ปีการศึกษา 2532

ระบบการเก็บข้อมูลส่วนหน้า
(Front End Data Acquisition)

โดย

นายบรีชา	สง่าเจริญกิจ	29.1118
นายพิชิต	เลิศศักดิ์วรกุล	29.1146
นางสาวสุภาภรณ์	บรีชากิจอนันต์	29.1251

อาจารย์ที่ปรึกษา

อาจารย์ชนิษฐา แซ่ตั้ง
อาจารย์วัชระ ฉัตรวิริยะ
อาจารย์วันชัย ร้วรุจา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ปีการศึกษา 2532

ภาควิชา อิเลคทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการเก็บข้อมูลส่วนหน้า(front end data acquisition)

ผู้จัดทำ

1)นายบรีชา	สงวาเจริญกิจ	291118
2)นายพิชิต	เลิศศักดิ์วรกุล	291146
3)น.ส.สุภาภรณ์	บรีชากิจอนันต์	291251

.....อาจารย์ที่ปรึกษา
(อาจารย์ ชนิษฐา แช่ตั้ง)

.....อาจารย์ที่ปรึกษา
(อาจารย์ วัชระ นัครวิริยะ)

.....อาจารย์ที่ปรึกษา
(อาจารย์ วันชัย ริ้วรุจา)

026939

22 พ.ย. 2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการเก็บข้อมูลส่วนหน้า

(FRONT END DATA ACQUISITION)

นายปรีชา สงาเจริญกิจ 29.1118

นายพิชิต เลิศศักดิ์วรกุล 29.1146

นางสาวสุภาภรณ์ บริชากิจอนันต์ 29.1251

อาจารย์ที่ปรึกษา อาจารย์ชนิษฐา แซ่ตั้ง

อาจารย์วันชัย ริ้วรุจา

อาจารย์วัชระ ฉัตรวิริยะ

บทคัดย่อ

ในปัจจุบันอุตสาหกรรมต่างๆต้องอาศัยเทคโนโลยีเป็นส่วนสำคัญ ซึ่งระบบการเก็บข้อมูล (DATA ACQUISITION) นั้นก็เป็นเทคโนโลยีอย่างหนึ่งที่มีความสำคัญในการพัฒนาอุตสาหกรรม

ระบบการเก็บข้อมูลส่วนหน้า (FRONT END DATA ACQUISTION) นั้นก็คือระบบการเก็บข้อมูลซึ่งควบคุมการทำงานด้วยระบบไมโครโปรเซสเซอร์ (MICROPROCESSOR) ซึ่งเป็นข้อได้เปรียบกว่าระบบการเก็บข้อมูล (DATA ACQUISITION) ทั่วไป คือสามารถประมวลผลข้อมูลบนตัวฟอร์นเอน (FRONT END) ได้โดยไม่ต้องอาศัยเครื่องคอมพิวเตอร์เข้ามาช่วย แต่เครื่องคอมพิวเตอร์ก็สามารถเข้ามามีบทบาทในระบบได้ด้วย คือช่วยในการกำหนดการทำงานและแสดงผลบนเครื่องคอมพิวเตอร์ได้ ทำให้สะดวกต่อการวิเคราะห์ข้อมูลต่างๆ นอกจากนั้นข้อมูลที่ผ่านการประมวลผลแล้วนั้นยังสามารถส่งออกไปควบคุมขบวนการภายนอกได้อีกด้วย

ระบบการเก็บข้อมูลส่วนหน้า (FRONT END DATA ACQUISITION) นี้จึงเหมาะกับขบวนการทางอุตสาหกรรมทุกชนิดที่ต้องการควบคุมข้อมูลและการทำงานของระบบให้เป็นไปได้อย่างสมบูรณ์

FRONT END DATA ACQUISITION

Mr.PREECHA SANGACHAROENKIT 29.1118
Mr.PICHIT LERTSAKWORAKUL 29.1146
MISS SUPAPORN PREECHAKITANANT 29.1251
ADVISOR: MISS KANITTA SEATUNG
Mr.WANCHAI REURUCHA
Mr.WATCHARA CHATVIRIYA
1989

abstract

Now; technology is very important for many industries. Data acquisition system is a technology that is important for developing industry.

Front end data acquisition system is a acquisition system that is controlled by micriprocessor system. And that is an adventage of this system ; it can process data by itself and not deoend on the helf of computer system. But it can link with computer by use it for help user to fix the function of the Front End System and display data make easily analyze data. Morever, data that is processed can supply outputs to control the physical processes.

So, Front end data acquisition system is the good technology for the perfect industry.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	หัวข้อ	หน้า
บทที่ 1	คำนำ	1
บทที่ 2	ทฤษฎี	2
บทที่ 3	ส่วนประกอบของระบบ	33
บทที่ 4	การทดสอบระบบ	72
บทที่ 5	บทวิจารณ์และสรุป	73
	กิตติกรรมประกาศ	74
	บรรณานุกรม	75
	ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่1
บทนำ

ในปัจจุบันการเก็บข้อมูลมีความสำคัญมากยิ่งขึ้นเนื่องจากการทดลองหรือการวิเคราะห์ข้อมูลที่มีจำนวนมากต้องการวประมวลผลที่มีความเร็วสูงและมีประสิทธิภาพมากยิ่งขึ้น รวมถึงการควบคุมคุณภาพในงานอุตสาหกรรมต้องการเก็บข้อมูลจากขั้นตอนต่างๆของการผลิต แล้วนำผลที่ได้จากการเก็บข้อมูลนี้ไปทำการควบคุมการผลิต เช่น การควบคุมอุณหภูมิในเตาเผา เป็นต้น ซึ่งจะเป็นการลดต้นทุนในการผลิตรวมถึงจะทำให้คุณภาพของสินค้ามีความแน่นอนมากยิ่งขึ้น ซึ่งการที่จะนำเอาข้อมูลมาทำการควบคุมได้นั้นจะต้องมีการแปลงข้อมูลที่รับเข้ามาให้อยู่ในรูปของข้อมูลที่สามารถประมวลผลได้ และเมื่อทำการประมวลผลแล้วก็จะทำการแปลงข้อมูลที่ได้นี้ให้ไปควบคุมระบบต่อไป

จากความจำเป็นดังกล่าวทำให้มีการนำคอมพิวเตอร์เข้ามาช่วยในการทำงาน เนื่องจากมีประสิทธิภาพและความแน่นอนในการทำงานสูงรวมถึงมีความรวดเร็วสูง จึงมีการคิดค้นระบบของ front end data acquisition (ระบบการเก็บข้อมูลส่วนหน้า) ซึ่งเป็นระบบที่จะทำหน้าที่ในการเก็บข้อมูลได้ครั้งละหลายช่องสัญญาณ แล้วนำเอาข้อมูลที่ได้นี้ไปแปลงเพื่อนำเข้าสู่ระบบของการประมวลผลแล้วนำผลที่ได้ไปควบคุมการทำงานของระบบ รวมถึงยังมีการส่งข้อมูลที่อ่านเข้ามาและทำการประมวลผลแล้วไปแสดงผลให้กับผู้ใช้ด้วย

ซึ่งจากประโยชน์ที่ได้กล่าวถึงข้างต้นผู้จัดทำหวังว่าโครงงานนี้คงจะสามารถสร้างประโยชน์ในการนำเอาเทคโนโลยีมาดัดแปลงและพัฒนาการประมวลผลข้อมูลได้บ้างไม่มากก็น้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎี

-ระบบทั่วไปของระบบการเก็บข้อมูลส่วนหน้า

-ตัวตรวจจับสัญญาณ(sensor or transducer)

-การแปลงสัญญาณอนาลอกไปเป็นดิจิตอล

-การแปลงสัญญาณดิจิตอลไปเป็นอนาลอก

-การรับส่งข้อมูลแบบอนุกรม

-รายละเอียดของ cpu 8031

1)register ทั่วไปของระบบ

2)การติดต่อของ serial ของตัว cpu

3)timing ในการทำงานของ cpu

-การควบคุมแบบ pid

1)การควบคุมแบบอุดมคติ(ideal pid control)

2)การควบคุม pid ในทางปฏิบัติ

3)การปรับแต่งและการเลือกช่วงของการสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ระบบดาต้าแอคควิซิชั่น (Data Acquisition)

ดาต้าแอคควิซิชั่นคือ ระบบที่ใช้ในการเก็บรวบรวมรายละเอียดของข้อมูลต่างๆที่ต้องการจากขบวนการภายนอก ซึ่งโดยปกติแล้วข้อมูลเหล่านี้จะแปรตามเวลาด้วย ดังนั้นในการออกแบบให้ระบบสามารถทำงานได้รวดเร็วเพียงใดขึ้นอยู่ความเร็วในการเปลี่ยนแปลงของข้อมูลและความละเอียดที่ต้องการ

เนื่องจากขบวนการต่างๆภายนอกนั้นข้อมูลจะเป็นสัญญาณอนาลอกเป็นส่วนใหญ่ ดังนั้นข้อมูลเหล่านี้จึงต้องถูกแปลงให้เป็นสัญญาณดิจิตอลเสียก่อนแล้วจึงถูกเก็บรวบรวม (หรืออาจจะนำไปประมวลผลก่อนก็ได้) และเนื่องจากข้อมูลเป็นแบบดิจิตอลจึงทำให้ระบบนี้สามารถต่อร่วมการทำงานกับระบบคอมพิวเตอร์ได้ ซึ่งเราอาจจะนำระบบคอมพิวเตอร์มาช่วยในการแสดงผลหรือวิเคราะห์ข้อมูลต่างๆได้

ลักษณะของข้อมูลที่ใช้กับระบบดาต้าแอคควิซิชั่นมีหลายประเภทได้แก่

1) แบ่งตามเวลา
   - ข้อมูลระยะสั้น
   - ข้อมูลระยะกลาง
   - ข้อมูลระยะยาว
2) แบ่งตามลักษณะของสัญญาณ
   - อนาลอก(analog)
   - ดิจิตอล (digital)
3) แบ่งตามลักษณะอื่นๆ

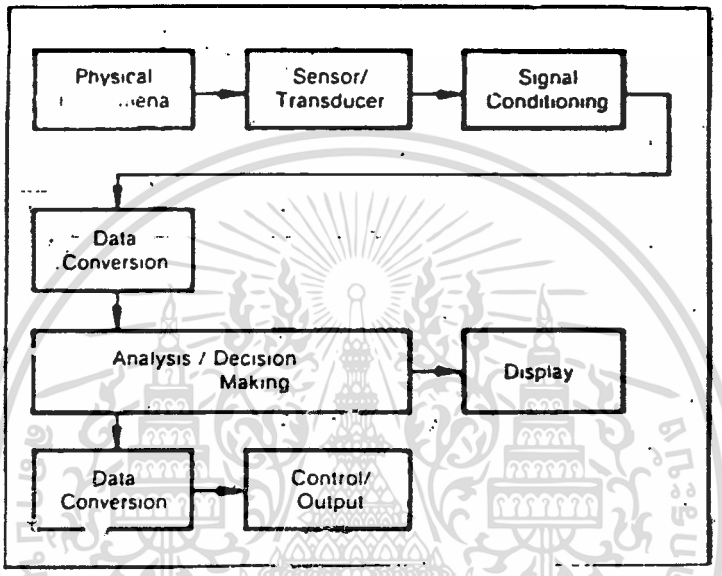
สำหรับขบวนการทั่วไปของระบบดาต้าแอคควิซิชั่น ได้แก่

1) การเก็บรวบรวมข้อมูล โดยมากจะเป็นการแปลงสัญญาณจากอนาลอกเป็นดิจิตอล
2) การประมวลผลข้อมูล ซึ่งอาจจะไม่มีก็ได้
3) การส่งสัญญาณไปยังอีกแหล่งหนึ่ง ซึ่งอาจจะเป็นระบบคอมพิวเตอร์หรือระบบหน่วยความจำใดๆ
4) การแสดงผล
5) การส่งสัญญาณกลับออกไปยังระบบภายนอกซึ่งก็คือการแปลงสัญญาณกลับจากดิจิตอลกลับไปเป็นอนาลอก

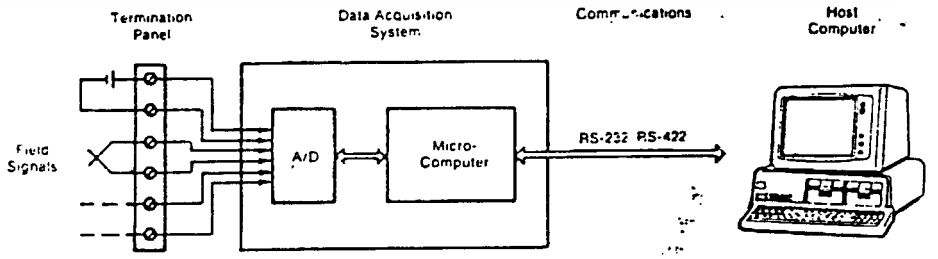
พรอนท์เอนดาต้าแอคควิซิชั่น (Data Acquisition)

ก็คือระบบดาต้าแอคควิซิชั่นอย่างหนึ่งเพียงแต่ไม่มีหน่วยแสดงผลในตัวมันเอง แต่สามารถทำงานได้โดยตัวของมันเอง แต่ก็สามารถทำงานร่วมกับระบบคอมพิวเตอร์ได้โดยใช้เป็นหน่วยแสดงผล ดังนั้นพรอนท์เอนจึงเสมือนเป็นตัวกลางเชื่อมระหว่างระบบคอมพิวเตอร์กับระบบภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ระบบ DATA ACQUISITION ทั่วไป



การต่อ DATA ACQUISITION เข้ากับระบบคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

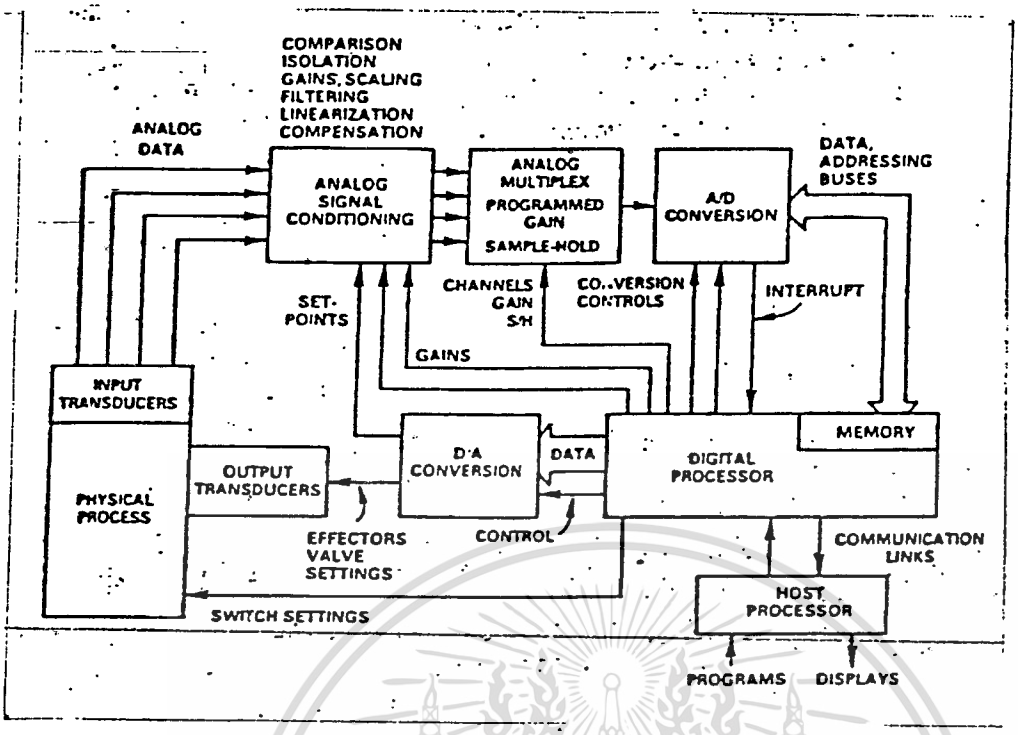
การทำงานโดยทั่วไปของระบบ front end data acquisition เริ่มจากจะมีการส่งผ่านคำสั่งต่างๆจาก host รวมถึงอุณหภูมิที่ต้องการควบคุม และ channel ที่ต้องการทำการควบคุมในระบบเมื่อเริ่มงานของระบบก็จะมีการส่ง channel ที่ต้องการทำการควบคุมให้กับ i/p port(i/p port ในที่นี้หมายถึงตัว transducer)เพื่อให้ i/p port ทำการเก็บข้อมูลเข้ามาที่ตัว front end โดยตัว transducer จะเก็บข้อมูลเข้ามาในรูปของ analog signal ต่อจากนั้นจะผ่านสัญญาณไปที่ a/d(analog to digital converter) ซึ่งจะทำการเปลี่ยนสัญญาณที่รับเข้ามาให้เป็นสัญญาณ digital ขนาด 8bit เพื่อส่งเข้าระบบ front end data acquisition เพื่อนำข้อมูลที่รับมานั้นมาทำการประมวลผลแล้วส่งไปควบคุมระบบต่อไป process ที่จะเกิดที่ตัว front end จะมีทั้งหมด 2 ส่วนคือ

1)ส่วนของ PID CONTROL

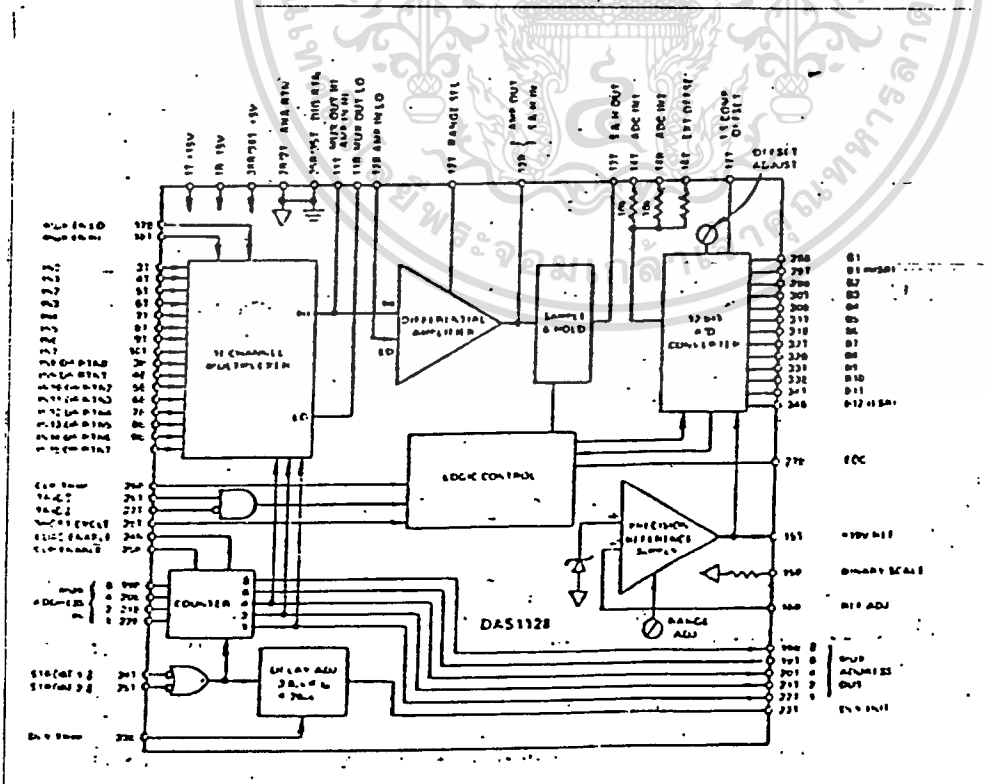
2)ส่วนที่ทำการเปรียบเทียบข้อมูลที่รับจาก i/p port กับค่า max,min, set point temperature ที่ส่งมาจาก host

เมื่อทำการ process ทั้ง2ขั้นแล้วพบข้อผิดพลาดจะมีการส่งสัญญาณที่แสดงข้อผิดพลาดไปที่ตัว front end เพื่อแสดงผลที่ failed port รวมทั้งแสดงผลที่ส่วนของ host ด้วยรวมทั้งยังมีการส่งผลลัพธ์ของ PID PROCESS ไปทำการควบคุมการทำงานของ o/p port เนื่องจาก o/p port ที่ติดต่อด้วยมีการทำงานในลักษณะของ analog signal จึงต้องมีการส่งสัญญาณเข้าไปที่ d/a(digital to analog convertor)เพื่อทำการแปลงสัญญาณตามที่ o/p port ต้องการ รวมถึงต้องมีการส่งข้อมูลทั้งที่มีการ process และยังไม่มีการprocess ไปทำการแสดงผลที่ host อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

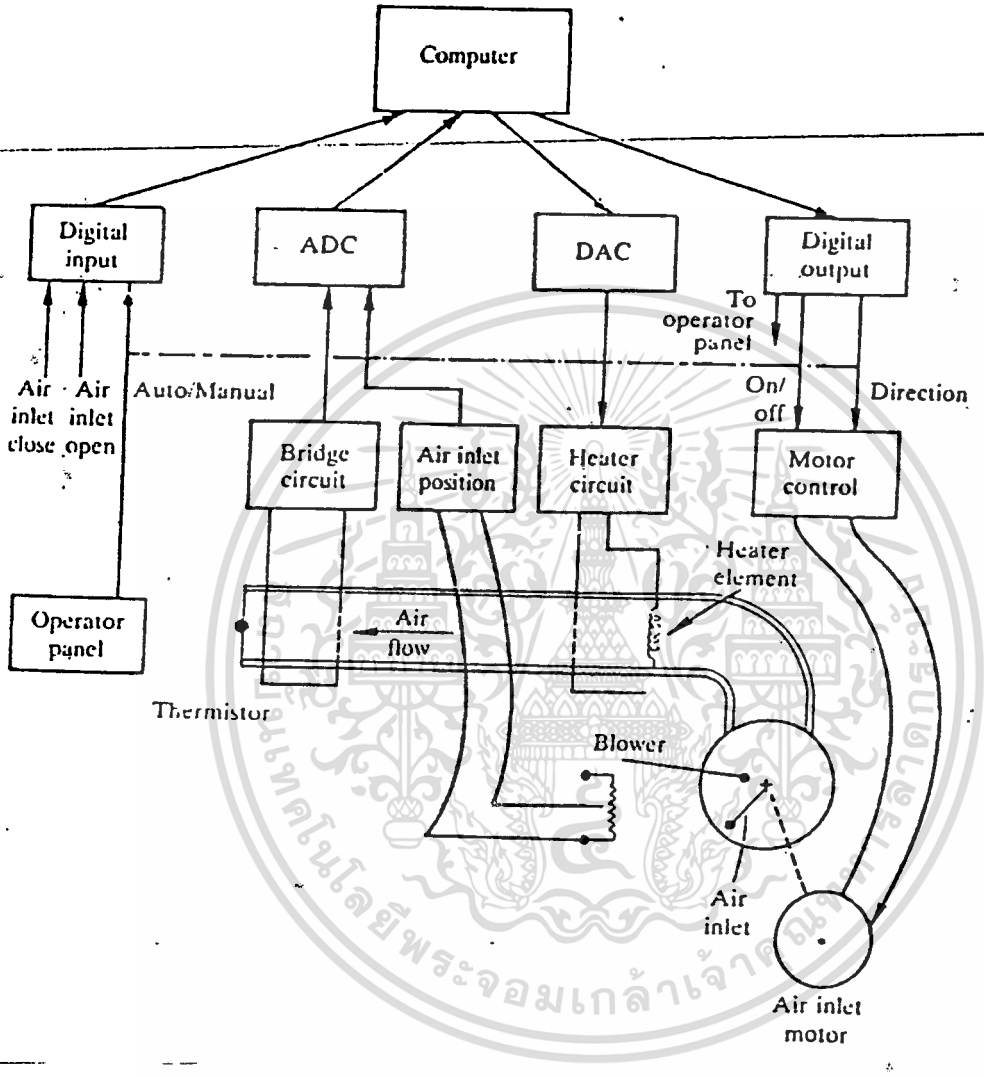


BLOCK DIAGRAM ของ FRONT END



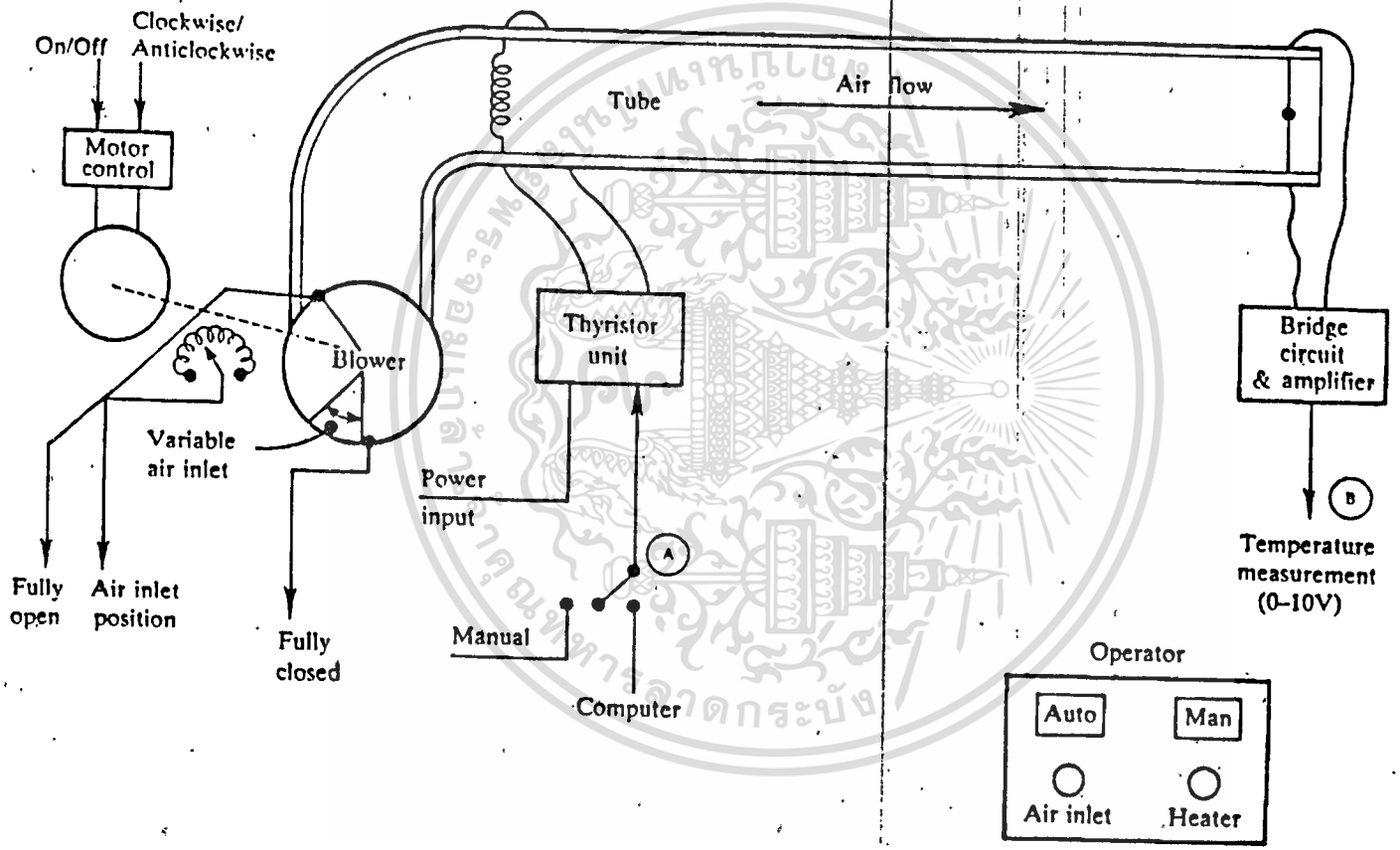
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง IC FRONT END DATA ACQUISITION



ตัวอย่าง ADC และ FRONT END DATA ACQUISITION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



A simple plant: a hot-air blower.

ตัวตรวจจับสัญญาณ(sensor or transducer)

เป็นตัวที่ใช้ในการอ่านค่าทางกายภาพ มาเป็นสัญญาณที่ใช้เปรียบเทียบได้ เช่น แรงดันไฟฟ้า เพื่อนำมาใช้ในการประมวลผลได้ ซึ่งการวัดทางกายภาพมีหลายลักษณะ เช่น วัดจากระยะทาง อุณหภูมิ แสง เป็นต้น

ซึ่งการตรวจจับที่จะกล่าวถึงในที่นี้จะเป็นตัวตรวจจับแรงต่างๆซึ่งแบ่งออกเป็นส่วนๆดังนี้คือ

1)pressure(แรงดัน)ใช้หลักการคือเมื่อได้รับแรงดันจะเกิดการบิดงอและเลื่อนตำแหน่ง ซึ่งจะแปรค่า ค.ต.ทานลักษณะ bridge เมื่อมีการจ่ายค่าแรงดันเข้าไปทำให้มีการแปรค่าของแรงดัน

2)แรงสั่นสะเทือนใช้หลักการของการเปลี่ยนค่าเหนี่ยวนำของสนามแม่เหล็กทำให้เกิดค่าแรงดันไม่เท่ากัน

3)load cell(โหลดเซล)คือเมื่อมีการเปลี่ยนน้ำหนักของวัตถุจะมีผลให้ค่าความต้านทานหรือค่าความเก็บประจุเปลี่ยนไป

4)การวัดอุณหภูมิมีวิธีต่างๆดังนี้คือ

-termister เป็นสารกึ่งตัวนำแปรค่าตามความต้านทาน แปรผกผันกับอุณหภูมิ

-resister thermal detector(rtd)เป็นโลหะมี ส.ป.ส ของอุณหภูมิเป็นบวก และมีความเป็นเชิงเส้นแต่ความไวน้อยกว่า termistor

-termocouple ทำจากตัวนำ 2 ชนิด เมื่อได้รับความร้อนจะเกิดแรงดันไฟฟ้าซึ่งแปรตามอุณหภูมิที่รอยต่อ

-IC thermosensor(ไอซีวัดอุณหภูมิ)ความแม่นยำสูงแต่ช่วงใช้งานแคบมีค่าแรงดันแปรผกผันกับอุณหภูมิ

การ multiplexing signal

เนื่องจากการรับส่งข้อมูลในระบบ data acquisition มี i/p port จำนวนมากทำให้ต้องมีการรับข้อมูลโดยแบ่งเวลาหรือที่เรียกว่า multiplex เพื่อนำเอาสัญญาณไปประมวลผล โดยมีการควบคุมการนำเข้าสัญญาณจากตัว microprocessor เมื่อมีการประมวลผลแล้วจะมีการส่งสัญญาณออกไปในรูปต่างๆคือ

1)digital signal คือการส่งสัญญาณแสดงการผิดพลาดของระบบซึ่งมีรูปแบบคร่าวๆดังรูปที่ 1

2)analog signal คือการส่งข้อมูลออกไปที่ analog switch โดยมีการ multiplex และ demultiplex ซึ่งสร้างโดยอุปกรณ์ต่างๆดังนี้คือ

-การใช้ jfet มีข้อดีคือความเพี้ยนต่ำแต่ช่วงการทำงานของ tr จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงแคบกว่าแรงดันไฟเลี้ยง. คือถ้าใช้ jfet-n channel แรงดันเกทต้องมีค่าต่ำกว่าสัญญาณอย่างน้อย = แรงดันพินท์ออฟ

-การใช้ pmos ข้อดีคือราคาถูก สร้างง่ายแต่มีความเพี้ยนสูง และมีความถี่ในการ switching ต่ำ

-การใช้ cmos สร้างจาก pmosกับ nmos มีความเป็นเชิงเส้นดีขึ้น และใช้กับพวก ttl gate และ cmos ได้ดี ซึ่งพวก cmos นี้แบ่งออกเป็น 2 ประเภทคือ

1)dielectrically isolation (การแบ่งแยกด้วยฉนวน) ซึ่งมีข้อดีคือการออกแบบง่าย ลดการแผ่งของ scr ได้ ลดค่าการเก็บประจุจากรอยต่อทำให้ทำงานได้ที่ความถี่สูงได้ดีขึ้น

2)junction isolation(การแบ่งส่วนด้วยรอยต่อ)เป็นการแยกอุปกรณ์ภายในด้วยการสร้างรอยต่อแรงดันย้อนกลับ ข้อเสียคือเกิดค่าเก็บประจุที่รอยต่อและทำให้เกิดการสั่นวงจรได้

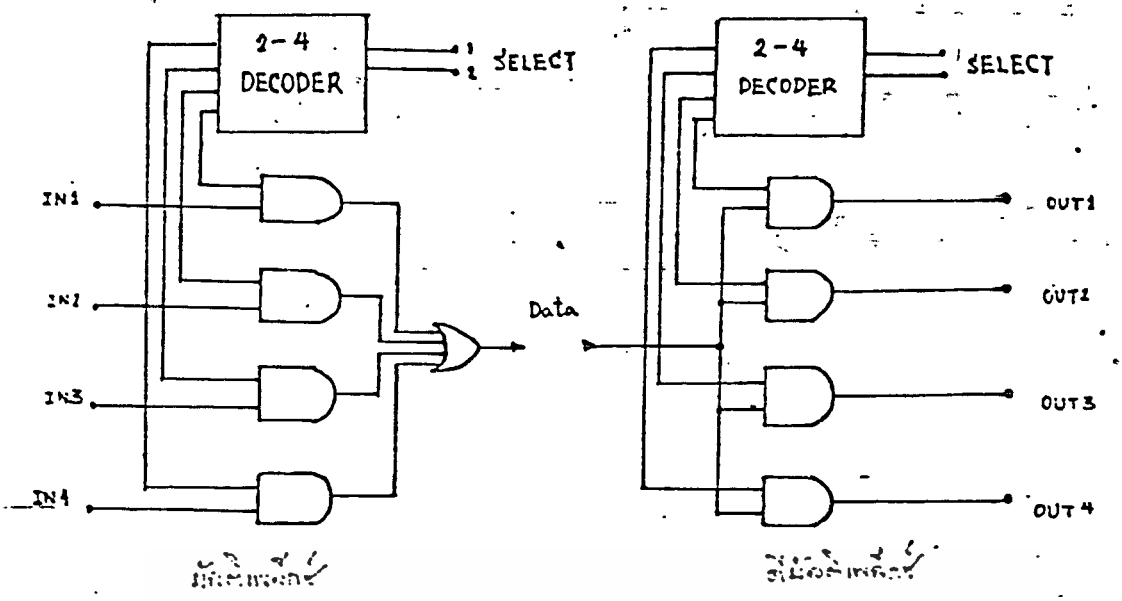
การสุ่มเก็บสัญญาณ(sample and hold)

ในการเปลี่ยนสัญญาณจาก analog ไปเป็น digital จะเกิดข้อผิดพลาดมากในกรณีที่มีการเปลี่ยนแปลงของสัญญาณเร็วมาก จึงต้องมีการ hold signal ไว้และต้องมีการสุ่มสัญญาณด้วยความถี่อย่างน้อย 2 เท่าของความถี่สูงสุดเพื่อลดการเพี้ยนของสัญญาณ วงจร sample and hold มี 2 ลักษณะคือ

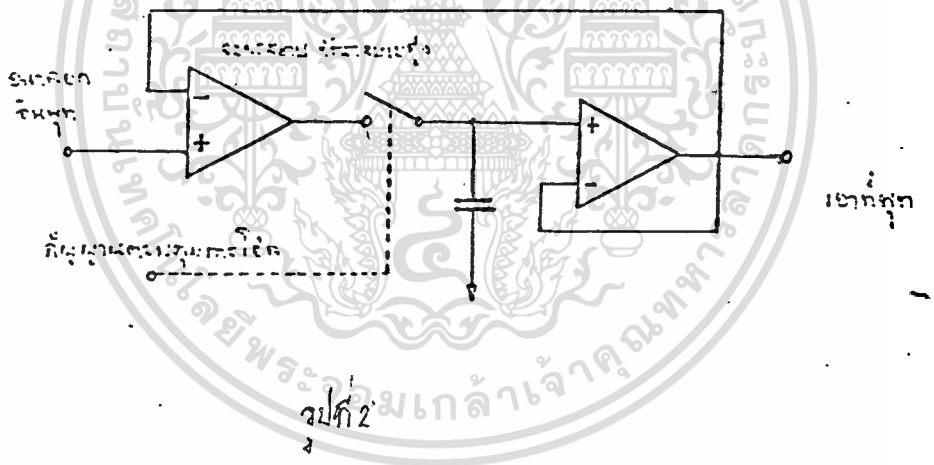
1)open loop(แบบเปิด)ข้อดีคือความเร็วสูงและถ้าต่อวงจรดังรูปที่ 2 จะทำให้การเปลี่ยนแปลงที่ตัวเก็บตัวประจุขึ้นกับค่า c.r.t.ที่ o/p ของ op amp

2)close loop(แบบปิด)ข้อดีคือความแม่นยำสูง และเปลี่ยนแปลงค่า gainได้ดังแสดงในรูปที่ 3

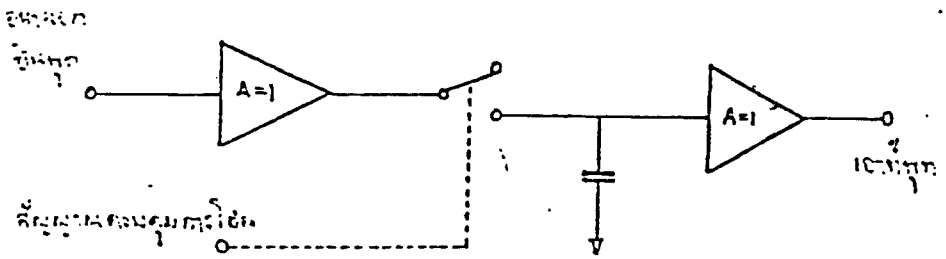
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
026939



รูปที่ 1



รูปที่ 2



รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

สัญญาณอนาลอกเป็นสัญญาณที่มีการเปลี่ยนแปลงปริมาณอย่างต่อเนื่องกับเวลาและขนาดสัญญาณ ส่วนสัญญาณดิจิตอลเป็นสัญญาณเชิงตัวเลขซึ่งเป็นกลุ่มของสัญญาณ ในหนึ่งกลุ่มสัญญาณจะประกอบไปด้วยจำนวนบิทเท่ากับ n ตัว สามารถแทนปริมาณได้ 2**n ค่า

ความสัมพันธ์ระหว่างสัญญาณดิจิตอลกับสัญญาณอนาลอกจะแสดงได้ดังสมการ

$$A = kV_{ref}\left[b_0 2^{0} + b_1 2^{-1} + b_2 2^{-2} + \ldots + b_{n-1} 2^{n-1}\right]$$

ซึ่ง b0 คือ บิทสูงสุด most significant bit

b n-1 คือ บิทต่ำสุด least significant bit

ดังนั้น วงจรแปลงสัญญาณดิจิตอลที่แปลงสัญญาณระหว่างดิจิตอลกับอนาลอกจะต้องมีคุณสมบัติสอดคล้องตามสมการ ค่าจากนิยามและความหมายต่างๆของการแปลงสัญญาณ

1)ความละเอียด(resolution) เป็นค่าที่แสดงถึงหน่วยย่อยสุดที่สามารถอ่านได้ ซึ่งจำนวนย่อย เท่ากับ 2n

2)ความแม่นยำ(accuracy) คือความถูกต้องของสัญญาณที่ได้จากการแปลง ซึ่งขึ้นอยู่กับปริมาณของความผิดพลาดต่างๆดังนี้

2.1)ออฟเซ็ท(offset) คือการเลื่อนไปของความถูกต้อง เป็นผลให้ค่าอื่นเลื่อนขึ้นหรือลงตามด้วยปริมาณเท่า

2.2)สัมประสิทธิ์ของสเกล(scale factor) เกิดจากการแปลงตามสมการความสัมพันธ์ที่มีค่าสัมประสิทธิ์ไม่ถูกต้อง กล่าวคือ จำนวนของอนาลอกต่อหนึ่งหน่วยของดิจิตอลไม่ถูกต้อง

2.3)ความเป็นเชิงเส้น(linearity) ซึ่งถ้าเกิดขึ้นแล้วจะทำให้เกิดความไม่แน่นอน

2.4)การเปลี่ยนแปลงในทิศทางเดียว(monotonicity) ทำให้เกิดการเพิ่มขึ้นของปริมาณทางแกนหนึ่งแต่อีกแกนหนึ่งมีค่าลดลง

2.5)เวลาในการเข้าสู่ภาวะคงตัว(settling time) เป็นเวลาที่ใช้ในการแปลงสัญญาณโดยเริ่มตั้งแต่จุดที่สัญญาณเริ่มถูกแปลง จนกระทั่งสัญญาณเข้าสู่สภาวะคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลนั้น มีอยู่หลายชนิด คือ

1) แบบวงจรอินทรีเกท(integrate types ADC)

- สโลปเดี่ยว
- สโลปคู่

2) แบบวงจรนับหรือเซอร์โว (counter of servo types ADC)

- แบบขั้นบันได (staircase ADC)
- แบบขั้นบันไดชนิดเลื่อนตาม (tracking ADC)
- แบบซัคเซสซีฟ แอพพรอกซิเมท(successive approximatation ADC)

3) แบบวงจรขนาด(parallel types)

- แบบแฟลช(flash ADC)

ในรายละเอียดของแต่ละเอียดแต่ละชนิดมีดังนี้

1)แบบอินทรีเกท อาศัยหลักการเปรียบเทียบแรงดันสัญญาณกับแรงดันอ้างอิง โดยนับสัญญาณพัลส์ที่กำเนิดขึ้นมาด้วยอัตราที่แน่นอนจะเริ่มต้นนับตั้งแต่แรงดันเปรียบเทียบเป็นศูนย์ จนกระทั่งแรงดันเปรียบเทียบเท่ากับแรงดันสัญญาณ และค่าที่จะเปรียบเทียบได้สูงสุดจะเท่ากับแรงดันอ้างอิง การนับมีได้ 2 วิธีคือ

1.1)single slope เมื่อเริ่มต้นการทำงาน s1 จะเปิดวงจรทำให้วงจรกำเนิดสัญญาณแรมป์ มีแรงดันเพิ่มขึ้นตามเวลาขณะเดียวกัน วงจรนับจะนับสัญญาณนาฬิกาที่เข้ามาจนกระทั่งแรงดันของแรมป์เท่ากับแรงดันอินพุท จึงหยุดการนับ ค่าที่นับได้คือค่าของดิจิตอลที่ต้องการ วงจรนี้ยังมีข้อเสียคือ ความแม่นยำการแปลงยังขึ้นอยู่กับค่า R และ C ด้วย

1.2)dual slope เมื่อเริ่มต้นการนับ s2 จะถูกเปิดออกและสวิทส์ 2 ต่ออยู่ที่ Vin ทำให้ประจุถูกเก็บไว้ที่ C1 โดยกระแสถูกกำหนดด้วยแรงดัน Vin กับค่า R เมื่อถึงเวลากำหนด S1 จะเปลี่ยนไปอยู่ที่ Vref ซึ่งมีค่าคงที่ทำให้ประจุถูกคายออกด้วยอัตราที่แน่นอนจนกระทั่งหมด

2)แบบวงจรนับหรือเซอร์โว จะประกอบด้วยวงจรเปรียบเทียบ วงจรแปลงดิจิตอลเป็นอนาลอกและวงจรนับ โดยวงจรนับเพิ่มค่าขึ้นเรื่อยๆ จนกระทั่งได้ค่าเท่ากับหรือมากกว่าแรงดันอินพุทจึงหยุดนับ การทำงานในแต่ละชนิดมีความแตกต่างกันดังนี้

2.1)แบบขั้นบันได วงจรนี้จะเริ่มต้นนับเพิ่มขึ้นในทางเดียวและจะเริ่มต้นที่ศูนย์ทุกครั้ง เวลาในการนับจึงขึ้นอยู่กับขนาดของสัญญาณอินพุทด้วย

2.2)แบบขั้นบันไดชนิดเลื่อนตาม ซึ่งจะแตกต่างจากแบบแรกคือสามารถนับได้ทั้งขึ้นและลง โดยตัดสินจากการเปรียบเทียบแรงดันก้าแรงดันลดลงต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

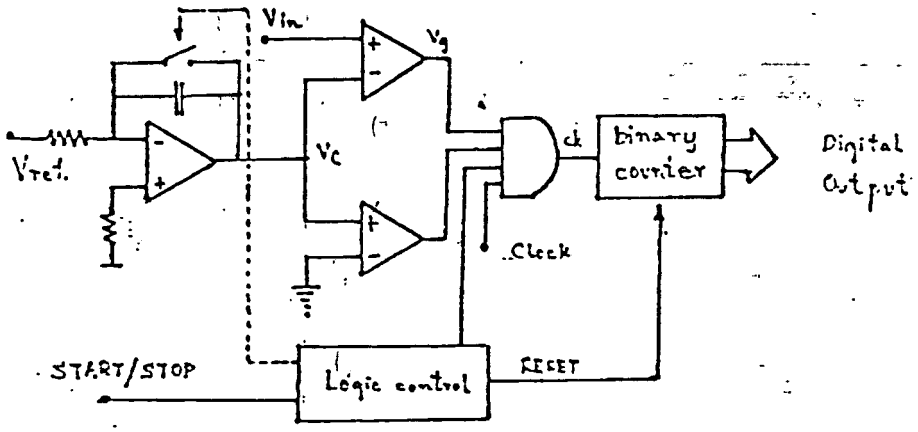
เอาท์พุทของวงจรแปลงดิจิตอลเป็นอนาลอก ก็จะทำการนับค่าลง ถ้าแรงดันสูงกว่า ก็จะนับขึ้น และจะหยุดเมื่อแรงดันมีค่าใกล้เคียงกันมากที่สุด แต่วงจรนี้ก็ยังไม่สามารถแก้ไขข้อเสียในเรื่องการแปลงสัญญาณที่ไม่แน่นอนได้ ส่วนวงจรเหมือนแบบขั้นบันได

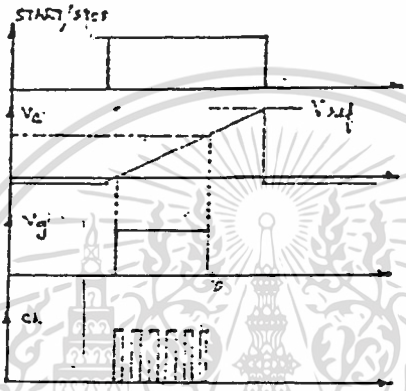
2.3)แบบซัคเซสซีฟ แอพพรอกซิเมท เป็นวงจรที่เปลี่ยนจากวงจรนับ วงจรซัคเซสซีฟ แอพพรอกซิเมท รีจิสเตอร์(successive approximate register)ซึ่งถูกกระตุ้นด้วยสัญญาณนาฬิกา โดยการแปลงสัญญาณจะเริ่มจากการแปรค่าของบิทสูงสุดมาถึงบิทต่ำสุด โดยเริ่มให้ทุกบิทมีค่าเป็นศูนย์ก่อน เมื่อบิทใดแปรค่าจากศูนย์เป็นหนึ่งแล้ว ทำให้แรงดันที่เปรียบเทียบสูงกว่าแรงดันอินพุท ก็จะแปรค่าบิทนั้นกลับเป็นศูนย์ ถ้าบิทใดแปรค่าแล้วค่าคงยังมีค่าน้อยกว่า ก็จะคงค่าที่ได้ที่แรงดันเปรียบเทียบจะใกล้เคียงกับแรงดันอินพุทการแปลงในลักษณะนี้จะใช้เวลาในการแปลงแน่นอน

3)วงจรแบบขนาน

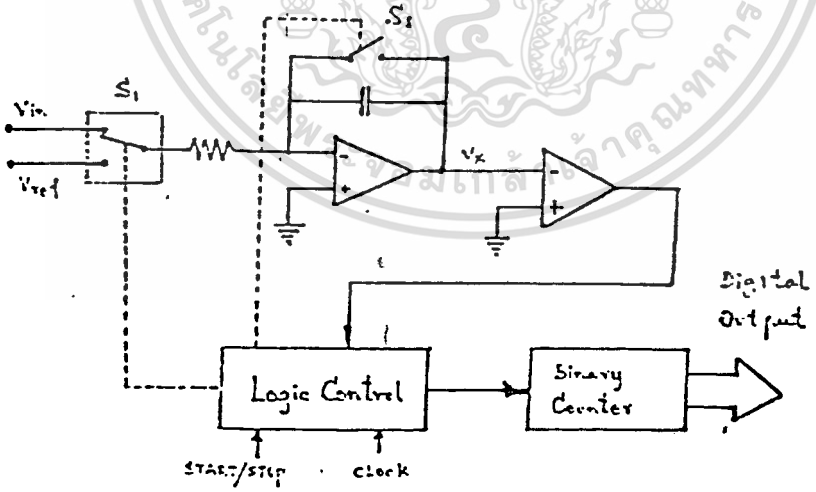
แบบแฟลช ใช้ในกรณีที่ต้องการความเร็วสูงมากๆ วงจรนี้จะต้องใช้วงจรเปรียบเทียบจำนวนมากเท่ากับ 2**n-1 ตัว เมื่อ n คือจำนวนบิทที่ต้องการ วงจรเปรียบเทียบแต่ละตัวจะถูกต่อกับแรงดันอ้างอิงที่สอดคล้องกับข้อมูลทางดิจิตอลดังนั้น จะต้องมีวงจรที่ถอดรหัสจากตัวเปรียบเทียบแต่ละตัว เป็นสัญญาณดิจิตอล วงจรนี้มีข้อยุ่งยากมากในการผลิตลงไอซี เนื่องจากใช้คอมพาราเตอร์

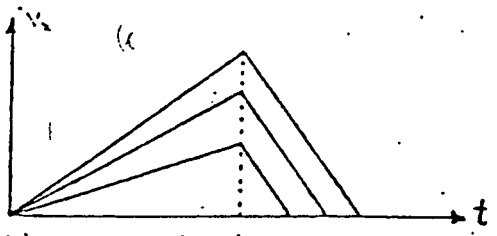
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





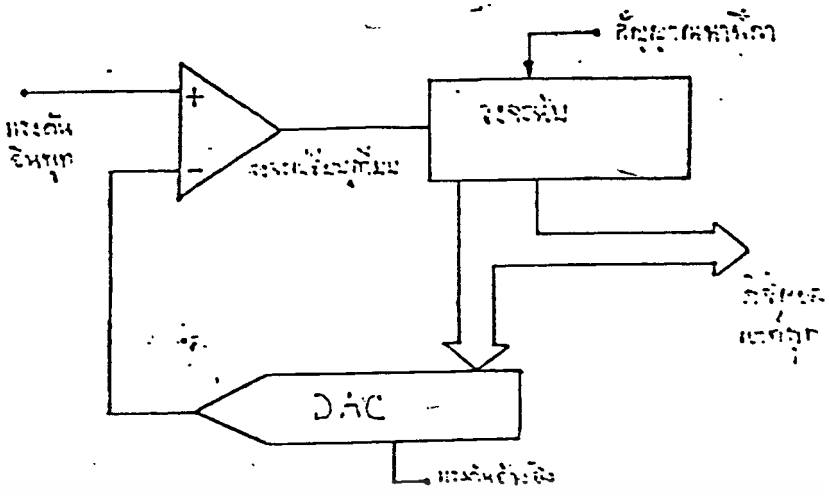
วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบอินทิเกรตชนิดสโลปเดียว



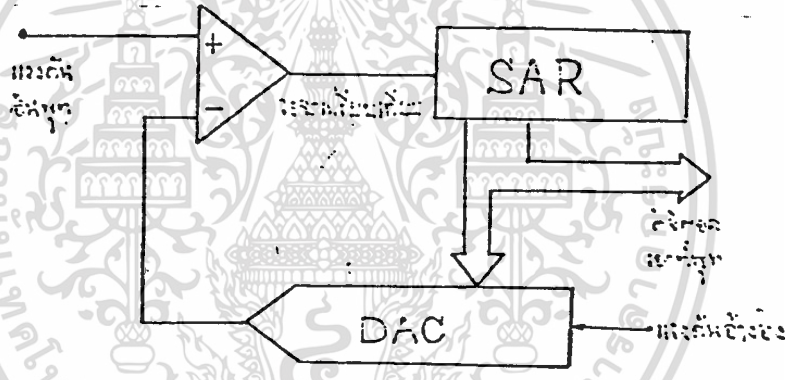


วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบอินทิเกรตชนิดสโลปคู่

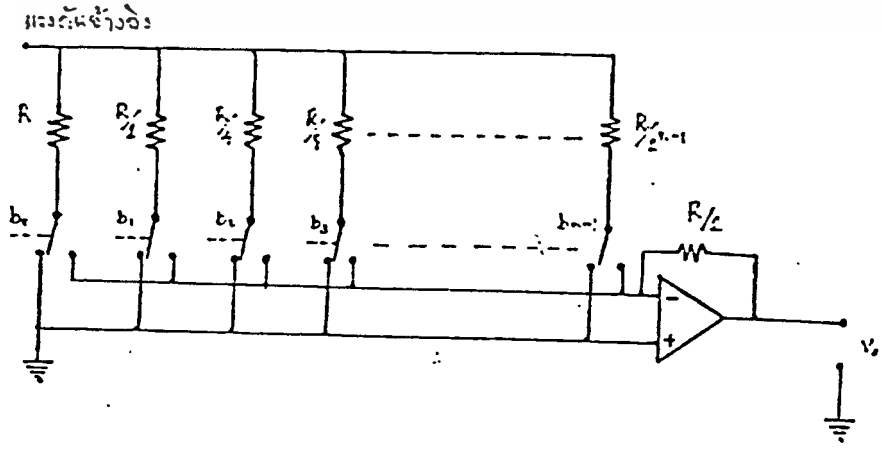
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบขั้นบันได



วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบซักเซสซีฟแอพพรอกซิเมชัน



วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบถ่วงน้ำหนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงสัญญาณดิจิตอลเป็นอนาลอก

เมื่อข้อมูลที่ถูกประมวลผลเสร็จสิ้นแล้ว จะนำเอาผลไปใช้งานจำเป็นต้องแปลงข้อมูลนั้นกลับมาเป็นอนาลอกอีก วงจรแปลงนี้มีหลายชนิด คือ

1)สวิทช์กระแส (current switching ADC)

2)สวิทช์แรงดัน (voltage switching ADC)

3)แลดเดอร์เนทเวอร์ค (ladder network)

4)สวิทช์ประจุ

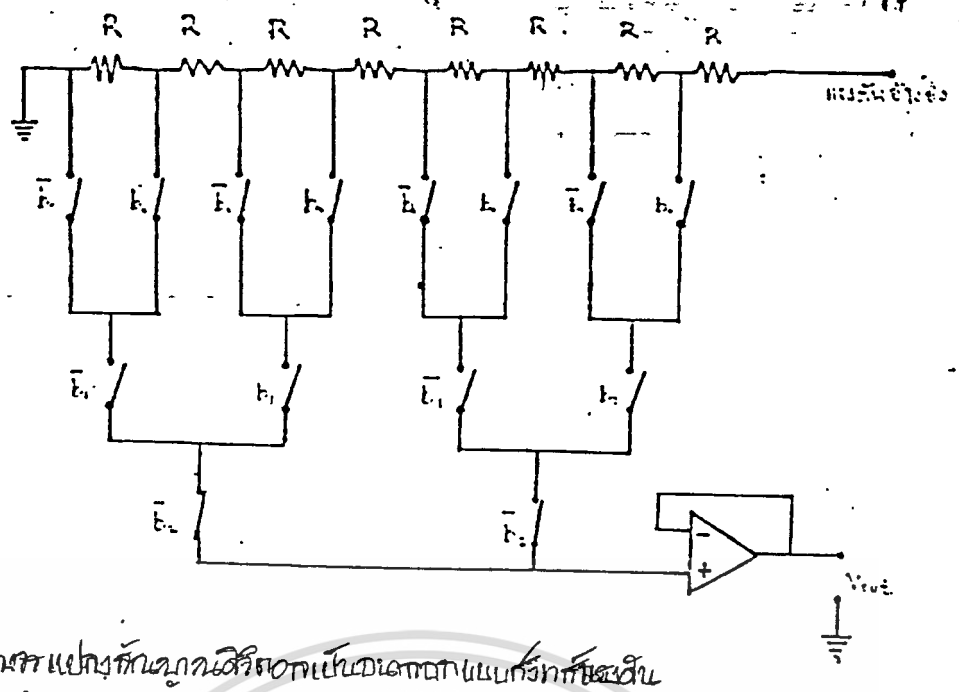
1)สวิทช์กระแส ใช้ตัวต้านทานค่าต่างๆ ปรับกระแสที่ไหลจากจุดแรงดันอ้างอิง เข้าสู่วงจรผลบวก โดยบิทที่มีค่าต่ำจะใช้ความต้านทานสูง

2)สวิทช์แรงดัน ใช้ตัวต้านทานที่มีค่าเท่ากันมาต่อกัน แบ่งแรงดันเป็นจำนวนชั้นเท่ากับ 2**n-1 ชั้น แล้วใช้วงจรเกทต่อกันเป็นสวิทช์ปิดเปิด โดยผลของการปิดเปิดสวิทช์จุดที่ต่อจากตัวต้านทานถึงวงจรขยายแบบบัฟเฟอร์เพียงจุดเดียว

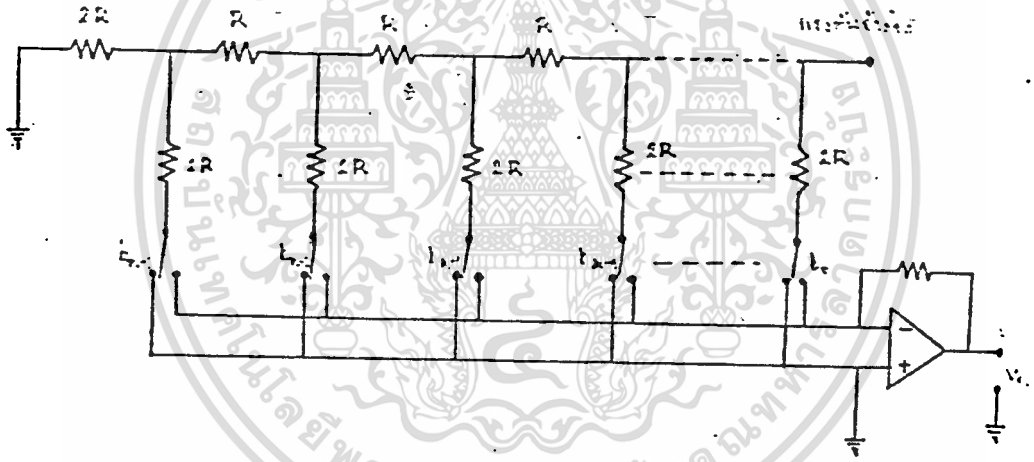
3)แลดเดอร์เนทเวร์ค เป็นลักษณะผสมระหว่างแบบสวิทช์กระแสและแรงดันโดยจะใช้ความต้านทานเพียง 2 ค่า คือ R,2R ต่อสลับกันเป็นบันได ทำให้มีการแบ่งแรงดันที่แต่ละจุด และแต่ละจุดจะมีตัวต้านทานต่อเข้าไปยังวงจรผลบวก ซึ่งจะได้ผลบวกเป็นค่าแรงดัน วงจรนี้ตัวต้านทานที่ต่อเข้ามาเข้าวงจรรวม จะต้องมีกระแสคงที่ตลอดเวลา จึงต้องต่อให้กราวด์ เมื่อบิทนั้นเป็นศูนย์ และต่อเข้าขาลบของออฟแอ็มป์ เมื่อมีค่าเป็นหนึ่ง เพราะออฟแอมป์มีค่าแรงดันเท่ากับกราวด์

4)สวิทช์ประจุ การต่อตัวเก็บประจุร่วมกับวงจรที่เหมาะสมก็สามารถสร้างวงจรแบ่งแรงดันได้ วงจรนี้เหมาะสมกับการสร้างด้วยเทคนิคลยีแบบมอส เนื่องจากสามารถสร้างตัวเก็บประจุได้ง่าย

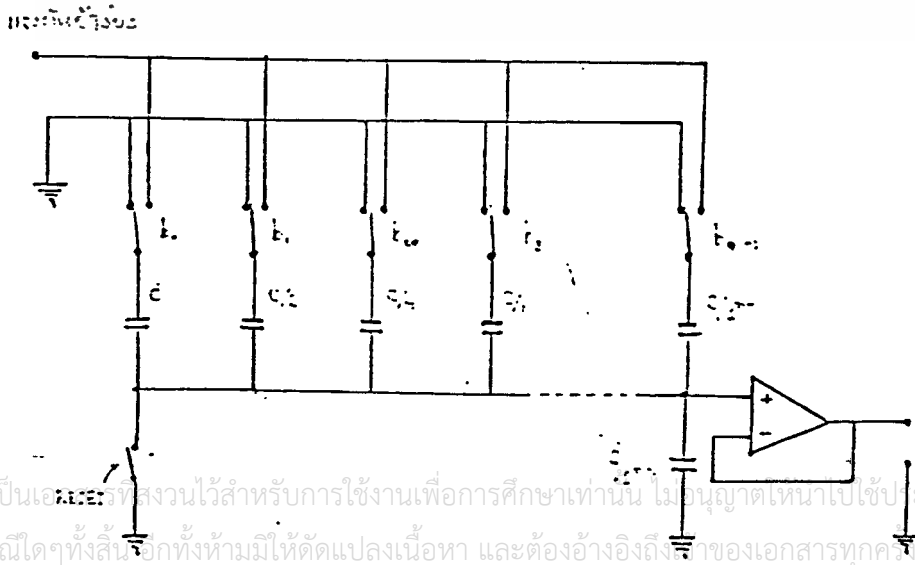
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรแปลงดิจิทัลเป็นอนาลอกแบบถ่วงน้ำหนักตัวต้านทาน



วงจรแปลงดิจิทัลเป็นอนาลอกแบบบันไดตัวต้านทานแบบ R-2R



วงจรแปลงดิจิทัลเป็นอนาลอกแบบถ่วงน้ำหนักตัวเก็บประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การรับส่งข้อมูลแบบอนุกรม</u>

การรับส่งข้อมูลแบบอนุกรมเป็นการส่งข้อมูลในลักษณะที่ข้อมูลถูกส่งเรียงไปเป็นขบวนของสัญญาณดิจิตอล ในขนาดที่เป็นไบต์หรือเป็นกลุ่มคำ ซึ่งข้อดีเปรียบของการส่งลักษณะนี้คือเป็นการส่งที่ใช้สายสัญญาณในการส่งน้อยมากและสามารถส่งได้เป็นระยะทางไกลอีกด้วย

การส่งข้อมูลแบบดิจิตอลมีการส่งได้ 2 ลักษณะคือ

1)การรับส่งแบบซิงโครนัส (synchronous)

2)การรับส่งแบบอะซิงโครนัส (asynchronous)

การส่งข้อมูลแบบอะซิงโครนัสเป็นที่นิยมมากกว่าเพราะว่างายต่อการใช้งานและมีความถูกต้องของข้อมูลสูง

<u>การรับส่งข้อมูลแบบอะซิงโครนัส</u>

ในการรับส่งข้อมูลนั้นได้มีการกำหนดมาตรฐานของรูปแบบการส่งไว้ซึ่งเรียกว่าโปรโตคอล (protocal) ซึ่งมีลักษณะดังนี้

1) บิทเริ่มต้น (start bit) เป็นบิทนำของข้อมูลที่จะส่งหรือรับ มีสถานะต่างจากสภาวะปกติ (mark state) ซึ่งมีสถานะเป็นหนึ่ง โดยบิทเริ่มต้นมีสถานะเป็นศูนย์

2) จำนวนบิทของข้อมูลที่จะใช้ในการรับส่ง ซึ่งสามารถเลือกได้คือขนาด 7 บิทและ 8 บิท ซึ่งจะใช้เป็นมาตรฐาน

3) บิทพาริตี้ (parity bit) บิทนี้จะแสดงว่าผลรวมของจำนวนบิทที่เป็นหนึ่งว่ามีเป็นจำนวนคู่หรือคี่ ซึ่งในการใช้นั้นสามารถเลือกได้ว่าจะใช้จำนวนคู่หรือคี่

4) บิทหยุด (stop bit) เป็นบิทสุดท้ายของการส่งใน 1 ไบท์ ซึ่งจะแสดงว่าการส่งสิ้นสุดในหนึ่งไบท์แล้ว มีสถานะเดียวกับ mark state

สำหรับพอร์ทที่ใช้ในการรับส่งข้อมูลแบบอนุกรมประกอบด้วยส่วนสำคัญต่างๆ เช่น

1) สัญญาณนาฬิกาของการส่ง (buad rate) ซึ่งได้กำหนดค่ามาตรฐานไว้ได้แก่ 9600,4800,2400,1200,600,300,150,110,75 buad/sec เพื่อให้พอร์ทที่ทำการรับส่งสามารถทำงานประสานกันได้อย่างถูกต้อง

2) สัญญาณควบคุมจากระบบประมวลผลส่วนกลาง(กรณีใช้ร่วมกับหน่วยประมวลผลส่วนกลาง) ซึ่งเป็นสัญญาณกำหนดการใช้งาน

3) สัญญาณควบคุมโมเด็ม พอร์ทอนุกรมสามารถใช้ร่วมกับระบบโมเด็มได้ ซึ่งจะสามารถทำการส่งมีประสิทธิภาพดีขึ้น คือ สามารถส่งได้ในระยะไกลขึ้นโดยจะมีโมเด็มเป็นหน่วยประสานงานทั้งในการรับและการส่ง (ระบบนี้อาจจะต่อร่วมกับเครือข่ายโทรศัพท์ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ส่วนเก็บข้อมูลชั่วคราวและการแปลงข้อมูล เนื่องจากในระบบคอมพิวเตอร์หรือในระบบไมโครโปรเซสเซอร์ทั่วไปจะมีการรับส่งข้อมูลแบบขนาน ดังนั้นข้อมูลที่ได้จากการรับแบบอนุกรมจะต้องทำการแปลงให้อยู่ในแบบขนานเสียก่อน โดยจะมีชิฟท์รีจิสเตอร์ทำหน้าที่เลื่อนบิทเข้ามาเก็บในหน่วยความจำชั่วคราว(buffer) ก่อนที่จะถูกส่งไปยังระบบไมโครโปรเซสเซอร์ ส่วนในการส่งข้อมูลนั้น ข้อมูลที่ได้จากระบบไมโครโปรเซสเซอร์จะต้องถูกแปลงเป็นแบบอนุกรมเสียก่อนที่จะถูกส่งออกไป

เนื่องจากสัญญาณเอาท์พุทจากไอซีต่างๆมีระดับสัญญาณเป็น 0 และ 5 โวล์ท เป็นช่วงของสัญญาณซึ่งมีขนาดต่ำ ดังนั้นสัญญาณรบกวนจึงสามารถรบกวนต่อสัญญาณได้ง่าย จึงได้มีการตกลงมาตรฐานใหม่ซึ่งในระบบ RS-232 ได้กำหนดให้ในระดับสูงของสัญญาณให้มีขนาดเป็น -5 โวล์ท ถึง -15 โวล์ท และในระดับต่ำของสัญญาณให้มีขนาดเป็น 15 โวล์ท ถึง 5 โวล์ท ซึ่งในระบบ RS-232 นี้ความต้านทานต่อสัญญาณรบกวนมากกว่าในระบบ TTL ทั่วไปจึงมีความเหมาะสมในการรับส่งสัญญาณในระยะทางไกลได้ดีกว่า

สัญญาณที่สำคัญที่ใช้ในระบบ RS-232 ได้แก่

- ขารับข้อมูล (RXD) และขาส่งข้อมูล (TXD)
- ขาสัญญาณควบคุมโมเด็มต่างๆ ใช้ในการประสานงานกับระบบโมเด็ม ซึ่งได้แก่
- RST (REQUEST TO SEND) คู่กับ CTS (CLEAR TO SEND)
- DTR (DATA TERMINAL READY) คู่กับ DST (DATA SET READY)

โดยจะมีการส่งสัญญาณออกมาเรียกและมีการส่งสัญญาณตอบรับการติดต่อ การสื่อสารโดยใช้ระบบการรับส่งข้อมูลแบบอนุกรมนี้จึงเหมาะสมกับการสื่อสารในระยะทางไกลๆ โดยผ่านทางเครือข่ายต่างๆ เช่นเครือข่ายโทรศัพท์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของอุปกรณ์ที่มีการใช้ในระบบ

เนื่องจากอุปกรณ์แต่ละตัวก็จะมีหลักการในการทำงานและสถาปัตยกรรมของตัวมันไม่เหมือนกันทำให้การใช้งานแตกต่างกันออกไป และการออกแบบระบบทั้งส่วนของ hardware และ software ก็ต้องเป็นไปตามข้อกำหนดของอุปกรณ์ที่ใช้งานจึงมีความจำเป็นที่ต้องรู้ถึงคุณสมบัติของอุปกรณ์ที่จะใช้ ซึ่งสามารถแยกออกได้ดังนี้คือ

1)cpu 8031 ซึ่งถือได้ว่าเป็นหัวใจของระบบ front end data acquisition ซึ่งเป็น MCS(microcontroller single chip)ของบริษัท intel ซึ่งมีรายละเอียดดังนี้

-เป็น cpu ขนาด 8 bit

-มีวงจร oscillator และวงจรสร้าง clock อยู่ในตัว

-มี i/o line 32 line

-ความสามารถขยาย data memory (ram) ได้ถึง 64 kbyte

-ความสามารถขยาย program memory (rom)ได้ถึง 64 kbyte

-มี timer/counter ขนาด 16 bit 2 ตัว

-การติดต่อกับ serial port ใช้วิธี full duplex

-การทำงานทำแบบ boolean coprocessor (คือการทำงานได้ทีละบิต)

ใน8031จะมี registor ที่ใช้ในงานต่างๆซึ่งเรียกรวมว่า SFR(special function registor) ซึ่งจะมี registor ต่างๆดังนี้คือ

-acc(accumulator)ใช้ในการทำงานทางคณิตศาสตร์

-b ใช้ในการคูณกับหารหรือจะทำการเก็บข้อมูลเหมือนข้อมูลอื่นก็ได้

-PSW(program status word) มีขนาด 8 bit ซึ่งแต่ละบิตมีค่าดังนี้คือ

bit1 เป็น reserved bit

bit2 เป็น overflow flag

bit3,4 เป็น registor bank select เพื่อเลือกการทำงานของ registor bank

bit5 เป็น flag

bit6(ac) เป็น auxillary carry flag

bit7 เป็น carry flag

-stack pointer(sp)เป็น registor ขนาด 8 bit เพิ่มและลดค่าโดยการ push,pop และเมื่อมีการ reset ค่าใน sp = 07h ซึ่งจะเห็นได้ว่าเมื่อมีการ reset แล้วจะเริ่มทำงานที่ตำแหน่ง 08h

-dptr(data pointer) เป็น registor ขนาด 16 bit ใช้ในการเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

address ประกอบด้วย dph,dpl ซึ่งจะมีการทำงานทีละ 16 bit หรือ 8 bit ก็ได้

-port0 - port3 เป็น port ในการทำงาน

port0 เป็นทั้งขา address,data ก็ได้ถ้าเป็น address จะเป็น address ในส่วนของ least significant byte ซึ่งการทำงานเป็นส่วนใดขึ้นกับ ALE signal(address latch enable) จาก 8031 ซึ่งจะกล่าวถึงรายละเอียดในภายหลัง

port1 ในที่นี้จะใช้เป็น port ที่ใช้รับข้อมูลจาก i/p port

port2 ทำหน้าที่เป็น address ในส่วนของ msb(most significant byte)

-ip(interrupt priority control)ใช้ในการจัดลำดับการทำงานของ interrupt ซึ่งแต่ละบิตมีความหมายดังนี้คือ

bit0 = external interrupt 0 priority level (PX0)

bit1 = timer 0 interrupt priority level (PT0)

bit2 = external interrupt 1 priority level (PX1)

bit3 = timer 1 interrupt priority level (PT1)

bit4 = serial port interrupt priority level(PS)

bit5 = timer 2 interrupt priority level (PT2)

bit6,7 = reserved bit

-ie(interrupt enable) ใช้ในการ enable interruptแต่ละบิตมีความหมายดังนี้คือ

bit0 เป็น EX0,bit2 เป็น EX1 (EX0,EX1 เป็นการ enable,disable external interrupt0,1 ตามลำดับ

bit1 เป็น ET1,bit3 เป็น ET2(เป็นการ enable หรือ disable timer1,2)

bit5 เป็นการ enable,disable serial port interrupt ถ้าเป็น 0 = disable

bit6 เป็น reserved bit

bit7 (EA) เป็นการ enable,disable interrupt

-tmod(timer/counter mode control)ใช้ในการเลือก mode การทำงานของ timer/counter โดยจะมีรายละเอียดในแต่ละบิตคือ

bit0 - bit3 เป็นของ timer0

bit4 - bit7 เป็นของ timer1 ซึ่งในแต่ละบิตจะมีความหมายคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bit0,1 เป็นการเลือก operating mode ซึ่งมี 3 mode คือ 0,1,2 และถ้ามีการตั้ง bit0,1ให้เป็น 1 ทั้งคู่จะทำให้ timer/counter หยุดทำงาน

bit2 เป็น 1 แสดงถึงการทำงานเป็น counter ถ้าเป็น 0 เป็น timer

bit3 เป็นขา gating control ในที่นี้ไม่มีการใช้จึงกำหนดให้เป็น 0

-tcon(timer/counter control register)ใช้ในการควบคุมการทำงานของ timer/counter ซึ่งจะแบ่งออกเป็น bit ต่างๆดังนี้คือ

bit0,2(bit IT0,IT1) เป็น interrupt0,1 type control bit คือการเลือกขอบของสัญญาณที่จะใช้ interrupt

bit1,3(bit IE0,IE1) เป็น interrupt0,1 edge flag

bit4,6(bit TR0,TR1) เป็น timer0,1 run control bit ตั้งโดยใช้ software ให้ timer on/off

bit5,7(bit TF0,1) timer0,1 overflow flag

-th0,tl0 timer/counter0 ที่ high byte,low byte

-th1,tl1 timer/counter1 ที่ high byte,low byte

-rcap2h,rcap2l timer/counter2 capture register high byte กับ low byte

-scon(serial control register) ใช้ควบคุมการทำงานของ serial port แต่ละบิตมีค่าดังนี้คือ

bit0 = received interrupt flag(ri)

bit1 = transmit interrupt flag(ti)

bit2,3 = tb8,rb8 ตามลำดับเป็น data bit ที่ 9 ในการใช้งาน mode 2,3

bit4 = enable serial reception(ren)

bit5,6,7 = การกำหนด mode ในการทำงานของ serial port (mode 0,1,2,3)

-sbuf(serial buffer) เป็นที่เก็บข้อมูลก่อนที่จะมีการส่งและมาจาก serial port แบ่งออกเป็น 2 ส่วนคือ

received buffer register เป็นตัวที่ใช้ในการเก็บข้อมูลที่จะส่งเข้ามาที่ตัว front end

transmit buffer register เป็นตัวที่ใช้ในการเก็บข้อมูลที่จะทำการส่งไปที่ภายนอก

-pcon(power control register) เป็นส่วนที่จะใช้ในการตั้งค่า SMOD

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปใช้ในการแสวงหาประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละบิตจะมีความหมายดังนี้คือ

bit0 = idle mode bit (IDL) เป็นส่วนที่จะใช้ในการกำหนด mode การทำงานของ idle mode operation

bit1 = power down bit (PD) เป็นส่วนที่ใช้ในการกำหนด mode การทำงานของ power down operation

bit2 = general purpose flag bit (gf0)

bit3 = general purpose flag bit (gf1)

bit4,5,6 = reserved bit

bit7 = SMOD เป็นการกำหนดค่า baud rate ถ้าเป็น 1 จะเป็น double baud rate ในการทำงานใน mode1,2,3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SERIAL INTERFACE

การทำงานของ serial port เป็นลักษณะของ full duplex ซึ่งสามารถทำการส่งและรับข้อมูลได้ในขณะเดียวกันและสามารถทำการอ่านข้อมูลมาเก็บที่ buffer ได้ ทำให้การอ่านข้อมูล byte ที่2โดยข้อมูลที่บิตแรกยังไม่ได้ส่งออกไปจาก buffer (แต่ถ้าข้อมูล byte ที่2ถูกอ่านข้อมูลโดยข้อมูล byte ที่1ยังไม่ได้อ่านออกไปข้อมูล byte ที่2จะเขียนไปทับ byte1) การอ่านเขียนข้อมูลของ serial ต้องผ่าน sbufเสมอ

mode การทำงานของ serial port มีดังนี้คือ

1)mode0 จะเป็นการรับและส่งข้อมูลขนาด 8bit และทำการส่ง LSB ก่อนใน mode นี้จะมีการกำหนดค่า baud rate ไว้แน่นอนแล้วคือ 1ใน12 เท่าของ oscillator freq

2)mode1 มีการส่งและรับข้อมูลทีละ 10 bit โดยมี 1start bit(0) ข้อมูล 8bit(ส่ง LSB ก่อน)และมีการส่ง stop bit(1) การส่งข้อมูลจะส่ง LSB ก่อนส่วนในการรับข้อมูลจะต้องมีการรับ stop bit เข้ามาที่ RB8 ใน SCON การทำงานใน mode นี้จะแปรค่า baud rate ได้

3)mode2 มีการส่งและรับข้อมูลทีละ 11bit โดยมี 1 start bit(0), 8data bit,1stop bit(1),1 pogrammable bit(bit9) โดยในบิตนี้จะถูกส่งมาที่ TB8(SCON) จะมีค่าเป็น 0,1 แล้วแต่ค่า parity bit (ใน PSW)ซึ่งจะถูกส่งออกมาที่ bit นี้ ส่วนในการรับข้อมูลข้อมูลในบิตนี้จะถูกส่งไปที่ RB8(SCON) ค่า baud rate จะสามารถแปรค่าได้

4)mode3 มีการรับส่งข้อมูลทีละ 11 bit โดยมี 1 start bit(0), 8 data bit,1 stop bit(1) ส่วนบิตที่ 9 เป็น programmable bit การทำงานใน mode นี้จะเหมือน mode2 แต่ค่า baud rate แปรค่าได้มากกว่า

การ reset ใน8031นี้จะทำได้ก็ต่อเมื่อมีการต่อขา RST มีค่าเป็น 1 อย่างน้อย 2 machine cycle เมื่อมีการ reset จะเกิด internal reset โดย internal reset นี้จะเกิดที่ cyc;e ที่2และจะเกิดการทำซ้ำจนกว่า RST เป็น 0 แต่การ reset จะไม่มีผลกับ internal ram และจะมีการเริ่มงานที่ address 0007h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING ในการทำงานของ CPU 8031

timing ในการทำงานของ cpu 8031 จะใช้ machine cycle ซึ่งในแต่ละ machine cycle จะประกอบด้วย 6 state ซึ่งในแต่ละ state ประกอบด้วย 2 phase (phase1,2) โดยในแต่ละเฟสจะใช้เวลา 1 clock(oscilator period)ซึ่งในระบบนี้จะใช้ crystal ขนาด 12 MHz จึงมีผลให้ 1 machine cycle ใช้เวลา 1 ไมโครวินาที และการทำงานในส่วนของคณิตศาสตร์และตรรกศาสตร์จะเกิดที่ phase1 ส่วนของการย้ายข้อมูลของ internal register จะเกิดที่เฟส2 และขณะที่ 8031จะส่ง address ออกมาที่ภายนอกจะมีการส่ง ALE ออกมาข้อมูลที่ออกมาจาก port0 จะเป็น address ส่วนกรณีที่ต้องการติดต่อกับ external memory (มี 2 ส่วนคือ rom,ram ซึ่งในแต่ละส่วนจะมีการติดต่อไม่เหมือนกัน)โดยในส่วนของ rom ต้องมีการส่ง PSEN(program strobe enable) ออกมาจากตัว 8031 และในส่วน rom จะทำการอ่านเท่านั้น ซึ่ง ALE จะถูกส่งออกมา 2 ครั้งใน 1 machine cycle และเมื่อมีการ enable ALE(1) ก็จะมีการตรวจว่าต้องการติดต่อกับ rom ก็ต้องมีการส่ง PSEN (enable '0')ออกมาด้วย จึงเปรียบได้ว่า PSEN คล้ายกับ read จาก cpu PSEN นั้นจะถูกสร้าง 2 ครั้งใน 1machine cycle ส่วนในกรณีของการติดต่อกับ ram จะมีการส่ง RD,WR จาก8031และไม่มีการส่ง PSEN จาก8031

การติดต่อกับส่วนของ program memory จะใช้ address 16 bit เท่านั้นแต่ถ้าติดต่อกับส่วนของ data memory จะทำได้ทั้ง 8bit,16 bit(การติดต่อแบบ 16 bit ใช้ในการติดต่อกับ internal,external ram ก็ได้)

ในกรณีที่มีการ interface ระหว่าง cpu กับส่วนใดต้องคำนึงถึงการ drive ที่8031ทำได้ด้วยคือ

-o/p buffer port1,2,3 drive ได้ 4 LSTTL

-o/p buffer port0 drive ได้ 8 LSTTL

คำสั่งที่มีการใช้ใน8031นั้นบางคำสั่งจะมีลักษณะของ read-modiflied-write คือมีการอ่านจากในส่วนของการ latch ไปทำตามคำสั่งแล้วทำการเขียนค่าใหม่ไปที่ตัว latch (ไม่ได้อ่านจากขา port) เช่นคำสั่งพวก anl,orl,xrl,jbc,cpl,inc,dec,djnz,mov bit,clr, set เป็นต้น

mode การทำงานของ timer/counter ใน8031มีลักษณะดังนี้คือ

1)mode0 ใน mode นี้จะใช้13 bit ของ register ซึ่งจะเริ่มจากทุก bit เป็น 1 จนถึงทุก bit เป็น 0 ซึ่งเมื่อมีค่าเป็น0จะไปทำการ set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timer interupt flag (tf1)13 bit ของ register จะใช้ 8 bit ของ TH1 5 bit ของ TL1 :

2)mode1 เหมือน mode1 แต่ใช้ register 16 bit(8bit ของ TH1, 8bit ของ TL1)

3)mode2 เป็น 8bit counter ซึ่งเก็บใน TL1 และเมื่อค่าใน TL1 เป็น 0 จะทำการ auto reload ค่าใน TH1และเมื่อมีการ reload ค่าใน TH1 ไปแล้วจะไม่มีการเปลี่ยนค่าใน TH1

4)mode3เป็นการเพิ่ม 8bit ใน timer/counter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขบวนการควบคุม

ขบวนการควบคุมเป็นขบวนการที่จะควบคุมสัญญาณเพื่อให้เอาท์พุทของขบวนการที่ถูกควบคุมให้มีค่าประมาณจุดกำหนด (set point) สำหรับตัวควบคุมทั้งหมดจะยึนอยู่บนพื้นฐานคือการสร้างค่าสัญญาณผิดพลาด (error signal) ขึ้น คือ

$$E = SP - PV \qquad ; SP = \text{set point}$$

ซึ่งจุดประสงค์ของขบวนการควบคุมคือพยายามให้ค่าผิดพลาดนี้มีค่าเข้าใกล้ศูนย์ที่สุด

การควบคุมที่ง่ายที่สุดคือขบวนการปิดหรือเปิดเพื่อลดค่าผิดพลาด แต่อาจจะทำให้ระบบภายนอกนั้นทำงานผิดพลาดขึ้น ดังนั้นการควบคุมที่ดีกว่าวิธีปิดเปิดระบบเป็นลักษณะที่มีการฟีดแบค (feed back) คือมีการนำเอาเอาท์พุทที่แล้วมาเข้าในการควบคุมด้วย ซึ่งการฟีดแบคนี้มีการฟีดแบคแบบบวก (positive) และการฟีดแบคแบบลบ (negative)

ระบบควบคุมที่ทำงานได้ดีกว่าระบบปิดเปิดมีหลายประเภท ได้แก่

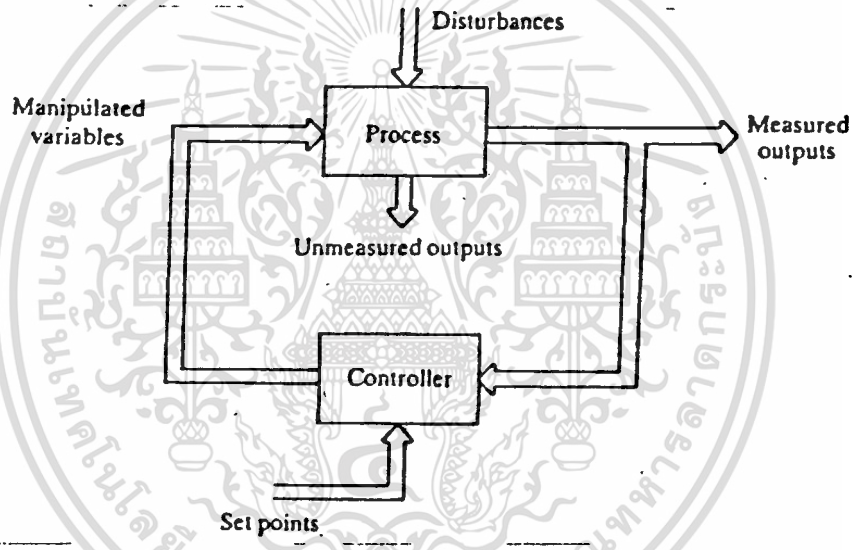
-proportional control (P) จะทำให้สัญญาณเอาท์พุทมีการเปลี่ยนแปลงแบบเชิงเส้น (linear) ด้วยค่าผิดพลาดที่มีค่าน้อยมากหรือประมาณศูนย์ ซึ่งค่าผิดพลาดนี้สามารถลดลงได้โดยการเพิ่มค่าอัตราการขยาย (gain) แต่จะทำให้ proportional band ลดลง

-integral control (I) ค่าผิดพลาดที่มีขนาดใหญ่นั้นเนื่องจากการเปลี่ยนแปลงอย่างรวดเร็วค่าเอาท์พุท ค่าเอาท์พุทนี้จะเปลี่ยนแปลงจนกระทั่งค่าเอาท์พุทมีค่าเป็นศูนย์ การที่ค่าผิดพลาดมีค่าเป็นศูนย์ได้นั้นเนื่องจากตัวควบคุมแบบ integral ตัวควบคุมแบบ integral นี้อย่างเดียวจะทำงานช้า แต่ก็เป็นการควบคุมที่ดี และสามารถนำมาควบคุมร่วมกับการควบคุมแบบ proportional (PI)

-derivative control การควบคุมแบบนี้จะมีผลต่ออัตราการเปลี่ยนแปลงของเอาท์พุทที่มีค่าสูงๆ ซึ่งอัตราการเปลี่ยนแปลงนี้ถ้ามีค่าสูงเกินไปจะทำให้ค่าผิดพลาดมีค่าสูง

การควบคุมทั้ง 3 แบบนี้ สามารถรวมกันเป็นการควบคุมแบบใหม่ได้เป็นการควบคุมแบบ PID (three-mode-control) ซึ่งเป็นการควบคุมที่ดีที่สุดระบบหนึ่ง และระบบนี้สามารถให้ค่า transient ที่ดีและให้ค่าเอาท์พุทที่สภาวะคงที่ (steady state) มีค่าใกล้จุด set point ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

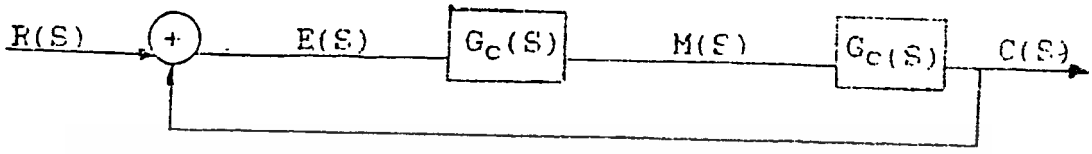


BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การควบคุมแบบ PID (PID controller)

PID control เป็นวิธีที่ถูกนำมาใช้อย่างกว้างขวางในโรงงานอุตสาหกรรม เพราะว่าการควบคุมแบบนี้ต้องการข้อมูลที่ใช้ในขบวนการน้อยมาก และวิธีในการพิจารณาค่าพารามิเตอร์ต่างๆก็เป็นที่รู้จักกันดีและเข้าใจได้ง่าย



G(S) คือการควบคุมแบบ 3 เทอม และตัวควบคุมซึ่งถูกออกแบบมาใช้ในงานนั้น

รูปแบบของระบบควบคุมแบบ PID (The PID Algorithm)

ระบบควบคุมแบบ PID นี้ สามารถเขียนในรูปของสมการได้ดังนี้

$$G_c(S) = M(S)/E(S) = K_c(1+1/T_{is}*T_{ds}) \qquad \text{-------(1)}$$

เมื่อ K_c คืออัตราการขยายของตัวควบคุม

T_i คือintegral action time

T_d คือderivative action time

ระบบควบคุมตามสมการข้างต้นนี้จะเป็นระบบแบบideal เพราะว่าสมมติว่าทำการควบคุมทั้ง 3 พจน์นี้ (P,I,D) ไม่ขึ้นต่อกัน

ระบบสมการของระบบควบคุมแบบ PID (Ideal PID Control)

จากสมการที่(1) สามารถเขียนให้อยู่ในรูปของเวลา(time domain) จะได้ดังสมการ

$$m(t) = K_cT_d(de(t)/dt)+K_ce(t)+K_c/T_i(\int e(t)d(t)) \qquad \text{----(2)}$$

จากสมการที่(2) จะสามารถแปลงให้อยู่ในรูปของสมการแบบไม่ต่อเนื่องได้ ซึ่งวิธีหนึ่งซึ่งสามารถทำได้คือใช้การทำอนุพันธ์ลำดับที่หนึ่ง(frist order finite difference) จะสามารถเขียนได้ดังสมการ

$$(df/dt)|_k = (f_k-f_{k-1})/\Delta t \quad , \quad \int e(t) = \sum_{k=0}^{n} e_k \Delta t \qquad \text{---(3)}$$

เมื่อนำไปใช้กับสมการที่(2) จะได้

$$m(n) = K_c[T_d(e_n-e_{n-1})/\Delta t + e_n + (\sum_{k=0}^{n} e_k \Delta t)/T_i] \qquad \text{------(4)}$$

ซึ่งจะได้พารามิเตอร์ใหม่ขึ้นมาเป็น three term control

$K_c = K_c$ (propotional term)

$K_i = K_c*T_s/T_i$ (integral term)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$K_d = K_c * T_d / T_s \quad \text{(differential term)}$$

เมื่อ $T_s = \Delta t$ เท่ากับช่วงเวลาในการสุ่มค่า (sampling interval)

แล้วกำหนด $S_n = S_{n-1} + e_n$; S_n = ผลรวมของค่าผิดพลาด

จะสามารถเขียนสมการ (4) ได้เป็น

$$m(n) = K_p e_n + K_i S_n + K_d(e_n - e_{n-1}) \quad \text{----------(5)}$$

ในทางปฏิบัติจริงแล้ว ช่วงเวลาของการสุ่ม T_s จะขึ้นความเร็วในกับการทำงานของเครื่องคอมพิวเตอร์ ดังนั้นในการปฏิบัติจะต้องมีการกำหนดค่าช่วงเวลาให้อยู่ในค่าที่ต้องการและสามารถที่จะทำได้

ในงานควบคุมจะมีวิธีต่างๆที่ใช้ในการซิงค์โครไนซ์ของระบบได้แก่

- polling
- การรับ interrupt จากภายนอก
- การ ballst coding
- การใช้สัญญาณ real time clock

สำหรับ 2 วิธีแรกคือpollingและinterrupt จะใช้การส่งสัญญาณเมื่อพร้อมที่จะทำการควบคุมซึ่งสัญญาณนี้จะถูกส่งในช่วงของการสุ่มแต่ละครั้ง แต่สำหรับข้อแตกต่างระหว่าง 2 วิธีนี้คือในการpolling เครื่องคอมพิวเตอร์ที่ใช้ในการควบคุมจะทำการอ่านข้อมูลซ้ำเป็นรอบ แต่ส่วนวิธีinterruptนั้นจะแตกต่างคือเครื่องจะทำการอ่านเมื่อมีสัญญาณinterruptส่งเข้ามาบอกเครื่องคอมพิวเตอร์

วิธีการpollingจะเหมาะสมกับระบบเล็กๆเท่านั้น แต่ก็มีประโยชน์ในการเขียนโปรแกรมซึ่งสะดวกกว่า

<u>การควบคุมแบบ PID ในทางปฏิบัติ</u>

ในทางปฏิบัติการควบคุมจะไม่เป็นไปตามสมการที่กล่าวมาแล้วในตอนต้น ดังนั้นจะต้องมีการปรับปรุงให้ใช้ในทางปฏิบัติ ซึ่งสรุปได้เป็นหัวข้อใหญ่ๆดังนี้

(1) bumpless transfer

(2) saturation (integral wind-up)

(3) noise

(4) modified integral and derivation calculation

<u>(1) bumpless transfer</u>

จากสมการที่(1)นั้น ในสภาวะคงที่(steady state) สมมติว่ามีค่าผิดพลาดเป็นศูนย์(zero error) แต่ในทางปฏิบัติจะไม่เป็นเช่นนั้น ดังนั้นจากสมการที่(1)จึงต้องมีการเพิ่มค่าคงที่ขึ้นมาอีกหนึ่งพจน์เพื่อแทนค่าผิดพลาดที่ไม่เป็นศูนย์ในสภาวะคงที่นี้ จะได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m_n = K_p e_n + K_i S_n + K_d(e_n - e_{n-1}) + MV$$

ขนาดของ MV คิดจากจุดปฏิบัติการของตัวควบคุม ซึ่งในสภาวะคงที่ MV จะแปรตามโหลดด้วย

ซึ่งวิธีแก้ปัญหานี้คือ จากสมการที่(1) นำมาหาการอนุพันธ์ จะได้

$$dm(t)/dt = K_c(de(t)/dt) + K_i e(t) + K_d(d^2e(t)/dt^2) \quad ----(6)$$

ซึ่งค่าอนุพันธ์นั้น $dm(t)/dt$ สามารถแทนด้วย $m_n - m_{n-1}$

จากสมการที่(4) จะได้

$$\Delta n = m_n - m_{n-1} = K_c[(e_n - e_{n-1}) + \Delta t/T_i * e_n + T_d/\Delta t * (e_n - 2e_{n-1} + e_{n-2})$$

--------(7)

ซึ่งแทนค่า Δt ด้วย T_s ได้

$$\Delta m_n = K_c[(1 + T_s/T_i + T_d/T_s)e_n - (1 + 2*T_d/T_s)e_{n-1} + T_d/T_s * e_{n-2}]$$

--------(8)

กำหนดให้ $k_1 = K_c(1 + T_s/T_i + T_d/T_s)$

$k_2 = -(1 + 2*T_d/T_s)$

$k_3 = T_d/T_s$

$$\Delta m_n = k_1 e_n + k_2 e_{n-1} + k_3 e_{n-2} \quad --------(9)$$

จากสมการที่(9) สามารถนำมาเขียนเป็นสมการได้ง่ายมาก

ถ้ากำหนดให้ $e_n = r - c_n$ เมื่อ r = จุดกำหนด(set point) จากสมการ(8) จะเห็นได้ว่าค่า r ซึ่งจะต้องคงที่จะปรากฏในเทอมอินทิกรัลเท่านั้น ดังนั้นในการปฏิบัติงานควรป้องกันโดยให้ค่า T_s/T_i มีค่าเท่ากับศูนย์หรือมีค่าน้อยมาก

(2)การอิ่มตัว(saturation)

ในทางปฏิบัติค่า m นั้นจะถูกจำกัดโดยข้อบังคับทางกายภาพของระบบ ซึ่งถ้าค่าต่างๆมีค่าเกินกว่าค่าทางเอาท์พุทสูงสุดจะทำให้ประสิทธิภาพของการผิดแบบเสียหาย ดังนั้นในการออกแบบระบบที่ดีจะต้องแน่ใจว่าเหตุการณ์นี้จะเกิดขึ้นเฉพาะสภาวะที่ผิดปรกติเท่านั้น

ในขบวนการที่มีการทำงานที่สูงหรือต่ำกว่าจุดกำหนดมากนั้น จะยากที่จะบังคับให้ระบบคงสภาพที่ได้ออกแบบไว้ อย่างเช่นถ้าเลือกการควบคุมแบบ PI ค่าเทอมของอินทิกรัลจะสูงขึ้นมาก ในขณะที่ช่วงอินพุทสูงขึ้น แต่เมื่อค่าอินพุทสูงกว่าจุดสูงสุดที่กำหนดไว้ ค่าอินทิกรัลกลับลดลงแม้อินพุทจะสูงขึ้นไปก็ตาม ดังนั้นค่าอินทิกรัลจึงเป็นส่วนที่รักษาสภาวะทางเอาท์พุท ณ จุดสูงสุดที่ได้กำหนดไว้ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์นี้เรียกว่า integral saturation หรือ integral wind-up ซึ่งเทคนิคแบบนี้มีหลายแบบเช่น จากัดช่วงกำหนด คือถ้าค่าผลรวมของค่าผิดพลาดสูงกว่าจุดกำหนดให้ค่าผลบวกนั้นเท่ากับค่าสูงสุดหรือถ้าค่ากว่าค่ากำหนดค่าสุดให้ผลรวมเท่ากับค่ากำหนดค่าสุด

(3)สัญญาณรบกวน(noise)

สัญญาณรบกวนที่ถูกสร้างขึ้นอาจจะมีขนาดและความถี่ซึ่งสามารถไปรบกวนต่อสัญญาณในระบบและช่วงเวลาการสุ่ม(sampling interval)ได้ ซึ่งในการลดสัญญาณรบกวนกระทำได้โดยการต่อ filter เข้าไปก่อนการสุ่มสัญญาณซึ่งรูปสมการลำดับที่หนึ่งของ filter ที่ใช้คือ

$$G(S) = 1/(1+T_f*S) \quad \text{เมื่อ } T_f = T_s/2$$

ซึ่งจะสามารถลดสัญญาณรบกวนลงได้ถึง 90%

(4)modifies integral and derivation calculator

ความเรียบของสัญญาณ(smoothing)จะสามารถทำได้โดยใช้วิธีอนุพันธ์เพื่อหาค่าเฉลี่ยของสัญญาณซึ่งได้จากการสุ่ม เทคนิคหนึ่งที่ใช้กันมากคือ four-point center difference method โดยกำหนด

$$de/dt = \Delta t/T_s = 1/6*T_s*(e_n+3*e_{n-1}-3*e_{n-2}-e_{n-3}) \quad ---(10)$$

แทนสมการที่(10)ลงในสมการที่(4)

$$m_n = K_c[T_d/(6*T_s)*(e_n+3*e_{n-1}-3*e_{n-2}-e_{n-3})+e_n+T_s/T_i*\sum_{k=0}^{n} e_k \quad --------(11)$$

และกำหนด $S_k = S_{k-1} + e_k$

$$m_k = p_1e_k - p_2e_{k-1} - p_3e_{k-3} + p_4e_{k-4} \quad --------(12)$$

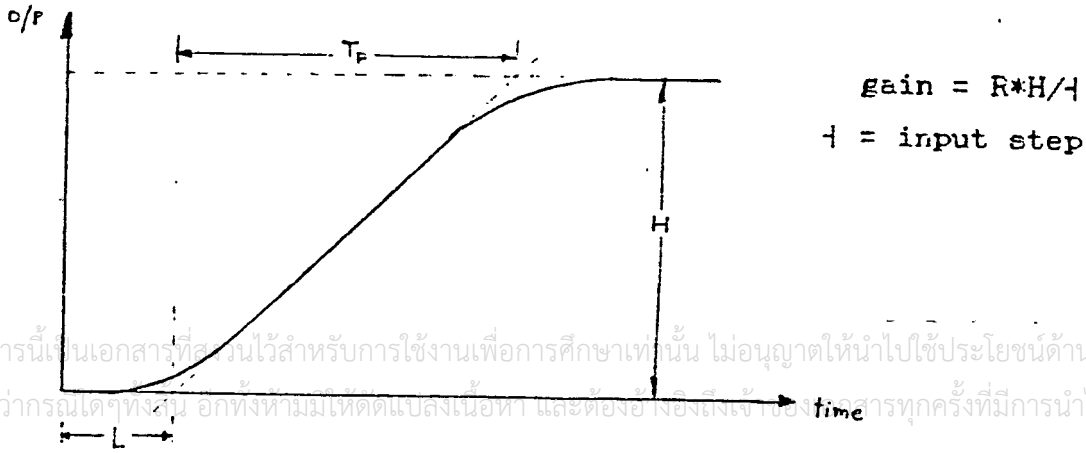
โดยที่

$$p_1 = 1+ (K_c*T_d/6/T_s) \quad ; \quad p_2 = (K_c*T_d)/(2*T_s)$$

$$p_3 = (K_c*T_d)/(6*T_s) \quad ; \quad p_4 = (K_c*T_s)/T_i$$

การปรับแต่งและการเลือกช่วงเวลาการสุ่ม

(tuning and chiose of sample interval)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 ส่วนประกอบของระบบ

-รายละเอียดของส่วนของ hardware

1)block ของการทำงานของ cpu 8031

2)block ของการทำงานของการ decode ส่วนของการติดต่อ

3)block การทำงานของส่วนของหน่วยความจำ

4)block ของการทำงานของการ display

5)block ของการทำงานของ input port

6)block ของการทำงานของ output port

7)block ของการทำงานในส่วนของ serial port

-รายละเอียดของส่วนของ software บน front end

1)ส่วนของการ initialization

2)ส่วนของการ self test

3)ส่วนของ input

4)ส่วนของการ process

5)ส่วนของ output

-แผนภูมิของการทำงานของ front end

1)แผนภูมิของการทำงานของระบบบน front end

2)แผนภูมิของการ initialization serial port

3)แผนภูมิของการรับ channel จาก host

4)แผนภูมิของการรับค่าอุณหภูมิจาก host

5)แผนภูมิของการ process และการส่งข้อมูลไป
แสดงผลที่ host

-รายละเอียดของส่วนของ software บน host

1)การ initialization

2)การติดต่อของ serial port

3)แผนภูมิของการทำงานทั้งหมดของ host

4)แผนภูมิของการส่งผ่านค่าต่างๆของ host

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของส่วน FRONT END DATA ACQUISITION

ในระบบของ front end จะแบ่ง hardware ออกเป็นส่วนต่างๆดังนี้

1)ส่วนของ i/p port คือส่วนของ transducer ที่จะทำหน้าที่ในการรับสัญญาณ i/p ผ่านมาทางa/d เข้ามาที่ port i/p ของตัว8031ซึ่งในส่วนของ port i/p ของ front end นี้จะประกอบด้วย port i/p ทั้งหมด 256 channels โดยจะแบ่งออกเป็น 32 card ในแต่ละ card จะประกอบด้วย 8 port i/p ซึ่งในขณะใช้งานอาจจะมีเพียงบาง card ที่มีการต่ออยู่เท่านั้น

2)ส่วนของระบบ front end จะทำหน้าที่ในการประมวลผลและติดต่อกับ host รวมถึงการติดต่อกับ port i/p ,port o/p และควบคุมการส่งผ่านข้อมูลต่างๆ

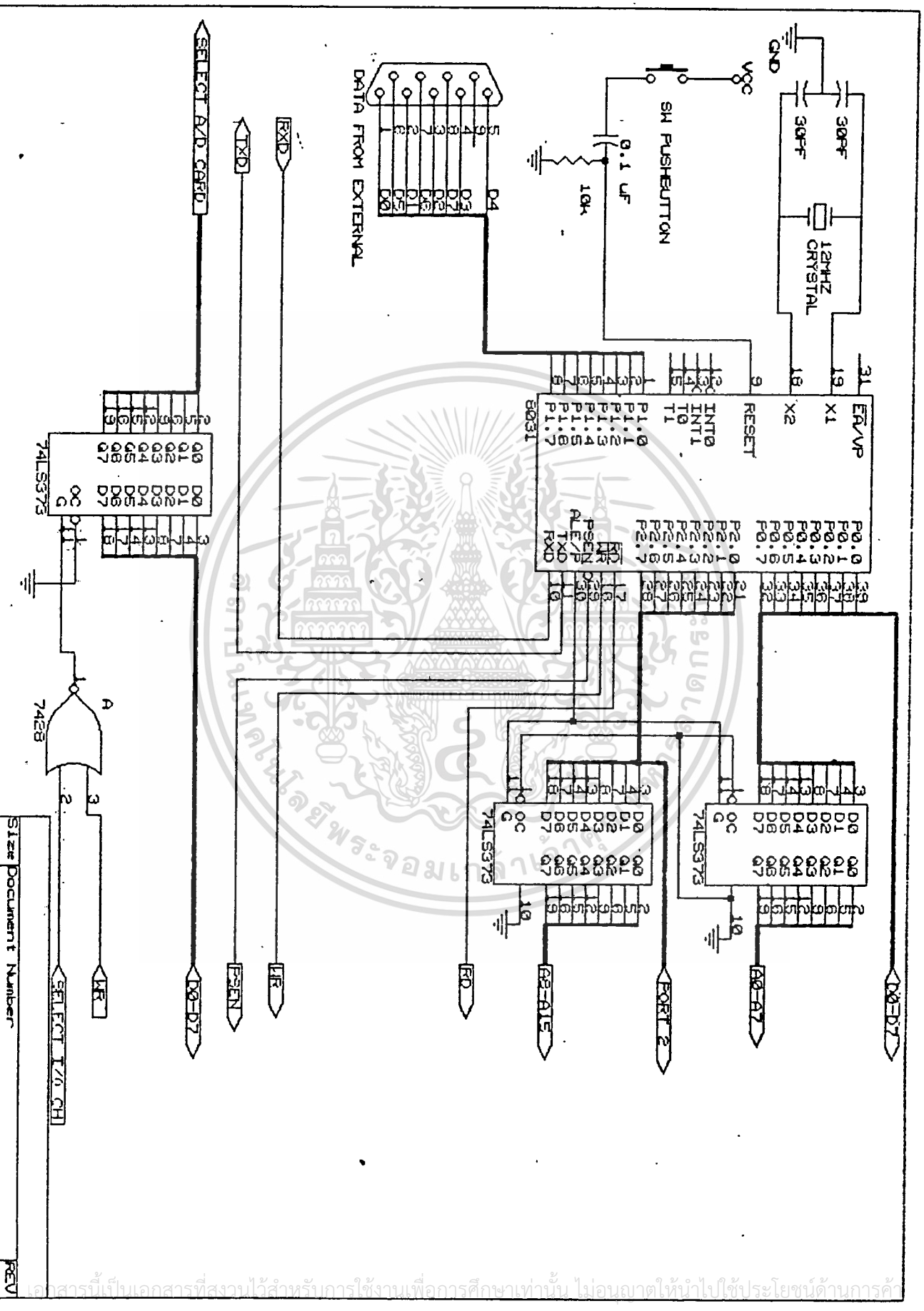
3)ส่วนของ port o/pจะทำหน้าที่ในการรับข้อมูลจาก front end (ข้อมูลในที่นี้เป็นข้อมูลที่ผ่าน DAC(digital to analog convertor) แล้วส่งมาทำการควบคุมระบบต่อไป

ในส่วนของ software แบ่งออกเป็นส่วนต่างๆดังนี้คือ

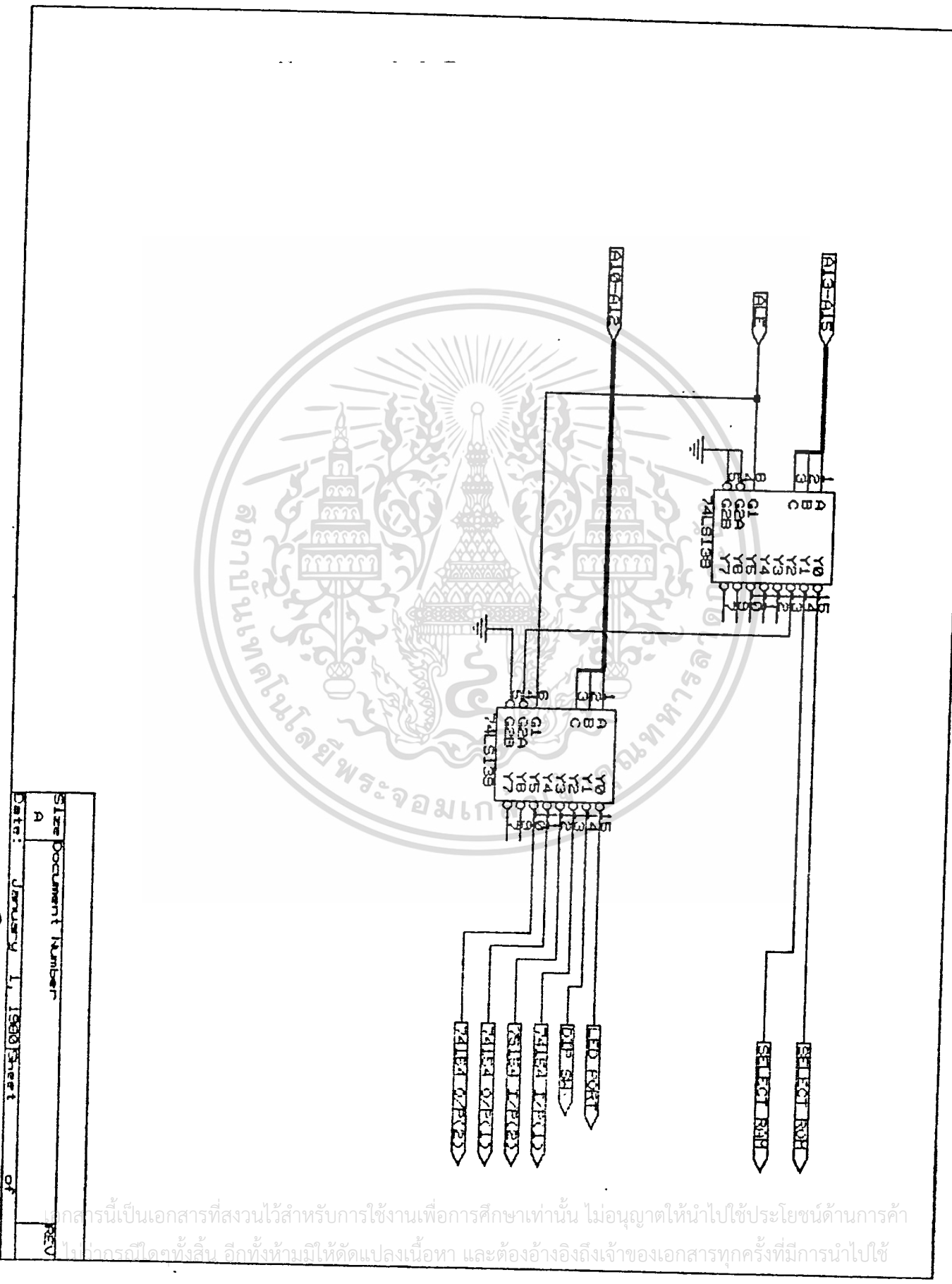
1)software ในส่วนของ host เป็น software ในส่วนของการควบคุมการทำงานของ host เช่น การรับคำสั่งและค่าต่างๆตั้งแต่ค่า max,min, set point, ค่าคงที่ใน pid process(kd,ki) เป็นต้น รวมทั้งติดต่อกับ front end และแสดงผลการทำงานของระบบให้ user โดยการแสดงกราฟที่ host ด้วย

2)software ในส่วนของ front end จะทำหน้าที่ในการตรวจสอบความพร้อมของระบบของตัว front end และส่วนที่ทำการติดต่อด้วย เช่น port i/p และทำการส่งข้อมูลไปที่ host เพื่อให้ host ตัดสินใจควบคุมได้ถูกต้อง รวมถึงทำหน้าที่ในการประมวลผลส่งให้ host อีกด้วยซึ่งรายละเอียดในแต่ละส่วนจะกล่าวในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK การทำงานในส่วนของการ DECODE

เนื่องจากในระบบมีทั้งส่วนของ port i/p, port o/p,failed port, rom,ram จึงต้องมีส่วนที่ทำหน้าที่ในการ decode ว่าส่วนที่ต้องการติดต่อขณะนั้นเป็นส่วนใด โดยกำหนดให้แต่ละส่วนมี address ที่แน่นอน แล้วสร้างส่วนของการ decode address ซึ่งในส่วนของระบบนี้จะใช้ 74LS138 เป็นตัว decode ส่วนที่มีการติดต่อว่าเป็นส่วนใด ส่วนของ port ที่ทำการติดต่อด้วยมีดังนี้คือ

-rom ใช้ address ในช่วง 0000h-1fffh(8kbyte)

-ram ใช้ address ในช่วง 2000h-3fffh(8kbyte)

-ส่วนที่ทำหน้าที่ในการติดต่อกับ channel ต่างๆ

ซึ่งจากรูปจะเห็นได้ว่าจะใช้ 74LS138 2ตัวเป็นตัว decode โดยเมื่อมีการส่ง address จาก8031ก็จะมีการส่ง address ไป decode ที่ 74LS138ตัวที่1และ2 โดยที่74LS138 ตัวที่1จะ decode จากบิตที่13-15 ของ address ที่ส่งออกมาโดยจะ decode ว่าส่วนที่ต้องการติดต่อนั้นเป็นส่วนของ rom,ram,ส่วนของ channel เป็นต้น ส่วน 74ls138 ตัวที่2 จะทำการ decode ที่บิตที่10 - 12 ของ address ซึ่งจะทำหน้าที่ในการตรวจสอบส่วนย่อยในการทำงาน(เช่นในกรณีที่ address อยู่ในส่วนของ channel ก็ต้องทำการ decode ต่อว่าเป็นการติดต่อกับ led port, serial เป็นต้น)

จากรูปจะเห็นได้ว่าขา enable ของ 74ls138 จะมีทั้งสิ้น3ขาคือ G2A,G2B ซึ่งต้องเป็น '0' เสมอและขา G1เป็น '1' จากข้อกำหนดข้างต้นจึงเป็นผลให้ขา G1 ของทั้ง2ขาต้องต่อกับขา ALE ของตัว8031และเมื่อมีการส่ง address จาก8031ก็จะมีการ enable(เป็น1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK การทำงานของหน่วยความจำ

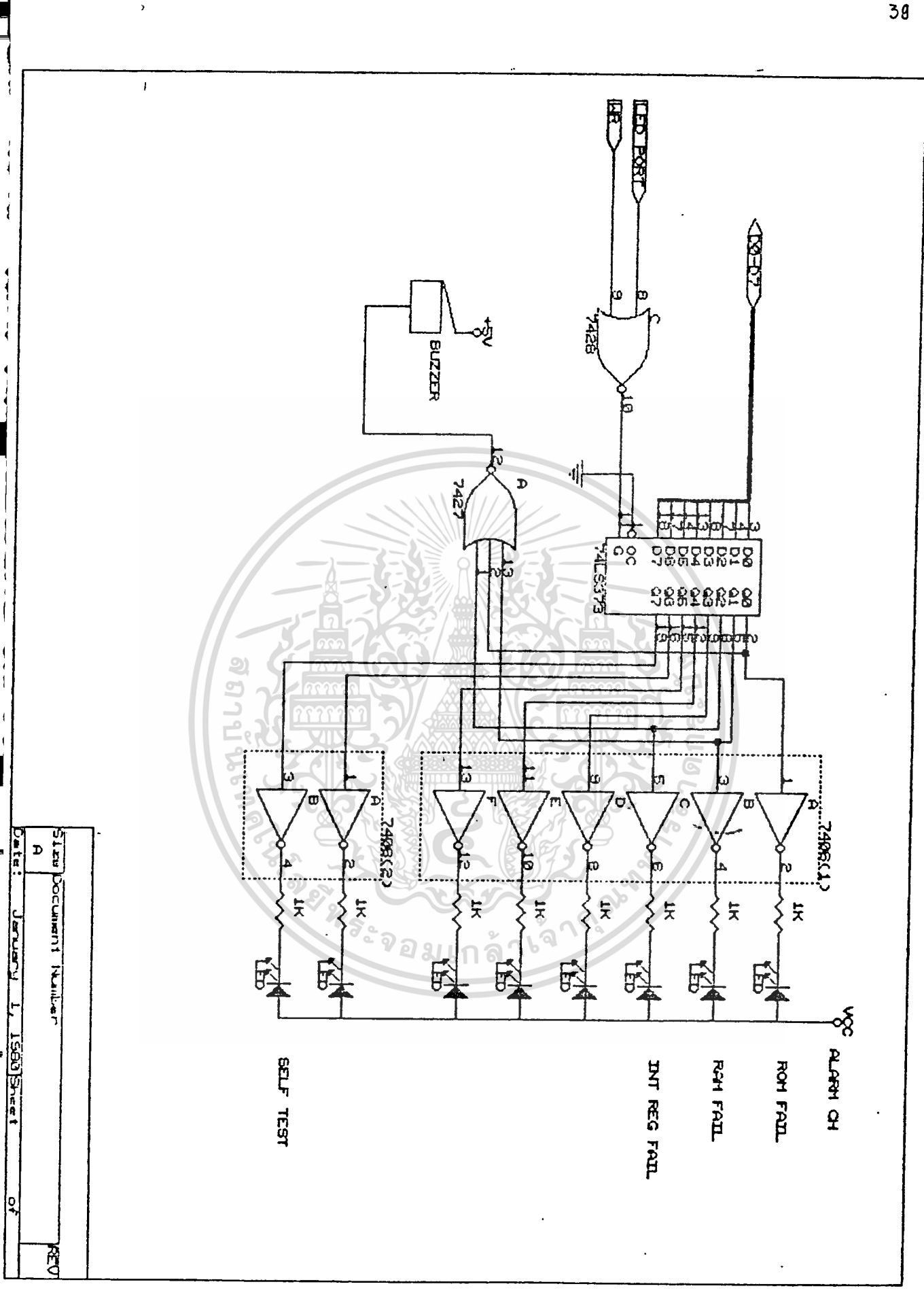
ในระบบนี้จะมีการติดต่อกับหน่วยความจำใน2ส่วนคือ rom,ram โดย rom จะเป็นส่วนที่ใช้ในการเก็บ program ที่ใช้ควบคุมการทำงานของ front end รวมถึงส่วนที่ใช้ควบคุมการติดต่อกับ host ส่วนของ ram จะเป็นส่วนที่ใช้เก็บข้อมูลที่ต้องมีการเปลี่ยนแปลงอยู่เสมอ เช่น อุณหภูมิที่ต้องการทำการควบคุม,channel ที่ต้องการควบคุม,ชนิดของ PID ที่ต้องการทำการควบคุม, ข้อมูลของ i/p port ที่มีการอ่านเข้ามา เป็นต้น

การเลือกการติดต่อกับ address ในส่วนของ rom,ram จะมีการ decode จากตัว 74LS138 ดังที่มีการกล่าวไว้ในส่วนของ block การ decode

นอกจากนั้นการติดต่อกับส่วนของ rom ยังจะมีการส่ง PSEN signal (program send enable)ออกมาจาก cpu 8031 ซึ่งสัญญาณนี้จะมีการส่งออกมาเมื่อมีการติดต่อกับ rom เท่านั้น และในส่วนของ PSEN นี้จะมีการ enable ที่ '1' จึงต้องมีการส่งไปผ่านตัว convertor เพื่อทำการกลับสัญญาณเป็น '0' เป็นสัญญาณไบหริกที่ขา output enable ของ rom และต้องมีการส่งสัญญาณ read ออกมาจาก8031เพื่อให้ rom ส่งข้อมูลออกมา ซึ่ง rom ที่ใช้อยู่ในระบบนี้จะมีขนาด8Kbyte

การติดต่อกับส่วนของ ram เมื่อมีการ enable จากตัว74LS138แล้วจะมีการส่งสัญญาณนี้ไป enable ที่ขา CS1 เพื่อให้ ram ทำงานรวมถึงยังต้องมีการส่งสัญญาณ write,read ออกมาจาก8031ออกมา enable ที่ตัว ram ซึ่ง ram ในขณะที่มีการใช้งานของ ram PSEN จะไม่ถูกส่งออกมา ในส่วนนี้จะใช้ ram ขนาด8Kbyte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK ในส่วนของการ DISPLAY

การแสดงผลในส่วนนี้จะเป็นการแสดงสภาวะในการทำงานของระบบคือ ถ้ามีความผิดพลาดของระบบเกิดขึ้นต้องมีการแสดงผลที่ในส่วนนี้ ซึ่งมีทั้งส่วนของ buzzle,led ซึ่งในแต่ละส่วนนี้จะมี address ต่างกันออกไป เมื่อมีความผิดพลาดขึ้นก็จะมีการส่ง code มาแจ้งที่ port เหล่านี้ ข้อผิดพลาดที่จะเกิดขึ้นในระบบมีดังนี้คือ

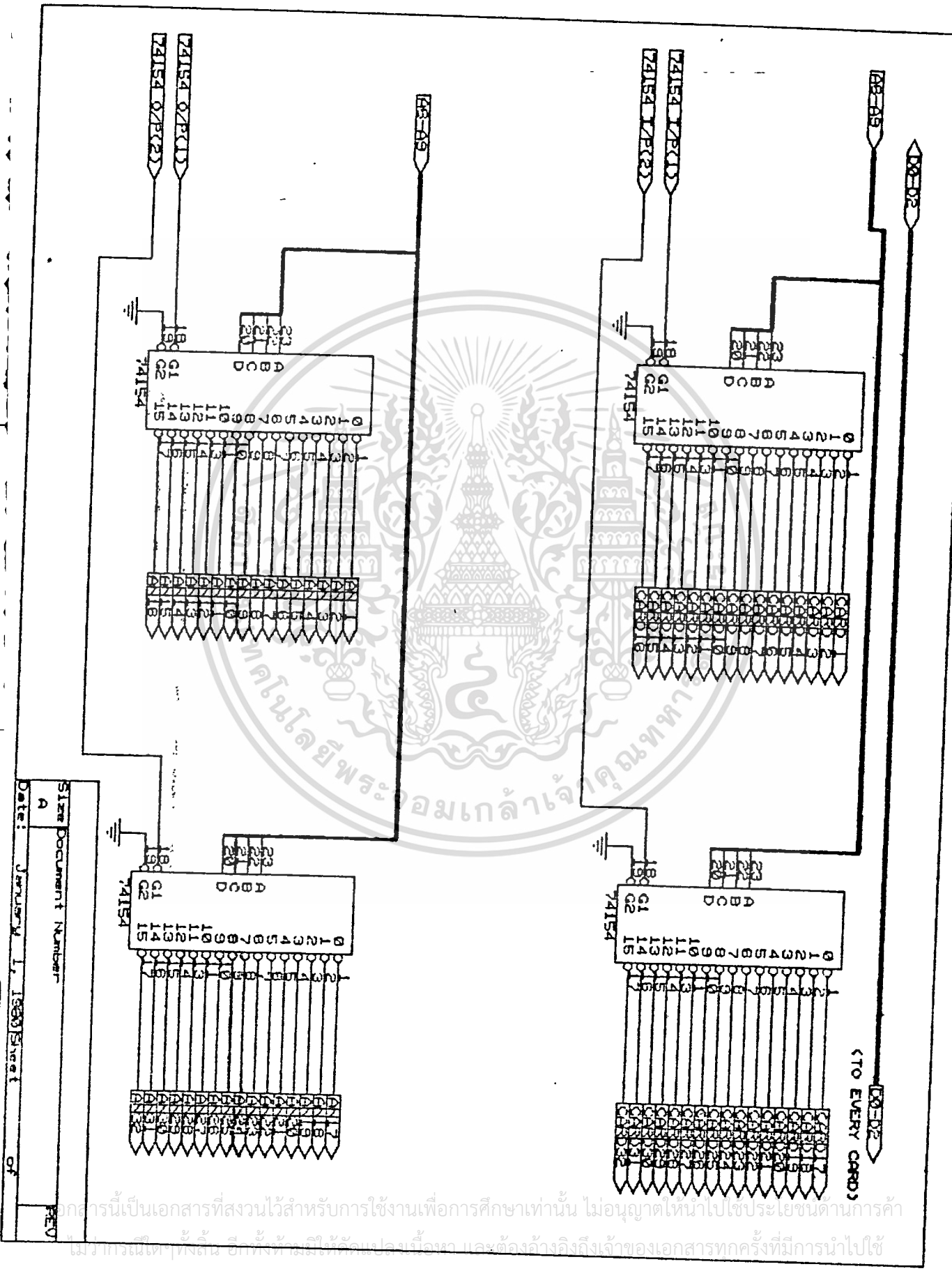
-failed ที่เกิดจากการ selftest คือการที่ front end ตรวจสอบพบว่าระบบที่กำลังจะเริ่มทำงานนี้มีข้อผิดพลาดเกิดขึ้นเช่น ผิดพลาดในส่วนของ ram,rom,หรือการติดต่อกับส่วนใดผิดพลาดไป ซึ่งรายละเอียดในการตรวจสอบข้อผิดพลาดนี้จะกล่าวถึงในภายหลัง

-ผลจากการทำงานของระบบพบว่าระบบมีการทำงานผิดพลาดจากค่าที่ตั้งไว้ เช่น มีค่าอุณหภูมิของการทำงานเกินค่าที่ตั้งไว้เป็นต้น

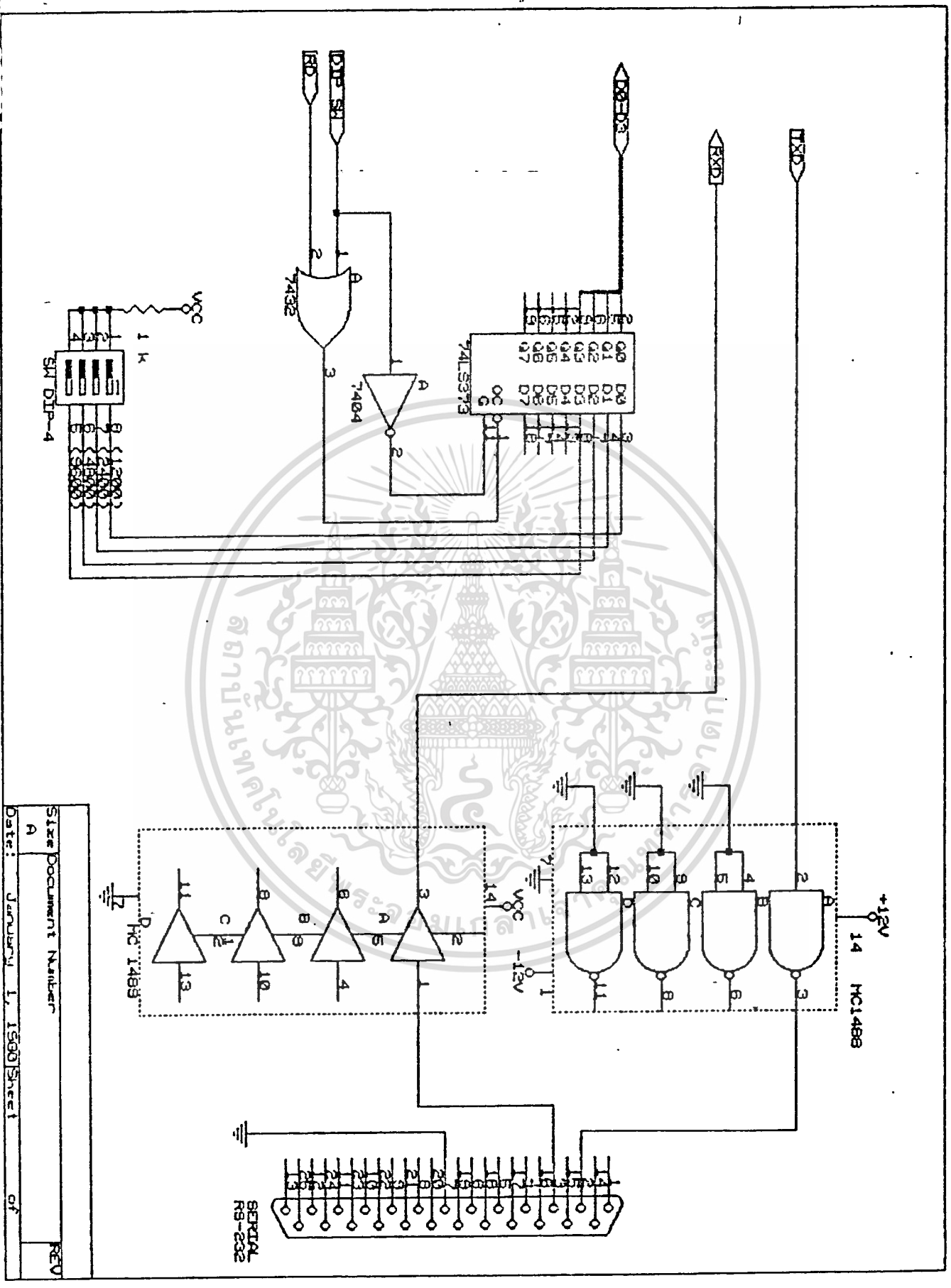
-การทำงานของ PID PROCESS ผิดจากที่ตั้งไว้

เมื่อมีข้อผิดพลาดในแต่ละส่วนก็จะมีการแสดงผลที่ led ในแต่ละส่วนแยกกันออกไป โดย address ของ led port ในแต่ละส่วนก็จะถูก decode จาก 74LS138 ตัวที่2แล้วทำการแสดงผลที่ portที่ถูกต้องต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

block การทำงานของส่วน serial และการกำหนด baud rate

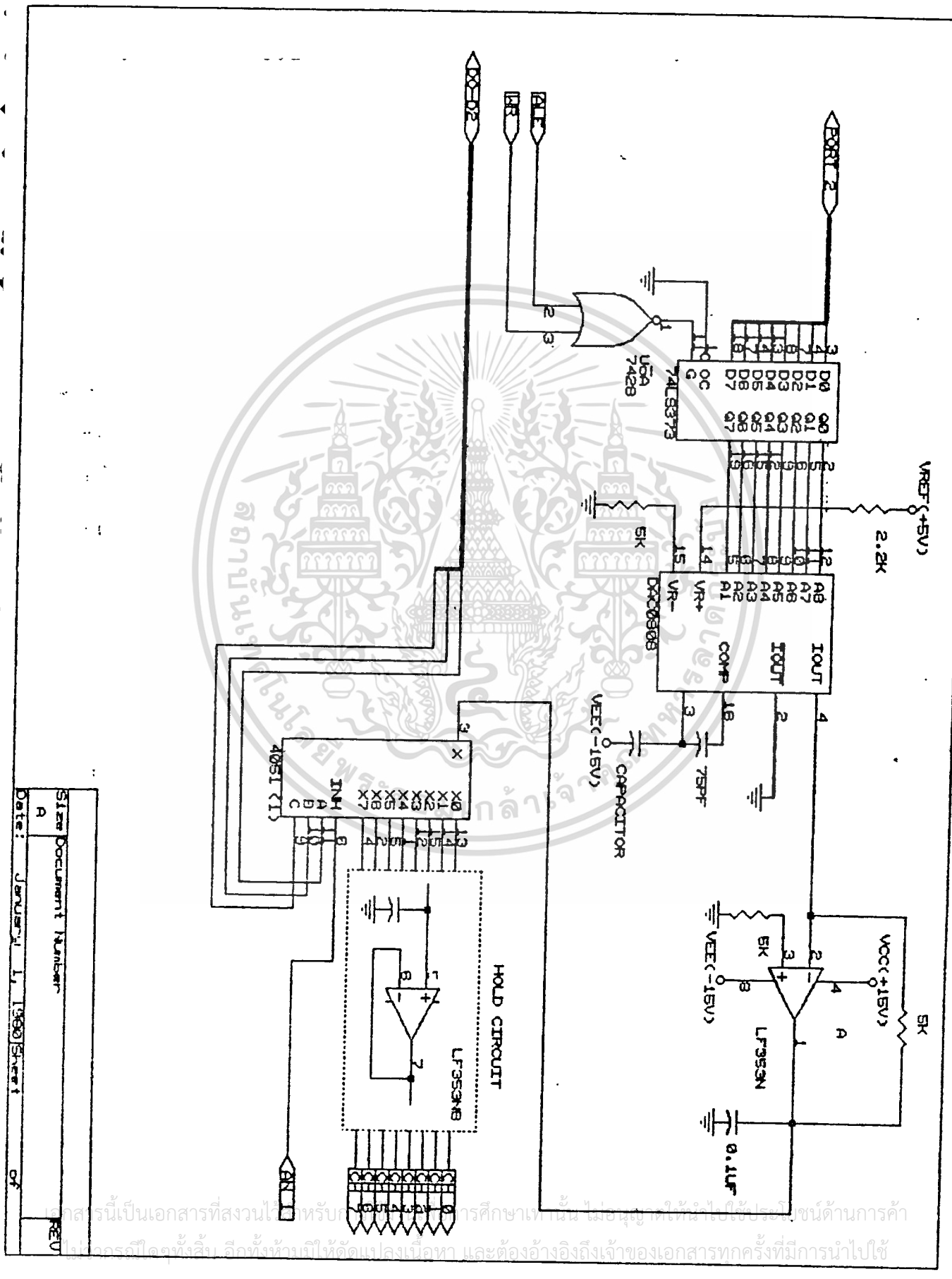
การทำงานในส่วนนี้เมื่อมีการ set ค่าที่ dip switch แล้วจะมีการส่งค่าที่แสดงถึงค่า baud rate ที่ต้องการไปเก็บที่ตัว 74LS373 ซึ่งเป็นตัว latch และเมื่อมีการอ่านค่า baud rate จากระบบก็จะมีการส่งสัญญาณ RD จาก cpu รวมถึงมีการส่ง address ของ dip switch ไปทำการ enable ให้กับ 74LS373 ให้มีการส่งค่า baud rate ให้กับ 8031

การทำงานในส่วนของ serial เมื่อมีการรับข้อมูลจาก RS-232 สัญญาณจะผ่าน MC1489 ซึ่งจะทำหน้าที่ในการเปลี่ยนระดับของสัญญาณจากระดับของ RS-232 ไปเป็นระดับของ TTL และเมื่อมีการส่งข้อมูลจากระบบไปให้กับ RS-232 ก็จะมีการส่งสัญญาณเข้าไปที่ MC1488 ซึ่งจะทำหน้าที่ในการแปลงระดับของสัญญาณ TTL ให้เป็นระดับของ RS-232

block การทำงานในส่วนของ input port

ในส่วนนี้จะมีการทำงานเมื่อมีการส่ง address ของ port i/p ซึ่งได้ทำการกำหนดไว้แล้วไปทำการถอดรหัสที่ 74138 (ดูที่ได้กล่าวไว้ในส่วนของการถอดรหัส port)ซึ่งจะมีผลให้มีการส่งสัญญาณไปทำการ enable ที่ 74154 รวมถึงเลือก card ที่ต้องการติดต่อด้วย ซึ่งในแต่ละ card นั้นจะมีการแบ่งออกไปเป็น channel ได้ 8 channel ซึ่งการเลือก channel ที่จะติดต่อด้วยก็ทำโดยการส่งสัญญาณ enable (เช่น ถ้าต้องการเรียก channel1 ก็ต้องส่งค่า 01h เป็นต้น)ในรูปของข้อมูลไบท์ให้กับทุกๆ card (แต่ card ที่จะมีการ enable ได้ต้องเป็น card ที่มีการส่ง address มาเรียก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK ของ OUTPUT PORT

block การทำงานของ o/p port จะมีการทำงานก็ต่อเมื่อมีการส่งการทำงานมาจาก host คือเมื่อ front end เริ่มการทำงานจะมีการทำ process ทั้งในส่วนของการเทียบค่า max,min รวมถึง PID PROCESS และจะทำการส่งผลของการทำงานไปแจ้งให้ host รับรู้ ต่อจากนั้น host จะทำการควบคุมการทำงานของระบบโดยส่งค่าต่างๆมาที่ port o/p โดยผ่านมาทาง front end(ก่อนที่จะมีการส่งข้อมูลออกมาที่ port o/p ต้องมีการส่งข้อมูลนี้ไปผ่าน D/A เพื่อให้เป็นสัญญาณ analog ก่อน) port o/pในระบบนี้จะต้องมีทั้งสิ้น 256 channel เช่นกันกับ port i/p port o/p ในที่นี้อาจจะเป็นพวก switch ควบคุมการปิดเปิดของระบบก็ได้ โดย port ที่ใช้เป็น port o/p ของระบบจะต่อออกมาจาก port 2ของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE ในส่วนของ FRONT END

software ในส่วนของ front end จะทำหน้าที่ในการควบคุมการทำงานและสภาวะในการทำงานของ front end ซึ่งจะแบ่งออกเป็นส่วนย่อยๆคือ

1)ในส่วนของ initialization จะทำหน้าที่ในการ initial ค่าในการทำงานของ front end ซึ่งต้องมีการระบุไว้ล่วงหน้าก่อนที่จะเริ่มการทำงาน เช่น การระบุค่าแหน่งของ port i/p,port o/p,serial port เป็นต้นรวมถึงต้องมีการกำหนดเนื้อที่ในหน่วยความจำที่จะใช้เก็บข้อมูลต่างๆ ซึ่งในส่วนนี้จะประกอบด้วยส่วนต่างๆดังนี้คือ

1.1)serial i/o port initialization ส่วนนี้จะเป็นการกำหนดรูปแบบของข้อมูลที่จะทำการส่งระหว่าง front end กับ host เช่น จำนวนบิตของข้อมูลที่จะส่งออกในแต่ละครั้ง,baud rate ที่จะทำการส่ง เป็นต้นโดยรูปแบบของการติดต่อต้องเป็นไปตามข้อกำหนดของตัว cpu ที่ใช้รวมถึงข้อกำหนดของ serial port มาตรฐานอีกด้วย

การควบคุม mode ในการทำงานใน8031นี้จะมี mode ของการติดต่อ serial 4 mode คือ mode 0,1,2,3 ซึ่งในแต่ละ mode ก็จะมีรายละเอียดต่างๆกันออกไป ซึ่งจะกล่าวถึงรายละเอียดในส่วนของตัว8031 ส่วนของ mode ที่มีการกล่าวถึงในที่นี้จะเป็น mode ที่ระบบนี้มีการใช้งานคือ mode 1 ซึ่งจะมีการส่งข้อมูลทีละ 10 bit โดยมีการส่ง 1 start bit (´0´),8 data bit ,1 stop bit (´1´) รวมถึงใน mode การทำงานนี้จะมีการตั้ง baud rate ได้ด้วย ส่วนการตั้ง baud rate ในระบบนี้จะมีการกำหนดจากตัว dip_switch ซึ่งจะมีทั้งหมด 4 ตัวแล้วอ่านค่าเหล่านี้ขึ้นมาก็จะรู้ถึงค่า buad rate ที่ต้องการ โดยรายละเอียดของส่วนของ dip_switch มีการกล่าวถึงรายละเอียดในส่วนของ hardware แล้วส่วนการทำงานของระบบตาม baud rate ที่ตั้งไว้จะถูกกำหนดโดยการกำหนดค่าใน register ของ8031ดังต่อไปนี้คือ

-การกำหนดค่าใน scon(serial control registor)ซึ่งจะทำหน้าที่ในการกำหนด mode ในการทำงานโดยในแต่ละบิตจะมีความหมายดังที่ได้กล่าวไว้ในส่วนของของ8031 แต่ในส่วนที่จะกล่าวถึงนี้จะหมายถึงการกำหนดค่าของ scon ในระบบนี้

RI = 0 เป็นการ clear received interrupt flag

TI = 1 เป็นการ set transmit interrupt flag

TB8 = RB8 = 0 เป็นการ clear ของ 2 bit นี้เนื่องจาก 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bit นี้ไม่เกี่ยวข้องกับการทำงานใน mode 1

REN = 1 เป็นการ set enable serial reception ถ้าไม่มีการ clear bit นี้จะทำให้ไม่สามารถรับข้อมูลทาง serial ได้

SM2 = 0 เป็นการ clear bit เนื่องจากในการทำงานของ mode 1 ถ้า SM2 = 1จะทำให้ RI ถูก disable ทำให้ไม่สามารถรับ stop bit ได้

SM1 = 1 SM0 = 0 เป็นการเลือก mode 1

-การกำหนดค่าใน tmod(timer/counter mode control) เป็นการเลือก mode ในการทำงานของ timer ซึ่งใน8031มี timer/counter 2 ตัวคือ timer0,timer1 ซึ่งในระบบจะใช้ timer1 เท่านั้น(การที่เป็นเช่นนี้เนื่องจากข้อจำกัดจากการกำหนด baud rate ในระบบนี้)

-baud rate = 9600 มี clock ในระบบ = 12 Mhz ต้องใช้ smod = 0 และใช้ timer1 ใน mode2 ซึ่งมีค่า auto reload = FDH

-baud rate = 4800 มี clock ในระบบ = 12 Mhz ต้องใช้ smod = 0 และใช้ timer1 ใน mode2 ซึ่งมีค่า auto reload = FAH

-baud rate = 2400 ต่างจากใน baud rate ข้างต้นตรงที่ค่า auto reload = f4h

-baud rate = 1200 ต่างจากใน baud rate ข้างต้นตรงที่ค่า auto reload = e8h

เนื่องจาก RS-232 จะคิดตั้งได้โดยใช้ baud rate สูงสุด = 9600 จึงเป็นการกำหนด baud rate ของระบบได้สูงสุดได้แค่ 9600(แต่ใน8031 สามารถกำหนดค่า baud rate ได้สูงกว่านี้) การทำงานของ auto reload คือต้องมีการ load ค่า reload value ไปเก็บไว้ที่ TH1 ซึ่งต้องมีการกำหนดค่าไว้ใน TMOD ดังนี้คือใน 4 bit แรก(bit0-bit3)จะเป็นการกำหนดของ timer0 จึงไม่ต้องมีการกำหนดค่าไว้ให้เป็น 0 ได้เลย ส่วนของ bit4 - bit7 จะเป็นของ timer1 จึงต้องมีการกำหนดค่าไว้ในที่นี้จะกำหนดค่า TMOD = 20H เป็นการเลือกค่า M1,M0 = 1,0 ตามลำดับเพื่อเป็นการเลือก mode2 ส่วน bit c/t = 0 เพราะเป็นการเลือกให้ทำงานเป็น timer และกำหนดให้ขา gate เป็น 0 (รายละเอียดในส่วนของการกำหนดค่าเหล่านี้จะกำหนดไว้ใน8031)

-การกำหนดค่าใน TCON(timer/counter control registor) จะเป็นการกำหนดขอบของสัญญาณในการทริก interrupt เป็นต้น ซึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบกำหนดให้ TCON = 10H(00010000)ใน 4 bit แรกจะเป็น 0 หมด เพราะไม่มีส่วนเกี่ยวข้องกับการทำงานในระบบที่ทำการกำหนดไว้

-การกำหนดค่าใน pcon(power control registor)เป็นการกำหนดค่า SMOD(serial mode control registor)เพื่อให้การทำงานเป็นไปตามการกำหนดค่า BAUD RATE ซึ่งการกำหนดค่า PCON นี้เราใช้ค่า 00h เพื่อเลือก SMOD = 0 ส่วนค่าต่างๆที่ไม่จำเป็นก็ไม่ต้องทำการกำหนดไว้

-ต้องมีการกำหนดค่าไว้ใน TL1 เป็น 00h ก่อนเพื่อให้มีการ reload ค่าใน TH1

1.2)data area initialization เป็นการกำหนดค่าตำแหน่งในการเก็บข้อมูลซึ่งข้อมูลในที่นี้มีหลายพวกคือ

-ตำแหน่งในหน่วยความจำที่ใช้ในการเก็บ monitor program

-ตำแหน่งที่ใช้ในการเก็บ unprocessed data (ข้อมูลที่ i/p port อ่านเข้ามาผ่าน A/D)ซึ่งข้อมูลในส่วนนี้จะถูกส่งผ่าน serial port ไปที่ host เพื่อแสดงผล รวมถึงส่งข้อมูลนี้ไปเข้า PID PROCESS, การเทียบค่า max,min

-ส่วนที่ใช้ในการเก็บ processed data (คือข้อมูลที่มีการผ่าน process บนตัว front end แล้วรวมทั้งยังต้องมีการส่งข้อมูลนี้ไปแสดงผลที่ host แล้วนำเอาข้อมูลไปทำการควบคุมระบบต่อไป

-ตำแหน่งที่ใช้ในการเก็บ port i/p,o/p ต้องมีการกำหนดตำแหน่งไว้ล่วงหน้าก่อนที่จะมีการใช้งานเพื่อให้มีการติดต่อได้ถูกต้อง

-ส่วนที่ใช้ในการเก็บข้อมูลจาก host ซึ่งจะแบ่งได้เป็นส่วนต่างๆดังนี้คือ

1)channel ที่ต้องมีการควบคุมในแต่ละ card

2)max temp, min temp,set point temp จะต้องมีการเก็บค่าเหล่านี้ไว้ในส่วนของ ram เพื่อนำไปใช้ใน process ของการเทียบค่า max,min รวมถึงต้องเอาไปใช้ใน PID process ด้วย

3)ค่าคงที่ต้องมีการใช้ใน PID process เช่น ค่า KD เป็นต้น

-ส่วนที่ใช้ในการเก็บ program PID process(ส่วนของ PID process นี้จะเป็นลักษณะของการแปลงเป็น execute file แล้วทำการส่ง file เหล่านี้จาก host โดยส่งมาทาง serial port มาเก็บไว้ที่ ram แล้วทำการ process ที่ตัว front end) การที่ไม่มีการเขียน program PID control ไว้ที่ส่วนของ monitor program ของ front end เลยเพราะจะสิ้นเปลืองหน่วยความจำรวมถึงไม่สะดวกในการแก้ไขหรือการขยายในอนาคต

-การกำหนดตำแหน่งในส่วนของ alarm port ที่ใช้ในการแสดงข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทำงานของระบบ

2)SELFTEST ในส่วนประกอบของระบบ เช่น internal register, ram,rom,serial port,ส่วนที่เป็น port ในการใช้งาน ซึ่งส่วนนี้จะเป็นการตรวจสอบความพร้อมในการทำงานของระบบทั้งหมดและวิธีที่ใช้ในการตรวจสอบก็ต่างกันออกไปตามแต่ละส่วนคือ

-ส่วนของ rom ก็จะใช้วิธี checksum คือเป็นการตรวจสอบผลรวมของค่าข้อมูลที่มีการบันทึกไว้ว่าถูกต้องหรือไม่

-ส่วนของ ram ใช้วิธี read/write stroage คือการเขียนข้อมูลเข้าไปแล้วทำการอ่านข้อมูลที่เขียนเข้าไปออกมาตรวจดูว่าเหมือนเดิมหรือไม่ ถ้าข้อมูลมีการเปลี่ยนไปแสดงถึงข้อผิดพลาดก็ต้องส่งผลผิดพลาดนี้ไปแสดงที่ port รวมทั้งแจ้งให้ host รู้ด้วย

-selftest ในส่วนของ i/p port ส่วนนี้จะต้องมีการตรวจสอบ card ที่มีการต่อไว้ว่ามี card ใดบ้าง เพื่อทำการส่งค่า card ที่ต่อไว้ไปให้กับ host เพื่อทำการตั้ง channel ได้ถูกต้อง

-selftest ในส่วนของ serial port ส่วนนี้ต้องมีการตรวจสอบความพร้อมในการติดต่อระหว่าง host กับ front end ซึ่งทำโดยการส่ง request signal ระหว่าง host กับ front end แล้วทำการตรวจสอบสัญญาณเหล่านี้

-selftest ในส่วนของ o/p port จะใช้การตรวจสอบลักษณะเดียวกับการตรวจสอบ i/p port คือมีการตรวจสอบว่ามี card ใดทำการต่ออยู่ด้วยแล้วทำการควบคุมให้ถูกต้อง

-selftest ในส่วนของ display ซึ่งในส่วนของการ display นี้จะแบ่งออกจะแบ่งออกเป็นส่วนย่อยๆดังนี้คือ

1)ส่วนของการ alarm โดยการใช้ buzzle ซึ่งเมื่อมีการส่งข้อผิดพลาดมาจากตัว front end ก็จะทำการแสดงผลในส่วนนี้ ซึ่งในส่วนนี้จะตรวจสอบโดยการส่ง failed code มาที่ alarm port แล้วดูการทำงานของ port นี้

2)ส่วนของ led port คือส่วนที่จะทำการแสดงผลในกรณีที่มีการส่งข้อผิดพลาดมาแจ้งที่ port ซึ่งส่วนนี้จะทำการตรวจโดยการส่ง failed code แล้วดูผลที่ port นี้

3)ส่วนของ i/p ของ front end ส่วนนี้จะทำหน้าที่ในการควบคุมข้อมูลทั้งหมดที่ทำการรับเข้ามาที่ front end จะแบ่งออกเป็นส่วนย่อยๆดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ส่วนของ i/p ที่รับมาจาก host ซึ่งแบ่งออกเป็นส่วนต่างๆดังนี้คือ

1)process selection คือค่าต่างๆที่ใช้ในการทำงานของระบบซึ่งจะแบ่งออกดังนี้คือ

ก)input selection channel คือค่า channel ในแต่ละ card ที่ต้องการทำการควบคุมจะต้องถูกส่งเข้ามาที่ตัว front end เพื่อทำการควบคุมได้ถูกต้อง

ข)temperature selection คือค่าของอุณหภูมิที่จะทำการควบคุมในแต่ละ channel ซึ่งจะประกอบด้วยค่าต่างๆดังนี้คือ max,min,setpoint temp โดย front end จะนำค่าต่างๆดังนี้ไปเข้า process โดยค่า max, min เป็นการบอกถึงขีดจำกัดสูงสุดและต่ำสุดของการควบคุมที่สามารถทำได้ และเมื่อมีการเกินค่าเหล่านี้จะต้องมีการแจ้งที่ port ที่แสดงผลผิดพลาด

ค)ค่าคงที่ต่างๆที่จะใช้ใน PID process เป็นค่าที่ต้องรับมาจาก host เพื่อนำมาเข้า process

ง)file execute ของ PID process ซึ่ง host ต้องทำการส่งไปที่ตัว front end เพื่อให้ front end ทำการควบคุม

2)ข้อมูลที่ host ส่งมาให้กับ front end เพื่อทำการควบคุมการทำงานของ o/p port ของระบบเช่นการไปปิดเปิด switch relay เป็นต้น

-ข้อมูลจากส่วนของ i/p port เป็นการอ่านข้อมูลที่ผ่านจาก ADC ทาง port 1 ซึ่งเป็นการอ่านข้อมูลแบบ polling (คือจะมีการอ่านข้อมูลในทุก channel ที่ทำการเลือกไว้โดยการอ่าน channel จะเป็นไปตามลำดับจาก channel แรกสุดไปจนถึง channel สุดท้ายแล้วทำการอ่านจาก channel แรกสุดใหม่)

4)ส่วนของ process เป็นส่วนที่จะทำการควบคุมระบบให้ถูกต้องซึ่ง process ที่จะเกิดบน front end นั้นจะประกอบด้วย 2 ส่วนคือ

4.1)process ที่จะทำหน้าที่ในการเทียบค่าข้อมูล i/p กับค่าของ max,min

4.2)process ของ PID process ซึ่งจะถูกส่งมาจาก host เมื่อมีข้อผิดพลาดก็จะส่งผลไปที่ failed port รวมทั้งยังส่งข้อมูลไปแสดงผลที่ host ด้วย

5)ส่วนของ o/p คือส่วนที่ front end จะทำการส่งข้อมูลออกไปยังส่วนต่างๆของระบบซึ่งแบ่งออกเป็นส่วนต่างๆดังนี้คือ

5.1)การส่งข้อมูลจาก front end ไปที่ host ทาง serial port ซึ่งในส่วนนี้จะแบ่งออกเป็นส่วนต่างๆดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ส่งข้อมูลที่ได้รับจาก i/p port ไปที่ host เพื่อให้แสดงกราฟของระบบขณะที่ไม่มีการควบคุม

-การส่งข้อมูลที่ได้จากการทำงานของ process ไปที่ host เพื่อทำการแสดงของระบบขณะที่ทำการควบคุม

-การส่ง alarm signal ไปแจ้งการทำงานของระบบในขณะที่มีข้อผิดพลาดที่ failed port

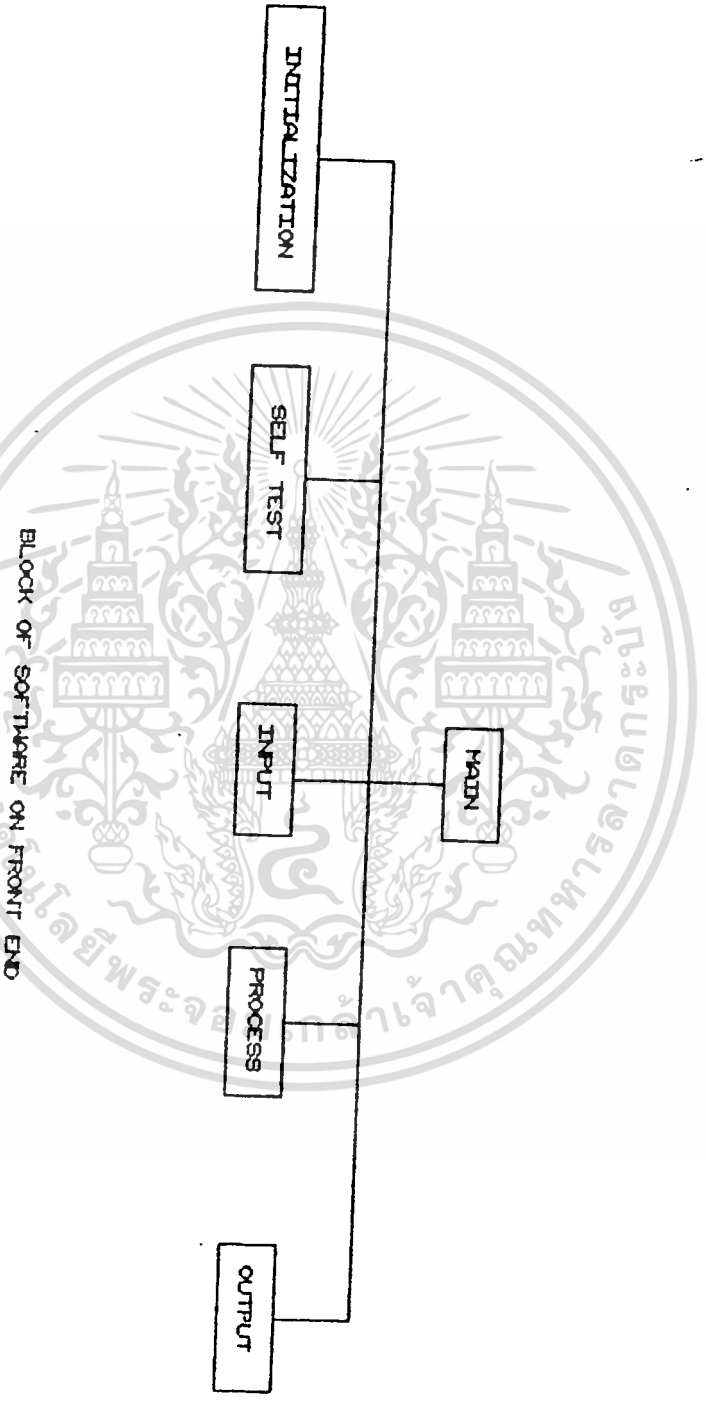
-selftest signal failed คือข้อมูลที่แสดงข้อผิดพลาดในการ selftest ของ front end

5.2)การส่งข้อมูลจาก front end ไปที่ o/p port เป็นการส่งข้อมูลที่ host ไปที่ front end แล้วให้ front end ทำการส่งต่อไปที่ o/p port เการควบคุมซึ่ง o/p port ในที่นี้จะแบ่งออกเป็น 2 พวกคือ

-การส่งข้อมูลในส่วนของ digital คือส่วนของ port ที่แสดงถึงการ failed ในส่วนนี้จะต้องส่งข้อมูลไปที่ port นี้ในลักษณะของ digital

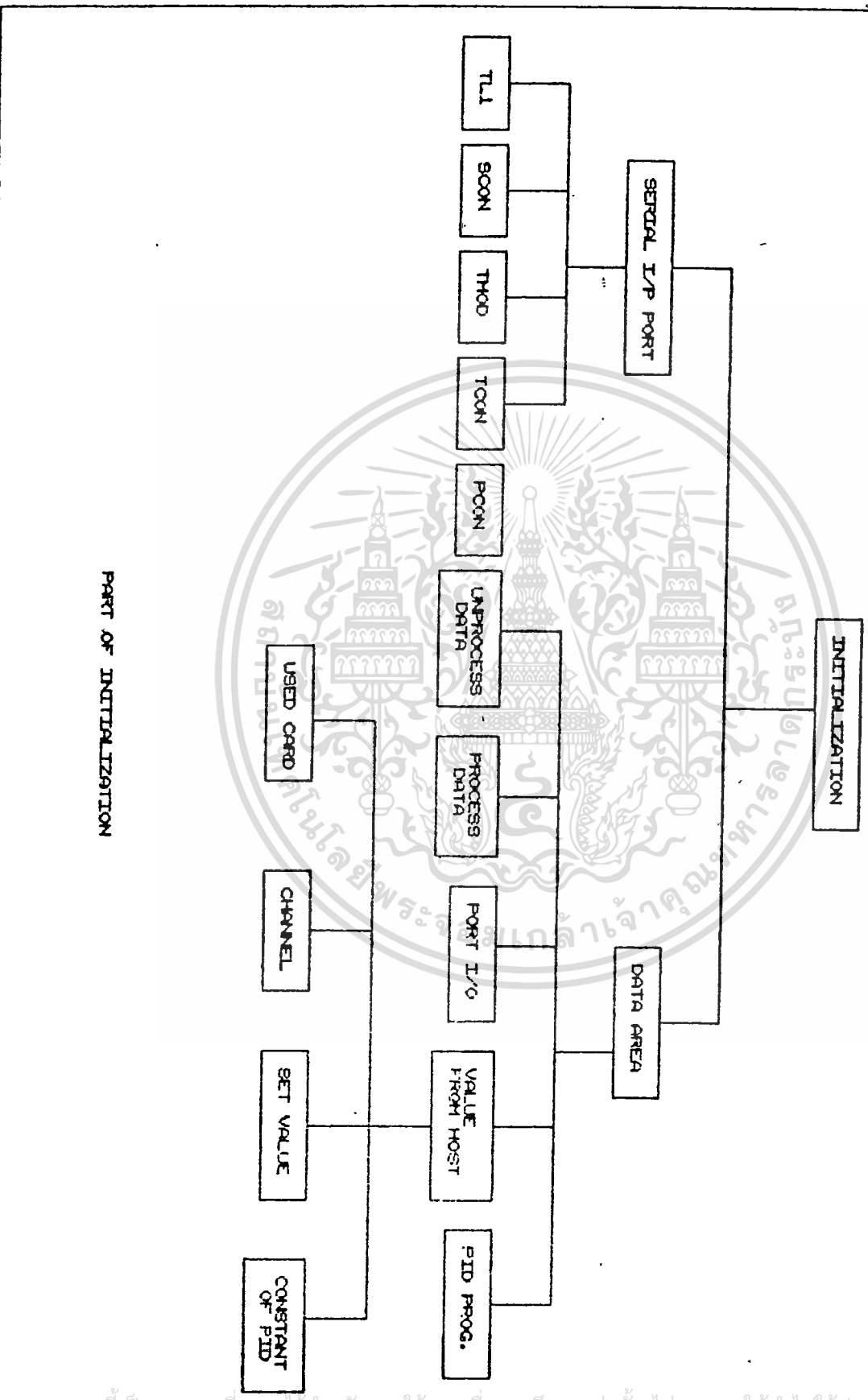
-การส่งข้อมูลในส่วนของ analog คือส่วนของ port ที่ต้องทำการส่งข้อมูลที่ analog ไปที่ port ข้อมูลพวกนี้จะต้องมีการส่งไปผ่านที่ DAC ก่อนที่จะทำกาประเภทของข้อมูลที่ใช้การส่งแบบนี้ก็จะเป็นการสั่งงานจาก host เพื่อให้ระบบเไปตามที่ host ต้องการ เช่นการปิดเปิด relay เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



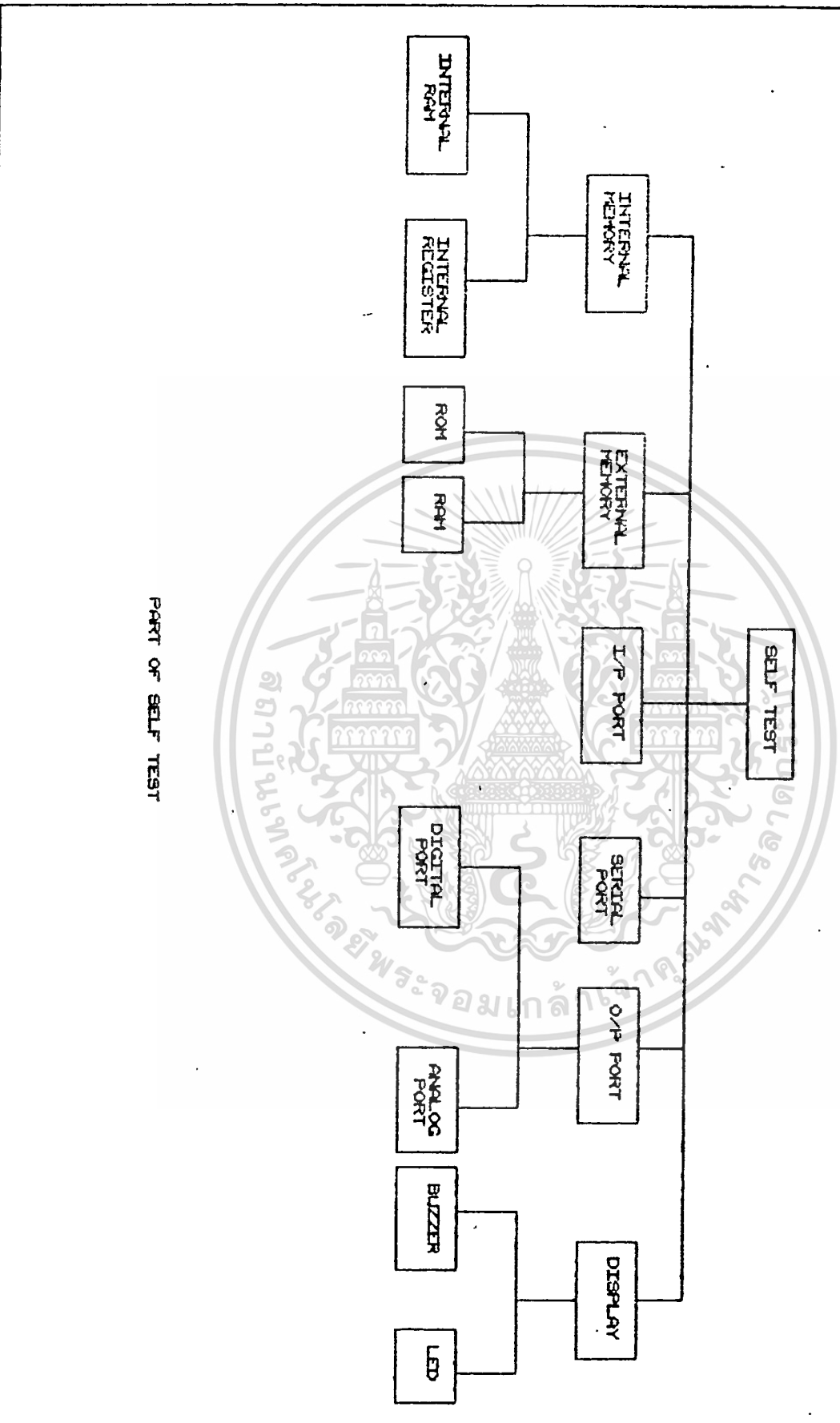
BLOCK OF SOFTWARE ON FRONT END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



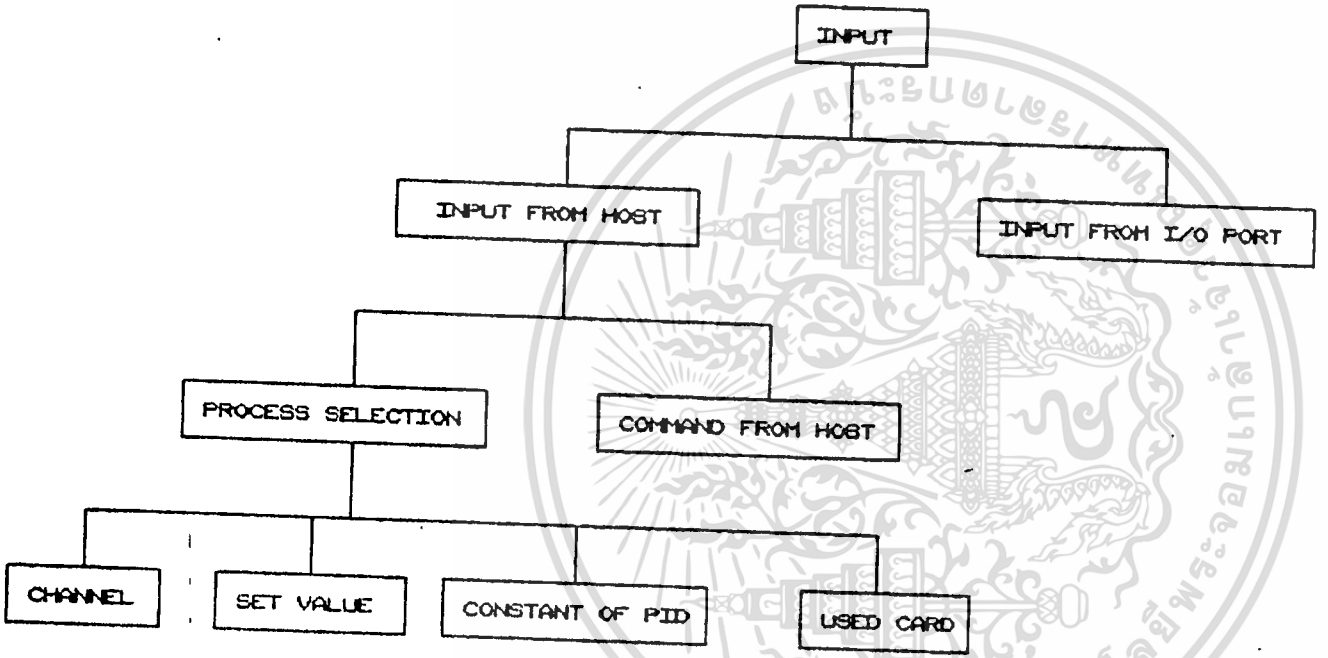
PART OF INITIALIZATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



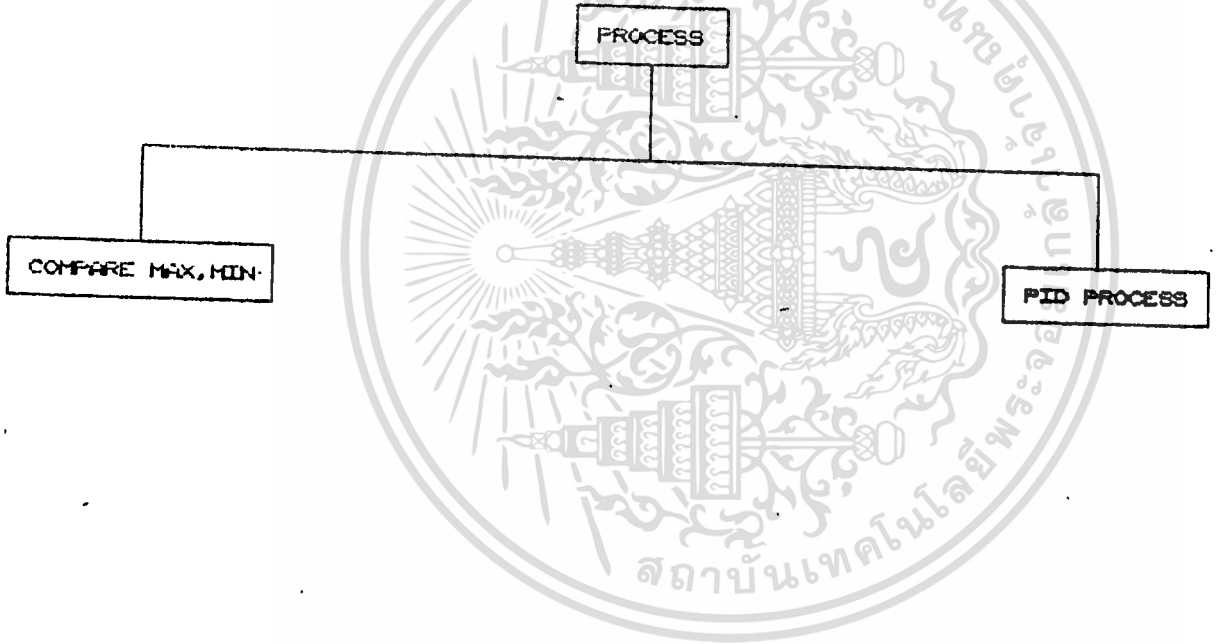
PART OF SELF TEST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



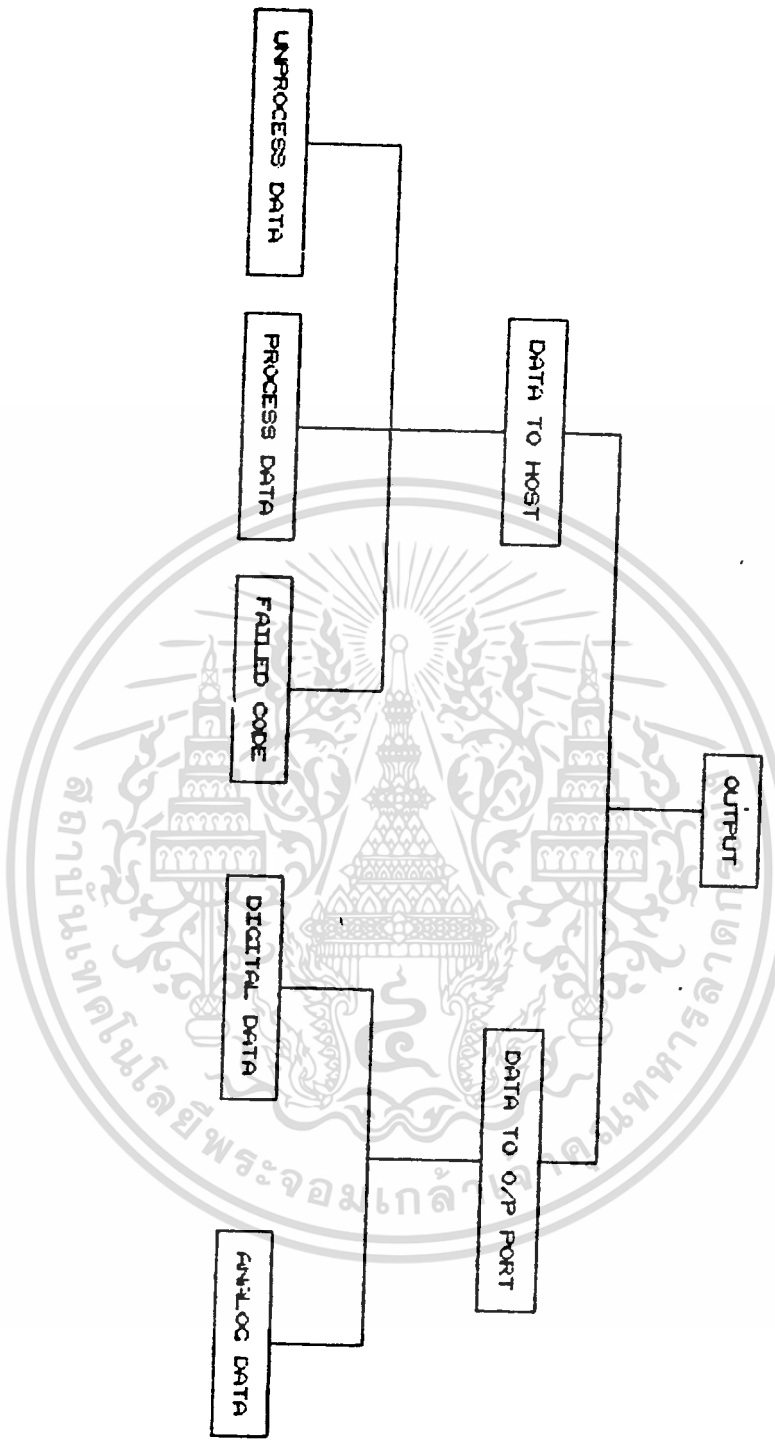
PART OF INPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PART OF PROCESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การแสวงหากำไร หรือการโฆษณาประชาสัมพันธ์ รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PART OF OUTPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดในส่วนของแผนภูมิของ SOFTWARE ในส่วนของ FRONT END :

ในส่วนนี้จะเป็นรายละเอียดของการทำงานในระบบว่ามีขั้นตอนใดบ้างซึ่งจะเริ่มจากการกำหนดความเร็วของการส่งข้อมูลระหว่าง host กับ front end ต่อจากนั้นจะตรวจสอบ card i/p ที่ทำการต่ออยู่กับระบบแล้วทำการส่งไปแจ้งให้กับ host รับรู้ แล้วทำการรับ channel ที่ถูกเลือกจาก host รวมถึงรับค่าคงที่ต่างๆเพื่อทำการควบคุม และเมื่อ front end ได้รับ code ให้เริ่มทำงานก็จะเริ่มส่งค่า card i/p และ channel ที่จะทำการควบคุมไปเรียกให้ i/p port ส่งข้อมูลมาให้กับ front end เพื่อให้ front end นำข้อมูลที่ได้เข้า process ต่อจากนั้นจะทำการส่งข้อมูลทั้งที่เข้าสู่ process และยังไม่ได้เข้า process ไปที่ host และจะทำการวนรับข้อมูลไปเรื่อยๆ

แผนภูมิของการ INITIALIZATION SERIAL PORT

ในส่วนนี้จะเริ่มจากการอ่านค่าใน dip switch ซึ่งจะเป็นการกำหนดค่า baud rate รวมถึงการกำหนดค่าที่เกี่ยวกับการทำงานใน serial mode ซึ่งจากการกำหนดค่า baud rate ก็จะทำการกำหนดค่าใน auto reload ได้ถูกต้องซึ่งถ้ากำหนด baud rate = 9600 ก็ต้องใช้ค่า auto reload = fdh เป็นต้น เมื่อกำหนดค่าใน auto reload เรียบร้อยแล้วก็จบส่วนของการ initialization serial port

แผนภูมิในส่วนของการรับ CHANNEL ที่จะควบคุมจาก HOST

ก่อนที่จะมีการรับค่า channel จาก host ต้องมีการรับค่าของ card ที่ทำการติดต่อในขณะนี้โดยจะรับค่า card i/p 4 byte แต่ละ byte แสดงถึงสถานะของ card ที่ทำการติดต่อด้วย 8 card ต่อจากนั้นจึงจะมีการรับ channel จาก host โดยเริ่มจากการรับ request signal จาก host (host จะส่ง ? มาให้ front end) ต่อจากนั้น front end จะส่ง acknowledge ไปให้ host(ส่ง . ไปให้กับ host)เมื่อ host ได้รับการตอบรับแล้วก็จะทำการส่ง channel ที่ต้องการควบคุมโดย channel ชุดแรกที่ส่งออกมาจะเป็นของ card i/p การ์ดแรกที่มีการต่อไว้ซึ่งจะตรวจสอบได้จากค่าที่ host มีการส่งมาให้ดังกล่าวแล้วข้างต้น ก็จะรู้ได้ว่า channel ที่ส่งออกมานั้นเป็นของ card i/p ใด ต่อจากนั้นจะทำการส่งค่า channel ที่ทำการรับมานี้ไปเก็บที่ internal ram เพื่อนำมาใช้ในการตรวจรับค่า temperature จาก host ต่อไปรวมถึงนำไปใช้ในการรับค่า i/p จาก i/p port ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

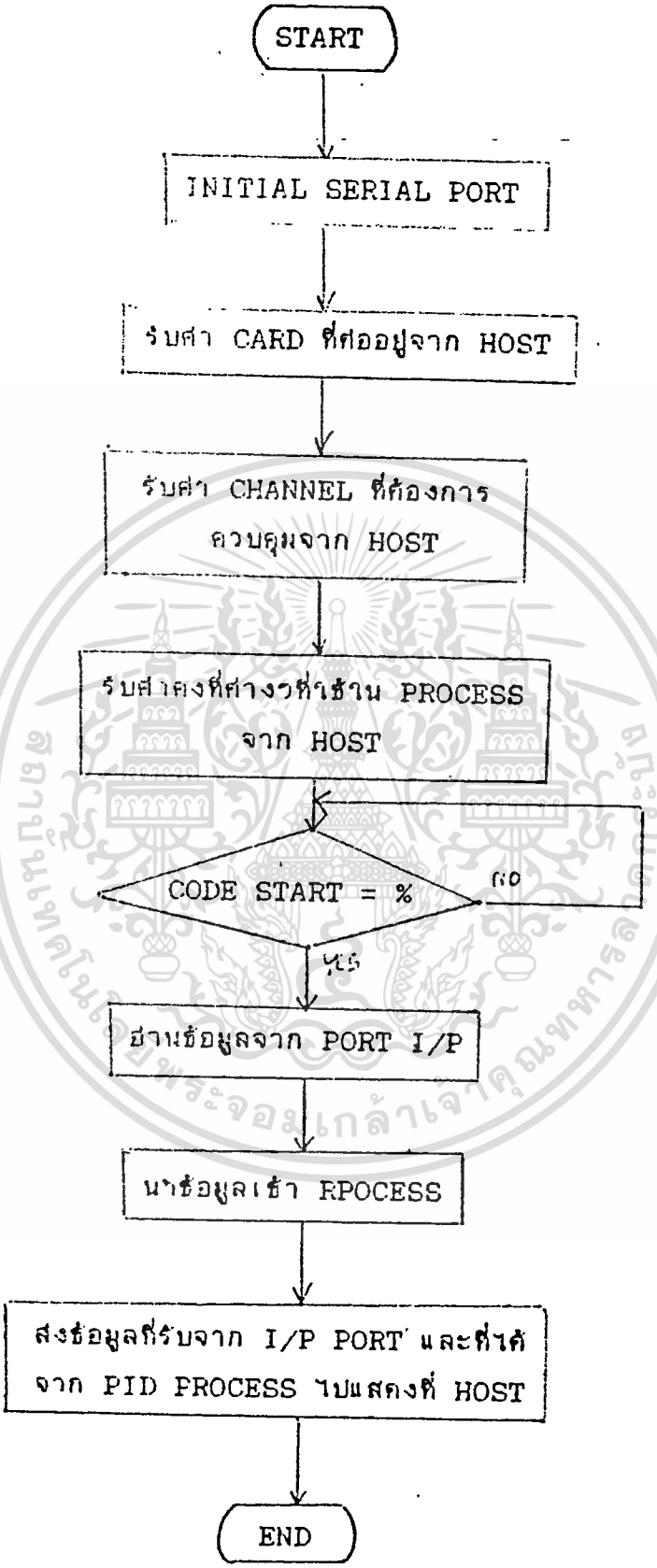
แผนภูมิของการรับค่าอุณหภูมิจาก HOST

การทำงานในส่วนนี้เริ่มจากการอ่านค่าจาก internal ram ซึ่งเก็บค่า channel ที่ต้องการควบคุมว่าต้องการควบคุม channel ใดบ้าง แล้วทำการรับค่าอุณหภูมิจาก host โดย host จะต้องส่ง request มาแล้ว front end ทำการส่ง acknowledge ไปตอบรับการทำงานของ host ต่อจากนั้น host จะทำการส่งค่าอุณหภูมิโดยต้องส่งค่า max temp,min temp, set point มาตามลำดับโดยจะส่งมาทีละ 1 byte มื่อ front end ได้รับแล้วจะส่งการตอบรับไปให้กับ host แล้ว front end จะนำค่าไปเก็บไว้ที่หน่วยความจำ เพื่อนำไปเข้า process ต่อไปเมื่อมีการรับค่าเหล่านี้แล้วก็จะจบขั้นตอนในส่วนนี้

แผนภูมิของ process ที่เกิดบน front end

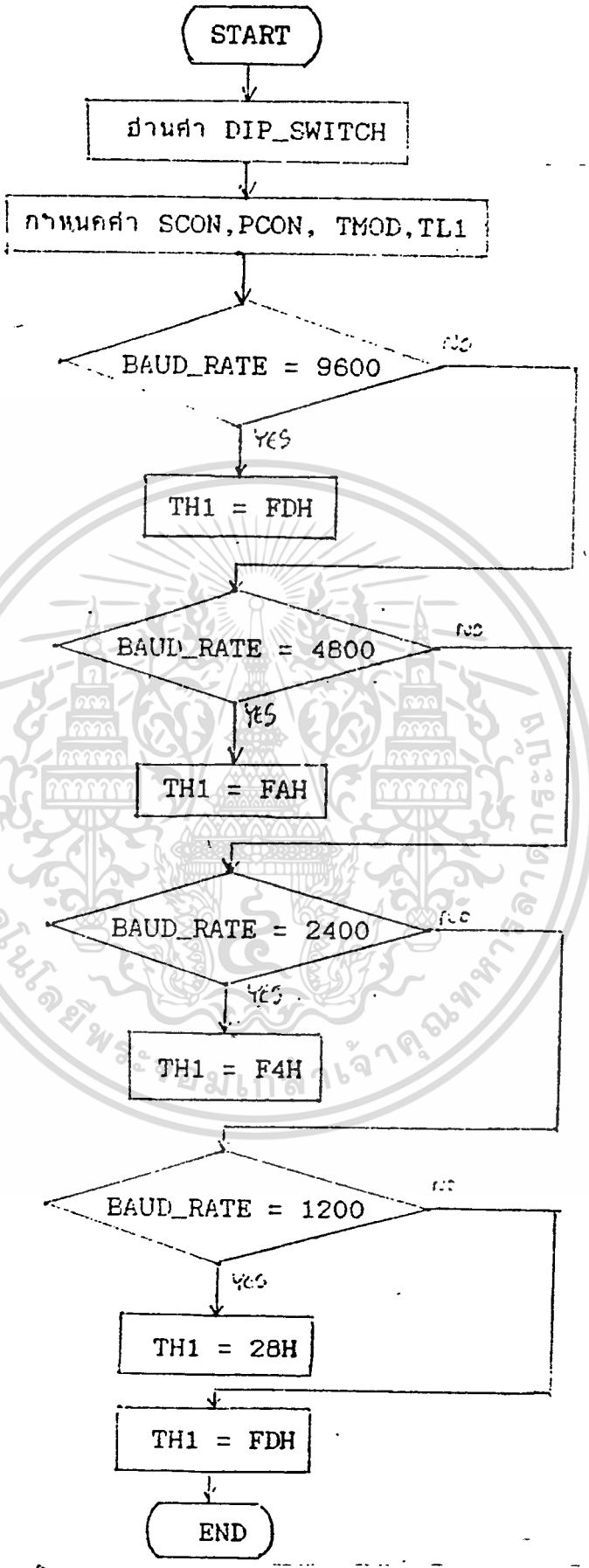
เมื่อมีการเก็บข้อมูลจาก i/p port เรียบร้อยแล้วจะมีการนำค่า max,min temp มาทำการเปรียบเทียบว่าค่าที่อ่านได้นั้นเกินค่าที่กำหนดไว้หรือไม่ ต่อจากนั้นจะนำข้อมูลไปเข้า pid process แล้วเก็บค่าผลลัพธ์ที่ได้ไว้ที่ ram ในส่วนที่กำหนดไว้เพื่อทำการส่งออกไป เมื่อมีการรับข้อมูลของทุกๆ channel ที่ต้องการควบคุมรวมถึงมีการ process ข้อมูลในส่วนนั้นเรียบร้อยแล้วจะนำเอาค่าของ i/p data และค่าของ o/p data ทั้งหมดที่เก็บไว้ทำการส่งออกไปให้กับ host แสดงผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



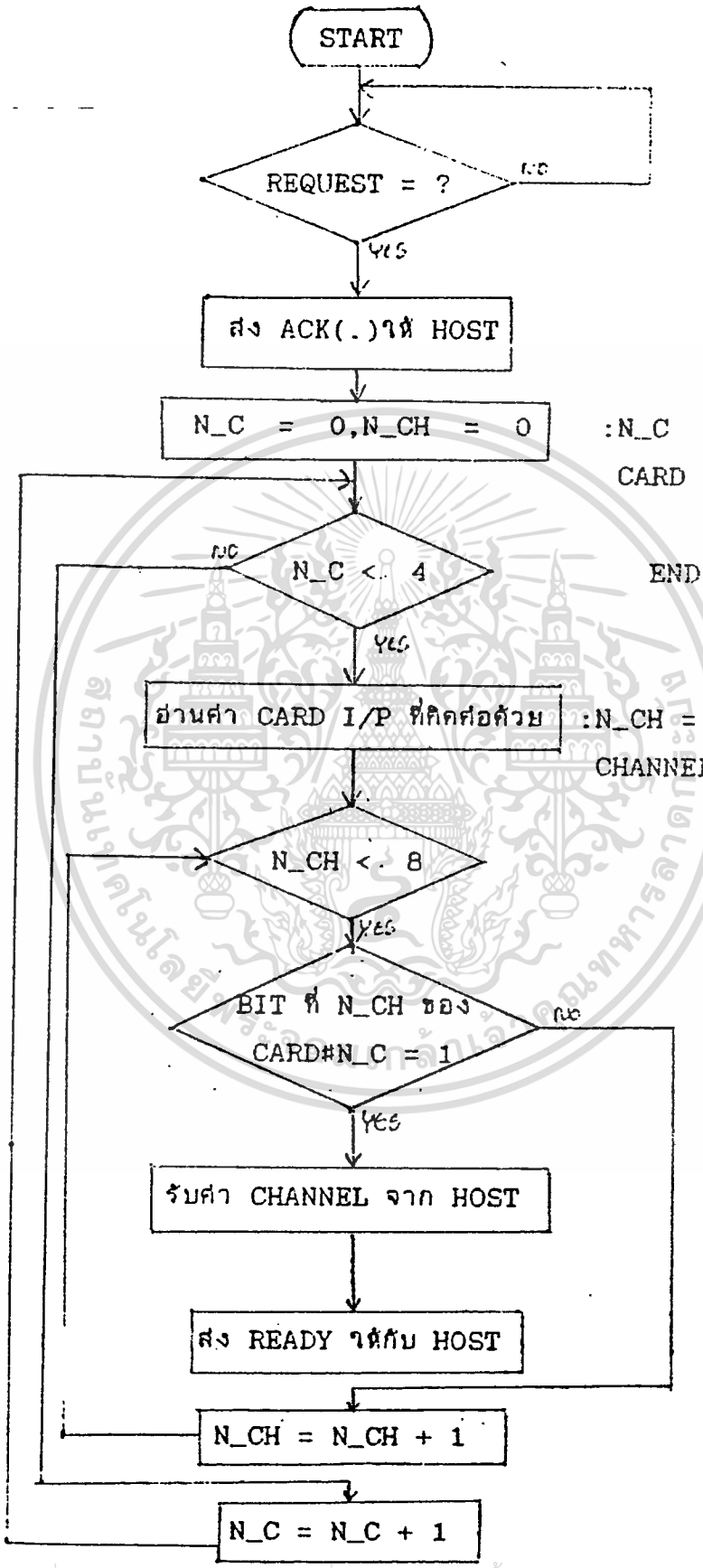
แผนภูมิของการทำงานของ FRONT END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



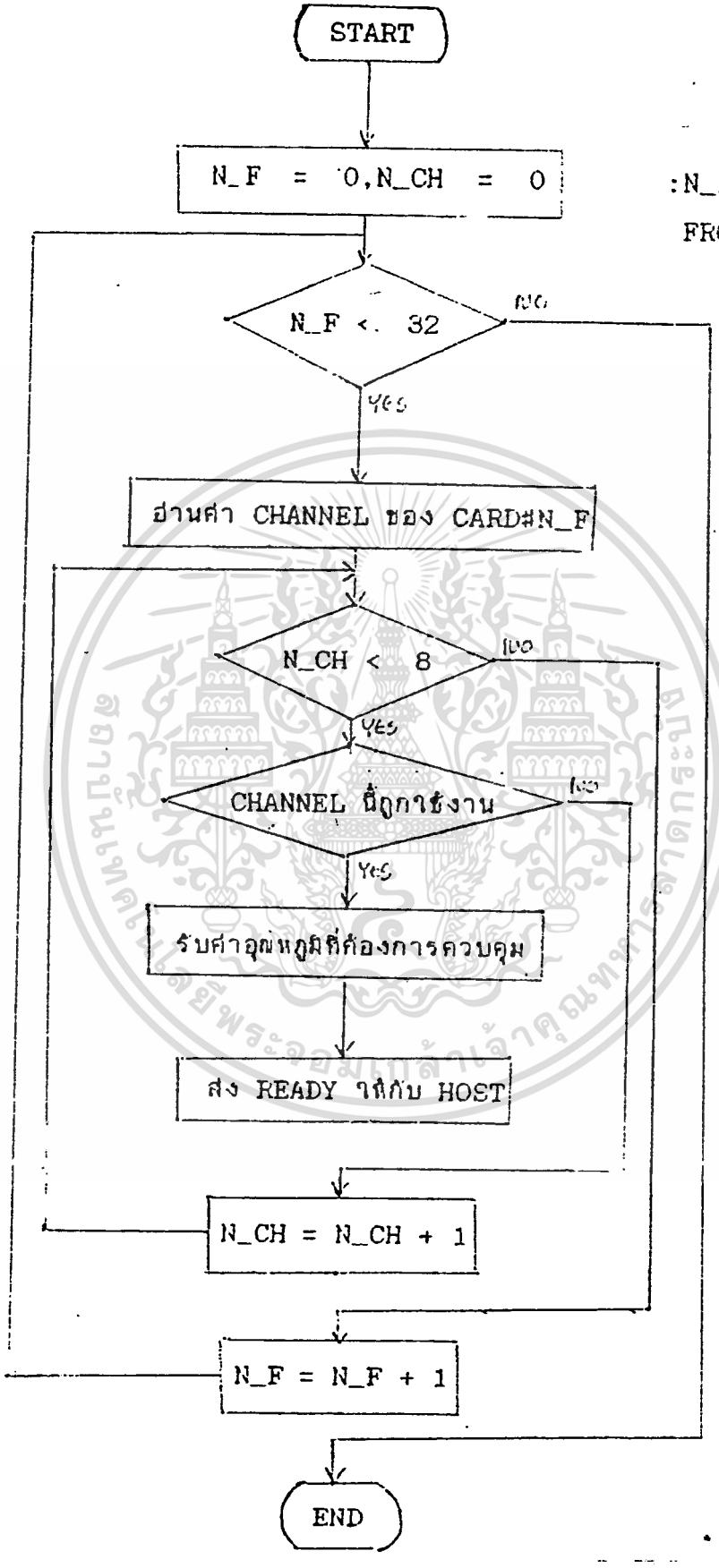
แผนผังของการ INITIAL SERIAL PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



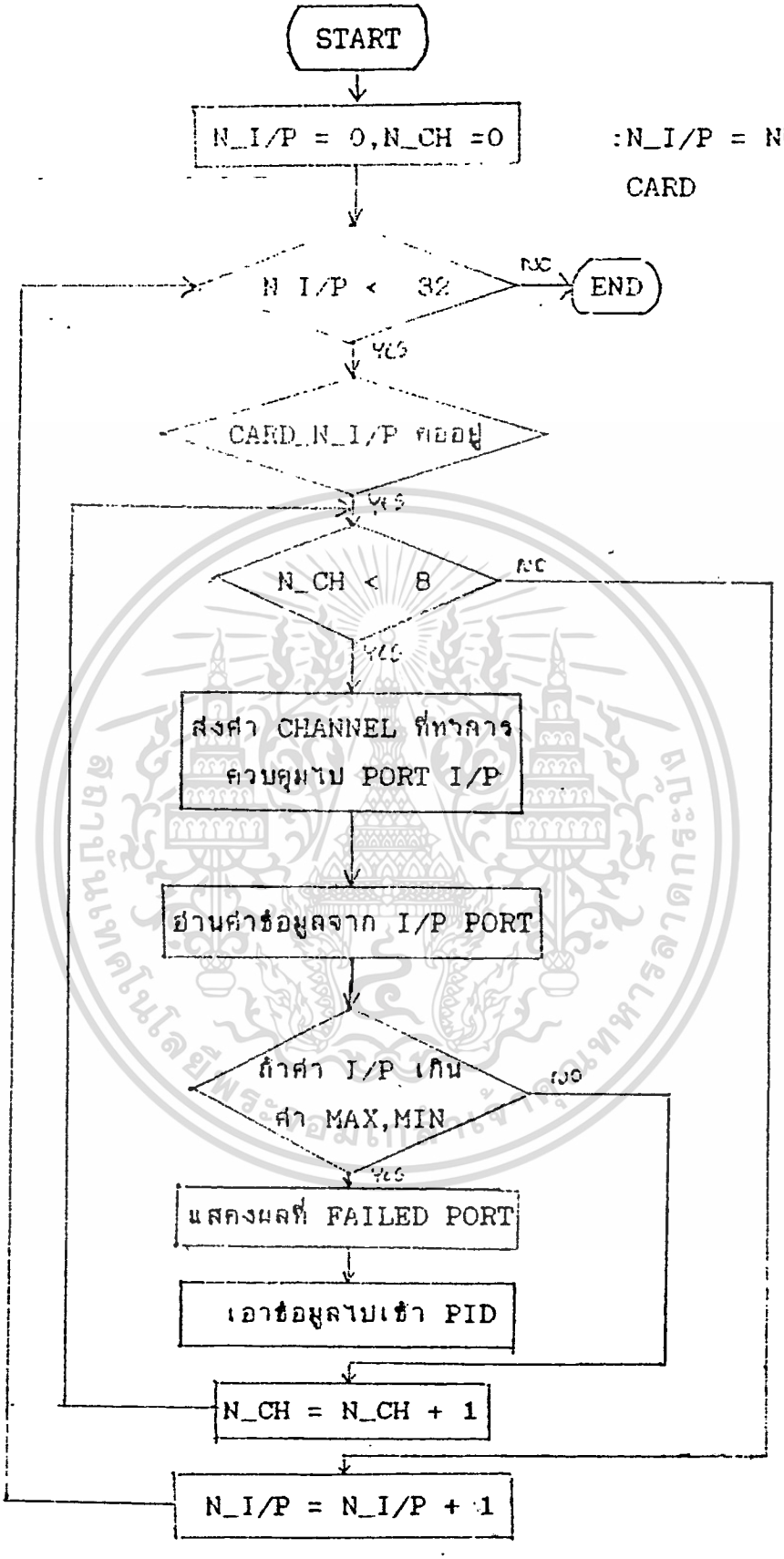
แผนภูมิของการรับ CHANNEL จาก HOST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



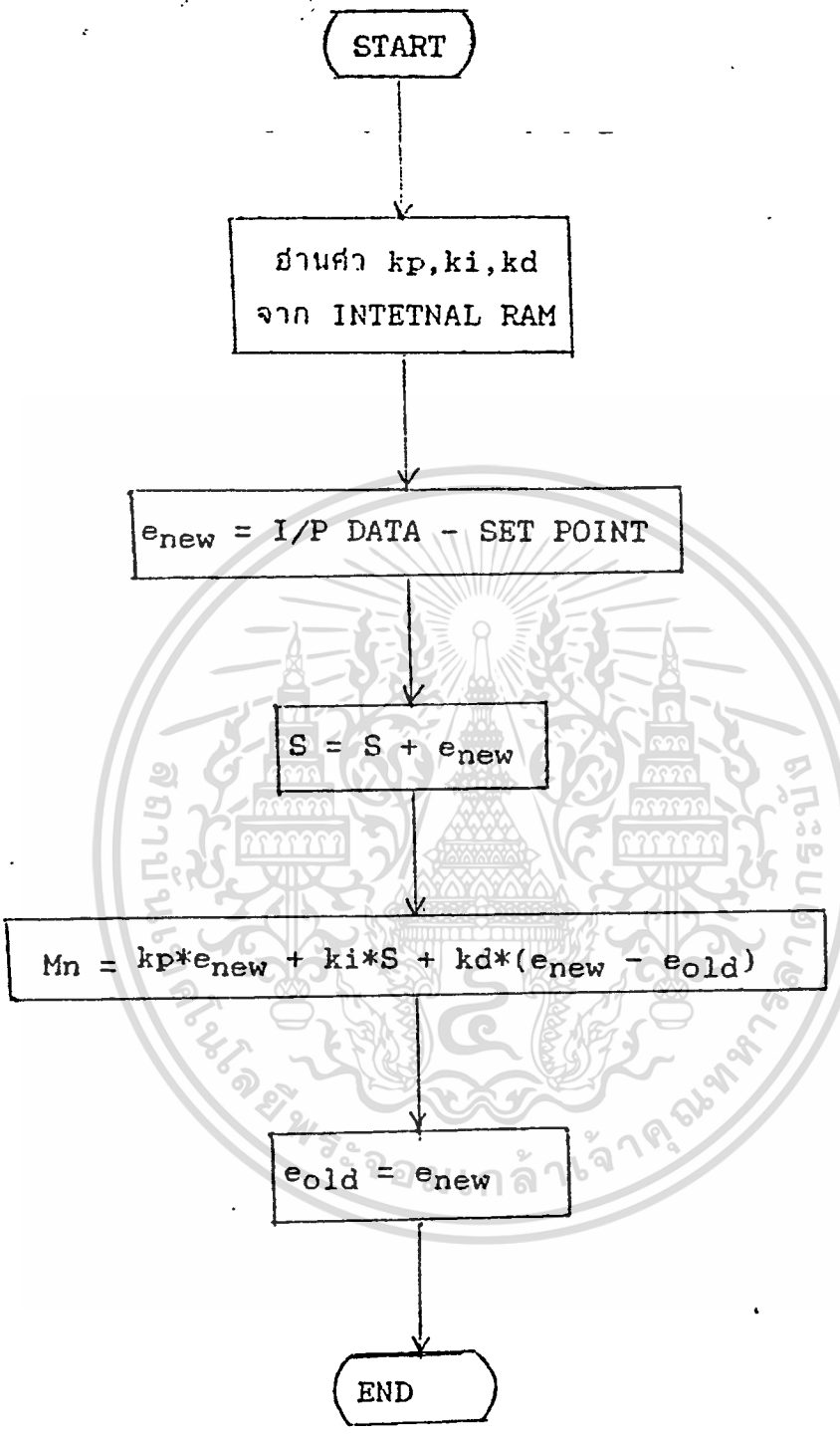
แผนภูมิของการรับอุณหภูมิจาก HOST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนภูมิของ PROCESS การเทียบ MAX,MIN บน FRONT END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนภูมิของการทำงานของ PID PROCESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนSOFT WARE ที่ทำงานอยู่บน COMPUTER(host)

เนื่องจากระบบ front end นี้ สามารถทำงานได้ด้วยตัวเองโดยไม่ต้องพึ่งระบบคอมพิวเตอร์ แต่คอมพิวเตอร์ก็สามารถเข้ามามีบทบาทเป็นส่วนประกอบในระบบเพื่อเพิ่มความสะดวกต่างๆแก่ผู้ใช้ ซึ่งหมายถึงการกำหนดค่าต่างๆและการแสดงผล

software ในส่วนนี้ส่วนใหญ่จะเป็นการกำหนดค่าต่างๆให้แก่ front end และการแสดงผล ซึ่งขั้นตอนการทำงานของโปรแกรมเป็นดังนี้

เมื่อเริ่มstart program การทำงานจะทำงานตามลำดับขั้นต่างๆต่อไปนี้

1) Initialization ส่วนนี้จะเริ่มต้นเช็คสภาวะ(condition)เริ่มต้นการทำงาน เช่น การจองเนื้อที่หน่วยความจำ(allocated memory area) และ การเช็คสภาวะเริ่มต้นให้แก่port (initialize serial port) โดยเฉพาะความเร็วของการส่งข้อมูล(baud rate) ที่ใช้นั้นผู้ใช้สามารถเลือกใช้ได้ตามความเหมาะสม

2) รอรับข้อมูลจาก front end ว่า slot ใดถูกใช้งานบ้าง

3) หลังจากนั้นจะเข้าสู่การกำหนดการทำงานของตัว front end ได้แก่

- การเลือกช่องใช้งาน(channel)ต่างๆในแต่ละ card
- เช็คค่า max,min,set point ของแต่ละช่องใช้งาน
- เช็คค่าคงที่ของ PID program

4) ส่งค่าต่างๆจากการเช็คให้กับตัว front end

5) load PID program จากดิสก์ พร้อมทั้งส่งไปที่ตัว front end

6) ส่งรหัส(code) ให้กับตัวfront end ให้ทำงานได้

7) รอรับข้อมูลจากตัว front end เพื่อแสดงผลในรูปของกราฟออกทางจอภาพ(monitor)

8) ทำข้อ(7)ซ้ำจนกว่าจะมีการรีเซ็ตเครื่อง

รายละเอียดของโปรแกรมในแต่ละส่วนมีดังนี้

1) INITIALIZATION จะทำหน้าที่ในการกำหนดสภาวะเริ่มต้นการทำงานต่างๆของเครื่องคอมพิวเตอร์ ประกอบด้วยส่วนต่างๆดังนี้

1.1) การ initial serial port เนื่องจากการส่งผ่านข้อมูลระหว่าง host กับ front end ใช้การส่งในระบบ RS-232 asynchronous serial communication standard และรูปแบบการส่งจะถูกกำหนดไว้ที่ serial port ซึ่งค่าต่างๆที่ถูกกำหนดไว้คือ 1 start bit, 8 data bit, non

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

parity,bit 1 stop bit และค่า baud rate นั้นกำหนดไว้ 4 ค่าให้เลือกใช้ซึ่งผู้ใช้จะต้องเลือก(ค่าเดียวกับที่เลือกในตัวfront end) ได้แก่ 9600, 4800,2400,1200การกำหนดค่าต่างๆเหล่านี้ใช้ interrupt 14H service 0 ช่วยในการกำหนดค่านี้

1.2) การจองเนื้อที่ในหน่วยความจำ(allocated data area)เป็นการกำหนดค่าแหน่งต่างๆของหน่วยความจำในการเก็บข้อมูลซึ่งถูกส่งมาจาก front end ซึ่งแบ่งเป็นส่วนย่อยต่างๆดังนี้

- การกำหนดค่าแหน่งต่างๆของหน่วยความจำสำหรับข้อมูลที่มาจากตัว front end

- การกำหนดค่าแหน่งต่างๆของหน่วนความจำสำหรับค่าที่กำหนดสภาวะการทำงานของตัวfront end ซึ่งได้แก่ ค่าset pointต่างๆ, ช่องสัญญาณที่ถูกคิดค่อด้วย,ค่าคงที่ต่างๆสำหรับ PID program

- การกำหนดค่าแหน่งต่างๆของหน่วนความจำสำหรับโปรแกมต่างๆ เช่น โปรแกรมการแสดงผล(กราฟ),โปรแกรมPID,

(2) serial port โปรแกรมส่วนนี้เป็นส่วนที่ควบคุมการส่งและการรับข้อมูลต่างๆระหว่างfront end กับตัว computer โดยค่าbaud rate ของการส่งรวมทั้งโปรโตคอลได้ถูกกำหนดไว้ในส่วนการinitialแล้ว ซึ่งโปรโตคอลใช้แบบ 1 start bit,8 data bit,non parity bit,1 stop bit สำหรับการ handshaking ระหว่างทั้งสองนี้ใช้ในการส่งและรับ โดยในการส่งตัวส่ง(transmitter) จะส่งรหัส '?' ให้กับตัวรับ(reciever)ซึ่งตัวรับนี้จะส่งสัญญาณกลับ(acknowleage)กลับมาด้วยรหัส ' ' หลังจากนั้นการส่งก็จะเกิดขึ้นและเพื่อป้องกันการรับข้อมูลไม่ทันของตัวรับ(reciever) จึงได้กำหนดให้การส่งในแต่ละไบต์(1 byte) ตัวส่ง(transmitter)จะต้องรอรับรหัส '&' จากตัวรับก่อนที่จะส่งไบต์ถัดไป

สำหรับขาสัญญาณ handshaking ตามมาตรฐาน RS-232 ได้ถูกต่อไว้ดังนี้คือ RST ถูกต่อร่วมกับ CTS และ DTR ถูกต่อร่วมกับ DSR เพื่อให้การคิดต่อสามารถเกิดขึ้นได้ตลอดเวลา

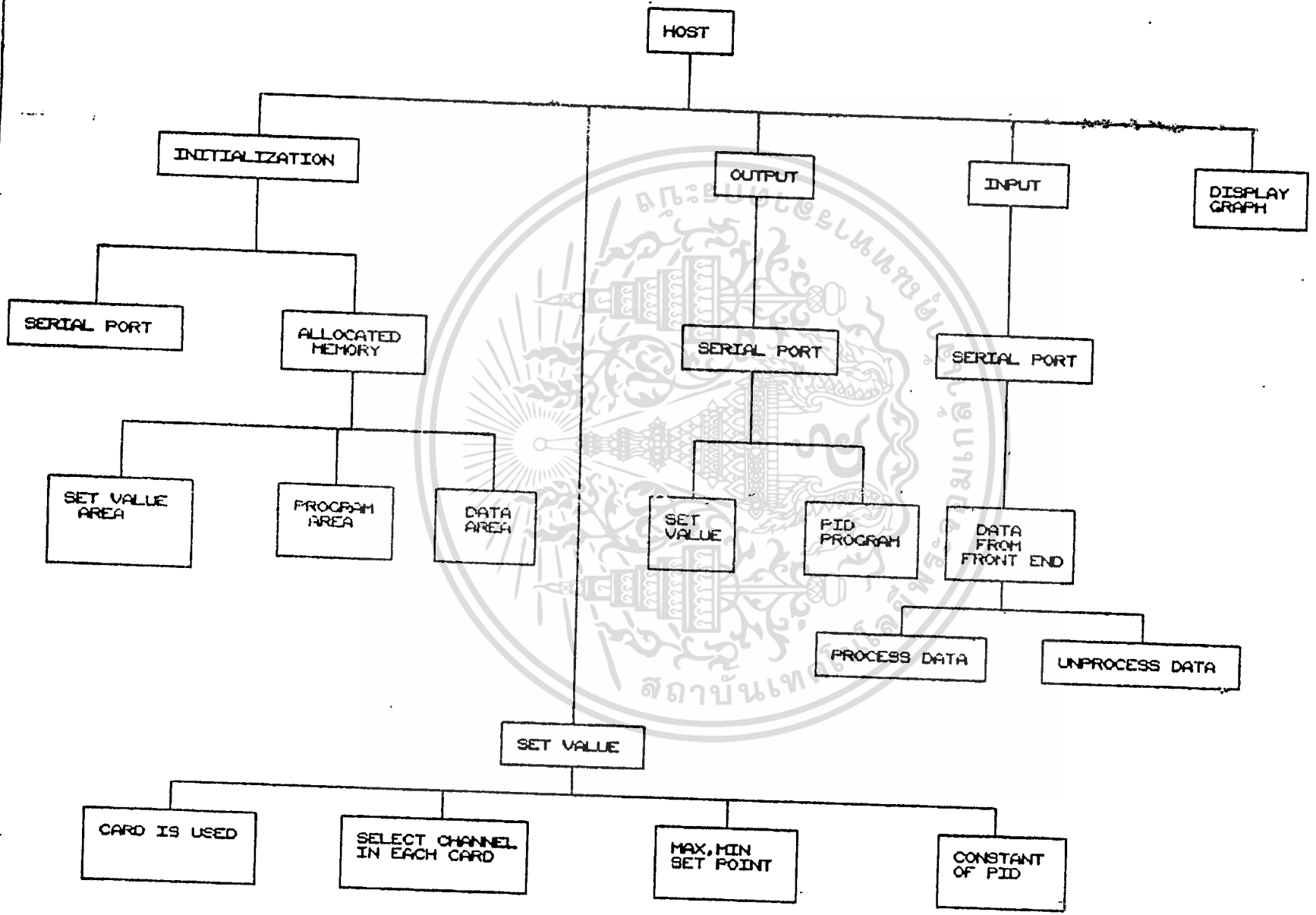
ส่วนโปรแกรมการรับส่งนั้นได้ใช้ DOS interrupt 14H service 1 และ 2 เข้าช่วยในการเขียนโปรแกรมทำให้สะดวกมากขึ้น

สำหรับค่ากำหนดที่ต้องส่งจากส่วน host ไปยังส่วน front end ซึ่งเป็นค่าที่กำหนดการทำงานต่างๆของ front end ค่าต่างๆคือ ค่าที่แสดงว่า

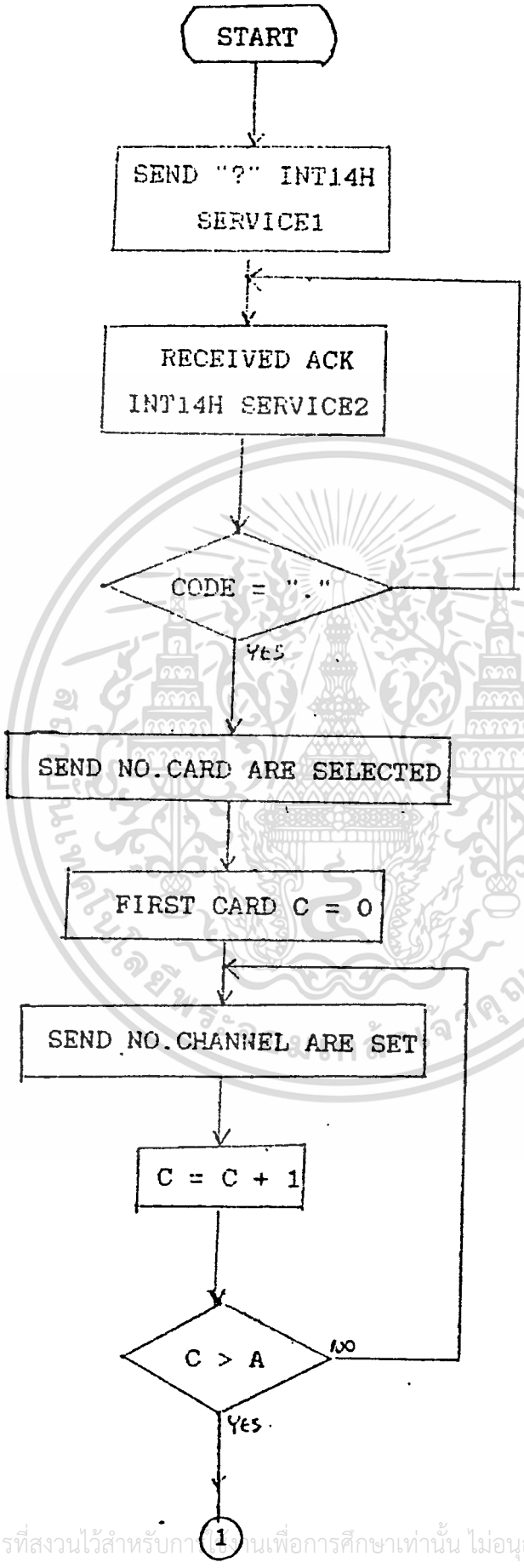
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

card ใดถูกติดต่ออยู่บ้าง,channelใดในแต่ละcardที่ถูกติดต่ออยู่บ้างและค่ากำหนด max,min,set point ของแต่ละchannel สำหรับลำดับการส่งนั้นเริ่มจากการส่งนั้นเริ่มจากการส่งตำแหน่งของ card ใดบ้างที่ถูกติดต่อด้วยโดยcard ที่ใช้สูงสุดคือ 32 card จึงแบ่งการส่งเป็น 4 byte แต่ละ bit แสดงสภาวะการติดต่อของแต่ละ card และค่าไบท์ที่ส่งต่อมาจะแสดงchannelที่ถูกเลือกในแต่ละcard (ในแต่ละcardมี 8 channel) สำหรับค่า max,min,set point จะส่งเรียงมาตามลำดับในแต่ละchannel โดยเรียงตามลำดับของchannelที่ถูกเลือก และการส่งในแต่ละbyteนั้น ตัวส่ง(transmitter)ก็จะต้องรอรหัส"&" จากตัวรับ(recieve)ก่อนที่จะส่งbyteถัดไป

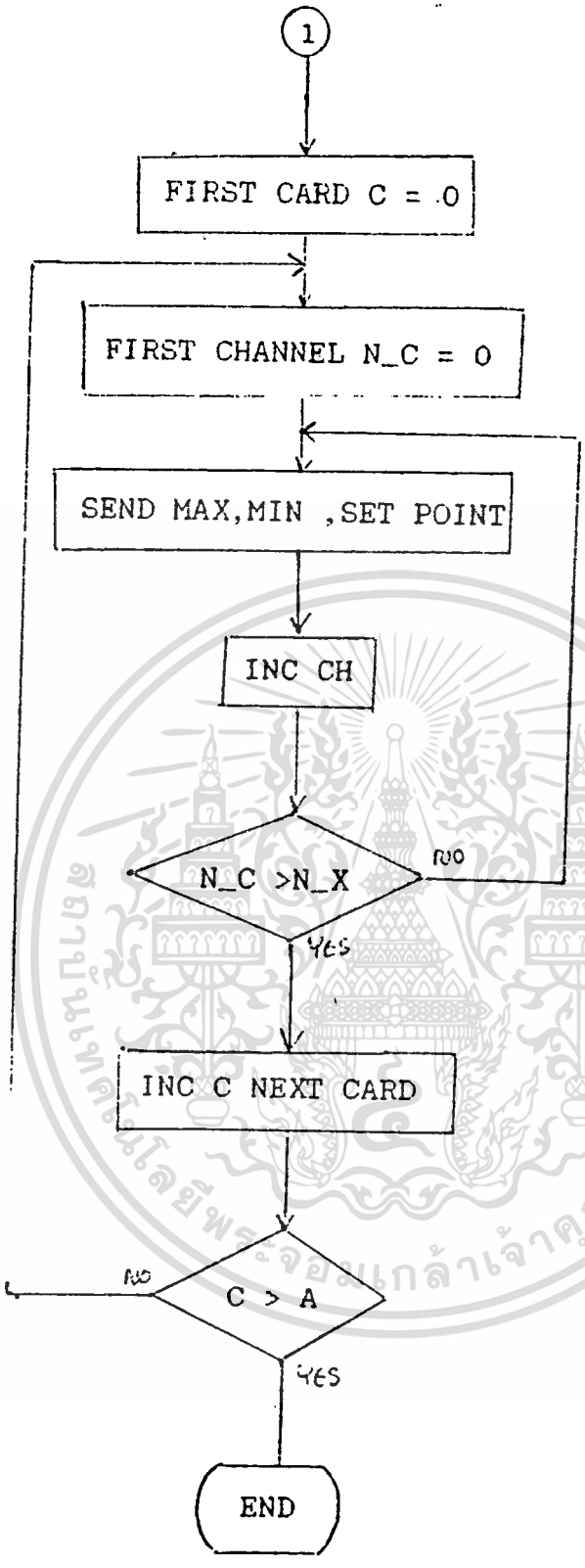
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



BLOCK DIAGRAM OF PROGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนภูมิแสดงการส่งผ่านค่าต่างๆให้กับ FRONT END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบระบบ

การทดสอบระบบเพื่อแสดงการทำงานอย่างง่ายของระบบเก็บข้อมูลส่วนหน้านี้คือ ทำการควบคุมอุณหภูมิของน้ำในกาต้มน้ำ โดยใช้รีเลย์ในการควบคุมรีเลย์ที่ใช้ในการปิดเปิดวาล์วทเจที่ป้อนให้กับคอยล์ของหม้อต้มน้ำ และตัวตรวจจับอุณหภูมิจะทำการวัดอุณหภูมิส่งป้อนให้กับระบบ

สัญญาณที่วัดได้จะเป็นสัญญาณอนาลอกระดับ 0-5 โวล์ทซึ่งค่าเหล่านี้จะถูกป้อนให้กับระบบ ระบบทำการรับข้อมูลแล้วแปลงเป็นสัญญาณดิจิตอลต่อไปแล้วจึงส่งต่อให้กับระบบประมวลผลทำการเปรียบเทียบและเข้าขบวนการควบคุมได้ค่าออกมาในรูปสัญญาณดิจิตอลส่งต่อไปยังคอมพิวเตอร์เพื่อแสดงผลและผ่านไปยังส่วนแปลงสัญญาณดิจิตอลเป็นอนาลอกออกไปควบคุมยังคอยล์ของหม้อต้มน้ำ

จากการทดลองดังนี้เริ่มจากการเพิ่มของอุณหภูมิขึ้นอย่างช้าๆและเมื่อเพิ่มจนถึงจุดที่ได้ทำการกำหนดไว้นั้น อุณหภูมิจะเพิ่มขึ้นอีกน้อยมาก

จากการทดลองยังมีข้อผิดพลาดอยู่เนื่องจากตัวตรวจจับอาจมีฮิสเทอร์รีซิสและอุปกรณ์ต่างๆในระบบจึงทำให้ค่าอุณหภูมิที่วัดได้มีค่ามากกว่าค่าที่ได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

การวิจารณ์และสรุป

ระบบการเก็บข้อมูลส่วนหน้าเป็นระบบที่มีความเหมาะสมในการควบคุมการทำงานแบบอัตโนมัติ ซึ่งมีการควบคุมโดยคอมพิวเตอร์รวมถึงสามารถทำการควบคุมการทำงานได้หลายๆ channel โดยทำการ multiplex สัญญาณได้ทีละหลายสัญญาณรวมถึงมีหลายลักษณะอีกด้วย ซึ่งระหว่างที่มีการทำงานนี้คอมพิวเตอร์จะมีหน้าที่ในการแสดงผลการทำงานเท่านั้น ทำให้สามารถนำเอาเวลาไปใช้ในงานอื่นได้ จากเหตุผลดังกล่าวทำให้สามารถทำการควบคุมการทำงานระยะไกลได้โดยการส่งผ่านสัญญาณผ่านทางโครงข่ายการสื่อสารสาธารณะ เพื่อใช้งานในระบบการควบคุมขนาดใหญ่

ข้อดีของระบบการเก็บข้อมูลส่วนหน้า

1)สามารถทำการควบคุมได้ในระยะไกลรวมถึงสามารถทำการควบคุมการทำงานของระบบได้เป็นจำนวนมากโดยผ่านคอมพิวเตอร์เพียงเครื่องเดียวซึ่งเป็นการลดค่าใช้จ่ายในการควบคุม

2)สามารถใช้กับการควบคุมได้หลายลักษณะโดยการเปลี่ยนค่าคงที่ที่ใช้ในระบบก็จะสามารถเปลี่ยนลักษณะของการควบคุมได้ง่ายกว่าการเปลี่ยนแปลงทาง hardware

3)สามารถทำการควบคุมระบบได้โดยไม่จำกัดลักษณะของตัว transducer มีข้อจำกัดเพียงต้องมีการส่งข้อมูลเข้าระบบเป็นข้อมูลขนาด 8 bit เท่านั้นทำให้มีความสะดวกในการใช้งานกับระบบต่างๆ

4)สามารถทำงานได้โดยไม่จำเป็นต้องมีคอมพิวเตอร์เนื่องจากคอมพิวเตอร์เป็นเพียงตัวแสดงผลในการควบคุมเท่านั้น

ข้อควรปรับปรุง

1)ระบบนี้ยังไม่สามารถทำการติดต่อกับระบบของ modem ได้ทำให้การควบคุมยังถูกจำกัดอยู่

2)ระบบนี้ทำให้สำหรับการควบคุมแบบ pid เท่านั้นควรมีการปรับปรุงให้มีการควบคุมในลักษณะอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สามารถประสบผลสำเร็จได้ ก็เนื่องจากได้รับคำปรึกษารวมถึงอุปกรณ์ที่ใช้จากบุคคลหลายฝ่าย ดังนี้คือ

อ.ชนิษฐา แซ่ตั้ง ภาควิชา อิเลคทรอนิคส์

อ.วัชระ ฉัตรวิริยะ ภาควิชา คอมพิวเตอร์

อ.วันชัย ริ้วรุจา ภาควิชา ระบบควบคุม

จึงขอขอบคุณมา ณ ที่นี้ ที่ได้ให้ความช่วยเหลือจนโครงงานนี้สำเร็จลุล่วงลงได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1)IBM," TECHNICAL REFERENCE ", PP.(1-215) - (1-243) , REVISED EDITION,1983

2)INTEL," MICROCONTROLLER HANDBOOK ",PP.(7-1) - (11-8), 1985

3)KRISHNA KANT," MICROPROCESSOR BASED DATA ACQUISITION SYSTEM DESIGN ", McGraw-Hill,PP.1-6,PP.47-126,1987

4)THE ENGINEERING STAFF OF ANALOG DEVICE,INC.," ANALOG - DIGITAL CONVERSION HANDBOOK ",PRENTICE - HALL , PP.17,PP.169 - 211,1986

5)DAROLD WOBSCHALL," CIRCUIT DESIGN FOR ELECTRONIC INSTRUMENT ",McGraw-Hill,PP.59 - 161,2nd EDITION,1987

6)STUARY BENNETT, " REAL TIME COMPUTER CONTROL ", PRENTICE HALL,PP.99 - 128,1988

7)BURR BROWN CORP.," THE HANDBOOK OF PERSONAL COMPUTER INSTRUMENTATION ",PP.1-1,2-2,(3-1) - (3-11),(4-1) - (4-4), THIRD EDITION,1988

8)HERBERT SCHILDT," C:POWER USER'S GUIDE ",OSBORNE McGraw-Hill,PP.207-208

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 7
MCS®-51 ARCHITECTURE

7.0 INTRODUCTION

The MCS®-51 family of 8-bit microcontrollers consists of the devices listed in Table 1, all of which are based on the MCS-51 architecture shown in Figure 7-1. The original 8051 was built in HMOS I technology. The HMOS II version, which is the device currently in production, is called the 8051AH. The term "8051," however, is still often used to generically refer to all of the MCS-51 family members. This is the case throughout this manual, except where specifically stated otherwise. Also for brevity, the term "8052" is used to refer to both the 8052 and the 8032, unless otherwise noted.

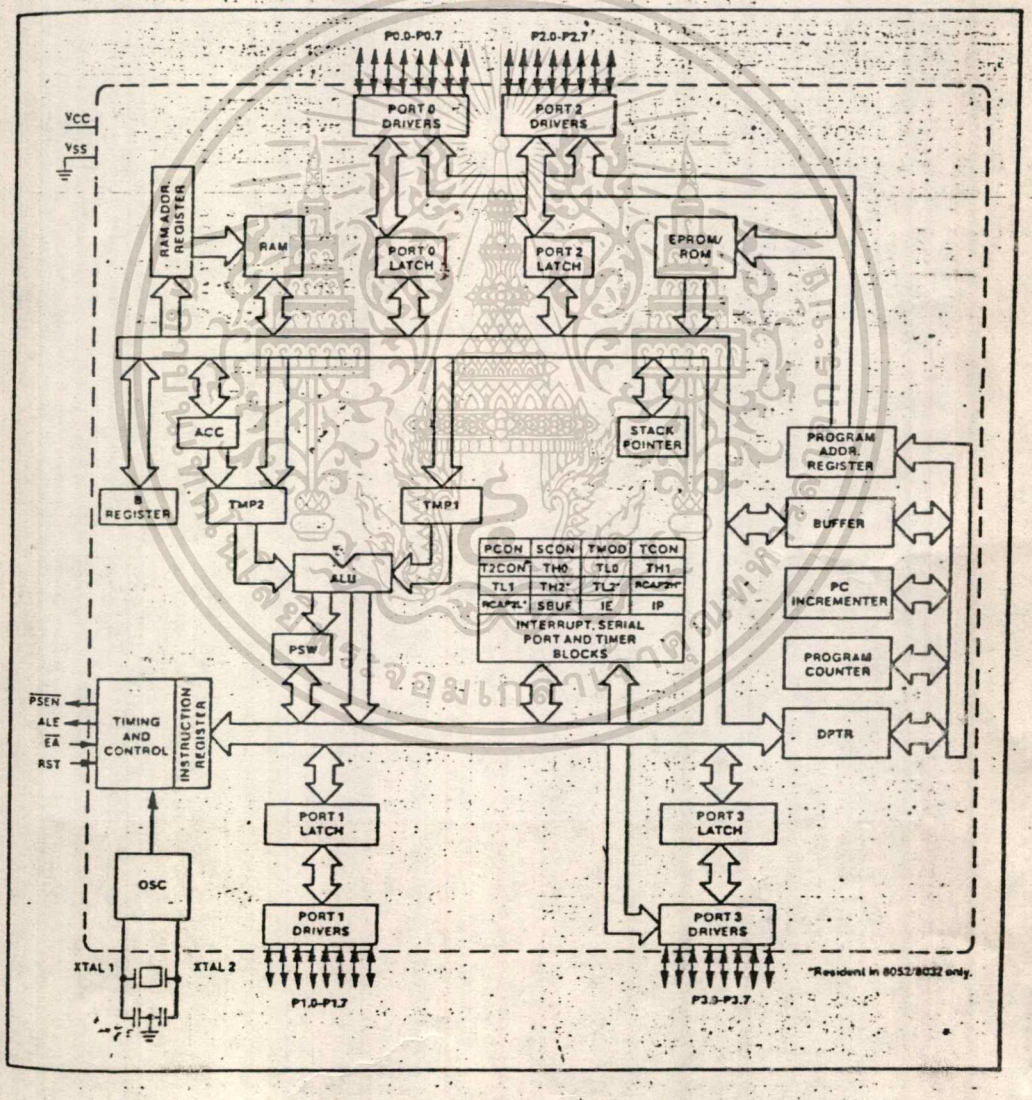


Figure 7-1. MCS-51 Architectural Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.13. A more detailed discussion will be found in Application Note AP-155, "Oscillators for Microcontrollers," which is included in this manual.

The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. The internal clocking signals are at half the oscillator frequency, and define the internal phases, states, and machine cycles, which are described in the next section.

7.3 CPU TIMING

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a Phase 1 half, during which the Phase 1 clock is active, and a Phase 2 half,

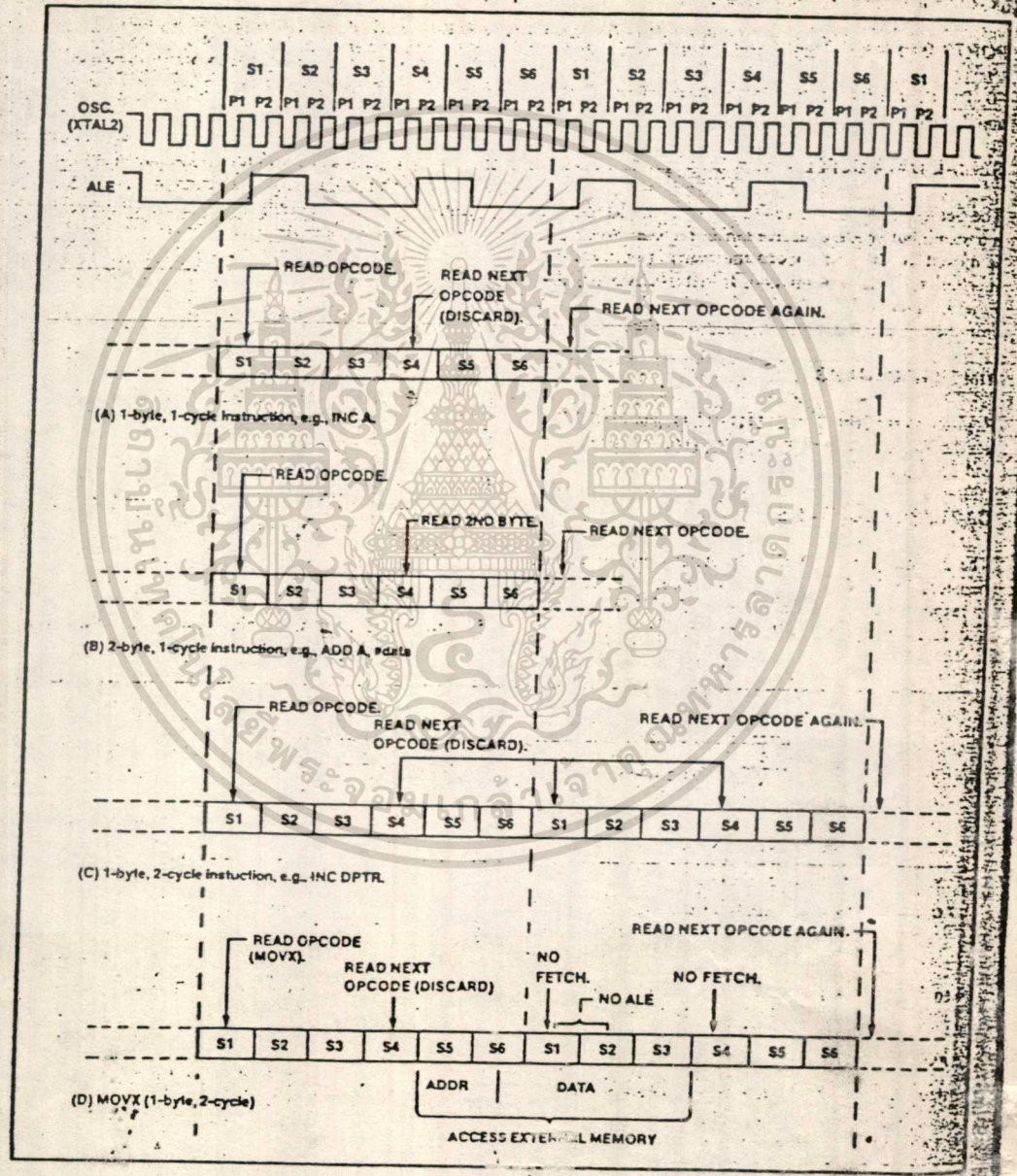


Figure 7-4. 8051 Fetch/Execute Sequences

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

low to enable the lower 4K (8K for the 8032) program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

7.5.1 PSEN

The read strobe for external fetches is $\overline{PSEN}$. $\overline{PSEN}$ is not activated for internal fetches. When the CPU is accessing external Program Memory, $\overline{PSEN}$ is activated twice every cycle (except during a MOVX instruction) whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 12 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$, takes 6 oscillator periods. The execution sequence for these two types of read cycles are shown in Figure 7-7 for comparison.

7.5.2 ALE

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 to an external latch during fetches from external Program Memory. For that purpose ALE is activated twice every machine cycle. This activation takes place even when the cycle involves no external fetch. The only time an ALE pulse doesn't come out is during an access to external Data Memory. The first ALE of the second cycle of a MOVX instruction is missing (see Figure 7-7). Consequently, in any system that does not use external Data Memory, ALE is activated at a constant rate of 1/6 the oscillator frequency, and can be used for external clocking or timing purposes.

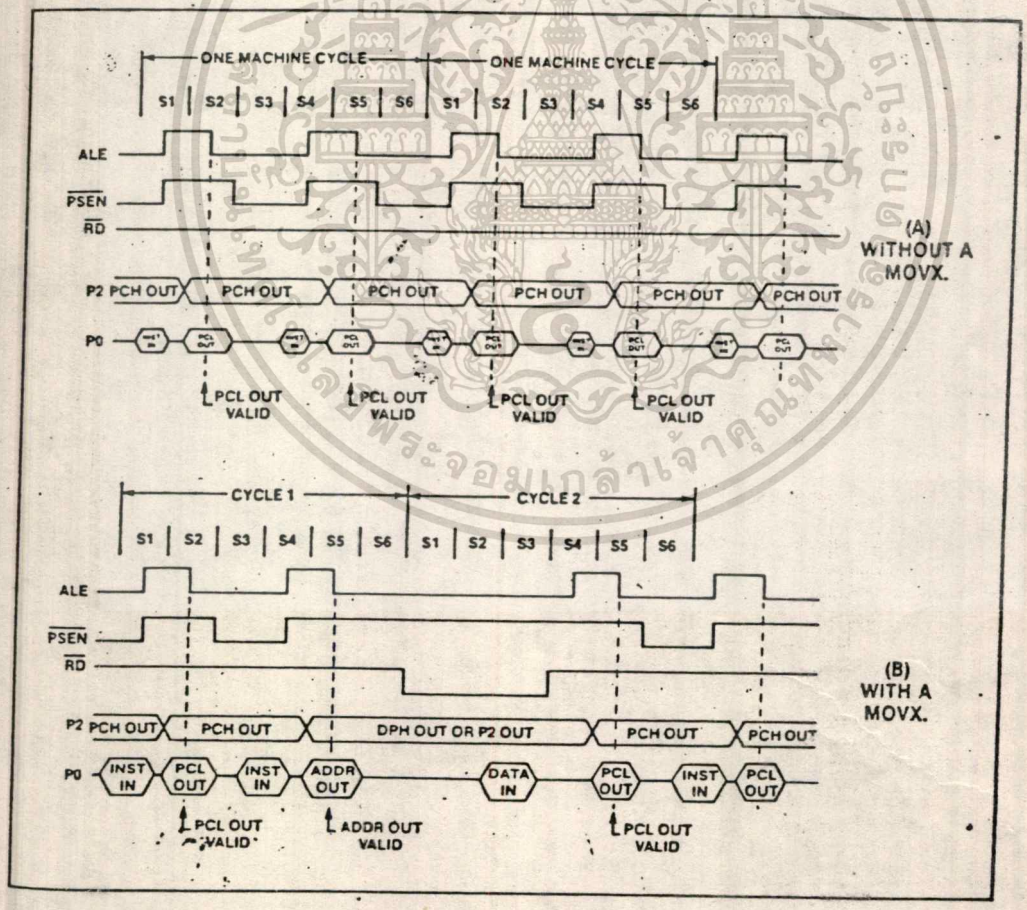


Figure 7-7. External Program Memory Execution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 8-1. 8051 Instruction Set Summary

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request, push the PC and to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7 μs @ 12 MHz).

INSTRUCTIONS THAT AFFECT FLAG SETTINGS[1]

INSTRUCTION	FLAG			INSTRUCTION	FLAG		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn —Register R7-R0 of the currently selected Register Bank.

direct —8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].

@Ri —8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.

#data —8-bit constant included in instruction.

#data 16 —16-bit constant included in instruction

addr 16 —16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 —11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel —Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.

bit —Direct Addressed bit in Internal Data RAM or Special Function Register.

• —New operation not provided by 8048AH/8049AH

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Oscillator Period
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12

ARITHMETIC OPERATIONS Cont.

Mnemonic		Description	Byte	Oscillator Period
SUBB	A,@Ri	Subtract indirect RAM from Acc with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12

All mnemonics copyrighted ©Intel Corporation 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 8-1. 8051 Instruction Set Summary (Continued)

LOGICAL OPERATIONS

Mnemonic		Description	Byte	Oscillator Period
ANL	A,Rn	AND register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12

LOGICAL OPERATIONS Cont.

Mnemonic		Description	Byte	Oscillator Period
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12

All mnemonics copyrighted © Intel Corporation 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 8-1. 8051 Instruction Set Summary (Continued)

DATA TRANSFER				
	Mnemonic	Description	Byte	Oscillator Period
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12

DATA TRANSFER Cont.				
	Mnemonic	Description	Byte	Oscillator Period
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12

All mnemonics copyrighted ©Intel Corporation 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 8-1. 8051 Instruction Set Summary (Continued)

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Oscillator Period
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to Carry	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24

PROGRAM BRANCHING

Mnemonic		Description	Byte	Oscillator Period
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24

PROGRAM BRANCHING Cont.

Mnemonic		Description	Byte	Oscillator Period
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump If Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	3	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

All mnemonics copyrighted ©Intel Corporation 1980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

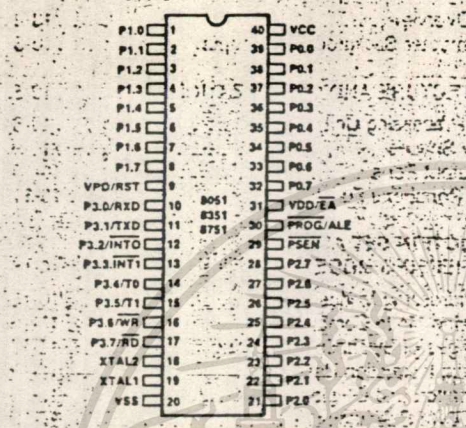



Figure 1a. 8051 Microcomputer Pinout Diagram

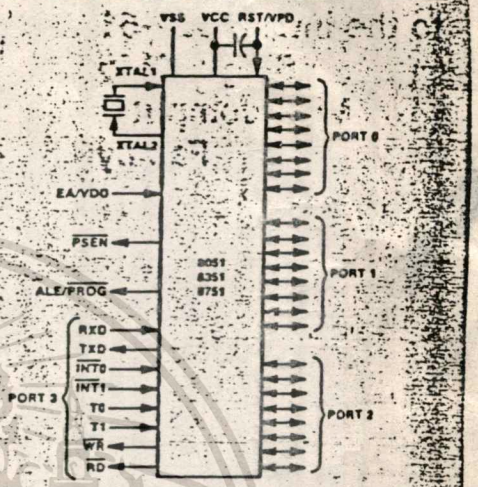


Figure 1b. 8051 Microcomputer Logic Symbol

1. INTRODUCTION

In 1976 Intel introduced the MCS-48™ family, consisting of the 8048, 8748, and 8035 microcomputers. These parts marked the first time a complete microcomputer system, including an eight-bit CPU, 1024 8-bit words of ROM or EPROM program memory, 64 words of data memory, I/O ports and an eight-bit timer/counter could be integrated onto a single silicon chip. Depending only on the program memory contents, one chip could control a limitless variety of products, ranging from appliances or automobile engines to text or data processing equipment. Follow-on products stretched the MCS-48™ architecture in several directions: the 8049 and 8039 doubled the amount of on-chip memory and ran 83% faster; the 8021 reduced costs by executing a subset of the 8048 instructions with a somewhat slower clock; and the 8022 put a unique two-channel 8-bit analog-to-digital converter on the same NMOS chip as the computer, letting the chip interface directly with analog transducers.

Now three new high-performance single-chip microcomputers—the Intel® 8051, 8751, and 8031—extend the advantages of Integrated Electronics to whole new product areas. Thanks to Intel's new HMOS technology, the MCS-51™ family provides four times the program memory and twice the data memory as the 8048 on a single chip. New I/O and peripheral capabilities both increase the range of applicability and reduce total system cost. Depending on the use, processing throughput increases by two and one-half to ten times.

This Application Note is intended to introduce the reader to the MCS-51™ architecture and features. While it does not assume intimacy with the MCS-48™ product line on the part of the reader, he/she should be familiar with some microprocessor (preferably Intel's, of course) or have a background in computer programming and digital logic.

Family Overview

Pinout diagrams for the 8051, 8751, and 8031 are shown in Figure 1. The devices include the following features:

- Single-supply 5 volt operation using HMOS technology.
- 4096 bytes program memory on-chip (not on 8031).
- 128 bytes data memory on-chip.
- Four register banks.
- 128 User-defined software flags.
- 64 Kilobytes each program and external RAM addressability.
- One microsecond instruction cycle with 12 MHz crystal.
- 32 bidirectional I/O lines organized as four 8-bit ports (16 lines on 8031).
- Multiple mode, high-speed programmable Serial Port.
- Two multiple mode, 16-bit Timer/Counters.
- Two-level prioritized interrupt structure.
- Full depth stack for subroutine return linkage and data storage.
- Augmented MCS-48™ instruction set.
- Direct Byte and Bit addressability.
- Binary or Decimal arithmetic.
- Signed-overflow detection and parity computation.
- Hardware Multiple and Divide in 4 usec.
- Integrated Boolean Processor for control applications.
- Upwardly compatible with existing 8048 software.

AFN-01502A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

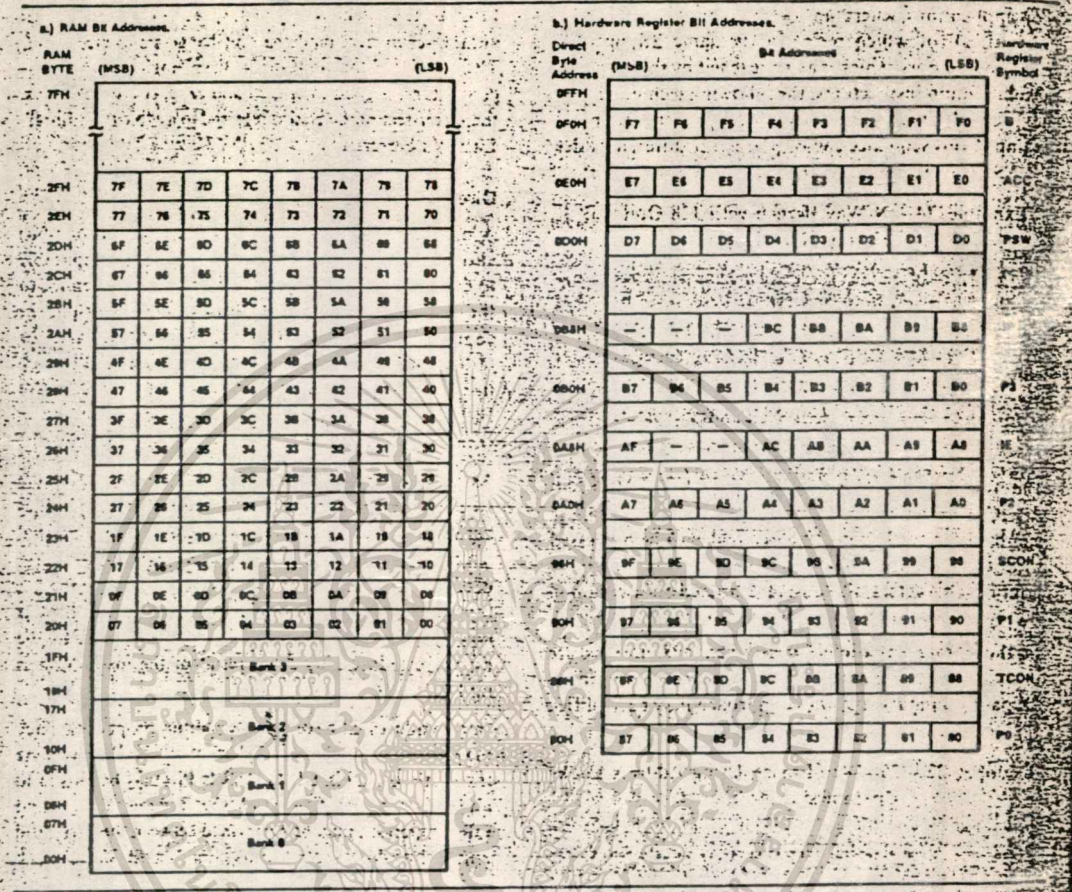


Figure 18. Bit Address Maps

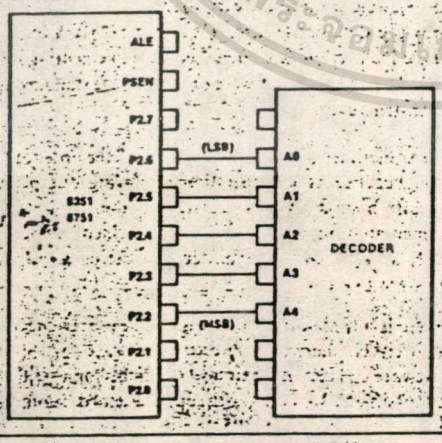


Figure 19. "Mismatch" Between I/O port and Decoder

Example 26 — Re-ordering I/O Port Configuration

```
OUT_P2:  RRC   A       ;MOVE ORIGINAL ACC 0 INTO CY
         MOV   P2.6,C  ;STORE CARRY TO PIN P26
         RRC   A       ;MOVE ORIGINAL ACC 1 INTO CY
         MOV   P2.5,C  ;STORE CARRY TO PIN P25
         RRC   A       ;MOVE ORIGINAL ACC 2 INTO CY
         MOV   P2.4,C  ;STORE CARRY TO PIN P24
         RRC   A       ;MOVE ORIGINAL ACC 3 INTO CY
         MOV   P2.3,C  ;STORE CARRY TO PIN P23
         RRC   A       ;MOVE ORIGINAL ACC 4 INTO CY
         MOV   P2.2,C  ;STORE CARRY TO PIN P22
         RET
```

Solving Combinatorial Logic Equations — ANL, ORL

Virtually all hardware designers are familiar with the problem of solving complex functions using combinatorial logic. The technologies involved may vary greatly, from multiple contact relay logic, vacuum tubes, TTL, or CMOS to more esoteric approaches like fluidics, but in each case, the goal is the same: a Boolean (true/false) function is computed on a number of Boolean variables.

AFN-01502A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

2764A
ADVANCED 64K (8Kx8) UV ERASABLE PROM

- Fast 180 nsec Access Time —HMOS II*-E Technology
- Low Power
  —60 mA Maximum Active
  —20 mA Maximum Standby
- Two Line Control
- Int$_e$ligent Programming™ Algorithm —Fastest EPROM Programming
- Int$_e$ligent Identifier™ Mode —Automated Programming Operations
- Compatible with 2764, 27128, 27256
- ±10% V_{CC} Tolerance Available

The Intel 2764A is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2764A is an advanced version of the 2764 and is fabricated with Intel's HMOSII-E technology which significantly reduces die size and greatly improves the device's performance, power consumption, reliability and producibility.

The 2764A provides access times to 180 ns/2764A-1). This is an improvement over the fastest 2764 access time of 200 ns. This is compatible with high-performance microprocessors, such as Intel's 8 MHz iAPX 186 allowing full speed operation without the addition of WAIT states. The 2764A is also directly compatible with the 12 MHz 8051 family.

Several advanced features have been designed into the 2764A that allow fast and reliable programming—the int$_e$ligent Programming Algorithm and the int$_e$ligent Identifier Mode. Programming equipment that takes advantage of these innovations will electronically identify the 2764A and then rapidly program it using an efficient programming method.

The 2764A also offers reduced power consumption compared to the 2764. The maximum active current on faster speed parts is 60 mA while the maximum standby current is only 20 mA. The standby mode lowers power consumption without increasing access time.

Two-line control and JEDEC-approved, 28 pin packaging are standard features of all Intel higher density EPROMs. This ensures easy microprocessor interfacing and minimum design efforts when upgrading, adding or choosing between non-volatile memory alternatives.

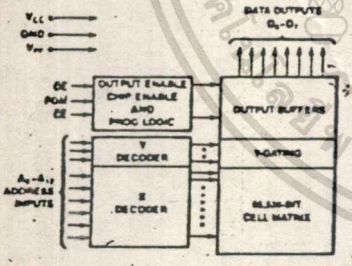
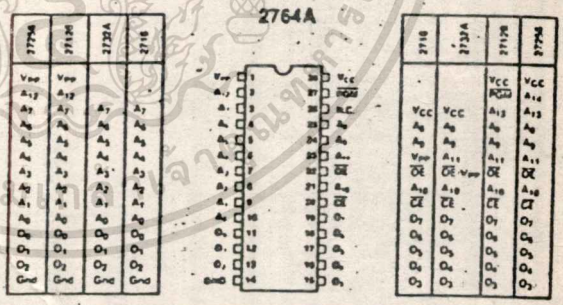


Figure 1. Block Diagram



NOTE: INTEL "UNIVERSAL BITE"-COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 2764A PINS

Figure 2. Pin Configurations

MODE SELECTION

MODE	CE (20)	OE (22)	PGM (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Algorithm Inhibit	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Algorithm Programming [2]	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ± 0.5V

*HMOS is a patented process of Intel Corporation

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
PGM	PROGRAM
N.C.	NO CONNECT

Intel Corporation Assumes No Responsibility for the Use of Any Circuitry Other Than Circuitry Embodied in an Intel Product. No Other Circuit Patent Licenses are Implied.
©INTEL CORPORATION, 1984.

JULY 1984
ORDER NUMBER: 210759-005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....–10°C to +80°C
Storage Temperature .....–65°C to +125°C
All Input or Output Voltages with Respect to Ground .....+6.5V to –0.6V
Voltage on Pin 24 with Respect to Ground .....+13.5V to –0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming .....+14V to –0.6V

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2764A-1, 2764A-2, 2764A 2764A-3, 2764A-4	2764A-20, 2764A-25 2764A-30, 2764A-45
Operating Temperature Range	0°–70°C	0°–70°C
V_{CC} Power Supply[1,2]	5V ±5%	5V ±10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min	Typ[3]	Max	Unit	Conditions
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby			20/35[5]	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active			60/75[5]	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	-.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA
V_{PP}[2]	V_{PP} Read Voltage	3.8		V_{CC}	V	$V_{CC} = 5.0V \pm 0.25V$

A.C. CHARACTERISTICS

Symbol	Parameter	2764A-1 Limits Min	Max	2764A-20 & 2764A-2 Limits Min	Max	2764A-25 & 2764A Limits Min	Max	2764A-30 & 2764A-3 Limits Min	Max	2764A-45 & 2764A-4 Limits Min	Max	Unit	Test Conditions
t_{ACC}	Address to Output Delay		180		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		180		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		65		75		100		120		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ or $\overline{CE}$ High to Output Data Float	0	55	0	55	0	60	0	105	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

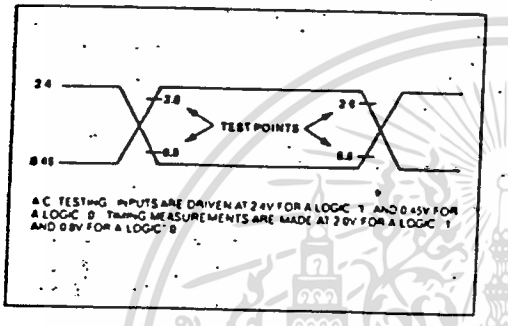
NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP}.
3. Typical values are for $T_A = 25°C$ and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on the following page.
5. Max I_{CC} rating differs with access time. Rating of 60 mA active and 20 mA standby are for 2764As at 200 nsec and 180 nsec access time only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAPACITANCE[2] ($T_A = 25°C$, $f = 1MHz$)

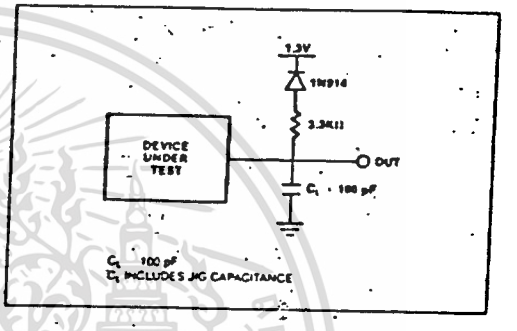
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

A.C. TESTING INPUT/OUTPUT WAVEFORM

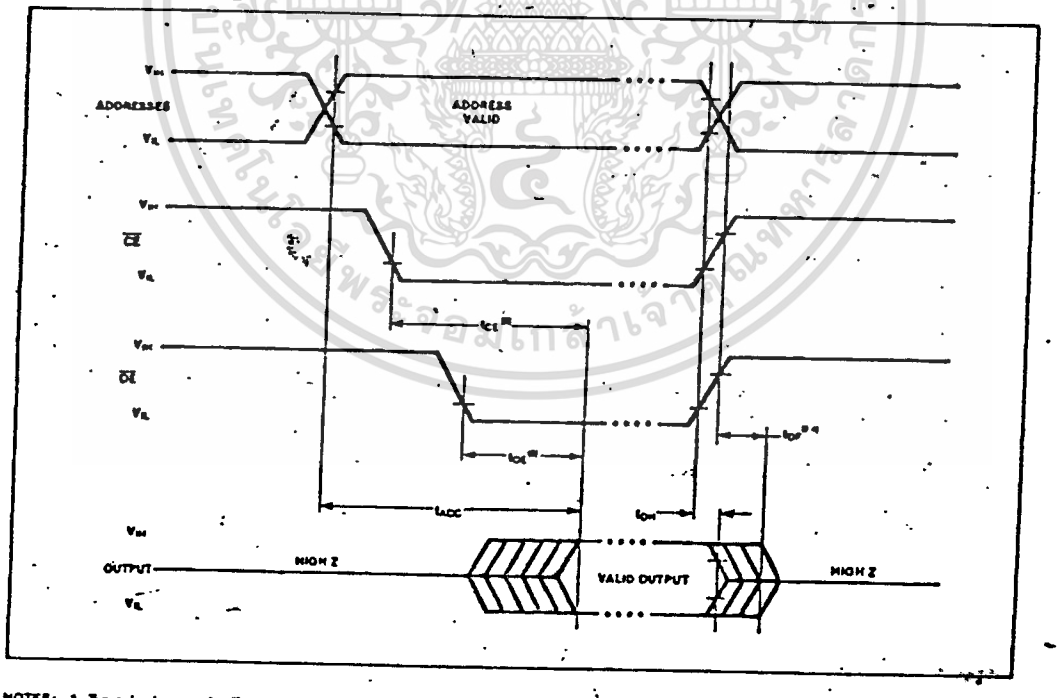


A.C. TESTING INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT



A.C. WAVEFORMS



NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

A to D, D to A

DAC0808, DAC0807, DAC0806 8-Bit D/A Converters

General Description

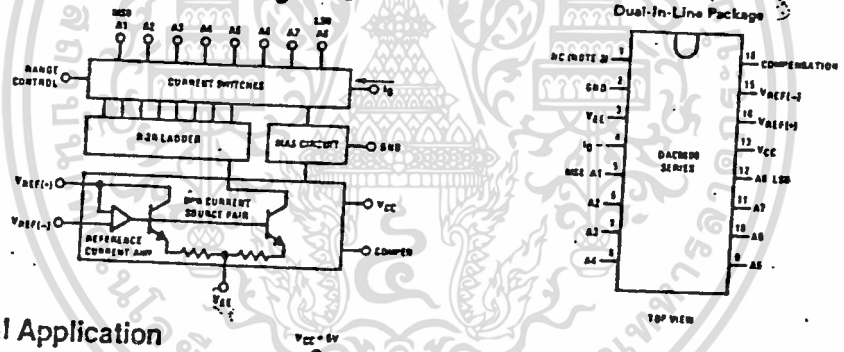
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 I_{REF}/256. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 μA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

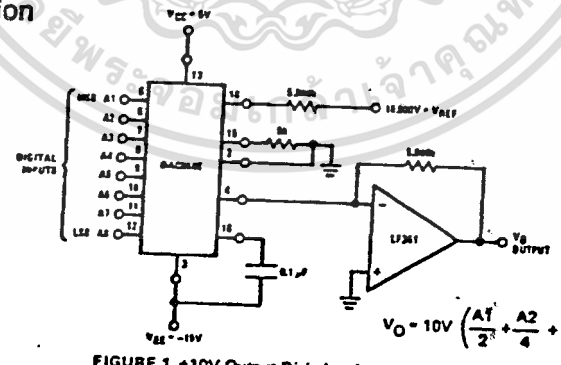
Features

- Relative accuracy: ±0.19% error maximum (DAC0808)
- Full scale current match: ±1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/μs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

Block and Connection Diagrams



Typical Application



$$V_O = 10V\left(\frac{A1}{2} + \frac{A2}{4} + \cdots \frac{A8}{256}\right)$$

FIGURE 1. +10V Output Digital to Analog Converter

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS*					
		D PACKAGE (D16C)		J PACKAGE (J16A)		N PACKAGE (N16A)	
8-bit	−55°C ≤ T_A ≤ +125°C	DAC0808LD	MC1508L8				
8-bit	0°C ≤ T_A ≤ +75°C			DAC0808LCJ	MC1408L8	DAC0808LCN	MC1408P8
7-bit	0°C ≤ T_A ≤ +75°C			DAC0807LCJ	MC1408L7	DAC0807LCN	MC1408P7
6-bit	0°C ≤ T_A ≤ +75°C			DAC0806LCJ	MC1408L6	DAC0806LCN	MC1408P6

*Note. Devices may be ordered by using either order number.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Power Supply Voltage	
V_{CC}	+18 V_{DC}
V_{EE}	−18 V_{DC}
Digital Input Voltage, V5–V12	−10 V_{DC} to +18 V_{DC}
Applied Output Voltage, V_O	−11 V_{DC} to +18 V_{DC}
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V14, V15	V_{CC}, V_{EE}
Power Dissipation (Package Limitation)	1000 mW
Derate above T_A = 25°C	6.7 mW/°C
Operating Temperature Range	
DAC0808L	−55°C ≤ T_A ≤ +125°C
DAC0808LC Series	0 ≤ T_A ≤ +75°C
Storage Temperature Range	−65°C to +150°C

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, $V_{REF}/R14$ = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
E_r	Relative Accuracy (Error Relative to Full Scale I_O)	*(Figure 4)*				%
	DAC0808L (LM1508-8), DAC0808LC (LM1408-8)				±0.19	%
	DAC0807LC (LM1408-7), (Note 1)				±0.39	%
	DAC0806LC (LM1408-6), (Note 1)				±0.78	%
	Settling Time to Within 1/2 LSB (Includes t_{PLH})	T_A = 25°C (Note 2), *(Figure 5)*		150		ns
t_{PLH}, t_{PHL}	Propagation Delay Time	T_A = 25°C, *(Figure 5)*		30	100	ns
TCI_O	Output Full Scale Current Drift			±20		ppm/°C
MSB	Digital Input Logic Levels	*(Figure 3)*				
V_{IH}	High Level, Logic "1"		2			V_{DC}
V_{IL}	Low Level, Logic "0"				0.8	V_{DC}
MSB	Digital Input Current	*(Figure 3)*				
	High Level	V_{IH} = 5V		0	0.040	mA
	Low Level	V_{IL} = 0.8V		−0.003	−0.8	mA
I_{15}	Reference Input Bias Current	*(Figure 3)*		−1	−3	μA
	Output Current Range	*(Figure 3)*				
		V_{EE} = −5V	0	2.0	2.1	mA
		V_{EE} = −15V, T_A = 25°C	0	2.0	4.2	mA
I_O	Output Current	V_{REF} = 2.000V, R14 = 1000Ω, *(Figure 3)*	1.9	1.99	2.1	mA
	Output Current, All Bits Low	*(Figure 3)*		0	4	μA
	Output Voltage Compliance Pin 1 Grounded, V_{EE} Below −10V	E_r ≤ 0.19%, T_A = 25°C			−0.55, +0.4	V_{DC}
					−5.0, +0.4	V_{DC}
SRI_{REF}	Reference Current Slew Rate	*(Figure 6)*	4	8		mA/μs
	Output Current Power Supply Sensitivity	−5V ≤ V_{EE} ≤ −16.5V		0.05	2.7	μA/V
	Power Supply Current (All Bits Low)	*(Figure 3)*				
I_{CC}				2.3	22	mA
I_{EE}				−4.3	−13	mA
	Power Supply Voltage Range	T_A = 25°C, *(Figure 3)*				
V_{CC}			4.5	5.0	5.5	V_{DC}
V_{EE}			−4.5	−15	−16.5	V_{DC}
	Power Dissipation					
	All Bits Low	V_{CC} = 5V, V_{EE} = −5V		33	170	mW
		V_{CC} = 5V, V_{EE} = −15V		106	305	mW
	All Bits High	V_{CC} = 15V, V_{EE} = −5V		90		mW
		V_{CC} = 15V, V_{EE} = −15V		160		mW

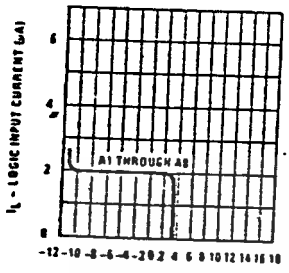
Note 1: All current switches are tested to guarantee at least 50% of rated current.
Note 2: All bits switched.
Note 3: Range control is not required.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

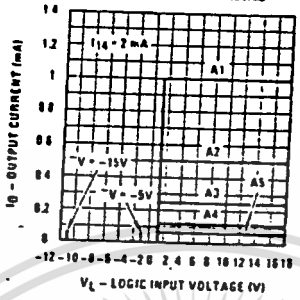
Typical Performance Characteristics

V_{CC} = 5V, V_{EE} = −15V, T_A = 25°C, unless otherwise noted

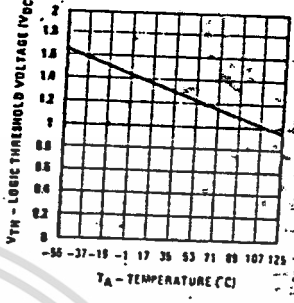
Logic Input Current vs Input Voltage



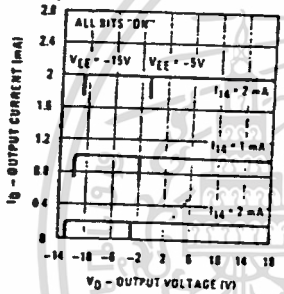
Bit Transfer Characteristics



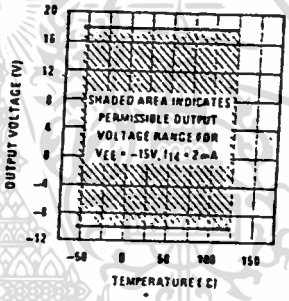
Logic Threshold Voltage vs Temperature



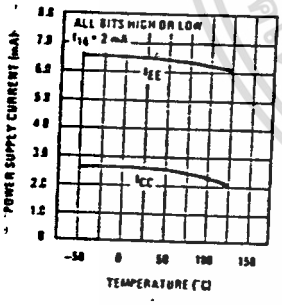
Output Current vs Output Voltage (Output Voltage Compliance)



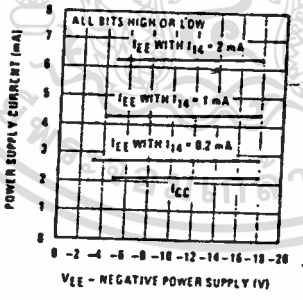
Output Voltage Compliance vs Temperature



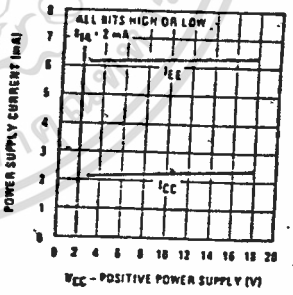
Typical Power Supply Current vs Temperature



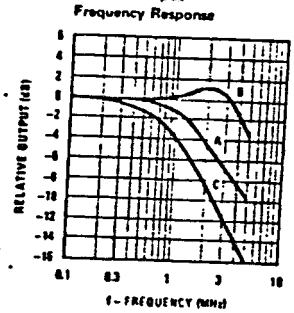
Typical Power Supply Current vs V_{EE}



Typical Power Supply Current vs V_{CC}



Reference Input Frequency Response



Unless otherwise specified: R14 = R15 = 1 kΩ, C = 15 pF, pin 16 to V_{EE}, R_L = 50Ω, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of *Figure 7*, V_{REF} = 2 Vp-p offset 1 V above ground

Curve B: Small Signal Bandwidth Method of *Figure 7*, R_L = 250Ω, V_{REF} = 50 mVp-p offset 200 mV above ground.

Curve C: Large and Small Signal Bandwidth Method of *Figure 9* (no op amp, R_L = 50Ω), R_S = 50Ω, V_{REF} = 2V, V_S = 100 mVp-p centered at 0V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

PRELIMINARY

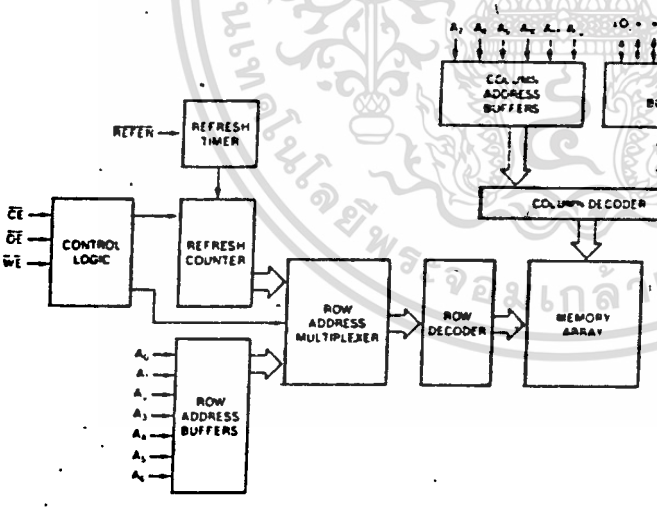
2187A FAMILY
8192 x 8 BIT INTEGRATED RAM

- Low-cost, high-volume HMOS technology
- High density one transistor cell
- Single +5V ± 10% supply
- Proven HMOS Reliability
- Low active current (70 mA)
- Simple synchronous refresh operation
- 2764 EPROM compatible pin-out
- Two-line bus control
- JEDEC standard 28-pin site
- Low standby current (20 mA)

The Intel 2187 is an 8192 word by 8-bit integrated random access memory (iRAM) fabricated on Intel's proven HMOS dynamic RAM technology. Packaged in the industry standard 28-pin DIP, the 2187 conforms to the industry standard JEDEC 28-pin site.

The 2187 is particularly suited for use in microcontroller applications, incorporating many requisite system features. These include low power dissipation, automatic initialization, extended cycle operation and two-line bus control to eliminate bus contention.

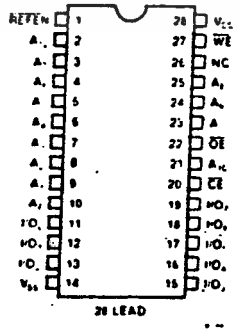
BLOCK DIAGRAM



PIN NAMES

A_0-A_{12}	ADDRESS INPUTS
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
$\overline{WE}$	WRITE ENABLE
I/O_1-I/O_8	DATA INPUT/OUTPUT
$\overline{REFEN}$	REFRESH ENABLE
V_{CC}	+5V POWER
V_{SS}	GROUND

PIN CONFIGURATION



28 LEAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −1.0V to +7V
D.C. Continuous Current per Output ..... 10 mA
D.C. Maximum Data Out Current ..... 50 mA
D.C. Power Dissipation ..... 1.0 W

** COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS[1]

TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted.

Symbol	Parameter	Limits Min.	Limits Max.	Unit	Test Conditions	Notes
ILI	Input Load Current (All Input Pins)		10	μA	V_{IN} = VSS to VCC	
ILO	Output Leakage Current		10	μA	$\overline{OE}$ = VIH	
ICC	Operating Current		70	mA	Minimum Cycle Time	2
ISB	Standby Current		20	mA	$\overline{CE}$ = VIH	
VIL	Input Low Voltage	−1.0	0.8	V		3
VIH	Input High Voltage	2.4	7.0	V		
VOL	Output Low Voltage		0.45	V	IOL = 2.1 mA	
VOH	Output High Voltage	2.4		V	IOH = −1.0 mA	

NOTES FOR D.C. CHARACTERISTICS:
1. Typical limits are VCC = +5V, TA = 25°C
2. ICC is dependent on outputs loading when the device output is selected. Specified ICC max. is measured with the outputs open
3. Specified VIL min. is for steady state operation. During transitions the inputs may overshoot to −2.0V for periods not to exceed 20 nsec

A.C. TEST CONDITIONS

Input Pulse and Timing
Reference Levels ..... 0.8V to 2.4V
Input Rise and Fall Times ..... 10 nsec.
Output Timing Reference Levels .... 0.6V and 2.4V
Output Load ..... See Figure 1

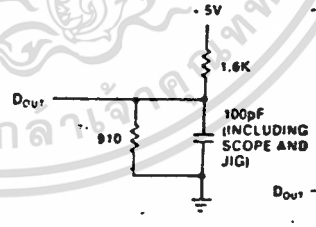


Figure 1.

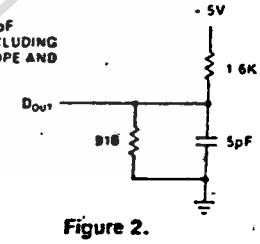


Figure 2.

(FOR HIGH IMPEDANCE MEASUREMENTS ONLY)

CAPACITANCE [4]

TA = 25°C, f = 1.0 MHz

Symbol	Parameter	Max.	Unit	Conditions
C_{ADD}	Address Capacitance	8	pF	V_{ADD} = 0V
$C_{I/O}$	I/O Capacitance	14	pF	$V_{I/O}$ = 0V
C_{IN}	Control Capacitance	14	pF	V_{IN} = 0V

NOTE: 4. This parameter is characterized and not 100% tested.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS

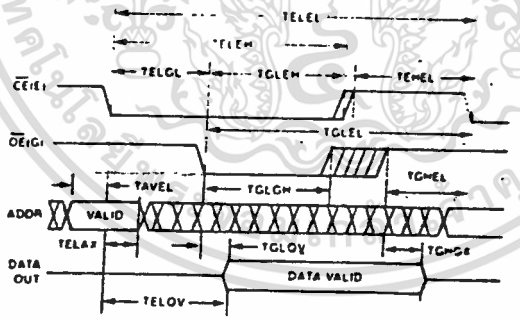
TA = 0°C to +70°C, VCC = +5V ± 10% unless otherwise noted

READ CYCLE ($\overline{WE}$ = VIH)

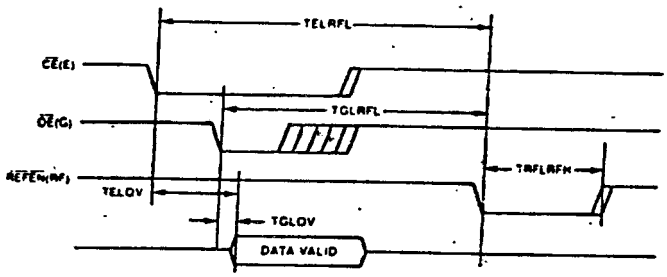
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELEL	Cycle Time	425		500		600		ns	
TELQV	Access Time from $\overline{CE}$		250		300		350	ns	
TGLQV	Access Time from $\overline{OE}$		65		70		75	ns	
TELEH	$\overline{CE}$ Pulse Width	40		40		40		ns	
TEHEL	$\overline{CE}$ High Time	40		40		40		ns	
TAVEL	Address Set-Up Time	0		0		0		ns	
TELAX	Address Hold Time	35		35		35		ns	
TGLEL	$\overline{OE}$ low to next $\overline{CE}$ low	250		275		300		ns	
TGLGH	$\overline{OE}$ Pulse Width	65		70		75		ns	
TGHEL	$\overline{OE}$ high to next $\overline{CE}$ low	40		40		40		ns	
TGHQX	$\overline{OE}$ high to Data Float	0	70	0	70	0	70	ns	1
TELGL	$\overline{CE}$ low to $\overline{OE}$ low		10,000		10,000		10,000	ns	
TGLEH	$\overline{OE}$ low to $\overline{CE}$ high	40		40		40		ns	
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TGLRFL	$\overline{OE}$ low to $\overline{REFEN}$ low	250		275		300		ns	
TRFLRFH	$\overline{REFEN}$ Pulse Width	70	9800	70	9800	70	9800	ns	5

WAVEFORMS

READ CYCLE



READ CYCLE FOLLOWED BY REFRESH



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS

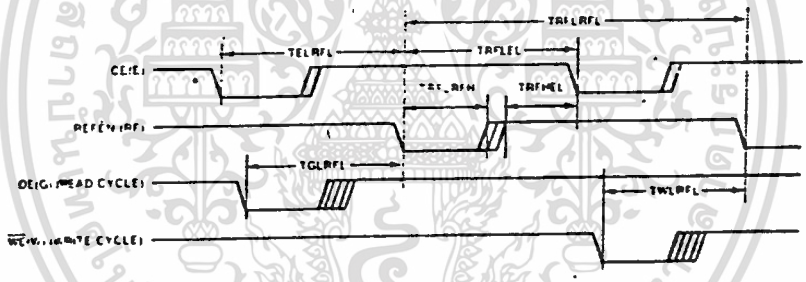
$T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 10\%$ unless otherwise noted.

REFRESH CYCLE

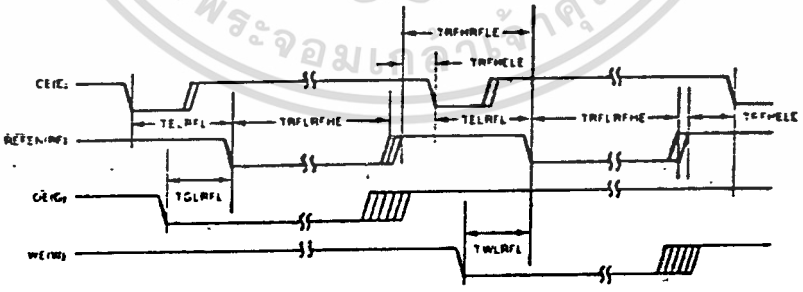
Symbol	Parameter	2187-25		2187-30		2187-35		Unit	Notes
		Min.	Max.	Min.	Max.	Min.	Max.		
TELRFL	$\overline{CE}$ low to $\overline{REFEN}$ low	425		500		600		ns	
TRFLRFH	$\overline{REFEN}$ Pulse Width	70	9800	70	9800	70	9800	ns	5
TRFHEL	$\overline{REFEN}$ high to $\overline{CE}$ low	40		40		40		ns	
TRFLRFL	$\overline{REFEN}$ cycle time to guarantee refresh	350	15600	400	15600	450	15600	ns	
TRFLEL	$\overline{REFEN}$ low to $\overline{CE}$ low	350		400		450		ns	
TRFHELE	$\overline{REFEN}$ high to $\overline{CE}$ low — extended cycle	350		400		450		ns	4
TGLRFL	$\overline{OE}$ low to $\overline{REFEN}$ low	250		275		300		ns	
TWLRFL	$\overline{WE}$ low to $\overline{REFEN}$ low	250		300		350		ns	
TRFLRFHE	$\overline{REFEN}$ Pulse Width — extended cycle	10,000		10,000		10,000		ns	4,5
TRFHRFLE	$\overline{REFEN}$ high to $\overline{REFEN}$ low — extended cycle	425		500		600		ns	

WAVEFORMS

REFRESH CYCLE



EXTENDED CYCLE REFRESH[5]



NOTES FOR A.C. CHARACTERISTICS:

1. Transition is measured ± 500 mV from steady state logic level with specified loading in Figure 2.
2. Maximum applies for P.M.C. Only
3. Note TEHELF > TEHEL.
4. TRFHELE > TRFHEL and TRFLRFHE > TRFLRFH
5. Extended cycles occur when the $\overline{REFEN}$ pulse width is ≥10 μsec.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

ADC0808, ADC0809 8-Bit μP Compatible A/D Converters With 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

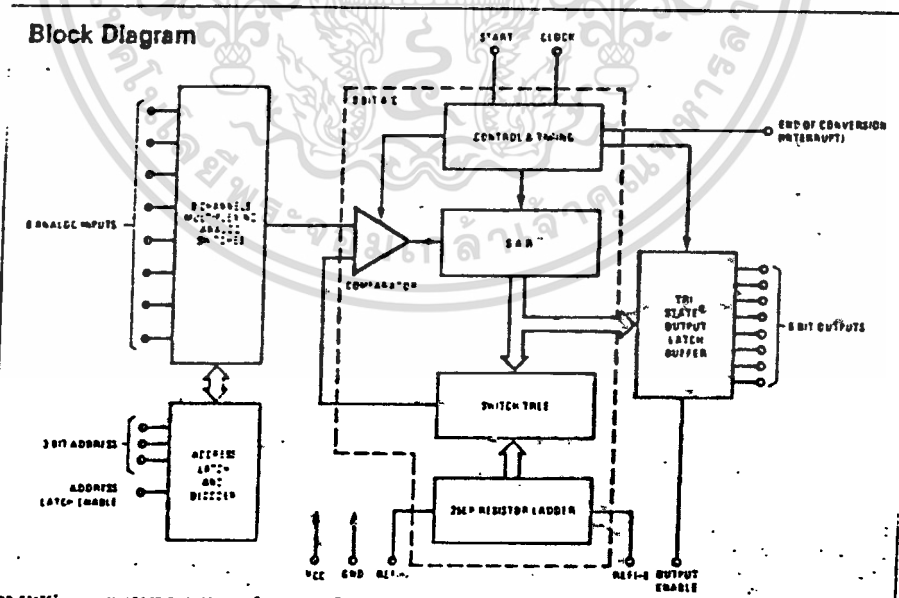
The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.

Features

- Resolution — 8-bits
- Total unadjusted error — ± 1/2 LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μs
- Single supply — 5 V_{DC}
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T^2L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range −40°C to +85°C or −55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE® output

Block Diagram



TRI-STATE® is a registered trademark of National Semiconductor Corp.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

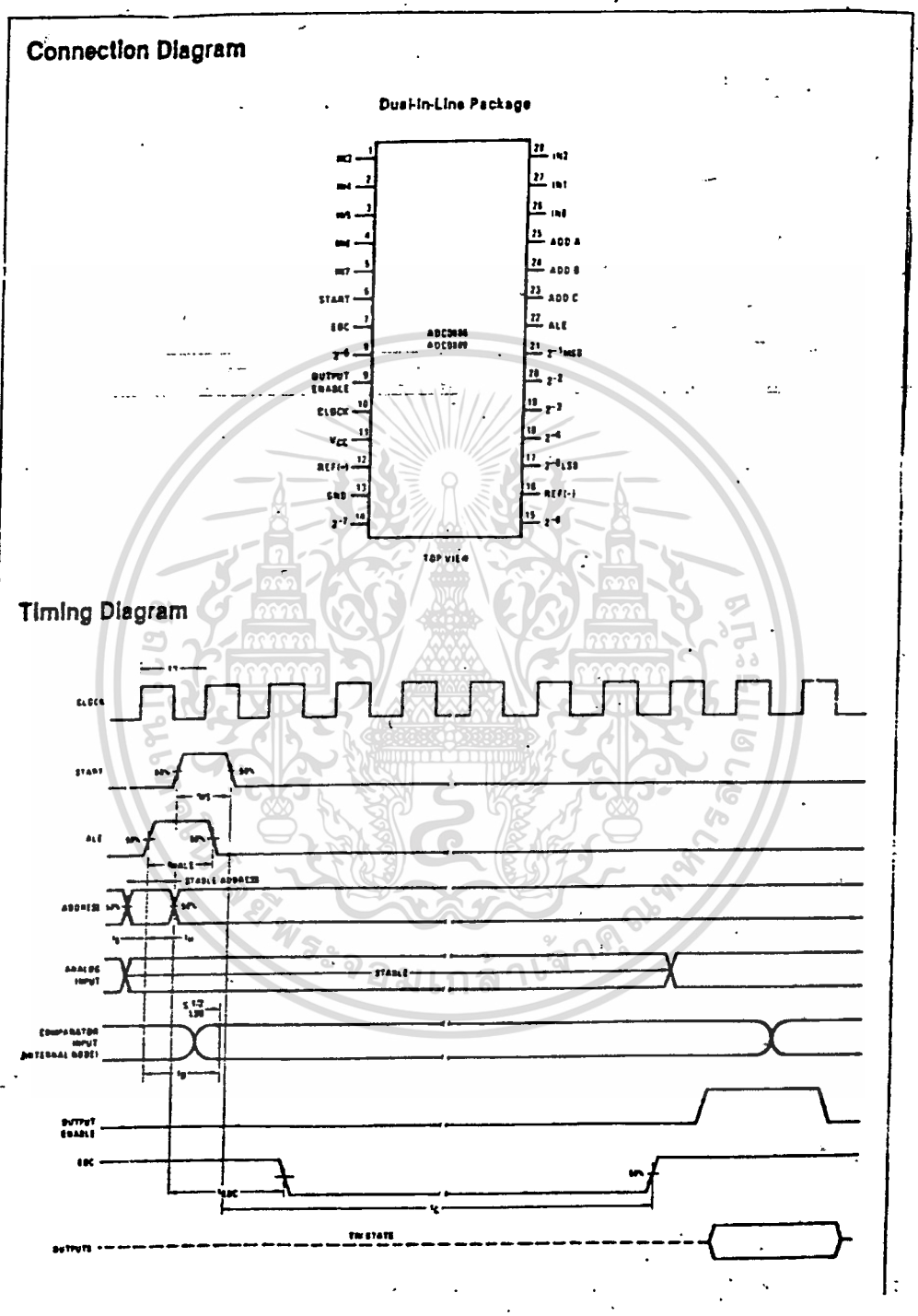


Figure 11.27 Connection diagram and timing relationships of the ADC0809 control signals. (Courtesy of National Semiconductor Corporation and Prentice-Hall, Inc.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้