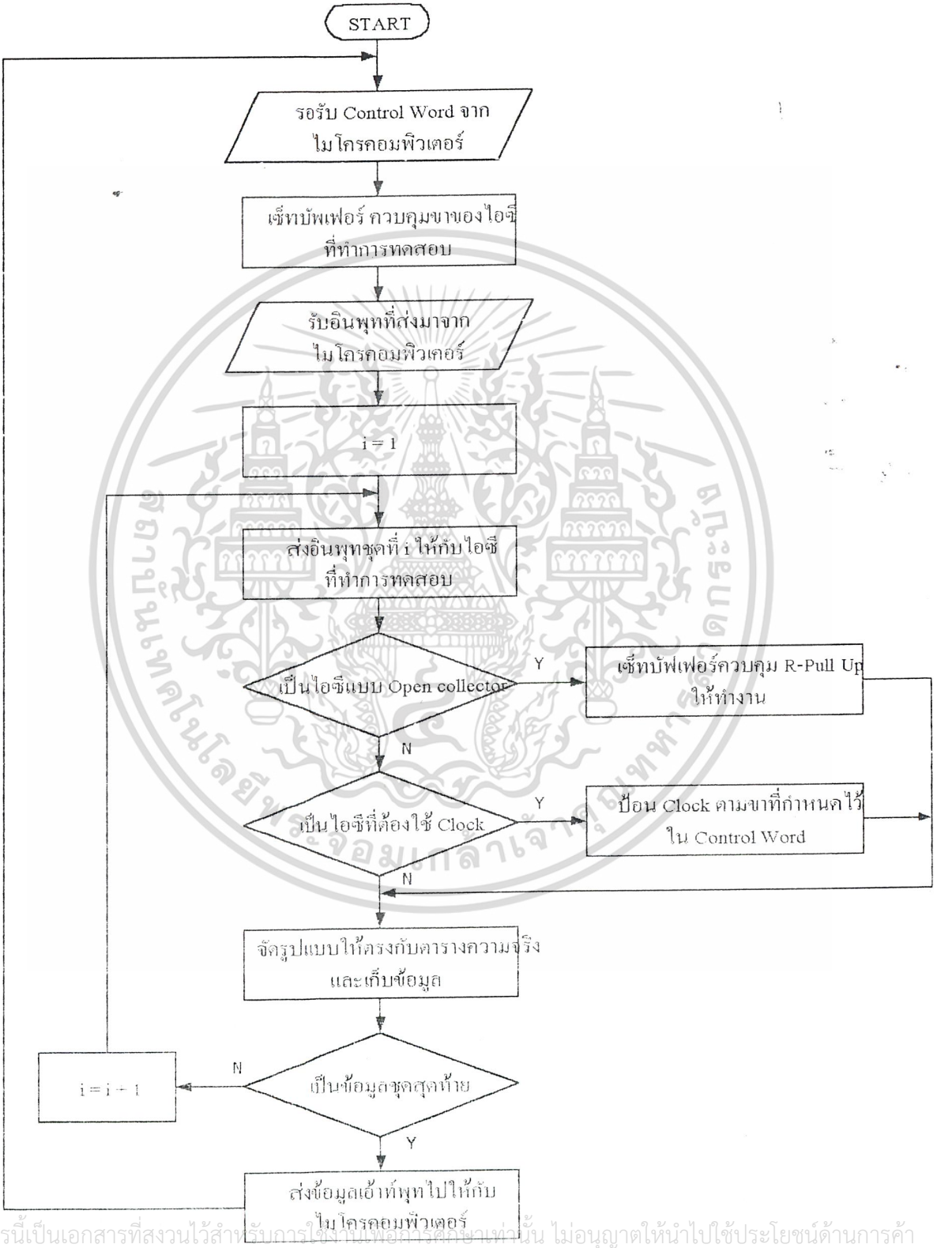


รูปที่ 5.8 แสดง FLOW CHART การทำงานในส่วนของ ไมโครคอมพิวเตอร์

การทำงานโดยรวมของโปรแกรม จะเป็นการทำงานที่มีการติดต่อกันระหว่าง ไมโครคอมพิวเตอร์ และ ไมโครคอนโทรลเลอร์ จาก Flow Chart ในรูปที่ 5.8 เป็นการทำงานในส่วนของ ไมโครคอมพิวเตอร์ ส่วน Flow Chart การทำงานในส่วนของ ไมโครคอนโทรลเลอร์ แสดงในหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 FLOW CHART แสดงการทำงานในส่วนของ ไมโครคอนโทรลเลอร์



รูปที่ 5.9 แสดง FLOW CHART การทำงานในส่วนของ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

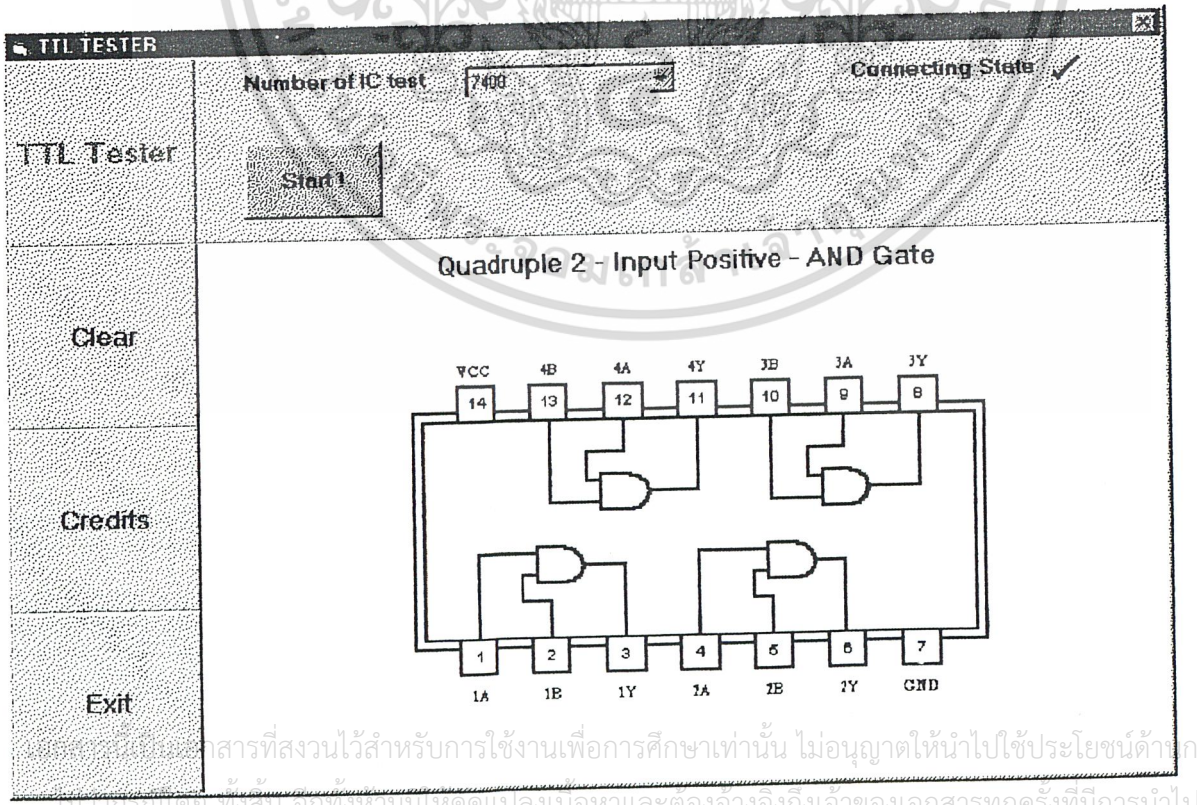
บทที่ 6

ผลการทดลอง

การทำงานของเครื่องทดสอบไอซี ทีทีแอล นี้ สามารถที่จะทำการทดสอบไอซี ทีทีแอล ที่เป็นแบบ ไอซีเกทธรรมดา , ไอซีประเภทที่ใช้สัญญาณนาฬิกา , ไอซีประเภทที่เป็น ไอซีเข้ารหัส , ถอดรหัส , ไอซีแบบ Open Collector เป็นต้น ซึ่งในบทนี้จะทำการแสดงผลการทดลอง การทดสอบไอซีแต่ละประเภทที่กล่าวมาข้างต้น การทดสอบจะแยกเป็นกรณีที่ไอซีสามารถใช้งานได้ตามปกติ และ กรณีที่ทำให้วงจรภายในไอซีเสียก่อนที่จะทำการทดสอบ เพื่อตรวจสอบดูว่าการทำงานของเครื่องทดสอบนี้ ให้ผลการทำงานที่มีความถูกต้อง และ แม่นยำ

6.1 การติดต่อกับผู้ใช้งานของโปรแกรม ในไมโครคอมพิวเตอร์

ขั้นตอนแรกของการทำการทดสอบ ก็คือเริ่มที่ผู้ใช้งาน เลือกเบอร์ของไอซีที่ต้องการทดสอบที่หน้าจอคอมพิวเตอร์ หลังจากที่เลือกเบอร์ไอซีแล้ว ทำการสั่งการให้ไมโครคอนโทรลเลอร์เริ่มทำการทดสอบ เมื่อไมโครคอนโทรลเลอร์ทำการทดสอบได้ผลลัพธ์มาแล้วก็ทำการส่งกลับมาประมวลผลว่ามีวงจรไหนภายในไอซีที่เสียบ้าง จากนั้นก็แสดงผลที่หน้าจอคอมพิวเตอร์ ลักษณะของการติดต่อกับผู้ใช้งาน แสดงดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

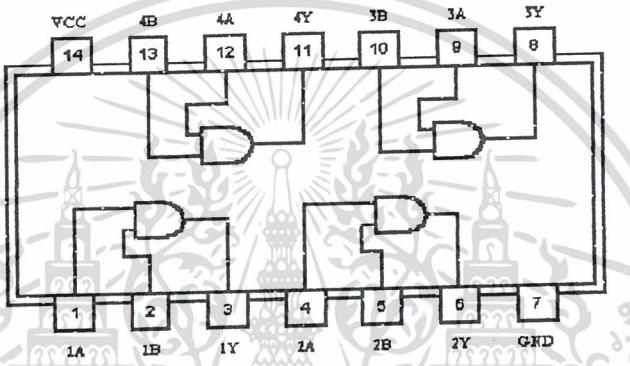
รูปที่ 6.1 แสดงหน้าจอการใช้งานของโปรแกรมทดสอบไอซี ทีทีแอล

6.2 การทดสอบไอซี BASIC GATE

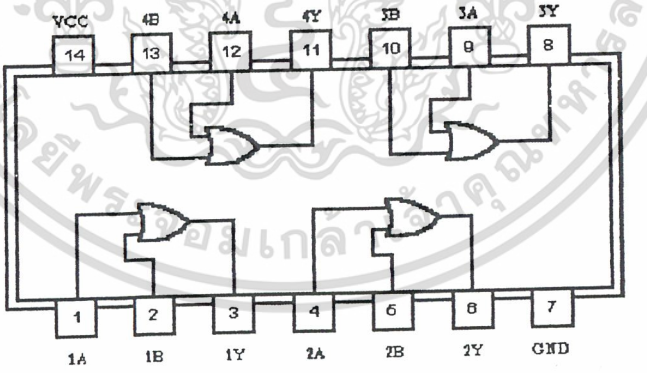
การทดสอบนี้ใช้ตัวอย่างของไอซีเบอร์ 7408 (AND GATE) , 7432 (OR GATE) , 7404 (NOT GATE)

6.2.1 กรณีที่ไอซีใช้งานได้ตามปกติ หลังจากการทดสอบแล้ว การแสดงผลก็จะไม่มีอะไรเกิดขึ้น แสดงได้ดังต่อไปนี้

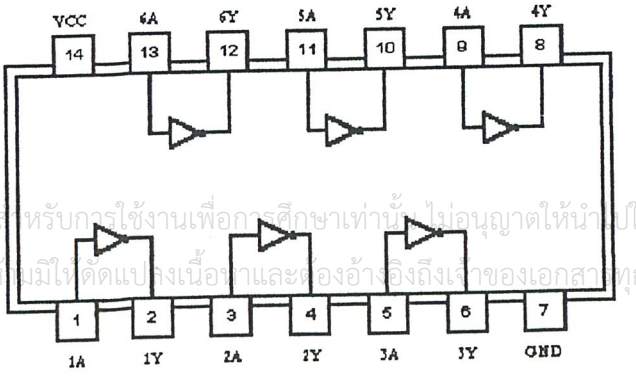
- AND GATE



- OR GATE



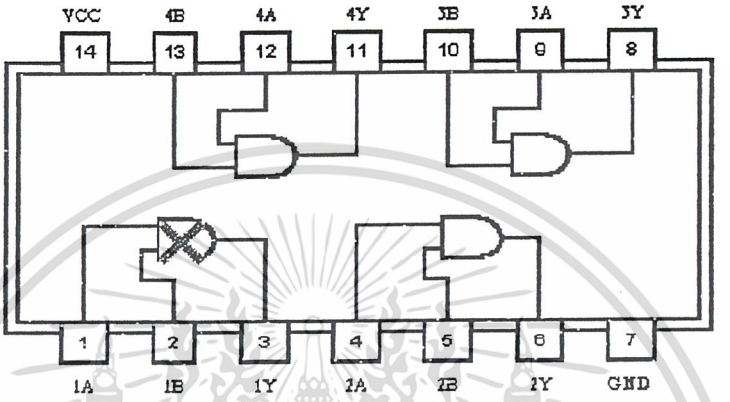
- NOT GATE



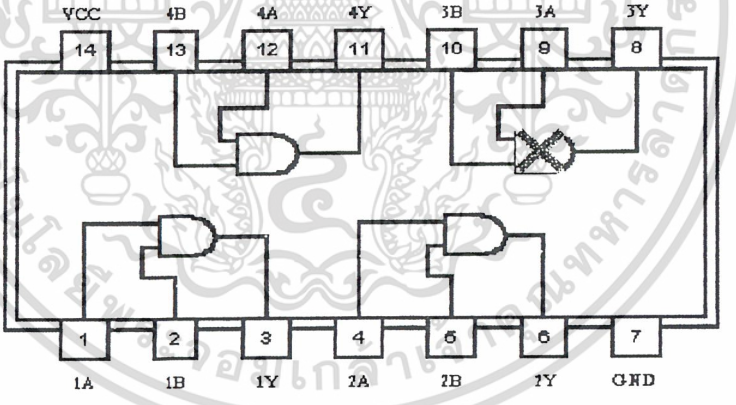
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 กรณีต่อไป ทำการทำให้เกทตัวใดตัวหนึ่งภายในไอซีเสีย ซึ่งในการทดลองจะทำได้โดยการตัดขาของเกทที่ต้องการจะทำให้เสียออก จากนั้นนำไอซีไปทำการทดสอบ

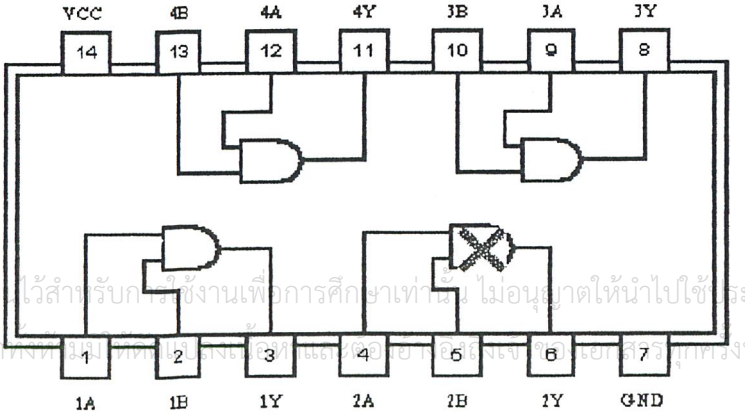
- ผลที่ได้จากการทดลองตัดขาที่ 2 ของ ไอซี 7408



- ผลที่ได้จากการทดลองตัดขาที่ 10 ของไอซี 7408

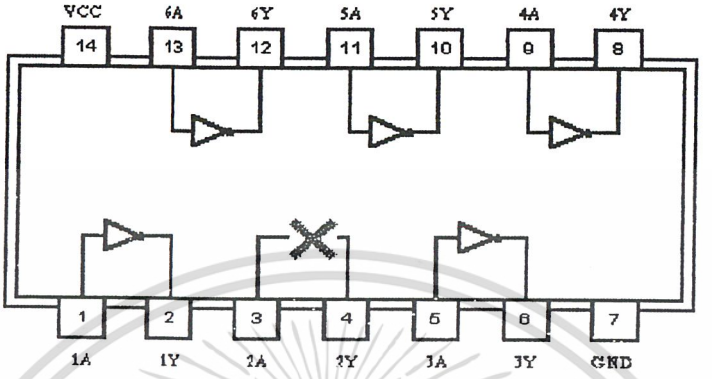


- ผลที่ได้จากการทดลองตัดขาที่ 6 ของไอซี 7408

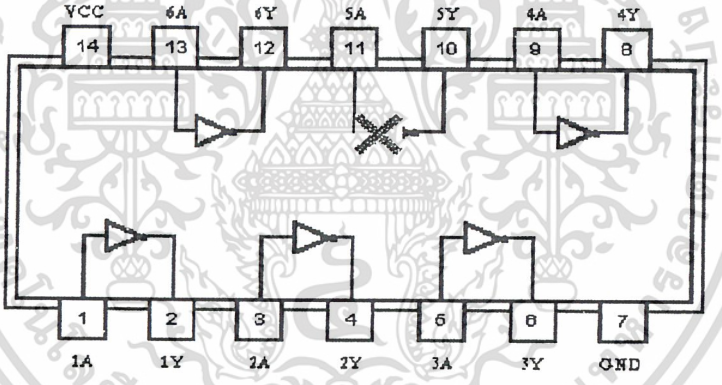


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

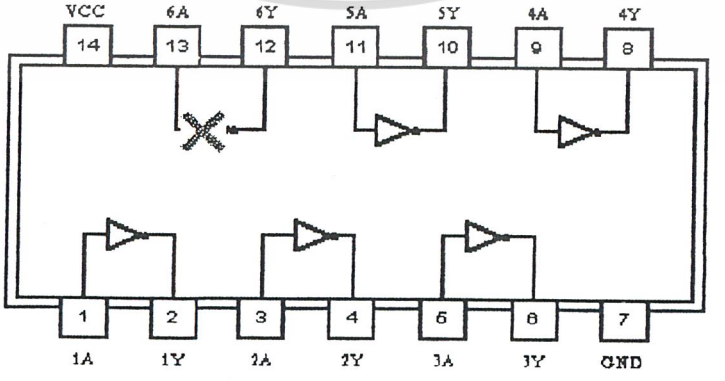
- ผลที่ได้จากการตัดขาที่ 3 ของไอซี 7404



- ผลที่ได้จากการตัดขาที่ 10 ของไอซี 7404

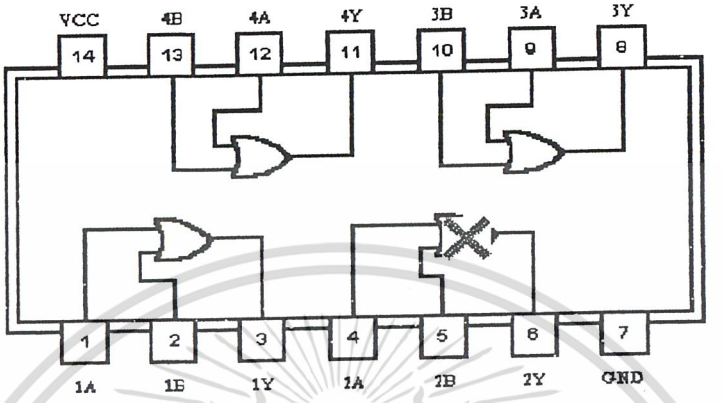


- ผลที่ได้จากการตัดขาที่ 13 ของไอซี 7404

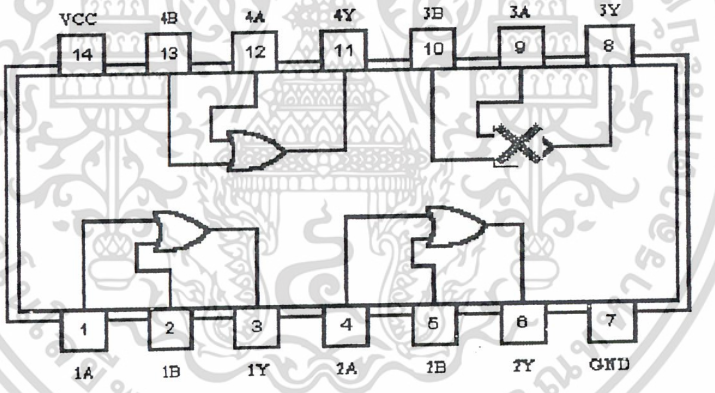


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

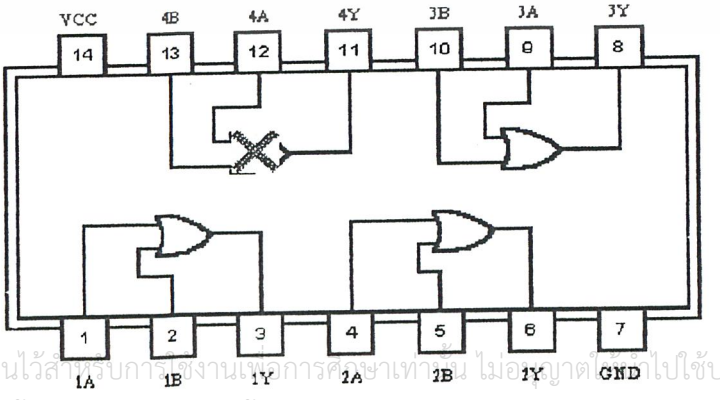
- ผลจากการตัดขาที่ 6 ของไอซี 7432



- ผลจากการตัดขา 10 ของไอซี 7432



- ผลจากการตัดขาที่ 13 ของไอซี 7432



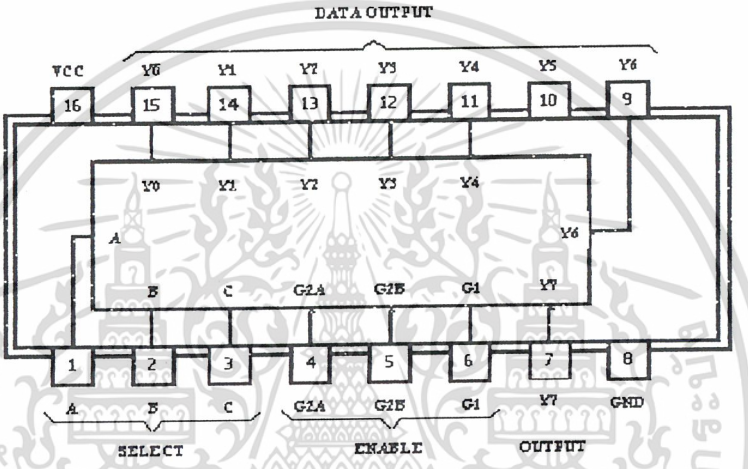
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดสอบไอซี ประเภทเข้ารหัส ถอดรหัส

ในการทดสอบไอซีประเภท เข้ารหัส ถอดรหัสนี้ ใช้ตัวอย่างไอซีเบอร์ 74138 ซึ่งเป็นไอซี Decoder เข้า 3 อินพุท ออก 8 อินพุท

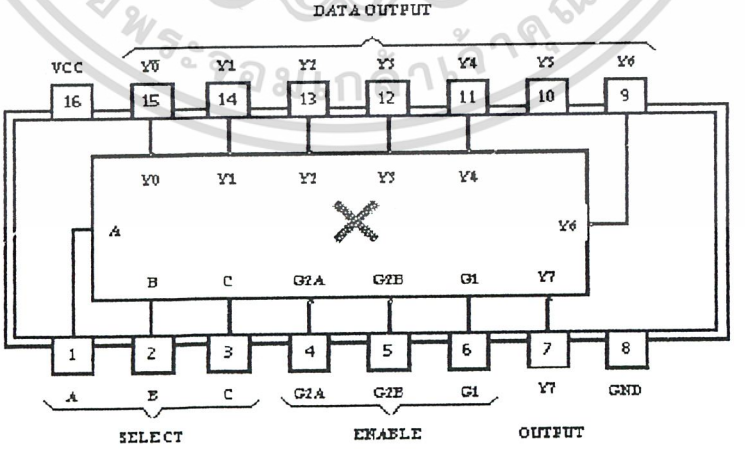
6.3.1 กรณีที่ไอซีใช้งานได้ตามปกติ

ได้ผลการทดสอบดังนี้



6.3.2 กรณีที่มีฟังก์ชันการทำงานที่ไม่ตรงกับตารางความจริง

ในการทดลองหัวข้อนี้ ทำการตัดขาที่ 3 ของไอซีออก แล้วทำการทดลอง ได้ผลดังนี้



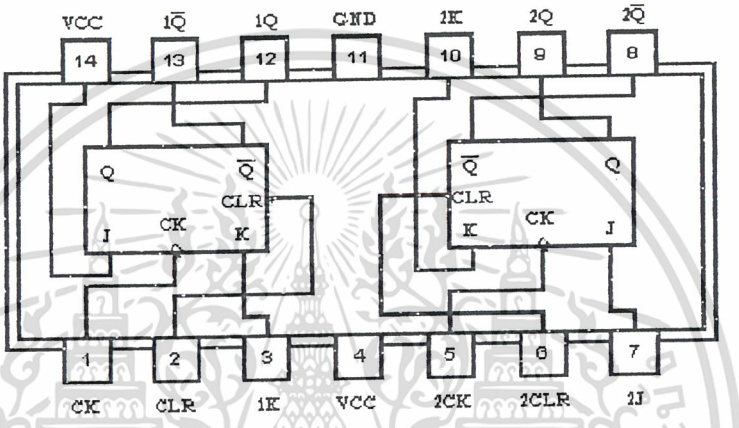
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 การทดสอบไอซีประเภทที่ใช้สัญญาณนาฬิกา

การทดลองในหัวข้อนี้ ใช้ตัวอย่างไอซีเบอร์ 7473 (J-K Master – Slave Flip-Flop with Clear)

6.4.1 กรณีที่ไอซีใช้งานได้ตามปกติ

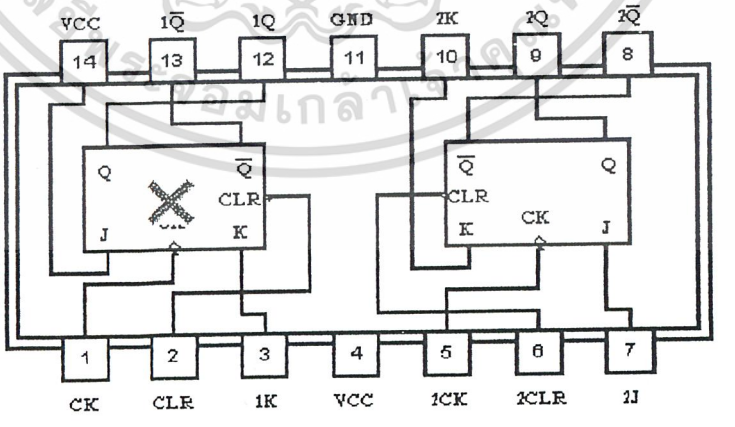
ได้ผลการทดลองดังนี้



4.4.1 กรณีที่มีฟังก์ชันการทำงานผิดพลาด

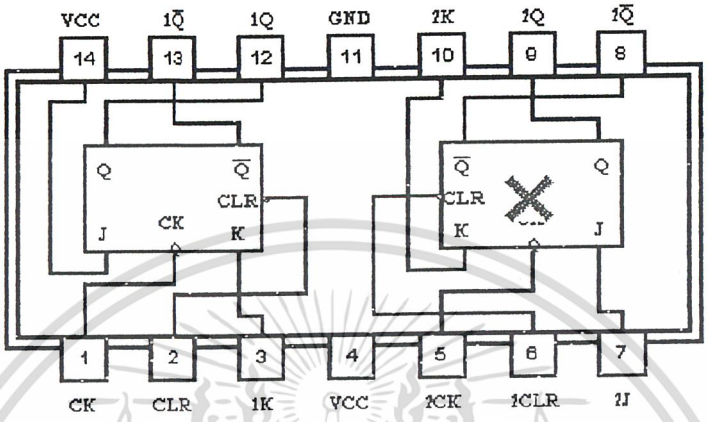
ทำได้โดยการตัดขาของฟังก์ชันภายในไอซี เพื่อดูผลการทดสอบ

- ตัดขา Clock ของ ฟลิป-ฟลอป ตัวที่ 1 ออก (ขา 1)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัดขา Clock ของ ฟลิป-ฟลอป ตัวที่ 2 ออก (ขาที่ 5)

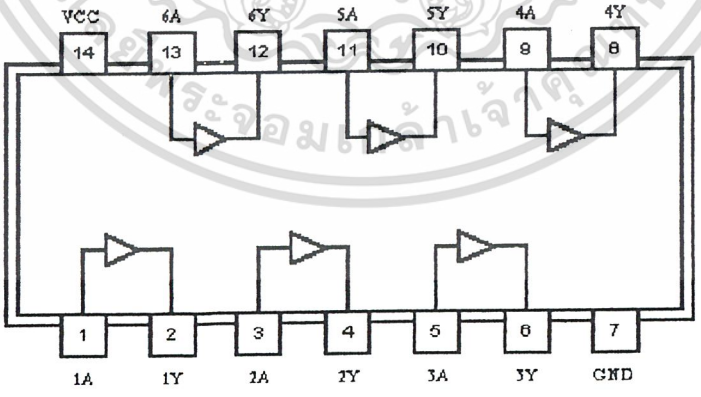


6.5 การทดสอบไอซีประเภท Open Collector

การทดสอบในหัวข้อนี้ ใช้ตัวอย่างไอซีเบอร์ 7407 (Buffer with Open – Collector) ทำการทดสอบได้ผลการทดลองดังนี้

6.5.1 กรณีที่ไอซีใช้งานได้ตามปกติ

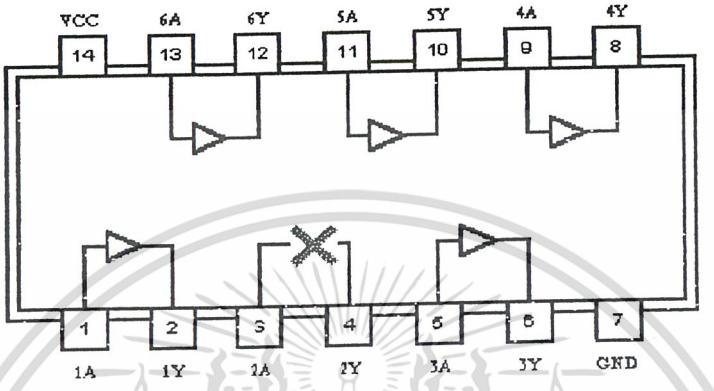
ได้ผลการทดลองดังนี้



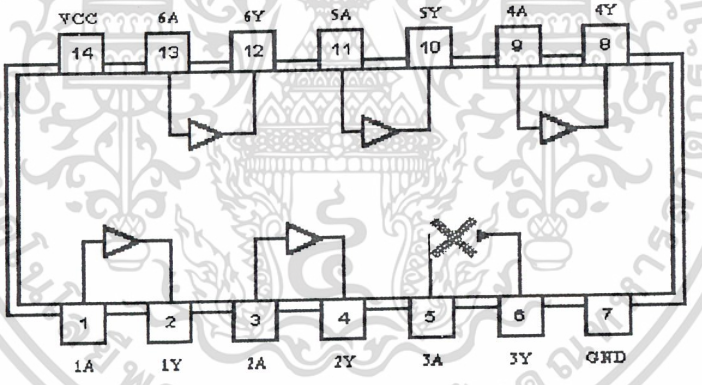
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.2 กรณีที่มีฟังก์ชันการทำงานภายในไอซีที่ผิดพลาด

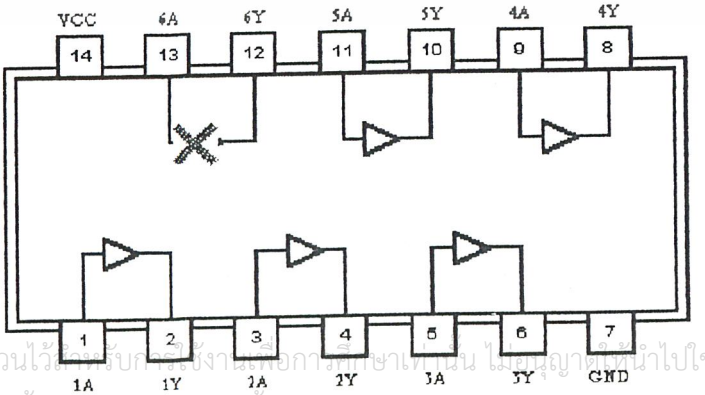
- ตัดขาที่ 3 ของไอซี ได้ผลดังนี้



- ตัดขาที่ 6 ของไอซี ได้ผลดังนี้



- ตัดขาที่ 13 ของไอซี ได้ผลดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุป

7.1 หลักการของเครื่องทดสอบไอซีทีทีแอล

เมื่อต้องการตรวจสอบไอซีทีทีแอล ผู้ใช้จะสั่งงานได้โดยการติดต่อที่หน้าจอคอมพิวเตอร์ ส่วนการทำงานที่ใช้ในการตรวจสอบไอซีนั้นจะเป็นหน้าที่ของไมโครคอนโทรลเลอร์ ซึ่งจะเป็นตัวที่ติดต่อสื่อสารกับคอมพิวเตอร์ โดยผ่านทางพอร์ตอนุกรม เมื่อคอมพิวเตอร์สั่งให้ตรวจสอบไอซี ก็จะส่งเป็นรหัสไปยังไมโครคอนโทรลเลอร์ จากนั้นส่วนของไมโครคอนโทรลเลอร์นั้นจะทำหน้าที่ตรวจสอบ และเมื่อได้ผลลัพธ์มาแล้วก็จะส่งค่าคืนไปให้คอมพิวเตอร์ ซึ่งคอมพิวเตอร์ก็จะนำไปประมวลผล และแสดงผลต่อไป จะเห็นว่าจากการทำงานนี้จะแบ่งการทำงานของเครื่องตรวจสอบไอซีทีทีแอลเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนของไมโครคอนโทรลเลอร์และส่วนของ ไมโครคอมพิวเตอร์

ส่วนของไมโครคอนโทรลเลอร์

ในการตรวจสอบไอซีทีทีแอล ใช้ 8255 ร่วมด้วยในการตรวจสอบ เนื่องจากพอร์ตของไมโครคอนโทรลเลอร์นั้นมีไม่มากเพียงพอ และ 8255 นี้ก็ยังแบ่งการทำงานออกเป็นอีกหลายส่วนด้วยกัน คือ

- 8255 ตัวที่1 ทำหน้าที่เป็นตัวควบคุมการเข้า-ออก ของข้อมูลที่แต่ละขาของไอซีที่ทดสอบ
- 8255 ตัวที่ 2 ทำหน้าที่ จ่ายข้อมูลเข้าไปยังขาของไอซีที่ทำการทดสอบ
- 8255 ตัวที่ 3 ทำหน้าที่ รับข้อมูล Output จากขาของไอซีที่ทำการทดสอบ
- 8255 ตัวที่ 4 ทำหน้าที่ ควบคุมการตัดต่อ R-Pull Up เข้ากับขาของไอซีที่ทำการทดสอบ

ข้อมูลที่จะสั่งการให้ 8255 แต่ละตัวทำงานอย่างไร จะส่งมาจาก ไมโครคอมพิวเตอร์ และหลังจากที่ได้ค่าของ Output มาทาง 8255 ตัวที่รับ Output มาแล้ว ก็ทำการส่งผลลัพธ์นี้กลับไปให้ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ ไมโครคอมพิวเตอร์

การทำงานที่ไมโครคอมพิวเตอร์ เริ่มที่ทำการติดต่อกับผู้ใช้ โดยผู้ใช้สามารถที่จะเลือกเบอร์ของไอซีที่ทำการทดสอบ เมื่อผู้ใช้เลือกเบอร์ไอซีแล้ว โปรแกรมก็จะทำการประมวลผลจัดเตรียมข้อมูลของไอซีเพื่อที่จะส่งไปให้กับไมโครคอนโทรลเลอร์ การทำงานจัดเตรียมข้อมูลนี้ จะทำงานร่วมกับฐานข้อมูลของไอซี ที่จัดเก็บไว้ในไมโครคอมพิวเตอร์ หลังจากทำการเตรียมข้อมูลเรียบร้อยก็จะทำการส่งข้อมูลนี้ไปให้กับ ไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ทำการตรวจสอบและส่งผลลัพธ์กลับมาแล้ว ก็นำผลลัพธ์ที่ส่งกลับมานี้ มาประมวลผลว่ามีฟังก์ชันตัวไหนภายในไอซีที่ใช้งานไม่ได้บ้าง จากนั้นก็ทำการแสดงผลให้ผู้ใช้งานทราบ

7.2 ผลการทำงานของเครื่องตรวจสอบไอซี ทีทีแอล

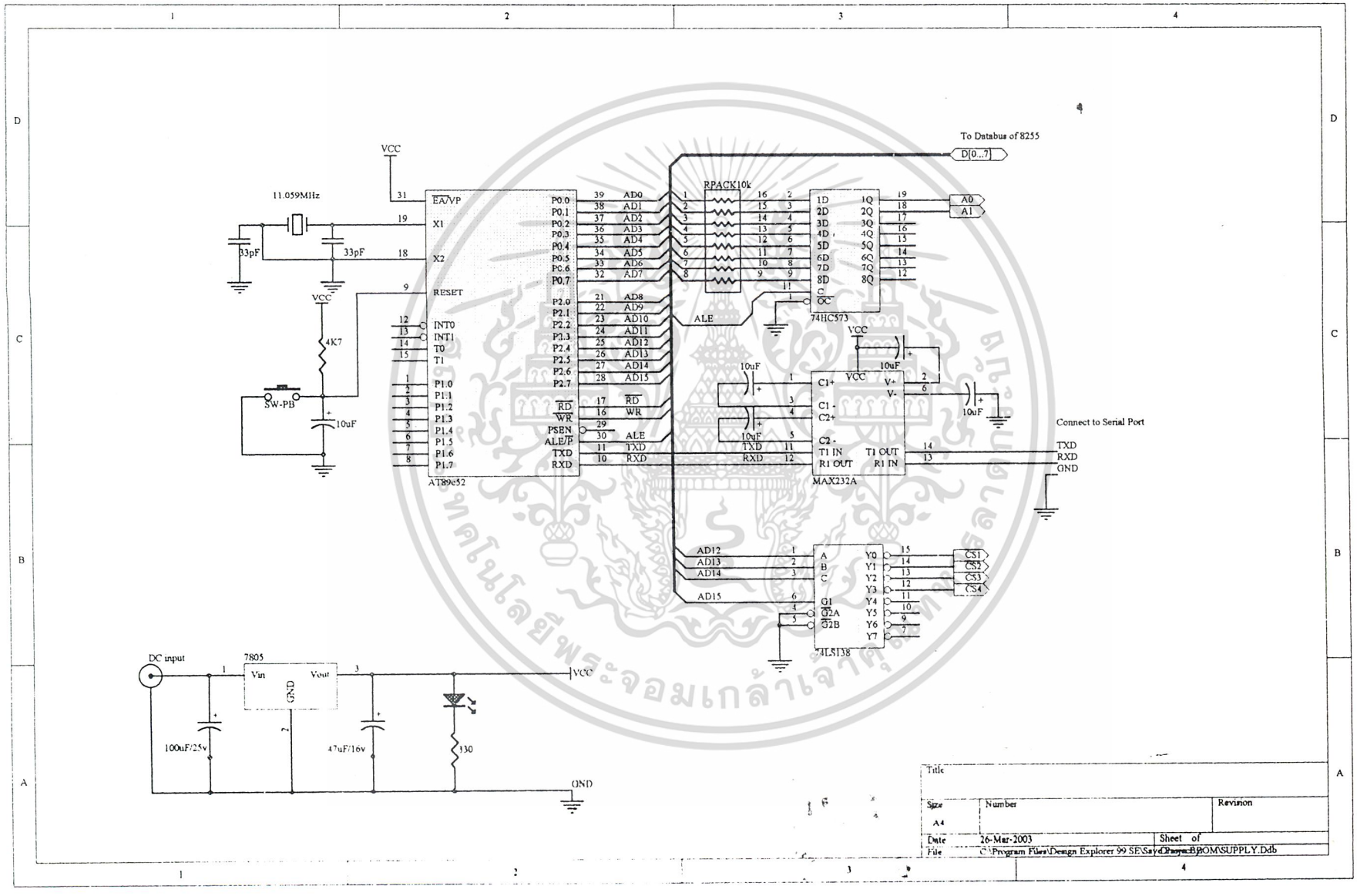
ในการทดสอบการทำงานจากผลการทดลองของเครื่องตรวจสอบไอซี ทีทีแอลนี้ สรุปได้ว่าสามารถที่จะทำการตรวจสอบไอซีที่เป็นแบบ Basic Gate, ไอซีประเภท เข้ารหัส และ ถอดรหัส, ไอซีประเภทที่ใช้สัญญาณนาฬิกา และไอซีประเภท Open Collector ได้อย่างถูกต้อง โดยมีตัวอย่างไอซีที่นำมาทดสอบ เช่น 7400, 7401, 7402, 7404, 7405, 7407, 7408, 7409, 7410,7411,7412, 7420, 7421, 7422, 7432, 7473, 7474, 7490, 74138, 74139, 74192 เป็นต้น

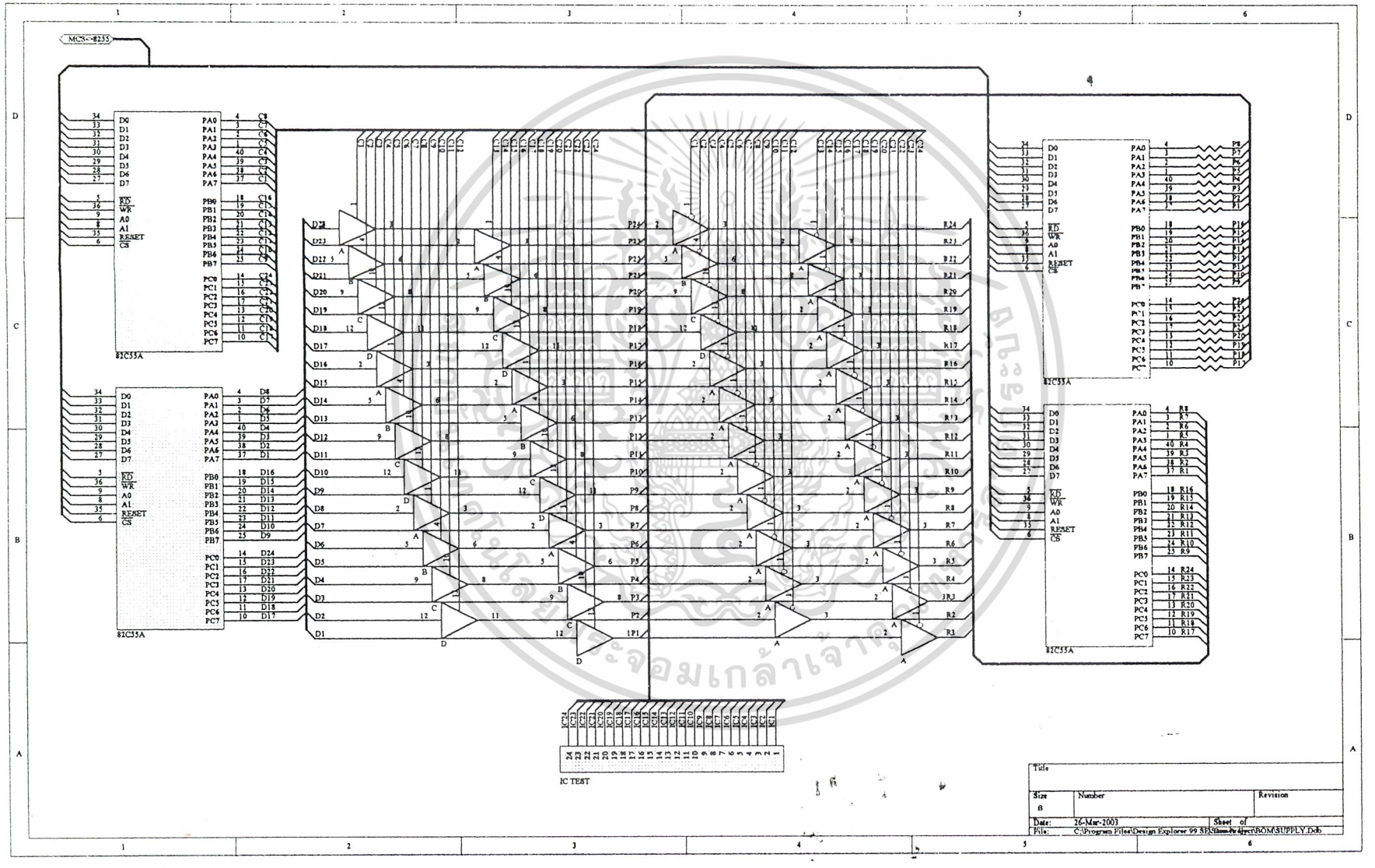
การที่จะเพิ่มจำนวนไอซีที่สามารถทดสอบได้ให้มากขึ้น ก็ทำได้โดยการเพิ่มข้อมูลต่าง ๆ ที่สำคัญของไอซีที่ต้องการจะทดสอบเข้าไปที่ไมโครคอมพิวเตอร์ซึ่งใช้เป็นฐานข้อมูล เช่น ค่าข้อมูลอินพุท,เอาท์พุทและรูปการจัดเรียงขาของไอซีเบอร์นั้น ๆ ที่จะนำมาใช้ในส่วนของการแสดงผล เป็นต้น และในส่วนของไมโครคอนโทรลเลอร์นั้น ไม่จำเป็นต้องทำการเพิ่มเติมใดๆอีก เพราะวิธีการที่ใช้ในการตรวจสอบไอซีนั้นจะเป็นหลักการเดียวกันในทุก ๆ เบอร์ที่นำมาตรวจสอบไม่ว่าจะเป็น Basic Gate, multiplex หรืออื่น ๆ โดยจะช่วยให้ประหยัดเนื้อที่หน่วยความจำของไมโครคอนโทรลเลอร์ เราจึงไม่จำเป็นต้องต่อหน่วยความจำภายนอกเพิ่มเติมให้ยุ่งยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


To Databus of 8255
D[0...7]
VCC
11.059MHz
33pF
33pF
4K7
SW-PB
10uF
EA/VP
X1
X2
RESET
INT0
INT1
T0
T1
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
P2.7
RD
WR
PSEN
ALE/P
TXD
RXD
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
AD8
AD9
AD10
AD11
AD12
AD13
AD14
AD15
AT89c52
RPACK10k
74HC573
ALE
A0
A1
10uF
C1+
C1 -
C2+
C2 -
T1 IN
R1 OUT
T1 OUT
R1 IN
V+
V-
MAX232A
Connect to Serial Port
TXD
RXD
GND
A
B
C
G1
G2A
G2B
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
74LS138
CS1
CS2
CS3
CS4
DC input
7805
Vin
GND
Vout
100uF/25v
47uF/16v
330
Title
Size
A4
Number
Revision
Date
26-Mar-2003
Sheet of
File
C:\Program Files\Design Explorer 99 SE\Sayed\Prayut\BROM\SUPPLY.Ddb


MCS-~8255
82C55A
82C55A
82C55A
82C55A
IC TEST
Title
Size
B
Number
Revision
Date: 26-Mar-2003
Sheet of
File: C:\Program Files\Design Explorer 99 SE\Examples\ISOM\SUPPLY.Ddb

เอกสารอ้างอิง

1. ชูชัย ธนสารตั้งเจริญ และคณะ, "การใช้งาน Z-80", หจก.สำนักพิมพ์ฟิสิกส์เซนเตอร์
2. สุนทร วิทูอินสุรพจน์, "การใช้งานโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล8051",บริษัท ซีเอ็ดยูเคชั่น จำกัด มหาชน, 198 หน้า,2537
3. "คู่มือเทียบเบอร์ไอซีทีทีแอล",บริษัท ซีเอ็ดยูเคชั่น จำกัด มหาชน
4. รัชนัย อินทุใส, ไตรภพ อินทุใส, "ไมโครคอนโทรลเลอร์ 8051", หจก.สำนักพิมพ์ฟิสิกส์เซนเตอร์
5. กฤษดา วิศวธีรานนท์, "เรียน/เล่น/ใช้ไอซีดิจิตอล", บริษัท ซีเอ็ดยูเคชั่น จำกัด มหาชน, 303 หน้า,2537
6. Digital Integrated Circuits/Thomas A.Demassa, Zack Cicone, John Wiley & Sons Inc., Print in USA.,1996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้