



เครื่องปรับแต่งเสียงอัตโนมัติ
(Automatic Equalizer)

โดย

นาย รัชชัย จิรรังคานนท์ 34103146

นาย นวชัย เกียรติก่อเกื้อ 34103173

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สัจวงศศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034870

เครื่องปรับแต่งเสียงอัตโนมัติ
(Automatic Equalizer)



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของวิชา Project I , II

ประจำปีการศึกษา 2537

ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องปรับแต่งเสียงอัตโนมัติ

(Automatic Equalizer)

นาย รัชชัย จีรวงศ์ ๘4108146

นาย นวชัย เกียรติก่อเกื้อ ๘4108178

อาจารย์ที่ปรึกษา

รศ.ดร. มนต์ สัจวงศศิลป์



อาจารย์ที่ปรึกษา

(รศ.ดร. มนต์ สัจวงศศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องปรับแต่งเสียงอัตโนมัติ

(Automatic Equalizer)

นาย ชวิชัย จิรังกานนท์ 34103146

นาย นวชัย เกียรติก่อเกื้อ 34103173

อาจารย์ที่ปรึกษา

รศ.ดร. มหัท สัจจวรรตศิลป์

ปีการศึกษา 2537



บทคัดย่อ

เครื่องปรับแต่งเสียง มีความสำคัญต่อระบบอุปกรณ์เครื่องเสียงเป็นอันมาก เพราะใช้ปรับแต่งเสียงในห้องรับฟังให้ได้ตามต้องการ จึงเหมาะสมที่จะทำการพัฒนาให้เป็นระบบอัตโนมัติ เพื่อให้ใช้งานได้สะดวกขึ้น โดยการเชื่อมต่อกับคอมพิวเตอร์ และใช้อุปกรณ์เครื่องมือวัดที่เชื่อมต่อกับคอมพิวเตอร์โดยใช้ IEEE - 488 (GPIB) เป็นตัวควบคุม โดยใช้ส่งสัญญาณเสียงที่ความถี่ต่างๆ และรับมาประมวลผล แล้วส่งงานไปยังเครื่องปรับแต่งเสียงอัตโนมัติ ให้ปรับแต่งเสียงให้ได้ตามระดับที่เราต้องการ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC EQUALIZER

Thawatchai Jirarungkanont 34103146

Navachai Kiartkorkuaa 34103173

Advisor:

Assoc.Prof.Dr. Manus Sungworasilpa

Project I , II . 1994



Abstract

Equalizer has importances for sound system. Because it uses for changing sound on purpose. Thus it is suitable to develop into automatic system for convenient uses. By interface it with controller (computer) and use the instruments that can control by computer with IEEE - 488 (GPIB). By use them to send the sound signal in different frequency and receive to process data. Then computer controls the automatic equalizer to calibrate sound for suitable use.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ปริญญาบัตรนี้จะอธิบายถึง วัตถุประสงค์, หลักการ, ทฤษฎี, แนวคิด และ กระบวนการ
สร้างเครื่องปรับแต่งเสียงอัตโนมัติ เครื่องปรับแต่งเสียงอัตโนมัตินี้จะมีประโยชน์ต่อการใช้งาน
ทางด้านระบบเสียง ผู้จัดทำหวังว่า ปริญญาบัตรนี้จะมีประโยชน์ต่อผู้ศึกษากันทั่วทั้งด้านนี้ หาก
มีข้อผิดพลาดประการใด ผู้จัดทำขอรับผิดชอบทุกประการขออภัยล่วงหน้าทั้งหมด ปริญญาบัตรนี้ได้
รับคำปรึกษา, ความช่วยเหลือ และ กำลังใจ จาก อาจารย์ที่ปรึกษา, อาจารย์ภาควิชาอิเล็กทรอนิกส์,
รุ่นพี่, เพื่อน และ รุ่นน้อง ทุกท่าน ผู้จัดทำขอขอบคุณทุกท่าน และ ความดีของปริญญาบัตรนี้ท่าน
เหล่านี้มีส่วนร่วมด้วยทั้งสิ้น.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	เรื่อง	หน้า
1	บทนำ	1
2	IEEE - 488 (GPIB)	4
3	การเขียนโปรแกรมสำหรับควบคุม IEEE - 488 (GPIB) บน MS - DOS	19
4	การเขียนโปรแกรมสำหรับควบคุม IEEE - 488 (GPIB) บน Windows	37
5	Digital Graphic Equalizer	38
6	ส่วนควบคุมระบบ (8081)	48
7	การ Interface กับ GPIB	55
8	ผลการทดลอง สรุปผลการทดลอง และ วิจารณ์ กิตติกรรมประกาศ หนังสืออ้างอิง	58
ภาคผนวก 1	รายการประกาศ NI-488.2 Routines และ NI-488.2 Functions ในการเขียนโปรแกรมควบคุม IEEE - 488 (GPIB)	
ภาคผนวก 2	Data Sheets	
ภาคผนวก 3	Source Codes ของโปรแกรม Automatic Equalizer	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันเราจะพบว่าวงจร Graphic Equalizer เป็นที่นิยมมากในระบบเครื่องเสียงสมัครเล่น จะพบเห็นได้ในเครื่องเสียงทั่วไป โดยใช้เป็นตัวแต่งเสียงแทนระบบ Tone Control สมัยเก่า เนื่องจากข้อดีของ Graphic Equalizer เองเช่น ความละเอียดในการปรับแต่งเสียงในแต่ละความถี่ของเสียงมีมากกว่า หรืออาจจะกล่าวได้ว่ามีจำนวนช่องมากกว่า ขึ้นอยู่กับวงจรที่ได้ออกแบบ ปกติที่พบเห็นในท้องตลาดได้แก่ 3 ช่อง 5 ช่อง 10 ช่อง เป็นต้น ซึ่งแตกต่างจาก Tone Control ซึ่งจะมีเพียง 2 ถึง 3 ช่องความถี่เท่านั้น ได้แก่ Base , Treble หรือบางวงจรอาจจะมี Midrange ด้วย แต่ประโยชน์ของ Graphic Equalizer จริง ๆ นั้นไม่ได้มีเพียงเท่านี้

ประโยชน์ที่แท้จริงของเครื่องปรับแต่งเสียงนั้นคือ ใช้ในการแก้สภาพเสียงที่เกิดจากสภาพแวดล้อมที่แตกต่างกัน เพื่อให้สภาพแวดล้อมนั้นมีสภาพการตอบสนองต่อความถี่ต่าง ๆ ใกล้เคียงกับห้องอัด ซึ่งห้องอัดมีคุณสมบัติพิเศษคือ ผนังห้องจะบุด้วยวัสดุกันการสะท้อนของเสียงเพื่อให้ได้สภาพอ้างอิงที่แท้จริง แต่ในสภาพแวดล้อมอื่น ๆ จะมีผลของเสียงสะท้อนมารบกวนด้วย เช่น อาจจะทำให้เกิดการ Beat ของบางความถี่ทำให้ความถี่เสียงช่องนั้นเกิดดังมากกว่าปกติ (คลื่นที่เกิดการสะท้อนมีเฟสที่เสริมกับคลื่นหลัก) หรือบางความถี่จะเบากว่าปกติ (เนื่องจากคลื่นที่เกิดจากสะท้อนมีเฟสหักล้างกับคลื่นหลัก) เมื่อเกิดสภาวะการเช่นนี้ขึ้นจะต้องมีการนำเอา Graphic Equalizer มาทำการปรับแต่งแก้สภาพแวดล้อมเพื่อให้ได้ใกล้เคียงกับสภาพอ้างอิง ถ้าความถี่ช่วงไหนเกิดการเสริมกันก็จะต้องทำการลดขนาดของสัญญาณลง ถ้าความถี่ไหนหักล้างกันก็จะต้องเพิ่มขนาดของสัญญาณที่ช่วงความถี่นั้นเพิ่มขึ้นเป็นการชดเชย

Graphic Equalizer ที่จะนำมาใช้ในสภาพนี้นั้น จะต้องมิจำนวนช่องมากพอซึ่งอาจจะมากกว่าถึง 31 ช่องต่อข้าง (ถ้าจำนวนช่องยิ่งมากเท่าใด การปรับแต่งเสียงสัญญาณก็จะมีผลละเอียดมากขึ้นเท่านั้น) แต่เนื่องจากจำนวนช่องที่มากนั่นเองทำให้ผู้ที่ทำการควบคุมจะต้องเป็นผู้ที่มีประสบการณ์หรือจะต้องเป็น Sound Engineer เท่านั้น แต่เนื่องจากประเทศไทยมีผู้ที่มีความรู้ด้านนี้น้อยมาก ดังนั้นวิทยานิพนธ์ชิ้นนี้จึงออกแบบเพื่อช่วยในการปรับแต่งง่ายขึ้น โดยออกแบบให้ Graphic Equalizer นี้สามารถทำงานในลักษณะอัตโนมัติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทํางานของ Automatic Equalizer

หน้าที่การทํางานของ Equalizer ได้กล่าวไปแล้วเบื้องต้น ในส่วนนี้เราจะอธิบายถึงหลักการทํางานอย่างคร่าว ๆ ซึ่งสามารถแบ่งได้เป็น 2 ส่วนคือ

1. ส่วนรับรู่การตอบสนองความถี่เสียงของสภาพแวดล้อม (Project I)
2. ส่วนของเครื่องปรับแต่งเสียงที่ทำหน้าที่ชดเชยความถี่ (Project II)

ในการทํางานนั้นเครื่องจะทำการผลิตสัญญาณ Sinewave ความถี่ต่าง ๆ ที่กำหนดไว้ออกมา ซึ่งมีขนาดของสัญญาณคงที่ ผ่านเข้าสู่ชุดของระบบเครื่องเสียงที่ใช้งานจริงออกสู่ลำโพง และเครื่องจะทำการรับสัญญาณกลับด้วยไมโครโฟน สัญญาณที่ได้จะมีการนำขนาดของสัญญาณไปหาผลการตอบสนองความถี่ของสภาพแวดล้อม (ไมโครโฟนที่ใช้จะต้องมีการตอบสนองความถี่ราบเรียบที่สุดตลอดช่วงความถี่เสียง ประมาณ 20 Hz - 20 K Hz) ที่กล่าวมาทั้งหมดนี้เป็นส่วนของ Project I หลังจากที่เราได้ผลการตอบสนองความถี่แล้ว เครื่องจะทำการ Scan ความถี่อีกครั้งหนึ่ง พร้อมกับการปรับแต่งเสียงเพื่อให้ได้ขนาดของสัญญาณเหมือนกับกราฟการตอบสนองความถี่อ้างอิง ซึ่งเป็นส่วนของ Project II

คั้งที่ได้กล่าวมาแล้วว่าเราจะต้องทราบผลการตอบสนองความถี่ของสภาพแวดล้อม คั้งนั้นส่วนนี้จะประกอบไปด้วย ส่วนผลิตสัญญาณ ส่วนรับ Feedback ส่วนบันทึกข้อมูล และส่วนการคำนวณข้อมูล โดยการติดต่อแต่ละส่วนจะเชื่อมต่อกันกับคอมพิวเตอร์ ด้วยระบบ IEEE - 488 (GPIB)

ส่วนผลิตสัญญาณจะใช้ฟังก์ชันเจนเนอเรเตอร์เป็นตัวสร้างสัญญาณ Sinewave เครื่องจะทำการ Scan ความถี่ที่ได้ตั้งไว้แต่ละความถี่ออกไป โดยความถี่ที่ตั้งไว้จะเป็นความถี่ Octave ที่มีขนาดของสัญญาณคงที่ตลอดช่วงการ Scan สัญญาณ Sinewave นี้จะทำหน้าที่เป็นอินพุทของระบบผ่านเข้าสู่ระบบเครื่องเสียงออกสู่ลำโพง และจะทำการรับสัญญาณ Feedback กลับด้วยไมโครโฟน แต่สัญญาณที่ได้จากไมโครโฟนมีขนาดเล็กมาก จึงต้องทำการขยายเบื้องต้นด้วยวงจรขยายสัญญาณก่อน ผ่านเข้าสู่ออสซิลโลสโคปที่ทำหน้าที่ในการ Sampling Amplitude ข้อมูลที่ได้จากการ Sampling ของแต่ละความถี่จะถูกนำไปเปรียบเทียบเพื่อให้ได้ค่าสูงสุดเพื่อนำไปเก็บเป็นข้อมูล หลังจากที่เราได้ข้อมูลของแต่ละความถี่ครบหมดแล้ว จะต้องทำการหาค่าเฉลี่ยเพื่อให้ได้ขนาดของสัญญาณอ้างอิง (ในที่นี้จะใช้การหาค่าเฉลี่ยแบบค่าเฉลี่ยเลขคณิต แต่อาจจะมีการเปลี่ยนแปลงโดยการหาค่าเฉลี่ยด้วยวิธีอื่นได้ตามความเหมาะสม เช่น พัฒนาการหาค่าเฉลี่ยแบบฐานนิยม เพื่อให้ได้เหมาะสมกับการปรับแต่งสัญญาณมากที่สุด) นำขนาดของสัญญาณอ้างอิงที่ได้ไปเปรียบเทียบกับข้อมูลของแต่ละความถี่เพื่อให้ได้ข้อมูลใหม่ในหน่วย db แต่จะเห็นได้ว่าข้อมูลจะเป็นจริงในด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็คือเมื่อผลการตอบสนองความถี่ของไมโครโฟนเป็น 0 db ตลอดย่านความถี่ ซึ่งไม่สามารถเป็นไปได้นั้น ข้อมูลใหม่นี้จะต้องถูกนำไปเปรียบเทียบกับผลการตอบสนองความถี่ของไมโครโฟนอีกครั้งหนึ่งจึงจะได้ข้อมูลที่แท้จริง แล้วจึงสามารถนำไปแสดงผลได้

การควบคุมฟังก์ชันเจเนอเรเตอร์และออสซิลโลสโคปเพื่อให้เชื่อมต่อกับคอมพิวเตอร์ได้นั้น จะใช้การควบคุมเชื่อมต่อ IEEE - 488 (GPIB)

ส่วนต่อไปคือการพัฒนาได้แก่ การออกแบบส่วนของการเชื่อมต่ออุปกรณ์ปรับแต่งเสียงอัตโนมัติ (Automatic Equalizer) กับคอมพิวเตอร์โดยใช้ IEEE - 488 (GPIB) เช่นเดียวกัน โดยการออกแบบระบบการปรับแต่งเสียงซึ่งสามารถควบคุมด้วยข้อมูลที่เป็นดิจิทัลได้ การโปรแกรมทางคอมพิวเตอร์เพื่อให้ง่ายต่อการใช้งานควบคุมเครื่องปรับแต่งเสียงอัตโนมัติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

IEEE - 488 (GPIB).

ในปัจจุบัน อุตสาหกรรมมีอุปกรณ์ที่ใช้งานกันมากขึ้น การเชื่อมต่อกันระหว่างอุปกรณ์ต่าง ๆ ก็มีมากขึ้นด้วย ดังนั้นจึงมีการคิดระบบการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ขึ้นเป็นมาตรฐานขึ้นมา นั่นคือ IEEE-488 นั่นเอง

2.1 ประวัติการพัฒนา IEEE - 488

การเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ นั้นในอดีตก่อให้เกิดความยุ่งยาก และเสียค่าใช้จ่ายมาก ดังนั้น บริษัทผู้ผลิตเครื่องมือวัดต่าง ๆ ในประเทศสหรัฐอเมริกาจึงร่วมกันจัดหาระบบเชื่อมต่อมาตรฐานขึ้นมา ซึ่งในประเทศเยอรมันก็มีการพัฒนาระบบเชื่อมต่อมาตรฐานขึ้นมาเช่นกันโดยความร่วมมือของ IEC (International Electrical Commission) จนในกระทั่งปี 1972 สหรัฐอเมริกาขอยกการนำของ IEEE (Institute of Electrical and Electronics Engineers : IEEE) จึงได้มีการประชุมเพื่อวางแผนพิจารณาเชื่อมต่อมาตรฐานร่วมกัน

บริษัทฮิวเลตต์แพคการ์ด (Hewlett-Packard) ผู้ผลิตเครื่องมือวัดรายใหญ่ในอเมริกา ได้ทำการพัฒนาระบบเชื่อมต่อมาตรฐานอยู่ก่อนแล้วชื่อว่า HPIB (Hewlett Packard Interface Bus) จึงได้เสนอโครงการให้ IEEE เพื่อพิจารณา และได้รับการยอมรับในปี 1975 โดย IEEE จัดให้เป็นมาตรฐานลำดับที่ 488 ดังนั้นจึงได้ชื่อว่า IEEE-Std 488-1975 ซึ่งต่อมาได้มีการปรับปรุงเป็น IEEE-Std 488-1978 หรือที่นิยมเรียกกันว่า GPIB (General Purpose Interface Bus)

2.2 โครงสร้างของ IEEE-488

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์คือ ผู้ส่ง (Talker) , ผู้รับ (Listener) และ ผู้ควบคุม (Controller)

- Talker ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัว แต่จะมีเพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่

- Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัว เช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลาย ๆ ตัวได้

- Controller ทำหน้าที่ควบคุมอุปกรณ์ต่าง ๆ ในระบบ โดยจะกำหนดให้ Talker ทำการส่งข้อมูลหรือกำหนดให้ Listener ทำการรับข้อมูล

ไม่ว่ากรณีใดก็ตาม อุปกรณ์เหล่านี้จำเป็นต้องมีอยู่อย่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งตามหน้าที่ได้ดังนี้

1. ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น
2. ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์ (Printer) , เครื่องบันทึก (Recorder) เป็นต้น
3. ทำหน้าที่เป็นทั้ง Talker และ Listener เช่น คอมพิวเตอร์ , เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น
4. ทำหน้าที่เป็น Talker , Listener , Controller ทั้งหมดในตัวเดียวกัน เช่น คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบ

2.3 พิจารณาข้อกำหนดของ IEEE-488

1. จำนวนอุปกรณ์ภายในระบบ (Talker , Listener , Controller) ที่ต่อกับสายสัญญาณ 1 เส้น จะต้องมีไม่เกิน 15 เครื่อง
2. สายเคเบิลที่ต่อระหว่างอุปกรณ์ จะต้องมียาวไม่เกิน 4 เมตร และ ความยาวรวมของสายเคเบิลในระบบจะต้องยาวไม่เกิน 20 เมตร
3. ความเร็วในการส่งข้อมูลจะต้องไม่เกิน 1 Mb / Sec (1 ล้านไบต์ต่อวินาที)
4. ต้องมีการจ่ายไฟให้อุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

2.4 รายละเอียดเกี่ยวกับ IEEE-488

ลักษณะทางกายภาพของ IEEE-488 นั้นคือ เป็นสายสัญญาณแบบ 24 เส้นขนานกัน และมีขั้วต่ออยู่ทางปลายทั้งสองของสาย เพื่อต่อกับอุปกรณ์ หรือ ต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น ในจำนวนสายสัญญาณ 24 เส้นนี้ มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนที่เหลืออีก 8 เส้น ทำหน้าที่เป็น กราวนด์ (Ground) และ ชีลด์ (Shield)

โดยจำนวนสายที่นำมาใช้นำสัญญาณ 16 เส้นนั้น ยังแบ่งออกได้เป็น 3 ประเภท คือ

1. บัสข้อมูล (Data Bus) จำนวน 8 สาย คือ
 - DIO 1 - DIO 8
2. สายสัญญาณควบคุม (Control Line) จำนวน 5 สาย คือ
 - IFC (Interface Clear)
 - ATN (Attention)
 - SRQ (Service Request)
 - REN (Remote Enable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สายแฮนด์เชค (Hand Shake Line) จำนวน 3 สาย คือ

DAV (Data Valid)

NRFD (Not Ready For Data)

NDAC (No Data Accepted)

2.5 ความหมายของสัญญาณต่าง ๆ ภายใน IEEE-488

สิ่งที่ได้กล่าวมาแล้วว่าสายสัญญาณต่าง ๆ ใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้จะอธิบายความหมายของสัญญาณต่าง ๆ ดังนี้

2.5.1 กลุ่มสัญญาณข้อมูล

1. DIO 1 - DIO 8 (Data Input / Output) สายสัญญาณทั้ง 8 เส้นนี้ทำหน้าที่เป็นทางผ่านของข้อมูลในระบบ

2.5.2 กลุ่มสัญญาณควบคุมการเชื่อมต่อ

1. IFC (Interface Clear) เป็นสัญญาณรีเซ็ต หรือ เคลียร์ระบบ กำหนดได้โดยตัวควบคุมเท่านั้น เมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้ จะกลับคืนสู่สถานะเริ่มต้นใหม่ ซึ่งเป็นสถานะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนแรกเปิดสวิตช์

2. ATN (Attention) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ที่เป็นตัวควบคุมเช่นเดียวกัน ใช้ในการสั่งให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป

3. SRQ (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่าง ๆ เพื่อเป็นการบอกแก่ระบบว่า ขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม

4. REN (Remote Enable) สัญญาณนี้เป็นสัญญาณที่ถูกส่งมาจากตัวควบคุมเพียงตัวเดียวเท่านั้น เพื่อให้สั่งให้อุปกรณ์ต่าง ๆ เปลี่ยนโหมดที่ใช้งานตามปกติมาเป็นการควบคุมโดยตัวควบคุมแทน

5. EOI (End Or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม หรือ อุปกรณ์ที่เป็นตัวส่ง ก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้เสร็จสิ้นลงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 กลุ่มสัญญาณควบคุมการรับส่งข้อมูล

1. DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก Low โดยอุปกรณ์ที่เป็นตัวส่ง เป็นการแจ้งแก่ระบบรับว่า ขณะนี้ตัวส่ง ได้ทำการส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว

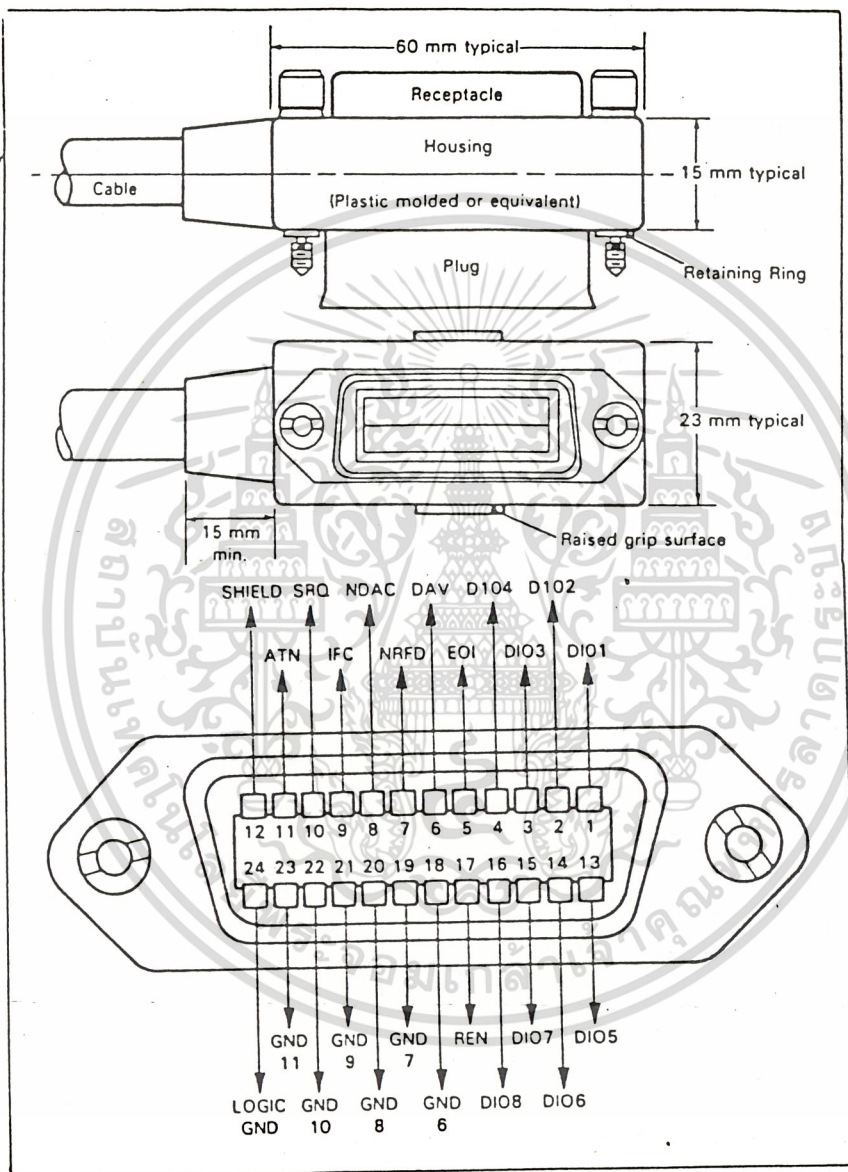
2. NRFD (Not Ready For Data) เมื่อสัญญาณนี้มีลอจิกเป็น Low จะเป็นการแสดงว่าในขณะที่ระบบรับยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ที่เป็นตัวรับกำลังเก็บข้อมูลจากสายข้อมูล และจะเป็น Hi เมื่ออุปกรณ์นั้นได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว

3. NDAC (No Data Accepted) สัญญาณเส้นนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ โดยสัญญาณนี้จะมีลอจิกเป็น Low ในขณะที่อุปกรณ์ที่เป็นตัวรับกำลังเก็บข้อมูลจากสายข้อมูล และจะเป็น Hi เมื่ออุปกรณ์นั้นได้ทำการอ่านข้อมูลเรียบร้อยแล้ว

โดยสัญญาณลอจิกที่ใช้ใน บัสข้อมูล (DIO 1 -DIO 8) ของ IBEE-488 นี้ มีลักษณะเป็นคอมพลิเมนต์ทั้งหมด คือ "1" เท่ากับ Low และ "0" เท่ากับ Hi ซึ่งตรงข้ามกับในวงจรที่ใช้โดยทั่วไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่ 2.1 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่าง ๆ ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2.6 การเชื่อมต่ออุปกรณ์ต่าง ๆ ในระบบ IEEE-488 Bus

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ในระบบ IEEE-488 Bus นั้น มีอยู่ 2 วิธี คือ

1. การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)
2. การเชื่อมต่อแบบกระจาย (Star Configuration)

2.7 ขบวนการแฮนด์เชก (Handshake Procedure)

เมื่อมีการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ในระบบ ดังนั้นเมื่อระบบจะทำงานจึงต้องมีการตรวจสอบก่อน ซึ่งการตรวจสอบนี้เรียกว่า ขบวนการแฮนด์เชก (Handshake Procedure)

พิจารณาระบบที่ไม่ซับซ้อนนัก โดยกำหนดให้มีตัวส่ง (Talker) และตัวรับ (Listener) อย่างละ 1 ตัว

การที่จะมีการสื่อสารระหว่างตัวส่งและตัวรับจะกระทำได้โดยการทำหน้าที่ส่งสัญญาณออกมาเพื่อให้ตัวรับทราบว่ามีการส่งข้อมูลเสร็จในสายสัญญาณ และตัวรับก็จะส่งสัญญาณเพื่อให้ตัวส่งทราบว่าได้รับการรับข้อมูลเป็นที่เรียบร้อยแล้ว

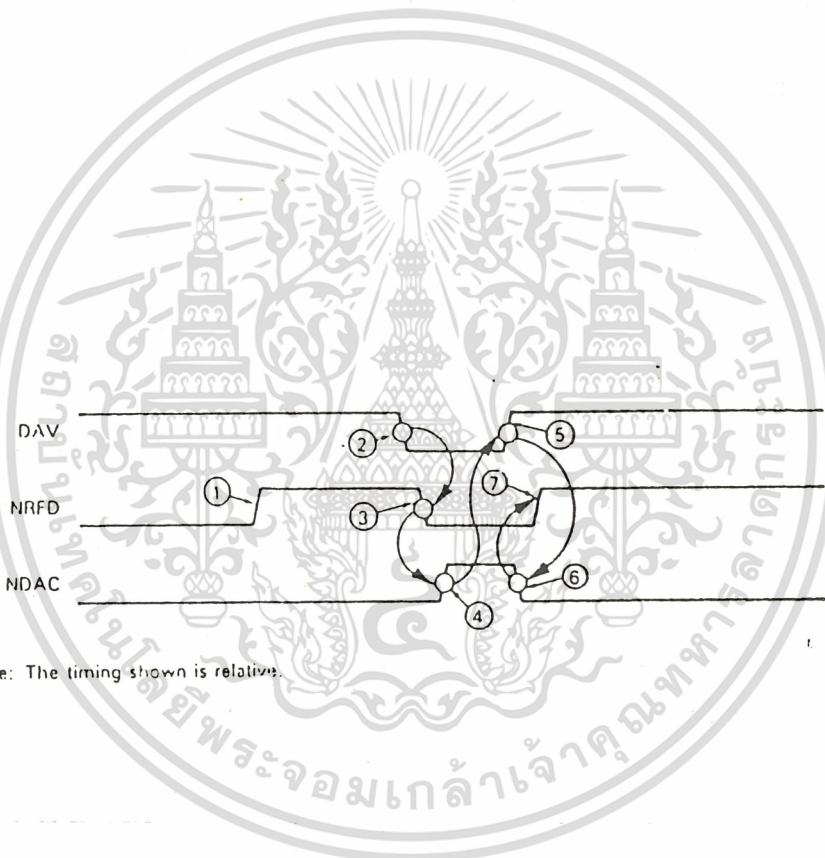
การส่งสัญญาณในการสื่อสารระหว่างตัวส่งและตัวรับนั้น จะส่งมาทางสายสัญญาณ 3 สาย ซึ่งสายสัญญาณดังกล่าวนี้เป็นสายสัญญาณในกลุ่มควบคุมการรับส่งข้อมูล คือ NFRD , DAV และ NDRC

DAV เป็นสัญญาณที่ถูกส่งโดยตัวส่ง NFRD และ NDAC เป็นสัญญาณที่ถูกส่งโดยตัวรับ โดยตัวรับจะใช้สัญญาณ NDAC เพื่อชี้ให้เห็นว่าพร้อมหรือไม่พร้อมที่จะรับข้อมูล

ขบวนการแฮนด์เชกเริ่มขึ้นเมื่อ ตัวส่งมีข้อมูลที่จะถ่ายทอลงสายสัญญาณไปยังตัวรับ เริ่มด้วยการที่ตัวรับจะทำให้สัญญาณ NFRD เป็น Hi (มีค่าเป็น 0) นั่นคือการบอกให้ทราบว่าตัวรับพร้อมที่จะให้ตัวส่งทำการถ่ายข้อมูลลงในสายสัญญาณแล้ว (ในจุดที่ 1) ตัวส่งจะทำการถ่ายข้อมูลลงในสายสัญญาณ DIO 1 - DIO 8 หลังจากนั้นทำการรอเวลา เพื่อให้ตัวส่งถ่ายข้อมูลแล้ว ตัวส่งจะทำให้สัญญาณ DAV มีลอจิกเป็น Low (มีค่าเป็น 1) เพื่อที่จะบอกให้ทราบว่าขณะนี้ได้มีข้อมูลในสายสัญญาณทั้ง 8 เส้นแล้ว (ในจุดที่ 2) ต่อมาเมื่อผู้รับทราบว่าข้อมูลอยู่ในบัสก็จะทำให้สัญญาณ NFRD ให้มีค่าเป็น Low เพื่อบอกให้ตัวส่งทราบว่ายังไม่พร้อมที่จะให้ตัวส่งทำการส่งข้อมูลชุดต่อไป และตัวรับพร้อมที่จะทำการเก็บข้อมูลจากสายสัญญาณ (ในจุดที่ 3) หลังจากตัวรับทำการเก็บข้อมูลจากสายสัญญาณเสร็จแล้ว ตัวรับจะทำให้สัญญาณ NDAC ให้มีสถานะเป็น Hi

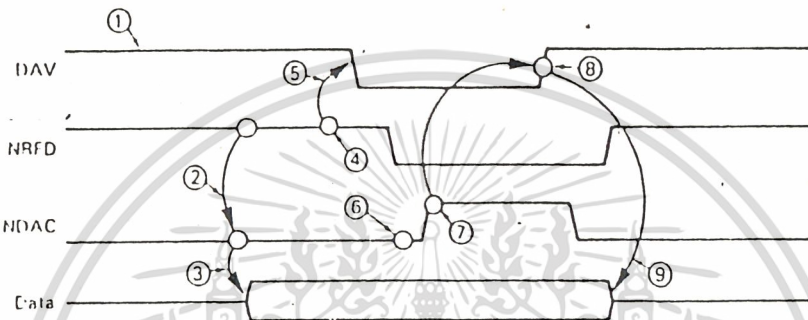
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อบอกให้ทราบว่าได้ทำการเก็บข้อมูลเสร็จแล้ว (ในจุดที่ 4) เมื่อตัวส่งตรวจพบว่าสัญญาณ NDAC มีลอจิกเป็น Hi ก็จะทำให้สัญญาณ DAV ให้มีลอจิกเป็น Hi ก็จะทำให้สัญญาณ NDAC มีลอจิกเป็น Low ทำให้ข้อมูลในบัตลูก้าจะออกไป หลังจากนั้นก็จะทำให้สัญญาณ NRFD เป็น Hi เพื่อบอกให้ทราบว่าจะพร้อมที่จะรับข้อมูลชุดต่อไปแล้วขบวนการแฮนด์เชคจึงเสร็จสิ้นลง และจะเริ่มขึ้นใหม่เมื่อมีสัญญาณชุดใหม่เข้ามา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 2.2 การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นคำสั่ง

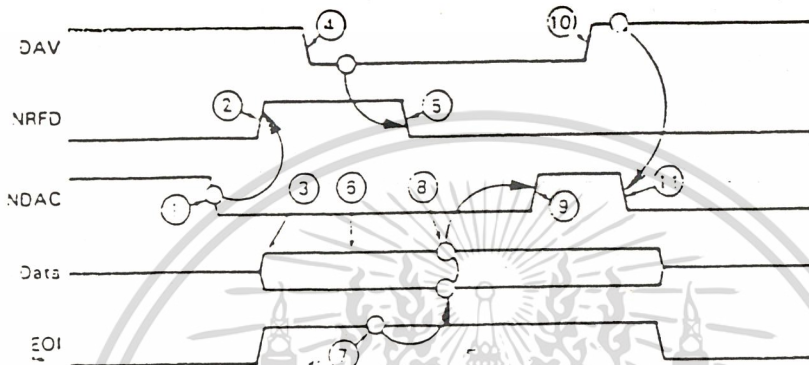


รูปที่ 2.3

ขบวนการแฮนด์เชคเริ่มขึ้นโดยการที่ตัวควบคุม ทำการเซตค่าให้สัญญาณ DAV มีค่าเป็น Hi (ในจุดที่ 1) ต่อมาเป็นการตรวจสอบว่าสัญญาณ NDAC และ NRFD มีลอจิกเป็น Hi ทั้งคู่หรือไม่ (ในจุดที่ 2) ถ้าทั้งคู่เป็น Hi นั่นคือเกิดการผิดพลาด คืออุปกรณ์นั้นไม่พร้อมที่จะทำงาน จึงทำการออกจากขบวนการแฮนด์เชค

กลับมาดูในจุดที่ 2 หากสัญญาณใดสัญญาณหนึ่งมีลอจิกเป็น Low ตัวควบคุมจะทำการส่งข้อมูลลงใน GPIB Bus (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบว่าสัญญาณ NRFD มีลอจิกเป็น Hi (ในจุดที่ 4) หลังจากนั้นตัวควบคุมจะทำให้สัญญาณ DAV ให้มีลอจิกเป็น Low เพื่อให้ตัวรับทราบว่ามีข้อมูลอยู่ในบัสข้อมูล (ในจุดที่ 5) ตัวควบคุมจะรอเวลาเพื่อให้ตัวรับทำการเก็บข้อมูล เมื่อครบกำหนดเวลาตัวควบคุมจะทำการเช็คในตัวรับทำงานเกินเวลาที่กำหนดหรือไม่ หากเกินเวลาก็จะทำการตรวจสอบต่อไปว่า สัญญาณ NDAC ทำลอจิกเป็น Hi ไรหรือไม่ ก็จะทำให้การตรวจสอบอีกครั้งว่าเกินเวลาหรือไม่ กลับไปที่จุดที่ 6 หากสัญญาณ NDAC เป็น Hi (ในจุดที่ 7) ตัวควบคุมจะทำการตอบสนองโดยการทำให้สัญญาณ DAV ให้มีลอจิกเป็น Hi เพื่อบอกให้ตัวรับทราบว่าให้หยุดการเก็บข้อมูลจากบัส (ในจุดที่ 8) หลังจากนั้นข้อมูลในบัสข้อมูลจะถูกนำออกไป (ในจุดที่ 9) จึงเป็นการเสร็จสิ้นขบวนการแฮนด์เชค

2.7.2 ขบวนการแฮนด์เชคของตัวควบคุมซึ่งทำหน้าที่เป็นตัวรับ



รูปที่ 2.4

ขบวนการแฮนด์เชคเริ่มขึ้น หลังจากการที่ตัวควบคุมรับรู้ว่าตัวส่งจะทำการส่งข้อมูลเมื่อตัวควบคุมทราบว่าตัวส่งจะทำการส่งข้อมูล ตัวควบคุมรับรู้ว่าตัวส่งจะทำให้สัญญาณ NDAC มีลอจิกเป็น Low เพื่อบอกให้ทราบว่าตัวควบคุมยังไม่ได้ทำการเก็บข้อมูล (ข้อมูลในที่นี้ คือ แอคเครส หรือ ข้อมูลทั่วไปซึ่งตัวส่งทำการส่งมาในบัสข้อมูล) นั่นคือในจุดที่ 1

ต่อมาตัวควบคุมจะทำการเช็คค่าสัญญาณ NRFD ให้เป็น Hi เพื่อที่จะให้ตัวส่งทราบว่าขณะนี้ตัวควบคุมพร้อมที่จะรับข้อมูลแล้ว (ในจุดที่ 2) เมื่อสัญญาณ NRFD มีค่าเป็น Hi และสัญญาณ NDAC มีค่าเป็น Low แล้ว ตัวส่งจะทราบทันทีว่าทำการส่งข้อมูลลงมารในบัสข้อมูลได้แล้ว (ในจุดที่ 3) หลังจากนั้นตัวควบคุมจะทำการรอเวลาให้ตัวส่งทำการส่งข้อมูลลงในบัสข้อมูลให้เสร็จ โดยจะทำการตรวจสอบไปด้วยว่าเกินเวลาที่กำหนดให้หรือยัง หากเกินเวลาที่กำหนดก็จะออกจากขบวนการแฮนด์เชค หากไม่เกินกำหนดเวลาจะทำการตรวจสอบเวลาว่าเกินเวลาที่ต้องคอยแล้วหรือยัง หากยังไม่เกินเวลาที่จะทำการเชคต่อไปจนกระทั่งหมดเวลาหรือจนกว่าสัญญาณ DAV เป็น Low หรือไม่ หากไม่ก็จะตรวจสอบเวลาว่าเกินเวลาที่ต้องรอแล้วหรือยัง หากยังไม่เกินเวลาที่จะทำการเชคต่อไปจนกระทั่งหมดเวลาหรือจนกว่าสัญญาณ DAV จะมีค่าเป็น Low

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสัญญาณ DAV เป็น Low แล้วตัวควบคุมจะตอบรับ โดยการทำให้สัญญาณ NRFD มีค่าเป็น Low (ในจุดที่ 5) นั้นเป็นการบอกว่าตัวควบคุมพร้อมที่จะเริ่มเก็บข้อมูล (ในจุดที่ 6) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ EOI เพื่อตรวจสอบว่าข้อมูลที่ตัวส่งทำการส่งลงมา นั้นหมดแล้วหรือยัง (ถ้าตัวส่งทำการส่งข้อมูลลงในบัสหมดแล้วจะทำการตั้งค่าสัญญาณ EOI ให้เป็น Low) (ในจุดที่ 7) หากสัญญาณ EOI เป็น Low ตัวควบคุมจะเช็คสถานะว่า ขณะนี้ตัวส่งได้ทำการส่งข้อมูลลงในบัสหมดแล้ว หากสัญญาณ EOI เป็น Hi ก็ไม่ทำการเช็คสถานะ หลังจากนั้นตัวควบคุมจะทำการเป็นข้อมูลในบัส (ในจุดที่ 8) แล้วทำให้สัญญาณ NDAC มีค่าเป็น Hi เพื่อบอกให้ทราบในตัวควบคุมได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว (ในจุดที่ 9) เมื่อตัวควบคุมทำให้สัญญาณ NDAC เป็น Hi แล้วตัวส่งจะลบข้อมูลในบัสข้อมูล โดยการทำให้สัญญาณ DAV เป็น Hi (ในจุดที่ 10) ซึ่งก่อนหน้านี้นั้นตัวควบคุมจะทำการตรวจสอบอยู่แล้วว่าสัญญาณ DAV เป็น Hi หรือยัง เมื่อสัญญาณ DAV เป็น Hi แล้วตัวควบคุมจะทำให้สัญญาณ NDAC มีค่าเป็น Low (ในจุดที่ 11) แล้วจึงออกจากขบวนการแฮนด์เชค

2.8 คำสั่งใช้งานของ GPIB

คำสั่งการต่าง ๆ เพื่อกำหนดหน้าที่การทำงานและกำหนดฟังก์ชัน เช่น กำหนดช่วงการวัด โหมดการวัด หรืออื่น ๆ ให้แก่เครื่องมือวัดที่ต่ออยู่ เหล่านี้มันตัวควบคุมจะเป็นตัวกำหนดโดยการส่งรหัสคำสั่งไปที่อุปกรณ์โดยผ่าน DI 1 - DI 6 รหัสคำสั่งนี้จะถูกส่งไปในช่วงที่สัญญาณ ATN เป็น Low

คำสั่งสำหรับกำหนดหน้าที่การทำงานต่าง ๆ ตามมาตรฐานของ GPIB มีอยู่ด้วยกัน 128 คำสั่งดังแสดงในตารางที่ 1 ต่อไปนี้ โดยแบ่งออกเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIB บัสนั้นใช้ร่วมกัน ทั้งรหัสข้อมูล และ รหัสคำสั่ง นั่นคือรหัสคำสั่งเดียวกันมีความหมายได้ 2 อย่าง คือ เมื่อ ATN เป็น Low จะหมายถึงรหัสคำสั่ง แต่ถ้า ATN เป็น High รหัสนี้จะแทนข้อมูลที่เป็น ASCII แทน ซึ่งในตาราง ก็ได้แบ่งความหมายออกเป็น 2 คอลัมน์

ASCII - IEEE 488 BUS MESSAGES (COMMANDS AND ADDRESSES) HEX CODES

MSD	0		1		2		3		4		5		6		7	
LGD	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16			p	
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	Q	17	a		q	
2	STX		DC2		"	02	2	18	B	02	R	18	b		r	
3	ETX		DC3		#	03	3	19	C	03	S	19	c		s	
4	EOT	SDC	DC4	DCL	\$	04	4	20	D	04	T	20	d		t	
5	ENO	PPC	NAK	PPU	%	05	5	21	E	05	U	21	e		u	
6	ACK		SYN		&	06	6	22	F	06	V	22	f		v	
7	BEL		ETB		'	07	7	23	G	07	W	23	g		w	
8	BS	GET	CAN	SPE	(08	8	24	H	08	X	24	h		x	
9	HT	TCT	EM	SPD)	09	9	25	I	09	Y	25	i		y	
A	LF		SUB		*	10		26	J	10	Z	26	j		z	
B	VT		ESC		+	11		27	K	11	[27	k		[
C	FF		FS		,	12		28	L	12	\	28	l		\	
D	CR		GS		-	13		29	M	13]	29	m]	
E	SO		RS		.	14		30	N	14	^	30	n		^	
F	SI		US		/	15		31	UNL	O	15	_	UNT	o		DEL

ADDRESSED
COMMAND
GROUP

UNIVERSAL
COMMAND
GROUP

LISTEN
ADDRESS
GROUP

TALK
ADDRESS
GROUP

SECONDARY
COMMAND
GROUP

PRIMARY COMMAND GROUP (PCG)

ตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1. กลุ่มคำสั่งเจาะจงจุดหมาย (Addressed Command Group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ที่เป็นตัวส่งหรือตัวรับที่กำหนดไว้ล่วงหน้าแล้ว คำสั่งนี้ประกอบด้วย

GTL (Got To Local) สั่งให้อุปกรณ์กลับสู่สภาพการควบคุมปกติด้วยมือ

SDC (Selected Device Clear) สั่งให้อุปกรณ์เคลียร์ตัวเองให้อยู่ในสภาวะเริ่มต้นใหม่

PPC (Paralled Poll Configure) เป็นคำสั่งสำหรับการจัดสายสัญญาณของการทำการจัดสรรสายสัญญาณของการทำกระบวนการตรวจสอบสภาพอุปกรณ์โดยวิธีขนานโดยใช้กับกลุ่มคำสั่งรอง

GET (Group Excute Trigger) ใช้สั่งเริ่มต้นการทำงานของอุปกรณ์ที่ละหลายตัว

TCT (Take Control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

2.8.2 กลุ่มคำสั่งครอบคลุม (Universal Command Group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่ต่ออยู่ในบัส ประกอบด้วย

LLO (Local Lockout) เป็นการสั่งให้อุปกรณ์ล๊อคอยู่ที่สภาวะควบคุมโดยปุ่มปรับที่หน้าปัทม์ตามปกติ

DCL (Device Clear) สั่งให้อุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น

PPU (Parallel Poll Unconfigure) ใช้ยกเลิกกระบวนการตรวจสอบสภาพแบบขนานทั้งหมด

SPE (Serial Poll Enable) เปลี่ยนโหมดของการตรวจสอบสภาพเป็นแบบอนุกรม ในโหมดนี้จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล

SPD (Serial Poll Disable) ยกเลิกโหมดการตรวจสอบแบบอนุกรม

2.8.3. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ (Listener Address Group) เป็นคำสั่งสำหรับกำหนดให้อุปกรณ์เป็นตัวรับ ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNT (untalker) สำหรับยกเลิก

2.8.4. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวส่ง (Talker Address Group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่ง ตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNL (unlistener) สำหรับยกเลิกเช่นกัน

กลุ่มคำสั่งที่ 1 ถึง 4 นั้น จัดเป็นกลุ่มคำสั่งหลัก ที่มีความหมายตายตัว ยังมีกลุ่มคำสั่งอีกกลุ่มที่ขึ้นอยู่กับข้อกำหนดภายหลัง นั่นคือกลุ่มคำสั่งรอง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.5. กลุ่มคำสั่งรอง (Secondary Command Group) เป็นคำสั่งที่กำหนดรายละเอียดย่อยของอุปกรณ์แต่ละตัวที่ต่ออยู่ในระบบ ให้มีการทำงานอย่างไร ตามจุดประสงค์ใช้งานของเครื่องมือชิ้นนั้น เช่นเกี่ยวกับการปรับปุ่มต่าง ๆ ด้วยมือตนเอง คำสั่งรองนี้จะตามหลังคำสั่งหลัก คือ จะใช้หลังจากอุปกรณ์ต่าง ๆ ถูกกำหนดวางตัวในระบบเรียบร้อยแล้ว

คำสั่งต่าง ๆ ที่กล่าวไป ซึ่งใช้การกำหนดสถานะการทำงานของอุปกรณ์ แต่ละสถานะที่กำหนดไปนั้นเป็นอย่างไร และมีจุดประสงค์เพื่ออะไร ดังต่อไปนี้

Device Clear / Interface Clear

Device Clear ใช้ในการทำให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปสู่สถานะเริ่มต้น ยังไม่มีการกำหนดฟังก์ชันใด ๆ สถานะเริ่มต้นนี้จะแตกต่างกันไปแล้วแต่ว่าอุปกรณ์นั้นออกแบบไว้อย่างไร Device Clear มีอยู่ 2 ลักษณะ คือ เคลียร์หมดทุกตัวที่ต่ออยู่ (DCL) กับเคลียร์เฉพาะจงอุปกรณ์ตัวใดตัวหนึ่ง (SDC)

แต่ว่าในการเคลียร์อุปกรณ์ให้อยู่ในสถานะเริ่มต้นนั้น ไม่ได้หมายความว่า Interface Function ของ GPIB จะถูกเคลียร์อุปกรณ์ให้ไปอยู่ในสถานะเริ่มต้นด้วยแต่อย่างใด Interface Function คือ สภาพการ Interface ที่ได้กำหนดในระบบประกอบด้วยฟังก์ชันต่าง ๆ

Remote / Local

Remote เป็นการกำหนดให้อุปกรณ์ที่อยู่ในระบบเช่น เครื่องมือวัดให้อยู่ในการควบคุมของอุปกรณ์ของอุปกรณ์ตัวอื่นแทน ซึ่งปุ่มปรับต่าง ๆ บนหน้าปัทม์เครื่องจะไม่มีผลต่อการทำงาน ส่วน Local เป็นการควบคุมการทำงานของเครื่องมือวัดด้วยปุ่มปรับบนหน้าปัทม์ตามปกติ

การใช้ Remote มีประโยชน์ในแง่ที่ขณะที่ตัวการควบคุม เช่น คอมพิวเตอร์กำลังติดต่ออุปกรณ์ตัวนั้นอยู่ หากไม่มีการจัดการควบคุมโดยปุ่มปรับบนหน้าปัทม์ออก ถ้ามีใครมาปรับแต่งก็จะทำให้การทำงานผิดพลาดไปได้ การทำงานของ GPIB ใน Remote and Local มี 4 ลักษณะดังนี้

1. LOCS ก็คือ Local นั่นเอง เป็นสภาพการควบคุมที่ปุ่มตามปกติ จะอยู่ในสภาพนี้ตอนเปิดเครื่อง หรือ REN เป็น Hr หรือเมื่อได้รับคำสั่ง GTL

2. REMS คือ Remote หมายถึงการตัดการควบคุมโดยปุ่มหน้าปัทม์ออก จะเกิดขึ้นเมื่อ REN เป็น Low และจะถูกล็อกไว้ เว้นแต่ว่าสวิทช์ Local ที่ตัวอุปกรณ์จะถูกเปลี่ยนไปตำแหน่ง Local

การนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. RWLS เป็นสภาพ Remote ที่ถูกล็อกไว้เช่นเดียวกัน แต่จะตัดการควบคุมตรงสวิตช์ Local ที่ตัวอุปกรณ์ออกไป สภาพ Remote โดย RWLS จึงมีความสำคัญสูงกว่า REMS อย่างไรก็ตามยังถูกยกเลิกได้ด้วยคำสั่ง LLO

4. LWLS มีสภาพเช่นเดียวกับ Local แต่จะแตกต่างกันตรงที่สภาพ Local โดย LWLS นี้เมื่อได้รับคำสั่งกำหนดอุปกรณ์ตัวรับจะเปลี่ยนไปอยู่ในสภาพแบบล็อก หรือ LWLS ทั้งนี้ในการที่จะมาที่สภาพ LWLS นี้ได้ก็มี 2 กรณีคือ เมื่ออยู่ในสภาพ Local ธรรมดา (LOCS) แล้วได้รับคำสั่ง LLO หรืออยู่ใน RWLS แล้วได้รับคำสั่ง GTL

2.9 การขอบริการและการตรวจสอบ (Service Request and Polling)

เมื่อตัวควบคุมได้รับ SRQ เป็น Low จะให้อุปกรณ์ส่งข้อมูลแสดงสถานะการทำงานซึ่งมีอยู่ 2 วิธี คือ

2.9.1. การตรวจสอบแบบอนุกรม ซึ่งมีขั้นตอนดังนี้

1.1 ATN ถูกดึงเป็น Low หลังจากได้รับ Low จากสายสัญญาณ SRQ

1.2 คำสั่ง UNL ถูกส่งไปยังอุปกรณ์

1.3 ตัวควบคุมจะแจ้งรหัสตัวรับของตน และกำหนดรหัสตัวส่งของอุปกรณ์ที่จะตรวจสอบไปที่บัส

1.4 ตามด้วยคำสั่ง SPE และสาย ATN กลายเป็น Hi ซึ่งอุปกรณ์ที่ถูกเรียกจะส่งข้อมูลแสดงสถานะออกมา 1 ไบต์ โดยที่บิตที่ 7 จะเป็นตัวชี้ว่าอุปกรณ์เส้นเป็นตัวขอบริการ ถ้าใช่จะเป็น Low ส่วนบิตอื่น ๆ ก็ใช้บอกข้อมูลอื่น ๆ ซึ่งมีได้กำหนดเฉพาะ

1.5 สาย ATN ถูกดึงเป็น Low อีกที เพื่อส่งคำสั่งยกเลิกการตรวจสอบคือ SPD

1.6 จากนั้นคำสั่ง UNT ก็ถูกส่งไปยังอุปกรณ์เพื่อยกเลิกการเป็นตัวส่งซึ่งถ้าหากเป็น SRQ ยังคงเป็น Low อยู่ ก็จะมีการตรวจสอบไปยังอุปกรณ์ตัวอื่น ๆ ต่อไปตามขั้นตอนเดิม

2.9.2. การตรวจสอบแบบขนาน สามารถทำได้เร็วกว่าแบบอนุกรมทั้งนี้เพราะสามารถอ่านข้อมูล เพราะสามารถอ่านข้อมูลเพียง ไบต์เดียวก็สามารถรู้ได้ทันที ว่าอุปกรณ์ตัวใดเป็นผู้ขอบริการ

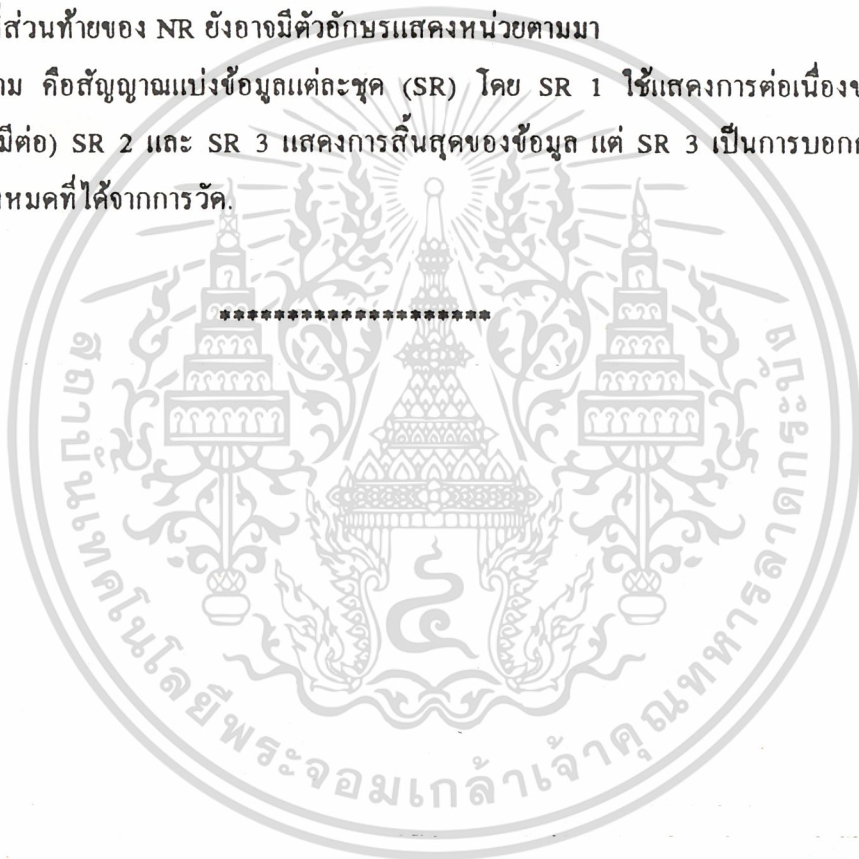
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 รูปแบบของข้อมูล

โดยทั่วไปข้อมูลจากอุปกรณ์ (Device Message) แบ่งได้ออกเป็น 3 ส่วน อันได้แก่ส่วนหัว (HR) ซึ่งจะอยู่ส่วนหน้าสุด เป็นตัวบอกชนิดข้อมูล ส่วนประกอบ HR จะประกอบด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ ช่วงว่างที่เป็นเว้นวรรค () ปกติจะมีอักษรประมาณ 1-3 ตัว

ส่วนที่สองคือเนื้อหาข้อมูล (NR) ซึ่งใช้แสดงค่าตัวเลข มีอยู่ 3 แบบ คือ NR 1 , NR 2 และ NR 3 โดยที่ส่วนท้ายของ NR ยังอาจมีตัวอักษรแสดงหน่วยตามมา

ส่วนที่สาม คือสัญญาณแบ่งข้อมูลแต่ละชุด (SR) โดย SR 1 ใช้แสดงการต่อเนื่องของข้อมูล (ข้อมูลยังมีต่อ) SR 2 และ SR 3 แสดงการสิ้นสุดของข้อมูล แต่ SR 3 เป็นการบอกการเสร็จสิ้นข้อมูลทั้งหมดที่ได้จากการวัด.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การเขียนโปรแกรมสำหรับควบคุม IEEE-488 (GPIB) บน MS-DOS

โครงการชิ้นนี้ได้ใช้อุปกรณ์ของบริษัท National Instrument Corporation. ซึ่งเป็น Interface Card สำหรับการควบคุม IEEE-488 โดยเฉพาะ ซึ่งอุปกรณ์ของ Interface Card นี้ ประกอบด้วย ส่วน Hardware และ Software ซึ่งเป็นพื้นฐานสำหรับการเขียนโปรแกรมสำหรับควบคุม IEEE-488 ซึ่งโครงการนี้จะใช้ภาษา C ในการเขียนโปรแกรมทั้งหมด

3.1 IEEE-488 Interface Board

GPIB-PCII/PIA เป็น IEEE-488 interface board สำหรับเครื่อง IBM PC ,PC/XT,PC AT หรือ คอมพิวเตอร์ซึ่งเทียบเท่า ซึ่งบริษัท National Instrument Corp. ได้สร้างขึ้นเพื่อใช้สำหรับควบคุม IEEE-488 โดยเฉพาะ

3.2 IEEE-488 Program

NI-488.2 คือ Software ซึ่งช่วยในการเขียนโปรแกรมสำหรับควบคุม IEEE-488 โดยเฉพาะ ซึ่งมี Routines สำหรับการเขียนโปรแกรม ซึ่งบริษัท National Instrument Corp. ผลิตเพื่อใช้ในการเขียนโปรแกรมควบคุม Interface Board สำหรับ IEEE-488 ซึ่งประกอบด้วยโปรแกรมในภาษา C และ QuickBasic โดยใช้สำหรับการเขียนโปรแกรมบน DOS

3.3 ขั้นตอนการเขียนโปรแกรมสำหรับควบคุม IEEE-488

การเขียนโปรแกรมสำหรับควบคุม IEEE-488 โดยใช้ NI-488.2 routines มีขั้นตอนต่าง ๆ ดังจะอธิบายต่อไปโดยจะกล่าวถึงเฉพาะโปรแกรมที่เขียนโดยภาษา C เท่านั้น โดยโปรแกรมที่จะเขียนนี้จะใช้ควบคุมอุปกรณ์โดยผ่าน GPIB-PCII/PIA Interface Board ซึ่งทั้ง NI-488.2 และ GPIB-PCII/PIA ถูกสร้างให้มาใช้งานร่วมกัน

ขั้นตอนการเขียนโปรแกรมสำหรับควบคุม IEEE-488 มีดังต่อไปนี้

1. การเตรียมพร้อมสำหรับโปรแกรม (Preparation)

ขั้นตอนแรกสำหรับการเขียนโปรแกรมคือการเรียก NI-488.2 routines จากแฟ้มข้อมูล ซึ่งเป็นการประกาศในแฟ้มข้อมูลจาก NI-488.2 C Language Interface ดังแสดงในตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
```

```
.....
```

```
#include "decl.h"
```

หลังจากนั้นก็เป็นการประกาศตัวแปรสำหรับการเขียนโปรแกรมต่อไป

สำหรับเพิ่มข้อมูล DECL.H เป็น Header File ซึ่งประกาศ Function สำหรับการเรียกใช้เพื่อช่วยในการเขียนโปรแกรมสำหรับควบคุม IEEE-488 ซึ่งจะกล่าวถึงในภายหลัง

2. การเริ่มต้นสำหรับการเขียนโปรแกรม (Initialization)

การเริ่มต้น IEEE-488 Bus และ GPIB-PCII/IA ให้ใช้งานควบคุม อุปกรณ์ IEEE-488 ใต้นั้นจะต้องมีการส่ง Interface Clear ไปยังอุปกรณ์ก่อนทุกครั้ง ดังแสดงในตัวอย่าง

```
SendIFC(Board);
```

ฟังก์ชัน SendIFC ใช้สำหรับการส่ง Interface Clear ซึ่งจะกล่าวอย่างละเอียดในภายหลัง

3. การเริ่มต้นสำหรับอุปกรณ์ต่าง ๆ (Initialize the Instrument)

จากนั้นก็เป็นการส่งเคลียร์อุปกรณ์ต่าง ๆ บน IEEE-488 Bus เพื่อให้เข้าสู่สภาวะเตรียมพร้อม โดยมีการส่ง Device Clear ไปยังอุปกรณ์บนบัสก่อน

```
DevClear(Board,Dev_address);
```

ฟังก์ชัน DevClear ใช้สำหรับการส่ง Device Clear ซึ่งจะกล่าวอย่างละเอียดภายหลัง

4. การตั้งอุปกรณ์ (Configure the Instrument)

หลังจากเตรียมความพร้อมและเริ่มต้นอุปกรณ์ทั้งหมดแล้วก็จะเป็นการตั้งอุปกรณ์ให้ปฏิบัติตามความต้องการ โดยเป็นการส่งคำสั่งให้แก่แต่ละอุปกรณ์ โดยคำสั่งที่ส่งไปให้แก่แต่ละอุปกรณ์จะแสดงอยู่ในคู่มือของแต่ละอุปกรณ์ ดังแสดงในตัวอย่าง

```
Send(Board,Dev_address,"<command>",counter,end);
```

ฟังก์ชัน Send ใช้สำหรับการส่งคำสั่งไปยังอุปกรณ์ต่าง ๆ ซึ่งจะกล่าวอย่างละเอียดภายหลัง

5. การรับข้อมูลจากอุปกรณ์ (Read the Instrument)

การอ่านข้อมูลจากการวัดนั้นจะเกิดขึ้นหลังจากเราส่งคำสั่งเพื่อให้อุปกรณ์ส่งข้อมูลที่เรากำลังต้องการไปแล้ว โดยจะใช้คำสั่ง

```
Recieve(Board,Dev_address,Data,counter,end);
```

ฟังก์ชันต่าง ๆ ที่ถูกเรียกใช้จะกล่าวอย่างละเอียดในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 NI-488.2 Routine

ในหัวข้อนี้จะกล่าวถึงฟังก์ชันต่าง ๆ ใน NI-488.2 โดยจะอธิบายเฉพาะฟังก์ชันที่สำคัญ ๆ (โดยเรียงลำดับตามตัวอักษรภาษาอังกฤษ) ซึ่งฟังก์ชันเหล่านี้อยู่ในแฟ้มข้อมูล DECL.H (เฉพาะใน ภาษา C)

DevClear

วัตถุประสงค์

ใช้สำหรับเคลียร์อุปกรณ์ทีละตัว

รูปแบบการใช้

```
void Devclear (short board,unsigned short address)
```

คำอธิบาย

board หมายถึงหมายเลขประจำบอร์ดของ GPIB-PCII/PIA ซึ่งโดยปกติ board จะเป็น 0 ที่ address 2B8 (สำหรับ GPIB-PCII) ส่วน address หมายถึง address ของอุปกรณ์ต่าง ๆ บนบัส

ตัวอย่าง

```
unsigned short address=12;
```

```
DevClear(0,address);
```

จะหมายถึงการตั้ง Device Clear ไปยังอุปกรณ์บนบอร์ด 0 ตัวที่ 12 นั้นเอง

EnableLocal

วัตถุประสงค์

ใช้สำหรับการยินยอมให้ทำการปฏิบัติการจากแผงหน้าปัดของอุปกรณ์แต่ละตัวได้

รูปแบบการใช้

```
void EnableLocal (short board,unsigned short address)
```

คำอธิบาย

เช่นเดียวกับคำสั่ง DevClear

ตัวอย่าง

```
unsigned short address=12;
```

```
EnableLocal(0,address);
```

EnableRemote

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์

ใช้สำหรับการควบคุมอุปกรณ์โดยตัว GPIB โปรแกรม

รูปแบบการใช้

```
void EnableRemote (short board,unsigned short address)
```

คำอธิบาย

เช่นเดียวกับคำสั่ง DevClear

ตัวอย่าง

```
unsigned short address=12;
```

```
EnableRemote(0,address);
```

Receive

วัตถุประสงค์

อ่านข้อมูลจากอุปกรณ์ GPIB

รูปแบบการใช้

```
void Recieve (short board,unsigned short address,unsigned char data[],unsigned long count,unsigned short termination)
```

คำอธิบาย

board หมายถึงหมายเลขประจำบอร์ด GPIB-PCII/PIA ส่วน address หมายถึง address ของอุปกรณ์ต่าง ๆ บนบัส data หมายถึงตัวแปรที่ใช้สำหรับเก็บข้อมูลที่รับมาได้ count หมายถึงจำนวนอักขระในสายข้อมูล data ส่วน termination หมายถึงรูปแบบการสิ้นสุดของข้อมูล

ตัวอย่าง

```
unsigned char data[100];
```

```
Receive (0,12,data,100,STOPend);
```

จะหมายถึงการสั่งให้รับข้อมูลจากบอร์ด 0 อุปกรณ์ตัวที่ 12 มาเก็บไว้ใน data และมีการสิ้นสุดของข้อมูลเป็นแบบ Stopend

Send

วัตถุประสงค์

ส่งคำสั่งไปยังอุปกรณ์ GPIB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ

void Send (short board,unsigned short address,unsigned char data[],unsigned long count,unsigned short eotmode)

คำอธิบาย

board หมายถึงหมายเลขประจำบอร์ด GPIB-PCII/LA ส่วน address หมายถึง address ของอุปกรณ์ต่าง ๆ บนบัส data หมายถึงตัวแปรที่ใช้สำหรับเก็บคำสั่งที่จะส่งไป count หมายถึงจำนวนอักขระในสายข้อมูล data ส่วน eotmode หมายถึงรูปแบบการสิ้นสุดของข้อมูลแบ่งออกเป็น NLEnd คือส่ง NL กับ EOI หลังจากส่งข้อมูล , DABend คือส่งเฉพาะ EOI ไป , NULLend คือไม่ส่งอะไรไปเลย

ตัวอย่าง

```
Send (0,12,"FRQ:",5,NLEnd);
```

หมายถึงการส่งคำสั่ง FRQ: ไปยังบอร์ดที่ 0 อุปกรณ์ตัวที่ 12

สำหรับคำสั่งอื่น ๆ จะไม่กล่าวโดยละเอียดแต่แสดงไว้ในตารางด้านล่าง

โดยที่ ตารางที่ 3.1 จะแสดงถึง Routines สำหรับภาษา C

และ ตารางที่ 3.2 จะแสดงถึง Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
AllSpoll(board, addressl,resultl)	Serial poll all devices
DevClear(board, address)	Clear a single device
DevClearList (board,addressl)	Clear multiple devices
EnableLocal(board , addresslist)	Enable operations from the front of devices
EnableRemote (board,addressl)	Enable remote GPIB programming of devices
FindLstm(board,ad dressl,resl,lim)	Find all listeners
FindRQS(board, addressl,result)	Determine which device is requesting service
GenerateREQF (board,addr)	Cancel service request
GenerateREQT (board,addr)	Request service

ตารางที่ 3.1 NI-488.2 Routines สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
GotoMultAddr(brd, type, addf, spollf)	Enable multiple primary or secondary address support
PassControl (board, address)	Pass control another device with Controller capability
Ppoll (board, result)	Perform a parallel poll
PpollCoonfig(brd, add, dtline, sense)	Configur a device for parallel polls
PpollUnconfig (board, address)	Unconfigure devices for parallel polls
RcvRespMsg(board, data, termination)	Read data bytes from already addressed device
ReadStatusByte (brd, add, result)	Serial poll a single device to get its status byte
Recieve(board, add count, terminate)	Read data bytes from a GPIB device
RecieveSetup (board, address)	Prepare a particular device to send data bytes and prepare GPIB to read

ตารางที่ 3.1 NI-488.2 Routines สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ResetSys (board,addressl)	Initialize a GPIB system on three levels
Send(brd,address, data,eotmode)	Send data bytes to a single GPIB device
SendCmds(board, commands,count)	Send GPIB command bytes
SendDataBytes(brd ,dat,cnt,eotmode)	Send data bytes to already addressed devices
SendIFC(board)	Clear the GPIB interface functions with IFC
SendList(brd,addl ,dat,cnt,eotmode)	Send data bytes to multiple GPIB devices
SendLLO(board)	Send the local lockout message to all devices
SendSetup(board, addresslist)	Prepare particular devices to receive data byte
SetRWLS(board, addresslist)	Place particular devices in the Remote with Lockout state

ตารางที่ 3.1 NI-488.2 Routines สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
TestSRQ (board,result)	Determine the current state of the SRQ line
TestSys(board, addl,resultlist)	Cause devices to conduct self-tests
Trigger (board,address)	Trigger a single device
Triggerlist(broad ,addresslist)	Trigger multiples devices
WaitSRQ (board,result)	Wait until a device asserts Service Request

ตารางที่ 8.1 NI-488.2 Routines สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ibbna (ud,bname[])	Change access board device
ibcac (ud,v)	Become Active Controller
ibclr (ud)	Clear specified device
ibcmd (ud,cmd[],cnt)	Send GPIB command messages
ibcmda (ud,cmd[],cnt)	Send commands asynchronously from string
ibconfig (ud,option,cnt)	Change the software configuration parameters
ibdev(brdind,pad, sad,tmo,eot,eos)	Open and initialize an unused device when device name is unknown
ibdma (ud,v)	Enable or disable DMA
ibeos (ud,v)	Change or disable End-of-String termination mode

ตารางที่ 3.2 NI-488.2 Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ibeot (ud,v)	Enable/disable END message on write operations
ibevent (ud,*event)	Return the next event
ibfind (udname[])	Open device and return unit descriptor associated with given name
ibgts (ud,v)	Go from Active Controller to Standby
ibist (ud,v)	Set or clear individual status bit for Parallel Polls
iblines (ud,*clines)	Return the status off the GPIB control lines
ibln(ud,pad,sad, *listen)	Check for the presence of a device on the bus
ibloc (ud)	Go to local
ibonl (ud,v)	Place the device or interface board online or offline

ตารางที่ 3.2 NI-488.2 Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ibpad (ud,v)	Change Primary Address
ibpct (ud)	Pass Control
ibppc (ud,v)	Parallel Poll Configure
ibrd (ud,rd[],cnt)	Read data from a device to a string
ibrda (us.rd[],cnt)	Read data asynchronously to string
ibrdf (ud,flname[])	Read data from GPIB to file
ibrpp (ud,*ppr)	Conduct a Parallel Poll
ibrsc (ud,v)	Request or release system control
ibrsp (ud,*spr)	Return serial poll byte

ตารางที่ 3.2 NI-488.2 Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ibrsv (ud,v)	Request service and/or set or change the serial poll status byte
ibsad (ud,v)	Change or disable Secondary Address
ibsic (ud)	Send interface clear for 100 usec
ibsre (ud,v)	Set or clear the Remote Enable line
ibsrq (*func)	Register an SRQ "interrupt routine"
ibstop (ud)	Abort asynchronous operation
ibtmo (ud,v)	Change or disable time limit
ibtrap (mask,mode)	Alter trap and display modes of the Applications Minitor
ibtrg (ud)	Trigger selected device

ตารางที่ 3.2 NI-488.2 Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
<code>ibwait</code> (ud,mask)	Wait for selected event
<code>ibwrt</code> (ud,wrt[],cnt)	Write data from string
<code>ibwrta</code> (ud,wrt[],cnt)	Write data asynchronously from string
<code>ibwrta</code> (ud,fname[])	Write data from file

ตารางที่ 3.2 NI-488.2 Functions สำหรับภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.5 รายละเอียดเกี่ยวกับอุปกรณ์เครื่องมือวัดที่ใช้ในโครงการงาน

8.5.1 ฟังก์ชันเจเนอเรเตอร์ HM8130 (Function Generator HM8130)

ฟังก์ชันเจเนอเรเตอร์ HM8130 เป็นอุปกรณ์อเนกประสงค์สำหรับเป็นตัวสร้างสัญญาณทุกชนิด โดยประกอบด้วยตัวซินธิไซเซอร์ (Synthesizer), สวิทช์เจเนอเรเตอร์ (Sweep Generator) และ รูปคลื่นอาร์บิทรารี (Arbitrary Waveform) หรือสร้างสัญญาณขึ้นมาเองโดยประกอบด้วยหน่วยความจำ 1024 ตำแหน่งสำหรับแนวตั้งและแนวขวาง โดยสามารถรับข้อมูล อ่านข้อมูลได้โดยผ่านเป็นควบคุมภายนอก หรือ ใช้การเชื่อมต่อแบบ IEEE-488

สำหรับอุปกรณ์นี้สามารถสร้างสัญญาณโดยการกวาดไปตามความถี่ต่าง ๆ โดยเป็นลักษณะเชิงเส้นได้ อุปกรณ์ชนิดนี้มีระยะความถี่ของสัญญาณ sine และ สัญญาณ square เป็นระยะ 10 mHz ถึง 10 MHz

อุปกรณ์นี้สามารถเขียนโปรแกรมควบคุมได้โดยผ่านการเชื่อมต่อทั้งแบบ IEEE-488 และ RS-232

การเชื่อมต่อ HM8130 กับ IEEE-488

การเชื่อมต่อกับ IEEE-488 มีคำสั่งต่าง ๆ ดังต่อไปนี้

คำสั่งที่ไม่มีข้อมูลด้วย

SIN	คำสั่งสำหรับสร้างสัญญาณ Sine
TR	คำสั่งสำหรับสร้างสัญญาณ Triangle
SQR	คำสั่งสำหรับสร้างสัญญาณ Square
PLS	คำสั่งสำหรับสร้างสัญญาณ Pulse
RMP	คำสั่งสำหรับสร้างสัญญาณ Sawtooth แบบ Positive-going
RMN	คำสั่งสำหรับสร้างสัญญาณ Sawtooth แบบ Negative-going
ARB	คำสั่งสำหรับสร้างสัญญาณ Arbitrary
SW1/0	คำสั่งสำหรับ Sweep mode on/off
CTM	คำสั่งสำหรับ Continuous mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ การสั่งให้สร้างสัญญาณรูปสามเหลี่ยม ความถี่ 12.3 KHz เอาท์พุทออกที่เอาท์พุท 1 โดยมีขนาด 10 โวลต์

3.5.2 ออสซิลโลสโคป HM1007 (Oscilloscope HM1007)

ออสซิลโลสโคป HM1007 เป็นออสซิลโลสโคปอเนกประสงค์ซึ่งสามารถปฏิบัติการในรูปแบบดิจิตอลได้เหมือนการปฏิบัติการในรูปแบบอนาล็อก (เวลาจริง) ซึ่งสามารถทำการเชื่อมต่อกับคอมพิวเตอร์ได้ ในรูปแบบการควบคุมของ IEEE-488 โดยใช้อุปกรณ์สำหรับการเชื่อมต่อกับสายสัญญาณของ IEEE-488 คือ HO 79-4/7 ซึ่งสามารถปฏิบัติการตามวัตถุประสงค์ต่าง ๆ ดังนี้

- ค่าอนาล็อกสำหรับ X-Y Recorder
- การต่อขานานสำหรับอุปกรณ์ Contronics connector
- RS 232 สำหรับการเชื่อมต่อสัญญาณแบบสองทาง
- IEEE-488 สำหรับการเชื่อมต่อสัญญาณแบบสองทางและการปฏิบัติการแบบ Talk-

only

ซึ่งมีรูปแบบของการส่งข้อมูลซึ่งแตกต่างกันในการใช้

3.5.3 อุปกรณ์เชื่อมต่อสำหรับออสซิลโลสโคป HO 79-4/7 (Interface for HAMEG Analog/Digital Oscilloscope)

สำหรับการเชื่อมต่อออสซิลโลสโคปกับ IEEE-488 นั้นสามารถใช้คำสั่งดังต่อไปนี้

- ID? คำสั่งสำหรับเรียกค่าประจำตัวของอุปกรณ์ออสซิลโลสโคป
- DIG คำสั่งสำหรับเรียกข้อมูลซึ่งถูกเก็บไว้ในหน่วยความจำของออสซิลโลสโคปโดยสามารถใช้ได้ทั้งในแบบไม่เจาะจงช่องหรือเจาะจงช่อง ตัวอย่างเช่น DIG 3 คือการเจาะจงว่าจะเรียกจากช่องที่ 1 และ 2
- GET คำสั่งสำหรับใช้รีเซ็ต Trigger function ของสโคปโดยใช้ได้ทั้งแบบไม่เจาะจงช่องและเจาะจงช่อง ตัวอย่างเช่น GET 2 คือการเจาะจงจะให้สิ่งที่ช่องที่ 2 ช่องเดียว
- STA คำสั่งสำหรับการเซ็ทช่องของออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การเขียนโปรแกรมควบคุม IEEE-488 บน Windows

ในบทนี้จะกล่าวถึงวิธีซึ่งสามารถใช้ในการติดต่อกับอุปกรณ์ GPIB ผ่านทาง Windows โดยการเขียนโปรแกรมบนวินโดวส์

วิธีการติดต่อกับ GPIB โดยการเขียนโปรแกรมผ่านทางวินโดวส์จะใช้ DLL (Dynamic Link Library) ซึ่งประกอบด้วย NI-488 Functions และ NI-488 Routines

DLL Direct Entry NI-488 Functions and NI-488.2 Routines

การเขียนโปรแกรมควบคุม GPIB ผ่านทางวินโดวส์นี้ จะใช้เพิ่มข้อมูล GPIB.DLL ซึ่งเป็น Dynamic Link Library ซึ่งสามารถเรียกใช้ได้จากภาษาหรือการโปรแกรมต่าง ๆ ซึ่งใช้บนวินโดวส์ ซึ่งในการโปรแกรมบนวินโดวส์นี้ จะใช้ ภาษา Visual Basic 3.0 ซึ่งเป็นภาษาซึ่งใช้สำหรับการเขียนโปรแกรมบนวินโดวส์โดยเฉพาะ

ตัวอย่างการเขียนโปรแกรมควบคุม IEEE-488 บน Windows โดยใช้ Visual Basic

ในส่วนนี้จะยกตัวอย่างการเขียนโปรแกรมบนวินโดวส์โดยใช้ภาษา Visual Basic

Declaration

Declare Function DLLFind Lib "gpiib.dll" (ByVal udname\$, ibsta%, iberr%, ibcntl&) As Integer

Declare Sub DLLSendIFC Lib "gpiib.dll" (ByVal board%, ibsta%, iberr%, ibcntl&)

Use functions or subs in application

Boardud% = DLLFind("GPIB0", ibsta%, iberr%, ibcntl&)

Call DLLSendIFC (0, ibsta%, iberr%, ibcntl&)

แต่การใช้ GPIB.DLL นั้นต้องมีข้อแม้ว่า เพิ่มข้อมูลนี้จะต้องอยู่ในไคลเรทอริที่ Windows อยู่เท่านั้น

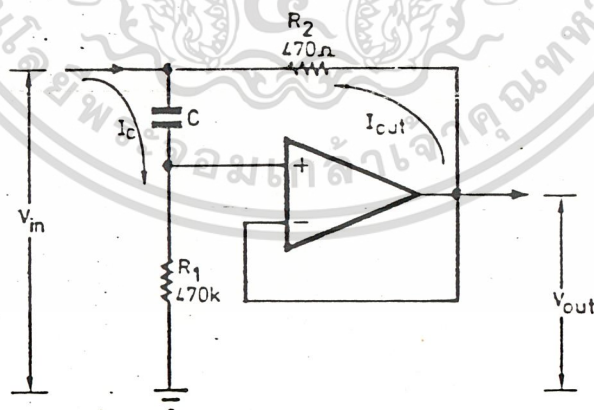
รายการการประกาศค่าของ NI-488.2 Routines และ NI-488 Functions สำหรับภาษา Visual Basic อยู่ในภาคผนวกที่ 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

Digital Graphic Equalizer

วงจรพื้นฐานของ Graphic Equalizer คือ การปรับแต่งอัตราขยายที่ความถี่ใด ๆ (ความถี่ resonance ใด ๆ) โดยมีวงจรพื้นฐานดังรูปที่ 5.1 เป็นวงจรพื้นฐานของภาคขยายเฉพาะความถี่ 1 ช่วงความถี่ สัญญาณที่เข้ามาซึ่งงานอนอินเวอร์ท (+) ของออปแอมป์ จะถูกขยายออกทางเอาต์พุต สัญญาณส่วนหนึ่งจากเอาต์พุตจะถูกป้อนกลับผ่านตัวต้านทาน 10 กิโลโอห์มไปยังขาอินเวอร์ท (-) ซึ่งเป็นการป้อนกลับแบบลบ (Negative Feedback) เมื่อตัวต้านทานปรับค่าได้ 50 กิโลโอห์มถูกปรับให้อยู่ในตำแหน่งตรงกลางวงจรนี้จะมีอัตราขยายเป็น 1 คือ ไม่มีการขยายเลข C และ L ในวงจรก็จะมีผลใด ๆ ทั้งสิ้น เมื่อปรับตัวต้านทานให้อยู่ในตำแหน่ง "เพิ่ม" คือต้องการขยายความถี่นั้นออกมา C และ L ก็คืออยู่ที่ตำแหน่งขาอินเวอร์ท สัญญาณที่ป้อนกลับแบบลบจะถูกค่อลงกราวด์ โดยวงจรจูน LC ซึ่งทำให้เพิ่มอัตราขยายเฉพาะความถี่เรโซแนนซ์ขึ้น ในทางกลับกันเมื่อปรับตัวต้านทานมาทางตำแหน่งลด การป้อนกลับแบบลบจะมีมากที่สุด อัตราขยายจะต่ำสุดที่ความถี่เรโซแนนซ์ ดังนั้นการปรับค่าความต้านทานในรูปที่ 5.1 จึงสามารถเปลี่ยนอัตราขยายของความถี่เรโซแนนซ์

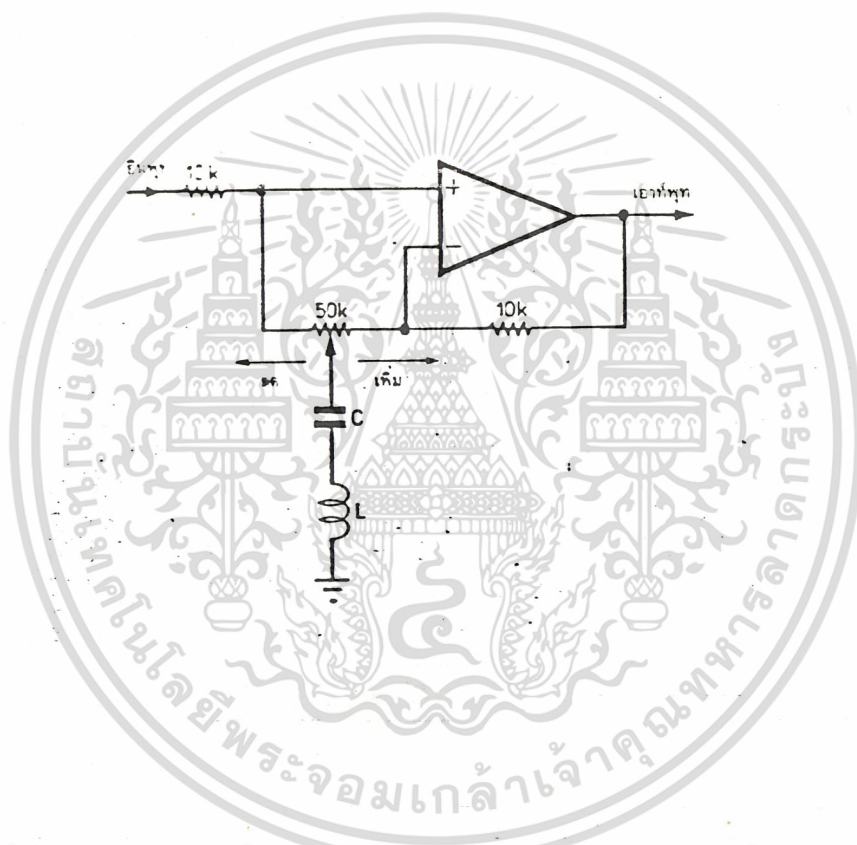


รูปที่ 5.1 วงจรพื้นฐานของภาคขยายเฉพาะความถี่ที่ไว้ค้นกับประจุและตัวเหนี่ยวนำ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี โดยสงวนสิทธิ์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

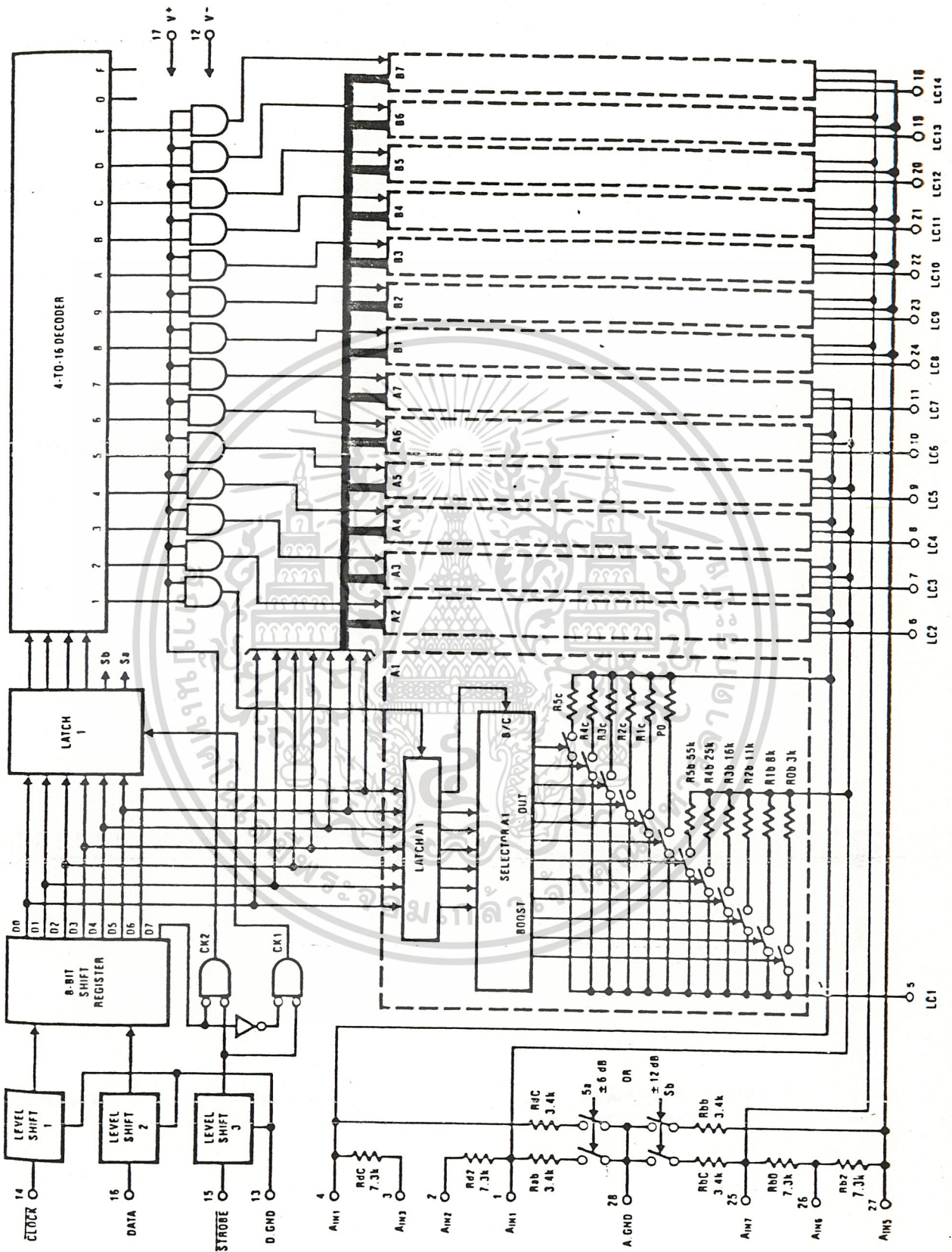
ในสมัยก่อนวงจรกราฟฟิเคิลควอดเรเซอร์จะใช้ค้วเหนียวมาจริง ๆ ตามวงจรในรูปที่ 5.1 แต่เนื่องจากค้วเหนียวมีราคาแพง หันยาก และมีโอกาสรับสัญญาณฮัมเข้ามารบกวนได้ง่าย จึงทำให้ภายหลังไม่เป็นที่นิยมใช้ กราฟฟิเคิลควอดเรเซอร์สมัยใหม่จึงใช้วงจรไอเรเตอร์ (Gyrtator) แทน

วงจรไอเรเตอร์ หมายถึงวงจรออปแอมป์ที่เปลี่ยนค้วเก็บประจุให้เป็นค้วเหนียวมา คือ เปลี่ยนคุณสมบัติของค้วเก็บประจุให้เป็นคุณสมบัติของค้วเหนียวมา



รูปที่ 5.2 วงจรไอเรเตอร์ที่ใช้ออปแอมป์กับค้วเก็บประจุมาทำเป็นค้วเหนียวมา

ในรูปที่ 5.2 จึงเป็นวงจรไอเรเตอร์ เมื่อป้อนสัญญาณ V_{in} เข้าไปยังวงจร กระแส I_c จะไหลผ่าน C และ R_1 แรงดันที่ตกคร่อม R_1 จะถูกขยายออกไปทางเอาต์พุต แรงดันที่ตกคร่อม R_2 เป็นผลต่างของแรงดันอินพุตและเอาต์พุต ดังนั้นจึงมีกระแส I_O ไหล จากเอาต์พุตของออปแอมป์ ผ่าน R_2 ไปยังอินพุตของวงจร ผลรวมของกระแส I_c และ I_O ที่อินพุตของวงจร โดยทิศทางเฟสรวมแล้วจะตามหลัง (lag) แรงดันอินพุตอยู่ ซึ่งเป็นคุณสมบัติของค้วเหนียวมา ดังนั้นวงจรนี้จึงเปรียบเสมือนกับเปลี่ยนคุณสมบัติของค้วเก็บประจุให้กลายเป็นค้วเหนียวมา รูปที่ 5.3 เป็น กราฟแสดงการปรับแต่ง Graphic Equalizer ขนาด 10 ช่องความถี่อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



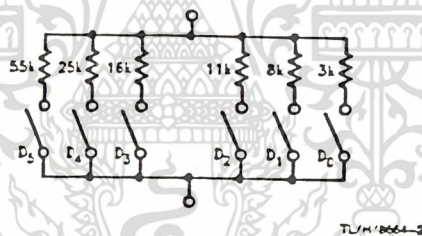
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.4

การทำงานแบบ Digital

ที่กล่าวไปแล้วเป็นพื้นฐานการทำงานแบบ Analog ในการทำงานแบบ Digital จำเป็นจะต้องหาจรรยาบรรณที่สามารถปรับค่าได้ด้วยสัญญาณ Digital เพื่อนำไปควบคุมการปรับอัตราขยายที่ความถี่ต่าง ๆ ซึ่ง IC ที่สามารถทำหน้าที่เช่นนี้ได้ ได้แก่ IC ของบริษัท National Semiconductor เบอร์ LMC 835 (Special Device) Digital Controlled Graphic Equalizer

LMC 835 สามารถทำการควบคุมเครื่องปรับแต่งเสียงได้ 14 bands mono หรือ 7 bands stereo ดังรูปที่ 5.4 โดยภายใน 1 band จะประกอบด้วยชุดความต้านทาน 2 ชุด ที่ทำหน้าที่เหมือนกัน โดยชุดแรกจะทำหน้าที่สำหรับการ Boost และอีกชุดจะทำหน้าที่สำหรับการ Cut โดยชุดความต้านทานจะแสดงไว้ดังรูปที่ 5.5



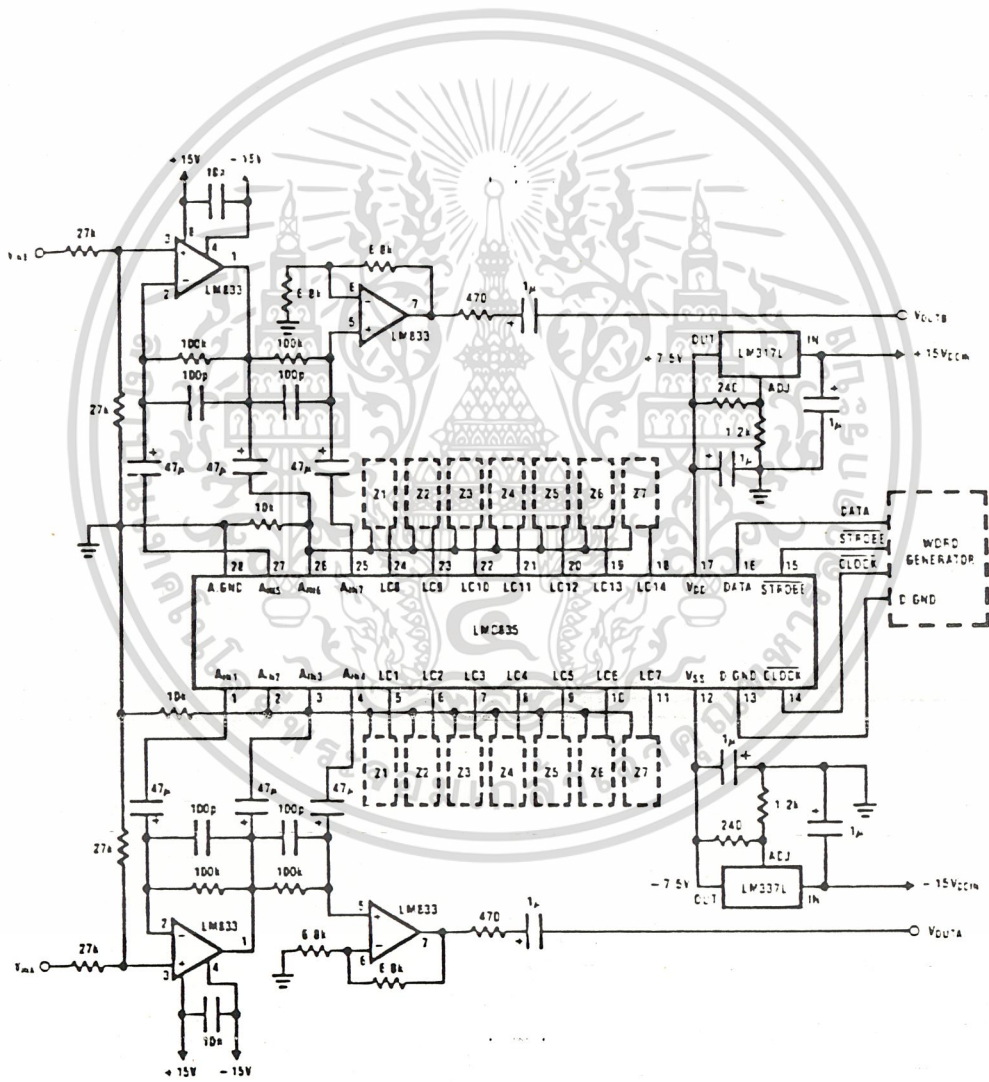
LEVEL	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
FLAT	0	0	0	0	0	0
1 dB	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	1	0	0	1	0
7	1	0	1	0	1	0
8	0	1	0	1	1	0
9	0	0	0	0	0	1
10	1	0	1	0	0	1
11	1	0	1	1	0	1
12	1	0	1	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.5 Digital Controlled Resistance

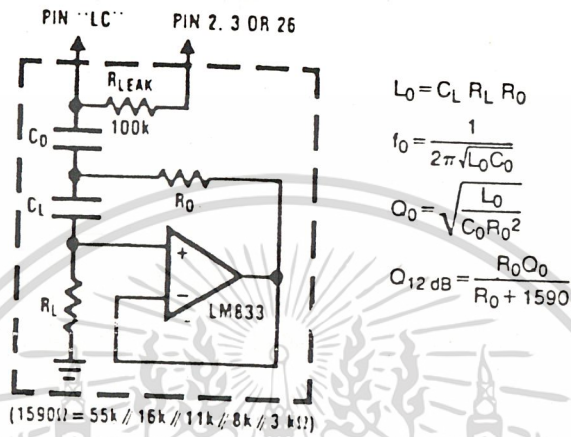
ชุดความค้ำทานนี้สามารถปรับ Level ได้ทั้งหมด 25 ระดับ จากตารางจะเป็นตัวอย่างของการปรับแต่ง 25 ระดับ โดยเป็น Flat 1 ระดับ , Boost (1-12 db) 12 ระดับ , Cut (1-12 db) 12 ระดับ การ Boost และ การ Cut จะเป็นข้อมูลเดียวกัน เพียงแค่ทำการปรับชุดความค้ำทานต่างชุดกัน และ IC ตัวนี้สามารถทำการปรับแต่ง +6 db - -6 db ได้ (25 ระดับ) โดยจะทำการปรับขั้นละ 0.5 db ขึ้นอยู่กับข้อมูลที่ให้กับ IC ตัวนี้

รูปที่ 5.6 เป็นตัวอย่างการต่อ LMC 835 เพื่อให้เป็น Stereo 7 Bands Equalizer โดย Z1 - Z7 จะเป็นวงจรรีโชนแอมป์ ดังรูปที่ 5.7 โดยจะมีสูตรการคำนวณทั้งหมดดังนี้



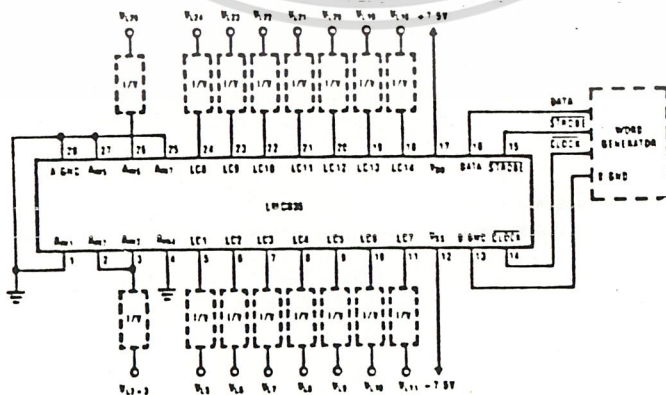
รูปที่ 5.6 Stereo 7 - Band Equalizer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 Tuned Circuit for Stereo 7 - Band Equalizer

จาก Data Sheet จะพบว่า LMC 835 ต้องการสัญญาณควบคุมทั้งหมด 3 สัญญาณ ได้แก่ DATA , STROBE และ CLOCK โดยการส่งข้อมูลแบบ data ให้จะต้องเป็นการส่งแบบอนุกรม (Serial) ดังรูปที่ 5.8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงรูปที่ 5.8 * ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยข้อมูลที่ส่งออกไปความถี่ LMC 835 นั้นเป็นคังรูปที่ 5.9

DATA I (Band Selection)

D7	D6	D5	D4	D3	D2	D1	D0
H	X	L	L	L	L	L	L
H	X	L	L	L	L	L	H
H	X	L	L	L	L	H	L
H	X	L	L	L	L	H	H
H	X	L	L	L	H	L	L
H	X	L	L	L	H	L	H
H	X	L	L	L	H	H	L
H	X	L	L	L	H	H	H
H	X	L	L	H	L	L	L
H	X	L	L	H	L	L	H
H	X	L	L	H	L	H	L
H	X	L	L	H	L	H	H
H	X	L	L	H	H	L	L
H	X	L	L	H	H	L	H
H	X	L	L	H	H	H	L
H	X	L	L	H	H	H	H
H	X	L	H	Valid Binary Input			
H	X	H	L	Valid Binary Input			
H	X	H	H	Valid Binary Input			
↑	↑	↑	↑	← Band Code →			
⊖	⊕	⊖	⊖				

(Ch A: Band 1~7, Ch B: Band 8~14)

- Ch A ± 12 dB Range, Ch B ± 12 dB Range, No Band Selection
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 1
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 2
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 3
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 4
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 5
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 6
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 7
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 8
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 9
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 10
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 11
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 12
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 13
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 14
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, No Band Selection
- Ch A ± 12 dB Range, Ch B ± 6 dB Range, Band 1~14
- Ch A ± 6 dB Range, Ch B ± 12 dB Range, Band 1~14
- Ch A ± 6 dB Range, Ch B ± 6 dB Range, Band 1~14

- ⊖ DATA 1
- ⊕ Don't Care
- ⊖ Ch A ± 6 dB/± 12 dB Range
- ⊖ Ch B ± 6 dB/± 12 dB Range

DATA II (Gain Selection)

D7	D6	D5	D4	D3	D2	D1	D0	
L	X	L	L	L	L	L	L	
L	H	H	L	L	L	L	L	
L	H	L	H	L	L	L	L	
L	H	L	L	H	L	L	L	
L	H	L	L	L	H	L	L	
L	H	L	L	L	L	H	L	
L	H	L	H	L	L	H	L	
L	H	H	L	H	L	H	L	
L	H	L	H	L	H	H	L	
L	H	L	L	L	L	L	H	
L	H	H	L	H	L	L	H	
L	H	H	L	H	H	L	H	
L	H	H	L	H	H	H	H	
L	L	Valid Above Input						
↑	↑	← Gain Code →						
⊖	⊕							

This is the gain if the ± 12 dB range is selected by DATA I. If the ± 6 dB range is selected, then the values shown must be approximately halved. See the characteristics curves for more exact data.

- Flat
- 1 dB Boost
- 2 dB Boost
- 3 dB Boost
- 4 dB Boost
- 5 dB Boost
- 6 dB Boost
- 7 dB Boost
- 8 dB Boost
- 9 dB Boost
- 10 dB Boost
- 11 dB Boost
- 12 dB Boost
- 1 dB ~ 12 dB Cut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ⊖ DATA II
- ⊖ Boost/Cut

TABLE 1 : CENTER FREQUENCY OF BAND

BAND	CENTER FREQUENCY	BAND	CENTER FREQUENCY
1	20 Hz	15	720 Hz
2	26 Hz	16	930 Hz
3	33 Hz	17	1.2 KHz
4	43 Hz	18	1.6 KHz
5	56 Hz	19	2 KHz
6	72 Hz	20	2.6 KHz
7	93 Hz	21	3.3 KHz
8	120 Hz	22	4.3 KHz
9	155 Hz	23	5.6 KHz
10	200 Hz	24	7.2 KHz
11	260 Hz	25	9.3 KHz
12	334 Hz	26	12 KHz
13	430 Hz	27	16 KHz
14	557 Hz	28	20 KHz

ตารางที่ 5.1 Center Frequency of Band

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BAND	FO	CO	CL	RL	RO	BAND	FO	CO	CL	RL	RO
1	20 Hz	0.62u	0.22u	560K	627	15	720 Hz	0.022u	0.0066u	505K	650
2	28 Hz	0.66u	0.18u	525K	578	16	930 Hz	0.018u	0.0047u	565K	614
3	38 Hz	0.47u	0.15u	482K	654	17	1.2 KHz	0.015u	0.0033u	622K	570
4	48 Hz	0.53u	0.1u	573K	723	18	1.6 KHz	0.01u	0.0033u	485K	668
5	56 Hz	0.33u	0.082u	541K	560	19	2 KHz	0.0082u	0.0022u	580K	627
6	72 Hz	0.22u	0.066u	505K	650	20	2.6 KHz	0.0066u	0.0018u	525K	578
7	93 Hz	0.18u	0.047u	585K	614	21	3.3 KHz	0.0047u	0.0015u	492K	654
8	120 Hz	0.15u	0.033u	622K	570	22	4.3 KHz	0.0033u	0.001u	573K	723
9	155 Hz	0.1u	0.033u	485K	668	23	5.6 KHz	0.0033u	0.001u	444K	560
10	200 Hz	0.082u	0.022u	560K	627	24	7.2 KHz	0.0022u	0.001u	343K	650
11	260 Hz	0.066u	0.018u	525K	578	25	9.3 KHz	0.0018u	0.001u	288K	614
12	334 Hz	0.047u	0.015u	492K	654	26	12 KHz	0.0015u	0.001u	205K	570
13	430 Hz	0.033u	0.01u	573K	723	27	16 KHz	0.001u	0.001u	160K	668
14	557 Hz	0.033u	0.0082u	540K	560	28	20 KHz	820p	0.001u	124K	627

ตารางที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

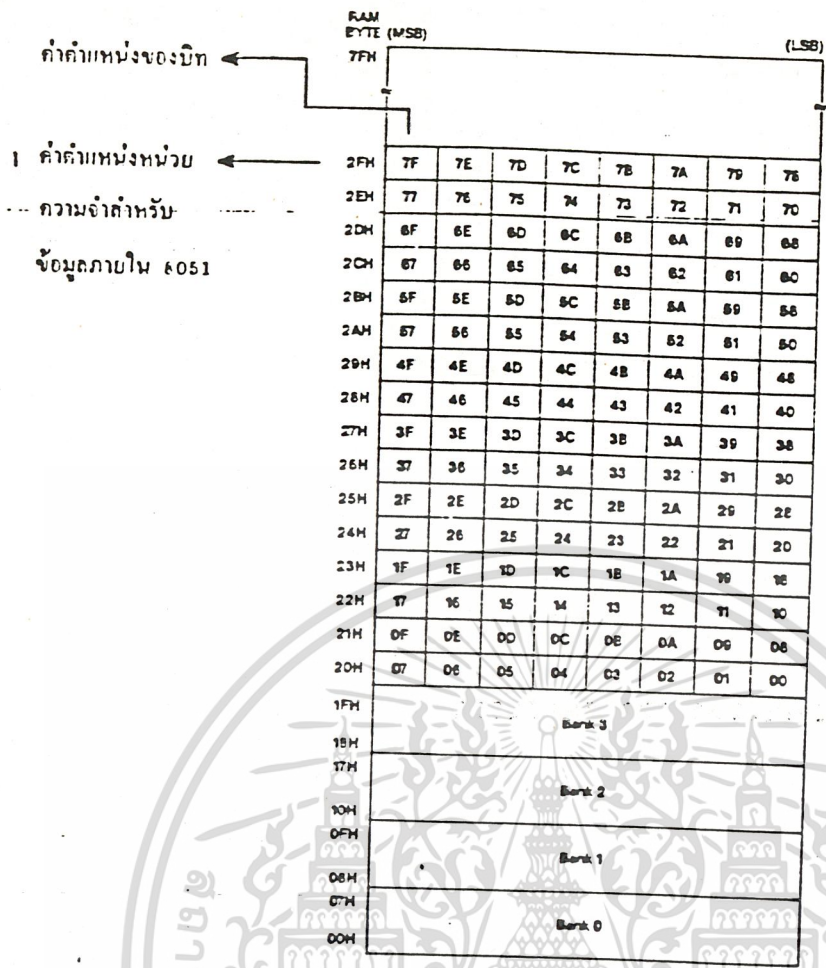
ส่วนควบคุมระบบ (80S1)

Microcontroller เบอร์ 8031 หรือ 8051 ปัจจุบันนี้เป็นที่นิยมในการใช้งานอย่างแพร่หลายมาก เนื่องจากคุณสมบัติและราคาที่เหมาะสม คุณสมบัติพิเศษที่พิจารณาเข้ามาใช้งานในวงจรนี้คือ

1. มี Internal RAM 128 Bytes ค้างแสดงในรูปที่ 6.1 , 6.2 และ 6.3
2. มี I / O แบบ Serial Port (ใช้ในการติดต่อกับ LMC 835)
3. สามารถอ้าง I / O Port แบบ Bit ได้
4. มีจำนวน I / O Port มากพอความต้องการ
5. ใช้งานได้ง่ายและราคาถูก

Direct Byte Address (MSB)	B: Address	Special Function Register Symbol (LSB)
0FBH	WDT T32 SERR ZC P3-PC P2-HZ P1-HZ ALF	IOCON
0F0H	F7 F6 F5 F4 F3 F2 F1 F0	B
0E0H	E7 E6 E5 E4 E3 E2 E1 E0	ACC
0D0H	CY AC FO RS1 RS0 DN F1 P	PSW
0C0H	Not B: Accessible	
0C0H	Not B: Accessible	
0CBH	Not B: Accessible	
0CAH	Not B: Accessible	
0C8H	TF2 EXF2 RCLK TCLK EXG2 TR2 C12 C123	T2CON
	CF CE CD CC CE CA C9 CE	
0B8H	PC1 PT2 PS PT1 PX1 PTD PXC	IP
	BF - BC BD BE BA B9 BE	
0B0H	B7 B6 B5 B4 B3 B2 B1 B0	P3
0A8H	EA ET2 ES ET1 EX1 ETO EX0	E
	AF - AD AC AE AA A9 AE	
0A0H	A7 A6 A5 A4 A3 A2 A1 A0	P2
99H	Not B: Accessible	
	SM0 SM1 SM2 REN TBE RBE TI R	SCON
98H	8F 8E 8D 8C 8B 8A 89 8E	
90H	87 8E 8E 84 85 82 81 8C	P1
8CH	Not B: Accessible	
80H	Not B: Accessible	
8BH	Not B: Accessible	
8AH	Not B: Accessible	
89H	Not B: Accessible	
	TF1 TR1 TFC TR0 IE1 IT1 IE0 IT0	TCON
88H	8F 8E 8C 8C BE BA BE BE	
87H	Not B: Accessible	
83H	Not B: Accessible	
82H	Not B: Accessible	
81H	Not B: Accessible	
80H	B7 BE B5 B4 B3 B2 B1 B0	DPH DPL SP P0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 6.1 ค่าตำแหน่งของแต่ละบิตไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิต

Table 1

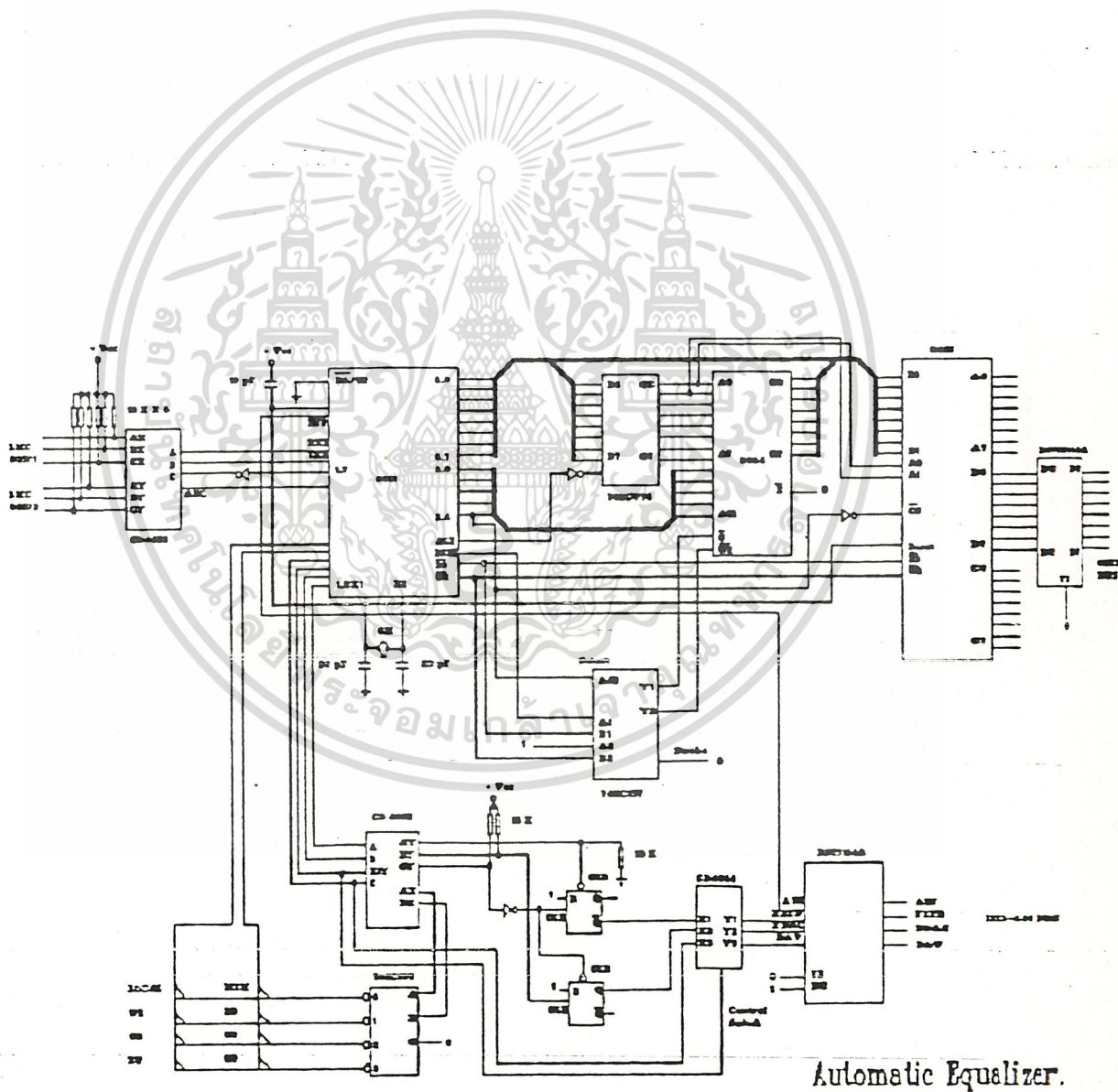
Symbol	Name	Address
*ACC	Accumulator	0E0H
*PC	Program Counter	0E1H
*PSW	Program Status Word	0E2H
SP	Stack Pointer	0E3H
DPTR	Data Pointer 2 Bytes	0E4H
DPH	Low Byte	0E5H
DPL	High Byte	0E6H
P0	Port 0	80H
P1	Port 1	81H
P2	Port 2	82H
P3	Port 3	83H
IP	Interrupt Priority Control	0A0H
IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter Control	8BH
-T2CON	Timer/Counter 2 Control	8BH
TH0	Timer/Counter 0 High Byte	0C8H
TL0	Timer/Counter 0 Low Byte	0C9H
TH1	Timer/Counter 1 High Byte	0EAH
TL1	Timer/Counter 1 Low Byte	0EBH
-TH2	Timer/Counter 2 High Byte	0CCH
-TL2	Timer/Counter 2 Low Byte	0CDH
-RCAP2H	T/C 2 Capture Reg. High Byte	0CEH
-RCAP2L	T/C 2 Capture Reg. Low Byte	0CFH
*SCON	Serial Control	0CAH
SBUF	Serial Data Buffer	9BH
PCON	Power Control	87H
*TCON (1)	I/O Control	F8H

- 80C52 and 83C154 only *b* addressable
 (1) 83C154 only

รูปที่ 6.3 Special Function Register (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรใช้งานในระบบควบคุม จะแสดงไว้ในรูปที่ 6.4 [Main Circuit] โดยมีกำหนด Port ต่าง ๆ ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแบบไปแก้ไขหรือดัดแปลงอย่างอื่นถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 6.4 วงจรรวม

Port0 ใช้ในการอ้าง Address ร่วมกับ Port 2 กับหน่วยความจำภายนอกได้แก่

1. EEPROM 2864 เป็นหน่วยความจำที่มีขนาด 8 Kbytes (EEPROM เป็นหน่วยความจำประเภทหนึ่ง แต่สามารถทำการลบหรือเขียนได้ด้วยประจุไฟฟ้า โดยมีอายุการใช้งานประมาณ 10,000 ครั้ง / Byte) ในการใช้งาน EEPROM นี้จะแบ่งออกเป็น 2 ส่วนคือ

- 4 Kbytes แรก ใช้ในการเก็บ Program คำสั่งของ 8031 (แสดงไว้ในส่วน Program ภาษา Assembly
- 4 Kbytes หลัง ใช้เป็น Memory ในการเก็บข้อมูลการปรับแต่งเครื่องปรับแต่งเสียงที่ระดับต่าง ๆ ของแต่ละ band ซึ่งสามารถเก็บได้สูงสุดถึง 99 memory

2. 8255 เป็น IC ที่ใช้ในการขยาย Port เพื่อให้ได้ Port เพิ่มขึ้น 3 Ports ได้แก่ Port A , Port B , Port C โดยมีการอ้าง Port ที่ทำการ ซึ่งหน้าที่การทำงานของ Port ต่าง ๆ จะอธิบายอีกครั้งหนึ่ง

การทำงานของ Port 0 คือ ตอนแรกจะทำการตั้ง Address ร่วมกับ Port 2 ค่า Address ที่อ้างไปจะถูก latch ไว้ที่ IC เบอร์ 74HC374 ซึ่งเป็น 8 bit latch ด้วยสัญญาณจากขา ALE หลังจากนั้น Port 0 จะทำการรับ Data หรือ Program จากหน่วยความจำภายนอกเข้ามา การอ้าง Address ของ EEPROM 4 Kbytes แรก ซึ่งเป็นส่วน โปรแกรม จะถูกอ่านเข้ามา ด้วยสัญญาณขา PSEN ส่วน 4 Kbytes หลัง ซึ่งเป็นส่วน Memory หรือ Data จะถูกอ่านและเขียน ด้วยสัญญาณจากขา RD และ WR ของ 8031

Port1 จะทำหน้าที่ทั้งหมด 4 หน้าที่ ซึ่งมีการทำงานแบบ Bit รวมอยู่ด้วยคือ

- Bit 7 ใช้สำหรับส่งสัญญาณ Strobe ให้กับ LMC 835 ร่วมกับ Serial Port (ใช้ในการส่ง data และ clock)
- Bit 6 ใช้ในการเลือก LMC 835 โดยถ้าเป็น Low จะทำการเลือก LMC 835 ตัวที่ 1 (Band 1 - 14) และถ้าเป็น High จะทำการเลือก LMC 835 ตัวที่ 2 (Band 15 -28) บิตนี้จะใช้ในการแสดงผลการเลือก Band (Decoder Display) ร่วมกับ Port C ของ 8255 ด้วย
- Bit 2 ใช้ในการกำหนดค่าให้บิตที่เหลือทำหน้าที่ในการ Scan Key เพื่อรับ Input การกด key ต่าง ๆ หรือใช้ในการเชื่อมต่อกับระบบแฮนด์เชค (Handshake) ของ IEEE - 488 Bus โดยจะมีการกล่าวถึงในรายละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 2 ใช้ร่วมกับ Port 0 ในการอ้างแอดเดรสกับหน่วยความจำภายนอก โดยจะใช้บิตที่ 4 ในการกำหนดว่าหน่วยความจำของ 2864 ที่ทำการติดต่อก็คือเป็นการติดต่อกับโปรแกรม หรือ บันทึความจำ ดังที่ได้กล่าวมาแล้ว หรืออาจแสดงได้ดังนี้

0000 H - 0FFF H	เป็นแอดเดรสของโปรแกรม (มีการติดต่อกับแบบ External ROM)
1000 H - 1FFF H	เป็นแอดเดรสของบันทึกความจำ (มีการติดต่อกับแบบ External RAM)
0000 H - 0003 H	เป็นแอดเดรสของ 8255 (มีการติดต่อกับแบบ External RAM)

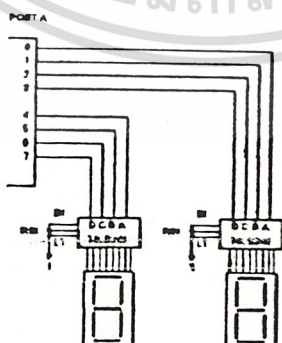
Port A (8255)

พอร์ตนีจะทำหน้าที่ติดต่อกับ Seven Segment 2 หลัก เพื่อเป็นการแสดงผลที่เป็นตัวเลขได้แก่

1. แสดงค่าแอดเดรสของ Device ในระบบ GPIB ซึ่งสามารถตั้งค่าแอดเดรสนี้ได้ขณะทำการเปิดเครื่อง

2. แสดงค่าระดับ Level ของ Band ที่ทำการเลือก Seven Segment นี้จะทำการแสดงค่าเพียง 2 หลักเท่านั้น คือ 0 - 12 db ส่วน Sign bit จะถูกแสดงใน Port C

3. แสดงค่าบันทึกความจำที่ทำการบันทึกเอาไว้ (สูงสุด 99 บันทึกความจำ) การที่จะเลือกให้ Seven Segment แสดงค่าใดนั้นขึ้นอยู่กับว่าขณะนั้น เครื่องกำลังทำงานอยู่ใน mode อะไร โดยที่ Port A จะมีการวงจรรูปที่ 6.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับที่ 6.5 IC Segment Drive (Port A) อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port B (8255)

ใช้ในการรับข้อมูลสำหรับการติดต่อกับ GPIB (IEEE - 488) โดยบิตที่ 0 ถึง 6 จะทำการรับข้อมูลจากบิต บิตที่ 0 ถึง 6 เหมือนกัน ส่วนบิตที่ 7 จะทำการรับสัญญาณ ATN (เป็นสัญญาณที่รับออกมาข้อมูลที่อยู่ในบิตเป็น command หรือ data) พอร์ตนี้อาจจะต้องการต่อกับ IC Buffer สำหรับ GPIB Bus เบอร์ DS 75160 (คอแบบ Open Collector)

Port C (8255)

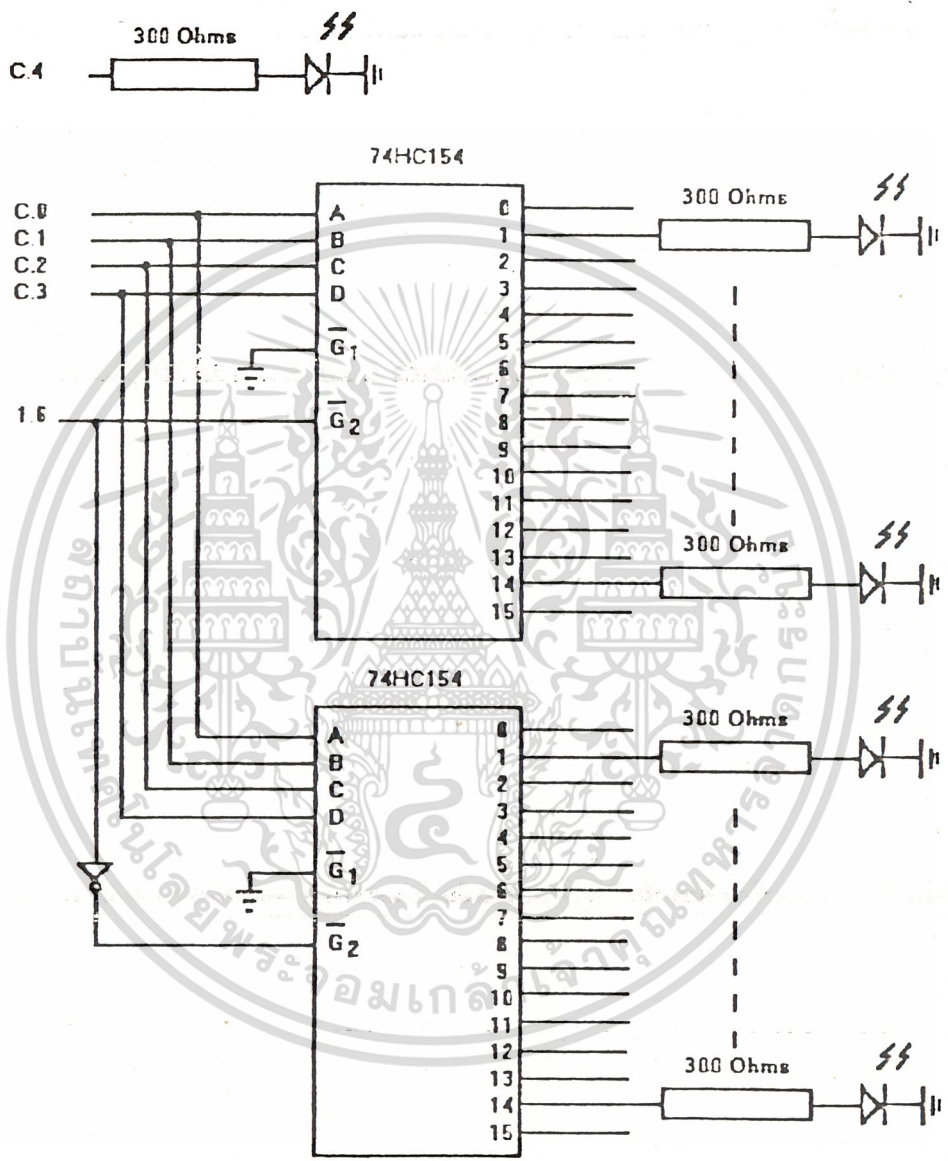
แบ่งการทำงานออกเป็น 2 ส่วนได้แก่

1. บิต 0 ถึง บิต 3 เป็นส่วนแสดงहर่วมกับ บิต 6 ของ Port 1 ว่าทำการเลือก Band ไหน (0 - 14) ๑๑ง IC LMC 835 คือเป็นวงจร Decoder แล้วทำการแสดงผลด้วย LED Display โดยที่ บิตที่ 6 ของ Port 1 จะทำการเลือกว่าเป็น LMC 835 ตัวที่ 1 หรือ ตัวที่ 2 โดยจะสามารถแสดงเป็นตารางได้ดังตารางที่ 6.1 และมีวงจรดังรูปที่ 6.6

PORT					HEX	BAND	PORT					HLX	BAND
1.6	C.3	C.2	C.1	C.0	CODE	DISPLAY	1.6	C.3	C.2	C.1	C.0	CODE	DISPLAY
0	0	0	0	0	0	NO USE	1	0	0	0	0	0	NO USE
0	0	0	0	1	1	1	1	0	0	0	1	1	15
0	0	0	1	0	2	2	1	0	0	1	0	2	16
0	0	0	1	1	3	3	1	0	0	1	1	3	17
0	0	1	0	0	4	4	1	0	1	0	0	4	18
0	0	1	0	1	5	5	1	0	1	0	1	5	19
0	0	1	1	0	6	6	1	0	1	1	0	6	20
0	0	1	1	1	7	7	1	0	1	1	1	7	21
0	1	0	0	0	8	8	1	1	0	0	0	8	22
0	1	0	0	1	9	9	1	1	0	0	1	9	23
0	1	0	1	0	A	10	1	1	0	1	0	A	24
0	1	0	1	1	B	11	1	1	0	1	1	B	25
0	1	1	0	0	C	12	1	1	1	0	0	C	26
0	1	1	0	1	D	13	1	1	1	0	1	D	27
0	1	1	1	0	E	14	1	1	1	1	0	E	28
0	1	1	1	1	F	NO USE	1	1	1	1	1	F	NO USE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1



รูปที่ 6.6 วงจร Decoder (Port C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

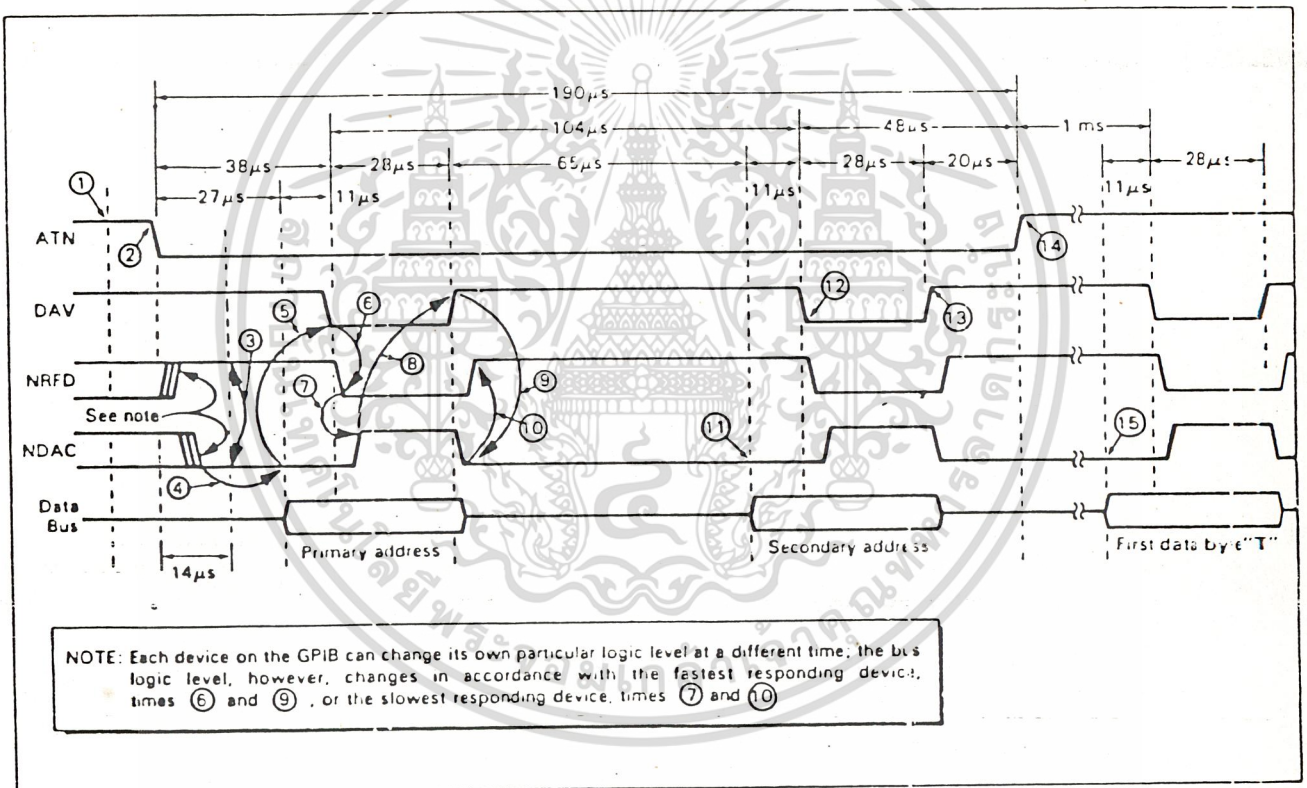
การ Interface กับ GPIB

GPIB จะต้องมีการ Handshake ในการที่จะส่งและรับข้อมูลครั้งที่ได้กล่าวมาแล้ว รูปที่ 7.1 เป็นรูปแสดงระบบการ Handshake และค่าคาบเวลาต่าง ๆ ในการตรวจสอบ ดังนั้นวงจรหรือโปรแกรม จะต้องทำงานให้ได้ความเร็วสูงกว่าค่าคาบเวลาในการตรวจสอบ เพราะถ้าหากการทำงานเลยค่าคาบเวลาที่กำหนดให้แล้ว GPIB จะถือว่าเกิด Error ขึ้นในระบบบัส

ปกติการทำงานของเครื่องจะอยู่ใน mode local หรือ เป็นการปรับแต่งด้วยมือ Port 1 จะทำหน้าที่ในการ scan key เท่านั้น เมื่อมีการติดต่อกับ GPIB สัญญาณ ATN จะทำให้เกิดการ Interrupt กับ 8031 ดังนั้น 8031 จะมีการเรียกโปรแกรม Interrupt ให้ทำงาน เมื่อเข้าสู่โปรแกรม Interrupt จะมีการตั้งให้ Port 1 ทำหน้าที่ในการ Handshake แทนการ scan key และใช้ Port B ของ 8255 ในการรับข้อมูลเข้ามา จากนั้น 8031 จะทำการ Handshake กับ GPIB เพื่อต้องการทราบว่า GPIB ต้องการติดต่อกับ Device ตัวใด ถ้าหากค่า Address ไม่ตรงกับของเครื่องปรับแต่งเสียง 8031 ก็จะทำการออกจากโปรแกรม Interrupt เพื่อมาทำงานใน mode local ต่อไป แต่ถ้าหาก GPIB ต้องการติดต่อกับเครื่องปรับแต่งเสียง เครื่องก็จะเข้าสู่ mode remote เพื่อติดต่อกับ GPIB ต่อไป การติดต่อกับ GPIB จะมีการกำหนด Data ตามตารางที่ 7.1 [Data ASCII] การออกจากโปรแกรม Interrupt สามารถกระทำได้ 2 วิธี คือ

1. ใช้คำสั่ง Goto Local
2. กดปุ่ม Local ที่ตัวเครื่อง

ส่วนของ Hardware สำหรับการ Interface มีอยู่ 2 ส่วนด้วยกันคือ Port 1 และ Port B ของ 8255 ดังแสดงในรูปวงจรรวม (รูปที่ 6.4)



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 7.1 Handshake Timing Diagram ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 3 : ASCII - IEEE 488 BUS MESSAGES (BAND AND LEVEL) HEX CODES

ASCII	HEX CODE	MESSAGE	ASCII	HEX CODE	MESSAGE	ASCII	HEX CODE	MESSAGE	ASCII	HEX CODE	MESSAGE
C	40	NO USE	P	50	NO USE	r	60	11dB	p	70	NO USE
A	41	BAND 1	Q	51	BAND 15	q	61	1 dB	l	71	-1 dB
B	42	BAND 2	R	52	BAND 16	t	62	2 dB	r	72	-2 dB
C	43	BAND 3	S	53	BAND 17	c	63	3 dB	s	73	-3 dB
D	44	BAND 4	T	54	BAND 18	d	64	4 dB	t	74	-4 dB
E	45	BAND 5	U	55	BAND 19	e	65	5 dB	u	75	-5 dB
F	46	BAND 6	V	56	BAND 20	f	66	6 dB	y	76	-6 dB
G	47	BAND 7	W	57	BAND 21	g	67	7 dB	w	77	-7 dB
H	48	BAND 8	X	58	BAND 22	h	68	8 dB	x	78	-8 dB
I	49	BAND 9	Y	59	BAND 23	i	69	9 dB	y	79	-9 dB
J	4A	BAND 10	Z	5A	BAND 24	j	6A	10 dB	z	7A	-10 dB
K	4B	BAND 11	[5B	BAND 25	k	6B	11 dB	{	7B	-11 dB
L	4C	BAND 12	\	5C	BAND 26	l	6C	12 dB		7C	-12 dB
M	4D	BAND 13]	5D	BAND 27	m	6D	NO USE	}	7D	NO USE
N	4E	BAND 14	^	5E	BAND 28	n	6E	NO USE	~	7E	NO USE
O	4F	NO USE	_	5F	NO USE	o	6F	NO USE	DEL	7F	NO USE

ตารางที่ 7.1 ASCII - IEEE 488 Bus Messages (Band and Level) Hex Codes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

ผลการทดลอง , สรุปผลการทดลอง และ วิจารณ์

ผลการทดลอง

จากการทดลองได้ทำการทดลองใช้โปรแกรม Automatic Equalizer บน Windows ได้ผลดังต่อไปนี้

1. โปรแกรม Automatic Equalizer สามารถสั่งงานผ่านการควบคุม IEEE - 488 (GPIB) ได้เป็นอย่างดี โดยสามารถส่งอุปกรณ์ Function Generator และ Oscilloscope ได้
2. โปรแกรม Automatic Equalizer คำนวณหาค่าการตอบสนองความถี่ของสภาพแวดล้อมต่าง ๆ ได้ (Frequency Response) เมื่อทำการ Scanning
3. การแสดงผลค่าการตอบสนองความถี่แสดงได้ทั้งในรูปแบบตัวเลขธรรมดา และ ในรูปแบบของกราฟทั่วไป
4. โปรแกรม Automatic Equalizer สามารถหาค่าการปรับแต่งที่เหมาะสมแก่สภาพแวดล้อมเหล่านั้นได้ และแสดงผลบนหน้าจอ Setting
5. สามารถเก็บบันทึกสภาพการปรับแต่งลงในหน่วยความจำได้ เพื่อให้สามารถนำข้อมูลที่เคยหาค่าการปรับแต่งแล้ว มาใช้ซ้ำได้อีก
6. สามารถพิมพ์ข้อมูลผลของการใช้โปรแกรมนี้ผ่านเครื่องพิมพ์ได้
7. โปรแกรม Automatic Equalizer สามารถสั่งการ เครื่อง Automatic Equalizer ได้เป็นอย่างดี โดยสามารถปรับแต่งได้ตามสภาพการปรับแต่งที่เหมาะสม
8. เครื่อง Automatic Equalizer สามารถปฏิบัติงานได้เป็นอย่างดี โดยสามารถปรับแต่งเสียงได้ถูกต้อง
9. เครื่อง Automatic Equalizer สามารถทำงานพิเศษต่าง ๆ ได้ อันได้แก่ การเก็บข้อมูลไว้ในหน่วยความจำภายในเครื่องเพื่อนำกลับมาเรียกใช้ใหม่ในลักษณะของรูปแบบที่ใช้เป็นประจำ และ การแสดงผลต่าง ๆ ได้ถูกต้อง

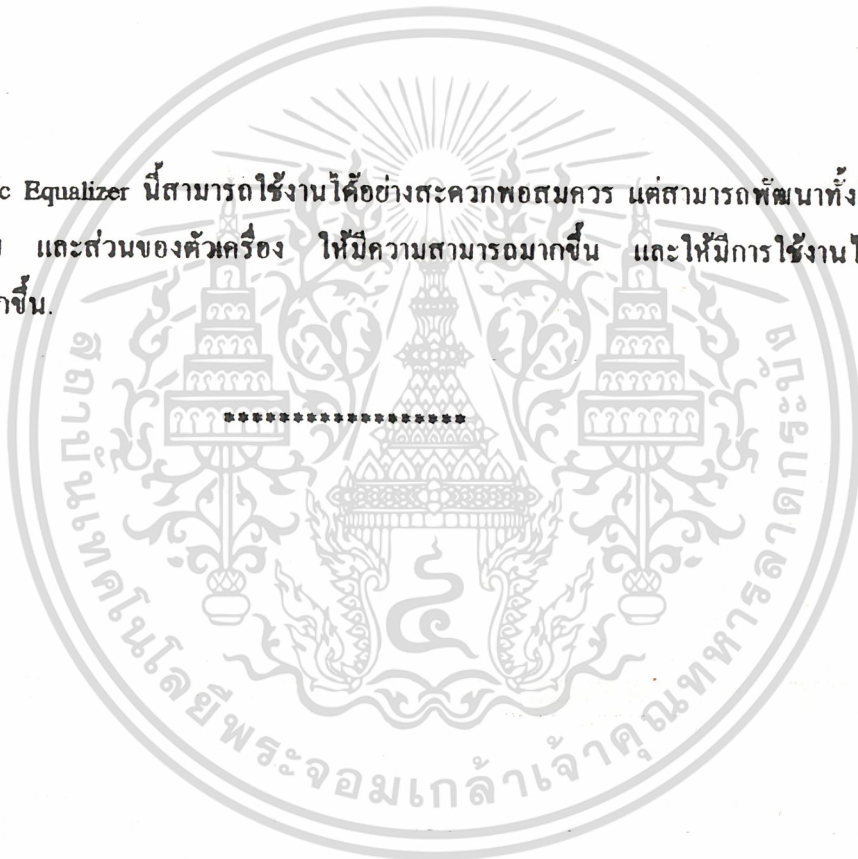
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

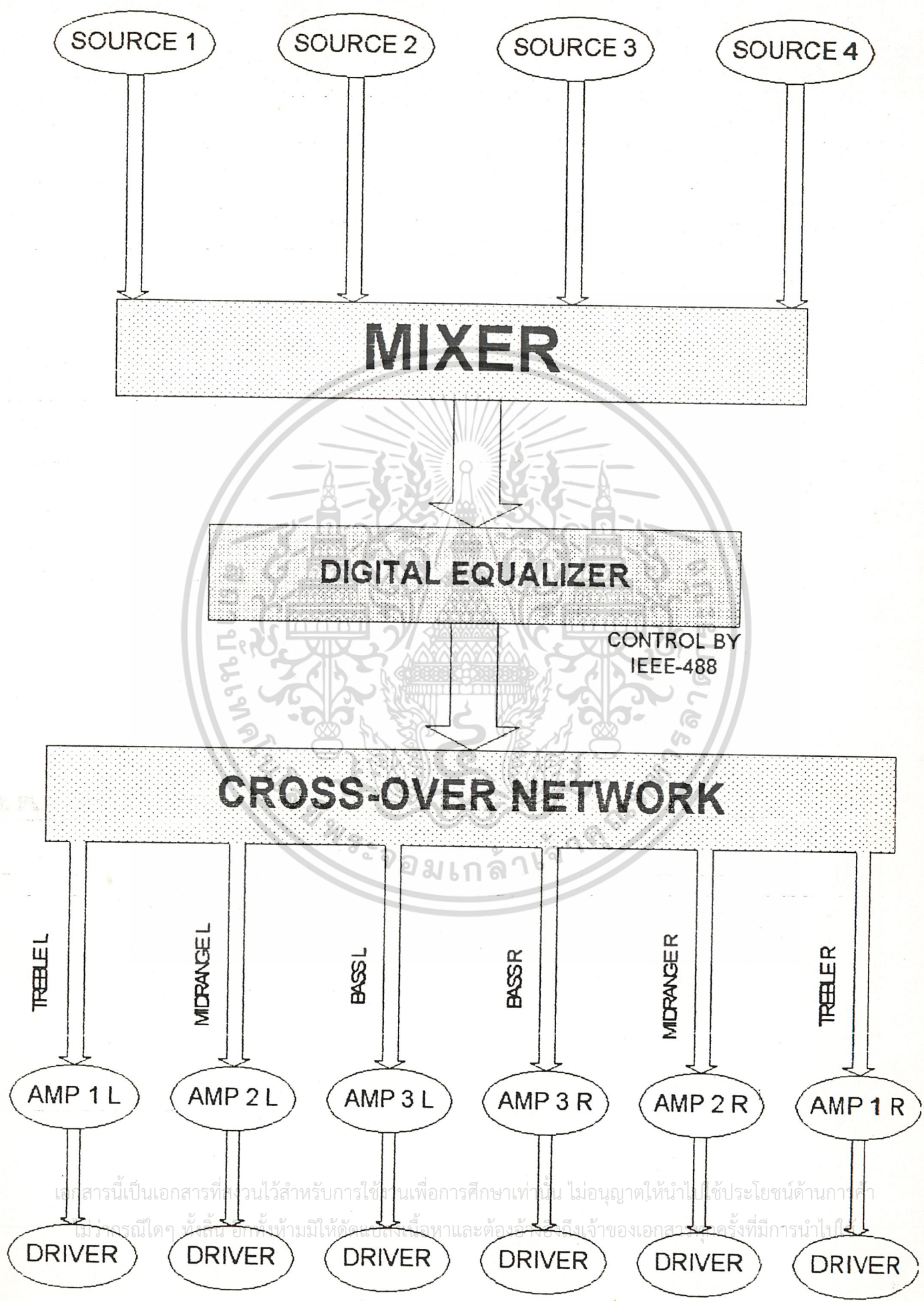
Automatic Equalizer สามารถนำไปใช้ในการปรับแต่งเสียงได้อย่างถูกต้อง และสะดวกแก่การใช้งานเป็นอย่างยิ่ง ทั้งในส่วนของโปรแกรม (Software) และในส่วนของตัวเครื่อง (Hardware).

วิจารณ์

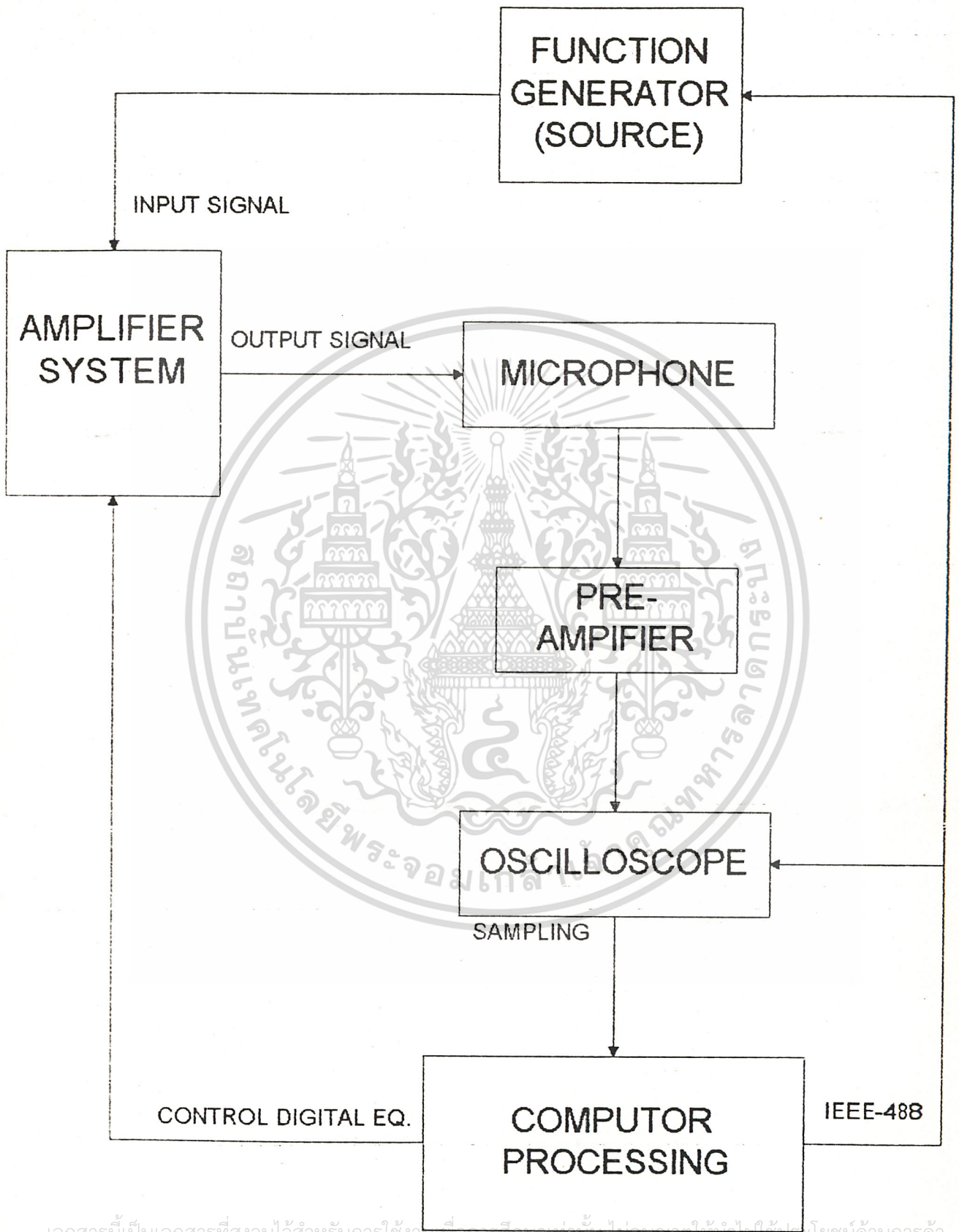
Automatic Equalizer นี้สามารถใช้งานได้สะดวกพอสมควร แต่สามารถพัฒนาทั้งในส่วนของโปรแกรม และส่วนของตัวเครื่อง ให้มีความสามารถมากขึ้น และให้มีการใช้งานได้สะดวกง่ายดายมากขึ้น.



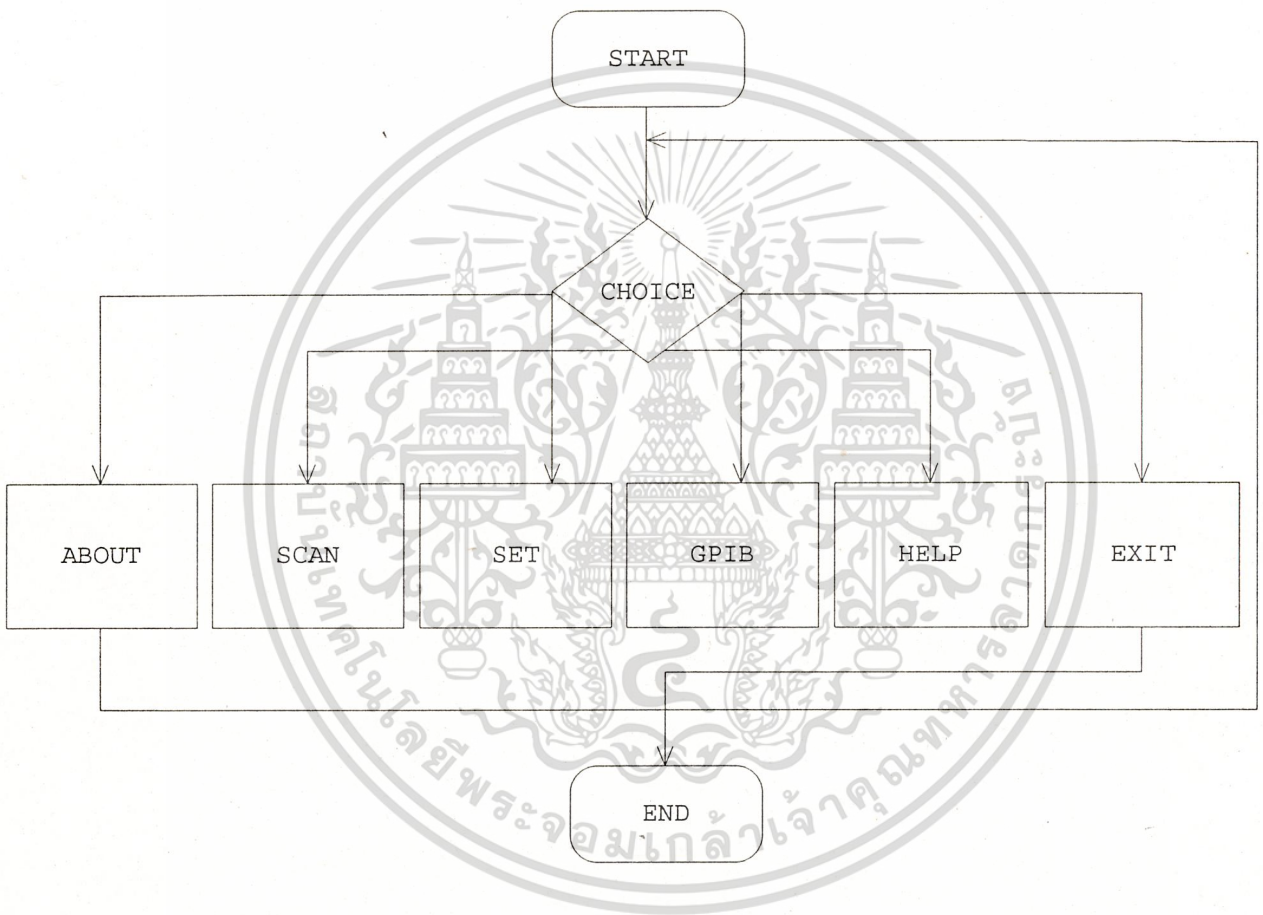
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



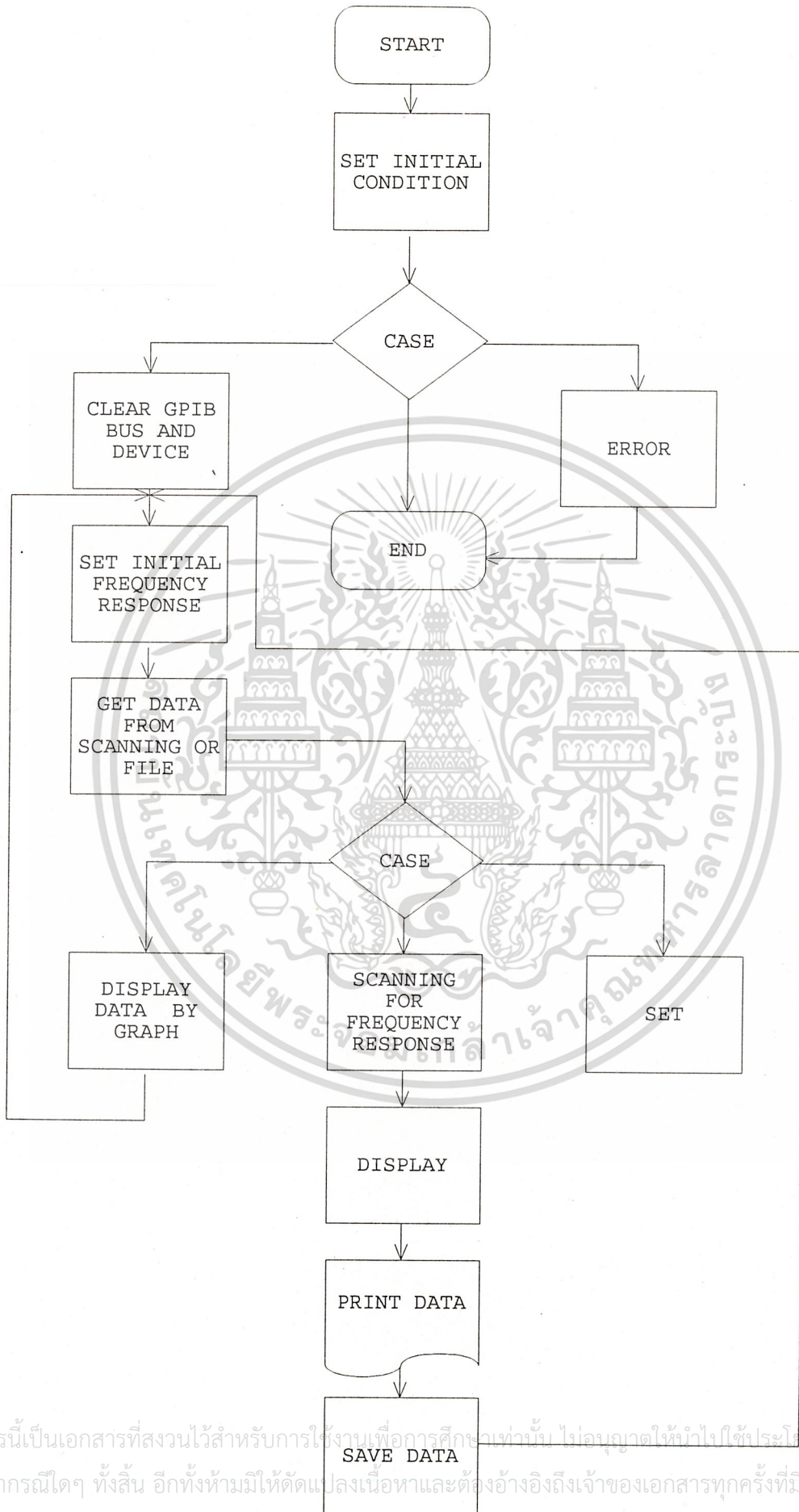
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเชิงเนื้อหาและต้องแจ้งชื่อเจ้าของเอกสารทุกครั้งที่มีการนำไป



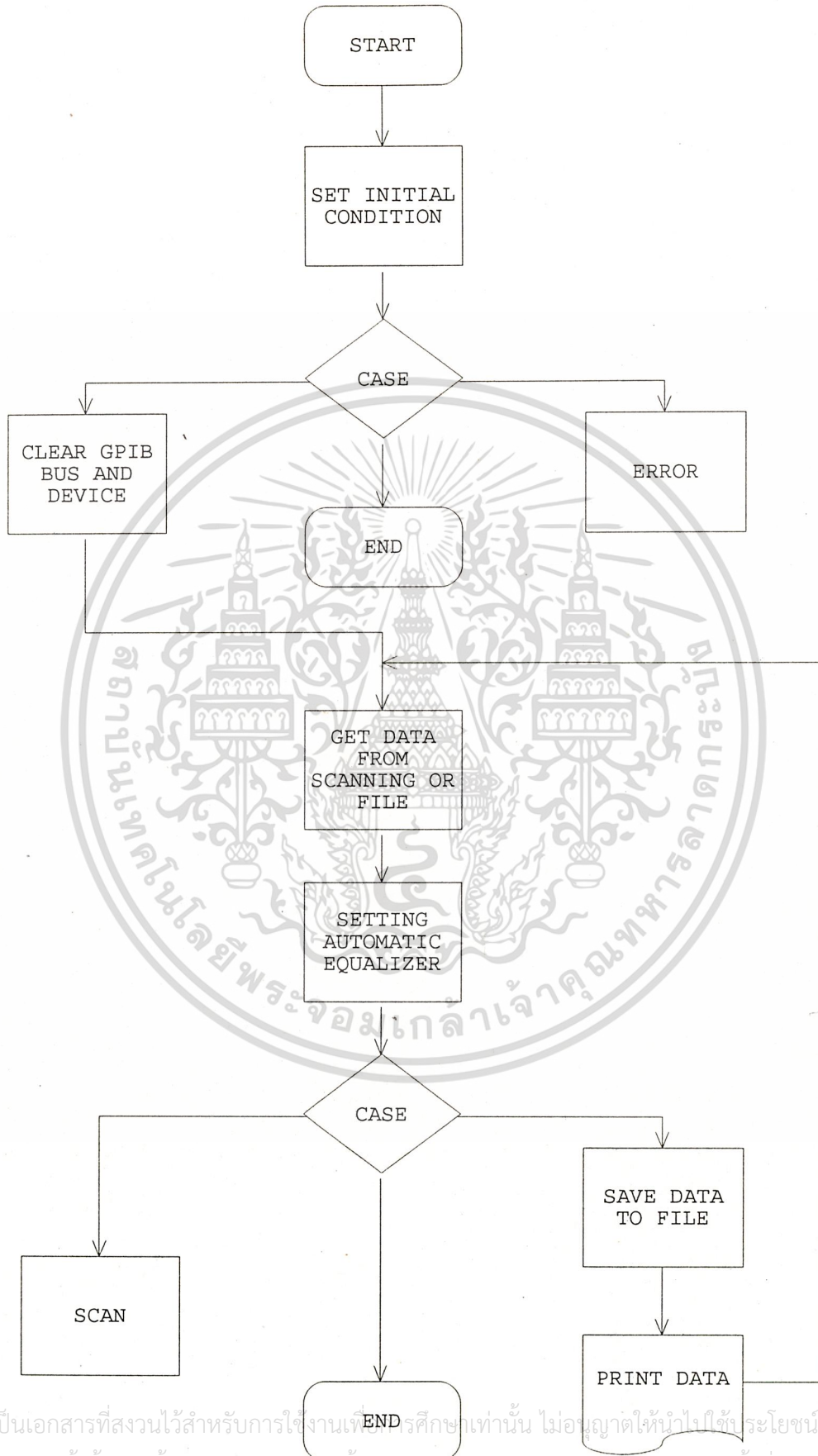
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



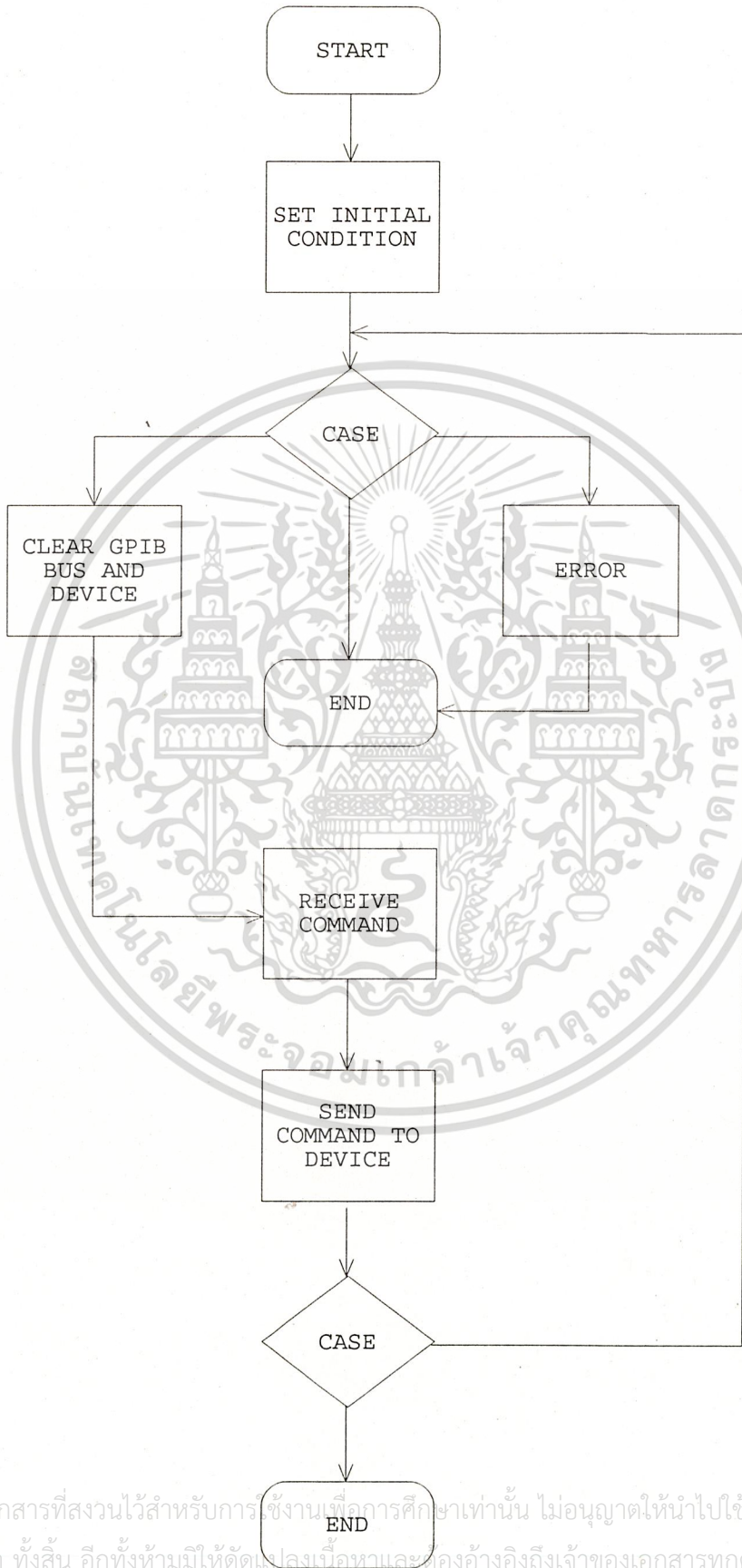
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ระยะเวลาในการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรนี้สำเร็จได้ด้วยดี เนื่องด้วยได้รับความช่วยเหลือจากผู้มีพระคุณต่าง ๆ คณะผู้จัดทำจึงขอแสดงความขอบคุณทุกท่าน อันได้แก่ บิคารมราคาที่ให้กำลังใจ , รศ.क्रमนัต ถังวรศิลป์ อาจารย์ที่ปรึกษา , อาจารย์ภาควิชาอิเล็กทรอนิกส์ที่ให้คำปรึกษา , รุ่นที่ทุกท่านที่ให้คำแนะนำ , เพื่อนซึ่งให้ความสนับสนุน และ รุ่นน้องที่ให้ความช่วยเหลือ ปริญญาบัตรนี้ได้รับความช่วยเหลือจากท่านเหล่านี้มาด้วยดี.




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- Digital Graphic Equalizer , วีรวิญญ์ ชั้นประเสริฐ , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- PET And The IEEE 488 Bus (GPIB) , Eugene Fisher , McGraw - Hill.
- การควบคุมโดยใช้ IEEE 488 (GPIB) , ชัยยุทธ สุพัฒน์โสภณ , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- การควบคุมโดยใช้ IEEE 488 (GPIB) , จรุง มิ่งขวัญ , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- Advanced C , Peter Hipson , SAMS Publishing.
- Visual BASIC How To , Zane Thomas.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวกที่ 1
รายการการประกาศ NI-488.2 Routines และ NI-488.2 Functions
ในการเขียนโปรแกรมควบคุม IEEE - 488 (GPIB)
โดยใช้ภาษา Microsoft Visual BASIC 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note: All of the routines listed in Table A-1 are of type `void _far _pascal`.

Table A-1. Direct Entry NI-488.2 Style Routines

Routine (ordinal entry value)	Syntax
AllSpoll (100)	DLLAllSpoll (short board, short addresslist [], short resultlist [], short *ibsta, short *iberr, unsigned long *ibcntl)
DevClear (101)	DLLDevClear (short board, short address, short *ibsta, short *iberr, unsigned long *ibcntl)
DevClearList (102)	DLLDevClearList (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
EnableLocal (103)	DLLEnableLocal (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
EnableRemote (104)	DLLEnableRemote (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
FindLstn (105)	DLLFindLstn (short board, short addresslist [], short resultlist [], short limit, short *ibsta, short *iberr, unsigned long *ibcntl)
FindRQS (106)	DLLFindRQS (short board, short addresslist [], short *result, short *ibsta, short *iberr, unsigned long *ibcntl)
GenerateREQF (53)	DLLGenerateREQF (short board, short addr, short *ibsta, short *iberr, unsigned long *ibcntl)
GenerateREQT (52)	DLLGenerateREQT (short board, short addr, short *ibsta, short *iberr, unsigned long *ibcntl)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุก (continues) ใช้

Table A-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	Syntax
GotoMultAddr (129)	DLLGotoMultAddr (short board, unsigned short type, unsigned short (_far_loads *addrfunc) (), unsigned short (_far_loads *spollfunc) (), short *ibsta, short *iberr, unsigned long *ibcntl)
PassControl (107)	DLLPassControl (short board, short address, short *ibsta, short *iberr, unsigned long *ibcntl)
PPoll (108)	DLLPPoll (short board, short *result, short *ibsta, short *iberr, unsigned long *ibcntl)
PPollConfig (109)	DLLPPollConfig (short board, short address, short dataline, short sense, short *ibsta, short *iberr, unsigned long *ibcntl)
PPollUnconfig (110)	DLLPPollUnconfig (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
RcvRespMsg (111)	DLLRcvRespMsg (short board, char data [], long count, short termination, short *ibsta, short *iberr, unsigned long *ibcntl)
ReadStatusByte (112)	DLLReadStatusByte (short board, short address, short *result, short *ibsta, short *iberr, unsigned long *ibcntl)
Receive (113)	DLLReceive (short board, short address, char data [], unsigned long count, short termination, short *ibsta, short *iberr, unsigned long *ibcntl)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ (continues) ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	Syntax
ReceiveSetup (114)	DLLReceiveSetup (short board, short address, short *ibsta, short *iberr, unsigned long *ibcntl)
ResetSys (115)	DLLResetSys (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
Send (116)	DLLSend (short board, short address, char data [], long count, short eotmode, short *ibsta, short *iberr, unsigned long *ibcntl)
SendCmds (117)	DLLSendCmds (short board, char commands [], unsigned long count, short eotmode, short *ibsta, short *iberr, unsigned long *ibcntl)
SendDataBytes (118)	DLLSendDataBytes (short board, char data [], long count, short eotmode, short *ibsta, short *iberr, unsigned long *ibcntl)
SendIFC (119)	DLLSendIFC (short board, short *ibsta, short *iberr, unsigned long *ibcntl)
SendList (120)	DLLSendList (short board, short addresslist [], char data [], long count, short eotmode, short *ibsta, short *iberr, unsigned long *ibcntl)
SendLLO (121)	DLLSendLLO (short board, short *ibsta, short *iberr, unsigned long *ibcntl)
SendSetup (122)	DLLSendSetup (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)
SetRWLS (123)	DLLSetRWLS (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcntl)

(continues)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	Syntax
TestSRQ (124)	DLLTestSRQ (short board, short *result, short *ibsta, short *iberr, unsigned long *ibcnt1)
TestSys (125)	DLLTestSys (short board, short addresslist [], short resultlist [], short *ibsta, short *iberr, unsigned long *ibcnt1)
Trigger (126)	DLLTrigger (short board, short address, short *ibsta, short *iberr, unsigned long *ibcnt1)
TriggerList (127)	DLLTriggerList (short board, short addresslist [], short *ibsta, short *iberr, unsigned long *ibcnt1)
WaitSRQ (128)	DLLWaitSRQ (short board, short *result, short *ibsta, short *iberr, unsigned long *ibcnt1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note: All of the functions listed in Table A-2 are of type `short_far_pascal`.

Table A-2. Direct Entry NI-488 Style Functions

Functions (ordinal entry value)	Syntax
ibbna (10)	DLLibbna (short ud, char bname[], short *ibsta, short *iberr, unsigned long *ibcntl)
ibcac (11)	DLLibcac (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibclr (12)	DLLibclr (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibcmd (13)	DLLibcmd (short ud, char cmd[], long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibcmda (14)	DLLibcmda (short ud, char cmd[], long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibconfig (15)	DLLibconfig (short ud, unsigned short option, unsigned short cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibdev (16)	DLLibdev (short boardindex, short pad, short sad, short tmo, short eot, short eos, short *ibsta, short *iberr, unsigned long *ibcntl)
ibdma (18)	DLLibdma (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibeos (19)	DLLibeos (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibeot (20)	DLLibeot (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ (continues) คำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-2. Direct Entry NI-488 Style Functions (continued)

Functions (ordinal entry value)	Syntax
ibevent (21)	DLLibevent (short ud, short event, short *ibsta, short *iberr, unsigned long *ibcntl)
ibfind (22)	DLLibfind (char udname[], short *ibsta, short *iberr, unsigned long *ibcntl)
ibgts (23)	DLLibgts (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibist (24)	DLLibist (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
iblines (25)	DLLiblines (short ud, short *clines, short *ibsta, short *iberr, unsigned long *ibcntl)
ibln (26)	DLLibln (short ud, short pad, short sad, short *listen, short *ibsta, short *iberr, unsigned long *ibcntl)
ibloc (27)	DLLibloc (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibonl (28)	DLLibonl (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibpad (29)	DLLibpad (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibpct (30)	DLLibpct (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibppc (32)	DLLibppc (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrd (33)	DLLibrd (short ud, short rd[], unsigned long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)

(continues)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-2. Direct Entry NI-488 Style Functions (continued)

Functions (ordinal entry value)	Syntax
ibrda (34)	DLLibrda (short ud, char rd[], unsigned long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrdf (35)	DLLibrdf (short ud, char flname[], short *ibsta, short *iberr, unsigned long *ibcntl)
ibrdkey† (36)	DLLibrdkey (short ud, char rd[], unsigned short cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrpp (37)	DLLibrpp (short ud, char *ppr, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrsc (38)	DLLibrsc (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrsp (39)	DLLibrsp (short ud, char *spr, short *ibsta, short *iberr, unsigned long *ibcntl)
ibrsv (40)	DLLibrsv (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibsad (41)	DLLibsad (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibsic (42)	DLLibsic (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibsre (43)	DLLibsre (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)
ibstop (44)	DLLibstop (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibtmo (45)	DLLibtmo (short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)

(continues)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-2. Direct Entry NI-488 Style Functions (continued)

Functions (ordinal entry value)	Syntax
ibtrg (46)	DLLibtrg (short ud, short *ibsta, short *iberr, unsigned long *ibcntl)
ibwait (47)	DLLibwait (short ud, short mask, short *ibsta, short *iberr, unsigned long *ibcntl)
ibwrt (48)	DLLibwrt (short ud, char wrt[], unsigned long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibwrta (49)	DLLibwrta (short ud, char wrt[], unsigned long cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
ibwrtf (50)	DLLibwrtf (short ud, char flname[], short *ibsta, short *iberr, unsigned long *ibcntl)
ibwrtkey† (51)	DLLibwrtkey (short ud, char wrt[], unsigned short cnt, short *ibsta, short *iberr, unsigned long *ibcntl)
† ibrdkey and ibwrtkey are OEM functions. Refer to the <i>NI-488 Hardware Key Functions Reference Guide</i> for a detailed description of these functions.	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
 *   If you are using the Microsoft C compiler version 5.1 or later,
 *   the following two constants should be defined.
 */

#define MSC_FAR
#define MSC_CDECL

/*
 *   If you are using the Microsoft C compiler version 5.0
 *   the following two constants should be defined.
 *
 *   NOTE:  If you are going to use the function GotoMultAddr()
 *   you MUST specify the "/Au" command-line option.  The
 *   Microsoft C compiler version 5.0 does not support the
 *   "_loadds" keyword.  The "/Au" option directs the compiler
 *   to set the Data Segment correctly for ALL functions in
 *   your program.  The data segment MUST be set by the compiler
 *   or you will not be able to access you program's global
 *   variables from within the GotoMultAddr() call-back functions.
 *
 *   #define MSC_FAR far
 *   #define MSC_CDECL cdecl
 */

#ifdef __cplusplus
extern "C" {
#endif

/* C Declarations */
/* Status variables declared public by mcib.obj. */

extern short ibsta; /* status word */
extern short iberr; /* GPIB error code */
extern unsigned short ibcnt; /* number of bytes sent */
extern unsigned long ibcntl; /* (same) number of bytes sent */

#ifdef __cplusplus
}
#endif

/* GPIB Commands */

#define UNL 0x3f /* GPIB unlisten command */
#define UNT 0x5f /* GPIB untalk command */
#define GTL 0x01 /* GPIB go to local */
#define SDC 0x04 /* GPIB selected device clear */
#define PPC 0x05 /* GPIB parallel poll configure */
#define GET 0x08 /* GPIB group execute trigger */
#define TCT 0x09 /* GPIB take control */
#define LLO 0x11 /* GPIB local lock out */
#define DCL 0x14 /* GPIB device clear */
#define PPU 0x15 /* GPIB parallel poll unconfigure */
#define SPE 0x18 /* GPIB serial poll enable */
#define SPD 0x19 /* GPIB serial poll disable */
#define PPE 0x60 /* GPIB parallel poll enable */
#define PPD 0x70 /* GPIB parallel poll disable */

/* GPIB status bit vector :
 * global variable ibsta and wait mask
 */

```

```

#define ERR      (1<<15) /* Error detected */
#define TIMO     (1<<14) /* Timeout */
#define END      (1<<13) /* EOI or EOS detected */
#define SRQI     (1<<12) /* SRQ detected by CIC */
#define RQS      (1<<11) /* Device needs service */
#define SPOLL    (1<<10) /* Board has been serially polled */
#define EVENT    (1<<9)  /* An event has occurred */
#define CMPL     (1<<8)  /* I/O completed */
#define LOK      (1<<7)  /* Local lockout state */
#define REM      (1<<6)  /* Remote state */
#define CIC      (1<<5)  /* Controller-in-Charge */
#define ATN      (1<<4)  /* Attention asserted */
#define TACS     (1<<3)  /* Talker active */
#define LACS     (1<<2)  /* Listener active */
#define DTAS     (1<<1)  /* Device trigger state */
#define DCAS     (1<<0)  /* Device clear state */

/* Error messages returned in global variable iberr */

#define EDVR 0 /* DOS error */
#define ECIC 1 /* Function requires GPIB board to be CIC */
#define ENOL 2 /* Write function detected no Listeners */
#define EADR 3 /* Interface board not addressed correctly */
#define EARG 4 /* Invalid argument to function call */
#define ESAC 5 /* Function requires GPIB board to be SAC */
#define EABO 6 /* I/O operation aborted */
#define ENEB 7 /* Non-existent interface board */
#define EOIP 10 /* I/O operation started before previous
/* operation completed */
#define ECAP 11 /* No capability for intended operation */
#define EFSO 12 /* File system operation error */
#define EBUS 14 /* Command error during device call */
#define ESTB 15 /* Serial poll status byte lost */
#define ESRQ 16 /* SRQ remains asserted */
#define ETAB 20 /* The return buffer is full.

/* EOS mode bits

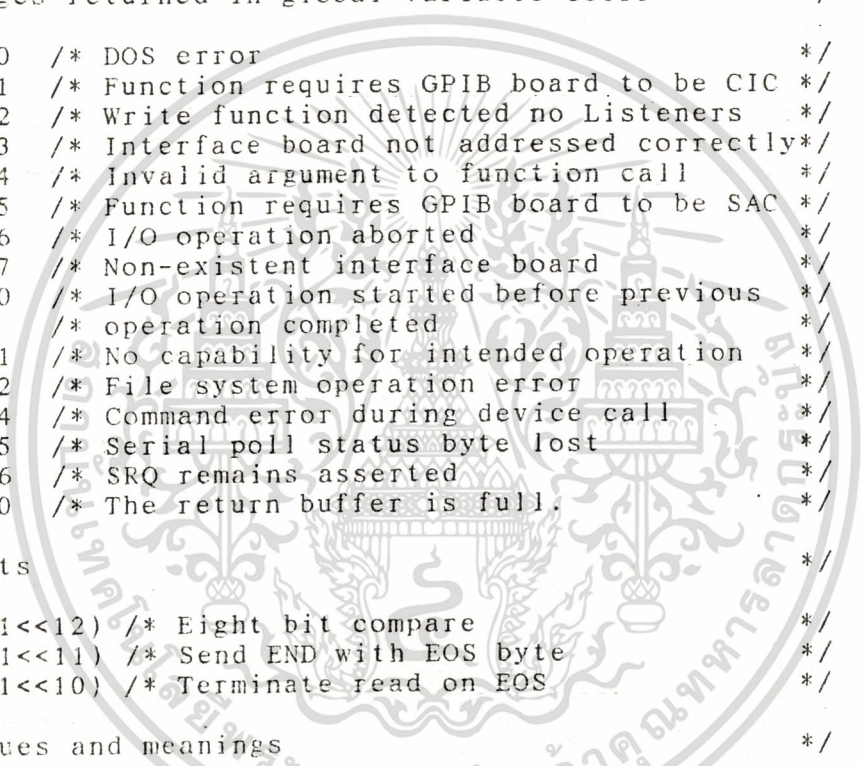
#define BIN (1<<12) /* Eight bit compare */
#define XEOS (1<<11) /* Send END with EOS byte */
#define REOS (1<<10) /* Terminate read on EOS

/* Timeout values and meanings

#define TNONE 0 /* Infinite timeout (disabled) */
#define T10us 1 /* Timeout of 10 us (ideal) */
#define T30us 2 /* Timeout of 30 us (ideal) */
#define T100us 3 /* Timeout of 100 us (ideal) */
#define T300us 4 /* Timeout of 300 us (ideal) */
#define T1ms 5 /* Timeout of 1 ms (ideal) */
#define T3ms 6 /* Timeout of 3 ms (ideal) */
#define T10ms 7 /* Timeout of 10 ms (ideal) */
#define T30ms 8 /* Timeout of 30 ms (ideal) */
#define T100ms 9 /* Timeout of 100 ms (ideal) */
#define T300ms 10 /* Timeout of 300 ms (ideal) */
#define T1s 11 /* Timeout of 1 s (ideal) */
#define T3s 12 /* Timeout of 3 s (ideal) */
#define T10s 13 /* Timeout of 10 s (ideal) */
#define T30s 14 /* Timeout of 30 s (ideal) */
#define T100s 15 /* Timeout of 100 s (ideal) */
#define T300s 15 /* Timeout of 300 s (ideal) */
#define T1000s 15 /* Timeout of 1000 s (ideal)

/* IBLN Constants

```



เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ (ส.ค.อ.) กระทรวงพาณิชย์
 ไม่สามารถนำเอกสารนี้ไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต
 หมายเหตุ: ห้ามนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต
 หมายเหตุ: ห้ามนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต

```
#define NO_SAD 0
#define ALL_SAD -1
```

```
/* IBEVENT Constants */
```

```
#define EventDTAS 1
#define EventDCAS 2
```

```
/*
 * GotoMultAddr flags
 */
```

```
#define MultAddrPrimary 0x00
#define MultAddrSecondary 0x01
#define MultAddrListen 0x00
#define MultAddrTalk 0x01
#define MultAddrSerialPoll 0x81
```

```
/* Miscellaneous */
```

```
#define S 0x08 /* parallel poll sense bit */
#define LF 0x0A /* ASCII linefeed character */
```

```
/* The following constants are used for the second parameter of the
 * ibconfig function. They are the "option" selection codes.
 */
```

```
#define IbcPAD 0x0001 /* Primary Address */
#define IbcSAD 0x0002 /* Secondary Address */
#define IbcTMO 0x0003 /* Timeout Value */
#define IbcEOT 0x0004 /* Send EOI with last data byte? */
#define IbcPPC 0x0005 /* Parallel Poll Configure */
#define IbcREADDR 0x0006 /* Repeat Addressing */
#define IbcAUTOPOLL 0x0007 /* Disable Auto Serial Polling */
#define IbcCICPROT 0x0008 /* Use the CIC Protocol? */
#define IbcIRQ 0x0009 /* Use PIO for I/O */
#define IbcSC 0x000A /* Board is System Controller? */
#define IbcSRE 0x000B /* Assert SRE on device calls? */
#define IbcEOSrd 0x000C /* Terminate reads on EOS */
#define IbcEOSwrt 0x000D /* Send EOI with EOS character */
#define IbcEOScmp 0x000E /* Use 7 or 8-bit EOS compare */
#define IbcEOSchar 0x000F /* The EOS character. */
#define IbcPP2 0x0010 /* Use Parallel Poll Mode 2. */
#define IbcTIMING 0x0011 /* NORMAL, HIGH, or VERY_HIGH timing. */
#define IbcDMA 0x0012 /* Use DMA for I/O */
#define IbcReadAdjust 0x0013 /* Swap bytes during an ibrd. */
#define IbcWriteAdjust 0x0014 /* Swap bytes during an ibwrt. */
#define IbcEventQueue 0x0015 /* Enable/disable the event queue. */
#define IbcSPollBit 0x0016 /* Enable/disable the visibility of SPOLL. */
#define IbcSendLLO 0x0017 /* Enable/disable the sending of LLO. */
#define IbcSPollTime 0x0018 /* Set the timeout value for serial polls. */
#define IbcPPollTime 0x0019 /* Set the parallel poll length period. */
#define IbcNoEndBitOnEOS 0x001A /* Remove EOS from END bit of IBSTA. */
```

```
/* These are the values are used by the Send 488.2 command. */
```

```
#define NULLend (unsigned short)0x00 /* Do nothing at the end of a transfer. */
#define NLend (unsigned short)0x01 /* Send NL with EOI after a transfer. */
#define DABend (unsigned short)0x02 /* Send EOI with the last DAB. */
```

```
/* This value is used by the 488.2 Receive command.
 */
```

```
#define STOPend (unsigned short)0x0100
```

```

/*
 * This macro can be used to easily create an entry in address list
 * that is required by many of the 488.2 functions. An address list is
 * just an array of unsigned integers. The primary address goes in the
 * lower 8-bits and the secondary address goes in the upper 8-bits.
 */
#define MakeAddr(pad, sad) ((unsigned short)((unsigned char)(pad) | ((unsigned char)(sad) << 8)))

/*
 * The following two macros are used to "break apart" an address list
 * entry. They take an unsigned integer and return either the primary
 * or secondary address stored in the integer.
 */
#define GetPAD(val) ((unsigned char)(val))
#define GetSAD(val) ((unsigned char)((val) >> 8))

/*
 * The following values are used by the iblines function. The unsigned
 * integer returned by iblines contains:
 * The lower byte will contain a "monitor" bit mask. If a bit is
 * set (1) in this mask, then the corresponding line can be
 * monitored by the driver. If the bit is clear (0),
 * then the line cannot be monitored.
 * The upper byte will contain the status of the bus lines.
 * Each bit corresponds to a certain bus line, and has
 * a corresponding "monitor" bit in the lower byte.
 */
#define ValidEOI ((unsigned short)0x0080)
#define ValidATN ((unsigned short)0x0040)
#define ValidSRQ ((unsigned short)0x0020)
#define ValidREN ((unsigned short)0x0010)
#define ValidIFC ((unsigned short)0x0008)
#define ValidNRFD ((unsigned short)0x0004)
#define ValidNDAC ((unsigned short)0x0002)
#define ValidDAV ((unsigned short)0x0001)
#define BusEOI ((unsigned short)0x8000)
#define BusATN ((unsigned short)0x4000)
#define BusSRQ ((unsigned short)0x2000)
#define BusREN ((unsigned short)0x1000)
#define BusIFC ((unsigned short)0x0800)
#define BusNRFD ((unsigned short)0x0400)
#define BusNDAC ((unsigned short)0x0200)
#define BusDAV ((unsigned short)0x0100)

/*
 * This value is used to terminate an address list. It should be
 * assigned to the last entry.
 */
#define NOADDR 0xFFFF

#ifdef __cplusplus
extern "C" {
#endif

extern short MSC_FAR MSC_CDECL ibbna(short handle, char MSC_FAR *bdname);
extern short MSC_FAR MSC_CDECL ibcac(short handle, short v);
extern short MSC_FAR MSC_CDECL ibclr(short handle);
extern short MSC_FAR MSC_CDECL ibcmd(short handle, char MSC_FAR *buffer, unsigned long cnt);

```

```
extern short MSC_FAR MSC_CDECL ibcmnda(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibconfig(short handle, unsigned short option, unsigned short value);
extern short MSC_FAR MSC_CDECL ibdev(short boardID, short padval, short sadval, short tmoval, short eotval, short eosval);
extern short MSC_FAR MSC_CDECL ibdma(short handle, short v);
extern short MSC_FAR MSC_CDECL ibeos(short handle, short v);
extern short MSC_FAR MSC_CDECL ibeot(short handle, short v);
extern short MSC_FAR MSC_CDECL ibevent(short handle, unsigned short MSC_FAR *event);
extern short MSC_FAR MSC_CDECL ibfind(char MSC_FAR *bdname);
extern short MSC_FAR MSC_CDECL ibgts(short handle, short v);
extern short MSC_FAR MSC_CDECL ibist(short handle, short v);
extern short MSC_FAR MSC_CDECL iblines(short handle, unsigned short MSC_FAR *lines);
extern short MSC_FAR MSC_CDECL ibln(short handle, short padval, short sadval, short MSC_FAR *listenflag);
extern short MSC_FAR MSC_CDECL ibloc(short handle);
extern short MSC_FAR MSC_CDECL ibonl(short handle, short v);
extern short MSC_FAR MSC_CDECL ibpad(short handle, short v);
extern short MSC_FAR MSC_CDECL ibpct(short handle);
extern short MSC_FAR MSC_CDECL ibppe(short handle, short v);
extern short MSC_FAR MSC_CDECL ibrd(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibrda(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibrdf(short handle, char MSC_FAR *filename);
extern short MSC_FAR MSC_CDECL ibrpp(short handle, char MSC_FAR *ppr);
extern short MSC_FAR MSC_CDECL ibrse(short handle, short v);
extern short MSC_FAR MSC_CDECL ibrsp(short handle, char MSC_FAR *spr);
extern short MSC_FAR MSC_CDECL ibrsv(short handle, short v);
extern short MSC_FAR MSC_CDECL ibsad(short handle, short v);
extern short MSC_FAR MSC_CDECL ibsic(short handle);
extern short MSC_FAR MSC_CDECL ibsre(short handle, short v);
extern short MSC_FAR MSC_CDECL ibsrq(void (MSC_FAR *func)(void));
extern short MSC_FAR MSC_CDECL ibstop(short handle);
extern short MSC_FAR MSC_CDECL ibtmo(short handle, short v);
extern short MSC_FAR MSC_CDECL ibtrap(short mask, short mode);
extern short MSC_FAR MSC_CDECL ibtrg(short handle);
extern short MSC_FAR MSC_CDECL ibwait(short handle, unsigned short mask);
extern short MSC_FAR MSC_CDECL ibwrt(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibwrtta(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibwrtf(short handle, char MSC_FAR *filename);

extern short MSC_FAR MSC_CDECL ibpoke(short handle, unsigned short option, unsigned short value);
extern short MSC_FAR MSC_CDECL ibdiag(short handle, char MSC_FAR *buffer, unsigned long cnt);
extern short MSC_FAR MSC_CDECL ibxtrc(short handle, char MSC_FAR *buffer, unsigned long cnt);

extern short MSC_FAR MSC_CDECL ibwrtkey(short handle, char MSC_FAR *buffer, unsigned short cnt);
extern short MSC_FAR MSC_CDECL ibrdkey(short handle, char MSC_FAR *buffer, unsigned short cnt);

extern void MSC_FAR MSC_CDECL SendCmds(short boardID, char MSC_FAR *buf, unsigned long cnt);
extern void MSC_FAR MSC_CDECL SendSetup(short boardID, unsigned int MSC_FAR *listen);
extern void MSC_FAR MSC_CDECL SendDataBytes(short boardID, char MSC_FAR *buffer, unsigned long cnt, unsigned int eot_mode);
extern void MSC_FAR MSC_CDECL Send(short boardID, unsigned int listener, char MSC_FAR *buf, unsigned long cnt);
```

```

_FAR *databuf, unsigned long datacnt, unsigned int eotMode);
extern void MSC_FAR MSC_CDECL SendList(short boardID, unsigned int MSC_FAR *list
ners, char MSC_FAR *databuf, unsigned long datacnt, unsigned int eotMode);
extern void MSC_FAR MSC_CDECL ReceiveSetup(short boardID, unsigned int talker);
extern void MSC_FAR MSC_CDECL RcvRespMsg(short boardID, char MSC_FAR *buffer, un
igned long cnt, unsigned int eotMode);
extern void MSC_FAR MSC_CDECL Receive(short boardID, unsigned int talker, char M
C_FAR *buffer, unsigned long cnt, unsigned int eotMode);
extern void MSC_FAR MSC_CDECL SendIFC(short boardID);
extern void MSC_FAR MSC_CDECL DevClear(short boardID, unsigned int address);
extern void MSC_FAR MSC_CDECL DevClearList(short boardID, unsigned int MSC_FAR *
ddrlist);
extern void MSC_FAR MSC_CDECL EnableLocal(short boardID, unsigned int MSC_FAR *l
ddrs);
extern void MSC_FAR MSC_CDECL EnableRemote(short boardID, unsigned int MSC_FAR *
addrs);
extern void MSC_FAR MSC_CDECL SetRWLS(short boardID, unsigned int MSC_FAR *laddr
);
extern void MSC_FAR MSC_CDECL SendLLO(short boardID);
extern void MSC_FAR MSC_CDECL PassControl(short boardID, unsigned int talker);
extern void MSC_FAR MSC_CDECL ReadStatusByte(short boardID, unsigned int talker,
unsigned int MSC_FAR *result);
extern void MSC_FAR MSC_CDECL Trigger(short boardID, unsigned int laddr);
extern void MSC_FAR MSC_CDECL TriggerList(short boardID, unsigned int MSC_FAR *l
ddrs);
extern void MSC_FAR MSC_CDECL PPollConfig(short boardID, unsigned int laddr, uns
igned int dataLine, unsigned int lineSense);
extern void MSC_FAR MSC_CDECL PPollUnconfig(short boardID, unsigned int MSC_FAR
laddr);
extern void MSC_FAR MSC_CDECL PPoll(short boardID, unsigned int MSC_FAR *res_ptr
);
extern void MSC_FAR MSC_CDECL TestSRQ(short boardID, int MSC_FAR *result);
extern void MSC_FAR MSC_CDECL WaitSRQ(short boardID, int MSC_FAR *result);
extern void MSC_FAR MSC_CDECL ResetSys(short boardID, unsigned int MSC_FAR *laddr
);
extern void MSC_FAR MSC_CDECL FindRQS(short boardID, unsigned int MSC_FAR *taddr
, unsigned int MSC_FAR *dev_stat);
extern void MSC_FAR MSC_CDECL AllSpoll(short boardID, unsigned int MSC_FAR *taddr
s, unsigned int MSC_FAR *res);
extern void MSC_FAR MSC_CDECL FindLstn(short boardID, unsigned int MSC_FAR *pads
, unsigned int MSC_FAR *results, unsigned int limit);
extern void MSC_FAR MSC_CDECL TestSys(short boardID, unsigned int MSC_FAR *addrs
, unsigned int MSC_FAR *result);
extern void MSC_FAR MSC_CDECL GotoMultAddr(short handle, unsigned int type,
unsigned int (MSC_FAR *addrfunc)(),
unsigned int (MSC_FAR *spollfunc)());
extern void MSC_FAR MSC_CDECL GenerateREQT(short handle, unsigned int addr);
extern void MSC_FAR MSC_CDECL GenerateREQF(short handle, unsigned int addr);

#ifdef __cplusplus
endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LMC835 Digital Controlled Graphic Equalizer

General Description

The LMC835 is a monolithic, digitally-controlled graphic equalizer CMOS LSI for Hi-Fi audio. The LMC835 consists of a Logic section and a Signal Path section made of analog switches and thin-film silicon-chromium resistor networks. The LMC835 is used with external resonator circuits to make a stereo equalizer with seven bands, ± 12 dB or ± 6 dB gain range and 25 steps each. Only three digital inputs are needed to control the equalization. The LMC835 makes it easy to build a μ P-controlled equalizer.

The signal path is designed for very low noise and distortion, resulting in very high performance, compatible with PCM audio.

Features

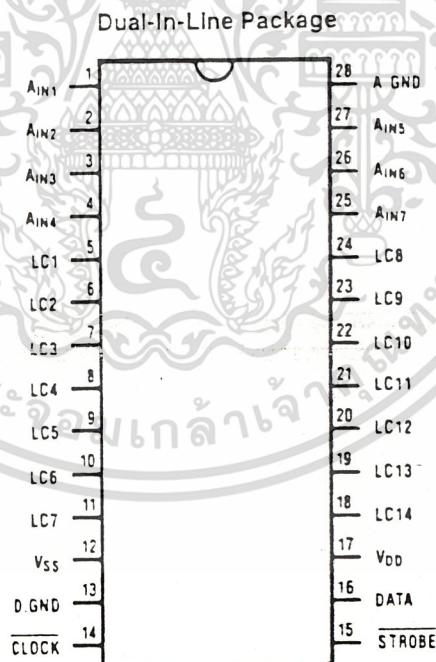
- No volume controls required
- Three-wire interface
- 14 bands, 25 steps each
- ± 12 dB or ± 6 dB gain ranges
- Low noise and distortion
- TTL, CMOS logic compatible

Applications

- Hi-Fi equalizer
- Receiver
- Car stereo
- Musical instrument
- Tape equalization
- Mixer
- Volume controller

Connection Diagram

Order Number LMC835N
See NS Package N28B

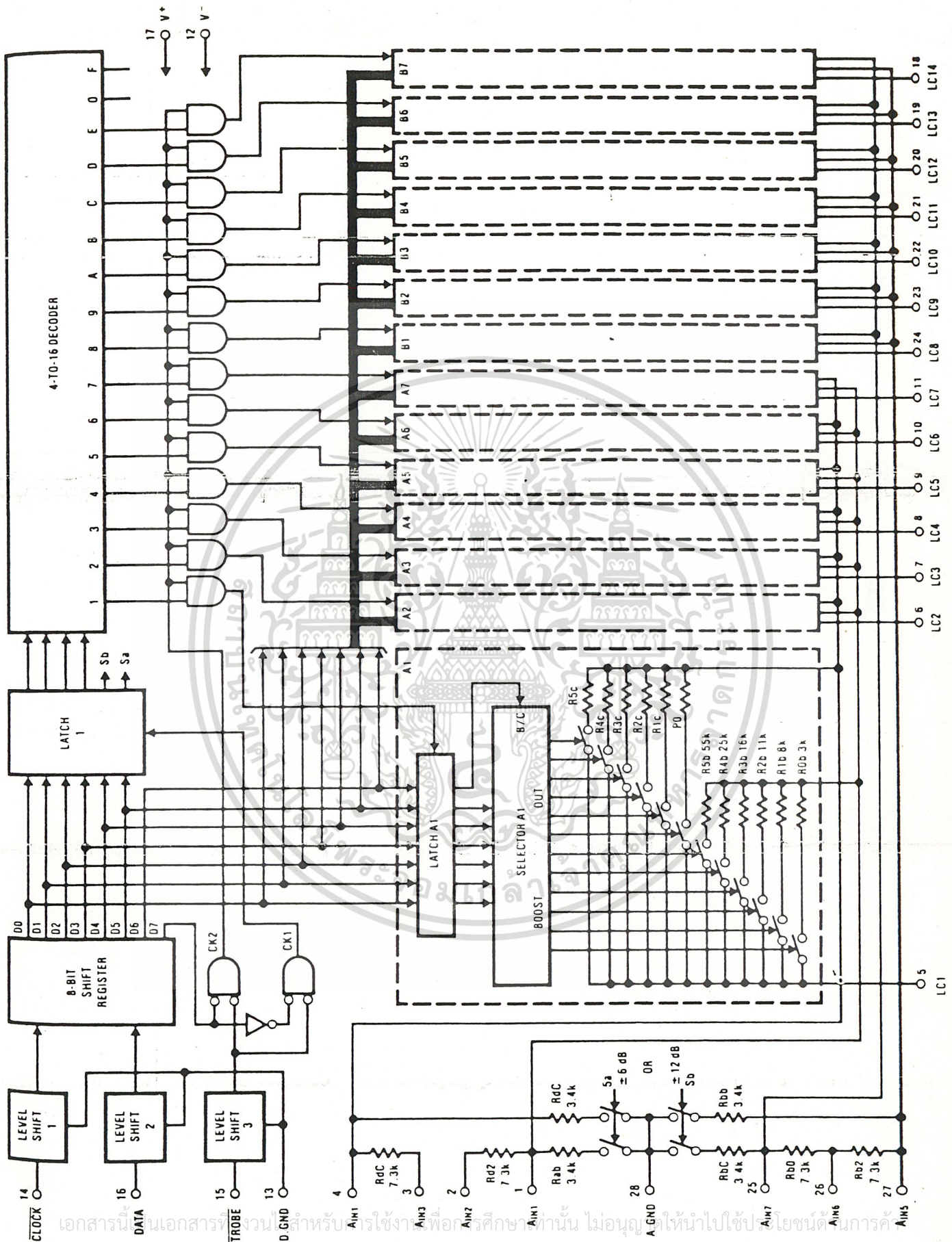


Top View

TL7H/6753-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ในการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, $V_{DD} - V_{SS}$	18V
Allowable Input Voltage (Note 1)	$V_{SS} - 0.3V$ to $V_{DD} + 0.3V$
Storage Temperature, T_{stg}	$-60^{\circ}C$ to $+150^{\circ}C$
Lead Temperature (Soldering, 10 sec), T_L	$+260^{\circ}C$

Operating Ratings

Supply Voltage, $V_{DD} - V_{SS}$	5V to 16V
Digital Ground (Pin 13)	V_{SS} to V_{DD}
Digital Input (Pins 14, 15, 16)	V_{SS} to V_{DD}
Analog Input (Pins 1, 2, 3, 4, 25, 26, 27) (Note 1)	V_{SS} to V_{DD}
Operating Temperature, T_{opr}	$-40^{\circ}C$ to $+85^{\circ}C$

Electrical Characteristics (Note 2) $V_{DD} = 7.5V$, $V_{SS} = -7.5V$, A.GND = 0V

LOGIC SECTION

Symbol	Parameter	Test Conditions	Typ	Tested Limit (Note 3)	Design Limit (Note 4)	Unit (Limit)
I_{DDL}	Supply Current	Pins 14, 15, 16 are 0V	0.01	0.5	0.5	mA (Max)
I_{SSL}		Pins 14, 15, 16 are 0V	0.01	0.5	0.5	mA (Max)
I_{DDH}		Pins 14, 15, 16 are 5V	1.3	5	5	mA (Max)
I_{SSH}		Pins 14, 15, 16 are 5V	0.9	5	5	mA (Max)
V_{IH}	High-Level Input Voltage	@Pins 14, 15, 16	1.6	2.3	2.5	V (Min)
V_{IL}	Low-Level Input Voltage	@Pins 14, 15, 16	0.9	0.6	0.4	V (Max)
f_o	Clock Frequency	@Pin 14	2000	500	500	kHz (Max)
$t_w(\overline{STB})$	Width of \overline{STB} Input	See Figure 1	0.25	1	1	μs (Min)
t_{setup}	Data Setup Time	See Figure 1	0.25	1	1	μs (Min)
t_{hold}	Data Hold Time	See Figure 1	0.25	1	1	μs (Min)
t_{cs}	Delay from Rising Edge of \overline{CLOCK} to \overline{STB}	See Figure 1	0.25	1	1	μs (Min)
I_{IN}	Input Current	@Pins 14, 15, 16 $0V < V_{IN} < 5V$	± 0.01	± 1		μA (Max)
C_{IN}	Input Capacitance	@Pins 14, 15, 16 $f = 1$ MHz	5			pF

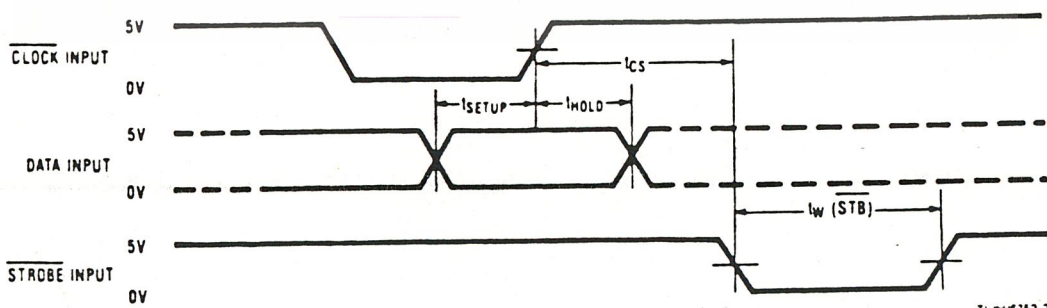
Note 1: Pins 2, 3 and 26 have a maximum input voltage range of $\pm 22V$ for the typical application shown in Figure 7.

Note 2: Bold numbers apply at temperature extremes. All other numbers apply at $T_A = 25^{\circ}C$, $V_{DD} = 7.5V$, $V_{SS} = -7.5V$, D.GND = A.GND = 0V as shown in the test circuit, Figures 3 and 4.

Note 3: Guaranteed and 100% production tested.

Note 4: Guaranteed (but not 100% production tested) over the operating temperature range. These limits are not used to calculate outgoing quality levels.

Timing Diagram



TL/H/6753-3

TL/H/6753-3

Note: To change the gain of the presently selected band, it is not necessary to send DATA 1 (Band Selection) each time.

FIGURE 1

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Note 2) $V_{DD} = 7.5V$, $V_{SS} = -7.5V$, D.GND = A.GND = 0V

SIGNAL PATH SECTION

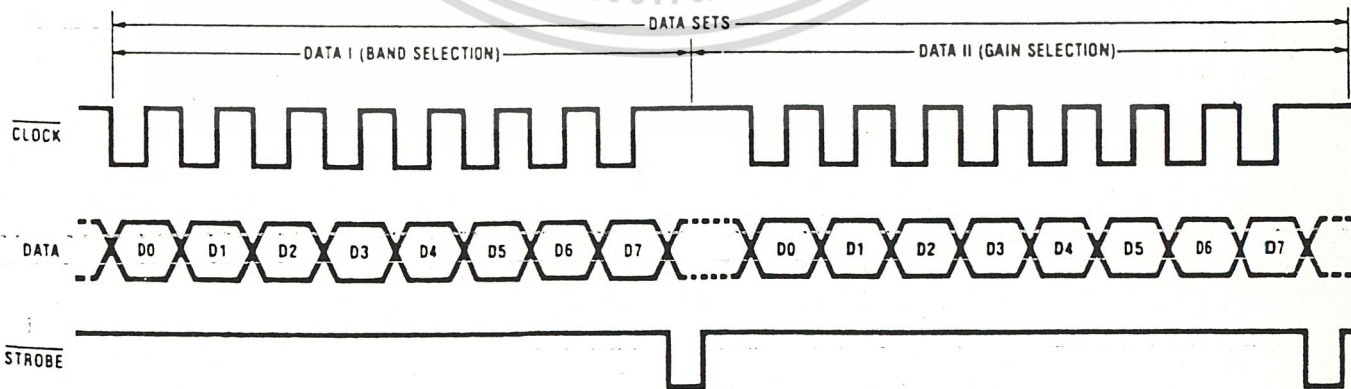
Symbol	Parameter	Test Conditions	Typ	Tested Limit (Note 3)	Design Limit (Note 4)	Unit (Limit)
EA	Gain Error	$A_V = 0 \text{ dB} @ \pm 12 \text{ dB Range}$	0.1	0.5	0.5	dB (Max)
		$A_V = 0 \text{ dB} @ \pm 6 \text{ dB Range}$	0.1	1	1	dB (Max)
		$A_V = \pm 1 \text{ dB} @ \pm \text{dB Range}$ (R_{5b} or R_{5c} is ON)	0.1	0.5	0.6	dB (Max)
		$A_V = \pm 2 \text{ dB} @ \pm 12 \text{ dB Range}$ (R_{4b} or R_{4c} is ON)	0.1	0.5	0.6	dB (Max)
		$A_V = \pm 3 \text{ dB} @ \pm 12 \text{ dB Range}$ (R_{3b} or R_{3c} is ON)	0.1	0.5	0.6	dB (Max)
		$A_V = \pm 4 \text{ dB} @ \pm 12 \text{ dB Range}$ (R_{2b} or R_{2c} is ON)	0.1	0.5	0.7	dB (Max)
		$A_V = \pm 5 \text{ dB} @ \pm 12 \text{ dB Range}$ (R_{1b} or R_{1c} is ON)	0.1	0.5	0.7	dB (Max)
		$A_V = \pm 9 \text{ dB} @ \pm 12 \text{ dB Range}$ (R_{0b} or R_{0c} is ON)	0.2	1	1.3	dB (Max)
THD	Total Harmonic Distortion	$A_V = 0 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{IN} = 4V_{rms}$, $f = 1 \text{ kHz}$	0.0015			%
		$A_V = 12 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{IN} = 1V_{rms}$, $f = 1 \text{ kHz}$	0.01	0.1		% (Max)
		$V_{IN} = 1V_{rms}$, $f = 20 \text{ kHz}$	0.1	0.5		% (Max)
		$A_V = -12 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{IN} = 4V_{rms}$, $f = 1 \text{ kHz}$	0.01	0.1		% (Max)
		$V_{IN} = 4V_{rms}$, $f = 20 \text{ kHz}$	0.1	0.5		% (Max)
$V_{O \text{ Max}}$	Maximum Output Voltage	$A_V = 0 \text{ dB} @ \pm 12 \text{ dB Range}$ THD < 1%, $f = 1 \text{ kHz}$	5.5	5.1	5	V_{rms} (Min)
S/N	Signal to Noise Ratio	$A_V = 0 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{ref} = 1V_{rms}$	114			dB
		$A_V = 12 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{ref} = 1V_{rms}$	106			dB
		$A_V = -12 \text{ dB} @ \pm 12 \text{ dB Range}$ $V_{ref} = 1V_{rms}$	116			dB
I_{LEAK}	Leakage Current	$A_V = 0 \text{ dB} @ \pm 12 \text{ dB Range}$ (All internal switches are OFF) Pin 2 + 3, Pin 26 Pin 5 ~ Pin 11, Pin 18 ~ Pin 24		500 50		nA (Max) nA (Max)

Note 2: Boldface numbers apply at temperature extremes. All other numbers apply at $T_A = 25^\circ\text{C}$, $V_{DD} = 7.5V$, $V_{SS} = -7.5V$, D.GND = A.GND = 0V as shown in the test circuit, Figures 3 and 4.

Note 3: Guaranteed and 100% production tested.

Note 4: Guaranteed (but not 100% production tested) over the operating temperature range. These limits are not used to calculate outgoing quality levels.

Timing Diagrams



Note: To change the gain of the presently selected band, it is not necessary to send DATA 1 (Band Selection) each time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

FIGURE 2

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits

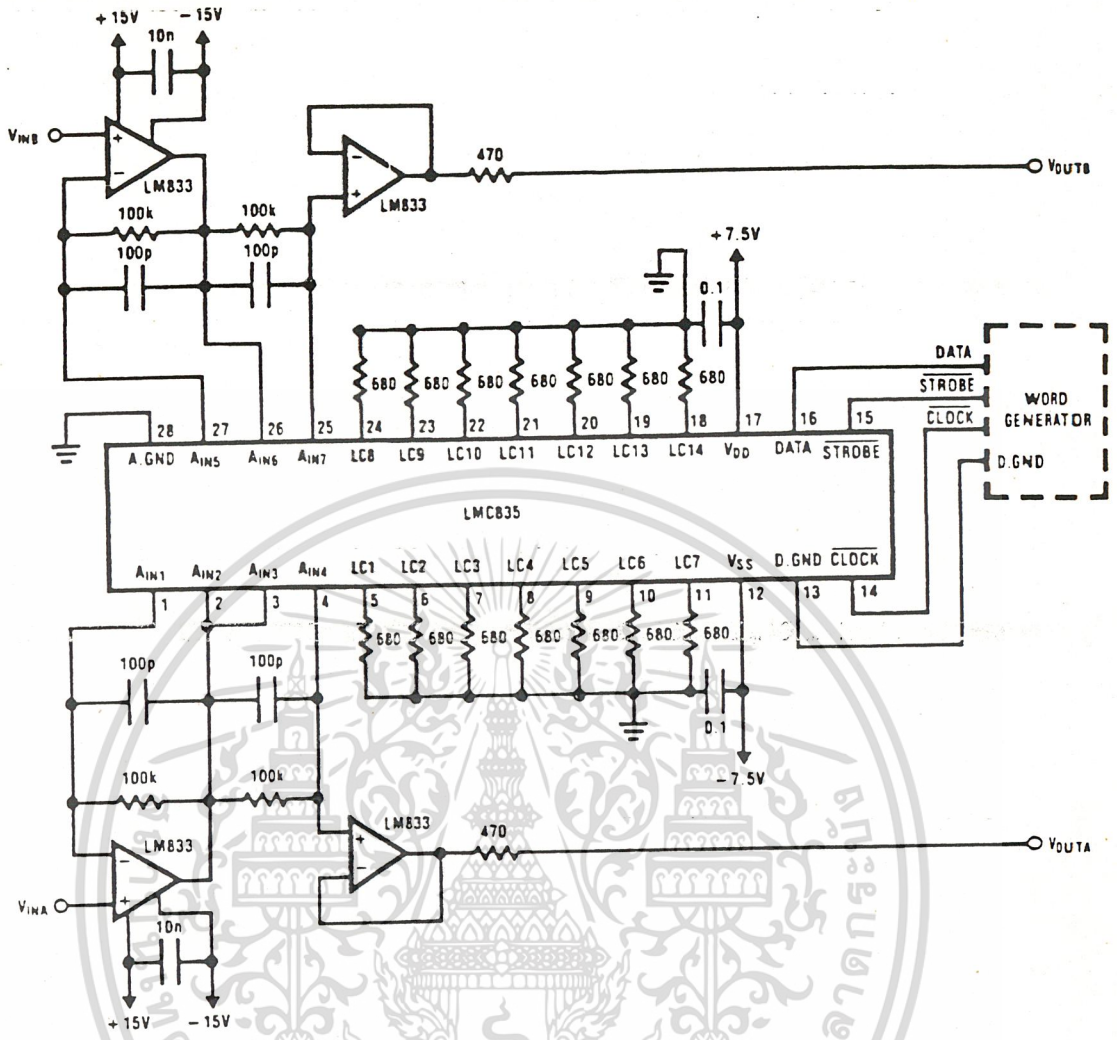


FIGURE 3. Test Circuit for AC Measurement

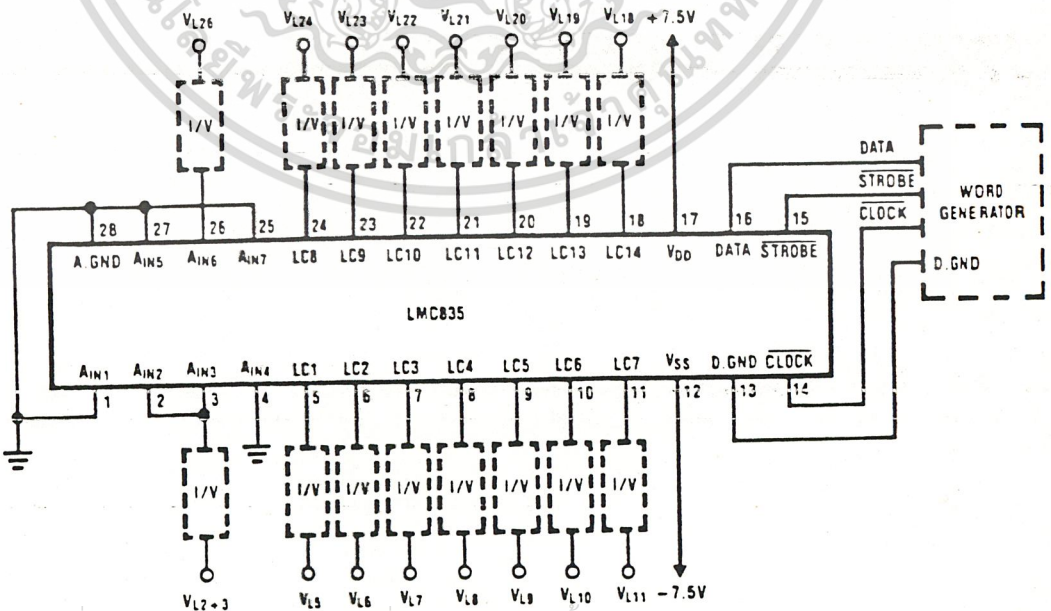


FIGURE 4. Test Circuit for Leakage Current Measurement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Truth Tables

DATA I (Band Selection)

D7	D6	D5	D4	D3	D2	D1	D0	
H	X	L	L	L	L	L	L	
H	X	L	L	L	L	L	H	
H	X	L	L	L	L	H	L	
H	X	L	L	L	L	H	H	
H	X	L	L	L	H	L	L	
H	X	L	L	L	H	L	H	
H	X	L	L	L	H	H	L	
H	X	L	L	L	H	H	H	
H	X	L	L	H	L	L	L	
H	X	L	L	H	L	L	H	
H	X	L	L	H	L	H	L	
H	X	L	L	H	L	H	H	
H	X	L	L	H	H	L	L	
H	X	L	L	H	H	L	H	
H	X	L	L	H	H	H	L	
H	X	L	L	H	H	H	H	
H	X	L	H	Valid Binary Input				
H	X	H	L	Valid Binary Input				
H	X	H	H	Valid Binary Input				
↑	↑	↑	↑	← Band Code →				
①	②	③	④					

- ① DATA 1
- ② Don't Care
- ③ Ch A ± 6 dB / ± 12 dB Range
- ④ Ch B ± 6 dB / ± 12 dB Range

(Ch A: Band 1 ~ 7, Ch B: Band 8 ~ 14)

- Ch A ± 12 dB Range, Ch B ± 12 dB Range, No Band Selection
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 1
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 2
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 3
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 4
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 5
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 6
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 7
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 8
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 9
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 10
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 11
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 12
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 13
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, Band 14
- Ch A ± 12 dB Range, Ch B ± 12 dB Range, No Band Selection
- Ch A ± 12 dB Range, Ch B ± 6 dB Range, Band 1 ~ 14
- Ch A ± 6 dB Range, Ch B ± 12 dB Range, Band 1 ~ 14
- Ch A ± 6 dB Range, Ch B ± 6 dB Range, Band 1 ~ 14

DATA II (Gain Selection)

D7	D6	D5	D4	D3	D2	D1	D0	
L	X	L	L	L	L	L	L	
L	H	H	L	L	L	L	L	
L	H	L	H	L	L	L	L	
L	H	L	L	H	L	L	L	
L	H	L	L	L	H	L	L	
L	H	L	L	L	L	H	L	
L	H	L	H	L	L	H	L	
L	H	H	L	H	L	H	L	
L	H	L	H	L	H	H	L	
L	H	L	L	L	L	L	H	
L	H	H	L	H	L	L	H	
L	H	H	L	H	H	L	H	
L	H	H	L	H	H	H	H	
L	L	Valid Above Input						
↑	↑	← Gain Code →						
⑤	⑥							

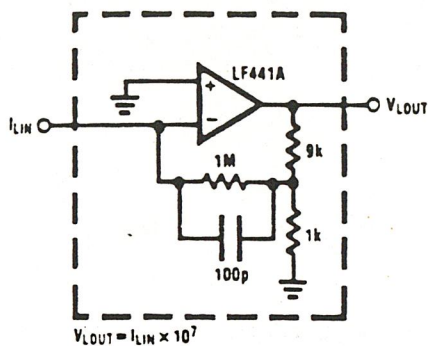
This is the gain if the ± 12 dB range is selected by DATA I. If the ± 6 dB range is selected, then the values shown must be approximately halved. See the characteristics curves for more exact data.

- Flat
- 1 dB Boost
- 2 dB Boost
- 3 dB Boost
- 4 dB Boost
- 5 dB Boost
- 6 dB Boost
- 7 dB Boost
- 8 dB Boost
- 9 dB Boost
- 10 dB Boost
- 11 dB Boost
- 12 dB Boost
- 1 dB ~ 12 dB Cut

- ⑤ DATA II
- ⑥ Boost/Cut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

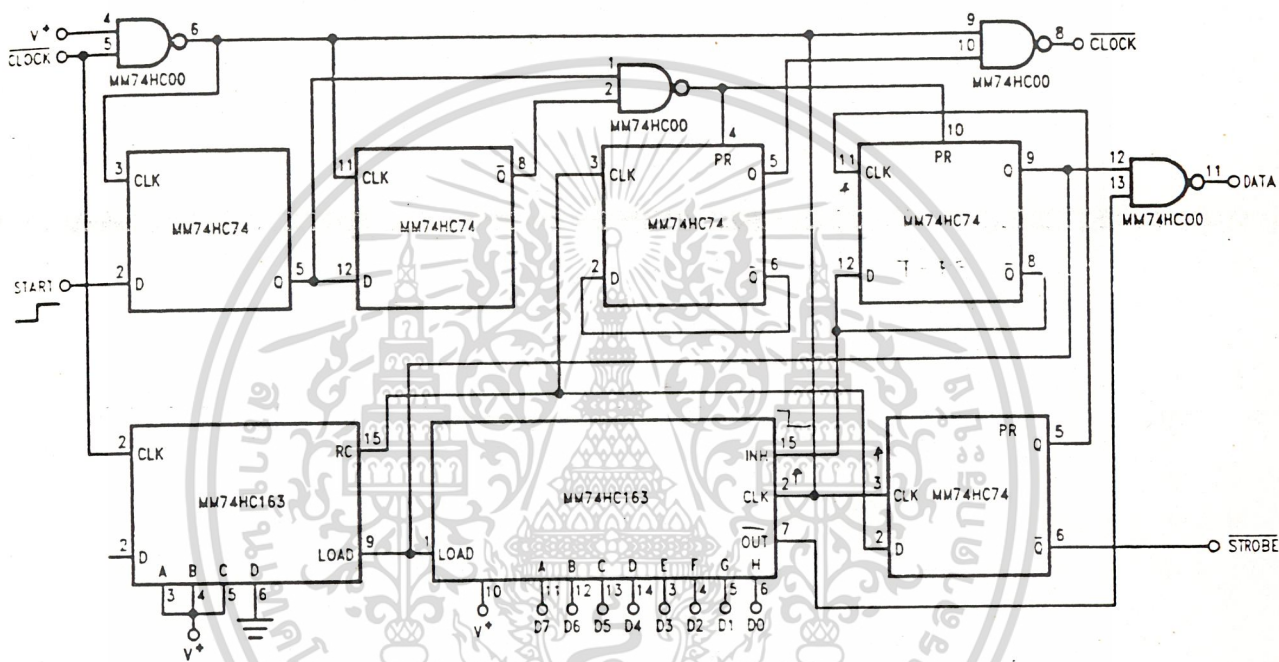
Test Circuits (Continued)



$$V_{LOUT} = I_{LIN} \times 10^7$$

TL/H/6753-7

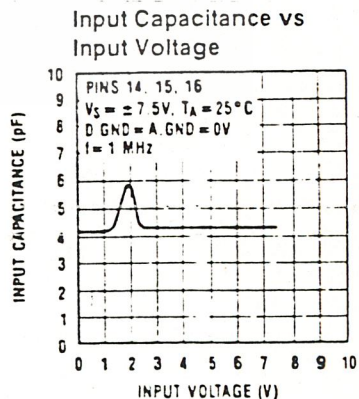
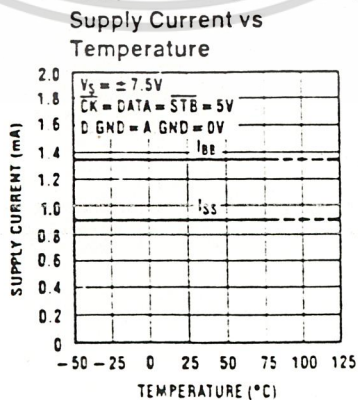
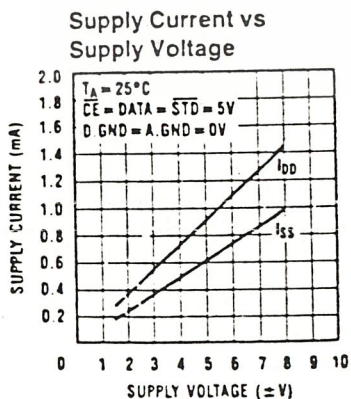
FIGURE 5. I to V Converter



TL/H/6753-B

FIGURE 6. Simple Word Generator

Typical Performance Characteristics

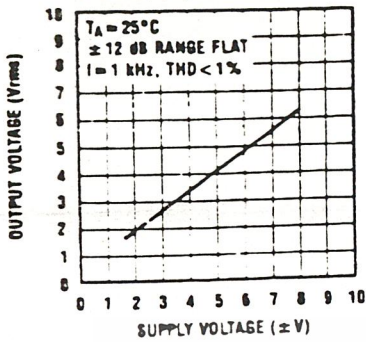


TL/H/6753-9

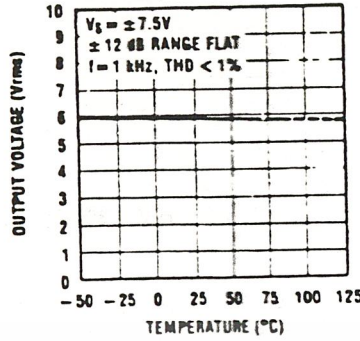
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)

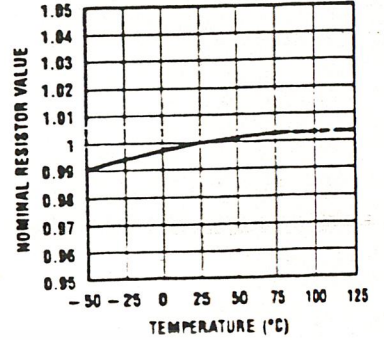
Maximum Output Voltage vs Supply Voltage



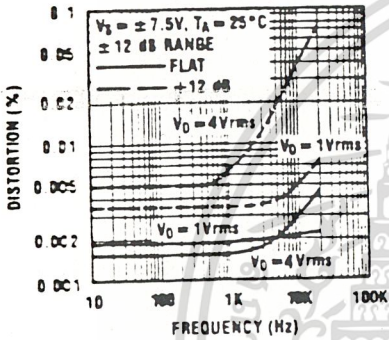
Maximum Output Voltage vs Temperature



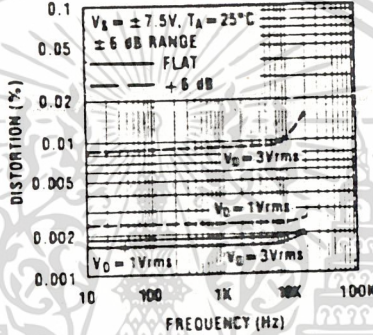
Nominal Resistor vs Temperature



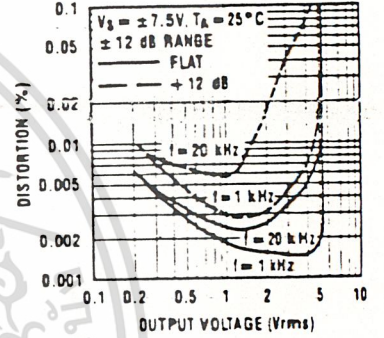
Distortion vs Frequency @ ±12 dB Range



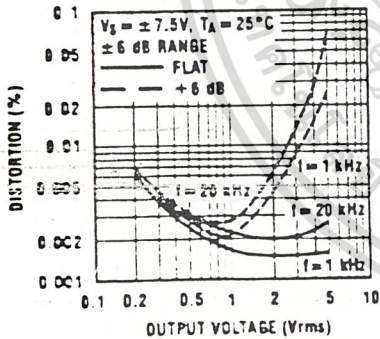
Distortion vs Frequency @ ±6 dB Range



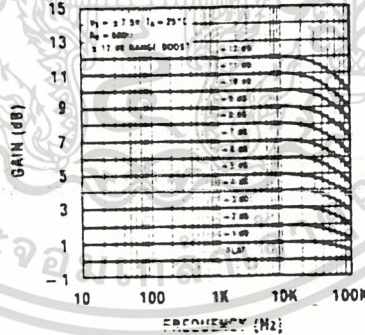
Distortion vs Output Voltage @ ±12 dB Range



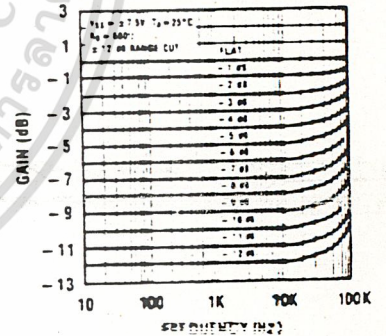
Distortion vs Output Voltage @ ±6 dB Range



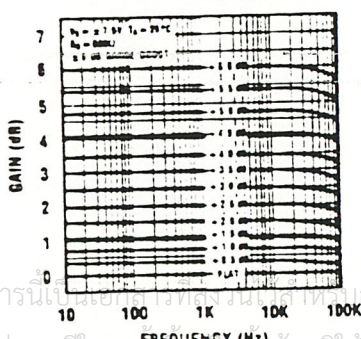
Gain vs Frequency @ ±12 dB Range (Boost)



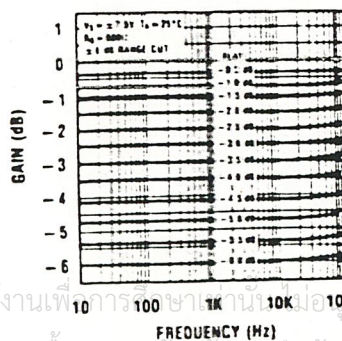
Gain vs Frequency @ ±12 dB Range (Cut)



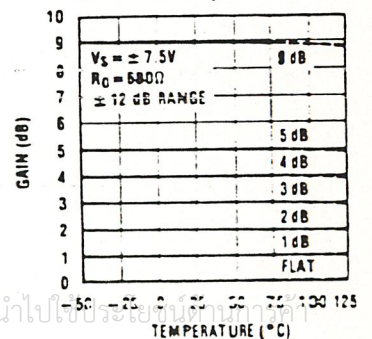
Gain vs Frequency @ ±6 dB Range (Boost)



Gain vs Frequency @ ±6 dB Range (Cut)



Gain vs Temperature



Typical Applications

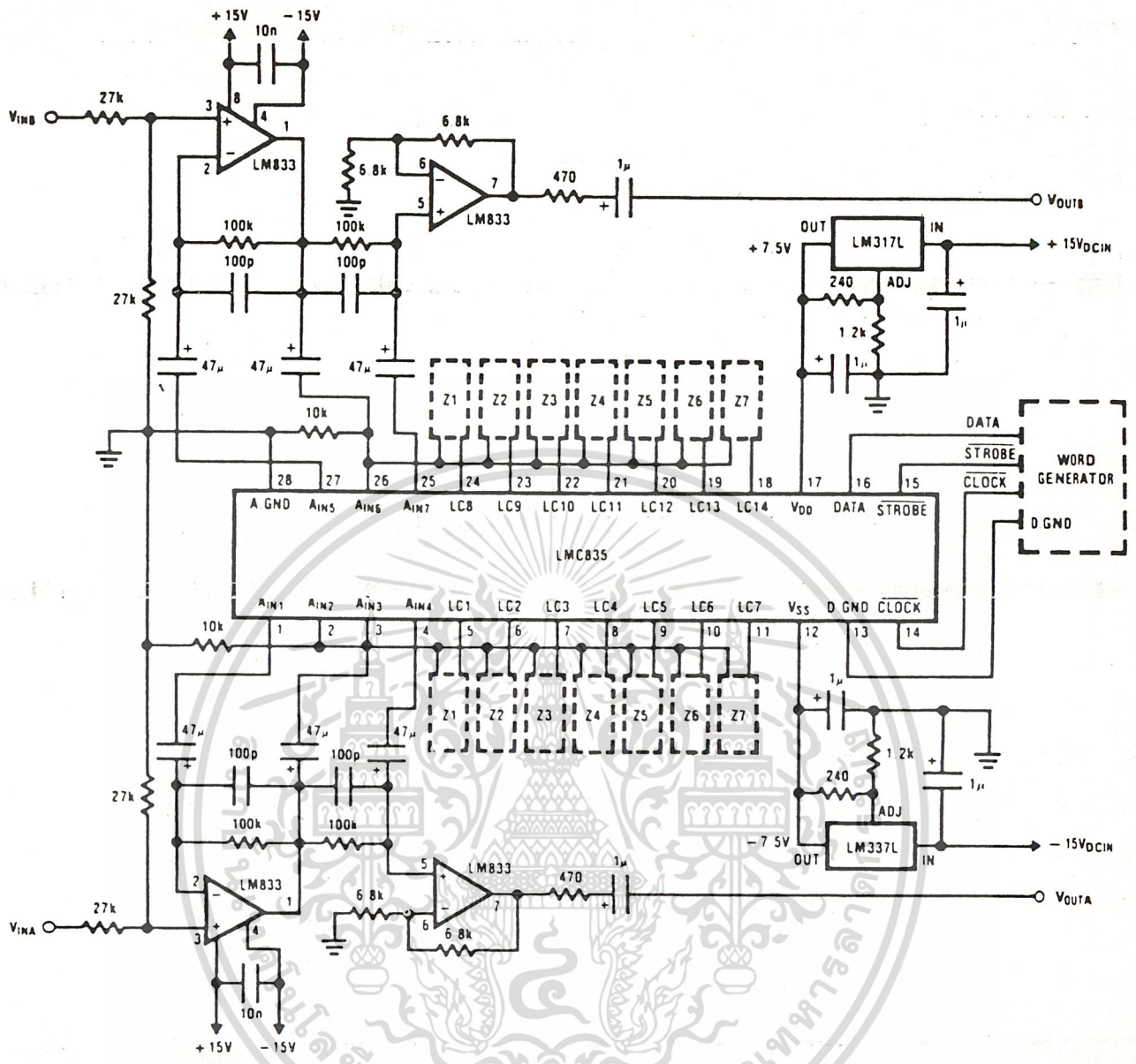
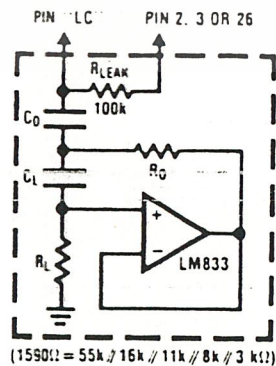


FIGURE 7. Stereo 7-Band Equalizer.

TL/H/6753-11

TABLE I: Tuned Circuit Elements

$Q_0 = 3.5, Q_{12dB} = 1.05$					
Z1	f_o (Hz)	C_o (F)	C_L (F)	R_L (Ω)	R_o (Ω)
Z1	63	1μ	0.1μ	100k	680
Z2	160	0.47μ	0.033μ	100k	580
Z3	400	0.15μ	0.015μ	100k	680
Z4	1k	0.068μ	0.0068μ	82k	680
Z5	2.5k	0.022μ	0.0033μ	82k	680
Z6	6.3k	0.01μ	0.0015μ	62k	680
Z7	16k	0.0047μ	680p	47k	680



$$L_0 = C_L R_L R_0$$

$$f_o = \frac{1}{2\pi\sqrt{L_0 C_0}}$$

$$Q_0 = \sqrt{\frac{L_0}{C_0 R_0^2}}$$

$$Q_{12dB} = \frac{R_0 Q_0}{R_0 + 1590}$$

(1590 Ω = 55k // 16k // 11k // 8k // 3 k Ω)

TL/H/6753-12

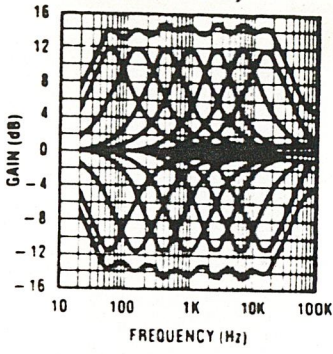
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้ง
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 8. Tuned Circuit for Stereo 7-Band Equalizer (Figure 7)

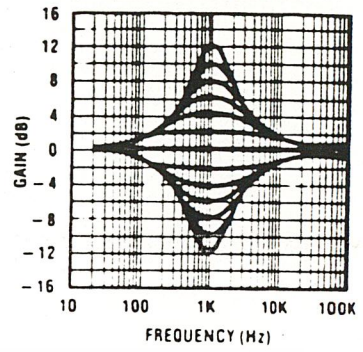
Typical Applications (Continued)

Performance Characteristics (Circuit of Figure 7)

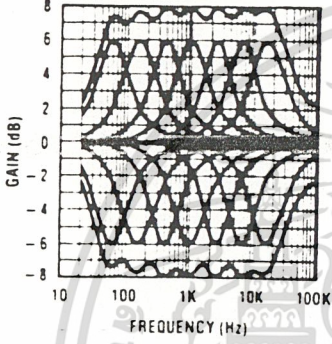
**LMC835 Gain vs Frequency
@ ± 12 dB Range
(All Boost or Cut)**



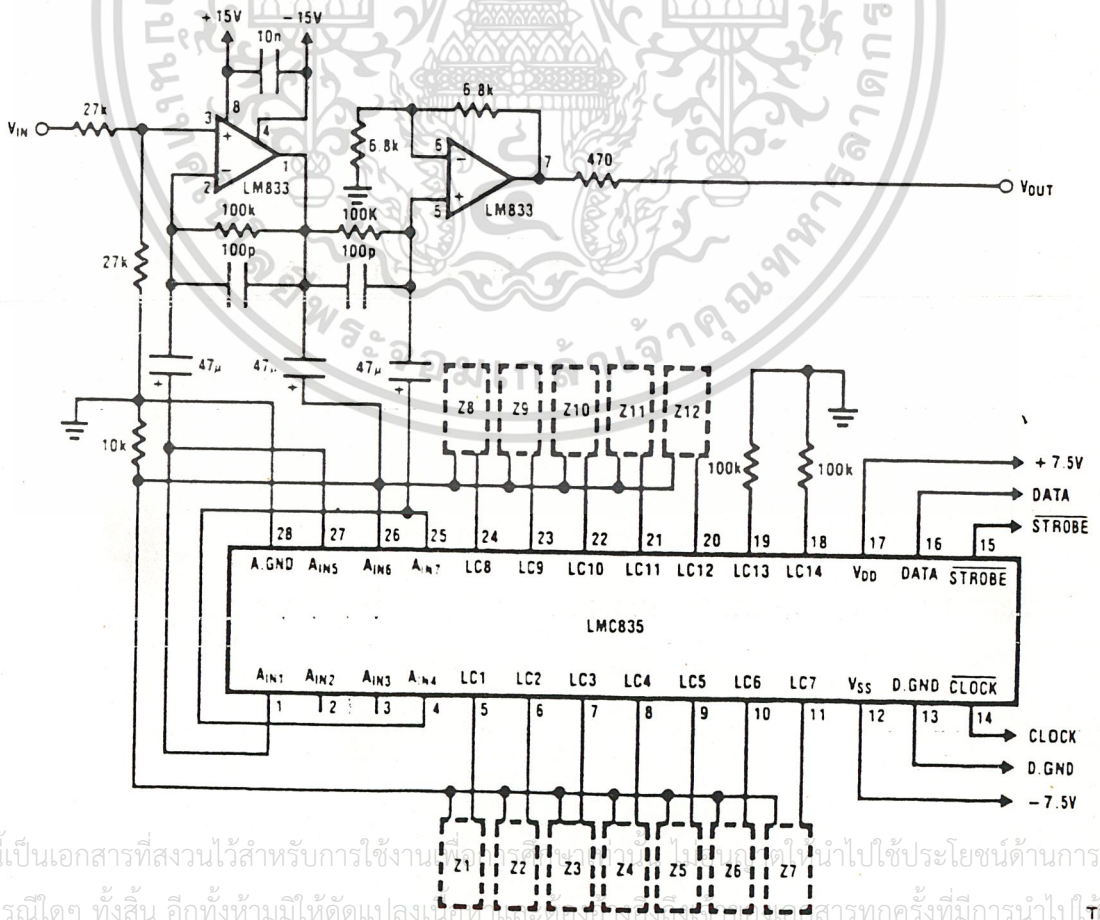
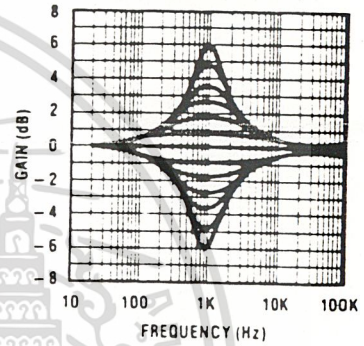
**LMC835 Gain vs Frequency
@ ± 12 dB Range
(1 kHz Boost or Cut)**



**LMC835 Gain vs Frequency
@ ± 6 dB Range
(All Boost or Cut)**



**LMC835 Gain vs Frequency
@ ± 6 dB Range
(1 kHz Boost or Cut)**



TL/H/6753-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในโครงการวิจัยเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารทุกครั้งที่มีการนำไปใช้

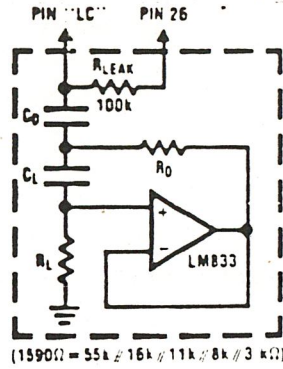
FIGURE 9. 12-Band Equalizer

TL/H/6753-14

Typical Applications (Continued)

TABLE II. Tuned Circuit Elements

$Q_0 = 4.7, Q_{12\text{ dB}} = 1.4$					
	f_0 (Hz)	C_0 (F)	C_L (F)	R_L (Ω)	R_0 (Ω)
Z1	16	3.3μ	0.47μ	100k	680
Z2	31.5	15μ	0.22μ	110k	680
Z3	63	1μ	0.1μ	100k	680
Z4	125	0.39μ	0.068μ	91k	680
Z5	250	0.22μ	0.033μ	82k	680
Z6	500	0.1μ	0.015μ	100k	680
Z7	1k	0.047μ	0.01μ	82k	680
Z8	2k	0.022μ	0.0047μ	91k	680
Z9	4k	0.01μ	0.0022μ	110k	680
Z10	8k	0.0068μ	0.001μ	82k	680
Z11	16k	0.0033μ	680p	62k	680
Z12	32k	0.0015μ	470p	68k	510



$$L_0 = C_L R_L R_0$$

$$f_0 = \frac{1}{2\pi\sqrt{L_0 C_0}}$$

$$Q_0 = \sqrt{\frac{L_0}{C_0 R_0^2}}$$

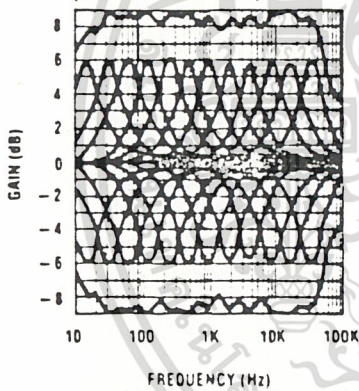
$$Q_{12\text{ dB}} = \frac{R_0 Q_0}{R_0 + 1590}$$

TL/H/6753-15

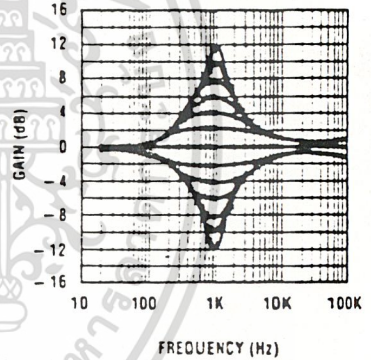
FIGURE 10. Tuned Circuit for 12-Band Equalizer (Figure 9)

Performance Characteristics (Circuit of Figure 9)

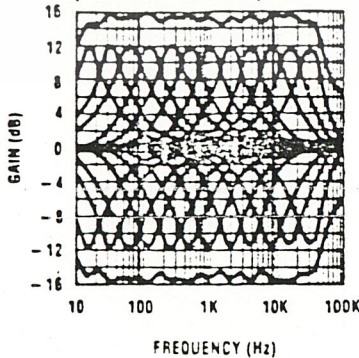
12 Band Equalizer Application
LMC835 Gain vs Frequency
@ ± 6 dB Range
(All Boost or Cut)



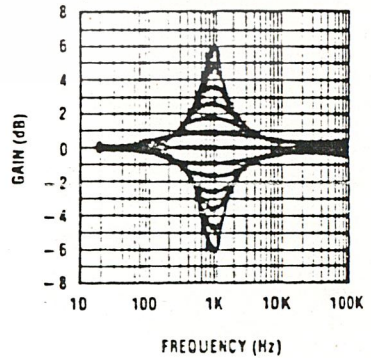
LMC835 12 Band E.Q. Application
Gain vs Frequency
@ ± 12 dB Range
(1 kHz Boost or Cut)



12 Band Equalizer Application
LMC835 Gain vs Frequency
@ ± 12 dB Range
(All Boost or Cut)



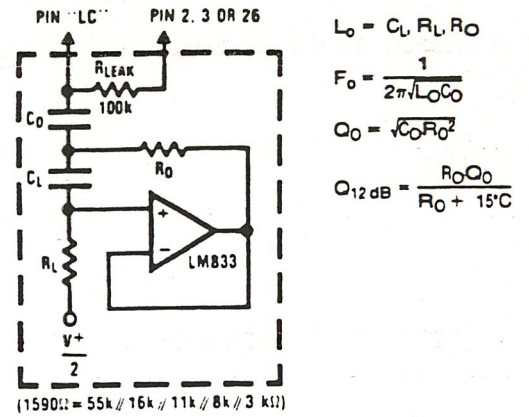
LMC835 12 Band E.Q. Application
Gain vs Frequency
@ ± 6 dB Range
(1 kHz Boost or Cut)



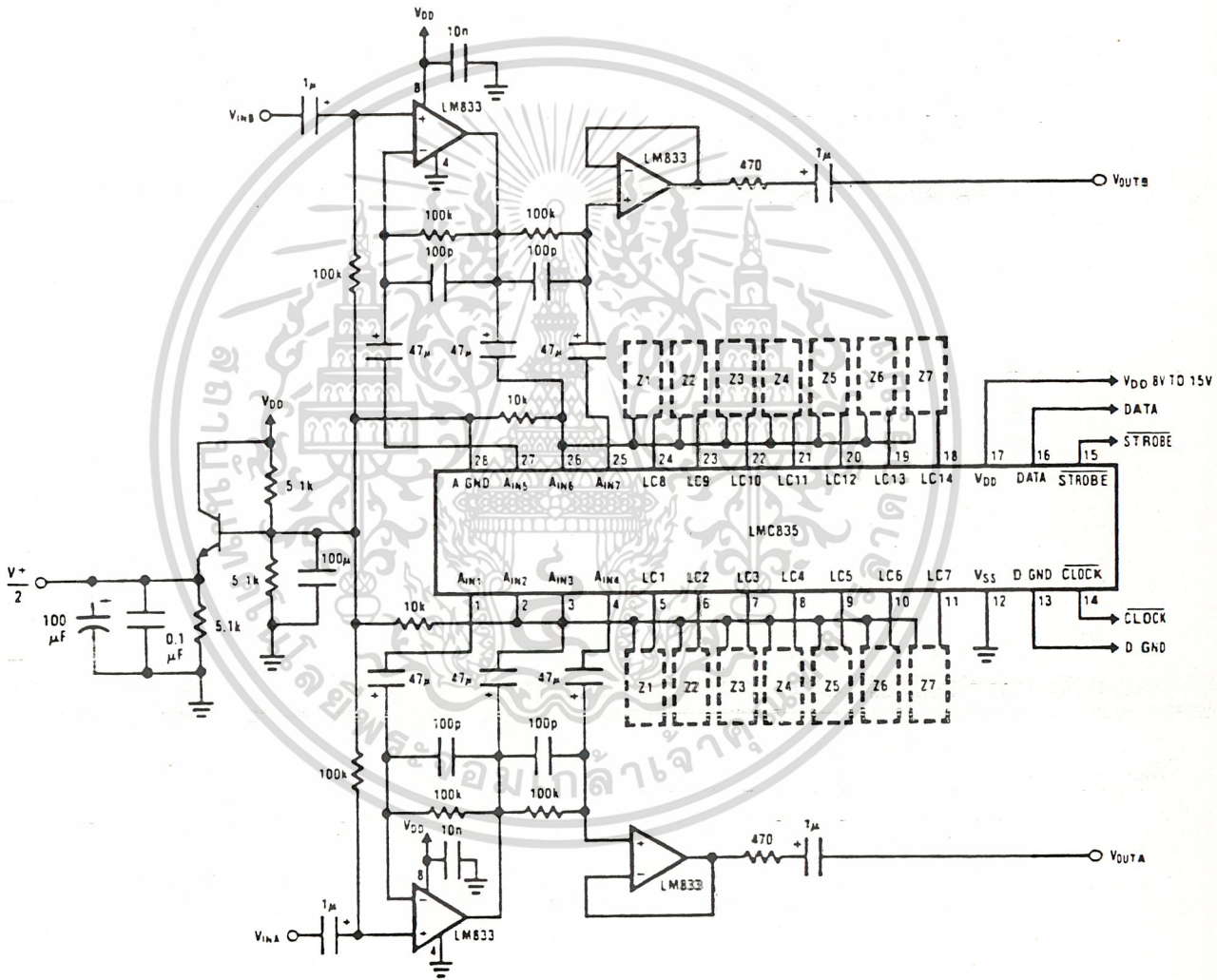
TL/H/6753-16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)



TL/H/6753-25



TL/H/6753-1

The $\frac{V^-}{2}$ output is used to bias the gyrators

FIGURE 11. Single Supply Stereo Equalizer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

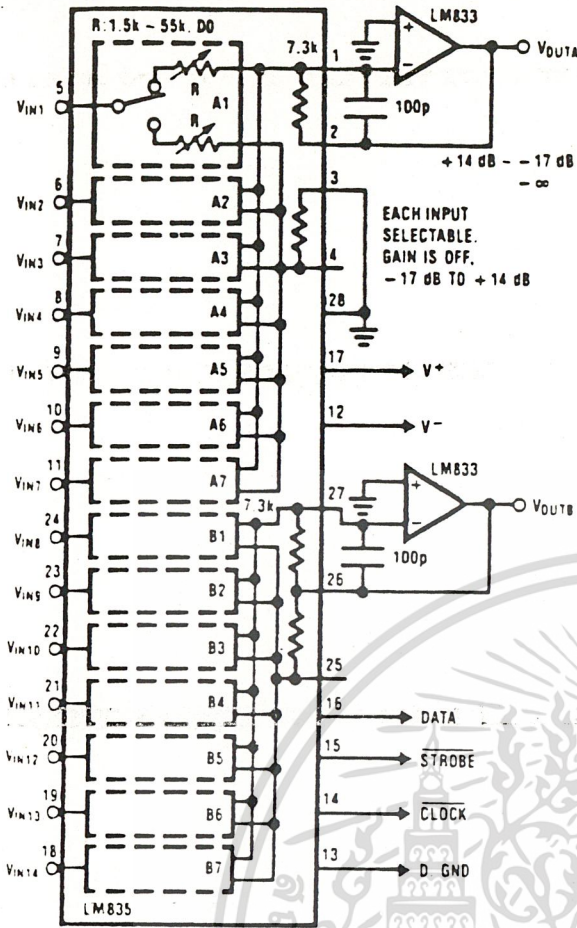


FIGURE 12. Stereo 7-Input/1-Output Mixers (THD is not as low as equalizer circuit)

TL/H/6753-18

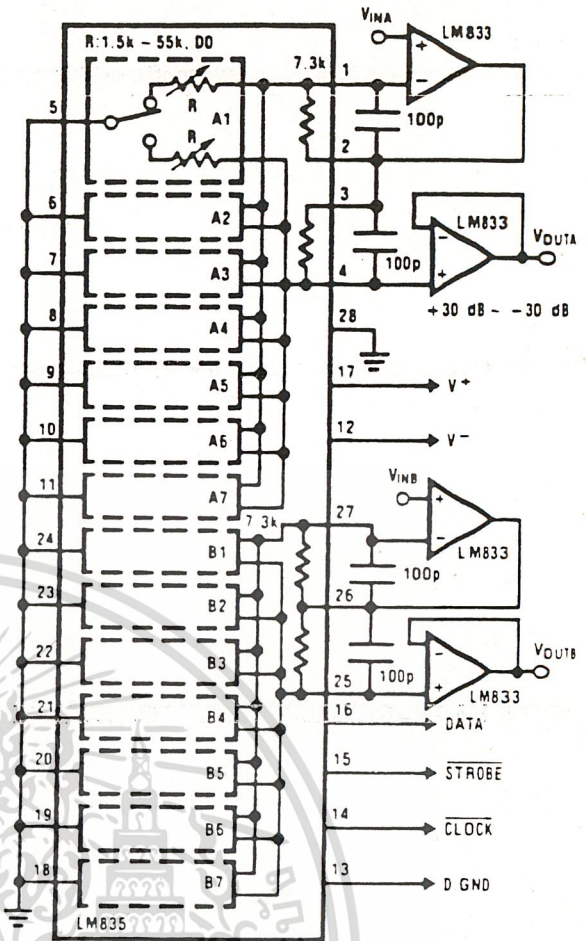


FIGURE 13. Stereo Volume Control, Very Low THD

TL/H/6753-19

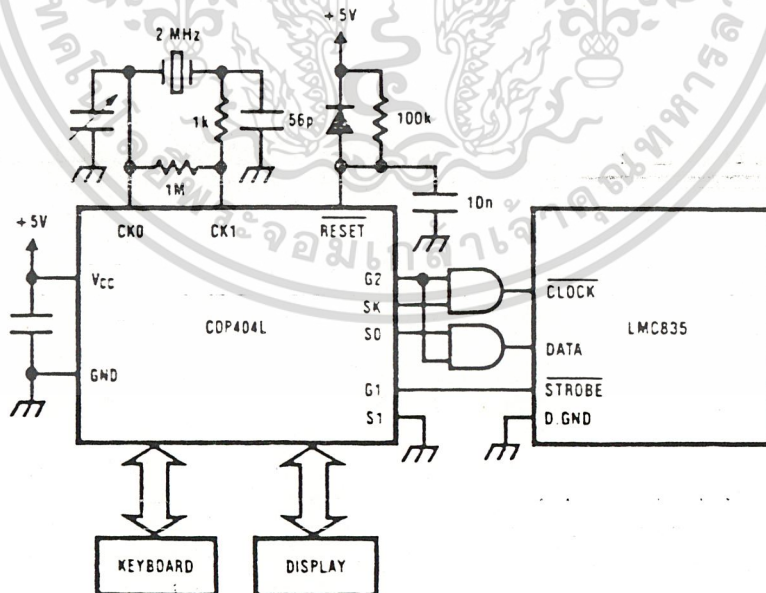


FIGURE 14. LMC835-COP404L CPU Interface

TL/H/6753-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

Sample Subroutine Program for Figure 14, LMC835-COP404L CPU Interface

HEX CODE	LABEL	MNEMONICS	COMMENTS
3F	LMC835:	LBI 3F	;POINT TO RAMADDRESS 3F
05	SEND	LD	;RAMDATA TO A
22		SC	; SET CARRY
335F		OGI	;SET PORT G= 1111, OPEN THE AND GATES
4F		XAS	;SWAP A AND SIO, CLOCK START
05		LD	;RAMDATA TO A, MAKE SURE A = DATA
07		XDS	;SWAP A AND RAMDATA, RAMADDRESS=RAMADDRESS-1
05		LD	;RAMDATA TO A
4F		XAS	;SWAP A AND SIO
05		LD	;RAMDATA TO A, MAKE SURE A=NEWDATA
07		XDS	;SWAP A AND RAMDATA, RAMADDRESS=RAMADDRESS-1
32		RC	;RESET CARRY
4F		XAS	;SWAP A AND SIO, CLOCK STOP
335D		OGJ 13	;SET PORT G=1101, MAKE STROBE LOW
335B		OGI 11	;SET PORT G=1011, MAKE STROBE HIGH, CLOSE THE GATES
4E		CBA	;BD TO A
43		AISC 3	;RAMADDRESS < 3C THEN RETURN
48		RET	
80		JP SEND	

RAM ADDRESS	DATA	COMMENTS
3C	DATA	;GAIN DATA D4-D7
3D	DATA	;GAIN DATA D0-D3
3E	DATA	;BAND DATA D4-D7
3F	DATA	;BAND DATA D0-D3

Application Hints

SWITCHING NOISE

The LMC835 uses CMOS analog switches that have small leakages (less than 50 nA). When a band is selected for flat gain, all the switches in that band are open and the resonator circuit is not connected to the LMC835 resistor network. It is only in the flat mode that the small leakage currents can cause problems. The input to the resonator circuit is usually a capacitor and the leakage currents will slowly charge up this capacitor to a large voltage if there is no resistive path to limit it. When the band is set to any value other than flat, the charge on the capacitor will be discharged by the resistor network and there will be a transient at the output. To limit the size of this transient, R_{LEAK} is necessary.

HOW TO AVOID SWITCHING NOISE DUE TO LEAKAGE CURRENT (Refer to Figures 7 and 8)

To avoid switching noise due to leakage currents when changing the gain, it is recommended to put $R_{LEAK} = 100$ k Ω between Pin 3 and Pin 5—11 each, Pin 26 and Pin 12—24 each. The resistor limits the voltage that the capacitor can charge to, with minimal effects on the equalization. The frequency response change due to R_{LEAK} are shown in Figure 15. The gain error is only 0.2 dB and Q error is only 5% at 12 dB boost or cut.

SIMPLE WORD GENERATOR (Figure 6)

Circuit operation revolves around an MM74HC165 parallel-in/serial-out shift register. Data bits D0 through D7 are applied to the parallel of the MM74HC165 from 8 toggle switches. The bits are shifted out to the DATA input of the LMC835 in sync with the clock. When all data bits have been loaded, CLOCK is inhibited and a STROBE pulse is generated: this sequence is initiated by a START pulse.

LMC835-COP404L CPU INTERFACE (Refer to Figure 14)

The diagram shows AND gates between the COP and the LMC835. These permit G2 to inhibit the CLOCK and DATA lines (SK and SO) during a STROBE (G1) pulse. This function may also be implemented in software. As shown in Figure 2, the data groups are shifted in D0 first. Data is loaded on positive clock edges.

POWER SUPPLIES

These applications show LM317/337 regulators for the ± 7.5 V supplies for the LMC835. Since the latter draws only 5 mA max., 1k series dropping resistors from the ± 15 V op amp supply and a pair of 7.5V zeners and bypass caps will also suffice.



DS75160A/DS75161A/DS75162A IEEE-488 GPIB Transceivers

General Description

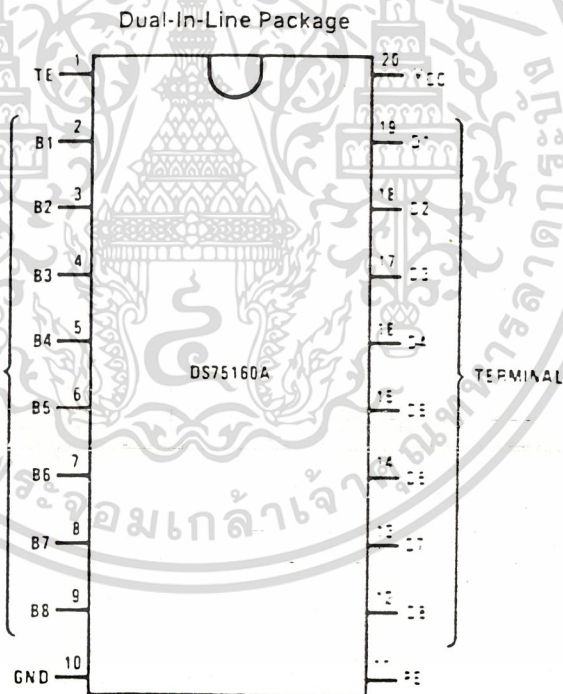
This family of high-speed-Schottky 8-channel bi-directional transceivers is designed to interface TTL/MOS logic to the IEEE Standard 488-1978 General Purpose Interface Bus (GPIB). PNP inputs are used at all driver inputs for minimum loading, and hysteresis is provided at all receiver inputs for added noise margin. The IEEE-488 required bus termination is provided internally with an active turn-off feature which disconnects the termination from the bus when V_{CC} is removed.

The General Purpose Interface Bus is comprised of 16 signal lines — 8 for data and 8 for interface management. The data lines are always implemented with DS75160A, and the management lines are either implemented with DS75161A in a single-controller system, or with DS75162A in a multi-controller system.

Features

- 8-channel bi-directional non-inverting transceivers
- Bi-directional control implemented with TRI-STATE[®] output design
- Meets IEEE Standard 488-1978
- High-speed Schottky design
- Low power consumption
- High impedance PNP inputs (drivers)
- 500 mV (typ) input hysteresis (receivers)
- On-chip bus terminators
- No bus loading when V_{CC} is removed
- Pin selectable open collector mode on DS75160A driver outputs
- Accommodates multi-controller systems

Connection Diagrams



Top View

Order Number DS75160AN or DS75161CAWM
See NS Package Number M20B or N20A

FIG. 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, V_{CC}	7.0V
Input Voltage	5.5V
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering, 4 sec.)	260°C
Maximum Power Dissipation* at 25°C	
Molded Package	1897 mW

*Derate molded package 15.2 mW/°C above 25°C.

Operating Conditions

V_{CC} , Supply Voltage	Min 4.75	Max 5.25	Units V
T_A , Ambient Temperature	0	70	°C
I_{OL} , Output Low Current			
Bus		48	mA
Terminal		16	mA

Electrical Characteristics (Notes 2 and 3)

Symbol	Parameter		Conditions	Min	Typ	Max	Units	
V_{IH}	High-Level Input Voltage			2			V	
V_{IL}	Low-Level Input Voltage					0.8	V	
V_{IK}	Input Clamp Voltage		$I_I = -18$ mA		-0.8	-1.5	V	
V_{HYS}	Input Hysteresis	Bus		400	500		mV	
		Terminal						
V_{OH}	High-Level Output Voltage	Terminal	$I_{OH} = -800$ μ A	2.7	3.5		V	
		Bus (Note 5)	$I_{OH} = -5.2$ mA	2.5	3.4			
V_{OL}	Low-Level Output Voltage	Terminal	$I_{OL} = 16$ mA		0.3	0.5	V	
		Bus	$I_{OL} = 48$ mA		0.4	0.5		
I_{IH}	High-Level Input Current	Terminal and TE, PE, DC, SC Inputs	$V_I = 5.5$ V		0.2	100	μ A	
			$V_I = 2.7$ V		0.1	20		
I_{IL}	Low-Level Input Current		$V_I = 0.5$ V		-10	-100	μ A	
V_{BIAS}	Terminator Bias Voltage at Bus Port	Driver Disabled	$I_{I(bus)} = 0$ (No Load)	2.5	3.0	3.7	V	
I_{LOAD}	Terminator Bus Loading Current	Bus	Driver Disabled	$V_{I(bus)} = -1.5$ V to 0.4 V	-1.3			mA
			Driver Disabled	$V_{I(bus)} = 0.4$ V to 2.5 V	0		-3.2	
			Driver Disabled	$V_{I(bus)} = 2.5$ V to 3.7 V			2.5	
			Driver Disabled	$V_{I(bus)} = 3.7$ V to 5 V	0		2.5	
			Driver Disabled	$V_{I(bus)} = 5$ V to 5.5 V	0.7		2.5	
			$V_{CC} = 0$ V, $V_{I(bus)} = 0$ V to 2.5 V			40	μ A	
I_{OS}	Short-Circuit Output Current	Terminal	$V_I = 2$ V, $V_O = 0$ V (Note 4)	-15	-35	-75	mA	
		Bus (Note 5)		-35	-75	-150		
I_{CC}	Supply Current	DS75160A	Transmit, TE = 2V, PE = 2V, $V_I = 0.8$ V		85	125	mA	
			Receive, TE = 0.8V, PE = 2V, $V_I = 0.8$ V		70	100		
		DS75161A	TE = 0.8V, DC = 0.8V, $V_I = 0.8$ V		84	125		
		DS75162A	TE = 0.8V, DC = 0.8V, SC = 2V, $V_I = 0.8$ V		85	125		
C_{IN}	Bus-Port Capacitance	Bus	$V_{CC} = 5$ V or 0V, $V_I = 0$ V to 2V, $f = 1$ MHz		20	30	pF	

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min/max limits apply across the 0°C to +70°C temperature range and the 4.75V to 5.25V power supply range. All values are for $T_A = 25^\circ\text{C}$ and $V_{CC} = 5.0$ V.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground, unless otherwise specified. All values shown as max or min are so classified on absolute value basis.

Note 4: Only one output at a time should be shorted.

Note 5: This characteristic does not apply to outputs on DS75161A and DS75162A that are open collector.

Switching Characteristics $V_{CC} = 5.0V \pm 5\%$, $T_A = 0^\circ C$ to $70^\circ C$ (Note 1)

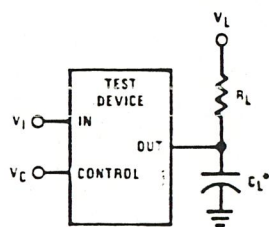
Symbol	Parameter	From	To	Conditions	DS75160A			DS75161A			DS75162A			Units
					Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
t _{PH}	Propagation Delay Time, Low to High Level Output	Terminal	Bus	$V_L = 2.3V$ $R_L = 38.3\Omega$ $C_L = 30 pF$ <i>Figure 1</i>		10	20		10	20		10	20	ns
t _{PL}	Propagation Delay Time, High to Low Level Output					14	20		14	20		14	20	ns
t _{PH}	Propagation Delay Time, Low to High Level Output	Bus	Terminal	$V_L = 5.0V$ $R_L = 240\Omega$ $C_L = 30 pF$ <i>Figure 2</i>		14	20		14	20		14	20	ns
t _{PL}	Propagation Delay Time, High to Low Level Output					10	20		10	20		10	20	ns
t _{EH}	Output Enable Time to High Level	TE, DC, or SC	Bus	$V_I = 3.0V$ $V_L = 0V$ $R_L = 480\Omega$ $C_L = 15 pF$ <i>Figure 1</i>		19	32		23	40		23	40	ns
t _{ED}	Output Disable Time From High Level					15	22		15	25		15	25	ns
t _{EL}	Output Enable Time to Low Level	(Note 2) (Note 3)	Bus	$V_I = 0V$ $V_L = 2.3V$ $R_L = 38.3\Omega$ $C_L = 15 pF$ <i>Figure 1</i>		24	35		28	48		28	48	ns
t _{EL}	Output Disable Time From Low Level				17	25		17	27		17	27	ns	
t _{EH}	Output Enable Time to High Level	TE, DC, or SC	Terminal	$V_I = 3.0V$ $V_L = 0V$ $R_L = 3 k\Omega$ $C_L = 15 pF$ <i>Figure 1</i>		17	33		18	40		18	40	ns
t _{ED}	Output Disable Time From High Level					15	25		22	33		22	33	ns
t _{EL}	Output Enable Time to Low Level	(Note 2) (Note 3)	Terminal	$V_I = 0V$ $V_L = 5V$ $R_L = 280\Omega$ $C_L = 15 pF$ <i>Figure 1</i>		25	39		28	52		28	52	ns
t _{EL}	Output Disable Time From Low Level				15	27		20	35		20	35	ns	
t _{EH}	Output Pull-Up Enable Time (DS75160A Only)	PE	Bus	$V_I = 3V$ $V_L = 0V$ $R_L = 480\Omega$ $C_L = 15 pF$ <i>Figure 1</i>		10	17		NA			NA		ns
t _{ED}	Output Pull-Up Disable Time (DS75160A Only)	(Note 2)				10	15		NA		NA		ns	

Note 1: Typical values are for $V_{CC} = 5.0V$ and $T_A = 25^\circ C$ and are meant for reference only.

Note 2: Refer to Functional Truth Tables for control input definition.

Note 3: Test configuration should be connected to only one transceiver at a time due to the high current stress caused by the V_I voltage source when the output is connected to that input becomes active.

Switching Load Configurations

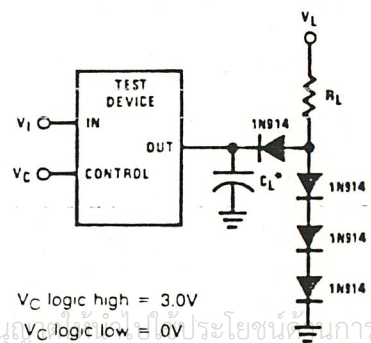


V_C logic high = 3.0V

V_C logic low = 0V

FIGURE 1

TL/F/5804-B



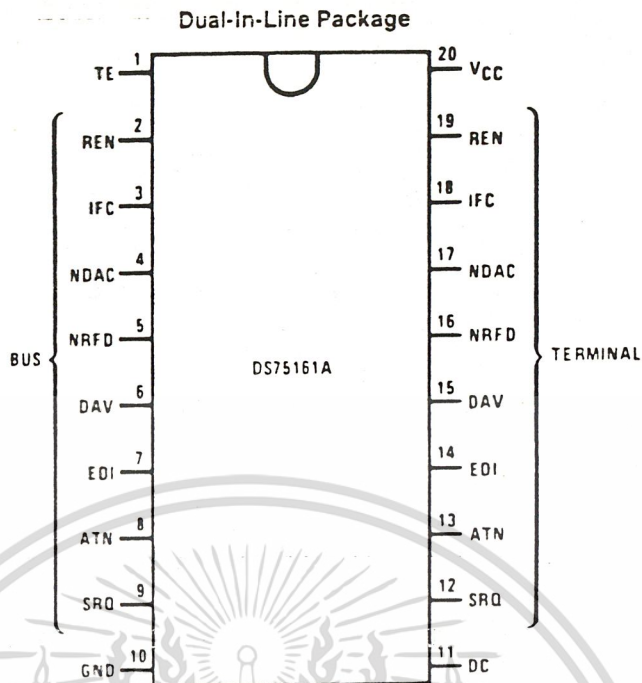
V_C logic high = 3.0V

V_C logic low = 0V

FIGURE 2

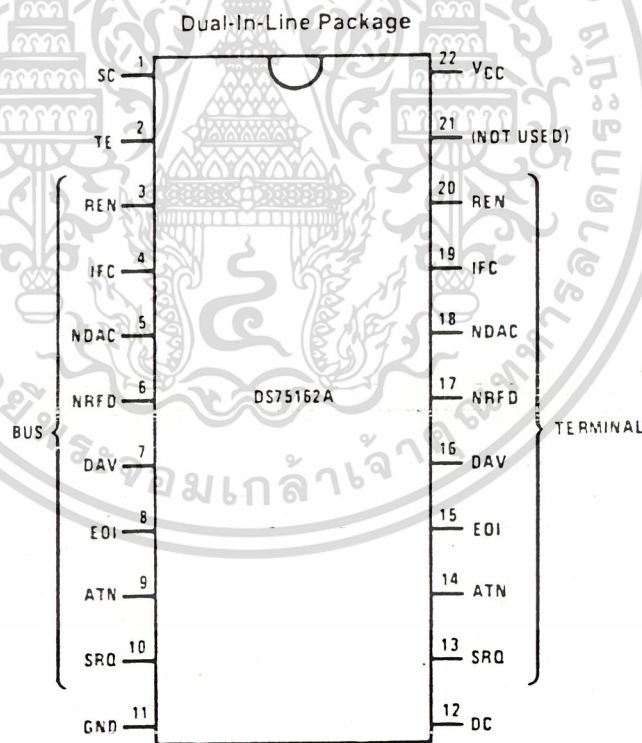
TL/F/5804-9

Connection Diagrams (Continued)



TL/F/5604-2

Top View



TL/F/5604-3

Top View

Order Number DS75161AN, DS75162AN or DS75161AWM
See NS Package Number M20B, N20A or N22A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

DS75160A

This device is an 8-channel bi-directional transceiver with one common direction control input, denoted TE. When used to implement the IEEE-488 bus, this device is connected to the eight data bus lines, designated DIO₁-DIO₈. The port connections to the bus lines have internal terminators, in accordance with the IEEE-488 Standard, that are deactivated when the device is powered down. This feature guarantees no bus loading when V_{CC} = 0V. The bus port outputs also have a control mode that either enables or disables the active upper stage of the totem-pole configuration. When this control input, denoted PE, is in the high state, the bus outputs operate in the high-speed totem-pole mode. When PE is in the low state, the bus outputs operate as open collector outputs which are necessary for parallel polling.

DS75161A

This device is also an 8-channel bi-directional transceiver which is specifically configured to implement the eight management signal lines of the IEEE-488 bus. This device, paired with the DS75160A, forms the complete 16-line interface between the IEEE-488 bus and a single controller instrumentation system. In compliance with the system organization of the management signal lines, the SRQ, NDAC, and NRFD bus port outputs are open collector. In contrast to the DS75160A, these open collector outputs are a fixed configuration. The direction control is divided into three groups. The DAV, NDAC, and NRFD transceiver directions are controlled by the TE input. The ATN, SRQ, REN, and IFC transceiver directions are controlled by the DC input. The EOI transceiver direction is a function of both the TE and DC inputs, as well as the logic level present on the ATN channel. The port connections to the bus lines have internal terminators identical to the DS75160A.

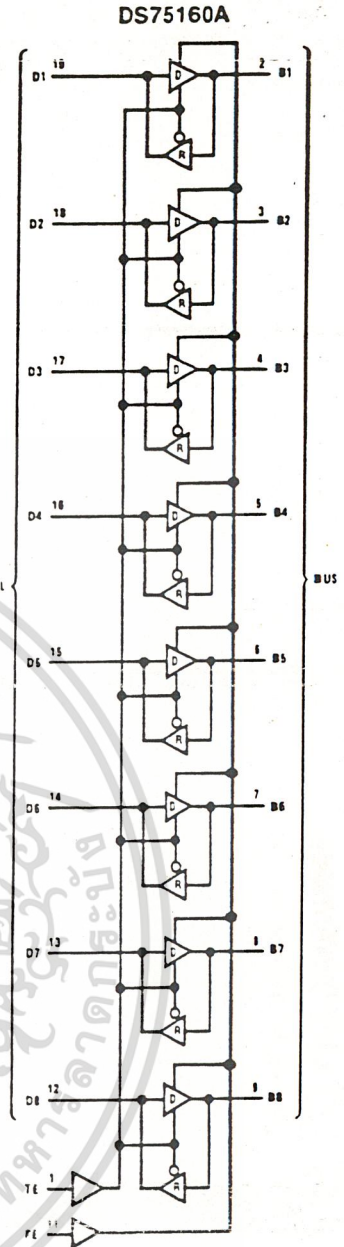
DS75162A

This device is identical to the DS75161A, except that an additional direction control input is provided, denoted SC. The SC input controls the direction of the REN and IFC transceivers that are normally controlled by the DC input on the DS75161A. This additional control function is instrumental in implementing multiple controller systems.

Table of Signal Line Abbreviations

Signal Line Classification	Mnemonic	Definition	Device
Control Signals	DC	Direction Control	DS75161A/ DS75162A
	PE	Pull-Up Enable	DS75160A
	TE	Talk Enable	All
	SC	System Controller	DS75162A
Data I/O Ports	B1-B8	Bus Side of Device	DS75160A
	D1-D8	Terminal Side of Device	
Management Signals	ATN	Attention	DS75161A/ DS75162A
	DAV	Data Valid	
	EOI	End or Identify	
	IFC	Interface Clear	
	NDAC	Not Data Accepted	
	NRFD	Not Ready for Data	
	REN	Remote Enable	
	SRQ	Service Request	

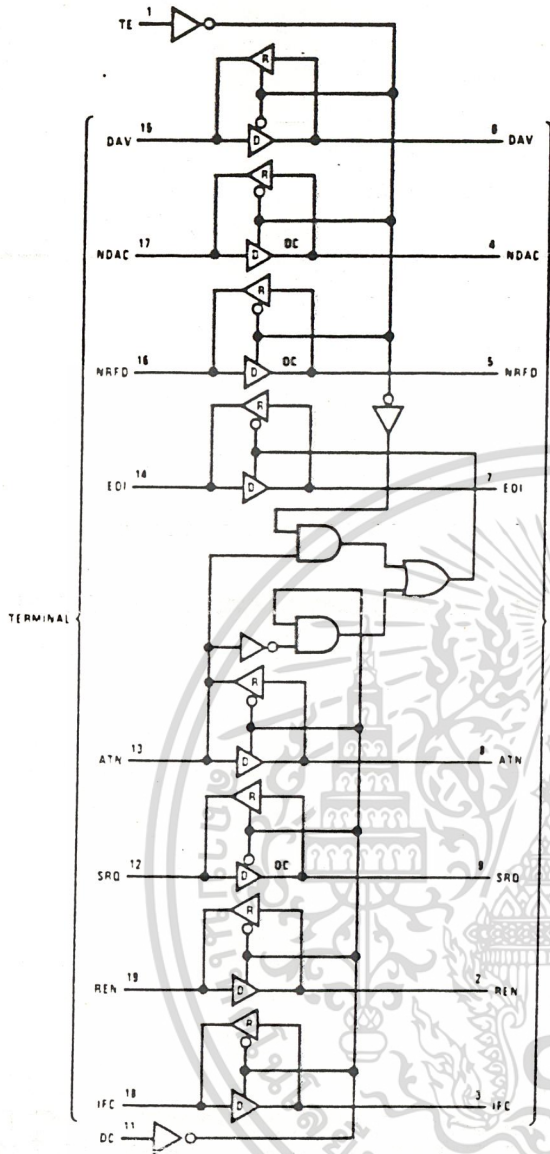
Logic Diagrams



- Note 1: Denotes driver
- Note 2: Denotes receiver
- Note 3: Driver and receiver outputs are totem-pole configurations
- Note 4: The driver outputs of DS75160A can have their active pull-ups disabled by switching the PE input (pin 11) to the logic low state. This mode configures the outputs as open collector.

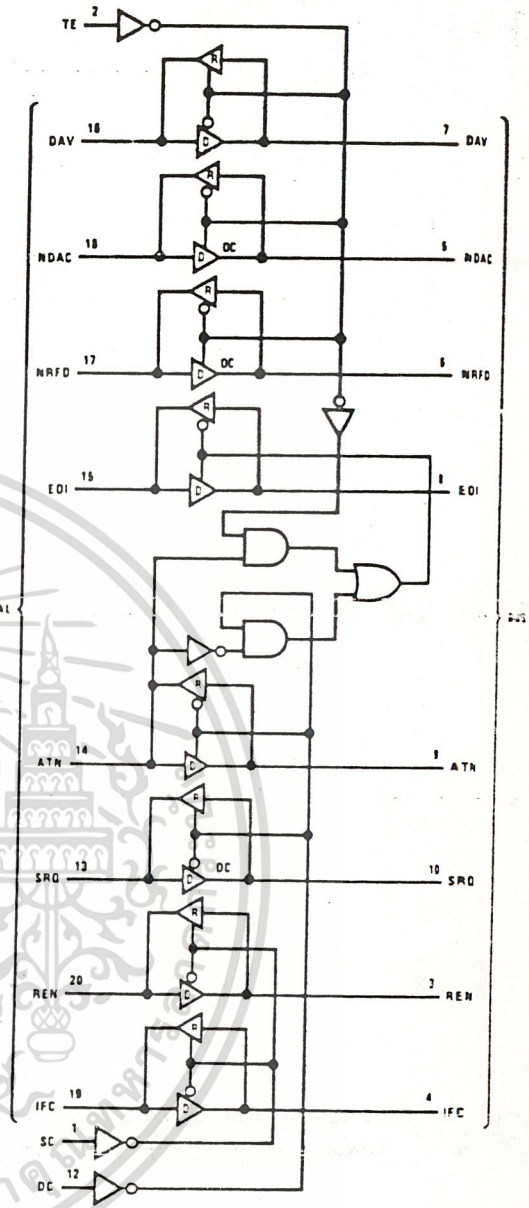
Logic Diagrams (Continued)

DS75161A




TL/F 5804-5

DS75162A



TL/F 5804-6

Note 1:  Denotes driver

Note 2:  Denotes receiver

Note 3: Symbol "OC" specifies open collector output

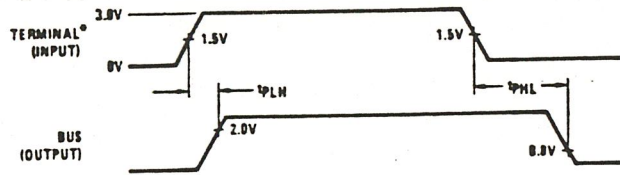
Note 4: Driver and receiver outputs that are not specified "OC" are totem-pole configurations

TL/F/5804-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

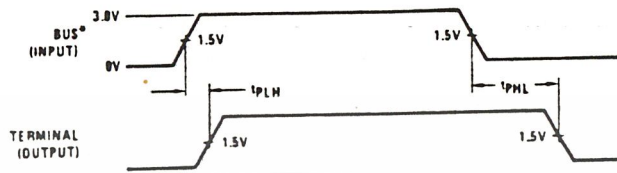
Switching Waveforms

Transmit Propagation Delays



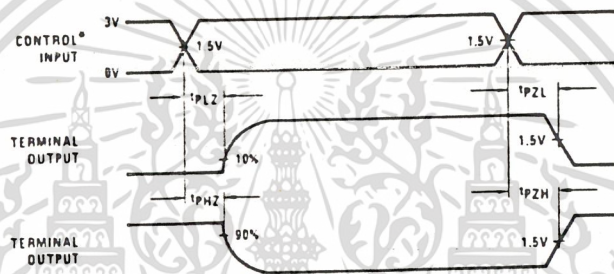
TL/F/5804-10

Receive Propagation Delays



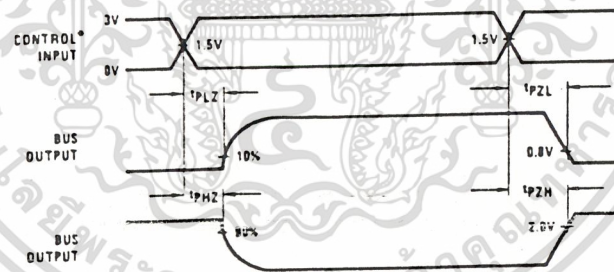
TL/F/5804-11

Terminal Enable/Disable Times



TL/F/5804-12

Bus Enable/Disable Times

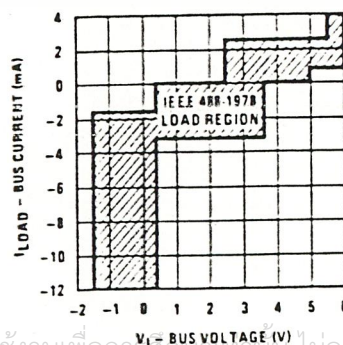


TL/F/5804-13

*Input signal: $f = 1.0$ MHz, 50% duty cycle, $t_r = t_f = 5$ ns

Performance Characteristics

Bus Port Load Characteristics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Refer to Electrical Characteristics table.

TL/F/5804-14

Functional Truth Tables

Functional Tr...

DS75160A

Control Input Level		Data Transceivers	
TE	PE	Direction	Bus Port Configuration
H	H	T	Totem-Pole Output
H	L	T	Open Collector Output
L	X	R	Input

DS75161A

Control Input Level			Transceiver Signal Direction							
TE	DC	ATN*	EOI	REN	IFC	SRQ	NRFD	NDAC	DAV	
H	H		R		R	R	T	R	R	T
H	L		T		R	T	R	R	R	T
L	H		R		R	R	T	T	T	R
L	L		T		T	T	R	T	T	R
H	X	H	T							
L	X	H	R							
X	H	L	R							
X	L	L	T							

DS75162A

Control Input Level				Transceiver Signal Direction							
SC	TE	DC	ATN*	EOI	REN	IFC	SRQ	NRFD	NDAC	DAV	
H	H	H		R		T	T	T	R	R	T
H	H	L		T		T	T	R	R	R	T
H	L	H		R		T	T	T	T	T	R
H	L	L		T		T	T	R	T	T	R
L	H	H		R		R	R	R	R	R	T
L	H	L		T		R	R	R	R	R	T
L	L	H		R		R	R	T	T	T	R
L	L	L		T		R	R	R	T	T	R
X	H	X	H	T							
X	L	X	H	R							
X	X	H	L	R							
X	X	L	L	T							

H = High level input

L = Low level input

X = Don't care

T = Transmit, i.e., signal outputted to bus

R = Receive, i.e., signal outputted to terminal

*The ATN signal level is sensed for internal multiplex control of EOI transmission direction logic.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MOTOROLA

MCM2833

Product Preview

32K-BIT ELECTRICALLY ERASABLE PROM

The MCM2833 is a 32,768-bit Electrically Erasable Programmable Read Only Memory (EEPROM) designed for handling data in applications requiring both nonvolatile memory and in-system reprogramming.

The MCM2833 saves time and money because of the in-system erase and reprogram capability. The device operates from a single +5 V power supply in the read, write, and erase mode. Word erase and write can be controlled entirely by TTL signal levels.

To ease system design, the high voltage needed by the device for write and erase cycles is generated internally.

Another ease-of-use feature is the choice of erase modes (bulk, byte, row, or column) to optimize system erase/write time. For microprocessor compatibility, on-chip latches are provided for addresses, data, and controls, allowing the microprocessor to perform other tasks while the MCM2833 is erasing or programming (writing) itself.

The MCM2833 is fabricated using Motorola's FETMOS technology (Floating-gate Electron Tunneling MOS), which has the advantages of good data retention, good endurance, and conventional processing.

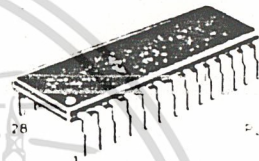
The device pinout is part of Motorola's industry standard byte-wide Nonvolatile Memory family, providing cost-effective density upgrades.

- Single +5 V Power Supply
- Organized as 4096 Bytes of 8 Bits
- Fast Access Time of 150 ns (MCM2833-15) and 200 ns (MCM2833-20)
- Low Power Dissipation
125 mA Maximum (Active)
35 mA Maximum (Standby)
- In-System Automatic Erase/Write Capability
- Data Protected During Power-Up and Power-Down
- 10,000 Erase/Program Cycles per Byte
- Data Integrity of 10 Years
- 9 ms for Byte Erase or Write
- Latched Address, Data, and Controls for Write/Erase
- Chip Enable and Output Enable for Two-Line Bus Control
- 28-Pin JEDEC Standard Pinout

HMOS

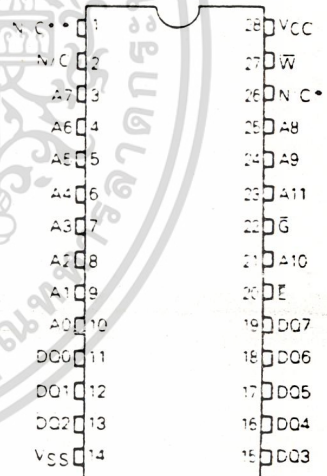
(IN CHANNEL, SILICON GATE)

4096 × 8-BIT
ELECTRICALLY ERASABLE
PROGRAMMABLE READ
ONLY MEMORY



P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENT

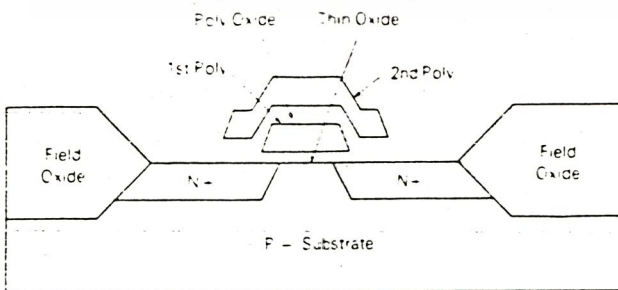


- * For normal operation, pin 26 can be tied to VSS or left floating.
- ** For normal operation, pin 1 can be tied to VSS or VCC.

PIN NAMES

PIN NAME	Function
A	Address
DQ	Data Input, Output
CE	Chip Enable
OE	Output Enable
W	Write Enable
N/C	No Connect
VCC	+5V Power Supply
VSS	Ground

FETMOS (Floating-Gate Electron Tunneling MOS)



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

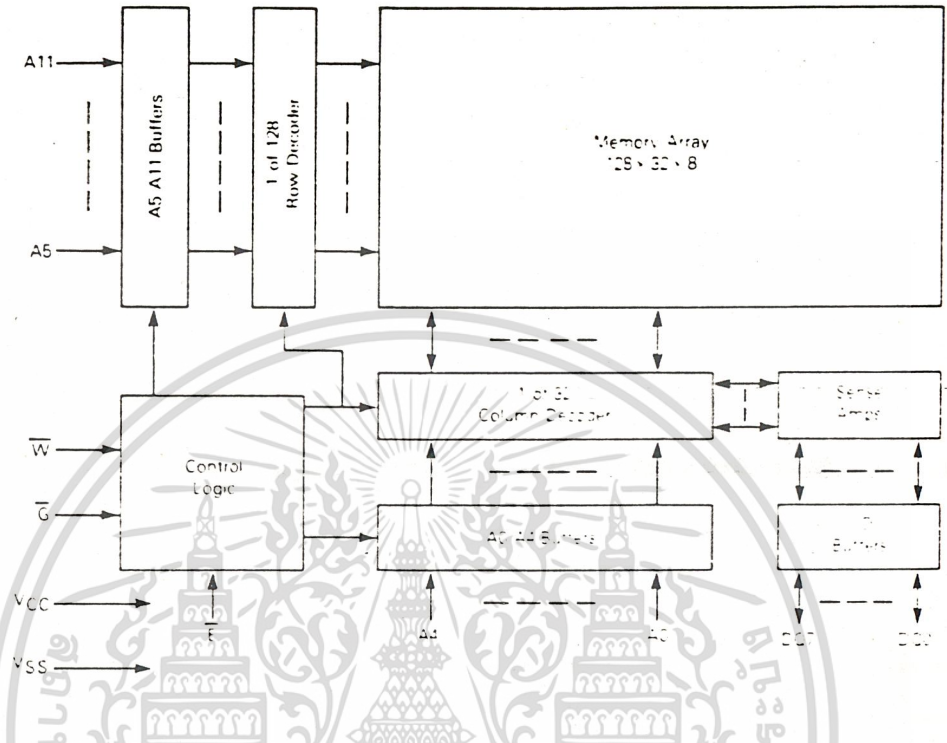
เอกสารนี้เป็นลิขสิทธิ์ของ Motorola และสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EEPROM

MCM2833

MCM2833 BLOCK DIAGRAM



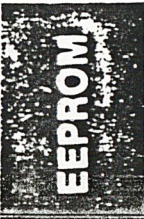
MODE SELECTION

Mode	Pin Number and Function				Notes
	Pin 11-13 15-19 DQ0-DC7	Pin 20 E	Pin 22 G	Pin 27 W	
Read	Data In	V _{CC}	V _{CC}	V _{CC}	-
Standby	High Z	V _{CC}	Don't Care	Don't Care	-
Output Disable	High Z	V _{CC}	V _{CC}	V _{CC}	-
Write	Data In	V _{CC}	V _{CC}	V _{CC}	-
Write or Erase Inhibit	High Z	V _{CC}	Don't Care	Don't Care	-
Word Erase (I)	V _{CC}	V _{CC}	V _{CC}	V _{CC}	-
Word Erase (II)	V _{CC}	V _{CC}	V _{CC}	V _{CC}	-
Page Erase (Row)	V _{CC}	V _{CC}	V _{CC}	V _{CC}	-
Page Erase (Column)	V _{CC}	V _{CC}	V _{CC}	V _{CC}	-
Bulk Erase	V _{CC}	V _{CC}	V _{CC}	V _{CC}	-

V_{IHH} = 11 V to 17 V

NOTES:

1. Row Page Erase Mode: There are 128 rows containing 32 bytes each. Individual bytes are selected with addresses A0-A4 (don't care).
2. Column Page Erase Mode: There are 32 columns containing 128 bytes each. Individual columns are selected with addresses A7-A11 (A0-A4 are don't care).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCM2833

ABSOLUTE MAXIMUM RATINGS (See Note)

Rating	Value	Unit
Temperature Under Bias	-10 to +80	°C
Operating Temperature Range	0 to +70	°C
Storage Temperature	-65 to +100	°C
Input or Output Voltages with Respect to V _{SS} (Except G, \bar{W})	+6 to -0.4	V
Input Voltages with Respect to V _{SS} for \bar{G} and \bar{W}	18 to -0.4	V

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.

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.

RECOMMENDED DC OPERATING CONDITIONS READ, WRITE, OR ERASE

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	MCM2833-15, -20	V _{CC} 4.50	5.0	5.50	V
	MCM2833-15.5, -20.5	V _{CC} 4.75	5.0	5.25	V
Input High Voltage		V _{IH} 2.0	-	6.0	V
	(\bar{G} Pin 22 and \bar{W} Pin 27)	V _{IHH} 11.0	-	17.0	V
Input Low Voltage*	V _{IL}	-0.1	-	0.5	V

*The device will withstand undershoots to the -0.4 V level for a maximum duration of 10 ns.

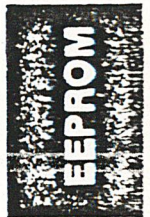
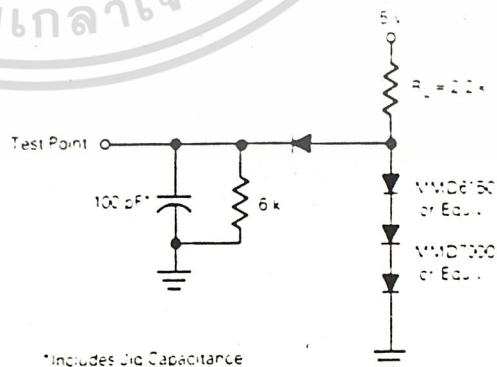
DC OPERATING CHARACTERISTICS

Characteristic	Condition	Symbol	Min	Typ	Max	Unit
Input Leakage Current (A0-A11, \bar{E})	V _{in} = V _{CC} Max	I _{in1}	-	-	10	μA
Input Leakage Current (\bar{G} , \bar{W})	V _{in} = 17 V	I _{in2}	-	-	10	μA
Output Leakage Current (DQ0-DQ7)	V _{out} = V _{CC} Max, \bar{G} = V _{IH}	I _{OL1}	-	-	10	μA
Output Leakage Current (DQ0-DQ7)	V _{out} = 0.4 V, \bar{G} = V _{IH}	I _{OL2}	-	-	10	μA
V _{CC} Supply Current, Standby	\bar{E} = V _{IH} , \bar{G} = V _{IL}	I _{CC1}	-	-	35	mA
V _{CC} Supply Current, Active (Read)	\bar{E} = V _{IL} , \bar{G} = V _{IL} , \bar{W} = V _{IH}	I _{CC2}	-	-	100	mA
V _{CC} Supply Current, Active (Erase/Write)	See Mode Selection Table	I _{CC3}	-	-	100	mA
Output Low Voltage	I _{OL} = 2.1 mA	V _{OL}	-	-	0.4	V
Output High Voltage	I _{OH} = -400 μA	V _{OH}	2.4	-	-	V

CAPACITANCE (f = 1.0 MHz, T_A = 25°C, Periodically Sampled Rather than 100% Tested)

Characteristic	Symbol	Typ	Max	Unit
Input Capacitance	C _{in}	5	10	pF
Output Capacitance	C _{out}	-	10	pF

FIGURE 1 - AC TEST LOAD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCM2833

AC OPERATING CONDITIONS AND CHARACTERISTICS (Full operating voltage and temperature range unless otherwise noted)

AC Test Conditions

Input Transition Times: $5 \text{ ns} \leq t_r = t_f \leq 10 \text{ ns}$
 Input Pulse Transition Levels: 0.45 Volts (V_{iL}) to 2.4 Volts (V_{iH})
 Output Load: See Figure 1

Test Timing Measurement
 Reference Levels

Input 1 V and 2 V
 Output 0.8 V and 2 V

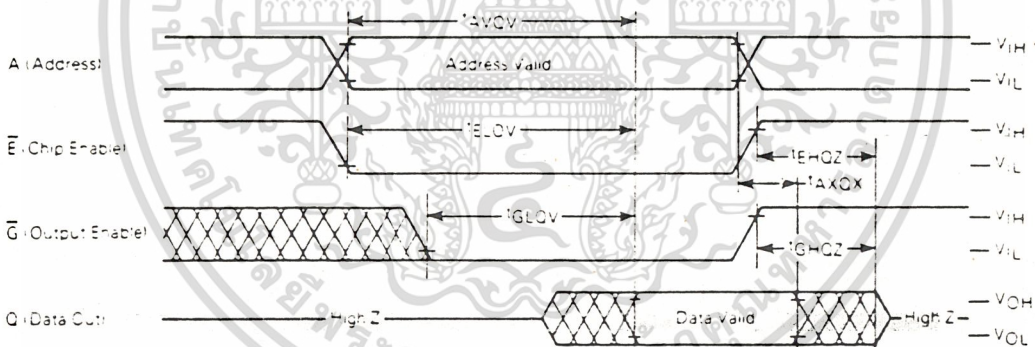
READ OPERATION ($\bar{W} = V_{iH}$)

Parameter	Condition	Symbol	MCM2833-15		MCM2833-20		Unit	Notes
			Min	Max	Min	Max		
Address Valid to Output Valid (Address Access Time)	$\bar{E} = \bar{G} = V_{iL}$	t_{AVQV}	-	150	-	200	ns	-
Output Enable to Output Valid (Output Enable Access Time)		t_{GLOV}	-	70	-	75	ns	-
Chip Enable to Output Valid (Chip Enable Access Time)		t_{ELOV}	-	150	-	200	ns	-
Output Disable to Output High Z	$\bar{E} = V_{iL}$	t_{GHQZ}	0	60	0	60	ns	3
Chip Disable to Output High Z	$\bar{G} = V_{iL}$	t_{GHCZ}	0	60	0	60	ns	3
Address Invalid to Output Invalid		t_{AXQV}	0	-	0	-	ns	-

NOTE:

- The parameters t_{GHQZ} and t_{GHCZ} may define the time at which the outputs achieve the open or High Z state and are not referenced to a level.

READ MODE TIMING DIAGRAM



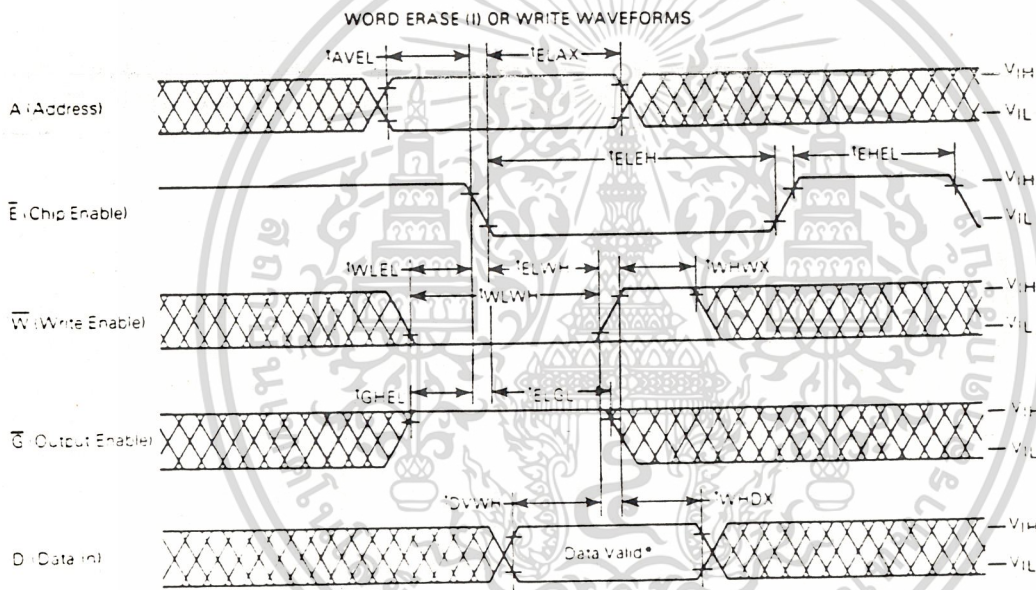
EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

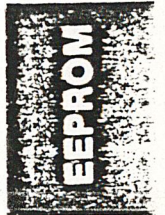
MCM2833

AC WRITE OR ERASE CHARACTERISTICS ($\bar{G} = V_{IH}$ During Write or Word Erase (II))

Parameter	Symbol	Min	Max	Unit
Address Valid to Chip Enable (Address Setup Time)	t_{AVEL}	0	-	ns
Write Enable to Chip Enable	t_{WLEL}	0	-	ns
Chip Enable to Address Don't Care	t_{ELAX}	150	-	ns
Write High to Data Don't Care	t_{WHDX}	20	-	ns
Data Valid to Write High	t_{DVWH}	100	-	ns
Write Enable Pulse Width	t_{WLWH}	150	-	ns
Write Enable Hold Time	t_{ELWH}	150	-	ns
Chip Disable to Chip Enable (Enable Latch Setup Time)	t_{EHEL}	350	-	ns
Write or Erase Time	t_{ELEH}	9	25	ms
(Chip Enable Pulse Width)	t_{ELEH}	25	50	ms
Data Latch Time	t_{LHWX}	50	-	ns
Output Enable to Chip Enable	t_{GHEL}	0	-	ns
Output Enable Hold Time	t_{ELGL}	150	-	ns



* Data in during Word Erase (II) (DCC-DC7) = V_{IH}



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCM2833

FUNCTIONAL DESCRIPTION

All inputs for the operating modes are TTL levels with the exception of bulk and page erase.

READ MODE

The MCM2833 uses 2-line control architecture for read operation to avoid bus contention problems. Data is available at the Data outputs of the selected device at t_{AVQV} with Chip Enable (\bar{E}), and Output Enable (\bar{G}) at V_{IH} or, at t_{GLQV} with Chip Enabled (\bar{E}), and address stable. In the read mode the device can be accessed similar to a static RAM. This can be done by holding Chip Enable active low and supplying the next address locations in a ripple through fashion with the next access determined by t_{AVQV} . The outputs of two or more EEPROMs may be Or'ed to the same data bus. Only one EEPROM should have its outputs selected to prevent data bus contention between two devices in this configuration. The outputs of other EEPROMs should be deselected with the Output Enable (\bar{G}) or Chip Enable (\bar{E}) input at a high TTL level.

STANDBY MODE

The Standby mode of the MCM2833 is achieved by applying a TTL high signal (V_{IH}) to Chip Enable (\bar{E}) input. When the device is in the Standby mode, the outputs are in the high impedance state, independent of the Output Enable (\bar{G}) input. When the MCM2833 is placed in the Standby mode, the active power dissipation is reduced by 72%.

WRITE OR ERASE

After each erasure, all bits of the selected byte(s) are in the "1" state. Data is introduced by selectively programming

(writing) "0s" into the desired bit locations. Although only "0s" will be programmed (written), both "1s" and "Cs" can be presented in the data word. The only way to change a "0" to a "1" is by electrical erasure.

Write or Erase Mode selection is controlled by applying the required sequence of signals to the device. The Output Enable (\bar{G}), Write Enable (\bar{W}), and AO-A11 address inputs are latched on the falling edge of Chip Enable (\bar{E}). DCC-DQ7 are latched on the rising edge of Write Enable (\bar{W}). To enter new address, control (\bar{G} and \bar{W}), and/or data, the Chip Enable (\bar{E}) signal must be clocked to V_{IH} for > 350 ns (t_{EHLL}), otherwise, the previous address and data may stay latched and inhibit the entry of the new information.

WRITE OR ERASE INHIBIT

Programming (writing) or erasure of multiple EEPROMs in parallel is easily accomplished. Except for Chip Enable (\bar{E}), all like inputs of the parallel devices may be common. A high level on the Chip Enable (\bar{E}) input inhibits the EEPROM from being programmed (written) or erased.

WRITE OR ERASE VERIFY

To determine that the word(s) was correctly programmed (written) or erased, a normal read operation can be performed. A read following after a write or erase cycle will require that Chip Enable (\bar{E}) goes to V_{IH} and is held for > 350 ns (t_{EHLL}) and that Write Enable (\bar{W}) is at a "1" and Output Enable (\bar{G}) at a "0" when Chip Enable (\bar{E}) goes to V_{IL} at the beginning of a read cycle. The data addressed will be valid on output lines at t_{GLQV} access time after Chip Enable (\bar{E}) goes low.

AC CHARACTERISTICS, SPECIAL ERASE MODE

Mode	Symbol	Min	Max	Unit
Address valid to Chip Enable (Address Setup Time)	t_{AEVLE}	0	-	ns
Output Disable to Chip Enable	t_{GHEH}	0	-	ns
Data Valid to Chip Enable	t_{DHEH}	0	-	ns
Write Enable to Chip Enable	t_{WLEH}	0	-	ns
Chip Enable to Address Don't Care	t_{ELAX}	150	-	ns
Chip Enable to Output Enable Don't Care	t_{ELGX}	150	-	ns
Chip Enable to Data In Don't Care	t_{ELDX}	150	-	ns
Chip Enable to Write Enable Don't Care	t_{ELWX}	150	-	ns
Chip Disable to Chip Enable (Enable Latch Setup Time)	t_{EHLL}	350	-	ns
Write or Erase Time				
(Chip Enable Pulse Width)				
Write Disable to Chip Enable	t_{WHEH}	0	-	ns
Data Latch Time	t_{LHAX}	50	-	ns
Data In High to Write High	t_{DHVH}	100	-	ns
Write High to Data Don't Care	t_{WHDx}	20	-	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EEPROM

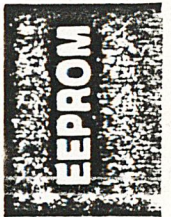
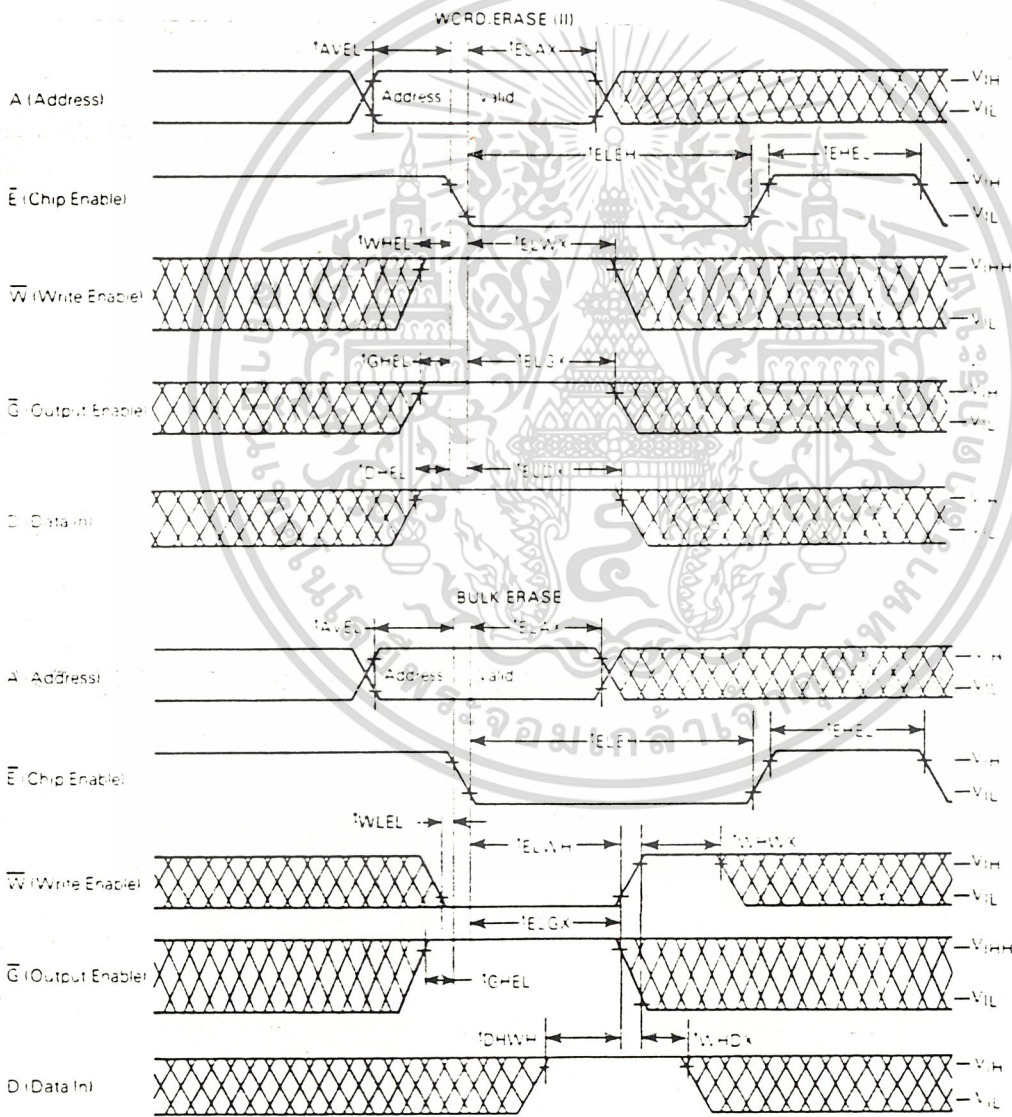
SPECIAL MODE SECTION

SPECIAL MODE SELECTIONS (V_{IHH} = 11 V to 17 V)

Special Modes	Pin Number and Function				Notes
	11-13, 15-19 DQ0-DQ7	Pin 20 E	Pin 22 G	Pin 27 W	
Word Erase (III)	V _{IH}	V _{IL}	V _{IH}	V _{IHH}	-
Page Erase (Row)	V _{IH}	V _{IL}	V _{IHH}	V _{IHH}	4
Page Erase (Column)	V _{IH}	V _{IL}	V _{IHH}	V _{IH}	5
Bulk Erase	V _{IH}	V _{IL}	V _{IHH}	V _{IL}	-

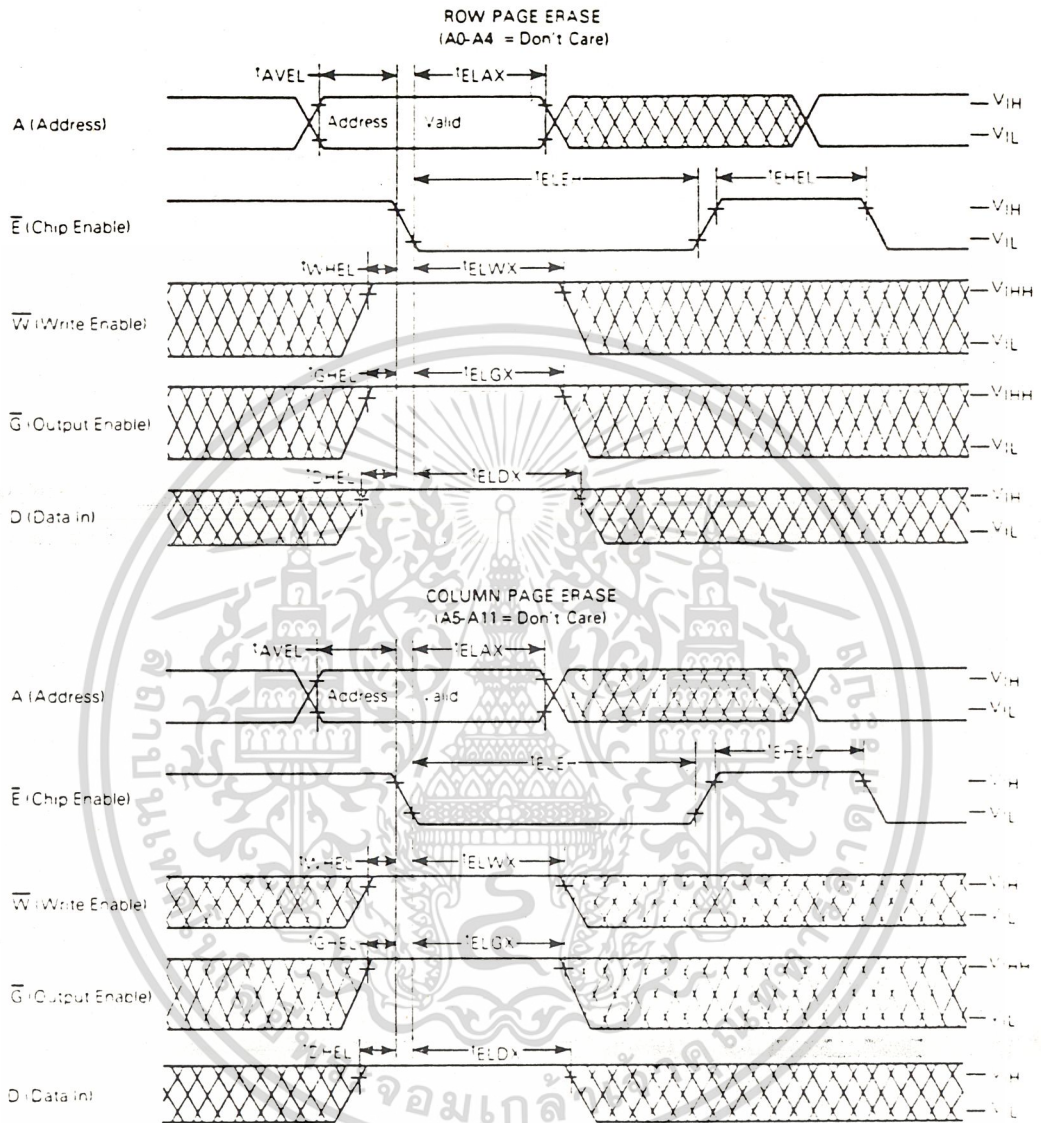
NOTES

- 4 Row Page Erase Mode: There are 128 rows containing 32 bytes each. Individual rows are selected with addresses A5-A11 (A0-A4 are don't care).
- 5 Column Page Erase Mode: There are 32 columns containing 128 bytes each. Individual columns are selected with addresses A0-A4 (A5-A11 are don't care).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCM2833



EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MOTOROLA

Product Preview

SMART 64K-BIT ELECTRICALLY ERASABLE PROM

The MCM2864 is a 65,536-bit Smart Electrically Erasable Programmable Read Only Memory (E²PROM) designed for handling data in applications requiring both nonvolatile memory and in-system reprogramming.

The MCM2864 improves processor throughput by reducing the system overhead due to its in-system transparent erase-before-write capability. The device operates from a single +5 V power supply in the read and smart write mode. Word read and write can be controlled entirely by TTL signal levels.

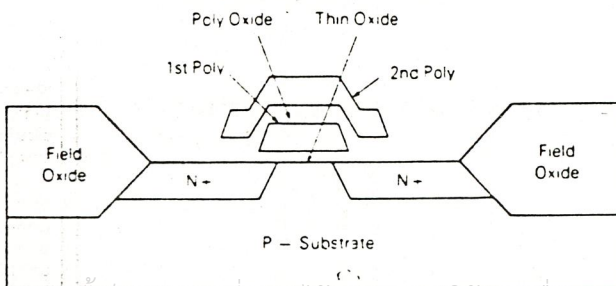
To ease system design, the high voltage needed by the device for the smart write cycle is generated internally. Another ease-of-use feature is the choice of erase modes (bulk, byte, row, or column) to optimize system erase/write time. For microprocessor compatibility, on-chip latches are provided for addresses, data, and controls allowing the microprocessor to perform other tasks while the MCM2864 is programming itself by the provision of a RDY, BUSY function to indicate status.

The MCM2864 is fabricated using Motorola's FETMOS technology (Floating-gate Electron Tunneling MOS), which has the advantages of good data retention, good endurance, and conventional processing.

The device pinout is part of Motorola's industry standard byte wide Nonvolatile Memory family, providing cost-effective density upgrades.

- Single +5 Volt Power Supply
- Organized as 8192 Bytes of 8 Bits
- Fast Access Time of 200 ns Maximum
- Low Power Dissipation
- In-System Automatic and Transparent Erase Before Write
- RDY, BUSY Function to Indicate Status
- Data Protected During Power Up and Power Down
- 10,000 Write Cycles Per Byte
- Data Integrity of 10 Years
- Latched Address, Data, and Controls During Write
- Chip Enable and Output Enable for Two Line Bus Control
- 10 ms for Byte Write and Internally Timed
- 28-Pin JEDEC Standard Pinout
- Internal Automatic Erase/Write Verify

FETMOS (Floating-Gate Electron Tunneling MOS)

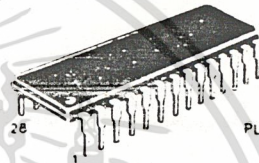


MCM2864

HMOS

(IN-CHANNEL, SILICON GATE)

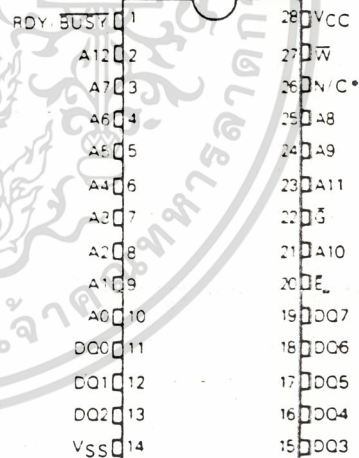
8192 x 8-BIT
ELECTRICALLY ERASABLE
PROGRAMMABLE READ
ONLY MEMORY



P SUFFIX
PLASTIC PACKAGE
CASE 710

L SUFFIX CERAMIC PACKAGE
ALSO AVAILABLE - CASE 719

PIN ASSIGNMENT



* For normal operation, pin 26 can be tied to VSS or VCC

PIN NAMES

A	Address
DQ	Data Input/Output
\bar{E}	Chip Enable
\bar{O}	Output Enable
\bar{W}	Write Enable
RDY, BUSY	Ready/Busy
N/C	No Connect
VCC	+5V Power Supply
VSS	Ground

E²PROM

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

เอกสารนี้เป็นเอกสารผลงานวิจัยสำหรับการใช้งานเพื่อการศึกษานี้ ไม่ควรนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากผู้เกี่ยวข้อง

เอกสารนี้เป็นเอกสารผลงานวิจัยสำหรับการใช้งานเพื่อการศึกษานี้ ไม่ควรนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากผู้เกี่ยวข้อง

เอกสารนี้เป็นเอกสารผลงานวิจัยสำหรับการใช้งานเพื่อการศึกษานี้ ไม่ควรนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากผู้เกี่ยวข้อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;9 MAR 1995

```
                LJMP      ORG          ;ORG.
ORG      0013H   LJMP      INTERFACE   ;INTERFACE MODE
                ORG      0023H   LJMP      SERIAL_INT   ;STROBE SIGNAL
                ORG      0100H
;BAND          EQU      R0
;DPTR LOWER   EQU      R1
;ADDRESS GPIB EQU      R2
;TIMER        EQU      R3
;TRUE CH      EQU      R4
;SCANNING     EQU      R5
;MEM          EQU      R7
;SAVE_DIV     EQU      21H
;LEVEL        EQU      23H
;DECODER & SIGNBIT EQU    24H
;HEX TO DEC   EQU      25H
;SAVE_KEY     EQU      26H
```

;CONTROL 8051

```
ORG:           MOV      TCON,#05H      ;SET TCON
               MOV      IE,#95H       ;SET IE
               MOV      IP,#10H       ;SET IP
               MOV      22H,#03H      ;SCAN_KEY
               MOV      R0,#60H       ;BAND
```

;CONTROL 8255

```
MOV      DPTR,#03H
MOV      A,#82H
MOVX    @DPTR,A      ;MOVE 82H TO CONTROL PORT
```

;GET ADDRESS FOR IEEE-488 (GPIB)

```
MOV      R3,#0FFH
MOV      P1,#03H
GPIB1:    LCALL    CHECK_KEY
          JZ      GPIB5
          CJNE   A,#10H,GPIB2      ;CHECK PRESS KEY
          INC    R2                ;INCREASE ADDRESS
          CJNE   R2,#1FH,GPIB3
          MOV    R2,#00H
          LJMP   GPIB3
GPIB2:    DEC     R2                ;DECREASE ADDRESS
          CJNE   R2,#0FFH,GPIB3
          MOV    R2,#1EH
GPIB3:    MOV     R3,#02H          ;LOOP R3,#0FFH***
          MOV    A,R2
          LCALL  SEGMENT          ;SEGMENT DISPLAY
          MOV    A,#02H          ;DELAY 400mSEC.***
GPIB4:    DEC     A
          JNZ    GPIB4
GPIB5:    MOV     A,#02H          ;DELAY 10mSEC.***
GPIB6:    DEC     A
          JNZ    GPIB6
          DJNZ   R3,GPIB1
          MOV    DPTR,#00H
          MOV    A,#0ABH
          MOVX   @DPTR,A
          MOV    A,#02H          ;DELAY 1 SEC.***
GPIB7:    DEC     A
          JNZ    GPIB7
          MOVX   @DPTR,A      ;CLEAR SEGMENT DISPLAY
```

;SCAN KEY SWITCH

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังให้ตัดแปะและต้องอ้างอิงถึงที่มาของเอกสารฉบับนี้ทุกครั้ง

```

RUN:          LCALL      CLEAR          ;CLEAR DEVICE
              MOV        R5,#04H
SCAN0:        DEC        R5
              CJNE       R5,#0FFH,SCAN1
              LJMP       RUN            ;SCAN LOOP
SCAN1:        MOV        P1,R5
              LCALL      CHECK_KEY
              JZ         SCAN0         ;NOT PRESS GOTO SCAN KEY
;KEYPRESS
              MOV        26H,A         ;SAVE KEY
              MOV        R3,#02H      ;DELAY R3,#00AH***
SCAN2:        LCALL      CHECK_KEY
              XRL        A,26H
              JNZ        SCAN0         ;NOT SAME GOTO SCAN KEY
              DJNZ       R3,SCAN2
              MOV        A,26H        ;RETURN VALUES
              ORL        A,R5         ;UPPER AND LOWER
              CJNE       A,#13H,SCAN3
              LJMP       LV_UP
SCAN3:        CJNE       A,#23H,SCAN4
              LJMP       LV_DOWN
SCAN4:        CJNE       A,#12H,SCAN5
              LJMP       CH_UP
SCAN5:        CJNE       A,#22H,SCAN6
              LJMP       CH_DOWN
SCAN6:        CJNE       A,#11H,SCAN7
              LJMP       READ
SCAN7:        CJNE       A,#21H,SCAN8
              LJMP       WRITE
SCAN8:        CJNE       A,#10H,SCAN9
              LJMP       MEM1
SCAN9:        LJMP       SCAN0         ;GOTO SCAN KEY
;CHECK KEY
CHECK_KEY:    MOV        A,P1
              ORL        A,#0CFH
              CPL        A
              RET
;CHANNEL COUNT UP
CH_UP:        CJNE       R7,#00H,CU4   ;CHECK MEM. 0
              INC        R0
              CJNE       R0,#6FH,CU1
              MOV        R0,#71H
CU1:          CJNE       R0,#7FH,CU2   ;CHECK MAX.
              MOV        R0,#60H
CU2:          MOV        23H,@R0      ;GET LV. FROM CH.
              LCALL      CH_LMC
              LCALL      TIMER
              JC         CH_UP
CU4:          LJMP       RUN          ;GOTO SCAN KEY
;CHANNEL COUNT DOWN
CH_DOWN:     CJNE       R7,#00H,CD4   ;CHECK MEM. 0
              DEC        R0
              CJNE       R0,#70H,CD1
              MOV        R0,#6EH
CD1:          CJNE       R0,#5FH,CD2   ;CHECK MIN.
              MOV        R0,#7EH
CD2:          MOV        23H,@R0      ;GET LV. FROM CH.
              LCALL      CH_LMC
              LCALL      TIMER
              JC         CH_DOWN
CD4:          LJMP       RUN          ;GOTO SCAN KEY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำเอกสารนี้ไปลงนิตยสารหรือสิ่งพิมพ์อื่นใดโดยไม่ได้รับอนุญาต

```

;LEVEL COUNT UP
LV_UP:      CJNE      R0,#00H,LU5      ;CHECK CH. 0
            LJMP      RUN              ;GOTO SCAN KEY
LU5:        MOV       A,23H
            JNB       ACC.4,LU2        ;JUMP IF (BIT 1C)=0
            CJNE     A,#11H,LU1        ;CHECK -1 dB
            CLR      A
            LJMP     LU4
LU1:        DEC      A
            LJMP     LU4
LU2:        CJNE     A,#0CH,LU3        ;CHECK 12 dB
            LJMP     RUN              ;GOTO SCAN KEY
LU3:        INC      A
LU4:        MOV       23H,A            ;GET LV.
            LCALL    LV_LMC
            LCALL    TIMER
            JC       LV_UP
            LJMP     RUN              ;GOTO SCAN KEY

;LEVEL COUNT DOWN
LV_DOWN:    CJNE     R0,#00H,LD5      ;CHECK CH. 0
            LJMP     RUN              ;GOTO SCAN KEY
LD5:        MOV       A,23H
            JB       ACC.4,LD2        ;JUMP IF (BIT 1C)=1
            JNZ      LD1              ;JUMP IF A<>0 (CHECK 0 dB)
            MOV      A,#11H
            LJMP     LD4
LD1:        DEC      A
            LJMP     LD4
LD2:        CJNE     A,#1CH,LD3        ;CHECK -12 dB
            LJMP     RUN              ;GOTO SCAN KEY
LD3:        INC      A
LD4:        MOV       23H,A            ;GET LV.
            LCALL    LV_LMC
            LCALL    TIMER
            JC       LV_DOWN
            LJMP     RUN              ;GOTO SCAN KEY

;CONTROL LMC835
CH_LMC:     MOV       A,R0
            MOV       C,ACC.4
            MOV       P1.6,C          ;CHOOSE LMC835
            ANL      A,#0FH
            MOV       24H,A           ;SAVE DECODER
            ADD      A,#80H
            MOV       SBUF,A          ;SEND CH. TO SERIAL PORT
            LCALL    DELAY
LV_LMC:     MOV       A,23H
            MOV       DPTR,#TABLE
            MOVC     A,@A+DPTR
            MOV       SBUF,A          ;SEND LV. TO SERIAL PORT
            LCALL    DELAY
            MOV       C,23H.4
            MOV       24H.4,C         ;SAVE SIGN BIT
            MOV       DPTR,#02H
            MOV       A,24H
            MOVX     @DPTR,A          ;OUT DECODER & SIGN BIT
            MOV       A,23H
            ANL      A,#0FH
            MOV       PSW,#00H        ;CLEAR PSW
            DA
            MOV       DPTR,#00H
            MOVX     @DPTR,A          ;OUT LV NO. TO PORTA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ปรึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET

;TIMER
TIMER:      MOV      R3,#02H      ;DELAY R3,#OFFH***500mSEC.
            CLR      C
TIME1:      MOV      A,#02H      ;DELAY A,#OFFH***
TIME2:      DEC      A
            JNZ     TIME2
            LCALL   CHECK_KEY
            XRL     A,26H      ;COMPARE
            JNZ     TIME3
            DJNZ    R3,TIME1
            SETB   C
TIME3:      RET

;DELAY TIME FOR SERIAL INT.
DELAY:      MOV      R3,#04H
DELAY1:     DJNZ    R3,DELAY1
            RET

;SERIAL INT. FOR STROBE OF LMC835
SERIAL_INT: NOP
            SETB   P1.7      ;SET STROBE
            NOP
            CLR    P1.7      ;CLEAR STROBE
            RETI

;CLEAR DEVICE
CLEAR:      MOV      R0,#7FH
CLR1:      DEC      R0
            MOV     @R0,#00H    ;CLEAR REGISTER
            CJNE   R0,#60H,CLR1
            MOV     23H,#00H    ;CLEAR LV. REGISTER
            MOV     DPTR,#00H
            CLR    A
            MOVX   @DPTR,A      ;CLEAR 7 SEGMENT
            MOV     DPTR,#02H
            MOVX   @DPTR,A      ;CLEAR DECODER
            SETB   P1.6      ;CHOOSE LMC835, 2
CLR2:      MOV     A,#8FH
            DEC     A
            MOV     SBUF,A      ;SEND CH. TO SERIAL PORT
            LCALL  DELAY
            MOV     SBUF,#00H    ;CLEAR LV.
            LCALL  DELAY
            CJNE   A,#80H,CLR2
            CLR    P1.6      ;CHOOSE LMC835/1
            MOV     A,#8FH
CLR3:      DEC     A
            MOV     SBUF,A      ;SEND CH. TO SERIAL PORT
            LCALL  DELAY
            MOV     SBUF,#00H    ;CLEAR LV.
            LCALL  DELAY
            CJNE   A,#80H,CLR2
            RET

;TABLE LEVEL
TABLE:      DB      00H,60H,50H,48H,44H,42H,52H,6AH
            DB      56H,41H,69H,6DH,6FH,00H,00H,00H
            DB      00H,20H,10H,08H,04H,02H,12H,2AH
            DB      16H,01H,29H,2DH,2FH,00H,00H,00H

;MEMORY COUNT UP
MEM1:      MOV      R0,#60H      ;CLEAR BAND REGISTER

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR, #02H
CLR A
MOVX @DPTR, A ;CLEAR BAND DISPLAY
LCALL M_COUNT
MOV R3, #0AH ;DELAY R3, #0FFH***1.5 SEC.
MEM2: MOV A, #02H ;DELAY A, #0FFH***
MEM3: DEC A
JNZ MEM3
LCALL CHECK_KEY
XRL A, 26H ;COMPARE
JNZ MEM6
DJNZ R3, MEM2
MEM4: LCALL M_COUNT
MOV R3, #02H ;DELAY R3, #0FH***80 mSEC.
MEM5: LCALL CHECK_KEY
XRL A, 26H ;COMPARE
JNZ MEM6
DJNZ R3, MEM5
LJMP MEM4
MEM6: LJMP RUN ;GOTO SCAN KEY

M_COUNT: INC R7
CJNE R7, #64H, M3 ;RUN
MOV R7, #00H
M3: MOV A, R7 ;HEX TO DEC
;SEVEN SEGMENT DISPLAY DECIMAL
SEGMENT: MOV B, #0AH
DIV AB
MOV B, 25H, B
MOV B, #10H
MUL AB
ADD A, 25H
MOV DPTR, #00H
MOVX @DPTR, A ;OUT MEM. TO PORTA
RET

;READ
READ: MOV R1, #00H ;CLEAR R1
CJNE R7, #00H, RD1
LJMP RUN ;GOTO SCAN KEY
RD1: LCALL SET_DPTR
RD2: INC R0
CJNE R0, #6FH, RD3
MOV R0, #71H
RD3: CJNE R0, #7FH, RD7 ;CHECK LOAD FINISH
RD4: DEC R0
CJNE R0, #70H, RD5
MOV R0, #6EH
RD5: CJNE R0, #60H, RD6
LJMP RUN ;GOTO SCAN KEY
RD6: MOV A, R0
MOV C, ACC.4
MOV P1.6, C ;CHOOSE LMC835
ANL A, #0FH
ADD A, #80H
MOV SBUF, A ;SEND CH. TO SERIAL PORT
LCALL DELAY
MOV DPTR, #TABLE
MOV A, @R0
MOVC A, @A+DPTR
MOV SBUF, A ;SEND LV. TO SERIAL PORT
LCALL DELAY
LJMP RD4
RD7: INC R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามดัดแปลงหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกประการ

```

MOV          DPL,R1          ;SEND MEM. TO DPTR(LOWER)
MOVX         A,@DPTR
MOV          @R0,A          ;LOAD LV. FROM EX-RAM
LJMP        RD2

;WRITE
WRITE:      MOV          R1,#00H          ;CLEAR R1
            CJNE        R7,#00H,WD1
            LJMP        RUN              ;GOTO SCAN KEY
WD1:        LCALL       SET_DPTR
WD2:        INC         R0
            CJNE        R0,#6FH,WD3
            MOV         R0,#71H
WD3:        CJNE        R0,#7FH,WD4
            MOV         R0,#60H
            LJMP        RUN              ;GOTO SCAN KEY
WD4:        INC         R1
            MOV         DPL,R1          ;SEND MEM. TO DPTR(LOWER)
            MOV         A,@R0
            MOVX        @DPTR,A
            LJMP        WD2

;SET_DPTR
SET_DPTR:   MOV         A,R7
            ANL         A,#0FH
            ORL         A,#10H
            MOV         DPH,A          ;SEND MEM. TO DPTR(UPPER)
            MOV         A,R7
            RL          A
            ANL         A,#0EOH
            MOV         R1,A          ;1---- --XXXXX
            RET

;INTERFACE MODE
INTERFACE:  MOV         P1,#07H          ;SET A,B,X/Y
            MOV         DPTR,#01H      ;PORT B (DATA)
IEEE3:     LCALL       H_SHAKE
            JB          ACC.7,IEEE1     ;CHECK ATN-L
            CLR         P1.0
            SETB        P1.0           ;SET NRFD-H
            CLR         P1.2           ;CLEAR X/Y
            RETI        ;QUIT TO RUN
IEEE1:     ANL         A,#7FH
            CJNE        A,02H,IEEE2    ;COMPARE ADDRESS
            MOV         DPTR,#WAIT
            MOV         09H,DPH
            MOV         08H,DPL
            MOV         SP,#09H
            MOV         R7,#00H        ;CLEAR MEM.
            LCALL       H_SHAKE
            JNB         ACC.7,CHAN      ;JUMP IF ATN-H
            ANL         A,#7FH
            ;SELECT DEVICE CLEAR
            CJNE        A,#04H,IEEE5   ;CHECK COMMAND SDC
            CLR         P1.0
            SETB        P1.0           ;SET NRFD-H
            LCALL       CLEAR
            RETI        ;GOTO WAIT
IEEE5:     LOCAL      ;GOTO LOCAL
            CJNE        A,#01H,IEEE6   ;CHECK COMMAND GTL
            MOV         22H,#04H
            MOV         DPTR,#RUN
            MOV         09H,DPH
            MOV         08H,DPL

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังห้ามดัดแปลงหรือทำซ้ำโดยไม่ได้รับอนุญาตของทางผู้จัดทำ

```

MOV          SP, #09H
IEEE6:      CLR          P1.0
            SETB        P1.0          ;SET NRFD-H
            RETI         ;QUIT TO RUN

;DEVICE CLEAR
IEEE2:      CJNE        A, #14H, IEEE3  ;CHECK COMMAND DCL
            CLR          P1.0
            SETB        P1.0          ;SET NRFD-H
            LCALL       CLEAR
            RETI         ;GOTO WAIT

;CHANNEL
CHAN:       ADD          A, #20H
            CJNE        A, #61H, CHAN1 ;CHECK CH >= 61
CHAN1:      JC          EXIT           ;GOTO WAIT
            CJNE        A, #7FH, CHAN2 ;CHACK CH < 7E
CHAN2:      JNC         EXIT           ;GOTO WAIT
            MOV         R0, A         ;GET CH.

;LEVEL
LCALL       H_SHAKE
LEV:        CJNE        A, #60H, LEV    ;CHECK LV >= 0dB
            JC          EXIT           ;GOTO WAIT
LEV1:       CJNE        A, #6DH, LEV1  ;CHECK LV <= 12dB
            JC          LEV2
LEV3:       CJNE        A, #71H, LEV3  ;CHECK LV <= -1dB
            JC          EXIT           ;GOTO WAIT
LEV6:       CJNE        A, #7DH, LEV6  ;CHECK LV >= -12dB
            JNC        EXIT           ;GOTO WAIT
LEV2:       ANL         A, #1FH
            MOV         @R0, A        ;GET LEVEL
            MOV         23H, A

;LINE FEED
LF:         LCALL       H_SHAKE
            CJNE        A, #0AH, LF
            CLR          P1.0
            SETB        P1.0          ;SET NRFD-H
            LCALL       CH_LMC
            RETI         ;GOTO WAIT

;HANDSHAKE
H_SHAKE:   CLR          P1.0
SHAKE1:    SETB        P1.0          ;SET NRFD-H
            JB          P1.3, SHAKE1  ;CHECK DAV-L (NRFD-L TOO)
            MOVX       A, @DPTR      ;GET DATA
            CPL
            CLR          P1.1
SHAKE2:    SETB        P1.1          ;SET NDAC-H
            JNB        P1.3, SHAKE2  ;CHECK DAV-H (NDAC-L TOO)
            RET

;EXIT
EXIT:      CLR          P1.0
            SETB        P1.0          ;SET NRFD-H
            RETI         ;GOTO WAIT

;WAIT & SCAN LOCAL KEY
WAIT:      MOV         P1, #00H      ;SCAN GTL
WAIT1:     JB          P1.5, WAIT1    ;GET KEY PRESS (LOCAL)
            MOV         22H, #04H   ;CAN SCAN LV.
            LJMP       RUN

END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AE.FRM

C:\WINDOWS\SYSTEM\GRID.VBX

C:\WINDOWS\SYSTEM\MSOLE2.VBX

C:\WINDOWS\SYSTEM\ANIBUTTON.VBX

C:\WINDOWS\SYSTEM\CMDIALOG.VBX

C:\WINDOWS\SYSTEM\CRYSTAL.VBX

C:\WINDOWS\SYSTEM\GAUGE.VBX

C:\WINDOWS\SYSTEM\GRAPH.VBX

C:\WINDOWS\SYSTEM\KEYSTAT.VBX

C:\WINDOWS\SYSTEM\MSCOMM.VBX

C:\WINDOWS\SYSTEM\MSMASKED.VBX

C:\WINDOWS\SYSTEM\MSOUTLIN.VBX

C:\WINDOWS\SYSTEM\PICCLIP.VBX

C:\WINDOWS\SYSTEM\SPIN.VBX

C:\WINDOWS\SYSTEM\THREED.VBX

FILEMSG.BAS

CONSTANT.TXT

ERROR.BAS

SCAN.FRM

SET.FRM

HELP.FRM

ERROR.FRM

SCANNING.FRM

SETTING.FRM

SAVE.FRM

LOAD.FRM

ABOUT.FRM

DISPLAY.FRM

ProjWinSize=45,416,221,273

ProjWinShow=0

IconForm='Ae'

Title='Automatic Equalizer.'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ExeName='AE.EXE'

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

