ปริญญานิพนธ์

เรื่อง การส่งสัญญาณควบคุมระยะไกล โดยใช้สาย ๒ เส้น

TWO WIRE REMOTE CONTROL SYSTEM

๑. นายธวัชชัย กลมเกลียว

๒. นายศิริรจน์ รัตน์ประสิทธิ์

อาจารย์ที่ปรึกษา

อ.สุพรรณ กุลพาณิชย์

ADVISOR

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมบัณฑิต

สาขา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๓๕

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032594

ภาควิชา เทคโนโลยีวัดคุมทางอุตสาหกรรม

สาขา เทคโนโลยีวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมระยะไกลด้วยสายคู่

TWO WIRE REMOTE CONTROL SYSTEM

ผู้จัดทำ

1. นาย ศิริรงน์ รัตน์ประสิทธิ์ 34161230
2. นาย ธวัชชัย กลมเกลียว 34161107

.....อาจารย์ที่ปรึกษา

(อ.สุพรรณ กุลพานิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032594

ปริญญานิพนธ์

เรื่อง การส่งสัญญาณควบคุมระยะไกล โดยใช้สาย ๒ เส้น

TWO WIRE REMOTE CONTROL SYSTEM

จุดประสงค์

1. เพื่อศึกษาหลักการสื่อสาร ของอุปกรณ์รับส่งข้อมูลระยะไกล โดยไม่ต้องใช้ MODEM
2. เพื่อออกแบบอุปกรณ์ ที่มีความสามารถในการควบคุม การติดต่อสื่อสารของระบบเครือข่ายสื่อสารข้อมูลระหว่างเครื่องส่งและเครื่องรับ
3. เพื่อสร้างอุปกรณ์รับและส่งข้อมูลระยะไกลโดยใช้สาย 2 เส้น
4. เพื่อแสดงผลของการควบคุมโหลด โดยใช้ระบบ TWO WIRE REMOTE
5. เพื่อสามารถประยุกต์วงจร TWO WIRE REMOTE ซึ่งใช้ในโรงงานอุตสาหกรรมขนาดใหญ่ ให้สามารถนำมาใช้ในโรงงานขนาดเล็ก และเผยแพร่ให้กับหน่วยงานที่สนใจ

ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้ทราบการทำงานของระบบ TWO WIRE REMOTE CONTROL
2. นำไปใช้ในการควบคุมระบบไฟฟ้าในโรงงานอุตสาหกรรม
3. ช่วยในการลดจำนวนสายที่ใช้ในการควบคุมเครื่องใช้ไฟฟ้า
4. ลดค่าใช้จ่ายในการใช้ไฟฟ้า
5. เผยแพร่ผลงานให้แก่หน่วยงานต่างๆ ที่สนใจในระบบ TWO WIRE REMOTE CONTROL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งสัญญาณควบคุมระยะไกล โดยใช้สาย ๒ เส้น

นายธวัชชัย กลมเกลียว

นายศิโรจน์ รัตน์ประสิทธิ์

อาจารย์ที่ปรึกษา

อ.สุพรรณ กุลพาณิชย์

ปีการศึกษา ๒๕๓๕

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำเสนอถึงการรับส่งข้อมูลระยะไกลโดยใช้สาย 2 เส้น ซึ่งใช้ในการควบคุมโหลดจากศูนย์ปฏิบัติกลาง สามารถตั้งเวลาและเปิดปิดโหลดจากศูนย์กลางได้ตามความต้องการ ในการรับและส่งข้อมูลใช้มาตรฐานการสื่อสารข้อมูลอนุกรมแบบ RS-485 ซึ่งให้ผลดีกว่ามาตรฐานการสื่อสารอนุกรมแบบ RS-232 โดยสามารถส่งข้อมูลได้ในระยะทางไกล มีความเร็วในการส่งสูง สามารถใช้งานในสิ่งแวดล้อมที่มีสัญญาณรบกวนสูงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TWO WIRE REMOTE CONTROL SYSTEM

BY

MR.TAWATCHAI KLOMKLEAL

MR.SIROT RATPRASIT

ADVISOR

MR.SUPHAN KULAPANIT

YEAR 1992

ABSTRACT

THIS THESIS IS INTRODUCE ABOUT TO THE TWO WIRE REMOTE CONTROL SYSTEM. IT WILL CONTROL LOAD AND SETTING TIMER FROM CENTER.

IN COMMUNICATION USED TO EIA STANDARD RS-485 BETTER THEN EIA STANDARD RS-232. FROM EIA STANDARD RS-485 DESIGNED FOR MULTIPOINT TRANSMISSION ON LONG BUS LINES IN NOISY ENVIRONMENTS.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

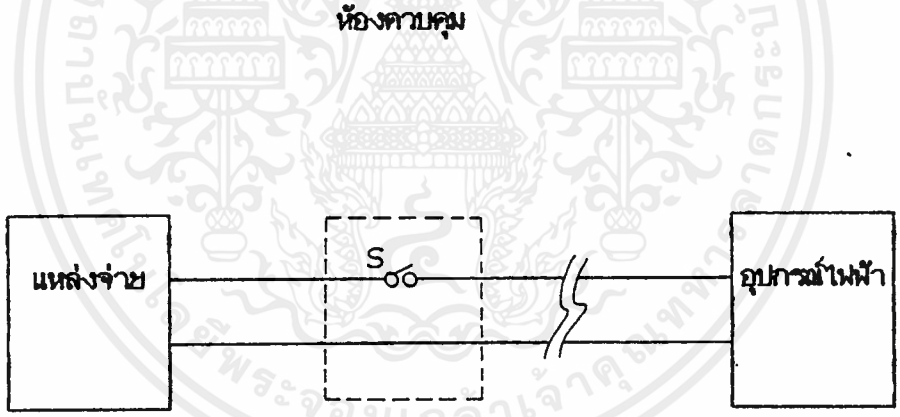
หัวข้อ	หน้า
บทที่ 1	
-พื้นฐานการใช้งานระบบ TWO WIRE	1
บทที่ 2	
-ทฤษฎีเบื้องต้นในการสื่อสารข้อมูลแบบอนุกรม	3
-มาตรฐาน RS-232-C	6
-มาตรฐาน RS-423	17
-มาตรฐาน RS-422	19
-มาตรฐาน RS-485	20
-Z-80 ไมโครโปรเซสเซอร์	35
-หน่วยอินพุทและเอ้าท์พุท	42
-การใช้ 8255 PIAกับZ-80	54
บทที่ 3 การออกแบบวงจรและการสร้าง	63
-ด้านส่ง	63
-ด้านรับ	70
บทที่ 4 การทดลองและผลการทดลอง	
บทที่ 5 วิจารณ์และสรุปผล	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

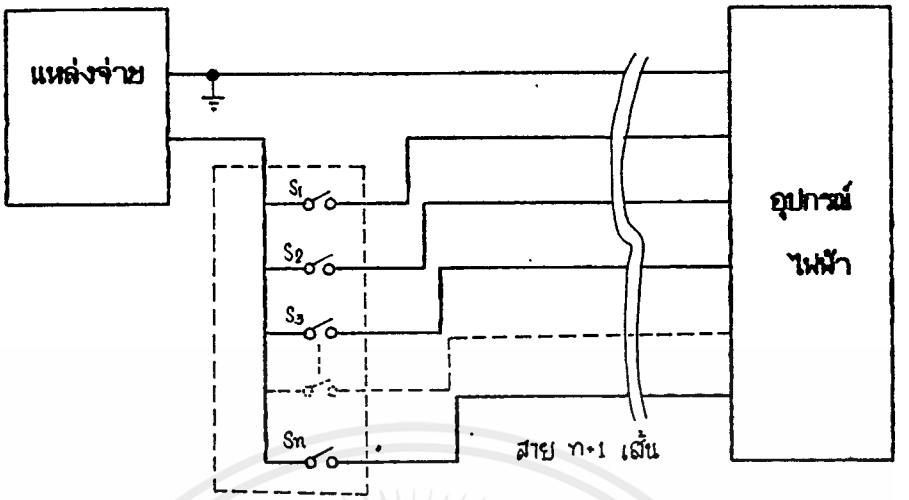
ปัจจุบันตามโรงงานอุตสาหกรรม นิยมใช้การควบคุมการทำงานตามจุดต่างๆภายในหน่วยงาน จากศูนย์ปฏิบัติกลาง เพื่อความรวดเร็วในการตรวจสอบและสั่งงาน

ในอดีตการปิด-เปิดอุปกรณ์ไฟฟ้าที่ง่ายที่สุด คือ สวิทช์ในรูป ก. ซึ่งเป็นการปิด-เปิดอุปกรณ์เพียงหนึ่งตัว แต่เมื่อต้องการปิด-เปิดอุปกรณ์มากกว่าหนึ่งตัว โดยมีศูนย์กลางการควบคุมเพียงแห่งเดียว สามารถทำได้ดังรูป ข. จะเห็นว่าเมื่อต้องการควบคุมอุปกรณ์ n ตัว จะต้องมีการเดินสายทั้งหมด n+1 เส้น จากห้องควบคุมไปยังตำแหน่งโหลด ซึ่งจะเป็นการสิ้นเปลืองค่าสายไฟเป็นอย่างมาก โดยเฉพาะเมื่ออุปกรณ์ไฟฟ้ากินกระแสมาก (ใช้สายเส้นใหญ่) ระบบ Two Wire Remote จึงถูกวิจัยขึ้นเพื่อแก้ปัญหานี้



ก. การควบคุมอุปกรณ์เพียงตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข. การควบคุมอุปกรณ์ไฟฟ้า n ตัว

วัตถุประสงค์ที่สำคัญอีกประการหนึ่งในการทำปริญญานิพนธ์ก็คือการประยุกต์ ระบบ Two Wire Remote ที่ใช้ในโรงงานอุตสาหกรรมขนาดใหญ่ ซึ่งมีราคาสูงมาให้ถูกลงและสามารถนำมาใช้ในหน่วยงานขนาดเล็กได้ ในด้านการประหยัดพลังงาน ระบบ Two Wire Remote ทำให้ประหยัดพลังงานได้โดยสามารถตั้งเวลาปิด-เปิดอุปกรณ์ไฟฟ้าได้ตามต้องการ

ระบบ Two Wire Remote เป็นระบบการสื่อสารข้อมูลแบบอนุกรมที่ใช้สายเพียง 2 สาย ควบคุมการส่ง-รับข้อมูลโดย Micro Processor ซึ่งในส่วนของ Line Driver ใช้มาตรฐานการส่งข้อมูลแบบ EIA Standard RS-485 ที่มีข้อดีกว่า RS-232 ทั้งในด้านอัตราในการส่งข้อมูลที่สูงกว่า และระยะทางในการส่งข้อมูลที่ไกลกว่า โดยไม่เกิดการผิดเพี้ยนของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎี หลักการ และงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีเบื้องต้นในการสื่อสารข้อมูลแบบอนุกรม

- การสื่อสารข้อมูลแบบขนาน

การสื่อสารแบบขนานก็คือข้อมูลทุกๆ บิทในแต่ละเวิร์ดจะถูกส่งออกไปพร้อมๆกันขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร โดยทั่วไปก็คือ 1 ไบท์ หรือ 8 บิท นั่นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อจำกัดทางด้านระยะทางระหว่างต้นทางและปลายทาง โดยทั่วไปจะส่งในระยะทางไม่เกิน 3-5 ฟุตเท่านั้น ทั้งนี้ขึ้นอยู่กับอัตราที่ใช้ในการส่งข้อมูลสูงก็จะส่งได้ระยะสั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูงมากๆ แต่อุปกรณ์อยู่ไม่ห่างกันมากนัก

- การสื่อสารข้อมูลแบบอนุกรม

สำหรับการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกทยอยส่งออกไปทีละบิทจนครบทั้งเวิร์ดในสายสัญญาณเพียงเส้นเดียว แต่ในการใช้งานจริงจะต้องมีสายสัญญาณอีกเส้นเป็นระดับ Ground ดังนั้นเมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่การส่งข้อมูลแบบขนาน จะต้องใช้อย่างน้อยเท่ากับจำนวนบิทบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น นอกจากนี้การส่งข้อมูลแบบอนุกรมนั้น จะสามารถส่งข้อมูลได้ไกลกว่ามาก เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่จะกล่าวต่อไปในภายหลังจะสามารถส่งได้ไกล 30-40 ฟุตโดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่อย่างใด อย่างไรก็ดีการส่งข้อมูลแบบอนุกรมนั้น จะต้องมีส่วนที่ทำหน้าที่แปลงข้อมูลจากข้อมูลแบบขนานมาเป็นแบบอนุกรม ซึ่งสามารถเป็นได้ทั้งฮาร์ดแวร์และซอฟต์แวร์ และในการส่งยังมีข้อกำหนดบางประการเพื่อให้การส่งข้อมูลมีความถูกต้องมากยิ่งขึ้นอีกด้วย

- SIMPLEX และ DUPLEX

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือการสื่อสารทั่วไปนั้น ย่อมจะต้องประกอบด้วยผู้รับและผู้ส่ง ผู้รับในขณะนี้อาจสามารถเป็นผู้ส่งในอนาคตได้ แต่มีบางการมีสำหรับการสื่อสารข้อมูลที่ผู้รับและผู้ส่งจะตายตัวอยู่ตลอดเวลา เช่นการสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบ Simplex กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลา ซึ่งจะมีที่ใช้น้อยมาก การสื่อสารโดยทั่วไปนั้นจะเป็นลักษณะ Duplex คือมีทิศทางการสื่อสาร 2 ทิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางทั้งไปและกลับ การสื่อสารในลักษณะ Duplex นี้ยังแบ่งออกได้เป็น 2 ชนิด คือ Half Duplex (นิยมเขียนย่อว่า HDX) ซึ่งจะมีทิศทางการสื่อสารที่สลับกันเป็นผู้รับและผู้ส่งในเวลาเดียวกัน เราอาจเปรียบเทียบเพื่อให้เห็นภาพชัดเจนยิ่งขึ้น โดยเปรียบเทียบการสื่อสารแบบ HDX หรือ Half Duplex เป็นทางรถไฟ ซึ่งจะมีรถไฟเพียงขบวนเดียวเท่านั้นที่วิ่งอยู่บนรางในเวลาหนึ่ง และเปรียบเทียบ FDX หรือ Full Duplex เป็นถนนที่รถสามารถวิ่งสวนกันได้ในเวลาเดียวกัน การสื่อสารระหว่างคอมพิวเตอร์มักจะอยู่ในลักษณะของ Duplex แบบใดแบบหนึ่ง

– โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมให้ดีจะพบว่า ปัญหาหนึ่งที่ควรจะเกิดขึ้นอยู่เสมอก็คือ การตัดสินว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้นจึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าวเราเรียกว่า โปรโตคอล (Protocal) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ โปรโตคอลสำหรับการสื่อสารข้อมูลอนุกรมแบบซิงโครนัส (Synchronous) และโปรโตคอลสำหรับการสื่อสารข้อมูลแบบอะซิงโครนัส (Asynchronous) การสื่อสารแบบซิงโครนัสนั้น ข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอ ช่วงเวลาระหว่างบิทและระหว่างเวิร์ดจะมีค่าเท่ากันเสมอ ดังนั้นในการสื่อสารข้อมูลอนุกรมในแบบซิงโครนัส จึงต้องมีสายสัญญาณเพิ่มเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใดและควรจะหยุดเมื่อใด ระบบที่เป็นซิงโครนัสจะเป็นระบบที่มีความเร็วสูง แต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน

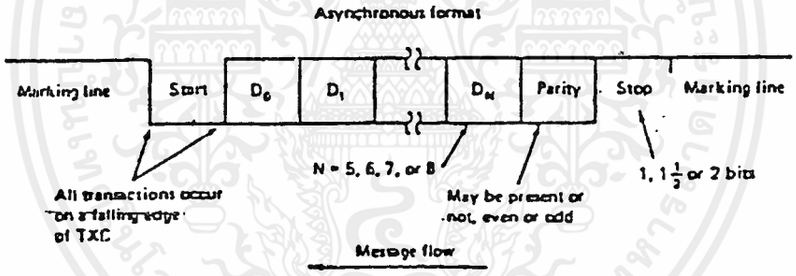
การสื่อสารแบบอซิงโครนัส เป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ในปัจจุบัน การสื่อสารแบบนี้ช่วงระยะเวลาระหว่างบิทจะมีค่าเท่ากันเช่นเดียวกับซิงโครนัส แต่จะแตกต่างกันที่ระยะห่างระหว่างเวิร์ด ซึ่งไม่มีข้อกำหนดว่าจะห่างกันเป็นระยะเวลานานเท่าใด กล่าวคือ ระหว่างเวิร์ดนั้น จะห่างกันกี่วินาที กี่นาที กี่ชั่วโมง กี่วัน กี่เดือน หรือ กี่ปี ก็ได้ทั้งสิ้นขึ้นอยู่กับว่าทางฝ่ายรับสามารถรอได้หรือไม่เท่านั้น

เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจตรงกันได้อย่างไรว่าที่ใดคือจุดเริ่มต้นและจุดสิ้นสุดของแต่ละเวิร์ด เพื่อแก้ปัญหานี้ จึงมีการกำหนดข้อตกลงเกี่ยวกับฟอร์แมทของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดเป็นจุดเริ่มต้นของเวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิทที่เรียกว่า Start Bit หรือ บิทเริ่มต้น ซึ่งจะต้องเป็นลอจิกศูนย์เสมอจากนั้นตามด้วยบิทข้อมูลที่ต้องการส่ง ซึ่งมีความยาว 5 ถึง 8 บิท ถัดจากบิทข้อมูลก็จะเป็นพาริตีบิท ซึ่งทำหน้าที่เป็นบิทสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ บิทพาริตีนี้มี 2 ประเภท คือ Even Parity ซึ่งจะเซ็ท (มีค่าเป็น 1) เมื่อจำนวนบิทที่เป็นลอจิก 1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคี่ และ Odd Parity ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเช็คเมื่อจำนวนบิทที่เป็นลอจิก 1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคู่ ในการส่งข้อมูล บางครั้งอาจจะไม่มีการใช้บิทพาริตีก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูง คือมีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการสื่อสารได้ด้วย บิทสุดท้ายในฟอร์แมทก็คือ Stop Bit ทำหน้าที่บอกทางผู้รับว่าขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้ว ขอให้เตรียมตัวรับเวิร์ดต่อไปได้ Stop Bit นี้ถูกกำหนดให้เป็นลอจิก 1 เสมอ ทั้งนี้เพื่อให้ระบบสามารถตรวจสอบบิทเริ่มต้นได้ (บิทเริ่มต้นมีลอจิกเป็น 0) Stop Bit นี้อาจมีความยาวเป็น 1 บิท 1.5 บิทหรือ 2 บิทก็ได้

จากฟอร์แมทดังกล่าว จะเห็นว่ามีฟอร์แมทสำหรับการสื่อสารมากมายไปหมด เช่น 5E1 (5 Data Bit, Even Parity, 1 Stop bit), 7E1 (7 Data Bit, Even Parity, 1 Stop Bit) และ 8N1 (8 Data Bit, No Parity, 1 Stop Bit) เป็นต้น ในการใช้งานทั่วไป เรานิยมใช้กันอยู่เพียง 2 ฟอร์แมท คือ 7E1 และ 8N1 จะเลือกใช้ฟอร์แมทใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำควรใช้ 8N1 ทั้งนี้จะต้องมีการตกลงกันล่วงหน้าระหว่างผู้รับและผู้ส่งว่าจะใช้ฟอร์แมทใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป



รูปที่ 1

- ทำให้เกิด Crosstalk ที่มีค่ามาก

ตามมาตรฐาน RS-232-C ที่ถูกตีพิมพ์โดย EIA ได้กล่าวถึงการสื่อสารข้อมูลระหว่าง Data Terminal Equpiment (DTE) และ Data Communication Equipment (DCE) (แต่ในปัจจุบันตัวย่อ DCE จะแทน Data Circuit Termination Equipment) คำจำกัดความของ DCE และ DTE ซึ่งแสดงไว้ข้างล่างได้คัดมาจากคำแปลศัพท์ในหนังสือ "Technical Aspect of Data Communication" ซึ่งเขียนโดย John Mcnamara (Digital press, 1977) ดังนี้

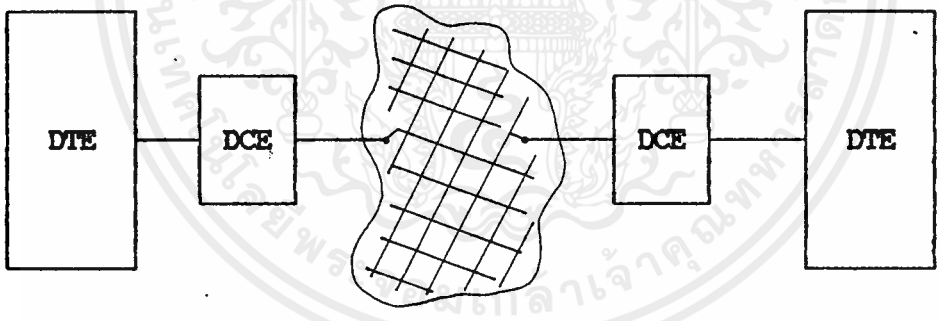
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DCE : อุปกรณ์ที่มีฟังก์ชั่นการทำงานต่างๆที่ทำให้เกิดการเชื่อมต่อ ทำให้การเชื่อมต่อยังดำรงต่อไปและยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่างๆที่จำเป็นต้องใช้ในการสื่อสารข้อมูลระหว่าง DTE และ Data Circuit โดย DCE อาจเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์หรือไม่ก็ได้

DTE : - เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (Data Source) หรือตัวรับข้อมูล (Data Sink) หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้

- เป็นอุปกรณ์ที่ประกอบด้วย Function Unit ต่อไปนี้ Control Logic, Buffer Store และอุปกรณ์อินพุทหรือเอาท์พุทจำนวนหนึ่งตัวหรือมากกว่าก็ได้ หรือรวมเครื่องคอมพิวเตอร์เข้าไปด้วยก็ได้ DTE อาจจะรวมส่วน Error Control, Synchronization และความสามารถในการบ่งหรือระบุว่าต้องการเกี่ยวข้องกับอุปกรณ์ตัวใด (Station Identification Capability) เข้าไปด้วยก็ได้

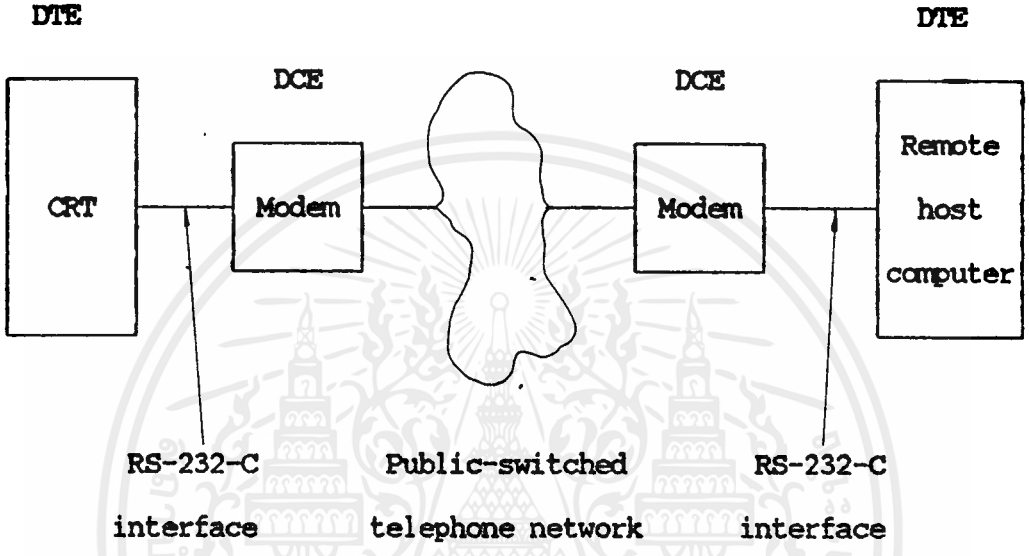
ลักษณะของ DCE และ DTE ที่ใช้ในการสื่อสารข้อมูลได้แสดงไว้ในรูปที่ 3 ตามลักษณะการทำงานที่ได้อธิบายไว้ข้างต้น



รูปที่ 3 ลักษณะของ DCE และ DTE ที่ใช้ในวงการสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในความเป็นจริงแล้ว DTE มักจะแทนแหล่งกำเนิดข้อมูลแหล่งแรกและหรืออุปกรณ์ที่เป็นแหล่งรับข้อมูลแหล่งสุดท้าย (ดังแสดงในรูปที่ 3) เช่น เครื่องพิมพ์ (เป็นอุปกรณ์ที่รับข้อมูลได้เพียงอย่างเดียว) จะเป็น DTE เพราะเป็นอุปกรณ์ที่รับข้อมูลเป็นตัวสุดท้าย หรือ CRT/คีย์บอร์ด เป็นทั้งตัวรับข้อมูลและตัวกำเนิดข้อมูล ส่วน DCE เป็นอุปกรณ์ที่ทำให้การสื่อสารข้อมูลระหว่างแหล่งกำเนิดกับตัวรับข้อมูลปลายทางทำได้สะดวกขึ้น ตัวอย่างหนึ่งของ DCE ก็คือ โมเด็ม



รูปที่ 4 ลักษณะของระบบที่ใช้แสดงเป็นตัวอย่าง

ถ้าเทอร์มินัลและคอมพิวเตอร์ของเราเป็น DCE และ DTE ทั้งคู่เราจะทำการสื่อสารข้อมูลได้อย่างไร ปัญหาที่เกิดขึ้นนี้เราแก้ไขได้โดยใช้เคเบิล (CABLE) ที่เรียกว่า "Null Modem"

1. Null หมายความว่าอุปกรณ์ตัวนี้ไม่สามารถทำงานอะไรได้ เราใช้อุปกรณ์ตัวนี้เมื่อต้องการเปลี่ยนทิศทางการเคลื่อนที่ของข้อมูลเท่านั้น

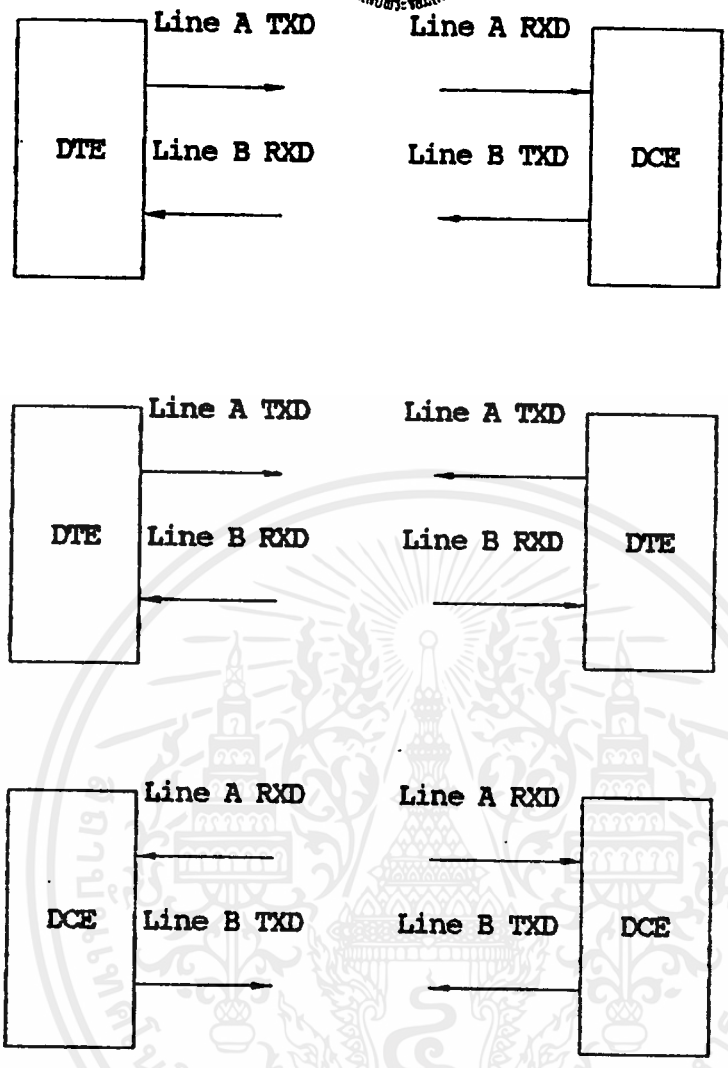
2. Modem แสดงว่าอุปกรณ์ตัวนี้เป็น DCE ด้วยเหตุนี้เราจึงใช้ Null Modem แทรกเข้าไประหว่าง DTE สองตัว เพื่อทำให้เราสามารถทำการสื่อสารข้อมูลโดยผ่าน RS-232-C ได้

สำหรับลักษณะของสาย (Line) ที่ใช้ในการรับและส่งข้อมูลของ DTE และ DCE เป็นดังนี้

1. สายที่ใช้ในการส่งและรับข้อมูลของ DTE และ DCE มีอยู่สองเส้น แต่ละเส้นจะมีทิศทางการเคลื่อนของข้อมูลกำหนดได้ต่างทิศทางกัน

2. DTE จะส่งข้อมูลทาง Line A และ DCE จะรับข้อมูลทาง Line(A เช่นเดียวกัน DCE จะส่งข้อมูลทาง Line B และ DTE รับข้อมูลทาง Line B ดังแสดงในรูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

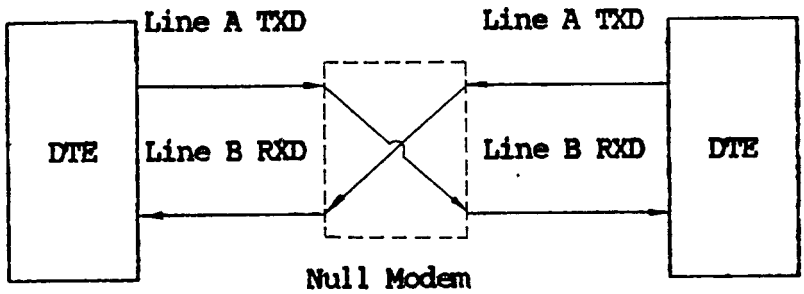


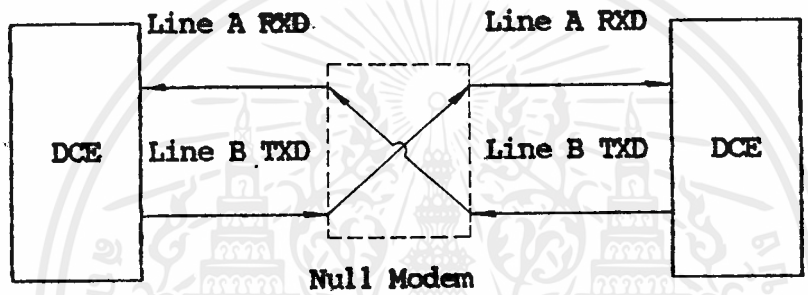
รูปที่ 5 ลักษณะการส่งและรับข้อมูลของ DCE และ DTE

จากรูปที่ 4 ถ้า DTE 2 ตัว ทำการแลกเปลี่ยนข้อมูลกัน ข้อมูลจะถูกส่งออกทาง Line A และรับทาง Line B ทั้งคู่ ดังนั้นการสื่อสารข้อมูลจะไม่สามารถเกิดขึ้นได้สำหรับการมีที่ DCE 2 ตัว ทำการแลกเปลี่ยนข้อมูลก็เป็นเช่นเดียวกัน เมื่อเราต่อ Line A เข้าด้วยกัน และต่อ Line B เข้าด้วยกัน ข้อมูลใน Line B จะต้านกันเอง ส่วนใน Line A ไม่มีข้อมูลที่จะรับ ปัญหานี้แก้ได้โดยใช้ Null Modem เข้าช่วย Null Modem Cable จะทำการไขว้ Line A เข้ากับ Line B ดังรูปที่ 6 ดังนั้น DTE 1 สามารถรับข้อมูลที่ส่งจาก DTE 2 ได้ และ DTE 2 ก็สามารถรับข้อมูลจาก DTE 1 ได้เช่นกัน

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032594





รูปที่ 6 ลักษณะการทำงานของ Null Modem ที่ใช้ในการอินเตอร์เฟส ระหว่าง DTE หรือ DCE สองตัว

เราใช้มาตรฐาน RS-232-C ในการสื่อสารข้อมูลแบบอนุกรมระหว่าง DCE กับ DTE โดยอัตราการส่งข้อมูลจะถูกกำหนดให้อยู่ระหว่าง 0 ถึง 20000 บิทต่อวินาที ในการประยุกต์ใช้งาน RS-232-C อัตราเร็วสูงสุดที่ใช้ควรจะมีค่าไม่เกิน 19.2 Kbps

มาตรฐานนี้ได้กำหนดความยาวของสายเคเบิลที่ใช้ในการสื่อสารข้อมูลไว้ไม่เกิน 50 ฟุต (ไม่ใช่ข้อกำหนดที่ตายตัว เนื่องจากระยะ 50 ฟุต นี้ได้มาจากประสบการณ์) เคเบิลอาจจะยาวกว่า 50 ฟุตก็ได้ เรารู้สภาพแวดล้อมของสายเคเบิลและอยู่ในเงื่อนไขที่ถูกกำหนดไว้ในมาตรฐาน

EIA ได้ระบุไว้ว่า เราไม่ควรใช้ RS-232-C ในการที่ต้องการให้มีการแบ่งแยกทางไฟฟ้า (Electrical Isolation) ระหว่างอุปกรณ์ทั้งสองด้านของการอินเตอร์เฟส ซึ่งค่าเดือนนี้เป็นสิ่งที่สำคัญที่เราต้องจำไว้เสมอถ้าเราคิดจะใช้ RS-232-C ในการอินเตอร์เฟสเครื่องคอมพิวเตอร์ของเราเข้ากับอุปกรณ์ที่สร้างขึ้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ของการใช้ RS-232-C คือใช้ในการอินเตอร์เฟสระหว่าง DTE กับ DCE ในกรณีที่ต้องการสื่อสารข้อมูลในระยะทางไกล ๆ โดยผ่านทางเครือข่ายโทรศัพท์ แต่อย่างไรก็ตาม RS-232-C ก็ยังใช้ในการสื่อสารข้อมูลในระยะทางใกล้ ๆ เช่น ใช้ระหว่างคอมพิวเตอร์กับเทอร์มินัล คอมพิวเตอร์กับเครื่องพิมพ์ และคอมพิวเตอร์กับดิสเก็ตไดรฟ์

คุณสมบัติของสัญญาณทางไฟฟ้า

สัญญาณที่ขาทุกขาที่คอนเน็คเตอร์ของ RS-232-Cจ จะเป็นสถานะใดสถานะหนึ่งในแต่ละคู่ของคู่ต่อไปนี้

MARK/SPACE

ON/OFF

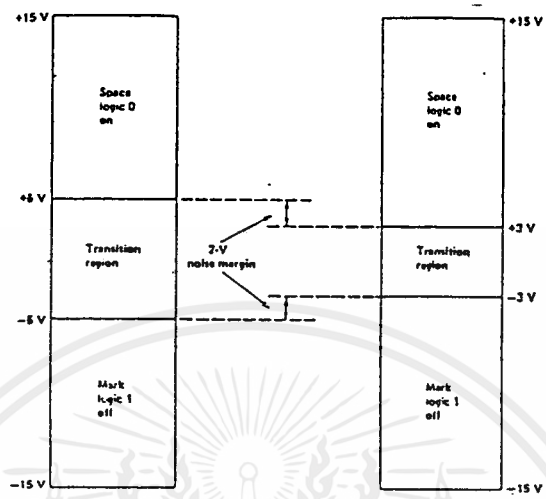
LOGIC 0/LOGIC 1

ความสัมพันธ์ระหว่างสถานะของสัญญาณคู่ต่าง ๆ กับระดับแรงดันได้แสดงไว้ในตารางที่ 1

Status	Signal Voltage	
	-25V < V1 < -3V	3V < V1 < 25V
Binary logic state	1	0
Signal condition	MARK	SPACE
Function	OFF	ON

ตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



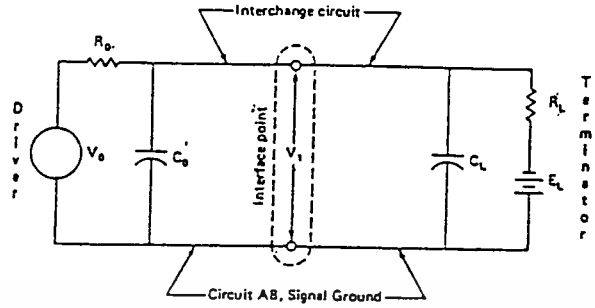
รูปที่ 7 คุณสมบัติทางไฟฟ้าของการอินเตอร์เฟสแบบ RS-232-C

ช่องของระดับแรงดันระหว่าง -3 ถึง +3 โวลท์ จะเป็นช่วงของการเปลี่ยนลอจิก ดังนั้นจึงไม่มีการระบุสภาวะของสัญญาณในช่วงนี้ ในการแทนลอจิก 1 หรือสถานะ Mark ตัวขับสัญญาณ (Driver) ต้องจ่ายแรงดันระหว่าง -5 ถึง -15 โวลท์ ส่วนในการแทนลอจิก 0 หรือ Space ตัวขับสัญญาณต้องจ่ายแรงดันระหว่าง +5 ถึง +15 โวลท์

RS-232-C ยอมให้มี Noise Margin ได้ไม่เกิน 2 โวลท์ สำหรับความสัมพันธ์ระหว่างระดับแรงดันและสถานะของสัญญาณได้แสดงไว้ในรูปที่ 7

จากรูปจะเห็นได้ว่า ถ้า Line Driver หรือตัวกำเนิดสัญญาณต้องการส่งลอจิก 0 Line Driver จะต้องจ่ายแรงดันระหว่าง +5 ถึง +15 โวลท์ ส่วน Line Receiver หรือตัวรับสัญญาณปลายทางจะถือว่าแรงดันที่อยู่ภายในช่วง +3 ถึง +5 โวลท์ แทนลอจิก 0 จากการเปรียบเทียบระดับสัญญาณของตัวส่งและตัวรับจะเห็นว่า RS-232-C ยอมให้มีการ Drop ของสัญญาณในช่วง 2 โวลท์เกิดขึ้นได้สำหรับในด้านการส่งลอจิก 1 ก็เป็นเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 RS-232-C Interface Circuit (EIA)

จากที่ได้อธิบายมาอาจมีข้อสงสัยว่า ทำไมไม่ใช้สถานะลอจิกแบบ TTL ซึ่งระดับของแรงดันมีค่าระหว่าง 0 ถึง +5โวลท์และทำไมถึงต้องใช้ระดับแรงดันระหว่าง -15 ถึง -3 และ +3 ถึง +15 โวลท์ด้วย

สาเหตุที่ไม่ใช้การแทนลอจิกแบบ TTL ก็เพราะสถานะลอจิกแบบ TTL ถูกรบกวนจากสัญญาณรบกวนต่าง ๆ ได้ง่าย นอกจากนั้นยังมีปัญหาเกี่ยวกับระยะทางที่สามารถทำการสื่อสารข้อมูลอีกด้วย สำหรับสาเหตุที่ต้องใช้แรงดันในช่วง -15 ถึง -3 และ +3 ถึง +15 ก็เพราะในขณะที่กำลังทำการพัฒนา RS-232-C นั้นในวงจรคอมพิวเตอร์ต่าง ๆ โดยทั่ว ๆ ไปมีการใช้ระดับแรงดันในช่วงเหล่านี้อยู่ อนึ่งทรานซิสเตอร์ที่มีขายกันทั่วไปสามารถทำงานได้ในช่วงแรงดันเหล่านี้ และยังทนต่อสัญญาณรบกวนต่าง ๆ ที่มีเข้ามาได้ นอกจากนี้ยังสามารถทำงานที่ความถี่สูง ๆ ได้ถึง 20000 บิทต่อวินาที (bps) ยิ่งกว่านั้น MARK และสถานะ SPACE ยังถูกแทนด้วยการไหลของกระแสในทิศทางที่ตรงข้ามกัน และความแตกต่างของแรงดันที่สถานะ MARK และ SPACE มีค่าสูงถึง 6 โวลท์เป็นอย่างน้อย ข้อดีต่าง ๆ ที่กล่าวมานี้ช่วยให้การส่งข้อมูลมีเสถียรภาพ

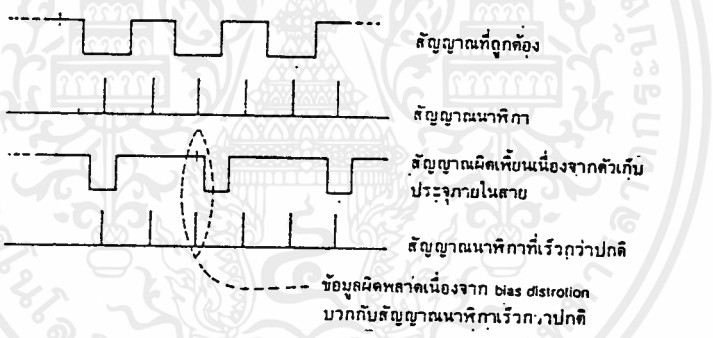
จากรูปที่ 8 นั้น ตัวเก็บประจุ C1 ที่ต่อขนานกับอุปกรณ์รับข้อมูลปลายทางจะต้องมีค่าไม่เกิน 2500 pF โดยค่านี้ไม่รวมค่าความจุไฟฟ้าของสายเคเบิลเข้าไปด้วย (ด้วยเหตุนี้จึงทำให้ระยะที่สามารถใช้ทำการสื่อสารข้อมูลได้ต้องไม่เกิน 50 ฟุต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาของ RS-232-C พอสรุปได้ 3 ประการ ดังนี้

1. ใช้ระดับแรงดันไฟเลี้ยง +15 V. นอกเหนือจาก 5 โวลท์ ซึ่งใช้ในวงจรลอจิก

2. ค่าตัวเก็บประจุของอุปกรณ์รับสัญญาณ RS-232-C รวมทั้งตัวเก็บประจุสเตย์ (Stay Capacitance) สายจะต้องไม่มากกว่า 2500 pF สายที่รวมกันหลาย ๆ สาย ส่วนมากจะมีตัวเก็บประจุสเตย์ ประมาณ 40-50 pF ต่อ 1 ฟุต ดังนั้นสายนี้จะต่อได้ยาวสุด 50 ฟุต ก่อนที่ค่าตัวเก็บประจุสเตย์จะมากกว่า 2500 pF ถ้าหากตัวเก็บประจุสเตย์มากกว่าที่กำหนดนี้ ช่วงเวลาการเปลี่ยนแปลงระดับของสัญญาณมากกว่า 4 เปอร์เซนต์ ตามที่ยอมให้ได้ในมาตรฐาน RS-232-C เมื่อเป็นเช่นนี้จะทำให้ฝ่ายรับตีความสัญญาณผิดไปจากความเป็นจริง มาร์กบิท (MARK BIT) จะยาวกว่าสเปซบิท (SPACE BIT) หรือสเปซยาวกว่ามาร์กบิทขึ้นอยู่กับวงจรการตรวจสอบการผิดเพี้ยนแบบนี้เรียกว่า "Bias Distrotion"

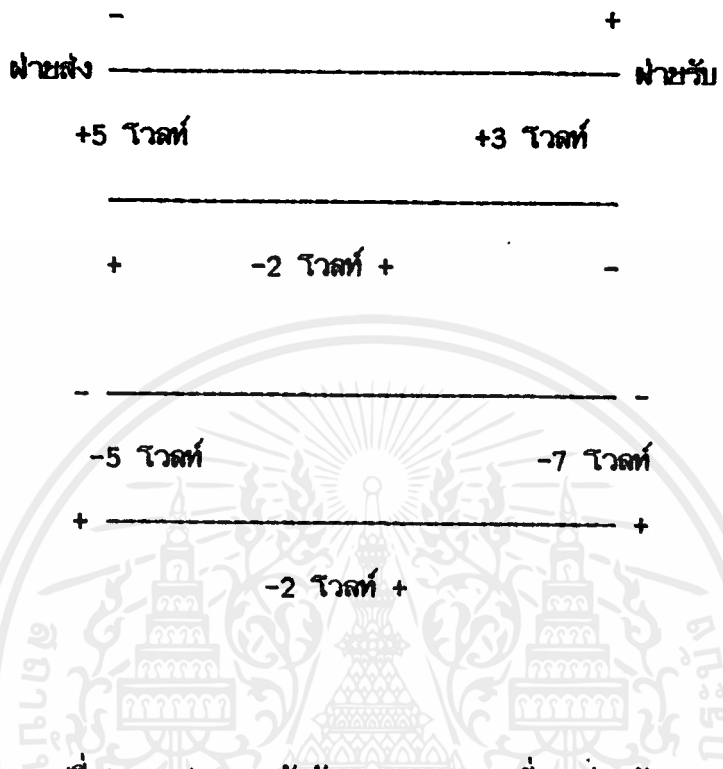


รูปที่ 9 แสดงถึงการตีความข้อมูลที่ผิดของฝ่ายรับอันเนื่องจาก Bias Distrotion และสัญญาณนาฬิกาเร็วกว่าปกติ

3. ปัญหาทางสัญญาณไฟฟ้าก็คือปัญหาเรื่อง Ground ที่แตกต่างกัน สัญญาณที่ส่งออกเทียบกับ Groud ของตัวส่งเท่านั้น ถ้าหากตัวรับกับตัวส่งมีระดับแรงดัน Ground ที่แตกต่างกัน สมมุติว่า 2 โวลท์ กระแสที่ไหลในเส้นที่เป็น Ground ก็จะเกิดขึ้น สมมุติว่าความต้านทานของสายเป็น 0 ความต่างศักย์ที่เกิดจากกระแส Ground ก็จะไม่มี ความต่างศักย์ของกราวน์ระหว่างตัวรับกับตัวส่งก็จะคงเท่าเดิมระดับของสัญญาณที่ฝ่ายส่งและฝ่ายรับ มองเห็นก็แตกต่างกัน สมมุติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าระดับแรงดัน Ground ต่างกัน 2 โวลท์ ตัวส่งป้อนแรงดันเข้าไป 5 โวลท์ ตัวรับก็จะมองเห็นแค่ 3 โวลท์เท่านั้นในทางกลับกัน ถ้าฝ่ายส่งป้อนแรงดัน -5 โวลท์จะมองเห็นเป็น -7 โวลท์



รูปที่ 10 ผลของระดับสัญญาณ Ground ที่แตกต่างกัน

ความต่างศักย์ของ Ground จะคงที่ 2 โวลท์ ไม่ว่าฝ่ายส่งจะใส่แรงดันเข้าไปเท่าไหร่ก็ตาม ผลของ Ground ที่แตกต่างกันนี้อาจจะเกิดมาจากตัวรับและตัวส่งมีระบบไฟฟ้าที่ Ground แตกต่างกันก็ได้

จากปัญหาต่าง ๆใน RS-232-C ทาง EIA จึงได้พัฒนาและกำหนดมาตรฐานใหม่ขึ้นอีกเป็น RS-422, RS-423 และ RS-485

RS-422, RS-423 และ RS-485 มีข้อได้เปรียบ RS-232-C ตรงที่อินพุทของพวกมันเป็นแบบ "การขยายความแตกต่าง (Differential Input)" ต่อไปจะอธิบายว่ามันได้เปรียบอย่างไรขอให้พิจารณาภาพต่อไปนี้ : เราวางตำแหน่งของ DCE และ DTE ของเราไว้ในสำนักงานเดียวกัน แต่ใช้ระบบจ่ายไฟคนละระบบ จากสาเหตุนี้เองทำให้ระดับแรงดันที่สาย Ground ของ DCE และ DTE มีระดับแรงดันแตกต่างกันโดยสมมุติให้ระดับแรงดันที่ Ground ของ DTE มีค่ามากกว่า DCE เท่ากับ 5 โวลท์ จากมาตราฐาน RS-232-C ถ้าเราต้องการส่งลอจิก 1 เราต้องป้อนแรงดันในช่วง -5 ถึง -15 โวลท์ เทียบกับ Ground สัญญาณ (Signal Ground)ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

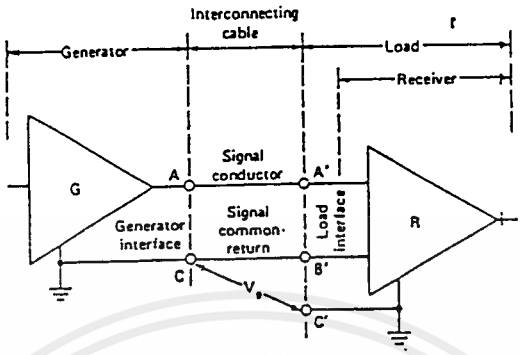
ตัวส่ง ดังนั้นถ้า DTE ป้อนระดับแรงดันในช่วง -5 ถึง -7.9 โวลท์ เทียบกับ Ground ของมัน (ของ DTE) เมื่อ DCE รับระดับแรงดันนี้เข้ามาโดยเทียบกับ Ground ของมัน (ของ DCE) ระดับแรงดันที่รับเข้ามาจะอยู่ในช่วง 0 ถึง -2.9 โวลท์แทน ซึ่งจากเหตุการณ์เช่นนี้ DTE จะคิดว่ามันส่งข้อมูลที่เป็นลอจิก 1 ออกมา แต่จริง ๆ กลายเป็นว่า DCE รับข้อมูลอยู่ในช่วงที่ไม่สามารถตัดสินได้ว่าเป็นลอจิกใดไปแทน (อยู่ใน Transition Region) ดังนั้นความแตกต่างของระดับแรงดันที่ Ground สัญญาณของ DTE และ DCE ซึ่งมีค่าเท่ากับ 5 โวลท์ เป็นตัวก่อให้เกิดความผิดพลาดข้อนี้ขึ้น

ตัวการอีกอย่างหนึ่ง ที่ทำให้ระดับแรงดันที่ส่งมีค่าเปลี่ยนแปลงไป คือ สัญญาณรบกวนทางไฟฟ้า (Electrical Noise) ถ้าเราวางเคเบิล RS-232-C ผ่านสนามแม่เหล็กไฟฟ้าที่แรงพอ สนามแม่เหล็กไฟฟ้านี้สามารถเหนี่ยวนำให้เกิดสัญญาณรบกวนที่มีค่ามากพอ ที่จะเปลี่ยนแปลงระดับลอจิกที่ส่งไปจากลอจิก 0 ไปเป็น 1 หรือลอจิก 1 ไปเป็น 0 ได้

เราสามารถลดปัญหาต่าง ๆ ที่กล่าวไว้ข้างต้นลงได้โดยใช้ตัวรับข้อมูลที่มีอินพุทเป็น แบบวงจรขยายความแตกต่าง (Differential Input) คุณสมบัติที่สำคัญของตัวรับข้อมูลแบบนี้ คือ มันจะวัดระดับความแตกต่างของระดับแรงดันที่อินพุททั้งสอง ดังนั้นอินพุทหนึ่งจะถูกต่อเข้ากับตัวนำสัญญาณที่ต้องการส่ง ส่วนอินพุทอีกตัวหนึ่งจะต่อเข้ากับสาย Ground ของตัวส่ง การใช้งานลักษณะนี้ถูกใช้ในมาตรฐาน RS-423 แต่ในมาตรฐาน RS-422 และ RS-485 นั้น อินพุทของตัวรับข้อมูลได้รับสัญญาณความแตกต่างที่ส่งมาจากตัวส่ง สำหรับระดับแรงดันที่ส่งจากตัวส่งจะมีค่าเท่ากับความแตกต่างของเอาท์พุททั้งสอง จากปัญหาที่กล่าวไว้ข้างต้น เราสามารถใช้วงจรแบบ Balanced และ Unbalanced แก้ไขได้ดังนี้ การอินเตอร์เฟสแบบ RS-423 สามารถแก้ไขปัญหาที่ระดับแรงดันที่ Ground ของ DTE มีค่าสูงกว่า DCE อยู่ 5 โวลท์ โดยอาศัยการต่อสาย Ground ของ DTE เข้าที่อินพุทแบบขยายความแตกต่างของ DCE ส่วน RS-422 และ RS-485 แก้ไขปัญหานี้โดยรับเป็นค่าความแตกต่างเข้ามา ซึ่งการแก้ปัญหาโดยการใช้การอินเตอร์เฟสแบบนี้ไม่ต้องใช้สาย Ground เลย สำหรับปัญหาเกี่ยวกับสัญญาณรบกวนนี้ถูกกำจัดไป เนื่องจากลวดตัวนำที่เป็นอินพุทของตัวรับข้อมูลทั้งสองอินพุทถูกวางผ่านสภาพแวดล้อมทางไฟฟ้า อย่างเดียวกันดังนั้นจึงได้รับสัญญาณรบกวนเหมือนกัน สมมุติให้สัญญาณรบกวนที่ได้รับมีแรงดันเป็น X โวลท์ ที่จุด A' และ B' จะมีแรงดันเท่ากับ Va + x และ Vb + x (สมมุติในนี้แรงดันที่จุด A' และ B' ในกรณีที่ไม่มีสัญญาณรบกวนมีค่าเป็น Va และ Vb ตามลำดับ) เนื่องจากอินพุทของตัวรับข้อมูลเป็นแบบขยายความแตกต่าง ดังนั้นสัญญาณรบกวนจะไม่มีผลเนื่องจาก (Va + X) - (Vb + X) มีค่าเท่ากับ Va - Vb ซึ่งเป็นแรงดันที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มาตราฐาน RS-423



รูปที่ 11 ลักษณะทางไฟฟ้าของการอินเทอร์เฟส RS-423

-ตัวผลิตสัญญาณเป็นแบบ Unbalance

-ตัวรับข้อมูลเป็นแบบขยายความแตกต่าง

-ในแต่ละ Circuit ใช้ลวดตัวนำในการส่งสัญญาณเพียง 1 สาย มีสาย Ground แยกสำหรับการไหลของสัญญาณกลับในแต่ละทิศทาง

-อัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 100 kbps

-ระยะทางที่ส่งได้ไกลถึง 4000 ฟุต

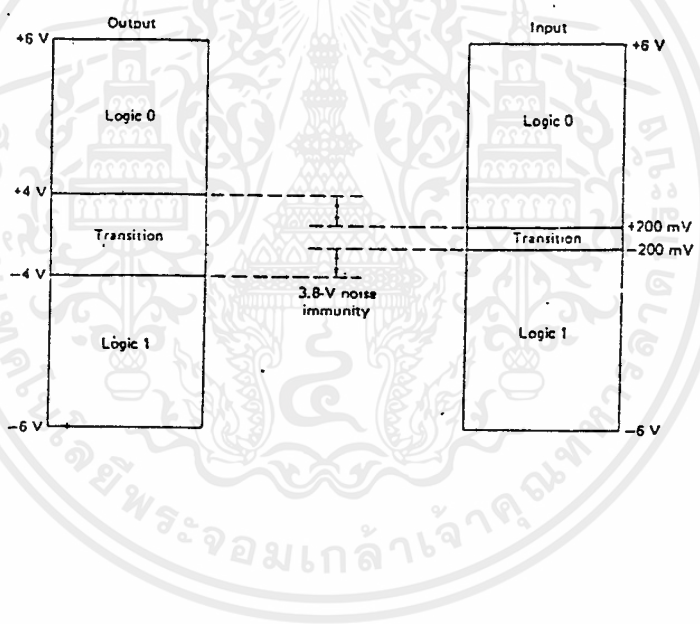
มาตราฐาน RS-423 เป็นมาตราฐานที่ได้รับการพัฒนามาจากมาตราฐาน RS-232-C ซึ่งใช้สายสัญญาณเส้นเดียวในการส่งสัญญาณ โดยสัญญาณที่ส่ง ส่งไปได้ทิศทางเดียว ตัวรับข้อมูลแบบขยายความแตกต่างของสัญญาณระหว่างสาย Ground กับตัวส่งสัญญาณ การทำเช่นนี้ช่วยแก้ปัญหาในกรณีที่เกิดความแตกต่างระหว่างแรงดันที่ Ground ของตัวรับข้อมูลกับตัวส่งสัญญาณ

ระดับแรงดันที่ใช้ใน RS-423 มีค่าน้อยกว่า RS-232-C ก็เป็นส่วนทำให้อัตราการส่งข้อมูลมีค่าสูงขึ้นหรือเรียกได้ว่าหนึ่งในส่วนที่เป็นตัวจำกัดอัตราการส่งข้อมูลก็คือ Slew Rate เพราะว่าระดับแรงดันของ RS-232-C นั้นสูงจึงจำเป็นต้องใช้ค่า Slew Rate ที่สูง เพื่อที่จะทำการส่งให้มีค่าอัตราการส่งข้อมูลสูงขึ้น แต่เรายังไม่สามารถให้ค่า Slew Rate นี้สูงได้ (เพราะมีข้อจำกัดอยู่หลายประการ) ดังนั้นเมื่อเราใช้ค่าระดับแรงดันในการส่งให้ต่ำลง ก็ทำให้เรานั้นใช้ค่า Slew Rate ต่ำลงด้วยสำหรับค่าอัตราการส่งข้อมูลที่เท่ากัน ดังนั้นที่ค่า Slew Rate เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ค่าระดับแรงดันที่ใช้ในการส่งนั้นลดลงมาก็สามารถที่จะทำให้ค่าอัตราการส่งข้อมูลสูงขึ้นได้

RS-423 ใช้ระดับแรงดัน 4 โวลท์ ถึง 6 โวลท์แทน ไบนารี่ 0 และ -4 โวลท์ ถึง -6 โวลท์ ไบนารี่แทน 1 เมื่อใช้ค่า Slew Rate ที่เท่ากับ RS-232-C ก็จะให้ค่าอัตราการส่งข้อมูลของ RS-423 สูงกว่า RS-232 ประมาณ 4 เท่า นอกจากนั้นในการรับค่าไบนารี่ 1 หรือ 0 ที่ตัวรับข้อมูลของ RS-423 ก็สามารถรับรู้ค่าได้ในที่ต่ำกว่า RS-232 คือถ้าที่ขาอินพุทของตัวรับข้อมูลมีระดับแรงดันแตกต่างกัน 200 มิลลิโวลท์ ก็จะรับรู้ได้ว่าเป็นไบนารี่ 0 และ -200 มิลลิโวลท์ ก็จะรับรู้ว่าเป็นไบนารี่ 1 ก็หมายความว่าการส่งข้อมูลแบบ RS-423 สามารถที่จะยอมให้เกิดค่าแรงดันสูญเสีย (Voltage Loss) ในสายได้มากกว่าแบบ RS-232-C



รูปที่ 12 แสดงระดับแรงดันไฟฟ้าของการอินเตอร์เฟสแบบ RS-423

ตัวอย่างการคำนวณอย่างง่าย ๆ ที่จะแสดงผลของการส่งระดับแรงดันที่ตัวรับข้อมูลสามารถที่รับรู้ได้บนอัตราการส่งข้อมูล และ Slew Rate ที่สามารถทำได้ สมมุติว่าตัวส่งข้อมูลส่งสัญญาณค่า Slew Rate 1000 V/S

ส่งแบบ RS-232-C (-25 v ถึง +25 v) 50/1000 = 0.05 S

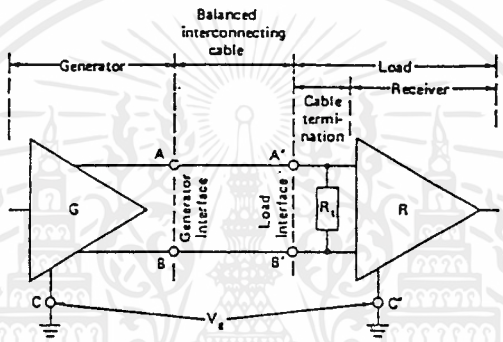
ส่งแบบ RS-423 (-6 v ถึง +6 v) 12/1000 = 0.012 S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะเห็นได้ว่าตัวส่งข้อมูลที่ค่า Slew Rate เดียวกันนั้น ถ้าส่งแบบ RS-423 จะสามารถทำให้ได้ค่าอัตราการส่งข้อมูลที่สูงกว่า เพราะว่าใช้ระดับแรงดันไฟฟ้าที่มีค่าต่ำกว่า

2.4 มาตรฐาน RS-422

มาตรฐาน RS-422 ได้พัฒนาจากมาตรฐาน RS-423 ทำให้อัตราเร็วในการส่งข้อมูล มีค่าสูงขึ้นและระยะทางที่ใช้ส่งข้อมูลระหว่างตัวส่งและตัวรับมีระยะไกลขึ้น



รูปที่ 13 ลักษณะสมบัติทางไฟฟ้าของการอินเตอร์เฟสแบบ RS-422

-ตัวส่งสัญญาณเป็นแบบ Balanced

-ตัวรับข้อมูลเป็นแบบขยายความแตกต่าง

-ในแต่ละวงจรใช้ลวดตัวนำในการส่งสัญญาณจำนวน 2 เส้น

-อัตราเร็วในการส่งข้อมูลสูงถึง 10 Mbps

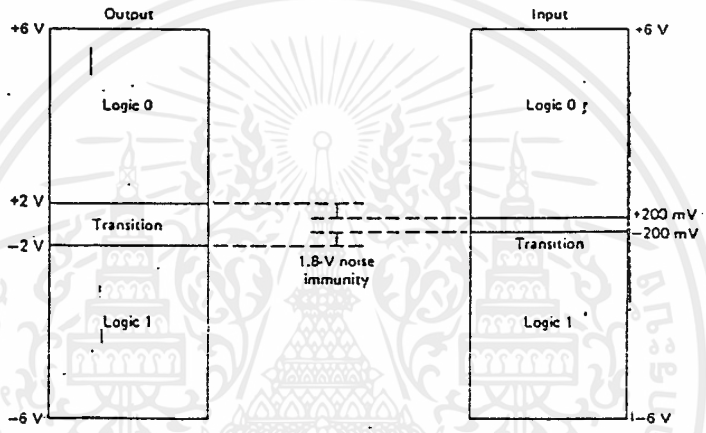
-ระยะทางที่ใช้ในการส่งข้อมูลได้ไกลถึง 4000 ฟุต

มาตรฐาน RS-422 ใช้ในการส่งข้อมูลในลักษณะของ One-Way Balance-Line ซึ่งมีตัวส่งข้อมูลบน Line 1 ตัว และตัวรับข้อมูล 10 ตัวโดยอัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 10 Mbps ที่ระยะทางเท่ากับ 40 ฟุต ในการมีที่ส่งข้อมูลในอัตราเร็วที่ต่ำกว่า 10 Mbps ระยะทางที่ใช้ในการส่งข้อมูลสามารถขยายได้ถึง 4,000 ฟุต

ตัวส่งสัญญาณเป็นแบบ Balanced ระดับแรงดันที่ส่งจากตัวส่งสัญญาณจะมีค่าระหว่าง +2 โวลท์ ถึง +6 โวลท์ โดยที่ระดับแรงดัน 2 โวลท์ ถึง 6 โวลท์ แทนไบนารี่ 0 และ -2 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึง -6 โวลท์ แทนไบนารี่ 1 ซึ่งจะทำให้มีอัตราเร็วในการส่งข้อมูลสูงกว่าแบบ RS-423 และในการรับรู้ค่าไบนารี่ 1 หรือ 0 ที่ตัวรับข้อมูล สามารถจับสัญญาณที่มีระดับต่ำถึง +200 มิลลิโวลท์คือถ้าที่ขาอินพุทของตัวรับข้อมูล มีระดับแรงดันแตกต่างกัน +200 มิลลิโวลท์ ก็จะรับรู้ได้ว่าเป็นไบนารี่ 0 และถ้าเป็น -200 มิลลิโวลท์ก็จะรับรู้ว่าเป็นไบนารี่ 1

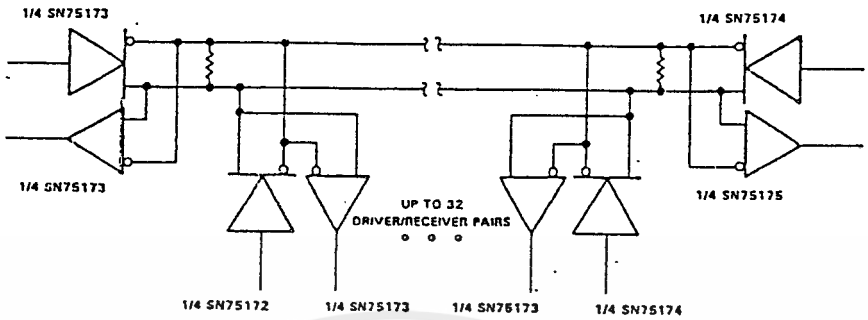


รูปที่ 14 แสดงระดับแรงดันไฟฟ้า ของการอินเตอร์เฟสแบบ RS-422

2.5 มาตรฐาน RS-485

มาตรฐาน RS-485 นี้ก็มีการพัฒนามาจาก RS-422 คือผู้ผลิตบางบริษัทได้ทำวงจรขับสัญญาณ (ตัวส่ง) เป็นแบบ Tri-State ทำให้เราสามารถส่งข้อมูลได้สองทิศทางบนสายคู่เดียว (Single Pair) คุณสมบัติข้อนี้จึงทำให้ ระบบส่งข้อมูลมีโครงสร้างเป็นแบบ Multidrop ซึ่งอุปกรณ์หลาย ๆ ตัวสามารถรับและส่งข้อมูลแบบ Half Duplex บนสายคู่เดียวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 15 ลักษณะสมบัติทางไฟฟ้าของการอินเตอร์เฟสแบบ RS-485

มาตรฐาน RS-485 นี้ ทางบริษัทผู้ผลิตได้ออกแบบตัวส่ง และตัวรับให้สามารถต่อร่วมอยู่บนบัสได้ถึงอย่างละ 32 ตัว แต่การส่งข้อมูลนั้นจะส่งได้ทีละตัวเท่านั้น ฉะนั้นจะต้องมีกรรมวิธีตรวจสอบบัสก่อนที่จะส่งข้อมูลไปในบัส เพื่อป้องกันการชนกันของข้อมูล ส่วนการรับข้อมูลนั้นสามารถรับได้พร้อม ๆกันถึง 32 ตัว แต่ถ้าเราต้องการกำหนดว่าจะให้ตัวรับข้อมูลตัวไหนนั้นรับข้อมูล เราจะต้องมีการตรวจสอบก่อนที่จะรับข้อมูลว่าข้อมูลที่ส่งมานั้นใช่ของตนเองหรือไม่ ซึ่งมันก็ขึ้นอยู่กับเทคนิคของการเขียนโปรแกรมตรวจสอบของแต่ละบุคคล

ส่วนประกอบของมาตรฐาน RS-485 นี้ก็มี ตัวส่ง, ตัวรับ, สายเคเบิล และ R-TERMINATING ซึ่งค่าความต้านทานของ RT ที่จะนำมาต่อนี้ก็จะขึ้นอยู่กับตัวส่งและตัวรับว่าเราจะใช้ IC เบอร์อะไรตาม DATA SHEET ของ IC เบอร์นั้นจะกำหนดค่า RT ไว้ให้แล้ว รายละเอียด และคุณสมบัติทางไฟฟ้าของตัวส่ง และตัวรับถูกออกแบบ และตรวจสอบโดยโรงงานผู้ผลิตให้เป็นไปตามข้อกำหนดของทาง EIA RS-485 (APRIL 1983) ส่วนในเรื่องของการกำหนดรูปแบบการส่งข้อมูล ,สายเคเบิล และเรื่องอื่นนั้นไม่ได้มีการกำหนดเอาไว้ให้ผู้ใช้กำหนด และเลือกเองตามความเหมาะสมของงานนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EIA STANDARD RS-485 TWO WIRE

มาตรฐานนี้จะแสดงรายละเอียดคุณสมบัติทางไฟฟ้าของเครื่องกำเนิดสัญญาณ และเครื่องรับ (Generators and Receivers)ว่าอาจถูกนำมาใช้เพื่อแสดงรายละเอียดสำหรับการสับเปลี่ยนของ Binary Signals ใน Multipoint Interconnection of Digital Equipments เมื่อได้รับการสนับสนุนภายในการนำของมาตรฐานนี้แล้ว เครื่องกำเนิดสัญญาณและเครื่องรับหลายอย่างอาจถูกนำมาเชื่อมถึงกันแบบ Common Interconnecting Cable

ระบบการสับเปลี่ยนหนึ่งประกอบด้วยเครื่องกำเนิดหนึ่งตัวหรือมากกว่า ซึ่งเชื่อมต่อกันโดยปลาย ลักษณะทางไฟฟ้าของวงจรจะแสดงรายละเอียดในเงื่อนไขของแรงดันไฟฟ้าที่ต้องใช้,กระแส และค่าความต้านทาน ซึ่งได้รับโดยเครื่องวัดในจุดต่อของเครื่องกำเนิดไฟฟ้า และเครื่องรับ Loading ที่เกิดขึ้นโดยวงจร จะถูกกำหนดตามจำนวนของ "unit loads" หรือส่วนใดส่วนหนึ่งของ "unit loads" เกณฑ์จะให้ในส่วนปิดจำกัดของอัตราการส่งสัญญาณข้อมูลซึ่งกำหนด โดยค่า parameter of cable lenght ความสมดุลและการสิ้นสุดของการติดตั้งแต่ละอย่าง

ค่าพารามิเตอร์ได้ถูกแสดงในมาตรฐานนี้ คล้ายกับใน RS-422-A ค่าเหล่านี้ทำให้เครื่องกำเนิดไฟฟ้าและเครื่องรับถูกออกแบบให้สามารถใช้ได้เหมาะสมกับความต้องการของมาตรฐานทั้งสอง มาตรฐานนี้ไม่ได้แสดงรายละเอียด คุณสมบัติอื่น ๆ เช่น คุณภาพสัญญาณ, การกะเวลา (timing), protocal, pinassignments ฯลฯ ซึ่งจำเป็นสำหรับการทำงานที่เหมาะสมของอุปกรณ์ภายใน และชี้แจ้งให้เห็นว่ามาตรฐานนี้มีความเกี่ยวข้องกับมาตรฐานอื่น ๆ และการแสดงรายละเอียดต่าง ๆ นั้นได้แสดงคุณสมบัติเพิ่มเติม ซึ่งจำเป็นต่อการ Interfacing ที่แน่นอนของเครื่องมือหรืออุปกรณ์นั้น

กลไกต่าง ๆ ซึ่งมีรูปแบบตามขอบเขตที่แน่นอนด้วยมาตรฐานนี้ จะเหมาะกับการแสดงรายละเอียดที่ใช้ภายในขอบเขตของปัจจัยต่าง ๆ เหล่านั้น ซึ่งอธิบายได้อย่างเหมาะสม สำหรับการทำงานของอุปกรณ์ เช่น Power supply voltage, อุณหภูมิของอากาศและความชื้น

APPLICABILITY

ในการจัดมาตรฐานนี้ ใช้เฉพาะกับส่วนประกอบของเครื่องกำเนิดสัญญาณและเครื่องรับ ซึ่งใช้ติดต่อระหว่างอุปกรณ์ที่นำส่งข้อความในรูป binary signals ซึ่งอาจมี DC หนึ่งเป็นส่วนประกอบ มาตรฐานนี้ไม่ใช่เพื่อ Interface Standard หนึ่ง แต่จะถูกเกี่ยวข้องโดย Interface Standard เหล่านั้นหรือแสดงรายละเอียดการใช้เครื่องกำเนิดไฟฟ้า และเครื่องรับที่มีคุณสมบัติทางไฟฟ้าเหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในวงจรที่มีคุณสมบัติที่ถูกแสดงในนี้ จะถูกใช้เป็นประจำในข้อมูล การกะเวลาหรือการควบคุมความต่อเนื่อง ซึ่งมีอัตราการให้สัญญาณข้อมูลสูงถึง 10 Mg/sec พารามิเตอร์จะแสดงว่าคลไกต่าง ๆ ในระบบสามารถทำงานได้ด้วยแรงดันไฟฟ้าแบบธรรมดาที่ +-7 volts การทำงานโดยมีหรือไม่มี Generator Offset Voltage นั้นก็จะถูกจัดหา อย่างไรก็ตาม the common mode voltage tolerance จะลดต่ำลง ถ้า offset voltage เป็นศูนย์ (0)

ELECTRICAL CHARACTERISTICS

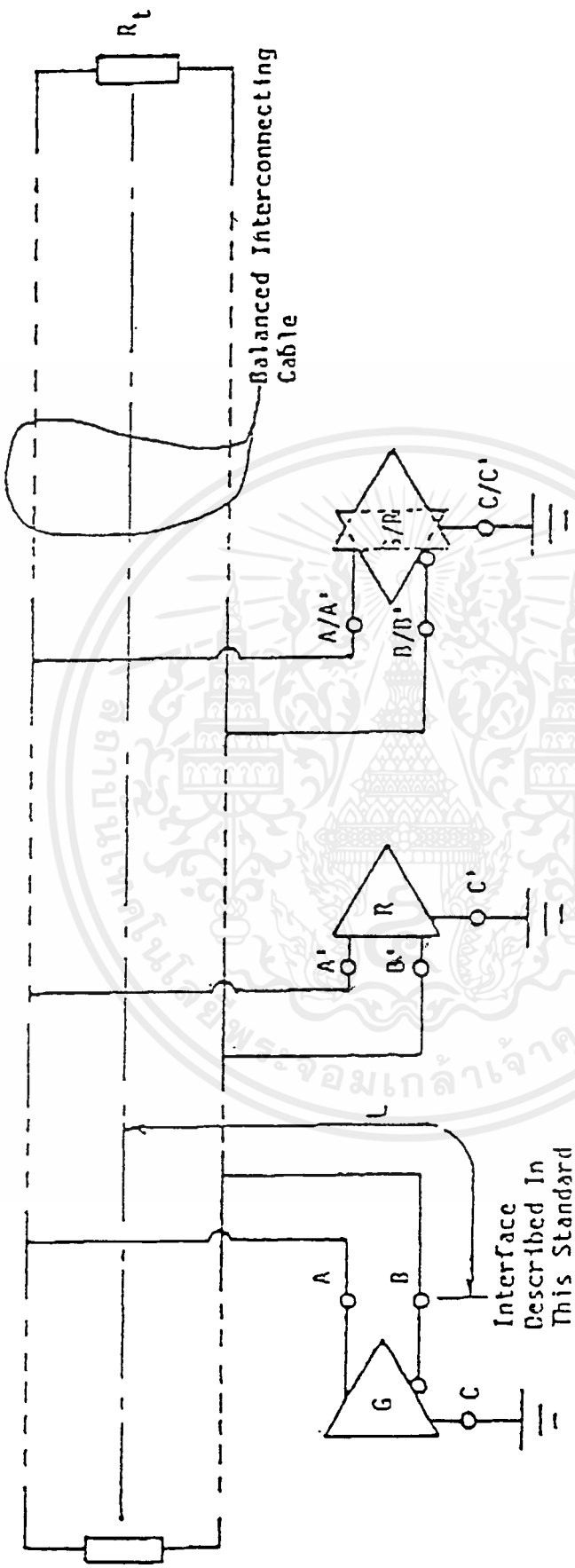
รูป แสดงการใช้ความสัมพันธ์ของเครื่องส่งและเครื่องรับ ที่มีค่าพารามิเตอร์ทางไฟฟ้าในมาตรฐานนี้ ส่วนที่นำมาใช้ประโยชน์คือ Generators, Receivers, Transmission cable และ Termination Resistance (R_t)ข้อมูลบน Active Generator หนึ่งจะถูกแสดงในเงื่อนไขของ unit loads (U.L.S.) ข้อมูลในระบบที่เกิดจากเครื่องรับ หรือเครื่องกำเนิดไฟฟ้าแต่ละตัว และ Passive Generator จะถูกแสดงโดยจำนวนหรือ unit loads ที่ปรากฏแต่ละครั้ง unit loads จะถูกอธิบายโดยลักษณะแรงดันของกระแสไฟฟ้า ข้อมูลนี้จะถูกอธิบายในรูป Passive Generator,Receiver (R) หรือทั้งสอง Electrical Parameters ที่เลือกมาแสดงในตอนต่อไป จะชี้ให้เห็นว่าเครื่องส่งหนึ่ง ๆ สามารถส่งข้อมูล 32 บิต ข้อมูล ผลความต้านทานทั้งหมดต่ำมาก

ลักษณะทางไฟฟ้าที่ได้แสดงรายละเอียดไปแล้วนั้น ถูกวัดที่จุด Interconnect จุดหนึ่งซึ่งทำโดยผู้ผลิตคลไกนั้น ๆ มีการทดสอบการใช้งานโดยมีมาตรฐานรับรองว่าหลักเกณฑ์ในจุดนั้นอาจแตกต่างกันได้ เนื่องจากการต่อสายภายใน และเครื่องรับอื่น ๆ และ Passive Generators ภายในอุปกรณ์ในภาพต่อมาจุด A, B และ C แสดง Generator Interconnect Points ในขณะที่ A', B' และ C' แสดง Interconnect Pionts ร่วมกับเครื่องรับ ตัวต้านทานที่ปลายสุดจะชี้ให้เห็นส่วนของ Interconnecting means

LOAD CHARACTERISTICS

ข้อมูลที่เกิดโดย Receivers และ Passive Generators บน Interconnect mean จะต้องพิจารณาในขอบเขตของลักษณะทางไฟฟ้าของอุปกรณ์ มี 2 บริเวณเกี่ยวกับ DC Load และ AC Load ใน DC Load คลไกแต่ละตัวจะวางอยู่บนระบบ ซึ่งจะหมายถึง จำนวนหรือส่วนของ "unit loads" ดังที่ได้อธิบายไว้ ส่วน AC Loding จะไม่กำหนดมาตรฐานเอาไว้แต่จะพิจารณาในส่วนของการใช้ระบบคลไก ที่เหมาะสมกับมาตรฐานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



G - Generator
R - Receiver
G/R- Combination Generator/Receiver
L - Length of stub; the guidelines assume length of stub to be effectively zero
R_t- Termination resistance; location and value are not specified in this standard, but a generator can drive 32 unit loads plus two termination resistances of 120 ohms each.

Guidance on the length of the interconnecting cable is given in the Appendix.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC UNIT LOAD SPECIFICATION

เพื่อให้กระแสที่ได้จากเครื่องกำเนิดไฟฟ้าเครื่องหนึ่ง ในสภาวะ Active State ที่ค่าที่เหมาะสม ค่าหนึ่ง การส่งข้อมูลของชุดส่งในสภาวะ Passive State และของเครื่องรับจะถูกแสดงในเงื่อนไขของประจุที่มีสิทธิโดย Hypotheical Unit Load (U.L.) พร้อมกับแรงดันไฟฟ้า Via (หรือ Vib) ในระยะระหว่าง -7 และ +12 โวลท์ ในขณะที่ Vib (หรือ Via) จะอยู่ที่ศูนย์โวลท์ (Grounded) กระแส Iia (หรือ Iib) ที่เกิดขึ้น จะยังอยู่ภายในเขตกำบัง (Shaded region) ซึ่งแสดงในกราฟในภาพ สำหรับข้อมูล 1 บิต เส้นโค้งที่แท้จริงของกระแสกับแรงดันไฟฟ้า (Voltage) จะลาดต่ำลงทางด้านขวาเสมอ

จะหาจำนวนของ Unit Loads a Passive Generator หรือ Direaciver ที่เกิดขึ้น ความลาดเอียง (Slope) ของปริมาณในภาพจะลดต่ำลงถึงจุดต่ำสุด จำเป็นต้องบรรจุ Current Voltage Characteristics ให้เต็มที่ ในขณะที่จุดตัด -3 V และ +5 V ยังคงอยู่

จำนวนของ Unit Loads ในขณะนั้นจะเท่ากับส่วนที่ใหญ่กว่าของสัดส่วน 2 สัดส่วน ของกระแสใหม่ ต่อกระแสเริ่มต้นที่จุดตัด -7 V และ +12 V ตัวอย่างการหาจะแสดงในภาพ จำนวนของ U.L.S. ที่เท่ากันจะแสดงให้เห็นโดย n U.L. ในตอนย่อย

วิธีการหาที่อธิบายไว้ข้างบนนั้นจะถูก Periormed ทั้งสภาวะ Power On และ Power - Off การใช้ระบบจะพิจารณาทั้งสองสภาวะนี้ ในการหาจำนวนทั้งหมดของประจุต่อหนึ่ง Generator ลักษณะของ Current-Voltage ตามการให้ข้อมูล จะเปิดหรือปิดนั้นไม่มีการกำหนดหลักเกณฑ์เอาไว้ แต่ผู้ใช้จะต้องจำไว้ว่าการเปลี่ยนแปลงเช่นนั้นอาจมีผลต่อการทำงานของระบบ

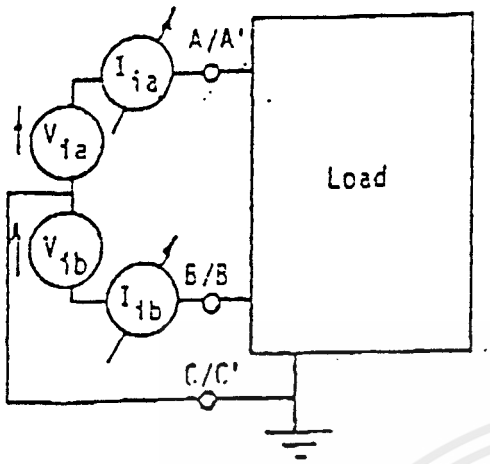
AC LOAD CHARACTERISTICS

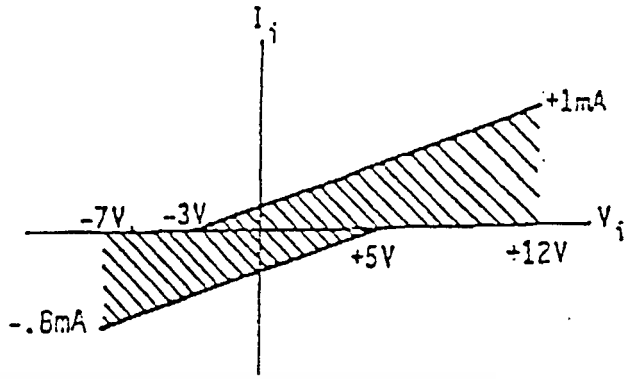
กลไกหลายอย่างมีความสัมพันธ์กับ Interconnect means ด้วยเช่นกัน อย่างลักษณะปฏิกิริยาที่เกิดขึ้น ซึ่งจะมีผลต่อลักษณะการส่งกระแสไฟฟ้าของ Interconnect means ผลกระทบเหล่านี้จะเกิดขึ้นใน

- การสะท้อนกลับ
- ความไม่สมดุล
- การผิดส่วนหรือความไม่ชัดเจนของสัญญาณ

การทำให้เกิด Parameters ของผลกระทบเหล่านี้ ได้แก่ ลักษณะตาม Differtial Impedance ความผิดจาก A' ไป C' และจาก B' ไป C' บนเครื่องรับหรือความผิดจาก A' ไป C และจาก B ไป C บน Passive Generator ซึ่งการพิจารณาเพิ่มเติมของพารามิเตอร์เหล่านี้จะมีอยู่ในภาคผนวก การกำหนดมาตรฐานของพารามิเตอร์เหล่านี้ยังไม่เป็นที่ยอมรับ

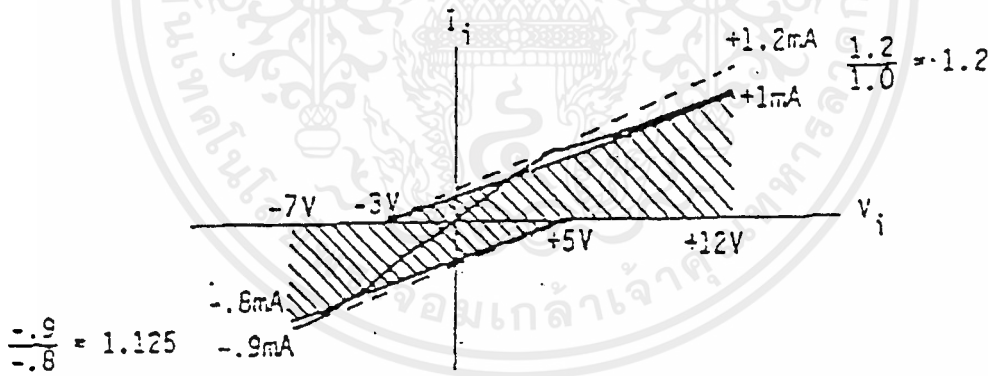
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



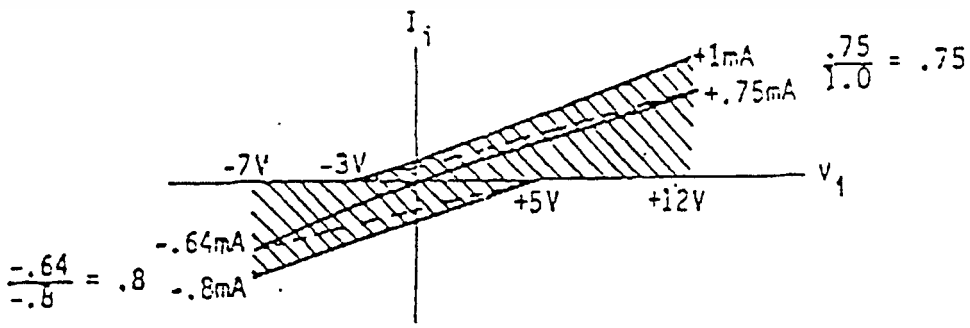


A load is a Passive Generator, a Receiver, or a combination Passive Generator-Receiver.

UNIT LOAD INPUT CURRENT-VOLTAGE MEASUREMENT



Example of 1.2 Unit Load



Example of .8 Unit Load

EXAMPLES OF LOADS THAT ARE NOT ONE UNIT LOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERATOR CHARACTERISTIC

Generator อาจอยู่ในสภาวะ Passive และ Active ขณะที่อยู่ในสภาวะ Passive ลักษณะข้อมูลของ Generator จะแสดงในเงื่อนไขของจำนวนที่เท่ากัน

ขณะที่อยู่ในสภาวะ Active ลักษณะทางไฟฟ้าของ Generator จะแสดงในทางตรงกันกับการวัด เครื่องส่งที่สมดุลจะผลิตแรงดันไฟฟ้าที่ต่างกัน เพื่อใช้ใน Interconnecting ในระดับ 1.5 โวลท์ ถึง 5.0 โวลท์ สัญญาณของการเกิด Voltages ไขว้ของสาย Interconnection Cable มีความหมายดังนี้

ขั้ว A ของ Generator จะเป็นลบในส่วนของขั้ว B สำหรับสภาวะ Binary I (MARK OR OFF)

ขั้ว A ของ Generator จะเป็นบวกในส่วนของขั้ว B สำหรับสภาวะ Binary 0 (SPACE OR ON)

OPEN CIRCUIT MEASUREMENT

สำหรับ Binary State ทั้งคู่ ขนาดของความต่างศักย์ (V) ซึ่งวัดระหว่างขั้ว Outset ทั้ง 2 ของ Generator จะไม่น้อยกว่า 1.5 โวลท์ และไม่มากกว่า 6 โวลท์ และขนาดของ V_{oa} และ V_{ob} ที่วัดได้ต่างหาก ระหว่างแต่ละขั้ว Output ของ Generator กับ Generator Circuit Ground จะไม่มากกว่า 6.0 โวลท์

TEST TERMINATION MEASUREMENT

การทดสอบ Load Termination 2 แบบ จะทำได้โดย ประการที่หนึ่งใช้ความต้านทาน 54 โอห์ม แล้ววัด Output Voltages และ Offset Voltage อีกประการหนึ่ง Common Mode Voltage จะเปลี่ยนจาก -7 โวลท์ ไปเป็น +12 โวลท์ และวัด Output Voltage

TEST TERMINATION MEASUREMENT 1, WITH 54 OHMS LOAD

ทดสอบส่งข้อมูลในสายความต้านทานตัวละ 27 โอห์ม ต่อกันแบบอนุกรม ระหว่างขั้ว Output ของ Generator ขนาดของความต่างศักย์ (V_t) ที่วัดระหว่างขั้ว Output ทั้งสองจะไม่ต่ำกว่า 1.5 โวลท์ หรือไม่สูงกว่า 5.0 โวลท์ สำหรับ Binary State ทั้งสองสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Binary State ที่ตรงกันข้าม กระแสของ V_t จะย้อนกลับ $\bar{V}_t$ ขนาดของความแตกต่างในขนาดของ V_t และ $\bar{V}_t$ จะน้อยกว่า 0.2 โวลท์ Offset Voltage ของ Generator Vas ที่วัดระหว่างจุดศูนย์กลางของ Test Load และ Generator Circuit Ground จะอยู่ระหว่าง -1 โวลท์ และ +3 โวลท์ สำหรับ Binary State ทั้ง 2 สภาวะ

ในขณะที่การแสดงรายละเอียดแบบธรรมดา จะถูกออกแบบสำหรับ Deliberate Offset Voltage ที่ผู้ใช้ยอมให้ใช้ที่ Offset ศูนย์ อย่างไรก็ตามถ้า Offset ศูนย์ ถูกใช้ความทนทานต่อ Common Mode จะลดลง ขนาดของความแตกต่างระหว่าง Vos สำหรับ Binary State-หนึ่ง และ $\bar{V}$ds สำหรับ Binary State ตรงข้ามจะน้อยกว่า 0.2 โวลท์

TEST TERMINATION MEASUREMENT 2, WITH VARYING COMMON MODE VOLTAGE

การทดสอบนี้ทำได้โดย ใช้ส่งสัญญาณในสาย ซึ่งมีตัวต้านทาน 60 โอห์ม ขนานกับตัวต้านทาน 375 โอห์ม 2 ตัว ซึ่งต่อกันแบบอนุกรม รอยต่อตรงกลางของตัวต้านทาน 375 โอห์ม ทั้งสองจะต่อกับชุดส่งแรงดันไฟฟ้าตัวหนึ่ง ซึ่งจะเปลี่ยนจาก -7 โวลท์ ไปเป็น +12 โวลท์ เพื่อแปลง Common Mode Voltages ขนาดของ Generator Output Voltage (Vt) จะไม่น้อยกว่า 1.5 โวลท์ หรือไม่มากกว่า 5.0 โวลท์ บน Voltage Range -7 โวลท์ ถึง +12 โวลท์ การทดสอบจะทำได้สำหรับ Binary State ทั้งสองคือ ทั้ง Vt และ $\bar{V}$t

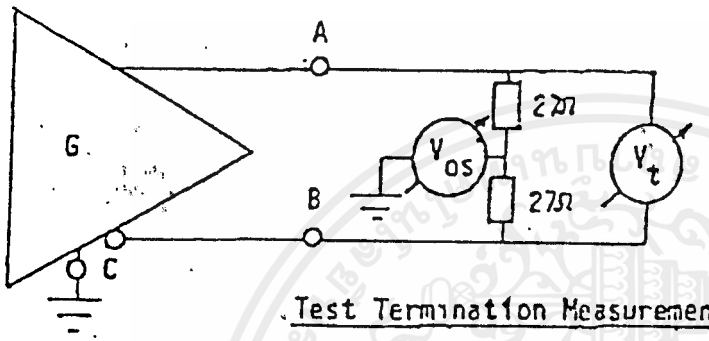
OUTPUT SIGNAL WAVE FORM

ในระหว่างการเปลี่ยนแปลงของ Generator Output ระหว่างการสลับ Binary State (1-0-1-0,etc) ความต่างศักย์ที่วัดข้ามสายทดสอบประกอบด้วยตัวต้านทาน 54 โอห์ม หนึ่งตัว ขนานกับตัวเก็บประจุ 50 picofarad หนึ่งตัว ต่อระหว่างขั้ว Generator Output ซึ่งแรงดันไฟฟ้าจะเปลี่ยนไปมาระหว่าง 0.1 และ 0.9 ของ Vss ภายใน Unit Interval ที่ต่ำกว่า 0.3 หลังจากนั้น Signal Voltage จะไม่เปลี่ยนเกินกว่า 10 % ของ Vss จากค่าสภาวะคงที่ จนกระทั่งการเปลี่ยนแปลง Binary ถัดไปเกิดขึ้น และในขณะนั้นขนาดของ Vt หรือ $\bar{V}$t จะเกิน 5.0 โวลท์ Vss จะถูกอธิบายตามความแตกต่างของแรงดันไฟฟ้า ระหว่างค่าสภาวะคงที่ 2 ค่า ของ Generator Output

RECEIVER CHARACTERISTICS

ลักษณะทางไฟฟ้าของเครื่องรับจะถูกแสดงในความเหมาะสมกับ Load Specifications และกับความไวและการวัดความสมดุลวงจรที่ต้องการจะแสดงผลใน Differential Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



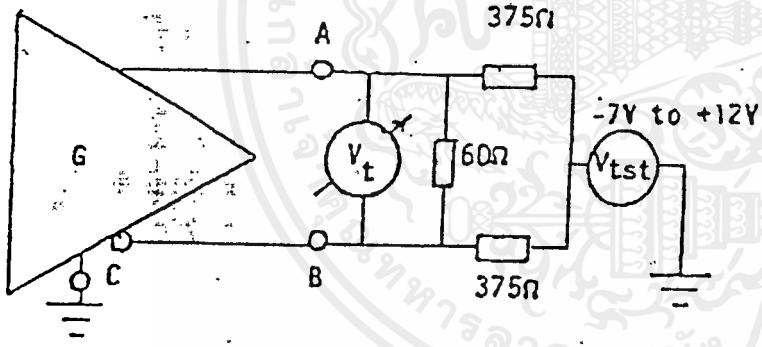
Test Termination Measurement 1

$1.5V < |V_t| < 5.0V$; $1.5V < |\overline{V}_t| < 5.0$

$||V_t| - |\overline{V}_t|| < 0.2V$

$-1.0V < V_{os} < 3.0V$, $-1.0V < \overline{V}_{os} < 3.0V$

$|V_{os} - \overline{V}_{os}| < 0.2V$



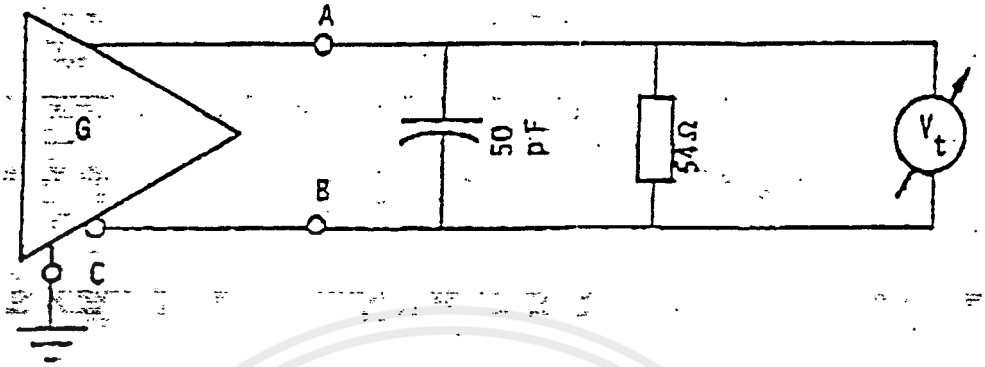
Test Termination Measurement 2

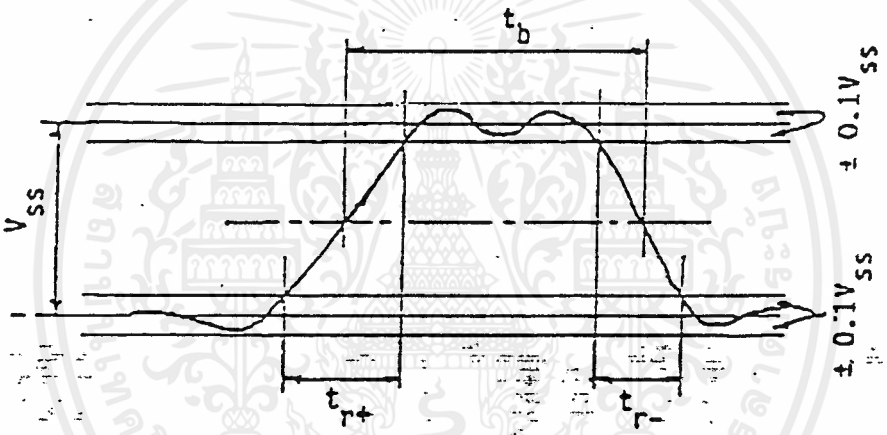
$1.5V < |V_t| < 5.0V$

$1.5V < |\overline{V}_t| < 5.0V$

$||V_t| - |\overline{V}_t|| < 0.2$
At any test voltage

GENERATOR TEST TERMINATION MEASUREMENTS





t_b = Time duration of the unit interval at the applicable data signalling rate

$t_r < 0.3t_b$

V_{ss} = Difference in steady state voltages

$V_{ss} = |V_t - \overline{V}_t|$

GENERATOR OUTPUT SIGNAL WAVEFORM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมี Input Impedance สูง และ Threshold วางอยู่ในบริเวณระหว่าง -0.2 และ +0.2 โวลท์

INPUT SENSITIVITY MEASUREMENT

ย่านทดสอบของ Input Voltages ($V_{A'}$ และ $V_{B'}$) ลดขึ้นที่ขั้ว Input ของเครื่องรับ (A' และ B') จะวัดในส่วน Receiver Common (C) จะอยู่ระหว่าง -7 โวลท์ และ +12 โวลท์ สำหรับการรวมกันของ Input Voltages ของเครื่องรับใด ๆ ภายในย่านทดสอบนี้ เครื่องรับจะถือเอา Binary State ที่ตั้งไว้กับ Applied Differential Input Voltage (Vi) ของ $\pm$0.2 โวลท์ หรือมากกว่า นอกจากนี้เครื่องรับยังจะไม่ได้รับความเสียหาย สำหรับการรวมกันใด ๆ ของ Receiver Input Voltages ภายใน Allowable Range นี้ การทำงานภายนอกของย่านทดสอบนี้ จะไม่ถูกครอบคลุมโดยมาตรฐานนี้ สิ่งที่ต้องคำนึงถึงเกี่ยวกับเครื่องรับมีดังนี้

ขั้ว A' ของเครื่องรับจะเป็นลบในส่วนของขั้ว B' สำหรับสภาวะ Binary 1 (MARK OR OFF)

ขั้ว A' ของเครื่องรับจะเป็นบวกในส่วนของขั้ว B' สำหรับสภาวะ Binary 0 (SPACE-OR ON)

INPUT BALANCE MEASUREMENT

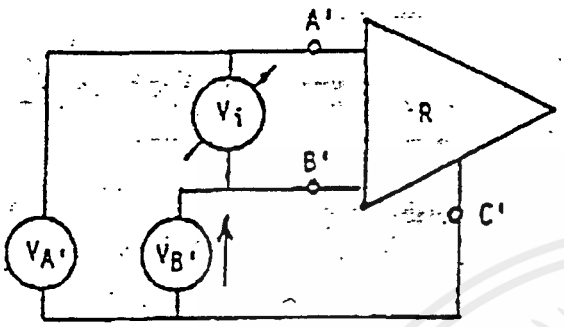
ความสมดุลของลักษณะ Input Voltage Current และ Bias Voltage ของเครื่องรับ จะยังคงอยู่ใน Binary State ที่ตั้งไว้เมื่อใช้ความต่างศักย์ (VR2) $\pm$0.4 volts ผ่านตัวต้านทานคู่จะเท่ากับ 1500/nU.L. ohms ต่อขั้ว Input แต่ละขั้ว ดังแสดงในรูป ด้วย Input Voltages (VR1 และ VR2) ที่ระยะระหว่าง -7 โวลท์ และ +12 โวลท์ เมื่อกระแสของ -VR3 ทวนกลับ Binary state ตรงข้ามจะยังคงอยู่ภายใต้สภาวะเดิม(การเปลี่ยนแปลงค่าความต้านทานขึ้นอยู่กับว่ามี Unit Loads (U.L.) มากเท่าใด หรือขึ้นกับส่วนของ Unit Load ที่ปรากฏในเครื่องรับ)

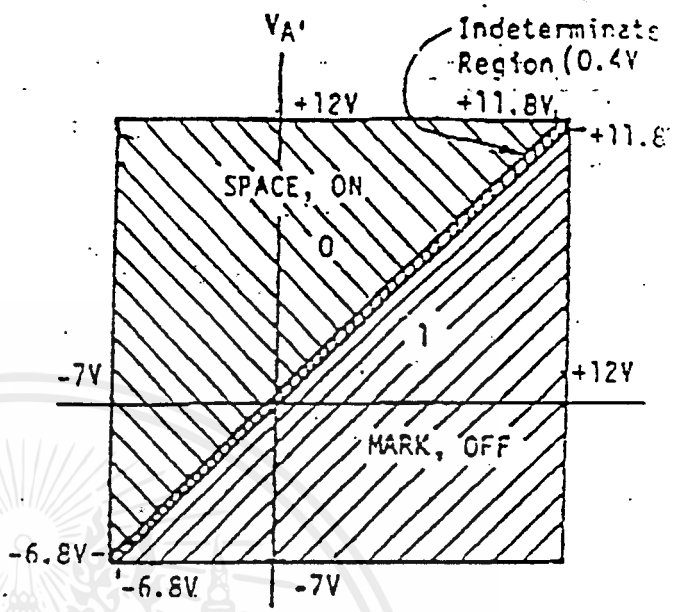
FAULT CONDITION MEASUREMENTS

การทดสอบต่อไปจะแสดงให้เห็นว่าถ้าลไก จะไม่ได้รับความเสียหาย เนื่องจากภาวะความผิดพลาดอย่างเดียว ในขณะที่กลไกหลักของวงจรศักย์ไฟฟ้าเสียไป ก็จะถูกทดสอบโดยส่วนต่าง ๆที่ตามมา ภาวะการเคลื่อนที่อื่น ๆ ที่ยังใช้ได้อยู่อาจจะเป็นตัวบังคับกลไก ภาวะเหล่านี้อาจต้องการ

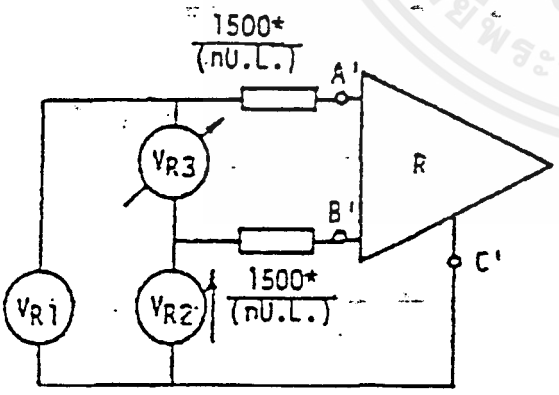
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-485
Page 10

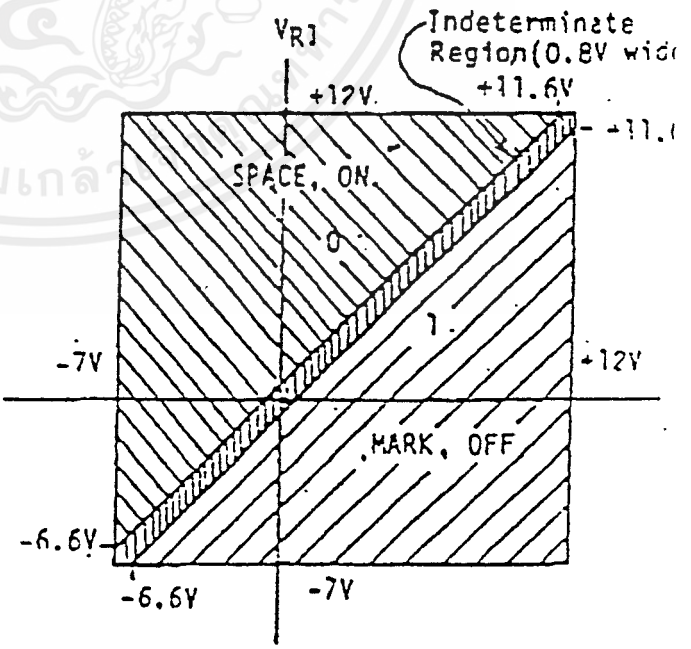




RECEIVER INPUT VOLTAGE RANGE



*Matched



RECEIVER INPUT BALANCE MEASUREMENT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบเพิ่มเติม เพื่อให้ผู้ใช้แน่ใจว่าเกลไกจะทำงานได้อย่างเหมาะสมภายใต้เงื่อนไข ในการปฏิบัติงาน

GENERATOR SHORT CIRCUIT MEASUREMENT

เมื่อขั้ว Output ของ Generator เกิดลัดวงจรกันขึ้น Generator จะไม่ได้รับความเสียหาย

GENERATOR CONTENTION TEST

Generator จะไม่ทนต่อการทำลายเหมือนกับผลของการต่อขั้ว Output ของมัน ต่อแหล่งกำเนิดแรงดันไฟฟ้าหนึ่ง ๆ ซึ่งจะเปลี่ยนจาก -7 ไปเป็น +12 โวลท์ ภายใต้ภาวะ Output, Binary 0, 1 หรือพัลส์บวก

GENERATOR CURRENT LIMIT

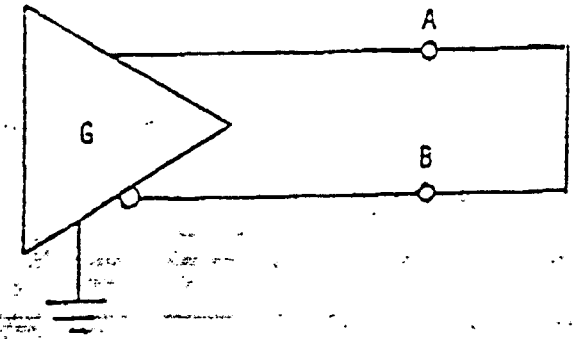
ภายใต้ภาวะการทดสอบการใช้แรงดันไฟฟ้าด้วยอัตราการแกว่งของ Variable Voltage Source เท่ากับหรือน้อยกว่า 1.2 โวลท์ ต่อ Microsecond กระแสสูงสุดในการนำไปยังขั้ว enerator จะไม่เกิน 250 MA. หรือพูดได้ว่า Sinking Generator จะไม่ให้ Composite Current เกิน 250 MA. ถ้าเครื่องกำเนิดไฟฟ้าหลาย ๆ ตัวให้กระแสนี้จะไม่ให้

TRANSIENT OVER VOLTAGE REQUIREMENT

ข้อกำหนดนี้ใช้กับทั้งเครื่องส่งและเครื่องรับ และจำเป็นต่อการเตรียมการป้องกันชั่วคราว ซึ่งอาจเกิดขึ้นบนสายไฟ เมื่อมีกระแสสูงขึ้น เนื่องจาก Single Contending Pair ถูกรบกวน การทดสอบอาจทำได้ทั้งสภาวะ Power On และ Power Off

Passive Generator หรือ Receiver จะต้องสามารถต้านทานโดยไม่มีความผิดพลาด ใช้ระยะเวลาความสั่นสะเทือน 15 microsec ที่ 1 % ต่อรอบจากแหล่งกำเนิด 25 โวลท์ ที่มีความผิด 100 โอห์ม ทั้ง Positive และ Negative Pulses จะใช้ระหว่าง A และ C และระหว่าง B และ C บน Passive Generator และระหว่าง A' และ C' และระหว่าง B' และ C' บนเครื่องรับ ถ้า Passive Device ไม่สามารถทำงานในสภาวะ Passive ระบบจะกลับไปสู่สภาวะ Passive ภายใน 200 ns. ของการ Oscilate

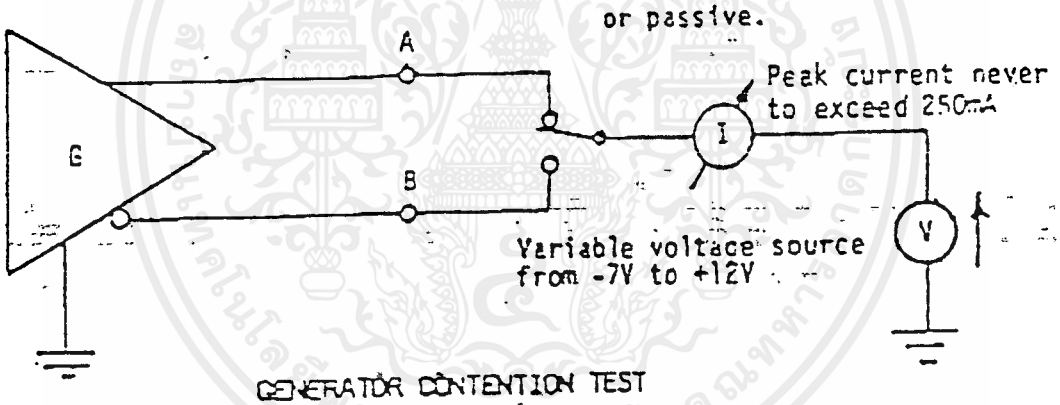
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



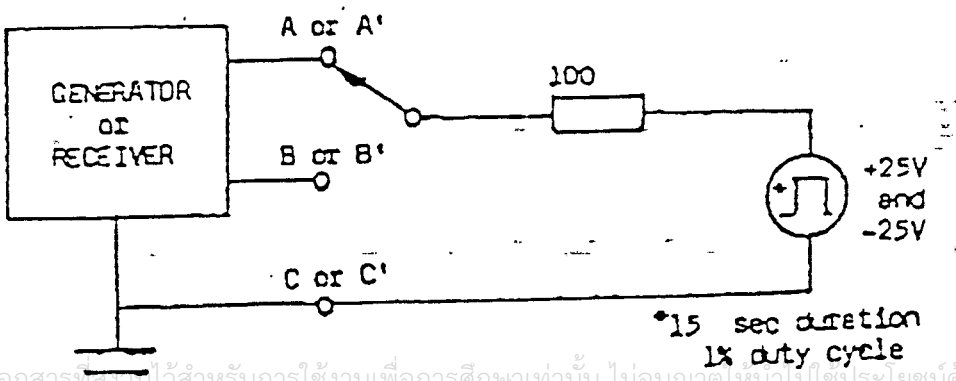
No generator damage when when terminals A and B are shorted together in any state, whether binary 0, 1, or passive.

GENERATOR SHORT CIRCUIT MEASUREMENT

No generator damage when variable voltage source is connected in any state, whether binary 1, 0, or passive.



GENERATOR CONTENTION TEST



TRANSIENT OVER VOLTAGE TEST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Z-80 ไมโครโปรเซสเซอร์

ในปัจจุบันโลกเรา อยู่ในสภาวะที่จะขาดเสียซึ่งการดำรงชีวิตที่สะดวกสบายไม่ได้ จำนวนประชากรที่เพิ่มขึ้นเป็นสาเหตุที่ทำให้ความต้องการในด้านต่าง ๆ เพิ่มขึ้นเป็นเงาตามตัว เพื่อที่จะตอบสนองความต้องการเหล่านี้ จึงจำเป็นที่จะต้องนำเอาเทคโนโลยีใหม่ ๆ มาใช้อย่างหลีกเลี่ยงไม่ได้ คอมพิวเตอร์เป็นสิ่งหนึ่งที่มีบทบาทอย่างมาก ในการพัฒนาชีวิตและความเป็นอยู่ของมนุษย์ให้ดีขึ้น

โดยทั่วไปโครงสร้างพื้นฐานของคอมพิวเตอร์ จะประกอบด้วย

- หน่วยควบคุม (CONTROL UNIT)
- หน่วยความจำ (MEMORY UNIT)
- หน่วยคำนวณ (ARITHMETIC UNIT)

ด้วยการพัฒนาทางด้านเทคโนโลยีที่ทันสมัย ทำให้ชิ้นส่วนต่าง ๆ ที่ประกอบขึ้นเป็นหน่วยย่อย ๆ ภายในเครื่องคอมพิวเตอร์มีขนาดเล็กลง แต่ประสิทธิภาพกลับสูงขึ้นและราคาก็ถูกลงอย่างมาก ทำให้ความต้องการที่จะนำเอาเครื่องคอมพิวเตอร์มาใช้ในชีวิตประจำวันมีมากขึ้นเป็นลำดับ

ในปัจจุบันนี้ เราสามารถนำเอาวงจรอิเล็กทรอนิกส์ ที่ยุ่งยากและซับซ้อนมาบรรจุลงบนแผ่นวงจรเดียว ที่มีขนาดเล็กมาก ซึ่งเรียกว่า LSI (LAST SCALE INTEGRATED CIRCUIT) และบรรจุอยู่ในตัวถัง ซึ่งต่อขาออกมาเพื่อใช้ในการติดต่อกับวงจรภายนอก สิ่งนี้เรียกว่า "ชิพ" (CHIP) องค์ประกอบย่อย ๆ ในไมโครคอมพิวเตอร์ทั้งหมด จะประกอบขึ้นจากชิพเหล่านี้ เช่น หน่วยความจำ ประเภท ROM (READ ONLY MEMORY), RAM (RANDOM ACCESS MEMORY) อุปกรณ์สนับสนุน (CHIP SUPPORT) ต่าง ๆ และสิ่งที่เป็นหัวใจของระบบไมโครคอมพิวเตอร์ คือ หน่วยประมวลผลกลาง หรือ CPU (CENTRAL PROCESSING UNIT) ซึ่งภายในประกอบไปด้วยส่วนต่าง ๆ ดังบล็อกไดอะแกรม รูปที่1

ซึ่งแต่ละบล็อกมีลักษณะการทำงานดังต่อไปนี้ คือ

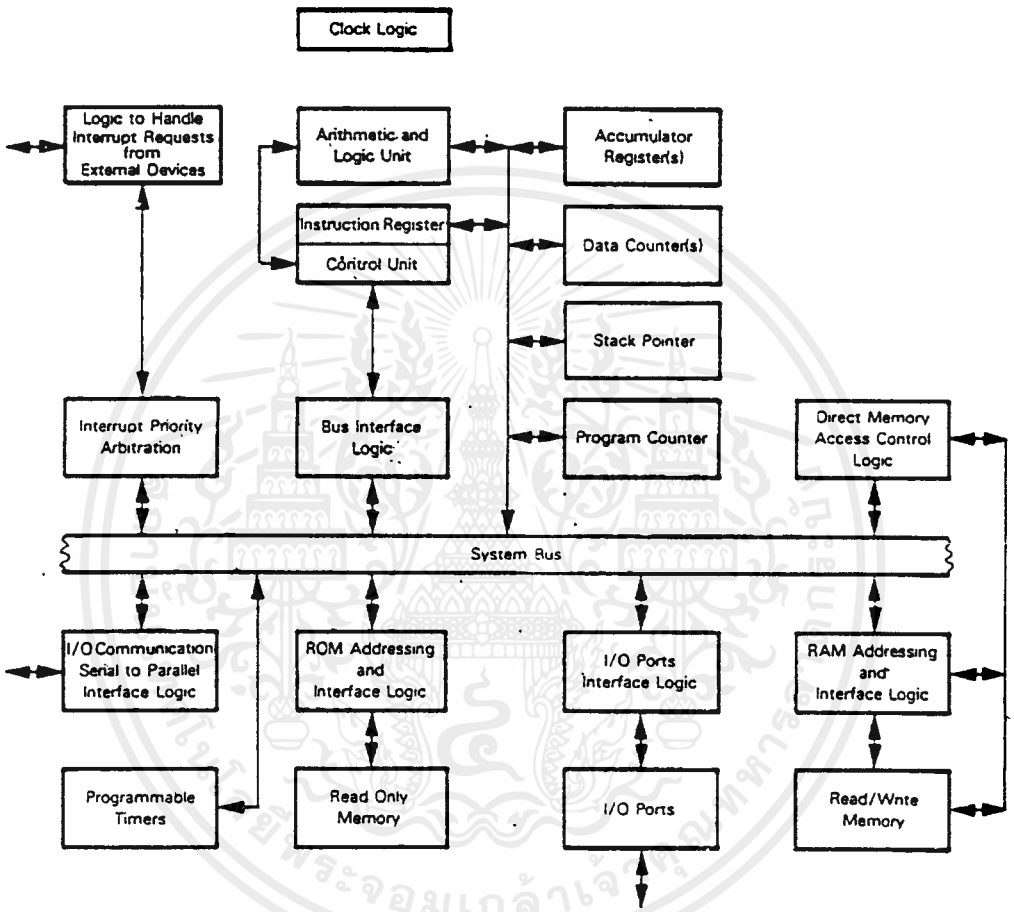
1. ARITHMETICS LOGIC UNIT (ALU) เป็นหน่วยที่ทำหน้าที่ในการคำนวณฟังก์ชั่นพื้นฐานทางคณิตศาสตร์ และการกระทำฟังก์ชันทางลอจิก เช่น AND และ OR ALU. จะสามารถทำหน้าที่ได้อย่างมีประสิทธิภาพมากเพียงใดนั้นขึ้นอยู่กับการออกแบบวงจรภายในของ ALU.

2. CONTROL UNIT เป็นหน่วยที่ทำหน้าที่ในการส่งสัญญาณไปควบคุมอุปกรณ์ต่าง ๆ ที่ต่อเชื่อมกับ CPU ให้ทำงานร่วมกันได้อย่างถูกต้อง

3. DATA BUS เป็นบัสสองทิศทาง (BI-DIRECTIONAL) ที่ใช้ในการส่งผ่านข้อมูลระหว่าง CPU กับอุปกรณ์อื่น ๆ ภายในระบบ จำนวนเส้นของบัสข้อมูล (DATA BUS) จะขึ้นอยู่กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ชนิดของ CPU เช่นในกรณีของ Z-80 CPU จะส่งผ่านข้อมูลทีละ 8 บิท ดังนั้นจะมีจำนวนเส้น-ของบัสข้อมูล 8 เส้น



รูปที่ 16 แสดงบล็อกไดอะแกรมของ Z-80

4. CONTROL BUS หรือ บัสควบคุม เป็นบัสทางเดียว (UNI-DIRECTIONAL BUS) ที่ใช้ในการส่งผ่านสัญญาณควบคุมให้กับอุปกรณ์ต่าง ๆ ในระบบ

5. ADDRESS BUS เป็นบัสทางเดียว ใช้ส่งผ่านค่าแอดเดรสจาก CPU ออกไปยังหน่วยความจำ เพื่อระบุตำแหน่งที่ต้องการรับหรือส่งข้อมูล หรือใช้ระบุตำแหน่งของพอร์ท I/O (INPUT OUTPUT PORT) ที่ CPU ต้องการติดต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

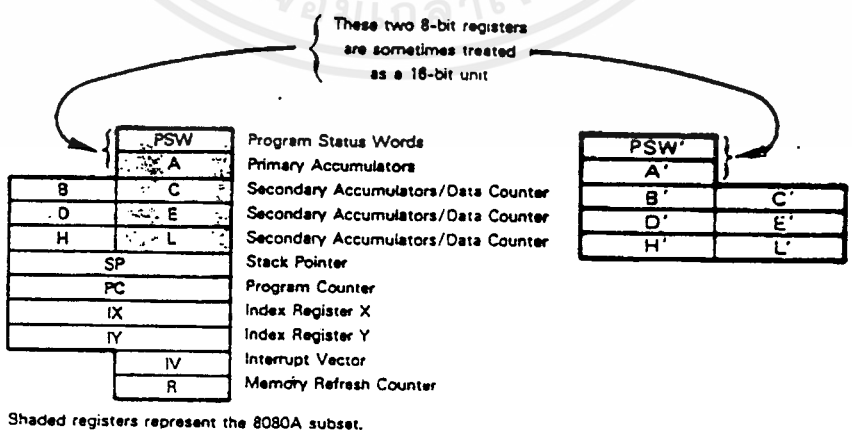
ต่อไปนี้เราจะกล่าวถึงรายละเอียดของ Z-80 CPU ซึ่งเป็นไมโครโปรเซสเซอร์ขนาด 8 บิท ที่นิยมใช้อย่างแพร่หลายในปัจจุบัน

Z-80 ไมโครโปรเซสเซอร์เป็น CPU ที่ผลิตจากบริษัท ZILOG INC. โดยกลุ่มวิศวกรชุดเดียวกับที่ผลิต 8080 CPU ของบริษัท INTEL COORPERATION, Z-80 CPU ได้รับการพัฒนาให้มีข้อดีเหนือกว่า 8080 เช่น มีชุดคำสั่งมากถึง 158 คำสั่ง โดยรวมชุดคำสั่งเดิมของ 8080 ไว้ 80 คำสั่ง นอกจากนี้ Z-80 ยังมีรีจิสเตอร์มากกว่าใน 8080 ถึง 12 ตัว และ 8080 เพียงตัวเดียวก็ยังไม่สามารถที่จะนำไปใช้งานได้ ต้องต่อกับอุปกรณ์สนับสนุนอีก 2 ตัว คือ CLOCK GENERATOR CHIP, SYSTEM CONTROLER CHIP รวมกันเรียกว่า THREE CHIP PROCESSOR แต่ใน Z-80 CPU ได้รวมเอาลักษณะพื้นฐานเหล่านี้ไว้ในชิพเดียวกัน และเพิ่มประสิทธิภาพทาง HARDWARE, SOFTWARE และการ INTERFACE ให้สูงขึ้น

Z-80 ไมโครโปรเซสเซอร์เพียงชิ้นเดียวไม่สามารถทำงานเป็นระบบคอมพิวเตอร์ได้ ต้องอาศัยอุปกรณ์อื่น ๆ อีก 2 ส่วน คือ หน่วยความจำ (MEMORY) และหน่วยรับส่งข้อมูลเข้าออก (I/O DEVICE) ซึ่งในการทำงานตามคำสั่งจากโปรแกรมที่ป้อนเข้ามา Z-80 CPU จะต้องทำการโอนย้ายคำสั่งหรือข้อมูลระหว่างหน่วยความจำ กับรีจิสเตอร์ (REGISTER) ก่อนอื่นเราจะกล่าวถึงรีจิสเตอร์ภายในของ Z-80 เสียก่อน

– รีจิสเตอร์ต่าง ๆ ใน Z-80 CPU

Z-80 CPU จะประกอบไปด้วยรีจิสเตอร์ถึง 22 ตัวดังในรูป รีจิสเตอร์เหล่านี้จะแบ่งได้เป็น 2 กลุ่มคือ รีจิสเตอร์ที่ทำหน้าที่ทั่ว ๆ ไป และรีจิสเตอร์ที่ทำหน้าที่เฉพาะงาน



รูปที่ 17 แสดงรีจิสเตอร์ต่าง ๆ ภายใน Z-80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **รีจิสเตอร์ที่ทำหน้าที่ทั่ว ๆ ไป** แบ่งเป็นรีจิสเตอร์หลัก ได้แก่ A, B, C, D, E, H และ L มีความจุขนาด 8 บิท รีจิสเตอร์เหล่านี้ใช้เก็บข้อมูลชั่วคราว นอกจากนี้ยังสามารถรับข้อมูลจากหน่วยความจำหรืออาจจะทำการย้ายข้อมูลไปเก็บไว้ในหน่วยความจำก็ได้ และรีจิสเตอร์สำรอง ได้แก่ A', B', C', D', E', H' และ L' ซึ่งเป็นรีจิสเตอร์ที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์หลัก ในกรณีที่ต้องใช้รีจิสเตอร์หลักในการทำงานอย่างอื่นก่อน ดังนั้นรีจิสเตอร์กลุ่มนี้จึงไม่สามารถกระทำขบวนการทางคณิตศาสตร์และลอจิกได้

รีจิสเตอร์ A เรียกว่า แอคคิวมูเลเตอร์ (ACCUMULATOR) ทำหน้าที่เก็บข้อมูลชั่วคราวที่ได้จากการทำขบวนการทางคณิตศาสตร์ เช่น บวกหรือลบข้อมูล 2 จำนวน ผลลัพท์ที่ได้จะเก็บไว้ในรีจิสเตอร์ A นี้ นอกจากนี้ ในการปฏิบัติตามคำสั่งที่ใช้กับข้อมูล 16 บิท Z-80 จะนำเอารีจิสเตอร์แฟลก "F" (FLAG REGISTER) มาใช้ร่วมกับรีจิสเตอร์ A เรียกว่า คู่รีจิสเตอร์ AF ซึ่งมีขนาด 16 บิท นอกจากนี้ยังมีคู่รีจิสเตอร์ 16 บิท อื่น ๆ อีก คือ BC, DE และ HL

2. **รีจิสเตอร์ที่ใช้งานเฉพาะอย่าง** ได้แก่ รีจิสเตอร์ I, R, IX, IY, SP และ PC ซึ่งทำหน้าที่ต่าง ๆ ดังนี้

รีจิสเตอร์ I (INTERRUPT PAGE ADDRESS REGISTER) เมื่อมีการอินเตอร์รัพท์เกิดขึ้น จำเป็นจะต้องบอกตำแหน่งของหน่วยความจำที่เก็บโปรแกรมตอบสนองการอินเตอร์รัพท์ รีจิสเตอร์ I จะทำหน้าที่เก็บค่า 8 บิทบนของตำแหน่งข้อมูลในหน่วยความจำ ส่วนค่า 8 บิทล่างจะป้อนจากภายนอกให้แก่ CPU ค่าทั้งสองจะประกอบกันเป็นค่าแอดเดรสที่ระบุตำแหน่งของโปรแกรมการตอบสนองการอินเตอร์รัพท์

รีจิสเตอร์ R (MEMORY REFRESH REGISTER) เป็นรีจิสเตอร์ขนาด 7 บิทที่ถูกใช้ในการรีเฟรช (REFRESH) DYNAMIC RAM และค่ารีจิสเตอร์ R จะเพิ่มขึ้นเองโดยอัตโนมัติ ในทุกครั้งที่มีการเฟทช์คำสั่งจากหน่วยความจำ

รีจิสเตอร์ IX และ IY (INDEX REGISTER) เป็นรีจิสเตอร์ที่มีขนาด 16 บิท มีประโยชน์ใช้บ่งบอกตำแหน่งในหน่วยความจำแบบ INDEX ADDRESSING MODE โดยจะกำหนดให้ค่าใน INDEX REGISTER เป็นค่าอ้างอิง แล้วใช้คำสั่งบ่งบอกว่าตำแหน่งของข้อมูลที่ต้องการอยู่ห่างจากค่าอ้างอิงนี้เท่าใด โดยจะบอกค่าระยะห่างในรูปของ TWO COMPLEMENT

รีจิสเตอร์ SP (STACK POINTER) มีขนาด 16 บิท ในหน่วยความจำชนิด RAM จะมีส่วนหนึ่งที่ถูกกำหนดให้เป็นที่เก็บข้อมูลแบบ LIFO (LAST IN FIRST OUT) เราสามารถที่จะเก็บข้อมูลลงบนสแตกโดยใช้คำสั่ง PUSH และเมื่อต้องการดึงข้อมูลออกจากสแตก ต้องใช้คำสั่ง POP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ PC (PROGRAM COUNTER) เป็นรีจิสเตอร์ขนาด 16 บิทที่ใช้ในการเก็บตำแหน่งของหน่วยความจำที่ CPU จะเฟทช์ (FETCH) คำสั่งหลังจากที่เฟทช์คำสั่งเรียบร้อยแล้ว ค่าในรีจิสเตอร์ PC จะเพิ่มขึ้น และจะชี้ไปยังตำแหน่งของคำสั่งถัดไป เราสามารถเปลี่ยนแปลงค่าใน PC ได้โดยใช้คำสั่ง CALL หรือ JUMP

รีจิสเตอร์ F (FLAG REGISTER) ประกอบด้วย

SIGN FLAG (S) : แฟลกเครื่องหมาย

ZERO FLAG (Z) : แฟลกศูนย์

HALF CARRY FLAG (H) : แฟลกทดครึ่ง

PARITY/OVERFLOW FLAG (P/V) : แฟลกพาริตี้หรือโอเวอร์โฟล์ว

SUBTRACT FLAG (N) : แฟลกลบ

CARRY FLAG (C) : แฟลกตัวทด

ผู้ผลิต Z 80 ได้นำเอาแฟลกเหล่านี้ มาประกอบร่วมกับบิทว่าง (X : ไม่มีความหมาย) อีก 2 บิทเพื่อทำเป็นรีจิสเตอร์ขนาด 8 บิท สำหรับรายละเอียดของแฟลกเหล่านี้จะไม่ขอกล่าวถึง

รายละเอียดของขา Z 80 (Z 80 PIN OUTS)

A0-A15 (ADDRESS BUS) : เป็นขาสัญญาณเอาท์พุทแบบ TRI-STATE ใช้บ่งบอกตำแหน่งหน่วยความจำได้ถึง 2^{16} = 65536 ตำแหน่ง A0-A7 จะแสดงตำแหน่งของพอร์ทที่ Z-80 ต้องการติดต่อด้วย นอกจากนี้ขา A0-A6 จะให้ค่ารีเฟรชแอดเดรสออกมาขณะที่ Z-80 ให้สัญญาณรีเฟรช

D0-D7 (DATA BUS) : เป็นขาสัญญาณอินพุท/เอาท์พุท TR-_STATE แบบสองทิศทาง ซึ่งเป็นทางผ่านของข้อมูลระหว่าง Z-80 กับหน่วยความจำและอุปกรณ์ I/O

$\overline{M1}$ (MACHINE CYCLE ONE) : เป็นขาเอาท์พุทแอคทีฟที่ลอจิก "0" ขา $\overline{M1}$ นี้จะแอคทีฟขณะที่ Z-80 ทำการเฟทช์ออฟโคดของคำสั่ง ในกรณีที่คำสั่งที่จะเฟทช์เข้ามานั้นมีขนาด 2 ไบท์ $\overline{M1}$ จะแอคทีฟในหุน ๆ ไซเคิลการเฟทช์แต่ละไบท์

$\overline{MREQ}$ (MEMORY REQUEST) : เป็นสายเอาท์พุทแบบ TRI-STATE แอคทีฟที่ลอจิก-"0" เพื่อเป็นการบ่งบอกว่า Z-80 กำลังกระทำการติดต่อกับหน่วยความจำ

$\overline{IORQ}$ (INPUT/OUTPUT REQUEST) : เป็นสายเอาท์พุทแบบ TRI-STATE จะแอคทีฟที่ลอจิก "0" เพื่อเป็นการบ่งบอกว่า Z-80 กำลังทำการติดต่อกับอุปกรณ์ I/O และเมื่อ $\overline{IORQ}$ และ $\overline{M1}$ แอคทีฟทั้งคู่ จะเป็นการบ่งบอกการตอบรับการอินเทอร์รัพท์ (INTERRUPT ACKNOWLEDGE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{RD}$ (MEMORY READ) : เป็นขาเอาท์พุท TRI-STATE จะแอคทีฟที่ลอจิก "0" เมื่อ Z-80 ต้องการอ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์ I/O และ Z-80 จะรับข้อมูลจากบัสข้อมูลเข้าไปเมื่อสัญญาณนี้เปลี่ยนระดับลอจิกจาก "0" เป็น "1"

$\overline{WR}$ (MEMORY WRITE) : เป็นขาเอาท์พุทแบบ TRI-STATE จะแอคทีฟที่ลอจิก "0" เมื่อ Z-80 ต้องการส่งข้อมูลออกไปให้หน่วยความจำหรืออุปกรณ์ I/O

$\overline{RFSH}$ (REFRESH) : เป็นขาเอาท์พุท จะแอคทีฟเมื่อ 7 บิทล่าง (A0-A6) ของบัสแอดเดรสให้ค่ารีเฟรชออกมา

$\overline{HALT}$ (HALT STATE) : เป็นขาเอาท์พุท จะแอคทีฟที่ลอจิก "0" เมื่อ Z-80 อยู่ในสภาวะของการ HALT คือ CPU จะทำคำสั่ง NOP (NO OPERATION) เพื่อให้เกิดการรีเฟรชได้และ Z-80 จะหลุดพ้นจากสภาวะการ HALT เมื่อได้รับการรีเซ็ทหรือถูกอินเทอร์รัพท์

$\overline{WAIT}$: เป็นขาอินพุท แอคทีฟที่ลอจิก "0" และจะมีการตรวจสอบสัญญาณนี้ที่ขอบขาลงของคล็อกลูกที่ 2 ของทุก ๆ MACHINE CYCLE เมื่อมีการตรวจพบว่าขาอินพุทนี้แอคทีฟ จะมีการแทรก WAIT STATE ให้กับแต่ละ MACHINE CICLE เพื่อเป็นการรอให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ Z-80 และ Z-80 จะแทรก WAIT STATE จนกว่าจะมีการตรวจสอบพบว่าขา WAIT จะมีลอจิกเป็น "1"

$\overline{INT}$ (INTERRUPT REQUEST) : เป็นขาอินพุท แอคทีฟที่ลอจิก "0" Z-80 จะตรวจสอบระดับสัญญาณที่ขานี้ทุก ๆ การสิ้นสุดของ INSTRUCTION CYCLE (LAST STATE)

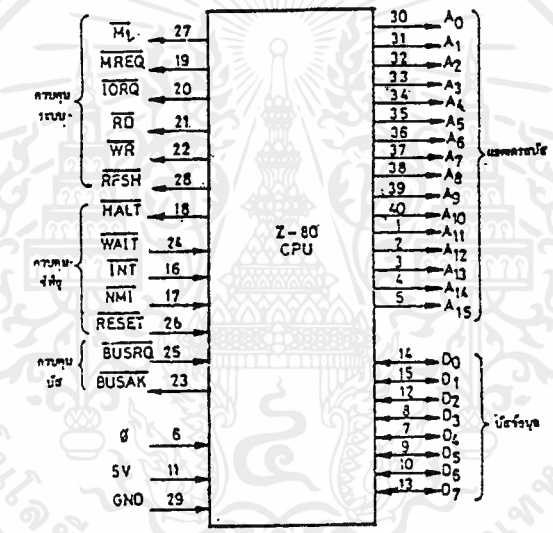
$\overline{NMI}$ (NON MASKABLE INTERRUPT) : เป็นขาอินพุทแอคทีฟที่ลอจิก "0" สัญญาณ NON MASKABLE INTERRUPT เป็นสัญญาณที่มีระดับความสำคัญ ในการขออินเทอร์รัพท์สูงกว่าสัญญาณ INTERRUPT REQUEST Z-80 จะตอบรับการอินเทอร์รัพท์ชนิดนี้เสมอโดยที่เราไม่สามารถ DISABLE ได้ด้วย SOFTWARE

$\overline{RESET}$: เป็นขาอินพุท แอคทีฟที่ลอจิก "0" สัญญาณนี้จะทำการ INITIALIZE CPU โดยทำการรีเซ็ท INTERRUPT FLIP-FLOP และเซ็ทค่าในโปรแกรมเคาน์เตอร์ (PROGRAM - COUNTER) ให้เป็น 0000H และในสภาวะการรีเซ็ทนี้ บัสแอดเดรสและบัสข้อมูลจะอยู่ในสภาวะ HIGH IMPEDANEC และสัญญาณควบคุมต่าง ๆ จะอยู่ในสภาวะ INACTIVE

$\overline{BUSRQ}$ (BUS REQUEST) : เป็นขาอินพุทที่แอคทีฟที่ลอจิก "0" สัญญาณ BUS REQUEST เป็นสัญญาณที่มีลำดับความสำคัญสูงกว่าสัญญาณ NON MASKABLE INTERRUPT และมีการตรวจสอบสัญญาณนี้ทุก ๆ การสิ้นสุดของ MACHINE CYCLE อุปกรณ์ภายนอกจะให้สัญญาณนี้แก่ Z-80 เมื่อต้องการใช้บัสข้อมูลและบัสแอดเดรส โดยเปรียบเสมือนว่าเป็นการถอด Z-80 ออกจากระบบบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{BUSAK}}$ (BUS ACKNOWLEDGE) : เป็นขาอินพุทแอคทีฟที่ลอจิก "0" ขานี้จะแอคทีฟ-เมื่อ Z-80 ตอบสนองการต่อสัญญาณ BUS REQUEST และจะทำให้บัสข้อมูล, บัสควบคุม และบัสแอดเดรสมีสภาวะเป็น HIGH IMPEDANCE ซึ่งทำให้อุปกรณ์ภายนอกใช้บัสเหล่านี้ได้โดยไม่มีผลต่อ CPU



รูปที่ 18 ลักษณะของขาไอซี Z-80 CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Z-80 กับอินพุทและเอาท์พุท

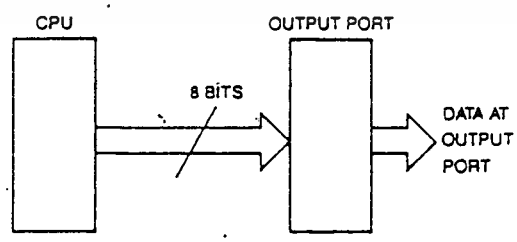
ในขบวนการทำงานของระบบไมโครโปรเซสเซอร์นั้น มีความจำเป็นอย่างยิ่งที่จะต้องทำการติดต่อกับอุปกรณ์อื่น ๆ เสมอ นอกเหนือจากหน่วยความจำแล้ว ในบางครั้ง CPU ยังมีความจำเป็นที่จะต้องทำการอ่านข้อมูลจากอุปกรณ์อินพุทและส่งข้อมูลออกไปทางอุปกรณ์เอาท์พุท ในที่นี้จะกล่าวถึงวิธีการเชื่อมต่อระหว่าง Z-80 เข้ากับอุปกรณ์ I/O (INPUT/OUTPUT DEVICE) นอกจากนี้เราจะกล่างถึงการสร้างพอร์ท I/O (INPUT/OUTPUT DEVICE) ทั่ว ๆ ไป โดยใช้อุปกรณ์ประเภท DISCRETE

ในที่นี้จะกล่าวถึงขบวนการอินพุทและเอาท์พุท เนื่องจากการติดต่อกับอุปกรณ์ I/O นั้น มีลักษณะคล้ายคลึงกับการติดต่อกับ STATIC RAM หากมีความเข้าใจเนื้อหาอย่างลึกซึ้งแล้ว จะเห็นว่าขบวนการติดต่อระหว่าง CPU กับ STATIC RAM และการติดต่อระหว่าง CPU กับอุปกรณ์ I/O นั้นมีลักษณะคล้ายคลึงกันมาก

ขบวนการอินพุทและเอาท์พุทของ Z-80

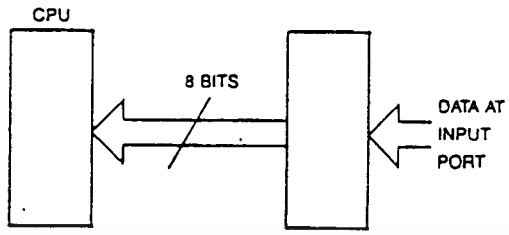
สัญญาณทางไฟฟ้าที่ใช้ในระบบฮาร์ดแวร์ในขณะที่ Z-80 ทำการติดต่อกับอุปกรณ์ I/O นั้น จะประกอบไปด้วยสัญญาณจากบัสแอดเดรส (ADDRESS BUS), สัญญาณจากบัสข้อมูล (DATA BUS) และสัญญาณจากบัสควบคุม (CONTROL BUS)

รูป19 ก.และ ข. จะแสดงบล็อกไดอะแกรมของการส่งข้อมูลขนาด 8 บิท ให้กับอุปกรณ์เอาท์พุทและรับข้อมูลจากอุปกรณ์อินพุท สำหรับรายละเอียดจะกล่าวถึงในภายหลัง



รูป19 ก. แสดงบล็อกไดอะแกรมของพอร์ทเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 19 ข. แสดงบล็อกไดอะแกรมของพอร์ทอินพุท

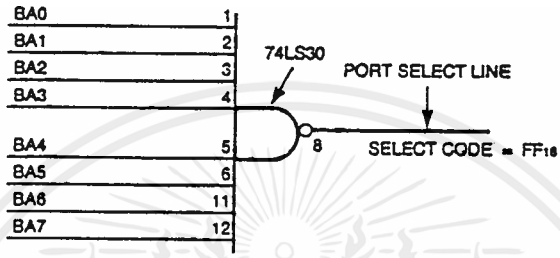
การบ่งบอกตำแหน่งของพอร์ท (PORT ADDRESS)

ในระบบของ Z-80 นั้นจะมีบัสแอดเดรสอยู่ 16 เส้นแต่จะถูกใช้ในการบ่งบอกตำแหน่งของพอร์ท I/O เพียง 8 เส้นเท่านั้น ซึ่งก็คือ Z-80 สามารถจะติดต่อกับ I/O ขนาด 8 บิท ได้ถึง 256 พอร์ท (2^8=256) โดยที่ระบบที่ใช้บ่งบอกตำแหน่งของพอร์ทนั้น จะต้องเป็นลักษณะการวาง I/O แบบมาตรฐาน (STANDARD I/O MAP) และแต่ละพอร์ทต้องใช้แอดเดรส เพียงแอดเดรสเดียวในการบ่งบอกตำแหน่ง (1 แอดเดรส = 1 พอร์ท) เท่านั้น แต่ถ้าเป็นการเลือกพอร์ทแบบ LINEAR SELECT I/O คือสายแอดเดรส 1 เส้นใช้เลือก I/O 1 พอร์ท ก็จะมีพอร์ท I/O เพียง 8 พอร์ท LINEAR SELECT I/O นี้เหมาะกับระบบที่มีอุปกรณ์ I/O จำนวนน้อย

เนื่องจากการบ่งตำแหน่งของ I/O และของหน่วยความจำในระบบของ Z-80 ใช้บัสแอดเดรสชุดเดียวกัน ไมโครโปรเซสเซอร์จึงต้องมีสัญญาณ IORQ (INPUT/OUTPUT REQUEST) แยกออกจาก MREQ (MEMORY REQUEST) เราทราบมาแล้วว่าในระบบหน่วยความจำนั้น ไม่จำเป็นต้องใช้บัสแอดเดรสทั้ง 16 เส้น (A0-A15) ในระบบของ I/O ก็ไม่จำเป็นที่ต้องใช้สายแอดเดรสทั้ง 8 เส้นเช่นกัน ตัวอย่างเช่น ระบบของเราต้องการใช้พอร์ท I/O เพียง 5 พอร์ท จะใช้สายแอดเดรส 3 เส้นก็พอ เพราะสายแอดเดรส 3 เส้นนั้นสามารถบ่งบอกตำแหน่งที่แตกต่างกันได้ถึง 8 ตำแหน่ง (2^3) แต่พอร์ท I/O ที่เราพิจารณาในที่นี้เราจะใช้บัสแอดเดรสทั้ง 8 เส้นในการถอดรหัสเลือกพอร์ท

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละพอร์ท I/O ในระบบไมโครโปรเซสเซอร์จะตอบสนองเพียงตำแหน่งเดียวของระบบบัสแอดเดรส (A0-A7) ตำแหน่งของพอร์ท I/O ที่ไมโครโปรเซสเซอร์ติดต่อด้วยนั้นจะถูกเรียกว่า "พอร์ทแอดเดรส" (PORT ADDRESS) รูปที่ 20 แสดงวงจรที่ใช้ในการถอดรหัสเลือกพอร์ทแอดเดรส 0FFH นี้



รูปที่ 20 แสดงวงจรถอดรหัสเพื่อสร้างสัญญาณ PORT SELECT โดยมีตำแหน่ง PORT อยู่ที่ 0FFH

จากไดอะแกรมในรูป20 นี้ เราจะเห็นว่าเอาท์พุทขา 8 ของ IC 74LS30 จะมีลอจิกเป็น "0" เมื่อขาอินพุทของ GATE ทั้งหมดมีลอจิกเป็น "1" สังเกตุว่าเอาท์พุทขา 8 ของ 74LS30 (PORT SELECT LINE) จะให้แอคทีฟที่ลอจิก "0" เมื่อบัสแอดเดรสมีลอจิกเท่ากับรหัสของพอร์ทที่จะเลือกเท่านั้น สำหรับตัวอย่างนี้จะให้ PORT SELECT LINE แอคทีฟที่ลอจิก "0"

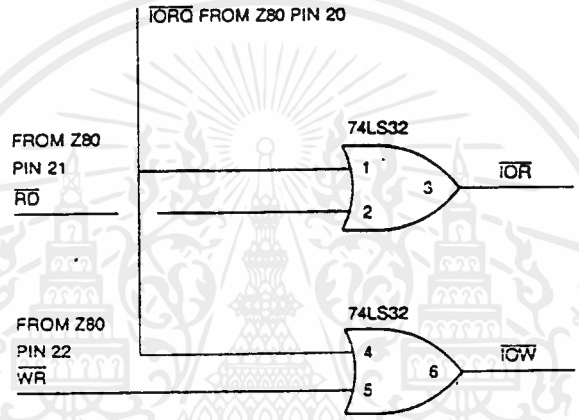
อย่างไรก็ตาม PORT SELECT LINE สามารถที่จะแอคทีฟได้ ในขณะที่ Z-80 ไม่ต้องการติดต่อกับ I/O ทั้งนี้ เนื่องจากระบบหน่วยความจำใช้บัสแอดเดรสชุดเดียวกับระบบของ I/O นั่นเอง ยกตัวอย่างเช่น ในการที่ไมโครโปรเซสเซอร์ต้องการที่จะอ่านข้อมูลจากหน่วยความจำที่ตำแหน่ง XXFFH, PORT SELECT LINE ก็จะแอคทีฟด้วย เพราะบัสแอดเดรส A0-A7 จะมีค่า-0FFH ซึ่งตรงกับรหัสของ PORT SELECT LINE ดังนั้นเราจึงจำเป็นต้องสร้างสัญญาณควบคุมขึ้นมา เพื่อเป็นการบ่งบอกว่าขณะนั้นไมโครโปรเซสเซอร์ต้องการติดต่อกับหน่วยความจำหรือ I/O

การสร้างสัญญาณควบคุม $\overline{IOW}$ และ $\overline{IOR}$

ในขณะที่ z-80 กระทำขบวนการติดต่อกับ I/O ขา $\overline{IORQ}$ ของ Z-80 จะให้ลอจิก "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมา (สังเกตุว่า คล้ายกับการติดต่อกับระบบหน่วยความจำ ซึ่งให้ขา $\overline{MREQ}$ เป็นลอจิก "0" ออกมา) เราจะใช้สัญญาณนี้ร่วมกับสัญญาณ $\overline{RD}$ และ $\overline{WR}$ เพื่อสร้างสัญญาณ $\overline{IOR}$ (I/O READ : เมื่อต้องการรับข้อมูลจากอุปกรณ์อินพุท) และสัญญาณ $\overline{IOW}$ (I/O WRITE : เมื่อต้องการส่งข้อมูลออกไปทางอุปกรณ์เอาท์พุท)

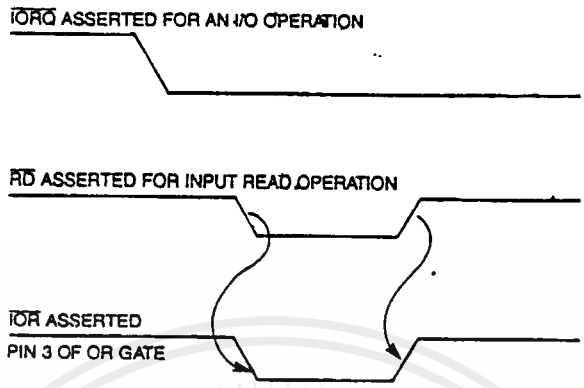


รูปที่21 แสดงวงจรการสร้างสัญญาณ IOR และ IOW

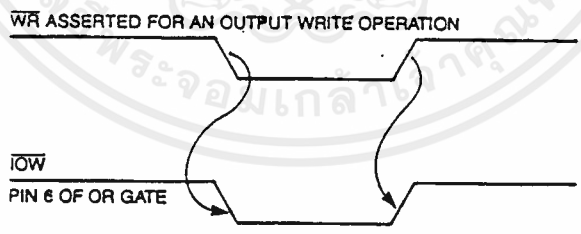
รูปที่ 21 แสดงให้เห็นว่าเราจะสร้างสัญญาณ $\overline{IOR}$ และ $\overline{IOW}$ ในระบบของ Z-80 ได้อย่างไร สังเกตุว่าสัญญาณเหล่านี้จะแอคทีพระหว่างกระทำการติดต่อกับอุปกรณ์ I/O เท่านั้น รูปที่-22 ก. และ 22 ข. แสดงไดอะแกรมเวลาของขบวนการดังกล่าว

สัญญาณการเขียนพอร์ท(PORT WRITE)ของระบบ Z-80 จะถูกใช้เป็นสัญญาณ WRITE ENABLE STROBE ของพอร์ทเอาท์พุทที่ถูกเลือก สัญญาณการเขียนพอร์ทนี้จะเกิดขึ้นได้ก็ต่อเมื่อ PORT SELECT LINE และ $\overline{IOW}$ แอคทีพพร้อมกัน (มีลอจิกเป็น "0" พร้อมกัน) สัญญาณการเขียนพอร์ท จะเป็นสัญญาณที่กระตุ้นให้อุปกรณ์เอาท์พุท รับข้อมูลจากไมโครโปรเซสเซอร์ไปแลทช์ (LATCH)เอาไว้ หรือเขียนข้อมูลลงบนพอร์ทเอาท์พุทที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 22 ก. แสดงไดอะแกรมเวลาของการเกิดสัญญาณ IOR

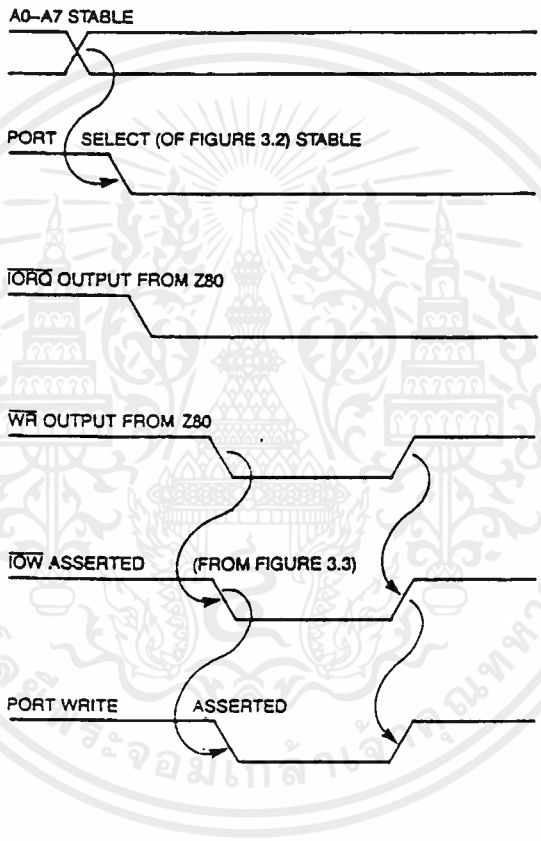


รูป 22 ข. แสดงไดอะแกรมเวลาของการเกิดสัญญาณ IOW

ไดอะแกรมเวลาในรูปที่ 23 แสดงลำดับเวลาของสัญญาณที่ปรากฏขึ้นในขณะข้อมูลถูกส่งไป ยังในรูปนี้จะเห็นว่าทุกๆสัญญาณยกเว้นสัญญาณ WR จะเป็นสัญญาณที่จะคงอยู่จนกว่าขบวนการที่ติดต่อ

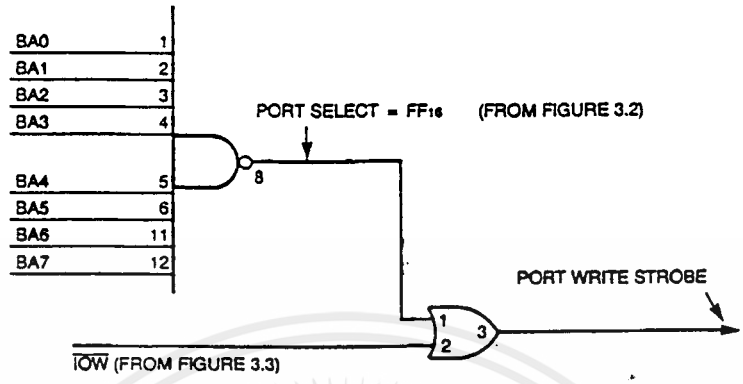
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ I/O จะสิ้นสุดลง ดังนั้นเราจะเขียนวงจรได้ดังรูป 24 วงจรนี้แสดงวิธีหนึ่งที่ใช้ในการสร้างสัญญาณการเขียนพอร์ทเท่านั้น



รูปที่ 23 แสดงไดอะแกรมเวลาของการเกิดสัญญาณ PORT WRITE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 24 แสดงวงจรการสร้างสัญญาณ PORT WRITE

จุดประสงค์ของวงจรในรูปที่ 24 คือ ต้องการสร้าง STROBE ซึ่งแอคทีฟที่ลอจิก "0" เมื่อ CPU ต้องการส่งข้อมูลไปยังพอร์ทเอาท์พุทที่ถูกกำหนด ในรูปนี้ระบบบัสแอดเดรส A0-A7 เป็นอินพุททั้ง 8 เส้นของ NAND HAND GATE ซึ่งเป็นวงจรลักษณะเดียวกับวงจรในรูปที่ 20

เมื่อบัสแอดเดรสทั้ง 8 เส้น (A0-A7) มีลอจิกเป็น "1" (รหัสเลือกพอร์ทคือ 0FFH) เอาท์พุทของ HAND GATE จะมีลอจิกเป็น "0" PORT SELECT LINE นี้จะต่อเข้ากับอินพุทหนึ่งของ 74LS32 OR GATE สำหรับอีกอินพุทหนึ่งของ OR GATE นั้นมาจากสัญญาณ $\overline{IOW}$ STROBE (แสดงในรูป 7) เมื่อ $\overline{IOW}$ STROBE มีลอจิกเป็น "0" จะบ่งบอกให้รู้ว่า CPU กำลังกระทำการส่งข้อมูลไปยังพอร์ทเอาท์พุท

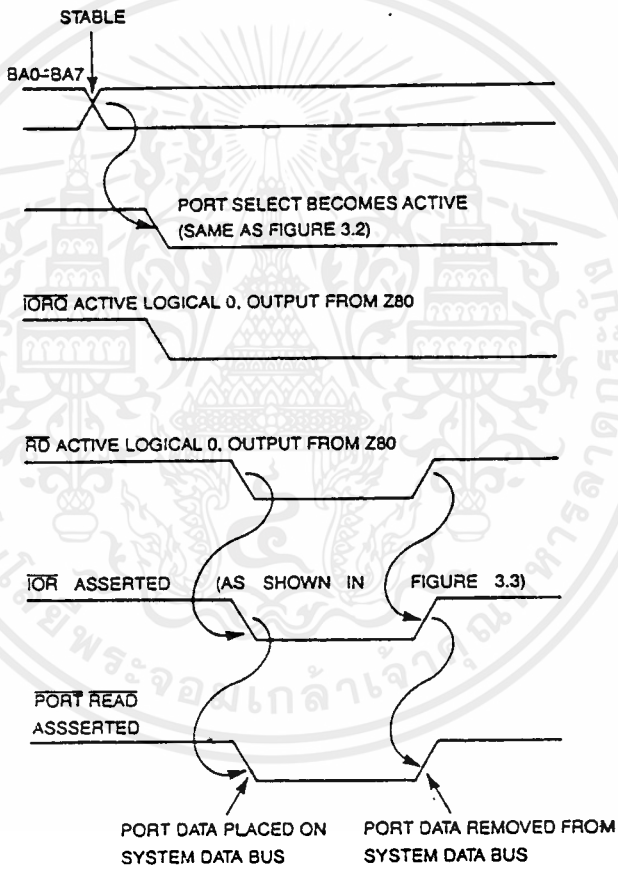
ดังนั้นถ้าสัญญาณ $\overline{IOW}$ และ PORT SELECT LINE มีลอจิกเป็น "0" พร้อมกัน CPU ก็จะส่งข้อมูลให้พอร์ทเอาท์พุทที่ถูกเลือก การใช้สัญญาณ $\overline{IOW}$ และ PORT SELECT LINE มาผ่าน OR GATE ก็เพียงพอที่จะสร้างสัญญาณ STROBE กระตุ้นพอร์ทเอาท์พุทที่ถูกเลือกให้ทำการรับเอาข้อมูลไป สัญญาณนี้เราจะเรียกว่า "PORT WRITE STROBE"

วิธีการที่กล่าวถึงนี้เป็นวิธีการหนึ่งในการสร้างสัญญาณ STROBE ให้พอร์ทเอาท์พุทรับเอาข้อมูลไปเท่านั้น ในระบบไมโครโปรเซสเซอร์ทั่วไปก็มักจะใช้วิธีดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างสัญญาณการอ่านพอร์ท

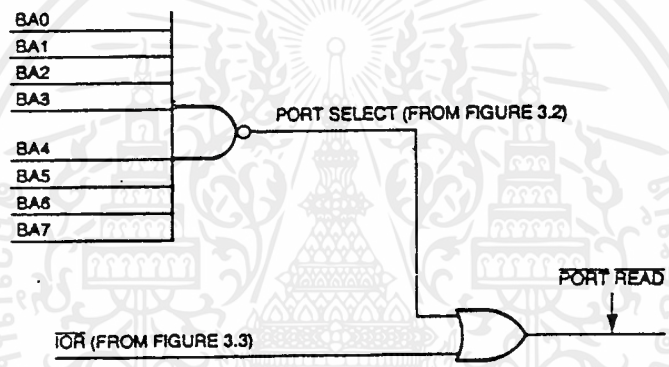
ที่จะกล่าวต่อไปนี้เราจะศึกษาขบวนการอ่านข้อมูลจากพอร์ทอินพุทของ Z-80 ไดอะแกรมเวลาในรูปที่ 25 แสดงลำดับของสัญญาณที่ปรากฏขึ้นระหว่างขบวนการอ่านข้อมูลจากพอร์ทอินพุท ในไดอะแกรมนี้สัญญาณ $\overline{IOR}$ จะมีลักษณะคล้ายกับสัญญาณ $\overline{IOW}$ ในขณะที่ทำการส่งข้อมูลไปยังพอร์ทเอาท์พุท



รูปที่ 25 แสดงไดอะแกรมเวลาของสัญญาณที่เกิดขึ้นในวงจรรูป 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

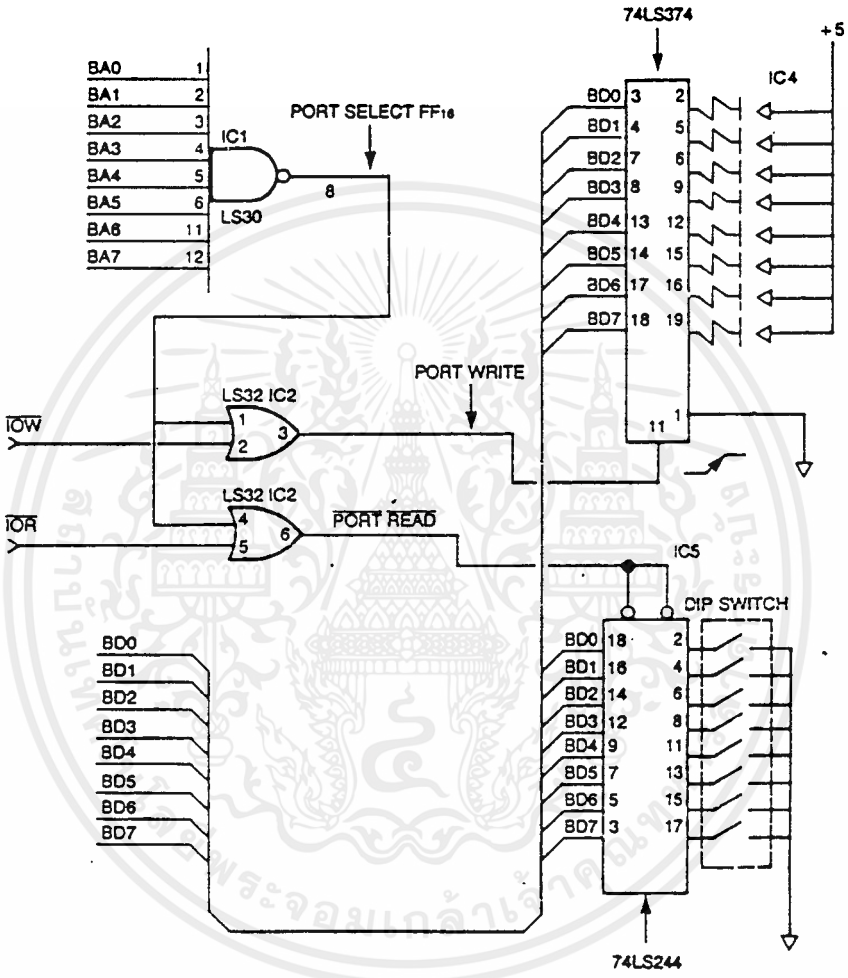
เรามาพิจารณาผลของสัญญาณ $\overline{RD}$ ซึ่งแสดงในรูปที่ 25 สัญญาณ $\overline{RD}$ นี้จะนำไปสร้างสัญญาณ $\overline{IOR}$ ให้แก่ฮาร์ดแวร์ส่วนพอร์ทอินพุท สัญญาณนี้จะทำให้เกิดการเคลื่อนย้ายข้อมูลจากพอร์ทอินพุทเข้าสู่ Z-80 เมื่อ $\overline{RD}$ เปลี่ยนลอจิกมาเป็น "0" ข้อมูลจากพอร์ทอินพุทจะปรากฏบนระบบบัสข้อมูล และจะถูก STROBE เข้าไปในรีจิสเตอร์ภายใน (INTERNAL REGISTER) ของ Z-80 สัญญาณ $\overline{RD}$ ก็จะกลับมาเป็นลอจิก "1" ใหม่ ขบวนการเคลื่อนย้ายข้อมูลก็จะสิ้นสุดลง



รูปที่ 26 แสดงวงจรการสร้างสัญญาณ PORT READ

รูปที่ 26 แสดงวิธีหนึ่งในหลาย ๆ วิธีที่จะสร้างสัญญาณการอ่านพอร์ท ซึ่งมาจากเงื่อนไขทางลอจิกในรูป 25 เราจะเห็นว่ารูป 26 นั้นมีวงจรคล้าย ๆ กับวงจรในรูป 24 มาก แตกต่างกันที่สัญญาณ $\overline{IOW}$ และ $\overline{IOR}$ เท่านั้น PORT READ STROBE จะแอคทีฟ (มีลอจิกเป็น "0") เมื่อ Z-80 ทำการรับข้อมูลจากพอร์ทที่เลือก ในที่นี้คือพอร์ท 0FFH ในรูป 27 แสดงวงจรที่สมบูรณ์ในการสร้างพอร์ท I/O ขนาด 8 บิททั่ว ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 27 แสดงวงจรสมบูรณ์ของพอร์ทอินพุท/เอาท์พุทขนาด 8 บิท ซึ่งมีตำแหน่ง I/O อยู่ที่ 0FFH

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ที่เกิดขึ้นขณะกระบวนการส่งข้อมูลไปยังพอร์ทเอาท์พุท

1. บัสแอดเดรส A0-A7 จะส่งตำแหน่งเอาท์พุทภายใต้การควบคุมของ Z-80 ซึ่งขณะเดียวกันบัสแอดเดรส A0-A7 จะถูกถอดรหัสโดยฮาร์ดแวร์ส่วน PORT SELECT LINE ในรูป 27 เอาท์พุทขา 8 ของ IC(74LS30) จะมีลอจิกเป็น "0" เมื่อ A0-A7 มีค่าตรงกับพอร์ทแอดเดรสที่กำหนด (ในที่นี้คือ 0FFH)

2. Z-80 จะส่งข้อมูลที่ต้องการส่งออก เข้าสู่บัสข้อมูล D0-D7 ในรูป 27 ข้อมูลจะรออยู่ที่ขาอินพุท D0-D7 ของ IC4 (74LS374 : OCTAL LATCH)

3. ขา $\overline{IORQ}$ ของ Z-80 จะถูกเซ็ทให้มีลอจิกเป็นศูนย์ ซึ่งจะเป็นการบ่งบอกว่า Z-80 ต้องการที่จะทำการติดต่อกับ I/O สัญญาณนี้จะยังคงค้างไว้จนกว่าจะสิ้นสุดขบวนการ

4. ขา $\overline{WR}$ ของ Z-80 จะถูกเซ็ทให้มีลอจิกเป็น "0" ซึ่งจะทำให้สัญญาณ $\overline{IOW}$ มีลอจิกเป็น "0" ก็จะทำให้ PORT WRITE STROBE เปลี่ยนลอจิกเป็น "0" ขา $\overline{WR}$ ของ Z-80 จะกลับมามีลอจิกเป็น "1" อีกครั้งหนึ่ง ขณะนี้ข้อมูลบนบัสข้อมูล D0-D7 จะถูกเขียนลงบน 74LS374 อันเป็นการสิ้นสุดการส่งข้อมูลไปยังพอร์ทเอาท์พุท

ในรูป 27 เป็นการนำข้อมูลของพอร์ทเอาท์พุทมาเปิด-ปิด LED บิทที่มีลอจิกเป็น "0" LED ตัวนั้นจะติด บิทที่มีลอจิกเป็น "1" LED จะดับ

ขบวนการอ่านข้อมูลจากพอร์ทอินพุท

ต่อไปเราจะพิจารณาขบวนการอ่านข้อมูลของ Z-80 จากพอร์ทอินพุท ดังแสดงในรูป 27 ขบวนการส่วนใหญ่จะมีลักษณะคล้าย ๆ กับการส่งข้อมูลไปยังพอร์ทเอาท์พุท ยกเว้นสัญญาณ $\overline{RD}$ ซึ่งจะถูกใช้แทนสัญญาณ $\overline{WR}$

1. สัญญาณ A0-A7 จะปรากฏขึ้นบนระบบบัสแอดเดรส IC 1 จะทำการถอดระหัส ขา 8 เอาท์พุทของ IC 1 จะมีลอจิกเป็น "0" ข้อมูลต่าง ๆ ของพอร์ทอินพุทจะยังไม่ปรากฏบนบัสข้อมูล จะรอจนกว่าสัญญาณ $\overline{IORQ}$ และสัญญาณ $\overline{RD}$ เกิดขึ้นพร้อมกันก่อน

2. $\overline{IORQ}$ จะถูกเซ็ทให้มีลอจิกเป็น "0" โดย Z-80 การกระทำนี้เป็นการบ่งบอกแก่ระบบว่า Z-80 ต้องการติดต่อกับ I/O ไม่ใช่หน่วยความจำ

3. สัญญาณ $\overline{RD}$ จะถูกเซ็ทให้มีลอจิกเป็น "0" โดย Z-80 ซึ่งจะทำให้เกิดสัญญาณ $\overline{IOR}$ ขึ้นในระบบ

4. เมื่อสัญญาณ $\overline{IOR}$ ปรากฏขึ้น ไมโครโปรเซสเซอร์จะทำการรับข้อมูลจากพอร์ทอินพุท เพราะว่าขา $\overline{RD}$ จาก Z-80 มีลอจิกเป็น "0" ขา ENABLE (ขา 1,19) ของ IC 5 (74LS244) จะถูกเซ็ทให้มีลอจิกเป็น "0" ลอจิก "0" บนขา ENABLE ของ 74LS244 จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ขาข้อมูลออก (DATA OUTPUT) มีข้อมูลส่งออกไปให้ระบบบัสข้อมูล เมื่อเหตุการณ์ทั้งหมดนี้เกิดขึ้น ข้อมูลที่ถูกเซ็ทโดยสวิทช์หรือที่สัญญาณที่ส่งให้กับ 74LS244 ซึ่งเป็นบัฟเฟอร์ ก็จะส่งข้อมูลให้กับบัสข้อมูล ขณะที่ 74LS244 ถูก ENABLE นั้น Z-80 จะรับเอาข้อมูลที่อยู่บนระบบบัสข้อมูลเข้าไปในรีจิสเตอร์ภายในของ Z-80

5. ท้ายสุดขา $\overline{RD}$ จะถูกเซ็ทให้มีลอจิกเป็น "1" อีกครั้งหนึ่ง ภายใต้การควบคุมของ Z-80 เมื่อ $\overline{RD}$ มีลอจิกเป็น "1" ข้อมูลจากพอร์ทอินพุทก็จะถูกเคลื่อนย้ายจากระบบบัสข้อมูลเข้าสู่ CPU ขบวนการอ่านข้อมูลจากพอร์ทอินพุทก็จะสิ้นสุดลง

สรุปขบวนการส่งข้อมูลออกไปยังพอร์ทเอาท์พุท

1. A0-A7 ถูกเซ็ทให้ตรงกับตำแหน่งของพอร์ทที่จะติดต่อด้วย
2. D0-D7 ถูกเซ็ทให้ตรงกับข้อมูลที่ต้องการส่งออกไป
3. $\overline{IORQ}$ จะมีลอจิกเป็น "0"
4. $\overline{WR}$ มีลอจิกเป็น "0"
5. $\overline{WR}$ จะถูกเซ็ทให้กลับมามีลอจิกเป็น "1"

สรุปขบวนการอ่านข้อมูลจากพอร์ทอินพุท

1. A0-A7 ถูกเซ็ทให้ตรงกับตำแหน่งพอร์ทที่จะอ่านข้อมูล
2. $\overline{IORQ}$ ถูกเซ็ทให้มีลอจิกเป็น "0" ภายใต้การควบคุมของ Z-80
3. $\overline{RD}$ ถูกเซ็ทให้มีลอจิกเป็น "0"
4. $\overline{RD}$ ถูกเซ็ทให้กลับมามีลอจิกเป็น "1"

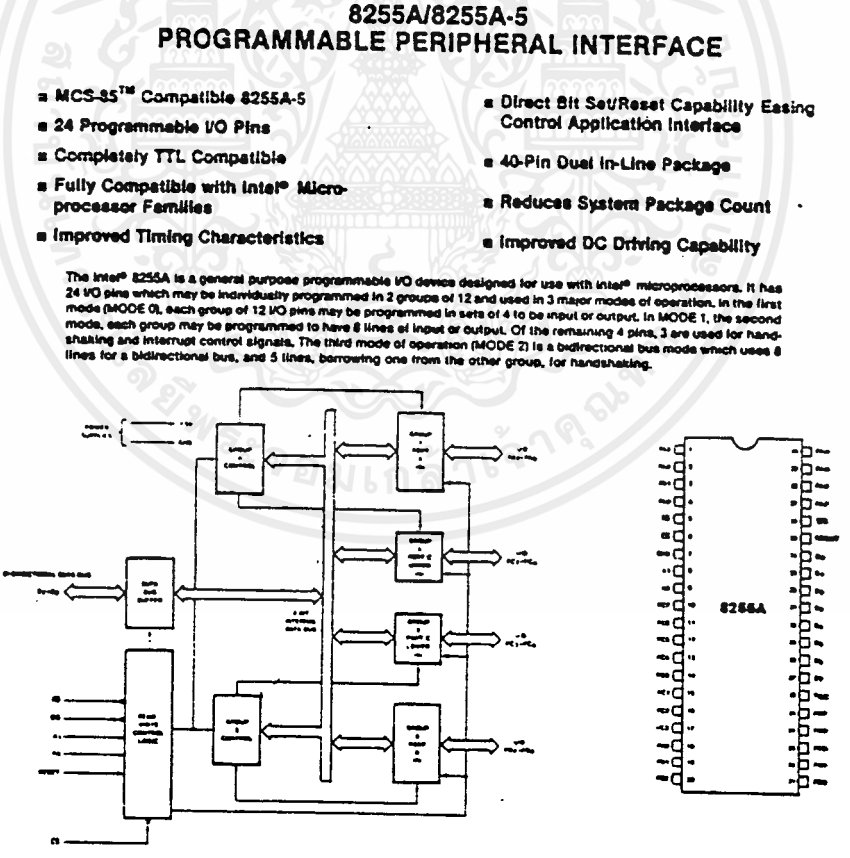
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การใช้ 8255 PIA กับ Z-80

รายละเอียดเกี่ยวกับ 8255

8255 เป็นอุปกรณ์ LSI (LARGE SCALE INTEGRATED CIRCUIT) บรรจุอยู่ใน PACKAGE 40 ขา แบบ DIP (DUAL-IN-LINE PACKAGE) เริ่มผลิตโดยบริษัท INTEL COORPERATION ผู้ผลิตไมโครโปรเซสเซอร์เบอร์ 8080 โดยเฉพาะแต่ในภายหลังได้มีการนำ 8255 ไปประยุกต์ใช้งานร่วมกับไมโครโปรเซสเซอร์เบอร์อื่น ๆ รวมทั้ง Z-80 ด้วย หากผู้อ่านเข้าใจการใช้งาน 8255 กับ Z-80 ที่จะกล่าวถึงนี้แล้วก็จะนำไปประยุกต์ใช้งานในลักษณะอื่น ๆ ได้ไม่ยากนัก

รูป 28 นี้แสดงบล็อกไดอะแกรมของ 8255 ซึ่งหน้าที่ของแต่ละบล็อกมีดังต่อไปนี้คือ



8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40-Pin Dual In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

รูปที่ 28 แสดงบล็อกไดอะแกรมและการวางตำแหน่งขาของ 8255

เอกสารนี้เป็นเอกสาร[illegible]ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกกลุ่มแรกที่เราจะพูดถึงนี้ ได้แก่ บล็อกจำนวน 4 บล็อก ที่อยู่ทางด้านขวาของรูป ซึ่งจะเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกอื่นๆ โดยมีสาย PA0-PA7, PB0-PB7 และ PC0-PC7 เป็นทางผ่านของข้อมูลระหว่างอุปกรณ์ภายนอกกับ 8255 สายสัญญาณเหล่านี้จะถูกแบ่งออกเป็น 3 I/O พอร์ท ได้แก่ พอร์ท A (PA), พอร์ท B (PB) และพอร์ท C (PC) พอร์ทเหล่านี้แต่ละพอร์ทสามารถเป็นได้ทั้งพอร์ทอินพุทและเอาท์พุท และแต่ละบล็อกจะมีสายสัญญาณเชื่อมเข้ากับบัสข้อมูลภายในของ 8255

บล็อกกลุ่มถัดมา ได้แก่ GROUP A CONTROL และ GROUP B CONTROL ซึ่งจะเป็นตัวกำหนดลักษณะการทำงานของทั้ง 3 I/O พอร์ท (8255 มีลักษณะการทำงานที่แตกต่างกันอยู่ 3 โหมด สามารถกำหนดได้โดยการโปรแกรมส่ง CONTROL WORD ให้กับ 8255 ซึ่งจะกล่าวถึงในภายหลัง) จากรูปที่ 28 จะเห็นว่า พอร์ท C นี้จะประกอบด้วยพอร์ทขนาด 4 บิท 2 พอร์ท กลุ่มหนึ่งจะถูกควบคุมโดย GROUP A CONTROL และอีกกลุ่มหนึ่งจะถูกควบคุมโดย GROUP B CONTROL สำหรับเหตุผลนั้นจะกล่าวถึงในภายหลัง

บล็อกกลุ่มสุดท้ายที่จะกล่าวถึง ได้แก่ DATA BUS BUFFER และ READ/WRITE CONTROL LOGIC ซึ่งบล็อกเหล่านี้จะเป็นส่วนที่ติดต่อกับ CPU, DATA BUS BUFFER นี้จะเป็นบัฟเฟอร์ให้กับบัสข้อมูลของ CPU ส่วน READ/WRITE CONTROL LOGIC จะเป็นส่วนที่ควบคุมให้ข้อมูลเข้าหรือออกจากรีจิสเตอร์ภายใน ตัวที่ถูกต้อง และในเวลาที่เหมาะสม

รายละเอียดการจัดเรียงขาของ 8255

ในส่วนนี้เราจะพิจารณาหน้าที่ของขาแต่ละขาของ 8255 ซึ่งข้อมูลเหล่านี้จะมีประโยชน์ในการเชื่อมต่อเข้ากับระบบบัสของ CPU สำหรับการจัดขาแสดงไว้ในรูปที่ 28 รายละเอียดของแต่ละขามีดังนี้คือ

D0-D7 : เป็นสายข้อมูลอินพุท/เอาท์พุทแบบสองทิศทาง (BI-DIRECTIONAL BUS) จะเป็นทางผ่านของข้อมูลระหว่างพอร์ทต่างๆของ 8255 กับบัสข้อมูลของ Z-80

CS (CHIP SELECT INPUT) : เมื่อขานี้มีสถานะลอจิกเป็น "0" CPU จะสามารถที่จะอ่านหรือเขียนข้อมูลกับ 8255 ได้

RD (READ INPUT) : เมื่อขานี้มีสถานะลอจิกเป็น "0" และสัญญาณ CS มีลอจิกเป็น "0" ข้อมูลจาก 8255 จะปรากฏสู่ระบบบัสข้อมูล CPU ก็จะสามารถอ่านข้อมูลออกไปได้ (ในการตั้งชื่อของขาสัญญาณนี้จะถือเอา CPU เป็นหลัก)

WR (WRITE INPUT) : เมื่อขานี้มีสถานะลอจิกเป็น "0" และขาสัญญาณ CS มีลอจิกเป็น "0" ข้อมูลจากระบบบัสข้อมูลจะถูกเขียนเข้าไปยัง 8255 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A0-A1 (ADDRESS INPUT) : จะเป็นตัวกำหนดการเลือกใช้รีจิสเตอร์ภายในของ 8255 ซึ่งจะกล่าวรายละเอียดในภายหลัง

RESET : เมื่อขานี้มีสถานะเป็น "1" 8255 จะอยู่ในสภาวะรีเซ็ททุกๆพอร์ทของ 8255 จะถูกเซ็ทให้อยู่ในโหมดอินพุท

PA0-PA7,PB0-PB7 : ขาสัญญาณเหล่านี้จะถูกใช้เป็นพอร์ท I/O ขนาด 8 บิท ใช้ต่อเข้ากับอุปกรณ์ภายนอกอื่น ๆ

PC0-PC7 : ขาสัญญาณนี้ถูกใช้เป็นพอร์ท I/O ขนาด 8 บิท เช่นเดียวกับ PA0-PA7 และ PB0-PB7 แต่กลุ่มของขาสัญญาณเหล่านี้สามารถแบ่งออกเป็น 2 กลุ่ม โดยแต่ละกลุ่มมีขนาด 4 บิทได้ กลุ่มแรกจะใช้ควบคุม PB0-PB7 และกลุ่มที่ 2 ใช้ควบคุม PA0-PA7 (ซึ่งจะกล่าวถึงรายละเอียดในภายหลัง)

การต่อ 8255 เข้ากับ Z-80

ในการต่อ 8255 เข้ากับระบบของ Z-80 นั้น สัญญาณต่าง ๆ ที่เกิดขึ้นจะเหมือนกับขบวนการติดต่อกับ I/O ดังที่เคยกล่าวมาแล้ว โดยจะต้องเอาสัญญาณ A0-A7 จาก Z-80 มาถอดรหัสเพื่อสร้างสัญญาณเลือกพอร์ท แต่เนื่องจาก 8255 มีขา ADDRESS INPUT อยู่แล้ว 2 ขา (A0, A1) ซึ่งโดยปกติแล้วขา A0,A1 นี้จะต่อเข้าโดยตรงกับ A0,A1 จากบัสแอดเดรส นั่นคือ 8255 หนึ่งตัวจะใช้ค่าพอร์ทแอดเดรสถึง 4 ค่า (2^2) ส่วนสัญญาณอีก 6 เส้น (A2-A7) จะนำไปถอดรหัสเพื่อทำสัญญาณเลือกชิพ (CHIP SELECT) ให้แก่ 8255

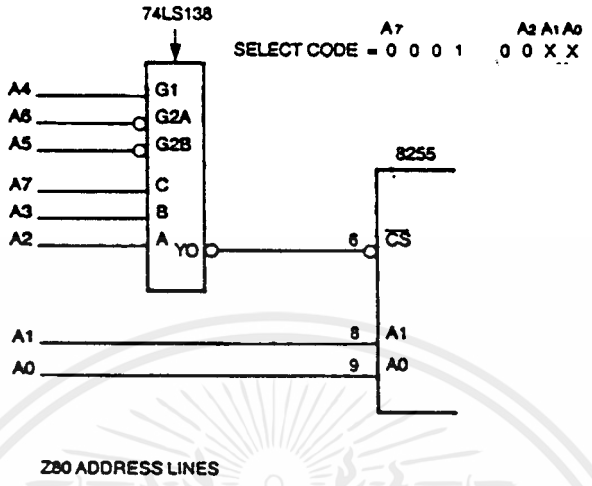
ในที่นี้เราจะสมมติให้ 8255 มีพอร์ทแอดเดรสอยู่ที่ 10H, 11H, 12H และ 13H ซึ่งวิธีหนึ่งที่สามารถจะถอดรหัสพอร์ทเหล่านี้ได้ แสดงไว้ในรูปที่ 29

จากรูปที่ 29 นี้ จะเห็นว่าขาอินพุท CS จะแอคทีฟต่อเมื่อ A7-A2 มีเท่ากับ 000100XXB (2 บิทล่างจะใช้เพื่อเลือกใช้รีจิสเตอร์ภายใน 4 ตัว)

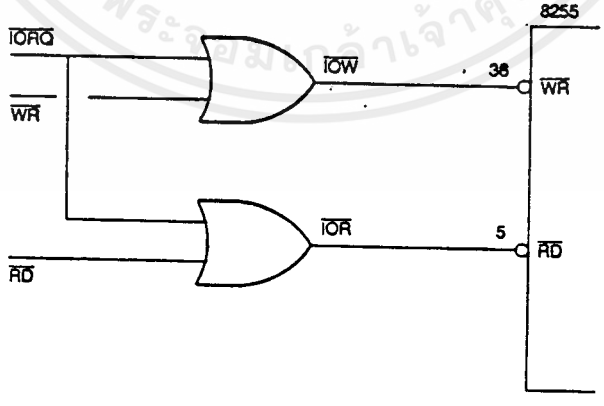
ขั้นต่อไปที่จะต้องทำคือ การต่อขา RD และ WR ของ 8255 เข้ากับสัญญาณควบคุม IOR และ IOW ของระบบ การที่เราไม่ต่อขา RD และ WR เข้าโดยตรง เพราะในตัวอย่าง วิธีการถอดรหัสนี้ อาจจะเกิดกรณีที่ A7-A0 มีค่าตรงกับ 000100XXB ซึ่งจะทำให้เกิดการอ่านหรือเขียนข้อมูลกับ 8255 โดยไม่ต้องการได้ ในการแก้ปัญหานี้จึงใช้สัญญาณ IORQ จาก CPU มาทำเป็นสัญญาณ IOR และ IOW เพื่อแยกว่าเป็นการติดต่อกับ I/O ไม่ใช่หน่วยความจำ ดังแสดงในรูปที่ 30

ในการต่อขา RESET ของ 8255 ซึ่งจะแอคทีฟที่ลอจิก "1" เข้ากับขา RESET ของ Z-80 ซึ่งแอคทีฟที่ลอจิก "0" นั้นจะต้องใช้ INVERTER คั่นกลางเสียก่อน

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



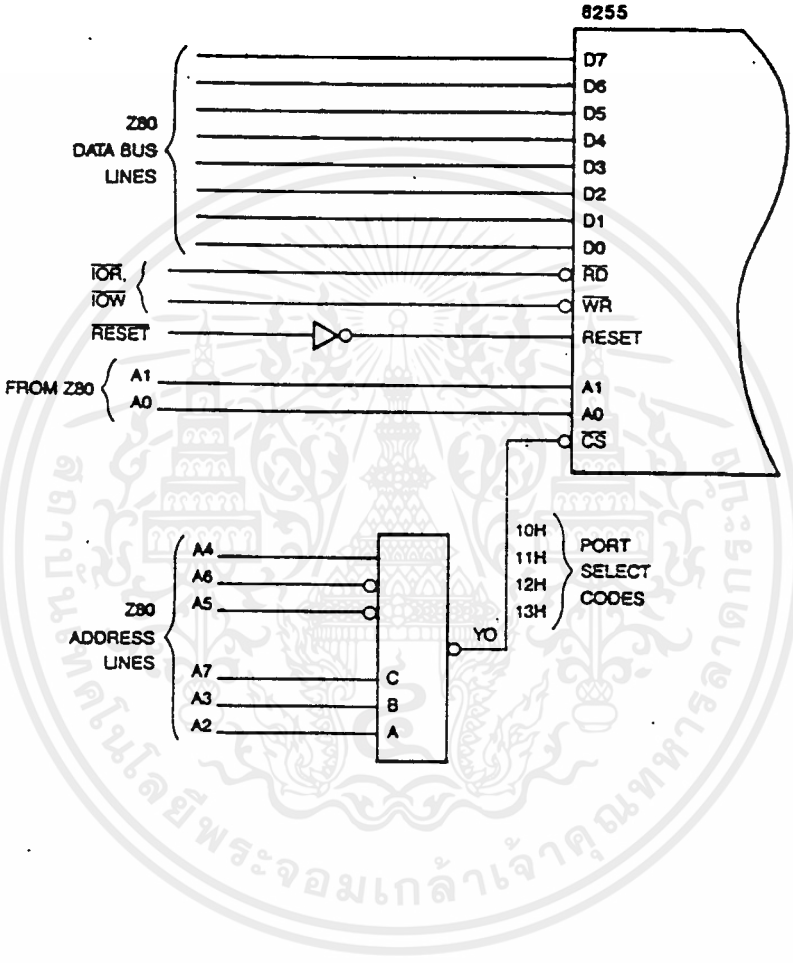
รูปที่ 29 แสดงวงจรการถอดรหัสการเลือกพอร์ทที่ติดต่อกับ 8255



รูปที่ 30 แสดงวิธีการต่อขา WR และ RD เข้ากับระบบของ Z-80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการต่อสายข้อมูล D0-D7 ของ 8255 เข้ากับระบบบัสข้อมูลของระบบเราจะสมมติว่าไม่มีการโหลดบนบัสข้อมูล ดังนั้นเราจึงสามารถต่อสายสัญญาณเหล่านี้เข้าโดยตรงกับระบบบัสข้อมูล ดังแสดงวงจรสมบูรณ์ของการเชื่อมต่อ 8255 เข้ากับระบบของ Z-80 ในรูปที่ 31



รูปที่ 31 แสดงผังวงจรสมบูรณ์ของการเชื่อมต่อ 8255 เข้ากับระบบของ Z-80

8255 READ และ WRITE REGISTER

ขณะนี้เราได้ทำการต่อ 8255 เข้ากับระบบของ Z-80 แล้ว ต่อไปจะศึกษาการโปรแกรมใช้งาน 8255 เพื่อที่จะให้ทำงานตามที่เราต้องการได้ จะเริ่มต้นพิจารณาที่รีจิสเตอร์ภายใน 4 ตัวของ 8255 สำหรับในตัวอย่างการถอดรหัสนี้ ตำแหน่งของรีจิสเตอร์จะอยู่ที่แอดเดรส 10H, 11H, 12H และ 13H ซึ่งรายละเอียดของรีจิสเตอร์เหล่านี้มีดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVICE PIN				REGISTER NAME
RD	WR	A1	A0	
1	0	0	0	WRITE PORT A DATA
0	1	0	0	READ PORT A DATA
1	0	0	1	WRITE PORT B DATA
0	1	0	1	READ PORT B DATA
1	0	1	0	WRITE PORT C DATA
0	1	1	0	READ PORT C DATA
1	0	1	1	WRITE CONTROL DATA
0	1	1	1	ILLEGAL READ REGISTER

หน้าที่ของรีจิสเตอร์หมายเลข 0-2 จะถูกกำหนดลักษณะการทำงานจากรีจิสเตอร์หมายเลข 3 (รีจิสเตอร์ควบคุม) รูปที่ 32 จะแสดงรายละเอียดของแต่ละบิทของรีจิสเตอร์ควบคุมนี้ ต่อไปเราจะกล่าวถึงลักษณะการทำงานของ 8255 ทั้ง 3 โหมด และการโปรแกรมให้อยู่ในโหมดต่าง ๆ ได้ดังต่อไปนี้คือ

โหมด 0 : BASIC REGISTER I/O

ในการเซ็ท 8255 ให้อยู่ในโหมด 0 นั้น เราจะต้องส่งคำสั่งควบคุม (CONTROL WORD) ให้แก่รีจิสเตอร์ควบคุมก่อน คำสั่งควบคุมนี้จะกำหนดลักษณะการทำงานให้แก่แต่ละพอร์ทของ 8255 ตัวอย่างหนึ่งของคำสั่งควบคุมที่จะสั่งให้ 8255 ทำงานอยู่ในโหมด 0 นี้ได้แก่

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

จากรูปที่ 32. จะเห็นว่า

บิท D7 เป็นตัวกำหนดว่าเป็นคำสั่งควบคุม (CONTROL WORD)

บิท D6 และ D5 กำหนดโหมดการทำงานของพอร์ท A D6,D5 มีค่าเป็น "0" แสดงว่าอยู่ในโหมด 0

บิท D4 = "0" กำหนดให้พอร์ท A เป็นพอร์ทเอาท์พุท

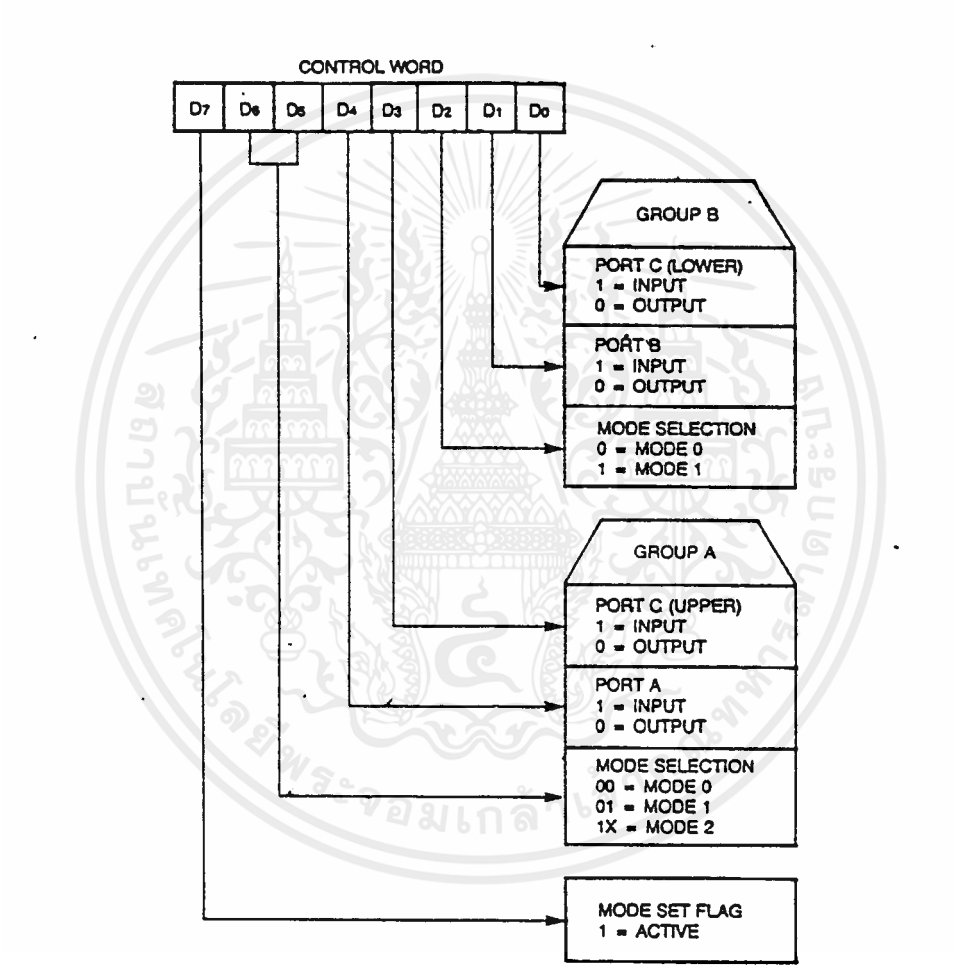
บิท D3 = "0" เซ็ทพอร์ท C 4 บิทบนเป็นพอร์ทเอาท์พุท

เอกสารนี้มีลิขสิทธิ์ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิท D2 = "0" เซ็ทโหมดของพอร์ท B ให้พอร์ท B อยู่ในโหมด 0

บิท D1 = "0" เซ็ทพอร์ท B เป็นพอร์ทเอาท์พุท

บิท D0 = "0" เซ็ทพอร์ท C ให้ 4 บิทล่างเป็นพอร์ทเอาท์พุท



รูปที่ 32 แสดงรายละเอียดแต่ละบิทของรีจิสเตอร์ควบคุมของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งควบคุมนี้จะกำหนดให้พอร์ททั้ง 3 ของ 8255 ทำงานอยู่ในโหมด 0 และเป็นพอร์ทเอาท์พุท ซึ่งจะได้สายสัญญาณซึ่งสามารถติดต่อกับอุปกรณ์ภายนอกได้ถึง 24 สาย คำสั่งของ Z80 ที่จะเซ็ทให้ 8255 อยู่ในลักษณะดังกล่าว ได้แก่

```
LD A, 80H          : เซ็ทคำสั่งควบคุม
OUT (13H), A       : ส่งคำสั่งควบคุมให้ 8255
```

เมื่อ Z80 ทำการ EXECUTE คำสั่งข้างต้นแล้ว 8255 จะถูกเซ็ทให้พอร์ททุกพอร์ท เป็นพอร์ทเอาท์พุท และอยู่ในโหมด 0 เราจะสามารถส่งข้อมูลออกไปยังพอร์ทต่าง ๆ ได้ด้วยคำสั่ง OUT ของ Z80 ตัวอย่างเช่น เราต้องการส่ง 23H ไปยังพอร์ท A, 41H ไปยังพอร์ท B และ 73H ไปยังพอร์ท C เราจะต้องให้ Z80 ทำตามโปรแกรมลักษณะดังนี้

```
LD A, 32H          : เซ็ทข้อมูลให้พอร์ท A
OUT (10H), A       : ส่งข้อมูลให้พอร์ท A
LD A, 41H          : เซ็ทข้อมูลให้พอร์ท B
OUT (11H), A       : ส่งข้อมูลให้พอร์ท B
LD A, 73H          : เซ็ทข้อมูลให้พอร์ท C
OUT (12H), A       : ส่งข้อมูลให้พอร์ท C
```

หลังจากที่คำสั่งเหล่านี้ถูก EXECUTE แล้วพอร์ท A, B และ C ของ 8255 จะมีข้อมูลต่าง ๆ ที่ส่งไปให้ปรากฏอยู่

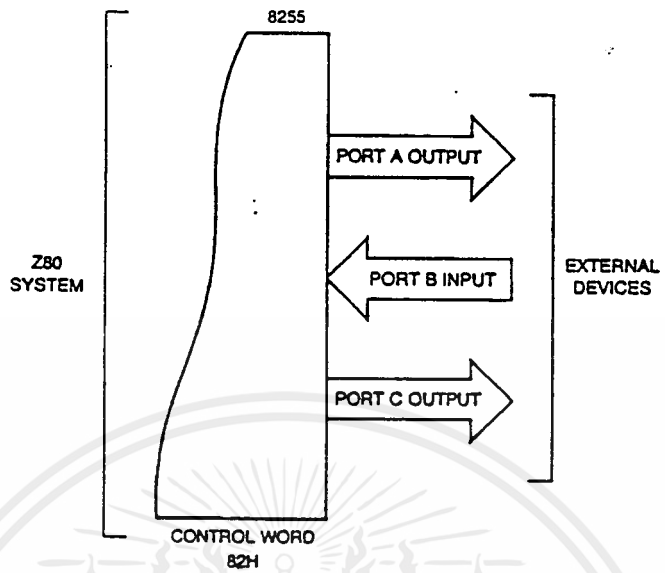
ในการทำงานในโหมด 0 ของ 8255 นี้อาจจะสั่งให้พอร์ทของ 8255 เป็นอินพุทหรือเอาท์พุทก็ได้ อย่างเช่น ให้พอร์ท A และพอร์ท C เป็นพอร์ทเอาท์พุท และพอร์ท B เป็นพอร์ทอินพุท เราจะต้องส่งคำสั่งควบคุมให้แก่รีจิสเตอร์ควบคุมในลักษณะดังนี้คือ

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0

หลังจากที่ส่งคำสั่งควบคุมให้แก่รีจิสเตอร์ควบคุมแล้ว 8255 จุถูกเซ็ทให้มีลักษณะการทำงานดังรูป 33 เราจะใช้คำสั่ง IN อ่านข้อมูลมาจากพอร์ท B ได้

```
IN A, (11H)        : อ่านข้อมูลจากพอร์ท B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

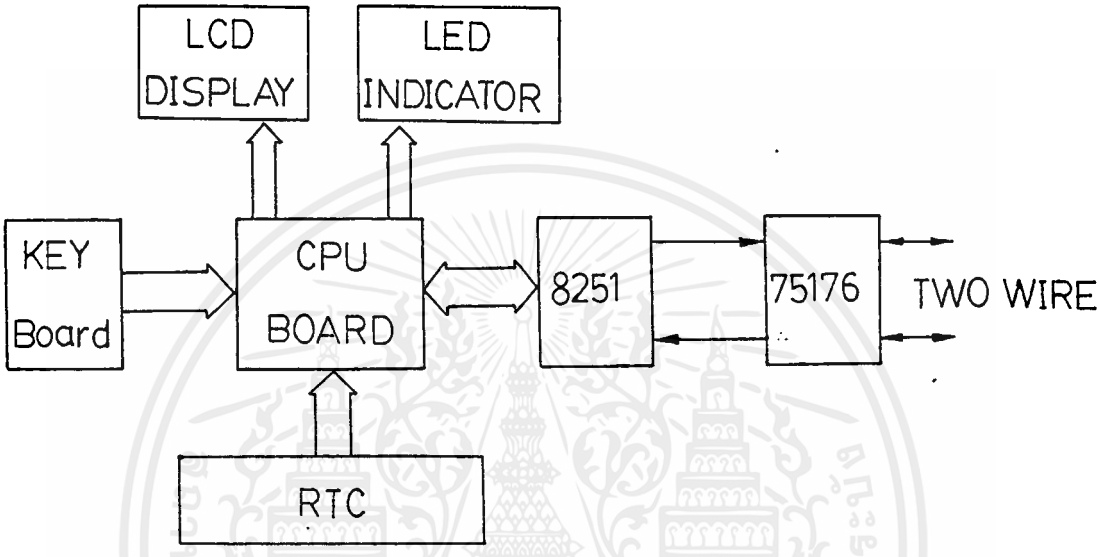


รูปที่ 33 บล็อกไดอะแกรมแสดงลักษณะการทำงานของ 8255 ในโหมด 0 หลังจากส่งคำสั่งควบคุมให้ 8255 แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบวงจรและการสร้าง

ด้านส่ง

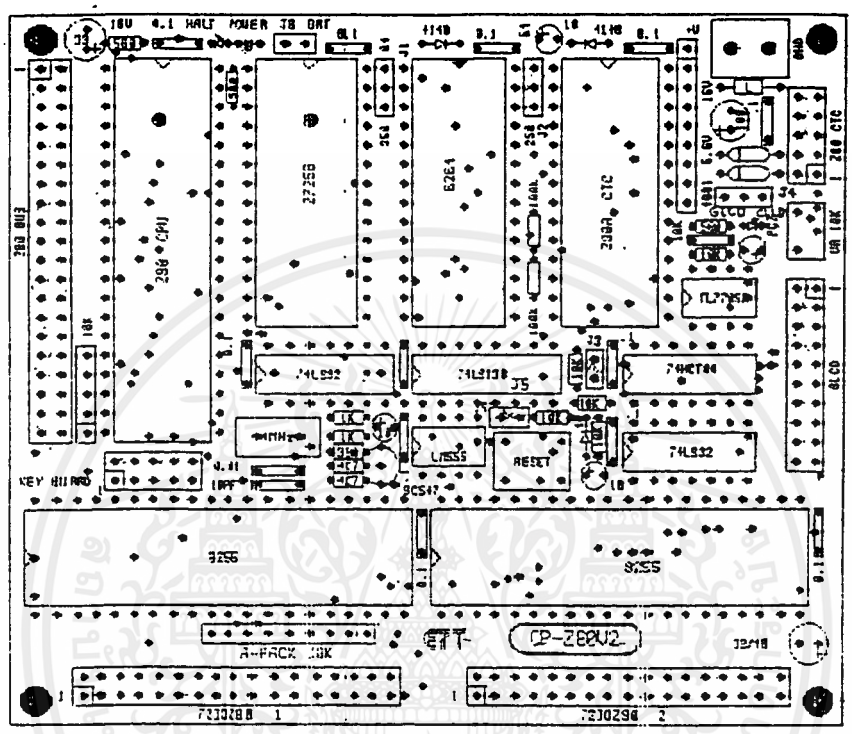


รูปที่ 1 แสดงบล็อกไดอะแกรมทางด้านส่ง

จากบล็อกไดอะแกรมในรูปที่ 1 CPU BOARD ทำหน้าที่เป็นตัวจัดการการทำงานทั้งหมดของเครื่อง โดยทำการรับ INPUT จาก KEYBOARD เพื่อจะให้ทราบว่า ต้องการให้ CPU ON-OFF LOAD ตัวใด หรือให้ตั้งเวลาอย่างไร ซึ่ง FUNCTION การทำงานต่างๆที่ป้อนโดย KEYBOARD จะแสดงผลโดย LCD DISPLAY เมื่อ CPU รับข้อมูลเข้ามา ก็จะนำมาประมวลผลเปรียบเทียบกับ PROGRAM ที่ตั้งไว้ จากนั้นจะส่งข้อมูลให้ 8251 ทำหน้าที่เปลี่ยนข้อมูลจากขนานเป็นอนุกรม และ 75176 จะเปลี่ยนข้อมูลอนุกรมที่ได้ให้เป็นมาตราฐาน EIA RS-485 เพื่อส่งข้อมูลไปให้ตัวรับได้ในระยะทางไกลๆ เมื่อตัวรับได้รับข้อมูลแล้วจะส่งข้อมูลนั้นกลับมาที่ตัวส่งอีกทีเพื่อเปรียบเทียบว่าข้อมูลที่ส่งออกไปกับข้อมูลที่ตัวรับรับตรงกันหรือไม่ ถ้าข้อมูลต่างกัน ก็จะแสดงผลการผิดพลาดโดยจอ LCD ถ้าข้อมูลตรงกัน CPU ก็จะ OUT PORT ไปที่ 8255 เพื่อแสดงผลสถานะของ LOAD โดย LED INDICATOR ว่ามี LOAD ตัวใด ON หรือ OFF อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลของ BOARD CP-Z80V2



CPU ใช้ CPU ยอดนิยม Z80 CPU เป็น CPU ประจำบอร์ดโดยเราใช้เบอร์ Z84C00-6 ของ ZILOG ซึ่งจะเทียบเท่า Z80 B แบบ CMOS เท่าใช้กินกำลังงานต่ำสามารถต่อใช้กับความถี่ได้สูงสุด 6 MHz แต่ในบอร์ดนี้ เราจะใช้ความถี่ 4 MHz เพื่อไม่จำเป็นต้องใช้ ROM หรือ EPROM ที่มี ACSSE TIME น้อยๆมากนักก็ได้ แต่ถ้าผู้ใช้จะเปลี่ยนใช้บอร์ดนี้ RUN 6 MHz ก็ได้ โดยการเปลี่ยนจาก X'TAL 4 MHz ของเดิมเป็น 6 MHz เท่านั้น

ROM หรือ EPROM โดยบอร์ด CP-Z80V2 จะต่อใช้ EPROM เป็น MONITOR PROGRAM ได้ 2 เบอร์ คือ เบอร์ 2764 หรือ เบอร์ 27256 โดยการเลือก JUMPER J1 หน่วยความจำนี้จะ DECODE อยู่ระหว่าง ADDRESS 0000H ถึง 7FFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RAM บอร์ด CP-Z80V2 จะต่อเลือกใช้หน่วยความจำ RAM ได้ 2 เบอร์ด้วยกัน คือ เบอร์ 6264 ขนาด 8K BYTE หรือเบอร์ 62256 ขนาด 32K BYTE โดยการเลือกใช้ JUMPER J2

หน่วยความจำนี้จะ DECODE อยู่ในตำแหน่ง 8000H ถึง FFFFH ใน RAM นี้ยังสามารถต่อใช้เก็บข้อมูลได้ในกรณีไฟดับ โดยต่อใส่ BATTERY ขนาดเล็ก 3V แบบตัวกลมใหญ่ BACK UP ข้อมูลใช้ J6 (BAT) ในการ ON/OFF BATTERY

PORT บอร์ด CP-Z80V2 จะมี IC PORT 8255 ให้ใช้งานได้ 2 ตัว หรือ 6 PORT ต่อใช้งาน โดยจะต่อออก PORT ทางขั้ว 34 PIN มาตรฐาน อีทีที 2 ชุด ทำให้สามารถเลือกใช้ต่อกับอุปกรณ์บอร์ดต่างๆ ของ อีทีทีได้มากมาย เช่น ET-SSRAC,ET-SMCC,ET-AD นอกจากนี้ยังมีการต่อขั้ว 10 PIN จาก PORT C ของ 8255 ให้สามารถเลือกต่อ KEY BOARD ขนาด 4x4 (16 KEY) ได้ด้วย ตำแหน่งการ DECODE นั้นเราสามารถใช้ J6 ซึ่งต่อจาก ADDRESS (A7) DECODE ตำแหน่งเบอร์ได้ 2 แบบ

LCD PORT จะเป็นลักษณะ CONNECTOR ขนาด 20 PIN มาตรฐาน อีทีที สามารถเลือกต่อ LCD ได้ 2 แบบ คือ แบบ DOT MATRIX LCD หรือ GRAPHIC LCD ได้โดยตรง เพียงต่อสายต่อจาก LCD นั้นๆ มายังขั้วของ LCD PORT 20 PIN นี้โดยดูขาสัญญาณให้ถูกต้องตรงกัน JUMPER J4 จะเป็นตัวเลือกว่าใช้ LCD แบบใดอยู่และ VR 10 K จะเป็นตัวปรับความคมชัดของจอภาพ LCD

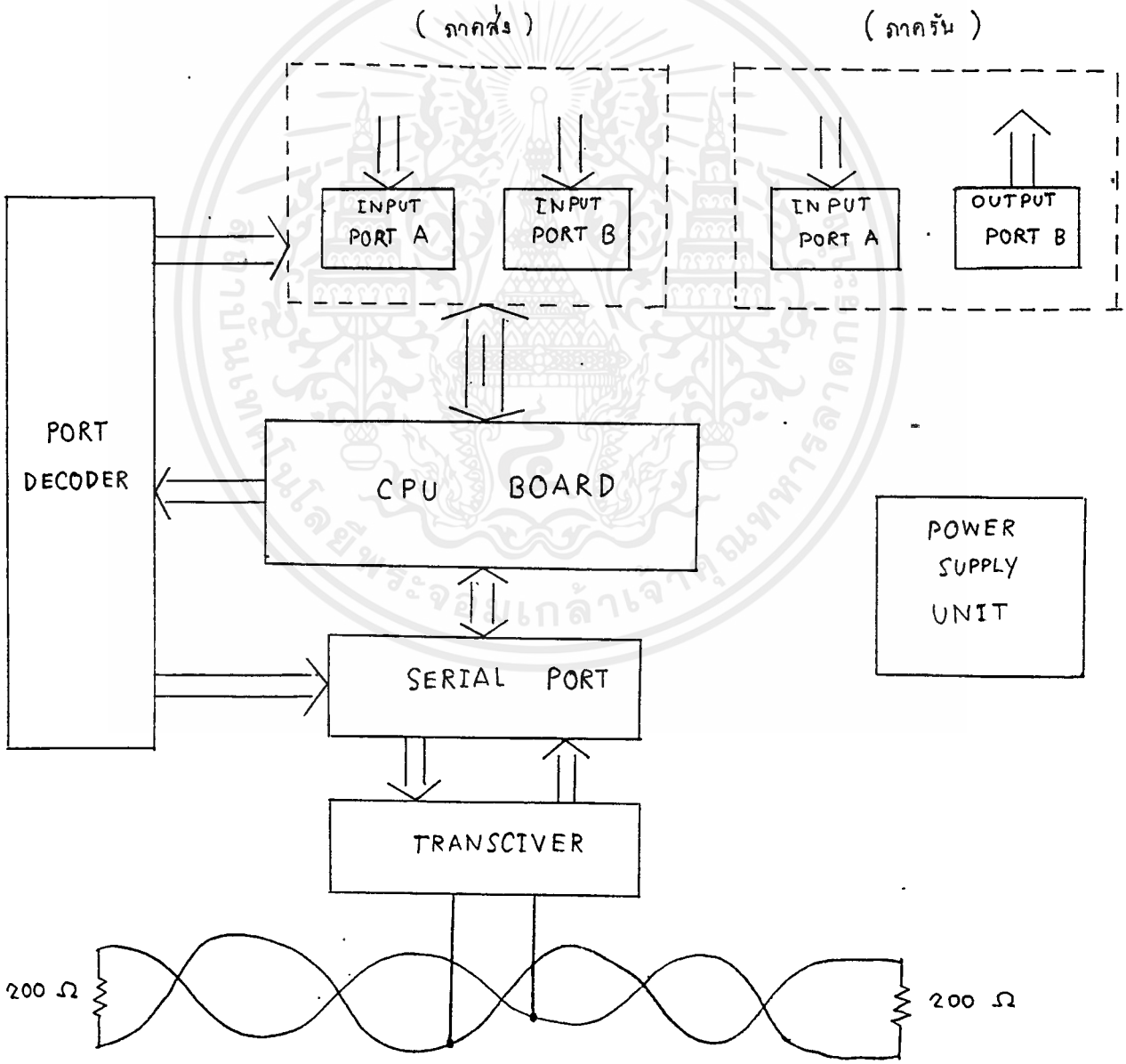
LCD ADDRESS PORT	
CHANNEL PORT	ADDRESS PORT SET J5=OFF
WRITE DATA INSTRUCTION	60H
WRITE DATA TO CG OR DD RAM	62H
READ BUSY FLAG AND ADDRESS	64H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของภาคส่งและภาครับ

โครงสร้างของภาคส่งและภาครับนี้จะมีลักษณะใกล้เคียงกันเกือบทั้งหมดจะแตกต่างกันก็เพียงส่วนพอร์ตข้อมูลเข้าฝออกเท่านั้น สามารถแยกเป็นส่วนประกอบหลัก ได้ดังนี้

1. CPU BOARD
2. วงจรเลือกพอร์ต
3. ส่วนอินพุทและเอาท์พุทพอร์ตแบบขนาน
4. ส่วนอินพุทและเอาท์พุทพอร์ตแบบอนุกรม
5. ส่วนแหล่งจ่ายไฟ



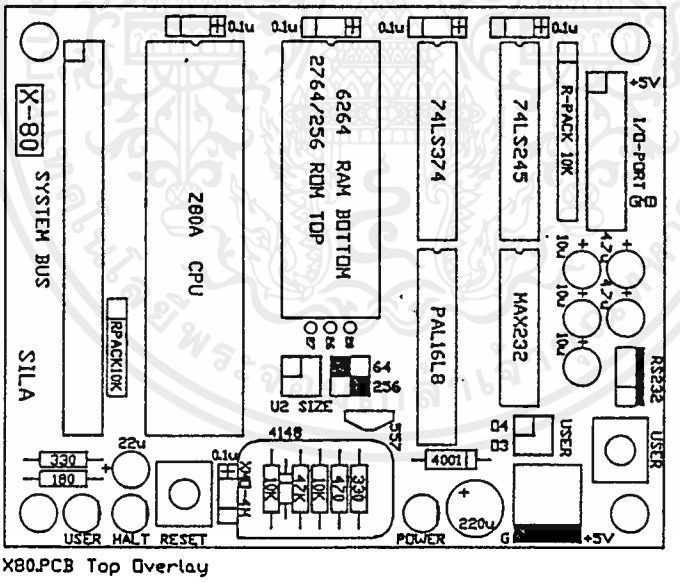
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของภาคส่ง/ภาครับ

ทางด้านภาคส่งจะประกอบไปด้วย CPU BOARD เป็นส่วนสำคัญโดยใช้ไมโครโปรเซสเซอร์ Z-80A ของ Zilog ทำงานที่ความถี่ 4.00 MH$_z$ ซึ่งใช้ภาคกำเนิดสัญญานนาฬิกาแบบ clock module ซึ่งมีความเที่ยงตรงสูงกว่าการใช้ x'tal ธรรมดามาก

โปรแกรมมอนิเตอร์ของ CPU BOARD นี้จะใช้ EPROM ขนาด 8Kbyte, RAM ชนิดสแตติก ขนาด 8 Kbyte ซึ่ง CPU BOARD นี้มีส่วนของการอินเตอร์เฟส กับ RS-232 อยู่ด้วยจึงช่วยให้การพัฒนาโปรแกรมโดยผ่านทางเครื่อง PC สามารถทำได้ง่ายกว่าการพัฒนาโปรแกรมโดยวิธีอื่นๆ

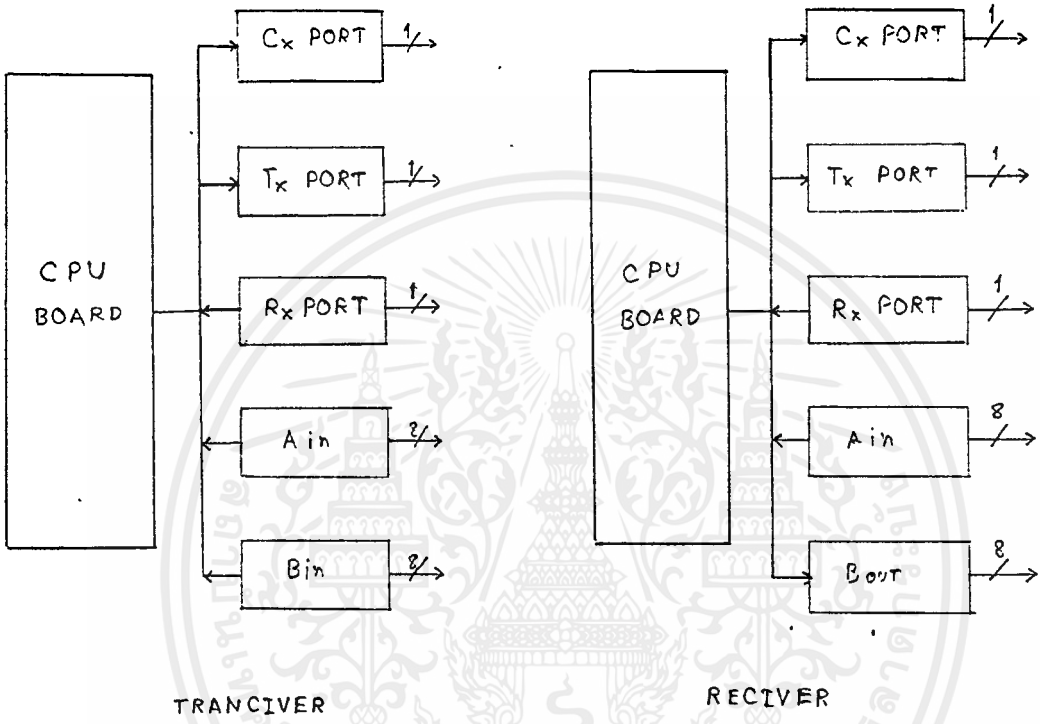
วงจรของ CPU BOARD และผังแสดงตำแหน่งของหน่วยความจำและ I/O ของ CPU BOARD นี้ แสดงดังรูปที่ 2 และรูปที่ 3 ตามลำดับ



รูปที่ 3 แสดง MEMORY MAP และ I/O MAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสำคัญอีกส่วนหนึ่งที่จำเป็นที่จะต้องออกแบบและสร้างขึ้นก็คือส่วน I/O PORT ซึ่งทั้งภาครับและภาคส่ง จะมีส่วนที่ใกล้เคียงกัน กล่าวคือส่วนภาคส่งจะมีพอร์ต ทั้งสิ้น 5 พอร์ต และภาครับก็จะมีพอร์ตทั้งสิ้น 5 พอร์ตเช่นกัน แต่จะแตกต่างกันที่ภาคส่งจะมี INPUT PORT 2 พอร์ท ในขณะที่ภาครับจะมี INPUT PORT และ OUTPUT PORT อย่างละ 1 พอร์ต ดังรูปที่ 4



รูปที่ 4 แสดงโครงสร้างด้านพอร์ตของภาคส่งและภาครับ

โดยที่ 5 พอร์ตนี้จะประกอบด้วย

Cx-PORT เป็นพอร์ต ขนาด 1 BIT ใช้ควบคุมการรับ/ส่งข้อมูลอนุกรม

Tx-PORT เป็นพอร์ต ขนาด 1 BIT ใช้ส่งข้อมูลอนุกรม

Rx-PORT เป็นพอร์ต ขนาด 1 BIT ใช้รับข้อมูลอนุกรม

Ain เป็นพอร์ต ขนาด 8 BIT ใช้รับข้อมูลแบบขนานจากอุปกรณ์ภายนอก

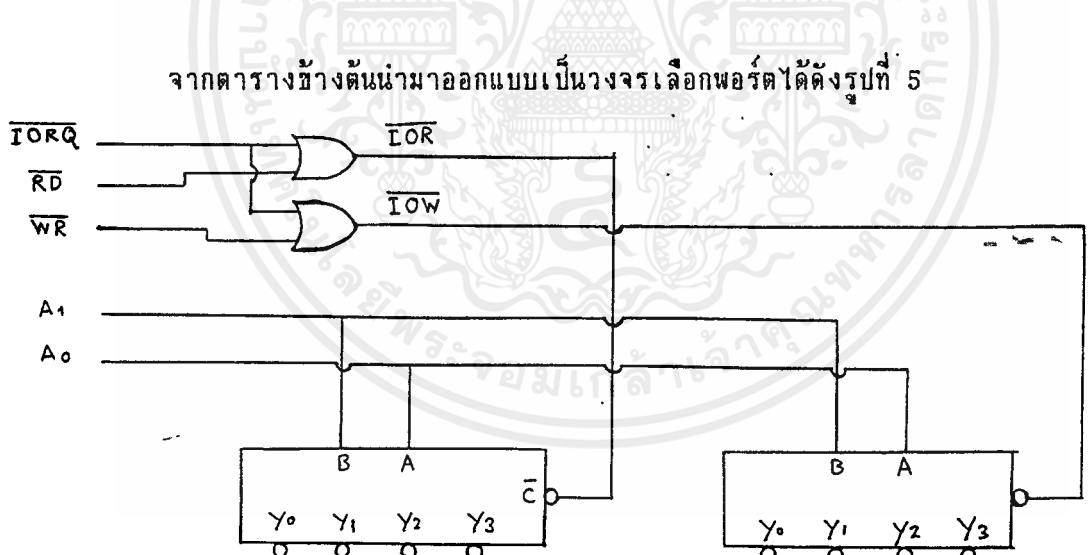
Bin/out เป็นพอร์ต ขนาด 8 BIT ในกรณีที่เป็นภาคส่งจะเป็น INPUT PORT ใช้รับข้อมูลกลุ่มที่ 2 แต่ในกรณีที่เป็นภาครับจะเป็น OUTPUT PORT ใช้ส่งข้อมูลควบคุมแบบขนานออกสู่อุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างการใช้พอร์ตของระบบสามารถสรุปได้ดังตารางต่อไปนี้

PORT NAME	FUNCTION	READ/WRITE	ADDRESS
Tx PORT	OUT	WRITE	01 H
Rx PORT	IN	READ	
Cx PORT	OUT	WRITE	02 H
AIN	IN	READ	
BIN /BOUT	IN/OUT	READ/WRITE	03 H

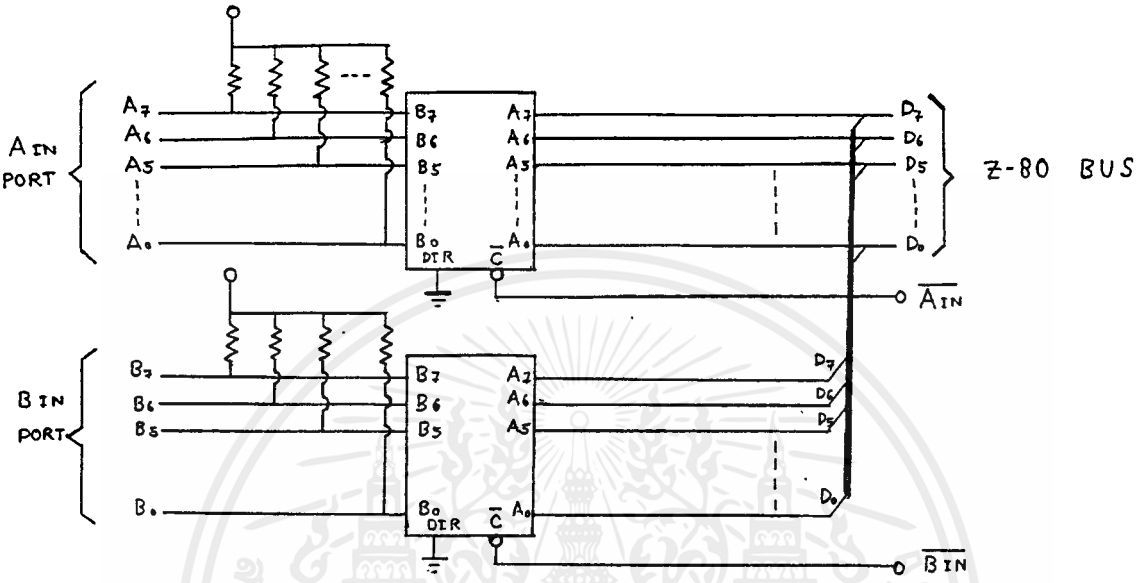
จากตารางข้างต้นนำมาออกแบบเป็นวงจรเลือกพอร์ตได้ดังรูปที่ 5



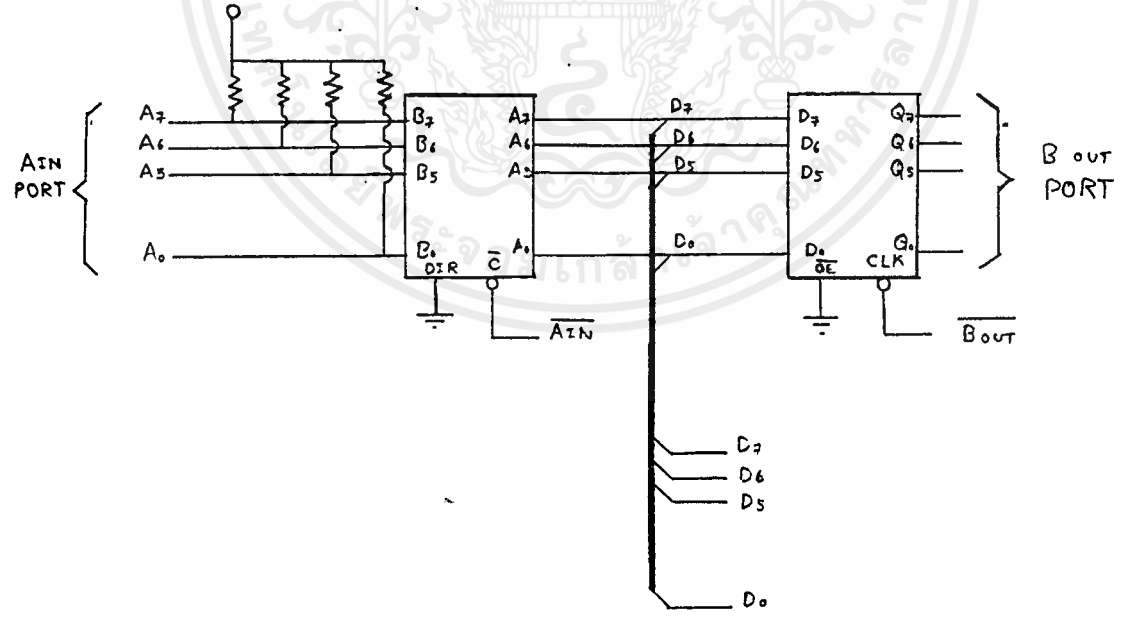
รูปที่ 5 แสดงวงจรเลือกพอร์ต (ใช้ทั้งภาครับและส่ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนอินพุทพอร์ตและเอาท์พุทพอร์ตแบบขนาน 8 บิต เป็นส่วนที่ภาคส่งและภาครับใช้ในการติดต่อกับอุปกรณ์ภายนอก ซึ่งมีการกำหนดคุณสมบัติของพอร์ตไว้ตายตัว จึงไม่จำเป็นต้องใช้ชิพชัพพอร์ตที่โปรแกรมได้ ในกรณีที่เป็นอินพุทพอร์ตจึงใช้ TTL IC เบอร์ 74LS245 และใช้ 74LS374 กรณีที่เป็นเอาท์พุทพอร์ต ส่วนพอร์ตของภาคส่งแสดงดังรูปที่ 6 และของภาครับแสดงดังรูปที่ 7



รูปที่ 6 แสดงส่วนอินพุทพอร์ตของภาคส่ง

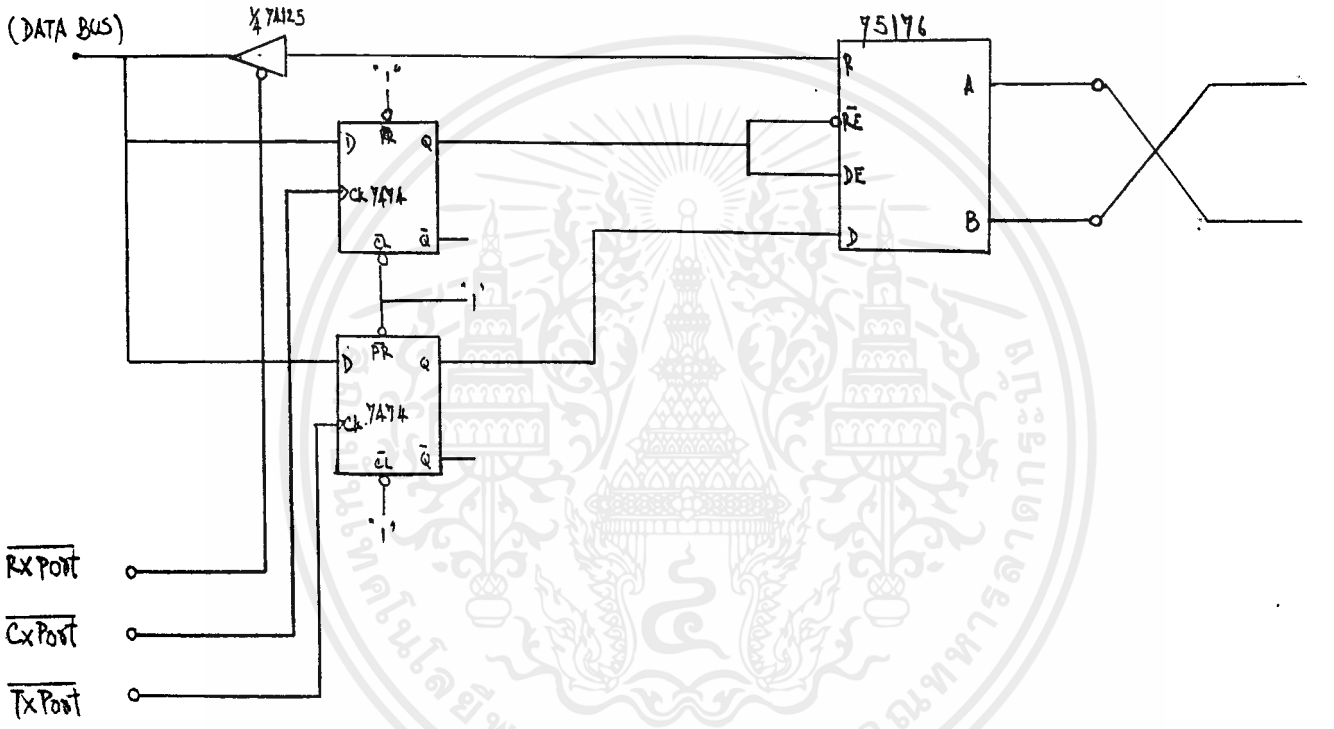


รูปที่ 7 แสดงส่วนอินพุทและเอาท์พุทพอร์ตของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

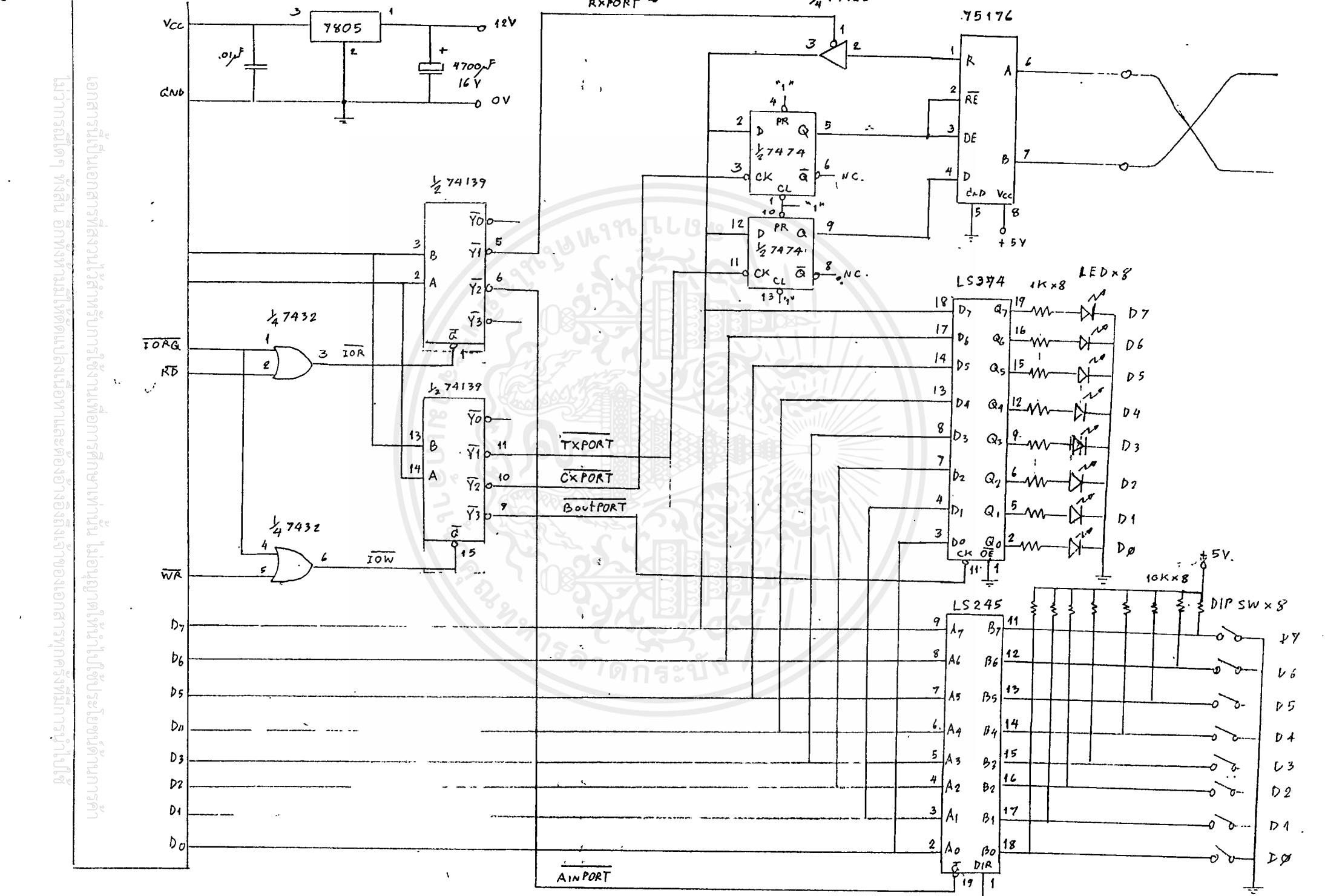
ส่วนอินพุทและเอาท์พุทพอร์ตแบบอนุกรมที่จะใช้ในการสื่อสารแต่ละเครื่องเข้าด้วยกันนั้น เพื่อหลีกเลี่ยงความซับซ้อนของส่วน Hardware และเพื่อประยุกต์ใช้ความสามารถของระบบไมโครโปรเซสเซอร์อย่างเต็มที่ จึงใช้วิธีสร้างพอร์ตชนิดอนุกรมขึ้นจากซอร์ฟแวร์

พอร์ตที่เกี่ยวข้องคือ CxPORT, TxPORT และ RxPORT ซึ่งแต่ละพอร์ตจะมีขนาดเพียง 1 บิต จึงสามารถนำอุปกรณ์ Tristate Buffer ธรรมดา มาใช้แทน INPUT PORT ได้ และนำ D-FLIPFLOP มาใช้แทน OUTPUT PORT ได้ วงจรในส่วนนี้จะเหมือนกันทั้งภาคส่งและภาครับ และนำมาเชื่อมต่อกับ IC 75176 ซึ่งเป็น Transceiver ตามมาตราฐาน EIA RS-485 ได้ทันที วงจรในส่วนนี้แสดงดังรูปที่ 8



รูปที่ 8 แสดงวงจรพอร์ตแบบอนุกรมและการต่อกับ 75176

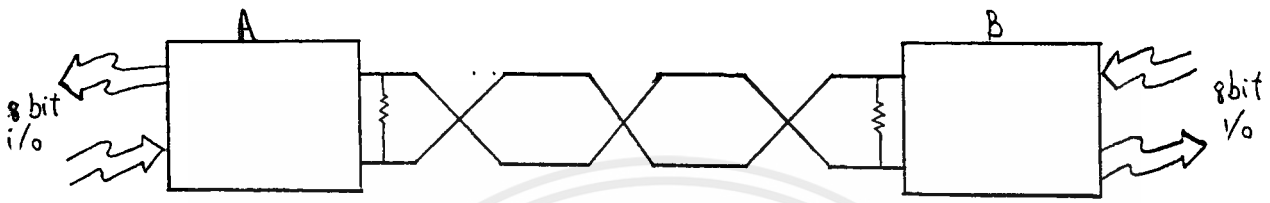
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Vcc
GND
7805
12V
0V
.01μF
4700μF
16V
1/2 74139
1/4 7432
IORQ
RD
WR
IOR
IOW
TXPORT
CXPORT
BoutPORT
AINPORT
RXPORT
1/2 7474
NC.
75176
R
RE
DE
D
A
B
GND
Vcc
+5V
LS374
1K×8
LED×8
LS245
10K×8
DIP SW×8
D7
D6
D5
D4
D3
D2
D1
D0

ส่วนประกอบทางซอฟท์แวร์

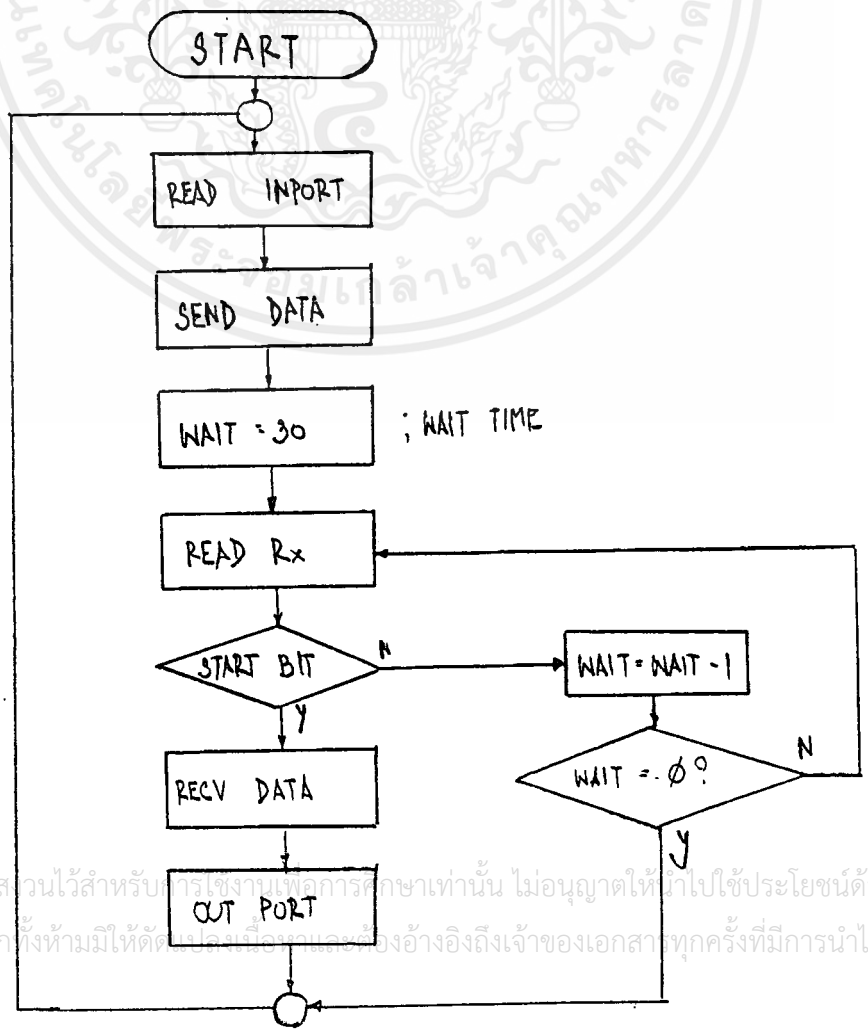
เนื่องจากการทำงานของทั้งภาคส่งและภาครับจะควบคุมโดย ไมโครโปรเซสเซอร์ (Z-80) โปรแกรมต่าง ๆ จะถูกเก็บไว้ในหน่วยความจำชนิด EPROM ที่แอดเดรส 0000H เพื่อสามารถเริ่มทำงานได้ทันทีเมื่อป้อนไฟเลี้ยงให้แก่วงจร

เนื่องจากการทำงานของโปรแกรมค่อนข้างจะซับซ้อน เพื่อความเข้าใจจะกล่าวถึงการรับส่งข้อมูลในลักษณะ INTERACTIVE และมีการต่อระหว่างแต่ละชุดเป็น แบบ Point to Point ดังรูปที่ 9



โดยที่ข้อมูลที่เข้าทาง A จะไปปรากฏที่ OUTPUT ของ B และในทำนองเดียวกัน ข้อมูล INPUT ทาง A ในเวลาเดียวกันและเปลี่ยนแปลงตาม INPUT อีกฝั่งหนึ่งอย่างต่อเนื่อง

จากข้อกำหนดดังกล่าว จะเห็นได้ว่าโปรแกรมของทั้งสองภาคจะมีลักษณะที่เหมือนกันสามารถเขียนเป็น FLOW CHART และโปรแกรมของ Z-80 ได้ดังนี้

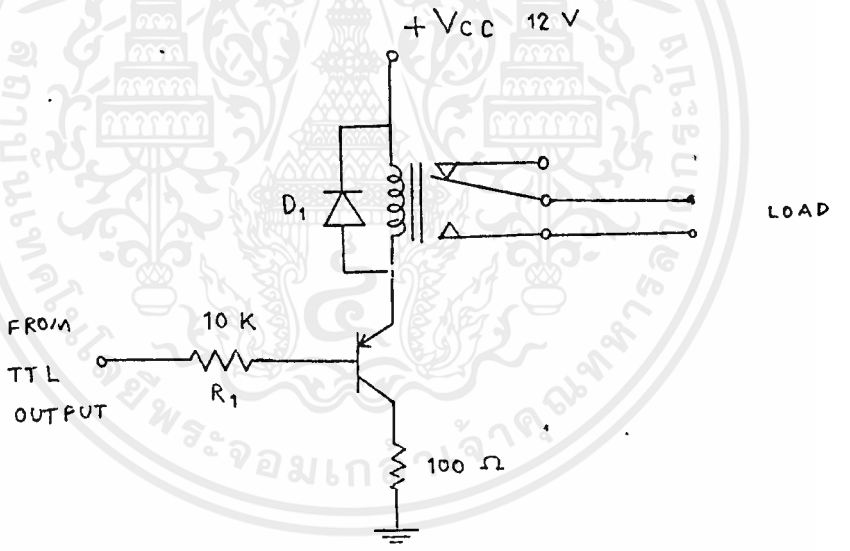


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร OUTPUT INTERFACE

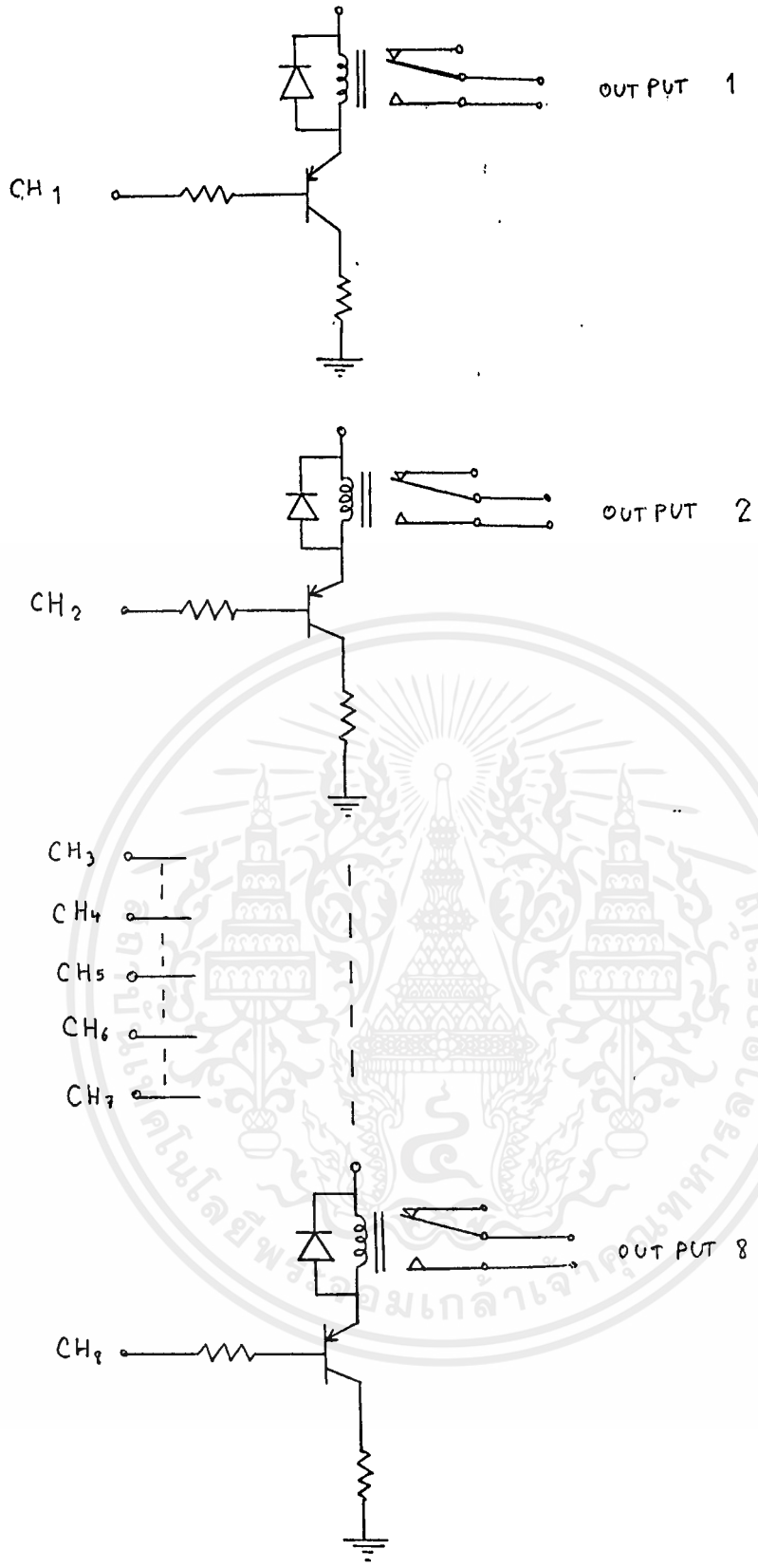
วงจรนี้จะทำหน้าที่เป็นตัวเชื่อมสัญญาณข้อมูลทางด้าน OUTPUT ของเครื่องรับกับ LOAD ที่ต้องการควบคุมโดยไม่กำหนดว่า LOAD จะเป็นอะไร จึงใช้ RELAY เนื่องจาก RELAY มีคุณสมบัติที่เป็น SW. ทางด้าน OUTPUT ซึ่งก็ตรงกับข้อมูลทางด้าน INPUT ของเครื่องส่ง ที่มีลักษณะเป็น SW. CONTROL เช่นกัน ดังนั้นลักษณะการทำงานของการควบคุม LOAD ทางด้านเครื่องรับจะเหมือนการควบคุมทางด้านเครื่องส่งทุกประการ

สัญญาณที่ได้จาก OUTPUT PORT ของเครื่องรับในแต่ละ CHANNEL จะถูกส่งผ่านไปยังวงจร OUTPUT INTERFACE ของในแต่ละ CHANNEL นั้น โดยสัญญาณข้อมูลจะถูกส่งผ่าน R_1 เข้าไปยัง TR_1 ทำให้การควบคุม RELAY เพื่อให้ OUTPUT ของ RELAY ทำงานตรงกับสัญญาณควบคุมจากทางเครื่องส่ง ส่วน D_1 นั้น ต่อไว้เพื่อกัน TR_1 เสียหาย เนื่องจากแรงดันย้อนกลับขณะ TR_1 ON-OFF



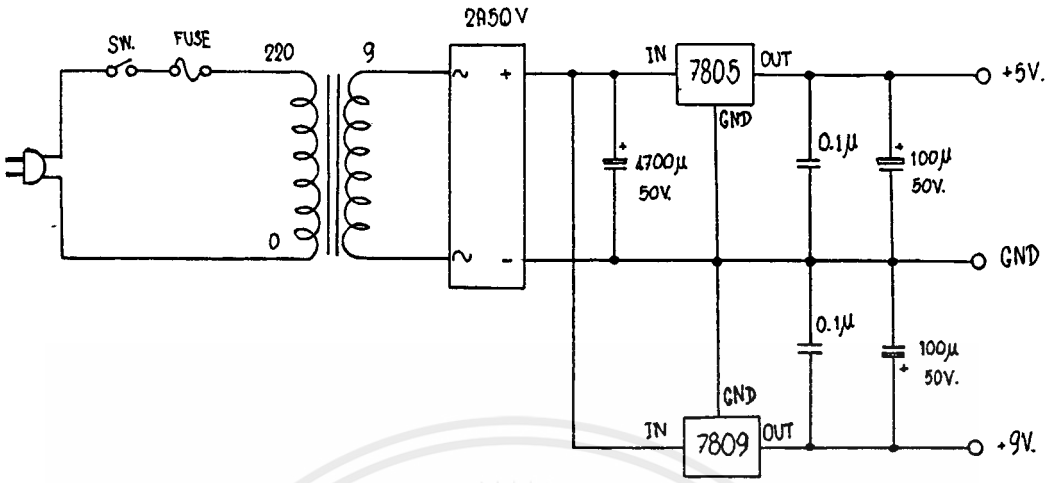
วงจร OUTPUT INTERFACE ของแต่ละ CHANNEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

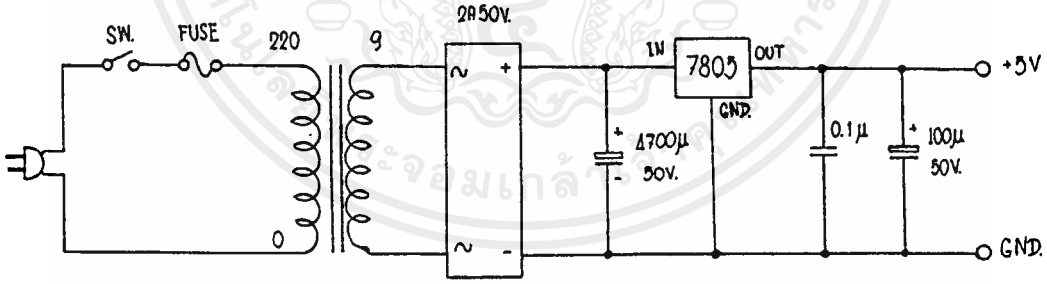


วงจร OUTPUT INTERFACE ของทุก CHANNEL แบบสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



POWER SUPPLY (ด้านส่ง)



POWER SUPPLY (ด้านรับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;INTERACTIVE TRANSCEIVERS PROGRAM
RXPORT     EQU  01H
TXPORT     EQU  01H
CXPORT     EQU  02H
AIN        EQU  02H
BOUT       EQU  03H
SEND_COM   EQU  80H
RECV_COM   EQU  00H
WAIT-TIME  EQU  7FH
BAUDMM     EQU  15H
INIT:  LD     HL,06FFFH
       LD     SP,HL          ;กำหนด STACK ของระบบ
       XOR    A              ;CLEAR สายสื่อสาร
       OUT    (TXPORT),A
       OUT    (CXPORT),A
       OUT    (BOUT),A       ;CLEAR OUTPORT
START: IN     A,(AIN)
       LD     D,A
       LD     A,SEND_COM
       OUT    (CXPORT),A
       CALL   SBYTE
       LD     B,WAIT_TIME
       LD     A,RECV_COM
       OUT    (CX PORT),A
       JR     R_BYTE
R_SUB: DJNZ   R_BYTE
       JR     START
R_BYTE:IN     A,(RXPORT)
       BIT    7,A
       JR     NZ,R_SUB
       NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        NOP
        NOP
        NOP
        CALL  RXDLY
        PUSH  BC
        LD    D,0
        LD    B,8
R_BYTE2:IN    A,(RXPORT)
        RLA
        RRD
        CALL RXDLY
        DJNZ R_BYTE2
        POP  BC
        LD   A,D
        OUT (BOUT),A
END:    JR    START
RXDLY: LD    A,BAUDMM
        ADD   A,1
RXDLY1:DEC   A
        JR    NZ,RXDLY1
        RET
SBYTE: PUSH  DE
        CALL  TXDLY
        XOR   A
        OUT   (TXPORT),A
        CALL  TXDLY
        PUSH  BC
        LD    B,8
SBYTE1:XOR   A
        RRC   D
        RRA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT    (TXPORT),A
        CALL   TXDLY
        DJNZ   SBYTE1
        POP    BC
        LD     A,80H
        OUT    (RXPORT),A
        CALL   TXDLY
        POP    DE
        RET
TXDLY:  LD     A,BAUDMM
TXDLY1: DEC    A
        JR     NZ,TXDLY1
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง

4.1 ผลการทดลอง

การส่งและรับสัญญาณโดยใช้ LINE DRIVE 75176

1. OUTPUT ที่ 1 ใช้สายหลาย ๆ ชนิดมาต่อกัน มีความยาว 66 เมตร
2. OUTPUT ที่ 2 และ 3 ใช้สายโทรศัพท์ยาว 100 เมตร มาต่อกลับไปกลับมาให้ได้ 1 กิโลเมตร และ 1.5 กิโลเมตร ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ปัญหาที่เกิดขึ้น

1. การประกอบวงจรใช้แผ่นวงจรเอนกประสงค์ ทำให้เกิดการผิดพลาดได้ง่าย

2. สายนำสัญญาณที่ใช้ในการประกอบวงจร ไม่เป็นไปตามสภาพที่จะใช้งานจริง โดยตามปกติที่จะใช้งานจริง สายนำสัญญาณจะต้องเหยียดตรง และเป็นคู่เดียวยาวตลอด แต่ที่ใช้ในการทดลองจะใช้สายโทรศัพท์ขนาด 25 คู่สาย ความยาว 100 เมตร นำมาต่อที่ปลายสายกลับไปกลับมาให้ได้ความยาว 1 กิโลเมตร และสายอยู่ในสภาพที่เป็นขด ซึ่งอาจจะทำให้มีผลต่อสัญญาณจากการทำตัวเป็น INDUCTANCE ของขดลวด

3. เนื่องจากระบบ TWO WORE REMOTE ยังถือว่าเป็นเทคโนโลยีสมัยใหม่อยู่ และมีราคาสูง ยังไม่ค่อยมีผู้ทำการวิจัยมากนัก ทำให้การค้นคว้าหาข้อมูลเป็นไปด้วยความยากลำบากและใช้เวลามาก แต่เวลาที่ใช้ในการทำโครงงานของภาควิชา มีระยะเวลาสั้น จึงทำให้โครงงานนี้ไม่เป็นไปตามเป้าหมายที่ตั้งไว้เท่าที่ควร

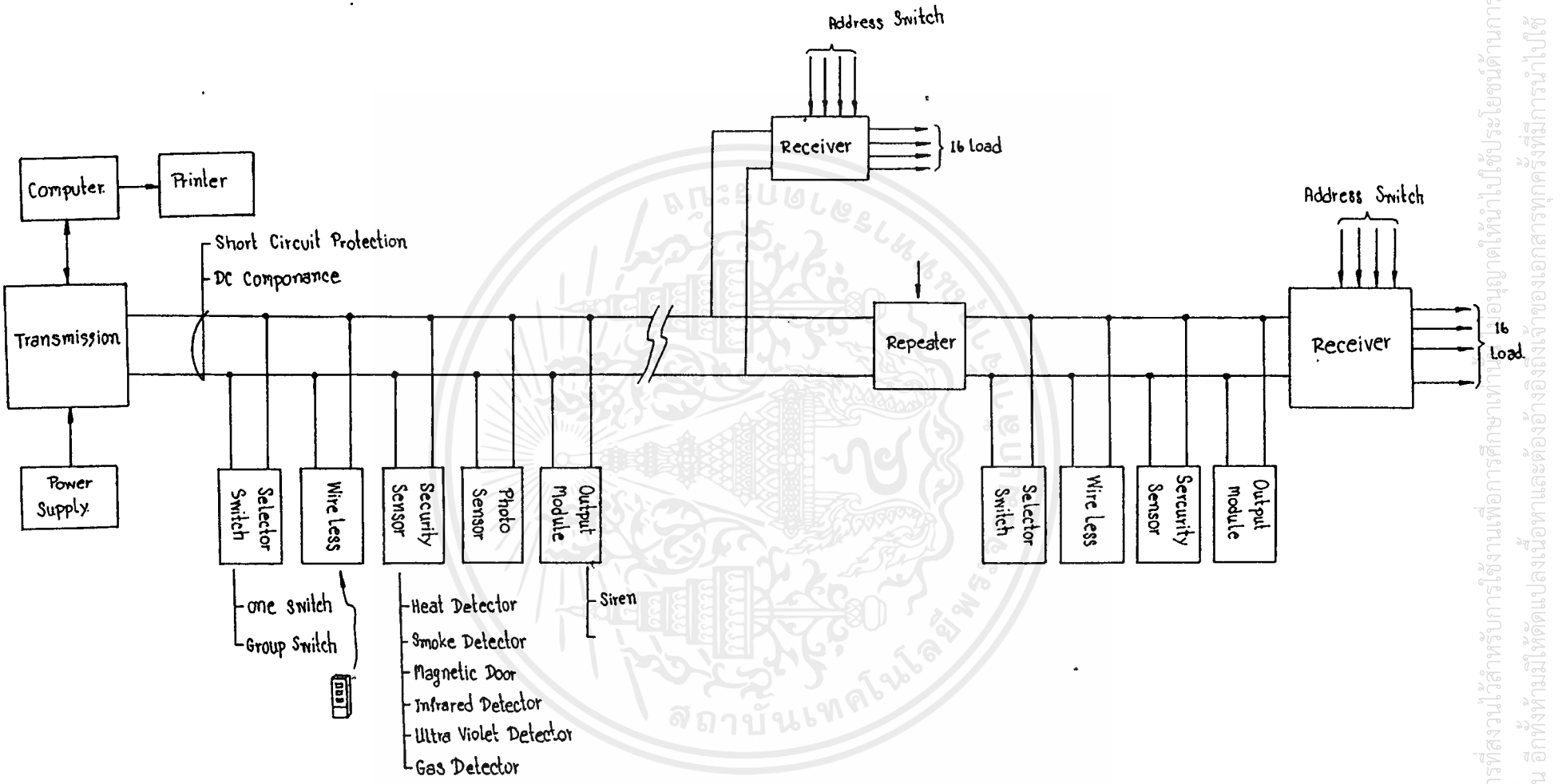
4.3 แนวทางแก้ไข

1. ออกแบบและทดสอบวงจรให้แน่นอน แล้วออกแบบเป็นแผ่นวงจรพิมพ์

2. ใช้สายนำสัญญาณที่ใช้งานจริงขนาด 1 คู่สาย ความยาว 1 กิโลเมตรขึ้นไป มาใช้ในการทดลองและทำให้สายเหยียดตรง ซึ่งต้องเสียค่าใช้จ่ายค่อนข้างสูง

3. ควรคิด วางแผน และหาข้อมูลในการทำโครงงานตั้งแต่เทอมแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แนวทางพัฒนา.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางพัฒนา

1.SELECTOR SWITCH

ทำหน้าที่ On-Off Load ตัวใดก็ได้ตามต้องการ ขึ้นอยู่กับ Address Switch และ Program ที่ตั้งไว้ซึ่ง Selector Switch นี้จะต่อขนานกับสายสัญญาณของ TWO WIRE REMOTE ทำให้สามารถติดตั้ง Selector Switch ที่ตำแหน่งใดของหน่วยงานก็ได้

-One Switch คือ Switch 1 ตัว จะควบคุม Load 1 ตัว จะเป็น Load ตัวใดก็ได้ขึ้นอยู่กับ Program ที่ตั้งไว้

-Group Switch คือ Switch 1 ตัว จะควบคุม Load ได้ทีละหลายๆตัว ขึ้นอยู่กับ Program ที่ตั้งไว้

2.WIRE LESS REMOTE CONTROL

ทำหน้าที่คล้าย Selector Switch แต่จะเป็นตัว Wire Less Remote Control และมีตัวรับ Remote ซึ่งจะต่อกับสายสัญญาณของ TWO WIRE REMOTE

3.PHOTO SENSOR

ทำหน้าที่ตรวจจับความสว่างของแสงในแต่ละพื้นที่ เช่น ตรวจจับความสว่างของแสงอาทิตย์ที่ส่องเข้ามาในห้อง แล้วส่งข้อมูลไปให้ Transmission ถ้า CPU ประมวลผลแล้วว่ามีแสงสว่างเพียงพอก็จะสั่งให้ Receiver ที่ควบคุม Load ในพื้นที่นั้น (ที่ตัว Photo Sensor อยู่) ทำการปิด Load ที่ไม่จำเป็น เพื่อการประหยัดพลังงาน

4.SERCURITY SENSOR

เป็น Sensor ในด้านความปลอดภัยต่างๆของหน่วยงาน เช่น ตรวจจับควันไฟ (Smoke Detector) ความร้อน (Heat Detector) ตรวจจับความเคลื่อนไหว (Infrared and Ultraviolet) แกส (Gas Detector) และ Magnetic Door เมื่อ Sensor ตัวใดตรวจจับความผิดปรกติได้ก็จะส่งข้อมูลไปให้ Transmission เพื่อแสดงผลที่ห้องปฏิบัติกลาง ว่าเกิดการผิดปรกติใดเกิดขึ้น ณ พื้นที่ใด

5.OUTPUT MODULE UNIT

ทำหน้าที่เป็น Output ของ Sercurity Sensor คือเมื่อ Sercurity Sensor ตรวจจับความผิดปรกติได้จะส่งข้อมูลไปที่ตัว Transmission ทำการแสดงผลที่ห้องปฏิบัติการกลางและ Transmission จะส่งข้อมูลมาที่ Output Module เพื่อแสดงผลให้ทราบอีกทางหนึ่งด้วย เช่น ทำให้ Siren ดัง หรือ Flasher Light สว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.REPEATER

ในมาตราฐาน RS-485 นั้น สามารถส่งข้อมูลได้ไกล 4000 ฟุต (1.2 กม.) ซึ่งถ้าต้องการส่งข้อมูลให้ได้ไกลกว่านี้ จำเป็นต้องมี Repeater ทำหน้าที่ทวนสัญญาณ

7.DC COMPONANCE

โดยการทำให้ระบบทั้งหมดของ TWO WIRE REMOTE มีแหล่งจ่ายอยู่ที่เดียว คือที่ Transmission ส่วนที่ตัว Receiver และอุปกรณ์ที่เชื่อมต่ออยู่กับสายสัญญาณของ TWO WIRE REMOTR ทุกตัว ไม่ต้องมีแหล่งจ่ายไฟของตัวเอง

8.SHORT CIRCUIT PROTECTION

เมื่ออุปกรณ์ตัวใดตัวหนึ่งที่เชื่อมต่ออยู่กับสายสัญญาณของ TWO WIRE REMOTE หรือตัวสายส่งเองเกิดการลัดวงจรขึ้น ตัว Transmission จะต้องทำการตัดสายส่งสัญญาณออก เพื่อป้องกันความเสียและแสดงผลว่าเกิดการลัดวงจรเกิดขึ้น จนกว่าการลัดวงจรนั้นจะได้รับการแก้ไข

9.COMPUTER AND PRINTER

เป็นการเชื่อมต่อระหว่าง Transmission กับ Computer และ Printer เพื่อการแสดงผลที่ Computer และสามารถ List ข้อมูลออกมาดูได้ว่าตั้ง Program อะไรไว้บ้าง และสามารถ Print ข้อมูลออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 วิจารณ์และสรุป

ปริญญานิพนธ์นี้เสนองานวิจัยเกี่ยวกับ การส่งสัญญาณควบคุมระยะไกล โดยใช้สาย 2 เส้น ที่ใช้ไมโครโปรเซสเซอร์ ในการควบคุมระบบทั้งหมด ทำให้ทำงานได้อย่างรวดเร็ว และมีประสิทธิภาพสูงไม่มีการผิดพลาด อีกทั้งอุปกรณ์ทั้งหมดที่ใช้ในโครงงานหาซื้อได้ทั่วไปตามท้องตลาด ทำให้ระบบ TWO WIRE REMOTE ในปริญญานิพนธ์ สามารถผลิตได้เองภายในประเทศในราคาที่ถูกกว่าของจากต่างประเทศอย่างมาก และสะดวกในการซ่อมบำรุง

ในการทดลองการส่งและรับข้อมูลซึ่งใช้สายโทรศัพท์ขนาด 25 คู่สาย ความยาว 100 เมตร นำมาต่อที่ปลายสายกลับไปกลับมา ให้ได้ความยาว 1 กิโลเมตร เป็นสายนำสัญญาณ ผลปรากฏเป็นที่น่าพอใจอย่างยิ่ง คือสัญญาณเอาท์พุทที่ได้ มีลักษณะเหมือนอินพุท เพียงแต่เฟสทางเอาท์พุทล่าช้ากว่าทางอินพุทเพียงเล็กน้อยเท่านั้น ซึ่งไม่มีผลในการส่งข้อมูลแบบอนุกรม

ระบบ TWO WIRE REMOTE ที่สมบูรณ์แบบนั้น ได้เสนอไว้แล้วในหัวข้อแนวทางพัฒนา ซึ่งทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่า คงจะมีผู้สนใจในระบบ TWO WIRE REMOTE นี้ เพื่อจะคิดค้นและพัฒนาความสามารถของระบบ ให้มีขีดความสามารถดังที่คณะผู้จัดทำคาดหวังไว้ ซึ่งเมื่อพัฒนาขีดความสามารถของระบบ ให้ได้สมบูรณ์แบบแล้ว ยังจะสามารถนำมาประกอบในเชิงธุรกิจได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

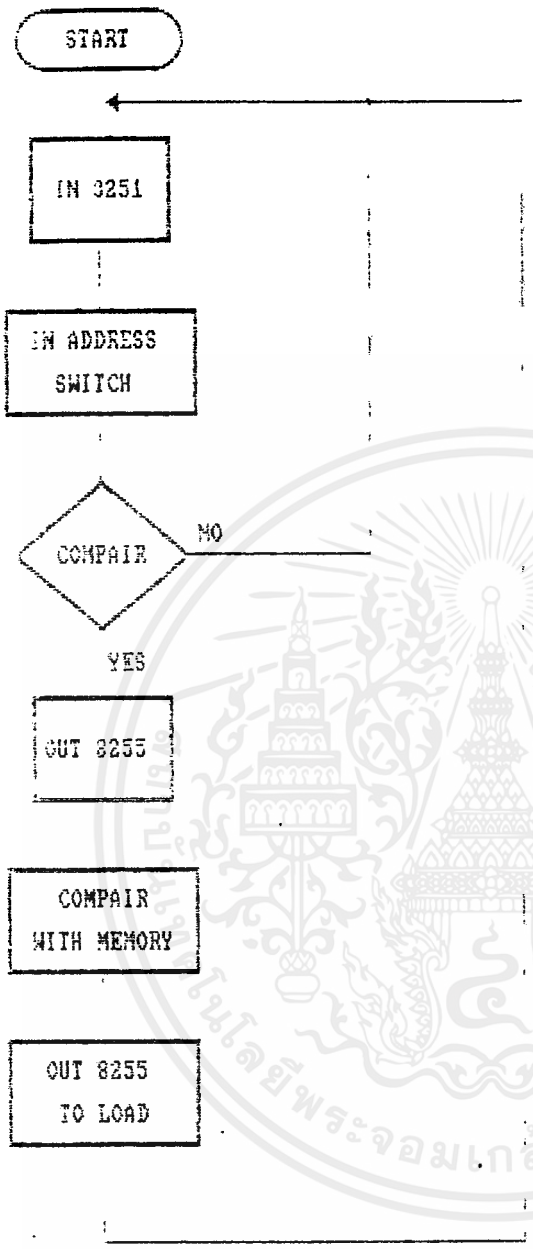
ภาคผนวก

-โฟลว์ชาร์ตการทำงานของ TWO WIRE REMOTE CONTROL SYSTEM

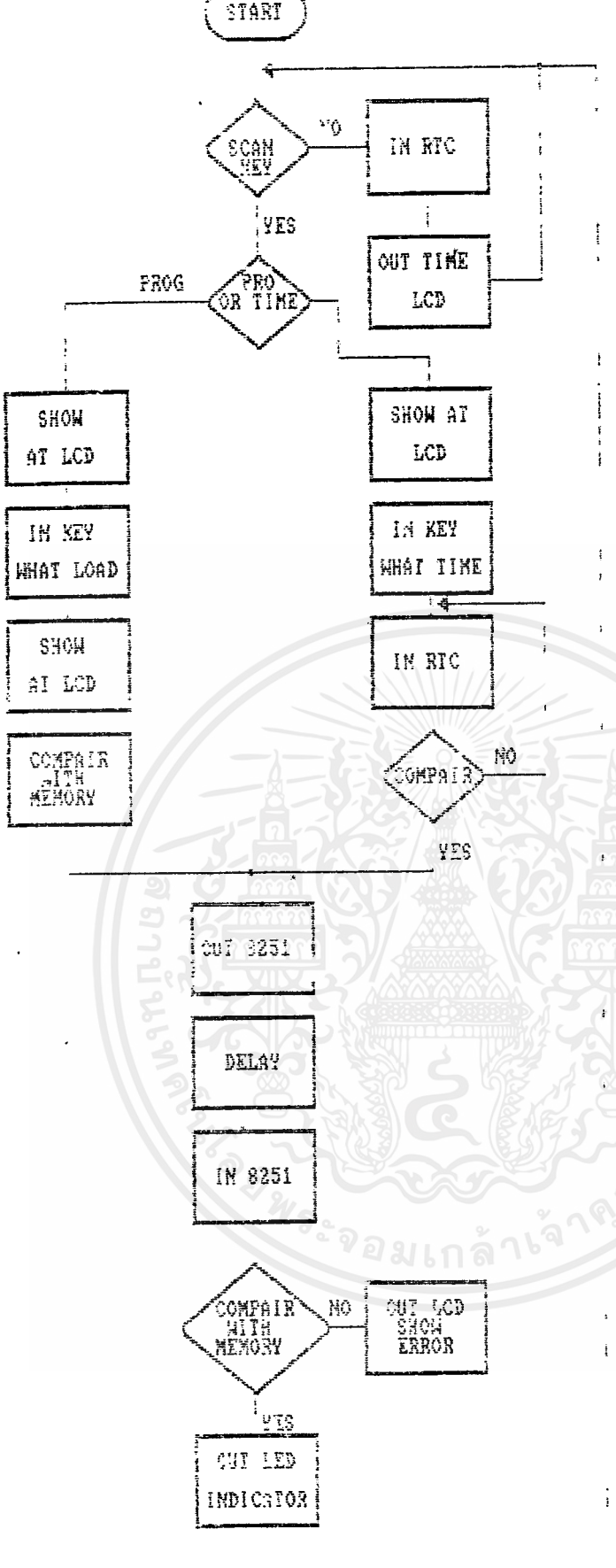
-โปรแกรมการทำงานของ TWO WIRE REMOTE CONTROL SYSTEM

-คู่มืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

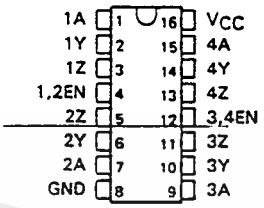


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D2601, OCTOBER 1980—REVISED OCTOBER 1986

- Meets EIA Standards RS-422-A and RS-485 and CCITT Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Outputs
- Common-Mode Output Voltage Range of −7 V to 12 V
- Active-High Enable
- Thermal Shutdown Protection
- Positive- and Negative-Current Limiting
- Operates from Single 5-V Supply
- Low Power Requirements
- Functionally Interchangeable with MC3487

J OR N DUAL-IN-LINE PACKAGE
(TOP VIEW)



FUNCTION TABLE (EACH DRIVER)

INPUT	ENABLE	OUTPUTS	
		Y	Z
H	H	H	L
L	H	L	H
X	L	Z	Z

H = TTL high level.
L = TTL low level.
X = irrelevant.
Z = High impedance (off)

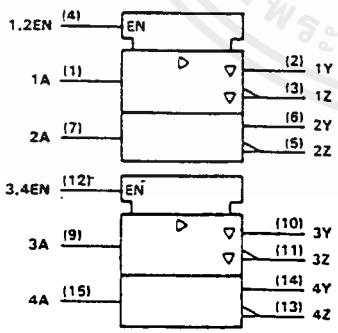
description

The SN75174 is a monolithic quadruple differential line driver with three-state outputs. It is designed to meet the requirements of EIA Standards RS-422-A and RS-485 and CCITT Recommendations V.11 and X.27. The device is optimized for balanced multipoint bus transmission at rates up to 4 megabaud. Each driver features wide positive and negative common-mode output voltage ranges making it suitable for party-line applications in noisy environments.

The SN75174 provides positive- and negative-current limiting and thermal shutdown for protection from line fault conditions on the transmission bus line. Shutdown occurs at a junction temperature of approximately 150°C. This device offers optimum performance when used with the SN75173 or SN75175 quadruple differential line receivers.

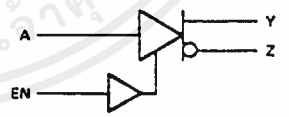
The SN75174 is characterized for operation from 0°C to 70°C.

logic symbol†



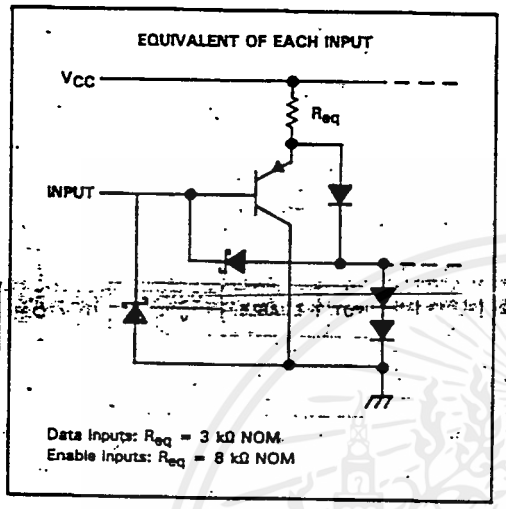
†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

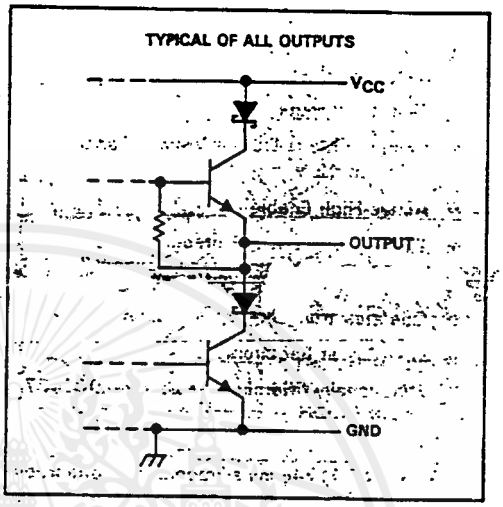
logic diagram, each driver (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematics of inputs and outputs





absolute maximum ratings over operating free-air temperature (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 5.5 V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2):
J package . . . . . 1375 mW
N package . . . . . 1625 mW
Operating free-air temperature . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: J package . . . . . 300°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: N package . . . . . 260°C

NOTES: 1. All voltage values are with respect to the network terminal.
2. For operation above 25°C free-air temperature, derate the J package to 880 mW at 70°C at the rate of 11.0 mW/°C and the N package to 1040 mW at 70°C at the rate of 13.0 mW/°C.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
High-level input voltage, V_{IH}	2			V
Low-Level input voltage, V_{IL}			0.8	V
Common-mode output voltage, V_{OC}			−7 to 12	V
High-level output current, I_{OH}			−60	mA
Low-level output current, I_{OL}			60	mA
Operating free-air temperature, T_A	0		70	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{IK}	Input clamp voltage	I_I = −18 mA				−1.5	V
V_{OH}	High-level output voltage	V_{IH} = 2 V, I_{OH} = −33 mA	V_{IL} = 0.8 V,		3.7		V
V_{OL}	Low-level output voltage	V_{IH} = 2 V, I_{OL} = 33 mA	V_{IL} = 0.8 V,		1.1		V
V_O	Output voltage	I_O = 0		0		6	V
$\|V_{OD1}\|$	Differential output voltage	I_O = 0		1.5		6	V
$\|V_{OD2}\|$	Differential output voltage	R_L = 100 Ω,	See Figure 1	½ V_{OD1}			V
		R_L = 54 Ω,	See Figure 1	1.5	2.5	5	V
V_{OD3}	Differential output voltage	See Note 3		1.5		5	V
$\Delta\|V_{OD}\|$	Change in magnitude of differential output voltage‡	R_L = 54 Ω or 100 Ω,	See Figure 1			±0.2	V
V_{OC}	Common mode output voltage					+3 −1	V
$\Delta\|V_{OC}\|$	Change in magnitude of common mode output voltage‡					±0.2	V
I_O	Output current with power off	V_{CC} = 0,	V_O = −7 V to 12 V			±100	μA
I_{OZ}	High-impedance-state output current	V_O = −7 V to 12 V				±100	μA
I_{IH}	High-level input current	V_I = 2.7 V				20	μA
I_{IL}	Low-level input current	V_I = 0.5 V				−360	μA
I_{OS}	Short-circuit output current	V_O = −7 V				−250	mA
		V_O = V_{CC}				180	
		V_O = 12 V				500	
I_{CC}	Supply current (all drivers)	No load	Outputs enabled		38	60	mA
			Outputs disabled		18	40	

† All typical values are at V_{CC} = 5 V and T_A = 25°C.
‡ $\Delta|V_{OD}|$ and $\Delta|V_{OC}|$ are the changes in magnitude of V_{OD} and V_{OC}, respectively, that occur when the input is changed from a high level to a low level.
NOTE 3: See EIA Standard RS-485 Figure 3.5, Test Termination Measurement 2.

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

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{DD}	Differential-output delay time	R_L = 54 Ω,	See Figure 2		45	65	ns
t_{TD}	Differential-output transition time				80	120	ns
t_{PZH}	Output enable time to high level	R_L = 110 Ω,	See Figure 3		80	120	ns
t_{PZL}	Output enable time to low level	R_L = 110 Ω,	See Figure 4		55	80	ns
t_{PHZ}	Output disable time from high level	R_L = 110 Ω,	See Figure 3		75	115	ns
t_{PLZ}	Output disable time from low level	R_L = 110 Ω,	See Figure 4		18	30	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL EQUIVALENTS

DATA SHEET PARAMETER	RS-422-A	RS-485
V_O	V_{oa}, V_{ob}	V_{oa}, V_{ob}
$\|V_{OD1}\|$	V_o	V_o
$\|V_{OD2}\|$	V_t (R_L = 100 Ω)	V_t (R_L = 54 Ω)
$\|V_{OD3}\|$		V_t (Test Termination Measurement 2)
$\Delta\|V_{OD}\|$	$\|\|V_t\| - \|\overline{V}_t\|\|$	$\|\|V_t\| - \|\overline{V}_t\|\|$
V_{OC}	$\|V_{os}\|$	$\|V_{os}\|$
$\Delta\|V_{OC}\|$	$\|V_{os} - \overline{V}_{os}\|$	$\|V_{os} - \overline{V}_{os}\|$
I_{OS}	$\|I_{sa}\|$, $\|I_{sb}\|$	
I_O	$\|I_{xa}\|$, $\|I_{xb}\|$	I_{ia}, I_{ib}

PARAMETER MEASUREMENT INFORMATION

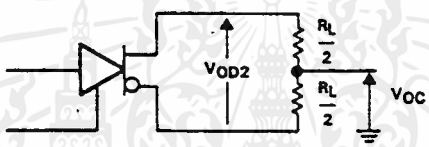
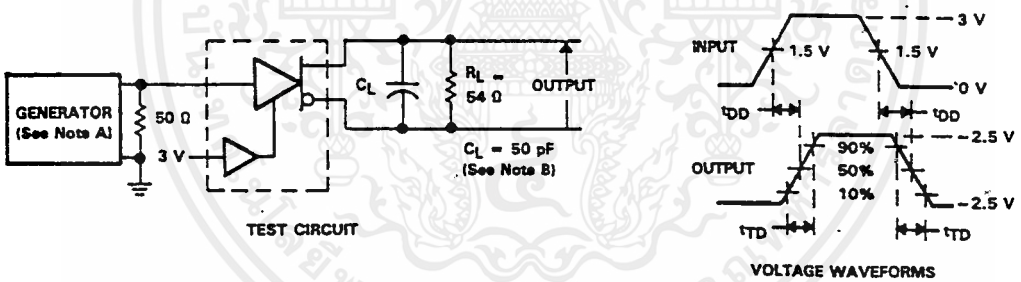


FIGURE 1. DIFFERENTIAL AND COMMON-MODE OUTPUT VOLTAGES

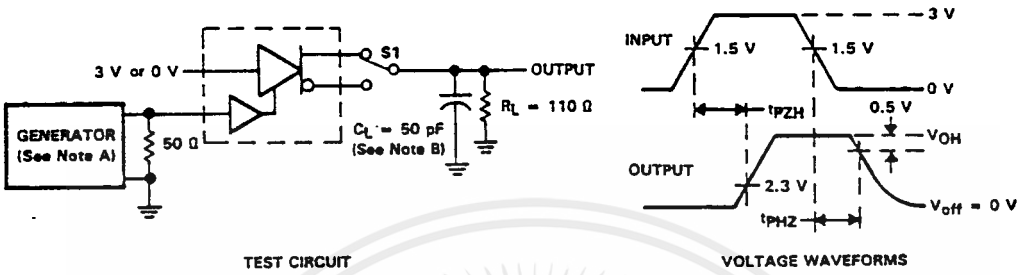


NOTES: A. The input pulse is supplied by a generator having the following characteristics: $t_r \leq 5$ ns, $t_f \leq 5$ ns, PRR ≤ 1 MHz, duty cycle = 50%, Z_O = 50 Ω.
B. C_L includes probe and stray capacitance.

FIGURE 2. DIFFERENTIAL-OUTPUT DELAY AND TRANSITION TIMES

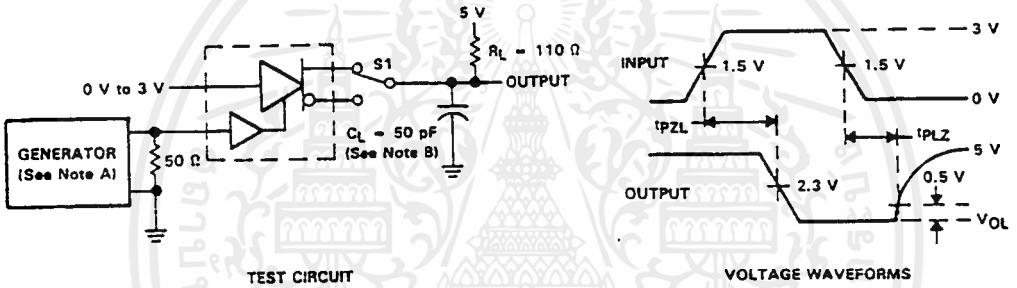
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT VOLTAGE WAVEFORMS

FIGURE 3. t_{PZH} AND t_{PHZ}



TEST CIRCUIT VOLTAGE WAVEFORMS

FIGURE 4. t_{PZL} AND t_{PLZ}

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR ≤ 1 MHz, duty cycle = 50%, t_r ≤ 5 ns, t_f ≤ 5 ns, Z_O = 50 Ω.
B. C_L includes probe and stray capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

HIGH-LEVEL OUTPUT VOLTAGE
vs
HIGH-LEVEL OUTPUT CURRENT

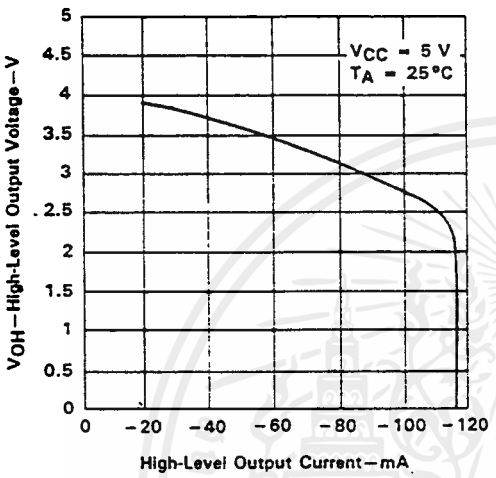


FIGURE 5

LOW-LEVEL OUTPUT VOLTAGE
vs
LOW-LEVEL OUTPUT CURRENT

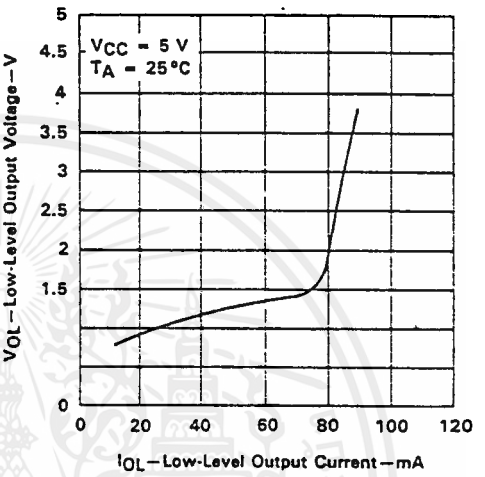


FIGURE 6

DIFFERENTIAL OUTPUT VOLTAGE
vs
OUTPUT CURRENT

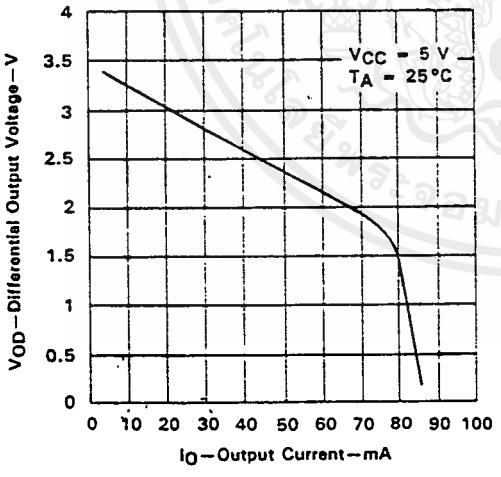


FIGURE 7

OUTPUT CURRENT
vs
OUTPUT VOLTAGE

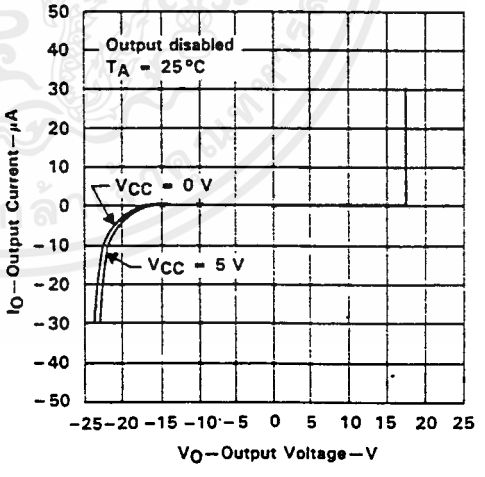


FIGURE 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

SUPPLY CURRENT
vs
SUPPLY VOLTAGE

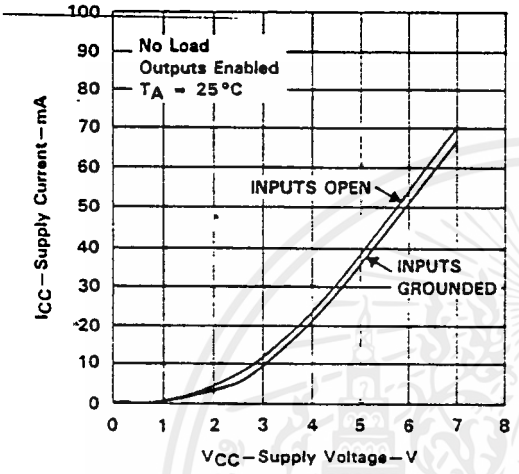


FIGURE 9

SUPPLY CURRENT
vs
SUPPLY VOLTAGE

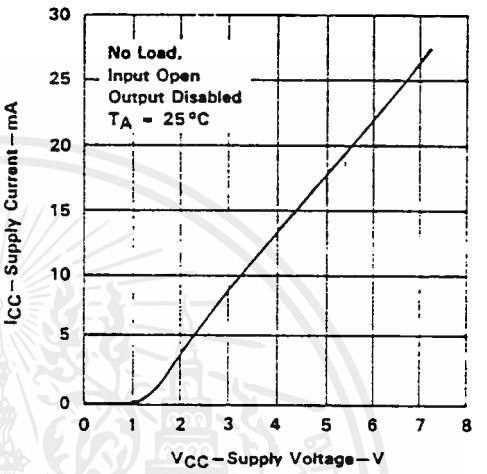
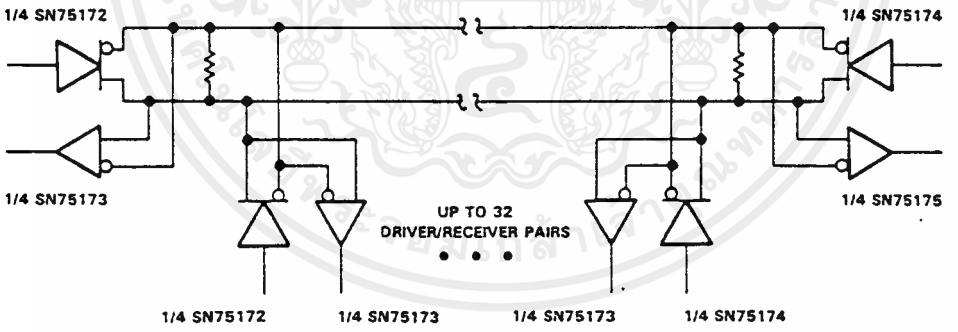


FIGURE 10

TYPICAL APPLICATION



NOTE: The line length should be terminated at both ends in its characteristic impedance. Stub lengths off the main line should be kept as short as possible.

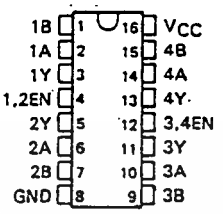
FIGURE 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D2602, OCTOBER 1980—REVISED SEPTEMBER 1986

- Meets EIA Standards RS-422-A, RS-423-A, and RS-485
- Meets CCITT Recommendations V.10, V.11, X.26, and X.27
- Designed for Multipoint Bus Transmission on Long Bus Lines in Noisy Environments
- 3-State Outputs
- Common-Mode Input Voltage Range −12 V to 12 V
- Input Sensitivity . . . ±200 mV
- Input Hysteresis . . . 50 mV Typ
- High Input Impedance . . . 12 kΩ Min
- Operates from Single 5-Volt Supply
- Low Power Requirements
- Plug-in Replacement for MC3486

. D, J, OR N
DUAL-IN-LINE PACKAGE
(TOP VIEW)



description

The SN75175 is a monolithic quadruple differential line receiver with three-state outputs. It is designed to meet the requirements of EIA Standards RS-422-A, RS-423-A, and RS-485 and several CCITT recommendations. The device is optimized for balanced multipoint bus transmission at rates up to 10 megabits per second. Each of the two pairs of receivers has a common active-high enable.

The receivers feature high input impedance, input hysteresis for increased noise immunity, and input sensitivity of ±200 millivolts over a common-mode input voltage range of ±12 volts. The SN75175 is designed for optimum performance when used with the SN75172 or SN75174 quadruple differential line drivers.

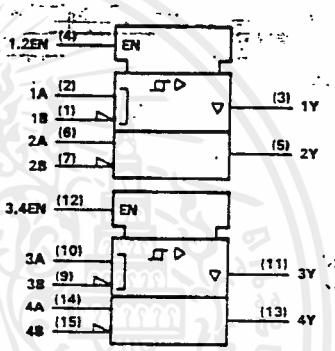
The SN75175 is characterized for operation from 0°C to 70°C.

FUNCTION TABLE (EACH RECEIVER)

DIFFERENTIAL INPUTS A — B	ENABLE	OUTPUT Y
$V_{ID} \geq 0.2$ V	H	H
-0.2 V $< V_{ID} <$ 0.2 V	H	?
$V_{ID} \geq -0.2$ V	H	L
X	L	Z

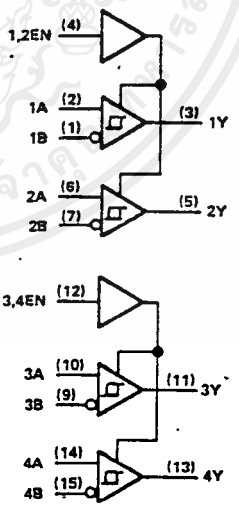
H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

logic symbol†



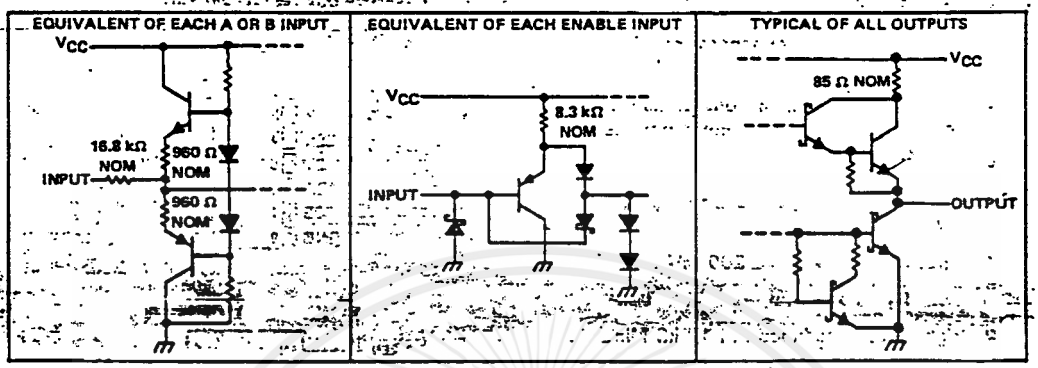
† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematics of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage, A or B inputs	±25 V
Differential input voltage (see Note 2)	±25 V
Enable input voltage	7 V
Low-level output current	50 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 3):	
D package	950 mW
J package	1025 mW
N package	1150 mW
Operating free-air temperature range	0°C to 70°C
Storage temperature range	−65°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 60 seconds: J package	−300°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds: D or N package	−260°C

NOTES: 1. All voltage values, except differential input voltage, are with respect to network ground terminal.
2. Differential input voltage is measured at the noninverting input with respect to the corresponding inverting input.
3. For operation above 25°C free-air temperature, derate the D package to 608 mW at 70°C at the rate of 7.6 mW/°C, the J package to 656 mW at 70°C at the rate of 8.2 mW/°C, and the N package to 736 mW at 70°C at the rate of 9.2 mW/°C. In the J package, SN75175 chips are glass mounted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
Common-mode input voltage, V_{IC}			±12	V
Differential input voltage, V_{ID}			±12	V
High-level enable input voltage, V_{IH}	2			V
Low-level enable input voltage, V_{IL}			0.8	V
High-level output current, I_{OH}			−400	μA
Low-level output current, I_{OL}			16	mA
Operating free-air temperature, T_A	0		70	°C

electrical characteristics over recommended ranges of common-mode input voltage, supply voltage, and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{TH}	Differential-input high-threshold voltage	V_O = 2.7 V, I_O = −0.4 mA				0.2	V
V_{TL}	Differential-input low-threshold voltage	V_O = 0.5 V, I_O = 16 mA		−0.2‡			V
V_{hys}	Hysteresis§				50		mV
V_{IK}	Enable-input clamp voltage	I_I = −18 mA				−1.5	V
V_{OH}	High-level output voltage	V_{ID} = 200 mV, I_{OH} = −400 μA, See Figure 1		2.7			V
V_{OL}	Low-level output voltage	V_{ID} = −200 mV, See Figure 1	I_{OL} = 8 mA			0.45	V
			I_{OL} = 16 mA			0.5	
I_{OZ}	High-impedance-state output current	V_O = 0.4 V to 2.4 V				±20	μA
I_I	Line input current	Other input at 0 V, See Note 4	V_I = 12 V			1	mA
			V_I = −7 V			−0.8	
I_{IH}	High-level enable-input current	V_{IH} = 2.7 V				20	μA
I_{IL}	Low-level enable-input current	V_{IL} = 0.4 V				−100	μA
r_i	Input resistance			12			kΩ
I_{OS}	Short-circuit output current¶			−15		−85	mA
I_{CC}	Supply current	Outputs disabled				70	mA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the less positive (more negative) limit is designated as minimum, is used in this data sheet for threshold voltage levels only.
§ Hysteresis is the difference between the positive-going input threshold voltage, V_{T+}, and the negative-going input threshold voltage, V_{T-}. See Figure 4.
¶ Not more than one output should be shorted at a time and the duration of the short-circuit should not exceed one second.
NOTE 4: Refer to EIA standards RS-422-A, RS-423-A, and RS-485 for exact conditions.

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

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 15 pF, See Figure 2		22	35	ns
t_{PHL}	Propagation delay time, high-to-low-level output			25	35	ns
t_{PZH}	Output enable time to high level	C_L = 15 pF, See Figure 3		13	30	ns
t_{PZL}	Output enable time to low level			19	30	ns
t_{PHZ}	Output disable time from high level	C_L = 15 pF, See Figure 3		26	35	ns
t_{PLZ}	Output disable time from low level			25	35	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

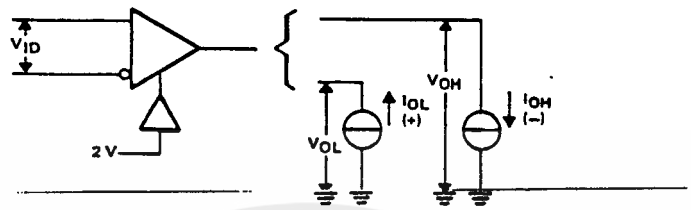


FIGURE 1. V_{OH}, V_{OL}

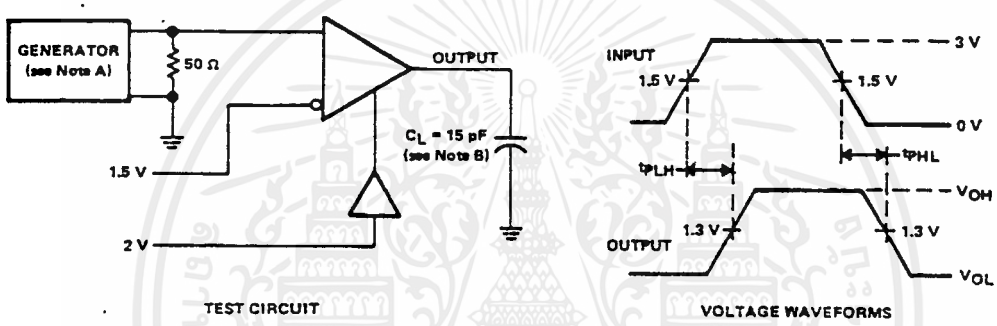


FIGURE 2. PROPAGATION DELAY TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR ≤ 1 MHz, duty cycle = 50%, t_r ≤ 6 ns, t_f ≤ 6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and stray capacitance.

4 Line Drivers/Receivers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

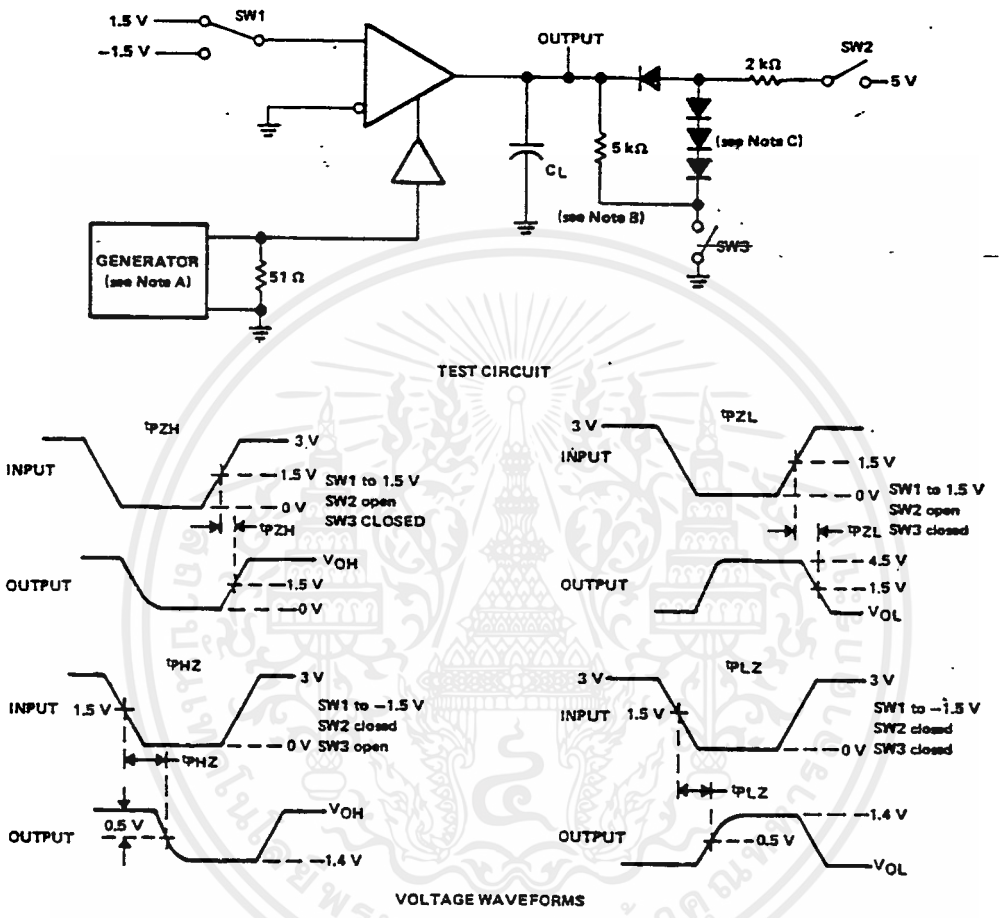


FIGURE 3. ENABLE AND DISABLE TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR ≤ 1 MHz, duty cycle = 50%, t_f ≤ 6 ns, t_r ≤ 6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and stray capacitance.
C. All diodes are 1N916 or equivalent.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

OUTPUT VOLTAGE
vs
DIFFERENTIAL INPUT VOLTAGE

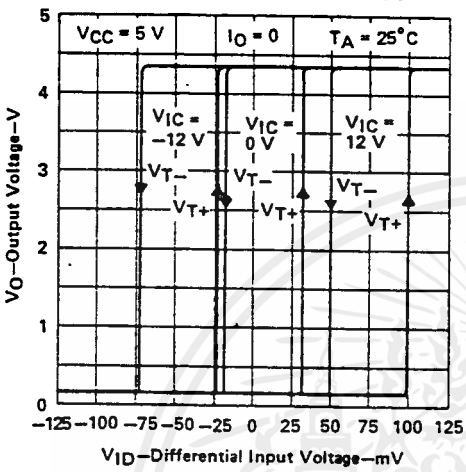


FIGURE 4

HIGH-LEVEL OUTPUT VOLTAGE
vs
HIGH-LEVEL OUTPUT CURRENT

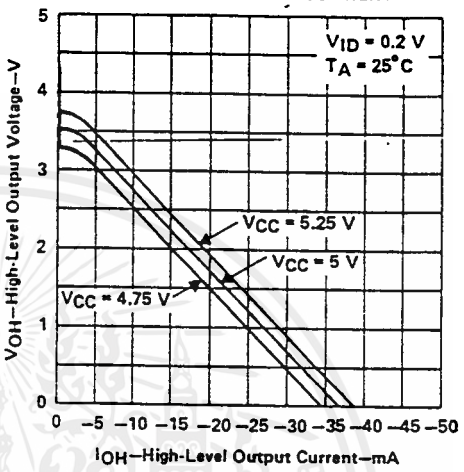


FIGURE 5

HIGH-LEVEL OUTPUT VOLTAGE
vs
FREE-AIR TEMPERATURE

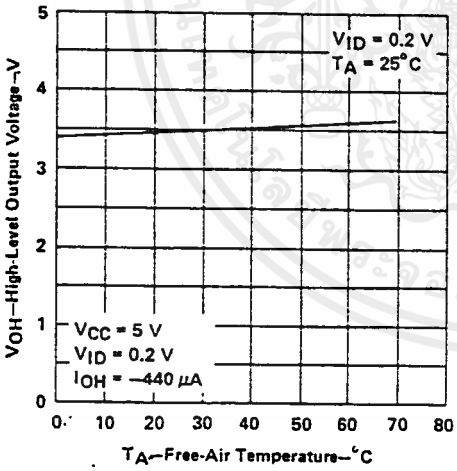


FIGURE 6

LOW-LEVEL OUTPUT VOLTAGE
vs
LOW-LEVEL OUTPUT CURRENT

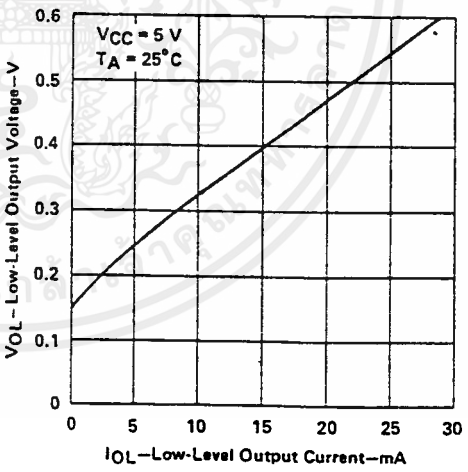


FIGURE 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

LOW-LEVEL OUTPUT VOLTAGE
vs
FREE-AIR TEMPERATURE

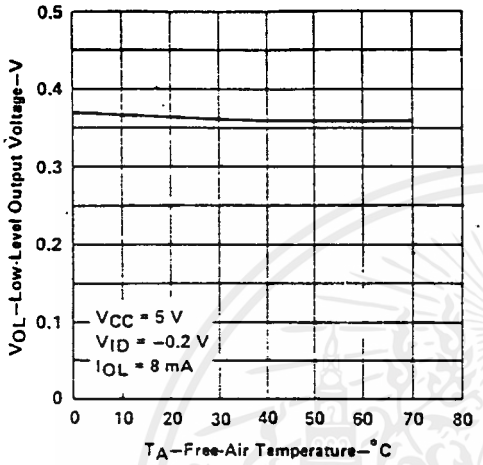


FIGURE 8

OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

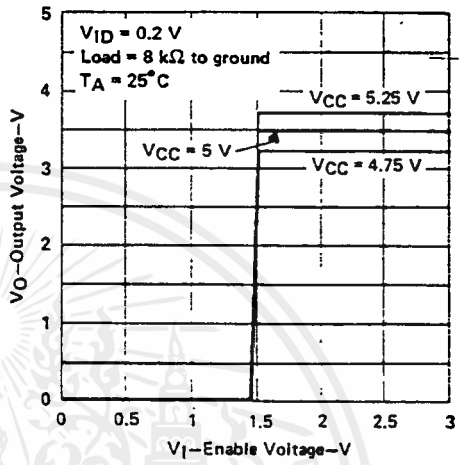


FIGURE 9

OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

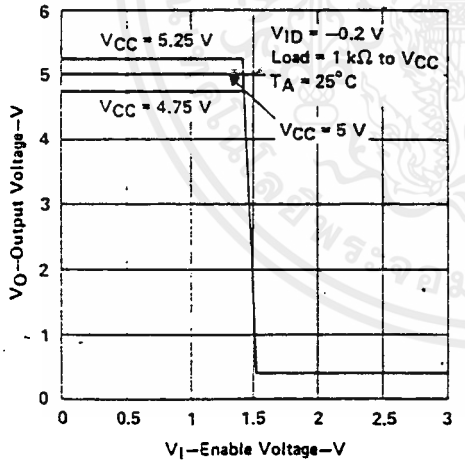


FIGURE 10

SUPPLY CURRENT
(ALL RECEIVERS)
vs
SUPPLY VOLTAGE

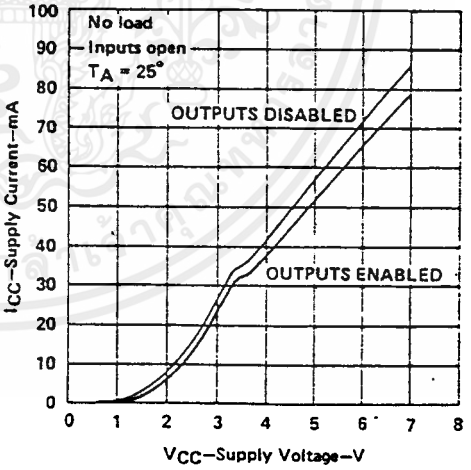


FIGURE 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

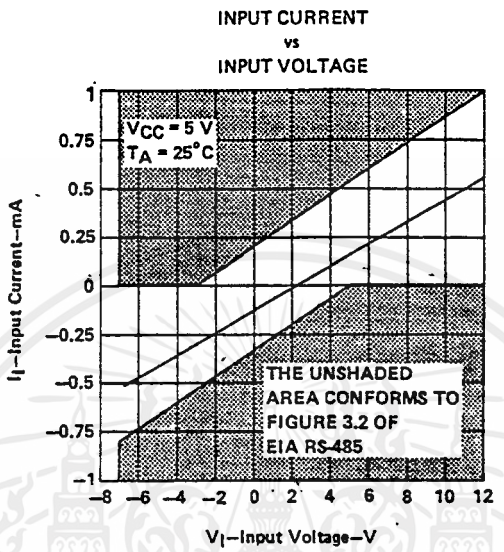


FIGURE 12

TYPICAL APPLICATION

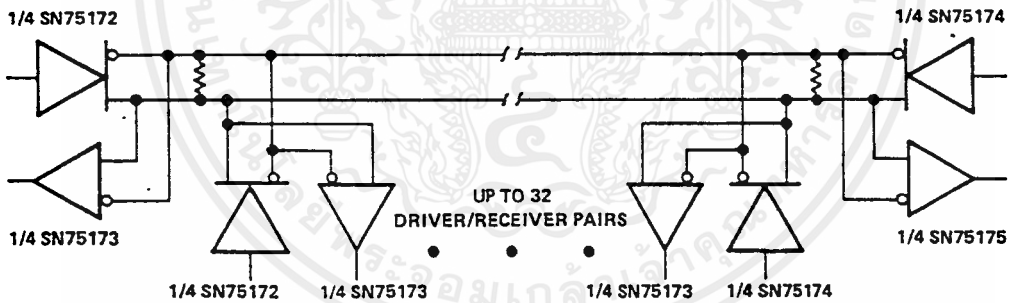


FIGURE 13

NOTE: The line should be terminated at both ends in its characteristic impedance. Stub lengths off the main line should be kept as short as possible.

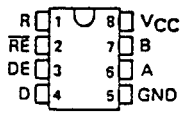
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN65176B, SN75176B
DIFFERENTIAL BUS TRANSCEIVERS

D2619, JULY 1985—REVISED OCTOBER 1988

- Bidirectional Transceiver
- Meets EIA Standards RS-422-A and RS-485 and CCITT Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability. . . ±60 mA Max
- Thermal Shutdown Protection
- Driver Positive and Negative Current Limiting
- Receiver Input Impedance . . . 12 kΩ Min
- Receiver Input Sensitivity . . . ±200 mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operates from Single 5-Volt Supply
- Low Power Requirements

D, JG, OR P PACKAGE
(TOP VIEW)



FUNCTION TABLE (DRIVER)

INPUT	ENABLE	OUTPUTS	
D	DE	A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

FUNCTION TABLE (RECEIVER)

DIFFERENTIAL INPUTS	ENABLE	OUTPUT
A − B	$\overline{RE}$	R
$V_{ID} \geq 0.2$ V	L	H
-0.2 V $< V_{ID} <$ 0.2 V	L	?
$V_{ID} \leq -0.2$ V	L	L
X	H	Z

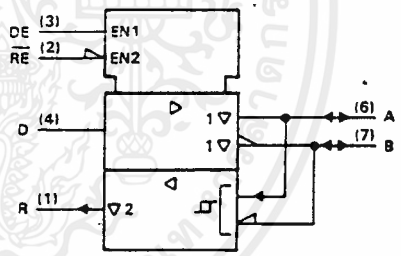
H = high level, L = low level, ? = indeterminate.
X = irrelevant, Z = high impedance (off)

description

The SN65176B and SN75176B differential bus transceivers are monolithic integrated circuits designed for bidirectional data communication on multipoint bus transmission lines. They are designed for balanced transmission lines and meet EIA Standard RS-422-A and RS-485 and CCITT Recommendations V.11 and X.27.

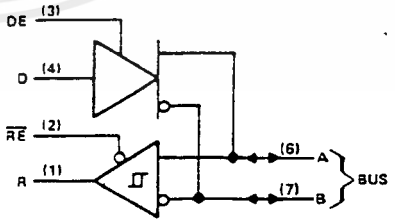
The SN65176B and SN75176B combine a three-state differential line driver and a differential input line receiver both of which operate from a single 5-volt power supply. The driver and receiver have active-high and active-low enables, respectively, that can be externally connected together to function as a direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus whenever the driver is disabled or V_{CC} = 0 volts. These ports feature wide positive and negative common-mode voltage ranges making the device suitable for party-line applications.

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram (positive logic)



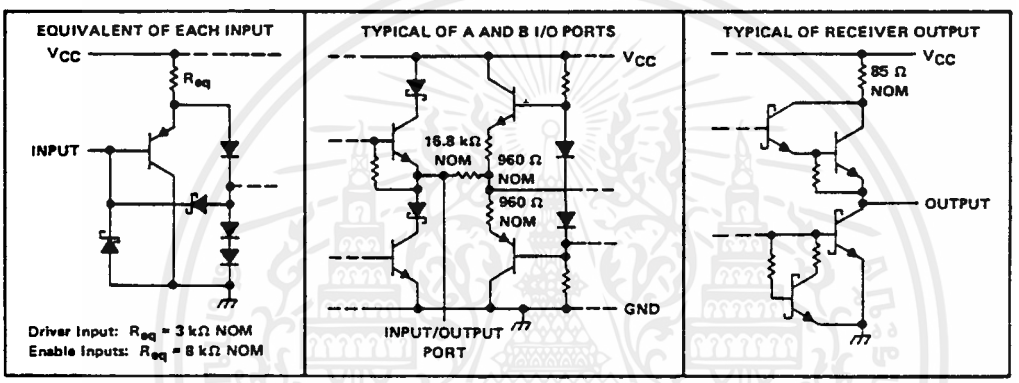
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The driver is designed to handle loads up to 60 milliamperes of sink or source current. The driver features positive- and negative-current limiting and thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 kΩ, an input sensitivity of ±200 millivolts, and a typical input hysteresis of 50 millivolts.

The SN65176B and SN75176B can be used in transmission line applications employing the SN75172 and SN75174 quadruple differential line drivers and SN75173 and SN75175 quadruple differential line receivers.

The SN65176B is characterized for operation from −40°C to 85°C and the SN75176B is characterized for operation from 0°C to 70°C.

schematics of inputs and outputs



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Voltage at any bus terminal . . . . . −10 V to 15 V
Enable input voltage . . . . . 5.5 V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2):
D package . . . . . 725 mW
JG package . . . . . 825 mW
P package . . . . . 1100 mW
Operating free-air temperature range: SN65176B . . . . . −40°C to 85°C
SN75176B . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from the case for 60 seconds: JG package . . . . . 300°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: D or P package . . . . . 260°C

NOTES: 1. All voltage values, except differential input/output bus voltage, are with respect to network ground terminal.
2. For operation above 25°C free-air temperature, refer to Dissipation Derating Table. In the JG package, the chips are glass mounted.

DISSIPATION DERATING TABLE

PACKAGE	T_A = 25°C POWER RATING	DERATING FACTOR ABOVE T_A = 25°C	T_A = 70°C POWER RATING	T_A = 85°C POWER RATING
D	725 mW	5.8 mW/°C	464 mW	377 mW
JG	825 mW	6.6 mW/°C	528 mW	429 mW
P	1100 mW	8.8 mW/°C	702 mW	570 mW

recommended operating conditions

			MIN	TYP	MAX	UNIT
Supply voltage, V_{CC}			4.75	5	5.25	V
Voltage at any bus terminal (separately or common mode), V_I or V_{IC}					12	V
					−7	
High-level input voltage, V_{IH}		D, DE, and $\overline{RE}$	2			V
Low-level input voltage, V_{IL}		D, DE, and $\overline{RE}$			0.8	V
Differential input voltage, V_{ID} (see Note 3)					±12	V
High-level output current, I_{OH}		Driver			−60	mA
		Receiver			−400	μA
Low-level output current, I_{OL}		Driver			60	mA
		Receiver			8	
Operating free-air temperature, T_A	SN651768		−40		85	°C
	SN751768		0		70	

NOTE 3: Differential-input/output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIVER SECTION

driver electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IK}	Input clamp voltage	I_I = −18 mA				−1.5	V
V_O	Output voltage	I_O = 0		0		6	V
$\lvert V_{OD1}\rvert$	Differential output voltage	I_O = 0		1.5		6	V
$\lvert V_{OD2}\rvert$	Differential output voltage	R_L = 100 Ω,	See Figure 1	½ V_{OD1}			
				2			V
		R_L = 54 Ω,	See Figure 1	1.5	2.5	5	V
V_{OD3}	Differential output voltage	See Note 4		1.5		5	V
$\Delta\lvert V_{OD}\rvert$	Change in magnitude of differential output voltage§	R_L = 54 Ω or 100 Ω,	See Figure 1			±0.2	V
V_{OC}	Common-mode output voltage					+3 −1	V
$\Delta\lvert V_{OC}\rvert$	Change in magnitude of common-mode output voltage§					±0.2	V
I_O	Output current	Output disabled, See Note 5	V_O = 12 V			1	mA
			V_O = −7 V			−0.8	
I_{IH}	High-level input current	V_I = 2.4 V				20	μA
I_{IL}	Low-level input current	V_I = 0.4 V				−400	μA
I_{OS}	Short-circuit output current	V_O = −7 V				−250	mA
		V_O = 0				−150	
		V_O = V_{CC}				250	
		V_O = 12 V				250	
I_{CC}	Supply current (total package)	No load	Outputs enabled		42	55	mA
			Outputs disabled		26	35	

† The power-off measurement in EIA Standard RS-422-A applies to disabled outputs only and is not applied to combined inputs and outputs.
‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.
§ $\Delta\lvert V_{OD}\rvert$ and $\Delta\lvert V_{OC}\rvert$ are the changes in magnitude of V_{OD} and V_{OC} respectively, that occur when the input is changed from a high level to a low level.

NOTES: 4. See EIA Standard RS-485 Figure 3.5, Test Termination Measurement 2.
5. This applies for both power on and off; refer to EIA Standard RS-485 for exact conditions. The RS-422-A limit does not apply for a combined driver and receiver terminal.

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

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{DD}	Differential-output delay time	R_L = 54 Ω,	See Figure 3		15	22	ns
t_{TD}	Differential-output transition time				20	30	ns
t_{PZH}	Output enable time to high level	R_L = 110 Ω,	See Figure 4		85	120	ns
t_{PZL}	Output enable time to low level	R_L = 110 Ω,	See Figure 5		40	60	ns
t_{PHZ}	Output disable time from high level	R_L = 110 Ω,	See Figure 4		150	250	ns
t_{PLZ}	Output disable time from low level	R_L = 110 Ω,	See Figure 5		20	30	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL EQUIVALENTS

DATA SHEET PARAMETER	RS-422-A	RS-485
V_O	V_{oa}, V_{ob}	V_{oa}, V_{ob}
$\lvert V_{OD1}\rvert$	V_o	V_o
$\lvert V_{OD2}\rvert$	V_t (R_L = 100 Ω)	V_t (R_L = 54 Ω)
$\lvert V_{OD3}\rvert$		V_t (Test Termination Measurement 2)
$\Delta\lvert V_{OD}\rvert$	$\lvert\lvert V_t\rvert - \lvert\overline{V}_t\rvert\rvert$	$\lvert\lvert V_t\rvert - \lvert\overline{V}_t\rvert\rvert$
V_{OC}	$\lvert V_{os}\rvert$	$\lvert V_{os}\rvert$
$\Delta\lvert V_{OC}\rvert$	$\lvert V_{os} - \overline{V}_{os}\rvert$	$\lvert V_{os} - \overline{V}_{os}\rvert$
I_{OS}	$\lvert I_{sa}\rvert$, $\lvert I_{sb}\rvert$	
I_O	$\lvert I_{xa}\rvert$, $\lvert I_{xb}\rvert$	I_{ia}, I_{ib}

RECEIVER SECTION

receiver electrical characteristics over recommended ranges of common-mode input voltage, supply voltage, and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{TH}	Differential-input high-threshold voltage	V_O = 2.7 V, I_O = −0.4 mA				0.2	V
V_{TL}	Differential-input low-threshold voltage	V_O = 0.5 V, I_O = 8 mA		−0.2‡			V
V_{hys}	Hysteresis§				50		mV
V_{IK}	Enable-input clamp voltage	I_I = −18 mA				−1.5	V
V_{OH}	High-level output voltage	V_{ID} = −200 mV, I_{OH} = −400 μA, See Figure 2		2.7			V
V_{OL}	Low-level output voltage	V_{ID} = −200 mV, I_{OL} = 8 mA, See Figure 2				0.45	V
I_{OZ}	High-impedance-state output current	V_O = 0.4 V to 2.4 V				±20	μA
I_I	Line input current	Other input = 0 V, See Note 6	V_I = 12 V			1	mA
			V_I = −7 V			−0.8	
I_{IH}	High-level enable-input current	V_{IH} = 2.7 V				20	μA
I_{IL}	Low-level enable-input current	V_{IL} = 0.4 V				−100	μA
r_I	Input resistance				12		kΩ
I_{OS}	Short-circuit output current				−15	−85	mA
I_{CC}	Supply current (total package)	No load	Outputs enabled		42	55	mA
			Outputs disabled		26	35	

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the less-positive (more-negative) limit is designated minimum, is used in this data sheet for common-mode input voltage and threshold voltage levels only.
§ Hysteresis is the difference between the positive-going input threshold voltage, V_{T+}, and the negative-going input threshold voltage, V_{T-}. See Figure 4.
NOTE 6: This applies for both power on and power off. Refer to EIA Standard RS-485 for exact conditions.

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

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	V_{ID} = 0 V to 3 V, C_L = 15 pF, See Figure 6		21	35	ns
t_{PHL}	Propagation delay time, high-to-low-level output			23	35	ns
t_{PZH}	Output enable time to high level	C_L = 15 pF, See Figure 7		10	20	ns
t_{PZL}	Output enable time to low level			12	20	ns
t_{PHZ}	Output disable time from high level	C_L = 15 pF, See Figure 7		20	35	ns
t_{PLZ}	Output disable time from low level			17	25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

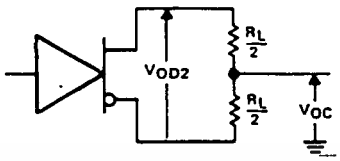


FIGURE 1. DRIVER V_{OD} AND V_{OC}

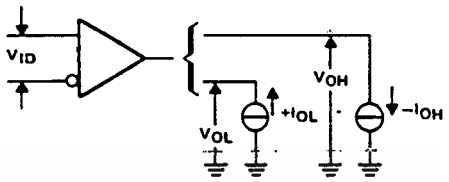


FIGURE 2. RECEIVER V_{OH} AND V_{OL}

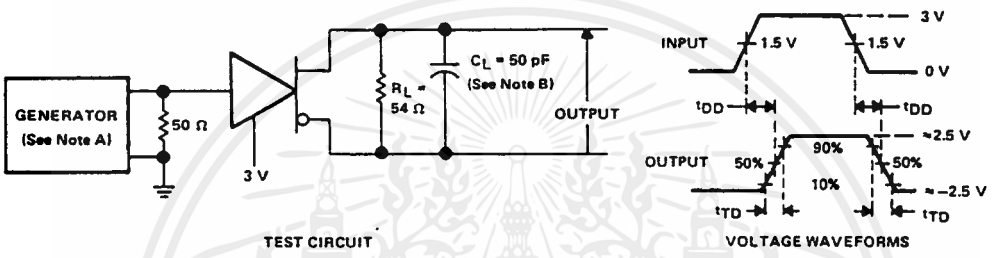


FIGURE 3. DRIVER DIFFERENTIAL-OUTPUT DELAY AND TRANSITION TIMES

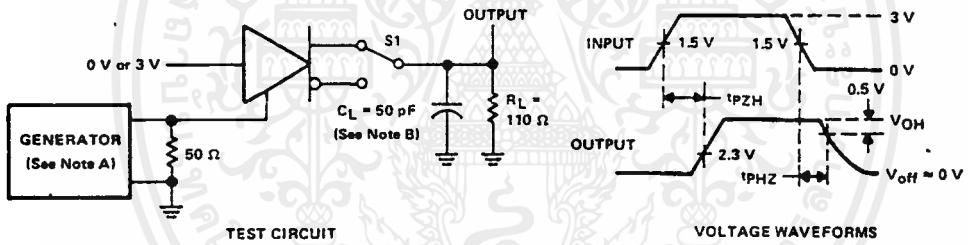


FIGURE 4. DRIVER ENABLE AND DISABLE TIMES

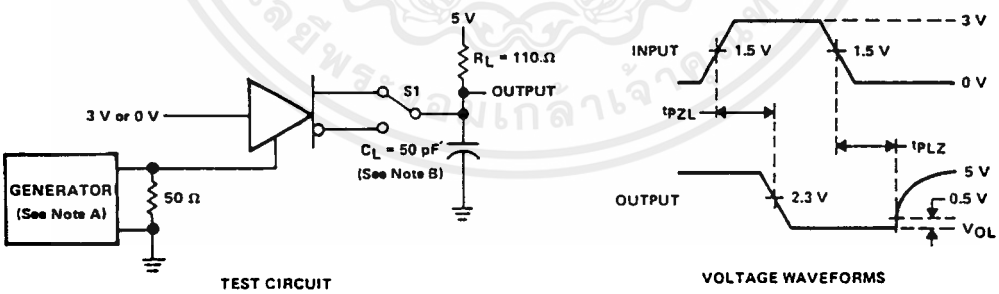


FIGURE 5. DRIVER ENABLE AND DISABLE TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR ≤ 1 MHz, 50% duty cycle, t_r ≤ 6 ns, t_f ≤ 6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and jig capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

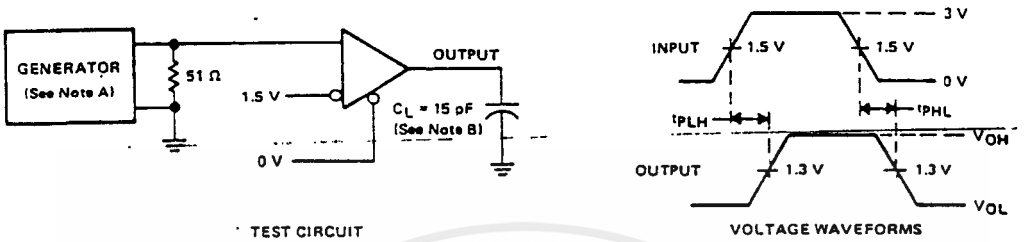


FIGURE 6. RECEIVER PROPAGATION DELAY TIMES

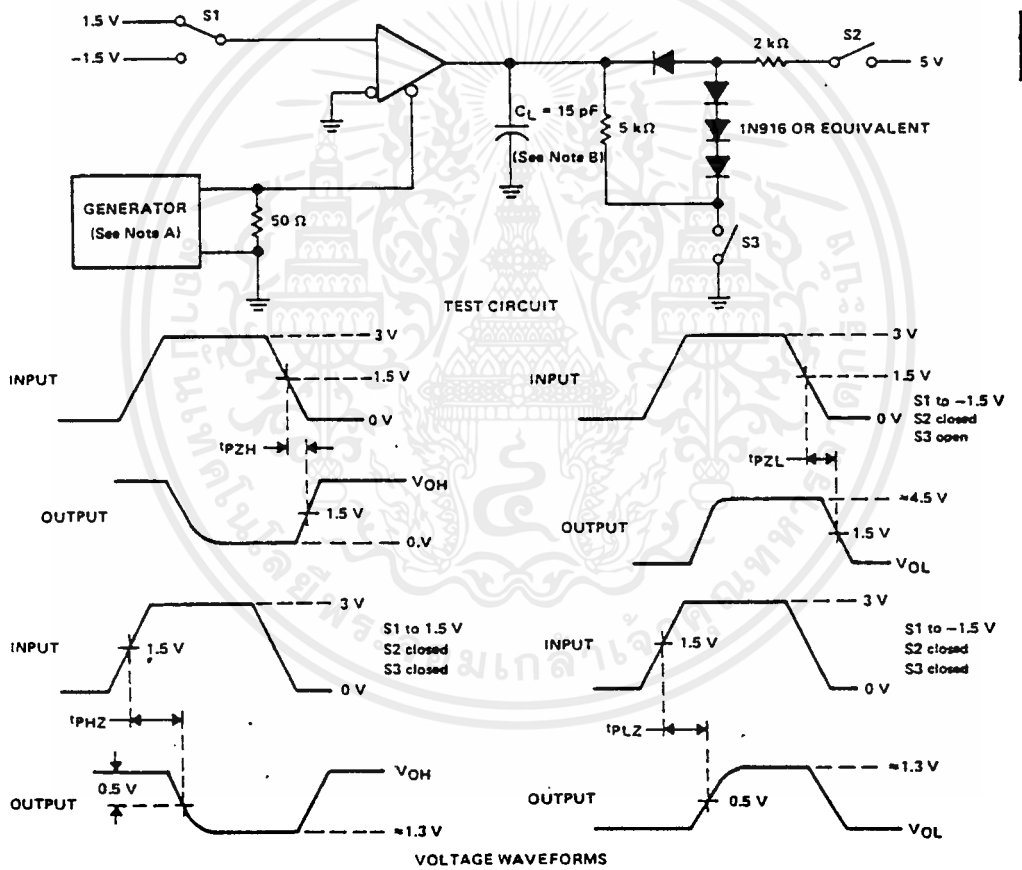


FIGURE 7. RECEIVER OUTPUT ENABLE AND DISABLE TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR ≤ 1 MHz, 50% duty cycle, t_r ≤ 6 ns, t_f ≤ 6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and jig capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

DRIVER HIGH-LEVEL OUTPUT VOLTAGE
vs
DRIVER HIGH-LEVEL OUTPUT CURRENT

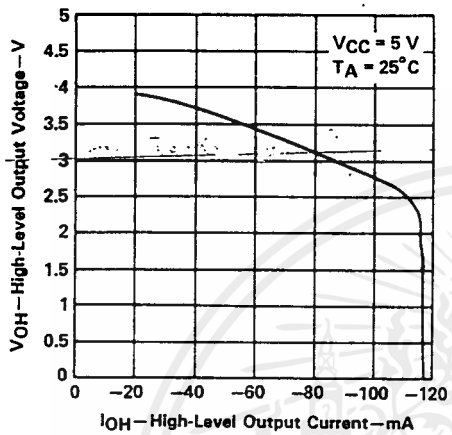


FIGURE 8

DRIVER LOW-LEVEL OUTPUT VOLTAGE
vs
DRIVER LOW-LEVEL OUTPUT CURRENT

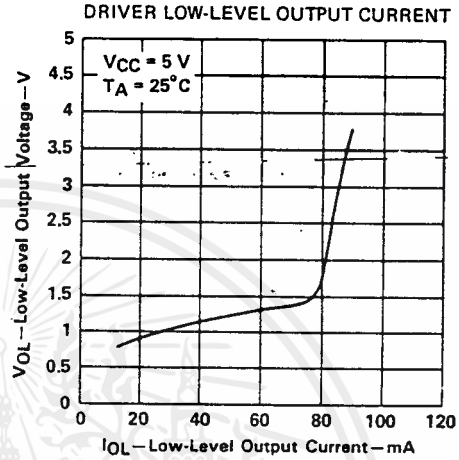


FIGURE 9

DRIVER DIFFERENTIAL OUTPUT VOLTAGE
vs
DRIVER OUTPUT CURRENT

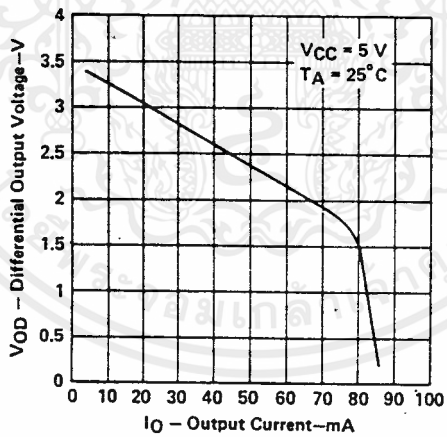


FIGURE 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

RECEIVER HIGH-LEVEL OUTPUT VOLTAGE
vs
HIGH-LEVEL OUTPUT CURRENT

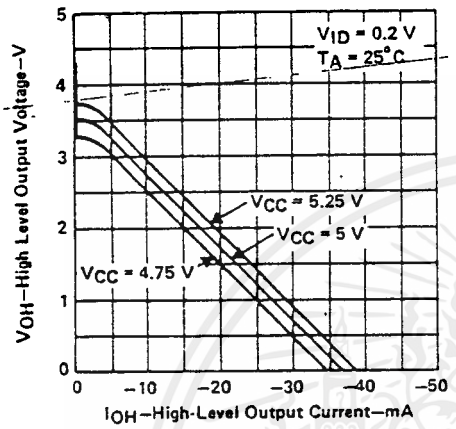


FIGURE 11

RECEIVER HIGH-LEVEL OUTPUT
vs
FREE-AIR TEMPERATURE

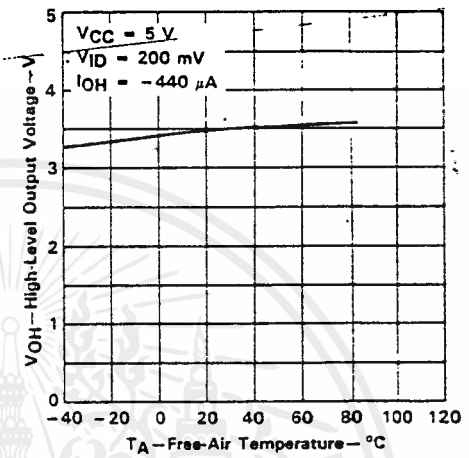


FIGURE 12

RECEIVER LOW-LEVEL OUTPUT VOLTAGE
vs
RECEIVER LOW-LEVEL OUTPUT CURRENT

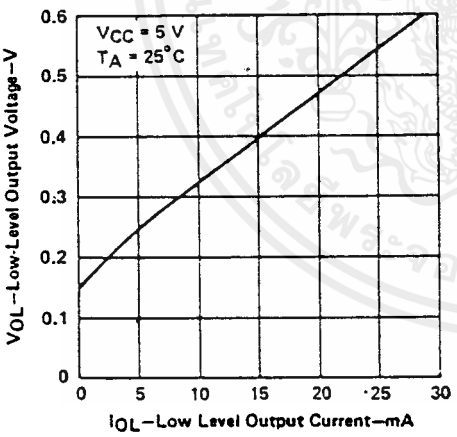


FIGURE 13

RECEIVER LOW-LEVEL OUTPUT VOLTAGE
vs
FREE-AIR TEMPERATURE

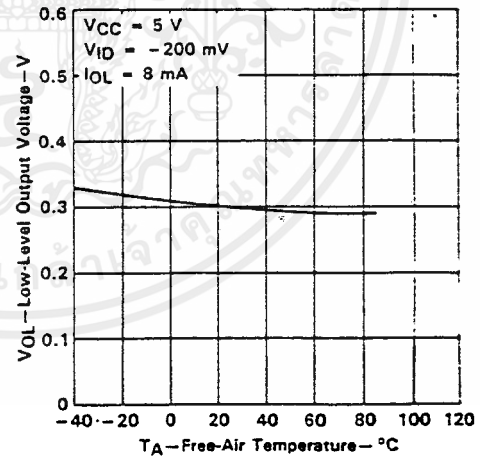


FIGURE 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

RECEIVER OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

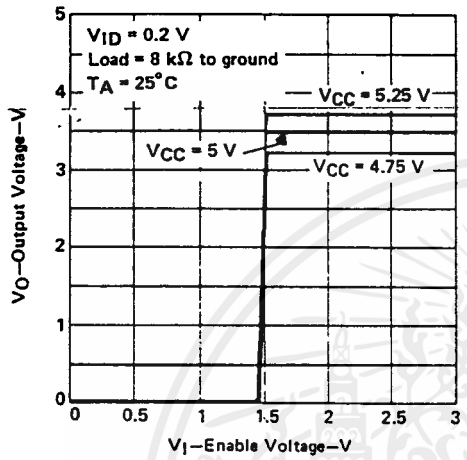


FIGURE 15

RECEIVER OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

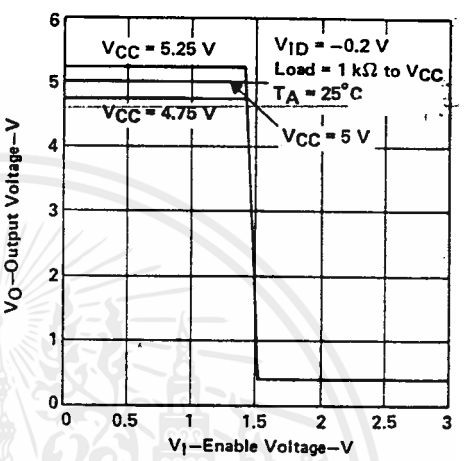


FIGURE 16

TYPICAL APPLICATION

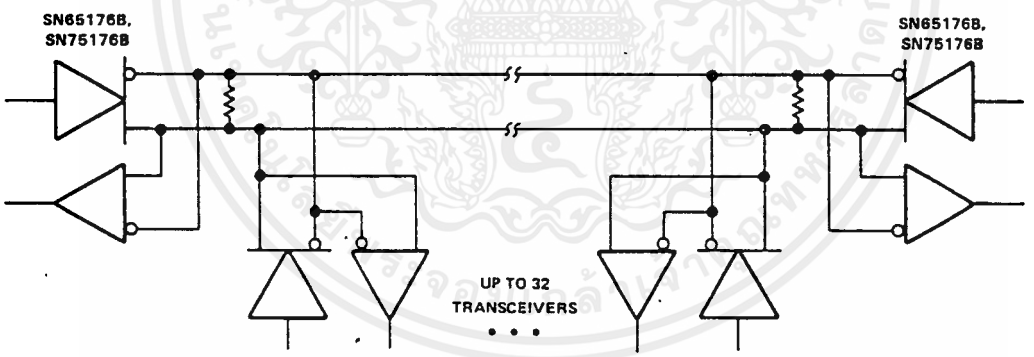


FIGURE 17. TYPICAL APPLICATION CIRCUIT

NOTE 7: The line should be terminated at both ends in its characteristic impedance. Stub lengths off the main line should be kept as short as possible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ชูชัย ธนสารสังเจริญ, ดนัย แสงสุวิชศิลป์, ทินกร ดุ๊ก, ธงชัย อุดมโภคล และธานินทร์ ถาวรศาสนวงศ์ "การใช้งาน Z80" กรุงเทพมหานคร : พิสิทส์เซ็นเตอร์การพิมพ์, 284 หน้า

[2] ชูชัย ธนสารสังเจริญ, ทินกร ดุ๊ก "การสื่อสารข้อมูล" กรุงเทพมหานคร : พิสิทส์เซ็นเตอร์การพิมพ์

[3] ธานินทร์ ถาวรศาสนวงศ์, ทินกร ดุ๊ก "การอินเตอร์เฟส" กรุงเทพมหานคร : พิสิทส์เซ็นเตอร์การพิมพ์

[4] แรงค์ เวศนาวัฒน์, สมพัตร์ เบ็ญจชัยพร และพิชิต รัชชลี "การพัฒนาเครื่องรับสัญญาณสื่อสารข้อมูล" กรุงเทพมหานคร : ศูนย์วิจัยและพัฒนางานคอมพิวเตอร์ KMIT รายงานการวิจัย 2531

[5] เอกชัย สันกำแพง "วงจรแปลง RS-232 เป็น RS-485" เซมิคอนดัคเตอร์, ฉบับที่ 115, มี.ค.-เม.ย., 2535, หน้า (132-136)

[6] ELECTRONIC INDUSTRIES ASSOCIATION : " EIA STANDARD RS-485", APRIL 1983

[7] RONALD L. KRUTZ : "INTERFACING TECHNIQUES IN DIGITAL DESIGN WITH EMPHASIS ON MICROPROCESSORS", JOHN WILEY & SONS, 1988, pp.382

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวิเคราะห์สัญญาณลอจิก
(LOGIC ANALYZER)

โดย
นายสรพงษ์ โชติพานิช
นายสัญญา ยิ้มศิริ

ปริญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต
สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องวิเคราะห์สัญญาณลอจิก
(LOGIC ANALYZER)

ผู้จัดทำ
นายสรพงษ์ โชติพานิช ระหัส 321361
นายสัญญา ยิ้มศิริ ระหัส 321366

(อาจารย์วิทยา ทิพย์สุวรรณพร)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวิเคราะห์สัญญาณลอจิก

นายสรพงษ์ โชติพานิช 321361
นายสัญญา ยิ้มศิริ 321366
อาจารย์วิทยา ทิพย์สุวรรณพร อาจารย์ที่ปรึกษา

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาและออกแบบการ์ดวิเคราะห์สัญญาณดิจิตอลที่ทำงานร่วมกับเครื่องไมโครคอมพิวเตอร์ของ IBM ให้สามารถใช้วิเคราะห์และตรวจสอบสัญญาณในวงจรดิจิตอลได้ 16 ช่อง ที่ความถี่ในการสุ่มของสัญญาณนาฬิกาภายในเป็น 20 MHz และความถี่ของสัญญาณจากภายนอก 40MHz โดยอาศัยหลักการแซมปลิง (Sampling) ข้อมูลด้วยความเร็วสูง และสามารถเปลี่ยนความถี่ในการแซมปลิงได้ตั้งแต่ 500 Hz จนถึง 20 MHz สำหรับสัญญาณนาฬิกาภายใน ข้อมูลที่ได้จะถูกนำไปเก็บในหน่วนความจำความเร็วสูง และจะถูกนำไปแสดงผลในรูปแบบต่างๆ ทางจอภาพหรือเครื่องพิมพ์ของไมโครคอมพิวเตอร์และสามารถเก็บข้อมูลไว้ใช้ภายหลังได้ โดยที่การทำงานทั้งหมด จะถูกควบคุมผ่านทางไมโครคอมพิวเตอร์และโปรแกรมควบคุมการทำงานที่เขียนด้วยภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOGIC ANALYZER

Mr. Sorapong Chotipanich 321361
Mr. Sanya Yimsiri 321366
Mr. Vittaya Thipsuwanporn Advisor

Abstract

This project is a development and design of Logic Analyzer Card add on IBM PC whose ability is analyzing and checking the digital signal . It can detect 16 channel of input simultaneously by 20 MHz of internal sampling clock and 40 MHz of external clock. Sampled datas are stored in the high speed memory, which showed on microcomputer display or printer and also can be save in the storage device. The function of the Logic Analyzer Card is control via microcomputer and user interface of controller programme which written by C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	หลักการเบื้องต้นของลอจิกอนาไลเซอร์	5
บทที่ 3	การออกแบบฮาร์ดแวร์ของลอจิกอนาไลเซอร์	16
บทที่ 4	การใช้งานจอภาพ Super VGA	45
บทที่ 5	โปรแกรมควบคุมการทำงาน	51
บทที่ 6	การใช้งานลอจิกอนาไลเซอร์	79
บทที่ 7	สรุปผลและวิจารณ์	87
บทที่ 8	ภาคผนวก	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ชนิดของความผิดพลาดในวงจรดิจิตอล

ความผิดพลาดที่เกิดในวงจรดิจิตอล อาจจะเป็นผลมาจากทางฮาร์ดแวร์ (Hard Ware) หรือทางซอฟแวร์ (Software) หรือทั้งสองอย่าง ซึ่งความผิดพลาดทางซอฟแวร์จะเกิดขึ้นนั้นอาจจะเกิดขึ้นมาจากการผิดพลาดของโปรแกรมการทำงาน ส่วนความผิดพลาดทางฮาร์ดแวร์นั้นอาจจะเกิดมาจาก

1. ความผิดพลาดของข้อมูล (No data or wrong data)
2. กลิทช์ (Glitches)
3. สไปค์ (Spikes)
4. เรส (Races)
5. การผิดพลาดในคาบเวลา (Timing Error)
6. ริงกิ้ง (Ringing)
7. ระดับของแรงดันผิดพลาด (Wrong Level)

ความผิดพลาดทางฮาร์ดแวร์ (Hardware Faults)

ดังที่ได้กล่าวมาแล้วว่าความผิดพลาดทางฮาร์ดแวร์จะมีอยู่ 7 ประเภท และมีรายละเอียดดังต่อไปนี้

1. ความผิดพลาดทางข้อมูล อาจเกิดจากการที่สายส่งสัญญาณในวงจรเกิดการเสียหาย , การลัดวงจร หรือจากการที่ไม่มีไฟเลี้ยงในบางวงจร

2. กลิทช์ เป็นสัญญาณพัลซ์เล็กๆ ซึ่งเป็นสัญญาณที่ไม่ต้องการจะพบในวงจรที่ทำการออกแบบ ในวงจรที่จะใช้งานจริง จะต้องกำจัดสัญญาณกลิทช์ออกไปให้หมด

3. สไปค์ มีลักษณะที่คล้ายๆ กันกับกลิทช์ มักที่จะเกิดในสายส่งในระบบบัสที่มีการ coupling โดยตัวเก็บประจุในสายส่ง

4. เรส มีลักษณะคล้ายกับกลิทช์และสไปค์โดยเกิดจากการรวมของสัญญาณที่มีความเร็วต่างกันในวงจรลอจิกหนึ่งทำให้เกิดเป็นสัญญาณพัลซ์ลูกเล็กๆ เกิดขึ้น

5. ความผิดพลาดในคาบเวลา มักเกิดขึ้นในระบบการส่งข้อมูลโดยบัส ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลไปยังอุปกรณ์อีกชุดหนึ่งซึ่งจะมีการสุ่มข้อมูลโดยพัลซ์ (Sampling pulse) ที่ขอบขาลงของพัลซ์ ถ้าพัลซ์ถูกทำให้มาช้ากว่าปกติด้วยเหตุผลใดเหตุผลหนึ่ง และมีผลทำให้พัลซ์ไปปรากฏที่ช่วงที่ข้อมูลมีการเปลี่ยนแปลง จะทำให้ได้ข้อมูลที่ผิดพลาดได้

6. ริงกิ้ง เกิดขึ้นโดยที่สัญญาณอินพุทเกิดการออสซิลเลต ผ่านระดับแรงดันหลายๆ ครั้งก็จะทำให้ข้อมูลที่ได้เกิดการผิดพลาดขึ้นมาได้

7. ระดับแรงดันผิดพลาด ในวงจรทางลอจิกแต่ละตระกูลจะมีระดับแรงดันที่ใช้ในการทำงานทำงานที่แตกต่างกัน เมื่อนำมาทำงานเข้าด้วยกันโดยไม่เหมาะสมก็จะทำให้มีการแปรความหมายของสัญญาณที่ผิดพลาดระหว่างวงจรลอจิกได้

ความผิดพลาดทางซอฟแวร์ (Software Faults)

มักที่จะเกิดจากความผิดพลาดของโปรแกรม ซึ่งแบ่งออกได้เป็น 3 ชนิด คือ

1. ความผิดพลาดของคำสั่ง (Wrong Instructions) เกิดจากการใช้คำสั่งที่ผิดทำให้มีการทำงานผิดรูปแบบไป นอกจากนี้ยังอาจเกิดจากผิดพลาดของแอดเดรสก็ได้ (Wrong address)

2. ความผิดพลาดแอบแฝง (Latent faults) อาจจะเป็นความผิดพลาดของคำสั่งหรือความผิดพลาดทางฮาร์ดแวร์ เช่น คำสั่ง JMP กับคำสั่ง JNZ ที่มี code ต่างกันเพียง 1 บิท เท่านั้น

JMP C3 11000011

JNZ C2 11000010

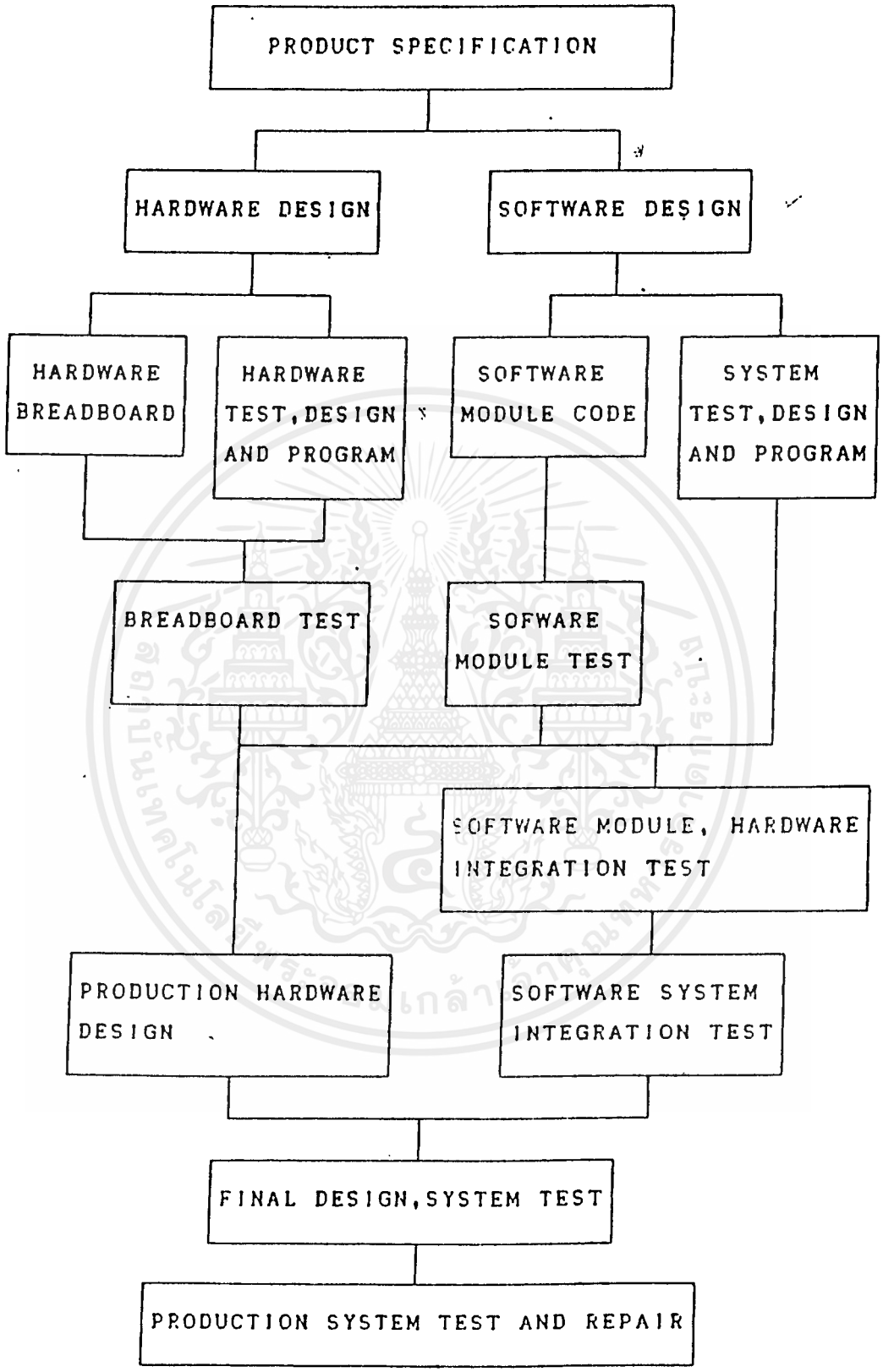
3. ความผิดพลาดในคาบเวลา (Timing faults) เช่นการติดต่อระหว่างผู้ส่งกับผู้รับ ถ้าผู้รับมีระบบการทำงานที่ช้ากว่า และโปรแกรมที่ใช้ควบคุมการทำงานไม่ได้โปรแกรมให้มีการรอคอยการทำงานในสภาวะ wait state ก็จะทำให้เกิดความผิดพลาดขึ้นมาได้

ในการที่จะตรวจสอบข้อผิดพลาดเหล่านี้ จำเป็นที่จะต้องมีครื่องมือที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัดสัญญาณต่างๆ ขึ้นมาเปรียบเทียบพร้อมกันได้ ซึ่งลอจิกอนาไลเซอร์ก็มีความสามารถเช่นนี้อยู่แล้ว ดังแสดงในแผนภาพที่แสดงกระบวนการในการออกแบบผลิตภัณฑ์ทางด้านไมโครโพรเซสเซอร์และดิจิตอล ซึ่งจะต้องใช้ทั้งฮาร์แวร์และซอฟแวร์ประกอบกัน ลอจิกอนาไลเซอร์จะถูกใช้ในส่วนที่เป็นการทดสอบการทำงานและแก้ไขข้อบกพร่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการเบื้องต้นของลอจิกอนาไลเซอร์

หน้าที่หลักของลอจิกอนาไลเซอร์ก็คือการแซมปลิ้ง (Sampling) ข้อมูลแล้วนำไปเก็บในหน่วยความจำ เพื่อใช้สำหรับการแสดงผลและวิเคราะห์ข้อมูลในภายหลัง ดังนั้นลอจิกอนาไลเซอร์จะประกอบด้วยส่วนประกอบ 3 ส่วน คือ

1. ภาคควบคุมการสุ่มข้อมูล (data acquisition)
2. ภาคควบคุมการหยุดสุ่มข้อมูล (trigger function)
3. ภาคแสดงผล (display)

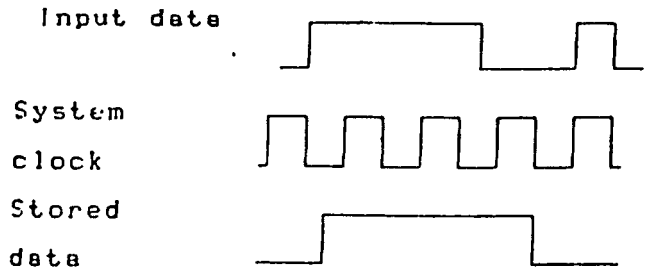
เราสามารถเขียน block diagram แสดงส่วนประกอบและการทำงานของลอจิกอนาไลเซอร์ได้ดังรูปที่ 2-1

ภาคควบคุมการสุ่มข้อมูล (Data Acquistion)

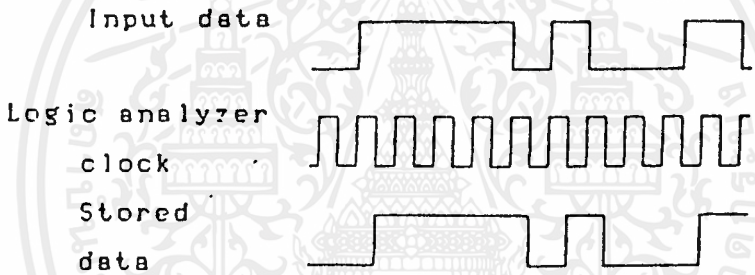
สัญญาณนาฬิกา (clock) ข้อมูลจะถูกสุ่มเข้ามาทุกๆ clock pulse จาก block diagram จะเห็นว่าเราสามารถเลือกใช้สัญญาณนาฬิกาได้ 2 แบบ คือสัญญาณนาฬิกาของระบบ (system clock) หมายถึงสัญญาณนาฬิกาของวงจรที่เราตรวจสอบซึ่งโดย
ทั่วไปข้อมูลเข้าจะมีความสัมพันธ์กับสัญญาณนี้ ในกรณีนี้เรียกว่าการทำงานแบบซิง-โครนัส (Synchronous Mode) อีกกรณีที่นิยมใช้กันมากกว่าคือใช้สัญญาณนาฬิกาที่สร้างขึ้นมาในตัวของลอจิกอนาไลเซอร์เอง เรียกว่าการทำงานแบบอะซิงโครนัส (Asynchronous Mode) ข้อมูลที่สุ่มจะมีความถูกต้องใกล้เคียงกับข้อมูลจริงเพียงใดจะขึ้นอยู่กับความถี่ของสัญญาณนาฬิกาที่ใช้ โดยปกติจะใช้ความถี่ที่สูงกว่าความถี่ของข้อมูลเข้าประมาณ 5-10 เท่า

เหตุผลหนึ่งที่ต้องมีการทำงานแบบอะซิงโครนัส เพราะว่าสัญญาณบางอย่าง เช่น สไปค์หรือกลิทช์ จะเกิดขึ้นโดยไม่ได้สัมพันธ์กับสัญญาณนาฬิกาของระบบและมีความแคบมาก ทำให้ข้อมูลไม่ถูก Sampling ถ้าใช้วิธีซิงโครนัส จึงต้องแก้ไขด้วยการเพิ่มความถี่สัญญาณนาฬิกาโดยการทำงานแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการทำงานแบบซิงโครนัส



รูปที่ 2.3 แสดงการทำงานแบบอะซิงโครนัส

การจับสัญญาณกลิทช์ (Glitch Capturing)

ในการสุ่มข้อมูลตามปกตินั้น เราจะใช้ขอบของสัญญาณเป็นตัวกำหนดเรียกว่า Sampling Mode ซึ่งมีข้อจำกัดคือ ถ้าอินพุทเป็นพัลซ์เล็กๆ อยู่ระหว่างสัญญาณนาฬิกา และมีความแคบมากกว่าคาบเวลาของสัญญาณนาฬิกา จะไม่ถูกนำไปเก็บ ซึ่งสามารถแก้ไขได้ด้วยการทำงานแบบ latch mode โดยใช้มาสเตอร์-สเลฟ ฟลิปฟลอปความเร็วสูงมาเป็นตัวเก็บพัลซ์ไว้ก่อนที่จะนำไปเก็บไว้ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

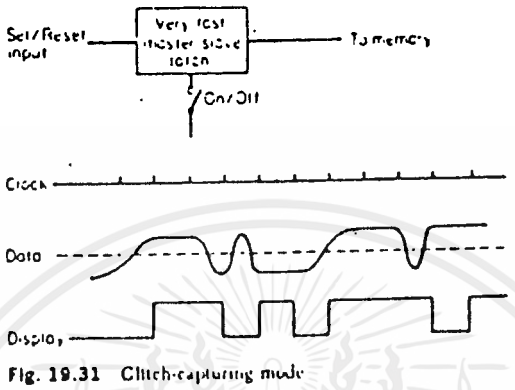


Fig. 19.31 Glitch-capturing mode

รูปที่ 2.4 แสดงการจับสัญญาณกลิทช์

ข้อมูลเข้า (Data Input)

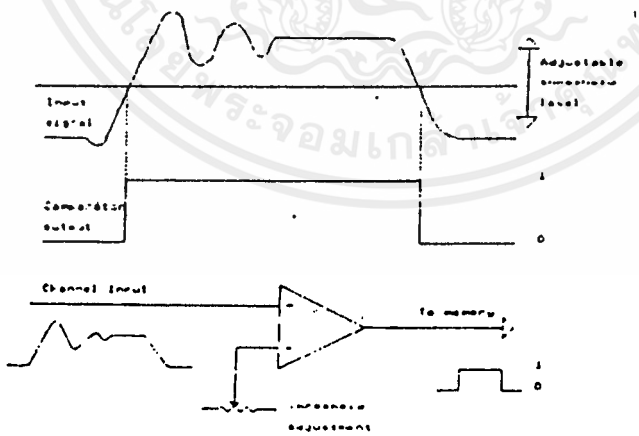
ก่อนที่จะรับข้อมูลเข้ามาเก็บในลอจิกนาไลเซอร์ จะต้องเลือกช่วงของระดับของอินพุทที่เหมาะสม โดยวิธีเลือกแรงดันเทรสโฮลด์ (Threshold Level) ซึ่งขึ้นอยู่กับว่าเป็นลอจิกตระกูลไหน (ตัวอย่างเช่น TTL มีแรงดันเทรสโฮลด์ เท่ากับ 1.4 V.) รูป
ที่ 2.5 แสดงถึงการใช้แรงดันเทรสโฮลด์เปลี่ยนระดับสัญญาณ การที่สารถเปลี่ยนค่าแรงดันเทรสโฮลด์ได้นี้มีประโยชน์มากในการตรวจสอบความผิดพลาดที่เกิดขึ้นมาจากแรงดันของสัญญาณที่ผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

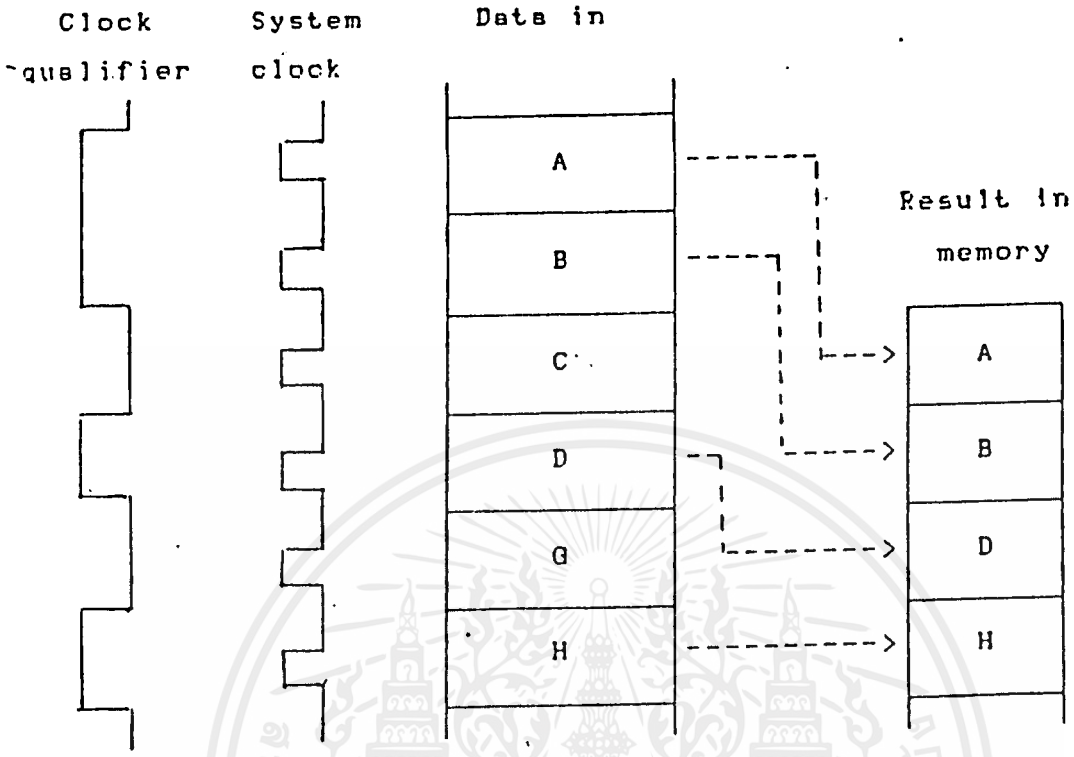
รูปที่ 2.5 การเปลี่ยนระดับของสัญญาณข้อมูล

คล็อกควอลิฟายเออร์ (Clock Qualifier)

ในการทำงานแบบซิงโครนัส ในบางครั้งเราต้องการที่จะสุ่มข้อมูลเฉพาะในบางช่วงเท่านั้น จึงจำเป็นที่จะต้องควบคุมการปล่อยสัญญาณนาฬิกาให้ได้ตรงกับช่วงเวลาของข้อมูลที่ต้องการ ซึ่งทำได้โดยการนำสัญญาณคล็อกควอลิฟายเออร์มา AND กับสัญญาณนาฬิกาดังในรูปที่ 2.6 จะเห็นข้อมูลที่เราต้องการอยู่ปะปนกับข้อมูลอื่นๆ สมมติให้คล็อกควอลิฟายเออร์คือสัญญาณอ่าน/เขียนหน่วยความจำของระบบที่กำลังทดสอบ โดยถ้าสัญญาณเป็น 1 หมายถึง ช่วงการอ่าน และถ้าเราต้องการที่จะสุ่มข้อมูลในช่วงนี้เราต้องเซ็ตให้คล็อกควอลิฟายเออร์แอคทีฟที่ลอจิก 1 ดังนั้นข้อมูลที่เราจะเก็บได้ก็คือ ช่วงที่ระบบที่กำลังทดสอบอ่านหน่วยความจำเท่านั้น วิธีใช้คล็อกควอลิ-ฟายเออร์มีข้อดีอยู่อีกอย่าง คือ ช่วยให้ใช้หน่วยความจำของลอจิกอนาไลเซอร์ที่มีอยู่จำกัดได้อย่างมีประสิทธิภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 การปรับปรุงสัญญาณนาฬิกา

การเริ่มต้นการทำงาน (Start Function)

อาจทำได้หลายวิธี ช่นให้ผู้ใช้เป็นผู้ส่งสัญญาณเริ่มต้นการทำงาน, รับสัญญาณมาจากภายนอก หรือตั้งเป็นแบบอัตโนมัติในทุกๆ ช่วงเวลาที่กำหนด

ภาคควบคุมการหยุดสุ่มข้อมูล (Trigger Function)

ในช่วงที่การสุ่มข้อมูลกำลังดำเนินอยู่ ข้อมูลจะถูกเก็บไปไว้ในหน่วยความจำตลอดเวลาและจะไม่มีการแสดงผลออกมาให้เห็น ดังนั้นจึงต้องมีส่วนที่จะมาทำหน้าที่ควบคุมหยุดการสุ่มข้อมูล เรียกว่าทริกเกอร์ แบ่งออกได้เป็น 3 ลักษณะ คือ การทริกภายนอก (External Triggering) , การทริกภายใน (Internal Triggering) และการทริก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวผู้ใช้เอง (Manual Triggering)

การทริกโดยตัวผู้ใช้เอง (Manual Triggering) อาจจะใช้เป็นลักษณะของสวิทช์กดเพื่อหยุดการทำงาน โดยปกติแล้วการทริกแบบนี้ไม่ค่อยจะได้ใช้กันมากนัก มักจะใช้เพื่อหยุดการสุ่มข้อมูลในทันทีทันใด

การทริกภายใน (Internal Triggering) การทริกแบบนี้เกิดจากสัญญาณที่สร้างมาจากการเปรียบเทียบข้อมูลกับทริกเกอร์เวิร์ด การสุ่มข้อมูลจะหยุดเมื่อพบว่าสัญญาณทั้งสองเหมือนกัน เราสามารถใช้ทริกเกอร์เวิร์ดหลาย ๆ ชุดให้ทำงานต่อเนื่องกันไปเพื่อให้ได้รับข้อมูล ที่ต้องการ ลักษณะอีกอย่างหนึ่งคือการใช้ทริกเกอร์ควอลิฟายเออร์มาควบคุมการทริกร่วมกันกับทริกเกอร์เวิร์ด ดังในรูปที่ 2.8 จะเห็นว่าถึงแม้ข้อมูลในช่วงสัญญาณนาฬิกาลูกที่สองจะตรงกันกับทริกเกอร์เวิร์ดแต่จะไม่มีการทริกเกิดขึ้นเพราะว่าทริกเกอร์ควอลิฟายเออร์ไม่ได้ตรงกับที่เรากำหนดการทริกจะเกิดขึ้นในสัญญาณนาฬิกาลูกที่สิบเพราะมีเงื่อนไขตรงกันทั้งสองอย่าง

Trigger Word – 1011 0111 0011 1110 –

Clock	Data Channel	
1	0100 1101 0010 0110	
2	1011 0111 0011 1110	<-- Trig

รูปที่ 2.7 ทริกเกอร์เวิร์ด

Trigger Qualifier	0	1011 0111 0011 1110	
Clock	Q	Data Channel	
1	0	0100 1111 1101 1010	
2	1	1011 0111 0011 1110	<-- Non
3	1	1010 1100 1110 1001	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

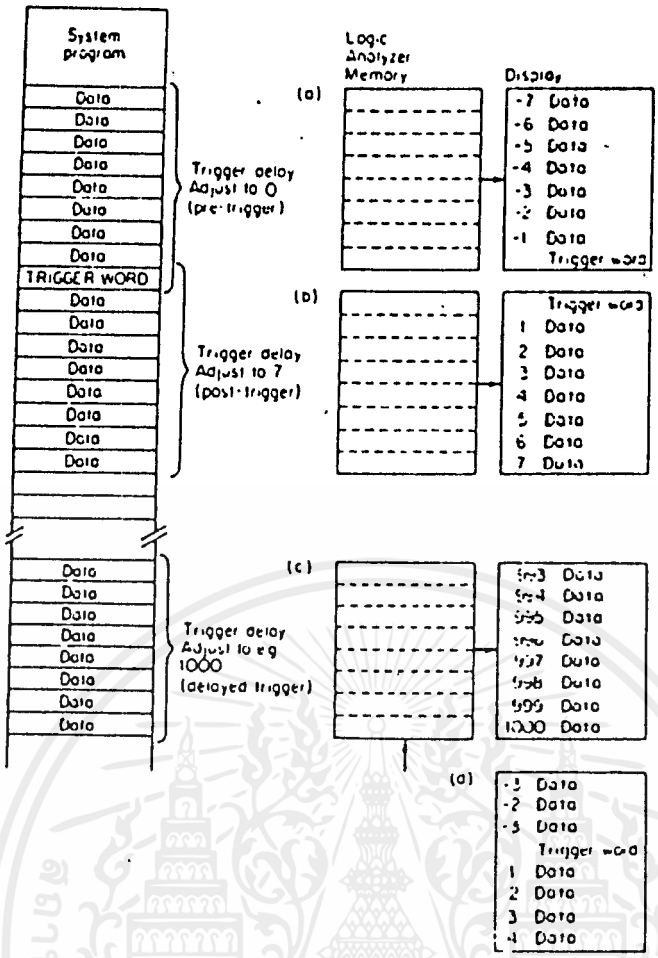
4	1	0001 0001 0101 1101
5	1	1110.0011 1011 0111
6	0	0000 1010 1000 1000
7	0	1100 1100 0111 0001
8	1	1010 0010 1011 1000
9	1	0110 1101 1100 0111
0	0	1011 0111 0011 1110 <-- Trig

รูปที่ 2.8 ทริกเกอร์เวิร์ดและทริกเกอร์ควอลิฟายเออร์

การทริกภายนอก (External Triggering) การทริกสามารถที่จะควบคุมได้จากภายนอกเช่นกัน โดยใช้สัญญาณทริกจากภายนอก

การหน่วงเวลาทริก (Trigger Delay) การหน่วงเวลาในการทริกจะทำให้เราสามารถเลือกจำนวนของข้อมูลที่อยู่ก่อนหรือ หลังทริกเกอร์เวิร์ดได้ เราหน่วงเวลาการเกิดของสัญญาณทริกเป็นจำนวนครั้งของการสุ่มข้อมูลซึ่งสามารถเลือกได้โดยผู้ใช้ และขอยกตัวกย่างการใช้งานทริกเกอร์ดีเลย์ดังในรูปที่ 2-9 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 การหน่วงเวลาทริก

ถ้าไม่มีการหน่วงเวลาดังในรูปที่ 2.9 A ทันทีที่ข้อมูลเข้าและทริกเกอร์เวิร์ดเท่ากันการสุ่มข้อมูลก็จะหยุดลงในทันที และข้อมูลก็จะนำออกแสดง ข้อมูลสุดท้ายก็คือ ทริกเกอร์เวิร์ด เราเรียกลักษณะการทริกแบบนี้ว่า พรีทริกเกอร์ (pretrigger)

สมมติว่าหน่วยความจำมีความยาวเท่ากับ 8 ข้อมูล (รูปที่ 2.9 B) เลือกให้มีการหน่วงเวลาเท่ากับ 7 คล็อก เมื่อพบกับทริกเกอร์เวิร์ดแล้วตัวนับ (counter) จะเริ่มนับถอยหลังในทุกๆ ครั้งที่มีการสุ่มข้อมูลจนเท่ากับศูนย์ สัญญาณทริกเกอร์จะถูกส่งออกมา ซึ่งจะทำให้เราได้ข้อมูลหลังจากทริกเกอร์เวิร์ด 7 ข้อมูลแล้วจึงนำไปแสดงผลเราจะเรียกการทริกในลักษณะนี้ว่า โพสท์ทริกเกอร์ (post-trigger)

การทริกในแบบสุดท้ายจะเป็นการนำเอาการทริกในแบบแรกและแบบที่สองมารวมไว้ด้วยกัน นั่นคือ ถ้าเราให้มีค่าดีเลย์เป็น 4 ก็จะได้ข้อมูลก่อนทริกเกอร์เวิร์ด 4 ข้อมูล และจะได้ข้อมูลหลังทริกเกอร์เวิร์ด 3 ข้อมูล เราเรียกการทริกในลักษณะนี้ว่า เซ็นเตอร์ทริกเกอร์ (รูปที่ 2.9 C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของภาคแสดงผล

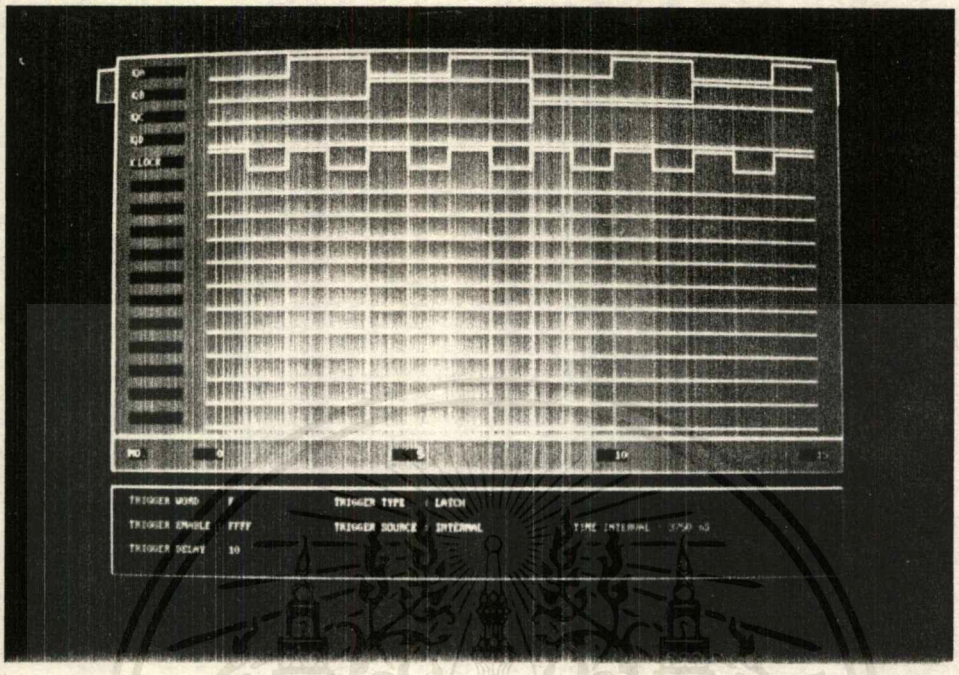
เป็นการนำข้อมูลที่เก็บอยู่ในหน่วยความจำมาจัดให้อยู่ในรูปแบบที่เหมาะสม แล้วจึงนำผลไปแสดงออกทางจอภาพหรือพริ้นท์เตอร์ รูปแบบของการแสดงผลแบ่งออกมาเป็นการ แสดงสถานะ (State Display) และการแสดงเวลา (Timing Display) ในการแสดงสถานะนั้นจะอยู่ในรูปความสัมพันธ์ระหว่างสถานะทางลอจิกของแต่ละอินพุทตามลำดับในการสุ่มข้อมูล งอาจจะแสดงในรูปของเลขฐานสอง (Binary Format) หรืออยู่ในรูปของเลขฐานสิบหก (Hexadecimal Format) ดังในรูปที่ 2.10 หรืออาจจะอยู่ในรูปแบบอื่นๆ ก็ได้

8 BIT	T=XXXX	C=S+0000	MEM-SRC
CURS	BIN	OCT	HEX
0490S	00001111	017	0F
0491	00000000	000	00
0492	00110000	060	30
0493	00110001	061	31
0494	10010010	222	92
0495	10010010	222	92
0496	00010010	022	12
0497	00000000	000	00
0498	00000000	000	00

รูปที่ 2.10 การแสดงผลแบบเลขฐานสองและเลขฐานสิบหก

ส่วนการแสดงเวลานั้นจะเป็นการนำเอาข้อมูลมาแสดงเป็นแผนผังเวลา (Timing Diagram) ตามเวลาจริงที่ข้อมูลเกิดขึ้น (รูปที่ 2.11) ซึ่งมีลักษณะที่คล้ายกันกับออสซิลโลสโคปและเรายังสามารถขยายขนาดของภาพได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การแสดงผลแบบสัมพันธ์กับเวลา

นอกจากรูปแบบการแสดงผลแบบต่างๆ ที่กล่าวมาแล้วอาจจะแสดงออกมาทางทางเครื่อง พิมพ์แทนจอภาพก็ได้ นอกจากนี้ถ้าเราต้องการเก็บข้อมูลเอาไว้ทำการวิเคราะห์ในภายหลัง ก็จะทำได้โดยการนำข้อมูลในหน่วยความจำไปเก็บเอาไว้ในหน่วยความจำถาวร เช่น จานแม่เหล็ก (Floppy Disk) หรือเทป (Tape) ก็ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ฮาร์ดแวร์ของลอจิกอนาไลเซอร์

ในการออกแบบฮาร์ดแวร์นั้น มีสิ่งที่ต้องคำนึงถึงก็คือขีดความสามารถที่เราต้องการ, ความสะดวกต่อการใช้งานและความเชื่อถือได้ ซึ่งเราจะยึดถือเป็นแนวทางในการออกแบบต่อไป

1. คุณสมบัติที่ต้องการ (Specification)

การกำหนดคุณสมบัติต่างๆ จะยึดถือตามความสามารถที่เครื่องลอจิกอนาไลเซอร์ควรที่จะมีดังที่ได้กล่าวไว้ในบทที่ 2 โดยจะทำการปรับปรุงไปตามความเหมาะสมในการออกแบบ

คุณสมบัติที่กำหนดจะมีดังนี้

- Internal Clock มีความถี่อยู่ระหว่าง 500 Hz จนถึง 20 MHz
  External Clock มีความถี่สูงสุดที่ 40 MHz
  สามารถเลือกใช้ Clock Qualifier ได้
- มีช่องรับสัญญาณอินพุททั้งหมด 16 ช่อง โดยในแต่ละช่องจะมีหน่วยความจำเอาไว้สำหรับเก็บข้อมูลที่สุ่มเข้ามาได้ 4 Kbit
- การทริกสามารถเลือกได้ว่าจะให้เป็น Internal, External หรือ Manual Triggeriing
- ทริกเกอร์เวิร์ดมีขนาด 16 บิท เพื่อให้สามารถใช้งานกับของอินพุททั้งหมดได้
- สามารถปรับค่าทริกเกอร์ดีเลย์ได้ระหว่าง 0-4096 clock
- สามารถเลือกค่าแรงดัน threshold level ได้ตั้งแต่ -10V ถึง 10V โดยเพิ่มค่าเป็น step ได้ครั้งละ 0.1V และยังเลือกให้ใช้กับระดับ threshold ของลอจิกเกทในตระกูล TTL ได้ (1.4V)
- มีตัวดักจับสัญญาณกลิทช์
- ใช้ IBM/PC Compatible เป็นโฮสท์ในการทำงาน โดยตัวลอจิก-อนาไลเซอร์ จะมีลักษณะเป็น interface card อันหนึ่ง ที่จะเชื่อมต่อเข้ากับ IBM/PC โดยผ่านทางสล็อตขนาด 8 bit ที่มีอยู่แล้วบนเมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ด

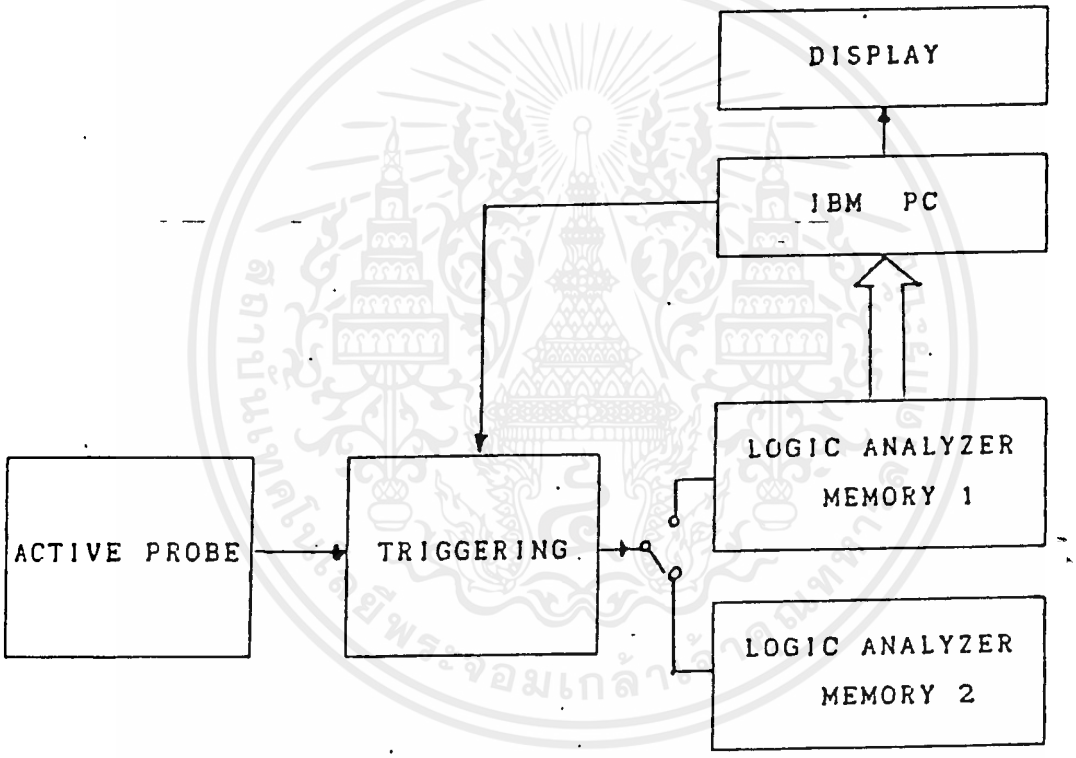
- ในส่วนของการแสดงผลนั้นจะกำหนดให้ต้องใช้จอภาพแบบ SVGA และจะทำงานในโหมดกราฟฟิกของ SVGA (1024 * 768 จุด) เพราะเป็นโหมดที่มีความเร็วในการทำงานสูงที่สุด
- สามารถใช้งานกับพริ้นเตอร์ได้สามตระกูลคือ EPSON LX, EPSON LQ และ HP Laserjet II

สำหรับการควบคุมการทำงานและการแสดงผลนั้นเลือกใช้วิธีการติดต่อผ่านเครื่อง IBM PC เพื่อความง่ายในการใช้งานและลดความยุ่งยากในการออกแบบวงจรแสดงผล ดังนั้นลักษณะของเครื่องจะเป็นการ์ดเสียบลงบนสล็อตของเครื่อง IBM PC

2. บล็อคไดอะแกรมของลอจิกอนาไลเซอร์

จากคุณสมบัติที่กำหนดมาเขียนเป็นบล็อคไดอะแกรมการทำงานได้ดังในภาพที่ 3.1 การควบคุมการทำงานจะตกเป็นหน้าที่ของซอฟแวร์ผ่านทางสล็อต 16 บิทของเครื่อง IBM/PC การทำงานจะเริ่มมาจากการ initial ฮาร์ดแวร์ทั้งหมดโดยการส่งข้อมูลที่จำเป็นมาที่พอร์ทต่างๆ ในลอจิกอนาไลเซอร์ รวมทั้งการส่งสัญญาณ START เพื่อที่จะเริ่มทำการสุ่มข้อมูล จากนั้นการสุ่มข้อมูลก็จะเกิดขึ้น โดยสัญญาณนาฬิกาจะถูกนำไปใช้สร้างแอดเดรสหน่วยความจำและสร้างสัญญาณสุ่มข้อมูลตามลักษณะการทำงานที่ได้เลือกการทำงานไว้การสุ่มข้อมูลจะ หยุดลงเมื่อมีสัญญาณ TRIG จากนั้นก็จะเป็นการย้ายข้อมูลจากหน่วยความจำของลอจิกอนาไลเซอร์ไปยังหน่วยความจำของ IBM/PC เพื่อทำการแสดงผลและวิเคราะห์การทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 บล็อคไดอะแกรมของลอจิกอนาไลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การออกแบบวงจร

การออกแบบวงจรนั้นจะแบ่งออกมาเป็น 7 ส่วนใหญ่ๆ คือ

1. ส่วนเชื่อมต่อกับ IBM/PC
2. ส่วนสร้างสัญญาณนาฬิกา
3. ส่วนสร้างแอดเดรสและสัญญาณควบคุมการสุ่มข้อมูล
4. ส่วนสร้างแรงดันเทรสโฮลด์
5. ส่วนรับข้อมูล
6. ส่วนสร้างสัญญาณทริก
7. หน่วยความจำ

ส่วนที่มีความสำคัญมากส่วนหนึ่งก็คือหน่วยความจำ ซึ่งเห็นได้ว่าต้องใช้หน่วยความจำที่มีความเร็วพอที่จะรับสัญญาณอินพุทได้ ดังนั้นจึงได้เลือกใช้ Static RAM ที่มี access time ที่ค่อนข้างน้อย ซึ่งในที่นี้เลือกใช้ 10 nS ทำให้ใช้กับอินพุทที่ความถี่ประมาณ 40 MHz ได้อย่างปลอดภัย

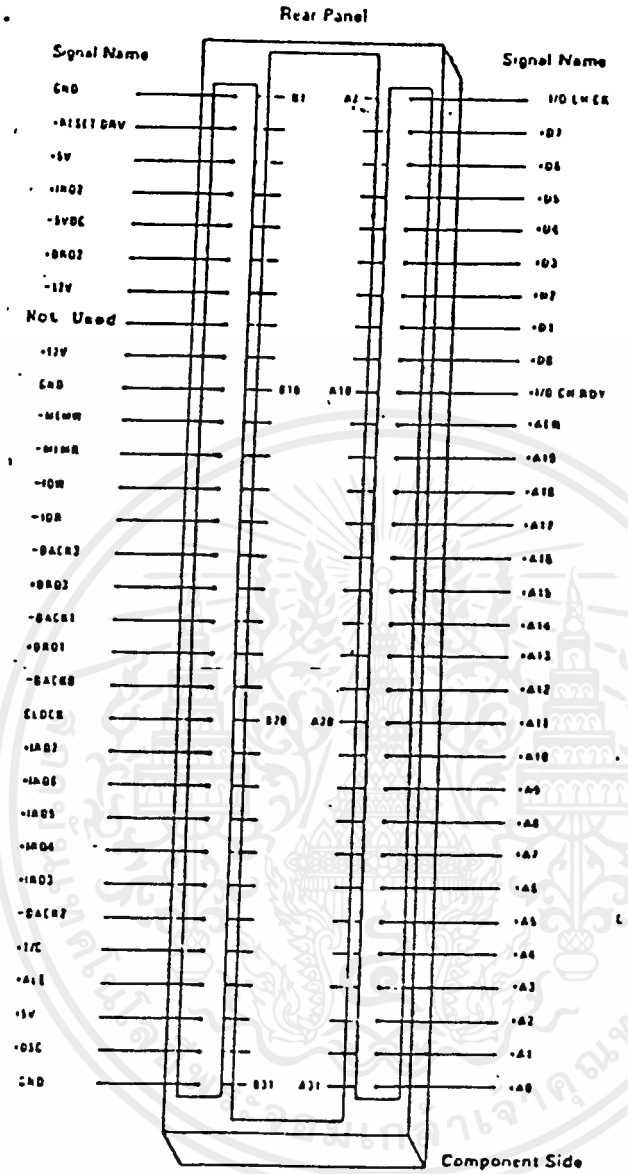
อีกส่วนที่มีความสำคัญในระหว่างการใช้งานมากๆ ก็คือ หน่วยแสดงผล เนื่องจากว่าจำเป็นที่จะต้องใช้กราฟในการทำงาน ถ้าเขียนโปรแกรมในส่วนนี้ไม่ดี ก็จะทำให้มีความเร็วในการเลื่อนของภาพค่อนข้างต่ำ ทำให้เห็นการกระพริบขึ้นมาได้ และการเขียนโปรแกรมทางด้านกราฟฟิกโดยทั่วไปนั้นจะอาศัย BGI (Basic Graphic Interface) เพื่อให้ง่ายในการเขียนโปรแกรม แต่ว่าใน BGI นั้นจะอาศัย BIOS Interrupt จึงมีความเร็วในการทำงานค่อนข้างที่จะต่ำ ดังนั้นจึงต้องเลี่ยงโดยเขียนการข้อมูลลงไปใน Video RAM โดยตรงซึ่งจะให้ความเร็วเพิ่มขึ้นได้มากแต่มีข้อเสียตรงที่โปรแกรมจะไม่เป็นมาตรฐาน และจะต้องเสียเวลาในการเขียนโปรแกรมมากขึ้น

3.1 ส่วนเชื่อมโยงกับ IBM PC

จากภาพที่ 3-2 แสดงถึงสัญญาณต่างๆ บนสล็อตของเครื่อง IBM PC ซึ่งจะมีสัญญาณที่เราใช้งานคือ

1) D0-D7 คือสายสัญญาณข้อมูล (Data Bus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3-2 สัญญาณต่างๆบนสล็อต IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Register
0AX*	NMI Mask Register
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communication (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C**	SDLC Communication
380-389**	Binary Synchronous Com(Secondary)
3A0-3A9	Binary synchronous Com(Primary)
3B0-3BF	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphic
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communication (Primary)

ภาพที่ 3-3 ตำแหน่งพอร์ทที่ถูกใช้โดย IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) A0-A9 คือสายสัญญาณแอดเดรสซึ่ง IBM ได้จัดเตรียมเอาไว้สำหรับการติดต่อกับอุปกรณ์ภายนอกจำนวน 10 เส้น (ที่เหลือจะใช้ในการขยายหน่วยความจำ)
3) AEN (Address Enable) เป็นสัญญาณที่บ่งบอกว่ากำลังอยู่ในช่วงของการทำ DMA อยู่ (AEN เป็น 1) ซึ่งอุปกรณ์อื่นๆ จะไม่สามารถใช้บัสของระบบได้
4) IOR สัญญาณบอกสถานะการอ่านข้อมูลจากอุปกรณ์ภายนอก
5) IOW สัญญาณบอกสถานะการเขียนข้อมูลไปที่อุปกรณ์ภายนอก
6) GND, +12V, -12V, +5V และ -5V เป็นไฟเลี้ยงของลอจิกอนาไลเซอร์

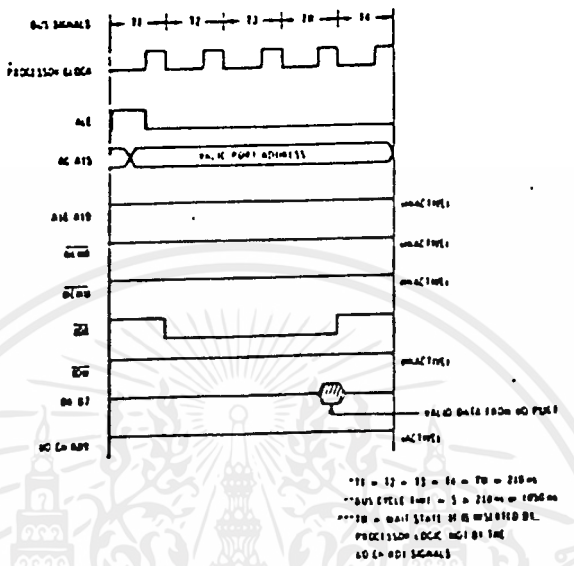
ในการเชื่อมโยงกับ IBM PC นี้จะทำผ่านทางพอร์ทอินพุทและเอาท์พุท ดังนั้นจึงต้องมีการดีโคดหมายเลขพอร์ท ซึ่งจะต้องไม่ให้ซ้ำกับของเดิมโดยใช้แอดเดรส A0-A9 มาทำการดีโคด โดยที่

ถ้า A9 = 0 หมายถึงพอร์ทบนเมนบอร์ด
ถ้า A9 = 1 หมายถึงพอร์ทบน I/O การ์ด

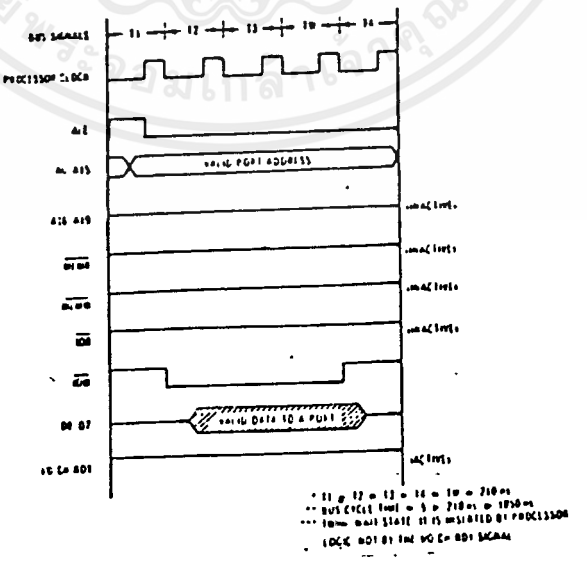
นอกจากนี้ยังมีสัญญาณอื่น ๆ ที่จะต้องนำมาพิจารณาในการดีโคดตำแหน่งพอร์ทได้แก่ IOR , IOW และ AEN ซึ่งจะพิจารณาถึงความสัมพันธ์ระหว่างสัญญาณต่างๆ ในขบวนการอ่านและเขียน I/O พอร์ทได้จากแผนผังเวลาในภาพที่ 3-4

จากข้อมูลทั้งหมดนำมาสร้างวงจรสำหรับติดต่อกับ IBM PC ได้ดังในภาพที่ 3-5 ซึ่งออกแบบการดีโคดแอดเดรสให้สามารถเปลี่ยนตำแหน่งได้ โดยเซ็ตที่ดิพ-สวิทช์ให้ตรงกับค่าที่แอดเดรสที่ต้องการ 74LS688 จะเป็นตัวเปรียบเทียบกับแอด-เดรสบัสว่าตรงกับค่าที่ตั้งเอาไว้หรือไม่ ถ้าตรงกันก็จะส่งสัญญาณไปอีนาเบิ้ลให้กับ 74LS138 เพื่อทำการดีโคดกลุ่มย่อยให้ไป เป็นสัญญาณเลือกพอร์ทอีกที ทำให้เราได้สัญญาณเลือกพอร์ท 8 เส้น (CS0-CS7) สำหรับในโครงงานนี้จะใช้อินพุทพอร์ทเพียง 4 พอร์ท และเอาท์พุทพอร์ท 11 พอร์ท แต่เนื่องจากว่า 8255 หนึ่งตัวต้องใช้พอร์ท 1 พอร์ทสำหรับทำการควบคุมคำสั่ง จึงต้องใช้ 8255 ทั้งหมด 5 ตัวและใช้ SA0, SA1, IOR และ IOW ต่อร่วมกับสัญญาณเลือกพอร์ทเพื่อที่จะควบคุมการเขียนและอ่านพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 ก. แผนผังเวลาการอ่านข้อมูลจากอินพุทพอร์ท



รูปที่ 3-4 ข. แผนผังเวลาการเขียนข้อมูลลงบนเอาท์พุทพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของสัญญาณที่ใช้ในแต่ละพอร์ทจะมีดังนี้

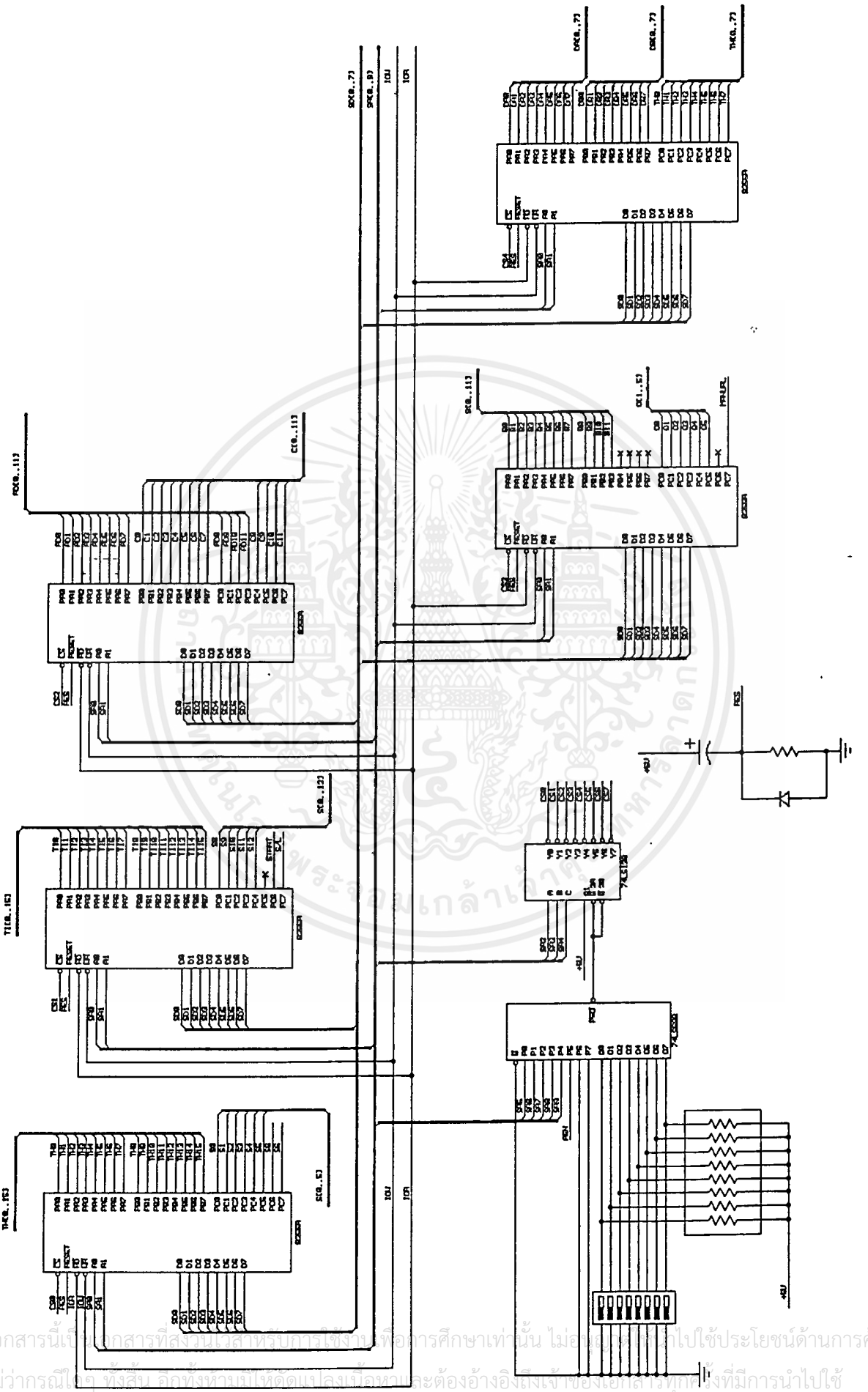
อินพุทพอร์ท

- DA0..DA7 ดาต้าบัสของหน่วยความจำชุดที่ 1
- DB0..DB7 ดาต้าบัสของหน่วยความจำชุดที่ 2
- B0..B11 แอดเดรสของข้อมูลสุดท้ายที่เก็บในหน่วยความจำ
- FULL แสดงสถานะของการสิ้นสุดการเก็บข้อมูลว่าเกิดจากการเก็บเต็มหน่วยความจำหรือไม่
- STOP แสดงการหยุดสุ่มข้อมูล เมื่อมีการตรวจสอบว่าบิทนี้เป็น 1 ก็จะทำการย้ายข้อมูลไปยังหน่วยความจำของ IBM PC

เอาท์พุทพอร์ท

- S0..S5 สัญญาณควบคุมการเลือกสัญญาณนาฬิกา
- S8 ใช้ในการดิสเอเบิลสัญญาณแอดเดรสจากลอจิกอนาไลเซอร์ เพื่อใช้ในการอ้างแอดเดรสโดยตรงจาก IBM PC
- S9,S10 เลือกแหล่งสร้างสัญญาณทริก
- S11 เลือก Threshold voltage ของสัญญาณนาฬิกา และของ Clock Qualifier
- S12 เลือก Threshold voltage ของสัญญาณข้อมูล
- TW0..TW15 Trigger Word
- TI0..TI15 Trigger Bit Enable
- S/L เลือกโหมดในการแซมปลิงว่าเป็น Sampling Mode หรือ Latch Mode
- C0..C11 เวลาหน่วงสำหรับการทริก
- TH0..TH7 ระดับแรงดันอ้างอิงที่โปรแกรมส่งออกมาให้กับ DAC0800
- Q0..Q5 ควบคุม Clock Qualifier
- Manual สัญญาณทริกที่ส่งออกมาจาก IBM PC
- AD0..AD10 แอดเดรสสำหรับการอ่านข้อมูลจากหน่วยความจำ
- START สัญญาณเริ่มต้นการสุ่มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนสร้างสัญญาณนาฬิกา

จากภาพที่ 3-6 เป็นวงจรเลือกสัญญาณนาฬิกา ซึ่งสามารถเลือกได้ทั้งจากสัญญาณนาฬิกา ภายใน (Internal Clock) และสัญญาณนาฬิกาภายนอก (External Clock) ซึ่งตัวสัญญาณนาฬิกาภายในสามารถเลือกความถี่ได้ตั้งแต่ 500 Hz ถึง 20 MHz โดยจะสามารถเลือกความถี่ได้ในลักษณะ 5-2-1 วงจรที่ใช้สร้างความถี่เป็น Oscillator ขนาด 20 MHz แล้วใช้ไอซีเบอร์ 74LS90 เป็นวงจรหารสิบและใช้ไอซีซีเบอร์ 74LS93 เป็นวงจรหาร 12 และใช้ไอซีเบอร์ 74F151 เป็นตัวเลือกความถี่ ซึ่ง

สัญญาณนาฬิกาที่สร้างได้จะสามารถเลือกความถี่ได้ดังตารางต่อไปนี้

S5	S4	S3	S2	S1	S0	ความถี่สัญญาณนาฬิกา
1	0	0	0	0	0	40 MHz
1	0	0	0	0	1	20 MHz
1	0	0	0	1	0	10 MHz
1	0	0	1	0	0	5 MHz
0	1	1	0	0	1	2 MHz
0	1	1	0	1	0	1 MHz
0	1	1	1	0	0	500 KHz
0	1	0	0	0	1	200 KHz
0	1	0	0	1	0	100 KHz
0	1	0	1	0	0	50 KHz
0	0	1	0	0	1	20 KHz
0	0	1	0	1	0	10 KHz
0	0	1	1	0	0	5 KHz
0	0	0	0	0	1	2 KHz
0	0	0	0	1	0	1 KHz
0	0	0	1	0	0	500 Hz
1	1	1	0	0	0	External Clock
1	1	1	1	1	1	External Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของสัญญาณนาฬิกาภายนอก (External Clock) จะมีวงจร Clock Qualifier ที่ใช้ในการเลือกสุ่มสัญญาณเป็นบางช่วงของสัญญาณตาม Qualifier ที่ตั้งเอาไว้ล่วงหน้า โดยใช้สัญญาณนาฬิกาภายนอกในการสุ่มข้อมูลดังนี้

	Logic	STATE
Q1, Q3, Q5	0	Don't Care
	1	H,L
Q0, Q2, Q4	0	L
	1	H

ตารางแสดงการเลือกใช้ Clock Qualifier

ในการเลือกใช้ Clock Qualifier นั้นจะมีลักษณะการทำงาน 3 สถานะ คือ X (don't care), H (High) และ L (Low) และจะมี Qualifier อยู่ 3 แชนแนล จะเห็นว่า ถ้า Q1, Q3 หรือ Q5 มีสถานะเป็น LOW แล้ว Qualifier นั้นจะมีสภาพเป็น don't care แต่ถ้า Q1, Q3 หรือ Q5 มีสถานะเป็น HIGH และ Q0, Q2 หรือ Q4 มีสถานะเป็น LOW แล้ว Qualifier นั้นจะทำงานที่สถานะ LOW แต่ถ้า Q0, Q2 หรือ Q4 มีสถานะเป็น HIGH แล้ว Qualifier นั้นก็จะทำงานที่สถานะ HIGH

จากวงจรจะเห็นว่าใช้ EXCLUSIVE OR เป็น GATE ในการเลือกสถานะของตัว Qualifier และใช้ NAND GATE แบบ Open Collector เป็นตัวรวมสัญญาณเข้าไปยัง D ฟลิป ฟลอป ซึ่งทำงานตามสัญญาณนาฬิกาภายนอกที่ใช้เป็น clock ให้กับ D-FF โดยสัญญาณที่ออกจะไปเข้า NAND GATE พร้อมกับสัญญาณนาฬิกาภายนอกได้เป็นสัญญาณนาฬิกาเข้าไปยังไอซีเบอร์ 74LS151

ตัวอย่างการทำงานของ CLock Qualifier

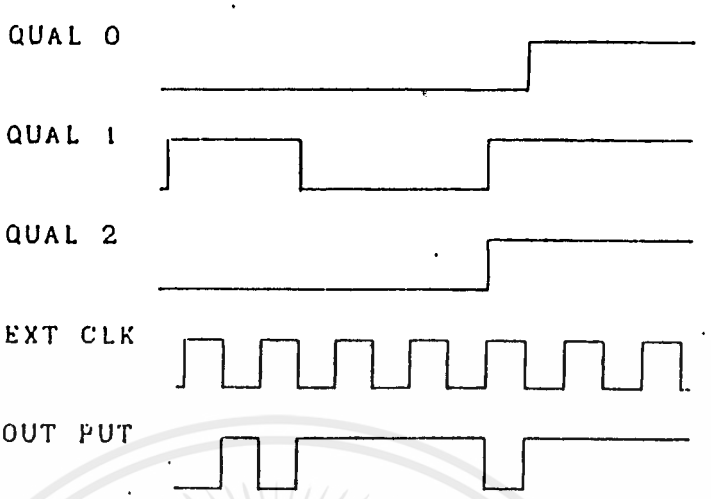
ถ้าให้ Qualifier 0 ทำงานที่สถานะ L

Qualifier 1 ทำงานที่สถานะ H

Qualifier 2 ทำงานที่สถานะ X

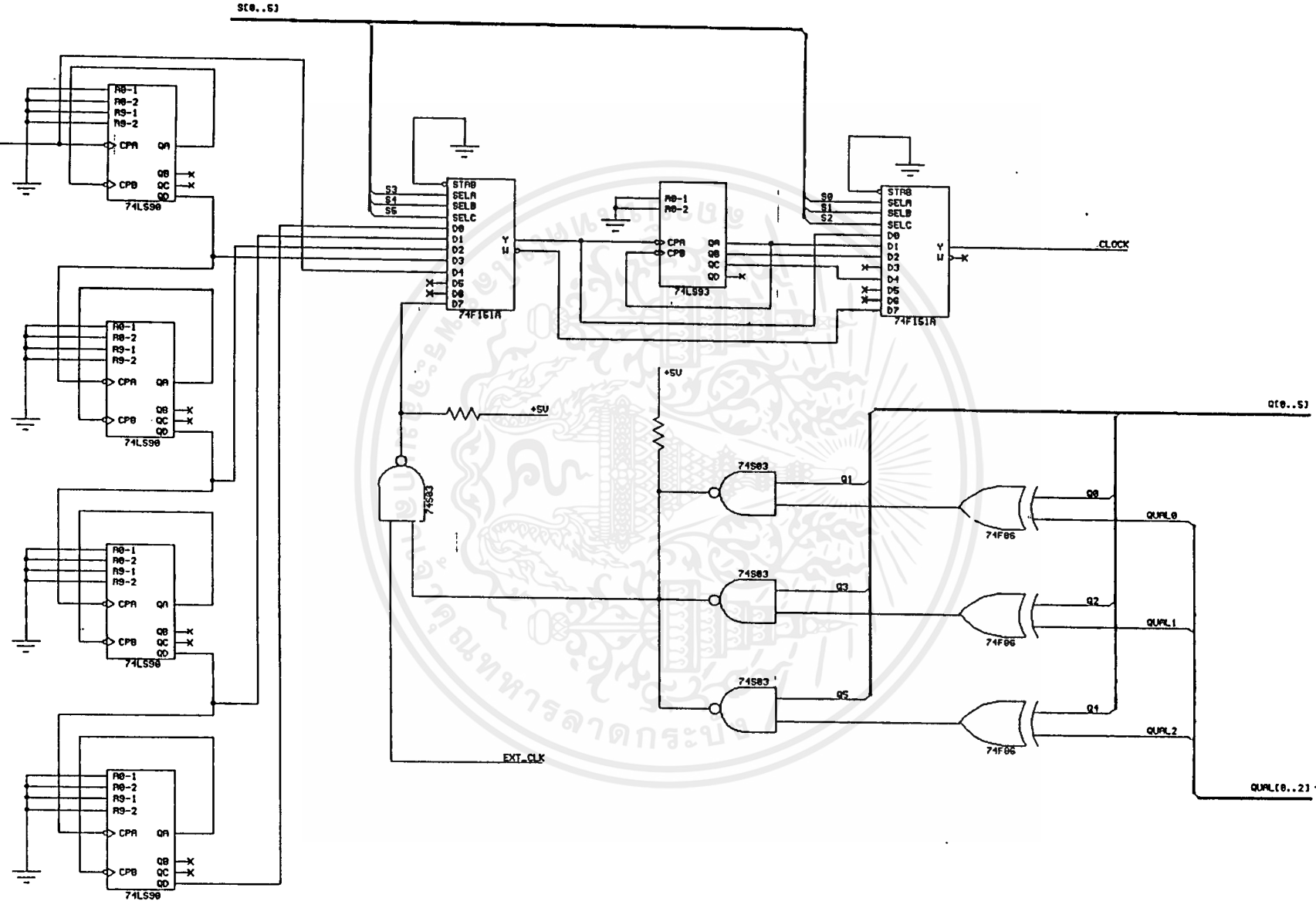
ก็จะได้ผลดังรูปข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตัวอย่างการทำงานของ Clock Qualifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


S[0..5]
20MHZ
74LS90
74F161A
74LS93
74F151A
CLOCK
+5V
74S03
74F86
EXT_CLK
Q[0..5]
QUAL0
QUAL1
QUAL2
QUAL[0..2]

3.3 ส่วนสร้างสัญญาณแอดเดรสและสัญญาณควบคุมการหยุดสุ่มข้อมูล

ในวงจรนี้ประกอบด้วยวงจรที่สำคัญสองส่วน คือ

1. วงจรหน่วงเวลาทริก
2. วงจรสร้างสัญญาณแอดเดรส

วงจรหน่วงเวลาทริก

ประกอบด้วย

- ไอซีเบอร์ 74F161 3 ตัว ใช้เป็นวงจรนับค่าหน่วงเวลา
- ไอซีเบอร์ 74F151 1 ตัว ใช้เป็นวงจรสร้างสัญญาณ CLEAR

จากภาพที่ 3-8, C0-C11 เป็นอินพุทของไอซีเบอร์ 74F161 ทั้ง 3 ตัว ซึ่งจะทำ การเซ็ตค่าหน่วงเวลา โดยมีขั้นตอนการทำงานดังต่อไปนี้

1) สัญญาณ START ที่มาจาก port จะมีสถานะเป็น HIGH ทำให้ขา LOAD ของ 74F161 เป็น LOW ทำให้ข้อมูลค่าหน่วงเวลาถูกส่งเข้าไปยัง 74F161 และ D-FFทั้งสองตัวจะถูกเคลียร์

2) เอาท์พุท Q ของ D-FF ทั้งสองตัวมีสถานะเป็น LOW และ Q เป็น HIGH ดังนั้นที่อินพุทของ D-FF ตัวแรกจึงมีสถานะเป็น HIGH และสัญญาณ WE เป็น HIGH

3) เมื่อเอาท์พุทของ D-FF เป็น LOW ทำให้ขา ENT ของ 74F161 เป็น LOW ดังนั้น 74F161 จึงยังไม่ทำงาน

4) เมื่อสัญญาณ START เป็น LOW ทำให้ขา CLK ของ D-FF เปลี่ยนจาก HIGH เป็น LOW ทำให้เอาท์พุทของ D-FF ตัวแรกเป็น HIGH และสัญญาณ WE เป็น LOW วงจรเริ่มทำการสุ่มข้อมูล

5) เมื่อสัญญาณ TRIG เปลี่ยนสถานะจาก HIGH เป็น LOW ทำให้เกิดการทริกขึ้นที่ขา CLK ของ D-FF ตัวที่ 2 ขา ENT ของ 74F161 เป็น HIGH จึงเริ่มทำการนับ

6) เมื่อทำการนับจนถึงค่า 0FFF จะทำให้สัญญาณที่ขา RC0 ของทั้ง 3 ตัว เป็น HIGH จึงทำให้เกิดสัญญาณ CLEAR ให้กับ D-FF ทั้งสองตัว และขา ENT ของไอซี 74F161 เป็น LOW วงจรจึงหยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสร้างสัญญาณแอดเดรส

ประกอบด้วย

- ไอซี 74F161 3 ตัว เป็นตัวนับ
- ไอซี 74F257 3 ตัวเป็นวงจรเลือกสัญญาณแอดเดรส

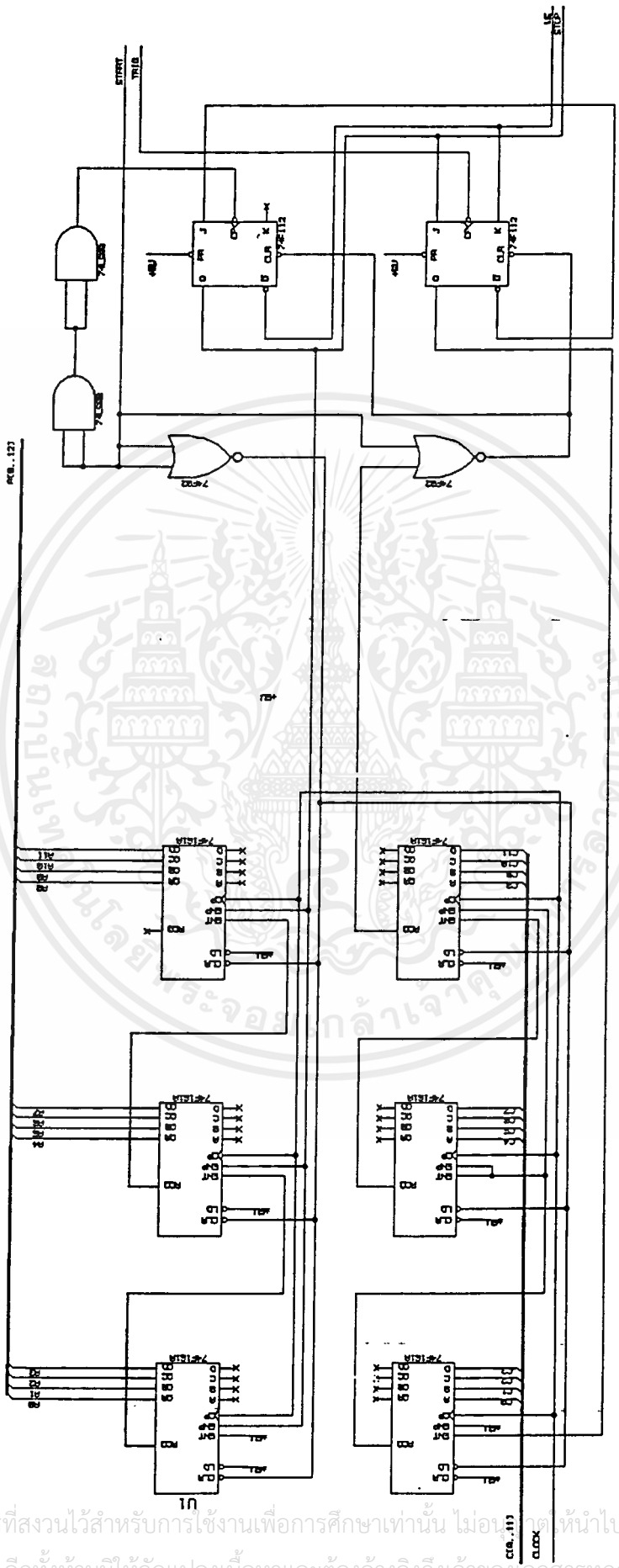
ขั้นตอนในการทำงาน

1) เมื่อสัญญาณ START มีสถานะเป็น HIGH ทำให้เอาท์พุทของ 74F151 เป็น LOW ส่งผลให้ 74F161 ถูกเคลียร์ และสัญญาณที่ขา ENP เป็น LOW ทำให้ยังไม่มีการนับเกิดขึ้น

2) เมื่อสัญญาณ START เป็น LOW เอาท์พุทของ 74F151 จะเป็น HIGH ทำให้ขา CLK ของ 74F161 เป็น HIGH ขา ENP จึงมีสถานะเป็น HIGH วงจรนับจึงเริ่มทำการนับ

3) เมื่อสัญญาณ TRIG เปลี่ยนจากสถานะ LOW เป็น HIGH จะทำให้วงจรนับในส่วนของวงจรนับค่าหน่วงเวลาทำงาน เมื่อวงจรนับค่าเวลาหน่วงนับจนถึงค่า 0FFF จะทำให้เอาท์พุทของ 74F151 มีสถานะเป็น LOW วงจรนับค่าเวลาหน่วงจึงหยุดนับและทำให้ขา ENP ของ 74F161 ในส่วนของวงจรสร้างสัญญาณแอดเดรสมีสถานะเป็น LOW วงจรนับก็จะหยุดนับก็จะทำให้ได้ค่าแอดเดรสตามที่ได้ตั้งเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



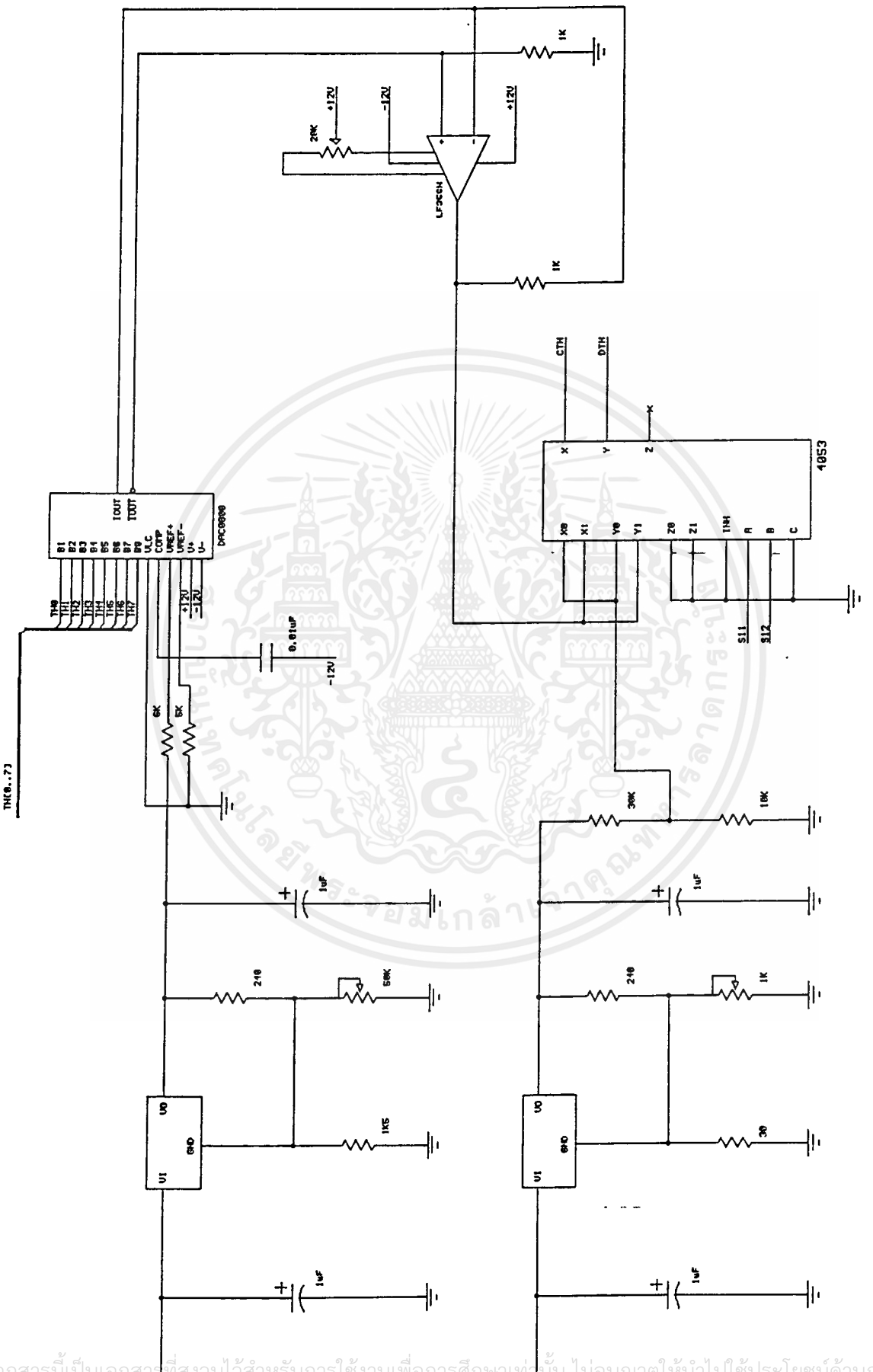
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนสร้างแรงดันเทรสโฮลด์

แรงดันเทรสโฮลด์แบ่งออกเป็น 2 ชุด คือ แรงดันคงที่ 1.4 โวลท์ สำหรับไอซีที่อยู่ในตระกูล TTL และแรงดันปรับค่าได้ตั้งแต่ -10 โวลท์ ถึง +10 โวลท์ สำหรับไอซีที่อยู่ในตระกูลอื่น และใช้ทดสอบความผิดพลาด เนื่องจากแรงดันผิดระดับ

การปรับค่าแรงดันเทรสโฮลด์นั้น ต้องสามารถทำได้ด้วยซอฟแวร์ ดังนั้นจึงต้องใช้วิธีการแปลงข้อมูลของ DAC เข้าช่วย โดยการใช้ข้อมูลจากเอาท์พุทพอร์ท (TH0-TH7) มาแปลงเป็นสัญญาณอนาลอกด้วยไอซี DAC0800 (ภาพที่ 3-9) เอาท์พุทที่ได้จะเป็นกระแส จึงต้องนำไปเข้าออปแอมป์เพื่อแปลงเป็นแรงดันและขยายให้ได้ขนาดตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ส่วนรับข้อมูล

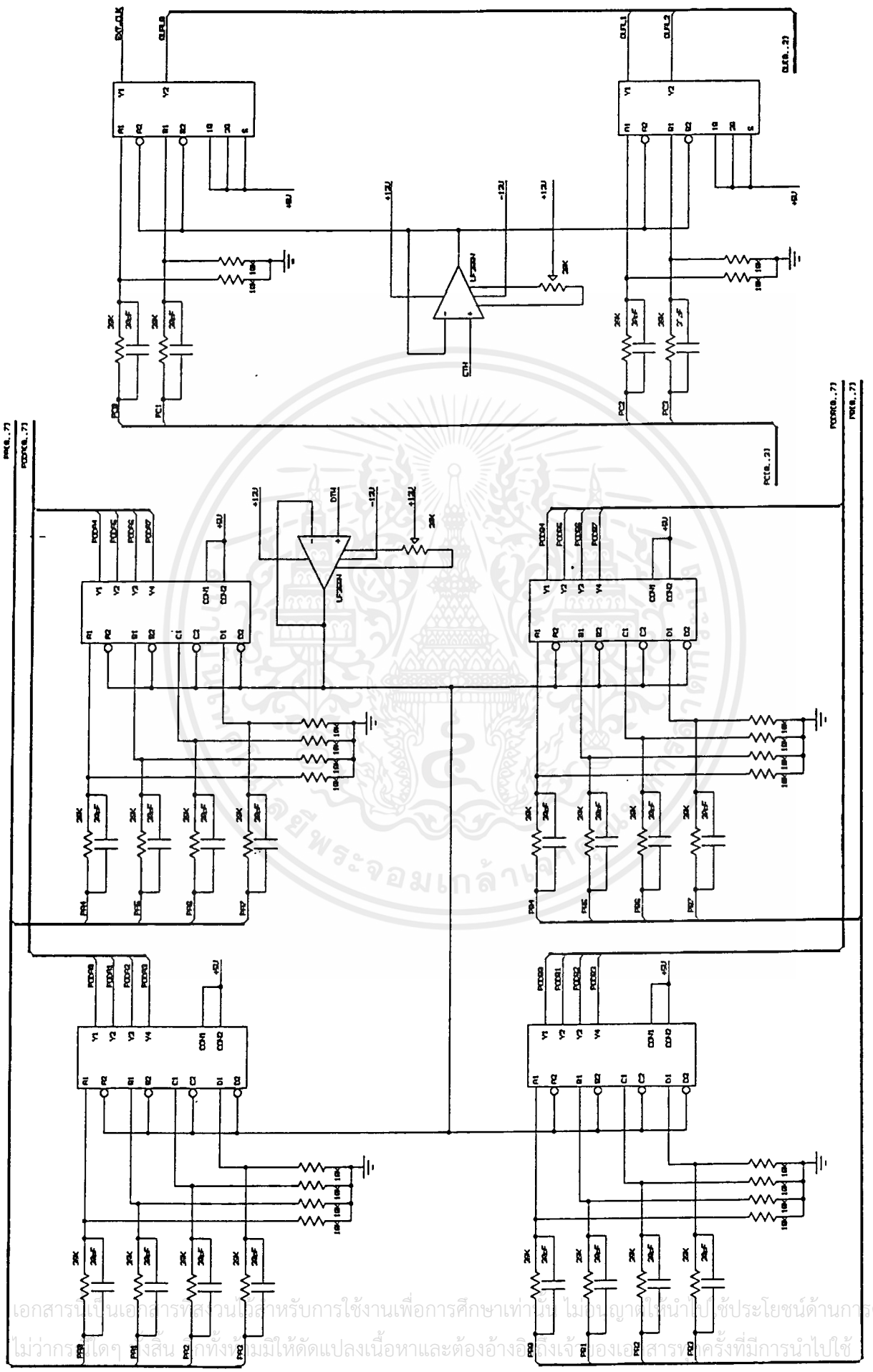
ส่วนนี้มีหน้าที่ในการแปลงระดับสัญญาณจากภายนอกให้อยู่ในระดับ TTL ซึ่งเป็นระดับที่ใช้งานในเครื่องลอจิกอนาไลเซอร์ โดยใช้หลักการเปรียบเทียบสัญ กับแรงดันเทรสโฮลด์ที่ตั้งเอาไว้ด้วยคอมพาราเตอร์ความเร็วสูง

สำหรับตัวเปรียบเทียบแรงดันใช้อุปกรณ์ประเภท Line Driver เนื่องจากมีข้อดีคือมีความเร็วสูง และให้เอาท์พุทที่เป็น TTL นอกจากนี้แล้วยังทำงานแบบสมิทซ์ทริกเกอร์ (Schmitt Trigger) ทำให้ป้องกันความผิดพลาดในการตรวจจับสัญญาณซึ่งเกิดขึ้นจากสัญญาณรบกวนได้อีกด้วย

สำหรับอุปกรณ์ที่ใช้ จะรับสัญญาณแบบคอมมอนโหมด (Common Mode) ได้ประมาณไม่เกิน 12 โวลท์ แต่ในการออกแบบต้องการให้รับอินทุทสูงสุดได้ถึง 50 โวลท์ จึงต้องต่อวงจรลดทอนสัญญาณก่อน โดยใช้วงจรแบ่งแรงดันธรรมดาที่มีอัตราการลดทอน 5 เท่า และที่วงจรแบ่งแรงดันนี้ ได้ต่อตัวเก็บประจุคร่อมตัวต้าน-ทานขนาด 39 K เพื่อชดเชยผลของตัวเก็บประจุแอบแฝง (Stray Capacitance) ซึ่งจะทำให้เกิดปรากฏการณ์ Low Pass Filter สำหรับค่าของตัวเก็บประจุที่ใช้ประมาณ 30-50 pF โดยได้มาจากการทดลอง

ผลที่เกิดจาดวงจรแบ่งแรงดันอีกอย่างก็คือ ทำให้ Total Input Impedance มีค่าสูงมากขึ้นกว่าเดิมประมาณ 12 K สำหรับ 75175

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ส่วนสร้างสัญญาณทริก

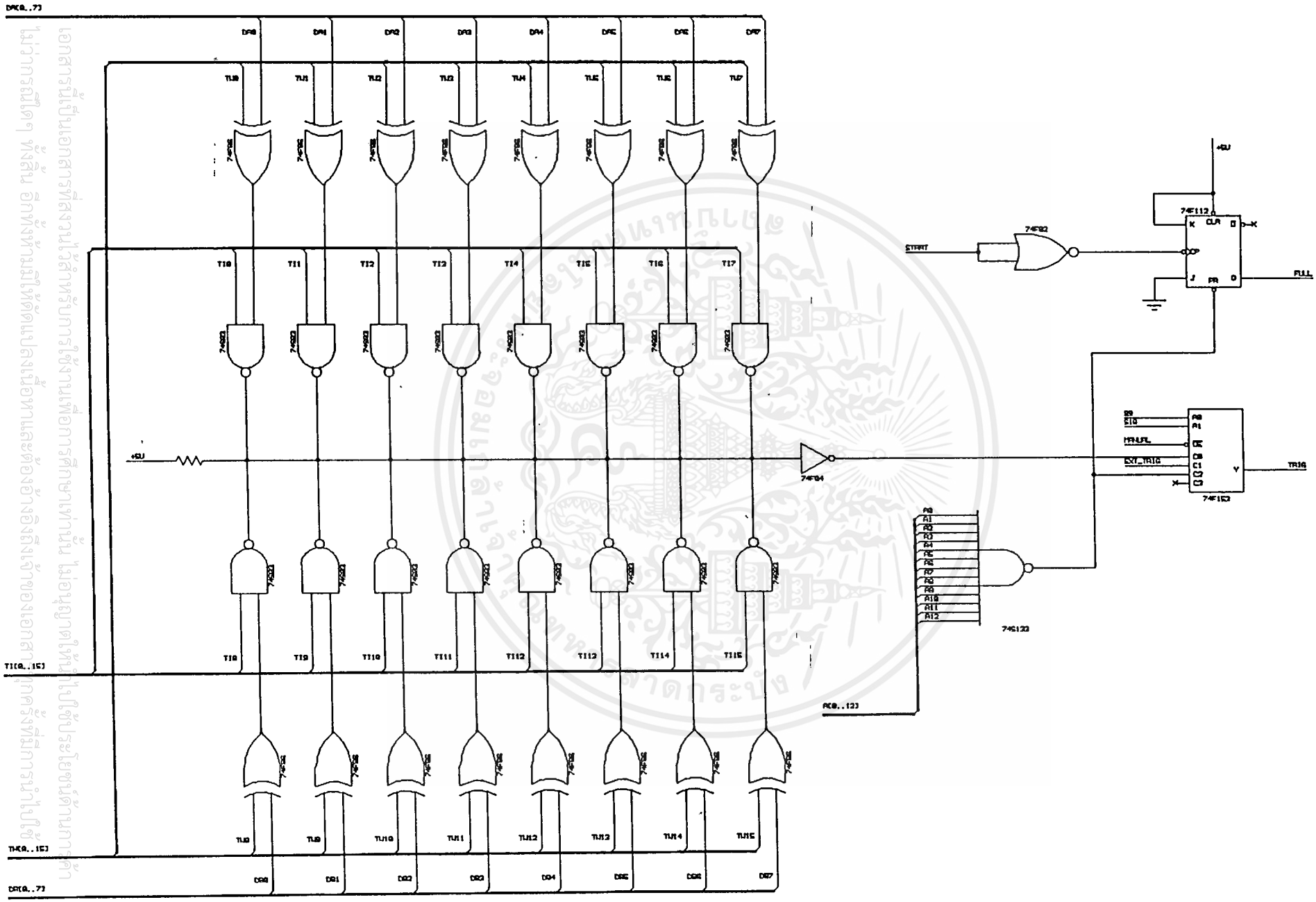
สัญญาณทริกสามารถเลือกได้ 4 แบบ ด้วยสัญญาณ S9 และ S10 ดังนี้

S10	S9	แหล่งสัญญาณทริก
0	0	Internal
0	1	External
1	0	NO TRIG
1	1	Don't Care

สัญญาณทริกภายใน (Internal) เกิดจากการเปรียบเทียบข้อมูลอินพุทกับทริกเกอร์เวิร์ด (TW0-TW15) แบบบิทต่อบิท ถ้าทุกบิทเหมือนกันจะทำให้สัญญาณทริกเป็น 0 แต่ถ้าไม่ต้องการเปรียบเทียบทุกบิทก็สามารถทำให้เป็น don't care ได้ โดยการเซ็ต TINH สำหรับบิทนั้นให้เป็น 0

สัญญาณทริกภายนอกรับมาจากวงจรภายนอก ส่วนสัญญาณ MANUAL นั้นมาจากเอาท์พุทพอร์ท ในกรณีที่เลือก NO TRIG การสุ่มข้อมูลจะหยุดต่อเมื่อข้อมูลเต็มเต็มหน่วยความจำ จึงต้องนำสัญญาณแอดเดรสมาแนนด์กันเป็นสัญญาณทริก นอกจากนี้เมื่อมีการหยุดสุ่มข้อมูลเกิดขึ้นโปรแกรมมีความจำเป็นที่จะต้องรู้ว่าเกิดการหยุดเมื่อข้อมูลเก็บเลยขนาดของหน่วยความจำหรือไม่ จึงต้องสร้างสัญญาณ FULL ขึ้นมา เมื่อสัญญาณนี้เป็น 1 แสดงว่าหน่วยความจำเต็ม เมื่อมีการสุ่มข้อมูลใหม่ สัญญาณ START จะถูกใช้ในการเคลียร์สัญญาณนี้ให้เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

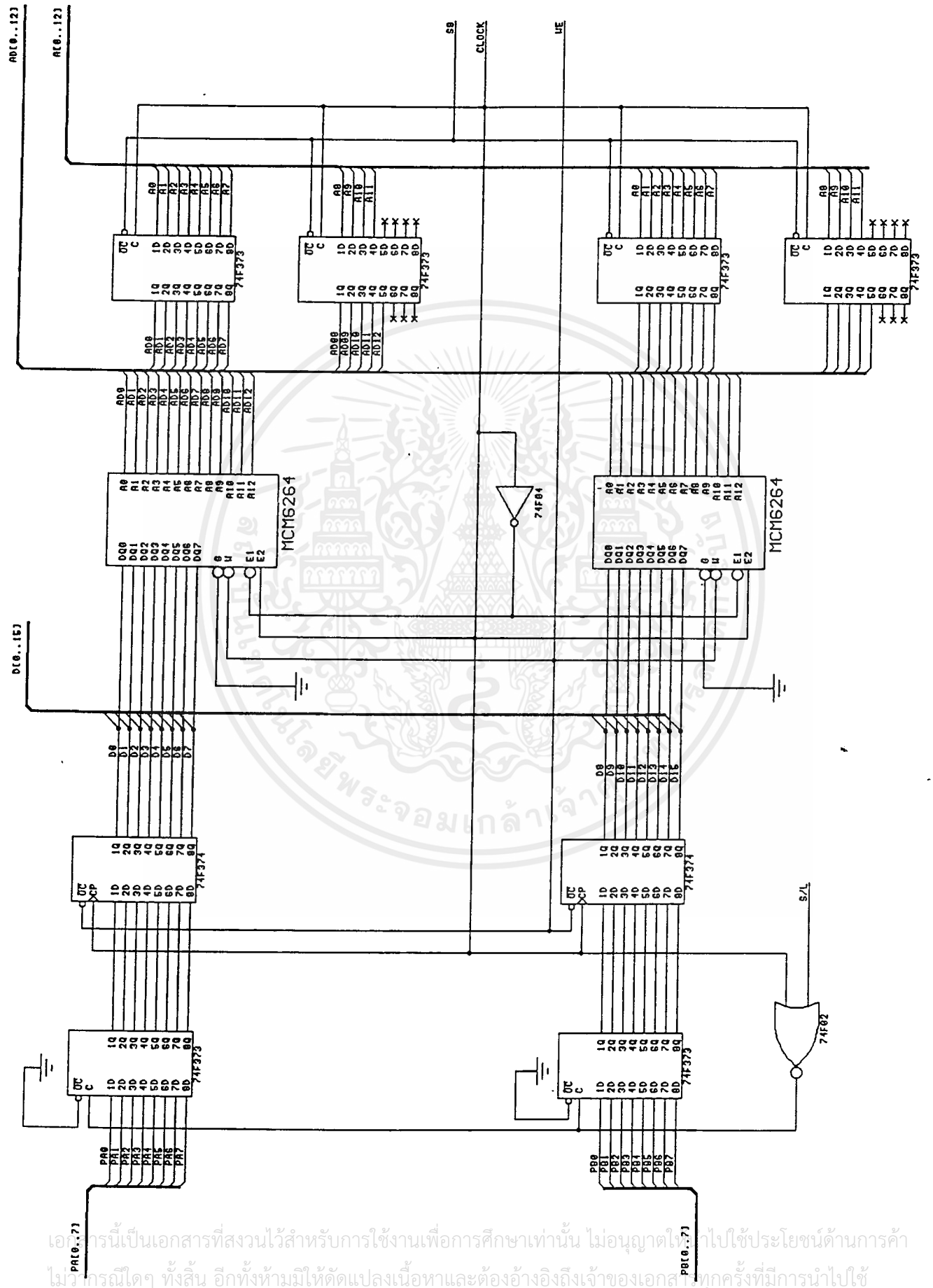

DR[0..7]
TI[0..15]
TH[0..15]
+5V
START
74F112
74F32
74F04
74F153
74S133
FULL
TRIG
EXT_TRIG

3.7 หน่วยความจำ

สัญญาณ S/L ใช้เลือกแบบในการสุ่มข้อมูล ถ้า S/L เป็น 1 ก็จะเป็นแบบแซมเปิล ตัว 74F373 จะทำตัวเป็นทางผ่านของสัญญาณเท่านั้น แต่เมื่อ S/L เป็น 0 จะทำงานแบบแลทช์ 74F373 และ 74F374 จะทำงานร่วมกันเป็นมาสเตอร์-สลาฟ ฟลิปฟลอป ซึ่งทำให้สามารถจับสัญญาณพัลซ์ที่มีขนาดเล็กกว่าสัญญาณนาฬิกาได้

ในการอ่านข้อมูลจาก RAM เพื่อนำไปเก็บลงบนหน่วยความจำของ IBM PC จะต้องเซ็ตให้ S8 เป็น 1 เพื่อที่จะดิสเอเบิลสัญญาณแอดเดรสจากลอจิกอนาไลเซอร์ แล้วจึงส่งสัญญาณแอดเดรสมายัง AD0..AD11 และอ่านข้อมูลจาก AD0..AD7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Super VGA

จอภาพในสมัยแรกๆ ที่เกิดมาพร้อมกับคอมพิวเตอร์เป็นจอโทรทัศน์ธรรมดา โดยเครื่องคอมพิวเตอร์จะมีวงจรแปลงสัญญาณภาพให้เหมาะสมกับเครื่องรับโทรศน์ทั่วไปเพื่อต้องการให้ต้นทุนถูก และลดการออกแบบ แต่จอโทรทัศน์ก็ดูไม่เหมาะสมนัก เนื่องจากมีหลายส่วนของโทรทัศน์ที่จะไม่ได้ถูกใช้งานเลย รวมถึงจำนวนจุดบนจอโทรทัศน์ก็ยังน้อยเกินไปที่จะแสดงตัวอักษรมากๆ

เมื่อ IBM ได้เป็นผู้ผลักดันให้เครื่อง XT เป็นที่แพร่หลายก็พร้อมกับจอสีเดียวที่เรียกกันว่า Monochrome Display (MD) และการ์ดแสดงผลแบบสีเดียว (MDA) ซึ่งจะแสดงผลได้เฉพาะตัวอักษรเท่านั้น แต่ด้วยความละเอียดสูงถึง 720*350 จุด ทำให้การ์ดชนิดนี้สามารถแสดงผลได้สูงถึง 80 ตัวอักษร จำนวน 25 บรรทัด

แต่ข้อเสียใหญ่ของจอประเภทนี้ก็คือ ไม่สามารถแสดงจุดภาพในตำแหน่งใดๆ ตามที่ผู้ใช้ต้องการได้ จึงมีหลายบริษัทได้พยายามคิดค้นการ์ดแสดงผลที่สามารถแสดงภาพได้ จากการ์ดแสดงผลแบบขาวดำก็ได้พัฒนาต่อมาจนเป็น Color Graphics Adapter (CGA) ซึ่งมีความละเอียดของจุดภาพสูงสุดถึง 320*200 จุด ในการแสดงภาพครั้งละสี่สี และ Enhance Graphics Adapter (EGA) ซึ่งมีความละเอียดสูงถึง 640*350 จุดในการแสดงภาพครั้งละ 16 สีซึ่งมากกว่าเป็น 2 เท่าของจอแบบ CGA

จอภาพที่กล่าวมาทั้งหมดนี้ แสดงผลจากสัญญาณดิจิตอลทั้งหมด สีที่ได้จากจอภาพจึงมีจำนวนสีจำกัด และไม่สามารถใช้จอผิดชนิดกันได้ ทางแก้ไขก็คือ หันมา
พัฒนาจอที่รับสัญญาณ แบบอนาล็อกแทนทำให้เราสามารถพัฒนาจอและการ์ดแสดงผลให้เป็นไปอย่างมีอิสระมากขึ้น และจอที่มีความละเอียดสูงกว่าก็จะแสดงผลจากการ์ดที่มีความละเอียดต่ำกว่าได้

การ์ดชนิดแรกที่ให้สัญญาณแบบอนาล็อกก็คือ Video Graphics Array หรือ VGA แสดงผลบน Analog Diaplay ซึ่งมีทั้งสีและขาวดำ ในแบบขาวดำจะต่างจากแบบสีตรงที่ใช้สัญญาณจากสีใดสีหนึ่งเพื่อใช้ในการแสดงภาพ เนื่องด้วยความจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบการ์ดและความสามารถในการเปลี่ยนข้อมูลจาก Digital เป็น Analog จึงทำให้การ์ดชนิดนี้ แสดงผลด้วยจำนวนสี 256 สูงสุด ในความละเอียด 320*200 จุด แต่ถ้าต้องการที่จะเพิ่มรายละเอียดก็ทำได้โดยลดจำนวนสี เหลือเพียง 16 สี ใน ความละเอียด 640*480 จุด

ถึงกระนั้นเทคโนโลยีทางการแสดงภาพก็ไม่ได้หยุดนิ่งมีการ์ดแสดงผลและจอที่มีความละเอียดสูงขึ้นอีกมาก เช่น การ์ดที่ใช้ในเครื่อง IBM รุ่น PS/2 ซึ่งใช้จอชนิด 8514/a , การ์ดแสดงผลที่ใช้กับจอ Super VGA ที่มีความละเอียดถึง 1024*768 จุดในสีจำนวนสีถึง 256 สี

ความสามารถของ VGA และ SVGA

ทั้ง VGA และ Super VGA ต่างก็มีการแสดงผลเป็นสองประเภทหลัก คือ text mode และ Graphic mode เฉพาะใน graphic mode ของ VGA จะสามารถทำงานได้หลายระดับดังแสดงในตารางข้างล่าง

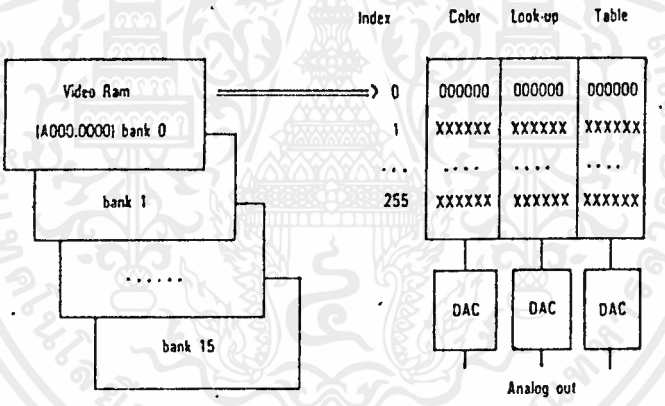
ระดับการแสดงภาพ	โหมด	ความละเอียดของภาพ	จำนวนสี
CGA	4,5	320*200	4
CGA	13	320*200	16
CGA	14	640*200	16
VGA (Mono)	15	640*350	2
EGA	16	640*350	16
VGA	17	640*480	2
VGA	18	640*480	16
VGA	19	320*200	256
SVGA	45	640*350	256
SVGA	47	640*400	256
SVGA	46	640*480	256
SVGA	48	800*600	256
SVGA	56	1024*768	256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงภาพของ VGA

ข้อมูลภาพซึ่งหมายถึงจุดแต่ละจุดที่จะแสดงบนจอภาพนั้น จะเก็บไว้ในในส่วนของหน่วยความจำที่เรียกว่า Video RAM ในจอ VGA และ SVGA จะอยู่ที่แอดเดรส A000:0000 ซึ่งจะกินพื้นที่เพียง 64 Kbytes แต่จะมีหลายชุด (page) ซ้อนกัน

ในการ์ดแสดงผลแบบ SVGA จะมีได้สูงสุดถึง 1 Mbytes หรือ มีจำนวน 16 pages ด้วยกัน การแสดงภาพในโหมดต่างๆจะใช้พื้นที่หน่วยความจำนี้ไม่เท่ากัน โดยจะใช้พื้นที่ 1 ไบต์ต่อข้อมูล 1 จุด ด้วยขนาดของข้อมูล 1 ไบต์นี้จะทำให้สามารถแสดงค่าสีได้ 256 สีด้วยกัน ในการแสดงภาพขนาด 1024*768 อันเป็นระดับความละเอียดสูงสุดจะใช้พื้นที่ หน่วยความจำถึง 12 page



ข้อมูลใน Video RAM จะถูกนำไปเปิดตารางหาค่าความเข้มของแม่สีแต่ละสีที่เก็บไว้ในตารางเทียบสี (Color Look-up Table) ตารางดังกล่าวมีขนาด 256 ชุดสี ใน แต่ละชุดประกอบไปด้วยค่าความเข้มของสี 3สี กินเนื้อที่สีละ 6 บิต ข้อมูลที่ได้มาจากการเปิดตารางจะถูกเปลี่ยนแปลงให้เป็นค่าสัญญาณอนาล็อก โดย DAC (Digital to Analog Converter) ก่อนจะส่งให้จอภาพแสดงผลต่อไป

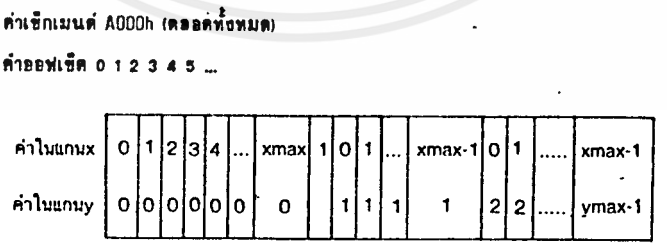
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลใน 1 เมกะไบท์ของวิดีโอแรม

ในจอ VGA ข้อมูลที่ใช้แสดงภาพกราฟฟิกทั้งหมดจะเก็บอยู่ใน video RAM ที่ bank 0 ทั้งหมด เมื่อเราคำนวณขนาดของหน่วยความจำที่จะต้องใช้ในการแสดงภาพก็จะได้เป็น 320*200 ซึ่งได้ผลลัพธ์เท่ากับ 640000 ไบท์ (ในแต่ละไบท์จะมีขนาด 8 บิท ซึ่งเป็นขนาดที่จอ VGA ใช้เก็บภาพ 1 จุดพอดี) ขนาดดังกล่าวน้อยกว่าขนาดของ Video RAM ในแต่ละ bank นิดหน่อย คือ 65536 ไบท์

เนื่องจากพื้นที่ที่เป็น Video RAM ของคอมพิวเตอร์ในตระกูล XT และ AT มีขนาดเพียง 64 กิโลไบท์เท่านั้น เมื่อเราต้องการจะแสดงภาพขนาด 640*400 ซึ่งจะต้องใช้ เนื้อที่ในการเก็บถึง 256 กิโลไบท์จึงเป็นไปไม่ได้เพราะจะเข้าไปซ้อนทับกับพื้นที่ BIOS ของระบบ ดังนั้น ทางออกที่ผู้ผลิตการ์ดแสดงผลออกแบบไว้ก็คือใช้วิธีกำหนดให้มีหน่วยความจำขนาด 64 กิโลไบท์หลายๆ ชุดซ้อนทับกันอยู่ในตำแหน่งแอดเดรสเดียวกัน การเลือกที่จะติดต่อกับชุดหน่วยความจำแต่ละชุดหรือแต่ละ bank นั้นจะกระทำได้โดยการใส่ค่าหมายเลข bank ไว้ในรีจิสเตอร์สำหรับอ้าง bank ที่ต้องการ รีจิสเตอร์ดังกล่าวเราสามารถ ติดต่อได้โดยผ่านพอร์ทหมายเลข 3CDH โดยการเขียนค่า bank ที่ต้องการลงในรีจิสเตอร์แล้วจึงอ่านหรือเขียนข้อมูลลงใน Video RAM ได้ตามที่เราต้องการ

ภายใน Video RAM จะเป็นข้อมูลจุดเรียงกันไป โดยเริ่มจากบรรทัดที่ 0 แถวที่ 0 แถวที่ 1 ไปเรื่อยๆ จนสุดหน้าจอของแต่ละโหมดซึ่งมีแตกต่างกันไปในแต่ละโหมด แล้วต่อ ด้วยบรรทัดที่ 1 บรรทัดที่ 2 บรรทัดที่ 3 เรื่อยไปจนสุดบรรทัดดังที่แสดงไว้ในแผนภาพ



ค่า xmax คือ ค่าจำนวนจุดในแนวนอนและค่า ymax คือค่าจำนวนจุดในแนวตั้งในโหมดการแสดงผลนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรีจิสเตอร์สำหรับกำหนด page นี้ จะแบ่งออกเป็นสองส่วน คือ สี่บิทต่ำจะทำหน้าที่สำหรับอ้าง page ที่ต้องการในขณะที่เขียนข้อมูลลงใน Video RAM โดยจะต้องเซ็ตบิทที่ 5 หรือบิทที่อยู่ถัดไปให้เป็น 1 เสมอด้วย ส่วนบิทอื่นๆ (3 บิทที่เหลือ) ให้เซ็ตเป็น 0 ทั้งหมด ดังนั้นการอ้างถึงข้อมูลใดๆ ใน Video RAM เพื่อการเขียนข้อมูลจะมีสูตรดังนี้

ค่าในรีจิสเตอร์ = ((x+y*ymax)/65536) OR 40h

ค่าออฟเซ็ต = (x+y*ymax) AND 0ffffh

เมื่อต้องการอ่านข้อมูล ก็จะกระทำเช่นเดียวกับการเขียนข้อมูล แต่ว่าการอ่านข้อมูลจะต้องกำหนดค่าในรีจิสเตอร์ 4 บิทบน ส่วนใน 4 บิทล่างให้เซ็ตเป็น 0 ดังนั้นการอ่านข้อมูลจึงมีสูตรดังนี้

ค่าในรีจิสเตอร์ = (((x+y*ymax)/65536) OR 40h)*16

ค่าออฟเซ็ต = (x+y*ymax) AND 0ffffh

เมื่อจะโปรแกรมในภาษา C การคูณหรือหารจะเปลี่ยนเป็นการเลื่อนบิทรีจิสเตอร์ไปทางขวาหรือซ้ายแทน ซึ่งจะทำให้การคำนวณรวดเร็วกว่าการคูณหรือหาร ดังนี้

ค่าในรีจิสเตอร์สำหรับการเขียน

```
(int)((long)(x+y*ymax)>>16)|0x40
```

ค่าในรีจิสเตอร์สำหรับการอ่าน

```
(int)((long)(x+y*ymax)>>16)|0x40)<<4
```

ค่าออฟเซ็ต

```
(x+y*ymax)&0x0ffff
```

ดังนั้นเมื่อการ์ดแสดงผลที่ใช้อยู่มีหน่วยความจำ 1 MB แสดงว่าการ์ดนั้นมีหน่วยความจำ 16 bank เต็ม ซึ่งในโหมดสูงสุดคือ โหมด 1024*768 จะใช้เท่ากับ 12 bank การแสดงโหมดดังกล่าวจึงไม่สามารถทำได้หากการ์ดนั้นมีหน่วยความจำเพียง 512 kB

สำหรับโครงงานนี้จะใช้งานในโหมด 1024*768 ซึ่งเป็นโหมดที่มีความละเอียดสูงที่สุด (โหมด 56h) และยังเป็นโหมดที่มีความเร็วในการโปรแกรมที่สูงที่สุดอีกด้วยเพราะว่าในการใช้งานตามมาตรฐาน VGA แบบปกติ เมื่อต้องการที่จะลากเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงไม่ว่าในแนวใดๆ ก็ตามจะใช้วิธีการพล็อตครั้งละจุดไปจนครบเส้นตรงหนึ่งเส้นและสำหรับการพล็อตในแต่ละจุดนั้นก็จะต้องทำการเลือก page ก่อน จึงเป็นการเสียเวลามาก แต่ถ้าเปลี่ยนมาใช้งานในโหมด 1024*768 นี้สำหรับทุกๆ จุดที่อยู่ในเส้นแนวนอนเส้นเดียวกันนั้นจะอยู่ใน page เดียวกันตลอด เพราะฉะนั้นเราจึงเลือก page เพียงครั้งเดียวแล้วจึงพล็อตเส้นตรงตามแนวนอนในครั้งเดียวเลย ซึ่งจะเพิ่มความเร็วให้กับการลากเส้นกราฟในแนวนอนมาก แต่ในการลากเส้นตรงในแนวดิ่งจะเหมือนกับการลากเส้นตรงในโหมด VGA ตามปกติ

จากที่กล่าวมาแล้ว เราจะแบ่งฟังก์ชันในการลากเส้นออกเป็น 2 ฟังก์ชันคือ ฟังก์ชันที่ใช้ลากเส้นในแนวนอน (hline) และฟังก์ชันที่ใช้ลากเส้นในแนวดิ่ง (vline) ดังนี้

```
void hline(int x1, int y, int x2, unsigned char color)
{
    register int i, t;

    if (x1>x2) {
      t = x1;  x1 = x2;  x2 = t;
    }
    outp(0x3cd, 0x40+y/64);
    for (i=x1; i<=x2; i++)
    pokeb(0xa000, y*1024U+i, color);
}
```

ฟังก์ชัน hline สำหรับลากเส้นตรงในแนวนอน

```
void vline(int y1, int x, int y2, unsigned char color)
{
    register int i, t;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (y1>y2) {
        t = y1;  y1 = y2;  y2 = t;
    }
    for (i=y1; i<=y2; i++)  {
          outp(0x3cd, 0x40+i/64);
          pokeb(0xa000, i*1024U+x, color);
    }
}
```

ฟังก์ชัน vline สำหรับลากเส้นตรงในแนวดิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/*    Logic Analyzer Menu Project                                        */
/*    Run on TSENG LAB ET4000  SVGA Adapter                              */
/*        Resolution  1024*748 pixels                                    */
/*    Written by                                                         */
/*        Sorapong Chotipanich  4J  321361                               */
/*        Sanya    Yimsiri      4J  321366                               */
/*    Finish at 13 May 1993                                              */
/*************************************************************************/


#include <mem.h>
#include <dos.h>
#include <time.h>
#include <alloc.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

#define  PAGEUP                18688
#define  PAGEDOWN              20736
#define  LEFT_ARROW            19200
#define  RIGHT_ARROW           19712
#define  UP_ARROW              18432
#define  DOWN_ARROW            20480
#define  ENTR                   7181
#define  ESC                     283
#define  HOME                  18176
#define  END                   20224
#define  CTRL_F                 8454
#define  CTRL_L                 9740
#define  CTRL_P                 6416
#define  CTRL_T                 5140
#define  CTRL_C                11779

#define  heigh                    25
#define  space                    55
#define  yinit                    68
#define  x_text                    7
#define  y_text                    3
#define  numwave                  16
#define  PLOT                      1
#define  CLEAR                     0
#define  MANUAL                    1
#define  AUTO                      0
#define  bit1                   0x01
#define  linegrid                  4
#define  xinit              (x_text+13)*8
#define  endx               (x_text+13)*8 + space*(numwave-1) + 35


/* MENUTYPE is the structure of the pulldown menu */

struct MENUTYPE {
                  int loc,active,item_num,maxlen;
                  char title[20];
                  char item[7][20];
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* PARAM is the structure of parameter sendind */

struct   PARAM {
                unsigned int wordtrig,wordenable,wordelay,startword,wordstop;
                float threshold_level;
};

/* LOGIC is the structure of view function in graphic mode */

struct   LOGIC {
                int logiccolor[16],maincolor,gridcolor,mask,prev,word;
                unsigned int data[16][numwave];
                char fname[20],charbase[3];
                int saveimage1[530],saveimage2[530];
                int pos1,pos2,timebase;
};

struct MENUTYPE menus[6];
struct PARAM parameter;
struct LOGIC logic_data;

unsigned int data_in[4096],dummy[4096];
int cond;
char channel_name[16][16];

initial()                              /* initial all defult value */
{
  int i;

  parameter.startword = 0xffff;
  parameter.wordtrig = 0xffff;
  parameter.wordenable = 0xffff;
  parameter.wordelay = 0x10;
  parameter.threshold_level = 1.50;

  strcpy(menus[0].title," File ");
  strcpy(menus[1].title," Freq Sampling ");
  strcpy(menus[2].title," Parameter ");
  strcpy(menus[3].title," Run ");
  strcpy(menus[4].title," F1 HELP ");

  menus[0].item_num = 5;
  menus[0].active = 0;
  menus[0].maxlen = 10;
  strcpy(menus[0].item[0]," Load ");
  strcpy(menus[0].item[1]," Save ");
  strcpy(menus[0].item[2]," View ");
  strcpy(menus[0].item[3]," OS Shell ");
  strcpy(menus[0].item[4]," Quit ");

  menus[1].item_num = 1;
  menus[1].active = 0;
  menus[1].maxlen = 8;
  strcpy(menus[1].item[0]," Clock ");

  menus[2].item_num = 6;
  menus[2].active = 0;
  menus[2].maxlen = 19;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void locate(int x, int y)
{
   _AH = 2;
   _BH = 0;
   _DH = y;
   _DL = x;
   geninterrupt(0x10);
}


/*
   getpixel() function is used for get one pixel color in location specfied
*/

unsigned getpixel(int x, int y)
{
  outp(0x3cd, y/4);
  return (int) peekb(0xa000, y*1024U+x);
}


/*
  putpixel() function is used for set one pixel on screen to be a color
  specfied
*/

void putpixel(int x, int y, int color)
{
 outp(0x3cd, y/64);
 pokeb(0xa000, y*1024U+x, (char) color);
}


/* gprintf() function is like printf() function */

int cdecl gprintf(char color, char *fmt, ... )
{
 va_list argptr;
 char str[140];
 int cnt, i;

 va_start(argptr, fmt);
 cnt = vsprintf(str, fmt, argptr);
 for (i=0; i<cnt; i++)
 {
    _AL = str[i];
    _AH = 0x0e;
    _BL = color;
    _BH = 0;
    geninterrupt(0x10);
 }
 va_end(argptr)
 return(cnt);
}


/* draw dash line in vertical */

 vdline(int y1, int x, int y2, unsigned char color)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    register int i, t;

    if (y1>y2)
    {
       t = y1;  y1 = y2;  y2 = t;
    }
    for (i=y1; i<=y2; i++)
    {
        outp(0x3cd, 0x40+i/64);
        pokeb(0xa000, i*1024U+x, color);
        if ( (i%5)==0 ) {
           i+=linegrid;
        }
    }
 }


 /* hline() function is a fast draw  line in horizontal */

 void hline(int x1, int y, int x2, unsigned char color)
 {
   register int i, t;

   if (x1>x2)
   {
      t = x1;  x1 = x2;  x2 = t;
   }
   outp(0x3cd, 0x40+y/64);
   for (i=x1; i<=x2; i++)
       pokeb(0xa000, y*1024U+i, color);
 }


 void vline(int y1, int x, int y2, unsigned char color)
 {
   register int i, t;

   if (y1>y2)
   {
      t = y1;  y1 = y2;  y2 = t;
   }
   for (i=y1; i<=y2; i++)
   {
       outp(0x3cd, 0x40+i/64);
       pokeb(0xa000, i*1024U+x, color);
   }
 }


 void bar256(int x1,int y1,int x2 ,int y2,int color)
 {
  register int i;

  for (i=y1;i<=y2;i++)
      hline(x1,i,x2,color);
 }


 void boxup(int x1, int y1, int x2 , int y2,
          int maincolor, int brightcolor,int lowcolor)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      bar256(x1+1,y1+1,x2-1,y2-1,maincolor);
      hline(x1,y1,x2,brightcolor);
      hline(x1+1,y1+1,x2-1,brightcolor);
      vline(y1,x1,y2,brightcolor);
      vline(y1+1,x1+1,y2-1,brightcolor);
      hline(x1+1,y2-1,x2-1,lowcolor);
      hline(x1,y2,x2,lowcolor);
      vline(y1+1,x2-1,y2-1,lowcolor);
      vline(y1,x2,y2,lowcolor);
    }


    void boxerase(int x1, int y1, int x2, int y2)
    {
      int i;

      for ( i=y1 ; i<=y2 ; i++ )
          hline (x1,i,x2,0);
    }


    /* write string on screen and wait for a key to delete */

    void wrt_str_erase(int x, int y, char c[20], int color)
    {
      int length;

      length = strlen(c);
      boxup((x-4)*8,(y-1)*16+8,(x+length+4)*8,(y+1)*16+8,0,15,7);
      locate(x,y);
      gprintf(color,"%s",c);
      getch();
      boxerase((x-4)*8,(y-1)*16+8,(x+length+4)*8,(y+1)*16+8);
    }


    /* only write string on screen */

    void wrt_str(int x, int y, char c[20], int color)
    {
      int length;

      length = strlen(c);
      boxup((x-4)*8,(y-1)*16+8,(x+length+4)*8,(y+1)*16+8,0,15,7);
      locate(x,y);
      gprintf(color,"%s",c);
    }


    void pulldown(int order)
    {
      int y_loc,tmp,i,loop;
      int lightcolor,brightcolor,maincolor;

      y_loc = 4;
      for ( loop=0 ; loop<2 ; loop++ ) {
          if ( loop==0 ) {
              lightcolor = brightcolor = maincolor = 0;
              boxup(menus[logic_data.prev].loc *8 ,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                (y_loc+2)*16+5 ,
                (menus[logic_data.prev].loc+menus[logic_data.prev].maxlen+2)*8,
                (y_loc+4+menus[logic_data.prev].item_num)*16-5 ,
                maincolor,brightcolor,lightcolor);
        }
        else {
           maincolor = 0;
           brightcolor = 15;
           lightcolor = 7;
           boxup(menus[order].loc *8 ,
                (y_loc+2)*16+5 ,
                (menus[order].loc+menus[order].maxlen +2)*8 ,
                (y_loc+4+menus[order].item_num)*16-5 ,
                maincolor,brightcolor,lightcolor);
        }
    }
    draw_text(order);
}


draw_text(int order)                    /* draw text in pulldown menu */
{
    int i,y_loc,color;

    y_loc=7;

    for ( i=0 ; i<menus[order].item_num ; i++,y_loc++ ) {
        locate(menus[order].loc +1,y_loc);
        if ( i==menus[order].active )
           color = 3;
        else
           color = 14;
        gprintf(color,"%s",menus[order].item[i]);
    }
}


void pulldown_update(int order)         /* update the entire menu */
{
    locate(menus[logic_data.prev].loc,4);
    gprintf(11,"%s",menus[logic_data.prev].title);
    locate(menus[order].loc,4);
    gprintf(14,"%s",menus[order].title);
    pulldown(order);
}


/* header() draw a top bar in the screen and is a part of pulldown menu */

header()
{
    int i,j,x,y;

    x = 2;
    y = 4;
    logic_data.maincolor = 12;
    boxup((x-1)*8,(y-1)*16,1022,(y+2)*16,12,15,7);
    x += 3;

    for ( i=0 ; i<4 ; i++ ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        menus[i].loc = x;
        locate(x,y);
        gprintf(11,menus[i].title);
        x += strlen(menus[i].title);
        x += 5;
    }

    j = strlen(menus[i].title);
    locate(126-j,y);
    gprintf(11,menus[i].title);
    pulldown(0);
    draw_text(0);
    locate(menus[0].loc,y);
    gprintf(14,"%s",menus[0].title);
}


footer()
{
    boxup((x_text-3)*8,(y_text+35)*16+20,endx,747,0,15,7);
    locate(7,40);  gprintf(11,"TRIGGER WORD   : %X",parameter.wordtrig);
    locate(7,42);  gprintf(11,"TRIGGER ENABLE : %X",parameter.wordenable);
    locate(7,44);  gprintf(11,"TRIGGER DELAY  : %X",parameter.wordelay);
    locate(42,40); gprintf(11,"TRIGGER TYPE   : LATCH    ");
    locate(42,42); gprintf(11,"TRIGGER SOURCE : INTERNAL");
    locate(42,44); gprintf(11,"THRESHOLD LEVEL   : %3.1fV",
                              logic_data.threshold_level);
}


void getfilename(char title[20],char name[20],int color)
{
  int i,j,x,y;
  char  c;

  x = 20;
  y = 15;
  boxup((x-1)*8,(y-1)*16+8,(x+50)*8,(y+2)*16-8,0,15,7);
  locate(x,y);
  gprintf(11,"%s",title);
  j = strlen(title);
  x+=j;
  for ( i=0,cond=0 ; i<20 && cond<20 ; i++,x++ ) {
      c = getch();
      name[i]=c;
      locate(x,y);
      gprintf(color,"%c",c);
      if ( c=='\r' )
         cond = 20;
      if ( c=='\x1B' ) {                  /* ESC key */
         x-=i;
      i=-1;
         locate(x,y);
         gprintf(0,"                       ");
      }
      if ( c=='\b' ) {                    /* BACK SPACE key */
         x--;
         i-=2;
         locate(x,y);
         gprintf(0," ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            x--;
          }
      }
      i--;
      name[i]='\0';
      strcat(name,".sav");
      x=20,y=15;
      boxerase((x-1)*8,(y-1)*16+8,(x+50)*8,(y+2)*16-8);
   }


   copydata(int num)
   {
     int index,item;
     for ( index=0 ; index<numwave ; index++ ) {
         for ( item=0 ; item<16 ; item++ ) {
             logic_data.data[item][index] = ((data_in[index+num])>>item)&bit1;
         }
     }
   }


   slide_left(int type,int pos)
   {

     int i,j,x,y;

     x = xinit + (pos+1) * space;
     y = (y_text-1) * 16;

     for ( i=0 ; i<530 ; i++,y++ ) {
         if ( type==0 )
            putpixel(x,y,logic_data.saveimage1[i]);
         else
            putpixel(x,y,logic_data.saveimage2[i]);
     }
     x-=space;
     y=(y_text-1)*16;
     for ( i=0 ; i<530 ; y++,i++ ) {
         if ( type==0 )
            logic_data.saveimage1[i] = getpixel(x,y);
         else
            logic_data.saveimage2[i] = getpixel(x,y);
     }
     y = (y_text-1) * 16;
     vline((y_text-1)*16,x,(y_text+32)*16,2);
}


slide_right(int type,int pos)
{
   int i,j,x,y;

   x = xinit+(pos-1) * space;
   y = (y_text-1) * 16;


   for ( i=0 ; i<530 ; i++,y++ ) {
       if ( type==0 )
          putpixel(x,y,logic_data.saveimage1[i]);
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
          putpixel(x,y,logic_data.saveimage2[i]);
    }
    x+=space;
    y = (y_text-1) * 16;
    for ( i=0 ; i<530 ; y++,i++ ) {
        if ( type==0 )
           logic_data.saveimage1[i] = getpixel(x,y);
        else
           logic_data.saveimage2[i] = getpixel(x,y);
    }
    y = (y_text-1) * 16;
    vline((y_text-1)*16,x,(y_text+32)*16,2);
}


wrt_time_interval()
{
    unsigned int  i;

    i = (logic_data.pos2-logic_data.pos1) * logic_data.timebase;
    locate(x_text+75,42);
    gprintf(13,"TIME INTERVAL : %d %s     ",i,logic_data.charbase);
}


change_signal_name()
{
   int i,j,x,check,temp;
   char name[16],num[4];

   boxup((x_text-3)*8,(y_text+35)*16+20,endx,747,0,15,7);

   for ( check=100 ; check>0 ; ) {
       x = x_text+15;
       locate(x,40);
       gprintf(14,"Enter signal number (1-16) :                    ");
       x+=30;
       for ( i=0 ; i<4 ; i++ ) {
           locate(x,40);
           num[i] = getch();
           gprintf(14,"%c",num[i]);
           x++;
           if ( num[i]=='\r' ) {
              num[i] = '\0';
              i=4;
           }
       }
       temp = atoi(num);
       if ( (temp>0) && (temp<17) )
          check=0;
   }
   x = x_text+15;
   locate(x,42);
   gprintf(13,"Enter name for channel %d :          ",temp);
   x+=28;
   for ( check=100,j=0 ; (j<9)&&(check>0) ; j++,x++ ) {
       name[j] = getch();
       locate(x,42);
       gprintf(14,"%c",name[j]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if ( name[j]=='\r' ) {
           check = 0;
        }
    }
    j--;
    for ( ; j<9 ; j++ ) {
        name[j]=' ';
    }
    name[j] = '\0';
    strupr(name);
    strcpy(channel_name[temp-1],name);
    locate(x_text,2*temp+1);
    gprintf(11,"%s",channel_name[temp-1]);
    footer();
}


border_setup()
{
    int i,y,x;

    for ( i=0 ; i<16 ; i++ ) {
        logic_data.logiccolor[i] = 15;
    }

    logic_data.gridcolor = 14;

    boxup((x_text-3)*8,(y_text-1)*16,endx,(y_text+35)*16,
          logic_data.maincolor,15,7);
    hline((x_text-3)*8,(y_text+32)*16,endx,15);
    hline((x_text-3)*8-1,(y_text+32)*16+1,endx,7);
    hline((x_text-3)*8-2,(y_text+32)*16+2,endx,7);

    for ( i=1,y=y_text ; i<17 ; y+=2,i++ ) {
        locate(x_text,y);
        gprintf(11,channel_name[i-1]);
    }
    y++;
    locate(x_text,y);
    gprintf(14,"%s","NO.");
}


void logic_plot(int status)
{
    int y[16];
    int i,j,x,z,color;

    for ( i=0 ; i<=16 ; i++ ) {
        y[i] = yinit+32*i;
    }

    x = xinit;
    for ( i=1 ; i<numwave ; i++) {
        for ( j=0 ; j<16 ; j++) {
            if ( status==1 )
               color = logic_data.logiccolor[j];   /* select color to plot */
            else
               color = logic_data.maincolor;
            if ( logic_data.data[j][i] < logic_data.data[j][i-1] ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if ( i-1==0 ) {
               hline(x,y[j]-heigh,x+space,color);
               vline(y[j]-heigh,x+space,y[j],color);
            }
            else {
               hline(x,y[j],x+space,color);
               vline(y[j],x+space,y[j]+heigh,color);
               y[j]+=heigh;
            }
         }
         if ( logic_data.data[j][i] > logic_data.data[j][i-1] ) {
            if ( i-1==0 ) {
               hline(x,y[j],x+space,color);
               vline(y[j],x+space,y[j]-heigh,color);
               y[j]-=heigh;
            }
            else {
               hline(x,y[j],x+space,color);
               vline(y[j],x+space,y[j]-heigh,color);
               y[j]-=heigh;
            }
         }
         if ( logic_data.data[j][i] == logic_data.data[j][i-1] ) {
            if (( i-1==0 ) && ( logic_data.data[j][i]==0 )) {
               hline(x,y[j],x+space,color);
            }
            if (( i-1==0 ) && ( logic_data.data[j][i]==1 )) {
               y[j]-=heigh;
               hline(x,y[j],x+space,color);
            }
            else
               hline(x,y[j],x+space,color);
         }
      }
      x+=space;
   }
}


void logic_update(int num, int pos1, int pos2)
{
   int j,x,i,z,k;

   logic_plot(CLEAR);
   for ( j=0,x=xinit,k=num ; j<numwave ; j++,k++ ) {
      if ( k==parameter.wordstop ) {
         vline((y_text-1)*16,x,(y_text+32)*16,logic_data.maincolor);
         vline((y_text-1)*16,x,(y_text+32)*16,10);
         x+=space;
      }
      else {
         vline((y_text-1)*16,x,(y_text+32)*16,logic_data.maincolor);
         vdline((y_text-1)*16,x,(y_text+32)*16,logic_data.gridcolor);
         x+=space;
      }
    }
    copydata(num);
    logic_plot(PLOT);
    for ( j=0,x=18 ; j<4 ; j++ ) {
       locate(x,y_text+33);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gprintf(11,"%5d",num);
        x+=34;
        num+=5;
    }
    x = xinit + pos1 * space;
    z = (y_text-1) * 16;
    for ( i=0 ; i<530 ; i++,z++ ) {
        logic_data.saveimage1[i] = getpixel(x,z);
    }
    z = (y_text-1) * 16;
    vline((y_text-1)*16,x,(y_text+32)*16,2);

    x = xinit + pos2 * space;
    z = (y_text-1) * 16;
    for ( i=0 ; i<530 ; i++,z++ ) {
        logic_data.saveimage2[i] = getpixel(x,z);
    }
    z = (y_text-1) * 16;
    vline((y_text-1)*16,x,(y_text+32)*16,2);
}


view_graph()
{
  int j,k,l,num;

  logic_data.pos1 = 0;
  logic_data.pos2 = numwave-1;

  border_setup();
  logic_update(0,logic_data.pos1,logic_data.pos2);
  wrt_time_interval();

  for ( num=0,j=100 ; j>0 ; ) {
      j = bioskey(0);
      switch (j) {
         case   ESC          :  j=0;
                                break;
         case  CTRL_T        :  num=parameter.wordstop-12;
                                if ( num < 1 ) {
                                   beep(2000,400);
                                   num = 0;
                                }
                                if ( num >4095-numwave ) {
                                   beep(2000,400);
                                   num = 4095-numwave;
                                }
                                logic_update(num,logic_data.pos1,logic_data.pos2);
                                break;
         case CTRL_P         :  epson1x_dump_scr();
                                break;
         case LEFT_ARROW     :  num-=1;
                                if ( num<1 ) {
                                   num = 0;
                                   beep(1000,400);
                                }
                                logic_update(num,logic_data.pos1,logic_data.pos2);
                                break;
         case RIGHT_ARROW    :  num+=1;
                                if ( num>(4096-numwave)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        num=4096-numwave;
                        beep(1000,400);
                     }
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case HOME         :  num = 0;
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case END          :  num = 4096-numwave;
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case PAGEUP       :  num+=100;
                     if ( num>(4096-numwave)) {
                        num=4096-numwave;
                        beep(1000,400);
                     }
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case PAGEDOWN     :  num-=100;
                     if ( num<0 ) {
                        num=0;
                        beep(1000,400);
                     }
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case UP_ARROW     :  num+=20;
                     if ( num>(4096-numwave)) {
                        num=4096-numwave;
                        beep(1000,400);
                     }
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case DOWN_ARROW   :  num-=20;
                     if ( num<0 ) {
                        num=0;
                        beep(1000,400);
                     }
                     logic_update(num,logic_data.pos1,logic_data.pos2);
                     break;
case  6747        :  logic_data.pos1--;
                     if ( logic_data.pos1<0 ) {
                        logic_data.pos1 = 0;
                        beep(1000,400);
                     }
                     slide_left(0,logic_data.pos1);
                     wrt_time_interval();
                     break;
case  7005        :  logic_data.pos1++;
                     if ( logic_data.pos1==logic_data.pos2 ) {
                        logic_data.pos1--;
                        beep(1000,400);
                     }
                     slide_right(0,logic_data.pos1);
                     wrt_time_interval();
                     break;
case 10043        :  logic_data.pos2--;
                     if ( logic_data.pos2==logic_data.pos1 ) {
                        logic_data.pos2++;
                        beep(1000,400);
                     }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                slide_left(1,logic_data.pos2);
                                wrt_time_interval();
                                break;
            case 10279          :  logic_data.pos2++;
                                if ( logic_data.pos2==numwave ) {
                                   logic_data.pos2 = numwave-1;
                                   beep(1000,400);
                                }
                                slide_right(1,logic_data.pos2);
                                wrt_time_interval();
                                break;
            case  ENTR          :  logic_update(num,logic_data.pos1,logic_data.pos2);
                                break;
            case CTRL_C         :  change_signal_name();
                                break;
        }
    }
    boxerase(0,0,1023,620);
    header();
}


write_text_display(int index)
{
    int y,i,x;

    gotoxy(6,1);
    textcolor(14);
    cprintf("16 BIT    DATA[n]     DATA[n+1]    DATA[n+2]    DATA[n+3]");
    textcolor(15);

    for ( y=2 ; y<23 ; y++ ) {
        gotoxy(6,y);
        cprintf("%5d",index);
        gotoxy(21,y);
        for ( x=25,i=0 ; i<4 ; i++,index++ ) {
            gotoxy(x,y);
            if ( index==logic_data.mask )  textcolor(11);
            else  textcolor(15);
            cprintf("%5X",data_in[index]);
            x+=15;
        }
    }
}


findword(int num)
{
    unsigned int i;

    gotoxy(5,24);
    textcolor(11);
    cprintf("Enter word to find (hex) : ");
    scanf("%4x",&logic_data.word);
    for ( i=0 ; i<4096 ; i++ ) {
        if ( logic_data.word==data_in[i] ){
            logic_data.mask = i;
            gotoxy(5,24);
            cprintf("Found %X at address  %d              ",logic_data.word,i);
            getch();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            gotoxy(5,24);
            printf("                                        ");
            return (i);
        }
    }
    logic_data.mask = 10000;
    gotoxy(5,24);
    cprintf("Not found %X at any address        ",logic_data.word);
    getch();
    gotoxy(5,24);
    printf("                                          ");
    return(num);
}


findnext(int num)
{
    int i;

    logic_data.mask++;
    textcolor(11);
    for ( i=logic_data.mask ; i<4096 ; i++ ) {
        if ( logic_data.word==data_in[i] ){
            logic_data.mask = i;
            gotoxy(5,24);
            cprintf("Found %X at address  %d            ",logic_data.word,i);
            getch();
            gotoxy(5,24);
            printf("                                    ");
            return (i);
        }
    }
    logic_data.mask = 10000;
    gotoxy(5,24);
    cprintf("Not found %X at any address        ",logic_data.word);
    getch();
    gotoxy(5,24);
    printf("                                          ");
    return(num);
}


view_text()
{
    int index,i,temp;

    logic_data.mask = 10000;

    temp = 100;
    graphics_close();
    clrscr();
    write_text_display(0);
    for ( index=i=0 ; temp>0 ; ) {
        i = bioskey(0);
        switch (i) {
            case HOME       :  index = 0;
                               write_display(index);
                               break;
            case END        :  index = 4012;
                               write_display(index);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                break;
            case PAGEDOWN   :   index=index+92;
                                if ( index>4012 ) {
                                   index = 4012;
                                   beep(3000,400);
                                }
                                write_display(index);
                                break;
            case PAGEUP     :   index-=92;
                                if ( index<0 ) {
                                   index = 0;
                                   beep(3000,400);
                                }
                                write_display(index);
                                break;
            case  CTRL_F    :   index = findword(index);
                                write_display(index);
                                break;
            case  CTRL_L    :   findnext(index);
                                write_display(index);
                                break;
            case DOWN_ARROW :   index+=4;
                                if (index>4012) {
                                   beep(3000,400);
                                   index = 4012;
                                }
                                write_display(index);
                                break;
            case UP_ARROW   :   index-=4;
                                if ( index<0 ) {
                                   beep(3000,400);
                                   index = 0;
                                }
                                write_display(index);
                                break;
            case   ESC      :   temp = 0;
                                break;
        }
    }
    graphics_open();
    header();
    footer();
}


void view()
{
    char item[2][12];
    int temp,x,active,y,i,color;

    strcpy(item[0]," Graphics ");
    strcpy(item[1]," Character ");

    x=20; active = 0;

    boxup(x*8-8,10*16-8,(x+13)*8+8,(10+2)*16+8,0,15,7);
    for ( temp=100 ; temp>0 ; ) {
        for ( i=0,y=10 ; i<2 ; i++,y++ ) {
            locate(x,y);
            if ( i==active )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            color = 3;
         else
            color = 14;
         gprintf(color,"%s",item[i]);
      }
      temp = bioskey(0);
      switch (temp) {
         case ENTR        :  temp = 0;
                             break;
         case DOWN_ARROW  :  if ( active==1 )  active = 0;
                             else active++;
                             break;
         case UP_ARROW    :  if ( active==0 )  active = 1;
                             else active--;
                             break;
      }
   }
   boxerase(60*8-8,10*16-8,(60+13)*8+8,(10+2)*16+8);
   switch (active) {
         case 0  :  view_graph();
                    break;
         case 1  :  view_text();
                    break;
   }
}


/* getint() get 4 characters and convert to decimal */

getint(int x1, int y1)
{
   int x,j,temp1,temp2;
   int i;
   char input[5];

   j = 100;
   temp1 = temp2 = 0;

   for ( ; ; j=100) {
      x = x1;
      locate(x,y1);
      for ( i=0 ; i<4 ; i++ ) {
         input[i] = getch();
         x++;
         locate(x,y1);
         gprintf(14,"%c",input[i]);
         if ( input[i]=='\x1B') {       /* if ESC character */
            x = x1;
            locate(x,y1);
            gprintf(15,"     ");
            i = -1;
         }
      }
      input[i] = '\0';
      for ( temp1=0,i=0 ; i<4 ; i++ ) {
         switch (input[i]) {
            case '0' :  temp1 = 0x00;  break;
            case '1' :  temp1 = 0x01;  break;
            case '2' :  temp1 = 0x02;  break;
            case '3' :  temp1 = 0x03;  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case '4' :  temp1 = 0x04;  break;
                case '5' :  temp1 = 0x05;  break;
                case '6' :  temp1 = 0x06;  break;
                case '7' :  temp1 = 0x07;  break;
                case '8' :  temp1 = 0x08;  break;
                case '9' :  temp1 = 0x09;  break;
                case 'a' :
                case 'A' :  temp1 = 0x0a;  break;
                case 'b' :
                case 'B' :  temp1 = 0x0b;  break;
                case 'c' :
                case 'C' :  temp1 = 0x0c;  break;
                case 'd' :
                case 'D' :  temp1 = 0x0d;  break;
                case 'e' :
                case 'E' :  temp1 = 0x0e;  break;
                case 'f' :
                case 'F' :  temp1 = 0x0f;  break;
                default  :  j=0;
            }
            if ( i != 0 )  temp2 *= 0x10;
            temp2 += temp1;
        }
        if ( j==0 ) {
           beep(1000,400);
           wrt_str_erase(60,20," Error! Type Again ",11);
        }
        else {
           beep(1000,400);
           beep(4000,300);
           beep(8000,200);
           return(temp2);
        }
     }
  }


void loadfile()
{
  FILE *fp;
  int i,tmp1,tmp2;

  getfilename("  File Name : ",logic_data.fname,14);
  fp = fopen(logic_data.fname,"rb");

  if ( fp==NULL ) {
     beep(1000,500);
     wrt_str_erase(20,15," File name not found.  Cannot loading file. ",12);
  }
  else {
     fread(data_in, sizeof(int), 4096, fp);
     fclose(fp);
     beep(1000,400);
     beep(4000,300);
     beep(8000,200);
     wrt_str_erase(30,15," Successful Loading. ",14);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void savefile()
{
  FILE *fp;
  int i;

  getfilename("  File Name : ",logic_data.fname,14);
  fp = fopen(logic_data.fname,"wb");

  if ( fp==NULL ) {
     beep(1000,500);
     wrt_str_erase(20,15," Cannot open file. Data not save. ",12);
  }
  else {
     fwrite(data_in, sizeof(int), 4096, fp);
     fclose(fp);
     beep(1000,400);
     beep(4000,300);
     beep(8000,200);
     wrt_str_erase(20,15," Successful Data Saving. ",14);
  }
}


void file()
{
  switch (menus[0].active) {
     case 0 :  loadfile();
               break;
     case 1 :  savefile();
               break;
     case 2 :  view();
               break;
     case 3 :  graphics_close();
               system("c:\command.com");
               graphics_open();
               header();
               footer();
               break;
     case 4 :  graphics_close();
               gotoxy(1,25);
               printf("THANK YOU TO SEE ME");
               textcolor(15);
               exit(0);
  }
}


freq()
{
  int x,y,i,active,color,temp;

  char item[16][10];

  strcpy(item[0],"  20 MHz ");
  strcpy(item[1],"  10 MHz ");
  strcpy(item[2],"   4 MHz ");
  strcpy(item[3]," 200 KHz ");
  strcpy(item[4]," 100 KHz ");
  strcpy(item[5]," 50 KHz ");
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
strcpy(item[6],"  20 KHz ");
strcpy(item[7],"  10 KHz ");
strcpy(item[8],"   2 KHz ");
strcpy(item[9],"   1 KHz ");

x=40; y=9; active = 0;
boxup(x*8-8,y*16-8,(x+10)*8,(y+10)*16+8,0,15,7);
for ( cond=100 ; cond>0 ; ) {
    for ( i=0,y=9 ; i<10 ; i++,y++ ) {
        locate(x,y);
        if ( i==active )
           color = 3;
        else
           color = 14;
        gprintf(color,"%s",item[i]);
    }
    temp = bioskey(0);
    switch (temp) {
        case DOWN_ARROW :  if ( active==9 )  active = 0;
                           else active++;
                           break;
        case UP_ARROW   :  if ( active==0 )  active = 9;
                           else active--;
                           break;
        case ENTR       :  cond=0;
                           break;

    }
}
switch(active) {
    case  0   :  logic_data.timebase = 50;
                 strcpy(logic_data.charbase,"nS");
                 break;
    case  1   :  logic_data.timebase = 100;
                 strcpy(logic_data.charbase,"nS");
                 break;
    case  2   :  logic_data.timebase = 250;
                 strcpy(logic_data.charbase,"nS");
                 break;
    case  3   :  logic_data.timebase = 5;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  4   :  logic_data.timebase = 10;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  5   :  logic_data.timebase = 20;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  6   :  logic_data.timebase = 50;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  7   :  logic_data.timebase = 100;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  8   :  logic_data.timebase = 500;
                 strcpy(logic_data.charbase,"uS");
                 break;
    case  9   :  logic_data.timebase = 1;
                 strcpy(logic_data.charbase,"mS");
                 break;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  boxerase(40*8-8,9*16-8,(40+10)*8,(9+17)*16+8);
  locate(x_text+75,40);
  gprintf(11,"FREQUENCY      : %s",item[active]);
}


void trigger_word()
{
  int x,y;

  x=60; y=15;                          /* locate position for text */
  wrt_str(x,y," Enter Trigger Word :          ",12);
  x+=23;
  parameter.wordtrig = getint(x,y);
  boxerase(448,232,760,264);
  locate(x_text,40);
  gprintf(11,"TRIGGER WORD   : %X   ",parameter.wordtrig);
}


void trigger_enable()
{
  int temp,j,x,y,i,color,count;
  char input[4];

  x=60; y=15;
  wrt_str(x,y," Enter Trigger Enable :          ",12);
  x+=25;
  parameter.wordenable = getint(x,y);
  boxerase(448,232,800,264);
  locate(x_text,42);
  gprintf(11,"TRIGGER ENABLE : %X   ",parameter.wordenable);
}


trigger_delay()
{
  int x,y;

  x=60; y=15;
  wrt_str(x,y," Enter Trigger Delay :          ",12);
  x+=25;
  parameter.wordelay = getint(x,y);
  boxerase(448,232,800,264);
  locate(x_text,44);
  gprintf(11,"TRIGGER DELAY  : %X   ",parameter.wordelay);
}


void trigger_start()
{
  int x,y;

  x=60; y=15;
  wrt_str(x,y," Enter Start Word :          ",12);
  x+=23;
  parameter.startword = getint(x,y);
  boxerase(448,232,760,264);
  locate(42,44);
  gprintf(11,"START WORD     : %X   ",parameter.startword);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void glitch()
{
  char item[2][12];
  int temp,x,active,y,i,color;

  strcpy(item[0]," Latch ");
  strcpy(item[1]," Non Latch ");

  x=60; active = 0;

  boxup(60*8-8,10*16-8,(60+13)*8+8,(10+2)*16+8,0,15,7);
  for ( temp=100 ; temp>0 ; ) {
      for ( i=0,y=10 ; i<2 ; i++,y++ ) {
          locate(x,y);
          if ( i==active )
             color = 3;
          else
             color = 14;
          gprintf(color,"%s",item[i]);
      }
      temp = bioskey(0);
      switch (temp) {
          case ENTR       :  temp = 0;
                             break;
          case DOWN_ARROW  :  if ( active==1 )  active = 0;
                             else active++;
                             break;
          case UP_ARROW    :  if ( active==0 )  active = 1;
                             else active--;
                             break;
      }
  }
  boxerase(60*8-8,10*16-8,(60+13)*8+8,(10+2)*16+8);
  locate(x_text+35,40);
  if ( active== 0 )
     gprintf(11,"TRIGGER TYPE   : LATCH    ");
  else
     gprintf(11,"TRIGGER TYPE   : NON LATCH");
}


void threshold()
{
   int i,j;

   parameter.threshold_level = 1.5;
   boxup(450,150,900,300,11,15,7);
   hline(452,250,900,7);
   hline(450,251,898,15);
   locate(74,11);
   gprintf(14," Threshold Voltage ");
   locate(67,17);
   gprintf(14," + to increment ,, - to decrement ");

   for ( j=100 ; j>0 ; ) {
       locate(81,13);
       gprintf(14," %3.1fV ",parameter.threshold_level);
```

เอกสารนี้เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        j = bioskey(0);
        switch (j) {
            case 18989 :  if (parameter.threshold_level==-10.0)
                             beep(1000,500);
                          else
                             parameter.threshold_level-=0.5;
                          break;
            case 20011 :  if (parameter.threshold_level==10.0)
                             beep(1000,500);
                          else
                             parameter.threshold_level+=0.5;
                          break;
            case ENTR  :  j=0 ;
                          break;
        }
    }
    boxerase(450,150,900,300);
    locate(42,44);
    gprintf(11,"THRESHOLD LEVEL    : %3.1fV",parameter.threshold_level);
}


parameter_var()
{
  switch (menus[2].active) {
      case 0  :  trigger_word();
                 break;
      case 1  :  trigger_enable();
                 break;
      case 2  :  trigger_delay();
                 break;
      case 3  :  glitch();
                 break;
      case 4  :  trigger_start();
                 break;
      case 5  :  threshold();
                 break;
  }
}


data_filter()
{
  int  check;
  register int i;

  for ( i=0 ; i<4096 ; i++ )
      dummy[i] = 0;

  for ( i=0 ; i<4096 ; i++ ) {
      data_in[i] = data_in[i] & parameter.wordenable;
  }

  for ( i=0,check=100 ; i<4096 && check>0 ; i++ ) {
      if ( data_in[i]==parameter.wordtrig )
         check = 0;
  }
  i--;
  parameter.wordstop = i;
  i += parameter.wordelay;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  for (  ; i<4096 ; i++ ) {
      data_in[i] = 0;
  }
}


run()
{
  unsigned int stop;
  register int index;

  for ( index=0 ; index<4096 ; index++ )
      dummy[index] = 0;

  outport(0x306,(char) 0x20);

  for ( stop=100 ; stop>0 ; ) {
      stop = (int) inport(0x30D);
  }

  for ( index=0 ; index<4096 ; index++ ) {
      data_in[index] = (int) inport(0x300);
  }

  data_filter();
  beep(8000,200);
  beep(4000,300);
  beep(1000,400);
  wrt_str_erase(60,15,"Data Sampling Is Complete",11);
}


selection(int order)
{
  switch (order) {
     case 0  :  file();            break;
     case 1  :  freq();            break;
     case 2  :  parameter_var();   break;
     case 3  :  run();             break;
  }
}


void cdecl main(void)
{
  unsigned int i,temp;
  char name[10];
  initial();
  graphics_open();
  header();
  footer();
  for ( i=0 ;  ; ) {
      temp = bioskey(0);
      logic_data.prev = i;
      switch (temp) {
         case LEFT_ARROW  :  if ( i==0 ) i = 3;
                             else  i--;
                             pulldown_update(i);
                             break;
         case RIGHT_ARROW :  if ( i==3 ) i = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        else  i++;
                        pulldown_update(i);
                        break;
        case UP_ARROW    :  if ( menus[i].active==0 )
                              menus[i].active = menus[i].item_num-1;
                           else
                              menus[i].active = menus[i].active-1;
                           draw_text(i);
                           break;
        case DOWN_ARROW  :  if ( menus[i].active==menus[i].item_num-1 )
                              menus[i].active = 0;
                           else
                              menus[i].active = menus[i].active+1;
                           draw_text(i);
                           break;
        case  ENTR       :  selection(i);
                           break;
      }
    }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <bios.h>
#include <ctype.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <alloc.h>
#include <stdlib.h>
#include <string.h>

#define READ_MODE  0
#define WRITE_MODE 1
#define FALSE      0
#define TRUE       1

unsigned char dit[8][8]={{ 19, 25, 23, 17, 14,  8, 10, 16},
                         { 21, 31, 29, 27, 12,  2,  4,  6},
                         { 28, 30, 32, 22,  5,  3,  1, 11},
                         { 18, 24, 26, 20, 15,  9,  7, 13},
                         { 14,  8, 10, 16, 19, 25, 23, 17},
                         { 12,  2,  4,  6, 21, 31, 29, 27},
                         {  5,  3,  1, 11, 28, 30, 32, 22},
                         { 15,  9,  7, 13, 18, 24, 26, 20}};
/*
unsigned char dit[8][8]={{ 22,  6, 18,  2, 21,  5, 17,  1},
                         { 14, 30, 10, 26, 13, 29,  9, 25},
                         { 20,  4, 24,  8, 19,  3, 23,  7},
                         { 12, 28, 16, 32, 11, 27, 15, 31},
                         { 21,  5, 17,  1, 22,  6, 18,  2},
                         { 13, 29,  9, 25, 14, 30, 10, 26},
                         { 19,  3, 23,  7, 20,  4, 24,  8},
                         { 11, 27, 15, 31, 12, 28, 16, 32}};
*/
unsigned char (huge *bwimage)[256];
unsigned char lut[256],dither[8][33];
int i,j,k,l,m,t;
unsigned char d;

void waiting(void)
{
  puts("ON LINE LASER PRINTER HP II");
  puts("press anykey");
  getch();
}

int fexist(char *filename)
{
FILE *ftest;
  if ((ftest = fopen(filename,"rb")) == NULL)
    return(FALSE);
  else
  {
    fclose(ftest);
    return(TRUE);
  }
}

void getfname(char *message, char *st, char *ext, int mode)
{
  *st = '\0';
  while (!strlen(st))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
    printf(message,ext);
    gets(st);
    if (strchr(st,'.') == NULL)
      strcat(st,ext);
    switch (mode)
    {
      case (READ_MODE): if (!fexist(st))
                        {
                           *st = '\0';
                           puts("ERROR: Can't open file ! ! !");
                        }
                        break;
      default:          if (fexist(st))
                        {
                          printf("Overwrite %s (Y/N)? ",st);
                          if (toupper(getchar()) != 'Y')
                            *st = '\0';
                        }
    }
  }
}

void showtime(void)
{
  struct time t1;
  gettime(&t1);
  printf("%2u:%2u:%2u.%2u \n",t1.ti_hour,t1.ti_min,t1.ti_sec,t1.ti_hund);
}

void init(unsigned char (huge *(huge *image))[256])
/*unsigned char (huge *(huge *image))[256];*/
{
   if ((*image=farcalloc(256,256)) == NULL)
   {
     puts("ERROR: Memory not enough ! ! !");
     exit('M');
   }
}

void initlut(unsigned char lut[256])
{
int i;

  for (i=0;i<256;i++) lut[i]=i*10/78;
}

void initdither(void)
{
int i,m,l,d;

  for (i=0;i<33;i++)
    for (m=0;m<8;m++)
    {
      dither[m][i]=0;
      d=0;
      for (l=0;l<8;l++)                          /* dither column # */
      {
        d<<=1;
        if ( i < dit[m][l]) d++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }
      dither[m][i]=d;
    }

  /*for (i=0;i<33;i++)
   for (m=0;m<8;m++) printf("%5x",dither[m][i]);*/
}

void writelineMGD(unsigned char huge *image, int y)
{
  while ((inportb(0x300) & 2)==0) ;
  outportb(0x300,0x01);
  movedata(FP_SEG(image),FP_OFF(image),0xd000,y*256,256);
  outportb(0x300,0x08);
}

void writeMGD(unsigned char (huge *image)[256])
{
int y;

  for (y=0;y<256;y++)
    writelineMGD(image[y],y);
}

void readimg(unsigned char (huge *image)[256])
{
char  fname[30];
FILE  *fp;

   getfname("Enter IMAGE data file name (read)[%s]: ",fname,".IMG",READ_MODE);
   if ( (fp=fopen(fname,"rb")) == NULL)
        {
           puts("ERROR: Can't open file ! ! !");
           exit('F');
        }
   else
   {
      fread((void *) image,256*128,2,fp);
      puts("Open O.K.");
      fclose(fp);
  }
}

void  print(void)
{
  puts("BW IMAGE LASER PRINT Version1.0");
  puts("by IMAGE CRSC KMITL\n");

  init(&bwimage);

  initlut(lut);

  initdither();

  readimg(bwimage);

  waiting();
  printf("success 0%\r");

                                                  /* print bw image */
  fprintf(stdprn,"%c*t300R",27);                  /* set 300 dpi */
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  fprintf(stdprn,"%c*r0A",27);                    /* Start graphic */
  for (i=0;i<256;i++)                                 /* line # */
  {
    for (m=0;m<8;m++)                                 /* dither line # */
    {
      fprintf(stdprn,"%c*b256W",27);                  /* transfer */
      for (j=0;j<256;j++)                             /* colume # */
        biosprint(0, dither[m][lut[bwimage[i][j]]], 0);
        /*fprintf(stdprn,"%c",dither[m][lut[bwimage[i][j]]]);*/
    }
    printf("success %d%%\r",100-(256-i)*100/256);
  }
  fprintf(stdprn,"%c*rB%c",27,12);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 #include <ctype.h>
#include <string.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <alloc.h>
#include <stdlib.h>
#include <mem.h>

#define  READ_MODE  0
#define  WRITE_MODE 1
#define  FALSE      0
#define  TRUE       1

unsigned char dit1[4][4]={{ 50,100,150,200},
                          { 50,100,150,200},
                          { 50,100,150,200},
                          { 50,100,150,200}};


unsigned char dit2[4][4]={{ 15,135, 45,165},
                          {195, 75,225,105},
                          { 60,180, 30,150},
                          {240,120,210, 90}};


unsigned char dit3[4][4]={{ 15, 30, 60,105},
                          { 45, 75,120,165},
                          { 90,135,180,210},
                          {150,195,225,240}};


unsigned char dit4[4][4]={{ 15, 30, 45, 60},
                          { 75, 90,105,120},
                          {135,150,165,180},
                          {195,210,225,240}};

unsigned char (huge *bwimage)[256];
unsigned char lut[256];
int i,j,k,l,m,t;
unsigned char d;
unsigned char dit[4][4]={{ 15,135, 45,165},
                         {195, 75,225,105},
                         { 60,180, 30,150},
                         {240,120,210, 90}};


void wait(void)
{
   puts("ON LINE PRINTER");
   puts("then press any key");
   getch();
}

int fexist(char *filename)
{
FILE *ftest,*fopen();
  if ((ftest = fopen(filename,"rb")) == NULL)
    return(FALSE);
  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
    fclose(ftest);
    return(TRUE);
  }
}

void getfname(char *message, char *st, char *ext, int mode)
{
  *st = '\0';
  while (!strlen(st))
  {
    printf(message,ext);
    gets(st);
    if (strchr(st,'.') == NULL)
      strcat(st,ext);
    switch (mode)
    {
      case (READ_MODE): if (!fexist(st))
                        {
                          *st = '\0';
                          puts("ERROR: Can't open file ! ! !");
                        }
                        break;
      default:          if (fexist(st))
                        {
                          printf("Overwrite %s (Y/N)? ",st);
                          if (toupper(getchar()) != 'Y')
                            *st = '\0';
                        }
    }
  }
}

void showtime(void)
{
  struct time t1;
  gettime(&t1);
  printf("%2u:%2u:%2u.%2u \n",t1.ti_hour,t1.ti_min,t1.ti_sec,t1.ti_hund);
}

void init(unsigned char (huge *(huge *image))[256])
{
   if ((*image=farcalloc(256,256)) == NULL)
   {
     puts("ERROR: Memory not enough ! ! !");
     exit('M');
   }
}

void initlut(unsigned char lut[256])
{
int i;

  for (i=0;i<256;i++) lut[i]=i;
}

void initdither(unsigned char dit[4][4])
{
int i,j,opt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    puts("Dither Pattern");
    puts("1:One dimension");
    puts("2:Jarvis *");
    puts("3:Triangle");
    puts("4:PANDA");
    printf("Use:");scanf("%d",&opt);getchar();
    switch(opt)
    {
     case 1: for (i=0;i<4;i++)
              for (j=0;j<4;j++)
                dit[i][j]=dit1[i][j];
             break;

     case 2: for (i=0;i<4;i++)
              for (j=0;j<4;j++)
                dit[i][j]=dit2[i][j];
             break;

     case 3: for (i=0;i<4;i++)
              for (j=0;j<4;j++)
                dit[i][j]=dit3[i][j];
             break;

     case 4: for (i=0;i<4;i++)
              for (j=0;j<4;j++)
                dit[i][j]=dit4[i][j];
             break;
    }
}

void readimg(unsigned char (huge *image)[256])
{
char  fname[30];
FILE  *fp;


    getfname("Enter IMAGE data file name (read)[%s]: ",fname,".IMG",READ_MODE);
    if ( (fp=fopen(fname,"rb")) == NULL)
         {
            puts("ERROR: Can't open file ! ! !");
            exit('F');
         }
    else
    {
       fread((void *) image, 256*128, 2, fp);
       puts("Open O.K.");
       fclose(fp);
    }
}

void main(void)
{
  puts("BW IMAGE PRINT Version1.0");
  puts("by IMAGE CRSC KMITL\n");
  init(&bwimage);

  initlut(lut);

  initdither(dit);
```

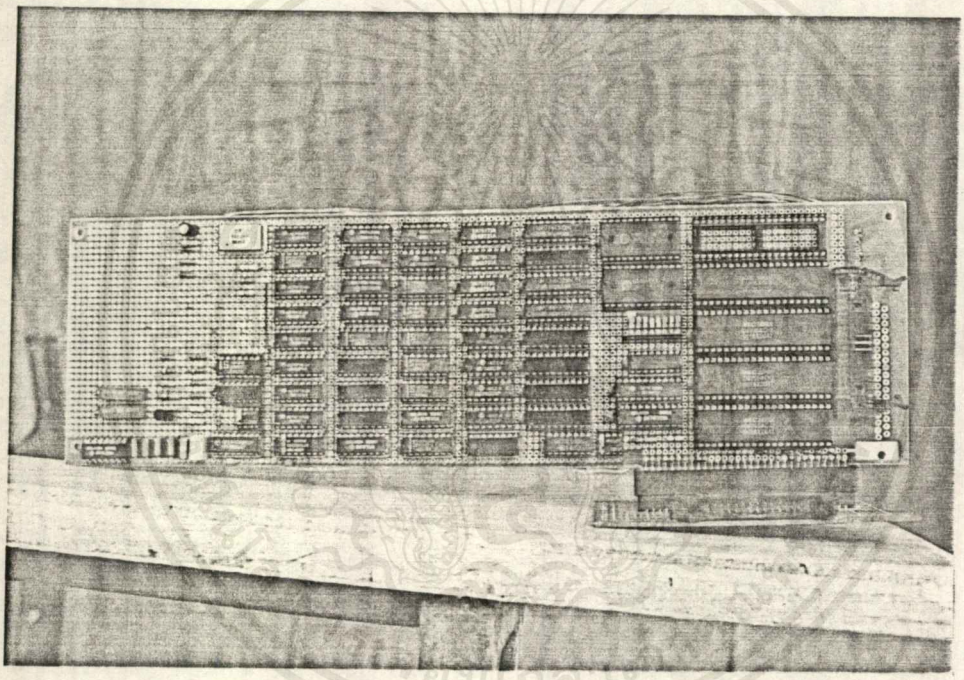
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    readimg(bwimage);

    wait();
    printf("success 0%\r");
    for (i=0;i<32;i++)                                  /* line # */
    {
                                                        /* print bw image */
      fprintf(stdprn,"%c%c%c",27,'A',8);                /* set 8/72 line space */
      fprintf(stdprn,"%c%c%c%c%c",27,'*',0,0,1);        /* set graphic 256 dots */
      for (j=0;j<256;j++)                               /* column # */
      { d=0;
          for (k=0;k<8;k++)                             /* dot in line */
          {
            d<<=1;
            if (lut[bwimage[i*8+k][j]] < dit[(i*8+k)%4][j%4]) d++;
          }
        fprintf(stdprn,"%c",d);
      }
      fprintf(stdprn,"\n");

      printf("success %d%%\r",100-(32-i)*100/32+100/32);
    }
    fprintf(stdprn,"%c%c",27,64);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การใช้งานลอจิกอนาไลเซอร์

ในการใช้งานลอจิกอนาไลเซอร์นั้น ก็จะเริ่มจากการติดตั้งตังลอจิกอนาไลเซอร์ลงไปในเครื่อง IBM PC โดยจะติดตั้งลงไปในสล็อตขนาด 8 บิท แต่จะติดตั้งลงไปในสล็อตขนาด 16 ก็ได้ซึ่งจะทำฝานได้เช่นกัน แต่ก็ยังคงใช้การถ่ายเทข้อมูลแบบ 8 บิท เหมือนเดิม



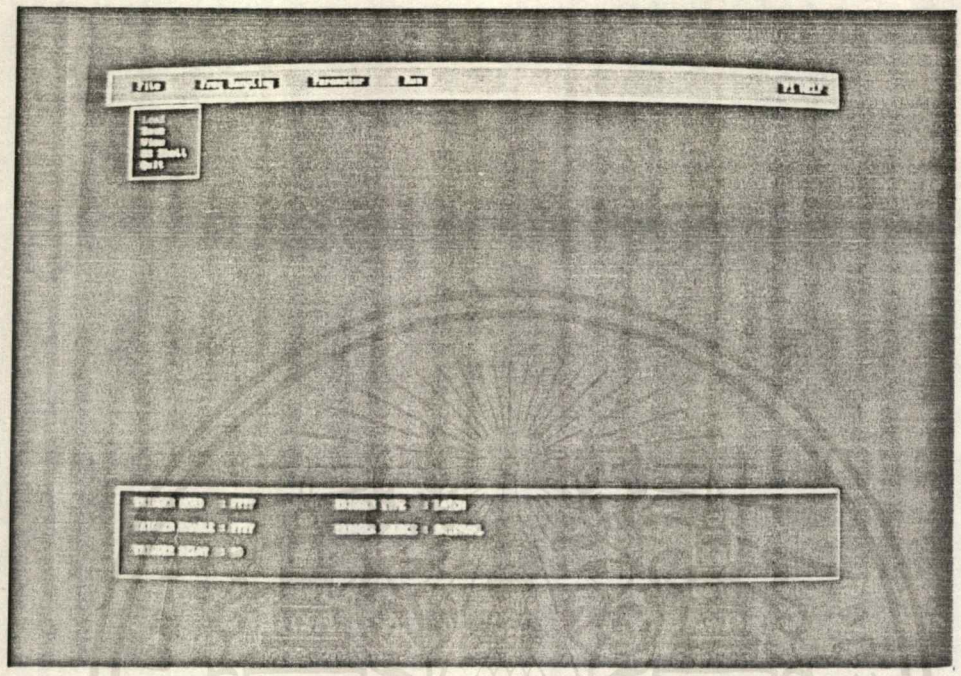
รูปที่ 5.1 ลักษณะของลอจิกอนาไลเซอร์

เมื่อได้ติดตั้งลงไปเป็นที่เรียบร้อยแล้วนั้น ในขั้นตอนต่อไปก็จะเป็นการติดตั้งสายวัด ซึ่งจะมีจุดเชื่อมต่ออยู่แล้วตรงขอบขวาของการ์ด จะมีลักษณะเป็นตัวคอนเนคเตอร์ตัวผู้ เมื่อนำคอนเนคเตอร์ตัวผู้เข้าไปเสียบแล้วจะล็อคกันทำให้มีความแข็งแรงในการใช้งาน

หลังจากที่ติดตั้งอุปกรณ์ฮาร์ดแวร์เรียบร้อยแล้ว ในขั้นตอนต่อไปก็จะเป็นการเรียกโปรแกรมที่จะทำหน้าที่ควบคุมการทำงานของลอจิกอนาไลเซอร์ขึ้นมาทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

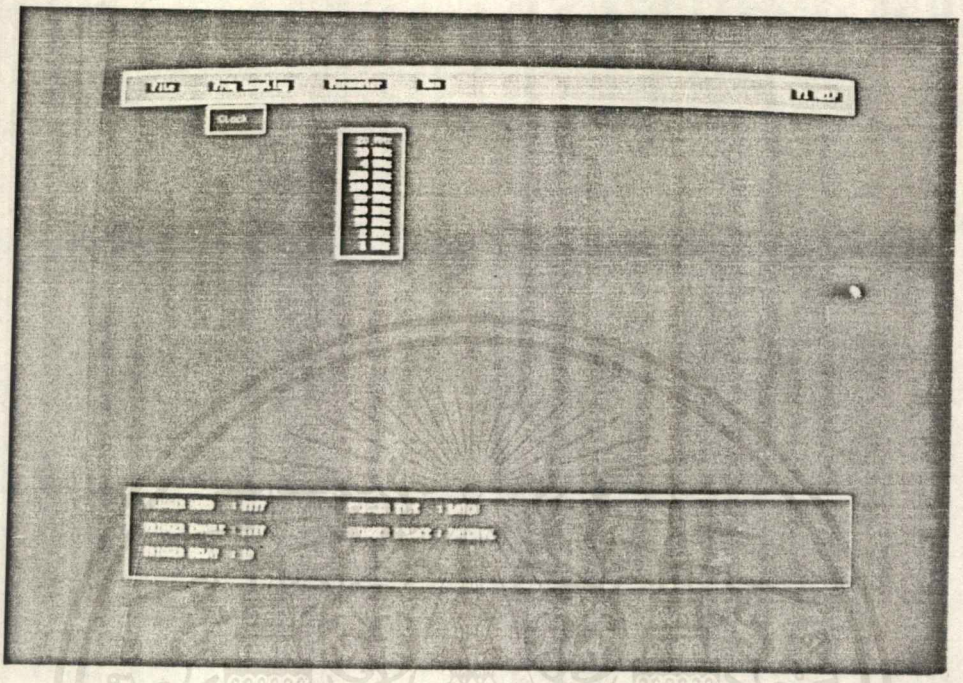
เมื่อได้ติดตั้งทุกอย่างเรียบร้อยแล้ว ก็จะได้หน้าจอที่มีลักษณะดังรูป



รูปที่ 5.2 เมนูหลักของลอจิกอนาไลเซอร์

ต่อไปก็จะต้องเลือกความถี่ที่ใช้ในการแซมปลิงข้อมูลเข้ามาเก็บ โดยเลื่อนเมนูไปที่หัวข้อ "Freq Sampling" แล้วกดปุ่ม <ENTR> ก็จะมีรายการความถี่ปรากฏขึ้นมา ให้เลือกความถี่ที่เหมาะสมโดยควรที่จะมีความถี่มากกว่าระบบที่จะวัดประมาณ 5-10 เท่า

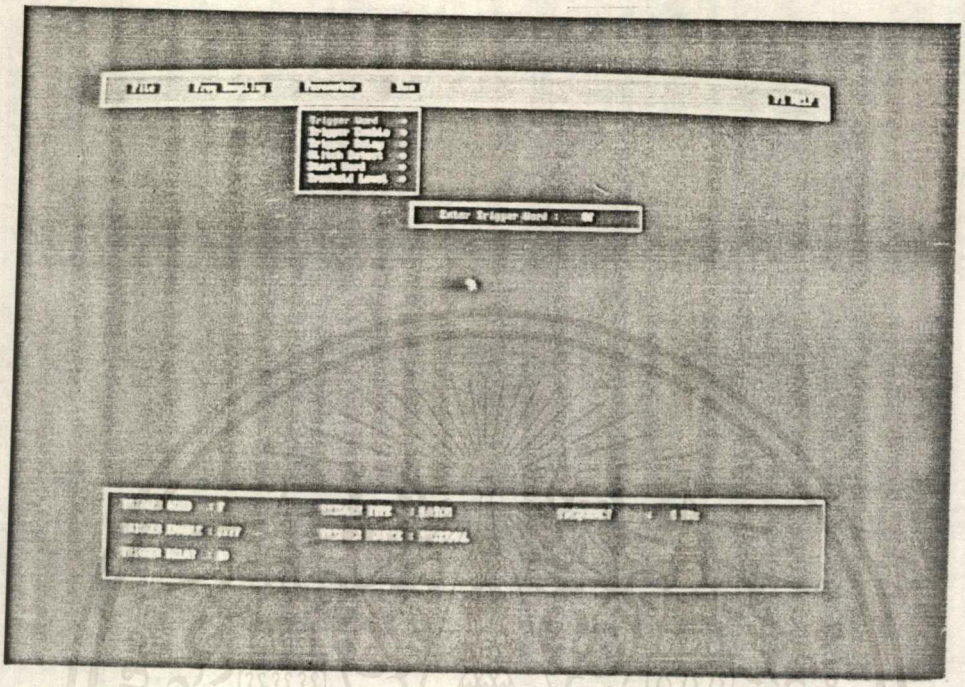
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 รายการในเมนูความถี่

หลังจากที่เลือกความถี่ในการแซมปลิงแล้ว ต่อไปก็จะต้องเลือกค่าพารา-มีเตอร์ต่างๆ เพื่อให้มีความสัมพันธ์กับการเลือกข้อมูล โดยให้เลื่อนเมนูไปที่เมนู "Parameter" แล้วก็จะมีรายการตัวเลือกต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

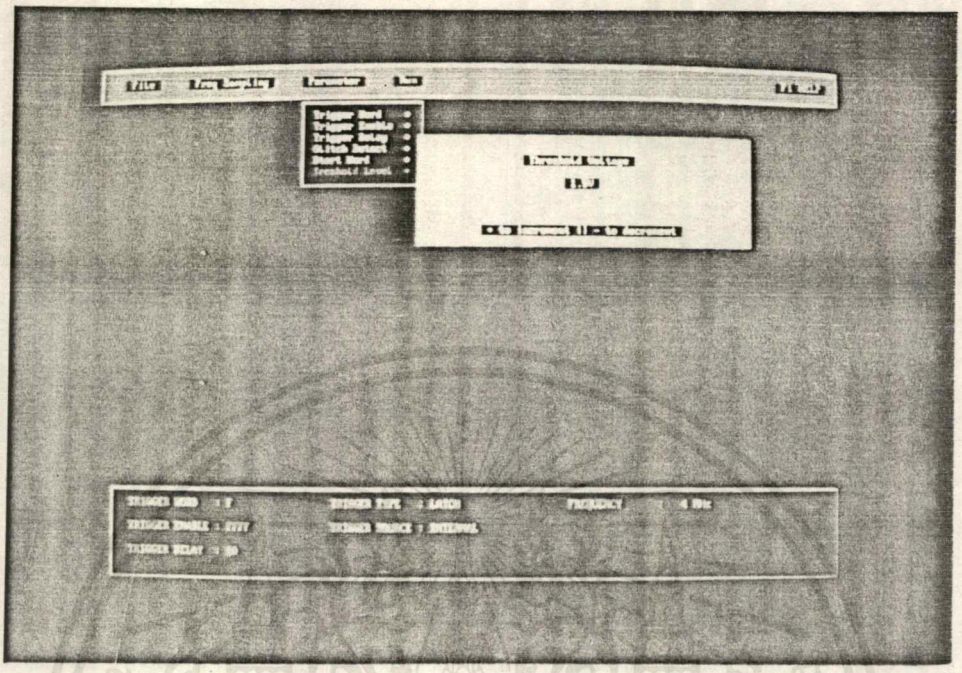


รูปที่ 5.4 รายการย่อยในเมนูพารามีเตอร์

- Trigger Word มีค่าเริ่มต้นเป็น FFFFh
- Trigger Enable มีค่าเริ่มต้นเป็น FFFFh
- Trigger Delay มีค่าเริ่มต้นเป็น 10h
- ใช้ลักษณะของการสุ่มข้อมูลแบบ LATCH เพื่อให้สามารถตรวจจับสัญญาณกลิทช์ได้
- แหล่งที่มาของการทริก โดยปกติจะเลือกให้เป็น Internal Trig

ค่าพารามีเตอร์ที่ได้กล่าวถึงทั้ง 5 อย่างนั้น เมื่อโปรแกรม เริ่มต้นทำงานก็จะมีการกำหนดค่าเริ่มต้น (Default Value) ให้กับตัวแปรเหล่านี้ แต่สำหรับ Threshold Level จะต้องปรับก่อนในครั้งแรก โดยเลื่อนไปที่เมนูย่อยอันล่างสุดที่อยู่ในเมนูพารามีเตอร์ การเพิ่มค่าแรงดันเทรสโฮลด์จะทำโดยการกดปุ่ม '+' ส่วนการลดค่าลงนั้นสามารถเลือกได้โดยการกดปุ่ม '-' ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

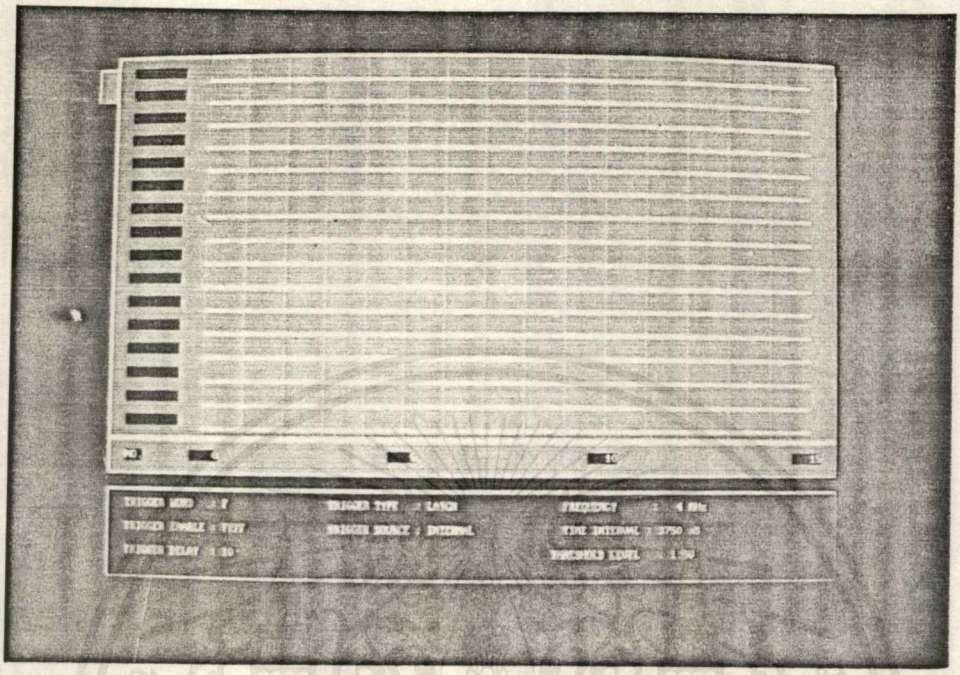


รูปที่ 5.5 การเลือกแรงดันเทรสโฮลด์

เมื่อผู้ใช้ได้ตั้งค่าต่างๆ เป็นที่เรียบร้อยแล้วก็ให้เลื่อนเมนูหลักไปที่ "RUN" แล้วกดปุ่ม <ENTR> และรอจนโปรแกรมแสดงข้อความว่าได้ทำการการเก็บข้อมูลเรียบร้อยแล้ว ก็จะเป็นการแสดงข้อมูลขึ้นมาดู โดยเลื่อนไปที่เมนู "FILE" แล้วให้เลือก 'View' พร้อมกับเลือกการแสดงผลแบบกราฟฟิก

จากนั้นโปรแกรมก็จะแสดงหน้าจอที่ใช้ในการแสดงผลขึ้นมา แต่จะปรากฏว่าไม่มีข้อมูลใดๆ จากการแซมปลิงเลย ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 หน้าจอในการแสดงผลแบบกราฟฟิก เมื่อยังไม่มีการตั้งชื่อสัญญาณ

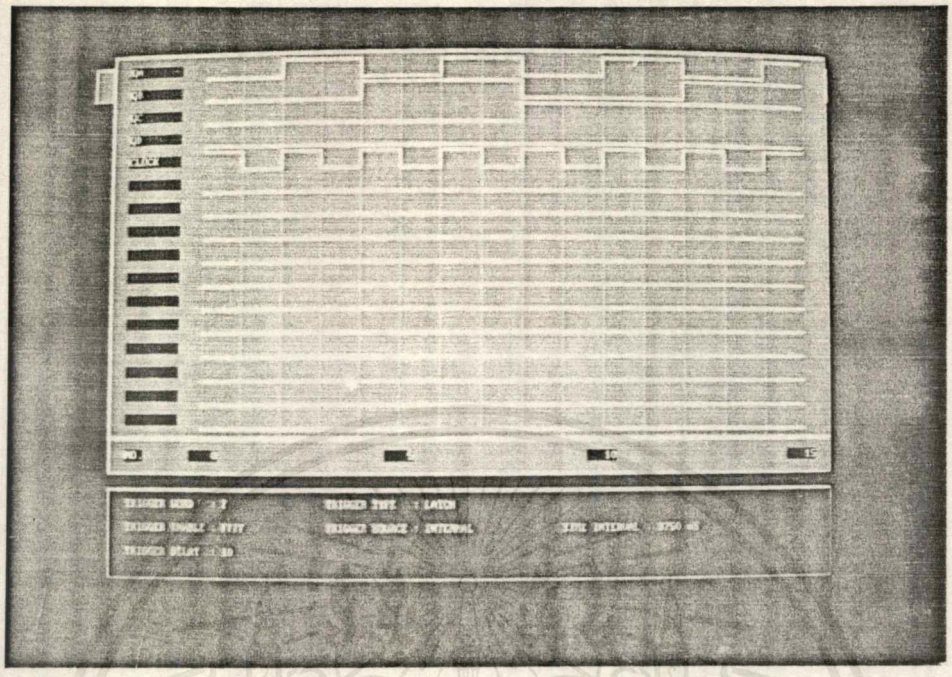
ที่เป็นอย่างนี้ก็เพราะว่าในช่องสัญญาณใด ๆ ก็ตามที่ยังไม่มีการตั้งชื่อให้ เมื่อตัวโปรแกรมรับข้อมูลจากฮาร์ดแวร์เข้ามาหมดแล้ว ก็จะรีเซ็ตสัญญาณในช่องนั้นให้กลายเป็นศูนย์ให้หมด ทั้งนี้เพื่อความง่ายในการวิเคราะห์ข้อมูลที่ได้มา

การตั้งชื่อช่องสัญญาณ จะทำได้โดยการกดปุ่ม <CTRL> + <C> แล้วใส่ตัวเลขว่าจะตั้งให้กับช่องสัญญาณไหน ซึ่งจะมีได้ตั้งแต่เลข 1-16 ตามด้วยชื่อของช่องสัญญาณที่เราต้องการนั้นๆ

ในการทดลองตัวอย่างนี้ จะให้ลอจิกอนาไลเซอร์จับสัญญาณเอาท์พุทของไอซีเบอร์ 74LS90 ซึ่งเป็น BCD Counter โดยจะอาศัยสัญญาณ clock ที่มาจากสล็อตของ IBM PC มีความถี่ประมาณ 2 MHz

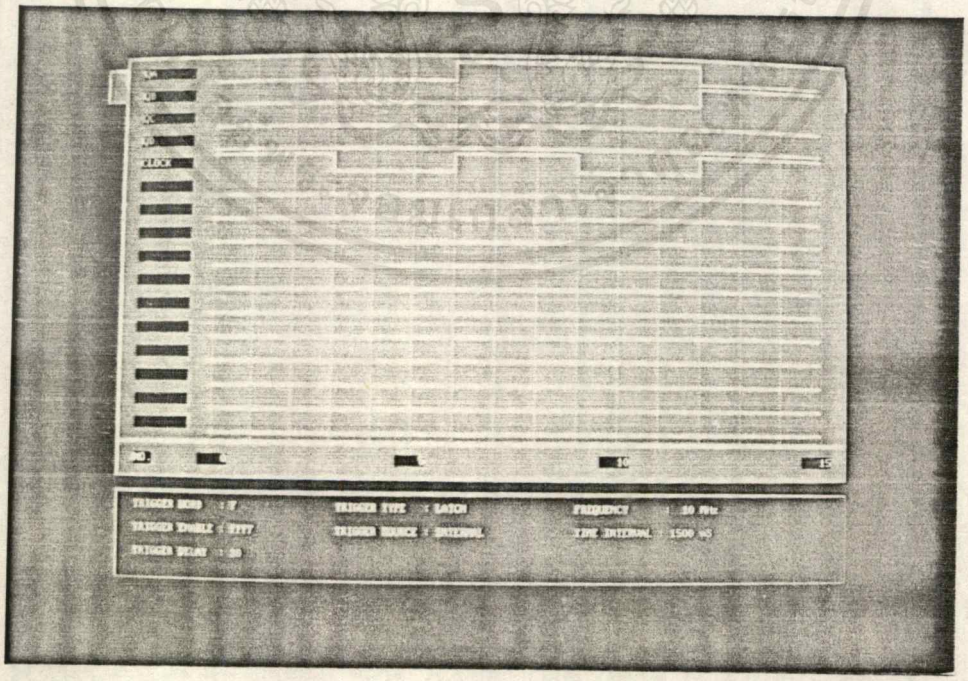
ในการจับสัญญาณครั้งแรก จะตั้งความถี่ในการแซมปลิงเอาไว้ที่ 4 MHz ซึ่งจะได้ลักษณะของสัญญาณดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



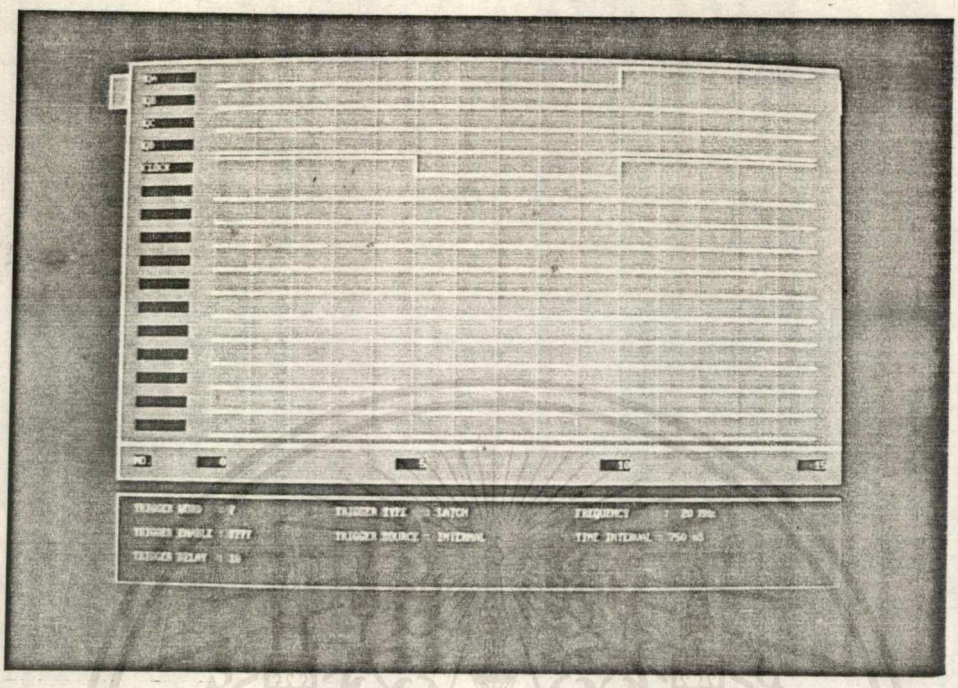
รูปที่ 5.7 เมื่อทำการแซมปลิงโดยใช้สัญญาณนาฬิกาขนาด 4 MHz

และเมื่อเปลี่ยนความถี่ที่ใช้ในการแซมปลิงไปเรื่อยๆ ก็จะได้ลักษณะดังนี้



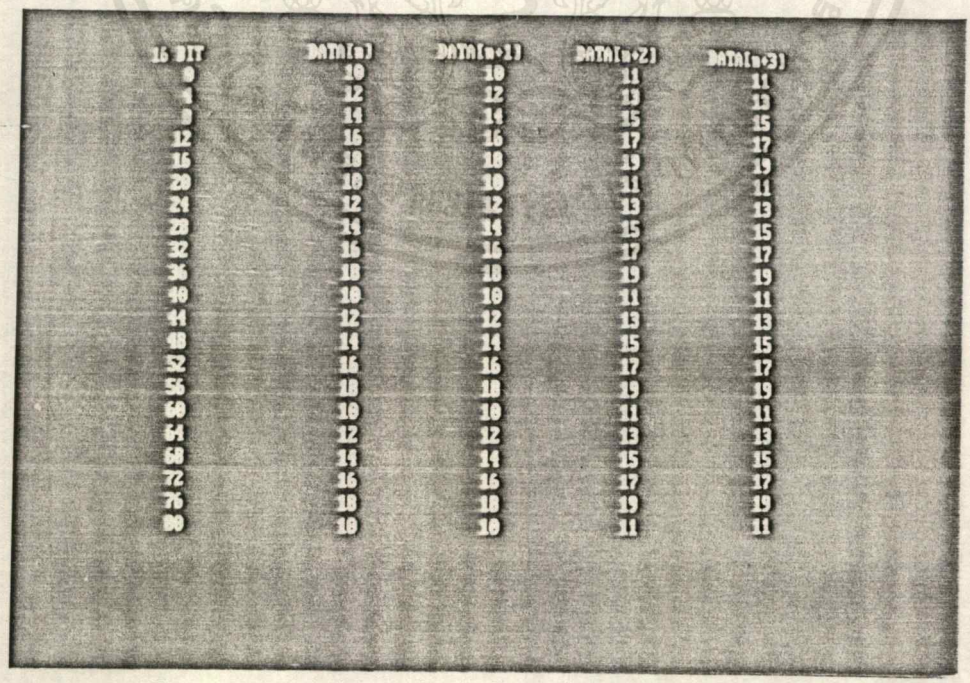
รูปที่ 5.9 เมื่อทำการแซมปลิงโดยใช้สัญญาณนาฬิกาขนาด 10 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 เมื่อทำการแซมปลิงโดยใช้สัญญาณนาฬิกาขนาด 20 MHz

และเมื่อแสดงข้อมูลในลักษณะของตัวอักษรก็จะได้ดังรูป



16 BIT	DATA[n]	DATA[n+1]	DATA[n+2]	DATA[n+3]
0	10	10	11	11
4	12	12	13	13
8	14	14	15	15
12	16	16	17	17
16	18	18	19	19
20	10	10	11	11
24	12	12	13	13
28	14	14	15	15
32	16	16	17	17
36	18	18	19	19
40	10	10	11	11
44	12	12	13	13
48	14	14	15	15
52	16	16	17	17
56	18	18	19	19
60	10	10	11	11
64	12	12	13	13
68	14	14	15	15
72	16	16	17	17
76	18	18	19	19
80	10	10	11	11

รูปที่ 5.11 การแสดงข้อมูลในลักษณะของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลและวิจารณ์

โครงงงานชิ้นนี้แสดงถึงการนำคอมพิวเตอร์เข้ามาร่วมใช้ในการวิเคราะห์สัญญาณลอจิก โดยมุ่งเน้นไปที่ไดอะแกรมเวลา(TimingDiagram) ตัวอินเทอร์เฟสและซอฟแวร์นั้นเราได้พัฒนาขึ้นมาเองทั้งหมดซึ่งใช้เวลาพอสมควร ในส่วนของซอฟแวร์จะได้ผลเป็นที่น่าพอใจมาก แต่ส่วนฮาร์ดแวร์จะไม่สามารถทำงานได้ถึงตามที่ออกแบบไว้ เนื่องจาก ไม่สามารถนำไปลงแผ่นวงจรพิมพ์ได้ เพราะว่า มีตัวอุปกรณ์และสายมากเกินไปจนเมื่อทำการลากสายแบบอัตโนมัติ โดยใช้โปรแกรม PCAD แล้ว ระยะห่างของสายจะน้อยมากจนไม่สามารถทำแผ่นวงจรพิมพ์ให้ได้ และไม่สามารถแก้ไขได้ จึงต้องวางอุปกรณ์ลงไปบนโปรโตบอร์ด ดังนั้นจึงใช้กับสัญญาณนาฬิกาได้ค่อนข้างต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. บุญยงค์ ภู่นันทพงษ์," การวัดและการควบคุมทางอุตสาหกรรม", คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ , pp 120-141

2. สมาคมส่งเสริมเทคโนโลยี (ไทย - ญี่ปุ่น)," เครื่องมือวัดอุตสาหกรรม ", pp 97-102

3. โยธิน เปรมปราณีรัชต์," เครื่องมือวัดอุตสาหกรรม ",วศบ. พระจอมเกล้าMS.E.E (Nihon University) , pp77-89

4. กิตติ ตีรณศรษฐ ," อุปกรณ์วัด และควบคุมในขบวนการ " , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

5. ชานินทร์ ถาวรศาสนวงศ์ , ทินกร ตุ๊ก ," การอินเตอร์เฟส IBM PC, ฟิสิกส์เซ็นเตอร์การพิมพ์

6. " Linear 1 Databook ", National Semiconductor

7. Jame W.Coffron,"z80 Application ",LSBN 0-89588-094-6

8. John Uffenbeck,"Microcomputer & Microprocessors", Prentice Hall international Edition

9. Herbert Schildt," Turbo C/C++ , The Complete Referance",McGraw-HILL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

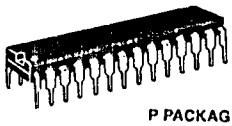
MCM6264

8K x 8 Bit Fast Static RAM

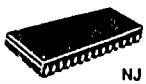
The MCM6264 is fabricated using Motorola's high-performance silicon-gate CMOS technology. Static design eliminates the need for external clocks or timing strobes, while CMOS circuitry reduces power consumption and provides for greater reliability.

This device meets JEDEC standards for functionality and pinout, and is available in plastic dual-in-line and plastic small-outline J-leaded packages.

- Single 5 V ±10% Power Supply
- Fully Static — No Clock or Timing Strobes Necessary
- Fast Access Times: 15, 20, 25 and 35 ns
- Equal Address and Chip Enable Access Times
- Output Enable ($\overline{G}$) Feature for Increased System Flexibility and to Eliminate Bus Contention Problems
- Low Power Operation: 110 –140 mA Maximum ac
- Fully TTL-Compatible — Three-State Output

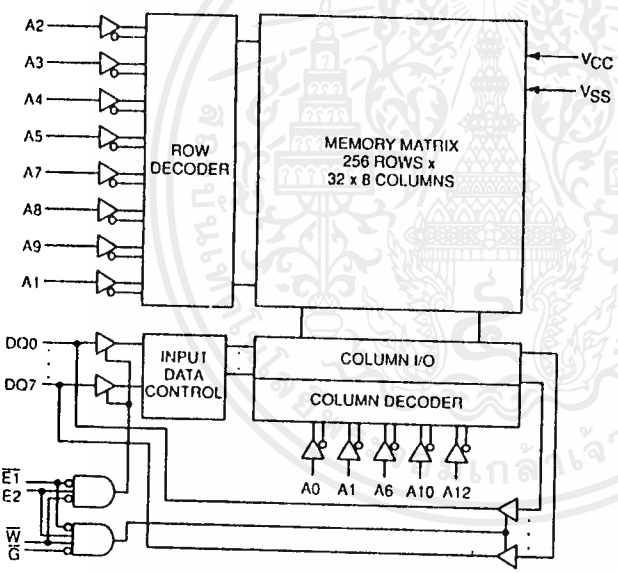


P PACKAGE
300-MIL PLASTIC
CASE 710B-01

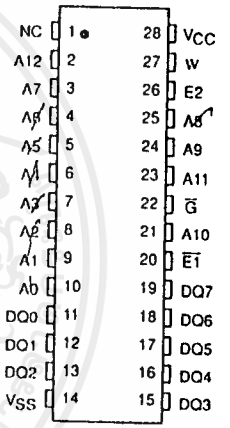


NJ PACKAGE
300-MIL SOJ
CASE 810B-03

BLOCK DIAGRAM



PIN ASSIGNMENT



PIN NAMES	
A0—A12	Address Input
DQ0—DQ7	Data Input/Data Output
$\overline{W}$	Write Enable
G	Output Enable
$\overline{E1}$, E2	Chip Enable
NC	No Connection
V_{CC}	Power Supply (+ 5 V)
V_{SS}	Ground

TRUTH TABLE (X = don't care)

$\overline{E1}$	E2	$\overline{G}$	$\overline{W}$	Mode	V_{CC} Current	Output	Cycle
H	X	X	X	Not Selected	I_{SB1}, I_{SB2}	High–Z	–
X	L	X	X	Not Selected	I_{SB1}, I_{SB2}	High–Z	–
L	H	H	H	Output Disabled	I_{CCA}	High–Z	–
L	H	L	H	Read	I_{CCA}	D_{out}	Read Cycle
L	H	X	L	Write	I_{CCA}	High–Z	Write Cycle

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit.

This CMOS memory circuit has been designed to meet the dc and ac specifications shown in the tables, after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow of at least 500 linear feet per minute is maintained.

ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage Relative to V_{SS}	V_{CC}	– 0.5 to + 7.0 V	V
Voltage Relative to V_{SS} For Any Pin Except V_{CC}	V_{in}, V_{out}	– 0.5 to V_{CC} + 0.5	V
Output Current (per I/O)	I_{out}	± 30	mA
Power Dissipation	P_D	1.0	W
Temperature Under Bias	T_{bias}	– 10 to + 85	°C
Operating Temperature	T_A	0 to +70	°C
Storage Temperature—Plastic	T_{stg}	– 55 to + 125	°C

NOTE: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability

DC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5.0 V ± 10%, T_A = 0 to 70°C, Unless Otherwise Noted)

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage (Operating Voltage Range)	V_{CC}	4.5	5.0	5.5	V
Input High Voltage	V_{IH}	2.2	—	V_{CC} + 0.3*	V
Input Low Voltage	V_{IL}	– 0.5**	—	0.8	V

*V_{IH} (max) = V_{CC} + 0.3 V dc; V_{IH} (max) = V_{CC} + 2.0 V ac (pulse width < 20 ns)
**V_{IL} (min) = – 0.5 V dc; V_{IL} (min) = – 2.0 V ac (pulse width < 20 ns)

DC CHARACTERISTICS

Parameter	Symbol	Min	Max	Unit
Input Leakage Current (All Inputs, V_{in} = 0 to V_{CC})	$I_{lkg(I)}$	—	± 1	µA
Output Leakage Current ($\overline{E}$ = V_{IH} or G = V_{IH}, V_{out} = 0 to V_{CC})	$I_{lkg(O)}$	—	± 1	µA
Output Low Voltage (I_{OL} = 8.0 mA)	V_{OL}	—	0.4	V
Output High Voltage (I_{OH} = – 4.0 mA)	V_{OH}	2.4	—	V

POWER SUPPLY CURRENTS

Parameter	Symbol	- 15	- 20	- 25	- 35	Unit
AC Active Supply Current (I_{out} = 0 mA, V_{CC} = Max, f = f_{max})	I_{CCA}	140	130	120	110	mA
AC Standby Current ($\overline{E}$ = V_{IH} or E2 = V_{IL}, V_{CC} = Max, f = f_{max})	I_{SB1}	40	35	30	30	mA
Standby Current ($\overline{E}$ ≥ V_{CC} – 0.2 V or E2 ≤ V_{SS} + 0.2 V, V_{in} ≤ V_{SS} + 0.2 V, or ≥ V_{CC} – 0.2 V)	I_{SB2}	20	20	20	20	mA

5

CAPACITANCE (f = 1 MHz, dV = 3 V, T_A = 25°C, Periodically sampled rather than 100% tested)

Characteristic	Symbol	Max	Unit
Address and Data Input Capacitance	C_{in}	6	pF
Control Pin Input Capacitance ($\overline{E}$, E2 $\overline{G}$, $\overline{W}$)	C_{in}	8	pF
Output Capacitance	C_{out}	7	pF

AC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5 V ± 10%, T_A = 0 to + 70°C, Unless Otherwise Noted)

Input Timing Measurement Reference Level ..... 1.5 V
Input Pulse Levels ..... 0 to 3 V
Input Rise/Fall Time ..... 5 ns

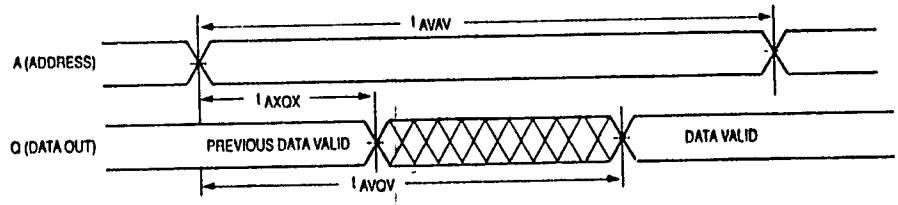
Output Timing Measurement Reference Level ..... 1.5 V
Output Load ..... Figure 1A Unless Otherwise Noted

READ CYCLE (See Notes 1 and 2)

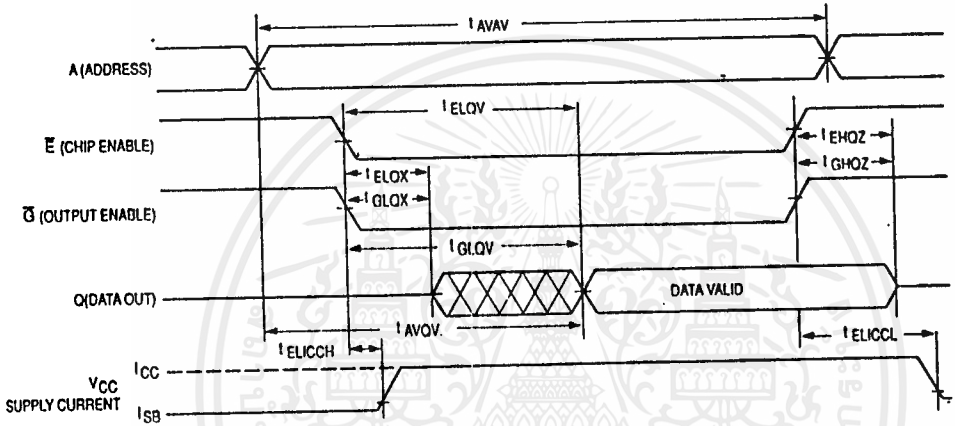
Parameter	Symbol		-15		-20		-25		-35		Unit	Notes
	Std	Alt	Min	Max	Min	Max	Min	Max	Min	Max		
Read Cycle Time	t_{AVAV}	t_{RC}	15	—	20	—	25	—	35	—	ns	3
Address Access Time	t_{AVQV}	t_{AA}	—	15	—	20	—	25	—	35	ns	
Enable Access Time	t_{ELQV}	t_{ACS}	—	15	—	20	—	25	—	35	ns	4
Output Enable Access Time	t_{GLQV}	t_{OE}	—	8	—	10	—	11	—	12	ns	
Output Hold from Address Change	t_{AXQX}	t_{OH}	4	—	4	—	4	—	4	—	ns	
Enable Low to Output Active	t_{ELQX}	t_{CLZ}	4	—	4	—	4	—	4	—	ns	5,6,7
Output Enable Low to Output Active	t_{GLQX}	t_{OLZ}	0	—	0	—	0	—	0	—	ns	5,6,7
Enable High to Output High-Z	t_{EHQZ}	t_{CHZ}	0	8	0	9	0	10	0	11	ns	5,6,7
Output Enable High to Output High-Z	t_{GHQZ}	t_{OHZ}	0	7	0	8	0	9	0	10	ns	5,6,7
Power Up Time	t_{ELICCH}	t_{PU}	0	—	0	—	0	—	0	—	ns	
Power Down Time	t_{EHICCL}	t_{PD}	—	15	—	20	—	25	—	35	ns	

NOTES:
1. $\overline{W}$ is high for read cycle.
2. $\overline{E}1$ and E2 are represented by $\overline{E}$ in this data sheet. E2 is of opposite polarity to $\overline{E}$.
3. All timings are referenced from the last valid address to the first transitioning address.
4. Addresses valid prior to or coincident with $\overline{E}$ going low.
5. At any given voltage and temperature, t_{EHQZ} max < t_{ELQX} min, and t_{GHQZ} max < t_{GLQX} min, both for a given device and from device to device.
6. Transition is measured ± 500 mV from steady-state voltage with load of Figure 1B.
7. This parameter is sampled and not 100% tested.
8. Device is continuously selected ($\overline{E}1$ = V_{IL}, E2 = V_{IH}, $\overline{G}$ = V_{IL}).

READ CYCLE 1 (See Note 8)



READ CYCLE 2 (See Note 4)



AC TEST LOADS

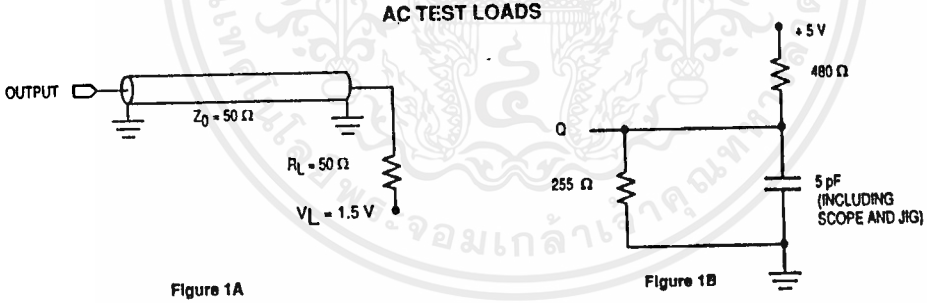
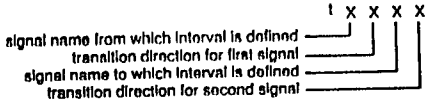


Figure 1A

Figure 1B

TIMING PARAMETER ABBREVIATIONS



The transition definitions used in this data sheet are:

- H = transition to high
- L = transition to low
- V = transition to valid
- X = transition to invalid or don't care
- Z = transition to off (high impedance)

TIMING LIMITS

The table of timing values shows either a minimum or a maximum limit for each parameter. Input requirements are specified from the external system point of view. Thus, address setup time is shown as a minimum since the system must supply at least that much time (even though most devices do not require it). On the other hand, responses from the memory are specified from the device point of view. Thus, the access time is shown as a maximum since the device output will be valid no later than that time.

WRITE CYCLE 1 (W Controlled) (See Notes 1, 2, and 3)

Parameter	Symbol		-15		-20		-25		-35		Unit	Notes
	Std	Alt	Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t_{AVAV}	t_{WC}	15	—	20	—	25	—	35	—	ns	4
Address Setup Time	t_{AVWL}	t_{AS}	0	—	0	—	0	—	0	—	ns	
Address Valid to End of Write	t_{AVWH}	t_{AW}	12	—	15	—	17	—	20	—	ns	
Write Pulse Width	t_{WLWH}, t_{WLEH}	t_{WP}	12	—	15	—	17	—	20	—	ns	
Write Pulse Width, High (Output Enable devices)	t_{WLWH}, t_{WLEH}	t_{WP}	10	—	12	—	15	—	17	—	ns	5
Data Valid to End of Write	t_{DVWH}	t_{DW}	7	—	8	—	10	—	12	—	ns	
Data Hold Time	t_{WHDX}	t_{DH}	0	—	0	—	0	—	0	—	ns	
Write Low to Output High-Z	t_{WLQZ}	t_{WZ}	0	7	0	8	0	10	0	12	ns	6,7,8
Write High to Output Active	t_{WHQX}	t_{OW}	4	—	4	—	4	—	4	—	ns	6,7,8
Write Recovery Time	t_{WHAX}	t_{WR}	0	—	0	—	0	—	0	—	ns	

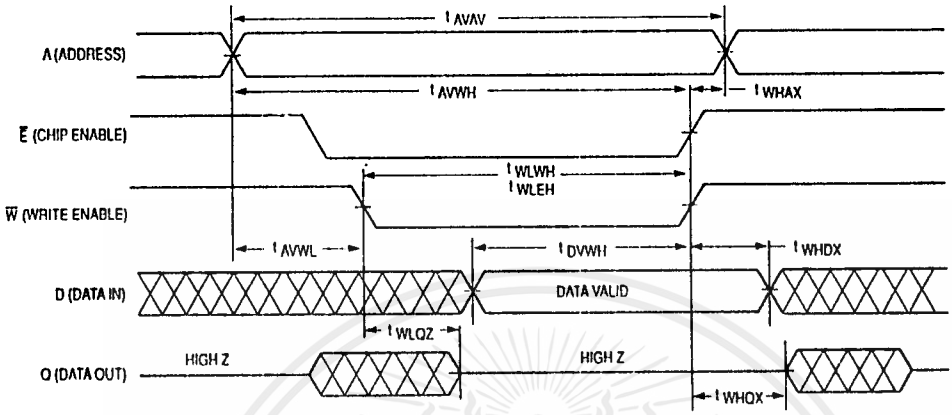
5

WRITE CYCLE 2 ($\overline{E}$ Controlled) (See Notes 1 and 2)

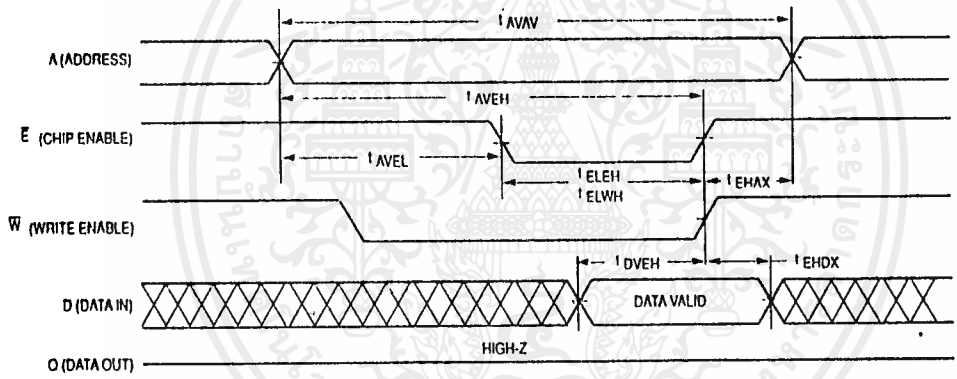
Parameter	Symbol		-15		-20		-25		-35		Unit	Notes
	Std	Alt	Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t_{AVAV}	t_{WC}	15	—	20	—	25	—	35	—	ns	4
Address Setup Time	t_{AVEL}	t_{AS}	0	—	0	—	0	—	0	—	ns	
Address Valid to End of Write	t_{AVEH}	t_{AW}	12	—	15	—	20	—	25	—	ns	
Enable to End of Write	t_{ELEH}, t_{ELWH}	t_{CW}	10	—	12	—	15	—	25	—	ns	9,10
Data Valid to End of Write	t_{DVEH}	t_{DW}	7	—	8	—	10	—	15	—	ns	
Data Hold Time	t_{EHDX}	t_{DH}	0	—	0	—	0	—	0	—	ns	
Write Recovery Time	t_{EHAX}	t_{WR}	0	—	0	—	0	—	0	—	ns	

NOTES:
1. A write occurs during the overlap of $\overline{E}$ low and $\overline{W}$ low.
2. $\overline{E1}$ and E2 are represented by $\overline{E}$ in this data sheet. E2 is of opposite polarity to $\overline{E}$.
3. If $\overline{G}$ goes low coincident with or after $\overline{W}$ goes low, the output will remain in a high-impedance state.
4. All timings are referenced from the last valid address to the first transitioning address.
5. If $\overline{G} \geq V_{IH}$, the output will remain in a high-impedance state.
6. At any given voltage and temperature, t_{WLQZ} max < t_{WHQX} min, both for a given device and from device to device.
7. Transition is measured ± 500 mV from steady-state voltage with load of Figure 1B.
8. This parameter is sampled and not 100% tested.
9. If $\overline{E}$ goes low coincident with or after $\overline{W}$ goes low, the output will remain in a high-impedance state.
10. If $\overline{E}$ goes high coincident with or before $\overline{W}$ goes high, the output will remain in a high-impedance state.

WRITE CYCLE 1 (See Notes 1, 2, and 3)



WRITE CYCLE 2 (See Notes 1 and 2)



ORDERING INFORMATION (Order by Full Part Number)

MCM	6264	X	XX	XX

- MCM — Motorola Memory Prefix
- 6264 — Part Number
- X — Package (P or BP = 300-mil Plastic DIP, NJ or BNJ = 300-mil SOJ)
- XX — Speed (15 = 15 ns, 20 = 20 ns, 25 = 25 ns, 35 = 35 ns)
- XX — Shipping Method (R2 = Tape & Reel, Blank = Rails)

Full Part Numbers—

MCM6264P15	MCM6264NJ15	MCM6264NJ15R2
MCM6264P20	MCM6264NJ20	MCM6264NJ20R2
MCM6264BP25	MCM6264BNJ25	MCM6264BNJ25R2
MCM6264BP35	MCM6264BNJ35	MCM6264BNJ35R2

DAC-08

HIGH SPEED 8-BIT MULTIPLYING D-TO-A CONVERTER

SILICON MONOLITHIC INTEGRATED CIRCUIT

HIGH SPEED 8-BIT MULTIFLYING D-TO-A CONVERTER

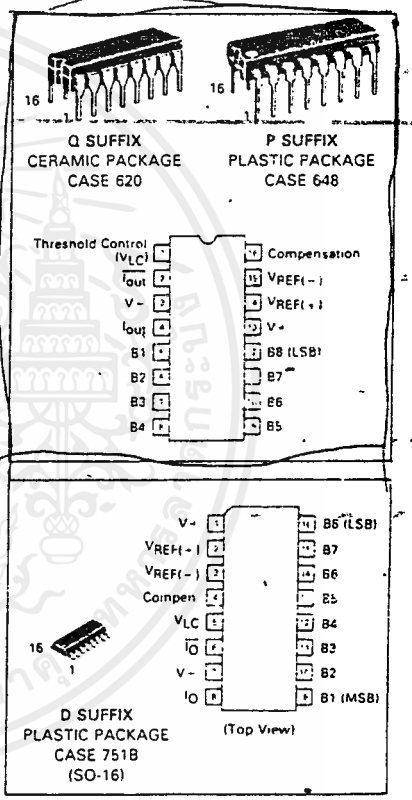
The DAC-08 series is a monolithic 8-bit high speed multiplying digital-to-analog converter, capable of settling to within 1/2 LSB (0.19%) in 85 ns. Monotonic multiplying performance is retained over a wide 40-to-1 reference current range. Full scale and reference currents are matched to within 1 LSB, therefore eliminating the need for full scale trim in most applications.

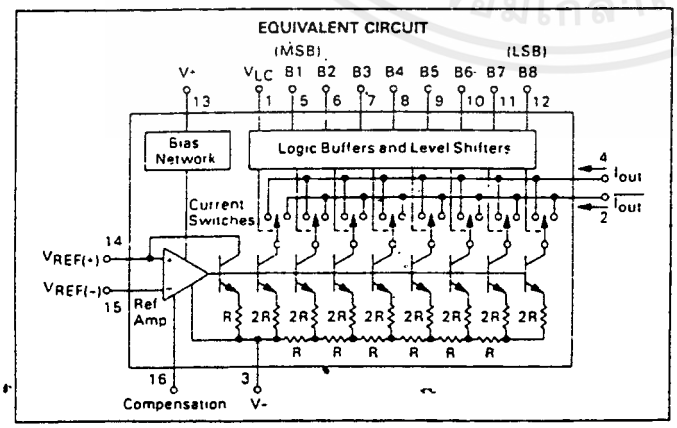
Dual complementry current outputs with high voltage compliance provide added versatility and allow differential mode of operation to effectively double the peak-to-peak output swing. In many applications, output current-to-voltage conversion can be accomplished without requiring an external op amp. Noise-immune inputs permit direct interface with TTL and DTL levels when the logic threshold control, V_{LC}, (Pin 1) is grounded. All other logic family thresholds are attainable by adjusting the voltage level of Pin 1. Performance characteristics are virtually unchanged over the entire ±4.5 V to ±18 V power supply range. Power consumption is typically 33 mW with ±5.0 V supplies.

The DAC-08 is available in several versions, with nonlinearity as tight as ±0.1% (±1/4 LSB) over temperature. All versions are guaranteed monotonic over 8 bits. For an extra margin of performance, Motorola utilizes thin-film resistors permitting very accurate resistive values which are extremely stable over temperature.

High performance characteristics along with low cost, make the DAC-08 an excellent selection for applications such as CRT displays, waveform generation, high speed modems, and high speed analog-to-digital converters.

- Fast Settling Time — 85 ns
- Full Scale Current Prematched to ±1 LSB
- Nonlinearity Over Temperature to ±0.1% Max
- Differential Current Outputs
- High Voltage Compliance Outputs −10 V to +18 V
- Wide Range Multiplying Capability
- Inputs Compatable With TTL, DTL, CMOS, PMOS, ECL, HTL
- Low Full Scale Current Drift
- Wide Power Supply Range ±4.5 V to ±18 V
- Low Power Consumption
- Thin-Film Resistors
- Low Cost





ORDERING INFORMATION

Device	Nonlinearity	Temperature Range	Package
DAC-08AQ	±0.1%	−55°C to +125°C	Ceramic
DAC-08Q	±0.19%		Ceramic
DAC-08HQ	±0.1%	0°C to +70°C	Ceramic
DAC-08EQ	±0.19%		Ceramic
DAC-08CQ	±0.39%		Ceramic
DAC-08CD	±0.39%		SO-16
DAC-08ED	±0.19%		SO-16
DAC-08HP	±0.1%		Plastic
DAC-08EP	±0.19%		Plastic
DAC-08CP	±0.39%		Plastic

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

Rating	Symbol	Value	Unit
V+ Supply to V- Supply	—	36	V
Logic Inputs	—	V- to V- Plus 36	V
Logic Threshold Control	V_{LC}	V- to V+	V
Analog Current Outputs	I_{out}	See Figure 7	mA
Reference Inputs (V14, V15)	V_{REF}	V- to V+	V
Reference Input Differential Voltage (V14 to V15)	$V_{REF(D)}$	±18	V
Reference Input Current (I14)	I_{REF}	5.0	mA
Operating Temperature Range DAC-08AQ, Q DAC-08HQ, EQ, CQ, HP, EP, CP, ED, CD	T_A	 -55 to +125 0 to +70	°C
Storage Temperature	T_A	-65 to +150	°C
Power Dissipation Derate above 100°C	P_D $R_{\theta JA}$	500 10	mW mW/°C

ELECTRICAL CHARACTERISTICS (V_S = ±15 V, I_{REF} = 2.0 mA, T_A = -55°C to +125°C, unless otherwise noted.)

Characteristic	Symbol	DAC-08A			DAC-08			Unit
		Min	Typ	Max	Min	Typ	Max	
Resolution	—	8	8	8	8	8	8	Bits
Monotonicity	—	8	8	8	8	8	8	Bits
Nonlinearity, T_A = 0°C to +70°C	NL	—	—	±0.1	—	—	±0.19	%FS
Settling Time to ± 1/2 LSB, Figure 24 (All Bits Switched On or Off, T_A = 25°C)(Note 1)	t_s	—	85	—	—	85	—	ns
Propagation Delay, T_A = 25°C (Note 1) Each Bit All Bits Switched	 t_{PLH} t_{PHL}	 — —	 35 35	 — —	 — —	 35 35	 — —	ns
Full Scale Tempco	TCI$_{FS}$	—	±10	—	—	±10	—	ppm °C
Output Voltage Compliance Full Scale Current Change < 1/2 LSB, R_{out} > 20 megohm typ	V_{OC}	-10	—	-18	-10	—	-18	V
Full Range Current (V_{REF} = 10.000 V; R14, R15 = 5 000 kΩ, T_A = 25°C)	I_{FR4}	1.984	1.992	2.000	1.94	1.99	2.04	mA
Full Range Symmetry (I_{FR4} - I_{FR2})	I_{FRS}	—	±0.5	±4.0	—	±1.0	±8.0	µA
Zero Scale Current	I_{ZS}	—	0.1	1.0	—	0.2	2.0	µA
Output Current Range V- = -5.0 V V- = -8.0 V to -18 V	 I_{OR1} I_{OR2}	 0 0	 — —	 2.1 4.2	 0 0	 — —	 2.1 4.2	mA
Logic Input Levels (V_{LC} = 0 V) Logic "0" Logic "1"	 V_{IL} V_{IH}	 — 2.0	 — —	 0.8 —	 — 2.0	 — —	 0.8 —	V
Logic Input Current (V_{LC} = 0 V) Logic Input "0" (V_{in} = -10 V to +0.8 V) Logic Input "1" (V_{in} = +2.0 V to +18 V)	 I_{IL} I_{IH}	 — —	 -2.0 0.002	 -10 10	 — —	 -2.0 0.002	 -10 10	µA
Logic Input Swing, V- = -15 V	V_{IS}	-10	—	+18	-10	—	+18	V
Logic Threshold Range, V_S = ±15 V	V_{THR}	-10	—	+13.5	-10	—	+13.5	V
Reference Bias Current	I_{15}	—	-1.0	-3.0	—	-1.0	-3.0	µA
Reference Input Slew Rate Figure 19 (Note 1)	dI/dt	—	8.0	—	—	8.0	—	mA µs
Power Supply Sensitivity (I_{REF} = 1.0 mA) V+ = 4.5 V to 18 V V- = -4.5 V to -18 V	 PSSI$_{FS+}$ PSSI$_{FS-}$	 — —	 ±0.0003 ±0.002	 ±0.01 ±0.01	 — —	 ±0.0003 ±0.002	 ±0.01 ±0.01	%/%
Power Supply Current V_S = ±5.0 V, I_{REF} = 1.0 mA V_S = +5.0 V, -15 V, I_{REF} = 2.0 mA V_S = ±15 V, I_{REF} = 2.0 mA	 I+ I- I+ I- I+ I-	 — — — — — —	 2.3 -4.3 2.4 -6.4 2.5 -6.5	 3.8 -5.8 3.8 -7.8 3.8 -7.8	 — — — — — —	 2.3 -4.3 2.4 -6.4 2.5 -6.5	 3.8 -5.8 3.8 -7.8 3.8 -7.8	mA
Power Dissipation V_S = ±5.0 V, I_{REF} = 1.0 mA V_S = +5.0 V, -15 V, I_{REF} = 2.0 mA V_S = ±15 V, I_{REF} = 2.0 mA	P_D	 — — —	 33 103 135	 48 136 174	 — — —	 33 108 135	 48 136 174	mW

Note 1. Parameter is not 100% tested; guaranteed by design.

ELECTRICAL CHARACTERISTICS (V_S = ±15 V, I_{REF} = 2.0 mA, T_A = 0°C to 70°C, unless otherwise noted.)

Characteristic	Symbol	DAC-08H			DAC-08E			DAC-08C			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Resolution	—	8	8	8	8	8	8	8	8	8	Bits
Monotonicity	—	8	8	8	8	8	8	8	8	8	Bits
Nonlinearity, T_A = 0°C to +70°C	NL	—	—	±0.1	—	—	±0.19	—	—	±0.39	%FS
Settling Time to ±1/2 LSB (All Bits Switched On or Off, (T_A = 25°C) Figure 24 (Note 1)	t_s	—	85	—	—	85	—	—	85	—	ns
Propagation Delay, T_A = 25°C (Note 1) Each Bit	t_{PLH}	—	35	—	—	35	—	—	35	—	ns
All Bits Switched	t_{PHL}	—	35	—	—	35	—	—	35	—	
Full Scale Tempco	TCI_{FS}	—	±10	—	—	±10	—	—	±10	—	ppm/°C
Output Voltage Compliance Full Scale Current Change < 1/2 LSB, R_{out} > 20 megohm typ.	V_{OC}	−10	—	+18	−10	—	+18	−10	—	+18	V
Full Range Current (V_{REF} = 10.000 V; R14, R15 = 5.000 kΩ) T_A = 25°C	I_{FR4}	1.984	1.992	2.000	1.94	1.99	2.04	1.94	1.99	2.04	mA
Full Range Symmetry (I_{FR4} − I_{FR2})	I_{FRS}	—	±0.5	±4.0	—	±1.0	±8.0	—	±2.0	±16.0	μA
Zero Scale Current	I_{ZS}	—	0.1	1.0	—	0.2	2.0	—	0.2	4.0	μA
Output Current Range V_- = −5.0 V	I_{OR1}	0	—	2.1	0	—	2.1	0	—	2.1	mA
V_- = −8.0 V to −18 V	I_{OR2}	0	—	4.2	0	—	4.2	0	—	4.2	
Logic Input Levels (V_{LC} = 0 V) Logic "0"	V_{IL}	—	—	0.8	—	—	0.8	—	—	0.8	V
Logic "1"	V_{IH}	2.0	—	—	2.0	—	—	2.0	—	—	
Logic Input Current (V_{LC} = 0 V) Logic Input "0" (V_{in} = −10 V to +0.8 V)	I_{IL}	—	−2.0	−10	—	−2.0	−10	—	−2.0	−10	μA
Logic Input "1" (V_{in} = +2.0 V to +18 V)	I_{IH}	—	0.002	10	—	0.002	10	—	0.002	10	
Logic Input Swing, V_- = −15 V	V_{IS}	−10	—	+18	−10	—	+18	−10	—	+18	V
Logic Threshold Range, V_S = ±15 V	V_{THR}	−10	—	+13.5	−10	—	+13.5	−10	—	+13.5	V
Reference Bias Current	I_{15}	—	−1.0	−3.0	—	−1.0	−3.0	—	−1.0	−3.0	μA
Reference Input Slew Rate Figure 19 (Note 1)	dI/dt	—	8.0	—	—	8.0	—	—	8.0	—	mA/μs
Power Supply Sensitivity (I_{REF} = 1.0 mA) V_+ = 4.5 V to 18 V	$PSSI_{FS+}$	—	±0.0003	±0.01	—	±0.0003	±0.01	—	±0.0003	±0.01	%/%
V_- = −4.5 V to −18 V	$PSSI_{FS-}$	—	±0.002	±0.01	—	±0.002	±0.01	—	±0.002	±0.01	
Power Supply Current V_S = ±5.0 V, I_{REF} = 1.0 mA	I_+	—	2.3	3.8	—	2.3	3.8	—	2.3	3.8	mA
	I_-	—	−4.3	−5.8	—	−4.3	−25.8	—	−4.3	−5.8	
V_S = +5.0 V, −15 V, I_{REF} = 2.0 mA	I_+	—	2.4	3.8	—	2.4	3.8	—	2.4	3.8	
	I_-	—	−6.4	−7.8	—	−6.4	−7.8	—	−6.4	−7.8	
V_S = ±15 V, I_{REF} = 2.0 mA	I_+	—	2.5	3.8	—	2.5	3.8	—	2.5	3.8	
	I_-	—	−6.5	−7.8	—	−6.5	−7.8	—	−6.5	−7.8	
Power Dissipation V_S = ±5.0 V, I_{REF} = 1.0 mA	P_D	—	33	48	—	33	48	—	33	48	mW
V_S = +5.0 V, −15 V, I_{REF} = 2.0 mA		—	108	136	—	108	136	—	108	136	
V_S = ±15 V, I_{REF} = 2.0 mA		—	135	174	—	135	174	—	135	174	

Note 1. Parameter is not 100% tested: guaranteed by design.

6

TYPICAL PERFORMANCE CURVES

FIGURE 1 — FULL SCALE CURRENT versus REFERENCE CURRENT

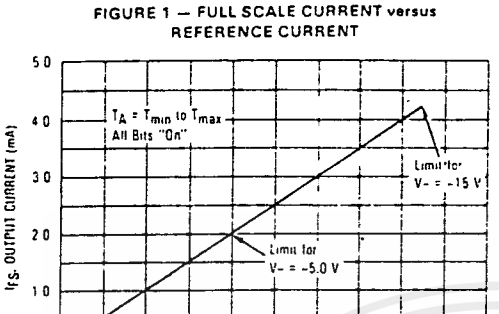
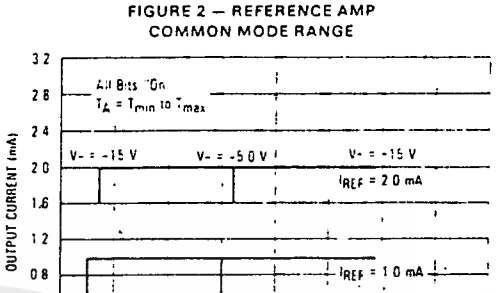


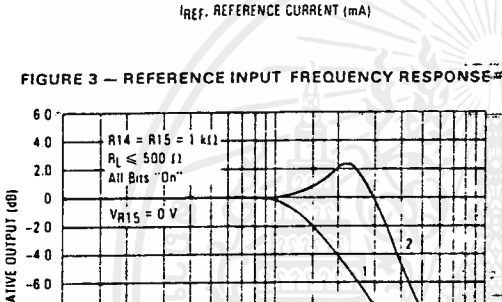
FIGURE 2 — REFERENCE AMP COMMON MODE RANGE



NOTE Positive Common Mode Range is Always (V+) -1.5 V

6

FIGURE 3 — REFERENCE INPUT FREQUENCY RESPONSE



Curve 1 — C_C = 15 pF, V_{in} = 2.0 V p-p Centered at +1.0 V (Large-Signal)
Curve 2 — C_C = 15 pF, V_{in} = 50 mV p-p Centered at +200 mV (Small-Signal)

FIGURE 4 — LSB PROPAGATION DELAY versus I_{FS}

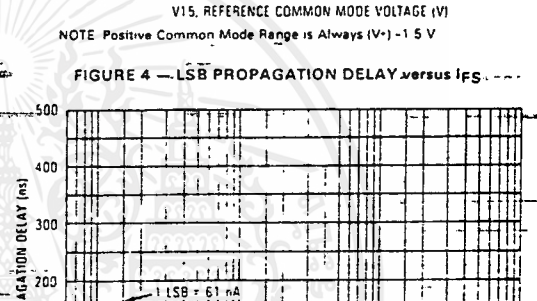


FIGURE 5 — LOGIC INPUT CURRENT versus INPUT VOLTAGE

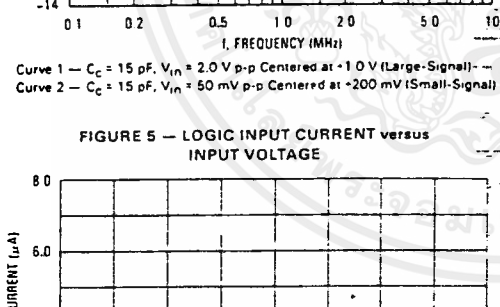
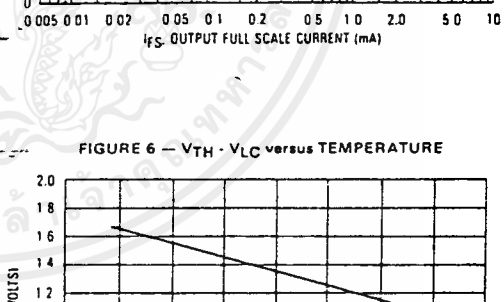


FIGURE 6 — V_{TH} - V_{LC} versus TEMPERATURE



TYPICAL PERFORMANCE CURVES

FIGURE 7 — OUTPUT CURRENT versus OUTPUT VOLTAGE (Output Voltage Compliance)

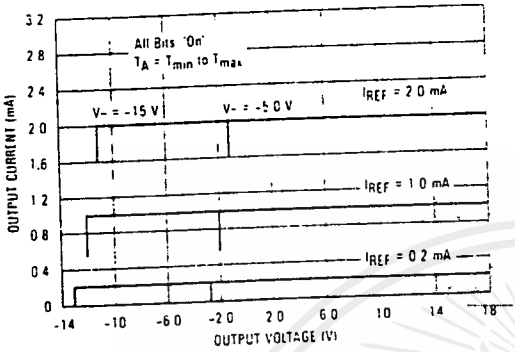


FIGURE 8 — OUTPUT VOLTAGE COMPLIANCE versus TEMPERATURE

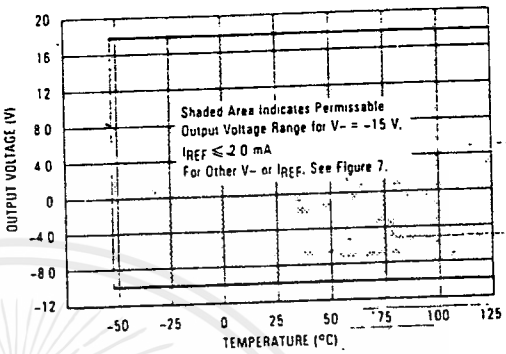
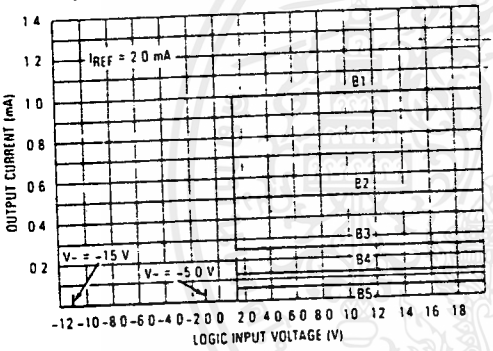


FIGURE 9 — BIT TRANSFER CHARACTERISTICS



NOTE B1–B8 have identical transfer characteristics. Bits are fully switched with less than 1/2 LSB error, at less than ±100 mV from actual threshold. These switching points are guaranteed to lie between 0.8 V and 2.0 V over operating temperature range (V_{LC} = 0 V).

FIGURE 10 — POWER SUPPLY CURRENT versus V+

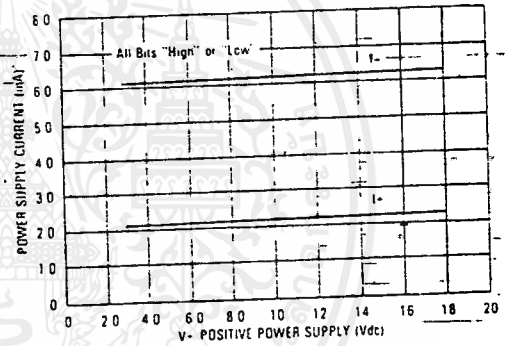


FIGURE 11 — POWER SUPPLY CURRENT versus V–

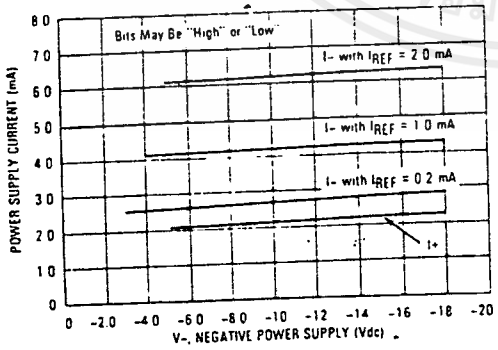
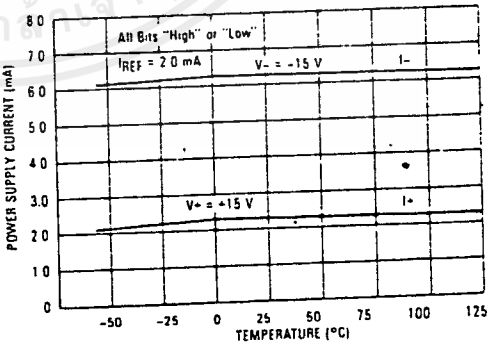


FIGURE 12 — POWER SUPPLY CURRENT versus TEMPERATURE



BASIC CIRCUIT CONFIGURATIONS

FIGURE 13 — RECOMMENDED FULL SCALE ADJUSTMENT CIRCUIT

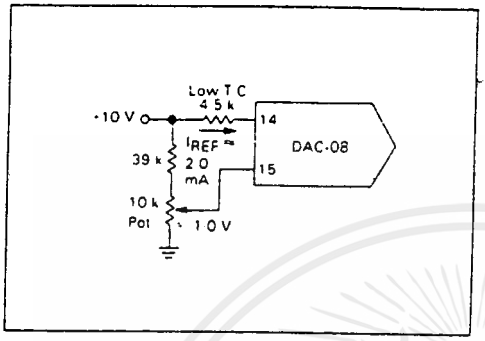
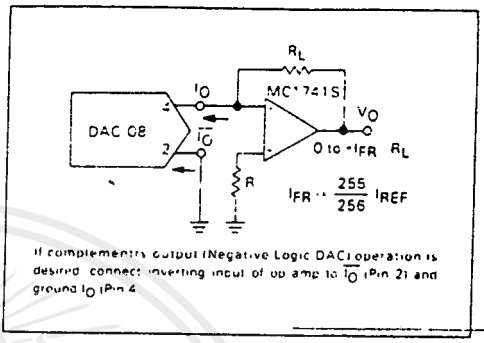
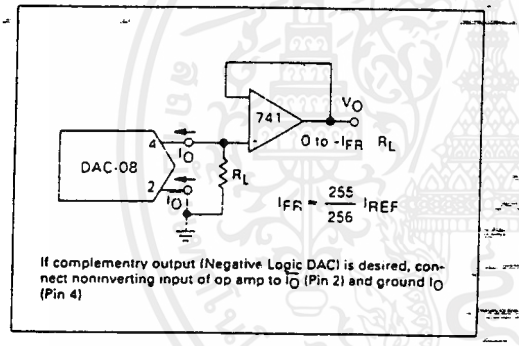


FIGURE 14 — POSITIVE LOW IMPEDANCE OUTPUT OPERATION



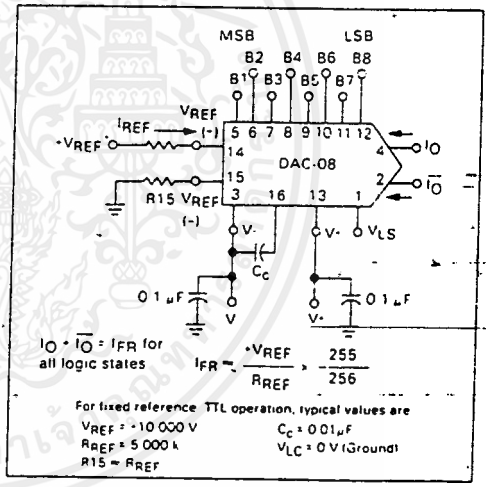
If complementry output (Negative Logic DAC) operation is desired, connect inverting input of op amp to $\overline{I_O}$ (Pin 2) and ground I_O (Pin 4

FIGURE 15 — NEGATIVE LOW IMPEDANCE OUTPUT OPERATION



If complementry output (Negative Logic DAC) is desired, connect noninverting input of op amp to $\overline{I_O}$ (Pin 2) and ground I_O (Pin 4)

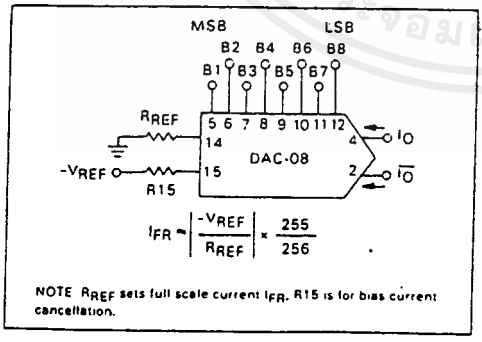
FIGURE 16 — BASIC POSITIVE REFERENCE OPERATION



For fixed reference TTL operation, typical values are

V_{REF} = +10.000 V
R_{REF} = 5.000 k
R15 = R_{REF}
C_C = 0.01 µF
V_{LC} = 0 V (Ground)

FIGURE 17 — BASIC NEGATIVE REFERENCE OPERATION



NOTE R_{REF} sets full scale current I_{FR}. R15 is for bias current cancellation.

BASIC CIRCUIT CONFIGURATIONS

FIGURE 18 — ACCOMMODATING BIPOLAR REFERENCES

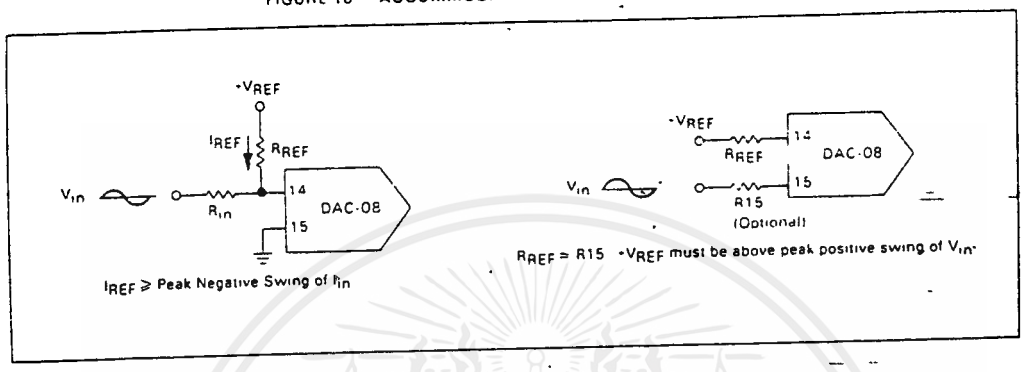


FIGURE 19 — PULSED REFERENCE OPERATION

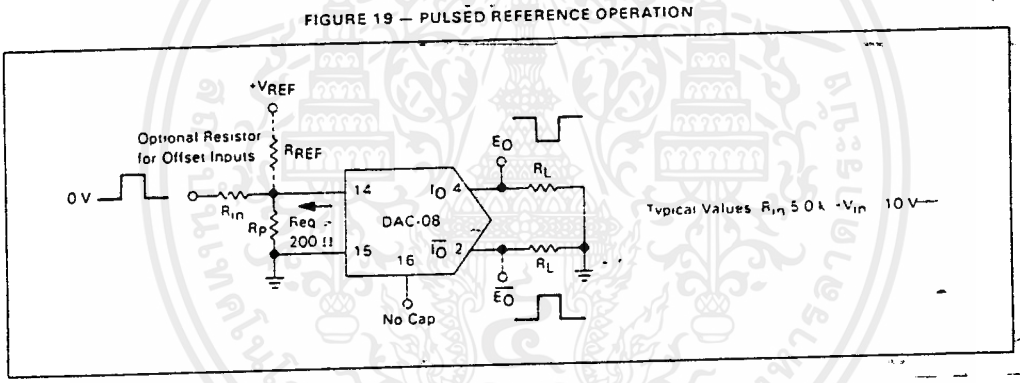
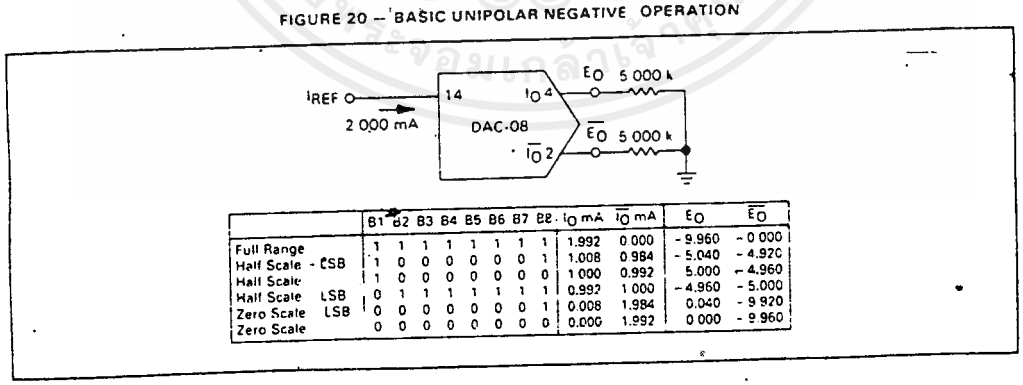


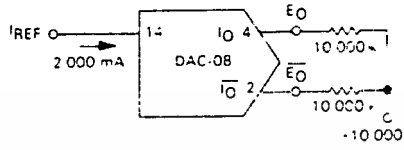
FIGURE 20 — BASIC UNIPOLAR NEGATIVE OPERATION



	B1	B2	B3	B4	B5	B6	B7	B8	IO mA	IO mA	EO	EO
Full Range	1	1	1	1	1	1	1	1	1.992	0.000	- 9.960	- 0.000
Half Scale - LSB	1	0	0	0	0	0	0	1	1.008	0.984	- 5.040	- 4.920
Half Scale	1	0	0	0	0	0	0	0	1.000	0.992	5.000	- 4.960
Half Scale + LSB	0	1	1	1	1	1	1	1	0.992	1.000	- 4.960	- 5.000
Zero Scale + LSB	0	0	0	0	0	0	0	1	0.008	1.984	0.040	- 9.920
Zero Scale	0	0	0	0	0	0	0	0	0.000	1.992	0.000	- 9.960

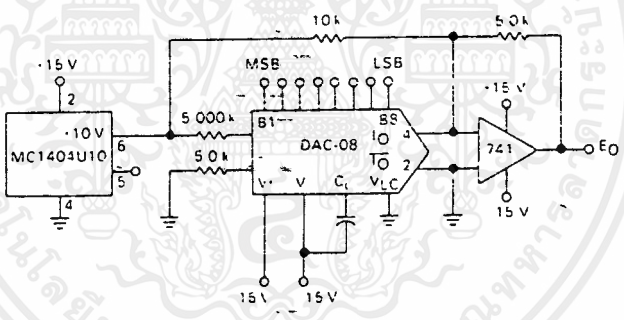
BASIC CIRCUIT CONFIGURATIONS

FIGURE 21 BASIC BIPOLAR OUTPUT OPERATION



	B1	B2	B3	B4	B5	B6	B7	B8	E_O	$\overline{E_O}$
Pos Full Range	1	1	1	1	1	1	1	1	9.920	-10.000
Pos Full Range -LSB	1	1	1	1	1	1	1	0	9.840	-9.920
Zero Scale +LSB	1	0	0	0	0	0	0	1	0.080	-0.160
Zero Scale	1	0	0	0	0	0	0	0	0.000	-0.080
Zero Scale -LSB	0	1	1	1	1	1	1	1	-0.080	0.000
Neg Full Scale +LSB	0	0	0	0	0	0	0	1	-9.920	9.840
Neg Full Scale	0	0	0	0	0	0	0	0	-10.000	9.920

FIGURE 22 OFFSET BINARY OPERATION



	B1	B2	B3	B4	B5	B6	B7	B8	E_O
Pos Full Range	1	1	1	1	1	1	1	1	4.960
Zero Scale	1	0	0	0	0	0	0	0	0.000
Neg Full Scale +1 LSB	0	0	0	0	0	0	0	1	-4.960
Neg Full Scale	0	0	0	0	0	0	0	0	-5.000

ในการทำปริญญานิพนธ์เรื่อง เครื่องวิเคราะห์สัญญาณลอจิกนี้ ผู้จัดทำได้รับคำปรึกษาและชี้แนะแนวทางมาจากผู้ทรงความรู้หลายท่านอาทิเช่น อาจารย์วิทยา ทิพย์สุวรรณพร , คุณอุดม ลิ่วลมไพศาล ผู้ซึ่งให้การสนับสนุนจนสำเร็จเป็นวิทยานิพนธ์ด้วยดี และยังรวมทั้งการอำนวยความสะดวกจากเพื่อนๆ ในห้อง DCS ที่ให้ความร่วมมือเป็นอย่างดี

สรพงษ์ โชติพานิช

สัญญา ยิ้มศิริ