เครื่องบันทึกค่าคุณสมบัติของสารละลาย

PH RECORDER

โดย

นาย	วัชระ	เพลิดเพลิน	34132123
นาย	ภูษิต	ภู่ประดิษฐ์	34132159
นาย	สมชาย	ประวัติยากูร	34132167
นาย	อนุชา	ระงับภัย	34132176
นาย	อนุสรณ์	ลิ้มตระกูล	34132178

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032658

ภาควิชา เทคนิคอุตสาหกรรม

สาขา เทคโนโลยีโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องบันทึกค่าคุณสมบัติของสารละลาย

(PH RECORDER)

ผู้จัดทำ

1.	นาย วัชระ	เพลิดเพลิน	34132123
2.	นาย ภูษิต	ภู่ประดิษฐ์	34132159
3.	นาย สมชาย	ประวัติยากูร	34132167
4.	นาย อนุชา	ระงับภัย	34132176
5.	นาย อนุสรณ์	ลิ้มตระกูล	34132178

..... อาจารย์ที่ปรึกษา

(อาจารย์ ดลชัย สุขเจริญผล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032658

สารบัญ

หน้า

บทคัดย่อ

Abstract

บทนำ

บทที่ 1 หลักการเบื้องต้นของ PH RECORDER

1.1 BLOCK DIAGRAM และหลักการเบื้องต้นของ PH RECORDER 1-2

1.2 ส่วนประกอบของ PH RECORDER 2

บทที่ 2 ดิจิตอล PH มิเตอร์ (DIGITAL PH METER)

2.1 คุณลักษณะเด่นของ PH METER 3

2.2 คุณสมบัติของ PH METER 3-5

2.3 รายละเอียดด้านหน้าของเครื่อง 5-6

2.4 ขั้นตอนการปรับเทียบ 6

2.5 วิธีการวัด 7

2.6 การเปลี่ยนแบตเตอรี่ 7

2.7 วิธีการแก้ปัญหาเบื้องต้น 7

บทที่ 3 CALIBRATE AMP AND ANALOG TO DIGITAL

3.1 หลักการทำงานของวงจร CALIBRATE AMP 8

3.2 วงจรแปลงอะนาลอกเป็นดิจิตอล 8-12

3.3 การคำนวณความถี่ CLOCK ของ A/D 12

3.4 การปรับ SPAN 13-14

3.5 การ START A/D 14

3.6 การกระตุ้นให้ A/D ปล่อยข้อมูลออกมา 14

บทที่ 4 พอร์ทสื่อสารอนุกรม

4.1 โครงสร้างของพอร์ทสื่อสาร 16-18

4.2 การอินเตอรัพต์ 19

4.3 รูปแบบข้อมูลที่รับหรือส่ง 19

4.4 ชิป 8250 20-23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 สถานะของ 8250 เมื่อเริ่มต้น 23-25
4.6 การต่อวงจรเข้ากับระบบ 25-28
4.7 การใช้งานรีจิสเตอร์ต่างๆบน 8250 29-38
บทที่ 5 การศึกษาโปรแกรม C ที่ใช้ส่งข้อมูลแบบอะซิงโครนัส
5.1 Port Initialization 40-42
5.2 การส่งข้อมูล 42-43
5.3 Checking a port status 43-44
5.4 การรับข้อมูล 44-45
บทที่ 6 วงจรที่ใช้งาน 46-52
บทที่ 7 FLOW CHART และ โปรแกรมที่ใช้งาน 53-69
บทที่ 8 การทดลองและผลการทดลอง 70-71
บทที่ 9 สรุปปัญหาการแก้ไขและแนวความคิดในการพัฒนา PH RECORDER
9.1 ปัญหาด้านฮาร์ดแวร์ (HARDWARE) 72
9.2 ปัญหาด้านซอฟท์แวร์ (SOFTWARE) 72-73
9.3 แนวความคิดในการพัฒนา PH RECORDER 73
กิตติกรรมประกาศ
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

การบันทึกผลของข้อมูล และแสดงผลการบันทึกทางคอมพิวเตอร์ในการทดลองนี้ จะใช้วิธีการแปลงข้อมูล ซึ่งเป็นสัญญาณ Analog ให้อยู่ในรูปของสัญญาณ Digital แล้วทำการ จัดรูปแบบของข้อมูล ให้เป็นการส่งข้อมูลแบบอนุกรม ซึ่งจะต้องทำการจัด Start bit, Parity check bit และ Stop bit ในแต่ละเฟรมของข้อมูลแล้วนำข้อมูลนั้น ไปบันทึกโดยเครื่อง Computer ซึ่งทำงานโดย Soft ware ที่ออกแบบขึ้นมาเพื่อการบันทึกและแสดงผลของข้อมูลในระยะเวลาที่ต้องการ(ใช้การติดต่อผ่านทาง Port RS-232C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

Recording and demonstrating of ph-data to the computer in this experiment provided with translating the data signals form analog signals to digital signals then transformed these to serial forms which each frame consisted of start bit,parity check bit and stop bit, then send these to recorded by the computer controlled with special software to recording and demonstrating the data whenever you required. (communications among the control ler cord and the computer used RS-232C)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ในการบันทึกข้อมูล ของการเปลี่ยนแปลงระดับค่าของสารละลาย ถ้าหากใช้มนุษย์เป็นผู้บันทึกผล เพื่อนำผลที่ได้มาทำการเปรียบเทียบการเปลี่ยนแปลงระดับค่าของสารละลาย จะทำให้เสียเวลามาก ยิ่งเป็นการบันทึกผลตลอดเวลาจะทำให้เกิดความยุ่งยาก ดังนั้น กลุ่มผู้ทดลองจึงมีแนวความคิดว่า น่าจะนำข้อมูลที่ต้องการจะทำการบันทึก จากแหล่งที่ทำการวัดระดับค่าของสารละลาย นำมาทำการติดต่อเพื่อจะทำการบันทึก และแสดงผลของการเปลี่ยนแปลง ทางด้านคอมพิวเตอร์ ซึ่งจะสามารถบันทึกผลและแสดงผลได้ตลอดเวลา ที่มีการเปลี่ยนแปลง

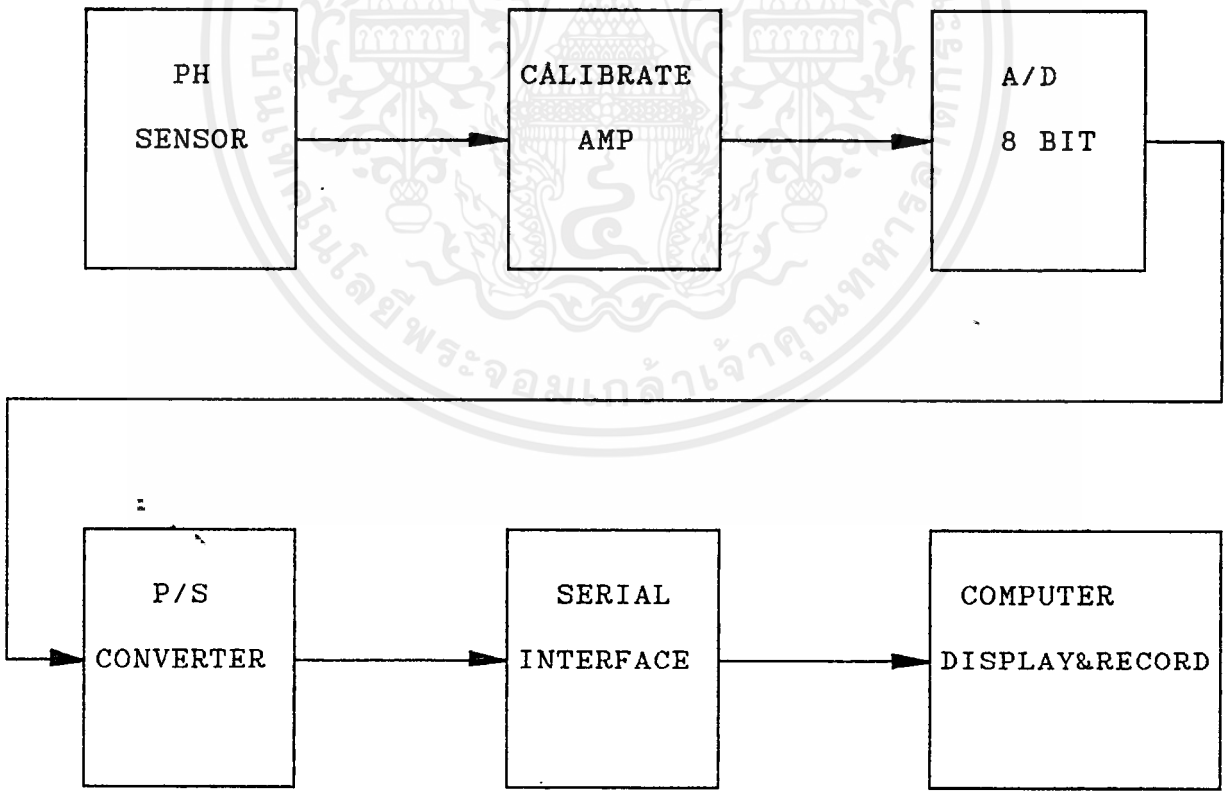
ในปริญญานิพนธ์ชุดนี้ ได้กล่าวถึง โครงสร้างของ PH METER ,หลักการของวงจร CALIBRATE และ วงจร A/D และการต่อ INTERFACE กับคอมพิวเตอร์ ซึ่งเป็นส่วนประกอบในการทำงานของ PH RECORDER ซึ่งกลุ่มผู้ทดลองได้สร้างเครื่องนี้ขึ้นมาเพื่อหวังว่า จะมีประโยชน์บ้างไม่มากก็น้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
หลักการเบื้องต้นของ PH RECORDER

PH RECORDER เป็นเครื่องที่ใช้สำหรับตรวจสอบ คุณสมบัติของสารละลาย และแสดงผลการบันทึกข้อมูล ค่าคุณสมบัติของสารละลายทางหน้าจอ(MONITOR)ของคอมพิวเตอร์การใช้คอมพิวเตอร์มาช่วยในการแสดงผล และบันทึกข้อมูลมีประโยชน์มากในด้านการศึกษาการเปลี่ยนแปลงของข้อมูล และปรับปรุงการทำงาน ให้มีประสิทธิภาพตามที่ต้องการเพิ่มขึ้นซึ่งจะสามารถตรวจสอบ การเปลี่ยนแปลงของข้อมูลได้ ทุกๆเวลาตามที่กำหนดไว้ในแต่ละช่วงก็จะแสดงที่มอนิเตอร์ ของคอมพิวเตอร์ ซึ่งจะทำให้สะดวกกว่าการวัดตรวจสอบแบบธรรมดาซึ่งจะต้องคอยอ่านค่าและจดบันทึกข้อมูลด้วยมนุษย์

1.1 BLOCK DIAGRAM และหลักการเบื้องต้นของ PH RECORDER



รูปที่ 1.1 บล็อกไดอะแกรมของ PH RECORDER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการเริ่มจากได้น่าสัญญาณ OUTPUT DC VOLTAGE จากPH SENSOR มาเข้าวงจร CALIBRATE AMP เพื่อปรับความละเอียดและถูกต้อง แล้วจึงน่ามาเข้าวงจร A/D (ANALOG TO DIGITAL) 8 บิต เพื่อแปลงเป็นสัญญาณดิจิตอล 8 บิต แล้วจึงน่ามาเข้า P/S CONVERTER (PARALLEL AND SERIES CONVERTER) เพื่อจัดรูปแบบข้อมูลเป็นข้อมูลแบบอนุกรม จากนั้นน่ามา INTERFACE กับ COMPUTER เพื่อแสดงผลและบันทึกข้อมูล ทางหน้าจอมอนิเตอร์ (MONITOR)

1.2 ส่วนประกอบของ PH RECORDER

ส่วนประกอบของ PH RECORDER นั้นมีส่วนสำคัญดังนี้

1.2.1 PH SENSOR เป็นส่วนที่ใช้ สำหรับตรวจสอบค่าคุณสมบัติของสารละลาย โดยจะมี 2 ส่วนคือ ตัวเครื่อง PH METER และ PH อีเลคโทรด เป็นแบบชีลด์ ตัวเรือนเป็นอีป๊อกซี่ ใช้งานทั่วๆไป 0-14 PH ซึ่งระดับค่าของสารละลาย จะถูกเปลี่ยนเป็นสัญญาณอนาลอก ประมาณ 0-2 มิลลิโวลท์ เมื่อเปรียบเทียบกับค่าของสารละลาย 0-14 PH

1.2.2 CALIBRATE AMP เป็นส่วนปรับความละเอียด และขยายสัญญาณอนาลอก ประมาณ 10 เท่าจาก PH SENSOR ก็จะได้สัญญาณอนาลอกที่ผ่าน CALIBRATE AMP มีขนาด 0-2 โวลท์ เพื่อที่จะป้อนให้กับ A/D (ANALOG TO DIGITAL CIRCUIT)

1.2.3 วงจร A/D เป็นส่วนเพื่อที่จะเปลี่ยนสัญญาณอนาลอก ให้เป็นสัญญาณดิจิตอล 8 บิต ซึ่งจะใช้ IC เบอร์ ADC 0804 ซึ่งมีโครงสร้างเป็น CMOS

1.2.4 P/S CONVERTER เป็นส่วนที่ทำการจัดรูปแบบข้อมูลให้เป็นข้อมูลแบบอนุกรมเพื่อทำการติดต่อกับคอมพิวเตอร์

1.2.5 SERIAL INTERFACE เป็นส่วนที่ทำการเชื่อมต่อกับคอมพิวเตอร์ ซึ่งคอมพิวเตอร์ จะมีพอร์ตสื่อสารอยู่ พอร์ตสื่อสารนี้มีชื่อเรียกอีกอย่างว่า คอม-พอร์ต (COM-PORT) ซึ่งพอร์ตสื่อสารนี้จะต้องเป็นไป ตามมาตรฐานการเชื่อมต่อแบบอนุกรม ที่เรียกว่า RS 232C

1.2.6 COMPUTER DISPLAY & RECORD เป็นส่วนที่ใช้ในการแสดงผลและบันทึกข้อมูลทางมอนิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ดิจิตอล PH มิเตอร์

(DIGITAL PH METER)

2.1 คุณลักษณะเด่นของ PH METER

2.1.1 ย่านการวัดกว้าง และมีความเที่ยงตรงสูง

2.1.2 จอแสดงผลเป็น LCD (LIQUID CRYSTAL DISPLAY) สิ้นเปลืองพลังงานน้อย

2.1.3 อินพุทอิมพีแดนซ์สูง จึงไม่ทำให้การวัดเกิดความผิดพลาด

2.1.4 ใช้กับ PH อีเล็คโทรดได้ทุกชนิด

2.1.5 มีปุ่มปรับการชดเชยอุณหภูมิ

2.1.6 มีปุ่มปรับ slope และ CALสะดวกในการปรับเทียบแบบจุดเดียวและ 2 จุด

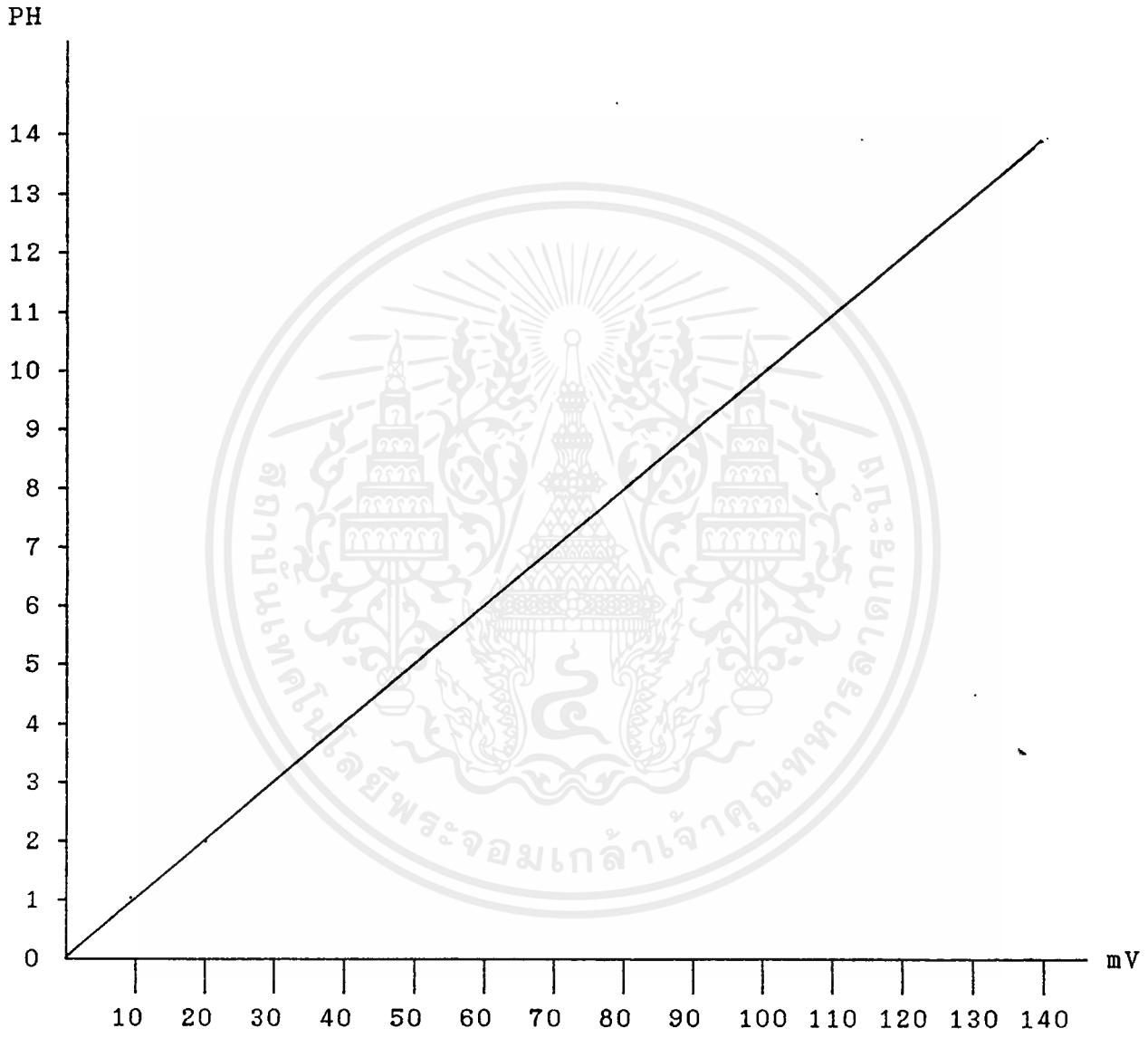
2.1.7 มีเครื่องหมายเตือนแบตเตอรี่หมด

2.1.8 ขนาดเล็กกะทัดรัด น้ำหนักเบา ทำให้ใช้งานได้สะดวก

2.2 คุณสมบัติของ PH METER

จอแสดงผล	: 13 มม.(0.5") LCD,3 1/2 หลัก
ย่านการวัด	: 0.00 ถึง 14.00 PH
ความละเอียด	: 0.01 PH
ความเที่ยงตรง	: +(0.1 PH + 1หลัก) หลังจากปรับเทียบแล้ว
อินพุทอิมพีแดนซ์	: 10^{12} โอมห์
การชดเชยอุณหภูมิ	: แบบธรรมดา,0 °C ถึง 100 °C
เวลาสุ่ม (Sampling)	: 0.5 วินาที
อุณหภูมิใช้งาน	: 0 °C ถึง 50 °C (32 °F ถึง 122 °F)
ความชื้นใช้งาน	: น้อยกว่า 80 % RH
VR ปรับเทียบ	: ปรับจากภายนอก PH 4 (ปรับ SLOPE), PH 7 (ปรับ CAL)
แหล่งจ่ายไฟ	: แบตเตอรี่ 006P DC 9V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

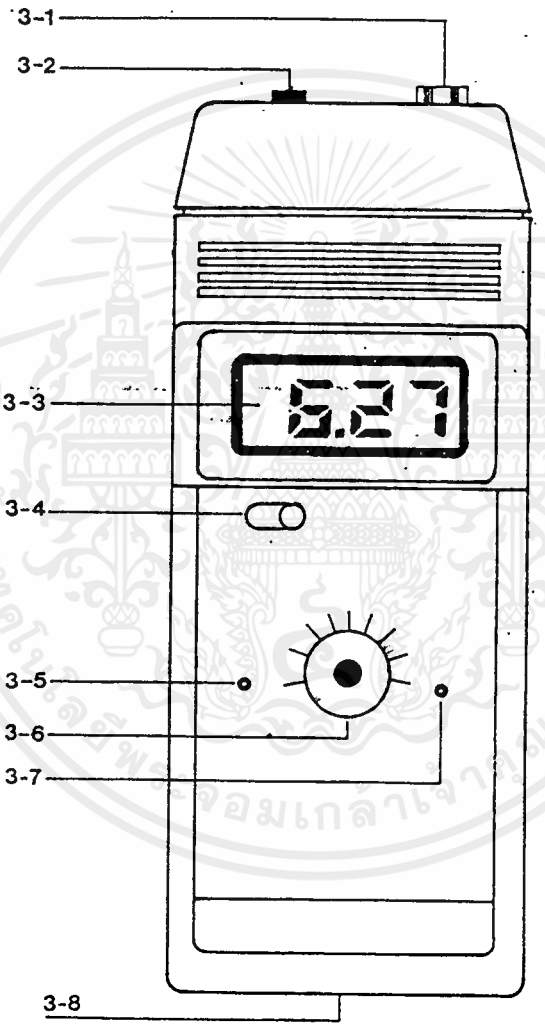


ความสัมพันธ์ระหว่างระดับสัญญาณ OUTPUT ของ PH METER กับค่าระดับ PH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ้นเปลืองพลังงาน	:	ประมาณ 2.7 mA
ขนาด	:	170*70*25 มม.(6.7*2.8*1 นิ้ว) สูง*กว้าง*หนา
น้ำหนัก	:	200 กรัม (0.5 ปอนด์)
PH อีเล็คโทรด	:	PH อีเล็คโทรดที่มีขั้วต่อเป็นแบบ BNC มีอิมพีแดนซ์มากกว่า 200 เมกกะโอมห์

2.3 รายละเอียดด้านหน้าของเครื่อง



รูปที่ 2.1 แสดงส่วนประกอบของ PH METER

2.3.1 ขั้วเสียบอีเล็คโทรดแบบ BNC

2.3.2 ขั้วดาต้า บัส เอาท์พุท

2.3.3 จอแสดงผล

2.3.4 สวิทช์ ON/OFF

2.3.5 VR ปรับ PH4 (SLOPE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 ปุ่มปรับชดเชยอุณหภูมิ

2.3.7 VR ปรับ PH7 (CAL)

2.3.8 ช่องใส่แบตเตอรี่และฝาปิด

2.4 ขั้นตอนการปรับเทียบ

2.4.1 อุปกรณ์ที่ใช้ในการปรับเทียบ

A.PH อีเล็คโทรด

B.สารละลายทดสอบ PH 7.00 และ PH 4.00

2.4.2 การปรับเทียบแบบ 2 จุด

A.ต่อสาย PH อีเล็คโทรดเข้ากับขั้วเสียบ BNC ของ PHมิเตอร์ แล้วจุ่มอีเล็คโทรดลงในสารละลาย PH 7

B.เลื่อนสวิทช์ ON/OFF ไปที่ ON (รูปที่ 2.1,2.3.4)

C.ปรับ VR PH7 (CAL) (รูปที่ 2.1,2.3.7) จนกระทั่งจอแสดงผลค่า 7.00

D.ปรับปุ่มชดเชยอุณหภูมิให้เท่ากับค่าอุณหภูมิของสารละลาย PH4

E.ล้างปลายอีเล็คโทรดด้วยน้ำกลั่น

F.จุ่มอีเล็คโทรดลงในสารละลาย PH4 ปรับ VR PH4 (SLOPE)(รูปที่ 2.1,2.3.5)

G.ขณะนี้เครื่องและอีเล็คโทรดก็ได้ถูกปรับเทียบให้เข้ากันแล้ว สามารถนำไปใช้งานได้ทันที

2.4.3 การปรับเทียบแบบจุดเดียว

A.ต่อสาย PH อีเล็คโทรดเข้ากับขั้วเสียบ BNC ของ PH มิเตอร์แล้วจุ่มอีเล็คโทรดลงในสารละลายมาตรฐาน (เช่น PH 7.00 เป็นต้น)

B.เลื่อนสวิทช์ ON/OFF ไปที่ตำแหน่ง ON

C.ปรับ VR PH7 (CAL) จนกระทั่งจอแสดงผลแสดงค่าเท่ากับค่าของสารละลายที่นำมาทดสอบ

D.ขณะนี้เครื่องและอีเล็คโทรดก็ได้ถูกปรับเทียบให้เข้ากันแล้ว สามารถนำไปใช้งานได้ทันที

E.โปรดจำไว้ว่า ต้องปรับปุ่มชดเชยอุณหภูมิให้เท่ากับค่าอุณหภูมิของสารละลายที่จะนำมาทดสอบเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 วิธีการวัด

หลังจากเครื่องและอีเล็คโทรดถูกปรับเทียบแล้วก็พร้อมที่จะใช้งานได้ทันที่

A. ต่อสาย PH อีเล็คโทรดเข้ากับขั้วเสียบ BNC ของ PH มิเตอร์

B. เลื่อนสวิทช์ ON/OFF ไปที่ตำแหน่ง ON

C. ปรับปุ่มชดเชยอุณหภูมิ ไปยังค่าที่เท่ากับค่าอุณหภูมิของสารละลายที่จะทำการทดสอบ

D. จุ่มอีเล็คโทรดลงในสารละลายแล้วอ่านค่าที่วัดได้บนจอแสดงผล

E. หลังจากวัดค่าเรียบร้อยแล้ว ให้ล้างอีเล็คโทรดด้วยน้ำกลั่นทุกครั้ง

2.6 การเปลี่ยนแบตเตอรี่

(1) เมื่อมีเครื่องหมาย "LO BAT" ปรากฏที่มุมซ้ายด้านบนของจอแสดงผล จำเป็นจะต้องมีการเปลี่ยนแบตเตอรี่ใหม่ แต่อย่างไรก็ตาม เครื่องยังสามารถใช้งานได้อีกหลายชั่วโมงก่อนที่จะเกิดการวัดผิดพลาด

(2) เลื่อนฝาปิดแบตเตอรี่ออก (รูปที่ 2.1,2.3.8) แล้วถอดแบตเตอรี่เก่าออก

(3) ใส่แบตเตอรี่ 9 V ก้อนใหม่ แล้วปิดฝาให้เรียบร้อย

2.7 วิธีการแก้ปัญหาเบื้องต้น

ในกรณีที่เกิดข้อบกพร่องขึ้น จะต้องพิจารณาดูว่าเกิดข้อบกพร่องขึ้นในส่วนของเครื่องหรืออีเล็คโทรดให้ทดลองเปลี่ยนอีเล็คโทรดอันใหม่แทนอันเก่า ถ้าเครื่องยังมีอาการเหมือนเดิม ให้ตรวจสอบทันที

A. เลื่อนสวิทช์ ON/OFF ไปที่ตำแหน่ง ON ปลดสาย PH อีเล็คโทรดออกจากตัวเครื่อง เครื่องจะต้องแสดงอาการอินพุทเกินคือ แสดงค่า "1" ที่ด้านซ้ายมือของจอแสดงผลเพียงตัวเดียว

B. ชอร์ทขั้วอินพุทของเครื่องด้วยสายไฟ เครื่องจะแสดงค่า ที่แน่นอนอยู่ระหว่าง 6 ถึง 8 ปรับ VR PH7 (CAL) ให้เครื่องแสดงค่า 7.00 ที่จุดนี้ ปุ่มชดเชยอุณหภูมิจะมีผลน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

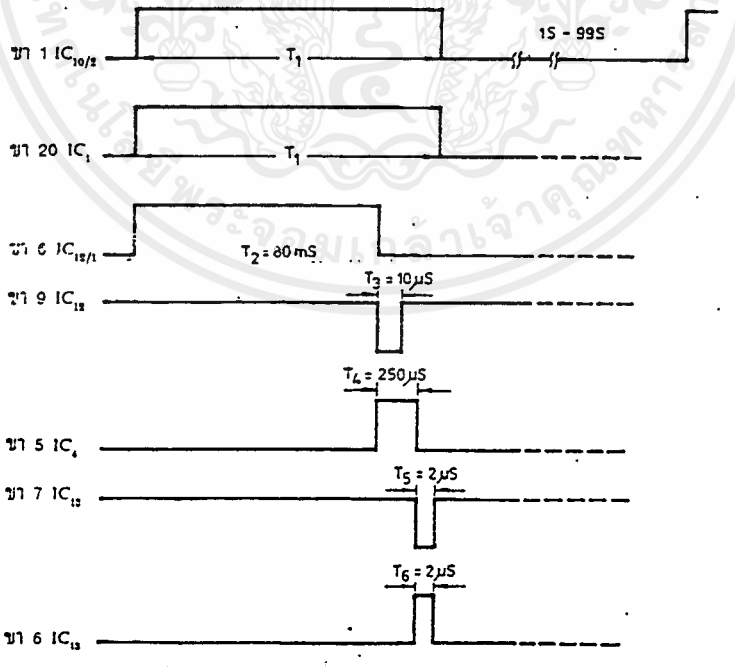
CALIBRATE AMP AND ANALOG TO DIGITAL CIRCUIT

3.1 หลักการทำงานของวงจร CALIBRATE AMP

สัญญาณจาก PH SENSOR จะนำมาเข้าวงจร Calibrate Amp เพื่อทำการปรับระดับของสัญญาณเพื่อให้ได้ระดับที่ต้องการ

วงจร Calibrate Amp ประกอบด้วยวงจร Buffer แบบละเอียดโดยใช้ TL 071 เพื่อจะได้ปรับค่า Voltage Drift ให้หมดไปจะได้สัญญาณที่ถูกต้องจากนั้นนำสัญญาณ มาเข้าวงจรขยายโดยใช้ TL 072 ซึ่งเป็นวงจรขยายแบบ NON - INVERTING ซึ่งจะทำการปรับ VR 100 กิโลโอห์ม เพื่อให้ได้ Gain = 10เท่า แล้วจึงนำสัญญาณนั้นมาเข้าวงจร Buffer อีกครั้งหนึ่ง เพื่อขยายสัญญาณ ก่อนจะนำมาเข้าวงจร A/D เพื่อเปลี่ยนสัญญานนี้ให้เป็นสัญญาณดิจิตอลต่อไป

3.2 วงจรแปลงอนาลอกเป็นดิจิตอล

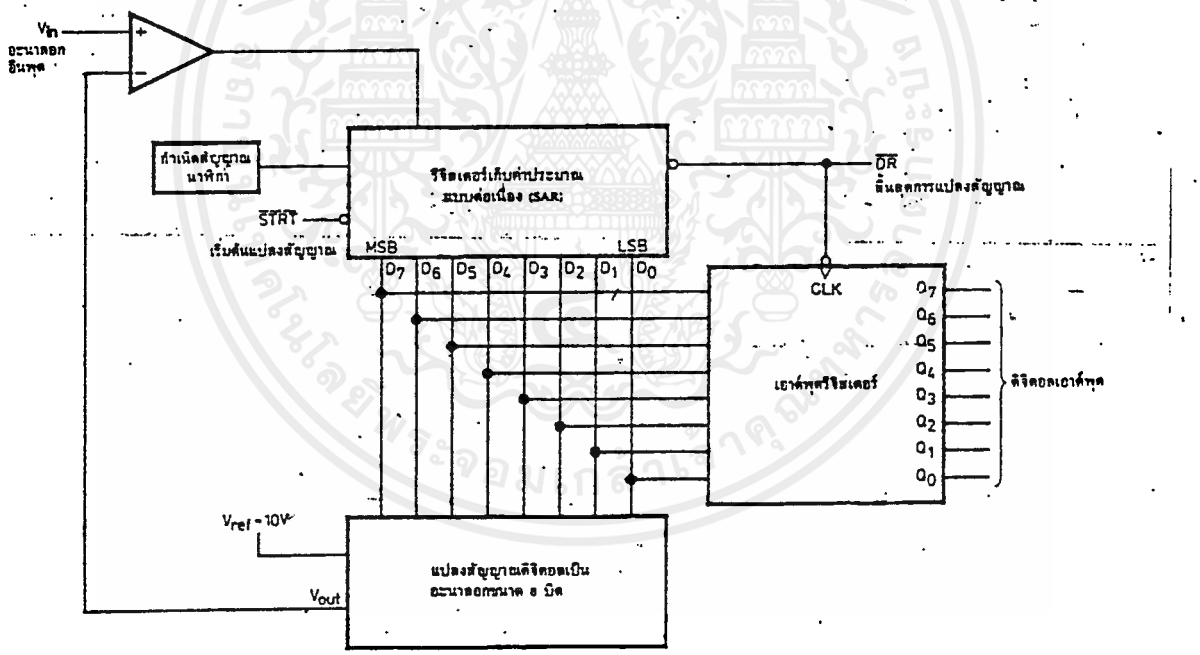


รูปที่ 3.1 แผนผังเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IC_1 เบอร์ ADC 0804 เป็นไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าอย่างต่อเนื่อง (SUCCESSIVE APPROXIMATION A/D)ซึ่งมีโครงสร้างเป็น CMOS ขนาด 8 บิท ไอซีเบอร์นี้จะมีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในโดยใช้ C_7 และ R_{11} เป็นตัวกำหนดคาบเวลา ความถี่ของวงจรผลิตความถี่ในวงจรนี้ออกแบบให้ผลิตความถี่ 600 กิโลเฮิรตซ์ โดยมีแรงดันที่ขา REF (ขา 9)เป็นตัวกำหนดแรงดันเต็มสเกลของ A/Dซึ่งจะต้องรักษาระดับแรงดันอ้างอิงที่ขานี้ไว้ที่ครึ่งหนึ่งของแรงดันเต็มสเกล ที่ต้องการเช่น รักษาแรงดันอ้างอิงไว้ที่ 2 โวลท์ โดยการใช้ ซีเนอร์ไดโอด จะได้แรงดันอินพุทเต็มที่เท่ากับ 4 โวลท์ (เอาท์พุทเป็น FF_h หรือ 11111111_b)

การแปลงสัญญาณแบบประมาณค่าอย่างต่อเนื่องนั้นมีการทำงานดังวงจรในรูปที่ 3.2 เป็นแผนผังวงจรการทำงานของ A/D

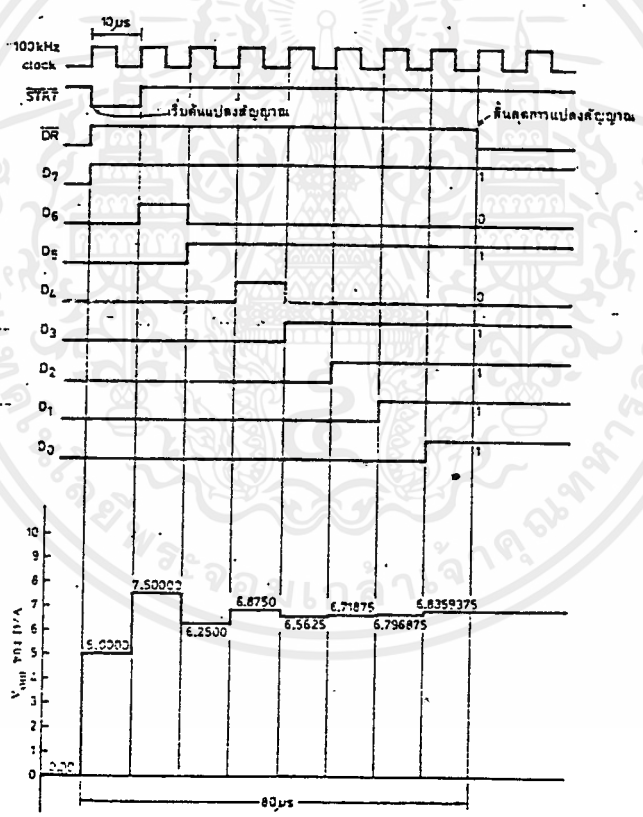


รูปที่ 3.2 แผนผังการทำงานของการแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032658

เริ่มต้นให้สัญญาณที่ขา STRT เป็นลอจิก "0" เข้ารีจิสเตอร์เก็บค่าประมาณอย่างต่อเนื่อง (SUCCESSIVE APPROXIMATION REGISTER ย่อว่า SAR) จะให้ขาเอาท์พุท D_7 (MSB) เป็น "1" นอกนั้นเป็น "0" หมด ดังนั้นภาคแปลงสัญญาณดิจิตอลเป็นอนาลอก (D/A) ก็จะแปลงสัญญาณดิจิตอลที่ได้ ไปเปรียบเทียบกับแรงดันอินพุทที่ส่งเข้ามา (D_7 จะให้แรงดันครึ่งหนึ่งของแรงดันเต็มสเกลของ D/A เมื่อแปลงเป็นสัญญาณอนาลอก)ถ้าแรงดันจาก D/A สูงกว่าแรงดันอินพุท SAR จะเปลี่ยน D_7 ให้เป็น "0" แต่ถ้าเอาท์พุทจาก D/A ต่ำกว่าแรงดันอินพุท SAR ก็จะ D_7 เป็น "1" ต่อไป

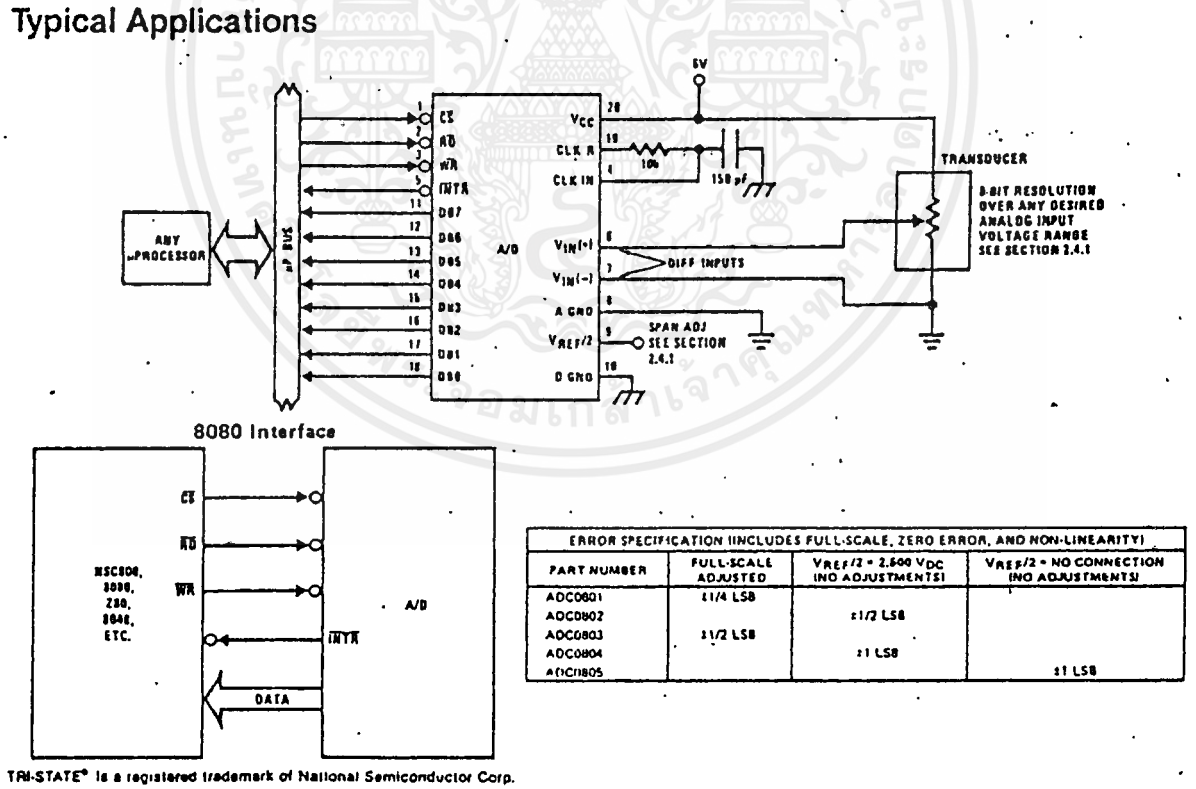


รูปที่ 3.3 แผนผังเวลาของ SAR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาก็จะให้บิทถัดไป (D_6) เป็น "1" แล้วแปลงสัญญาณดิจิตอลที่ได้เป็นสัญญาณอนาลอกไปเปรียบเทียบกับอินพุทอีกว่า สูงกว่าแรงดันอินพุทหรือเปล่า ถ้าสูงกว่าก็ให้ D_6 เป็น "0" ถ้าต่ำกว่าก็ให้เป็น "1" ขบวนการทำงานจะซ้ำไปอย่างนี้จนกระทั่งครบ 8 บิท เมื่อได้ข้อมูลครบทั้ง 8 บิทแล้ว SAR ก็จะให้ขา DR เป็น "0" เพื่อแสดงผลของข้อมูลออกมาจากเอาท์พุทรีจิสเตอร์เป็นสัญญาณดิจิตอล 8 บิท ทางขา Q จาก Q_0-Q_7 ดังรูปที่ 3.3 เป็นแผนผังเวลาของ A/D เมื่อแรงดันอนาลอกอินพุทมีค่า 6.4 โวลท์ และแรงดันอ้างอิงเป็น 10 โวลท์ ซึ่งจากรูปนี้ เราจะเห็นข้อดีของการแปลงสัญญาณแบบประมาณค่าอย่างต่อเนื่อง คือมีความเร็วในการแปลงสัญญาณ เพราะสัญญาณนาฬิกาไม่เกิน 8 ลูกก็จะแปลงสัญญาณเสร็จเรียบร้อย

ในการจัดรูปแบบของสัญญาณ ในการส่งข้อมูลจำเป็นที่จะต้องแปลงข้อมูลให้อยู่ในระบบดิจิตอล โดยใช้วงจร A/D



ERROR SPECIFICATION (INCLUDES FULL-SCALE, ZERO ERROR, AND NON-LINEARITY)			
PART NUMBER	FULL-SCALE ADJUSTED	$V_{REF}/2$ = 2.500 V_{DC} (NO ADJUSTMENTS)	$V_{REF}/2$ = NO CONNECTION (NO ADJUSTMENTS)
ADC0801	±1/4 LSB		
ADC0802		±1/2 LSB	
ADC0803	±1/2 LSB		
ADC0804		±1 LSB	
ADC1805			±1 LSB

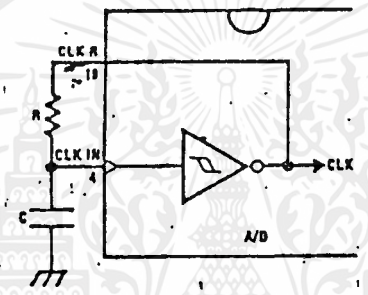
รูปที่ 3.4 การต่อ ADC 0804 เข้ากับระบบ MICROPROCESSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองนี้เราเลือกใช้ IC เบอร์ ADC 0804 เป็น IC A/D แบบ SUCCESSIVE APPROXIMATION ขนาด 8 บิตที่มี CONVERSIONTIME = 100 uS ต้องการแหล่งจ่ายไฟ + 5 V การปรับช่วงกว้างของแรงดันอินพุตที่จะป้อนให้ A/D ถูกกำหนด โดยขาVREF/2 และความถี่สัญญาณคล็อก ที่ป้อนให้ A/D ถูกกำหนดโดยขา 19 และขา 4 ซึ่งเป็น CLKRและ CLKIN เนื่องจาก IC เบอร์นี้มี TRI-STATE OUTPUT LATCH ในตัว ดังนั้น จึงไม่จำเป็นที่จะต่อ BUFFER การต่อ แสดงดังรูปที่ 3.4

3.3 การคำนวน ความถี่คล้อกของ A/D

ความถี่คำนวนได้จากสูตร $f_{clk} = 1/1.1\ RC$



รูปที่ 3.2 การคำนวนความถี่คล้อกของ A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การปรับ SPAN

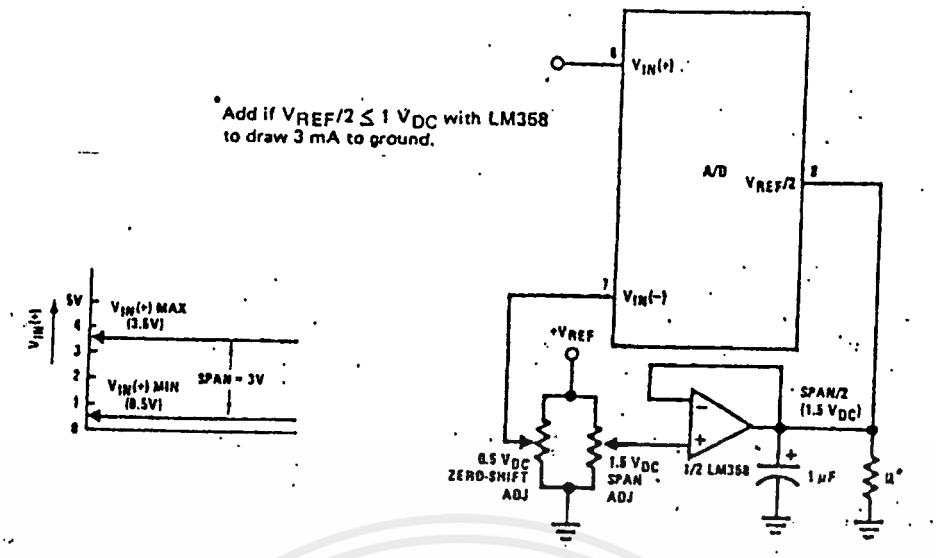
ขา V_{REF} จะเป็นขาที่กำหนดช่วงกว้างแรงดันอินพุท ดังแสดงในตารางที่ 3.1

$V_{ref/2}$ (V)	SPAN (V)
1	2
1.5	3
2	4
2.5	5

ตารางที่ 3.1 แสดงการปรับช่วงกว้างของแรงดันอินพุต โดยการปรับ $V_{REF/2}$ ค่าต่างๆ

ส่วนขา Vin (-) จะเป็นขาที่กำหนดแรงดันอินพุตต่ำสุดจากรูที่ 3.3 แสดงการปรับ $V_{ref/2}$ = 1.5 V และ Vin (-) เท่ากับ 0.5 V จะได้ SPAN = 3 V ดังนั้นจะได้แรงดันอินพุตอยู่ในช่วง 0.5 V ถึง 3.5 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



a) Analog Input Signal Example

b) Accommodating an Analog Input from 0.5V (Digital Out = 00_{HEX}) to 3.5V (Digital Out = FF_{HEX})

รูปที่ 3 ตัวอย่างการปรับ SPAN ของ A/D(เมื่อใช้ $V_{ref/2}$ =1.5 V) และ V_{in}(MIN =0.5 V)

3.5 การ START A/D

ทำได้โดยให้ขา CS และ WR เป็น " LOW "

3.6 การกระตุ้นให้ A/D ปล่อยข้อมูลออกมา

ทำได้โดยให้ขา CS และ RD เป็น " LOW " เนื่องจาก A/D ต้องการเวลาถึง100 US ในการ CONVERT,เมื่อกระทำเสร็จแล้วก็จะมีสัญญาณออกมาที่ขา INT เป็น " LOW ", A/D บางเบอร์เรียกขานี้ว่า EOC (END OF CONVERSION)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสาร เป็นส่วนสำคัญส่วนหนึ่งที่มีอยู่บนไมโครคอมพิวเตอร์ 16 บิต พอร์ตสื่อสารนี้ มีชื่อเรียกอีกอย่างหนึ่งว่า คอม-พอร์ต (com port) ผู้ออกแบบพอร์ตสื่อสารต้องการให้เป็นไปตามมาตรฐานการเชื่อมต่อแบบอนุกรมที่เรียกว่า RS232C พอร์ตนี้เป็นทางออกของข้อมูลที่ผู้ใช้สามารถส่งหรือรับกับระบบอื่นได้ พอร์ตสื่อสารจึงเป็นพอร์ตที่จำเป็นและผู้จัดซื้อไมโครคอมพิวเตอร์มักจะกำหนดอยู่ในสเปกด้วยเสมอ

พอร์ตสื่อสาร RS232C บนเครื่องไมโครคอมพิวเตอร์ 16 บิตมีโครงสร้างที่สามารถโปรแกรมด้วยการส่งรหัสคำสั่งให้กับชิปหลักได้ อย่างไรก็ตามผู้ออกแบบได้ให้ทางเลือกในการสื่อสารด้วยกระแสวนรอบ (current loop) หรือแบบแรงดันคือ RS232C โดยใช้จัมเปอร์เพื่อเลือกระบบ สำหรับตัว RS232C นี้เป็นมาตรฐานแบบอะซิงโครนัสที่โปรแกรมสตาร์ตบิต สต๊อปบิตและพาริตี้บิต อัตราการส่งก็สามารถกำหนดได้ตั้งแต่ 50 บอดถึง 9600 บอด ลักษณะพิเศษของวงจรคือสามารถส่งสัญญาณมาอินเตอร์รัพต์ซีพียูตามเงื่อนไขได้ และยังมีโครงสร้างฮาร์ดแวร์ป้อนกลับเพื่อใช้ในการตรวจสอบระบบว่าทำงานปกติดีหรือไม่ได้อีกด้วย วงจรพอร์ตสื่อสารของไมโครคอมพิวเตอร์ 16 บิตนี้ใช้ไอซีหมายเลข 8250 เป็นตัวสำคัญของระบบ 8250 เป็นไอซีขนาด 40 ขา มีขีดความสามารถพิเศษดังนี้

- มีบัฟเฟอร์ในตัวเพื่อทำให้ไม่จำเป็นต้องซิงโครนัสการรับส่ง
- ใช้สัญญาณนาฬิกาอิสระต่างหากไม่ขึ้นกับสัญญาณนาฬิกาของระบบ
- มีสัญญาณตอบโต้ควบคุมโมเดมทั้ง CTS (clear to send), RTS (request to send), DSR (data set read), DTR (data terminal ready),RC (ring indicator) และสัญญาณดีเทค ตัวพาหะ (carrier detect)

ตรวจสอบสตาร์ตบิตที่ผิดพลาด

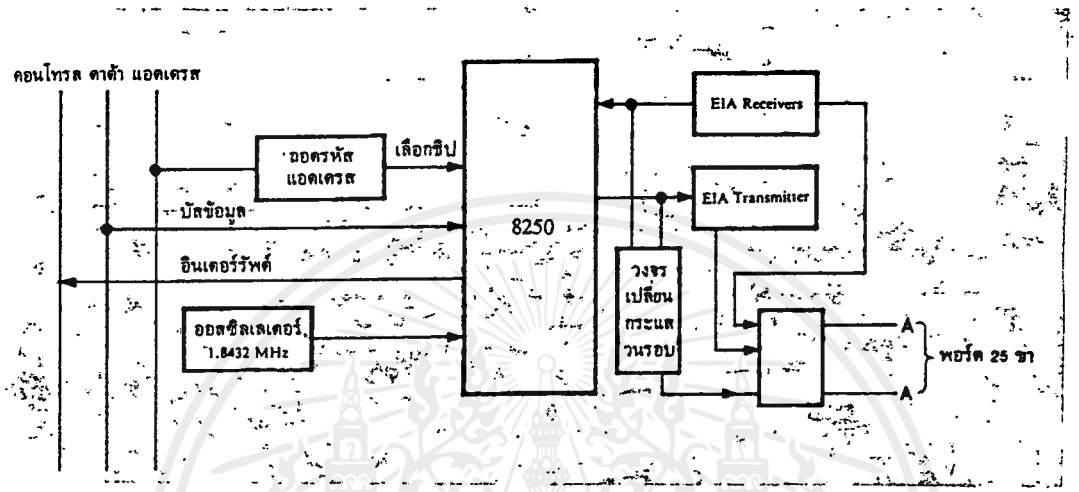
ตรวจสอบและสร้างสัญญาณสายขาด (line break) เพื่อใช้ในการตรวจสอบการทำงานของระบบ

ชิป 8250 มีโครงสร้างการทำงานคล้ายกับ 8251 ที่นักฮาร์ดแวร์ทั่วไปรู้จักดี การทำงานของ 8250 ต้องได้รับการโปรแกรมก่อน หลังจากนั้นจะทำงานตามรูปแบบที่ได้โปรแกรมไว้จนกว่าจะมีการโปรแกรมค่าใหม่ อย่างไรก็ตาม 8250 มีขีดความสามารถในการทำงานได้สูงกว่า8251 หลายอย่าง และด้วยเหตุนี้เอง 8250 จึงเป็นชิปประจำที่ใช้กับเครื่องไมโครคอมพิวเตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 โครงสร้างของพอร์ตสื่อสาร

พอร์ตสื่อสารของเครื่องไมโครคอมพิวเตอร์ 16 บิต ที่จะกล่าวถึงนี้เป็นระบบมาตรฐานตามแบบเครื่องไอบีเอ็มพีซีเอ็กซ์ที โครงสร้างบล็อกไดอะแกรมแสดงดังรูปที่ 4.1



รูปที่ 4.1 โครงสร้างพอร์ตสื่อสารข้อมูลที่ใช้ 8250

พิจารณาได้ว่า 8250 เป็นตัวรับข้อมูลจากบัสของระบบ ซึ่งก็คือสล็อตขนาด 31 x 2(62) ขาของระบบนั่นเอง ซีพียูติดต่อกับ 8250 ในลักษณะพอร์ตที่เป็นอินพุตเอาท์พุต การจัดพอร์ตนี้กำหนดหมายเลขพอร์ตอย่างเจาะจง ระบบไมโครคอมพิวเตอร์ 16 บิต มีพอร์ตสื่อสารสองพอร์ตคือ คอม 1 (COM_1) และคอม 2 (COM_2) ทั้ง COM_1 และ COM_2 มีหมายเลขอินพุตเอาต์พุตพอร์ต ดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 หมายเลขอินพุตเอาต์พุตพอร์ตของ COM_1 และ COM_2

อินพุตเอาต์พุตพอร์ต		เลือกรีจิสเตอร์	สถานะ DLAB
พอร์ต COM 1	พอร์ต COM 2		
3F8	2F8	บัฟเฟอร์ IX	DLAB = 0 (เขียน)
3F8	2F8	บัฟเฟอร์ RX	DLAB = 0 (อ่าน)
3F8	2F8	แลตช์ตัวหาร (LSB)	DLAB = 1
3F9	2F9	แลตช์ตัวหาร (MSB)	DLAB = 1
3F9	2F9	อีนาเบิลอินเตอร์รัพต์	
3FA	2FA	กำหนดอินเตอร์รัพต์	
3FB	2FB	ควบคุมสายสื่อสาร	
3FC	2FC	ควบคุมโมเดม	
3FD	2FD	แสดงสถานะสายสื่อสาร	
3FE	2FE	แสดงสถานะโมเดม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การกำหนดแอดเดรสสำหรับหมายเลขพอร์ตต่างๆ

แอดเดรส 3F8-3FE และ 2F8-2FE											
A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	DLAB	รีจิสเตอร์
1	1/0	1	1	1	1	1	X	X	X		
							0	0	0	0	บัฟเฟอร์สำหรับตัวรับข้อมูล(อ่าน)โฮลดิงสำหรับตัวส่งข้อมูล(เขียน)
							0	0	1	0	อีนาเบิลอินเตอร์รัพต์
							0	1	0	X	กำหนดอินเตอร์รัพต์
							0	1	1	X	ควบคุมสายสื่อสาร
							1	0	0	X	ควบคุมโมเดม
							1	0	1	X	แสดงสถานะสายสื่อสาร
							1	1	0	X	แสดงสถานะโมเดม
							1	1	1	X	ไม่ใช้
							0	0	0	1	แลตช์ตัวหาร (LSB)
							0	0	1	1	แลตช์ตัวหาร (MSB)

การเลือกหมายเลขอินพุตเอาต์พุตพอร์ตแยกเป็นสองกลุ่ม กลุ่มหนึ่งคือ COM_1 จะกำหนดหมายเลขพอร์ตจาก 3F8 ถึง 3FE อีกกลุ่มหนึ่ง ถ้ากำหนดหมายเลขพอร์ตเป็น 2F8-2FEในการเลือกหมายเลขรีจิสเตอร์ภายในกำหนดด้วยแอดเดรส 3 บิต คือ A_0,A_1 และ A_2 สำหรับการเลือก COM_1 และ COM_2 เราใช้แอดเดรส A_8 เป็นตัวเลือกการเลือกนี้ กำหนดเป็นตารางได้ดังตารางที่ 4.2

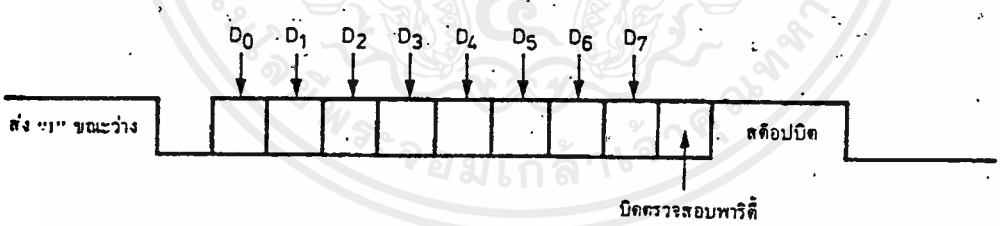
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การอินเตอร์รัพต์

จากที่เคยกล่าวถึงโครงสร้างการควบคุมอินเตอร์รัพต์ของชิป 8259 มาแล้ว 8259ได้จัดลำดับการอินเตอร์รัพต์ไว้ 8 ระดับสัญญาณรับอินเตอร์รัพต์ที่เข้าทางชิป 8259 มี 8 เส้น คือ IRQ_0-IRQ_7 สำหรับกรณีของระบบสื่อสารอนุกรมได้ กำหนดสัญญาณการอินเตอร์รัพต์ไว้แล้วคือให้ IRQ_4 เป็นสัญญาณอินเตอร์รัพต์ของระบบ COM_1 และ COM_3 เป็นของ COM_2 ในการที่จะ ส่งสัญญาณอินเตอร์รัพต์ต้องให้บิต 3 ของรีจิสเตอร์ควบคุมโมเดมได้รับการเซตค่าเป็น "1" ก่อนจากนั้นข้อมูลอินเตอร์รัพต์ที่อยู่ในรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์จะเป็นตัวสั่งการอินเตอร์รัพต์

4.3 รูปแบบข้อมูลที่รับหรือส่ง

การสื่อสารข้อมูลของระบบนี้เป็นการสื่อสารแบบอะซิงโครนัส รูปแบบของข้อมูลจะมีสตาร์ตบิต บิตตรวจสอบพาริตี้ และสต็อปบิตโครงสร้างของข้อมูลแต่ละเฟรมเป็นดังรูปที่4.2



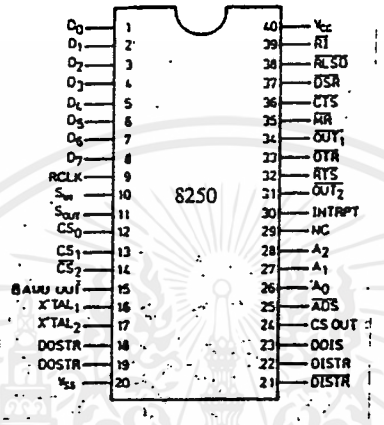
รูปที่ 4.2 รูปแบบข้อมูล 1 เฟรม

ข้อมูลบิตแรกของการส่งเป็นสตาร์ตบิต ระบบจะส่งสตาร์ตบิตก่อนหลังจากนั้นจะตามด้วยข้อมูลและแทรกด้วยบิต ตรวจสอบพาริตี้ตามด้วยสต็อปบิต ขนาดของข้อมูลมีค่าได้ตั้งแต่ 5-8บิต แต่ทั่วๆ ไปใช้ 8 บิต สต็อปบิตมีได้ 1, 1.5 หรือ 2 บิต ค่าเหล่านี้สามารถกำหนดลงไปในรีจิสเตอร์ควบคุมสายสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ชิป 8250

8250 เป็นไอซีขนาด 40 ขามีการทำงานเพื่อควบคุมสิ่งต่างๆ ที่เกี่ยวกับการสื่อสารแบบอนุกรมได้หมด โครงสร้างทางฮาร์ดแวร์ของอะแดปเตอร์การ์ดนี้จึงไม่ยุ่งยากมากนัก การจัดวางขาของไอซี 8250 มีรายละเอียดดังรูปที่ 4.3



รูปที่ 4.3 การจัดวางขาของไอซี 8250

ขาสัญญาณอินพุต ชิป CS_0, CS_1, CS_2 (chip select) คือ ขา 12, ขา 13,และขา 14 ตามลำดับ เป็นสัญญาณเลือกชิป โดยที่เมื่อต้องการเลือกชิปจะให้ CS_0 CS_1 เป็น "1" และ CS_2 เป็น "0" สัญญาณเลือกชิปนี้จะได้รับการเลือกแลตช์ไว้ในขณะที่สัญญาณ $\overline{ADS}$ มีค่าเป็น "0" การเลือกชิปนี้จะมีไว้เพื่อให้ซีพียูติดต่อกับ 8250

ขาสโตรบข้อมูลอินพุต (data input strobe) คือ DISTR (ขา 22) $\overline{DISTR}$(ขา 21) เมื่อสัญญาณที่ DISTR เป็น "1" และ DISTR เป็น "0" ในขณะที่มีการเลือกชิปเป็นขณะที่ซีพียูจะอ้างข้อมูลจากรีจิสเตอร์ภายใน ที่ได้รับการกำหนดไว้แล้วมายังซีพียู สัญญาณนี้จึงเป็นสัญญาณอ่านข้อมูลหรือ read นั่นเอง

ขาสโตรบข้อมูลเอาต์พุต คือ DOSTR (ขา 19) $\overline{DOSTR}$ (ขา 18) เป็นสัญญาณที่แอกทีฟขึ้นเพื่อให้ซีพียูเขียนข้อมูลลงมายังรีจิสเตอร์ของ 8250

ขาสโตรบแอดเดรส คือ $\overline{ADS}$ (ขา 25) เมื่อมีค่าเป็น "0" จะแอกทีฟเพื่อแลตช์ค่า A_0-A_1 เลือกรีจิสเตอร์ภายใน การเลือกรีจิสเตอร์จะทำขณะที่การเลือกชิปแอกทีฟอยู่

ขาเลือกรีจิสเตอร์ คือ A_0 (ขา 26), A_1 (ขา 27), A_2 (ขา 28)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามลำดับเป็นตัวกำหนดแอดเดรสของรีจิสเตอร์ภายใน เพื่อให้ซีพียูทำการติดต่อกับ 8250 ตามค่ารีจิสเตอร์ที่กำหนด เพื่อการเขียนหรืออ่าน อย่างไรก็ตามการทำงานของ A_0-A_2 นี้ก็ขึ้นกับสัญญาณเลือกตัวหาร LAB-drivisor latch access bit ซึ่งกำหนดค่ารีจิสเตอร์ได้ดังตารางที่ 4.3

ตารางที่ 4.3 การกำหนดค่ารีจิสเตอร์

DLAB	A	A	A	รีจิสเตอร์
0	0	0	0	บัฟเฟอร์สำหรับตัวรับข้อมูล (อ่าน)
0				โฮลดิงสำหรับตัวส่งข้อมูล (เขียน)
0	0	0	1	อีนาเบิลอินเตอร์รัพต์
X	0	1	0	กำหนดอินเตอร์รัพต์
X	0	1	1	ควบคุมสายสื่อสาร
X	1	0	0	ควบคุมโมเดม
X	1	0	1	แสดงสถานะสายสื่อสาร
X	1	1	0	แสดงสถานะโมเดม
X	1	1	1	ไม่ใช้
1	0	0	0	แลตช์ตัวหาร (LSB)
1	0	0	1	แลตช์ตัวหาร (MSB)

ขารีเซต (master reset) คือ MR (ขา 35) เมื่อมีค่าเป็น "1" จะรีเซตการทำงานของชิป 8250 โดยทำให้ค่าต่างๆ ในรีจิสเตอร์ถูกเคลียร์หมด (ยกเว้นบัฟเฟอร์ของตัวรับตัวส่งและตัวหาร) ขณะทำการรีเซตจะมีผลต่อสัญญาณเอาต์พุตด้วย ผลของการรีเซตแสดงไว้ในตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาสัญญาณนาฬิกาตัวรับ (receiver clock) คือ RCLK (ขา 9) เป็นขาที่ตัวรับสัญญาณนาฬิกาเพื่อกำหนดอัตราบอด สัญญาณนาฬิกานี้จะมีค่าเป็น16เท่าของที่นำมาใช้

ขาอินพุตข้อมูลอนุกรม (serial input) คือ SIN (ขา 10) เป็นขารับข้อมูลอนุกรมจากสายส่งในการเชื่อมโยงการติดต่อสื่อสาร

ขาเคลียร์ทูเซนด์ (clear to send) คือ $\overline{\text{CTS}}$ (ขา 36) เป็นสัญญาณที่ใช้ในการติดต่อกับโมเดม เงื่อนไขของสัญญาณนี้สามารถเก็บไว้ภายในชิป 8250 ที่จะให้ซีพียูอ่านไปตรวจสอบได้โดยเก็บไว้ที่บิต 4 ของรีจิสเตอร์ แสดงสถานะโมเดมส่วนบิตของรีจิสเตอร์แสดงสถานะจะเป็นตัวบอกว่า CTS ได้เปลี่ยนสถานะไปหลังจากการอ่านครั้งก่อนแล้วหรือไม่

ขาดาต้าเซตรีดี้ (data set ready) คือ $\overline{\text{DSR}}$ (ขา 37)เมื่อเป็น "0" จะแสดงว่าโมเดม หรือข้อมูลได้รับการเซตเตรียมพร้อมแล้ว สำหรับการเชื่อมต่อกับสายสื่อสาร และส่งข้อมูลระหว่าง 8250 กับโมเดมสัญญาณ DSR เป็นสัญญาณอินพุตของ 8250 ที่ซีพียูสามารถอ่านไปดูได้ทางบิตที่ 5 ของรีจิสเตอร์แสดงสถานะ ส่วนบิต 1 ของรีจิสเตอร์แสดงสถานะ เป็นตัวบอกว่า สัญญาณ DSR ได้เปลี่ยนสถานะไปหลังจากที่อ่านครั้งก่อนแล้วหรือไม่

หมายเหตุ ทั้ง $\overline{\text{CTS}}$ และ $\overline{\text{DSR}}$ เมื่อมีการเปลี่ยนสถานะและถ้าได้รับการอีนาเบิลที่ modem status interrupt จะส่งผลในการสร้างสัญญาณอินเตอร์รัพต์

ขาตรวจสอบสายสื่อสาร(received line signal detect) คือ $\overline{\text{RLSD}}$ (ขา 38) ถ้าเป็น "0" หมายถึงแอกทีฟ คือ 8250 รับสัญญาณตรวจสอบสัญญาณพาหะจากโมเดม ว่าโมเดมตรวจสอบได้แล้ว หรือข้อมูลได้รับการเซตแล้ว ซีพียูสามารถตรวจสอบสัญญาณนี้ทางบิต 7 ของรีจิสเตอร์แสดงสถานะ ส่วนบิต 3 จะเป็นบิตที่แสดงสถานะว่าสัญญาณนี้ได้รับการเปลี่ยนแปลงหลังจากอ่านไปแล้วหรือยัง

ขาแสดงวงจรเรียก (ring indicator) คือ RI(ขา 39)สัญญาณนี้แอกทีฟด้วยลอจิก"0"เป็นสัญญาณที่ส่งมาจากโมเดมโมเดมตรวจสอบสัญญาณการเรียก (ringing) สัญญาณนี้ตรวจสอบได้ทางบิต 6 และดูสถานะการเปลี่ยนหลังจากอ่านแล้วจากบิต 2

ขาไฟเลี้ยงคือ V_{cc} (ขา 40), V_{ss} (ขา 20) เป็นสัญญาณจากแหล่งจ่ายไฟเลี้ยง 5 โวลต์และกราวนด์

ขารีเควสต์ทูเซน(request to sent) คือ $\overline{\text{RTS}}$ (ขา 32)เริ่มขานี้มีลอจิกเป็น"0"หมายความว่า8250 พร้อมที่จะส่งข้อมูลแล้ว สัญญาณขานี้จะได้รับการเซตให้แอกทีฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยการโปรแกรมค่าลงไปในรีจิสเตอร์ควบคุมที่บิต 1

ขาเอาต์พุต 1 คือ OUT1(ขา 34) เป็นขาที่ผู้ใช้สามารถโปรแกรมให้แอกทีฟเป็น "0" ด้วยการโปรแกรมลงไปในบิต 2 ของรีจิสเตอร์ควบคุมโมเดม

ขาเอาต์พุต 2 คือ OUT2(ขา 31) เป็นขาที่ผู้ใช้สามารถโปรแกรมให้แอกทีฟเป็น "0" ด้วยการโปรแกรมค่าลงในรีจิสเตอร์ควบคุมโมเดมทางบิต 3

ขาเลือกชิปเอาต์(chip select out)คือ CSOUT (ขา 24)เมื่อมีค่าเป็น "1" จะบอกว่าชิปนี้ได้รับการเลือกชิปโดยซีพียูทางขา CS_0, CS_1 และ CS_2

ขาไดรฟ์เวอร์ดีสเอเบิล (driver disable) คือ DDIS (ขา 23)เป็นลอจิก"0"เมื่อซีพียูกำลังอ่านข้อมูลจาก 8250 สัญญาณ DDIS เป็น "1" มีไว้สำหรับการดิสเอเบิลการรับส่งภายนอก ในกรณีที่ใช้ 8250 กับซีพียูผ่านทางบิต D_0-D_7 เพื่อบอกเวลาว่าซีพียูเป็น 8250 ติดต่อกันอย่างไร

ขาสัญญาณกำหนดบอด คือ $\overline{\text{BAUDOUT}}$ (ขา 15) เป็นสัญญาณนาฬิกาที่มีความถี่เป็น16 เท่าของสัญญาณนาฬิกา แล้วหารด้วยค่าที่โปรแกรมกำหนดในตัวหาร

ขาอินเตอร์รัพต์ คือ INTRPT(ขา 30) เป็น "1"เป็นการส่งสัญญาณอินเตอร์รัพต์ออกไปจาก 8250

ขาข้อมูลเอาต์พุต คือ SOUT (ขา 11) เป็นขาที่ใช้ส่งข้อมูลอนุกรมออกไปยังสายสื่อสาร

ขาสัญญาณอินพุต/เอาต์พุตข้อมูล D_0-D_7 เป็นสัญญาณต่อเชื่อมกับบัสข้อมูลของระบบขาสัญญาณ $X'TAL_1$,$X'TAL_2$ คือ ขา 16, ขา 17 เป็นขาต่อกับคริสตอลเพื่อสร้างสัญญาณนาฬิกา

4.5 สถานะของ 8250 เมื่อเริ่มต้น

ก่อนการทำงาน หรือโปรแกรมเราควรจะทราบว่าสถานะต่างๆ ของ 8250 เป็นอย่างไร โดยเฉพาะอย่างยิ่งเมื่อเริ่มเปิดเครื่อง จะมีสัญญาณรีเซตซึ่งเป็นสัญญาณเดียวกับซีพียูมาทำการรีเซต 8250 ขณะนั้นจะมีสถานะเป็นอย่างไร เอาต์พุตของวงจรจะให้สัญญาณอะไร ตารางที่จะให้รายละเอียด 8250 ขณะเริ่มต้นมีดังตารางที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ค่าเริ่มต้นและเอาต์พุตของ 8250

รีจิสเตอร์/สถานะ	การควบคุม	สถานะเมื่อรีเซต
อินเตอร์รัพต์ริจิสเตอร์ภายใน	มาสเตอร์รีเซต	ทุกบิตเป็น "0"(0-3 ถูกกำหนด 4-7 จะเป็นถาวร)
รีจิสเตอร์กำหนดอินเตอร์รัพต์	มาสเตอร์รีเซต	บิต 0 เป็น "1"บิต 1-2 เป็น "0" บิต 3-7 เป็น "0" ถาวร
รีจิสเตอร์ควบคุมสายสื่อสาร	มาสเตอร์รีเซต	ทุกบิตเป็น "0"
รีจิสเตอร์ควบคุมโมเดม	มาสเตอร์รีเซต	ทุกบิตเป็น "0"
รีจิสเตอร์แสดงสถานะสายสื่อสาร	มาสเตอร์รีเซต	ยกเว้นบิต 5 และ 6 เป็น "1"
รีจิสเตอร์แสดงสถานะโมเดม	มาสเตอร์รีเซต	บิต 0-3 เป็น "0" บิต 4-7 เป็นสัญญาณอินพุต
SOUT	มาสเตอร์รีเซต	เป็น "1"
INTRPT (RCVR Errors)	Read LSR/ มาสเตอร์รีเซต	เป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 (ต่อ) ค่าเริ่มต้นและเอาต์พุตของ 8250

รีจิสเตอร์/สถานะ	การควบคุม	สถานะเมื่อรีเซต
INTRPT (RCVR Data Ready)	Read RBR/มาสเตอร์รีเซต	เป็น "0"
INTRPT (RCVR Data Ready)	Read IIR/Write THR/ มาสเตอร์รีเซต	เป็น "0"
INTRPT (เปลี่ยนสถานะโมเดม)	Read MSR/มาสเตอร์รีเซต	เป็น "0"
OUT2	มาสเตอร์รีเซต	เป็น "1"
RTS	มาสเตอร์รีเซต	เป็น "1"
DTR	มาสเตอร์รีเซต	เป็น "1"
OUT1	มาสเตอร์รีเซต	เป็น "1"

4.6 การต่อวงจรเข้ากับระบบ

บอร์ดอะแดปเตอร์สื่อสารนี้ มีโครงสร้างการเชื่อมต่อตามพอร์ต 3F8-3FE และ 2F8-2FE ดังนั้นจะใช้บิต A8 เป็นตัวเลือกบนบอร์ดจะมีจัมเปอร์ เพื่อจะบอกว่าเป็นพอร์ตสื่อสารแบบ COM_1 หรือ COM_2 การเลือกแอดเดรสใช้ 74LS30 ซึ่งเป็น NAND เกต แบบ 8 อินพุตมา เป็นตัวเลือก โดยมี U_{15} ในรูปเป็นตัวเลือก A_1 ดังได้กล่าวแล้ว

ส่วนของบัสข้อมูล D_0-D_7 จะผ่านบัฟเฟอร์ คือ 74LS245 ก่อนเข้าสู่ 8250 ส่วนสัญญาณควบคุมบน 8250 ทั้ง 10 มีดังนี้

ขามาสเตอร์รีเซต MR จะต่อโดยตรงกับสัญญาณรีเซตของระบบ

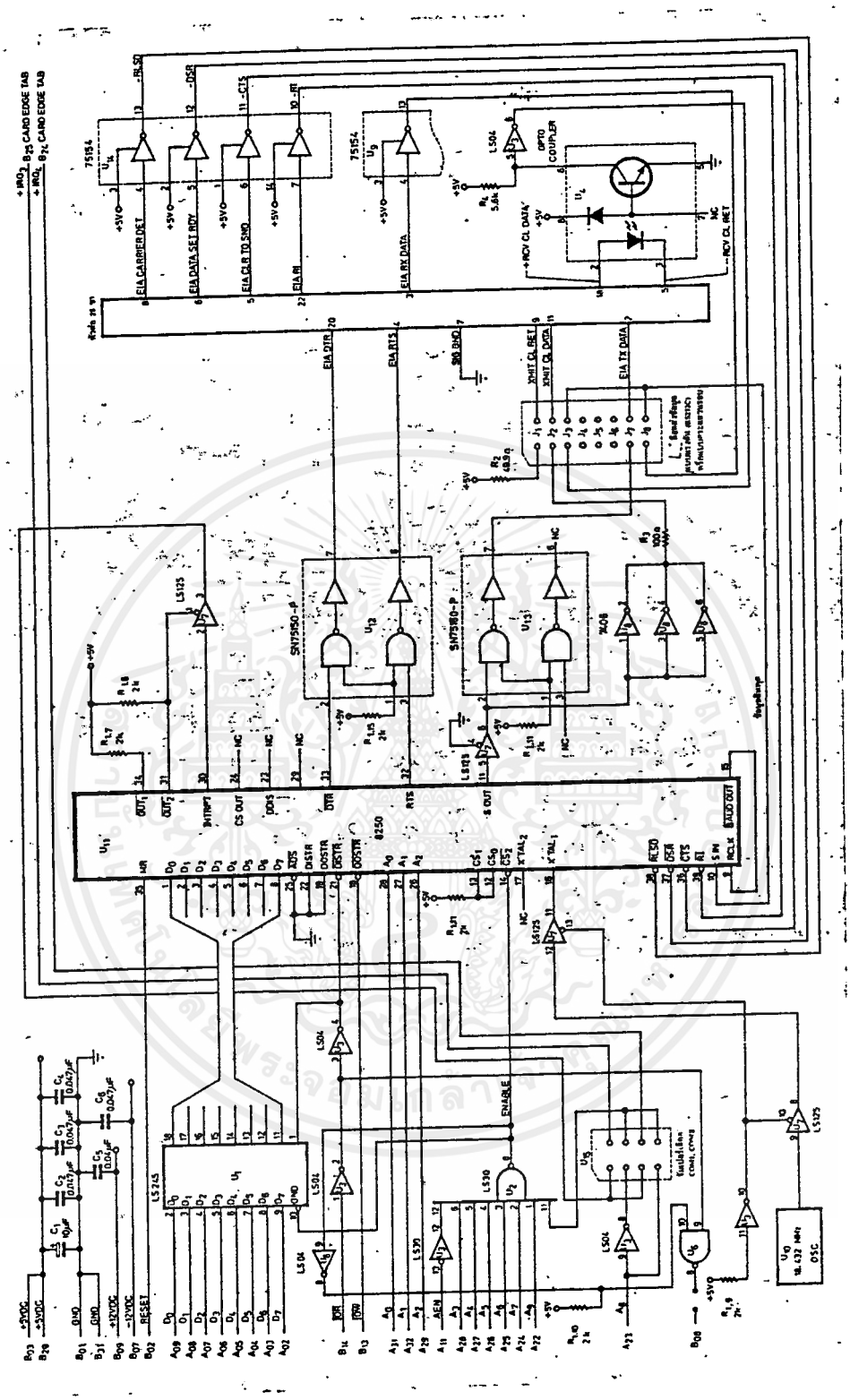
ขา $\overline{ADS}$, DISTR, DOSTR ต่อลงกราวนด์ทั้งนี้เพราะแอดเดรสที่ต่อมาที่ชิป 8250นี้เป็นสัญญาณแอดเดรสล้วนแล้วไม่ต้องสโตรบอีก

ขา $\overline{DISTR}$ ต่อกับ $\overline{IOR}$ ของระบบ โดยผ่านอินเวอร์เตอร์ 2 ตัว

ขา $\overline{DOSTR}$ ต่อกับ $\overline{IOW}$ ของระบบ

ขา CS_1 CS_2 ต่อขึ้นเป็นลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา $\overline{CS}_2$ มาจากการถอดรหัสให้เป็นพอร์ตของCOM_1หรือ COM_2 ตามต้องการ

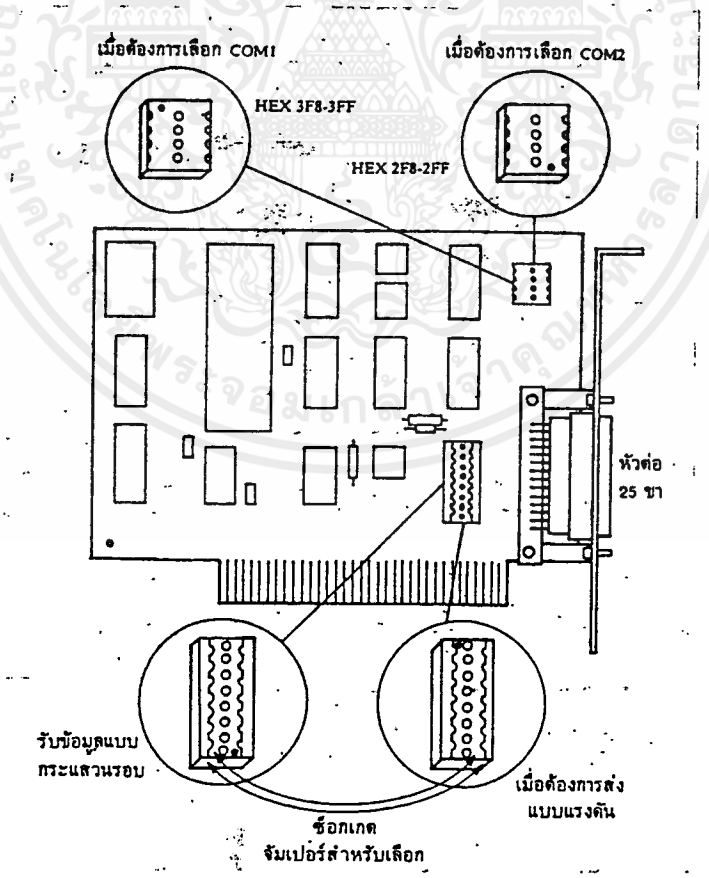
ขา $XTAI_2$ไม่ใช้ส่วน $XTAL_1$ ต่อมาจากออสซิลเลเตอร์ 18.432 MHzของระบบ

วงจรของบอร์ดอะแดปเตอร์สื่อสารเป็นดังรูปที่ 4.4

ส่วนของสัญญาณเอาต์พุตประกอบด้วยสัญญาณควบคุมโมเดม $\overline{RLSD}$, $\overline{DSR}$, $\overline{CTS}$, $\overline{RI}$ต่อออกไปยังหัวต่อตามมาตรฐาน EIA RS232

ขา SIN, SOUT ต่อออกไปขาเอาท์พุทเช่นกันแต่มีการปรับเป็นสัญญาณแรงดันตามมาตรฐาน EIA หรือเลือกส่งเป็นกระแสวนรอบก็ได้ การเลือกใช้จัมเปอร์ I_1-I_8 เป็นตัวเลือกการแปลงกระแสเป็นแรงดันใช้ออปโตคัปเปลอร์ดังรูปที่ 4.4

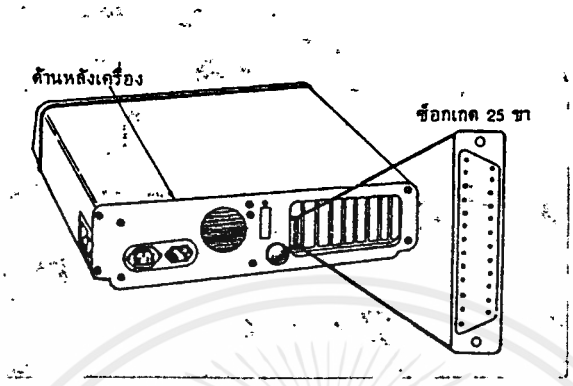
การเลือกจัมเปอร์บนบอร์ดอะแดปเตอร์นั้นผู้ออกแบบบอร์ดทำให้ง่ายต่อการใช้ด้วยการให้ จัมเปอร์มา เพียงผู้ใช้เลือกทิศทางก็จะได้พอร์ต COM_1 หรือ COM_2 หรือถ้าเลือกจัมเปอร์ อีกตัวก็เลือกการส่งแบบกระแสหรือแรงดันได้ ลักษณะของหัวต่อจัมเปอร์แสดงดังรูปที่ 4.5

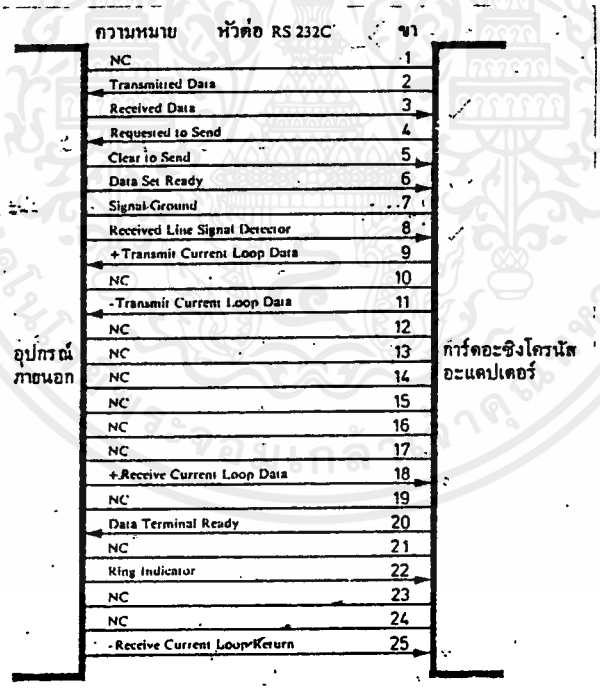


รูปที่ 4.5 การเลือกใส่จัมเปอร์เพื่อกำหนดวงจรตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาต่างๆ ของหัวต่อ 25 ขา RS232C ที่ใช้กันนั้นมีการจัดเรียงสัญญาณตามมาตรฐานสากล ขั้วต่อของสายแต่ละขาแสดงได้ดังรูปที่ 4.6





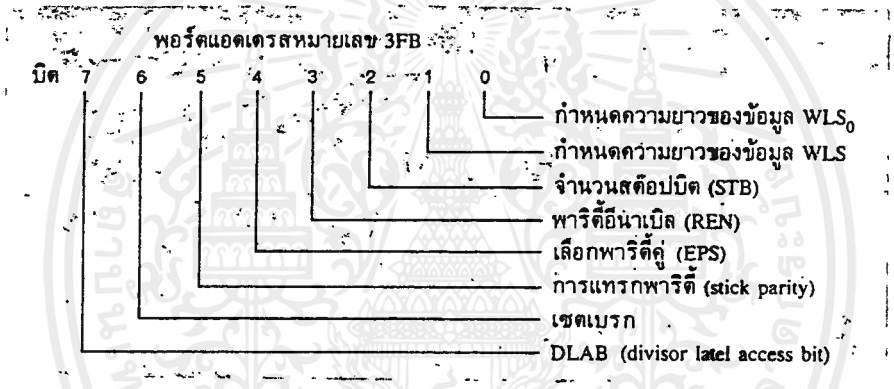
รูปที่ 4.6 วงจรต่างๆ ของหัวต่อ D-Shell 25 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การใช้งานรีจิสเตอร์ต่างๆ บน 8250

เพียงจากการใช้งานบอร์ดอะแดปเตอร์สื่อสาร ที่จะต้องโปรแกรมค่าไมโครโค้ดเข้าไปใน 8250 ก่อนดังนั้นผู้ใช้งาน 8250 จำเป็นต้องเข้าใจว่ารีจิสเตอร์ของ 8250 มีความหมายอย่างไร อธิบายได้ดังนี้

รีจิสเตอร์ควบคุมสายสื่อสาร (line control register) ในการควบคุมรูปแบบของข้อมูลแบบอะซิงโครนัสนั้น ผู้โปรแกรมจะต้องกำหนดค่าลงในรีจิสเตอร์ควบคุมสายสื่อสารรีจิสเตอร์ตัวนี้มี 8 บิต โดยแต่ละบิตมีความหมายดังนี้



รูปที่ 4.7 ค่าของบิตในรีจิสเตอร์ควบคุมสายการสื่อสาร

บิต 0 และ 1 เป็นตัวกำหนดความยาวของข้อมูลในการรับส่ง โดยที่

บิต 0	บิต 1	ความหมาย
0	0	หมายถึงข้อมูลขนาด 5 บิต
0	1	หมายถึงข้อมูลขนาด 6 บิต
1	0	หมายถึงข้อมูลขนาด 7 บิต
1	1	หมายถึงข้อมูลขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 2 เป็นบิตที่ใช้ในการกำหนดจำนวนสต๊อปบิต ถ้าเป็น "0" หมายถึงใช้สต๊อปบิต1 บิต แต่ถ้าบิต 2 เป็น "1" ในกรณีส่งแบบ 5 บิตจะมีความยาวของสต๊อปบิตเป็น 1.5 บิต แต่ถ้าส่งแบบ 6,7 หรือ 8 บิตความยาวของสต๊อปบิตจะเป็น 2

บิต 3 บิตนี้เป็นบิตแสดงการอีนาเบิลให้มีการตรวจสอบพาริตี้ โดยถ้าบิตนี้มีค่าเป็น 1จะมีการเพิ่มพาริตี้

บิต 4 มีค่าเป็น "0" และบิต 3 มีค่าเป็น "1" จะมีการกำหนดเป็นพาริตี้คี่ แต่ถ้าบิตนี้มีค่าเป็น "1" จะเป็นพาริตี้คู่

บิต 5 เมื่อบิต 3 มีค่าเป็น "1" และบิต 5 มีค่าเป็น "1" และบิต 4 มีค่าเป็น "1" จะมีการแทรกหรือตรวจสอบพาริตี้ (stick parity) ด้วยเงื่อนไขกำหนดให้เป็น "0" และถ้าบิต 4 มีค่าเป็น "0" บิต 3 มีค่าเป็น "1" และบิต 5 มีค่าเป็น "1" จะมีการกำหนดบิตพาริตี้เป็น "1"

บิต 6 เป็นบิตที่ควบคุมการเบรก เมื่อบิต 6 มีค่าเป็น "1" ส่วนของ SOUT จะได้รับการกำหนดให้เป็น "0" ตลอด

บิต 7 บิตนี้ทำหน้าที่เป็น DLAB บิตที่จะมีผลต่อการแลตช์ตัวหารดังที่กล่าวมาแล้วจากตารางที่ 4.1 และตารางที่ 4.2

การโปรแกรมอัตราบอด (boud rate generator) อัตราบอดได้รับการกำหนดเทียบกับสัญญาณนาฬิกา 1.8432 MHz และสามารถโปรแกรมตัวหารได้ตั้งแต่ 1-(2^{16}-1) ค่าความถี่เอาต์พุตของตัวกำหนดอัตราบอด มีค่าเท่ากับ 16 x อัตราบอด ดังนั้นตัวหาร = ความถี่สัญญาณนาฬิกา/(อัตราบอดx16) การกำหนดอัตราบอดด้วยการกำหนดตัวหารนี้ตัวหารจึงเป็นค่าที่กำหนดในรีจิสเตอร์ 2 ตัว ตัวหารนี้จะต้องถูกกำหนดค่าก่อนแล้วโปรแกรมลงมาในรีจิสเตอร์นี้การกำหนดต้องให้ DLAB = 1 แล้วให้ลดลงมาในรีจิสเตอร์ $3F_8$ ซึ่งเรียงกันเป็น LSB ของตัวหาร ส่วน $3F_9$ เมื่อ DLAB = 1 จะเป็นค่าของตัวหาร MSB ค่าของตัวหารเมื่อเทียบกับสัญญาณ 1.8432 MHz เป็นดังตารางที่ 4.5

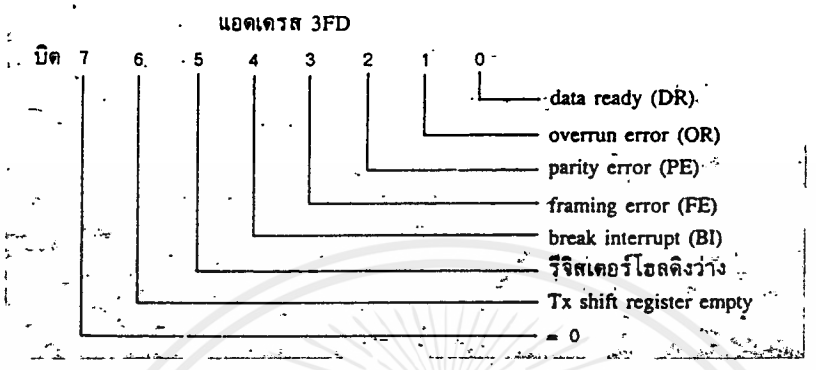
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ค่าตัวหารสำหรับการกำหนดอัตราบอด

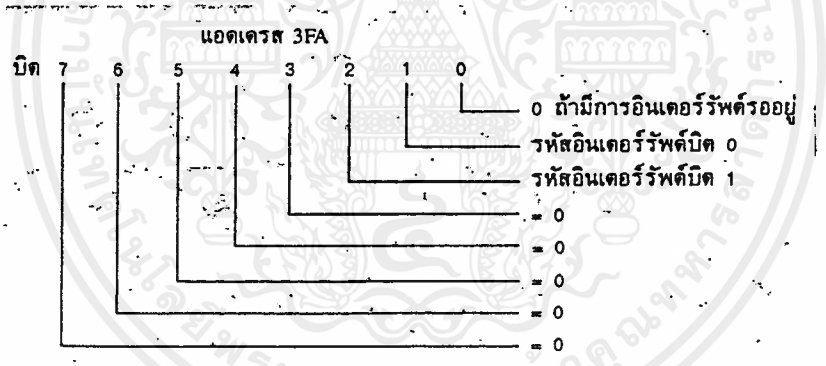
อัตราบอด	ตัวหาร		ค่าผิดพลาด
	ฐานสิบ	ฐานสิบหก	
50	2304	900	-
75	1536	600	-
110	1047	417	0.026
134.5	857	359	0.058
150	768	300	-
300	384	180	-
600	192	0C0	-
1200	96	060	-
1800	64	040	-
2000	58	03A	0.69
2400	48	030	-
3600	32	020	-
4800	24	018	-
7200	16	010	-
9600	12	00C	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์แสดงสถานะสายสื่อสาร (line status register)รีจิสเตอร์ตัวนี้เป็นรีจิสเตอร์ที่จะให้ข้อมูลแก่ซีพียูที่เกี่ยวกับการสื่อสารข้อมูลในสายสื่อสาร ค่าของบิตต่างๆ ในรีจิสเตอร์นี้เป็นดังนี้



รูปที่ 4.8 ค่าของบิตในรีจิสเตอร์แสดงสถานะสายสื่อสาร



รูปที่ 4.9 ค่าของบิตในรีจิสเตอร์กำหนดอินเตอร์รัพต์

บิต 0 บิตนี้เป็นบิตที่บอกสถานะการรับข้อมูล ถ้าบิตนี้เป็น "1" แสดงว่าการรับข้อมูลเข้ามาในบัฟเฟอร์ได้ครบทุกบิตแล้ว บิตนี้จะได้รับการรีเซตให้เป็น "0" เมื่อซีพียูได้อ่านข้อมูลในบัฟเฟอร์ไปแล้ว หรือจะให้ซีพียูเขียนข้อมูลกลับมายังรีจิสเตอร์นี้ก็ได้

บิต 1 บิตนี้ถ้ามีค่าเป็น "1" แสดงว่าเกิด overrun error (OR)กล่าวคือขณะที่มีข้อมูลที่บัฟเฟอร์แต่ซีพียูยังไม่ได้อ่านไป ปรากฏว่ามีข้อมูลชุดใหม่มาเขียนทับบนบัฟเฟอร์นี้ บิตนี้จะรีเซตโดยซีพียู เมื่อซีพียูอ่านค่าจากรีจิสเตอร์นี้ไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 2 บิตนี้ถ้ามีค่าเป็น "1" แสดงว่าเกิด parity error (PE) กล่าวคือ ถ้ามีการตรวจสอบบิตพาริตี้แล้วไม่เป็นไปตามที่กำหนดไว้ บิตนี้จะได้รับการรีเซตโดยซีพียู เมื่อซีพียูอ่านค่าจากรีจิสเตอร์นี้ไปแล้ว

บิต 3 บิตนี้ถ้ามีค่าเป็น "1" แสดงว่าเฟรมของข้อมูลไม่เป็นไปตามที่กำหนด เช่นตรวจสอบจำนวนบิตโดยดูที่พาริตี้และสต๊อปบิตไม่เป็นไปตามที่กำหนด

บิต 4 บิตนี้เรียกว่า break interrupt (BI) บิตนี้จะได้รับการเซตให้มีค่าเป็น"1" ถ้าหากว่ารับข้อมูลอินพุตเป็น "0" เป็นเวลายาวนานกว่าเวิร์ดของการสื่อสาร

บิต 5 บิตนี้เป็นบิตที่บอกว่า 8250 พร้อมที่จะรับข้อมูลจากสายสื่อสาร บิตนี้จะได้รับการเซตให้มีค่าเป็น "1" บิตนี้ยังคงสร้างสัญญาณอินเตอร์รัพต์ เพื่อส่งไปบอกซีพียูด้วย บิตนี้จะมีสถานะเซตเมื่อมีการส่งถ่ายข้อมูลจากโฮลด์รีจิสเตอร์ไป ยังชิปรีจิสเตอร์เพื่อพร้อมที่จะส่ง

บิต 6 เป็นบิตที่จะบอกว่าชิปรีจิสเตอร์ว่างเปล่า บิตนี้จะได้รับการเซต ให้มีค่าเป็น "1" เพื่อบอกว่าพร้อมส่งแล้ว

บิต 7 จะเป็น "0" ตลอด

รีจิสเตอร์กำหนดอินเตอร์รัพต์ (IIR-interrupt identification register)ไอซี8250มีขีดความสามารถในการส่งอินเตอร์รัพต์ภายในชิปเพื่อว่าให้การทำงานระหว่าง 8250 กับซีพียูเป็นไปอย่างมีประสิทธิภาพสูง และเพื่อให้ผู้เขียนซอฟแวร์สามารถเขียนซอฟต์แวร์ได้ง่ายและสั้นลงได้มาก 8250 กำหนดความสำคัญของอินเตอร์รัพต์ไว้ 4 ระดับคือ ระดับแรกสถานะ การรับข้อมูลจากสายสื่อสาร ระดับที่สอง -การพร้อมรับข้อมูล ระดับที่สาม-ขณะรีจิสเตอร์โฮลดิงสำหรับส่งข่าว ระดับที่สี่-สัญญาณสถานะโมเดม

ในขณะที่มีความต้องการอินเตอร์รัพต์หลายระดับพร้อมกัน 8250 จะให้ระดับที่มีความสำคัญน้อยกว่ารอไว้ก่อน โดยเก็บสถานะการอินเตอร์รัพต์นี้ไว้ในรีจิสเตอร์ กำหนดอินเตอร์รัพต์ความหมายของรีจิสเตอร์นี้มีดังนี้

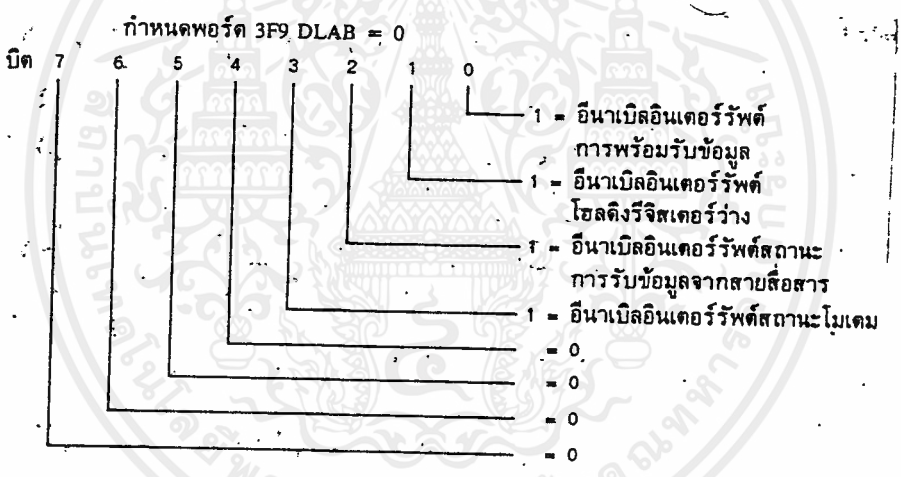
บิต 0 เป็นบิตที่ใช้แสดงว่ามีอินเตอร์รัพต์เกิดขึ้นหรือไม่ ซึ่งสามารถให้ซีพียูตรวจสอบดูด้วยวิธีการ polling ได้ ถ้าบิตนี้เป็น "1" หมายถึง ไม่มีอินเตอร์รัพต์เกิดขึ้น

บิต 1-2 เป็นบิตที่แสดงความหมาย บอกว่าการอินเตอร์รัพต์ที่เกิดขึ้นนั้นมาจากการอินเตอร์รัพต์ตามฟังก์ชันใด

บิต 3-7 มีค่าเป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์อีนาเบิลอินเตอร์รัพต์(INTRPT-interrupt enable register) ในCOM$_1$ เมื่อให้ DLAB = 0 พอร์ต $3F_9$ จะเป็นรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์ผู้ใช้สามารถกำหนดให้เกิดอินเตอร์รัพต์หรือไม่ก็ได้โดยการกำหนดค่าลง ในรีจิสเตอร์นี้ จากที่กล่าวแล้วว่าการอินเตอร์รัพต์ของ 8250 นี้มี 4 แบบดังนั้นจึงต้องกำหนดการอีนาเบิลได้ทั้ง 4 แบบ โดยการใช้ข้อมูลแต่บิตของรีจิสเตอร์นี้เพื่อการกำหนดการอีนาเบิล ข้อมูลที่อยู่ในรีจิสเตอร์นี้มีความหมายดังนี้



รูปที่ 4.10 ค่าของบิตในรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ฟังก์ชันการอินเตอร์รัพต์รหัสอินเตอร์รัพต์

บิต 2	บิต 1	บิต 0	ระดับความสำคัญ	ชนิดของอินเตอร์รัพต์	แหล่งเกิดอินเตอร์รัพต์	การรีเซตควบคุมอินเตอร์รัพต์
0 1	0 1	1 0	สูงสุด	ไม่เกิดสถานะการรับข้อมูลจากสายสื่อสาร	ไม่เกิด overrun error parity error framming error break interrupt	อ่านข้อมูลจากรีริสเตอร์สถานะสายสื่อสาร
1	0	0	ที่สอง	การพร้อมรับ	มีข้อมูลที่ตัวรับ	การอ่านข้อมูลจากบัฟเฟอร์
0	1	0	ที่สาม	โฮลดิงรีจิส-เตอร์สำหรับส่งข่าว	โฮลดิงรีจิสเตอร์สำหรับส่งข่าว	อ่านรีจิสเตอร์กำหนดอินเตอร์รัพต์ IIR หรือเขียนลงไปยังโฮดดิงรีจิสเตอร์สำหรับส่ง
0	0	0	ที่สี่	สถานะโมเดม	CLS DSR RI ตรวจสอบสายส่งโดยตรง	อ่านรีจิสเตอร์แสดงสถานะของโมเดม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 0 บิตนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์การพร้อมรับข้อมูล

บิต 1 บิตนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์โฮลดิงรีจิสเตอร์ว่าง

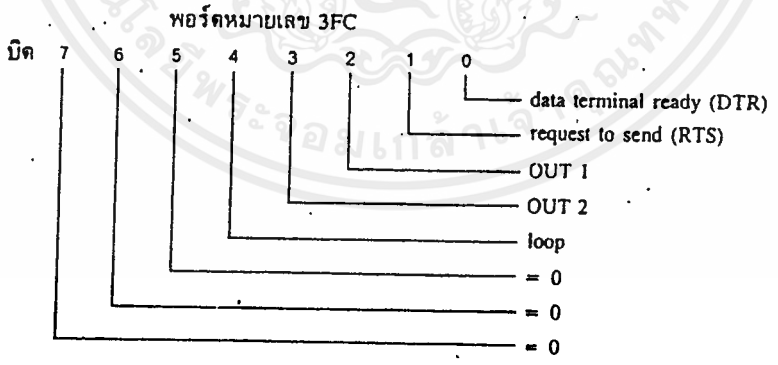
บิต 2 บิตนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์จากสถานะ การรับข้อมูลจากสายสื่อสาร

บิต 3 บิตนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์จากสถานะโมเดม

บิต 4-7 ได้รับการกำหนดให้เป็น 0 เสมอ

รีจิสเตอร์ควบคุมโมเดม (modem control register) รีจิสเตอร์ตัวนี้มีไว้ให้ซีพียูส่งผ่านข้อมูลมาเก็บ เพื่อเป็นรหัสสำหรับควบคุมการทำงานของโมเดม การกำหนดพอร์ตของรีจิสเตอร์ตัวนี้คือ 3FC ข้อมูลต่างๆ ที่มีในรีจิสเตอร์ตัวนี้มีความหมายดังนี้

บิต 0 บิตนี้มีความหมายถึงการควบคุมสัญญาณ $\overline{DTR}$ เมื่อบิตนี้มีค่าเป็น "1" เอาต์พุตที่ $\overline{DTR}$ จะได้รับการกำหนดให้เป็น "0" และถ้าบิตนี้มีค่าเป็น "0" เอาต์พุตที่ $\overline{DTR}$ จะได้รับการกำหนดให้เป็น "1"



รูปที่ 4.11 ค่าของบิตในรีจิสเตอร์ควบคุมโมเดม

บิต 1 บิตนี้มีความหมายถึงสัญญาณRTSซึ่งจะมีผลเหมือนกับบิต0ในกรณีของDTR

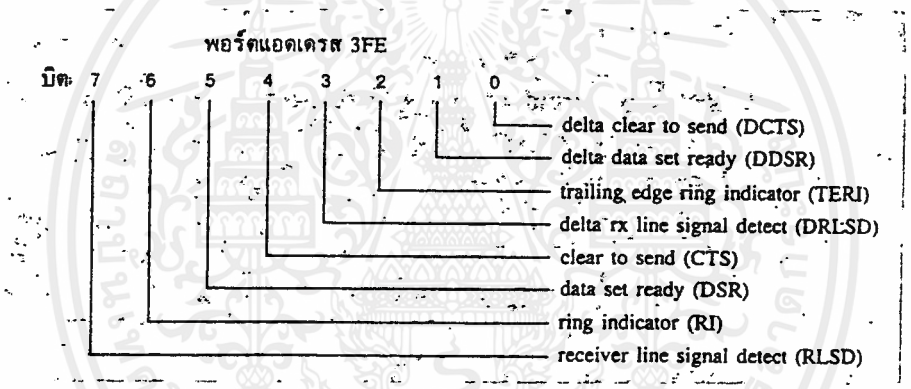
บิต 2 บิตนี้ใช้ควบคุมขาเอาต์พุต 1 (OUT 1) ซึ่งจะมีผลเหมือนบิต 0

บิต 3 บิตนี้ใช้ควบคุมเอาต์พุต 2 (OUT 2) ซึ่งจะมีผลเหมือนบิต 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 4 บิตนี้จะใช้สำหรับการกำหนดวงรอบสำหรับการตรวจสอบ 8250 เมื่อบิต4 นี้ได้รับการเซตเป็น "1" สิ่งที่จะเกิดขึ้นเป็นดังนี้ข้อมูลที่ SOUT จะได้รับการเซตให้เป็นลอจิก "1" ขาข้อมูลอินพุต SIN จะได้รับการแยกตัวออก ข้อมูลของเอาต์พุตชิปรีจิสเตอร์จะได้รับการป้อนกลับมายังรีจิสเตอร์ข้อมูลอินพุตส่วนสัญญาณ CTS, DSR, RLSD และ RIจะได้รับการแยกออกจากระบบแต่สัญญาณควบคุมโมเดม คือ DTR, RTS, OUT 1 และ OUT 2 จะต่อเข้ากับสัญญาณทั้งสี่ที่เป็นอินพุต ดังนั้นจึงตรวจสอบระบบการทำงานได้

รีจิสเตอร์แสดงสถานะโมเดม รีจิสเตอร์ตัวนี้จะเป็นตัวที่รับสถานะจากโมเดมมาเก็บไว้เพื่อให้ซีพียูสามารถอ่านตรวจสอบดูได้ สถานะของข้อมูลจะแอกทีฟเมื่อมีข้อมูลเป็น "1" และจะได้รับการรีเซตเมื่อซีพียูอ่านข้อมูลในรีจิสเตอร์นี้ไป พอร์ตที่ใช้กำหนดเป็นพอร์ตหมายเลข 3FE ข้อมูลภายในรีจิสเตอร์นี้เป็นดังนี้



รูปที่ 4.12 ค่าของบิตในรีจิสเตอร์แสดงสถานะโมเดม

บิต 0 บิตนี้ใช้สำหรับแสดงการเปลี่ยนแปลงของสัญญาณ CTSกล่าวคือเมื่อเขา CTSของ 8250 ได้เปลี่ยนสถานะหลังจากที่ซีพียูได้อ่านสถานะนี้ไปแล้ว บิตนี้ก็จะบอกด้วยเซต และเมื่อซีพียูอ่านก็จะได้รับการรีเซต "0" และจะได้รับการเซต "1" เมื่อมีการเปลี่ยนสถานะที่ขา CTS

บิต 1 เหมือนบิต 0 แต่เป็นบิตที่แสดงสถานะ การเปลี่ยนแปลงของขา DSR

บิต 2 บิตนี้เป็นบิตแสดงว่าสัญญาณ RI ซึ่งเป็นอินพุตของ 8250 ได้รับการเปลี่ยนจากออน "1" มาเป็นออฟ "0"

บิต 3 บิตนี้เหมือนบิต 0 แต่เป็นบิตแสดงสถานะการเปลี่ยนแปลงของ line signaldetector ซึ่งเป็นขาอินพุต RLSD

บิต 4 บิตนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา CTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 5 บิตนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา DSR

บิต 6 บิตนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา RI

บิต 7 บิตนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา RLSD

อนึ่งถ้าบิต 4 ของ MCR ได้รับการเซตหรือให้ทำลูป (loop) ตรวจสอบข้อมูลในบิต4 จะเหมือนกับ RTS ใน MCR ข้อมูลในบิต 5 จะเหมือนกับ DTR ใน MCR ข้อมูลใน บิต 6 จะเหมือนกับ OUT_1 ใน MCR ข้อมูลในบิต 7 จะเหมือนกับ OUT_2 ใน MCR

รีจิสเตอร์บัฟเฟอร์สำหรับตัวรับข้อมูล (receiver buffer register) เป็นรีจิสเตอร์สำหรับการรับข้อมูลที่มาจาก สายสื่อสารสัญญาณ พอร์ตที่กำหนด คือ หมายเลขแอดเดรส 3F8 ขณะที่ DLAB = 0 หากซีพียูอ่านข้อมูลที่รีจิสเตอร์นี้ก็หมายถึงได้อ่านข้อมูลที่มาจากสายสัญญาณสื่อสารนั่นเอง

รีจิสเตอร์โฮลดิงสำหรับตัวส่งข้อมูล(transmitter holding register) เป็นรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล รีจิสเตอร์นี้จะรับข้อมูลจากซีพียู โดยที่กำหนดพอร์ตเป็นหมายเลข 3F8 เมื่อ DLAB = 0 ข้อมูลของซีพียูที่เอาต์พุตมาที่พอร์ตนี้ก็ เพื่อจะส่งต่อออกไปยังสายสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การศึกษาโปรแกรม C ที่ใช้ส่งข้อมูลแบบอะซิงโครนัส

(C PROGRAMMING FOR ASYNCHRONOUS SERIAL TRANSMISSION)

การสื่อสารแบบ Serial synchronous นั้น ข้อมูลจะถูกส่งผ่าน Serial Port บิทต่อบิท ซึ่งจะต่างจากการสื่อสารที่ผ่าน Parallel port ที่จะมีการส่งข้อมูลที่ละไบท์ และคำว่า อะซิงโครนัสนั้น เนื่องจากไม่สนใจ จังหวะเวลาในการส่งข้อมูล อย่างไรก็ตามความยาวของข้อมูลจะถูกกำหนด โดยแต่ละไบท์ของข้อมูลประกอบด้วย

1.บิทเริ่มต้น 1 บิท

2.บิทข้อมูล 7 หรือ 8 บิท

3.พาริตี้บิท (มี หรือ ไม่มี)

4.บิทสุดท้าย 1 หรือ 2 บิท

สภาวะสายว่างนั้น แรงดันจะสูง เมื่อส่งบิท "0" แรงดันจะต่ำและ "1" แรงดันจะสูงนั่นคือ เมื่อส่งสัญญาณบิทเริ่มต้นนั้น จะขับให้แรงดันสายต่ำลง จากนั้นข้อมูลก็จะถูกส่ง และจะตามด้วยพาริตี้บิท และการส่งบิทสุดท้าย ส่งจะขับให้แรงดันสายต่ำลง

พาริตี้บิทจะใช้ในการตรวจสอบความผิดพลาดในการส่ง อาจจะใช้พาริตี้คู่หรือคี่ ค่าพาริตี้จะถูกเช็คให้ค่าไบท์ ที่ส่งรวมกับค่าพาริตี้บิทนั้น ออกมาเป็นเลขคี่

สำหรับการสื่อสารนี้จะผ่าน RS232 ซึ่งจะประกอบด้วยตัวคอนเนคเตอร์25ขา ในแต่ละปลาย ซึ่งขาที่สำคัญในการสื่อสารมีดังนี้

สัญญาณ	ชื่อย่อ	ขาของคอนเนคเตอร์
Reques to send	RTS	4
Clear to send	CTS	5
Data set ready	DSR	6
Data terminal ready	DTR	20
Transmit data	TxD	2
Recieve data	RxD	3
Ground	GRD	7

การแอดเซส Serial Port โดยอาศัย BIOS บน PC นั้น สามารถทำได้โดยการใช้ DOS หรือ ROM-BIOS หรือผ่าน DOS และ BIOS โดยการควบคุมฮาร์ดแวร์ การแอดเซสโดยใช้ DOS นั้นไม่ใช้วิธีที่ดีเนื่องจาก DOS ไม่ได้เตรียมการป้อนกลับสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ PORT นั่นคือมันจะเตรียมการเขียนและอ่าน โดยไม่มีการตอบรับ วิธีที่ดีที่สุดก็คือการ Interrupt ROM-BIOS

ROM-BIOS ได้เตรียมบริการในการแอสเซสไว้ 4 อย่าง โดยการ Interrupt 14H ดังนี้

5.1 Port Initialization

ก่อนที่จะมีการใช้งาน Serial port เราจำเป็นที่จะต้องมีการปรับค่าเริ่มแรกโดยการ Interrupt 14H และ Service ที่ "0" ดังเช่นการ Interrupt อื่นๆ Register AL จะใช้สำหรับเก็บค่า Service AL จะใช้เก็บค่าเริ่มแรก ซึ่งจะแปลงให้อยู่ในรูปข้อมูล 1 ไบท์ดังนี้

สำหรับค่า BAUD นั้นแสดงดังตารางที่ 1และ PARITY BITแสดงตารางที่ 2

ตารางที่ 1 แสดงการเข้าระหัสบิทที่ 5,6,7 เพื่อให้ได้ความเร็ว ที่ระบุไว้

BAUD	BIT PATTERN
9600	1 1 1
4800	1 1 0
2400	1 0 1
1200	1 0 0
600	0 1 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

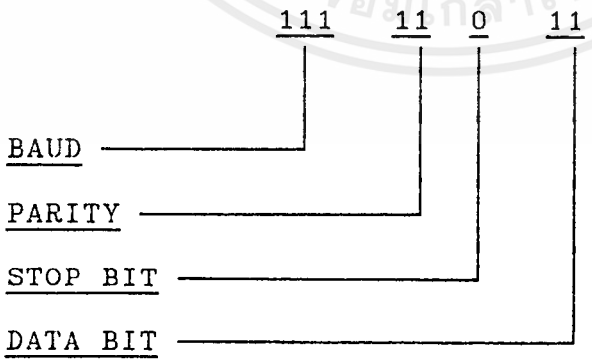
300	0 1 0
150	0 0 1
110	0 0 0

ตารางที่ 2 ใช้ในการเข้ารหัสบิทที่ 3,4 เพื่อแทนค่า ที่ระบุไว้

PARITY	BIT PATTERN
NO PARITY	00 หรือ 10
ODD	01
EVEN	11

สำหรับจำนวนของ STOP BIT จะตัดสินใจโดยที่ 2 ถ้าบิทที่ 2 = 1 จะใช้ 2 บิทในการหยุด และบิทที่ 2 = 0 จะใช้ 1 บิทในการหยุดและสุดท้ายเป็น DATA BITซึ่งแสดงโดยบิทที่ 0 และ 1 อย่างไรก็ตาม จะใช้ 10 แทน 7 ข้อมูล 7 บิท และ 11ใช้แทนข้อมูล 8 บิทเท่านั้น

ตัวอย่างเช่น ต้องการเซตพอร์ท ให้มีการส่ง 9600 baud, even pority มี stop bit = 1, data bit = 8 เราจะได้ค่าในเลขฐานสิบ = 251



จะมี 7 ที่ใช้ใน PC และต้องการใช้ PORT ใดนั้นจะระบุใส่ในรีจิสเตอร์ DX โดย PORT แรกหมายเลข = 0 ที่ 2 = 1 และฟังค์ชั่นที่ใช้ในการนี้เราเรียกว่า init-port() ซึ่งเขียนเป็นโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /* initialize the port */
void port_init(port,code)
        int port;
unsign character code;
{
        union REGS r;
r.x.dx = port  ;/*serial port*/
r.h.ah = 0     ;/*initialize port function*/
r.h.al = code  ;/*initialization code*/
int 86 (0x14,&r,&r)
}
```

5.2 การส่งข้อมูล

interrupt 14H และ source ที่ 1 จะใช้ในการส่งข้อมูล 1 ไบท์ ผ่าน SERIAL PORT ที่ระบุไว้ใน DX และข้อมูลที่จะส่งจะต้องเก็บไว้ใน AL สำหรับสภาวะการส่งจะถูกเก็บไว้ใน AH และฟังค์ชั่น sport() จะใช้ในการส่งข้อมูล 1 ไบท์ผ่าน PORT ที่ระบุไว้

```
        /*Send a character out the serial port */
        void sport(port,code)
        int port;/*I/O port*/
        char c;/*7 character to send */
{
        union REGS r;
r.x.dx = port  ;/*serial port*/
r.h.ah = 1     ;/*send character function*/
r.h.al = 0     ;/*char to send*/
int 86 (0x14,&r,&r)
if (r.h.ah & 128) { /*check bit 7*/
printf("send error detected in serial port.")
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
exit(1);
}
}
```

5.3 Checking a port status

interrupt 14H และ source ที่ 3 จะใช้สำหรับตรวจสอบสภาวะของ port ที่ต้องการตรวจสอบจะถูกระบุใน DX โดยที่ AH กับ AL จะระบุสภาวะของ PORT นั้นตามตารางที่ 3

ตารางที่ 3 แสดงสภาวะของ serial port

สำหรับเงื่อนไขหนึ่งที่สำคัญที่สุดก็คือ DATA READY ซึ่งจะเป็นตัวตัดสินว่า ข้อมูลได้ถูกรับแล้ว และกำลังจะถูกอ่านและฟังก์ชั่น rport() ใช้สำหรับอ่านข้อมูลจาก port

TEBLE 3 THE STATUS OF A SERIAT PORT

LINE STATUS (AH)	BIT
MEANING WHEN SET	
DATA READY	0
OVERRUN ERROR	1
PARITY ERROR	2
FRAMING ERROR	3
BREAK-DETECT ERROR	4
TRANSFER HOLDING REGISTER EMPTY	5
TRANSFER SHIFT REGISTER EMPTY	6
TIME-OUT ERROR	7

MODEM STATUS (AL)	
Modem Status (AL)	
Meaning when set	Bit
Change in clear-to-send	0
Change in data-set-ready	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Trailing-edge ring detector	2
Change in line signal	3
Clear-to-send	4
Data-set-ready	5
Ring indicator	6
Line signal detected	7

5.4 การรับข้อมูล

BIOS interrupt 14H, source 2 ใช้ในการอ่านข้อมูลจาก serial port โดยที่ serial port ที่ใช้จะถูกระบุไว้ใน DX และอักษรที่อ่านได้จะถูกเก็บไว้ใน AL และจากการส่งอักษรนั้นสภาวะของบิท 7 ใน AH แสดงถึงว่าการส่งนั้นสำเร็จหรือผิดพลาด

ฟังก์ชั่น rport() จะใช้ในการอ่าน byte ข้อมูลจาก port ที่ระบุไว้

```
/* Read a character from a byte */
rport(port)
int port;/* i/o port */
{
   union  REGS  r;
/* wait for a character */
while(!(check_statat(PORT)&256))
if(kbit()){/*abort on keypress */
getch();
exit(1);
}
r.x.dx = port; /*serial port */
r.h.ah = 2; /*read character function */
int86(0x14,&r,&r);
if(r.h.ah&128)
printf("read error detected in serial port");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
return r.h.al ;
}
```

การอ่านจะรอจนกว่าสามารถรับข้อมูลจาก serial port อย่างไรก็ตาม เพื่อหลีกเลี่ยงปัญหาความผิดพลาด ฟังก์ชั่นนี้จะประกอบด้วย การอ่านสภาวะของบิท 7 ใน AH และถ้าหากมีการกดคีย์ฟังก์ชั่นจะสิ้นสุด อย่างไรก็ตาม เมื่อรับข้อมูลเข้ามาแล้ว จะถูกเรียก และข้อมูลจะถูกอ่าน และบิทของข้อมูลจะถูกตรวจสอบว่า การทำงานสำเร็จหรือไม่

การส่งไฟล์ข้อมูลระหว่างคอมพิวเตอร์

- Handshaking ทาง Software

หลักการ: คอมพิวเตอร์ด้านส่งจะส่งไบท์แรกและรอจนกว่าคอมพิวเตอร์ด้านรับจะส่งไบท์ตอบรับกลับ และเมื่อได้รับไบท์ตอบรับแล้ว คอมพิวเตอร์ด้านส่งก็จะส่งไบท์ต่อไป และรอไบท์ตอบรับอีก ครั้งหนึ่งวิธีการนี้จะกระทำไปจนหมดไฟล์

```
Send()
{
  While (There are byte to send){
  send( a byte);
  wait()
}
Receive()
do {
  Receive_byte();
  Send(acknowledgement);
  } while(there are still byte to read);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
วงจรที่ใช้งาน

ในบทนี้จะได้กล่าวถึง การนำวงจรต่างๆ มาใช้ใน PH RECORDER ซึ่งส่วนวงจรนี้ จะประกอบกันขึ้นเพื่อที่จะให้การดำเนินการ บันทึกผลและแสดงผลของ PH RECORDER ให้ถูกต้องและมีประสิทธิภาพ ในการทำงาน ดังจะกล่าวถึงแต่ละส่วนดังนี้

CALIBRATE AMP CIRCUIT

วงจรภาคนี้มีหน้าที่ ในการปรับระดับสัญญาณ จาก PH METER ให้เหมาะสมกับ Input ของวงจร A/D เริ่มจากวงจร Buffer ที่ใช้ TL 071 เป็นวงจร Buffer ชนิดละเอียด Input มีความเที่ยงตรงสูง โดยสามารถปรับ Voltage Drift ได้ เพื่อให้มีค่าผิดพลาดน้อยที่สุด จากนั้นจึงนำมาเข้าวงจรขยาย 10 เท่า ต่อจากนั้นจึงนำมาเข้า Buffer อีกครั้งหนึ่ง ก่อนจะไปเข้าภาค A/D

อัตราขยาย (Gain) ขอองวงจรคำนวณได้จากสูตร

$$Gain = 1 + Rf/Rin$$

เมื่อ

$$Rf = 200 \text{ กิโลโอห์ม}$$

$$Rin = VR\ 100 \text{ กิโลโอห์ม}$$

A/D CIRCUIT

วงจรภาคนี้ มีไว้สำหรับแปลงค่าสัญญาณอนาลอก เป็นสัญญาณดิจิตอล 8 Bit ใช้ IC เบอร์ ADC 0804 เป็นตัวแปลงสัญญาณ โดยปรับให้มีย่านของ Input มีค่าตั้งแต่ 0-140 mV นั่นคือ สัญญาณ Full scale ซึ่งมีค่า 140 mV จะเท่ากับข้อมูล FF_H จากนั้นจึงนำ Output ที่ได้ไปผ่าน Buffer เพื่อส่งข้อมูลไปยัง Data bus สัญญาณที่ใช้กระตุ้น A/D ให้ทำการ Converse สัญญาณ เราใช้สัญญาณ Time base 1 Hz ซึ่งจะกล่าวถึงภาคต่อไป

OSCILLATOR CIRCUIT (1 Hz)

วงจรภาคนี้ เป็นตัวกำหนดสัญญาณ Time base 1Hz ที่จะนำไปกระตุ้นขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Write ของ A/D เพื่อให้ทำการ Converse สัญญาณ วงจรเริ่มจาก IC time base 60 Hz เบอร์ MM 5369 ซึ่งใช้ร่วมกับ Crystal ความถี่ 3.579545 MHz ร่วมกับ R_1, R_2 และ C_1,C_2 ได้ Output 60 Hz ที่ขา 1 จากนั้นจึงนำสัญญาณ 60 Hz มาเข้าวงจรหาร 6 โดยใช้ IC 4017 เป็นวงจรนับหาร จึงได้สัญญาณความถี่ 10 Hz ออกมาที่ขา 12 ของ IC 4017 จากนั้นจึงนำมาเข้าวงจรหาร 10 โดยใช้ IC 4017 อีกตัวหนึ่ง Output ที่ได้ จึงได้ความถี่ 1 Hz ที่จะนำไปใช้งาน

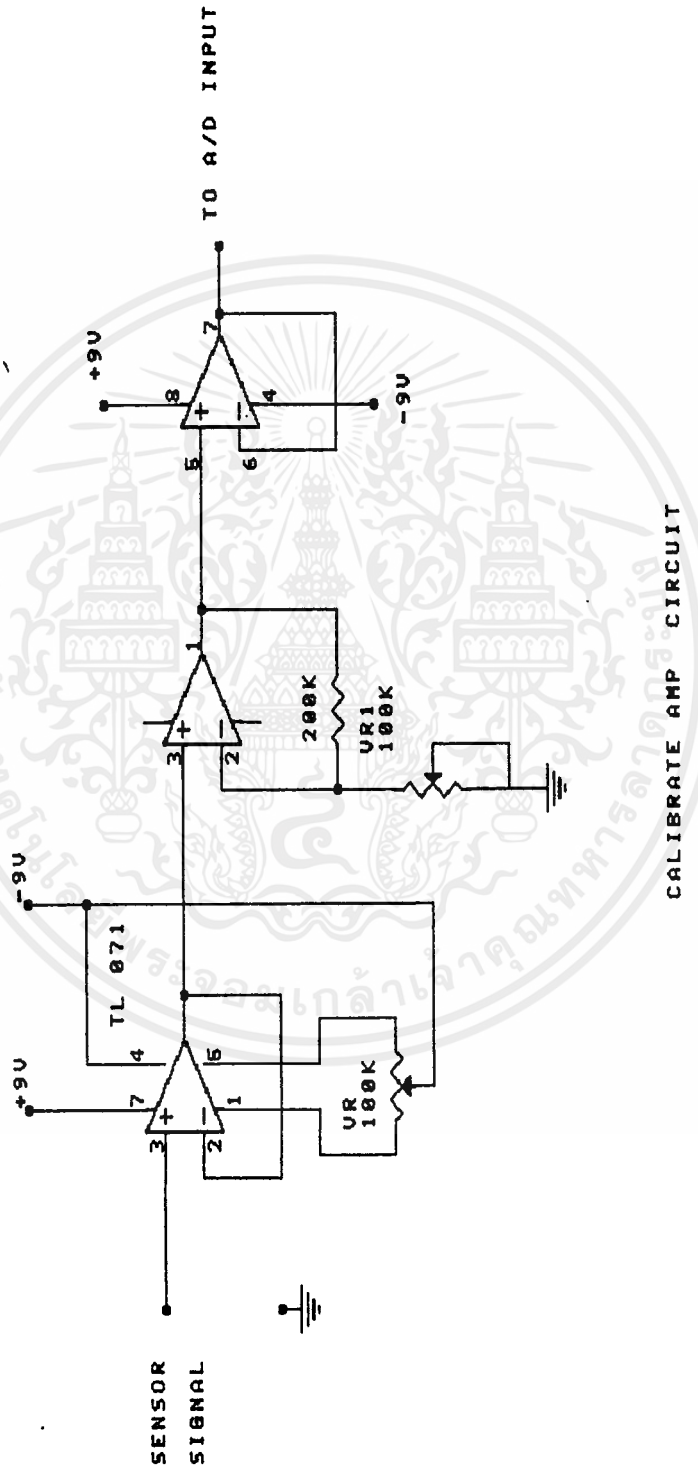
INTERFACE CIRCUIT

วงจรภาคนี้ เราใช้ IC เบอร์ UM 8250 เป็นตัวจัดรูปแบบข้อมูล ในรูปแบบข้อมูลอนุกรม การใช้งาน 8250 นี้ การเซ็ตค่าเราเซ็ตโดยใช้ Dip switch ตั้งค่าที่ต้องการ ทั้ง Data bus และ address bus จากนั้นจึงผ่าน Buffer เพื่อเข้าไปยัง Data bus และ address bus การกำหนดค่าที่จะเซ็ตต้องกำหนด แลตช์ตัวหารทั้ง DLL และ DLM เพื่อกำหนด อัตราความเร็วในการส่งข้อมูล และต้องเซ็ตค่า LCR (Line control register) เพื่อกำหนดรูปแบบของข้อมูล ซึ่งค่าต่างๆ ที่ต้องการจะเซ็ตนั้นดูได้จาก บทที่ 4 สำหรับสัญญาณนาฬิกา ที่ป้อนให้กับ 8250 เราใช้ IC 7404 ร่วมกับ Crystal ความถี่ 1.8432 MHz ส่วน Output จากขา 11 จะผ่านวงจร Driver เบอร์ MC 1488 เพื่อ Interface เข้ากับ port RS 232

POWER SUPPLY CIRCUIT

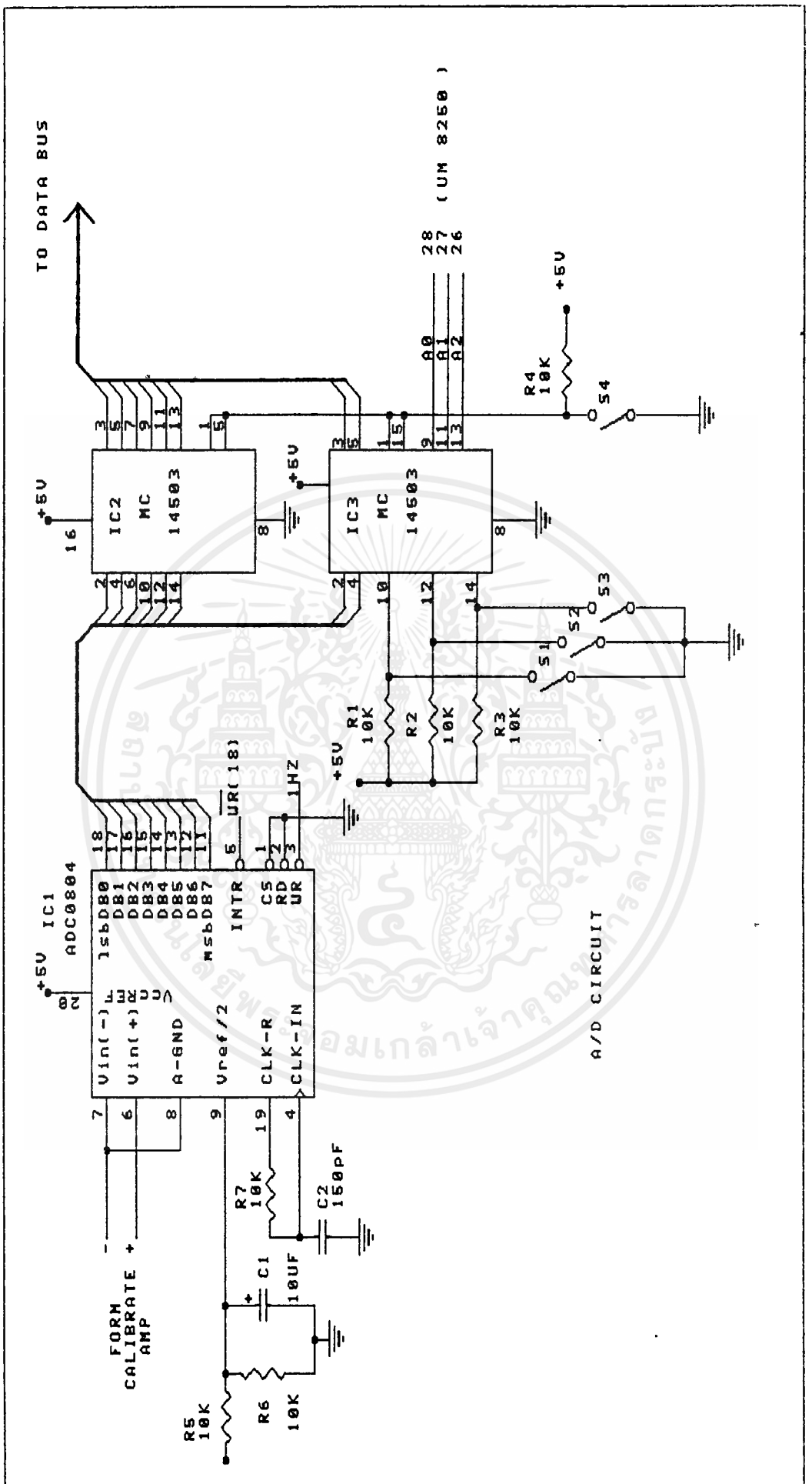
เป็นแหล่งจ่ายไฟของวงจร จากไฟ AC 220 V จะมาผ่าน Transformer เพื่อที่จะทำการแปลงลง(step down) ให้เป็น 9-0-9 V จากนั้นผ่านวงจร Bride rectifier เพื่อจัดเรียงกระแส เป็นแบบ fullwave rectifier และจะมี Capaciter เพื่อจะทำให้กระแสเรียบขึ้น จนเป็นกระแสตรง แล้วก็จะมาเข้า MC 7805T,'UA 78L09 CF, 7909 ซึ่งเป็น IC Regulater เพื่อที่จะปรับให้ แรงดันออกมาคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

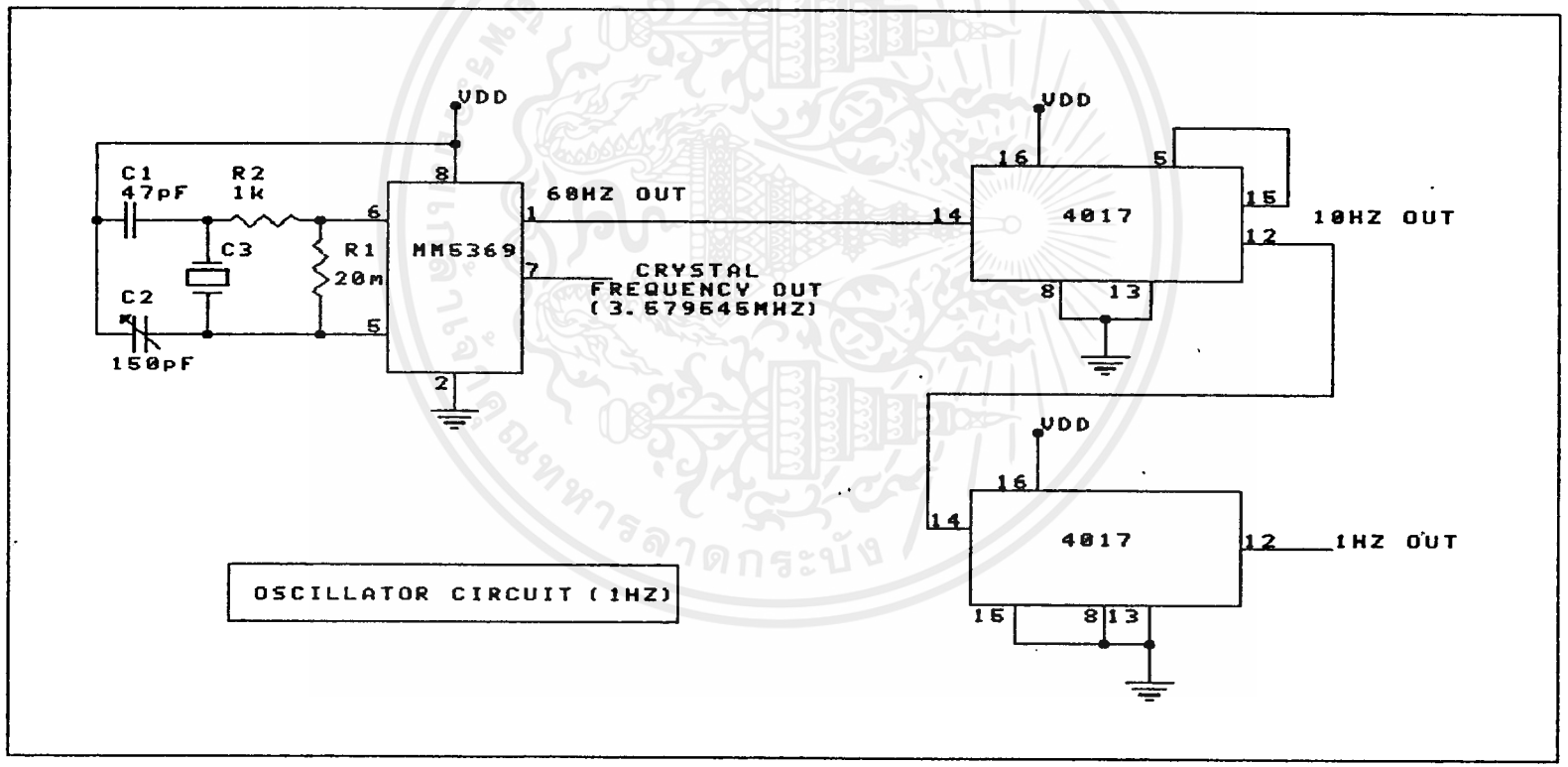


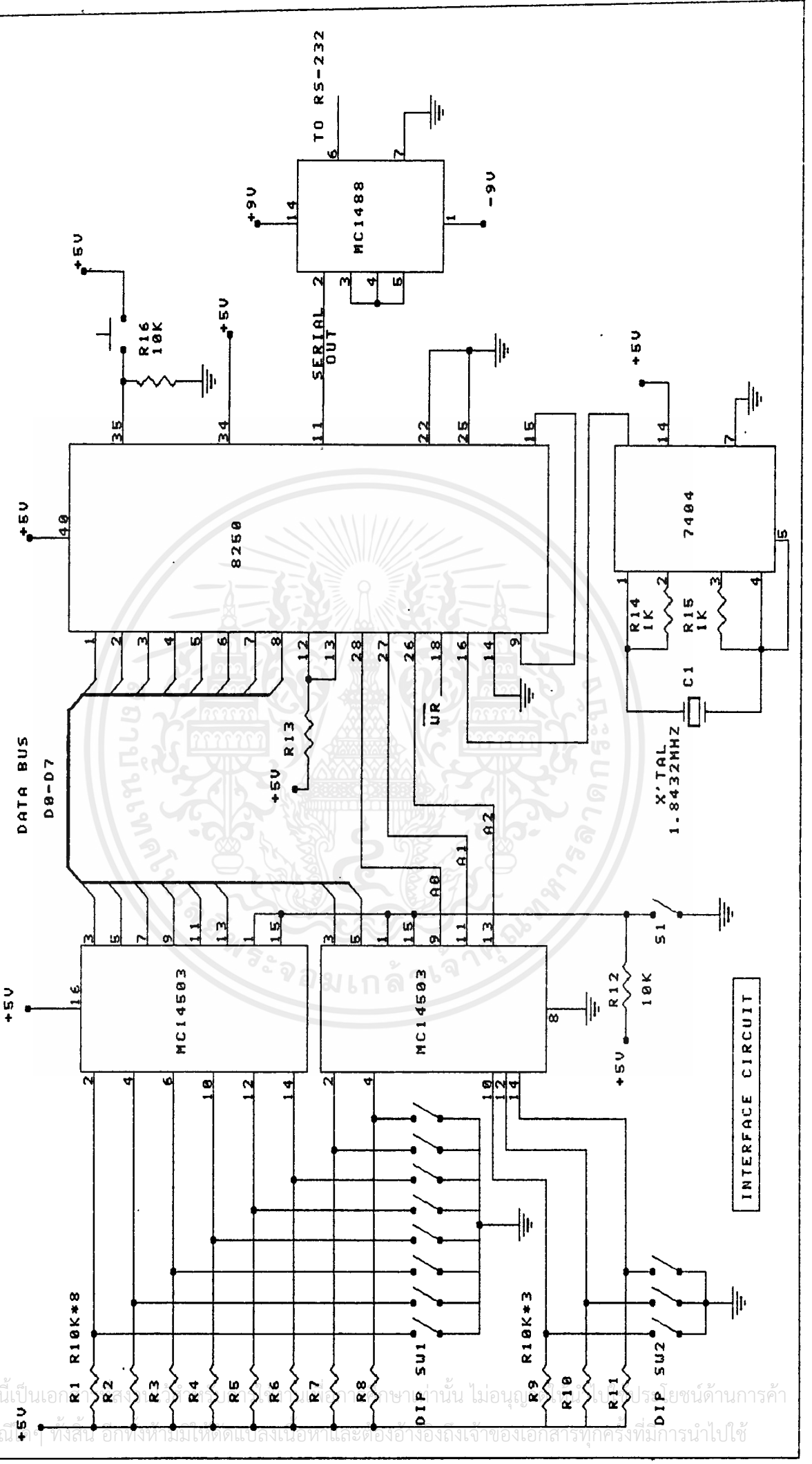
CALIBRATE AMP CIRCUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

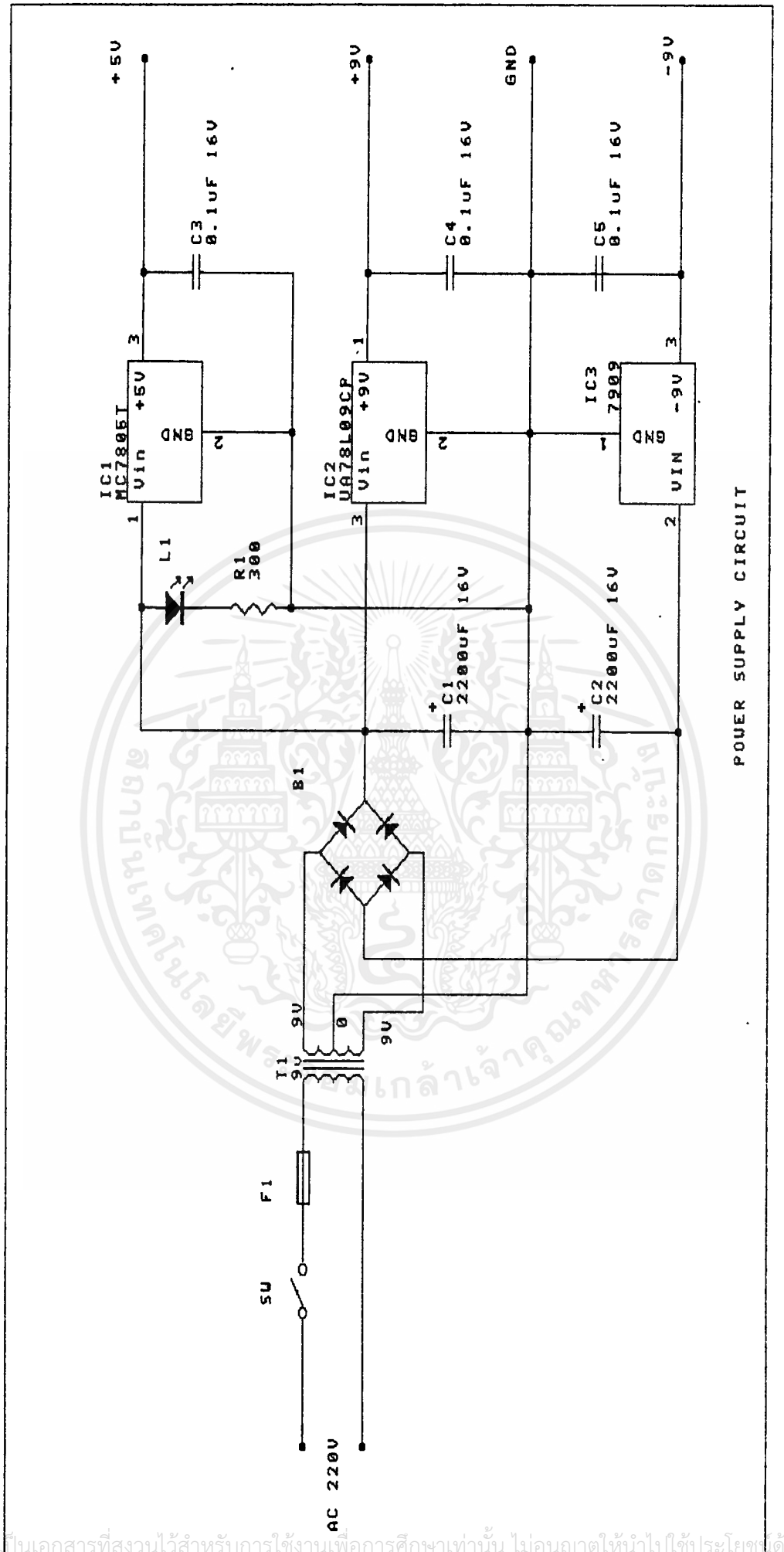


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VDD
C1
47pF
R2
1k
C3
C2
150pF
R1
20m
MM5369
8
6
5
2
1
7
60HZ OUT
CRYSTAL
FREQUENCY OUT
(3.579645MHZ)
VDD
16
5
15
14
4017
12
8
13
10HZ OUT
VDD
16
14
4017
12
1HZ OUT
15
8
13
OSCILLATOR CIRCUIT (1HZ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

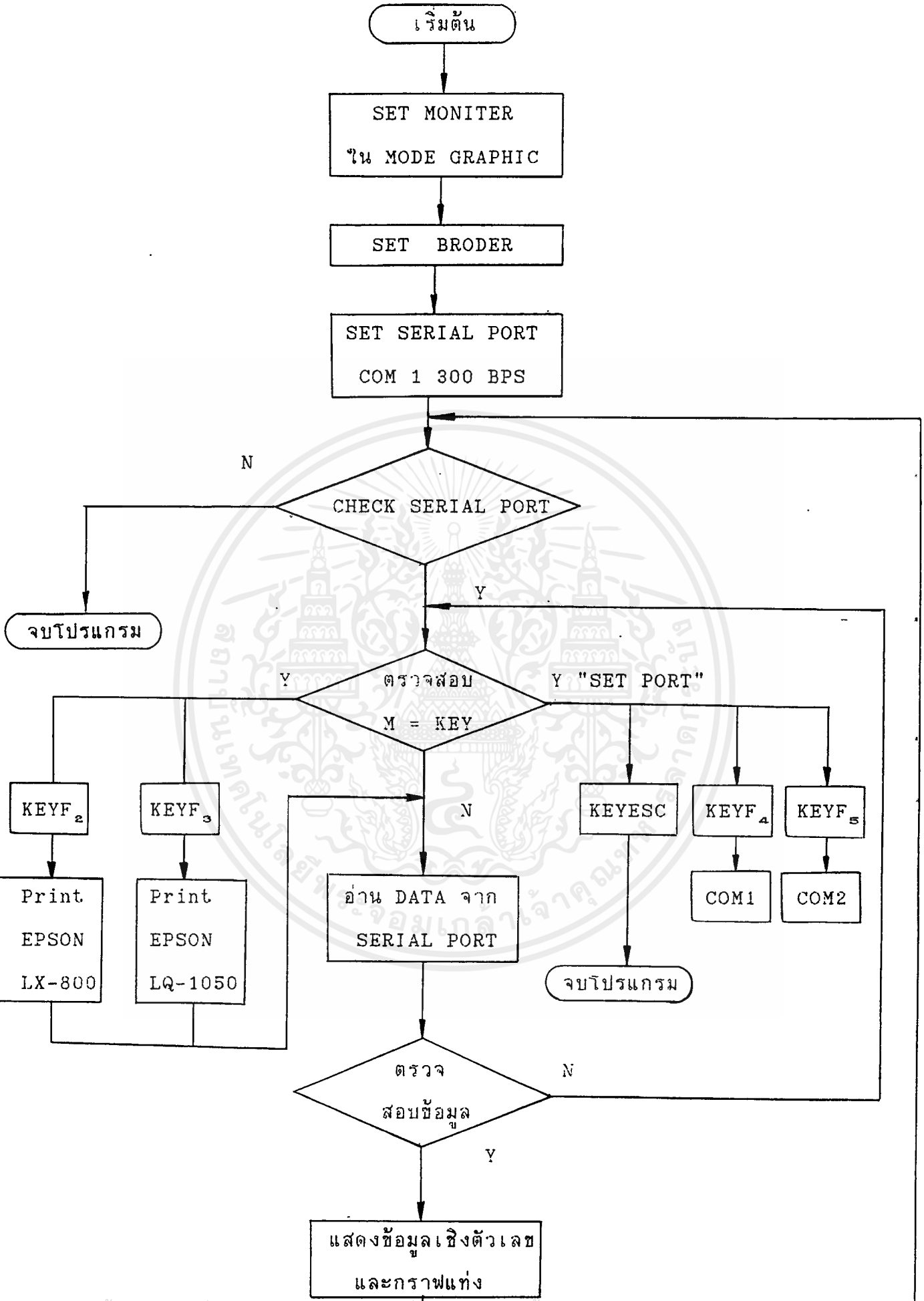
FLOW CHART และ โปรแกรมที่ใช้งาน

ในบทนี้ จะได้กล่าวถึง ในส่วนของงานทางด้าน Software ซึ่งจะมี Flow chart แสดงการทำงาน และมีโปรแกรมที่ใช้ในการแสดงผล และบันทึกค่าออกมาทางจอ Moniter

ในส่วนของโปรแกรม เราใช้ภาษาซีเขียนโปรแกรม ซึ่งภาษาซีสามารถติดต่อในระดับ Hardware ได้ดีกว่าภาษาระดับสูงอื่นๆ เช่น เบสิก ฟอร์แทรน ขณะเดียวกันก็มีคุณสมบัติของภาษาระดับสูงอยู่ด้วย ดังนั้นจึงได้จัดให้ภาษาซีเป็นภาษาระดับกลาง (Middle -level language) ด้วยเหตุผลดังกล่าว ภาษาซีเป็นภาษาคอมพิวเตอร์ชนิดคอมไพล์ (Compile language) ซึ่งมีคอมไพเลอร์ (Compiler) คอมไพล์ (Compile)คำสั่งทั้งหมดในโปรแกรม (Source code) เป็นภาษาเครื่อง (Object code) ซึ่งเป็นภาษาที่คอมพิวเตอร์เข้าใจ แล้วจะทำตามคำสั่งต่างๆในโปรแกรม นอกจากนี้ภาษาซีมีลักษณะเป็นภาษาโครงสร้าง (Structured language) เหมือนกับภาษาปาสคาล

ในส่วนของ Flow chart ก็ได้เริ่มโดย เมื่อมีการต่อด้าน Output ของเครื่อง PH RECORDER เข้ากับ Port RS 232 คอมพิวเตอร์ก็จะทำการ Check ดูว่ามีสัญญาณมาไหม ถ้าไม่มีก็จะไม่มีการ Run โปรแกรม แต่ถ้ามีสัญญาณมาคอมพิวเตอร์ก็จะอ่าน Data จาก Port RS 232 และทำการตรวจสอบข้อมูลที่ส่งมา เพื่อทำการแสดงผลที่หน้าจอ Moniter ให้ตรงกับสัญญาณของเครื่อง ถ้าต้องการบันทึกผลลงบนกระดาษ ก็จะทำได้โดยการกดคีย์ "F_2" เมื่อเครื่อง Printer พร้อมก็จะสามารถ Print ลงบนกระดาษได้ทันทีโดยเครื่อง Printer รุ่น EPSON LX-800 แต่ถ้ากดคีย์ "F_3" จะต้องเป็นเครื่อง Printer รุ่น EPSON LQ-1050 ส่วน "F_4" และ "F_5" เป็นการเลือก Port Com1 หรือ Com2 ตามลำดับ แต่ถ้าต้องการยกเลิกโปรแกรมก็ให้ ทำการกดคีย์ "Esc" ก็จะออกจากโปรแกรมนั้นทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>

int     GraphDriver;
int     GraphMode;
int     MaxX, MaxY;/* The maximum resolution of the screen */
int     MaxColors;/* The maximum # of colors available*/
int     ErrorCode;/* Reports any graphics errors*/

int    MaxX1;
int    MaxY1;
int    MaxX2;

float height;

  float xa1,xa2,xa3,xa4;
  float xb1,xb2,xb3,xb4;
  float ya1,ya2,ya3,ya4;
  float yb1,yb2,yb3,yb4;

  float xc1,xc2,xc3;
  float yc1,yc2,yc3;

struct palettetype palette;/* Used to read palette info*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Initialize(void);
void MakeBorder(void);
void DrawBorder(int left,int top,int right,int bottom);
void set(void);
void linep(void);
void linep1(void);
void show_g(void);
void clear_g(void);


void set_port();
void rec_data();
void out_item();
void make_h();
void print_graph();

/* int check_stat(int port);  */
int graphm;

 int port=0;
 int speed=0x43;
 int num_hex[10];

 float data_com[10];
 float data_f[10];

 float count = 1;

 int print_10 =0;
 int set_p;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
main()
{
  int i;

  Initialize(); /* Set system into Graphics mode*/
  set();
  MakeBorder();
  make_h();
  linep();
  linep1();
  set_port();

 for(;;) {
  count = 1;

  for(i=1;i<11;i++)
    show_g();

  count = 1;

  for(i=1;i<11;i++) {
   if(count==1)
      delay(1000);
      clear_g();
  }
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Initialize(void)
{
  union REGS r;

  r.h.al = 0;
  r.h.ah = 0xf;
  int86(0x10,&r,&r);
  graphm = r.h.al;

  GraphDriver = DETECT; /* Request auto-detection*/
  initgraph( &GraphDriver, &GraphMode, "" );
  ErrorCode = graphresult();/* Read result of initialization*/
  if( ErrorCode != grOk ){/* Error occured during init*/
    printf(" Graphics System Error: %s\n", grapherrormsg( Error
            Code ) );
    exit( 1 );
  }
  getpalette( &palette );/* Read the palette from board*/
  MaxColors = getmaxcolor() + 1;/*Read maximum number of colors*/

  MaxX = getmaxx();
  MaxY = getmaxy();/* Read size of screen*/



  if(graphm == 3) {
    MaxX1 = MaxX * 1.5;                    /* monitor  vga or ega  */
    MaxY1 = MaxY * 1.5;
    MaxX2 = MaxX * 0.5;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(graphm == 7) {
     MaxX1 = MaxX * 1;                    /* monitor  monochone  */
     MaxY1 = MaxY * 1;
     MaxX2 = MaxX * 0;
   }
}

void set(void)
{
  cleardevice();
  setcolor(MaxColors-1);
  setlinestyle(SOLID_LINE,0,NORM_WIDTH);
}

void MakeBorder(void)
{

  settextstyle(DEFAULT_FONT, HORIZ_DIR,1);
  height = textheight("H");

  /* makeborder1 */
  xa1 = 5;
  ya1 = 2;
  xb1 = MaxX-5;
  yb1 = MaxY-2;

  DrawBorder(xa1,ya1,xb1,yb1);

  /* makeborder2 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    xa2 = height+5;
    ya2 = height+2;
    xb2 = (MaxX/3)*2 - height ;
    yb2 = (MaxY/3)*2 - height ;

    DrawBorder(xa2,ya2,xb2,yb2);

    /* makeborder3 */
    xa3 = (MaxX/3)*2 + height ;
    ya3 = height+2 ;
    xb3 = (MaxX - height -5) ;
    yb3 = (MaxY/3)*2 - height;

    DrawBorder(xa3,ya3,xb3,yb3);

    /* makeborder4 */
    xa4 = height+5 ;
    ya4 = (MaxY/3)*2 + (2*height) ;
    xb4 = MaxX - height - 5;
    yb4 = MaxY - height -2;

   DrawBorder(xa4,ya4,xb4,yb4);
  }

  void DrawBorder(int left,int top,int right,int bottom)
  {
    rectangle(left,top,right,bottom);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void make_h()
{

  if(graphm == 3) {
    outtextxy((xa4+110),(ya4+15),"GRAPHIC  OF  PH.  RECORDER");
    outtextxy((xa4+120),(ya4+35),"MAJOR  TELECOMMUNICATION");
    outtextxy((xa4+140),(ya4+55),"FACULTY  ENGINEER");
    outtextxy((xa4+110),(ya4+75),"KING  MONGKUT'S  INSTITUTE");
    outtextxy((xa4+200),(ya4+95),"OF");
    outtextxy((xa4+120),(ya4+110),"TECHNOLOGY  LADKRABANG");
  }
  else
   if(graphm == 7) {
    outtextxy((xa4+110),(ya4+10),"GRAPHIC  OF  PH.  RECORDER");
    outtextxy((xa4+120),(ya4+23),"MAJOR  TELECOMMUNICATION");
    outtextxy((xa4+140),(ya4+33),"FACULTY  ENGINEER");
    outtextxy((xa4+110),(ya4+43),"KING  MONGKUT'S  INSTITUTE");
    outtextxy((xa4+200),(ya4+53),"OF");
    outtextxy((xa4+120),(ya4+63),"TECHNOLOGY  LADKRABANG");
  }

  line(xa3,ya4,xa3,yb4);

  outtextxy((xa3+10),(ya4+20),"KEY  OPERATE");
  outtextxy((xa3+10),(ya4+40),"ESC==> QUIT");
  outtextxy((xa3+10),(ya4+55),"F2==> PRINTER LX-800");
  outtextxy((xa3+10),(ya4+70),"F3==> PRINTER LQ-1050");
  outtextxy((xa3+10),(ya4+85),"F4==> COM1");
  outtextxy((xa3+100),(ya4+85)," F5==> COM2");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void linep()
{

  float xstep,ystep;
  float i;
  char buffer[40];

  xc1 = xa2 + (2*20) + height;
  yc1 = yb2 - (2*15) - height;
  xc2 = xc1;
  yc2 = ya1 + (2*20) + height;
  xc3 = xb2 - (2*20) - height;
  yc3 = yc1;
  line(xc2,yc2,xc1,yc1);
  line(xc1,yc1,xc3,yc3);

  outtextxy((xc3+10),(yc3-5),"NO.");

  outtextxy((xc1-15),(ya3+12),"PH.");

  xstep = ((xc3-xc1)-(height))/10;
  ystep = ((yc1-yc2)-(height))/14;

  settextjustify(CENTER_TEXT,CENTER_TEXT);

  for(i=0;i<15;i++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    line(xc1,(yc1-(ystep*i)),(xc1-(1*height)),(yc1-(ystep*i)));
    itoa(i,buffer,10);
    outtextxy((xc1-(3*height)),(yc1-(ystep*i)),buffer);
  }

  for(i=0;i<11;i++) {
    line((xc1+(xstep*i)),yc1,(xc1+(xstep*i)),(yc1+(1*height)));
    itoa(i,buffer,10);
    outtextxy((xc1+(xstep*i)),(yc1+(3*height)),buffer);

  }
}

void linep1()
{
  float xstep,ystep;
  float i;

  ystep = (yb3-ya3)/11;

  for(i=1;i<11;i++) {
    line(xa3,(ya3+(ystep*i)),xb3,(ya3+(ystep*i)));
  }

  line((xa3+(4*height)),ya3,(xa3+(4*height)),yb3);
  line((xa3+(4*height))+(10*height),ya3,(xa3+(4*height))+
      (10*height),yb3);

  out_item();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}



void out_item()
{
   float ystep;
   int i;
   char buffer[10];

   ystep = (yb3-ya3)/11;

   for(i=1;i<11;i++) {
    itoa(i,buffer,10);
    outtextxy(((xa3+(4*height))-15),(ya3+(ystep*(i)))+14,buffer);
   }
      outtextxy(((xa3+(4*height))-15),(ya3+14),"NO.");
      outtextxy(((xa3+(9*height))),(ya3+14),"PH.");

}

void show_g()
{
  float xstep;
  float ystep;
  float ystepi;
  char buffer[20];

  ystepi = (yb3-ya3)/11;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    rec_data();

    xstep = ((xc3-xc1)-(height))/10;
    ystep = ((yc1-yc2)-(height))/14;
    setfillstyle(SOLID_FILL,9);
    bar((xc1+(xstep*count))-3,yc1,(xc1+(xstep*count))+3,(yc1-
        (ystep*data_f[count])));


   gcvt((data_f[count]),3,buffer);
   outtextxy(((xa3+(9*height))),(ya3+(ystepi*(count))+14),buffer);

   if(print_10 == 1 && count == 10) {
      print_graph();
      print_10 = 0;
   }
    count++;
}


void clear_g()
{
   float xstep;
   float ystep;
   float stepi;

   float clear_x1;
   float clear_y1;
   float clear_x2;
   float clear_y2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   stepi = (yb3-ya3)/11;

   clear_x1 = (xa3+(4*height))+2;
   clear_y1 = (ya3+(stepi*count))+10;
   clear_x2 = (xa3+(4*height))+(9*height);
   clear_y2 = (ya3+(stepi*(count))+17);

   xstep = ((xc3-xc1)-(height))/10;
   ystep = ((yc1-yc2)-(height))/14;


    setfillstyle(EMPTY_FILL,14);
    bar((xc1+(xstep*count))-3,yc1,(xc1+(xstep*count))+3,(yc1-
        (ystep*data_f[count])));

    bar(clear_x1,clear_y1,clear_x2,clear_y2);
    count++;
}

void print_graph()
{
 char m;
 int i,j,k,Msb,Lsb;

 setviewport(0,0,MaxX1,MaxY1,0);

   if(set_p=1)
     fprintf(stdprn,"\x1b\x31");
   if(set_p=2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fprintf(stdprn,"\x1b\x33\25");

    Lsb = MaxY1 & 0x00ff;
    Msb = MaxY1>>8;
    for(j=0;j<=(MaxX1-(MaxX2));j+=8) {
       fprintf(stdprn,"\x1B*%c%c%c",1,Lsb,Msb);
       for(i=(MaxY1);i>=0;i--) {
  m=0;
  for(k=0;k<8;k++) {
     m<<=1;
     if(getpixel(j+k,i))m++;
  }
  fprintf(stdprn,"%c",m);
       }
    fprintf(stdprn,"\x0D\x0A");
    }
  }


void set_port()
{
   union REGS r;

  r.x.dx = port;
  r.h.ah = 0;
  r.h.al = speed;
  int86(0x14,&r,&r);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void rec_data()
{
  union REGS r;
  char anw;

  while(!(check_stat(port)&256))

       if(kbhit()) {

       switch(anw=getch())
       {
 case 27:
 closegraph();
 exit(1);
 break;
 case 60:
 print_10 = 1;
 set_p = 1;
 break;
 case 61:
 print_10 = 1;
 set_p = 2;
 break;
 case 62:
 port = 0;           /* com1 */
 break;
 case 63:
 port = 1;           /* com2 */
 break;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    r.x.dx = port;
    r.h.ah = 2;
    int86(0x14,&r,&r);


    if(r.h.ah & 128)
       printf("read error");

    num_hex[count] = r.h.al;

    data_com[count] = r.h.al;
    data_f[count] = (data_com[count]/18.214);

}

check_stat(port)
{
    union REGS r;

    r.x.dx = port;
    r.h.ah = 3;
    int86(0x14,&r,&r);
    return r.x.ax;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การทดลองและผลการทดลอง

เป็นการทดสอบเครื่อง และระบบการทำงานของ PH RECORDER โดยการนำไปวัดค่า ของน้ำประปา ซึ่งพอจะสรุปผลได้ดังนี้

วิธีการทดลอง

8.1 ทำการเซ็ตเครื่องโดย

8.1.1 กดสวิทช์ Reset

8.1.2 กดสวิทช์ Run

8.1.3 กดสวิทช์ S_1 เพื่อกำหนด Register ตัวหาร

8.1.4 กดสวิทช์ S_2 เพื่อกำหนดค่าแลตช์ตัวหาร (DLM) ในที่นี้ใช้ค่า 01_H

8.1.5 กดสวิทช์ S_3 เพื่อกำหนดค่าแลตช์ตัวหาร (DLL) ในที่นี้ใช้ค่า 80_H จะได้อัตราความเร็วในการส่ง เท่ากับ 300 bits/sec

8.1.6 กดสวิทช์ S_4 เพื่อกำหนด LCR(Line Control Register) ในที่นี้ใช้ค่า 03_H มีรูปแบบข้อมูลดังนี้ ความยาวข้อมูลเท่ากับ 8 bit,ไม่มี Parity check,ความยาว Stop bit = 1bit

8.1.7 กดสวิทช์ S_5 เพื่อนำข้อมูลจาก A/D ไปใช้ในข้อมูล สำหรับส่ง

8.2 ต่อสัญญาณจาก PH METER เข้ามายัง Input ของเครื่อง

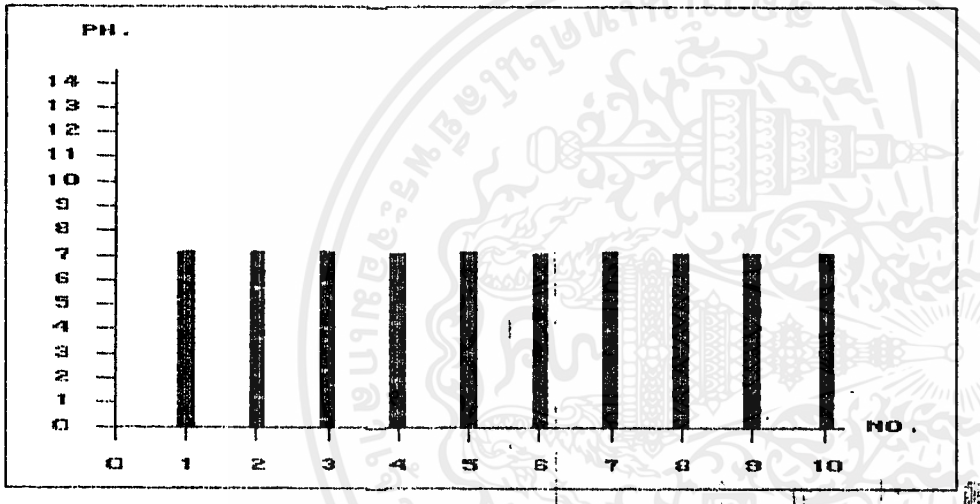
8.3 ต่อขั้ว Output ของเครื่องเข้ากับ Port RS 232

8.4 เรียกโปรแกรม PH.EXE

ผลการทดลอง

จากการทดลอง วัดค่าสัญญาณ ณ.จุดต่างๆ ในวงจรได้ดังนี้ ที่ขา 6 ของ TL 071 ได้สัญญาณ 70 mV PH มิเตอร์แสดงค่า 7.14 ที่ขา 1 ของ TL 072 วัดสัญญาณได้ 702 mV Data จาก A/D มีค่าเท่ากับ 81_H จากการทดลอง บันทึกค่าในช่วง 10 วินาที ดูผลได้จากกราฟที่บันทึก ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



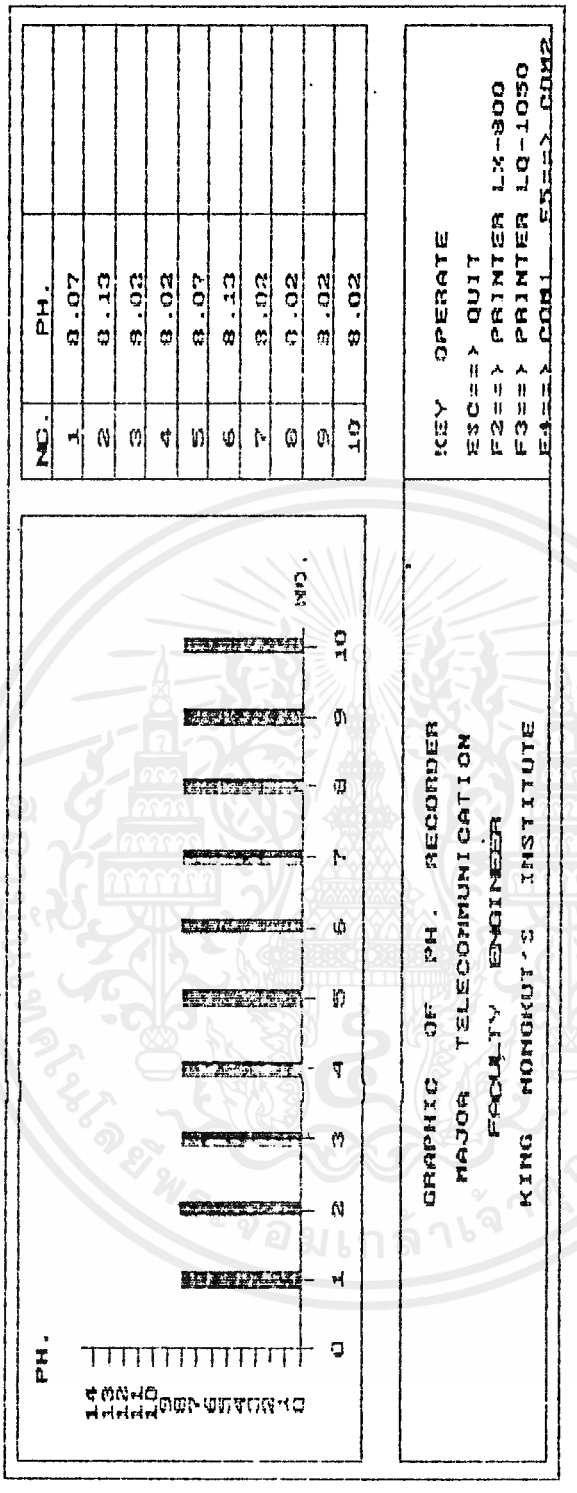
NO.	PH.	
1	7.14	
2	7.14	
3	7.14	
4	7.08	
5	7.14	
6	7.08	
7	7.14	
8	7.08	
9	7.08	
10	7.08	

GRAPHIC OF PH. RECORDER
MAJOR TELECOMMUNICATION
FACULTY ENGINEER
KING MONGKUT'S INSTITUTE
OF
TECHNOLOGY LADKRABANG

KEY OPERATE
ESC ==> QUIT
F2 ==> PRINTER

8.1 รูปแสดงค่า PH ต่ำกว่า 7 แสดงว่าสารละลายเป็นกรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8.2 รูปแสดงค่า PH สูงกว่า 7 แสดงว่าสารละลายเป็นด่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

สรุปปัญหาการแก้ไขและแนวความคิดพัฒนา PH RECORDER

9.1 ปัญหาด้าน HARD WARE

ปัญหาในช่วงต้น พบว่า OUTPUT ของ PH METER นั้นมีค่า IMPEDANCEที่สูงมาก เมื่อนำมาต่อเข้ากับ CALIBRATE AMP แล้วจะทำให้เกิดปัญหาการโหลดของสัญญานขึ้น ก่อให้เกิดความผิดพลาดมาก

การแก้ไขทำได้โดยการใช้วงจร BUFFER มาคั่นระหว่าง PH METER กับวงจร CALIBRATE AMP ซึ่งสามารถแก้ปัญหาดังกล่าวได้ดี ซึ่ง OP AMP ที่นำมาใช้ทำ BUFFER AMP นั้น ต้องใช้ชนิดที่เป็น FET INPUT เพราะจะมี INPUT IMPEDANCE สูงมาก ส่วน OP AMP ธรรมดานั้นแก้ปัญหาได้ไม่ดีเท่าที่ควร

ปัญหาในการ SET IC SERIAL PORT 8250 BUFFER ที่ใช้เป็น CMOS เมื่อต่อเชื่อมกับ 8250 ซึ่งเป็น TTL เมื่อมาดูถึงคุณสมบัติทางด้าน INPUT และ OUTPUT ของ TTL และ CMOS โดยทางด้าน OUTPUT ของ CMOS นั้นสัญญานระดับ "1" จะมีระดับแรงดันสูงกว่า 3.5 V และสัญญานระดับ "0" จะมีระดับแรงดันต่ำกว่า 1.5 V และทางด้าน INPUT ของ TTL นั้น สัญญานระดับ "1" จะต้องมีระดับแรงดันสูงกว่า 2 V และสัญญานระดับ "0" จะมีระดับแรงดันต่ำกว่า 0.8 V เปรียบเทียบระดับแรงดันของสัญญานทั้งสองแล้วเห็นว่าสัญญานระดับ "1" นั้นไม่มีปัญหา เพราะ CMOSให้สัญญานที่มีระดับสูงพอ แต่เมื่อเป็นสัญญานระดับ "0" นั้นจะมีปัญหา เมื่อ CMOS ให้แรงดันต่ำกว่า 1.5 V ขณะที่ TTL ต้องการแรงดันต่ำกว่า 0.8 V ปัญหาใหญ่อีกก็คือเมื่อสัญญานเข้าเป็น "0" TTL จะต้องจ่ายกระแสไหลออกไปเข้าเกต CMOS ถึง 1.6 mA ซึ่งกระแสขนาดนี้ CMOS จะรับไม่ไหว เมื่อเป็นเช่นนี้ CMOS จะไม่สามารถต่อโดยตรงกับ TTL ได้ จะแก้ปัญหาโดยการเปลี่ยน TTL ไปใช้ชนิดที่กินไฟน้อยลง และจ่ายกระแสไหลออกน้อยลง พอที่ CMOS จะรับได้ โดย TTL ตระกูลกินไฟน้อย(LOW POWER TTL) คือตระกูล 74LXX และตระกูล TTL ชอตต์กีกินไฟน้อย (LOW POWER SHCOTTKY TTL) คือตระกูล 74LSXX โดยเฉพาะตระกูล 74LSXX นั้นกินไฟน้อยกว่า TTL มาตรฐานถึง 5 เท่า ในขณะที่ความเร็วในการทำงานพอ ๆ กัน

9.2 ปัญหาด้าน SOFTWARE

รูปแบบการต่อสายสัญญาน ระหว่างตัวคอมพิวเตอร์ และตัวเครื่องที่ทำขึ้นมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลายรูปแบบ ทำให้เสียเวลาในการทดสอบ เพื่อหาการเชื่อมที่เหมาะสม

แนวทางตัว PROGRAM ที่ใช้เขียน (ภาษา C)เพื่อเชื่อมรับข้อมูลจากเครื่องวัดค่า PH (ผ่านเครื่องแปลงสัญญาน ANALOG TO DIGITAL ก่อน) มีตัวอย่างน้อย

ข้อมูลเกี่ยวกับการเขียน PROGRAM ควบคุม PRINTER มีน้อยมากทำให้เสียเวลาในการทดลองสุ่มเขียน PROGRAM

9.3 แนวความคิดพัฒนา PH RECORDER

ทางด้าน Hardware จากงานที่ได้ทำมา ซึ่งเป็นการวัดค่า PH ของสารละลาย ซึ่งสามารถจะบอกได้ว่า สารละลายนั้นเป็นกรดหรือด่าง โดยใช้ค่า 7 เป็นค่ากลาง ถ้ามีค่า PH สูงกว่า 7 แสดงว่าเป็นด่าง และถ้ามีค่า PH ต่ำกว่า 7 แสดงว่าเป็นกรด ซึ่งแนวทางในการพัฒนาต่อไป อาจจะใช้วัดสัญญาณแบบใดก็ได้ ซึ่งให้ Outputออกมาเป็นอนาลอก ก็จะสามารถนำสัญญาณนั้น เข้าภาค A/D ด้าน Input ของ Hardware ได้ เช่น การวัดออกซิเจนในน้ำ หรือจะเป็นการวัดแรงลม ซึ่งอาจจะนำไปใช้ได้อย่างกว้างขว้าง หรืออาจจะเป็นการนำเอา Output ของ Hardware ซึ่งเป็นสัญญาณดิจิตอล ที่มีการจัดรูปแบบของข้อมูลในแบบอนุกรม ส่งออกอากาศไป และก็ทำการรับสัญญาณ นำสัญญาณมาดีเทคเพื่อแยกเอาสัญญาณดิจิตอลออกมา แล้ว Interface กับคอมพิวเตอร์เพื่อแสดงออกทางจอ Monitor ซึ่งการทำแบบนี้ ยังจะสามารถใช้กับ การวัดที่มีหลาย Input และมีระยะทางที่ห่างกัน มาออกที่ศูนย์ควบคุม ซึ่งเป็นจอ Monitor ของคอมพิวเตอร์ แต่ในการทำแบบนี้นั้น จะต้องทำการพัฒนาทางด้าน Software ให้สามารถแสดงผลได้หลายอินพุท

ส่วนทางด้าน Foftware จากงานที่ได้ทำมาผลที่แสดงออกทางจอ Monitor ของคอมพิวเตอร์ ไม่สามารถที่จะกำหนดเวลาในการแสดงผลได้ ซึ่งจะต้องไปแก้ในโดยการเพิ่ม สัญญาณ Clock เข้าไป และทางด้าน Software ตัวนี้ยังไม่สามารถตรวจสอบได้ว่า Serial port Comใด Comหนึ่ง มีหรือไม่ ถ้าไม่มีเครื่องคอมพิวเตอร์จะ Hang หากมี การเลือกใช้ ณ.ที่ Com นั้นๆ ส่วน Software ตัวนี้ สามารถที่จะแสดงผลได้ทั้งจอ Monocrome, EGA และ VGA ส่วนรูปแบบที่แสดงผลออกมาทาง Printer จะแสดงได้แค่รูปภาพที่ออกมาทางมอนิเตอร์ ของคอมพิวเตอร์เท่านั้น อาจจะนำไปพัฒนาโดยการเขียน Software ให้มีฟังค์ชั่นรูปแบบการพิมพ์ต่างๆ เช่น อาจจะเป็นกราฟแบบอื่นก็ได้ แล้วแต่ลักษณะที่จะนำไปใช้งาน

แนวทางต่างๆเหล่านี้จะเป็นประโยชน์สำหรับผู้ที่สนใจและนักศึกษารุ่นต่อๆไป ที่มีความคิดที่จะพัฒนาคุณภาพ PH RECORDER ให้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5404/7404 Hex Inverter

	Schottky TTL					High-Speed TTL					Low-Power Schottky TL					Standard TTL					Low-Power TTL				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF
T. I.	SN54S04	J			W	SN54H04	J			W	SN54LS04	J			W	SN5404	J			W	SN54L04	J			
	SN74S04	J	N			SN74H04	J	N			SN74LS04	J	N			SN7404	J	N			SN74L04	J	N		
FAIRCHILD	FM54S04/FM9S04	D				FM54H04/FM9H04	D			F	FM54LS04/FM9LS04	D			F	FM5404/FM9N04	D			F					
	FC74S04/FC9SC4	D	P			FC74H04/FC9H04	D	P			FC74LS04/FC9LS04	D	P		F	FM7404/FC9N04	D	P		F					
MOTOROLA						MC3108	L			F						MC5404	L			F					
						MC3008	L	P		F	SN74LS04		P			MC7404	L	P							
N. S. C.						DM54H04	J	N			DM54LS04					DM5404	J	N		W	DM54L04	J	N		F
	DM74S04		N			DM74H04	J	N			DM74LS04					DM7404	J	N			DM74L04	J	N		F
PHILIPS																									
	N74S04					N74H04					N74LS04					FJH241/7404									
SIGNETICS	S54S04	F	A		W	S54H04	F	A		W						S5404	F	A		W					
	N74S04	F	A		W	N74H04	F	A			N74LS04		A			N7404	F	A							
SIEMENS																									
																FLH211									
FUJITSU																									
											74LS04		M			MB418		M							
HITACHI																									
	HD74S04		P								HD74LS04		P			HD7404/HD2522		P							
MITSUBISHI																									
	M5S004		P								M74LS04		P			M53204		P							
NEC																									
	74S04		C								74LS04		C			μPB235									
TOSHIBA																									
																TD3404A		P							

Electrical Characteristics SN54LS04/SN74LS04

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

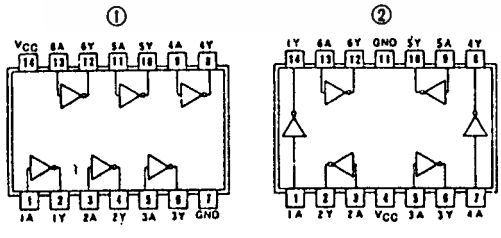
recommended operating conditions

	SN54LS04			SN74LS04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

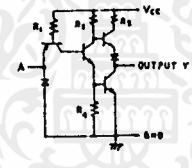
PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN,	I_I = −18mA			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, I_{OH} = MAX	V_{IL} = V_{IL} max,	2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, I_{OL} = 4mA	V_{IH} = 2V,			0.4	V
I_I	Input current at maximum input voltage	V_{CC} = MAX,	V_I = 7V			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX,	V_{IH} = 2.7V			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX,	V_{IL} = 0.4V			−0.4	mA
I_{OS}	Short-circuit output current *	V_{CC} = MAX	54LS Family	−20		−100	mA
			74LS Family	−20		−100	
I_{CCH}	Supply current	V_{CC} = MAX	Total, outputs high		1.2	2.4	mA
I_{CCL}	Supply current		Total, outputs low		3.6	6.6	mA
I_{CC}	Supply current	V_{CC} = 5V	Average per gate (50% duty cycle)		0.4		mA
t_{PLH}	Propagation delay time, low-to-high-level output	V_{CC} = 5V, C_L = 15pF,	T_A = 25°C, R_L = 2kΩ		9	15	ns
t_{PHL}	Propagation delay time, high-to-low-level output				10	15	ns

Pin Assignments (Top View)



positive logic: $Y = \overline{A}$

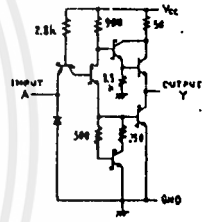
Schematics (each gate)



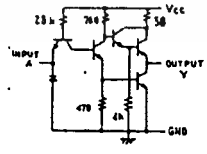
CIRCUIT	R1	R2	R3	R4
'04	4k	1.6k	130	1k
'L04	40k	20k	500	12k

Input clamp diodes not on SN54LS/SN74LS circuits.

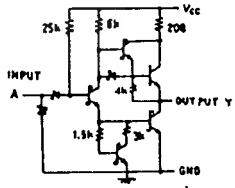
'04, 'L04 CIRCUITS



'S04 CIRCUIT



'H04 CIRCUIT



'L304 CIRCUIT

Resistor values shown are nominal and in ohms.

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5V, T_A = 25°C.
*Not more than one output should be shorted at a time, and for SN54H'/SN74H' and SN54S'/SN74S', duration of short-circuit should not exceed 1 second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES TL070, TL070A, TL071, TL071A, TL071B, TL072, TL072A, TL072B, TL074, TL074A, TL074B, TL075
LOW-NOISE JFET-INPUT OPERATIONAL AMPLIFIERS

BULLETIN NO. DL-S 12640, SEPTEMBER 1978—REVISED OCTOBER 1979

20 DEVICES COVER COMMERCIAL, INDUSTRIAL, AND MILITARY TEMPERATURE RANGES

- Low Noise . . . V_n = 18 nV/$\sqrt{Hz}$ Typ
- Low Harmonic Distortion . . . 0.01% Typ
- Wide Common-Mode and Differential Voltage Ranges
- Low Input Bias and Offset Currents
- Output Short-Circuit Protection
- High Input Impedance . . . JFET-Input Stage
- Internal Frequency Compensation
- Low Power Consumption
- Latch-Up-Free Operation
- High Slew Rate . . . 13 V/μs Typ

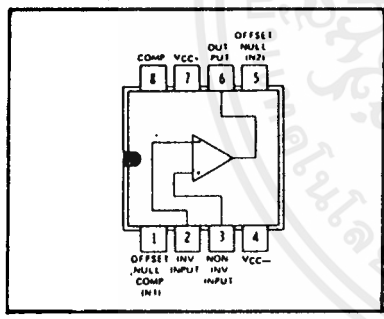
description

The JFET-input operational amplifiers of the TL071 series are designed as low-noise versions of the TL081 series amplifiers with low input bias and offset currents and fast slew rate. The low harmonic distortion and low noise make the TL071 series ideally suited as amplifiers for high-fidelity and audio preamplifier applications. Each amplifier features JFET-inputs (for high input impedance) coupled with bipolar output stages all integrated on a single monolithic chip.

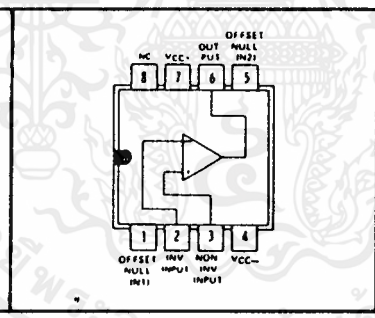
Device types with an "M" suffix are characterized for operation over the full military temperature range of −55°C to 125°C, those with an "I" suffix are characterized for operation from −25°C to 85°C, and those with a "C" suffix are characterized for operation from 0°C to 70°C.

4

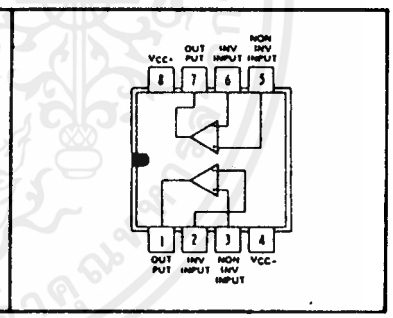
TL070, TL070A
JG OR P DUAL-IN-LINE
PACKAGE (TOP VIEW)



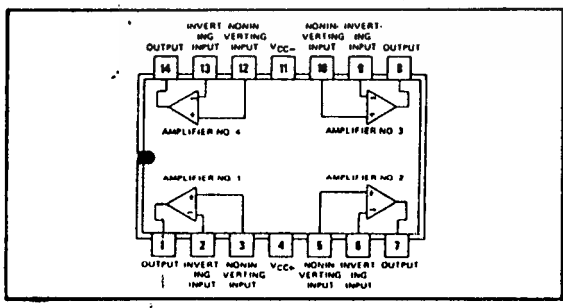
TL071, TL071A, TL071B
JG OR P DUAL-IN-LINE
PACKAGE (TOP VIEW)



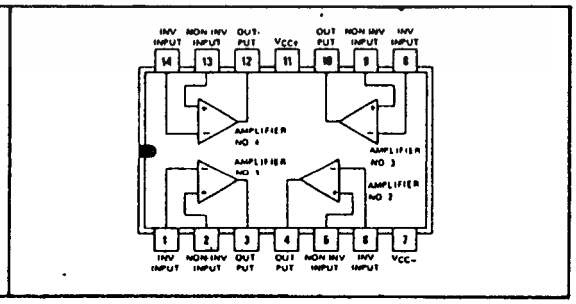
TL072, TL072A, TL072B
JG OR P DUAL-IN-LINE
PACKAGE (TOP VIEW)



TL074, TL074A, TL074B
J OR N DUAL-IN-LINE
OR W PACKAGE (TOP VIEW)



TL075
N DUAL-IN-LINE
PACKAGE (TOP VIEW)

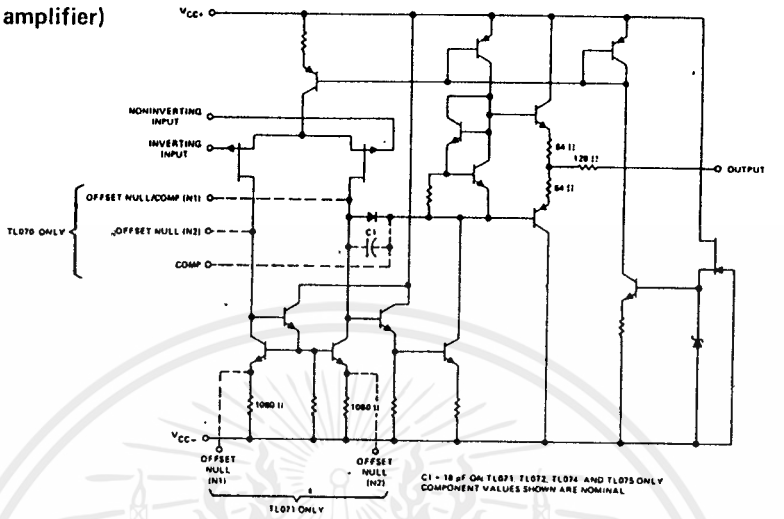


Copyright © 1979 by Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES TL070, TL070A, TL071, TL071A, TL071B, TL072, TL072A, TL072B, TL074, TL074A, TL074B, TL075
LOW-NOISE JFET-INPUT OPERATIONAL AMPLIFIERS

schematic (each amplifier)



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

		TL07_C	TL07_I	TL07_C TL07_AC TL07_BC	UNIT
Supply voltage, V_{CC+} (see Note 1)		18	18	18	V
Supply voltage, V_{CC-} (see Note 1)		−18	−18	−18	V
Differential input voltage (see Note 2)		±30	±30	±30	V
Input voltage (see Notes 1 and 3)		±15	±15	±15	V
Duration of output short circuit (see Note 4)		Unlimited	Unlimited	Unlimited	
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 5)		680	680	680	mW
Operating free-air temperature range		−55 to 125	−25 to 85	0 to 70	°C
Storage temperature range		−65 to 150	−65 to 150	−65 to 150	°C
Lead temperature 1/16 inch (1,6 mm) from case for 60 seconds	J, JG or W package	300	300	300	°C
Lead temperature 1/16 inch (1,6 mm) from case for 10 seconds	N or P package		260	260	°C

NOTES:
1. All voltage values, except differential voltages, are with respect to the midpoint between V_{CC+} and V_{CC-}.
2. Differential voltages are at the noninverting input terminal with respect to the inverting input terminal.
3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
4. The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.
5. For operation above 25°C, free-air temperature, refer to Dissipation Derating Table. In the J and JG packages, TL07_M chips are alloy-mounted; TL07_I, TL07_C, TL07_AC, and TL07_BC chips are glass-mounted.

DISSIPATION DERATING TABLE

PACKAGE	POWER RATING	DERATING FACTOR	ABOVE T_A
J (Alloy-Mounted Chip)	680 mW	11.0 mW/°C	88°C
J (Glass-Mounted Chip)	680 mW	8.2 mW/°C	67°C
JG (Alloy-Mounted Chip)	680 mW	8.4 mW/°C	69°C
JG (Glass-Mounted Chip)	680 mW	6.6 mW/°C	47°C
N	680 mW	9.2 mW/°C	76°C
P	680 mW	8.0 mW/°C	65°C
W	680 mW	8.0 mW/°C	65°C

Also see Dissipation Derating Curves, Section 2.

DEVICE TYPES, SUFFIX VERSIONS, AND PACKAGES

	TL070	TL071	TL072	TL074	TL075
TL07_M	JG,	JG,	JG,	J, W	•
TL07_I	JG, P	JG, P	JG, P	J, N	•
TL07_C	JG, P	JG, P	JG, P	J, N	N
TL07_AC	JG, P	JG, P	JG, P	J, N	•
TL07_BC	•	JG, P	JG, P	J, N	•

•These combinations are not defined by this data sheet.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics, $V_{CC\pm}$ = ±15 V

PARAMETER		TEST CONDITIONS†		TL07_M			TL07_I			TL07_C †TL07_AC TL07_BC			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	R_S = 50 Ω, T_A = 25°C	'70, '71, '72, '75‡		3	6		3	6		3	10	mV
			'74		3	9		3	6		3	10	
			'70A, '71A, '72A, '74A								3	6	
			'71B, '72B, '74B								2	3	
		R_S = 50 Ω, T_A = full range	'70, '71, '72, '75‡			9			9			13	
			'74			15			9			13	
			'70A, '71A, '72A, '74A									7.5	
			'71B, '72B, '74B									5	
α_{VIO}	Temperature coefficient of input offset voltage	R_S = 50 Ω,	T_A = full range		10			10			10		μV/°C
I_{IO}	Input offset current§	T_A = 25°C	'70, '71, '72, '74, '75‡		5	50		5	50		5	50	pA
			'70A, '71A, '72A, '74A								5	50	
			'71B, '72B, '74B								5	50	
		T_A = full range	'70, '71, '72, '74, '75‡			20			10			2	nA
			'70A, '71A, '72A, '74A									2	
			'71B, '72B, '74B									2	
I_{IB}	Input bias current§	T_A = 25°C	'70, '71, '72, '74, '75‡		30	200		30	200		30	200	pA
			'70A, '71A, '72A, '74A								30	200	
			'71B, '72B, '74B								30	200	
		T_A = full range	'70, '71, '72, '74, '75‡			50			20			7	nA
			'70A, '71A, '72A, '74A									7	
			'71B, '72B, '74B									7	
V_{ICR}	Common-mode input voltage range	T_A = 25°C	'70, '71, '72, '74, '75‡	±11	±12		±11	±12		±10	±11		V
			'70A, '71A, '72A, '74A							±11	±12		
			'71B, '72B, '74B							±11	±12		
V_{OPP}	Maximum peak-to-peak output voltage swing	T_A = 25°C,	R_L = 10 kΩ	24	27		24	27		24	27		V
		T_A = full range	R_L ≥ 10 kΩ	24			24			24			
			R_L ≥ 2 kΩ	20	24		20	24		20	24		
A_{VD}	Large-signal differential voltage amplification	R_L ≥ 2 kΩ, V_O = ±10 V, T_A = 25°C	'70, '71, '72, '74, '75‡	35	200		50	200		25	200		V/mV
			'70A, '71A, '72A, '74A							50	200		
			'71B, '72B, '74B							50	200		
		R_L ≥ 2 kΩ, V_O = ±10 V, T_A = full range	'70, '71, '72, '74, '75‡	20			25			15			
			'70A, '71A, '72A, '74A							25			
			'71B, '72B, '74B							25			
B_1	Unity-gain bandwidth	T_A = 25°C,	R_L = 10 kΩ		3			3			3		MHz
r_i	Input resistance	T_A = 25°C			10^{12}			10^{12}			10^{12}		Ω
CMRR	Common-mode rejection ratio	R_S ≤ 10 kΩ, T_A = 25°C	'70, '71, '72, '74, '75‡	80	86		80	86		70	76		dB
			'70A, '71A, '72A, '74A							80	86		
			'71B, '72B, '74B							80	86		
k_{SVR}	Supply voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$)	R_S ≤ 10 kΩ, T_A = 25°C	'70, '71, '72, '74, '75‡	80	86		80	86		70	76		dB
			'70A, '71A, '72A, '74A							80	86		
			'71B, '72B, '74B							80	86		
I_{CC}	Supply current (per amplifier)	No load, T_A = 25°C	No signal,		1.4	2.5		1.4	2.5		1.4	2.5	mA
V_{o1}/V_{o2}	Channel separation	A_{VD} = 100,	T_A = 25°C		120			120			120		dB

†All characteristics are specified under open-loop conditions unless otherwise noted. Full range for T_A is −55°C to 125°C for TL07_M; −25°C to 85°C for TL07_I; and 0°C to 70°C for TL07_C, TL07_AC, and TL07_BC.
‡Types TL075I and TL075M are not defined by this data sheet.
§Input bias currents of a FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive as shown in Figure 18. Pulse techniques must be used that will maintain the junction temperatures as close to the ambient temperature as is possible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS†

LARGE-SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION vs FREE-AIR TEMPERATURE

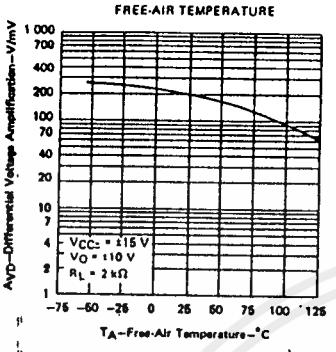


FIGURE 15

LARGE-SIGNAL DIFFERENTIAL VOLTAGE AMPLIFICATION and PHASE SHIFT vs FREQUENCY

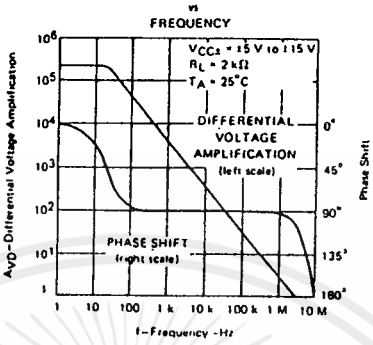


FIGURE 16

TL070 DIFFERENTIAL VOLTAGE AMPLIFICATION vs FREQUENCY WITH FEED-FORWARD COMPENSATION

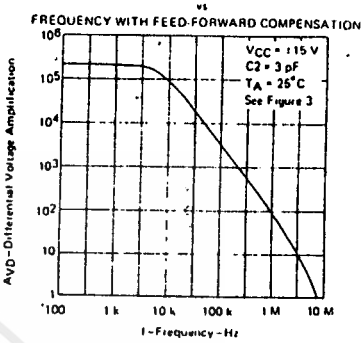


FIGURE 17

INPUT BIAS CURRENT vs FREE-AIR TEMPERATURE

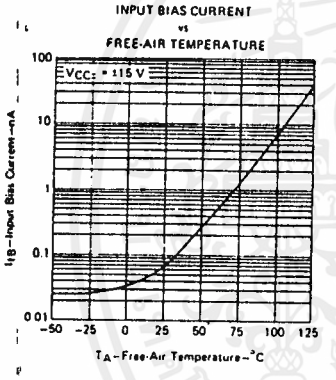


FIGURE 18

NORMALIZED UNITY GAIN BANDWIDTH and PHASE SHIFT vs TEMPERATURE

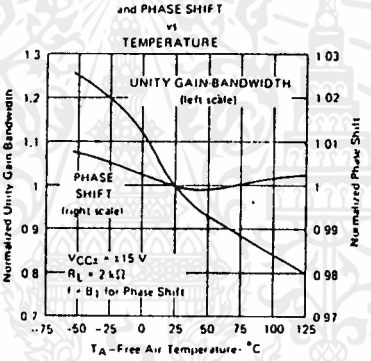


FIGURE 19

NORMALIZED SLEW RATE vs TEMPERATURE

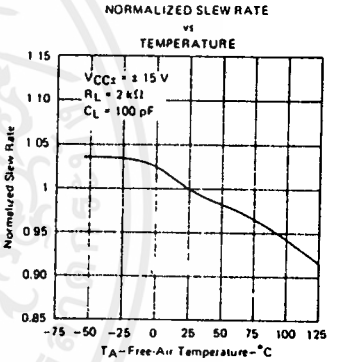


FIGURE 20

ALL EXCEPT TL07–C COMMON-MODE REJECTION RATIO vs FREE-AIR TEMPERATURE

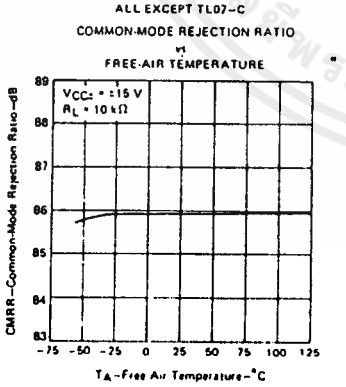


FIGURE 21

EQUIVALENT INPUT NOISE VOLTAGE vs FREQUENCY

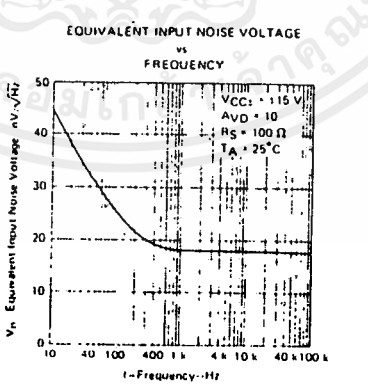


FIGURE 22

TOTAL HARMONIC DISTORTION vs FREQUENCY

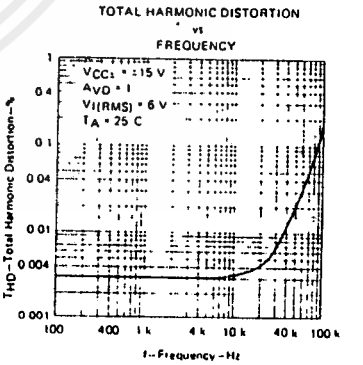


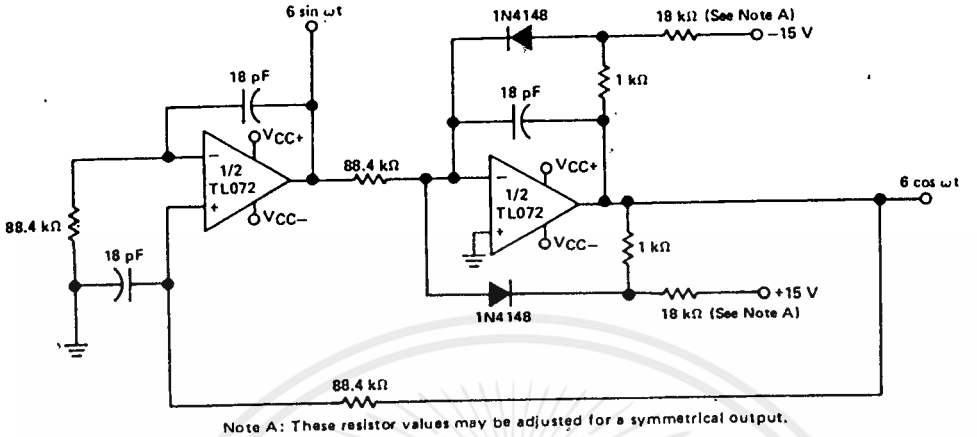
FIGURE 23

†Data at high and low temperatures are applicable only within the rated operating free-air temperature ranges of the various devices. A 18-pF compensation capacitor is used with TL070 and TL070A.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA



Note A: These resistor values may be adjusted for a symmetrical output.

FIGURE 29—100-KHz QUADRATURE OSCILLATOR

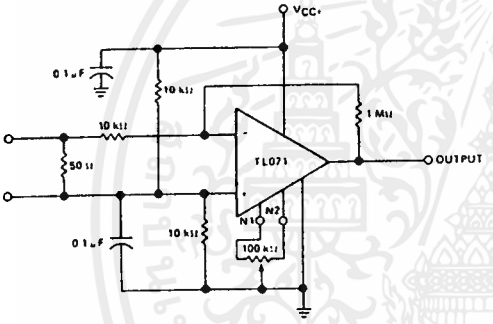


FIGURE 30—AC AMPLIFIER

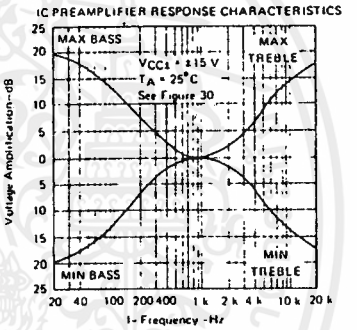


FIGURE 31

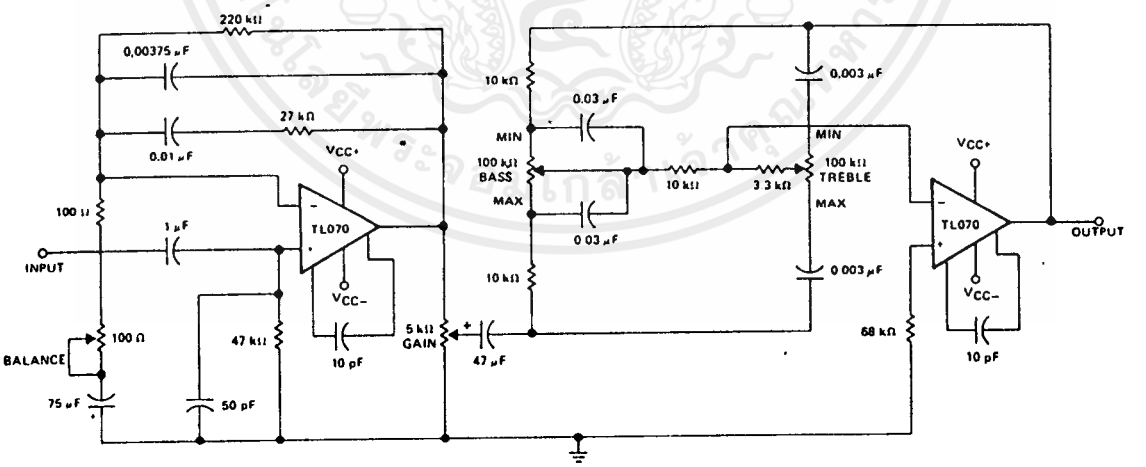


FIGURE 32—IC PREAMPLIFIER

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS†

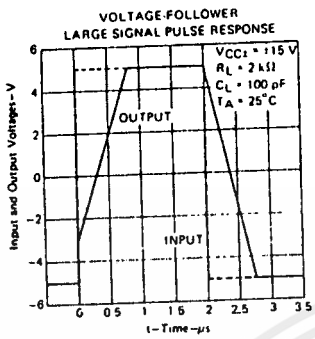


FIGURE 24

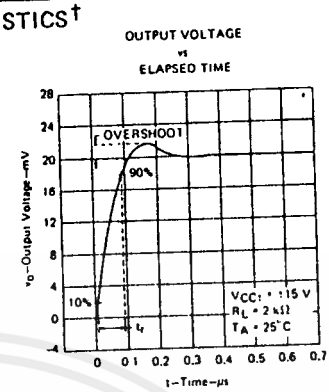


FIGURE 25

†Data at high and low temperatures are applicable only within the rated operating free-air temperature ranges of the various devices. A 18-pF compensation capacitor is used with TL070 and TL070A.

TYPICAL APPLICATION DATA

0.5-Hz SQUARE-WAVE OSCILLATOR

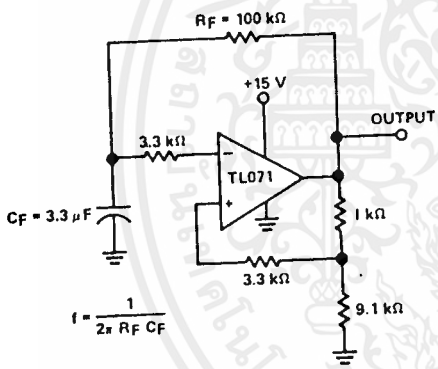


FIGURE 26–0.5-Hz SQUARE-WAVE OSCILLATOR

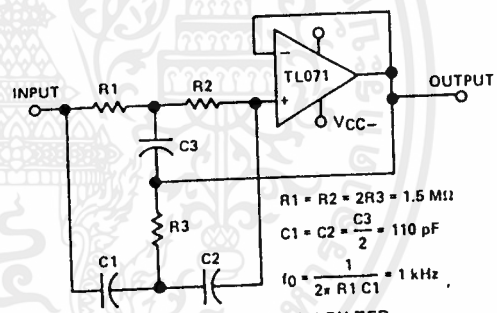
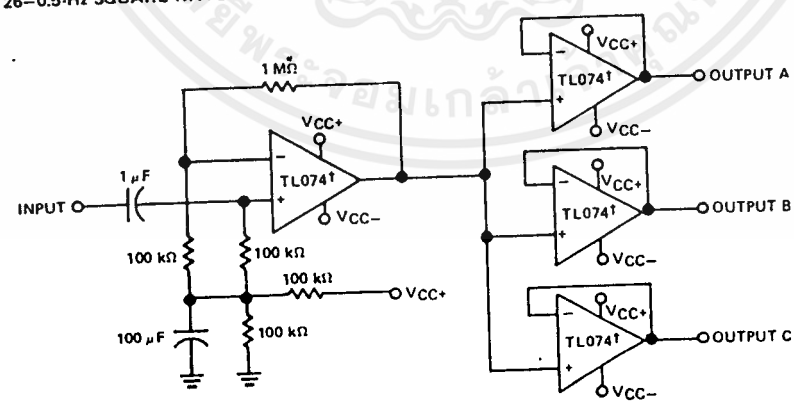


FIGURE 27–HIGH-Q NOTCH FILTER



† or TL075

FIGURE 28–AUDIO DISTRIBUTION AMPLIFIER

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805
8-Bit μP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

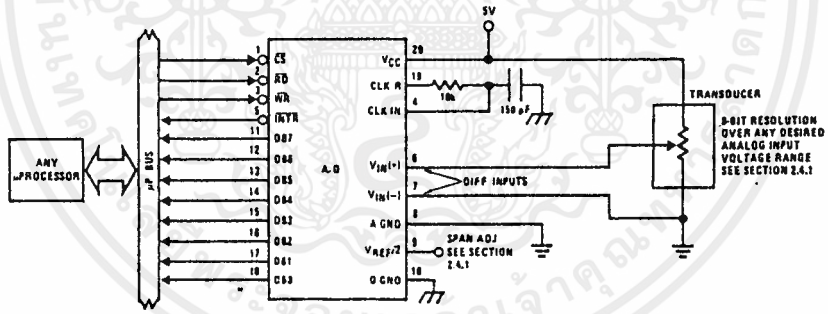
Features

- Compatible with 8080 μP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

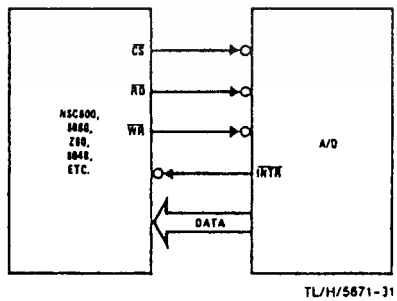
- Resolution: 8 bits
- Total error: ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time: 100 μs

Typical Applications



TL/H/5671-1

8080 Interface



TL/H/5671-31

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Notes 1 & 2)

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

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage	
Logic Control Inputs	−0.3V to +18V
At Other Input and Outputs	−0.3V to (V_{CC} + 0.3V)
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A = 25°C	875 mW
ESD Susceptibility (Note 10)	800V

Operating Ratings (Notes 1 & 2)

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
ADC0801/02LJ	$-55°C \le T_A \le +125°C$
ADC0801/02/03/04LCJ	$-40°C \le T_A \le +85°C$
ADC0801/02/03/05LCN	$-40°C \le T_A \le +85°C$
ADC0804LCN	$0°C \le T_A \le +70°C$
ADC0802/03/04LCV	$0°C \le T_A \le +70°C$
ADC0802/03/04LCWM	$0°C \le T_A \le +70°C$
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

The following specifications apply for V_{CC} = 5 V_{DC}, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK} = 640 kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±¼	LSB
ADC0802: Total Unadjusted Error (Note 8)	$V_{REF}/2$ = 2.500 V_{DC}			±½	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±½	LSB
ADC0804: Total Unadjusted Error (Note 8)	$V_{REF}/2$ = 2.500 V_{DC}			±1	LSB
ADC0805: Total Unadjusted Error (Note 8)	$V_{REF}/2$-No Connection			±1	LSB
$V_{REF}/2$ Input Resistance (Pin 9)	ADC0801/02/03/05 ADC0804 (Note 9)	2.5 0.75	8.0 1.1		kΩ kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	Gnd−0.05		V_{CC} + 0.05	V_{DC}
DC Common-Mode Error	Over Analog Input Voltage Range		±1/16	±⅛	LSB
Power Supply Sensitivity	V_{CC} = 5 V_{DC} ±10% Over Allowed $V_{IN}(+)$ and $V_{IN}(-)$ Voltage Range (Note 4)		±1/16	±⅛	LSB

AC Electrical Characteristics

The following specifications apply for V_{CC} = 5 V_{DC} and T_A = 25°C unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_C	Conversion Time	f_{CLK} = 640 kHz (Note 6)	103		114	μs
T_C	Conversion Time	(Note 5, 6)	66		73	$1/f_{CLK}$
f_{CLK}	Clock Frequency Clock Duty Cycle	V_{CC} = 5V, (Note 5) (Note 5)	100 40	640	1460 60	kHz %
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$ = 0 V_{DC}, f_{CLK} = 640 kHz	8770		9708	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}$ = 0 V_{DC} (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	C_L = 100 pF		135	200	ns
t_{1H}, t_{0H}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	C_L = 10 pF, R_L = 10k (See TRI-STATE Test Circuits)		125	200	ns
t_{WI}, t_{RI}	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (1)	Logical "1" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 5.25 V_{DC}	2.0		15	V_{DC}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Continued)

The following specifications apply for $V_{CC} = 5V_{DC}$ and $T_{MIN} \leq T_A \leq T_{MAX}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	$V_{CC} = 4.75\ V_{DC}$			0.8	V_{DC}
I_{IN} (1)	Logical "1" Input Current (All Inputs)	$V_{IN} = 5\ V_{DC}$		0.005	1	μA_{DC}
I_{IN} (0)	Logical "0" Input Current (All Inputs)	$V_{IN} = 0\ V_{DC}$	−1	−0.005		μA_{DC}
CLOCK IN AND CLOCK R						
V_T+	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_T-	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis $(V_T+)-(V_T-)$		0.6	1.3	2.0	V_{DC}
V_{OUT} (0)	Logical "0" CLK R Output Voltage	$I_O = 360\ \mu A$ $V_{CC} = 4.75\ V_{DC}$			0.4	V_{DC}
V_{OUT} (1)	Logical "1" CLK R Output Voltage	$I_O = -360\ \mu A$ $V_{CC} = 4.75\ V_{DC}$	2.4			V_{DC}
DATA OUTPUTS AND $\overline{INTR}$						
V_{OUT} (0)	Logical "0" Output Voltage Data Outputs $\overline{INTR}$ Output	$I_{OUT} = 1.6$ mA, $V_{CC} = 4.75\ V_{DC}$ $I_{OUT} = 1.0$ mA, $V_{CC} = 4.75\ V_{DC}$			0.4 0.4	V_{DC} V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	$I_O = -360\ \mu A$, $V_{CC} = 4.75\ V_{DC}$	2.4			V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	$I_O = -10\ \mu A$, $V_{CC} = 4.75\ V_{DC}$	4.5			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	$V_{OUT} = 0\ V_{DC}$ $V_{OUT} = 5\ V_{DC}$	−3		3	μA_{DC} μA_{DC}
I_{SOURCE}		V_{OUT} Short to Gnd, $T_A = 25°C$	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC}, $T_A = 25°C$	9.0	16		mA_{DC}
POWER SUPPLY						
I_{CC}	Supply Current (Includes Ladder Current) ADC0801/02/03/04LCJ/05 ADC0804LCN/LCV/LCWM	$f_{CLK} = 640$ kHz, $V_{REF}/2 =$ NC, $T_A = 25°C$ and $\overline{CS} = 5V$		1.1 1.9	1.8 2.5	mA mA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{DC}.

Note 4: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Accuracy is guaranteed at $f_{CLK} = 640$ kHz. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see *Figure 2* and section 2.0.

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and *Figure 5*.

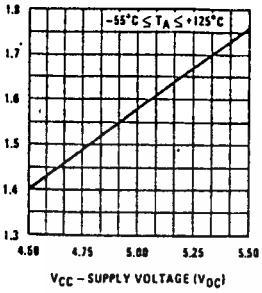
Note 9: The $V_{REF}/2$ pin is the center point of a two-resistor divider connected from V_{CC} to ground. In all versions of the ADC0801, ADC0802, ADC0803, and ADC0805, and in the ADC0804LCJ, each resistor is typically 16 kΩ. In all versions of the ADC0804 except the ADC0804LCJ, each resistor is typically 2.2 kΩ.

Note 10: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

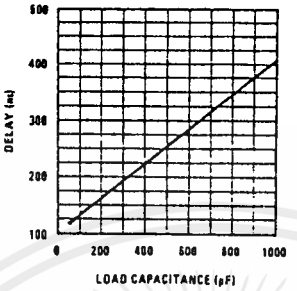
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ypical Performance Characteristics

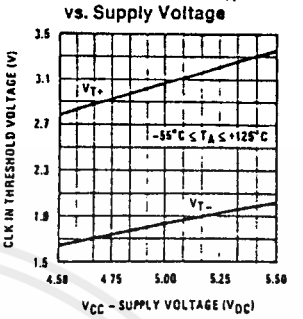
Logic Input Threshold Voltage vs. Supply Voltage



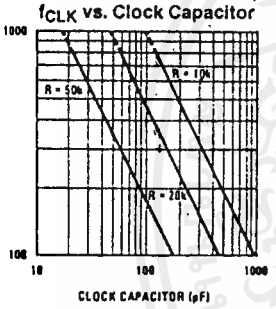
Delay From Falling Edge of $\overline{RD}$ to Output Data Valid vs. Load Capacitance



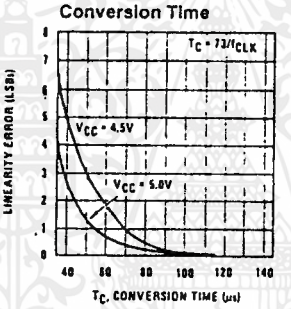
CLK IN Schmitt Trip Levels vs. Supply Voltage



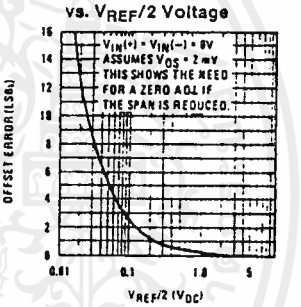
f_{CLK} vs. Clock Capacitor



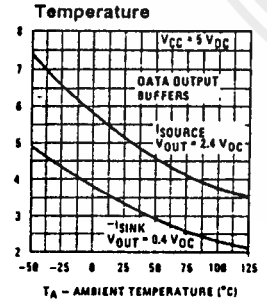
Full-Scale Error vs Conversion Time



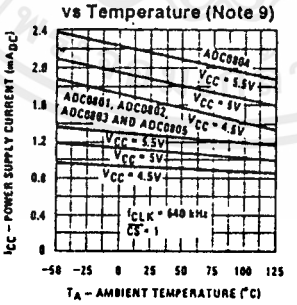
Effect of Unadjusted Offset Error vs $V_{REF}/2$ Voltage



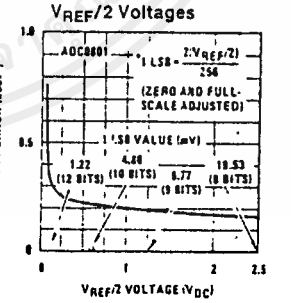
Output Current vs Temperature



Power Supply Current vs Temperature (Note 9)



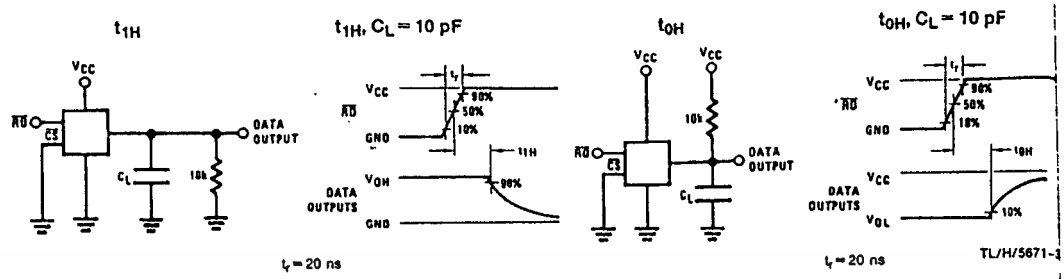
Linearity Error at Low $V_{REF}/2$ Voltages



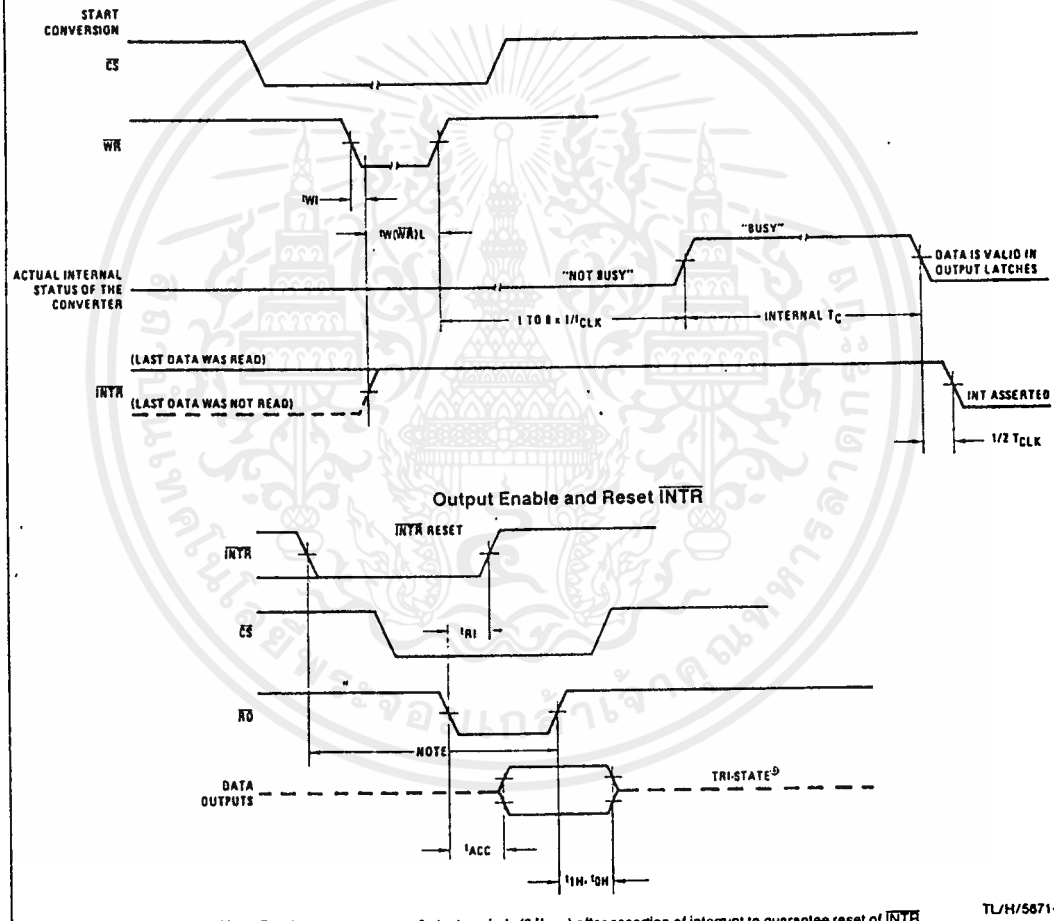
TL/H/5671–2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRI-STATE Test Circuits and Waveforms



Timing Diagrams (All timing is measured from the 50% voltage points)

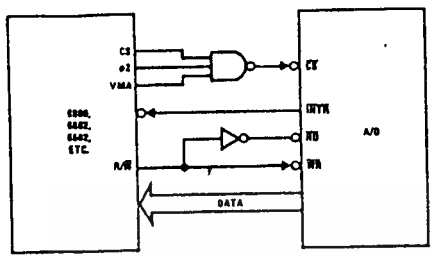


Note: Read strobe must occur 8 clock periods ($8/f_{CLK}$) after assertion of interrupt to guarantee reset of $\overline{INTR}$.

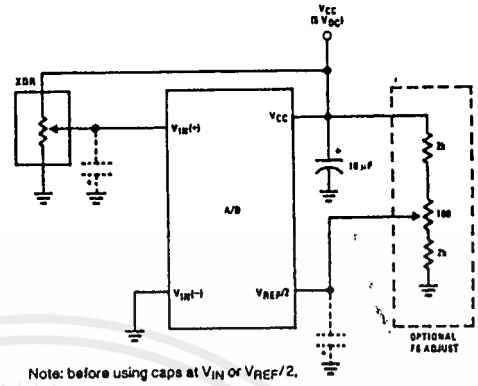
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

6800 Interface

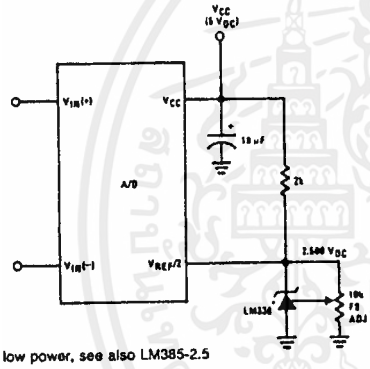


Ratiometric with Full-Scale Adjust



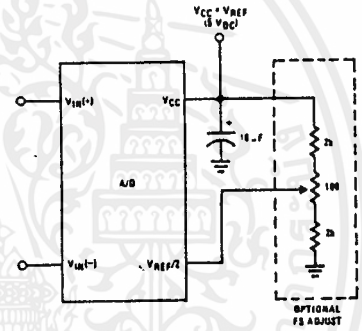
Note: before using caps at V_{IN} or $V_{REF}/2$, see section 2.3.2 Input Bypass Capacitors.

Absolute with a 2.500V Reference

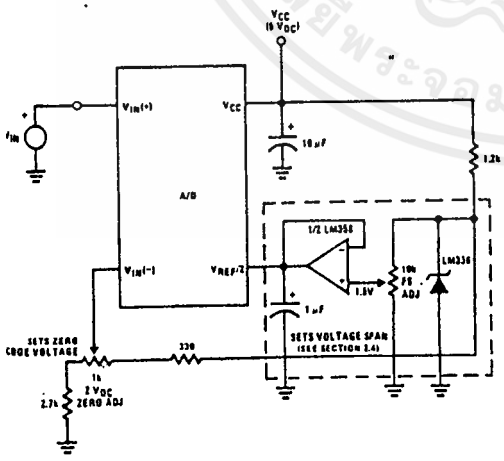


*For low power, see also LM385-2.5

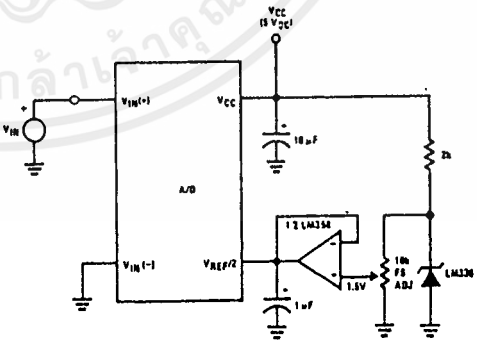
Absolute with a 5V Reference



Zero-Shift and Span Adjust: $2V \leq V_{IN} \leq 5V$



Span Adjust: $0V \leq V_{IN} \leq 3V$

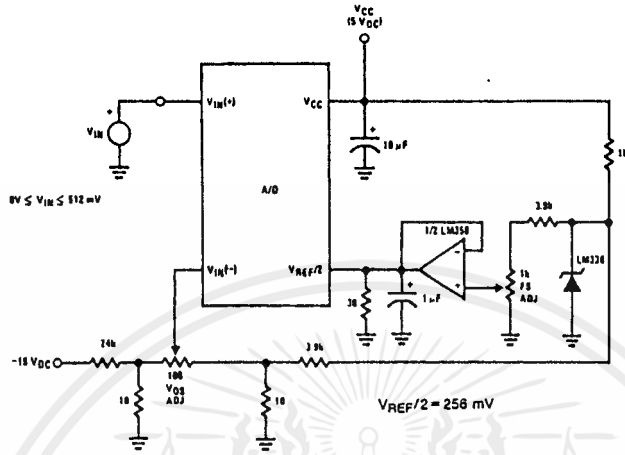


TL/H/5671–5

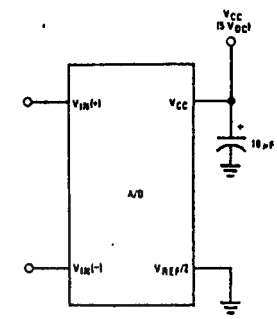
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

Directly Converting a Low-Level Signal



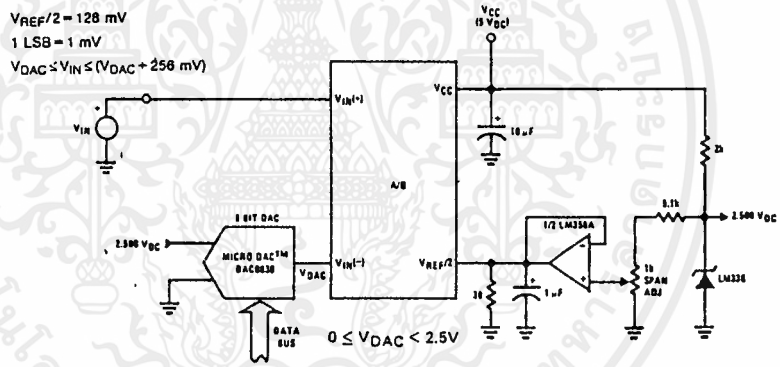
A µP Interfaced Comparator



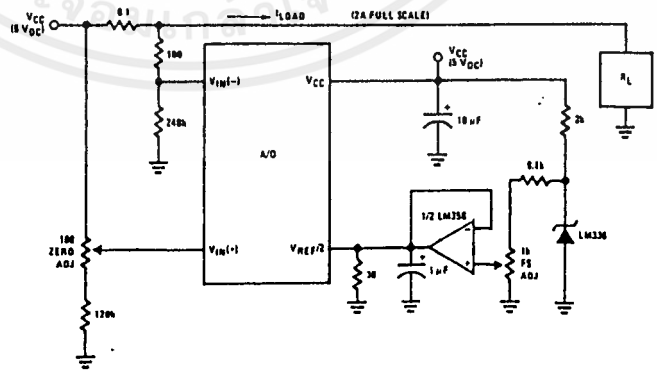
For: $V_{IN}(+) > V_{IN}(-)$
Output = FF_{HEX}
For: $V_{IN}(+) < V_{IN}(-)$
Output = 00_{HEX}

1 mV Resolution with µP Controlled Range

$V_{REF}/2$ = 128 mV
1 LSB = 1 mV
$V_{DAC} \leq V_{IN} \leq (V_{DAC} + 256\ mV)$



Digitizing a Current Flow

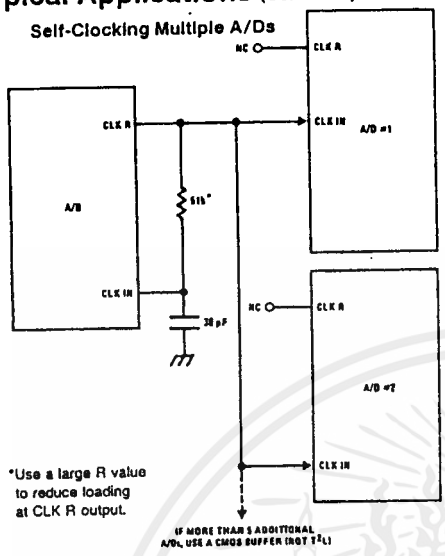


TL/H/5671

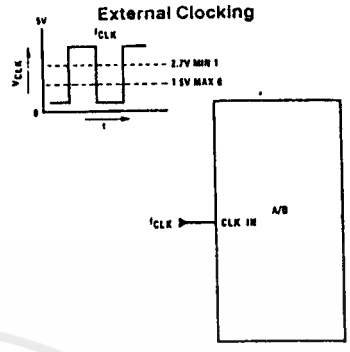
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

Self-Clocking Multiple A/Ds

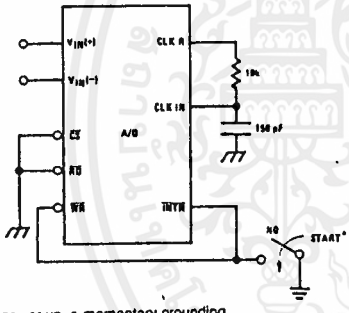


External Clocking



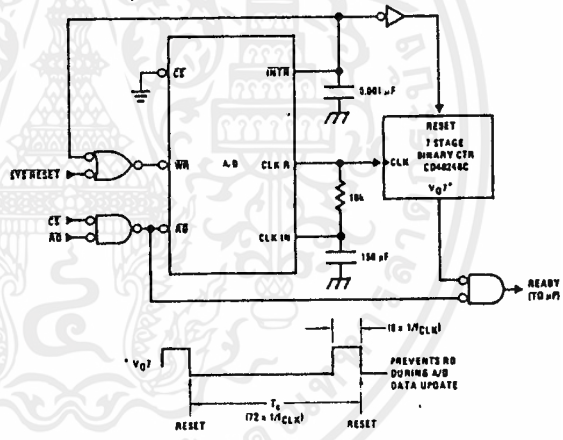
100 kHz ≤ f_{CLK} ≤ 1460 kHz

Self-Clocking in Free-Running Mode

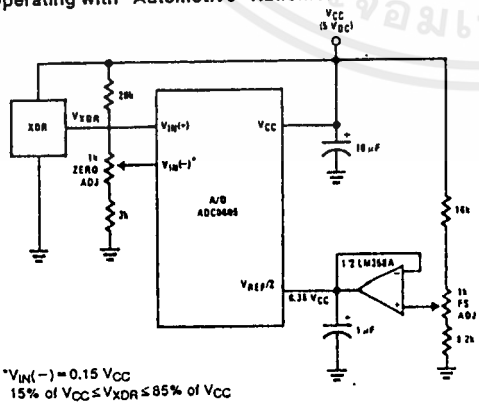


*After power-up, a momentary grounding of the $\overline{WR}$ input is needed to guarantee operation.

μP Interface for Free-Running A/D

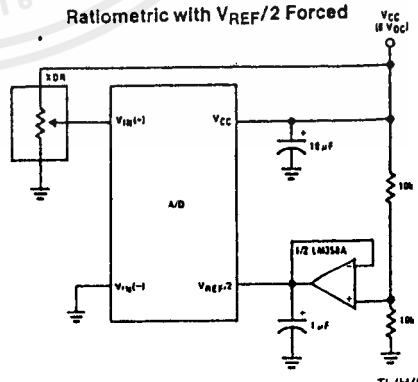


Operating with "Automotive" Ratiometric Transducers



*$V_{IN}(-) = 0.15\ V_{CC}$
15% of $V_{CC} \leq V_{XDR} \leq 85\%$ of V_{CC}

Ratiometric with $V_{REF}/2$ Forced

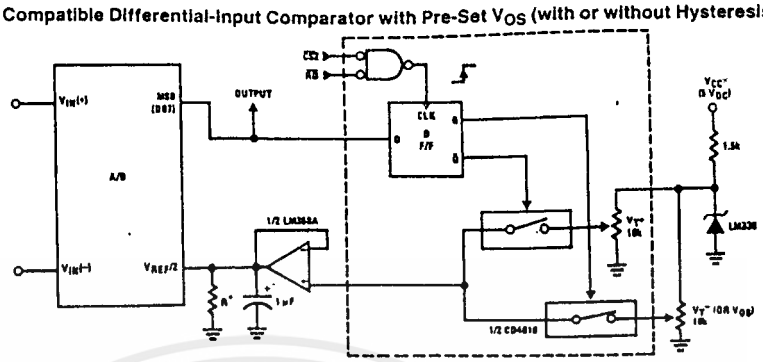


TL/H/5671–7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

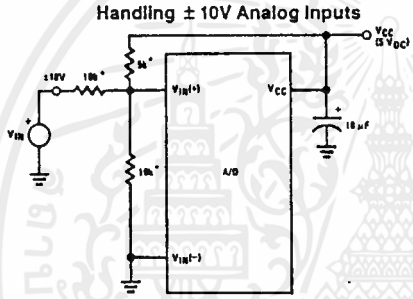
Typical Applications (Continued)

μP Compatible Differential-Input Comparator with Pre-Set V_{OS} (with or without Hysteresis)



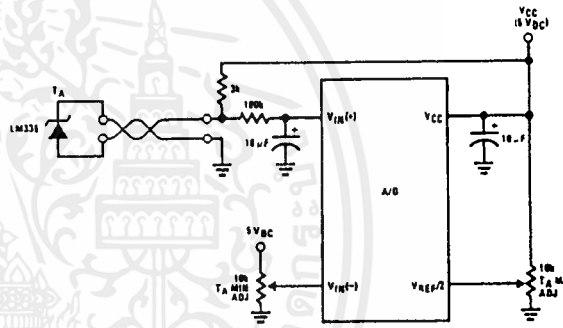
*See *Figure 5* to select R value
DB7 = "1" for $V_{IN}(+) > V_{IN}(-) + (V_{REF}/2)$
Omit circuitry within the dotted area if hysteresis is not needed

Handling ± 10V Analog Inputs

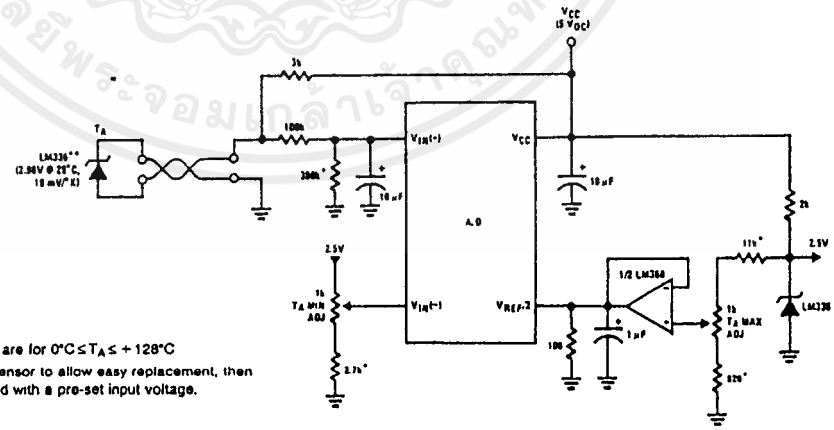


*Beckman Instruments #694-3-R10K resistor array

Low-Cost, μP Interfaced, Temperature-to-Digital Converter



μP Interfaced Temperature-to-Digital Converter



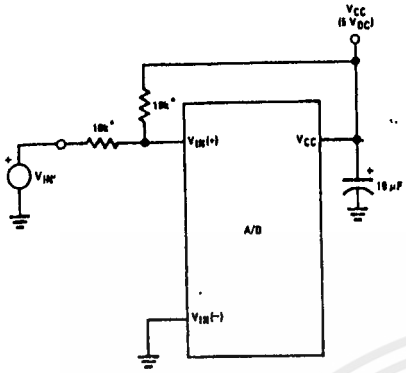
*Circuit values shown are for $0°C \leq T_A \leq +128°C$
**Can calibrate each sensor to allow easy replacement, then A/D can be calibrated with a pre-set input voltage.

TL/H/5671

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

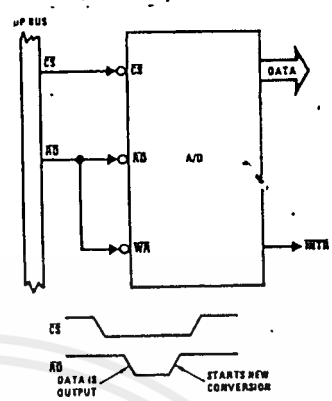
Handling ±5V Analog Inputs



TL/H/5671-33

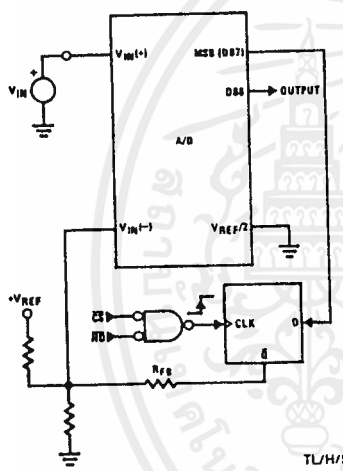
*Beckman Instruments # 694-3-R10K resistor array

Read-Only Interface



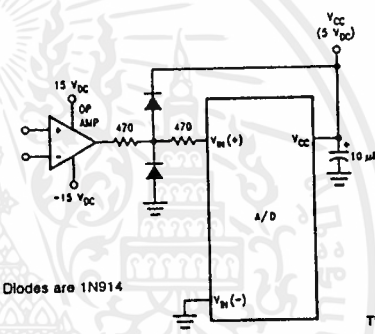
TL/H/5671-34

µP Interfaced Comparator with Hysteresis



TL/H/5671-35

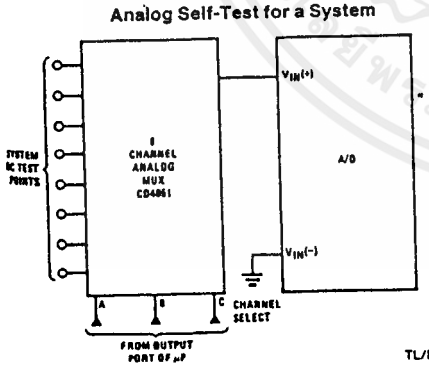
Protecting the Input



Diodes are 1N914

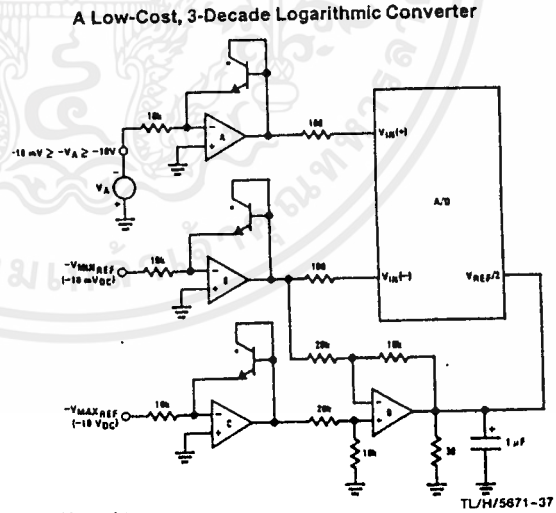
TL/H/5671-9

Analog Self-Test for a System



TL/H/5671-36

A Low-Cost, 3-Decade Logarithmic Converter



TL/H/5671-37

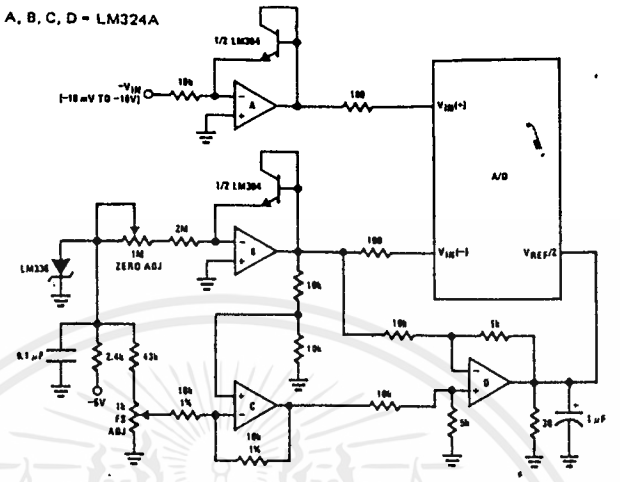
*LM389 transistors

A, B, C, D = LM324A quad op amp

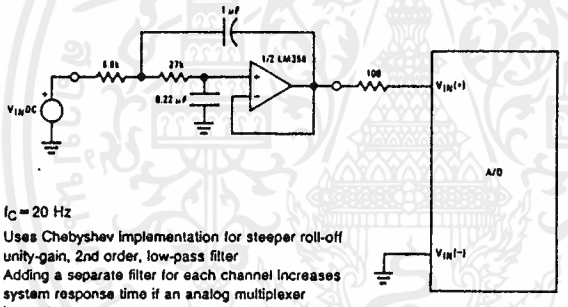
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

3-Decade Logarithmic A/D Converter



Noise Filtering the Analog Input

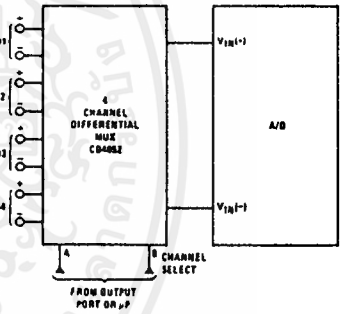


$f_C = 20$ Hz

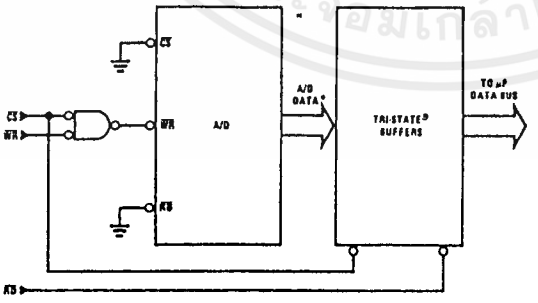
Uses Chebyshev implementation for steeper roll-off unity-gain, 2nd order, low-pass filter

Adding a separate filter for each channel increases system response time if an analog multiplexer is used

Multiplexing Differential Inputs

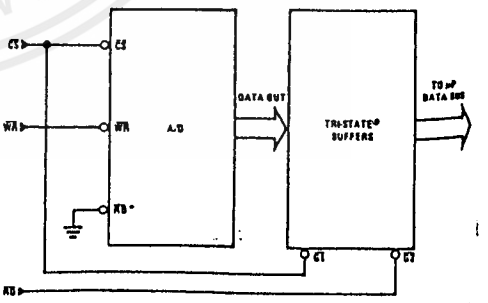


Output Buffers with A/D Data Enabled



*A/D output data is updated 1 CLK period prior to assertion of $\overline{INTR}$

Increasing Bus Drive and/or Reducing Time on Bus



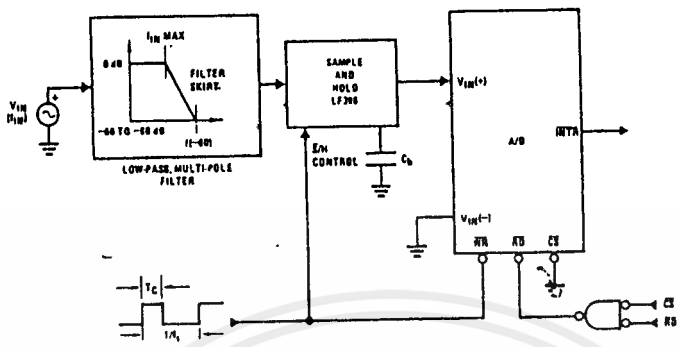
TL/H/5671–10

*Allows output data to set-up at falling edge of $\overline{CS}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

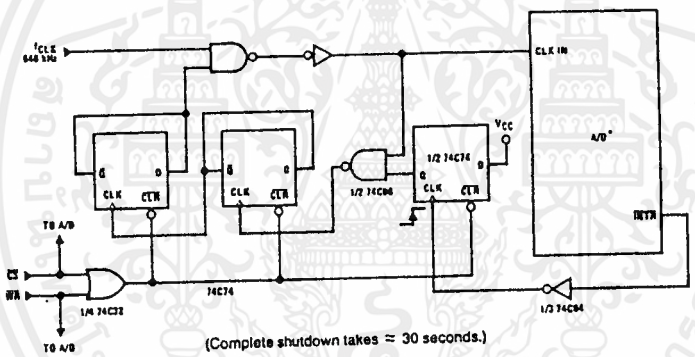
Sampling an AC Input Signal



Note 1: Oversample whenever possible (keep $f_S > 2f(-60)$) to eliminate input frequency folding (aliasing) and to allow for the skirt response of the filter.

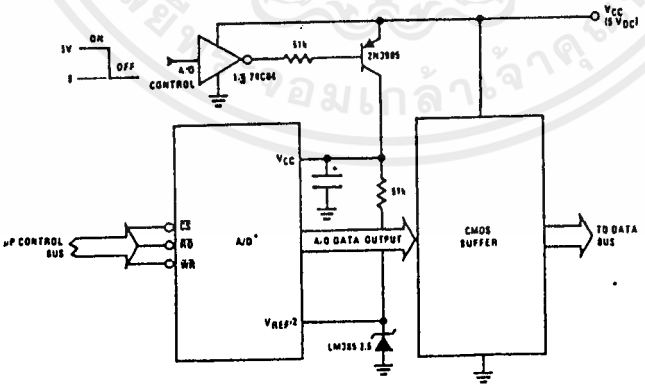
Note 2: Consider the amplitude errors which are introduced within the passband of the filter.

70% Power Savings by Clock Gating



(Complete shutdown takes ≈ 30 seconds.)

Power Savings by A/D and V_{REF} Shutdown



TL/H/5671–11

*Use ADC0801, 02, 03 or 05 for lowest power consumption.

Note: Logic inputs can be driven to V_{CC} with A/D supply at zero volts.

Buffer prevents data bus from overdriving output of A/D when in shutdown mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

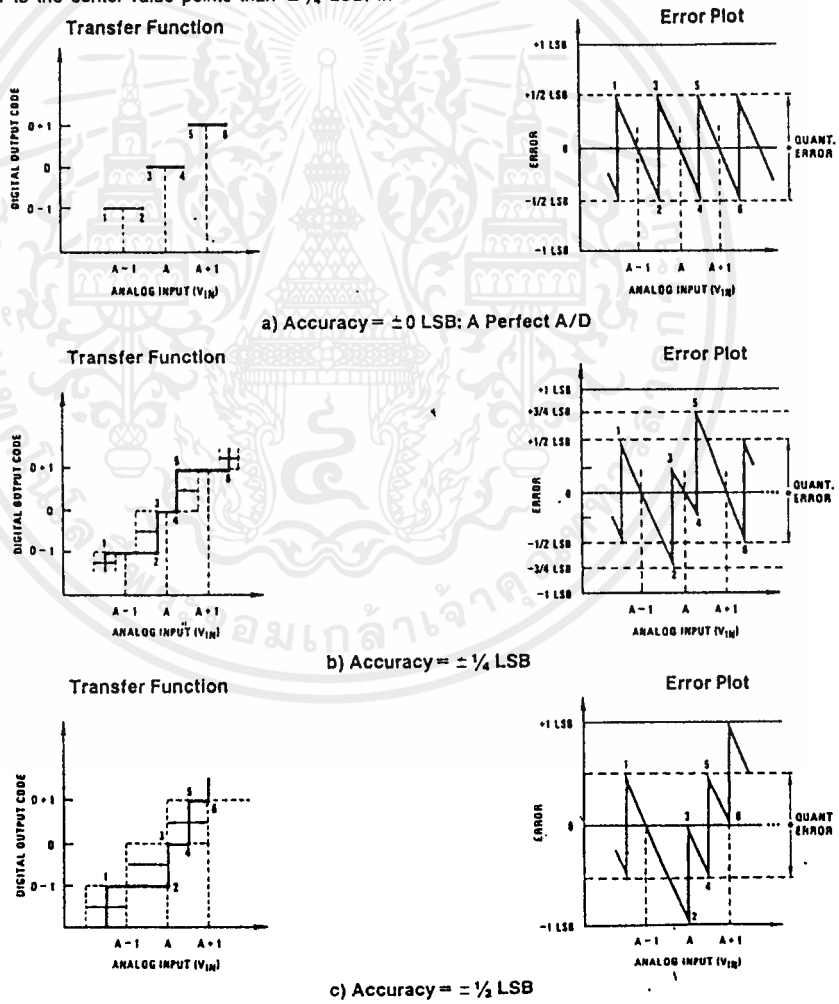
1.0 UNDERSTANDING A/D ERROR SPECS

A perfect A/D transfer characteristic (staircase waveform) is shown in *Figure 1a*. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53 mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes that correspond to these inputs are shown as D−1, D, and D+1. For the perfect A/D, not only will center-value (A−1, A, A+1, ...) analog inputs produce the correct output ditigal codes, but also each riser (the transitions between adjacent output codes) will be located ± ½ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages that extend ± ½ LSB from the ideal center-values. Each tread (the range of analog input voltage that provides the same digital output code) is therefore 1 LSB wide.

Figure 1b shows a worst case error plot for the ADC0801. All center-valued inputs are guaranteed to produce the correct output codes and the adjacent risers are guaranteed to be no closer to the center-value points than ± ¼ LSB. In other words, if we apply an analog input equal to the center-value ± ¼ LSB, *we guarantee* that the A/D will produce the correct digital code. The maximum range of the position of the code transition is indicated by the horizontal arrow and is guaranteed to be no more than ½ LSB.

The error curve of *Figure 1c* shows a worst case error plot for the ADC0802. Here we guarantee that if we apply an analog input equal to the LSB analog voltage center-value the A/D will produce the correct digital code.

Next to each transfer function is shown the corresponding error plot. Many people may be more familiar with error plots than transfer functions. The analog input voltage to the A/D is provided by either a linear ramp or by the discrete output steps of a high-resolution DAC. Notice that the error is continuously displayed and includes the quantization uncertainty of the A/D. For example the error at point 1 of *Figure 1a* is + ½ LSB because the digital code appeared ½ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude.



TL/H/5671–12

FIGURE 1. Clarifying the Error Specs of an A/D Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

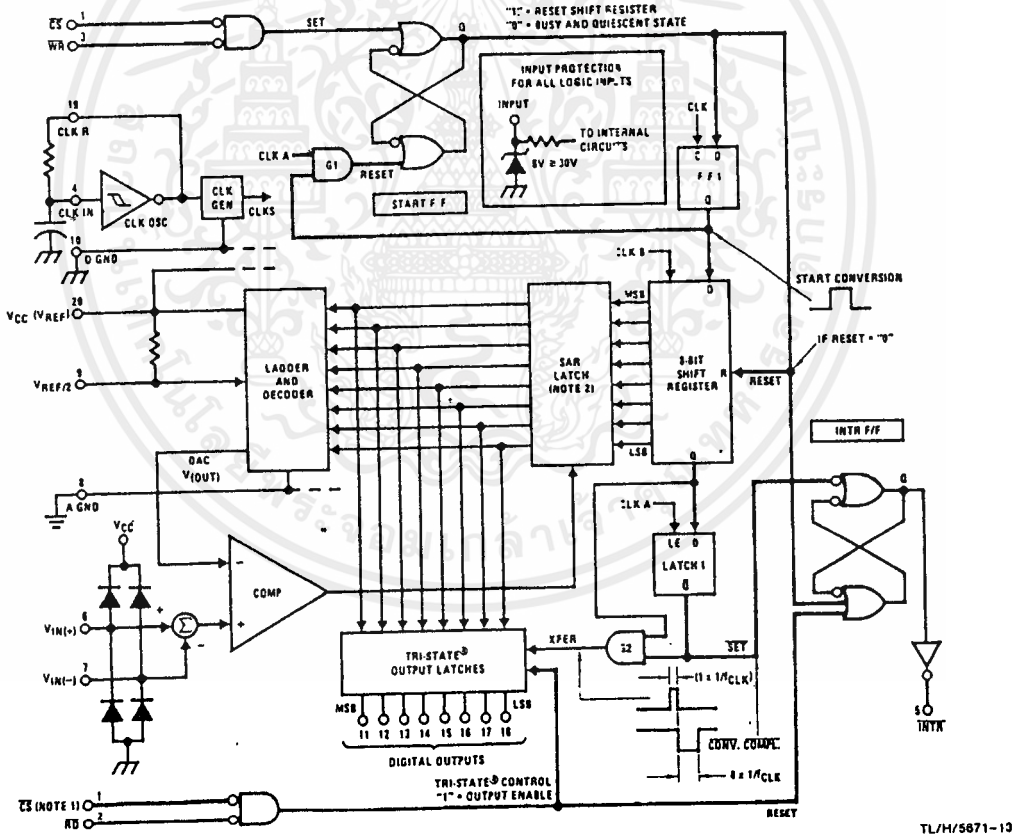
2.0 FUNCTIONAL DESCRIPTION

The ADC0801 series contains a circuit equivalent of the 256R network. Analog switches are sequenced by successive approximation logic to match the analog difference input voltage [$V_{IN}(+) - V_{IN}(-)$] to a corresponding tap on the R network. The most significant bit is tested first and after 8 comparisons (64 clock cycles) a digital 8-bit binary code (1111 1111 = full-scale) is transferred to an output latch and then an interrupt is asserted ($\overline{INTR}$ makes a high-to-low transition). A conversion in process can be interrupted by issuing a second start command. The device may be operated in the free-running mode by connecting $\overline{INTR}$ to the $\overline{WR}$ input with $\overline{CS} = 0$. To ensure start-up under all possible conditions, an external $\overline{WR}$ pulse is required during the first power-up cycle.

On the high-to-low transition of the $\overline{WR}$ input the internal SAR latches and the shift register stages are reset. As long as the $\overline{CS}$ input and $\overline{WR}$ input remain low, the A/D will remain in a reset state. *Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition.*

A functional diagram of the A/D converter is shown in *Figure 2*. All of the package pinouts are shown and the major logic control paths are drawn in heavier weight lines.

The converter is started by having $\overline{CS}$ and $\overline{WR}$ simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt (INTR) F/F and inputs a "1" to the D flop, F/F1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of F/F1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either $\overline{WR}$ or $\overline{CS}$ is a "1") the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would momentarily be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This logic therefore allows for wide $\overline{CS}$ and $\overline{WR}$ signals and the converter will start after at least one of these signals returns high and the internal clocks again provide a reset signal for the start F/F.



Note 1: $\overline{CS}$ shown twice for clarity.

Note 2: SAR = Successive Approximation Register.

FIGURE 2. Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

After the "1" is clocked through the 8-bit shift register (which completes the SAR search) it appears as the input to the D-type latch, LATCH 1. As soon as this "1" is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the TRI-STATE output latches. When LATCH 1 is subsequently enabled, the Q output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the $\overline{INTR}$ input signal.

Note that this $\overline{SET}$ control of the INTR F/F remains low for 8 of the external clock periods (as the internal clocks run at 1/8 of the frequency of the external clock). If the data output is continuously enabled ($\overline{CS}$ and $\overline{RD}$ both held low), the $\overline{INTR}$ output will still signal the end of conversion (by a high-to-low transition), because the $\overline{SET}$ input can control the Q output of the INTR F/F even though the RESET input is constantly at a "1" level in this operating mode. This $\overline{INTR}$ output will therefore stay low for the duration of the $\overline{SET}$ signal, which is 8 periods of the external clock frequency (assuming the A/D is not started during this interval).

When operating in the free-running or continuous conversion mode ($\overline{INTR}$ pin tied to $\overline{WR}$ and $\overline{CS}$ wired low—see also section 2.8), the START F/F is SET by the high-to-low transition of the $\overline{INTR}$ signal. This resets the SHIFT REGISTER which causes the input to the D-type latch, LATCH 1, to go low. As the latch enable input is still present, the $\overline{Q}$ output will go high, which then allows the INTR F/F to be RESET. This reduces the width of the resulting $\overline{INTR}$ output pulse to only a few propagation delays (approximately 300 ns).

When data is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the TRI-STATE output latches will be enabled to provide the 8-bit digital outputs.

2.1 Digital Control Inputs

The digital control inputs ($\overline{CS}$, $\overline{RD}$, and $\overline{WR}$) meet standard T^2L logic voltage levels. These signals have been renamed when compared to the standard A/D Start and Output Enable labels. In addition, these inputs are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the $\overline{CS}$ input (pin 1) can be grounded and the standard A/D Start function is obtained by an active low pulse applied at the $\overline{WR}$ input (pin 3) and the Output Enable function is caused by an active low pulse at the $\overline{RD}$ input (pin 2).

2.2 Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D has additional applications flexibility due to the analog differential voltage input. The $V_{IN}(-)$ input (pin 7) can be used to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4 mA–20 mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{IN}(+)$ and $V_{IN}(-)$ is 4-1/2 clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_e(MAX) = (V_P)(2\pi f_{cm})\left(\frac{4.5}{f_{CLK}}\right),$$

where:

ΔV_e is the error voltage due to sampling delay

V_P is the peak value of the common-mode voltage

f_{cm} is the common-mode frequency

As an example, to keep this error to 1/4 LSB (~5 mV) when operating with a 60 Hz common-mode frequency, f_{cm}, and using a 640 kHz A/D clock, f_{CLK}, would allow a peak value of the common-mode voltage, V_P, which is given by:

$$V_P = \frac{[\Delta V_{e(MAX)}(f_{CLK})]}{(2\pi f_{cm})(4.5)}$$

or

$$V_P = \frac{(5 \times 10^{-3})(640 \times 10^3)}{(6.28)(60)(4.5)}$$

which gives

$V_P \approx 1.9V$.

The allowed range of analog input voltages usually places more severe restrictions on input common-mode noise levels.

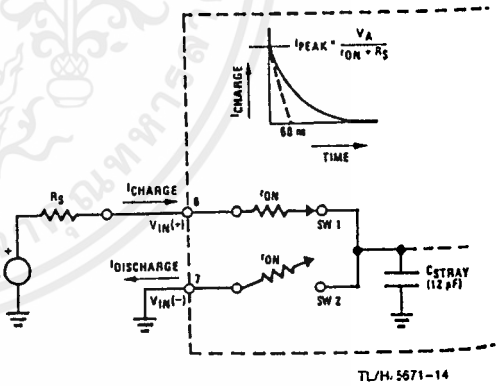
An analog input voltage with a reduced span and a relatively large zero offset can be handled easily by making use of the differential input (see section 2.4 Reference Voltage).

2.3 Analog Inputs

2.3.1 Input Current

Normal Mode

Due to the internal switching action, displacement currents will flow at the analog inputs. This is due to on-chip stray capacitance to ground as shown in *Figure 3*.



r_{ON} of SW 1 and SW 2 = 5 kΩ

$\tau = r_{ON} C_{STRAY} \approx 5\ k\Omega \times 12\ pF = 60\ ns$

FIGURE 3. Analog Input Impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

The voltage on this capacitance is switched and will result in currents entering the $V_{IN}(+)$ input pin and leaving the $V_{IN}(-)$ input which will depend on the analog differential input voltage levels. These current transients occur at the leading edge of the internal clocks. They rapidly decay and *do not cause errors* as the on-chip comparator is strobed at the end of the clock period.

Fault Mode

If the voltage source applied to the $V_{IN}(+)$ or $V_{IN}(-)$ pin exceeds the allowed operating range of $V_{CC}+50$ mV, large input currents can flow through a parasitic diode to the V_{CC} pin. If these currents can exceed the 1 mA max allowed spec, an external diode (1N914) should be added to bypass this current to the V_{CC} pin (with the current bypassed with this diode, the voltage at the $V_{IN}(+)$ pin can exceed the V_{CC} voltage by the forward voltage of this diode).

2.3.2 Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances of the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{IN}(+)$ input voltage at full-scale. For continuous conversions with a 640 kHz clock frequency with the $V_{IN}(+)$ input at 5V, this DC current is at a maximum of approximately 5 μA. Therefore, *bypass capacitors should not be used at the analog inputs or the $V_{REF}/2$ pin* for high resistance sources (> 1 kΩ). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the detrimental effects of the voltage drop across this input resistance, which is due to the average value of the input current, can be eliminated with a full-scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage.

2.3.3 Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used, *will not cause errors* as the input currents settle out prior to the comparison time. If a low pass filter is required in the system, use a low valued series resistor (≤ 1 kΩ) for a passive RC section or add an op amp RC active low pass filter. For low source resistance applications, (≤ 1 kΩ), a 0.1 μF bypass capacitor at the inputs will prevent noise pickup due to series lead inductance of a long wire. A 100Ω series resistor can be used to isolate this capacitor—both the R and C are placed outside the feedback loop—from the output of an op amp, if used.

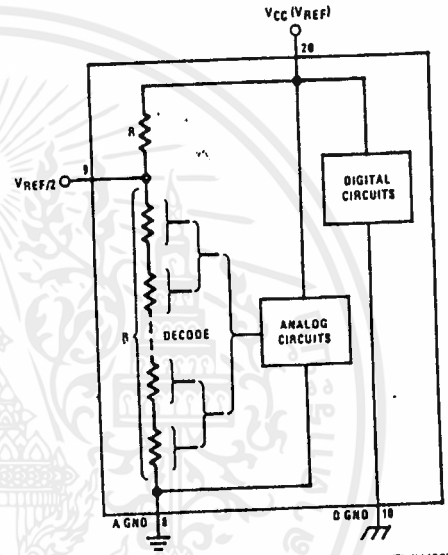
2.3.4 Noise

The leads to the analog inputs (pin 6 and 7) should be kept as short as possible to minimize input noise coupling. Both noise and undesired digital clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below 5 kΩ. Larger values of source resistance can cause undesired system noise pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate system noise pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see section 2.3.1.). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be eliminated by doing a full-scale adjustment of the A/D (adjust $V_{REF}/2$ for a proper full-scale reading—see section 2.5.2 on Full-Scale Adjustment) with the source resistance and input bypass capacitor in place.

2.4 Reference Voltage

2.4.1 Span Adjust

For maximum applications flexibility, these A/Ds have been designed to accommodate a 5 V_{DC}, 2.5 V_{DC} or an adjusted voltage reference. This has been achieved in the design of the IC as shown in *Figure 4*.



TL/H/5671–15

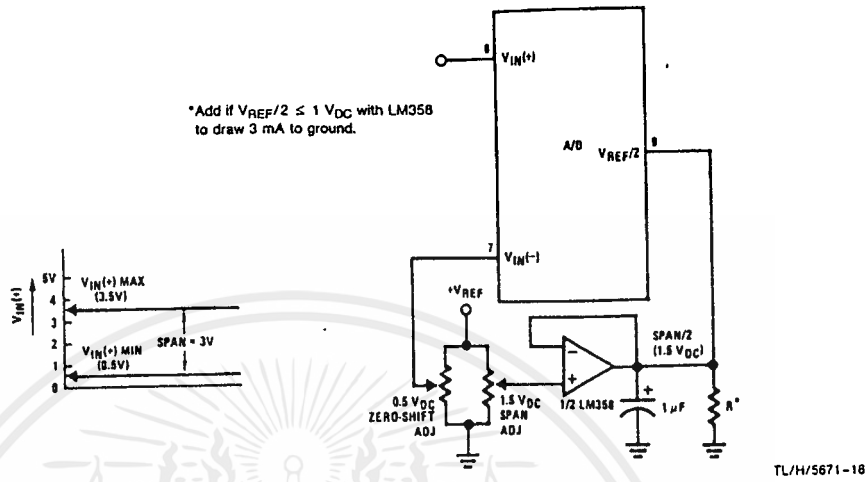
FIGURE 4. The $V_{REFERENCE}$ Design on the IC

Notice that the reference voltage for the IC is either ½ of the voltage applied to the V_{CC} supply pin, or is equal to the voltage that is externally forced at the $V_{REF}/2$ pin. This allows for a ratiometric voltage reference using the V_{CC} supply, a 5 V_{DC} reference voltage can be used for the V_{CC} supply or a voltage less than 2.5 V_{DC} can be applied to the $V_{REF}/2$ input for increased application flexibility. The internal gain to the $V_{REF}/2$ input is 2, making the full-scale differential input voltage twice the voltage at pin 9.

An example of the use of an adjusted reference voltage is to accommodate a reduced span—or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5 V_{DC} to 3.5 V_{DC}, instead of 0V to 5 V_{DC}, the span would be 3V as shown in *Figure 5*. With 0.5 V_{DC} applied to the $V_{IN}(-)$ pin to absorb the offset, the reference voltage can be made equal to ½ of the 3V span or 1.5 V_{DC}. The A/D now will encode the $V_{IN}(+)$ signal from 0.5V to 3.5 V with the 0.5V input corresponding to zero and the 3.5 V_{DC} input corresponding to full-scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)



a) Analog Input Signal Example

b) Accommodating an Analog Input from 0.5V (Digital Out = 00_{HEX}) to 3.5V (Digital Out = FF_{HEX})

FIGURE 5. Adapting the A/D Analog Input Voltages to Match an Arbitrary Input Signal Range

2.4.2 Reference Accuracy Requirements

The converter can be operated in a ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. The ADC0805 is specified particularly for use in ratiometric applications with no adjustments required. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important factors in the accuracy of the A/D converter. For $V_{REF}/2$ voltages of 2.4 V_{DC} nominal value, initial errors of ±10 mV_{DC} will cause conversion errors of ±1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20 mV (5V span) to 10 mV and 1 LSB at the $V_{REF}/2$ input becomes 5 mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the magnitude of the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full-scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive. The LM336B 2.5V IC reference diode (from National Semiconductor) has a temperature stability of 4.8 mV typ (6 mV max) over $0°C \leq T_A \leq +70°C$. Other temperature range parts are also available.

2.5 Errors and Reference Voltage Adjustments

2.5.1 Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D $V_{IN}(-)$ input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the V_{IN} (−) input and applying a small magnitude positive voltage to the V_{IN} (+) input. Zero error is the difference between the actual DC input voltage that is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal ½ LSB value (½ LSB = 9.8 mV for $V_{REF}/2 = 2.500\ V_{DC}$).

2.5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage that is 1½ LSB less than the desired analog full-scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9 or the V_{CC} supply if pin 9 is not used) for a digital output code that is just changing from 1111 1110 to 1111 1111.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

2.5.3 Adjusting for an Arbitrary Analog Input Voltage Range

If the analog zero voltage of the A/D is shifted away from ground (for example, to accommodate an analog input signal that does not go to ground) this new zero reference should be properly adjusted first. A $V_{IN}(+)$ voltage that equals this desired zero reference plus ½ LSB (where the LSB is calculated for the desired analog span, 1 LSB = analog span/256) is applied to pin 6 and the zero reference voltage at pin 7 should then be adjusted to just obtain the 00_{HEX} to 01_{HEX} code transition.

The full-scale adjustment should then be made (with the proper $V_{IN}(-)$ voltage applied) by forcing a voltage to the $V_{IN}(+)$ input which is given by:

$$V_{IN}(+) \text{ fs adj} = V_{MAX} - 1.5\left[\frac{(V_{MAX} - V_{MIN})}{256}\right],$$

where:

V_{MAX} = The high end of the analog input range

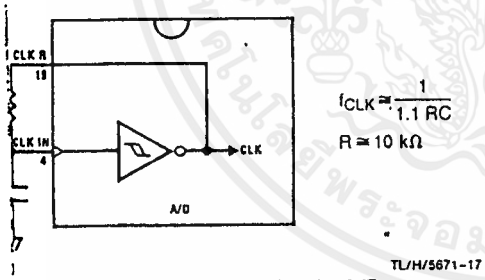
and

V_{MIN} = the low end (the offset zero) of the analog range. (both are ground referenced.)

The $V_{REF}/2$ (or V_{CC}) voltage is then adjusted to provide a code change from FE_{HEX} to FF_{HEX}. This completes the adjustment procedure.

2.6 Clocking Option

The clock for the A/D can be derived from the CPU clock or an external RC can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in *figure 6.*



TL/H/5671–17

FIGURE 6. Self-Clocking the A/D

Heavy capacitive or DC loading of the clock R pin should be avoided as this will disturb normal converter operation. Loads less than 50 pF, such as driving up to 7 A/D converter clock inputs from a single clock R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power logic buffer or PNP input logic should be used to minimize clock loading on the clock R pin (do not use a standard TTL driver).

2.7 Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in process is not allowed to be completed, therefore the data of the previous conversion remains in this latch. The $\overline{INTR}$ output simply remains at the "1" level.

2.8 Continuous Conversions

For operation in the free-running mode an initializing pulse should be used, following power-up, to ensure circuit operation. In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the $\overline{INTR}$ output. This $\overline{WR}$ and $\overline{INTR}$ node should be momentarily forced to logic low following a power-up cycle to guarantee operation.

2.9 Driving the Data Bus

This MOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in TRI-STATE (high impedance mode). Backplane bussing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see typical characteristics curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be TRI-STATE buffers (low power Schottky such as the DM74LS240 series is recommended) or special higher drive current products which are designed as bus drivers. High current bipolar bus drivers with PNP inputs are recommended.

2.10 Power Supplies

Noise spikes on the V_{CC} supply line can cause conversion errors as the comparator will respond to this noise. A low inductance tantalum filter capacitor should be used close to the converter V_{CC} pin and values of 1 μF or greater are recommended. If an unregulated voltage is available in the system, a separate LM340LAZ-5.0, TO-92, 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V_{CC} supply.

2.11 Wiring and Hook-Up Precautions

Standard digital wire wrap sockets are not satisfactory for breadboarding this A/D converter. Sockets on PC boards can be used and all logic signal wires and leads should be grouped and kept as far away as possible from the analog signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup, therefore shielded leads may be necessary in many applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

A single point analog ground that is separate from the logic ground points should be used. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any $V_{REF}/2$ bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of ¼ LSB can usually be traced to improper board layout and wiring (see section 2.5.1 for measuring the zero error).

3.0 TESTING THE A/D CONVERTER

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in *Figure 7*.

For ease of testing, the $V_{REF}/2$ (pin 9) should be supplied with 2.560 V_{DC} and a V_{CC} supply voltage of 5.12 V_{DC} should be used. This provides an LSB value of 20 mV.

If a full-scale adjustment is to be made, an analog input voltage of 5.090 V_{DC} (5.120−1½ LSB) should be applied to the $V_{IN}(+)$ pin with the $V_{IN}(-)$ pin grounded. The value of the $V_{REF}/2$ input voltage should then be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of $V_{REF}/2$ should then be used for all the tests.

The digital output LED display can be decoded by dividing the 8 bits into 2 hex characters, the 4 most significant (MS) and the 4 least significant (LS). Table I shows the fractional binary equivalent of these two 4-bit groups. By adding the voltages obtained from the "VMS" and "VLS" columns in Table I, the nominal value of the digital display (when $V_{REF}/2$ = 2.560V) can be determined. For example, for an output LED display of 1011 0110 or B6 (in hex), the voltage values from the table are 3.520 + 0.120 or 3.640 V_{DC}. These voltage values represent the center-values of a perfect A/D converter. The effects of quantization error have to be accounted for in the interpretation of the test results.

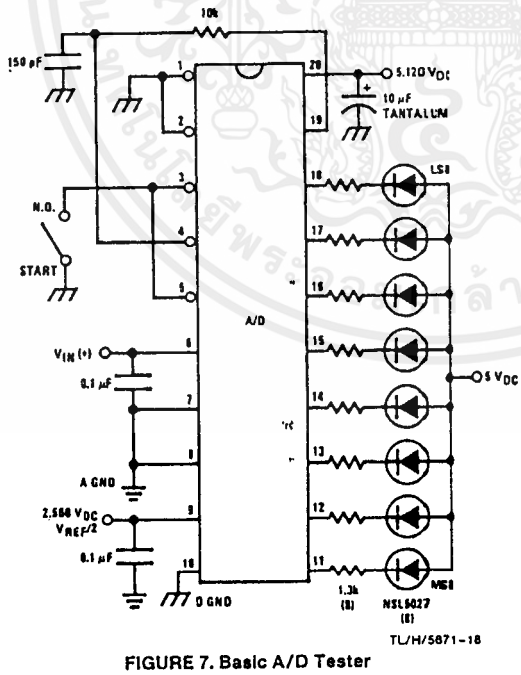


FIGURE 7. Basic A/D Tester

For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be expressed as either analog voltages or differences in 2 digital words.

A basic A/D tester that uses a DAC and provides the error as an analog output voltage is shown in *Figure 8*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to read the difference voltage, "A−C", directly. The analog input voltage can be supplied by a low frequency ramp generator and an X-Y plotter can be used to provide analog error (Y axis) versus analog input (X axis).

For operation with a microprocessor or a computer-based test system, it is more convenient to present the errors digitally. This can be done with the circuit of *Figure 9*, where the output code transitions can be detected as the 10-bit DAC is incremented. This provides ¼ LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.

4.0 MICROPROCESSOR INTERFACING

To dicuss the interface with 8080A and 6800 microprocessors, a common sample subroutine structure is used. The microprocessor starts the A/D, reads and stores the results of 16 successive conversions, then returns to the user's program. The 16 data bytes are stored in 16 successive memory locations. All Data and Addresses will be given in hexadecimal form. Software and hardware details are provided separately for each type of microprocessor.

4.1 Interfacing 8080 Microprocessor Derivatives (8048, 8085)

This converter has been designed to directly interface with derivatives of the 8080 microprocessor. The A/D can be mapped into memory space (using standard memory address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{I/O\ R}$ and $\overline{I/O\ W}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15 as they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. An example of an A/D in I/O space is shown in *Figure 10*.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

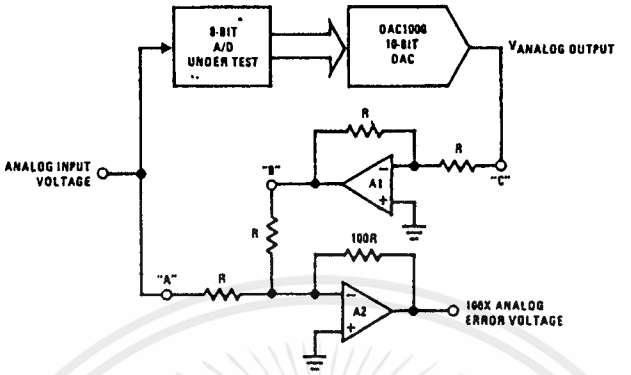


FIGURE 8. A/D Tester with Analog Error Output

TL/H/5671-19

FIGURE 9. Basic "Digital" A/D Tester

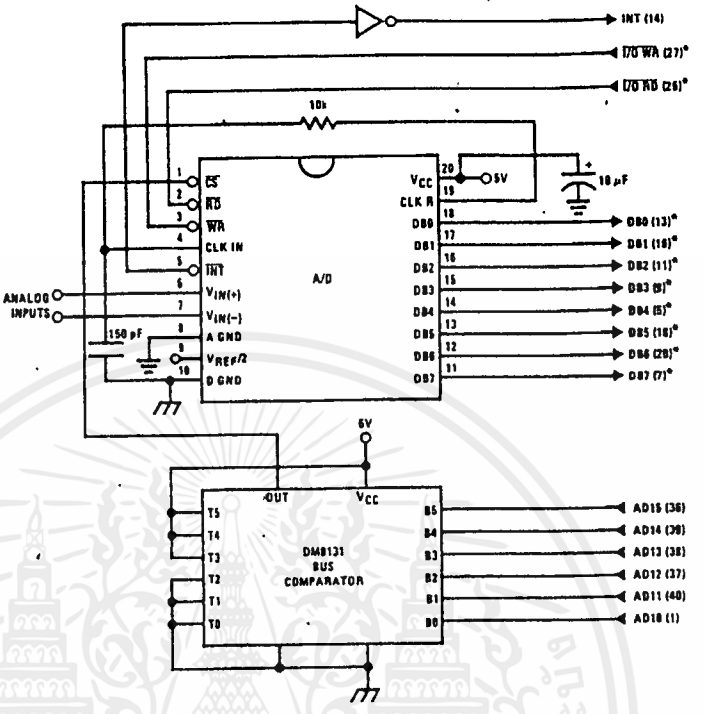
TABLE I. DECODING THE DIGITAL OUTPUT LEDs

HEX	BINARY	FRACTIONAL BINARY VALUE FOR		OUTPUT VOLTAGE CENTER VALUES WITH $V_{REF}/2 = 2.560\ V_{DC}$	
		MS GROUP	LS GROUP	VMS GROUP*	VLS GROUP*
F	1 1 1 1	15/16	15/256	4.800	0.300
E	1 1 1 0	7/8	7/128	4.480	0.280
D	1 1 0 1	13/16	13/256	4.160	0.260
C	1 1 0 0	3/4	3/64	3.840	0.240
B	1 0 1 1	11/16	11/256	3.520	0.220
A	1 0 1 0	5/8	5/128	3.200	0.200
9	1 0 0 1	9/16	9/256	2/880	0.180
8	1 0 0 0	1/2	1/32	2/560	0.160
7	0 1 1 1	7/16	7/256	2.240	0.140
6	0 1 1 0	3/8	3/128	1.920	0.120
5	0 1 0 1	5/16	2/256	1.600	0.100
4	0 1 0 0	1/4	1/64	1/280	0.080
3	0 0 1 1	3/16	3/256	0.960	0.060
2	0 0 1 0	1/8	1/128	0.640	0.040
1	0 0 0 1	1/16	1/256	0.320	0.020
0	0 0 0 0			0	0

*Display Output = VMS Group + VLS Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)



TL/H/5671–20

Note 1: *Pin numbers for the DP8228 system controller, others are INS8080A.

Note 2: Pin 23 of the INS8228 must be tied to +12V through a 1 kΩ resistor to generate the RST 7 instruction when an interrupt is acknowledged as required by the accompanying sample program.

FIGURE 10. ADC0801–INS8080A CPU Interface

SAMPLE PROGRAM FOR *FIGURE 10* ADC0801–INS8080A CPU INTERFACE

```
0038    C3 00 03    RST 7:      JMP     LD DATA
  •         •           •
  •         •           •
0100    21 00 02    START:      LXI H 0200H     ; HL pair will point to
                                                ; data storage locations
0103    31 00 04    RETURN:     LXI SP 0400H    ; Initialize stack pointer (Note 1)
0106    7D                      MOV A, L        ; Test # of bytes entered
0107    FE 0F                   CPI 0F H        ; If # = 16. JMP to
0109    CA 13 01                JZ CONT         ; user program
010C    D3 E0                   OUT E0 H        ; Start A/D
010E    FB                      EI              ; Enable interrupt
010F    00          LOOP:       NOP             ; Loop until end of
0110    C3 0F 01                JMP LOOP        ; conversion
0113        •       CONT:           •
  •         •           •               •
  •         •           •               •
  •         •       (User program to    •
  •         •       process data)       •
  •         •           •               •
  •         •           •               •
0300    DB E0       LD DATA:    IN E0 H         ; Load data into accumulator
0302    77                      MOV M, A        ; Store data
0303    23                      INX H           ; Increment storage pointer
0304    C3 03 01                JMP RETURN
```

Note 1: The stack pointer must be dimensioned because a RST 7 instruction pushes the PC onto the stack.

Note 2: All address used were arbitrarily chosen.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

The standard control bus signals of the 8080 $\overline{CS}$, $\overline{RD}$ and $\overline{WR}$) can be directly wired to the digital control inputs of the A/D and the bus timing requirements are met to allow both starting the converter and outputting the data onto the data bus. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100 pF.

4.1.1 Sample 8080A CPU Interfacing Circuitry and Program

The following sample program and associated hardware shown in *Figure 10* may be used to input data from the converter to the INS8080A CPU chip set (comprised of the INS8080A microprocessor, the INS8228 system controller and the INS8224 clock generator). For simplicity, the A/D is controlled as an I/O device, specifically an 8-bit bi-directional port located at an arbitrarily chosen port address, E0. The TRI-STATE output capability of the A/D eliminates the need for a peripheral interface device, however address decoding is still required to generate the appropriate $\overline{CS}$ for the converter.

It is important to note that in systems where the A/D converter is 1-of-8 or less I/O mapped devices, no address decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as $\overline{CS}$ inputs—one for each I/O device.

4.1.2 INS8048 Interface

The INS8048 interface technique with the ADC0801 series (see *Figure 11*) is simpler than the 8080A CPU interface. There are 24 I/O lines and three test input lines in the 8048. With these extra I/O lines available, one of the I/O lines (bit 0 of port 1) is used as the chip select signal to the A/D, thus eliminating the use of an external address decoder. Bus control signals $\overline{RD}$, $\overline{WR}$ and $\overline{INT}$ of the 8048 are tied directly to the A/D. The 16 converted data words are stored at on-chip RAM locations from 20 to 2F (Hex). The $\overline{RD}$ and $\overline{WR}$ signals are generated by reading from and writing into a dummy address, respectively. A sample interface program is shown below.

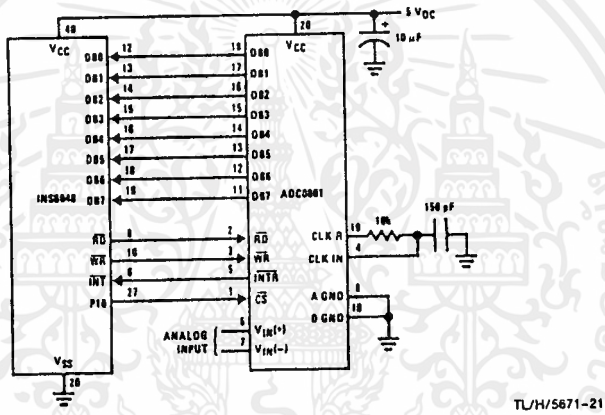


FIGURE 11. INS8048 Interface

SAMPLE PROGRAM FOR *FIGURE 11* INS8048 INTERFACE

```
04 10                    JMP     10H          ;Program starts at addr 10
                         ORG     3H
04 50                    JMP     50H          ;Interrupt jump vector
                         ORG     10H          ;Main program
99 FE                    ANL     P1, #0FEH    ;Chip select
81                       MOVX    A, @R1       ;Read in the 1st data
                                              ;to reset the intr
89 01          START:    ORL     P1, #1       ;Set port pin high
B8 20                    MOV     R0, #20H     ;Data address
B9 FF                    MOV     R1, #0FFH    ;Dummy address
BA 10                    MOV     R2, #10H     ;Counter for 16 bytes
23 FF          AGAIN:    MOV     A, #0FFH     ;Set ACC for intr loop
99 FE                    ANL     P1, #0FEH    ;Send CS (bit 0 of P1)
91                       MOVX    @R1, A       ;Send WR out
05                       EN      I            ;Enable interrupt
96 21          LOOP:     JNZ     LOOP         ;Wait for interrupt
EA 1B                    DJNZ    R2, AGAIN    ;If 16 bytes are read
00                       NOP                  ;go to user's program
00                       NOP
                         ORG     50H
81             INDATA:   MOVX    A, @R1       ;Input data, CS still low
A0                       MOV     @R0, A       ;Store in memory
18                       INC     R0           ;Increment storage counter
89 01                    ORL     P1, #1       ;Reset CS signal
27                       CLR     A            ;Clear ACC to get out of
93                       RETR                 ;the interrupt loop
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

4.2 Interfacing the Z-80

The Z-80 control bus is slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals are used which have to be combined with the generalized strobes to provide the equivalent 8080 signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in *Figure 13*.

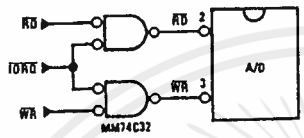


FIGURE 13. Mapping the A/D as an I/O Device for Use with the Z-80 CPU

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

4.3 Interfacing 6800 Microprocessor Derivatives (6502, etc.)

The control bus for the 6800 microprocessor derivatives does not use the $\overline{RD}$ and $\overline{WR}$ strobe signals. Instead it employs a single R/$\overline{W}$ line and additional timing, if needed, can be derived fom the $\phi 2$ clock. All I/O devices are memory mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. *Figure 14* shows an interface schematic where the A/D is memory mapped in the 6800 system. For simplicity, the $\overline{CS}$ decoding is shown using ½ DM8092. Note that in many 6800 systems, an already decoded $\overline{4}$–$\overline{5}$ line is brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D, provided that no other devices are addressed at HX ADDR: 4XXX or 5XXX.

The following subroutine performs essentially the same function as in the case of the 8080A interface and it can be called from anywhere in the user's program.

In *Figure 15* the ADC0801 series is interfaced to the M6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter, (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory mapped in the M6800 system and no $\overline{CS}$ decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D $\overline{RD}$ pin can be grounded.

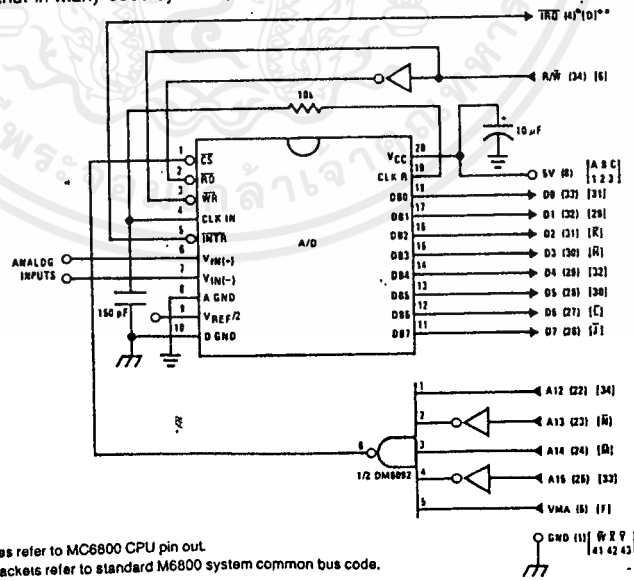
A sample interface program equivalent to the previous one is shown below *Figure 15*. The PIA Data and Control Registers of Port B are located at HEX addresses 8006 and 8007, respectively.

5.0 GENERAL APPLICATIONS

The following applications show some interesting uses for the A/D. The fact that one particular microprocessor is used is not meant to be restrictive. Each of these application circuits would have its counterpart using any microprocessor that is desired.

5.1 Multiple ADC0801 Series to MC6800 CPU Interface

To transfer analog data from several channels to a single microprocessor system, a multiple converter scheme presents several advantages over the conventional multiplexer single-converter approach. With the ADC0801 series, the differential inputs allow individual span adjustment for each channel. Furthermore, all analog input channels are sensed simultaneously, which essentially divides the microprocessor's total system servicing time by the number of channels, since all conversions occur simultaneously. This scheme is shown in *Figure 16*.



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.
Note 2: Number or letters in brackets refer to standard M6800 system common bus code.

TL/H/5671–24

FIGURE 14. ADC0801-MC6800 CPU Interface

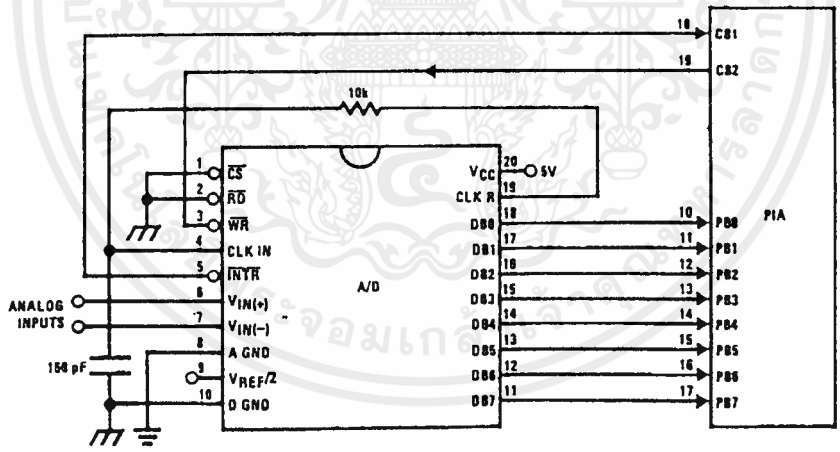
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 14* ADC0801-MC6800 CPU INTERFACE

```
0010   DF 36      DATAIN   STX    TEMP2    ; Save contents of X
0012   CE 00 2C            LDX    #$002C   ; Upon IRQ low CPU
0015   FF FF F8            STX    $FFF8    ; jumps to 002C
0018   B7 50 00            STAA   $5000    ; Start ADC0801
001B   0E                  CLI
001C   3E          CONVRT  WAI             ; Wait for interrupt
001D   DE 34               LDX    TEMP1
001F   8C 02 0F            CPX    #$020F   ; Is final data stored?
0022   27 14               BEQ    ENDP
0024   B7 50 00            STAA   $5000    ; Restarts ADC0801
0027   08                  INX
0028   DF 34               STX    TEMP1
002A   20 F0               BRA    CONVRT
002C   DE 34      INTRPT   LDX    TEMP1
002E   B6 50 00            LDAA   $5000    ; Read data
0031   A7 00               STAA   X        ; Store it at X
0033   3B                  RTI
0034   02 00      TEMP1    FDB    $0200    ; Starting address for
                                            ; data storage
0036   00 00      TEMP2    FDB    $0000
0038   CE 02 00   ENDP     LDX    #$0200   ; Reinitialize TEMP1
003B   DF 34               STX    TEMP1
003D   DE 36               LDX    TEMP2
003F   39                  RTS              ; Return from subroutine
                                            ; To user's program
```

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.



TL/H/5671-25

FIGURE 15. ADC0801-MC6820 PIA Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 15* ADC0801–MC6820 PIA INTERFACE

```
0010  CE 00 38   DATAIN   LDX    #$0038     ; Upon IRQ low CPU
0013  FF FF F8            STX    $FFF8      ; jumps to 0038
0016  B6 80 06            LDAA   PIAORB     ; Clear possible IRQ flags
0019  4F                  CLRA
001A  B7 80 07            STAA   PIACRB
001D  B7 80 06            STAA   PIAORB     ; Set Port B as input
0020  0E                  CLI
0021  C6 34               LDAB   #$34
0023  86 3D               LDAA   #$3D
0025  F7 80 07   CONVRT   STAB   PIACRB     ; Starts ADC0801
0028  B7 80 07            STAA   PIACRB
002B  3E                  WAI                ; Wait for interrupt
002C  DE 40               LDX    TEMP1
002E  8C 02 0F            CPX    #$020F     ; Is final data stored?
0031  27 0F               BEQ    ENDP
0033  08                  INX
0034  DF 40               STX    TEMP1
0036  20 ED               BRA    CONVRT
0038  DE 40   INTRPT      LDX    TEMP1
003A  B6 80 06            LDAA   PIAORB     ; Read data in
003D  A7 00               STAA   X          ; Store it at X
003F  3B                  RTI
0040  02 00   TEMP1       FDB    $0200      ; Starting address for
                                            ; data storage
0042  CE 02 00   ENDP     LDX    #$0200     ; Reinitialize TEMP1
0045  DF 40               STX    TEMP1
0047  39                 RTS                ; Return from subroutine
                PIAORB    EQU    $8006      ; To user's program
                PIACRB    EQU    $8007
```

The following schematic and sample subroutine (DATA IN) may be used to interface (up to) 8 ADC0801's directly to the MC6800 CPU. This scheme can easily be extended to allow the interface of more converters. In this configuration the converters are (arbitrarily) located at HEX address 5000 in the MC6800 memory space. To save components, the clock signal is derived from just one RC pair on the first converter. This output drives the other A/Ds.

All the converters are started simultaneously with a STORE instruction at HEX address 5000. Note that any other HEX address of the form 5XXX will be decoded by the circuit, pulling all the $\overline{CS}$ inputs low. This can easily be avoided by using a more definitive address decoding scheme. All the interrupts are ORed together to insure that all A/Ds have completed their conversion before the microprocessor is interrupted.

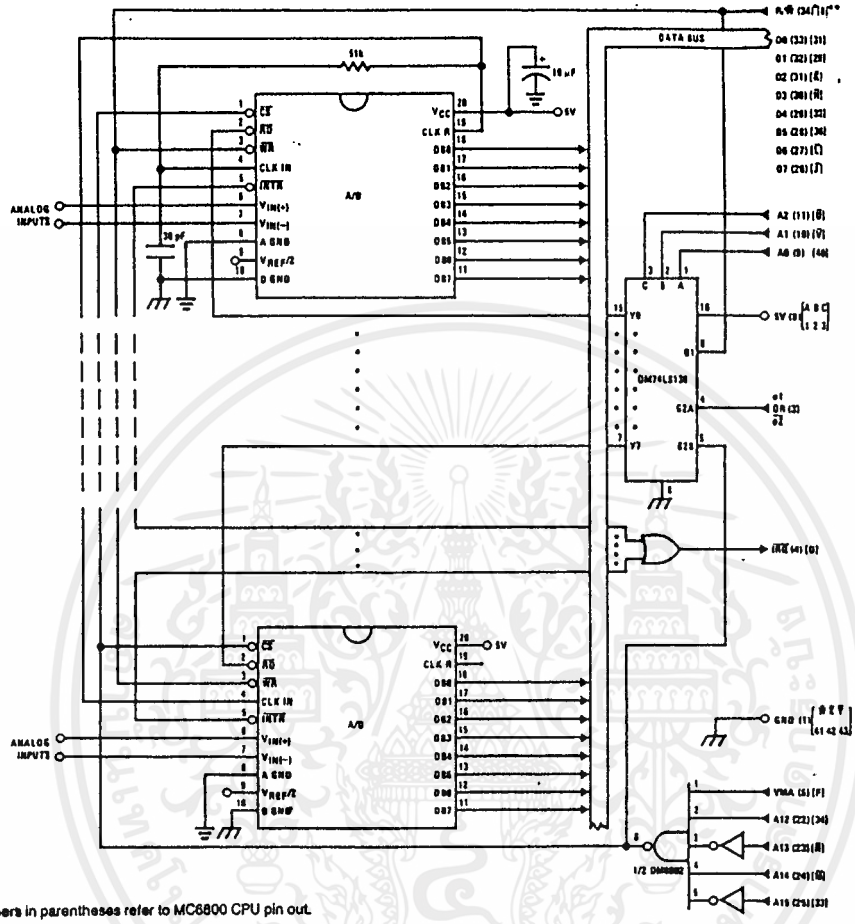
The subroutine, DATA IN, may be called from anywhere in the user's program. Once called, this routine initializes the CPU, starts all the converters simultaneously and waits for the interrupt signal. Upon receiving the interrupt, it reads the converters (from HEX addresses 5000 through 5007) and stores the data successively at (arbitrarily chosen) HEX addresses 0200 to 0207, before returning to the user's program. All CPU registers then recover the original data they had before servicing DATA IN.

5.2 Auto-Zeroed Differential Transducer Amplifier and A/D Converter

The differential inputs of the ADC0801 series eliminate the need to perform a differential to single ended conversion for a differential transducer. Thus, one op amp can be eliminated since the differential to single ended conversion is provided by the differential input of the ADC0801 series. In general, a transducer preamp is required to take advantage of the full A/D converter input dynamic range.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Numbers of letters in brackets refer to standard M6800 system common bus code.

TL/H/5671–26

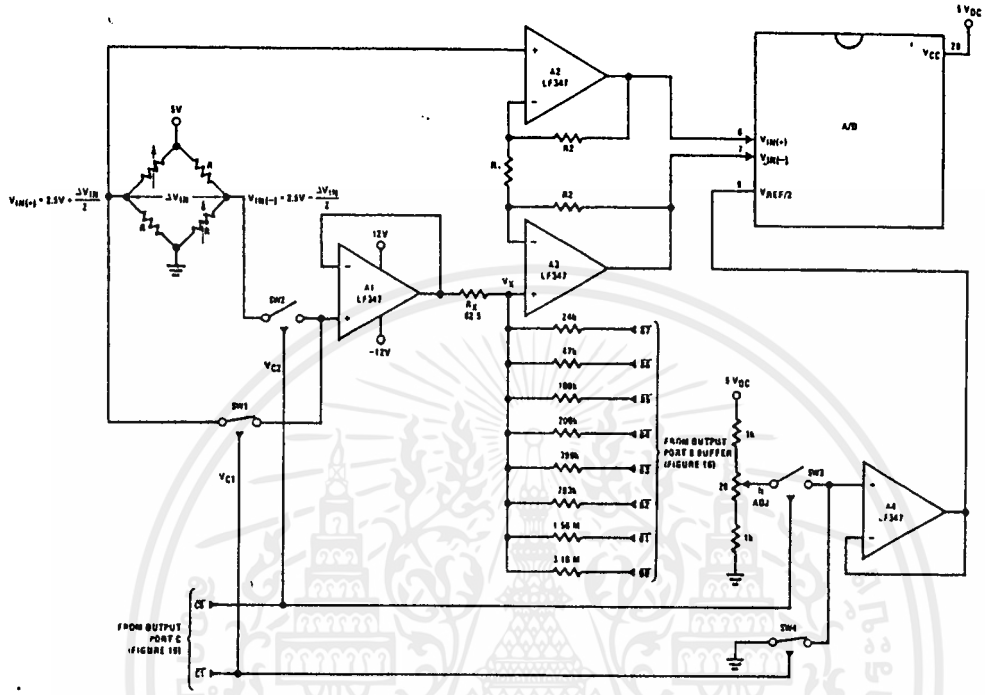
FIGURE 16. Interfacing Multiple A/Ds in an MC6800 System

SAMPLE PROGRAM FOR *FIGURE 16* INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE	MNEMONICS			COMMENTS
0010	DF 44	DATAIN	STX	TEMP	; Save Contents of X
0012	CE 00 2A		LDX	#$002A	; Upon $\overline{IRQ}$ LOW CPU
0015	FF FF F8		STX	$FFF8	; Jumps to 002A
0018	B7 50 00		STAA	$5000	; Starts all A/D's
001B	0E		CLI		
001C	3E		WAI		; Wait for interrupt
001D	CE 50 00		LDX	#$5000	
0020	DF 40		STX	INDEX1	; Reset both INDEX
0022	CE 02 00		LDX	#$0200	; 1 and 2 to starting
0025	DF 42		STX	INDEX2	; addresses
0027	DE 44		LDX	TEMP	
0029	39		RTS		; Return from subroutine
002A	DE 40	INTRPT	LDX	INDEX1	; INDEX1 → X
002C	A6 00		LDAA	X	; Read data in from A/D at X
002E	08		INX		; Increment X by one
002F	DF 40		STX	INDEX1	; X → INDEX1
0031	DE 42		LDX	INDEX2	; INDEX2 → X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

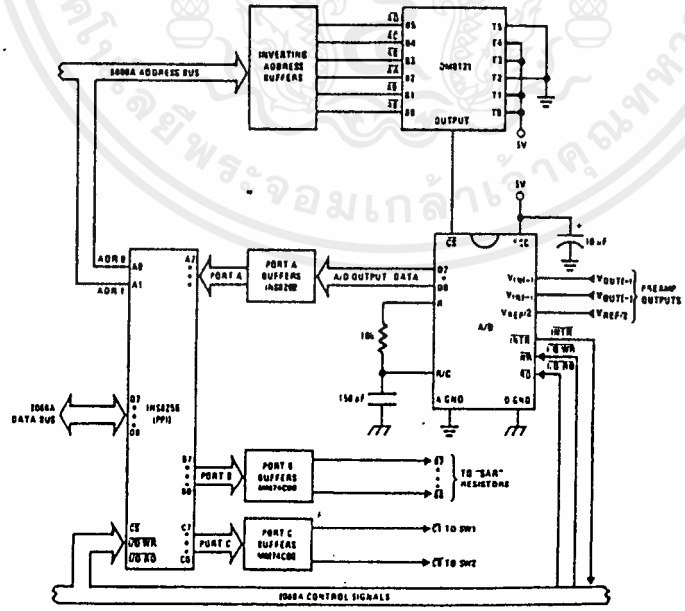


Note 1: R2 = 49.5 R1

Note 2: Switches are LMC13334 CMOS analog switches.

Note 3: The 9 resistors used in the auto-zero section can be ±5% tolerance.

FIGURE 17. Gain of 100 Differential Transducer Preamp



TL/H/5671-27

FIGURE 18. Microprocessor Interface Circuitry for Differential Preamp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 16* INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE	MNEMONICS			COMMENTS
0033	A7 00		STAA	X	;Store data at X
0035	8C 02 07		CPX	#$0207	;Have all A/D's been read?
0038	27 05		BEQ	RETURN	;Yes: branch to RETURN
003A	08		INX		;No: increment X by one
003B	DF 42		STX	INDEX2	;X → INDEX2
003D	20 EB		BRA	INTRPT	;Branch to 002A
003F	3B	RETURN	RTI		
0040	50 00	INDEX1	FDB	$5000	;Starting address for A/D
0042	02 00	INDEX2	FDB	$0200	;Starting address for data storage
0044	00 00	TEMP	FDB	$0000	

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.

For amplification of DC input signals, a major system error is the input offset voltage of the amplifiers used for the preamp. *Figure 17* is a gain of 100 differential preamp whose offset voltage errors will be cancelled by a zeroing subroutine which is performed by the INS8080A microprocessor system. The total allowable input offset voltage error for this preamp is only 50 μV for ¼ LSB error. This would obviously require very precise amplifiers. The expression for the differential output voltage of the preamp is:

$$V_O = \underbrace{[V_{IN}(+) - V_{IN}(-)]}_{\text{SIGNAL}} \underbrace{\left[1 + \frac{2R2}{R1}\right]}_{\text{GAIN}} +$$

$$\underbrace{(V_{OS_2} - V_{OS_1} - V_{OS_3} \pm I_X R_X)}_{\text{DC ERROR TERM}} \underbrace{\left(1 + \frac{2R2}{R1}\right)}_{\text{GAIN}}$$

where I_X is the current through resistor R_X. All of the offset error terms can be cancelled by making $\pm I_X R_X = V_{OS1} + V_{OS3} - V_{OS2}$. This is the principle of this auto-zeroing scheme.

The INS8080A uses the 3 I/O ports of an INS8255 Programable Peripheral Interface (PPI) to control the auto zeroing and input data from the ADC0801 as shown in *Figure 18*. The PPI is programmed for basic I/O operation (mode 0) with Port A being an input port and Ports B and C being output ports. Two bits of Port C are used to alternately open or close the 2 switches at the input of the preamp. Switch SW1 is closed to force the preamp's differential input to be zero during the zeroing subroutine and then opened and SW2 is then closed for conversion of the actual differential input signal. Using 2 switches in this manner eliminates concern for the ON resistance of the switches as they must conduct only the input bias current of the input amplifiers.

Output Port B is used as a successive approximation register by the 8080 and the binary scaled resistors in series with each output bit create a D/A converter. During the zeroing subroutine, the voltage at V_X increases or decreases as required to make the differential output voltage equal to zero. This is accomplished by ensuring that the voltage at the output of A1 is approximately 2.5V so that a logic "1" (5V) on any output of Port B will source current into node V_X thus raising the voltage at V_X and making the output differential more negative. Conversely, a logic "0" (0V) will pull current out of node V_X and decrease the voltage, causing the differential output to become more positive. For the resistor values shown, V_X can move ± 12 mV with a resolution of 50 μV, which will null the offset error term to ¼ LSB of full-scale for the ADC0801. It is important that the voltage levels that drive the auto-zero resistors be constant. Also, for symmetry, a logic swing of 0V to 5V is convenient. To achieve this, a CMOS buffer is used for the logic output signals of Port B and this CMOS package is powered with a stable 5V source. Buffer amplifier A1 is necessary so that it can source or sink the D/A output current.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A flow chart for the zeroing subroutine is shown in *Figure 19*. It must be noted that the ADC0801 series will output an all zero code when it converts a negative input [$V_{IN}(-) \geq V_{IN}(+)$]. Also, a logic inversion exists as all of the I/O ports are buffered with inverting gates.

Basically, if the data read is zero, the differential output voltage is negative, so a bit in Port B is cleared to pull V_X more negative which will make the output more positive for the next conversion. If the data read is not zero, the output voltage is positive so a bit in Port B is set to make V_X more positive and the output more negative. This continues for 8 approximations and the differential output eventually converges to within 5 mV of zero.

The actual program is given in *Figure 20*. All addresses used are compatible with the BLC 80/10 microcomputer system. In particular:

Port A and the ADC0801 are at port address E4

Port B is at port address E5

Port C is at port address E6

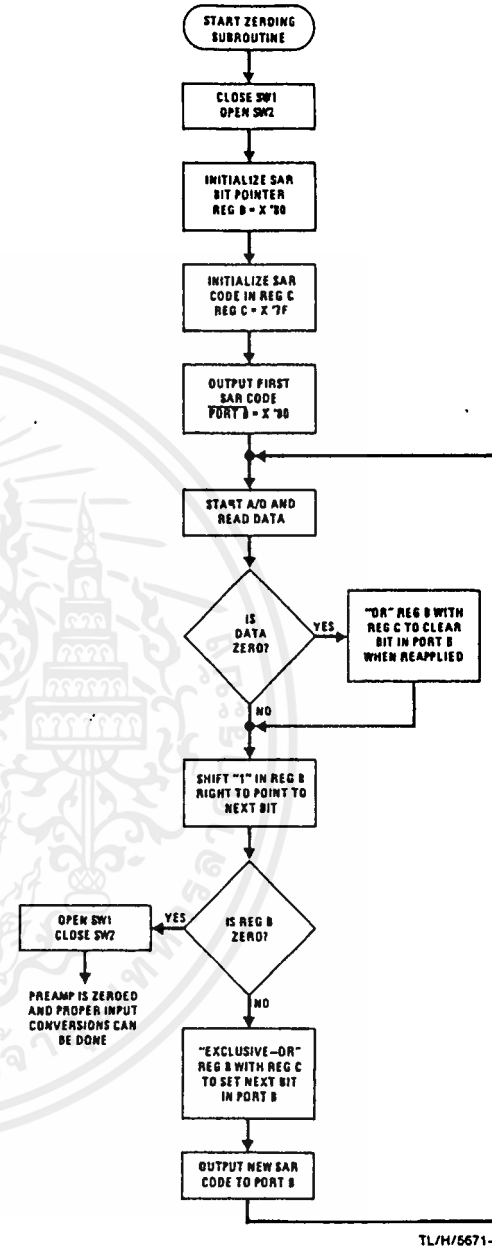
PPI control word port is at port address E7

Program Counter automatically goes to ADDR:3C3D upon acknowledgement of an interrupt from the ADC0801

5.3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode

In data acquisition systems where more than one A/D converter (or other peripheral device) will be interrupting program execution of a microprocessor, there is obviously a need for the CPU to determine which device requires servicing. *Figure 21* and the accompanying software is a method of determining which of 7 ADC0801 converters has completed a conversion ($\overline{INTR}$ asserted) and is requesting an interrupt. This circuit allows starting the A/D converters in any sequence, but will input and store valid data from the converters with a priority sequence of A/D 1 being read first, A/D 2 second, etc., through A/D 7 which would have the lowest priority for data being read. Only the converters whose INT is asserted will be read.

The key to decoding circuitry is the DM74LS373, 8-bit D type flip-flop. When the Z-80 acknowledges the interrupt, the program is vectored to a data input Z-80 subroutine. This subroutine will read a peripheral status word from the DM74LS373 which contains the logic state of the $\overline{INTR}$ outputs of all the converters. Each converter which initiates an interrupt will place a logic "0" in a unique bit position in the status word and the subroutine will determine the identity of the converter and execute a data read. An identifier word (which indicates which A/D the data came from) is stored in the next sequential memory location above the location of the data so the program can keep track of the identity of the data entered.



TL/H/5571–28

FIGURE 19. Flow Chart for Auto-Zero Routine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
3D00   3E90     MVI 90
3D02   D3E7     Out Control Port                    ; Program PPI
3D04   2601     MVI H 01          Auto-Zero Subroutine
3D06   7C       MOV A,H
3D07   D3E6     OUT C                              ; Close SW1 open SW2
3D09   0680     MVI B 80                           ; Initialize SAR bit pointer
3D0B   3E7F     MVI A 7F                           ; Initialize SAR code
3D0D   4F       MOV C,A           Return
3D0E   D3E5     OUT B                              ; Port B = SAR code
3D10   31AA3D   LXI SP 3DAA       Start            ; Dimension stack pointer
3D13   D3E4     OUT A                              ; Start A/D
3D15   FB       IE
3D16   00       NOP               Loop             ; Loop until INT asserted
3D17   C3163D   JMP Loop
3D1A   7A       MOV A,D           Auto-Zero
3D1B   C600     ADI 00
3D1D   CA2D3D   JZ Set C                           ; Test A/D output data for zero
3D20   78       MOV A,B           Shift B
3D21   F600     ORI 00                             ; Clear carry
3D23   1F       RAR                                ; Shift "1" in B right one place
3D24   FE00     CPI 00                             ; Is B zero? If yes last
3D26   CA373D   JZ Done                            ; approximation has been made
3D29   47       MOV B,A
3D2A   C3333D   JMP New C
3D2D   79       MOV A,C           Set C
3D2E   B0       ORA B                              ; Set bit in C that is in same
3D2F   4F       MOV C,A                            ; position as "1" in B
3D30   C3203D   JMP Shift B
3D33   A9       XRA C             New C            ; Clear bit in C that is in
3D34   C30D3D   JMP Return                         ; same position as "1" in B
3D37   47       MOV B,A           Done             ; then output new SAR code.
3D38   7C       MOV A,H                            ; Open SW1, close SW2 then
3D39   EE03     XRI 03                             ; proceed with program. Preamp
3D3B   D3E6     OUT C                              ; is now zeroed.
3D3D            •                 Normal
                •
                •
                Program for processing
                proper data values
3C3D   DBE4     IN A              Read A/D Subroutine  ; Read A/D data
3C3F   EEFF     XRI FF                             ; Invert data
3C41   57       MOV D,A
3C42   78       MOV A,B                            ; Is B Reg = 0? If not stay
3C43   E6FF     ANI FF                             ; in auto zero subroutine
3C45   C21A3D   JNZ Auto-Zero
3C48   C33D3D   JMP Normal
```

Note: All numerical values are hexadecimal representations.

FIGURE 20. Software for Auto-Zeroed Differential A/D

5.3 Multiple A/D Converters In a Z-80® Interrupt Driven Mode (Continued)

The following notes apply:

1) It is assumed that the CPU automatically performs a RST 7 instruction when a valid interrupt is acknowledged (CPU is in interrupt mode 1). Hence, the subroutine starting address of X0038.

2) The address bus from the Z-80 and the data bus to the Z-80 are assumed to be inverted by bus drivers.

3) A/D data and identifying words will be stored in sequential memory locations starting at the arbitrarily chosen address X 3E00.

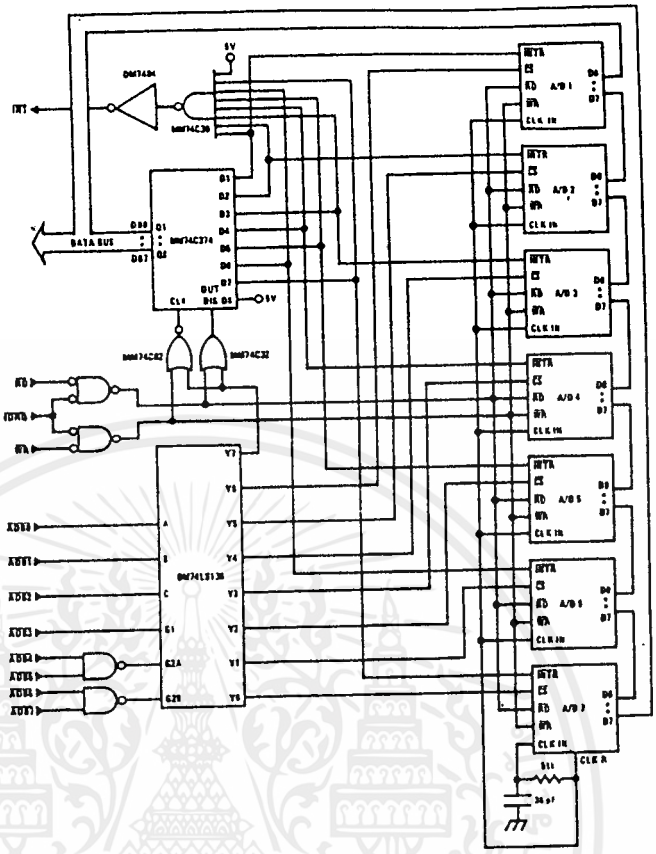
4) The stack pointer must be dimensioned in the main program as the RST 7 instruction automatically pushes the PC onto the stack and the subroutine uses an additional 6 stack addresses.

5) The peripherals of concern are mapped into I/O space with the following port assignments:

HEX PORT ADDRESS	PERIPHERAL
00	MM74C374 8-bit flip-flop
01	A/D 1
02	A/D 2
03	A/D 3
04	A/D 4
05	A/D 5
06	A/D 6
07	A/D 7

This port address also serves as the A/D identifying word in the program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TL/H/5671-29

FIGURE 21. Multiple A/Ds with Z-80 Type Microprocessor

INTERRUPT SERVICING SUBROUTINE

LOC	OBJ CODE	SOURCE STATEMENT		COMMENT
0038	E5		PUSH HL	; Save contents of all registers affected by
0039	C5		PUSH BC	; this subroutine.
003A	F5		PUSH AF	; Assumed INT mode 1 earlier set.
003B	21 00 3E		LD (HL), X3E00	; Initialize memory pointer where data will be stored.
003E	0E 01		LD C, X01	; C register will be port ADDR of A/D converters.
0040	D300		OUT X00, A	; Load peripheral status word into 8-bit latch.
0042	DB00		IN A, X00	; Load status word into accumulator.
0044	47		LD B, A	; Save the status word.
0045	79	TEST	LD A, C	; Test to see if the status of all A/D's have
0046	FE 08		CP, X08	; been checked. If so, exit subroutine
0048	CA 60 00		JPZ, DONE	
004B	78		LD A, B	; Test a single bit in status word by looking for
004C	1F		RRA	; a "1" to be rotated into the CARRY (an $\overline{INT}$
004D	47		LD B, A	; is loaded as a "1"). If CARRY is set then load
004E	DA 5500		JPC, LOAD	; contents of A/D at port ADDR in C register.
0051	0C	NEXT	INC C	; If CARRY is not set, increment C register to point
0052	C3 4500		JP, TEST	; to next A/D, then test next bit in status word.
0055	ED 78	LOAD	IN A, (C)	; Read data from interrupting A/D and invert
0057	EE FF		XOR FF	; the data.
0059	77		LD (HL), A	; Store the data
005A	2C		INC L	
005B	71		LD (HL), C	; Store A/D identifier (A/D port ADDR).
005C	2C		INC L	
005D	C3 51 00		JP, NEXT	; Test next bit in status word.
0060	F1	DONE	POP AF	; Re-establish all registers as they were
0061	C1		POP BC	; before the interrupt.
0062	E1		POP HL	
0063	C9		RET	; Return to original program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

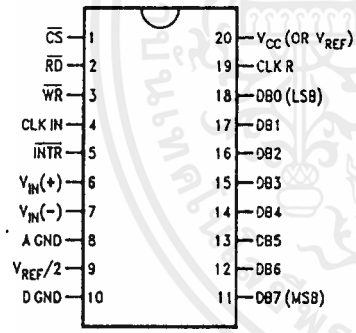
Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	0°C TO 70°C	−40°C TO +85°C
ERROR	±¼ Bit Adjusted				ADC0801LCN
	±½ Bit Unadjusted	ADC0802LCWM	ADC0802LCV		ADC0802LCN
	±½ Bit Adjusted	ADC0803LCWM	ADC0803LCV		ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCV	ADC0804LCN	ADC0805LCN
PACKAGE OUTLINE		M20B—Small Outline	V20A—Chip Carrier	N20A—Molded DIP	

TEMP RANGE		−40°C TO +85°C	−55°C TO +125°C
ERROR	±¼ Bit Adjusted	ADC0801LCJ	ADC0801LJ
	±½ Bit Unadjusted	ADC0802LCJ	ADC0802LJ
	±½ Bit Adjusted	ADC0803LCJ	
	±1Bit Unadjusted	ADC0804LCJ	
PACKAGE OUTLINE		J20A—Cavity DIP	J20A—Cavity DIP

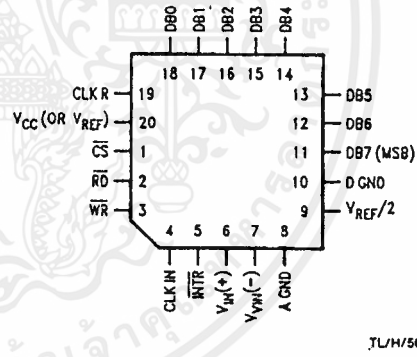
Connection Diagrams

ADC080X
Dual-In-Line and Small Outline (SO) Packages



TL/H/5671-30

ADC080X
Molded Chip Carrier (PCC) Package



TL/H/5671-32

See Ordering Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ อ. ดลชัย สุขเจริญผล เป็นอย่างสูงที่ได้ให้คำแนะนำปรึกษาในเรื่องต่าง ๆแก่ คณะผู้จัดทำ และขอขอบคุณ รุ่นพี่ ที่ได้ให้คำแนะนำปรึกษาทางด้าน SOLFWARE และขอขอบคุณ เพื่อนๆ ทุกคนที่ได้ช่วยเหลือ และให้กำลังใจในการทำปริญญานิพนธ์นี้ จนสำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ยืน ภู่วรวรรณ, " เทคโนโลยี ฮาร์แวร์ IBM PC " ,บริษัท ซีเอ็ด ยูเคชั่น จำกัด, 2533
2. กฤษดา วิศวชีรานนท์, " เรียน/เล่น/ใช้ ไอซีดิจิตอล " ,บริษัท ซีเอ็ด ยูเคชั่น จำกัด, 2532
3. เกรียงไกร จันทรา, " เครื่องบันทึกข้อมูลอัตโนมัติ " ,วารสารเซมิคอนดัคเตอร์ อิเล็กทรอนิคส์ ฉบับที่ 96 เดือนตุลาคม-พฤษจิกายน, 2532
4. เอกชัย สันกำแพง, " วงจรแปลง RS-232 เป็น RS-422 " ,วารสารเซมิคอนดัคเตอร์ อิเล็กทรอนิกส์ ฉบับที่ 115 เดือนมีนาคม-เมษายน, 2535
5. วิวัฒน์ ศิวะบวร, พรชัย จักรธำรงค์, จิรศักดิ์ ชัยวิริยะกุล, " การประยุกต์ใช้งานภาษาซี " ,บริษัท ซีเอ็ด ยูเคชั่น จำกัด, 2535
6. ดร.วิทยา วัชระวิทยากุล, " ภาษาและการโปรแกรมซี " ,บริษัท ซีเอ็ด ยูเคชั่น จำกัด, 2535
7. Ben Ezzell, " Graphics Programming in Turbo C 2.0 " ,บริษัท ซีเอ็ด ยูเคชั่น จำกัด, 1988
8. LEON A. WORTMAN,THOMAS O. SIDEBOTTOM, " THE C PROGRAMMING TUTOR " ,1987

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้