



ภาควิชาครุศาสตร์วิศวกรรม
 คณะครุศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใบรับรองปริญญาโท

ชื่อหัวข้อ เครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
 Digital Storage Oscilloscope

ชื่อนักศึกษา 1. นายเจษฎา ถวิล รหัสประจำตัว 42035331
 2. นายพงศ์ศักดิ์ สิงห์ศิริสังกุล รหัสประจำตัว 42035340
 3. นายวินัย ดอกดวง รหัสประจำตัว 42035351
 4. นายสิโรช โดดเครือ รหัสประจำตัว 42035355

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์วรวิทย์ สมหา

อาจารย์ที่ปรึกษาร่วม อาจารย์ไพบุลย์ พวงวงศ์ตระกูล

| คณะกรรมการสอบปริญญาโท | ลายมือชื่อ |
|---------------------------------|------------|
| 1. อาจารย์วรวิทย์ สมหา | |
| 2. อาจารย์ไพบุลย์ พวงวงศ์ตระกูล | |
| 3. อาจารย์สุชิน อาจหาญ | |
| 4. อาจารย์อำพล ทองระอา | |
| 5. อาจารย์สุระชัย พิมพ์สวัสดิ์ | |

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 11.45 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
 ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่.....เดือน.....พ.ศ.....



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์

DIGITAL STORAGE OSCILLOSCOPE



นายเจษฎา ฤทธิ
นายพงศ์ศักดิ์ สิงห์ศิริสังกุล
นายวินัย ดอกดวง
นายสิโรช โดดเครือ

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

พ.ท.

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ท ๗๕๕ ค

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

๒๕๔๓

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๔๓

เลขที่.....
เลขทะเบียน 40158
วัน, เดือน, ปี 17 ส.ค. ๒๕๔๔

b.....11092191.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์

Digital Storage Oscilloscope

วัตถุประสงค์

1. เพื่อศึกษาวิธีการเขียนโปรแกรม Turbo C++ ควบคุมเครื่องวัดและเก็บบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
2. เพื่อออกแบบเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
3. เพื่อสร้างเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. เพื่อทดสอบการทำงานของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
5. เพื่อนำเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้ในการเขียนโปรแกรม Turbo C++ ในการควบคุมการทำงานของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
2. ได้ความรู้ในการออกแบบเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
3. ได้ชุดต้นแบบเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. ได้ความรู้ในการทดลองการทำงานของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
5. ได้เครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ที่สามารถนำไปใช้งานทางด้านอิเล็กทรอนิกส์ได้

| | | |
|----------------------|---|-----------------|
| ชื่อหัวข้อ | เครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ | |
| นักศึกษา | นายเจษฎา | ถวิล |
| | นายพงศ์ศักดิ์ | สิงห์สิริสังกุล |
| | นายวินัย | คอกดวง |
| | นายสิโรช | โตดเครือ |
| อาจารย์ที่ปรึกษา | อาจารย์วรวิทย์ | สมหา |
| อาจารย์ที่ปรึกษาร่วม | อาจารย์ไพบุลย์ | พวงวงศ์ตระกูล |
| หลักสูตร | ครุศาสตร์อุตสาหกรรมบัณฑิต | |
| สาขาวิชา | อิเล็กทรอนิกส์และคอมพิวเตอร์ | |
| ปีการศึกษา | 2543 | |

บทคัดย่อ

ปริญญาโทฉบับนี้ได้กล่าวถึงการออกแบบและสร้าง เครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ ที่สามารถใช้วัดสัญญาณแอนะล็อกที่มีความถี่ 0-20 กิโลเฮิร์ตซ์ โดยมีหลักการทำงานคือนำสัญญาณแอนะล็อกมาเปลี่ยนเป็นข้อมูลดิจิทัลแล้วส่งข้อมูลดิจิทัลผ่านทางพอร์ตขนานของเครื่องไมโครคอมพิวเตอร์เข้าไปเปรียบเทียบข้อมูล เพื่อนำไปพล็อตเป็นรูปสัญญาณด้วยโปรแกรมที่เขียนขึ้นจากภาษาซี++ โดยแสดงผลการทำงานที่จอมอนิเตอร์และสามารถเก็บบันทึกข้อมูลที่ต้องการลงในแผ่นดิสก์เก็ตได้

| | | |
|------------------------|---|-----------------|
| Thesis Title | Digital Storage Oscilloscope | |
| Students | Mr. Jessada | Thawin |
| | Mr. Phongsak | Singsirisajakul |
| | Mr. Winai | Dorkduang |
| | Mr. Sirote | Dotkrua |
| Advisor | Mr. Worawit | Somha |
| Co-Advisor | Mr. Paiboon | Pongwongtragull |
| Education Level | Bachelor of Science in Industrial Education | |
| Program in | Electronics and Computer | |
| Academic Year | 2000 | |

ABSTRACT

This thesis presents about designing and creation of Digital Storage Oscilloscope with can measure analog having about frequency 0-20 kHz. The working principle of Digital Storage-Oscilloscope is changing the analog signal into digital data. Then send this digital data through Microcomputer 's parallel port to compare the data in order to plot the signal with the program written in Turbo C++ by showing the result of work on monitor and can save the wanted data on the diskette.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ถูกล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกในกลุ่มทุกท่าน ขอขอบพระคุณอาจารย์ไพบูลย์ พวงวงศ์ตระกูล อาจารย์สุชิน อางหาญ อาจารย์โกศล ตราชู และคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ให้ความอนุเคราะห์ให้คำแนะนำ แนวความคิด ความรู้ต่างๆ ตลอดจนแนวทางในการแก้ปัญหา ในการจัดทำปริญญานิพนธ์ ขอขอบคุณเจ้าหน้าที่ห้องสมุดคณะครุศาสตร์อุตสาหกรรม เจ้าหน้าที่สำนักหอสมุดกลาง และเจ้าหน้าที่ห้องปริญญานิพนธ์คณะวิศวกรรมศาสตร์ที่ช่วยอำนวยความสะดวก และเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูลสุดท้ายที่ควรระลึกถึงเป็นอย่างยิ่ง คือ บิดา และมารดา ที่เป็นผู้ให้ความสนับสนุนด้านการศึกษาและเป็นผู้ให้กำลังใจด้วยดีตลอดมาตั้งแต่อดีตจนถึงปัจจุบัน



สารบัญ

| เรื่อง | หน้า |
|---|------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญรูป | VIII |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์ | 1 |
| 1.2 จิตความสามารถของโครงการ | 2 |
| 1.3 เนื้อหาโดยสังเขป | 2 |
| บทที่ 2 ทฤษฎี และหลักการ | 4 |
| 2.1 การรับและการแปลงข้อมูล | 4 |
| 2.1.1 ทฤษฎีการสุ่มตัวอย่าง | 5 |
| 2.1.2 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล | 9 |
| 2.2 หลักการของดิจิทัลสตอเรจออกสซิด โลส โคป | 12 |
| 2.2.1 การทำงานของดิจิทัลสตอเรจออกสซิด โลส โคป | 12 |
| 2.2.2 ขอบเขตการวัด | 14 |
| 2.2.3 การสุ่มตัวอย่างของดิจิทัลสตอเรจออกสซิด โลส โคป | 20 |
| 2.2.4 ขั้นตอนหลังการสุ่มตัวอย่าง | 23 |
| 2.2.5 การเก็บข้อมูลของดิจิทัลสตอเรจออกสซิด โลส โคป | 24 |
| 2.2.6 ปัญหาพื้นฐานของดิจิทัลสตอเรจออกสซิด โลส โคป | 25 |
| 2.2.7 ความละเอียดถูกต้องแม่นยำ | 25 |
| 2.2.8 สัญญาณชั่วขณะ (Transient Capture) | 26 |
| 2.2.9 แบนวิดธ์และอัตราการสุ่มตัวอย่าง | 28 |
| 2.2.10 ประโยชน์ของหน่วยความจำที่ยาวในดิจิทัลสตอเรจออกสซิด โลส โคป | 29 |

สารบัญ(ต่อ)

| เรื่อง | หน้า |
|---|------|
| บทที่ 3 การออกแบบ การสร้าง และการทำงาน | 30 |
| 3.1 การออกแบบวงจร | 30 |
| 3.2 ภาควงจรขยายและลดทอนแรงดัน (Attenuater) | 31 |
| 3.3 ภาควงแปลงสัญญาณแอนะล็อกเป็นดิจิตอล | 32 |
| 3.3.1 วงจรยกระดับสัญญาณ | 33 |
| 3.3.2 วงจรมัลติเพล็กซ์ | 33 |
| 3.3.3 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล | 33 |
| 3.4 วงจรแหล่งจ่ายไฟ | 34 |
| 3.5 โปรแกรมการทำงาน | 35 |
| บทที่ 4 การทดลองและผลการทดลอง | 37 |
| 4.1 การทดลองรับสัญญาณอินพุต | 37 |
| 4.1.1 การทดลองรับสัญญาณคลื่นไซน์ความถี่ 200 Hz | 37 |
| 4.1.2 การทดลองรับสัญญาณคลื่นไซน์ความถี่ 20 kHz | 38 |
| 4.1.3 การทดลองรับสัญญาณคลื่นรูปสามเหลี่ยมความถี่ 200 Hz | 38 |
| 4.1.4 การทดลองรับสัญญาณคลื่นรูปสามเหลี่ยมความถี่ 20 kHz | 39 |
| 4.1.5 การทดลองรับสัญญาณคลื่นรูปสี่เหลี่ยมความถี่ 200 Hz | 40 |
| 4.1.6 การทดลองรับสัญญาณคลื่นรูปสี่เหลี่ยมความถี่ 20 kHz | 40 |
| 4.2 การทดลองโปรแกรมควบคุมการทำงานของวงจขยายและลดทอนสัญญาณ | 41 |
| 4.2.1 การทดลองขยายสัญญาณคลื่นไซน์ขนาด 0.5 โวลต์ | 41 |
| 4.2.2 การทดลองลดสัญญาณคลื่นไซน์ขนาด 20 โวลต์ | 42 |
| 4.2.3 การทดลองขยายสัญญาณคลื่นรูปสามเหลี่ยมขนาด 0.5 โวลต์ | 43 |
| 4.2.4 การทดลองลดสัญญาณคลื่นรูปสามเหลี่ยมขนาด 20 โวลต์ | 43 |
| 4.2.5 การทดลองขยายสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 0.5 โวลต์ | 44 |
| 4.2.6 การทดลองลดสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 20 โวลต์ | 45 |

สารบัญ(ต่อ)

| เรื่อง | หน้า |
|---|------|
| บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา | 46 |
| 5.1 บทสรุป | 46 |
| 5.2 ปัญหาที่เกิดขึ้นในการจัดทำโครงการ | 46 |
| 5.3 การแก้ไข | 47 |
| 5.4 แนวทางในการพัฒนา | 47 |
| ภาคผนวก ก เครื่องต้นแบบ | 48 |
| ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์ | 51 |
| ภาคผนวก ค ผังงาน และโปรแกรมควบคุมการทำงาน | 57 |
| ภาคผนวก ง รายละเอียด และคุณสมบัติของอุปกรณ์ | 101 |
| บรรณานุกรม | 124 |
| ประวัติผู้แต่ง | 125 |



สารบัญตาราง

| ตาราง | หน้า |
|--|------|
| ตารางที่ 2.1 ความสัมพันธ์ระหว่างแรงดันอินพุตที่เป็นแอนะล็อกกับเอาต์พุตที่เป็นดิจิตอล | 10 |



สารบัญรูป

| รูป | หน้า |
|---|------|
| รูปที่ 2.1 ระบบควบคุมโดยการประมวลผลข้อมูลในระบบดิจิทัล | 4 |
| รูปที่ 2.2 วิธีการพื้นฐานของการแปลงสัญญาณแอนะล็อกเป็นดิจิทัล | 5 |
| รูปที่ 2.3 ทรานสเฟอร์ฟังก์ชันของคอมพาราเตอร์ | 5 |
| รูปที่ 2.4 ค่าผิดพลาดจากการวัดใน Aperture Time | 6 |
| รูปที่ 2.5 Spectrum ของสัญญาณแอนะล็อกที่จะถูกสุ่ม | 7 |
| รูปที่ 2.6 หลังจากการสุ่มเกิด Frequency Folding | 7 |
| รูปที่ 2.7 การเกิด Alias Frequency จากการสุ่มด้วยความถี่ต่ำกว่า 2 เท่าของสัญญาณแอนะล็อก | 8 |
| รูปที่ 2.8 การต่อวงจร Parallel Comparator A/D Converter | 10 |
| รูปที่ 2.9 แผนผังการทำงานของดิจิทัลสโตเรจอสซิลโลสโคป | 13 |
| รูปที่ 2.10 ลักษณะของการเกิดเอเลียสซิ่ง | 15 |
| รูปที่ 2.11 ทฤษฎีการสอดแทรก | 17 |
| รูปที่ 2.12 การจัดช่วงขอบขาขึ้น ซึ่งค่าคลาดเคลื่อนที่เกิดขึ้นอยู่กับตำแหน่งในการสุ่มตัวอย่าง | 19 |
| รูปที่ 2.13 เทคนิคการสุ่มตัวอย่างแบบต่างๆ ของดิจิทัลสโตเรจอสซิลโลสโคป | 21 |
| รูปที่ 2.14 ผลที่ออกมาจากการสุ่มตัวอย่างสัญญาณอินพุตแบบเป็นลำดับ | 22 |
| รูปที่ 2.15 การควอนไทซ์ ซึ่งใช้วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล คอนเวอร์เตอร์เปลี่ยนแรงดันแอนะล็อกเป็นเลขฐานสองที่ระดับต่างๆ | 23 |
| รูปที่ 2.16 แผนผังการทำงานของ RIS ดิจิไทเซอร์ | 27 |
| รูปที่ 2.17 แผนผังการทำงานของดิจิไทเซอร์การสุ่มตัวอย่าง | 28 |
| รูปที่ 3.1 แผนผังการทำงานของเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ | 30 |
| รูปที่ 3.2 วงจรขยายและลดทอนแรงดัน | 31 |
| รูปที่ 3.3 การต่อ MCS-51 เพื่อเลือกการทำงานของวงจร Attenuater | 32 |
| รูปที่ 3.4 วงจรยกระดับสัญญาณและวงจรมัลติเพล็กซ์ | 33 |
| รูปที่ 3.5 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล | 34 |
| รูปที่ 3.6 วงจรแหล่งจ่ายไฟ | 35 |
| รูปที่ 3.7 ผังงานการทำงานแสดงขั้นตอนการทำงาน | 36 |
| รูปที่ 4.1 สัญญาณอินพุตเป็นคลื่นไซน์ความถี่ 200 Hz | 37 |
| รูปที่ 4.2 สัญญาณอินพุตเป็นคลื่นไซน์ความถี่ 20 kHz | 38 |

สารบัญรูป(ต่อ)

| รูป | หน้า |
|---|------|
| รูปที่ 4.3 สัญญาณอินพุตเป็นคลื่นรูปสามเหลี่ยมความถี่ 200 Hz | 39 |
| รูปที่ 4.4 สัญญาณอินพุตเป็นคลื่นรูปสามเหลี่ยมความถี่ 20 kHz | 39 |
| รูปที่ 4.5 สัญญาณอินพุตเป็นคลื่นรูปสี่เหลี่ยมความถี่ 200 Hz | 40 |
| รูปที่ 4.6 สัญญาณอินพุตเป็นคลื่นรูปสี่เหลี่ยมความถี่ 20 kHz | 41 |
| รูปที่ 4.7 การขยายสัญญาณคลื่นไซน์ขนาด 0.5 โวลต์ | 42 |
| รูปที่ 4.8 การลดทอนสัญญาณคลื่นไซน์ขนาด 20 โวลต์ | 42 |
| รูปที่ 4.9 การขยายสัญญาณคลื่นรูปสามเหลี่ยมขนาด 0.5 โวลต์ | 43 |
| รูปที่ 4.10 การลดทอนสัญญาณคลื่นรูปสามเหลี่ยมขนาด 20 โวลต์ | 44 |
| รูปที่ 4.11 การขยายสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 0.5 โวลต์ | 45 |
| รูปที่ 4.12 การลดทอนสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 20 โวลต์ | 45 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญญาประดิษฐ์

ในสภาวะปัจจุบันความก้าวหน้าและการพัฒนาทางเทคโนโลยีของระบบอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ได้เปลี่ยนแปลงไปตลอดเวลา ซึ่งทั้งสองสิ่งนี้เป็นตัวกำหนดที่บ่งบอกถึงการพัฒนาของประเทศ แต่ถ้าสิ่งทั้งสองนี้ขาดอุปกรณ์ที่มีความพร้อมและถูกต้องตามมาตรฐานสากลแล้วก็จะไม่สามารถยืนยันความถูกต้องได้ และสิ่งที่จะนำมาใช้ในการทดสอบและเปรียบเทียบมาตรฐานของระบบอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ได้ ก็ต้องมีเครื่องมือทดสอบที่สมบูรณ์แบบ

ในยุคแรกของระบบเครื่องมือทดสอบนั้น จะเป็นระบบที่มีการแยกชุดทดสอบเป็นส่วนๆ ซึ่งต้องใช้เครื่องมือและอุปกรณ์ต่างๆ เป็นจำนวนมากเช่น มัลติมิเตอร์, เครื่องกำเนิดสัญญาณ และออสซิลโลสโคป เป็นต้น ซึ่งระบบเหล่านี้เป็นระบบที่ยุ่งยากและเป็นอุปสรรคในการจัดหาเครื่องมือทดสอบเพราะมีราคาสูง ดังนั้นในปัจจุบันนี้ได้มีการพัฒนาการใช้เครื่องมือทดสอบที่มีพื้นฐานมาจากเครื่องคอมพิวเตอร์ส่วนบุคคลหรือ PC ได้กลายเป็นวิธีที่ง่ายและให้ความสะดวกรวดเร็วกับผู้ใช้รวมทั้งผู้ที่มีความสนใจทั่วไป เนื่องจากการได้ประโยชน์จากการทดสอบที่ใช้เครื่องคอมพิวเตอร์ช่วยงาน โดยไม่ก่อให้เกิดความสูญเสียค่าใช้จ่ายในการพัฒนาโปรแกรมและเสียเวลามากนัก ซึ่งเครื่องมือเหล่านี้สามารถลดค่าใช้จ่ายในด้านแรงงาน, สามารถเพิ่มผลผลิต และสามารถจำกัดความผิดพลาดในการอ่านค่า และการวัดค่าที่เกิดขึ้นเนื่องจากความผิดพลาดของผู้ใช้ได้ โดยเครื่องมือทดสอบนี้จะใช้ประโยชน์จากความสามารถของซอฟต์แวร์ทำให้สามารถทำหน้าที่ต่างๆ ได้หลากหลายซึ่งในอดีตนั้นเมื่อจะทำหน้าที่ต่างๆ แบบนี้ต้องใช้เครื่องมือและอุปกรณ์ต่างๆ เป็นจำนวนมาก

โดยทั่วไปแล้วอุปกรณ์ที่ใช้กับเครื่องคอมพิวเตอร์ส่วนบุคคลที่ถูกออกแบบสำหรับการใช้ภายในจะประดิษฐ์ให้อยู่ในแผ่นอแคปเตอร์ การที่จะติดตั้งอุปกรณ์เชื่อมต่อภายในเครื่องทำได้โดยการเสียบแผ่นการ์ดเข้าไปในช่องสล๊อต ที่วางไว้บนแผ่นเมนบอร์ดและระบบก็ป้อนไฟเลี้ยงตามปกติ หรืออีกวิธีหนึ่งคือ การประยุกต์ใช้พอร์ตพรินเตอร์เพื่อติดตั้งอุปกรณ์ทดสอบให้เชื่อมต่อกับเครื่องคอมพิวเตอร์ ซึ่งเป็นวิธีที่ดีเนื่องจากไม่ต้องเสียบแผ่นการ์ดเข้าไปในช่องสล๊อต แต่สามารถนำอุปกรณ์ทดสอบมาต่อกับสายที่นำมาต่อกับพรินเตอร์ได้นั่นเอง จากนั้นการใช้อุปกรณ์ทดสอบนี้ก็ถูกควบคุมโดยซอฟต์แวร์ที่ถูกผู้ใช้สั่งงานมาจากแป้นพิมพ์ หรือเมาส์ของเครื่องคอมพิวเตอร์ ไม่มีปุ่มลูกบิด, สวิตช์หรือส่วนแสดงผลอื่นๆ กับผู้ใช้อีกต่อไป การทำงานของเครื่องทดสอบทั้งหมด

จะทำงานโดยผ่านคอมพิวเตอร์และอินเทอร์เน็ตของตัวเอง ซึ่งประโยชน์และเทคโนโลยีอันทันสมัยที่ได้กล่าวมาแล้วเป็นแรงจูงใจและเกิดความร่วมมือกันในกลุ่มโครงการเพื่อที่จะศึกษาเทคโนโลยีเหล่านั้นนำมาประยุกต์เพื่อจะได้เป็นประโยชน์ในการศึกษา และเป็นการช่วยให้สามารถประหยัดค่าใช้จ่ายในการสั่งซื้อเครื่องมือทดสอบจากต่างประเทศ เพราะอุปกรณ์เหล่านั้นมีราคาแพงมาก ซึ่งทำให้ผู้ใช้ไม่สามารถซื้อมาใช้กันอย่างทั่วไปได้

จากความต้องการดังกล่าวมาทั้งหมดได้ก่อให้เกิดความคิดว่า น่าจะทำโครงการเครื่องมือทดสอบที่ใช้ความสามารถพื้นฐานของคอมพิวเตอร์ ในส่วนของเครื่องออสซิลโลสโคปที่ใช้วิเคราะห์ซึ่งมีชื่อเรียกว่า เครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ โดยมีข้อดีมากกว่าออสซิลโลสโคปธรรมดาทำ คือสามารถที่จะเก็บข้อมูลของสัญญาณที่วัดได้เพื่อนำมาวิเคราะห์ในภายหลัง

1.2 ขีดความสามารถของโครงการ

- 1) สามารถวัดสัญญาณแอนะล็อกที่มีความถี่ได้ 0- 20 กิโลเฮิร์ตซ์
- 2) สามารถรับสัญญาณทางอินพุตได้ 2 ช่อง
- 3) สามารถเก็บบันทึกรูปสัญญาณที่วัดได้
- 4) สามารถควบคุมการทำงานของเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ผ่านโปรแกรมที่แสดงบนจอภาพ
- 5) ติดต่อส่งข้อมูลผ่านทางพอร์ตขนานของคอมพิวเตอร์

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญานិพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการประกอบด้วยเนื้อหาดังนี้ คือ การรับและการแปลงข้อมูล ทฤษฎีการสุ่มตัวอย่าง การแปลงสัญญาณแอนะล็อกเป็นดิจิตอล หลักการและการทำงานของดิจิตอลออสซิลโลสโคป

บทที่ 3 การออกแบบ และการสร้าง กล่าวถึงเนื้อหาเกี่ยวกับวงจรที่ใช้ในการทำงานคือวงจรรขยายและลดทอนสัญญาณ, วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอลและวงจรระดับแรงดัน

บทที่ 4 การทดลอง และผลการทดลอง ประกอบด้วย การทดลองการใช้งานเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ โปรแกรมส่วนแสดงผลทางหน้าจอ, การควบคุมเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ผ่าน โปรแกรมที่แสดงบนจอภาพ

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา ขั้นการสรุปผล ในการจัดทำโครงการ ปัญหาที่เกิดขึ้น และได้เสนอแนวทางแก้ไขปัญหา รวมทั้งแนวทางการพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์

ภาคผนวก ค ผังงาน และ โปรแกรมควบคุมการทำงาน

ภาคผนวก ง รายละเอียด และคุณสมบัติของอุปกรณ์

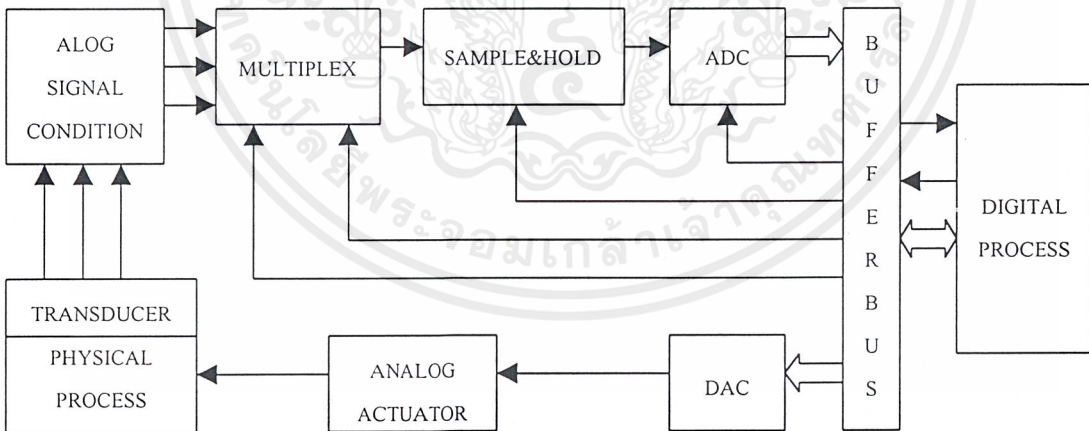


บทที่ 2

ทฤษฎีและหลักการ

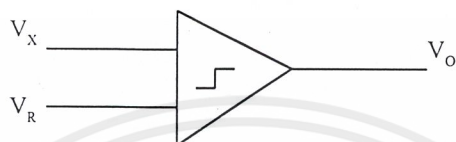
2.1 การรับและการแปลงข้อมูล

รูปแบบสัญญาณไฟฟ้าที่เราสามารถพบเห็นและคุ้นเคยในชีวิตประจำวัน จะอยู่ในรูปแบบของสัญญาณที่ต่อเนื่องกัน หรือเรียกว่าสัญญาณแอนะล็อก ซึ่งแต่เดิมการนำสัญญาณไฟฟ้างดกล่าวมาประมวลผล ใช้ในรูปแบบที่มีประโยชน์จะกระทำในแบบแอนะล็อกนั่นเอง แต่เมื่อเทคนิคการประมวลผลสัญญาณเชิงดิจิทัลได้รับการพัฒนาขึ้นมาเนื่องจากพบว่าในรูปแบบของดิจิทัลการประมวลผล, เก็บ, สื่อสาร และแสดงกระทำได้ง่าย และมีประสิทธิภาพมากกว่า ดังนั้นการเปลี่ยนรูปแบบสัญญาณ จึงได้มีความจำเป็นขึ้นจากสัญญาณแอนะล็อกที่มีอยู่ตามธรรมชาติถูก เปลี่ยนมาเป็นสัญญาณดิจิทัล และถูกประมวลผลโดยตัวประมวลผลทางดิจิทัล เช่น คอมพิวเตอร์ จากนั้นจึงจะถูกนำกลับมาแสดงผล หรือถูกเปลี่ยนกลับมาอยู่ในรูปแบบแอนะล็อก และใช้งานได้ง่ายกว่า โดยจะใช้วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก (Digital to Analog Converters) หรือเรียกสั้นๆ ว่า “DAC” ในรูปที่ 2.1 จะแสดงระบบควบคุมโดยการประมวลผลข้อมูลในระบบดิจิทัล

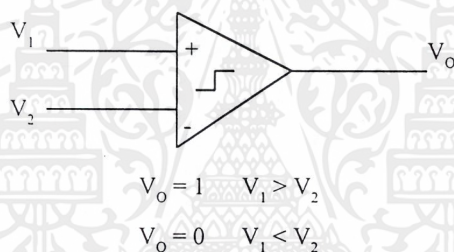


รูปที่ 2.1 ระบบควบคุมโดยการประมวลผลข้อมูลในระบบดิจิทัล

วิธีการแปลงสัญญาณแอนะล็อกเป็นดิจิทัลแบบง่าย ๆ ดังแสดงในรูปที่ 2.2 โดยใช้หลักการของวงจรคอมพาราเตอร์ ค่าแรงดันอินพุตที่ไม่ทราบค่า V_x จะต่อเข้ากับขาอินพุตขาหนึ่งของแอนะล็อกคอมพาราเตอร์ และแรงดันอ้างอิงที่ขนาดแปรตามเวลา V_R ต่อเข้ากับอินพุตอีกขาหนึ่งของคอมพาราเตอร์ ลักษณะของทรานสเฟอร์ฟังก์ชันของคอมพาราเตอร์ แสดงในรูป 2.3 ถ้าแรงดันอินพุต V_1 มากกว่าอินพุต V_2 แล้วแรงดันเอาต์พุตจะเป็น “1” ถ้าอินพุต V_1 มากกว่า V_2 แล้ว เอาต์พุตจะได้เป็น “0”



รูปที่ 2.2 วิธีการพื้นฐานของการแปลงสัญญาณแอนะล็อกเป็นดิจิทัล



รูปที่ 2.3 ทรานสเฟอร์ฟังก์ชันของคอมพาราเตอร์

วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล ที่มีการใช้งานกันทั่วไปมีหลายชนิด เช่น Counter type ADC, Integrating ADC, Successive Approximation ADC หรือ Parallel ADC เป็นต้น

2.1.1 ทฤษฎีการสุ่มตัวอย่าง

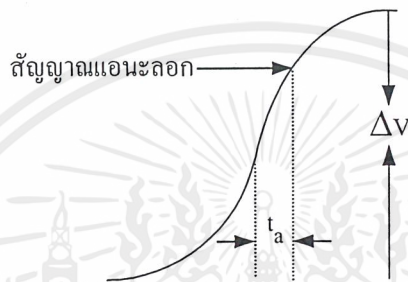
ในขั้นการดำเนินการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลนั้น วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลจะต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวขึ้นอยู่กับตัวแปรหลายตัวเช่น

1. ความละเอียดในการเปลี่ยนสัญญาณ
2. เทคนิคของการแปลงสัญญาณ
3. ความเร็วในการทำงานของอุปกรณ์อื่นๆ ความเร็วของการแปลงสัญญาณนี้จำเป็นสำหรับการใช้งานเฉพาะอย่างและความแม่นยำที่ต้องการ

Aperture Time : คือช่วงเวลาในการแปลงสัญญาณซึ่งคำว่า Aperture Time โดยทั่วไปหมายถึงช่วงเวลาที่เกิดความไม่แน่นอนในการวัดและผลก็คือเกิดความผิดพลาดต่อค่าที่วัดได้

ในรูปที่ 2.4 สัญญาณแอนะล็อก $V(t)$ มีอัตราการเปลี่ยนแปลง dV/dt ช่วง Aperture Time t_a ดังนั้นช่วงการเปลี่ยนแปลงสัญญาณแอนะล็อกจะเท่ากับ ΔV ดังสมการที่ 2.1

$$\Delta V = t_a \frac{dV(t)}{dt} \quad (2.1)$$



รูปที่ 2.4 ค่าผิดพลาดจากการวัดใน Aperture Time

ดังนั้นหากเวลาที่วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล ใช้ในการเปลี่ยนสัญญาณในเวลา t_a นี้ รหัสของสัญญาณที่ได้ อาจตรงกันข้ามกับขนาดของสัญญาณแอนะล็อกค่าใดค่าหนึ่งในช่วงนี้ และส่วนอื่นๆที่เหลือคือความผิดพลาดที่เกิดขึ้น ซึ่งแน่นอนในบางครั้งเป็นไปได้ที่รหัสของสัญญาณ ดิจิตอลจะตรงกับค่าของสัญญาณแอนะล็อกที่ถูกต้อง

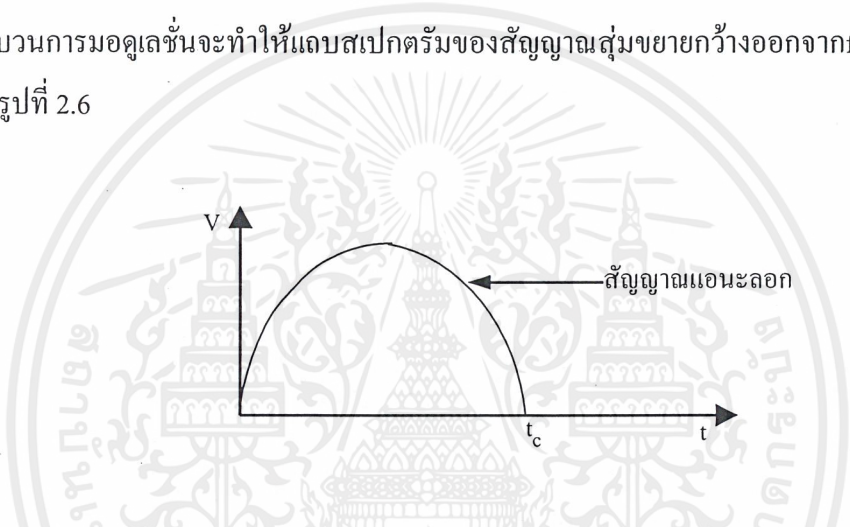
1) Sample and Hold Aperture error

วงจร Sample and Hold จะทำการสุ่มสัญญาณอินพุต และนำสัญญาณนั้นมาเก็บไว้ในเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้เวลาการประจุแรงดันนั้นในตัวเก็บประจุที่รั่วไหลค่า ดังนั้นในเมื่อแรงดันอินพุตยังคงอยู่ได้นานพอทำให้วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลไม่จำเป็นต้องมีเวลาในการแปลง (Conversion Time) อย่างรวดเร็วนัก Aperture Time ของ Sample and Hold คือเวลาดั้งแต่เวลาที่เริ่มสุ่มสัญญาณจนตัวเก็บประจุมีค่าแรงดันจนถึงค่าที่สุ่ม ซึ่งสำหรับ Sample and Hold แล้ว Aperture Time ขึ้นอยู่กับแบนวิดและสวิตซ์ิงใหม่ของอุปกรณ์เอคทีฟที่ใช้ในวงจรซึ่งหาและสร้างได้ง่ายและราคาถูกกว่าการสร้างวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลความเร็วสูง

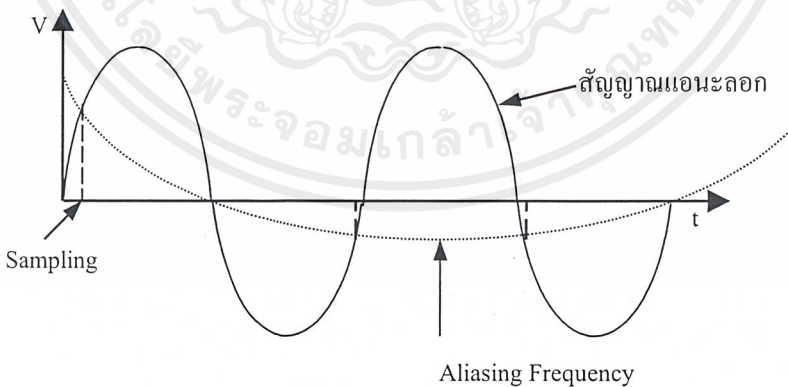
มีปัญหาว่าอัตราการสุ่มสัญญาณนั้นควรจะมีค่าเท่าใด ที่จะไม่ทำให้ข้อมูลเสียหายไปเมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเช่นเดิม อันนี้ขึ้นอยู่กับความถี่ของสัญญาณแอนะล็อก และทฤษฎีการสุ่มที่กล่าวไว้ว่า “ถ้าสัญญาณต่อเนื่องที่มีความถี่และฮาโมนิกส์ ไม่เกิน f_c สัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาได้อย่างเดิมโดยไม่สูญเสียรายละเอียดหรือเพี้ยนไปถ้าอัตราการสุ่มไม่น้อยกว่า $2f_c$ ต่อวินาที”

2) Frequency Folding and Aliasing

จากทฤษฎีการสุ่มสามารถอธิบายลักษณะรูปสเปกตรัม ของสัญญาณในรูปที่ 2.5 แสดงให้เห็นสเปกตรัมของสัญญาณที่ถูกสุ่มซึ่งแบนวิดไม่เกินกว่า f_s ในขณะที่สัญญาณนี้จะถูกสุ่มด้วยความถี่ ขบวนการมอดูเลชันจะทำให้แถบสเปกตรัมของสัญญาณสุ่มขยายกว้างออกจาก f_s เป็น $2f_s$, $3f_s$... ได้ดังรูปที่ 2.6



รูปที่ 2.5 Spectrum ของสัญญาณแอนะล็อกที่จะถูกสุ่ม



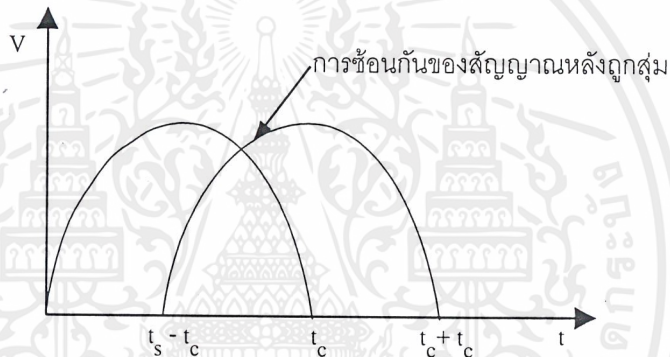
รูปที่ 2.6 หลังจากการสุ่มเกิด Frequency Folding

ถ้าความถี่ของสัญญาณสุ่ม f_s ไม่สูงพอหลังจากการสุ่มสเปกตรัมบางส่วนของ f_s จะมาซ้อนทับกับสเปกตรัมของสัญญาณซึ่งเรียกว่า Frequency Folding หากเป็นเช่นนี้ก็จะทำให้เกิดความเพี้ยนแก่สัญญาณแวนะลอกจากการซ้อนกันของสเปกตรัม เมื่อสัญญาณถูกเปลี่ยนกลับให้อยู่ในรูปเดิมและถ้าเลื่อนความถี่ของการสุ่มให้สูงขึ้น จนโอกาสการซ้อนกันของสเปกตรัมหมดไปจะทำให้การเปลี่ยนของสัญญาณหลังจากการถูกสุ่มก็ยังคงเหมือนเดิมได้

จากที่กล่าวมาแสดงการสนับสนุนทฤษฎีการสุ่มที่ว่าให้ $f_s < f_c$ นั่นคือการกำจัดการซ้อนกันของสเปกตรัมทำได้สองวิธีคือ

1. ใช้อัตราการสุ่มที่สูงพอ
2. การทำการกรองความถี่ของสัญญาณแวนะลอกก่อนการสุ่มเพื่อให้แบนวิด ไม่เกินไปกว่า

$f_s / 2$



รูปที่ 2.7 การเกิด Alias Frequency จากการสุ่มด้วยความถี่ต่ำกว่า 2 เท่าของสัญญาณแวนะลอก

ในทางปฏิบัติแล้วจะยังคงเกิด Frequency Folding ได้เสมอจากส่วนฮาร์โมนิกของสัญญาณรวมทั้งสเปกตรัมของสัญญาณรบกวนที่ยังคงอยู่ แม้ว่าจะทำการกรองความถี่ก่อนหน้ามาแล้วก็ตาม การกำจัดการซ้อนกันของสเปกตรัมนี้ วิธีที่ได้ผลคือพยายามให้การสุ่มสัญญาณเป็นไปอย่างรวดเร็วที่สุด

ผลของการใช้อัตราการสุ่มที่ไม่เหมาะสมอีกประการหนึ่งเกิดขึ้น ดังรูปที่ 2.7 เรียกว่า Alias Frequency ซึ่งเกิดกับสัญญาณที่เปลี่ยนกลับมาเช่นเดิมหลังจากถูกสุ่มแล้ว

2.1.2 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

ในการติดต่อสื่อสารกันระหว่างระบบแอนะล็อกและระบบดิจิทัลนั้นเป็นไปได้โดยที่จะให้สัญญาณแอนะล็อกเข้ามาทำงานในระบบดิจิทัล ดังนั้นจึงต้องมีการแปลงสัญญาณแอนะล็อกให้เป็นดิจิทัล ส่วนที่ทำหน้าที่ดังกล่าวนี้เราเรียกว่า “วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล” สำหรับดิจิทัลสโปกแล้วจะมีวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลที่เข้ามาทำหน้าที่เกี่ยวกับกลุ่มสัญญาณที่ได้จากการควอนไทซ์ซึ่งอยู่ 4 แบบด้วยกันคือ

1) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบใช้วงจรเปรียบเทียบ

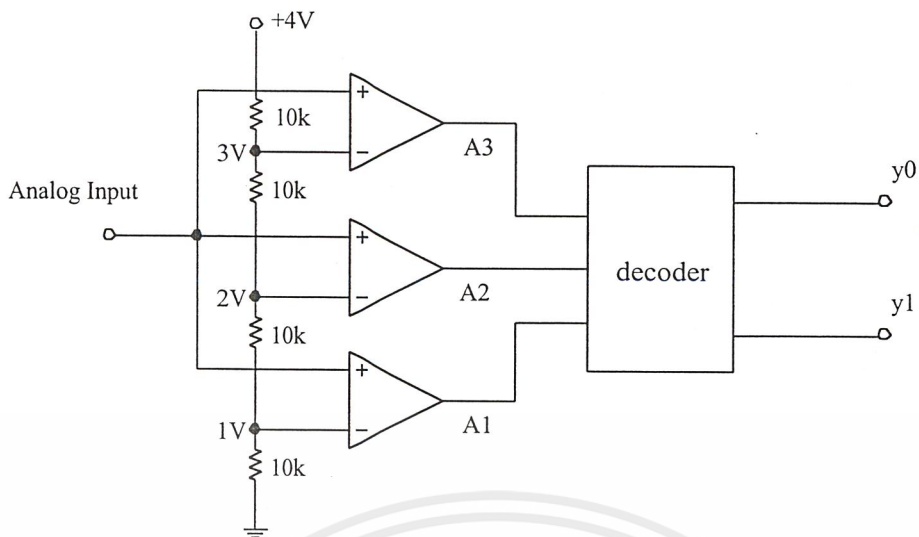
สำหรับการแปลงสัญญาณที่ต้องการความเร็วสูงมากๆ เช่นการแปลงสัญญาณภาพโทรทัศน์, เรดาร์ จำเป็นต้องใช้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบพิเศษ ที่เรียกว่า Flash ADC ซึ่งจะแสดงในรูปที่ 2.8

วงจรแบบนี้เป็นแบบที่นิยมใช้กันมากเนื่องจากมีความเร็วในการทำงานสูงมาก จะเห็นว่าการทำงานแบบนี้ประกอบด้วยตัวต้านทานต่อเป็นตัวแบ่งแรงดัน มีออปแอมป์ต่อเป็นวงจรเปรียบเทียบและมีวงจรถอดรหัส ซึ่งเป็นตัวกำหนดจำนวนของตัวเลขฐานสองที่จะใช้เป็นเอาต์พุต ออกมาเพียง 4 บิต เท่านั้น

สำหรับสัญญาณที่ใช้ในวงจรนี้จะมีสัญญาณอินพุตที่จะเข้ามาเปรียบเทียบกับระดับแรงดันอ้างอิงหรือระดับแรงดันไบแอส ซึ่งได้มาจากแรงดันตกคร่อมตัวต้านทาน และจะถูกนำไปใช้เป็นส่วนหนึ่งของวงจรเปรียบเทียบ เพื่อการเปรียบเทียบกับสัญญาณที่เข้ามา

เมื่อแรงดันอินพุตเข้ามามีขนาดใหญ่กว่าระดับแรงดันอ้างอิง เอาต์พุตของวงจรเปรียบเทียบก็จะมีค่าสูง และหากแรงดันอินพุตที่เข้ามามีค่าต่ำกว่าระดับแรงดันอ้างอิง เอาต์พุตของวงจรเปรียบเทียบก็จะมีค่าต่ำ แรงดันเอาต์พุตที่ได้จากวงจรเปรียบเทียบ ซึ่งจะมีระดับแรงดันแตกต่างกันก็จะเข้าสู่การถอดรหัสเพื่อจะให้ได้อาต์พุตออกมาเป็นเลขฐานสอง

ในการทำงานของระบบนี้จะมีการทำงานด้วยความเร็วสูงมาก และจะให้เอาต์พุตเลขฐานสองออกมาอย่างสม่ำเสมอด้วย แต่มีได้หมายความว่าคอนเวอร์เตอร์แบบนี้จะไม่มีข้อเสียเลยสำหรับความบกพร่องของคอนเวอร์เตอร์แบบนี้ก็คือ ประสิทธิภาพด้านความละเอียดจะลดลงทันทีเมื่อการสุ่มตัวอย่างมีการเปลี่ยนแปลงไป



รูปที่ 2.8 การต่อวงจร Parallel Comparator A/D Converter

หลักการทำงานคือ ใช้คอมพาราเตอร์ทำการเปรียบเทียบสัญญาณแอนะล็อกอินพุตกับแรงดันอ้างอิงที่แบ่งแรงดันให้สอดคล้องกับรหัสดิจิทัล โดยใช้ตัวต้านทานและแปลงเอาต์พุตจากคอมพาราเตอร์ให้ตรงกับรหัสดิจิทัล ซึ่งจะเห็นว่าอุปสรรคทางด้านความเร็วถูกจำกัดเพียง Propagation Time ของคอมพาราเตอร์เท่านั้นแต่อุปสรรคสำคัญต่อการพัฒนาวงจรบนไอซี คือ วงจรนี้ต้องการคอมพาราเตอร์ จำนวน $2n-1$ ตัวซึ่งเป็นจำนวนที่มากพอสมควร ในตารางที่ 2.1 แสดงการเปรียบเทียบสัญญาณอินพุตและผลของสัญญาณดิจิทัลที่ได้

ตารางที่ 2.1 ความสัมพันธ์ระหว่างแรงดันอินพุตที่เป็นแอนะล็อกกับเอาต์พุตที่เป็นดิจิทัล

| แรงดันอินพุต (โวลต์) | เอาต์พุตของวงจรเปรียบเทียบ | | | เอาต์พุตเลขฐานสอง | |
|-------------------------|----------------------------|----|----|-------------------|----|
| | A1 | A2 | A3 | D1 | D2 |
| 0-1 | 0 | 0 | 0 | 0 | 0 |
| 1-2 | 1 | 0 | 0 | 0 | 1 |
| 2-3 | 1 | 0 | 1 | 1 | 0 |
| 3-4 | 1 | 1 | 1 | 1 | 1 |

2) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบประมวลค่า

เป็นการประมวลค่าอย่างต่อเนื่อง โดยการเปรียบเทียบแรงดันอินพุตกับเอาต์พุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล จากนั้นจะมีการเลือกตำแหน่งที่จะนำมาใช้เป็นบิตสูงสุด (MSB) สำหรับเอาต์พุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล และในการทำงานของคอนเวอร์เตอร์แบบนี้ จะมีลำดับขั้นตอนในการทำงานแต่ละขั้นตอน โดยใช้เวลาที่คงที่ จากผลดังกล่าวจะมีผลกระทบต่อความละเอียดและความเร็วของวงจร ซึ่งนั่นหมายความว่าหากสัญญาณที่มีรายละเอียดมากๆ แล้วการแปลงสัญญาณก็จะต้องใช้เวลามากพอสมควร

3) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบ CCD

CCD สามารถตอบสนองการทำงานเมื่อมีค่าที่ได้จากการสุ่มตัวอย่างที่เร็วกว่า 100 MS/s ได้ โดยคงความละเอียดไว้ได้ตามความเป็นจริงแล้ว CCD นั้น ไม่ใช่วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลอย่างแท้จริง แต่จะเป็นลักษณะของการสุ่มตัวอย่างแบบแอนะล็อกเสียมากกว่า โดยการสุ่มตัวอย่างแบบนี้จะมีการทำงานเร็วพอสมควร ซึ่งลักษณะการทำงานจะอาศัยอุปกรณ์ที่เรียกว่า Bucket Brigade

การทำงานของ Bucket Brigade มีลักษณะดังนี้ เมื่อการสุ่มตัวอย่างมีค่าต่ำลงส่วนที่เรียกว่า Bucket Cell จะถูกชาร์ตประจุทันที ผลจากการชาร์ตนี้จะทำให้เกิดระดับแรงดันขึ้นระหว่างช่วงของการสุ่มตัวอย่างที่เข้ามา ก่อให้เกิดเป็นจุดเสริมเข้าไปในช่วงของจุดที่ได้จากการสุ่มตัวอย่าง ผลคือทำให้ได้ระดับสัญญาณที่ต้องการแปลงเพิ่มขึ้น ภาพที่ได้จากกรณีนี้จะมีรายละเอียดและจำนวนเซลล์หลายๆ เซลล์ที่มีอยู่ใน CCD จะเป็นตัวควบคุมการทำงานด้วย

นอกจากที่จะสามารถทำงานได้มากกว่า 100 MS/s แล้ว เมื่อเทียบราคาแบบ CCD จะถูกกว่าแบบแฟลชมาก ความละเอียดของสัญญาณที่ได้ก็จะไม่เปลี่ยนตามอัตราการสุ่มตัวอย่าง แต่ CCD นี้ก็มีข้อเสียตรงที่ว่าสโคปจะไม่สามารถรับข้อมูลใดๆ ได้ระหว่างที่ CCD ทำงานอยู่

4) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลแบบสแกน (Scan conversion)

ในการทำงานจะอาศัยการทำงานของหลอด CRT แบบดับเบิลเอนด์ (Double-ended CRT) ซึ่งหลอดชนิดนี้มีความสามารถที่จะเก็บข้อมูลไว้บนด้านข้างของทาร์เก็ต และเมื่อต้องการที่จะอ่านข้อมูลออกมาก็สามารถทำได้ โดยใช้ลำอิเล็กตรอนอีกชุดหนึ่งกวาดลงไปบนด้านหลังของทาร์เก็ต วิธีการแบบนี้จะมีการทำงานที่รวดเร็วมก แต่มีราคาแพง

2.2 หลักการของดิจิทัลสตอเรจออสซิลโลสโคป

ดิจิทัลสตอเรจออสซิลโลสโคป ได้รับการออกแบบให้มีขีดความสามารถในการตอบสนองความต้องการของผู้ใช้ ซึ่งไม่สามารถทำได้ในแอนะล็อกสโคปแบบเดิมๆ ที่มีอยู่ อาทิเช่น การเก็บข้อมูลของสัญญาณที่วัดได้เพื่อการนำมาวิเคราะห์ในภายหลัง การวิเคราะห์รูปคลื่นอย่างละเอียดโดยการซูมรูปคลื่นส่วนที่ต้องการการจับสัญญาณที่เกิดขึ้นเพียงชั่วขณะ การทำปรีทริกเกอร์ และการตรวจจับสัญญาณที่มีการเปลี่ยนแปลงทางความถี่อย่างรวดเร็วหรือช้าเกินไป ซึ่งไม่สามารถตรวจจับได้ด้วยแอนะล็อกสโคปแบบปกติ

จุดสำคัญของสโคปแบบนี้อยู่ที่วิธีการในการเก็บข้อมูลและวิธีการทางดิจิทัลที่นำมาใช้ โดยปกติแล้วข้อมูลที่ถูเก็บเอาไว้สามารถนำไปใช้บันทึกลงในเครื่องคอมพิวเตอร์ได้และยังสามารถต่อพ่วงพล็อตเตอร์หรือพรินเตอร์ได้ด้วย

2.2.1 การทำงานของดิจิทัลสตอเรจออสซิลโลสโคป

สัญญาณแอนะล็อกและสัญญาณดิจิทัลนั้นมีการนำมาเปรียบเทียบกันอยู่เสมอ ในเรื่องของสโคปก็เช่นเดียวกันของสตอเรจออสซิลโลสโคปนั้น ได้ถูกประดิษฐ์ขึ้นมาทั้งแบบแอนะล็อกและแบบดิจิทัล สำหรับแอนะล็อกสตอเรจออสซิลโลสโคปนั้น จะมีวิธีการเก็บบันทึกที่รูปคลื่นแบบแอนะล็อกนั่นคือ การใช้หลอดสโตเรจ (Storage CRT) ซึ่งหลอดนี้มีความสามารถทำให้สัญญาณที่เกิดขึ้นเพียงชั่วขณะแสดงผลบนจอได้นานเป็นเวลา 1 ชั่วโมงหรือมากกว่า โดยวิธีการในการควบคุมแรงไฟและอิเล็กทรอนิกส์ภายในหลอด

แต่การใช้หลอดดังกล่าวจะมีอายุการใช้งานเพียง 1,000 ชั่วโมงเท่านั้น หากนำไปใช้แสดงผลตามปกติก็จะใช้งานได้ยาวนานกว่า

ส่วนดิจิทัลสตอเรจออสซิลโลสโคปจะมีวิธีเก็บบันทึกที่รูปคลื่นด้วยระบบดิจิทัล ข้อมูลจะถูกบันทึกไว้ในหน่วยความจำแบบดิจิทัล ซึ่งจะเห็นว่าเมื่อเปรียบเทียบกับแอนะล็อกแล้วแบบแอนะล็อกจะเก็บรูปคลื่นด้วยวงจร นี่คือนี่ที่แตกต่างกัน

สำหรับดิจิทัลสตอเรจออสซิลโลสโคปจะมีขบวนการในการทำงานอยู่ด้วยกัน 3 ช่วงใหญ่ๆ คือ ช่วงของการสุ่มตัวอย่างและการแปลงเป็นสัญญาณดิจิทัล, ช่วงของการเก็บข้อมูลและสุดท้ายคือช่วงของการแสดงผลค่าต่างๆ

เริ่มต้นด้วยการสุ่มตัวอย่างสัญญาณที่เข้ามาเพื่อให้ได้จำนวนของจุดบนสัญญาณเหล่านี้จากนั้นจะเป็นขั้นตอนการนำค่าที่ได้จากการสุ่มตัวอย่างมาแปลงเป็นสัญญาณดิจิทัลด้วยวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล หากเป็นวงจรที่ใช้สัญญาณดิจิทัล 8 บิต ก็จะได้ระดับของสัญญาณที่แปลงออกมา 2^8 หรือ 256 ระดับ

2.2.2 ขอบเขตการวัด

ทุกสิ่งทุกอย่างย่อมมีขีดจำกัดในตัวของมันเองเมื่อนำสิ่งนั้นมาใช้งานเกินความสามารถของมันผลที่ได้ก็จะไม่ถูกต้องสมบูรณ์ ดิจิตอลสโตเรจอสซิดโลสโคปก็เช่นเดียวกัน เมื่อดิจิตอลสโตเรจอสซิดโลสโคปถูกสร้างขึ้นมาจากขบวนการทางดิจิตอล ดังนั้นขีดความสามารถของมันจึงได้ถูกจำกัดด้วยวิธีการทางดิจิตอลในแอนะล็อกสโคปหรือดิจิตอลสโคปทุกๆ ไป จะเห็นว่าผู้ผลิตกำหนดขอบเขตการวัดเป็นค่าความถี่สูงสุดที่เครื่องสามารถทำการได้ เช่น 20 MHz, 100 MHz 200 MHz เป็นต้น ซึ่งเวลาซื้อก็ต้องไต่ตรงถึงขีดจำกัดอันนี้

สำหรับดิจิตอลสโตเรจอสซิดโลสโคปนั้นขอบเขตของการวัดนอกจากสามารถบอกเป็นค่าความถี่สูงสุดที่เครื่องสามารถวัดได้แล้ว ยังต้องบอกค่าอัตราการสุมตัวอย่างข้อมูลควบคู่ไปด้วย โดยปกติแล้วจะมีค่าเป็นล้านครั้งต่อวินาที เช่นความถี่สุม 20 MS/s วัดความถี่สูงสุด 20MHz, ความถี่สุม 40 MS/s วัดความถี่สูงสุด 100 MHz, ความถี่ 100 MS/s วัดความถี่สูงสุด 100 MHz เป็นต้น

เริ่มด้วยค่าอัตราการสุมตัวอย่างข้อมูลจะชี้ให้เห็นถึงความละเอียดและคุณภาพของภาพที่ปรากฏบนจอ ซึ่งประกอบขึ้นมาจากจุดที่ได้จากการสุมตัวอย่างบนรูปคลื่นที่วัด ถ้าจำนวนจุดที่ได้มีจำนวนมากภาพที่ได้ก็就会有ความถูกต้องสมบูรณ์มากกว่าการสูญเสียจุดจากการสุมตัวอย่างสูญเสียเพียงเล็กน้อยในบางรูปคลื่นก็อาจเป็นการสูญเสียรายละเอียดของรูปคลื่นนั้นไป ถ้าอัตราการสุมตัวอย่างมีค่ามาก นั่นคือ ดิจิตอลสโตเรจอสซิดโลสโคปจะมีความละเอียดของรูปคลื่นมากขึ้นด้วย ส่วนความถี่ที่บอกมานั้นก็ยังคงเป็นขอบเขตการวัดความถี่สูงสุดของดิจิตอลสโตเรจอสซิดโลสโคปในลักษณะเดียวกับแอนะล็อกสโคป

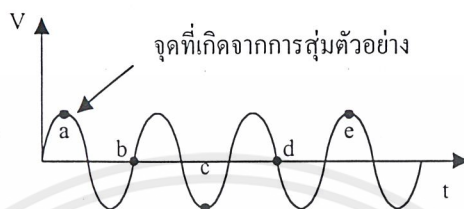
หากต้องการหารูปคลื่นที่ปรากฏบนจอที่มีความละเอียดเท่าไร ก็สามารถทำได้ด้วยการคำนวณตามสมการที่ 2.2

$$\text{จำนวนจุดต่อรูปคลื่น} = \frac{\text{อัตราการสุมตัวอย่างของดิจิตอลสโตเรจอสซิดโลสโคป}}{\text{ความถี่ที่ทำการวัด}} \quad (2.2)$$

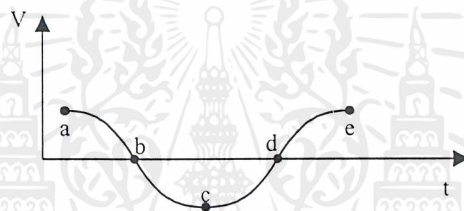
ตัวอย่างเช่น ดิจิตอลสโตเรจอสซิดโลสโคปมีอัตราการสุมตัวอย่าง 50 MS/s ใช้วัดสัญญาณที่มีความถี่ 500 kHz จากสมการที่ 2.2 จะได้ว่า

$$\begin{aligned} \text{จำนวนจุดต่อรูปคลื่น} &= \frac{50 \text{ MS/s}}{500 \text{ kHz}} \\ &= 100 \end{aligned}$$

จะเห็นว่าภาพที่แสดงผลบนจอในหนึ่งรูปคลื่นจะเกิดจากการเรียงกันของจุดจำนวน 100 จุด แต่ถ้าหากความถี่ที่ใช้วัดเปลี่ยนเป็น 10 MHz ที่อัตราการสุมตัวอย่างเดิม ความละเอียดของภาพในหนึ่งรูปคลื่นจะมีเพียง 5 จุดต่อหนึ่งรูปคลื่นเท่านั้น ซึ่งจากทั้ง 2 กรณี ทำให้สรุปได้ว่าเมื่อนำมาวัดค่าสัญญาณที่มีความถี่ต่ำๆ จะให้ความละเอียดของภาพสูง แต่เมื่อนำมาวัดสัญญาณที่มีความถี่สูงๆ แล้ว จะให้ภาพที่มีความละเอียดชัดเจนได้น้อยลง



ก)



ข)

รูปที่ 2.10 ลักษณะของการเกิดเอเลียสซึ่ง

- ก) สัญญาณอินพุตเมื่อมีการสุมตัวอย่างต่ำเกินไป
ข) เมื่อแสดงผลเกิดเป็นสัญญาณเอเลียส

ปัญหาอีกอย่างหนึ่งซึ่งเกิดขึ้นกับดิจิทัลอสซิลโลสโคป คือ การที่สัญญาณที่จะทำการวัดมีความถี่สูงมากๆ เมื่อมีการสุมตัวอย่างดังกล่าวแล้วนำมาสร้างให้ปรากฏบนจออีกครั้งผลที่ได้บนจอจะเป็นภาพที่ต่างไปจากสัญญาณที่วัดจริง คือสัญญาณเอเลียสอันเกิดจากอัตราการสุมต่ำกว่าความถี่ของสัญญาณที่วัดจริง โดยดูได้จากตัวอย่างในรูปที่ 2.10 จุด a, b, c, d และ e เป็นจุดที่เกิดจากการสุมสัญญาณ ในการใช้งานควรพยายามหลีกเลี่ยงอย่าให้เกิดสัญญาณเอเลียสขึ้น เพราะว่าการที่สัญญาณเอเลียสเป็นรูปคลื่นที่มีความถี่ต่างไปจากรูปคลื่นจริงอย่างมาก ซึ่งไม่สามารถนำไปวิเคราะห์ได้

การที่ทำให้รูปคลื่นปรากฏบนจอถูกต้องสมบูรณ์นั้นจำนวนจุดที่ได้จากการสุ่มตัวอย่างต้องมีมากกว่า 2 เท่าของความถี่ที่วัดนั้นใน 1 รูปคลื่นและเมื่อทำการตรวจวัดขอบขาขึ้นและขอบขาลงของพัลส์จะต้องได้จำนวนครั้งที่สุ่มตัวอย่างมากกว่า 10 ครั้งต่อ 1 รูปคลื่น นอกจากการสุ่มตัวอย่างที่ต้องการความเร็วในการทำงานแล้วยังต้องพิจารณาอีกด้วยว่า วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลนั้น สามารถทำงานได้ด้วยความเร็วขนาดไหนซึ่งจะพิจารณากันในหัวข้อต่อไป

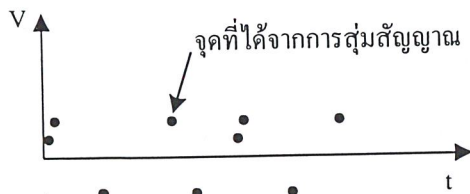
สิ่งที่จะทำให้เรารู้ถึงขอบเขตของการวัดของดิจิทัลสโตเรจอสซิคัลโลสโคปอีกอย่างหนึ่งคือค่าความถี่สูงสุดของสัญญาณที่ดิจิทัลสโตเรจอสซิคัลโลสโคปสามารถทำการเก็บข้อมูลได้ ซึ่งเราเรียกว่าค่า USB (Useful Storage Bandwidth) ค่า USB นี้จะมีค่าเป็นเท่าไรนั้นขึ้นอยู่กับความสามารถในการสุ่มตัวอย่างของดิจิทัลสโตเรจอสซิคัลโลสโคปด้วย ซึ่งค่า USB สามารถหาได้จากสมการที่ 2.3

$$USB = \frac{\text{อัตราการสุ่มตัวอย่างสูงสุด}}{25} \quad (2.3)$$

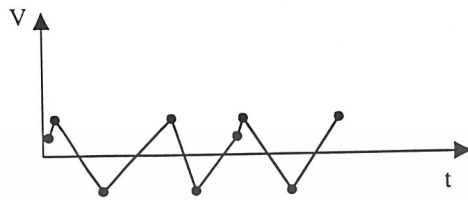
ตัวเลข 25 เป็นจำนวนจุดที่ใช้ในการสร้างภาพบนหน้าจอ 1 รูปคลื่นซึ่งเป็นค่าที่ใช้โดยทั่วไป เนื่องจากดิจิทัลสโตเรจอสซิคัลโลสโคปไม่สามารถแสดงรายละเอียดของสัญญาณที่มีความถี่สูงกว่าขีดจำกัดได้มากพอ ดังนั้นจึงได้มีการคิดค้นวิธีการเพิ่มรายละเอียดของสัญญาณโดยการสร้างความถี่ในการสุ่มตัวอย่างแบบปกติ ผลคือ จะมีจุดเสริมเข้าไปในรูปคลื่นเดิมภาพที่ได้จะมีความต่อเนื่องสม่ำเสมอมากขึ้น และเรียกรูปนี้ว่า “ทฤษฎีการสอดแทรกแบบลิเนียร์” ซึ่งแสดงตัวอย่างการสอดแทรกสัญญาณในรูปที่ 2.11

ในรูปที่ 2.11 ก) จะเป็นการแสดงให้เห็นถึงการสร้างภาพให้ปรากฏบนจอโดยการเรียงจุดที่ได้จากการสุ่มตัวอย่างโดยปกติจะได้ประมาณ 25 จุดต่อรูปคลื่น ส่วนรูปที่ 2.11 ข) เป็นการลากเส้นเชื่อมต่อกันระหว่างจุดที่ได้มาทำให้ได้ภาพเป็นเส้นตรงเชื่อมต่อกัน 10 เส้น ต่อหนึ่งรูปคลื่น ซึ่งวิธีการนี้เรียกว่า การสอดแทรกแบบลิเนียร์ หรือ การแสดงผลบวกแบบเวกเตอร์ (Vector Display) ดังนั้นเมื่อเป็นการแสดงผลของรูปคลื่นไซน์ ค่า USB จะเป็นดังสมการที่ 2.4

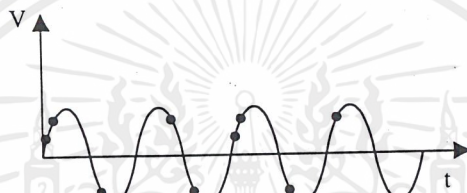
$$USB = \frac{\text{อัตราการสุ่มตัวอย่างสูงสุด}}{10} \quad (2.4)$$



ก)



ข)



ค)

รูปที่ 2.11 ทฤษฎีการสอดแทรก

- ก) การแสดงผลแบบจุด
- ข) การสอดแทรกด้วยวิธีเวกเตอร์
- ค) การสอดแทรกแบบไซน์

สำหรับรูปที่ 2.11ค) เป็นการสอดแทรกโดยการนำเอาฟังก์ชันไซน์มาคำนวณลากเส้นเชื่อมต่อระหว่างจุด (Sine interpolation) ซึ่งวิธีการนี้จะใช้จุดเพียง 2.5 จุดต่อรูปคลื่นเท่านั้นในการแสดงผล ดังนั้นค่า USB จะเป็น

$$USB = \frac{\text{อัตราการสุ่มตัวอย่างสูงสุด}}{2.5} \quad (2.5)$$

ซึ่งทำให้ค่า USB มีค่ามากขึ้นเท่ากับว่าขอบเขตของดิจิตอลสตรเรจออกซิดัลโพลก็ถูกขยายกว้างไปด้วย

1) ดิจิตอลสต่อเรจอสซิลโลสโคปกับรูปคลื่นพัลส์

หากเราจะพุดถึงรูปคลื่นพัลส์แล้ว ส่วนที่มีความสำคัญมากต่อรูปคลื่นชนิดนี้ก็คือ ช่วงเวลาขอบขาขึ้นและขอบขาลงของพัลส์ ดังนั้นเครื่องมือที่จะใช้วัดรูปคลื่นเหล่านี้จะต้องสามารถวัดช่วงขอบขาขึ้นและลงได้ โดยมีความผิดพลาดน้อยที่สุดในแอนะล็อกสโคปสามารถคำนวณหาช่วงเวลาขอบขาขึ้น (rise time : t_r) ของพัลส์ได้ดังสมการที่ 2.6

$$t_r = \frac{0.35}{\text{แบนด์วิดท์}} \quad (2.6)$$

จากสูตรดังกล่าวหากแอนะล็อกสโคปมีย่านการวัด 100 MHz เราจะได้ช่วงเวลาขอบขาขึ้นของพัลส์เท่ากับ 3.5 ns นี่ก็คือวิธีการของแอนะล็อกสโคป ส่วนในดิจิตอลสโคปช่วงเวลาของพัลส์ขาขึ้นที่วัดได้จะมีการเปลี่ยนแปลงอยู่ระหว่าง 0.8 ถึง 1.6 เท่า ของช่วงเวลากการสุ่มตัวอย่าง หากเรากิดช่วงเวลาขอบขาขึ้นเป็น 10 ถึง 90 เปอร์เซ็นต์ของแอมพลิจูดของพัลส์จะได้ดังสมการที่ 2.7

$$t_r = 1.6 \times \text{อัตราการสุ่มตัวอย่าง} \quad (2.7)$$

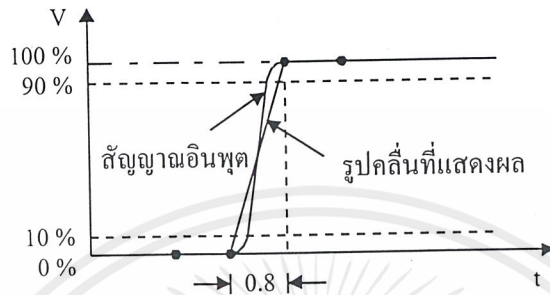
เพื่อให้การคำนวณมีความคลาดเคลื่อนน้อยลง สูตรการหาช่วงเวลาขอบขาขึ้นที่นำไปใช้งานจริง จึงใช้ค่า 1.6 ตลอด ดังสมการที่ 2.8

$$U_r = 1.6 \times \text{ช่วงเวลาการสุ่มตัวอย่างค่าสุด} \quad (2.8)$$

สำหรับดิจิตอลสต่อเรจอสซิลโลสโคปแล้วในการหาช่วงขอบขาขึ้นที่สามารถจะวัดได้จะมีการคำนวณเช่นเดียวกับกับดิจิตอลสโคป เช่นถ้า มีอัตราการสุ่มตัวอย่างเป็น 100 MS/s ก็จะมีช่วงเวลาการสุ่มตัวอย่างเป็น $0.01 \mu\text{s}$ และ $t_r = 0.01 \mu\text{s} \times 1.6 = 16 \text{ ns}$ นั่นคือดิจิตอลสต่อเรจอสซิลโลสโคปเครื่องนี้ จะมีความสามารถแสดงผลช่วงขอบขาขึ้นในการวัดได้ภายในเวลา 16 ns เมื่อคิดช่วงขอบขาขึ้นเป็น 10 ถึง 90 เปอร์เซ็นต์ของแอมพลิจูด

จะเห็นได้ว่าความสามารถในการแสดงผลรูปคลื่นพัลส์นั้นจะขึ้นอยู่กับช่วงเวลาในการสุ่มตัวอย่างของดิจิตอลสต่อเรจอสซิลโลสโคปเอง รูปที่ 2.12 ก) จะเป็นการแสดงให้เห็นถึงช่วงเวลาขอบขาขึ้นที่อยู่ระหว่างช่วงการสุ่มตัวอย่าง โดยรูปคลื่นที่แสดงผลจะอยู่ในช่วง $0.8 \times (\text{ช่วงเวลาการสุ่มตัวอย่าง})$ รูปคลื่นที่แสดงผลออกมาจะมีความชันของขอบขาขึ้นต่างจากรูปจริงอยู่บ้างพอ

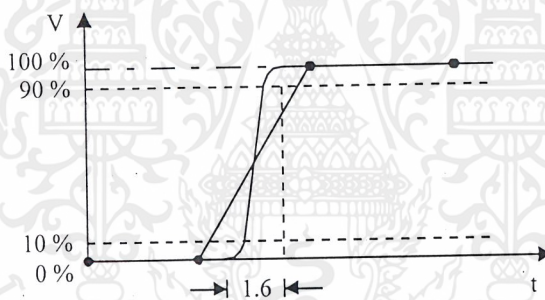
สมการและเมื่อรูปร่างสัญญาณอินพุตเดียวกันแต่มีช่วงเวลาในการสุ่มตัวอย่างต่างกันตามรูปที่ 2.12 ข) ภาพที่ได้จากการแสดงผลช่วงขอบขาขึ้นจะอยู่ในช่วง $1.6\times$ (ช่วงเวลาการสุ่มตัวอย่าง) ซึ่งถ้าเปรียบกับรูปที่ 2.12 ก) แล้ว ความชันของขอบขาขึ้นตามรูปที่ 2.12 ข) จะแตกต่างจากรูปร่างสัญญาณอินพุตจริงค่อนข้างมากทีเดียว



ช่วงเวลาการสุ่มตัวอย่าง

การสุ่มตัวอย่าง

ก)



ช่วงเวลาการสุ่มตัวอย่าง

การสุ่มตัวอย่าง

ข)

รูปที่ 2.12 การจัดช่วงขอบขาขึ้น ซึ่งค่าคลาดเคลื่อนที่เกิดขึ้นอยู่กับตำแหน่งในการสุ่มตัวอย่าง

ก) ช่วงขอบขาขึ้นอยู่ระหว่าง 0.8

ข) ช่วงขอบขาขึ้นอยู่ระหว่าง 1.6

จากที่ได้ทราบในตอนต้นว่า การจัดช่วงขอบขาขึ้นและขอบขาลงของพัลส์จะต้องทำการสุ่มตัวอย่างมากกว่า 10 ครั้ง ดังนั้นสามารถหาช่วงเวลาที่ต้องใช้ในการวัดขอบขาขึ้นได้ โดยหารอัตราการสุ่มตัวอย่างด้วย 10 แล้วกลับเศษเป็นส่วน ดังสมการที่ 2.9

$$\text{resolution} = \frac{10}{\text{อัตราการสุ่มตัวอย่าง}} \quad (2.9)$$

ถ้าเป็นดิจิตอลสต่อเรจอสซิดโลสโคปที่มีอัตราสุ่มตัวอย่าง 100 MS/s จะมีการสุ่มตัวอย่างทุกๆ 10 ns (1/100MS/s) และจะใช้การทำงานกับขอบขาขึ้นเป็นเวลา 100 ns แล้วถ้าหากว่าพัลส์ที่เข้ามาที่มีความถี่สูงมาก เราก็ยังต้องการดิจิตอลสต่อเรจอสซิดโลสโคปที่มีความเร็วในการสุ่มตัวอย่างสูงขึ้นด้วย เช่น 500 MS/s ซึ่งจะมีการสุ่มตัวอย่างทุกๆ 2 ns และสามารถวัดช่วงขอบขาขึ้นที่อยู่ระหว่างช่วงการสุ่มตัวอย่างโดยใช้เวลา 1.17 ns

จะเห็นว่าดิจิตอลสต่อเรจอสซิดโลสโคปตอบสนองการวัดรูปคลื่นพัลส์ได้ไม่ค่อยจะดีเท่ากับแอนะล็อกสโคป และเราไม่สามารถนำค่า U_T ที่วัดได้จากดิจิตอลสต่อเรจอสซิดโลสโคปไปคำนวณเปรียบเทียบกับรูปคลื่นจริงได้มากพอ ดังนั้น เราจะถือว่าค่า U_T เป็นค่าซึ่งแสดงถึงขอบเขตจำกัดของดิจิตอลสต่อเรจอสซิดโลสโคปก็ได้

2.2.3 การสุ่มตัวอย่างของดิจิตอลสต่อเรจอสซิดโลสโคป

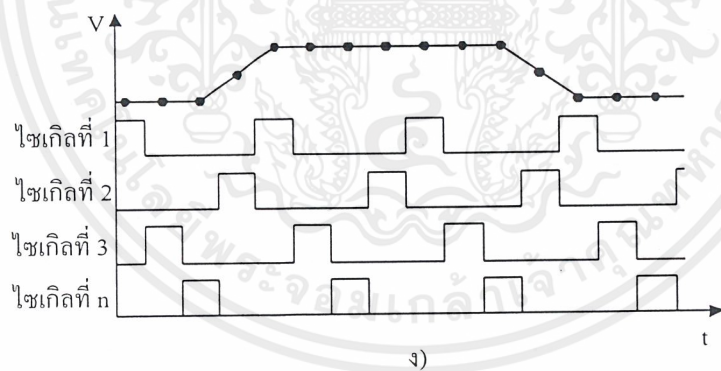
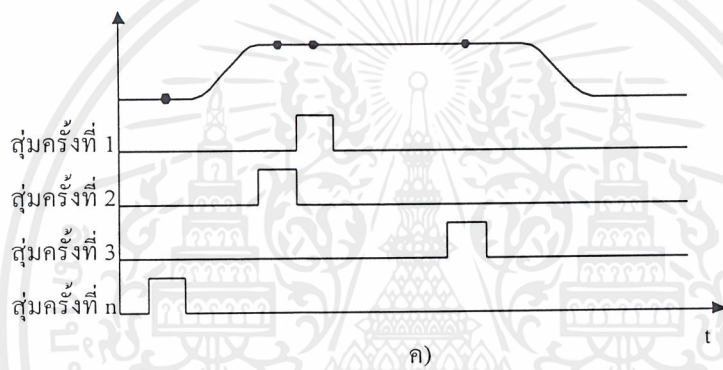
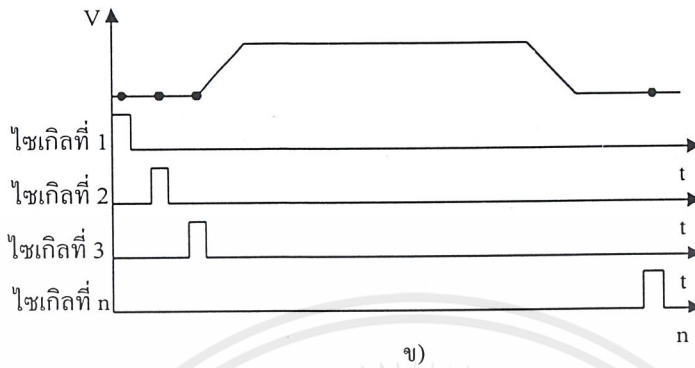
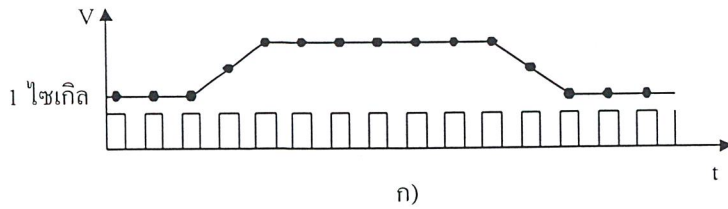
ดิจิตอลสต่อเรจอสซิดโลสโคปใช้การสุ่มตัวอย่างด้วยกัน 2 แบบคือ

1) การสุ่มตัวอย่างแบบเวลาจริง (Real Time or One-Shot)

จะเกิดการสุ่มตัวอย่างขึ้นทันทีที่มีการจัดสัญญาณอินพุตได้ โดยผลที่ได้จะมาจากรูปคลื่นจริงๆ และจะมีความเหมือนกับรูปคลื่นจริงมาก ส่วนมากจะใช้กับสัญญาณที่เกิดขึ้นเพียงชั่วขณะ หรือใช้ในการสุ่มตัวอย่างบางส่วนของสัญญาณที่ต้องการวิเคราะห์เป็นพิเศษ ดังนั้นความเร็วของการสุ่มตัวอย่างแบบเวลาจริงนี้จึงต้องสูงพอที่จะทำงานในกรณีดังกล่าวได้ การสุ่มตัวอย่างแบบนี้จะทำงานแบบผ่านครั้งเดียวคือ จะมีการสุ่มตัวอย่างจุดเดียวในแต่ละครั้งของการทำงาน จึงสามารถนำไปตรวจจับสัญญาณที่เกิดขึ้นเพียงชั่วขณะได้ เช่น แรงไฟกระชาก, สัญญาณควบคุมเครื่องกลสัญญาณในลักษณะชีววิทยา ฯลฯ ซึ่งจะปรากฏให้สามารถวัดได้ในระยะเวลาที่สั้นมาก การจะใช้วิธีสุ่มตัวอย่างแบบเวลาจริงคือแบบที่เมื่อเริ่มสุ่มและสิ้นสุดการสุ่มต้องได้รูปสัญญาณที่เกิดขึ้น รูปที่ 2.13 ก) จะเห็นการสุ่มตัวอย่างแบบเวลาจริงที่ใช้ในการวัดรูปคลื่นพัลส์หนึ่งลูก

2) การสุ่มตัวอย่างแบบเวลาเสมือน (Equivalent Time)

การสุ่มตัวอย่างแบบนี้โดยมากจะใช้กับสัญญาณที่เกิดขึ้นซ้ำๆ กันหรือใช้ตรวจจับบางส่วนของสัญญาณที่เกิดขึ้นซ้ำๆ กัน สามารถแบ่งออกได้เป็น 2 ชนิดย่อยๆ อีกคือ การสุ่มชนิดเป็นอันดับ (Sequential Sampling) และการสุ่มชนิดไม่เป็นลำดับ (Random Sampling)



รูปที่ 2.13 เทคนิคการสุ่มตัวอย่างแบบต่างๆ ของคิจิตอลสตอเรจอสซิดโลสโคป

ก) การสุ่มแบบเวลาจริง

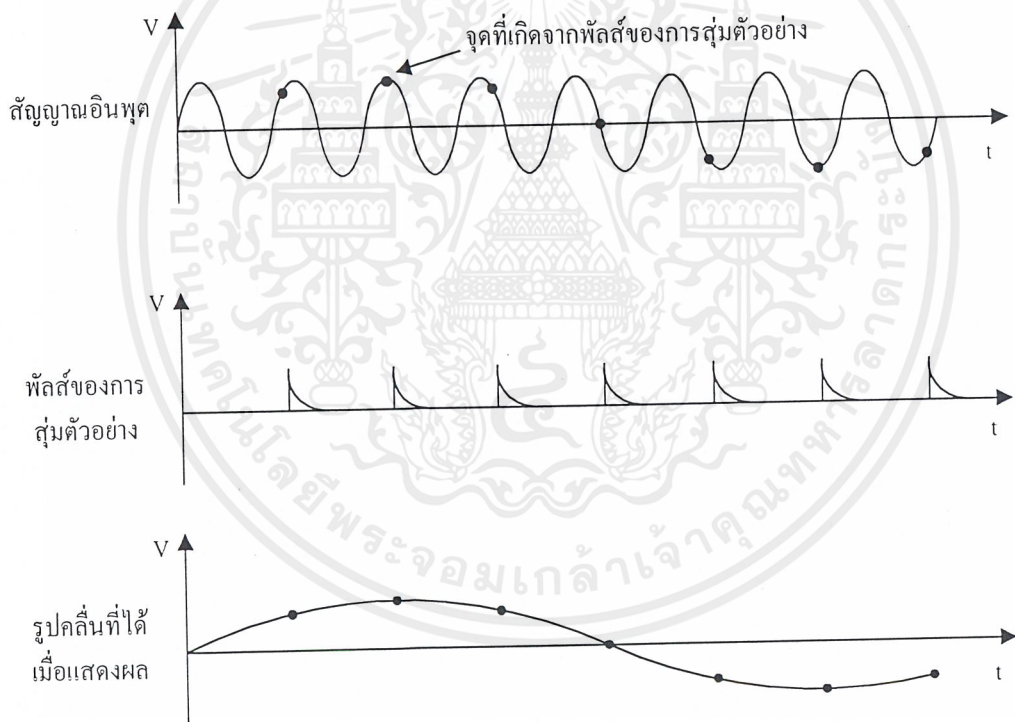
ข) การสุ่มแบบเวลาเสมือนชนิดเป็นลำดับ

ค) การสุ่มแบบเวลาเสมือนชนิดไม่เป็นลำดับโดยใช้จุดเดียวในการทำงานหนึ่งครั้ง

ง) การสุ่มแบบเวลาเสมือนชนิดไม่เป็นลำดับโดยใช้หลายจุดในการทำงานหนึ่งครั้ง

ในการสุ่มตัวอย่างชนิดลำดับเป็นดังรูปที่ 2.13 ข) ซึ่งการสุ่มตัวอย่างชนิดนี้เป็นการทำงานหนึ่งครั้งหรือหนึ่งไซเคิลจะต้องได้จุดบนรูปคลื่น 1 จุด แต่จะเป็นจุดตรงส่วนใดของรูปคลื่นแล้วนำไปเก็บไว้ในหน่วยความจำ จนกระทั่งหน่วยความจำเต็มเมื่อนำไปแสดงผลก็จะเป็นดัง รูปที่ 2.14 ส่วนการสุ่มตัวอย่างแบบเวลาเสมือนชนิดไม่เป็นลำดับนั้นจะเป็นดัง รูป 2.13 ค) ซึ่งการสุ่มตัวอย่างแบบนี้ จะใช้เมื่อมีการทำพริทริกเกอร์ เพื่อดูรูปคลื่นที่อยู่ด้านหลังจุดพริทริกซึ่งการสุ่มตัวอย่างแบบเป็นลำดับจะทำได้

ในรูปที่ 2.13 ง) จะเป็นการสุ่มตัวอย่างแบบไม่เป็นลำดับเช่นกันแต่จะต่างกันตรงที่ในการทำงาน 1 ไซเคิลนั้นจะได้หลายจุดบนรูปคลื่น และในการทำงานครั้งต่อไปก็จะได้จุดหลายจุดแต่จะเป็นจุดที่ไม่ซ้ำจุดเดิม (จุดใดก็ได้) การสุ่มตัวอย่างในรูปนี้นำไปใช้ประโยชน์เมื่อมีการวิเคราะห์รูปคลื่น โดยการนำข้อมูลที่ได้จากการสุ่มตัวอย่างแบบนี้มาเปรียบเทียบกับรูปคลื่นอื่นๆ โดยเฉพาะอย่างยิ่งการวิเคราะห์รูปคลื่นในส่วนที่เป็นการพริทริกในลักษณะต่างๆ กัน



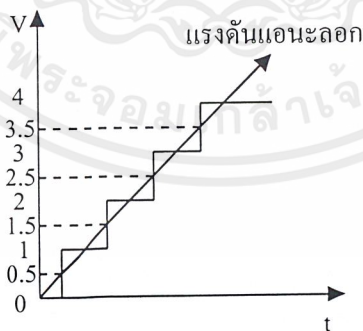
รูปที่ 2.14 ผลที่ออกมาจากการสุ่มตัวอย่างสัญญาณอินพุตแบบเป็นลำดับ

สำหรับดิจิตอลสตอเรจอสซิลโลสโคปที่มีการแสดงผลแบบแอนะล็อก ซึ่งระบุความสามารถในการวัดไว้ เช่น 100 MHz ความหมายของ 100 MHz นี้ก็คือ ความสามารถในการวัดสัญญาณความถี่ 100 MHz แล้วดิจิตอลสตอเรจอสซิลโลสโคปจะแสดงผลในแบบแอนะล็อกได้ทันที 100 MHz เฉพาะสำหรับสัญญาณที่เกิดขึ้นซ้ำๆ กัน แต่หากเป็นสัญญาณชั่วขณะเกิดขึ้นแล้วใช้การสุ่มตัวอย่างแบบเวลาจริงแล้ว ดิจิตอลสตอเรจอสซิลโลสโคปนี้จะสามารถวัดสัญญาณชั่วขณะได้เพียง 10 MHz เท่านั้น

เมื่อคิดเทียบเป็นอัตราส่วนระหว่างการทำงานกับสัญญาณปกติและการทำงานกับสัญญาณชั่วขณะของดิจิตอลสตอเรจอสซิลโลสโคป ก็จะได้เป็น 10 : 1 และสามารถจะเพิ่มเป็น 5 : 1 หรือ 2.5 : 1 ได้ ซึ่งหมายถึง ดิจิตอลสตอเรจอสซิลโลสโคป จะต้องมีวิธีการสอดแทรกอย่างมากมายทีเดียว สาเหตุที่เกิดกรณีดังกล่าวขึ้นนั้นมาจากการที่ความสามารถในการสุ่มตัวอย่างถูกกำหนดด้วยสัญญาณนาฬิกาภายใน แต่สัญญาณที่เข้ามาวัดจะเป็นการทำงานแสดงผลแบบแอนะล็อก ซึ่งขอบเขตของความถี่จะกว้างมาก นี่คือการนำข้อดีของแอนะล็อกมาใช้

2.2.4 ขั้นตอนหลังการสุ่มตัวอย่าง

เมื่อเราได้ทำการสุ่มตัวอย่างไปแล้วค่าที่ได้จะไม่ทำการแปลงเป็นสัญญาณดิจิตอลในทันที ทั้งนี้เนื่องจากค่าที่มาจากการสุ่มตัวอย่างนั้นเป็นค่าที่มีระดับแตกต่างกันอย่างมากมาย ในทางปฏิบัติเราจะไม่สามารถแปลงเป็นสัญญาณดิจิตอลได้มากขนาดนั้นจึงจำเป็นต้องแบ่งค่าที่ได้มาจากการสุ่มตัวอย่างออกเป็นกลุ่มๆ เราเรียกว่า “ควอนไทซ์” ซึ่งการควอนไทซ์นี้จะเป็นการนำค่ามาจากการสุ่มตัวอย่างที่มีค่าใกล้เคียงกันมากอยู่ในกลุ่มเดียวกัน ต่อจากนั้นจึงจะนำค่าตัวเลขมาใช้แทนกลุ่มสัญญาณดังกล่าว



รูปที่ 2.15 การควอนไทซ์ โดยใช้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลคอนเวอร์เตอร์ เปลี่ยนแรงดันแอนะล็อกเป็นเลขฐานสองที่ระดับต่างๆ

การแบ่งกลุ่มสัญญาณจะมีลักษณะเป็นขั้นบันไดตาม รูปที่ 2.15 โดยให้มีการเปลี่ยนแปลงของค่าที่ได้จากการสุ่มตัวอย่างระหว่าง 0-4 โวลต์ และจัดเป็นกลุ่มได้ 4 กลุ่ม คือ ช่วงระดับ 0-1 , 1-2 2-3 และ 3-4 โวลต์ โดยให้สัญญาณที่อยู่ในระดับ 0-1 มีค่า 0.5 แทนกลุ่มนี้ และ 1.5 แทนกลุ่ม 1-2 ระดับ 2.5 แทนกลุ่ม 2-3 และระดับ 3.5 แทนกลุ่ม 3-4 ค่า 0.5, 1.5, 2.5 และ 3.5 จะนำไปใช้ใน ADC อีกครั้งเพื่อทำเป็นสัญญาณดิจิทัล

เนื่องจากวิธีการควอนไทซ์ดังกล่าวมีโอกาสที่จะเกิดค่าคลาดเคลื่อนได้ อันเนื่องมาจากสัญญาณที่จะเกิดค่าคลาดเคลื่อนได้ อันเนื่องมาจากสัญญาณรบกวนที่เข้ามาในระบบทำให้ค่าที่ได้ในการจัดกลุ่มผิดไปจากความเป็นจริงได้ และเนื่องจากการทำควอนไทซ์นั้นประกอบจากวงจรดิจิทัลอิเล็กทรอนิกส์ ดังนั้นจึงเกิดความไม่แน่นอนของผลที่ได้อยู่บ้าง โดยความไม่แน่นอนเหล่านี้จะแสดงค่าบวก ลบ ของบิตสุดท้าย

เมื่อผ่าน ADC แล้ว นอกจากบิตสุดท้ายจะเป็นค่าของความไม่แน่นอนแล้ว บิตอื่นๆ ก็ยังเป็นค่าผิดพลาดที่เกิดจากสัญญาณรบกวนได้ด้วยเหมือนกันแต่สัญญาณดิจิทัลที่ได้ส่วนใหญ่ก็ยังคงมีความเที่ยงตรงอยู่มาก

2.2.5 การเก็บข้อมูลของดิจิทัลสตอเรจอสซิลโลสโคป

ความสามารถของดิจิทัลสตอเรจอสซิลโลสโคปที่สำคัญอีกอย่างหนึ่งก็คือ ความสามารถในการเก็บข้อมูล ซึ่งสำหรับดิจิทัลสตอเรจอสซิลโลสโคปแล้ว หน่วยความจำที่สามารถรองรับข้อมูลได้นั้นก็มีมากมายเช่นเดียวกับบรรดาคอมพิวเตอร์ทั้งหลาย ดังนั้น จึงเป็นที่แน่นอนว่า หน่วยความจำที่ใช้ก็ต้องมี 2 แบบ คือ RAM (Random Access Memory) และROM (Read Only Memory)

สำหรับแรมสามารถเก็บข้อมูลที่เป็นปัจจุบันได้อยู่ตลอดเวลา ในขณะที่รอมนั้นจะเป็นหน่วยความจำถาวร ไม่สามารถบันทึกข้อมูลเพิ่มเติมได้ ส่วนมากจะใช้ในการแสดงคุณสมบัติอุปกรณ์หรือวงจรต่างๆ รวมทั้งการแก้ไขปัญหาในลักษณะต่างๆ กัน

การเก็บบันทึกข้อมูลนั้นยังสามารถที่จะบันทึกลงบนแผ่นดิสก์เก็ต, เทปแม่เหล็ก หรือแม้แต่การพ่วงเข้ากับคอมพิวเตอร์เพื่อโอนถ่ายข้อมูลได้ด้วย

สำหรับการเก็บข้อมูลแบบรอมนั้น โดยปกติจะมีทั้งแบบ CMOS, NMOS หรือแม้แต่ ECL (Emitter Coupled Bipolar Logic) ซึ่งแบบต่างๆ เหล่านี้นั้นจะเป็นหน่วยความจำแรม ซึ่งอยู่ภายในดิจิทัลสตอเรจอสซิลโลสโคป และมีขนาดหน่วยความจำที่จำกัดสำหรับหน่วยความจำขนาดใหญ่ และมีความยาวของข้อมูลมากๆ ก็จะต้องทำการแก้ไขข้อมูลให้ทันสมัยอยู่ตลอดเวลาด้วยและขนาดหน่วยความจำที่ดิจิทัลสตอเรจอสซิลโลสโคปใช้จะอยู่ในช่วง 4 กิโลไบต์ ถึง 32 กิโลไบต์ นอกจากนี้ในปัจจุบันยังมีหน่วยความจำแบบแผ่นการ์ดด้วย

2.2.6 ปัญหาพื้นฐานของดิจิตอลสโตเรจอสซิลโลสโคป

ปัญหาที่มักจะเกิดขึ้นบ่อยๆ กับแอนะล็อกสโคปและดิจิตอลสโคปคือ การสั่นไหวของภาพทางแนวนอน (Horizontal Jitter) อาการดังกล่าวจะทำให้ภาพที่ได้จากการวัดเกิดอาการพร่ามัว เนื่องจากมีการสั่นของภาพ และจะเกิดกับรูปคลื่นไซน์ไปจนถึงรูปคลื่นพัลส์ด้วย

อย่างไรก็ดีปัญหานี้ก็อาจเกิดขึ้นได้อีก อันเนื่องมาจากสิ่งแวดล้อมภายนอกอย่างเช่น การที่สัญญาณแอนะล็อกที่ทำการวัดมีระดับแรงดันที่ไม่คงที่ มีการเปลี่ยนแปลงตลอดเวลา ซึ่งแรงดันดังกล่าวจะเข้าไปเกี่ยวพันกับการทริกของสโคปโดยตรงด้วย ซึ่งจะสรุปได้ว่าถ้าทุกอย่างภายนอกเป็นปกติปัญหานี้ก็จะไม่เกิดขึ้นมาอีกอย่างแน่นอนสำหรับดิจิตอลสโตเรจอสซิลโลสโคป

2.2.7 ความละเอียดถูกต้องแม่นยำ

ความละเอียดและความแม่นยำเป็นสิ่งที่มีความสำคัญมาก สำหรับเครื่องวัดทุกชนิด รวมทั้งดิจิตอลสโตเรจอสซิลโลสโคปด้วยเช่นกัน เมื่อพูดถึงความละเอียดทางด้านแนวตั้งย่อมาจะหมายถึงความสามารถในการแยกแยะระหว่างสัญญาณซึ่งอยู่ติดๆ กัน

โดยปกติแล้วความละเอียดทางด้านแนวตั้งของดิจิตอลสโตเรจอสซิลโลสโคปถูกกำหนดด้วยจำนวนเลขฐานสองที่เป็นเอาต์พุตของ ADC ตัวอย่างเช่น ADC ให้เอาต์พุตออกมา 8 บิต ก็จะมีความละเอียดได้ถึง $256 (2^8)$ ตำแหน่ง หรือคิดเป็น 0.391% เพื่อความชัดเจนยิ่งขึ้น นอกจากจะต้องรู้จำนวนบิตของ ADC แล้ว ยังต้องรู้อีกว่าถ้าหากจะตั้งค่าโวลต์ต่อช่องไว้ที่ค่าหนึ่งจะสามารถอ่านค่าความละเอียดออกมาได้เท่าไร ตัวอย่างเช่น ถ้าหากตั้งค่าโวลต์ต่อช่องไว้ที่ 50 mV/div เมื่ออ่านค่าเดิมน้ำจอซึ่งมี 8 ช่อง ก็จะอ่านค่าได้ทั้ง 400 mV และถ้า ADC ใช้ 8 บิตก็จะได้ค่าความละเอียดของแนวตั้งทั้งสิ้น 256 ตำแหน่ง ซึ่งแต่ละตำแหน่งจะมีค่าเป็น $400 \times 10^{-3} / 256 = 1.5626$ mV

นั่นคือ 1 ตำแหน่งจะแทนเป็นระดับแรงดันได้ 1.5626 mV ในทำนองเดียวกันหาก ADC ให้เอาต์พุตออกมา 4 บิต ความละเอียดทางแนวตั้งจะได้ 16 ตำแหน่งและตั้งโวลต์ต่อช่องค่าเดิม ดังนั้นสำหรับ 1 ตำแหน่งก็จะมีค่าเป็น 2.5 mV ซึ่งจะเห็นได้ว่าจำนวนบิตของ ADC และมาตรฐานส่วนทางแนวตั้งที่ตั้งค่าไว้ จะเป็นตัวกำหนดความสามารถในการแยกแยะรายละเอียดต่างๆ ของรูปคลื่นนั่นเอง

เมื่อมาพิจารณาถึงความละเอียดทางด้านแนวนอน สำหรับความละเอียดทางด้านแนวนอนนั้น เป็นค่าที่เกี่ยวข้องกับคาบเวลาที่ถูกรับไว้ในหน่วยความจำแบบดิจิตอล หากสัญญาณแนวนอนถูกรับไว้ภายในข้อมูล 1024 เวิร์ด จะได้ค่าความละเอียดของด้านแนวนอนเป็น 1/1024 หรือคิดเป็น 0.098%

หากนำแอนะล็อกสโคปและดิจิตอลสโคปมาเปรียบเทียบกันจะพบว่า ถ้าเป็นแอนะล็อกสโคป ความละเอียดของภาพที่ได้ จะขึ้นอยู่กับพื้นที่ผิวด้านหน้าจอ CRT และขนาดรูปร่างของลำ

อิเล็กทรอนิกส์ที่ใช้ในการเขียน รวมไปถึงความสามารถในการทำงานของวงจรมอนิเตอร์สัญญาณทางแนวตั้งและแนวนอน ซึ่งกลายเป็นปัจจัยอันหนึ่งที่เป็นข้อจำกัดอย่างมากสำหรับแอนะล็อกสโคป

แต่สำหรับดิจิตอลสโคปความละเอียดทางแนวตั้งจะขึ้นอยู่กับจำนวนบิตที่ ADC ใช้ และในเรื่องของความถูกต้องแม่นยำ แอนะล็อกจะคล้ายกับดิจิตอลในบางกรณี ซึ่งโดยปกติค่าความคลาดเคลื่อนทางแนวตั้งจะไม่เกิน $\pm 2.4\%$ และทางแนวนอนจะไม่เกิน $\pm 1.3\%$

อย่างไรก็ดี ดิจิตอลสโคปมีข้อได้เปรียบเรื่องความถูกต้องแม่นยำเพิ่มขึ้นอีกเล็กน้อย ด้วยลักษณะของภาพที่เรียกว่า Frozen ที่ปรากฏบนจอ CRT และโดยการใช้ตัวชี้ค่าที่เรียกว่า Cursors ซึ่งมีการชี้ค่าได้แม่นยำมากทีเดียว สิ่งนี้เสริมเข้ามามีทำให้ดิจิตอลสโคปสามารถลบจุดด้อยด้านความแม่นยำบางเรื่องลงไปได้มาก

สำหรับในเรื่องฐานเวลานั้นดิจิตอลสโคปเรจิสเตอร์สโคป จะไม่ใช่วงจรกำเนิดฐานเวลาแบบทีเนียร์สวีฟ แต่จะใช้วงจรกำเนิดฐานเวลาแบบคริสตอลแทน เนื่องจากฐานเวลาที่ได้จากแบบคริสตอลจะมีค่าความคลาดเคลื่อนน้อยกว่า $\pm 0.01\%$

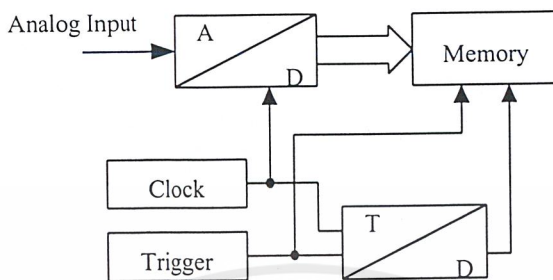
เมื่อมาถึงจุดนี้ก็จะเห็นได้ว่าความแม่นยำด้านแนวนอนจะเกี่ยวข้องกับสัญญาณนาฬิกา, ขนาดของหน่วยความจำ, และตัวชี้ค่า จะเห็นว่าความแม่นยำด้านแนวนอนจะต้องอาศัยองค์ประกอบมากทีเดียว ดังนั้นเพื่อให้เกิดความแม่นยำด้านแนวนอนสูงสุดสำหรับดิจิตอลสโคปเรจิสเตอร์สโคป เราจึงต้องทำการวัดสัญญาณนั้นๆ ซ้ำๆ กันและต้องลดทอนค่าคลาดเคลื่อนที่เกิดจากการเลือกใช้ตัวชี้ค่าที่ไม่เหมาะสมลงด้วย

2.2.8 สัญญาณชั่วขณะ (Transient Capture)

แอนะล็อกสโคปส่วนใหญ่ไม่สามารถแสดงสัญญาณชั่วขณะที่เกิดขึ้นได้ แต่ในทางตรงข้ามดิจิตอลสโคปถูกออกแบบเพื่อจับสัญญาณชั่วขณะ โดยมีโครงสร้างของดิจิไทเซอร์ 3 แบบ สำหรับทรานเซียนดิจิไทเซอร์ และ RIS (Random Interleaved Sampling) สามารถจับสัญญาณชั่วขณะและบันทึกข้อมูลรูปคลื่นปริทริกเกอร์ได้ ในขณะที่แซมปลิงดิจิไทเซอร์ไม่สามารถทำได้ ดิจิไทเซอร์ทั้ง 3 ชนิด สามารถบันทึกสัญญาณที่เกิดขึ้นซ้ำๆ หลายครั้งได้

ในทรานเซียนดิจิไทเซอร์ จะมีอุปกรณ์แปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล และหน่วยความจำรูปคลื่น อุปกรณ์เปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลจะเปลี่ยนสัญญาณต่อเนื่องให้อยู่ในรูปตัวเลข และส่งสัญญาณที่สุ่มได้ไปยังหน่วยความจำแบบ Circular Addressing หลังจากนั้นหน่วยความจำตำแหน่งสุดท้าย ระบบจะเขียนข้อมูลทับอันเดิม โดยจะเริ่มเขียนทับที่หน่วยความจำเริ่มต้น หลังจากทีทริกเกอร์ถูกผลิตขึ้น หน่วยความจำจะยังคงทำงานต่อไปโดยเป็นค่าที่ถูกเลือกใช้ของตัวอย่างโพสทริกเกอร์ จากนั้นอุปกรณ์ที่เปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลจะหยุดป้อนข้อมูลสู่หน่วยความจำ ถ้าหากว่าผู้เลือกใช้ข้อมูลปริทริกเกอร์ 100% แล้ว

อุปกรณ์เปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลจะหยุดส่งข้อมูลทันทีที่ทรiggerมาถึง แต่ถ้าผู้เลือกใช้เลือก โปสทรigger 100% ระบบจะเติมข้อมูลในทุกแอดเรสของหน่วยความจำมากกว่า 1 ครั้งแล้วหยุดหน่วยความจำจะมีข้อมูลรูปคลื่นที่เกิดขึ้นที่หลังทรigger

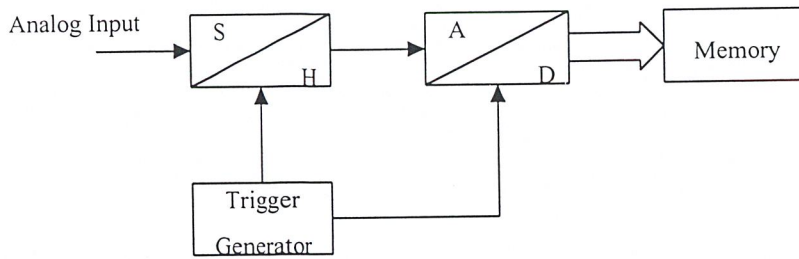


รูปที่ 2.16 แผนผังการทำงานของ RIS ดิจิไลเซอร์

RIS ดิจิไลเซอร์ จะประกอบด้วยทรานเซียนดิจิไลเซอร์ที่เพิ่มโหมดการสอดแทรกสำหรับทรiggerแต่ละอัน ดังรูปที่ 2.16 RIS ดิจิไลเซอร์จะบันทึกจุดของจุดตัวอย่างรูปคลื่น จุดตัวอย่างในดิจิไลเซอร์มาจากการได้มาของการเพิ่มทรigger เพื่อสร้างรายละเอียดของรูปร่างสัญญาณต้นกำเนิดเนื่องจากดิจิไลเซอร์ไม่มีทางรู้ได้ว่ามีสัญญาณทรiggerแล้ว สัญญาณนาฬิกาและจุดทรigger จึงไม่สอดคล้องกัน ดังนั้นเวลาระหว่างทรiggerและทุกๆ สัญญาณนาฬิกาเปลี่ยนแปลงอย่างสุ่มจากรูปคลื่นที่ได้มาโครงสร้างของ RIS จะใช้ TDC ในการวัดความสัมพันธ์และความแม่นยำของการได้มาของรูปคลื่น TDC มีการแก้ปัญหาเรื่องเวลามากกว่าการเว้นช่วงของเวลาดังตัวอย่าง ดังนั้น RIS จะช่วยแสดงผลให้เห็นรายละเอียดที่ทรานเซียนดิจิไลเซอร์อย่างเฉยพลาดไป RIS สามารถเลือกการบันทึกปริทรiggerเช่นเดียวกับทรานเซียนดิจิไลเซอร์

ดิจิไลเซอร์การสุ่มตัวอย่าง (Sampling Digitizer) ที่มีประสิทธิภาพจะประกอบไปด้วยส่วนหัวของการสุ่มตัวอย่าง, อุปกรณ์แปลงสัญญาณแอนะล็อกเป็นดิจิทัล, หน่วยความจำ, รูปคลื่นและวงจรสัญญาณเวลาส่วนหัวของการสุ่มตัวอย่างจะเก็บค่าศักดาไฟฟ้าและคงค่าไว้ ขณะที่อุปกรณ์แปลงสัญญาณแอนะล็อกเป็นดิจิทัลทำการแปลงเป็นตัวเลข ดิจิไลเซอร์การสุ่มตัวอย่างจะได้ 1 ตัวอย่างต่อทรigger สำหรับแต่ละทรiggerตามลำดับ ดังรูปที่ 2.17 วงจรสัญญาณเวลาจะทำการหน่วงเวลาจากทรiggerถึงจุดตัวอย่าง

เช่นอัตราการสุ่ม 1 GS/s จุดตัวอย่างแรกจะอยู่ที่จุดทรiggerจุดที่ 2 หน่วงไป 1 นาโนวินาที จุดที่ 3 หน่วงไป 2 นาโนวินาที เนื่องจากจุดตัวอย่างถูกหน่วงจากจุดทรigger ดังนั้นดิจิไลเซอร์การสุ่มตัวอย่างไม่สามารถบันทึกข้อมูลที่มีก่อนการทรiggerได้



รูปที่ 2.17 แผนผังการทำงานของดิจิทัลิไซเซอร์การสุ่มตัวอย่าง

2.2.9 แบนวิดธ์และอัตราการสุ่มตัวอย่าง

แบนวิดธ์เป็นส่วนสำคัญที่ต้องระบุสำหรับดิจิทัลิไซเซอร์ เช่นเดียวกับแอนะล็อกสโคปการขยายข้อมูลที่ป้อนเข้าของดิจิทัลิไซเซอร์และการกรองเป็นตัวตัดสินกำหนดแบนวิดธ์ ขอบสัญญาณพัลส์ที่เร็ว และจุดยอดของรูปคลื่นที่ฉับพลัน ประกอบด้วยส่วนประกอบสัญญาณความถี่สูงเพื่อที่จะบันทึกยอดและขอบของสัญญาณ ได้ถูกต้องแม่นยำ ดิจิทัลิไซเซอร์ต้องมีแบนวิดธ์เพียงพอที่จะผ่านสัญญาณความถี่สูงที่มีการลดทอนต่ำ

ต้องมีแบนวิดธ์เท่าใด เพื่อแสดงการขยายจุดยอดสัญญาณอย่างถูกต้องแม่นยำ แบนวิดธ์ของดิจิทัลิไซเซอร์ควรมีมากกว่าของแบนวิดธ์สัญญาณ ดังนั้นสิ่งแรกที่ต้องพิจารณาแบนวิดธ์ของสัญญาณ โดยการประเมินค่าเวลาที่เพิ่มขึ้นเร็วกว่า สมมติผลการตอบสนองต่อระบบมีโพลเดียวแบนวิดธ์ของสัญญาณจะเป็นดังนี้

1) แบนวิดธ์ของสัญญาณ 0.35/(10% -90% rise time)

แบนวิดธ์ของดิจิทัลิไซเซอร์แสดงความถี่ซึ่งถูกลดทอนลง 3 เดซิเบล การลดทอนนี้เกิดขึ้นอย่างค่อยเป็นค่อยไป เริ่มจากความถี่ต่ำกว่า ดังนั้นต้องเลือกดิจิทัลิไซเซอร์ซึ่งมีแบนวิดธ์สูงกว่าของสัญญาณ ผลกระทบต่ออัตราการสุ่มตัวอย่างของแบนวิดธ์ที่ใช้

อัตราการสุ่มตัวอย่างของดิจิทัลิไซเซอร์สามารถลดแบนวิดธ์ที่ใช้ เพื่อให้แน่ใจว่าการสุ่มตัวอย่างที่ใช้เพียงพอ เมื่อรับมา 4 ตัวอย่างต่อรอบกับการแทรก $\sin x/x$ หรือ 10 ตัวอย่างต่อรอบกับการแทรกเส้นตรง ถ้าเป็นสัญญาณทรานเซียนพิจารณารายละเอียดอัตราการสุ่มตัวอย่างซึ่งเกิดข้อผิดพลาด ถ้าเกิดซ้ำๆ หลายครั้งสามารถใช้อัตราการสุ่มที่เท่าๆ กันอย่างรวดเร็ว

ดิจิตเซอร์ในอุดมคติที่ไม่มีสัญญาณรบกวนและให้สัญญาณจำกัด หลักการของไนควิสต์ (Nyquist Theorem) ต้องได้ออย่างน้อย 2 ตัวอย่างต่อแต่ละรูปคลื่นออกจากกล่าวได้ว่าความถี่ข้อมูลป้อนเข้าที่สูงที่สุดห้ามเกินครึ่งหนึ่งของอัตราการสุ่มตัวอย่าง ขั้นตอนการสอดแทรก $\sin x/x$ สามารถจำลองสัญญาณข้อมูลป้อนเข้าในดิจิตเซอร์อย่างแม่นยำถูกต้องเหมาะสม ขั้นตอน $\sin x/x$ เหมาะกับส่วนโค้งระหว่างจุดตัวอย่างเพื่อแสดงรูปคลื่นที่เรียบ มีข้อเสียการสอดแทรก $\sin x/x$ สามารถขยายสัญญาณรบกวน เนื่องจากว่ามีสัญญาณรบกวนอยู่ในสัญญาณจริงและในดิจิตเซอร์ ดังนั้น $\sin x/x$ ควรจะใช้อย่างระมัดระวังโดยเฉพาะการสุ่มตัวอย่างเพียง 2 ตัวอย่างต่อลูกคลื่น

ขั้นตอน $\sin x/x$ จะสร้างโอเวอร์ซุ่มและพรีซุ่มบนขอบที่ไม่เป็นที่ต้องการด้วย ดังนั้นจึงต้องการตัวอย่างข้อมูลอย่างน้อย 2 อันบนขอบของสัญญาณ ผู้ใช้สามารถตรวจสอบจุดข้อมูลดิบที่ได้มาในสโคปโดยใช้แสดง $\sin x/x$

สำหรับการแทนรูปคลื่นที่แม่นยำกว่า ดิจิตเซอร์ควรบันทึกอย่างน้อย 4 จุดตัวอย่างต่อลูกคลื่นของส่วนประกอบของสัญญาณไซน์ที่มีความถี่สูงสุด จุดตัวอย่างที่เพิ่มเติมเพิ่มอัตราผ่านของสัญญาณต่อสัญญาณรบกวน สำหรับการสอดแทรก $\sin x/x$ การสอดแทรกเส้นตรงสามารถแสดงรูปคลื่นที่ถูกต้องแม่นยำโดยปราศจากการขยายสัญญาณรบกวน เพื่อผลที่ดีจะต้องการอย่างน้อย 10 จุดตัวอย่างต่อลูกคลื่น

2.2.10 ประโยชน์ของหน่วยความจำที่ยาวในดิจิตอลสตอเรจอสซิลโลสโคป

การเพิ่มความยาวของหน่วยความจำในดิจิตอลสตอเรจอสซิลโลสโคปนำมาซึ่งข้อดีหลายอย่าง คือ

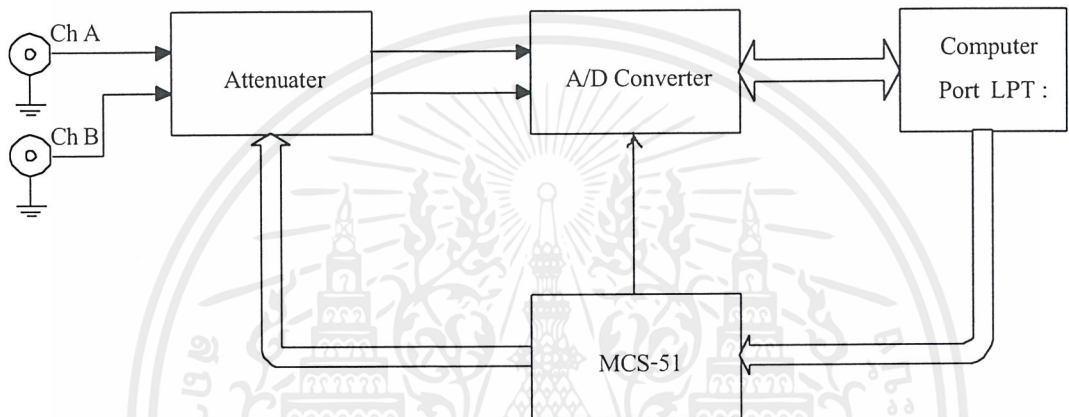
- 1.) ไม่สูญเสียข้อมูลของรูปคลื่น เนื่องจากอัตราการสุ่มลูกคลื่นที่มีประสิทธิภาพสูง
- 2.) จับสัญญาณกติกได้โดยไม่ทำให้รูปคลื่นผิดรูปหรือบิดเบือนไป
- 3.) ความละเอียดเรื่องความถี่และเวลาที่ดีกว่า
- 4.) การจับที่เชื่อถือได้ของเหตุการณ์ซึ่งไม่สามารถทำนายได้ในเวลานั้น
- 5.) ไม่เสียเวลาระหว่างเหตุการณ์แต่ละเหตุการณ์ที่ได้รับ

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 การออกแบบวงจร

วิธีการออกแบบวงจรที่ใช้งานในเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ ทำโดยการกำหนดขั้นตอนการทำงานเป็นแผนผังการทำงาน

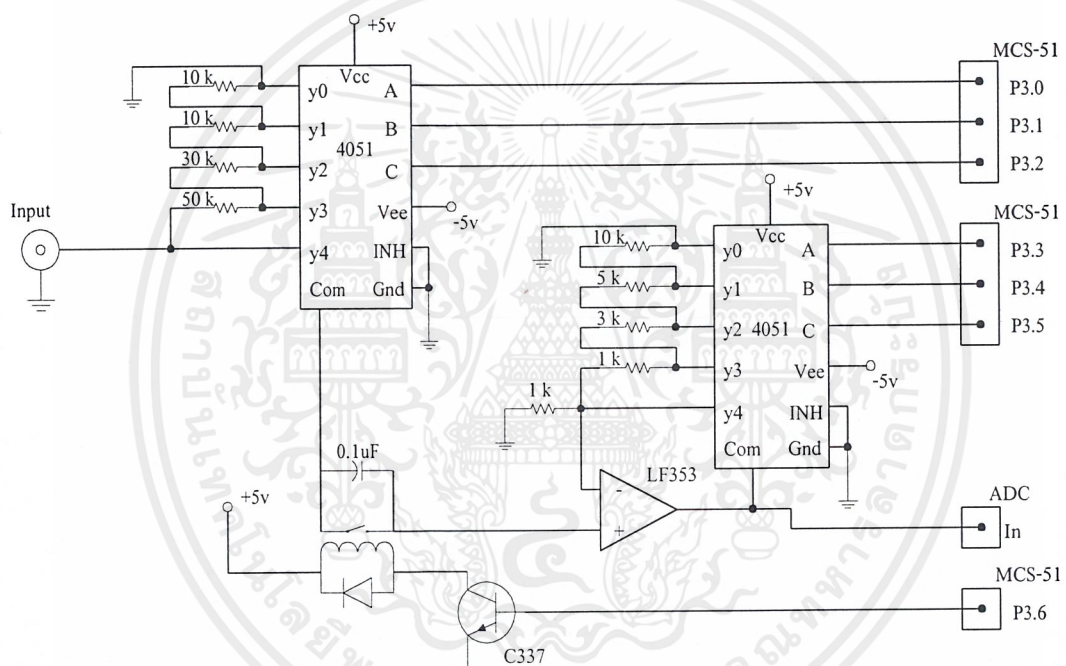


รูปที่ 3.1 แผนผังการทำงานของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์

การออกแบบวงจรสามารถแบ่งออกเป็นส่วนใหญ่ได้ดัง รูปที่ 3.1 วงจรแรกที่ทำหน้าที่รับสัญญาณอินพุตก็คือ วงจร Attenuater ซึ่งใช้สำหรับขยายและลดทอนสัญญาณให้เหมาะสมก่อนนำเข้ามาแปลงสัญญาณ โดยมี MCS-51 ควบคุมการการเลือกอัตราขยายและลดทอนแรงดัน ภาคการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลนั้นคือ A/D Converter จะทำหน้าที่รับสัญญาณมาเปลี่ยนเป็นสัญญาณดิจิตอลและส่งเข้าไปยังพอร์ต LPT : ของคอมพิวเตอร์ เพื่อนำไปแสดงผลการทำงาน และสร้างสัญญาณนาฬิกาขึ้นจากโปรแกรมมาควบคุมการทำงานของ วงจร A/D Converter

3.2 ภาควงจรขยายและลดทอนสัญญาณ (Attenuater)

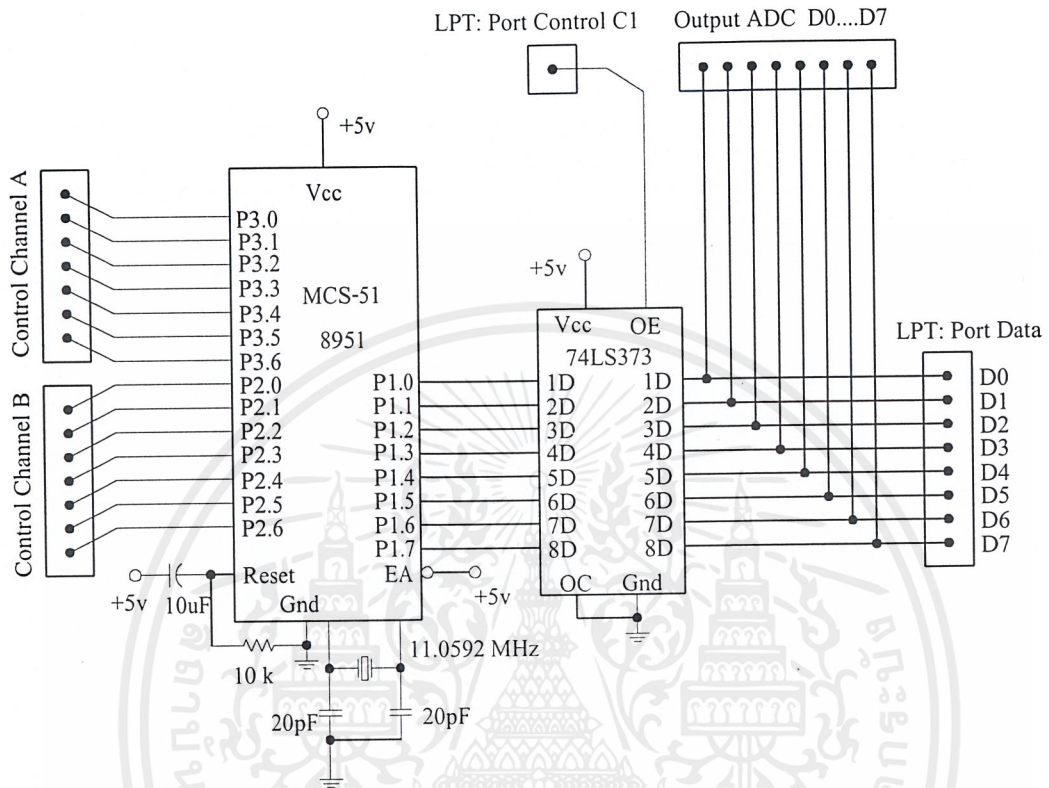
สัญญาณที่เข้ามามีขนาดแตกต่างกัน จึงต้องมีการปรับระดับสัญญาณให้เหมาะสมก่อนที่จะไปยังภาคอื่นต่อไป เพราะหากไม่มีการควบคุมแรงดันทางอินพุตอาจจะเกิดความเสียหายกับตัวอุปกรณ์ที่ใช้ภายในวงจรได้ จึงใช้การลดทอนสัญญาณโดยใช้หลักการแบ่งแรงดัน (Voltage Divider) และเนื่องจากสัญญาณที่จะวัดมีแถบความถี่ (Bandwidth) กว้าง ดังนั้นการลดทอนสัญญาณอินพุตจะต้องเป็นอิสระต่อความถี่ซึ่งต้องใช้วงจรลดทอนแบบชดเชยความถี่ (Compensated Attenuater) ในวงจรจะมีการลดทอนทั้งหมดอยู่ 3 ระดับคือ 0.5 เท่า, 0.2 เท่า และ 0.1 เท่า ส่วนการขยายมีอยู่ 4 ระดับ คือ 2 เท่า, 5 เท่า, 10 เท่า และ 20 เท่า ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 วงจรขยายและลดทอนสัญญาณ

ใช้ MCS-51 เป็นตัวสั่งการเลือกอัตราการขยายหรืออัตราการลดทอนให้มีความเหมาะสมกับอินพุตที่รับเข้ามา โดยใช้พอร์ต P3.0 ถึง P3.6 ควบคุมการเลือกของ Channel A และใช้พอร์ต P2.0 ถึง P2.6 ควบคุมการเลือกของ Channel B ส่วนพอร์ต P1.0 ถึง P1.7 ต่อกับพอร์ตขนานเพื่อใช้รับคำสั่งในการเลือกว่าจะใช้อินพุต Channel A หรือ Channel B และเลือกการขยายหรือลดทอน

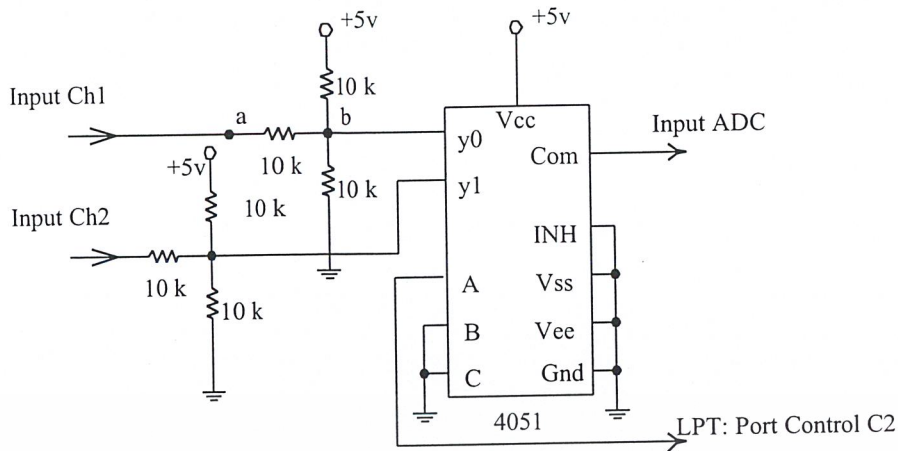
สัญญาณ แต่ละมีไอซีเบอร์ 74LS373 ต่อกันระหว่างพอร์ตขนานกับตัว MSC-51 ไว้เพื่อป้องกันการเสียหายของข้อมูล ดังรูปที่ 3.3



รูปที่ 3.3 การต่อ MCS-51 เพื่อเลือกการทำงานของวงจร Attenuator

3.3 ภาควจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล

วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลที่ใช้เป็นเบอร์ ADC0820 ทางอินพุตของวงจรจะต่อวงจรยกระดับสัญญาณและมีการใช้มัลติเพล็กซ์เพื่อเพิ่มช่องสัญญาณให้มากขึ้นและทางเอาต์พุตของ ADC0820 ต่อกับ ไอซีแลตข้อมูลป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลที่ส่งออก



รูปที่ 3.4 วงจรระดับสัญญาณและวงจรมัลติเพล็กซ์

3.3.1 วงจรระดับสัญญาณ

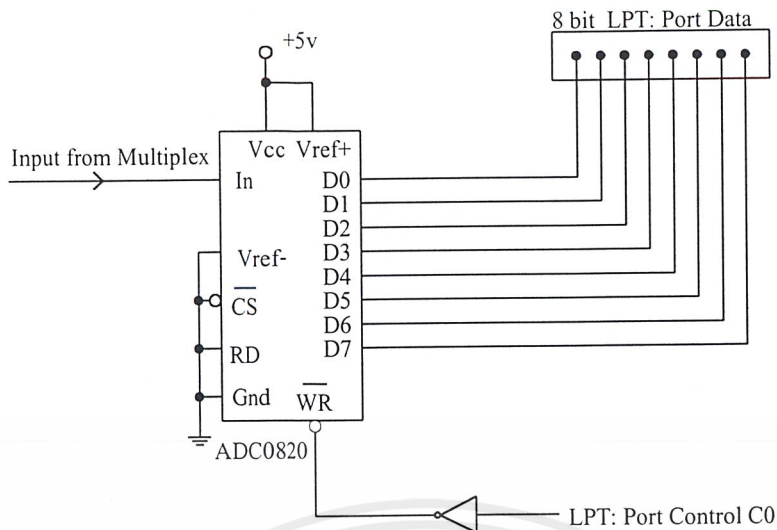
การทำงานของวงจรระดับสัญญาณอาศัยหลักการของวงจรแบ่งแรงดัน จากวงจรที่จุด b ดังรูปที่ 3.4 จะมีค่าแรงดันเท่ากับ 2.5 V เมื่อเทียบกับกราวด์ และในลักษณะการใช้งานเมื่อป้อนอินพุตเข้าที่จุด a หากเป็นรูปคลื่นในลักษณะของค่าแรงดัน V_{pp} สัญญาณสวิงทั้งด้านบวกและด้านลบ จะทำให้สัญญาณทางด้านลบถูกยกระดับบวกเพิ่มขึ้นมา 2.5 V และสัญญาณที่ถูกยกระดับขึ้นมานี้จะถูกลดขนาดแรงดันลงครึ่งหนึ่งของสัญญาณอินพุต

3.3.2 วงจรเลือกช่องทางสัญญาณ

คือวงจรมัลติเพล็กซ์ใช้ไอซี 4051 ซึ่งสามารถรับสัญญาณอินพุตได้มากถึง 8 ช่องสัญญาณ แต่จะเลือกใช้เพียง 2 ช่องสัญญาณเท่านั้น การควบคุมไอซี 4051 ให้ทำงานจะใช้การสั่งงานโดยผ่าน Port Control C2 ให้เลือกช่องสัญญาณการทำงานโดยปกติจะเลือกไว้ที่ Channel A เมื่อต้องการเปลี่ยนช่องสัญญาณก็ส่งโปรแกรมเปลี่ยนระดับลอจิกที่ Port Control C2 อีกครั้ง

3.3.3 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล

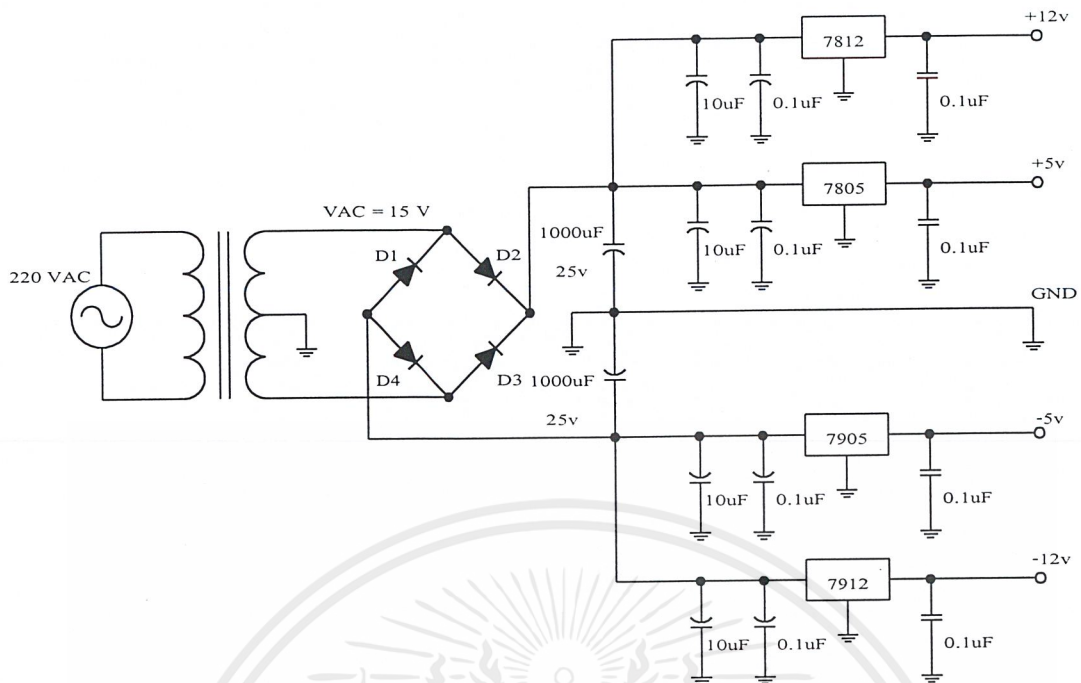
วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลใช้ไอซีเบอร์ ADC0820 ทำหน้าที่ในการแปลงซึ่งเป็นแบบ Half-Flash มีความละเอียด 8 บิต มีอัตราการแปลงสัญญาณที่ $2.5\mu s$ หรือมีอัตราการสุ่มสูงสุดที่ 400 kHz ใช้สัญญาณนาฬิกาที่ผลิตจากโปรแกรมภาษาซีควบคุมการทำงาน ดังรูปที่ 3.5 ซึ่งจะทำให้ได้ข้อมูลที่ผ่านการแปลงออกมามีความถูกต้อง เอาต์พุตมีไอซีแลตข้อมูลเบอร์ 74LS374 เพื่อรักษาสถานะของข้อมูลให้คงค่าไว้ก่อนที่จะทำการอ่านเข้าไปและใช้ Port Control C3 ในการควบคุมการทำงานของไอซี 74LS374



รูปที่ 3.5 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล

3.4 วงจรแหล่งจ่ายไฟ

วงจรภายในโครงการใช้แรงดันไฟเลี้ยงวงจรอยู่ 5 ระดับคือ ระดับแรงดัน +12 โวลต์, ระดับแรงดัน +5 โวลต์ กราวด์ 0 โวลต์, ระดับแรงดัน -5 โวลต์ และระดับแรงดัน -12 โวลต์ วงจรที่จะต้องใช้ระดับแรงดันไฟบวก +12 โวลต์และระดับแรงดันไฟ -12 โวลต์ คือออปแอมป์ที่ใช้ในการขยายสัญญาณเบอร์ LF 353 และไอซีมัลติเพิลิเคเตอร์ 4051 ส่วนวงจรอื่นใช้แหล่งจ่ายระดับแรงดัน +5 โวลต์ การสร้างแหล่งจ่ายไฟ จึงต้องสร้างแหล่งจ่ายไฟที่มีหลายระดับ โดยใช้ไอซี Regulator เบอร์ 7812 ในการสร้างระดับแรงดันไฟ+12โวลต์ ไอซี Regulator เบอร์ 7805 ในการสร้างระดับแรงดันไฟ + 5 โวลต์ ไอซี Regulator เบอร์ 7905 สร้างระดับแรงดันไฟ -5 โวลต์และไอซี Regulator เบอร์ 7912 สร้างระดับแรงดันไฟ -12 โวลต์ ดังแสดงในรูปที่ 3.6



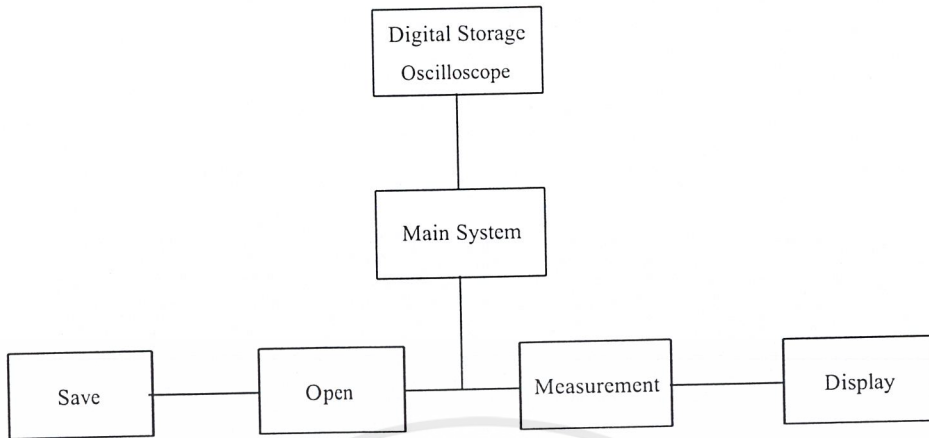
รูปที่ 3.6 วงจรแหล่งจ่ายไฟ

3.5 โปรแกรมการทำงาน

ในส่วนของการออกแบบซอร์ฟแวร์นั้น ก่อนอื่นต้องศึกษาหลักการการทำงานของฮาร์ดแวร์ เพื่อที่จะสามารถเขียนโปรแกรมควบคุมการทำงานของฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ และศึกษาเกี่ยวกับการใช้คำสั่งที่จะใช้ในการควบคุมอุปกรณ์ ศึกษาการเขียนกราฟฟิกเพื่อใช้ในการเขียนส่วนแสดงผลการทำงานออกทางจอคอมพิวเตอร์ให้มีความสวยงามน่าใช้งาน ภายในโปรแกรมจะประกอบไปด้วยฟังก์ชันการทำงานต่าง ที่ใช้สำหรับควบคุมดังต่อไปนี้

- 1) ฟังก์ชันในการแสดงผลสัญญาณทางหน้าจอคอมพิวเตอร์
- 2) ฟังก์ชันที่ใช้ในการควบคุมวงจรมองขยายและลดทอนสัญญาณ
- 3) ฟังก์ชันที่ใช้ในการเก็บบันทึกข้อมูลของสัญญาณ
- 4) ฟังก์ชันที่ใช้อ่านข้อมูลที่บันทึกไว้ในแผ่นดิสก์เก็ต

รูปที่ 3.7 เป็นผังแสดงขั้นตอนการทำงาน ส่วนผังการทำงานของโปรแกรมทั้งหมดสามารถดูได้ที่ ภาคผนวก ค ผังการทำงานและโปรแกรม



รูปที่ 3.7 ฟังงานการทำงานแสดงขั้นตอนการทำงาน



บทที่ 4

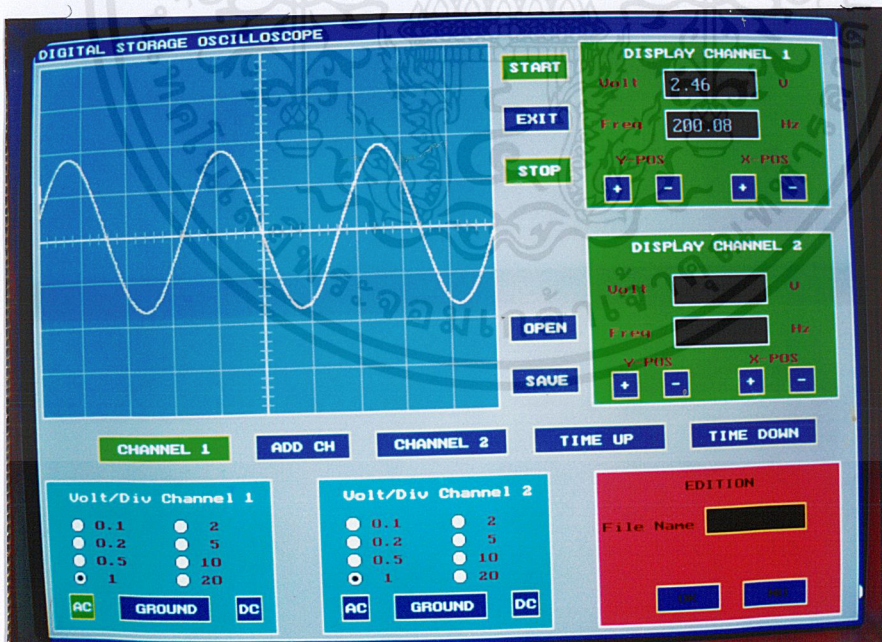
การทดลองและผลการทดลอง

4.1 การทดลองรับสัญญาณอินพุต

เนื่องจากโครงการนี้จัดทำจัดทำเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ขึ้น การทดลองจึงต้องมีการวัดสัญญาณอินพุตที่มีหลายลักษณะและหลายความถี่ และได้ใช้ Function Generator ในการผลิตสัญญาณทางอินพุต เพราะสามารถปรับเปลี่ยนความถี่และรูปคลื่นที่จะนำมาทดลองได้สะดวก

4.1.1 การทดลองรับสัญญาณคลื่นไซน์ความถี่ 200 Hz

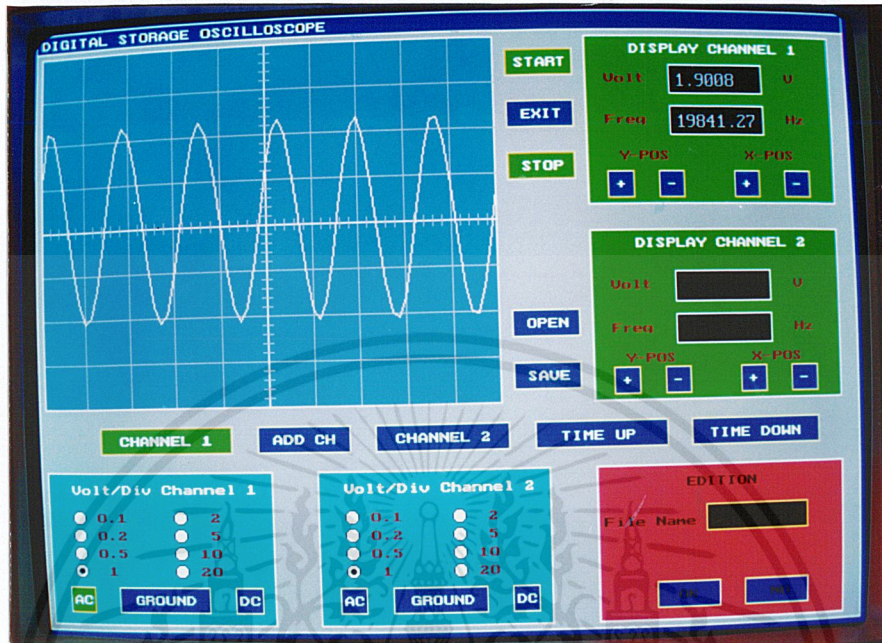
1. ปรับ Function Generator ให้สร้างรูปคลื่นไซน์ มีแอมพลิจูดของสัญญาณเท่ากับ 5 โวลต์ ความถี่เท่ากับ 200 Hz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.1 สัญญาณอินพุตเป็นคลื่นไซน์ความถี่ 200 Hz

4.1.2 การทดลองรับสัญญาณคลื่นไซน์ความถี่ 20 kHz

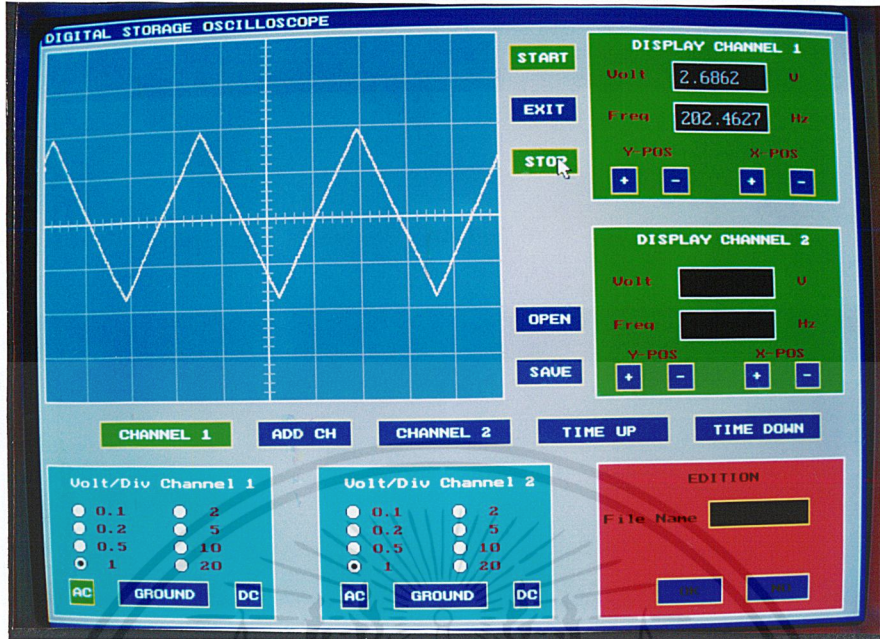
ทำตามการทดลองที่ 4.1.1 แต่เปลี่ยนความถี่ของสัญญาณจาก 200 Hz เป็น 20 kHz



รูปที่ 4.2 สัญญาณอินพุตเป็นคลื่น ไซน์ความถี่ 20 kHz

4.1.3 การทดลองรับสัญญาณคลื่นรูปสามเหลี่ยมความถี่ 200 Hz

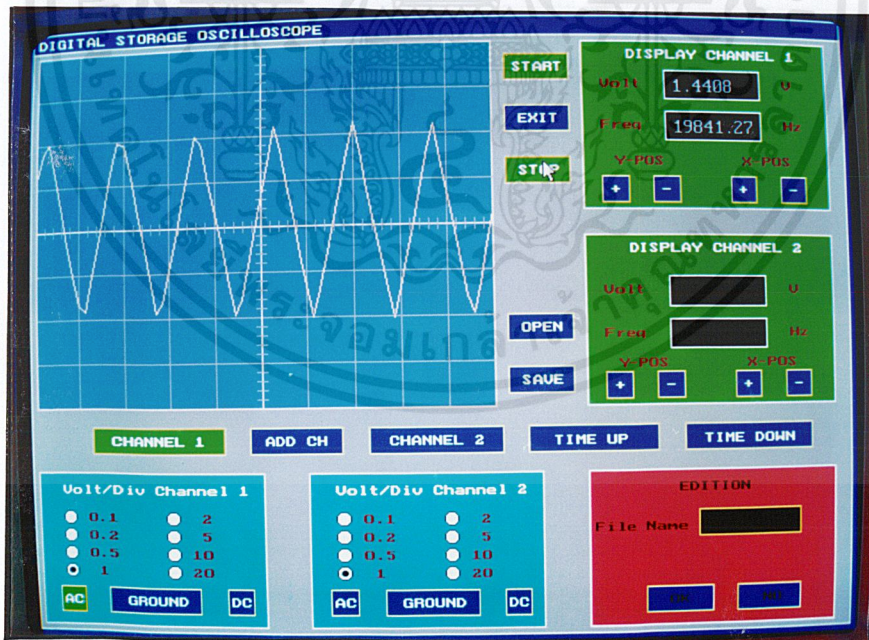
1. ปรับ Function Generator ให้สร้างคลื่นรูปสามเหลี่ยม โดยมีแอมพลิจูดของสัญญาณเท่ากับ 5 โวลต์ ความถี่เท่ากับ 200 Hz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.3 สัญญาณอินพุตเป็นคลื่นรูปสามเหลี่ยมความถี่ 200 Hz

4.1.4 การทดลองรับสัญญาณคลื่นรูปสามเหลี่ยมความถี่ 20 kHz

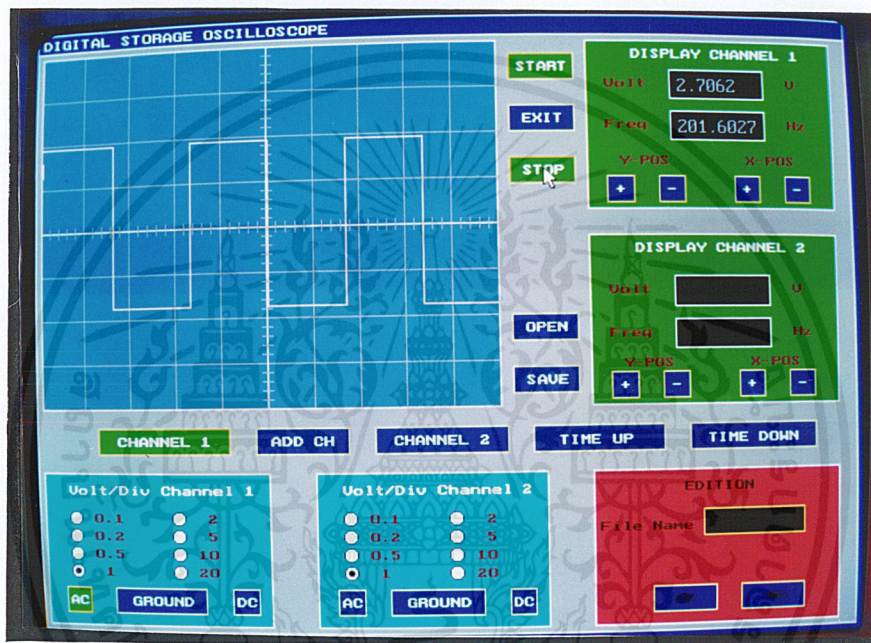
ทำตามการทดลองที่ 4.1.3 แต่เปลี่ยนความถี่ของสัญญาณจาก 200 Hz เป็น 20 kHz



รูปที่ 4.4 สัญญาณอินพุตเป็นคลื่นรูปสามเหลี่ยมความถี่ 20 kHz

4.1.5 การทดลองรับสัญญาณคลื่นรูปสี่เหลี่ยมความถี่ 200 Hz

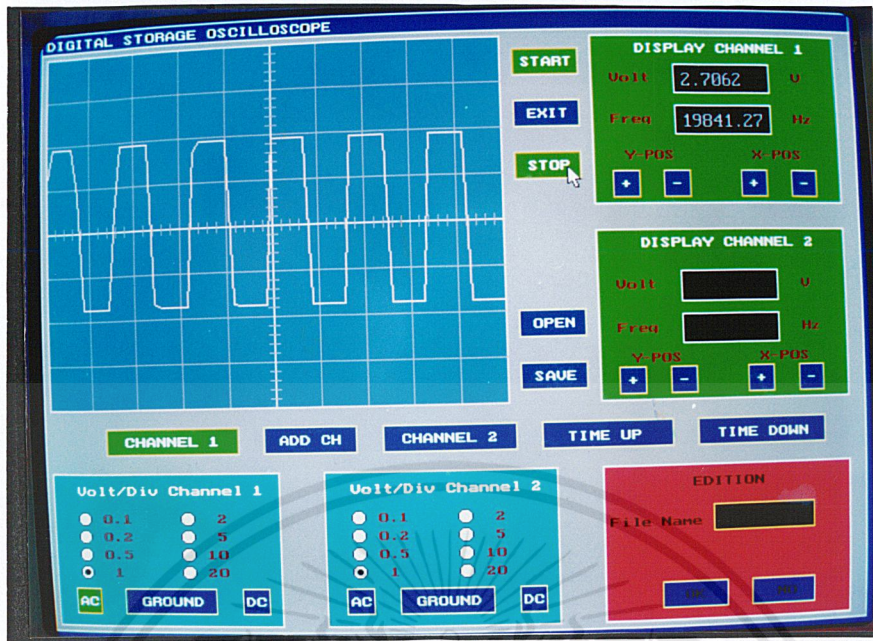
1. ปรับ Function Generator ให้สร้างคลื่นรูปสี่เหลี่ยม โดยมีแอมพลิจูดของสัญญาณเท่ากับ 5 โวลต์ ความถี่เท่ากับ 200 Hz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.5 สัญญาณอินพุตเป็นคลื่นรูปสี่เหลี่ยมความถี่ 200 Hz

4.1.6 การทดลองรับสัญญาณคลื่นรูปสี่เหลี่ยมความถี่ 20 kHz

ทำตามการทดลองที่ 4.1.5 แต่เปลี่ยนความถี่ของสัญญาณจาก 200 Hz เป็น 20 kHz



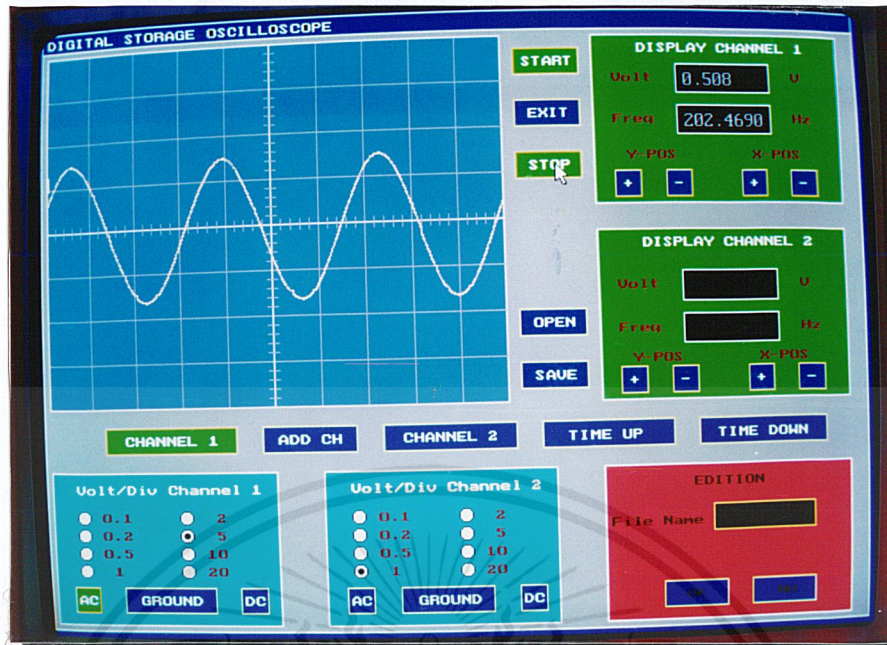
รูปที่ 4.6 สัญญาณอินพุตเป็นคลื่นรูปสี่เหลี่ยมความถี่ 20 kHz

4.2 การทดลองควบคุมการทำงานของวงจรรขยายและลดทอนสัญญาณ

เป็นการทดลองขยายสัญญาณและลดทอนสัญญาณที่เข้ามาทางอินพุตของ ADC0820 ให้มีความเหมาะสมก่อนที่จะนำไปทำการแปลง เนื่องจาก ADC0820 มีความสามารถรับแรงดันทางอินพุตอยู่ในช่วงระหว่าง 0 ถึง 5 โวลต์ หากสัญญาณอินพุตที่มีแรงดันสูงกว่านี้เข้ามาจะทำให้การแปลงข้อมูลเกิดความผิดพลาดและจะเกิดความเสียหายกับตัวไอซีได้ ใช้ความถี่ 2 kHz ทดลอง

4.2.1 การทดลองขยายสัญญาณคลื่นไซน์ขนาด 0.5 โวลต์

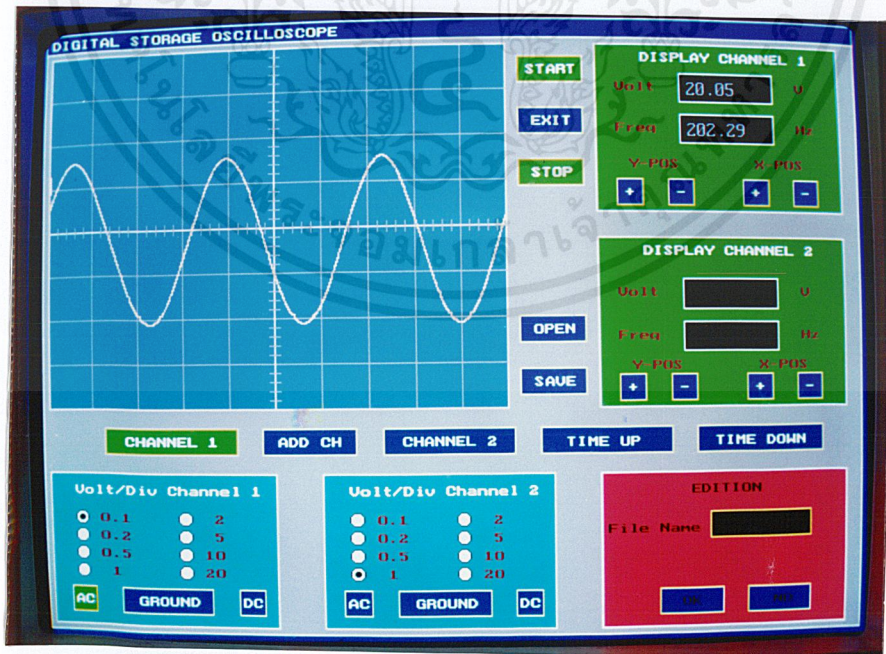
1. ปรับ Function Generator สร้างรูปคลื่น ไซน์มีแอมพลิจูดของสัญญาณเท่ากับ 0.5 โวลต์ ความถี่เท่ากับ 2 kHz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.7 การขยายสัญญาณคลื่นไซน์ขนาด 0.5 โวลต์

4.2.2 การทดลองลดทอนสัญญาณคลื่นไซน์ขนาด 20 โวลต์

ทำการทดลองตามการทดลองที่ 4.2.1 แต่เปลี่ยนแอมพลิจูดของสัญญาณจาก 0.5 โวลต์เป็น 20 โวลต์

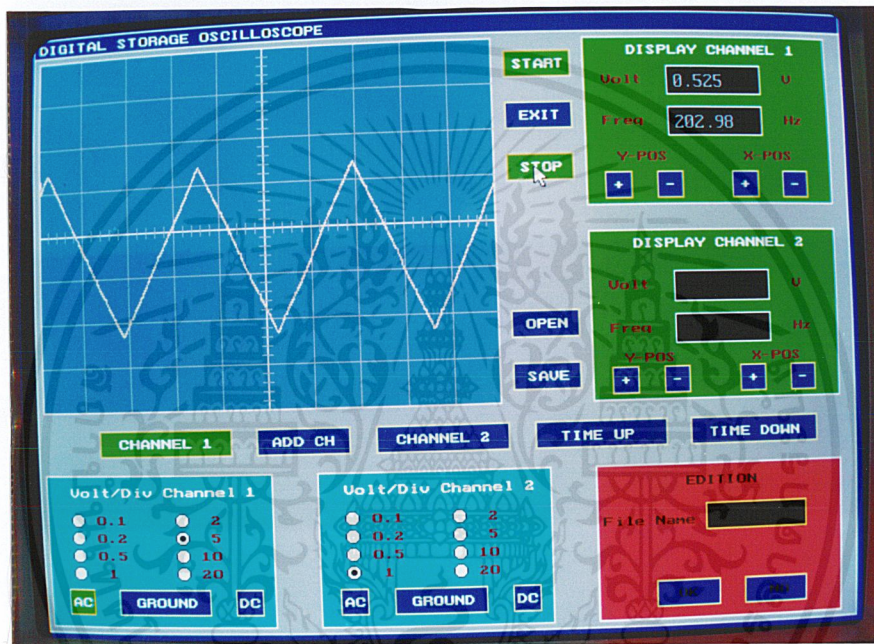


รูปที่ 4.8 การลดทอนสัญญาณคลื่นไซน์ขนาด 20 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองขยายสัญญาณคลื่นรูปสามเหลี่ยมขนาด 0.5 โวลต์

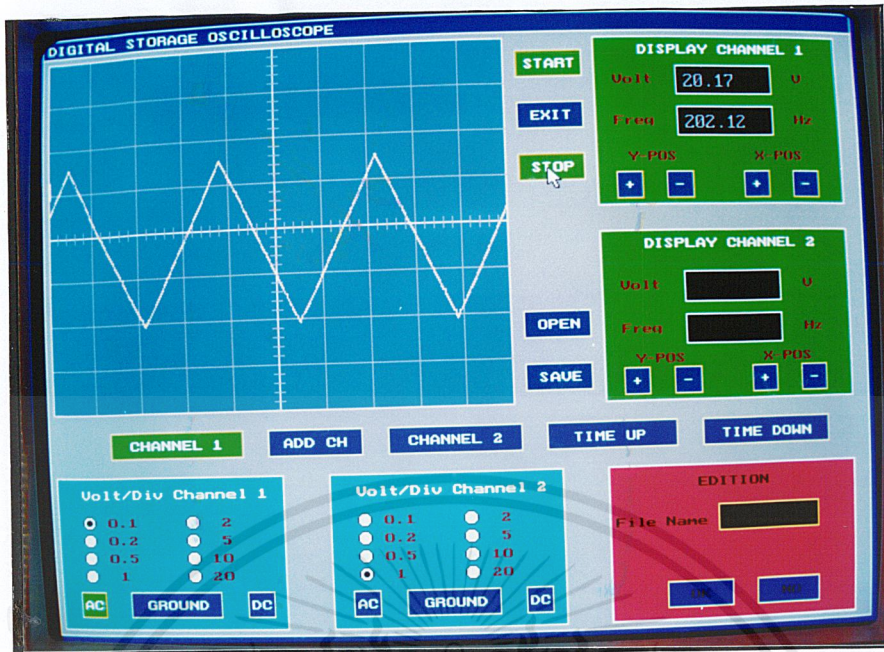
1. ปรับ Function Generator ให้สร้างคลื่นรูปสามเหลี่ยม มีแอมพลิจูดของสัญญาณเท่ากับ 0.5 โวลต์ ความถี่เท่ากับ 2 kHz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.9 การขยายสัญญาณคลื่นรูปสามเหลี่ยมขนาด 0.5 โวลต์

4.2.4 การทดลองขยายสัญญาณคลื่นรูปสามเหลี่ยมขนาด 20 โวลต์

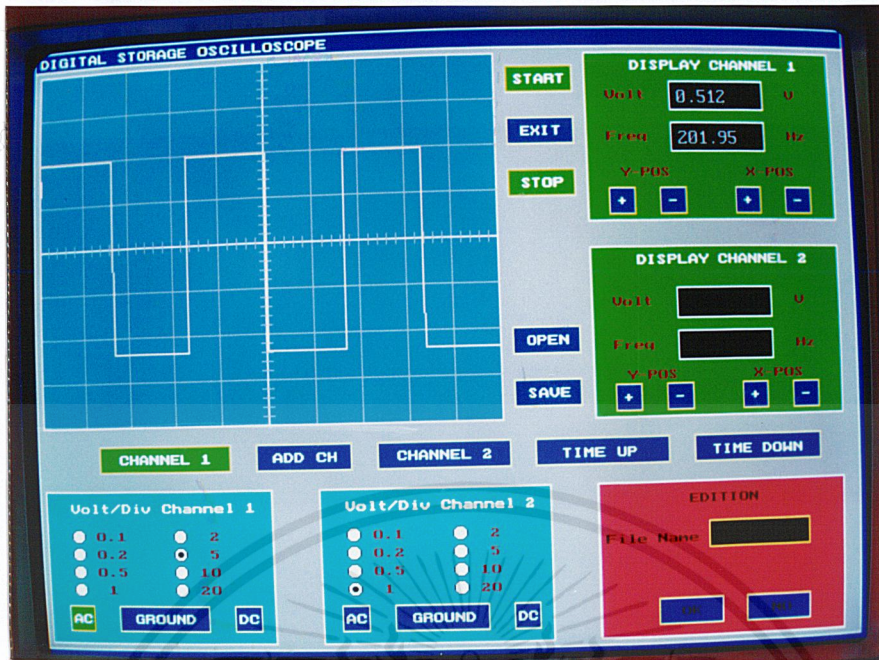
ทำการทดลองตามการทดลองที่ 4.2.3 แต่เปลี่ยนแอมพลิจูดของสัญญาณจาก 0.5 โวลต์เป็น 20 โวลต์



รูปที่ 4.10 การลดทอนสัญญาณคลื่นรูปสามเหลี่ยมขนาด 20 โวลต์

4.2.5 การทดลองขยายสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 0.5 โวลต์

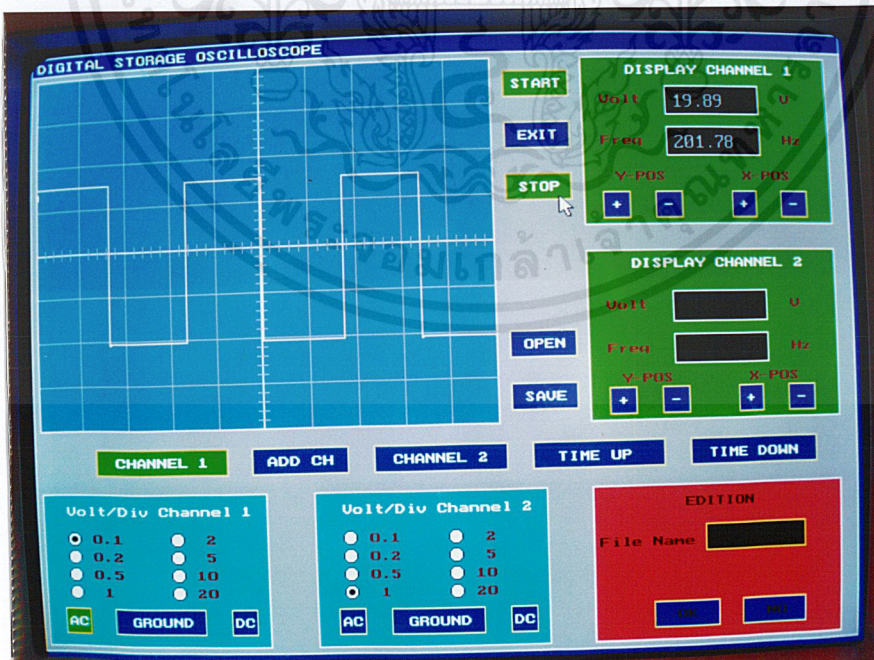
1. ปรับ Function Generator สร้างคลื่นรูปสี่เหลี่ยมแอมพลิจูดของสัญญาณเท่ากับ 0.5 โวลต์ ความถี่เท่ากับ 2 kHz
2. นำสัญญาณจาก Function Generator ป้อนให้กับอินพุตของเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ Channel A หรือ Channel B ก็ได้
3. เปิดเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์
4. รันโปรแกรม Scope.exe



รูปที่ 4.11 การขยายสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 0.5 โวลต์

4.2.6 การทดลองขยายสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 20 โวลต์

ทำการทดลองตามการทดลองที่ 4.2.5 แต่เปลี่ยนแอมพลิจูดของสัญญาณจาก 0.5 โวลต์เป็น 20 โวลต์



รูปที่ 4.12 การลดทอนสัญญาณคลื่นรูปสี่เหลี่ยมขนาด 20 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

ในการสร้างเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์ตามที่ได้นำเสนอไปในปริิญาณิพนธ์ฉบับนี้ ผลที่ได้ก็คือคือ จิตความสามารถจริงของเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์ต่ำกว่าที่ได้ตั้งเกณฑ์เอาไว้ ในด้านความถี่นั้นสามารถแสดงสัญญาณได้ตั้งแต่ 1 Hz จนถึง 20 kHz ถ้าสัญญาณมีความถี่สูงกว่านี้จะได้รูปสัญญาณที่ผิดเพี้ยนไป แต่สัญญาณที่นำมาแสดงนั้นไม่สม่าเสมอกันสังเกตได้จากรูปในผลการทดลองจะเห็นว่ามึบางรูปคลื่นที่บิดเบี้ยวไป

ส่วนในเรื่องของความสามารถในการรับแรงดันด้านอินพุตนั้นยังไม่สามารถควบคุมได้ตามความต้องการเนื่องยังไม่สามารถเขียนโปรแกรมควบคุมให้ดีได้ แต่อย่างไรก็ตามการสร้างเครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์นี้ขึ้นมาทำให้ได้รับประโยชน์ดังต่อไปนี้

1. จากการทดลองของโครงการนี้ ทำให้เราทราบถึงเทคนิคในการเขียนโปรแกรมภาษาซี และการประยุกต์คำสั่งในการใช้งานได้หลายรูปแบบ ซึ่งทำให้สามารถนำความรู้ที่ได้ไปใช้ในการเรียนการสอนหรืองานที่เกี่ยวข้องได้

2. ทำให้ทราบถึงคุณสมบัติ และการทำงานของไอซีแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิทัล ซึ่งจะเป็ประโยชน์มากเมื่อต้องการใช้งานในอนาคต

3. สามารถนำโครงการนี้ไปใช้ประโยชน์ได้จริงเพื่อวัดสัญญาณแอนะลอกชนิดต่างๆ ตามความต้องการของผู้ใช้งาน และยังสามารถพัฒนาให้มีจิตความสามารถสูงขึ้นจนมีความใกล้เคียงกับออสซิลโลสโคปจริง

4. โครงการนี้ใช้การแสดงผลทางหน้าจออมินเตอร์ และสามารถควบคุมการทำงานโดยผ่านโปรแกรมที่แสดงผลอยู่เป็นการใช้งานที่สะดวก

5.2 ปัญหาที่เกิดขึ้นในการจัดทำโครงการ

1. มีสัญญาณรบกวนเกิดขึ้นกับไฟเลี้ยงของอุปกรณ์แต่ละตัวภายในวงจร ทำให้การทำงานของอุปกรณ์แต่ละตัวถูกรบกวนเอาต์พุตที่ได้จึงมีความผิดพลาด

2. ใช้สัญญาณนาฬิกาที่ป้อนให้กับ ADC0820 มีความถี่ 400 กิโลเฮิร์ตซ์ ทำให้การสุ่มตัวอย่างของ ADC0820 มีความผิดพลาดทางข้อมูลที่แปลงออกมา

3. อินพุตที่รับได้ถูกจำกัดให้สามารถรับเพียงหนึ่งอินพุต และมีขนาดไม่เกิน 0 ถึง 5 โวลต์
4. ข้อมูลที่รับได้หลังจากผ่านการแปลงเป็นสัญญาณดิจิทัล เมื่อนำข้อมูลไปพล็อตเป็นรูปสัญญาณ โดยผ่านโปรแกรมภาษาซีผลที่ได้จะเป็นรูปคลื่นที่มีความบิดเบี้ยวของสัญญาณเกิดขึ้นบ่อย

5.3 การแก้ไข

1. ต่อตัวเก็บประจุ 0.1 ไมโครฟารัดชนิดเซรามิกอนุกรมกับขา Vcc ของไอซีแต่ละตัวจะช่วยลดสัญญาณรบกวนลดลง
2. ลดความถี่สุ่มที่ป้อนให้กับ ADC0820 เหลือเพียง 200 กิโลเฮิร์ตซ์ ทำให้ข้อมูลที่ได้อจากการสุ่มมีความผิดพลาดน้อยลง
3. ต่อดวงจรมัลติเพล็กซ์ก่อนเข้าอินพุตของ ADC0820 ทำให้สามารถรับสัญญาณได้หลายช่องทาง และต่อดวงจรถะดับสัญญาณที่ได้จากเอาต์พุตของวงจรมัลติเพล็กซ์จะทำให้สัญญาณอยู่ในช่วงบวกทั้งหมด
4. ในการนำข้อมูลไปพล็อตเป็นรูปสัญญาณ ต้องทำการเลือกเฉพาะข้อมูลบางชุดมาพล็อตจะทำให้รูปที่ได้มีความสวยงามขึ้น

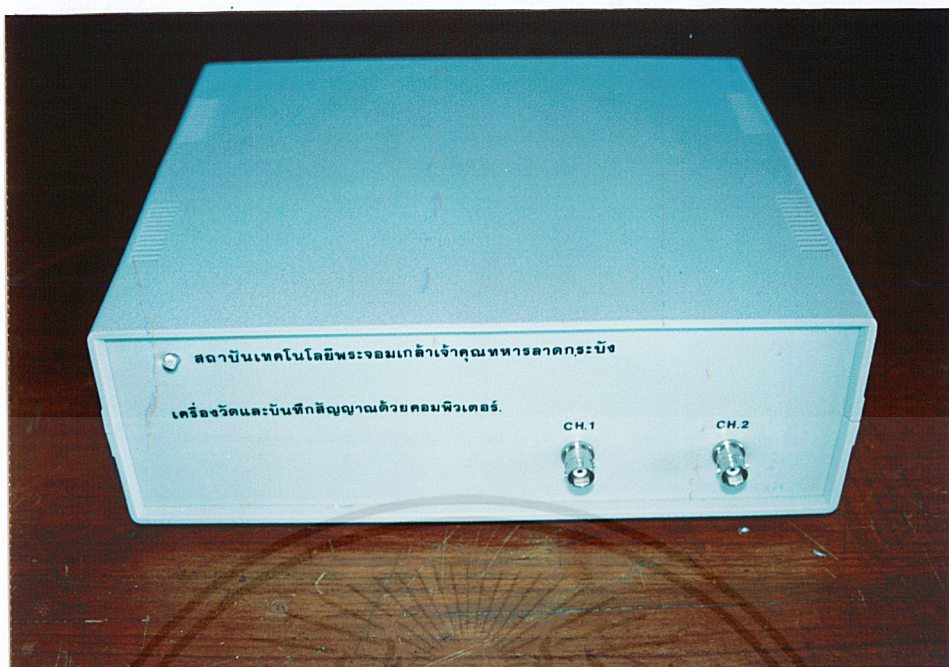
5.4 แนวทางในการพัฒนา

1. ควรใช้ไอซีแปลงสัญญาณแอนะล็อกเป็นดิจิทัลที่มีความสามารถในการแปลงข้อมูลที่เร็วกว่าเบอร์ ADC0820 เช่นเบอร์ CA3318 ซึ่งมีความเร็วในการแปลงข้อมูลสูงมากทำงานที่ความถี่สูงสุดถึง 15 MHz
2. ในการออกแบบควรให้แรงดันที่ป้อนให้กับไฟเลี้ยงของไอซีแต่ละตัวมี สัญญาณรบกวนให้น้อยที่สุดเพราะจะเป็นผลให้อเอาต์พุตของอุปกรณ์มีสัญญาณรบกวนเกิดขึ้นด้วย
3. การเขียนโปรแกรมการเลือกคอมพิวเตอร์ที่ทำงานบนระบบปฏิบัติการ Windows 95/98 ได้โดยที่ไม่ทำให้การทำงานทางฮาร์ดแวร์ช้าลง จะทำให้โปรแกรมมีความน่าใช้งานมากกว่าโปรแกรมที่ทำงานบนระบบปฏิบัติการดอส



ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

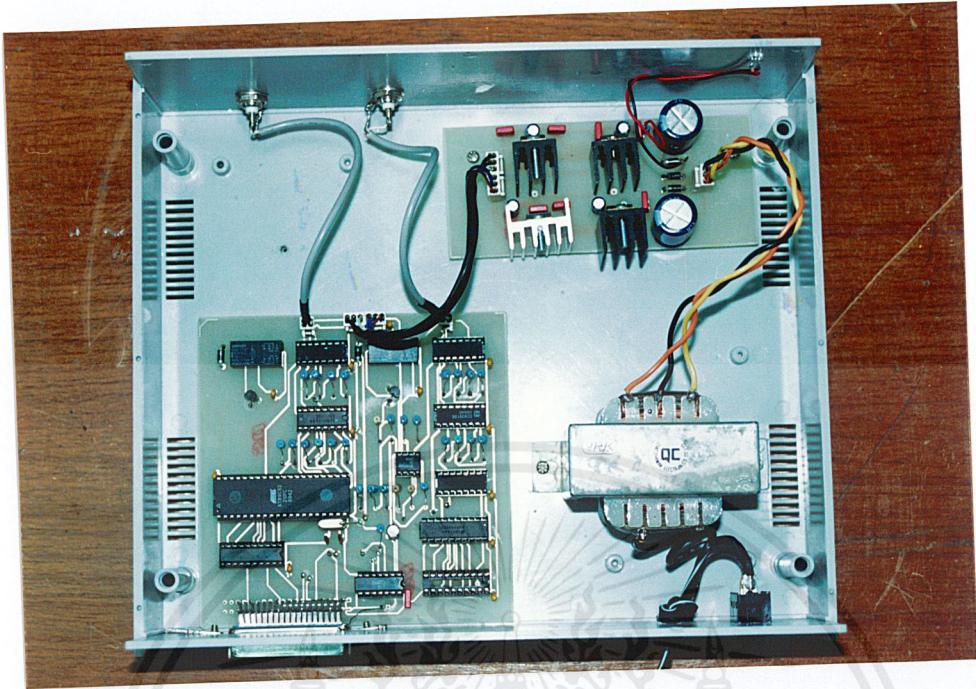


รูปที่ ก.1 ด้านหน้าเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์



รูปที่ ก.2 ด้านหลังเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



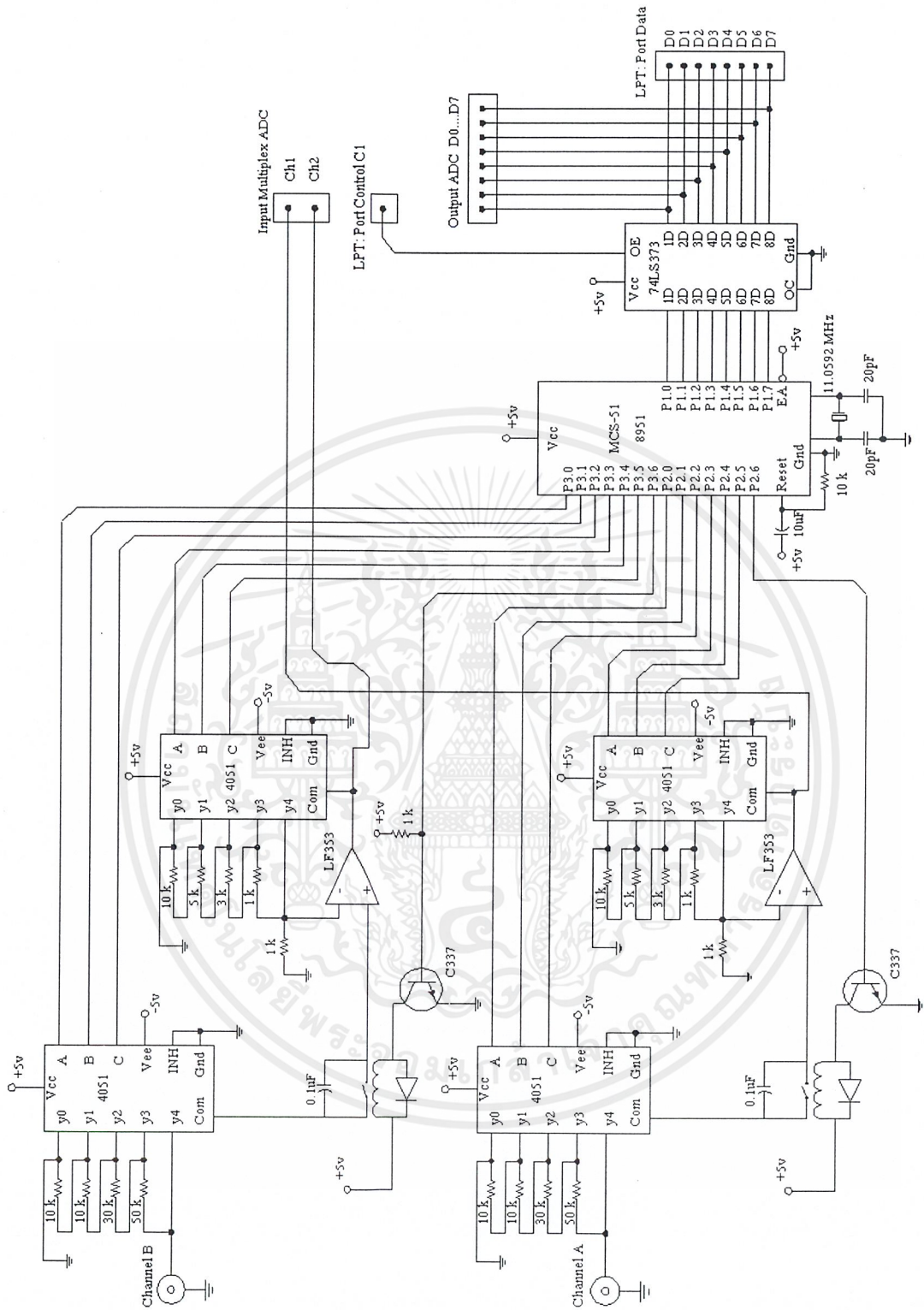
รูปที่ ก.3 วงจรภายในเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



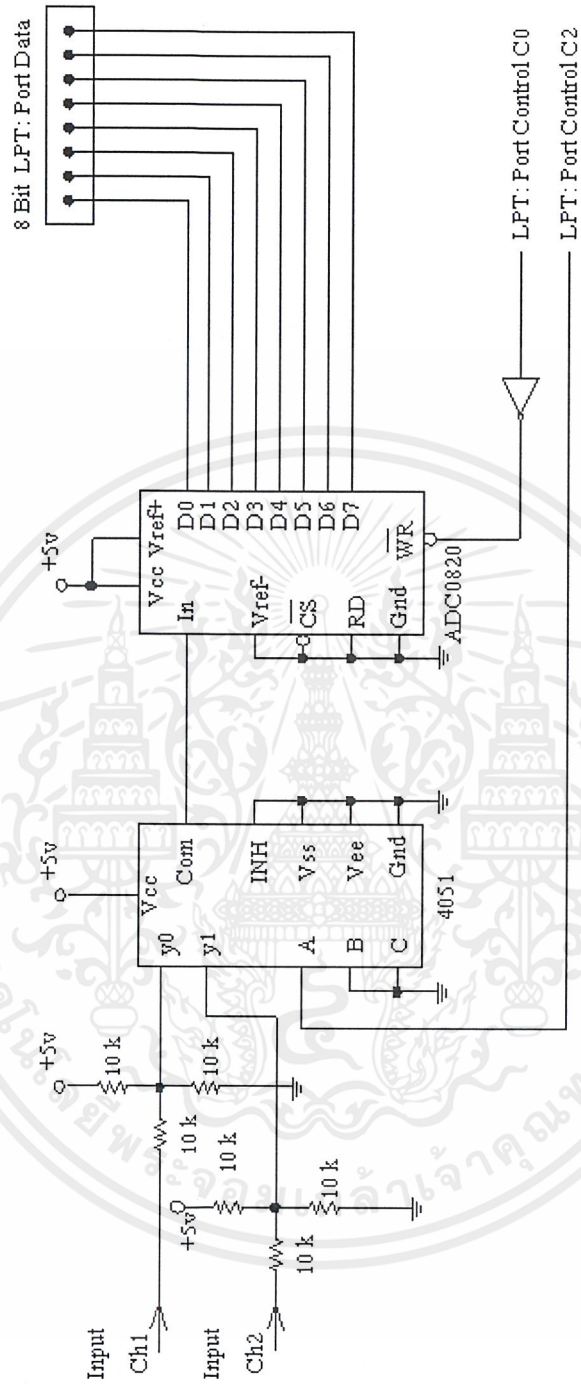
ภาคผนวก ข
วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



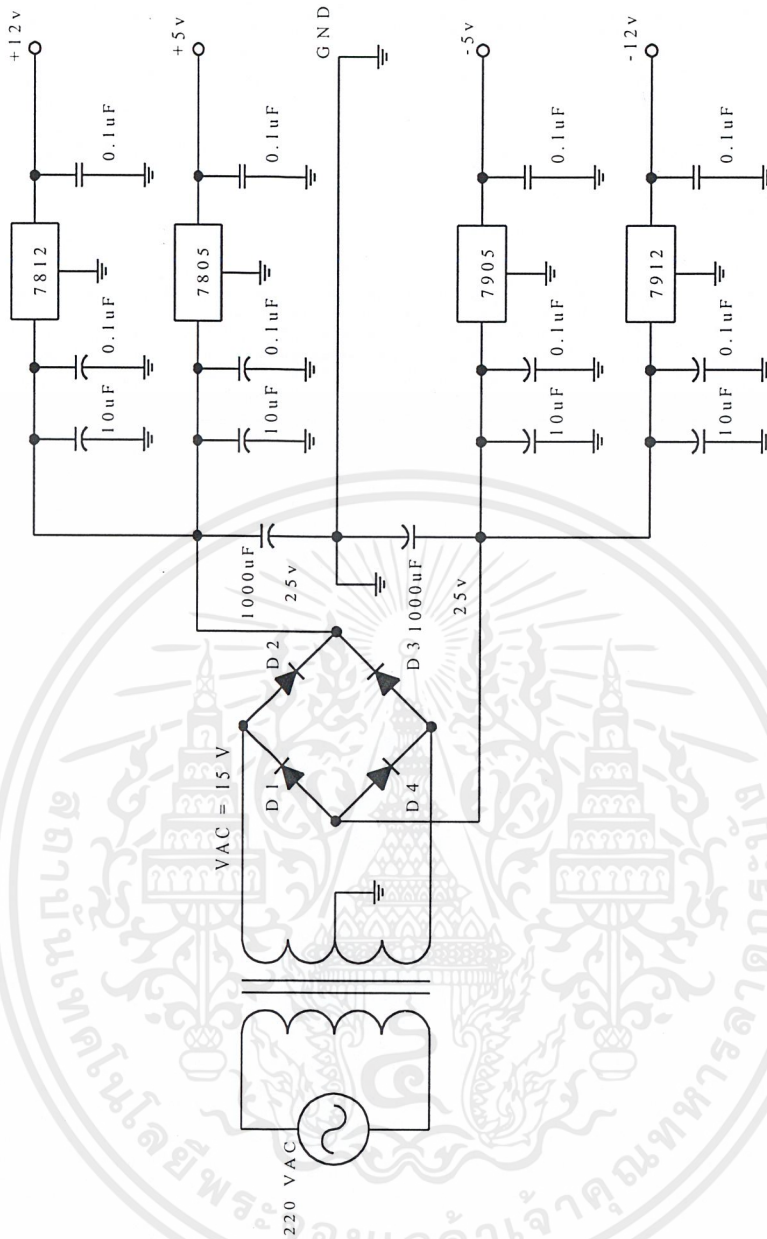
รูปที่ ข.1 วงจรขยายและลดทอนสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



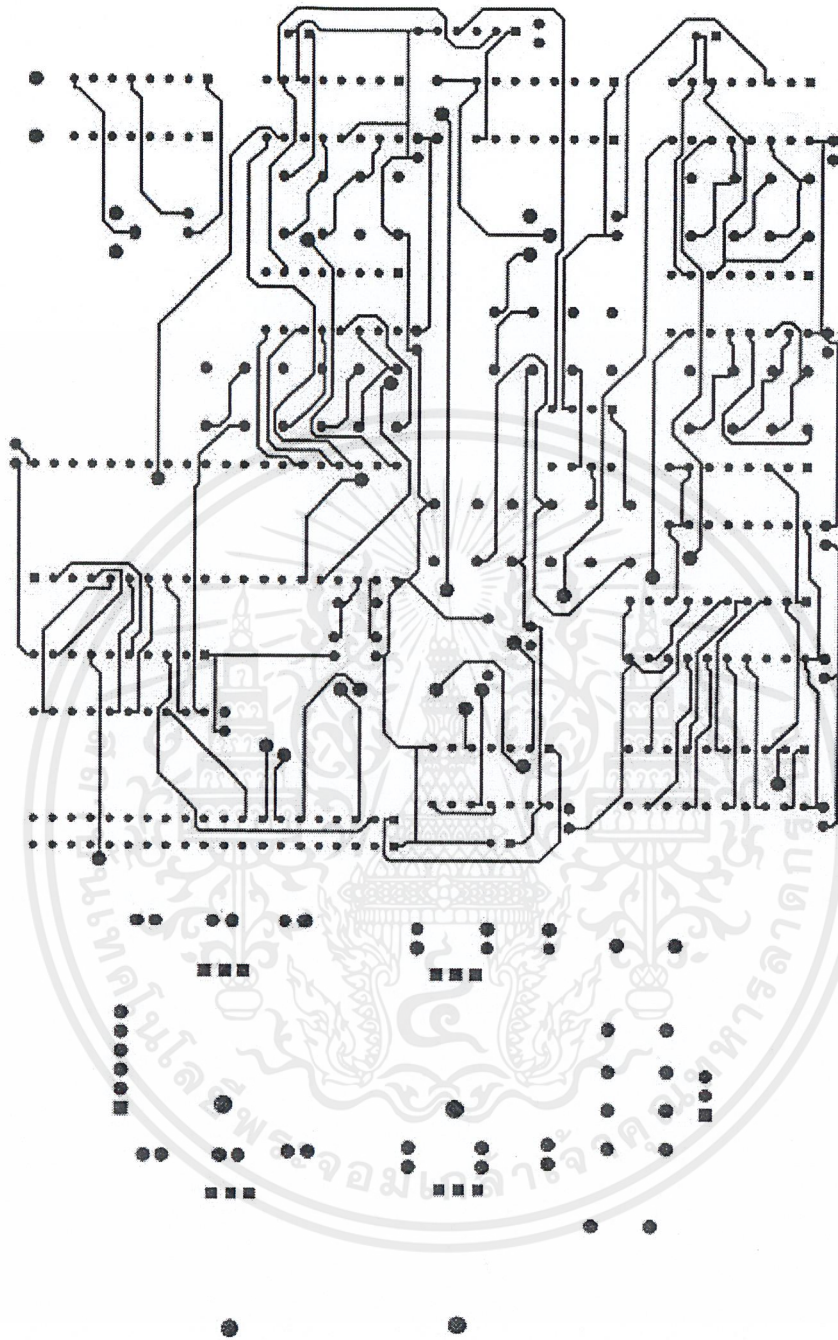
รูปที่ ข.2 วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



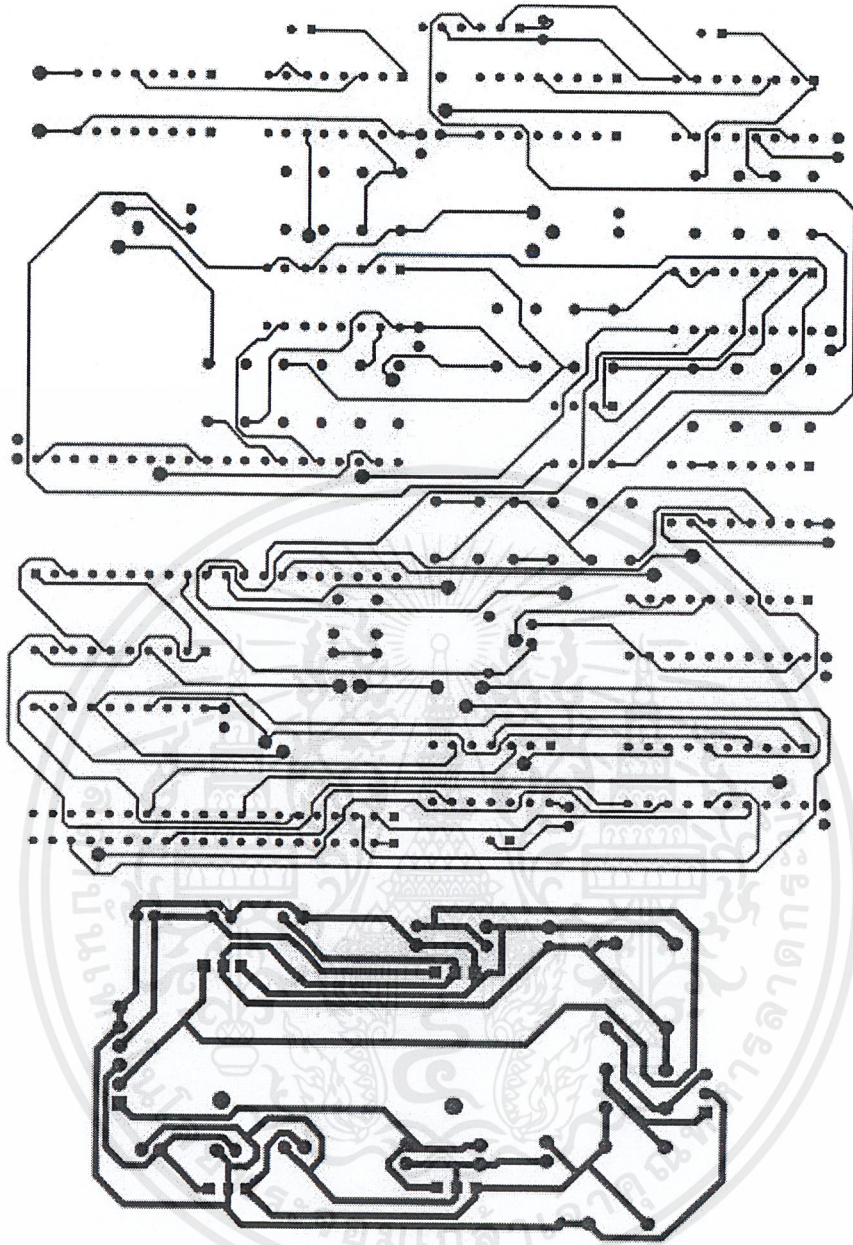
รูปที่ ข.3 วงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 แผ่นวงจรพิมพ์ด้าน Top Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

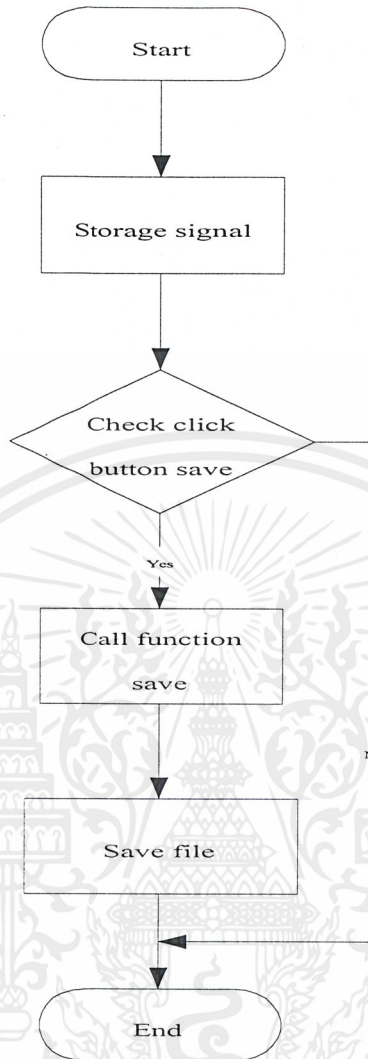


รูปที่ ข.5 แผ่นวงจรพิมพ์ด้าน Bottom Layer

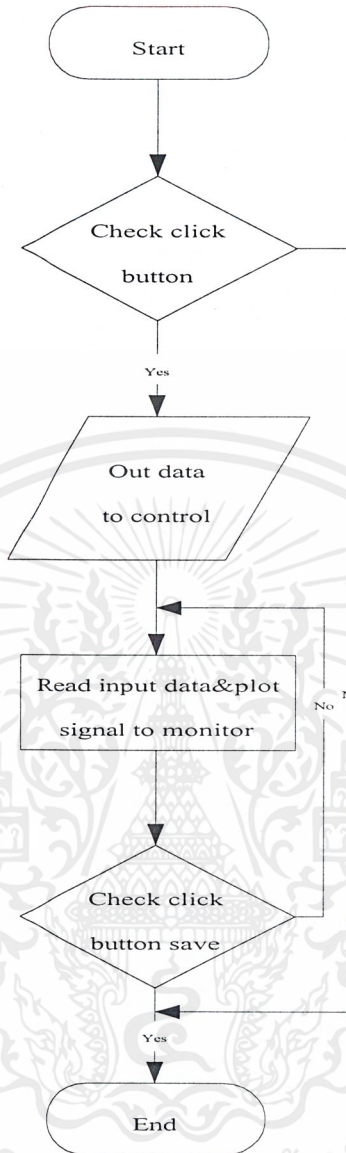
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



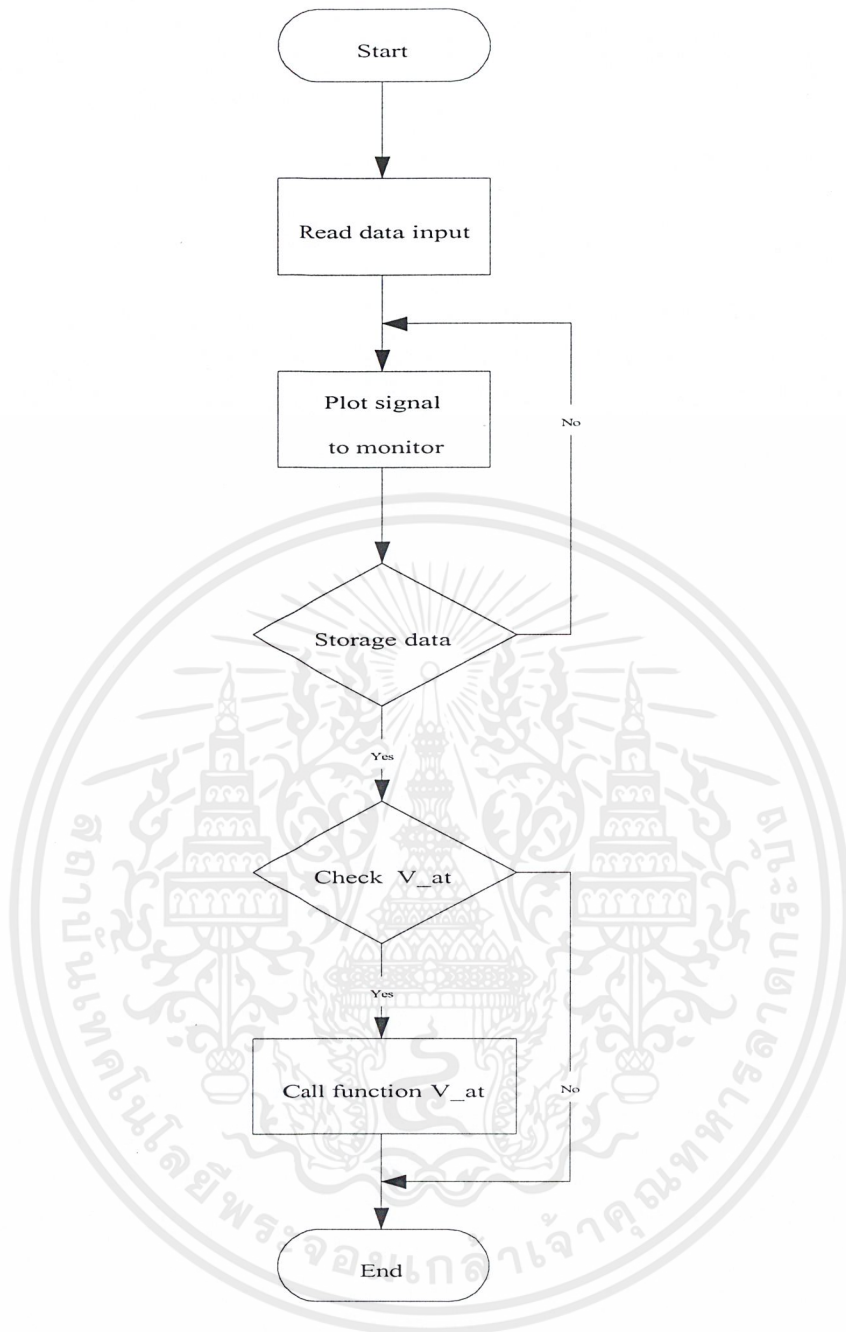
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



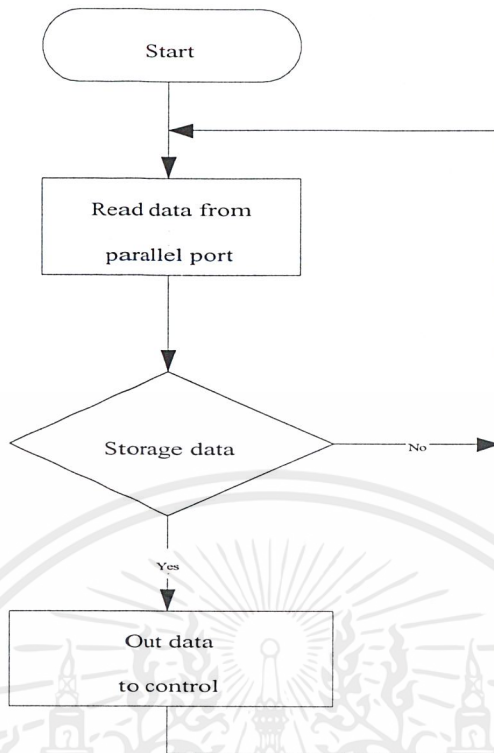
รูปที่ ค.1 ผังงานของโปรแกรมส่วนควบคุมการสั่งบันทึกข้อมูล



รูปที่ ค.2 ฟังงานของโปรแกรมส่วนควบคุมการวัดค่า



รูปที่ ๓.3 ฟังก์ชันของโปรแกรมส่วนควบคุมการแสดงผล



รูปที่ ค.4 ฟังงานของโปรแกรมส่วนควบคุมการทำงานของวงจร Attenuater

```

#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<bios.h>
#include<dos.h>

int signal,ber,ber2;
int xis,yis,buttom,out=0,ex=0,ox,ob,oy;
int ch1=0,ch2=0,add_ch=0,ac1=0,ac2=0,
dc1=0,dc2=0,gnd1=0,gnd2=0,ok=0;
int start=0,stop=0,v_at1=0,v_at2=0,save=0,
ope=0,ac_1=0,dc_1=0,ac_2=0,dc_2=0;
int in_ac_ch1_1[200],in_ac_ch1_2[200];
int port_ac_ch1=0,i,t,mmm=100,freqavg,first=1,
second=1,third=1,fourth=1;
int c=0,count=1,xpos=6,ypos=115;
float num,nn=1.0,vh,vl,vd=1;
float freq;
char buff[20];
union REGS regs;
FILE *fp;
Struct data{
    Char file_name[10];
    Int data_signal[9000];
}list;
  
```

```

/*-----MAIN PROGRAM-----*/
main()
{
    int i,j,D=0,M=0;
    initgraph(&D,&M,"C:\\tc\\bgi");
    BLACK_GROUND(); //<-- CALL FUNC BLACK GROUND
    MOUSE();
    START_SYSTEM();
    closegraph();
}

/*-----BLACK GROUND-----*/

BLACK_GROUND()
{
    setcolor(WHITE);
    outtextxy(5,5,"DIGITAL STORAGE
OSCILLOSCOPE");
    line(0,15,635,15);
    setfillstyle(1,BLUE);
    floodfill(1,1,WHITE);
    rectangle(0,0,635,475);
    setfillstyle(1,LIGHTGRAY);
    floodfill(1,17,WHITE);
    MONITOR(); //<-- CALL FUNCTION MONITOR
    CONTROL();
    DISPLAY();
    return(0);
}

/*-----FUNCTION MONITOR-----*/

MONITOR()
{
    int i,j,sx,sy,ex,ey;
    setcolor(WHITE);
    rectangle(5,20,355,300);
    setfillstyle(1,LIGHTBLUE);
    floodfill(6,21,WHITE);
    setcolor(LIGHTGRAY);

/*-----LINE VERTICAL-----*/

    sx=40;sy=21;ex=40;ey=299;
    for(i=1;i<=9;i++)
    {
        line(sx,sy,ex,ey);
        sx=sx+35;
        ex=sx;
    }

/*-----LINE HORIZONTAL-----*/

    sx=6;sy=55;ex=355;ey=55;
    for(i=1;i<=7;i++)

```

```

    {
        line(sx,sy,ex,ey);
        sy=sy+35;
        ey=sy;
    }
    setcolor(WHITE);
    line(180,20,180,300);
    line(6,160,355,160);
    setcolor(LIGHTGRAY);
    sx=12;sy=156;ex=12;ey=164;
    for(i=1;i<=10;i++)
    {
        for(j=1;j<=4;j++)
        {
            line(sx,sy,ex,ey);
            sx=sx+7;
            ex=sx;
        }
        sx=sx+7;
        ex=sx;
    }
    sx=176;sy=27;ex=184;ey=27;
    for(i=1;i<=8;i++)
    {
        for(j=1;j<=4;j++)
        {
            line(sx,sy,ex,ey);
            sy=sy+7;
            ey=sy;
        }
        sy=sy+7;
        ey=sy;
    }
}

/*-----FUNCTION CONTROL-----*/

CONTROL()
{
    int i,j,sx,sy;

/*-----VOLT AND TIME FOR CHANNEL 1-----*/
    setcolor(WHITE);
    rectangle(5,350,180,470);
    setfillstyle(1,CYAN);
    floodfill(6,351,WHITE);
    outtextxy(10,360,"Volt and Time for CH1");
    VOLT_CH1();

/*-----VOLT AND TIME FOR CHANNEL 2-----*/
    setcolor(WHITE);
    rectangle(215,350,390,470);
    setfillstyle(1,CYAN);
    floodfill(216,351,WHITE);
    outtextxy(220,360,"Volt and Time for CH2");
}

```

```
VOLT_CH2 ();
```

```
/*-----EDITION-----*/
```

```
setcolor(WHITE);  
rectangle(425,350,620,470);  
setfillstyle(1,LIGHTRED);  
floodfill(426,351,WHITE);  
setcolor(BLACK);  
outtextxy(495,360,"EDITION");  
setcolor(YELLOW);  
rectangle(510,381,590,401);  
setfillstyle(1,BLACK);  
floodfill(511,382,YELLOW);  
rectangle(470,440,520,460);  
setfillstyle(1,BLUE);  
floodfill(471,441,YELLOW);  
rectangle(540,440,590,460);  
setfillstyle(1,BLUE);  
floodfill(541,441,YELLOW);  
setcolor(BLACK);  
outtextxy(430,390,"File Name");  
outtextxy(487,447,"OK");  
outtextxy(560,447,"NO");
```

```
/*-----CONTROL BUTTOM-----*/
```

```
setcolor(YELLOW);  
rectangle(365,30,415,50);  
/*-- Buttom Switch Start --*/  
setfillstyle(1,BLUE);  
floodfill(366,31,YELLOW);  
rectangle(365,70,415,90);  
/*-- Buttom Exit Program --*/  
setfillstyle(1,BLUE);  
floodfill(366,71,YELLOW);  
rectangle(365,110,415,130);  
/*-- Buttom Storage Signal --*/  
setfillstyle(1,BLUE);  
floodfill(366,111,YELLOW);  
rectangle(365,230,415,250);  
/*-- Buttom Open File (Signal) --*/  
setfillstyle(1,BLUE);  
floodfill(366,231,YELLOW);  
rectangle(365,270,415,290);  
/*-- Buttom Save Signal --*/  
setfillstyle(1,BLUE);  
floodfill(366,271,YELLOW);  
rectangle(50,315,150,335);  
/*-- Buttom Channel 1 --*/  
setfillstyle(1,BLUE);  
floodfill(51,316,YELLOW);  
rectangle(260,315,360,335);  
/*-- Buttom Channel 2 --*/  
setfillstyle(1,BLUE);  
floodfill(261,316,YELLOW);
```

```

rectangle(380,315,480,335);
setfillstyle(1,BLUE);
floodfill(381,316,YELLOW);
rectangle(500,315,600,335);
setfillstyle(1,BLUE);
floodfill(501,316,YELLOW);
rectangle(60,440,130,460);
/*-- Buttom Ground CH 1 --*/
setfillstyle(1,BLUE);
floodfill(61,441,YELLOW);
rectangle(20,440,40,460);
/*-- Buttom AC for CH1 --*/
setfillstyle(1,BLUE);
floodfill(21,441,YELLOW);
rectangle(150,440,170,460);
/*-- Buttom DC for CH1 --*/
setfillstyle(1,BLUE);
floodfill(151,441,YELLOW);
rectangle(270,440,340,460);
/*-- Buttom Ground CH 2 --*/
setfillstyle(1,BLUE);
floodfill(271,441,YELLOW);
rectangle(230,440,250,460);
/*-- Buttom AC for CH2 --*/
setfillstyle(1,BLUE);
floodfill(231,441,YELLOW);
rectangle(360,440,380,460);
/*-- Buttom DC for CH2 --*/
setfillstyle(1,BLUE);
floodfill(361,441,YELLOW);
rectangle(170,315,240,335);
/*-- Buttom Add Channel --*/
setfillstyle(1,BLUE);
floodfill(171,316,YELLOW);
setcolor(WHITE);
outtextxy(372,37,"START");
outtextxy(376,77,"EXIT");
outtextxy(376,117,"STOP");
outtextxy(400,323,"TIME UP");
outtextxy(515,323,"TIME DOWN");
outtextxy(376,237,"OPEN");
outtextxy(376,277,"SAVE");
outtextxy(65,322,"CHANNEL 1");
outtextxy(275,322,"CHANNEL 2");
outtextxy(73,447,"GROUND");
outtextxy(283,447,"GROUND");
outtextxy(182,322,"ADD CH");
outtextxy(23,447,"AC");
outtextxy(153,447,"DC");
outtextxy(233,447,"AC");
outtextxy(363,447,"DC");
}

```

```

/*-----FUNCTION DISPLAY CH1-CH2-----*/

```

```
DISPLAY()
{
/*-----DISPLAY CHANNEL 1-----*/
    setcolor(WHITE);
    rectangle(425,20,620,150);
    setfillstyle(1,GREEN);
    floodfill(426,21,WHITE);
    rectangle(490,43,565,65);
    setfillstyle(1,BLACK);
    floodfill(491,44,WHITE);
    rectangle(490,75,565,97);
    setfillstyle(1,BLACK);
    floodfill(491,76,WHITE);
    outtextxy(460,27,"DISPLAY CHANNEL 1");
    setcolor(YELLOW);
    rectangle(440,125,460,145);
    setfillstyle(1,BLUE);
    floodfill(441,126,YELLOW);
    rectangle(480,125,500,145);
    setfillstyle(1,BLUE);
    floodfill(481,126,YELLOW);
    rectangle(540,125,560,145);
    setfillstyle(1,BLUE);
    floodfill(541,126,YELLOW);
    rectangle(580,125,600,145);
    setfillstyle(1,BLUE);
    floodfill(581,126,YELLOW);
    setcolor(WHITE);
    outtextxy(447,132,"+");
    outtextxy(487,132,"-");
    outtextxy(547,132,"+");
    outtextxy(587,132,"-");
    setcolor(RED);
    outtextxy(440,50,"Volt V");
    outtextxy(440,82,"Freq Hz");
    outtextxy(450,110,"Y-POS");
    outtextxy(550,110,"X-POS");
/*-----DISPLAY CHANNEL 2-----*/
    setcolor(WHITE);
    rectangle(425,170,620,300);
    setfillstyle(1,GREEN);
    floodfill(426,171,WHITE);
    rectangle(490,202,565,225);
    setfillstyle(1,BLACK);
    floodfill(491,206,WHITE);
    rectangle(490,235,565,257);
    setfillstyle(1,BLACK);
    floodfill(491,236,WHITE);
    outtextxy(460,177,"DISPLAY CHANNEL 2");
    setcolor(YELLOW);
    rectangle(440,275,460,295);
    setfillstyle(1,BLUE);
    floodfill(441,276,YELLOW);
}
```

```

rectangle(480,275,500,295);
setfillstyle(1,BLUE);
floodfill(481,276,YELLOW);
rectangle(540,275,560,295);
setfillstyle(1,BLUE);
floodfill(541,276,YELLOW);
rectangle(580,275,600,295);
setfillstyle(1,BLUE);
floodfill(581,276,YELLOW);
setcolor(WHITE);
outtextxy(447,282,"+");
outtextxy(487,282,"-");
outtextxy(547,282,"+");
outtextxy(587,282,"-");
setcolor(RED);
outtextxy(440,209,"Volt          V");
outtextxy(440,242,"Freq          Hz");
outtextxy(450,265,"Y-POS");
outtextxy(550,265,"X-POS");
}

/*-----START SYSTEM-----*/

START_SYSTEM()
{
do
{
    Checkmouse();
    Mouse_show();
    CHECK_CH1();
    CHECK_CH2();
    if((xis>=52&&xis<=147)&&(yis>=318&&yis
<=333)&&button==1) /*-- CHANNEL 1--*/
    {
        Mouse_off();
        ch1=1;
        ch2=0;
        add_ch=0;
        CHANNEL_1();
    }
    else if((xis>=262&&xis<=356)&&
(yis>=318&&yis<=333)&&button==1)
/*-- CHANNEL 2 --*/
    {
        Mouse_off();
        ch1=0;
        ch2=1;
        add_ch=0;
        CHANNEL_2();
    }
    else if((xis>=172&&xis<=237)&&
(yis>=318&&yis<=333)&&button==1)
/*-- ADD CHANNEL --*/
    {
        Mouse_off();

```

```

        ch1=0;
        ch2=0;
        add_ch=1;
        ADD_CH();
    }

    else if((xis>=368&&xis<=413)&&
(yis>=233&&xis<=247)&&button==1)
    {
        Mouse_off();
        OPEN_FILE();
    }
    else if((xis>=368&&xis<=412)&&
(yis>=73&&xis<=87)&&button==1)
/*-- EXIT PROGRAM --*/
    {
        ex=1;
    }
    else delay(0);
}while(ex!=1);
}

/*-----CHANNEL 1-----*/

CHANNEL_1()
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x00);
    setcolor(YELLOW);
    rectangle(50,315,150,335);
/*-- Button Channel 1 --*/
    setfillstyle(1,GREEN);
    floodfill(51,316,YELLOW);
    rectangle(260,315,360,335);
/*-- Button Channel 2 --*/
    setfillstyle(1,BLUE);
    floodfill(261,316,YELLOW);
    rectangle(170,315,240,335);
/*-- Button Add Channel --*/
    setfillstyle(1,BLUE);
    floodfill(171,316,YELLOW);
    setcolor(WHITE);
    outtextxy(65,322,"CHANNEL 1");
    outtextxy(275,322,"CHANNEL 2");
    outtextxy(182,322,"ADD CH");

INIT_2();
Do
{
    Mouse_show();
    Checkmouse();
    CHECK_CH1();
    START();

    if((xis>=22&&xis<=37)&&(xis>=442&&xis

```

```

<=457) &&button==1)
    {
        Mouse_off();
        AC_1();
        START();
    }
    else if((xis>=62&&xis<=127) &&(yis>=442&&yis
<=457) &&button==1)
    {
        Mouse_off();
        GND_1();
        START();
    }
    else if((xis>=151&&xis<=167) &&
(yis>=442&&yis<=457) &&button==1)
    {
        Mouse_off();
        DC_1();
        START();
    }
    else if((xis>=368&&xis<=412) &&(yis>=32&&yis
<=47) &&button==1)
    {
        Mouse_off();
        start=1;
        START();
    }
    else if((xis>=262&&xis<=356) &&
(yis>=318&&yis<=333) &&button==1)
    {
        ch1=0; // CH2
        ch2=1;
        add_ch=0;
        START();
    }
    else if((xis>=172&&xis<=237) &&
(yis>=318&&yis<=333) &&button==1)
    {
        ch1=0; // ADD CH
        ch2=0;
        add_ch=1;
        START();
    }
    else if((xis>=368&&xis<=412) &&(yis>=73&&yis
<=87) &&button==1)
    {
        ch1=0;
        ex=1;
    }
    else if((xis>=368&&xis<=413) &&
(yis>=233&&yis<=247) &&button==1)
    {
        Mouse_off();
        OPEN_FILE();
    }
}

```

```

        else delay(0);
    }while(ch1!=0);
    CLEAR_START_BUTTON();
}

CLEAR_START_BUTTON()
{
    setcolor(YELLOW);
    rectangle(365,30,415,50);
    /*-- Button Switch Start --*/
    setfillstyle(1,BLUE);
    floodfill(366,31,YELLOW);
    setcolor(WHITE);
    outtextxy(372,37,"START");
    start=0;
}

START();
{
    if(start==1&&ch1==1&&ac_1==1)
    {
        Mouse_off();
        MONITOR();
        PLOT1();
        /*-- PLOT GRAPH --*/
        setcolor(YELLOW);
        rectangle(365,30,415,50);
        /*-- Button Switch Start --*/
        setfillstyle(1,GREEN);
        floodfill(366,31,YELLOW);
        setcolor(WHITE);
        outtextxy(372,37,"START");
        do
        {
            Mouse_show();
            Checkmouse();
            CHECK_CH1();
            CHECK_CH2();
            delay(200);
            if((xis>=368&&xis<=413)&&
            (yis>=113&&yis<=127)&&button==1)
            {
                Mouse_off();
                if(stop==0)
                {
                    stop=1;
                    setcolor(YELLOW);
                    rectangle(365,110,415,130);
                    setfillstyle(1,GREEN);
                    floodfill(366,111,YELLOW);
                    setcolor(WHITE);
                    outtextxy(376,117,"STOP");
                    STOP();
                }
            }
        }
    }
}

```

```

else if(stop==1)
{
    stop=0;
    setcolor(YELLOW);
    rectangle(365,110,415,130);
    setfillstyle(1,BLUE);
    floodfill(366,111,YELLOW);
    setcolor(WHITE);
    outtextxy(376,117,"STOP");
}
else delay(0);
}
else delay(0);
}while(stop!=0);
}
else if(start==1&&ch2==1&&ac_2==1)
{
    Mouse_off();
    MONITOR();
    PLOT2();
    /*-- PLOT GRAPH --*/
    Setcolor(YELLOW);
    Rectangle(365,30,415,50);
    Setfillstyle(1,GREEN);
    Floodfill(366,31,YELLOW);
    Setcolor(WHITE);
    outtextxy(372,37,"START");
    do
    {
        Mouse_show();
        Checkmouse();
        CHECK_CH1();
        CHECK_CH2();
        delay(200);
        if((xis>=368&&xis<=413)&&
(yis>=113&&yis<=127)&&button==1)
        {
            Mouse_off();
            if(stop==0)
            {
                stop=1;
                setcolor(YELLOW);
                rectangle(365,110,415,130);
                setfillstyle(1,GREEN);
                floodfill(366,111,YELLOW);
                setcolor(WHITE);
                outtextxy(376,117,"STOP");
                STOP();
            }
            else if(stop==1)
            {
                stop=0;
                setcolor(YELLOW);
                rectangle(365,110,415,130);
                setfillstyle(1,BLUE);

```

```

        floodfill(366,111,YELLOW);
        setcolor(WHITE);
        outtextxy(376,117,"STOP");
    }
    else delay(0);
}
else delay(0);

}while(stop!=0);
}
else if(start==1&&ch1==1&&dc_1==1)
{
    Mouse_off();
    MONITOR();
    PLOT_DC1();
    Setcolor(YELLOW);
    Rectangle(365,30,415,50);
    Setfillstyle(1,GREEN);
    floodfill(366,31,YELLOW);
    setcolor(WHITE);
    outtextxy(372,37,"START");
    do
    {
        Mouse_show();
        Checkmouse();
        delay(200);
        if((xis>=368&&xis<=413)&&
(yis>=113&&ysis<=127)&&button==1)
        {
            Mouse_off();
            if(stop==0)
            {
                stop=1;
                setcolor(YELLOW);
                rectangle(365,110,415,130);
                setfillstyle(1,GREEN);
                floodfill(366,111,YELLOW);
                setcolor(WHITE);
                outtextxy(376,117,"STOP");
                STOP();
            }
        }
    }
    else if(stop==1)
    {
        stop=0;
        setcolor(YELLOW);
        rectangle(365,110,415,130);
        setfillstyle(1,BLUE);
        floodfill(366,111,YELLOW);
        setcolor(WHITE);
        outtextxy(376,117,"STOP");
    }
}
else delay(0);
}
else delay(0);
}while(stop!=0);

```

```

    }
    else if (start==1&&ch2==1&&dc_2==1)
    {
        Mouse_off();
        MONITOR();
        PLOT_DC1();
        setcolor(YELLOW);
        rectangle(365,30,415,50);
        setfillstyle(1, GREEN);
        floodfill(366,31, YELLOW);
        setcolor(WHITE);
        outtextxy(372,37, "START");
        do
        {
            Mouse_show();
            Checkmouse();
            delay(200);
            if ((xis>=368&&xis<=413) &&
                (yis>=113&&yis<=127) &&buttom==1)
            {
                Mouse_off();
                if (stop==0)
                {
                    stop=1;
                    setcolor(YELLOW);
                    rectangle(365,110,415,130);
                    setfillstyle(1, GREEN);
                    floodfill(366,111, YELLOW);
                    setcolor(WHITE);
                    outtextxy(376,117, "STOP");
                    STOP();
                }
                else if (stop==1)
                {
                    stop=0;
                    setcolor(YELLOW);
                    rectangle(365,110,415,130);
                    setfillstyle(1, BLUE);
                    floodfill(366,111, YELLOW);
                    setcolor(WHITE);
                    outtextxy(376,117, "STOP");
                }
                else delay(0);
            }
            else delay(0);
        }while (stop!=0);
    }
    else delay(0);
}

CHANNEL_2()
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x00);
}

```

```

setcolor(YELLOW);
rectangle(50,315,150,335);
setfillstyle(1,BLUE);
floodfill(51,316,YELLOW);
rectangle(260,315,360,335);
setfillstyle(1,GREEN);
floodfill(261,316,YELLOW);
rectangle(170,315,240,335);
setfillstyle(1,BLUE);
floodfill(171,316,YELLOW);
setcolor(WHITE);
outtextxy(65,322,"CHANNEL 1");
outtextxy(275,322,"CHANNEL 2");
outtextxy(182,322,"ADD CH");
INIT_1();

do
{
    Mouse_show();
    Checkmouse();
    START();
    CHECK_CH2();
    if((xis>=231&&xis<=247)&&(yis>=442&&yis
<=457)&&button==1)
    {
        Mouse_off();
        AC_2();
        START();
    }
    else if((xis>=272&&xis<=338)&&
(yis>=442&&yis<=457)&&button==1)
    {
        Mouse_off();
        GND_2();
        START();
    }
    else if((xis>=362&&xis<=377)&&
(yis>=442&&yis<=457)&&button==1)
    {
        Mouse_off();
        DC_2();
        START();
    }
    else if((xis>=368&&xis<=412)&&
(yis>=32&&yis<=47)&&button==1)
    {
        Mouse_off();
        start=1;
        START();
    }
    else if((xis>=52&&xis<=147)&&(yis>=318&&yis
<=333)&&button==1)
    {
        ch1=1;
        ch2=0;
    }
}

```

```

        add_ch=0;
        START();
    }

    else if((xis>=172&&xis<=237)&&
(yis>=318&&ysis<=333)&&button==1)
    {
        ch1=0;
        ch2=0;
        add_ch=1;
        START();
    }
    else if((xis>=368&&xis<=412)&&(ysis>=73&&ysis
<=87)&&button==1)
    {
        ch2=0;
        ex=1;
    }
    else delay(0);
}while(ch2!=0);
CLEAR_START_BUTTON();
}

ADD_CH()
{
    setcolor(YELLOW);
    rectangle(50,315,150,335);
    setfillstyle(1,BLUE);
    floodfill(51,316,YELLOW);
    rectangle(260,315,360,335);
    setfillstyle(1,BLUE);
    floodfill(261,316,YELLOW);
    rectangle(170,315,240,335);
    setfillstyle(1,GREEN);
    floodfill(171,316,YELLOW);
    setcolor(WHITE);
    outtextxy(65,322,"CHANNEL 1");
    outtextxy(275,322,"CHANNEL 2");
    outtextxy(182,322,"ADD CH");
    INIT_1();
}

INIT_2();
do
{
    Mouse_show();
    Checkmouse();
    if((xis>=231&&xis<=247)&&(ysis>=442&&ysis
<=457)&&button==1)
    {
        Mouse_off();
        AC_2();
    }
    else if((xis>=272&&xis<=338)&&
(ysis>=442&&ysis<=457)&&button==1)
    {

```

```

        Mouse_off();
        GND_2();
    }
    else if((xis>=362&&xis<=377)&&
(yis>=442&&ysis<=457)&&button==1)
    {
        Mouse_off();
        DC_2();
    }
    else if((xis>=52&&xis<=147)&&
(yis>=318&&ysis<=333)&&button==1)
    {
        ch1=1;
        ch2=0;
        add_ch=0;
    }
    else if((xis>=172&&xis<=237)&&
(yis>=318&&ysis<=333)&&button==1)
    {
        ch1=0;
        ch2=0;
        add_ch=1;
    }
    else if((xis>=368&&xis<=412)&&
(yis>=73&&ysis<=87)&&button==1)
    {
        add_ch=0;
        ex=1;
    }
    else delay(0);

    if((xis>=22&&xis<=37)&&(ysis>=442&&ysis
<=457)&&button==1)
    {
        Mouse_off();
        AC_1();
    }
    else if((xis>=62&&xis<=127)&&(ysis>=442&&ysis
<=457)&&button==1)
    {
        Mouse_off();
        GND_1();
    }
    else if((xis>=151&&xis<=167)&&
(yis>=442&&ysis<=457)&&button==1)
    {
        Mouse_off();
        DC_1();
    }
    else if((xis>=262&&xis<=356)&&
(yis>=318&&ysis<=333)&&button==1)
    {
        ch1=0;
        ch2=1;
        add_ch=0;
    }

```

```

    }
    else if ((xis>=172&&xis<=237) &&
(yis>=318&&yis<=333) &&button==1)
    {
        ch1=0;
        ch2=0;
        add_ch=1;
    }
    else if ((xis>=368&&xis<=412) &&
(yis>=73&&yis<=87) &&button==1)
    {
        add_ch=0;
        ex=1;
    }
    else delay(0);
}while(add_ch!=0);
INIT_1();
INIT_2();
CLEAR_START_BUTTON();
}
AC_1()
{
    ac_1=1;
    dc_1=0;
    ac_2=0;
    dc_2=0;
    outportb(0x37a, 0x0a);
    outportb(0x378, 0x00);
    setcolor(YELLOW);
    rectangle(20, 440, 40, 460);
    setfillstyle(1, GREEN);
    floodfill(21, 441, YELLOW);
    rectangle(150, 440, 170, 460);
    setfillstyle(1, BLUE);
    floodfill(151, 441, YELLOW);
    rectangle(60, 440, 130, 460);
    setfillstyle(1, BLUE);
    floodfill(61, 441, YELLOW);
    setcolor(WHITE);
    outtextxy(23, 447, "AC");
    outtextxy(153, 447, "DC");
    outtextxy(73, 447, "GROUND");
    STOP();
}

AC_2()
{
    ac_2=1;
    dc_2=0;
    ac_1=0;
    dc_1=0;
    outportb(0x37a, 0x0a);
    outportb(0x378, 0x00);
    setcolor(YELLOW);
    rectangle(230, 440, 250, 460);

```

```

setfillstyle(1, GREEN);
floodfill(231, 441, YELLOW);
rectangle(360, 440, 380, 460);
setfillstyle(1, BLUE);
floodfill(361, 441, YELLOW);
rectangle(270, 440, 340, 460);
setfillstyle(1, BLUE);
floodfill(271, 441, YELLOW);
setcolor(WHITE);
outtextxy(233, 447, "AC");
outtextxy(363, 447, "DC");
outtextxy(283, 447, "GROUND");
STOP();
}

```

```

GND_1()
{
setcolor(YELLOW);
rectangle(20, 440, 40, 460);
setfillstyle(1, BLUE);
floodfill(21, 441, YELLOW);
rectangle(150, 440, 170, 460);
setfillstyle(1, BLUE);
floodfill(151, 441, YELLOW);
rectangle(60, 440, 130, 460);
setfillstyle(1, GREEN);
floodfill(61, 441, YELLOW);
setcolor(WHITE);
outtextxy(23, 447, "AC");
outtextxy(153, 447, "DC");
outtextxy(73, 447, "GROUND");
}

```

```

GND_2()
{
setcolor(YELLOW);
rectangle(230, 440, 250, 460);
setfillstyle(1, BLUE);
floodfill(231, 441, YELLOW);
rectangle(360, 440, 380, 460);
setfillstyle(1, BLUE);
floodfill(361, 441, YELLOW);
rectangle(270, 440, 340, 460);
setfillstyle(1, GREEN);
floodfill(271, 441, YELLOW);
setcolor(WHITE);
outtextxy(233, 447, "AC");
outtextxy(363, 447, "DC");
outtextxy(283, 447, "GROUND");
}

```

```

DC_1()
{
ac_1=0;
dc_1=1;
}

```

```

ac_2=0;
dc_2=0;
setcolor(YELLOW);
rectangle(20,440,40,460);
setfillstyle(1,BLUE);
floodfill(21,441,YELLOW);
rectangle(150,440,170,460);
setfillstyle(1,GREEN);
floodfill(151,441,YELLOW);
rectangle(60,440,130,460);
setfillstyle(1,BLUE);
floodfill(61,441,YELLOW);
setcolor(WHITE);
outtextxy(23,447,"AC");
outtextxy(153,447,"DC");
outtextxy(73,447,"GROUND");
STOP();
}

DC_2()
{
ac_2=0;
dc_2=1;
ac_1=0;
dc_1=0;
setcolor(YELLOW);
rectangle(230,440,250,460);
setfillstyle(1,BLUE);
floodfill(231,441,YELLOW);
rectangle(360,440,380,460);
setfillstyle(1,GREEN);
floodfill(361,441,YELLOW);
rectangle(270,440,340,460);
setfillstyle(1,BLUE);
floodfill(271,441,YELLOW);
setcolor(WHITE);
outtextxy(233,447,"AC");
outtextxy(363,447,"DC");
outtextxy(283,447,"GROUND");
STOP();
}

/*-----CHANNEL 2-----*/
/*----INITIAL AC DC AND GND OF CHANNEL 1----*/

INIT_1()
{
setcolor(YELLOW);
rectangle(20,440,40,460);
setfillstyle(1,BLUE);
floodfill(21,441,YELLOW);
rectangle(150,440,170,460);
setfillstyle(1,BLUE);
floodfill(151,441,YELLOW);
}

```

```

rectangle(60,440,130,460);
setfillstyle(1,BLUE);
floodfill(61,441,YELLOW);
setcolor(WHITE);
outtextxy(23,447,"AC");
outtextxy(153,447,"DC");
outtextxy(73,447,"GROUND");
}

INIT_2()
{
setcolor(YELLOW);
rectangle(270,440,340,460);
setfillstyle(1,BLUE);
floodfill(271,441,YELLOW);
rectangle(230,440,250,460);
setfillstyle(1,BLUE);
floodfill(231,441,YELLOW);
rectangle(360,440,380,460);
setfillstyle(1,BLUE);
floodfill(361,441,YELLOW);
setcolor(WHITE);
outtextxy(283,447,"GROUND");
outtextxy(233,447,"AC");
outtextxy(363,447,"DC");
}

/*-----FUNCTION OPEN FILE-----*/
OPEN_FILE()
{
int i;
char fname[20];
MONITOR();
ope=1;
setcolor(YELLOW);
rectangle(365,230,415,250);
setfillstyle(1,GREEN);
floodfill(366,231,YELLOW);
setcolor(WHITE);
outtextxy(376,237,"OPEN");
setcolor(WHITE);
outtextxy(495,360,"EDITION");
outtextxy(430,390,"File Name");
outtextxy(487,447,"OK");
outtextxy(560,447,"NO");
Mouse_show();
start1:
gotoxy(66,25);scanf("%s",fname);
do
{
Mouse_show();
Checkmouse();
delay(200);
if((xis>=470&&xis<=520)&&(yis>=440&&yis
<=460)&&button==1)

```

```

        {
            Mouse_off();
        }
    if ((fp=fopen("%s",fname), "rb")) == NULL
    {
        exit(0);
    }
    for(;;)
    {
        if(fread(&list,sizeof(struct
data),1,fp)!=1);
        {
            if(feof(fp))
            {
                fclose(fp);
                break;
            }
        }
        if(!strcmp(fname,list.file_name))
        {
            moveto(6,160);
            for(i=1;i<=100;i++)
            {
                setcolor(YELLOW);
                if (list.data_signal[i]<180)
                [i]);
                lineto(5+i*3.5,(250)-list.data_signal
                }
            }
            setcolor(YELLOW);
            rectangle(510,381,590,401);
            setfillstyle(1,BLACK);
            floodfill(512,382,YELLOW);
            ope=0;
            setcolor(YELLOW);
            rectangle(365,230,415,250);
            setfillstyle(1,BLUE);
            floodfill(366,231,YELLOW);
            setcolor(WHITE);
            outtextxy(376,237,"OPEN");
            setcolor(BLACK);
            outtextxy(495,360,"EDITION");
            outtextxy(430,390,"File Name");
            outtextxy(487,447,"OK");
            outtextxy(560,447,"NO");

        }
        else if((xis>=540&&xis<=590)&&
(yis>=440&&ysis<=460)&&button==1)
        {
            Mouse_off();
            setcolor(YELLOW);
            rectangle(510,381,590,401);
            setfillstyle(1,BLACK);

```

```

        floodfill(512,382,YELLOW);
        i=0;x=520;
        buff[0];
        goto start1;
    }
    else if((xis>=365&&xis<=415)&&
(yis>=230&&ysis<=250)&&button==1)
    {
        Mouse_off();
        ope=0;
        setcolor(YELLOW);
        rectangle(365,230,415,250);
        setfillstyle(1,BLUE);
        floodfill(366,231,YELLOW);
        setcolor(WHITE);
        outtextxy(376,237,"OPEN");
        setcolor(BLACK);
        outtextxy(495,360,"EDITION");
        outtextxy(430,390,"File Name");
        outtextxy(487,447,"OK");
        outtextxy(560,447,"NO");
    }
    else delay(0);
}while(ope!=0);
}
STOP()
{
do
{
    Mouse_show();
    Checkmouse();
    delay(200);
    if((xis>=365&&xis<=415)&&(ysis>=270&&ysis
<=290)&&button==1)
    {
        Mouse_off();
        If(save==0)
        {
            v_at1=0;v_at2=0;save=1;
            setcolor(YELLOW);
            rectangle(365,270,415,290);
            setfillstyle(1,GREEN);
            floodfill(366,271,YELLOW);
            setcolor(WHITE);
            outtextxy(376,277,"SAVE");
            setcolor(WHITE);
            outtextxy(495,360,"EDITION");
            outtextxy(430,390,"File Name");
            outtextxy(487,447,"OK");
            outtextxy(560,447,"NO");
            MODE_SAVE();
        }
    }
    else if(save==1)
    {

```

```

        v_at1=0;v_at2=0;save=0;
        setcolor(YELLOW);
        rectangle(365,270,415,290);
        setfillstyle(1,BLUE);
        floodfill(366,271,YELLOW);
        setcolor(WHITE);
        outtextxy(376,277,"SAVE");
        setcolor(BLACK);
        outtextxy(495,360,"EDITION");
        outtextxy(430,407,"File Name");
        outtextxy(487,447,"OK");
        outtextxy(560,447,"NO");
    }
    else delay(0);
}
else if((xis>=365&&xis<=415)&&(yis>=110&&yis
<=130)&&button==1)
{
    Mouse_off();
    Stop=0;
    V_at1=0;v_at2=0;save=0;
    Setcolor(YELLOW);
    Rectangle(365,110,415,130);
    Setfillstyle(1,BLUE);
    Floodfill(366,111,YELLOW);
    Setcolor(WHITE);
    Outtextxy(376,117,"STOP");
    Setcolor(YELLOW);
    Rectangle(365,270,415,290);
    Setfillstyle(1,BLUE);
    Floodfill(366,271,YELLOW);
    Setcolor(WHITE);
    Outtextxy(376,277,"SAVE");
}
else if((xis>=368&&xis<=413)&&(yis>=233&&yis
<=247)&&button==1)
{
    Mouse_off();
    OPEN_FILE();
}
else delay(0);
}while(stop!=0);
Mouse_show();
}

MODE_SAVE()
{
    int i=0,x=520,no=0;
    start:
    Mouse_show();
    Gotoxy(66,25);scanf("%s",list.file_name);
    Do{
        Mouse_show();
        Checkmouse();
        Delay(200);
    }
}

```

```

        If ((xis>=470&&xis<=520) &&(yis>=440&&yis
<=460) &&button==1)
        {
            Mouse_off();

L) If ((fp=fopen("%s",list.file_name), "a+b"))==NULL)
        {
            exit(0);
        }
        fwrite(&list,sizeof(struct data),1,fp);
        fclose(fp);
        setcolor(YELLOW);
        rectangle(510,381,590,401);
        setfillstyle(1,BLACK);
        floodfill(512,382,YELLOW);
        save=0;
        setcolor(YELLOW);
        rectangle(365,270,415,290);
        setfillstyle(1,BLUE);
        floodfill(366,271,YELLOW);
        setcolor(WHITE);
        outtextxy(376,277,"SAVE");
        setcolor(BLACK);
        outtextxy(495,360,"EDITION");
        outtextxy(430,390,"File Name");
        outtextxy(487,447,"OK");
        outtextxy(560,447,"NO");
    }
    else if ((xis>=540&&xis<=590) &&(yis>=440&&yis
<=460) &&button==1)
    {
        Mouse_off();
        setcolor(YELLOW);
        rectangle(510,381,590,401);
        setfillstyle(1,BLACK);
        floodfill(512,382,YELLOW);
        i=0;x=520;
        buff[0];
        goto start;
    }
    else if ((xis>=365&&xis<=415) &&(yis>=270&&yis
<=290) &&button==1)
    {
        Mouse_off();
        save=0;
        setcolor(YELLOW);
        rectangle(365,270,415,290);
        setfillstyle(1,BLUE);
        floodfill(366,271,YELLOW);
        setcolor(WHITE);
        outtextxy(376,277,"SAVE");
        setcolor(BLACK);
        outtextxy(495,360,"EDITION");
        outtextxy(430,390,"File Name");
    }

```

```

        outtextxy(487,447,"OK");
        outtextxy(560,447,"NO");
    }
    else delay(0);
}while(save!=0);
Mouse_show();
}

/*-----FUNCTION MOUSE-----*/
MOUSE()
{
Mouse_init();
Mouse_show();
//Checkmouse();
}

/*-----CHECK MOUSE-----*/
Checkmouse()
{
    regs.x.ax=3;
    int86(0x33,&regs,&regs);
    xis=(regs.x.cx);
    yis=(regs.x.dx);
    buttom=regs.x.bx;
    ox=xis;
    oy=yis;
    ob=buttom;
return(0);
}

/*-----MOUSE INITIAL-----*/
Mouse_init()
{
    union REGS regs;
    regs.x.ax=0;
    int86(0x33,&regs,&regs);
    return regs.x.ax;
}

/*-----MOUSE SHOE-----*/
Mouse_show()
{
    union REGS regs;
    regs.x.ax=1;
    int86(0x33,&regs,&regs);
    return regs.x.ax;
}

/*-----MOUSE OFF-----*/
Mouse_off()
{
    union REGS regs;
    regs.x.ax=2;
    int86(0x33,&regs,&regs);
    return regs.x.ax;
}

```

```

}

PLOT1()
{
int t,i;
char k;
gotoxy(63,16);printf("          ");
gotoxy(63,14);printf("          ");
setcolor(YELLOW);
  moveto(xpos,ypos);
  t = 100;
delay(0);
  while( t >= 100)
  {
    outportb(0x37a,0x00);
    outportb(0x37a,0x00);
    t = inportb(0x37c);
    outportb(0x37a,0x01);
  }
  while(t<100)
  {
    outportb(0x37a,0x00);
    outportb(0x37a,0x00);
    t = inportb(0x37c);
    outportb(0x37a,0x01);
  }
for(i=1;i<=9000;i++)
{
  outportb(0x37a,0x00);
  list.data_signal[i]=inportb(0x37c);
  outportb(0x37a,0x01);
}
while(list.data_signal[c]>=128)
{
  c++;
  first++;
}
while(list.data_signal[c]<128)
{
  first++;
  c++;
}
while(list.data_signal[c]>=128)
{
  second++;
  c++;
}
while(list.data_signal[c]<128)
{
  second++;
  c++;
}
while(list.data_signal[c]>=128)
{
  third++;
}

```

```

        c++;
    }
    while(list.data_signal[c]<128)
    {
        third++;
        c++;
    }
    while(list.data_signal[c]>=128)
    {
        if((list.data_signal[c]>list.data_signal[c-3])&&(list.data_signal[c]<=list.data_signal[c+3]))
        {
            vh=list.data_signal[c];
        }
        else delay(0);
        fourth++;
        c++;
    }
    while(list.data_signal[c]<128)
    {
        if((list.data_signal[c]<list.data_signal[c-3])&&(list.data_signal[c]>=list.data_signal[c+3]))
        {
            vl=list.data_signal[c];
        }
        else delay(0);
        fourth++;
        c++;
    }
    for(i=1;i<=(700/(nn*2));i++)
    {
        setcolor(YELLOW);
        if(list.data_signal[i]<200)
        ber=list.data_signal[i];
        pot(ypos);
        lineto((i*nn)+5,ber2);
    }
    freqavg=(third+fourth)/2;
    freq=1/(0.000042*freqavg);
    gotoxy(63,6);
    printf("%.2f",freq);
    gotoxy(63,4);
    printf("%.2f", (vh-vl)*0.020);
    first=1;second=1;third=1;fourth=1;
    c=0;
    delay(10);
    if (kbhit())
    {
        k=getch();
        switch(k){
            case '+':
                if (nn <= 10)    nn=nn+0.05;
                break;
            case '-':
                if (nn >0.05)    nn=nn-0.05;

```

```

        break;
    case '/':
        if (ypos<=250) ypos=ypos+5;
        break;
    case '*':
        if (ypos>=80) ypos=ypos-5;
        break;
    case '1':
        nn=0.1;
        break;
    case '2':
        nn=0.5;
        break;
    case '3':
        nn=1;
        break;
    case '4':
        nn=3;
        break;
    case '5':
        nn=5;
        break;
    case '9':
        if (vd <= 2) vd=vd+0.05;
        break;
    case '6':
        if (vd>=0.05) vd=vd-0.05;
        break;
    case '7':
        if (xpos <= 170) xpos=xpos+5;
        break;
    case '8':
        if (xpos>10) xpos=xpos-5;
        break;
    case 27:
    case 13:
        nn = 2000;
    }
}

pot(int yy)
{
    if(ber<=128)
    {
        ber2=yy+(128-ber)*vd;
    }
    else if(ber>128)
    {
        ber2=yy-(ber-128)*vd;
    }
    else delay(0);
}

PLOT2()

```

```

{
int t,i;
char k;
gotoxy(63,6);printf("          ");
gotoxy(63,4);printf("          ");
setcolor(RED);
  moveto(xpos,ypos);
  t = 100;
  delay(0);
  while( t >= 100){
    outportb(0x37a,0x04);
    outportb(0x37a,0x04);
    t = inportb(0x37c);
    outportb(0x37a,0x05);
  }
  while(t<100){
    outportb(0x37a,0x04);
    outportb(0x37a,0x04);
    t = inportb(0x37c);
    outportb(0x37a,0x05);
  }
for(i=1;i<=9000;i++)
{
  outportb(0x37a,0x04);
  list.data_signal[i]=inportb(0x37c);
  outportb(0x37a,0x05);
}
while(list.data_signal[c]>=128)
{
  c++;
  first++;
}
while(list.data_signal[c]<128)
{
  first++;
  c++;
}
while(list.data_signal[c]>=128)
{
  second++;
  c++;
}
while(list.data_signal[c]<128)
{
  second++;
  c++;
}
while(list.data_signal[c]>=128)
{
  third++;
  c++;
}
while(list.data_signal[c]<128)
{
  third++;
}

```

```

    c++;
}
while(list.data_signal[c]>=128)
{
    if((list.data_signal[c]>list.data_signal[c-3])&&(list.data_signal[c]<=list.data_signal[c+3]))
    {
        vh=list.data_signal[c];
    }
    else delay(0);
    fourth++;
    c++;
}
while(list.data_signal[c]<128)
{
    if((list.data_signal[c]<list.data_signal[c-3])&&(list.data_signal[c]>=list.data_signal[c+3]))
    {
        vl=list.data_signal[c];
    }
    else delay(0);
    fourth++;
    c++;
}
for(I=1;i<=(700/(nn*2));i++)
{
    setcolor(LIGHTGREEN);
    if(list.data_signal[i]<200)
    ber=list.data_signal[i];
    pot(ypos);
    lineto((i*nn)+5,ber2);
}
freqavg=(third+fourth)/2;
freq=1/(0.00000425*freqavg);
gotoxy(63,16);
printf("%.2f",freq);
gotoxy(63,14);
printf("%.2f", (vh-vl)*0.020);
first=1;second=1;third=1;fourth=1;
c=0;
delay(10);
if (kbhit())
{
    k=getch();
    switch(k){
        case '+':
            if (nn <= 10)    nn=nn+0.05;
            break;
        case '-':
            if (nn >0.05)    nn=nn-0.05;
            break;
        case '/':
            if(ypos<=250) ypos=ypos+5;
            break;
        case '*':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (ypos >= 80)    ypos = ypos - 5;
        break;
    case '1':
        nn = 0.1;
        break;
    case '2':
        nn = 0.5;
        break;
    case '3':
        nn = 1;
        break;
    case '4':
        nn = 3;
        break;
    case '5':
        nn = 5;
        break;
    case '9':
        if (vd <= 2)    vd = vd + 0.05;
        break;
    case '6':
        if (vd >= 0.05)    vd = vd - 0.05;
        break;
    case '7':
        if (xpos <= 170)    xpos = xpos + 5;
        break;
    case '8':
        if (xpos > 10)    xpos = xpos - 5;
        break;

    case 27:
    case 13:
        nn = 2000;
    }
}
PLOT_DC1()
{
    moveto(6,160);
    output(0x37a,0x0a);
    output(0x378,0x59);
    for(I=1;i<=1000;i++)
    {
        outputb(0x37a,0x00);
        list.data_signal[i]=inportb(0x37c);
        outputb(0x37a,0x01);
    }
    for(i=1;i<=350;i++)
    {
        setcolor(YELLOW);
        list.data_signal[i];
        lineto(i+5,232-(list.data_signal[i]*0.80));
    }
    gotoxy(63,4);
}

```

```

        printf("%.2f",list.data_signal[i]*0.01);
    }

PLOT_DC2()
{
    moveto(6,160);
    outport(0x37a,0x0a);
    outport(0x378,0x99);
    for(i=1;i<=1000;i++)
    {
        outportb(0x37a,0x04);
        list.data_signal[i]=inportb(0x37c);
        outportb(0x37a,0x05);
    }
    for(i=1;i<=350;i++)
    {
        setcolor(YELLOW);
        list.data_signal[i];
        lineto(i+5,232-(list.data_signal[i]*0.80));
    }
}

VOLT_CH1()
{ int sx,sy,i,j;
  sx=30,sy=385;
  for(j=1;j<=2;j++)
  {
      for(i=1;i<=4;i++)
      {
          setcolor(BLACK);
          circle(sx,sy,5);
          setfillstyle(1,WHITE);
          floodfill(sx+1,sy,BLACK);
          sy=sy+14;
      }
      sy=385;sx=sx+80;
  }
  setcolor(BLACK);
  circle(30,427,2);
  setfillstyle(1,BLACK);
  floodfill(31,427,BLACK);
  outportb(0x37a,0x0a);
  outportb(0x378,0x00);
  setcolor(RED);
  outtextxy(45,382,"0.1");
  outtextxy(45,396,"0.2");
  outtextxy(45,410,"0.5");
  outtextxy(45,424," 1");
  outtextxy(125,382," 2");
  outtextxy(125,396," 5");
  outtextxy(125,410,"10");
  outtextxy(125,424,"20");
}

VOLT_CH2()
{ int sx,sy,i,j;
  sx=240,sy=385;

```

```

for(j=1;j<=2;j++)
{
    for(i=1;i<=4;i++)
    {
        setcolor(BLACK);
        circle(sx,sy,5);
        setfillstyle(1,WHITE);
        floodfill(sx+1,sy,BLACK);
        sy=sy+14;
    }
    sy=385;sx=sx+80;
}
setcolor(BLACK);
circle(240,427,2);
setfillstyle(1,BLACK);
floodfill(241,427,BLACK);
setcolor(RED);
outtextxy(255,382,"0.1");
outtextxy(255,396,"0.2");
outtextxy(255,410,"0.5");
outtextxy(255,424," 1");
outtextxy(335,382," 2");
outtextxy(335,396," 5");
outtextxy(335,410,"10");
outtextxy(335,424,"20");
}

CLEAR_CH1()
{ int sx,sy,i,j;
sx=30,sy=385;
for(j=1;j<=2;j++)
{
    for(i=1;i<=4;i++)
    {
        setcolor(WHITE);
        circle(sx,sy,2);
        setfillstyle(1,WHITE);
        floodfill(sx+1,sy,WHITE);
        sy=sy+14;
    }
    sy=385;sx=sx+80;
}
}

CLEAR_CH2()
{ int sx,sy,i,j;
sx=240,sy=385;
for(j=1;j<=2;j++)
{
    for(i=1;i<=4;i++)
    {
        setcolor(WHITE);
        circle(sx,sy,2);
        setfillstyle(1,WHITE);
        floodfill(sx+1,sy,WHITE);
        sy=sy+14;
    }
}
}

```

```

    }
    sy=385;sx=sx+80;
  }
}

CHECK_CH1()
{
if((xis>=25&&xis<=35)&&(yis>=380&&yis
<=390)&&button==1)
{
  Mouse_off();
  CLEAR_CH1();
  setcolor(BLACK);
  circle(30,385,2);
  setfillstyle(1,BLACK);
  floodfill(31,385,BLACK);
  if(ac_1==1)
  {
    outportb(0x37a,0x0a);
    outportb(0x378,0x3f);
  }
  else if(dc_1==1)
  {
    outportb(0x37a,0x0a);
    outportb(0x378,0x90);
  }
  else delay(0);
  Mouse_show();
}
else if((xis>=25&&xis<=35)&&(yis>=394&&yis
<=404)&&button==1)
{
  Mouse_off();
  CLEAR_CH1();
  setcolor(BLACK);
  circle(30,399,2);
  setfillstyle(1,BLACK);
  floodfill(31,399,BLACK);
  if(ac_1==1)
  {
    outportb(0x37a,0x0a);
    outportb(0x378,0x36);
  }
  else if(dc_1==1)
  {
    outportb(0x37a,0x0a);
    outportb(0x378,0x87);
  }
  else delay(0);
  Mouse_show();
}
else if((xis>=25&&xis<=35)&&(yis>=408&&yis
<=418)&&button==1)
{
  Mouse_off();

```

```

CLEAR_CH1();
setcolor(BLACK);
circle(30,413,2);
setfillstyle(1,BLACK);
floodfill(31,413,BLACK);
if(ac_1==1)
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x2d);
}
else if(dc_1==1)
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x7e);
}
else delay(0);
Mouse_show();
}
else if((xis>=25&&xis<=35)&&(yis>=422&&yis
<=432)&&button==1)
{
    Mouse_off();
    CLEAR_CH1();
    setcolor(BLACK);
    circle(30,427,2);
    setfillstyle(1,BLACK);
    floodfill(31,427,BLACK);
    if(ac_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x00);
    }
    else if(dc_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x51);
    }
    else delay(0);
    Mouse_show();
}
else if((xis>=105&&xis<=115)&&(yis>=380&&yis
<=390)&&button==1)
{
    Mouse_off();
    CLEAR_CH1();
    setcolor(BLACK);
    circle(110,385,2);
    setfillstyle(1,BLACK);
    floodfill(111,385,BLACK);
    if(ac_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x09);
    }
    else if(dc_1==1)

```

```

        {
            outportb(0x37a,0x0a);
            outportb(0x378,0x5a);
        }
    else delay(0);
    Mouse_show();
}
else if((xis>=105&&xis<=115)&&(yis>=394&&yis
<=404)&&button==1)
{
    Mouse_off();
    CLEAR_CH1();
    setcolor(BLACK);
    circle(110,399,2);
    setfillstyle(1,BLACK);
    floodfill(111,399,BLACK);
    if(ac_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x12);
    }
    else if(dc_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x63);
    }
    else delay(0);
    Mouse_show();
}
else if((xis>=105&&xis<=115)&&(yis>=408&&yis
<=418)&&button==1)
{
    Mouse_off();
    CLEAR_CH1();
    setcolor(BLACK);
    circle(110,413,2);
    setfillstyle(1,BLACK);
    floodfill(111,413,BLACK);
    if(ac_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x1b);
    }
    else if(dc_1==1)
    {
        outportb(0x37a,0x0a);
        outportb(0x378,0x6c);
    }
    else delay(0);
    Mouse_show();
}
else if((xis>=105&&xis<=115)&&(yis>=422&&yis
<=432)&&button==1)
{
    Mouse_off();
}

```

```

CLEAR_CH1();
setcolor(BLACK);
circle(110,427,2);
setfillstyle(1,BLACK);
floodfill(111,427,BLACK);
if(ac_1==1)
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x24);
}
else if(dc_1==1)
{
    outportb(0x37a,0x0a);
    outportb(0x378,0x75);
}
else delay(0);
Mouse_show();
}
else delay(0);
}

CHECK_CH2()
{
if((xis>=235&&xis<=245)&&(yis>=380&&yis
<=390)&&button==1)
{
    Mouse_off();
    CLEAR_CH2();
    setcolor(BLACK);
    circle(240,385,2);
    setfillstyle(1,BLACK);
    floodfill(241,385,BLACK);
    outportb(0x37a,0x0a);
    outportb(0x378,0x08);
    Mouse_show();
}
else if((xis>=235&&xis<=245)&&(yis>=394&&yis
<=404)&&button==1)
{
    Mouse_off();
    CLEAR_CH2();
    setcolor(BLACK);
    circle(240,399,2);
    setfillstyle(1,BLACK);
    floodfill(241,399,BLACK);
    outportb(0x37a,0x0a);
    outportb(0x378,0x07);
    Mouse_show();
}
else if((xis>=235&&xis<=245)&&(yis>=408&&yis
<=418)&&button==1)
{
    Mouse_off();
    CLEAR_CH2();
    setcolor(BLACK);

```

```

circle(240,413,2);
setfillstyle(1,BLACK);
floodfill(241,413,BLACK);
outportb(0x37a,0x0a);
outportb(0x378,0x06);
Mouse_show();
}
else if((xis>=235&&xis<=245)&&(yis>=422&&yis
<=432)&&button==1)
{
Mouse_off();
CLEAR_CH2();
setcolor(BLACK);
circle(240,427,2);
setfillstyle(1,BLACK);
floodfill(241,427,BLACK);
outportb(0x37a,0x0a);
outportb(0x378,0x00);
Mouse_show();
}
else if((xis>=315&&xis<=325)&&(yis>=380&&yis
<=390)&&button==1)
{
Mouse_off();
CLEAR_CH2();
setcolor(BLACK);
circle(320,385,2);
setfillstyle(1,BLACK);
floodfill(321,385,BLACK);
outportb(0x37a,0x0a);
outportb(0x378,0x01);
Mouse_show();
}
else if((xis>=315&&xis<=325)&&(yis>=394&&yis
<=404)&&button==1)
{
Mouse_off();
CLEAR_CH2();
setcolor(BLACK);
circle(320,399,2);
setfillstyle(1,BLACK);
floodfill(321,399,BLACK);
outportb(0x37a,0x0a);
outportb(0x378,0x02);
Mouse_show();
}
else if((xis>=315&&xis<=325)&&(yis>=408&&yis
<=418)&&button==1)
{
Mouse_off();
CLEAR_CH2();
setcolor(BLACK);
circle(320,413,2);
setfillstyle(1,BLACK);
floodfill(321,413,BLACK);

```

```

    outportb(0x37a,0x0a);
    outportb(0x378,0x03);
    Mouse_show();
}
else if((xis>=315&&xis<=325)&&(yis>=422&&yis
<=432)&&button==1)
{
    Mouse_off();
    CLEAR_CH2();
    setcolor(BLACK);
    circle(320,427,2);
    setfillstyle(1,BLACK);
    floodfill(321,427,BLACK);
    outportb(0x37a,0x0a);
    outportb(0x378,0x04);
    Mouse_show();
}
else delay(0);
}

CHECK_RL()
{
if((xis>=365&&xis<=415)&&(yis>=150&&yis
<=170)&&button==1)
{
    Mouse_off();
    setcolor(YELLOW);
    rectangle(365,150,415,170);
    setfillstyle(1,GREEN);
    floodfill(366,151,YELLOW);
    setcolor(WHITE);
    outtextxy(372,157,"V_at1");
    Mouse_show();
    do
    {
    Checkmouse();
    if(button==0)
        {
            Mouse_off();
            setcolor(YELLOW);
            rectangle(365,150,415,170);
            setfillstyle(1,BLUE);
            floodfill(366,151,YELLOW);
            setcolor(WHITE);
            outtextxy(372,157,"V_at1");
            Mouse_show();
        }
        else delay(0);
    }while(button!=0);
}
else if((xis>=365&&xis<=415)&&(yis>=190&&yis
<=210)&&button==1)
{
    Mouse_off();
    setcolor(YELLOW);

```

```

rectangle(365,190,415,210);
setfillstyle(1, GREEN);
floodfill(366,191, YELLOW);
setcolor(WHITE);
outtextxy(372,197, "V_at2");
Mouse_show();
do
{
Checkmouse();
if(button==0)
{
Mouse_off();
setcolor(YELLOW);
rectangle(365,190,415,210);
setfillstyle(1, BLUE);
floodfill(366,191, YELLOW);
setcolor(WHITE);
outtextxy(372,197, "V_at2");
Mouse_show();
}
else delay(0);
}while(button!=0);
else delay(0);
}

```

รูปที่ ค.5 โปรแกรมควบคุมการทำงานเครื่องวัดและบันทึกสัญญาณ โดยใช้ไมโครคอมพิวเตอร์



ภาคผนวก ง
รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



June 1999

ADC0820 8-Bit High Speed μ P Compatible A/D Converter with Track/Hold Function

General Description

By using a half-flash conversion technique, the 8-bit ADC0820 CMOS A/D offers a 1.5 μ s conversion time and dissipates only 75 mW of power. The half-flash technique consists of 32 comparators, a most significant 4-bit ADC and a least significant 4-bit ADC.

The input to the ADC0820 is tracked and held by the input sampling circuitry eliminating the need for an external sample-and-hold for signals moving at less than 100 mV/ μ s. For ease of interface to microprocessors, the ADC0820 has been designed to appear as a memory location or I/O port without the need for external interfacing logic.

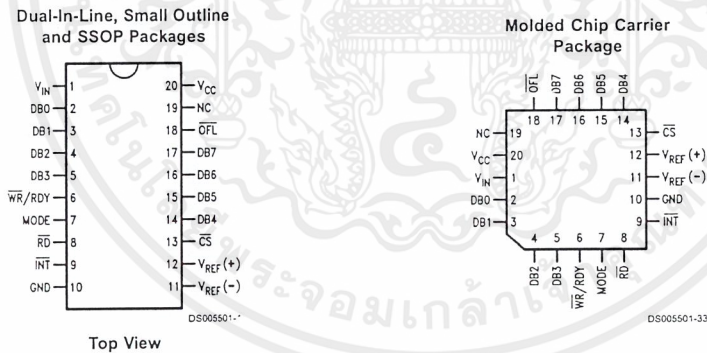
Key Specifications

- Resolution 8 Bits
- Conversion Time 2.5 μ s Max (RD Mode)
1.5 μ s Max (WR-RD Mode)
- Low Power 75 mW Max
- Total Unadjusted Error $\pm 1/2$ LSB and ± 1 LSB

Features

- Built-in track-and-hold function
- No missing codes
- No external clocking
- Single supply — 5 V_{DC}
- Easy interface to all microprocessors, or operates stand-alone
- Latched TRI-STATE® output
- Logic inputs and outputs meet both MOS and T²L voltage level specifications
- Operates ratiometrically or with any reference value equal to or less than V_{CC}
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Overflow output available for cascading
- 0.3" standard width 20-pin DIP
- 20-pin molded chip carrier package
- 20-pin small outline package
- 20-pin shrink small outline package (SSOP)

Connection and Functional Diagrams

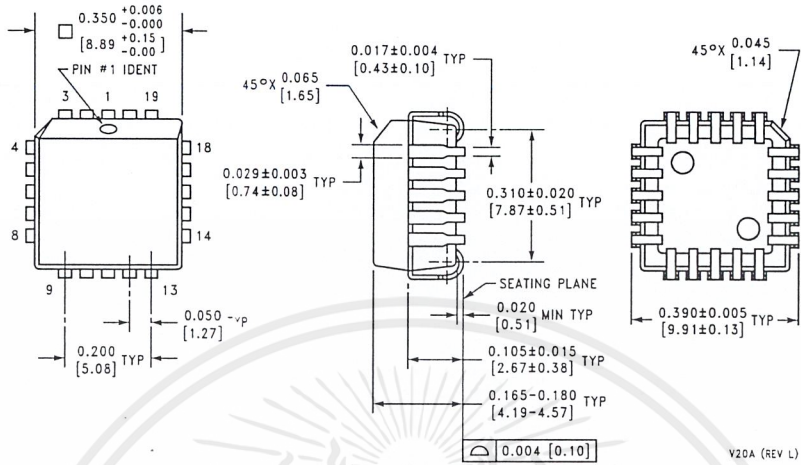


TRI-STATE® is a registered trademark of National Semiconductor Corporation.

ADC0820 8-Bit High Speed μ P Compatible A/D Converter with Track/Hold Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Chip Carrier Package (V)
 Order Number ADC0820BCV
 NS Package Number V20A

V20A (REV L)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation Americas
 Tel: 1-800-272-9959
 Fax: 1-800-737-7018
 Email: support@nsc.com

National Semiconductor Europe
 Fax: +49 (0) 1 80-530 85 86
 Email: europe.support@nsc.com
 Deutsch Tel: +49 (0) 1 80-530 85 85
 English Tel: +49 (0) 1 80-532 78 32
 Français Tel: +49 (0) 1 80-532 93 58
 Italiano Tel: +49 (0) 1 80-534 16 80

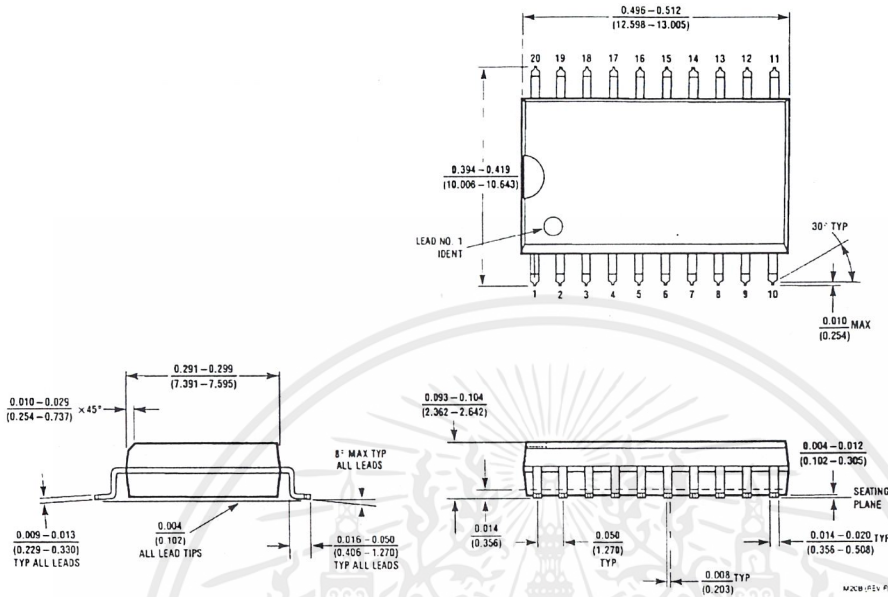
National Semiconductor Asia Pacific Customer Response Group
 Tel: 65-2544466
 Fax: 65-2504466
 Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
 Tel: 81-3-5639-7560
 Fax: 81-3-5639-7507

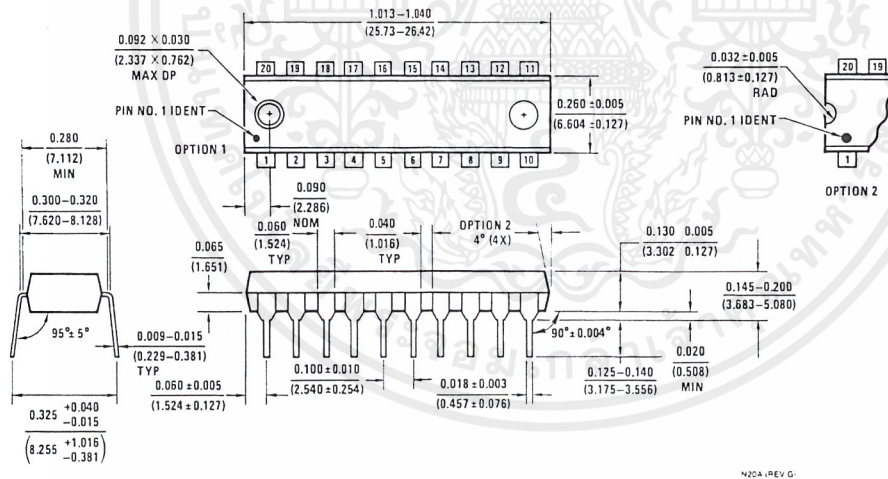
www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



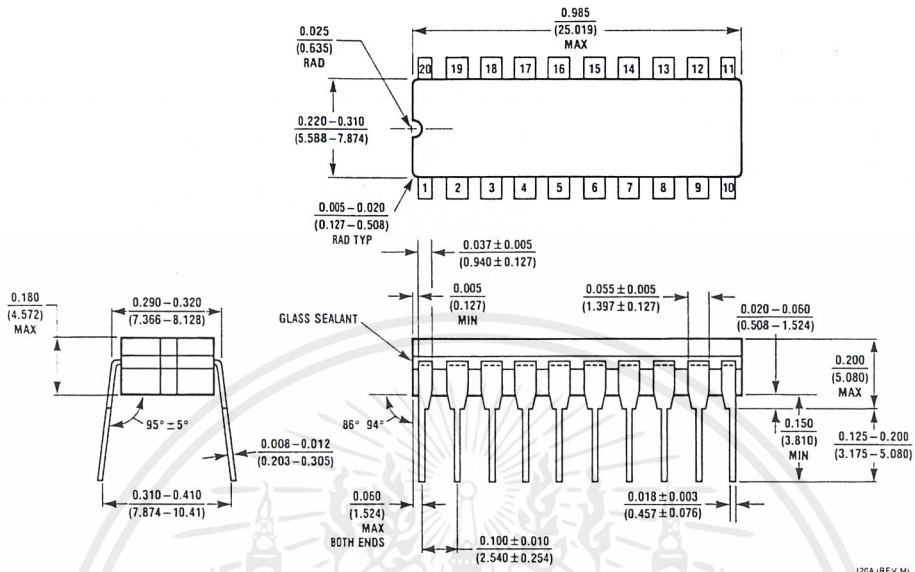
SO Package (M)
 Order Number ADC0820BCWM, ADC0820CCWM or ADC0820CIWM
 NS Package Number M20B



Molded Dual-In-Line Package (N)
 Order Number ADC0820BCN or ADC0820CCN
 NS Package Number N20A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

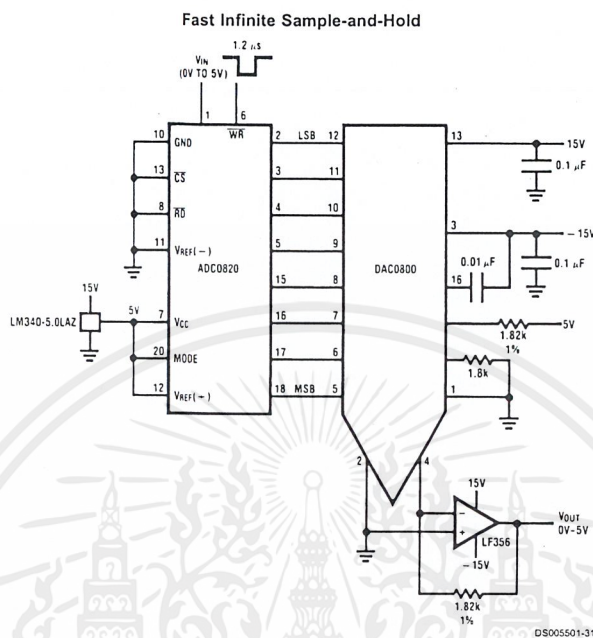
Physical Dimensions inches (millimeters) unless otherwise noted



Hermetic Dual-In-Line Package (J)
Order Number ADC0820CCJ
NS Package Number J20A

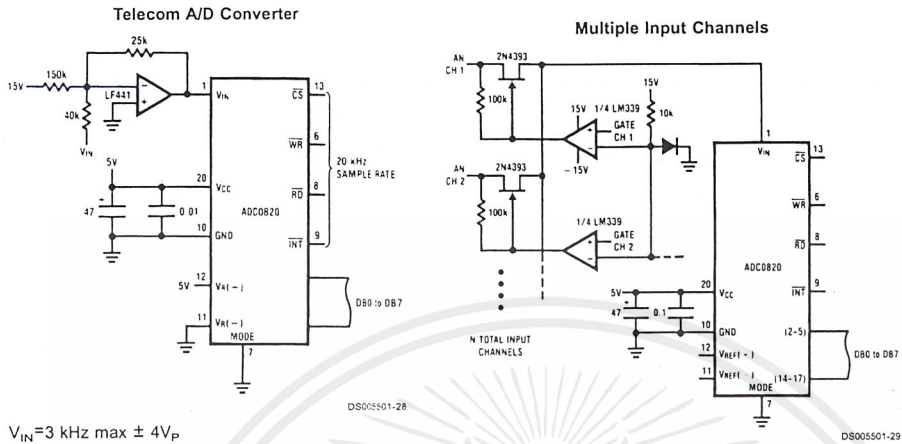
J20A (REV M)

3.0 Typical Applications (Continued)

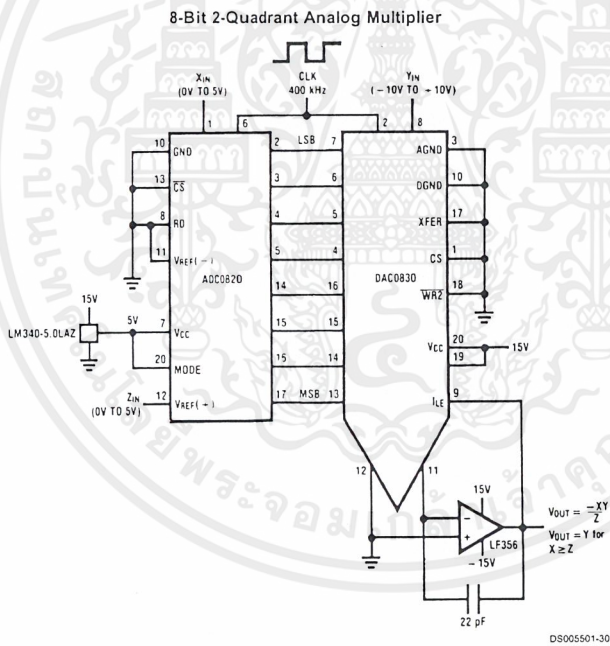


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.0 Typical Applications (Continued)



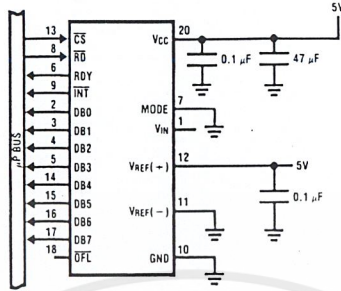
- V_{IN} = 3 kHz max \pm 4V_P
- No track-and-hold needed
- Low power consumption



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

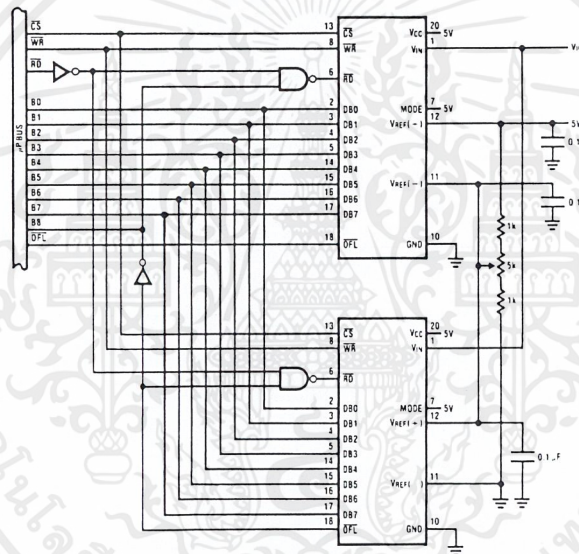
3.0 Typical Applications

8-Bit Resolution Configuration



DS005501-26

9-Bit Resolution Configuration



DS005501-27

2.0 Analog Considerations (Continued)

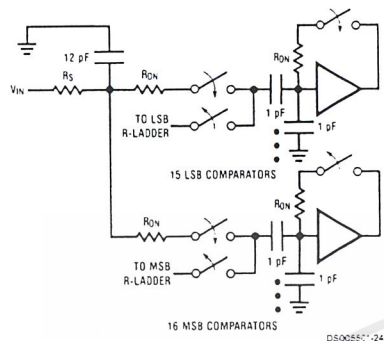


FIGURE 14.

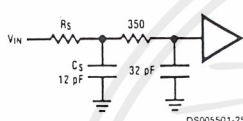


FIGURE 15.

2.3 INPUT FILTERING

It should be made clear that transients in the analog input signal, caused by charging current flowing into V_{IN} , will not degrade the A/D's performance in most cases. In effect the ADC0820 does not "look" at the input when these transients occur. The comparators' outputs are not latched while WR is low, so at least 600 ns will be provided to charge the ADC's input capacitance. It is therefore not necessary to filter out these transients by putting an external cap on the V_{IN} terminal.

2.4 INHERENT SAMPLE-HOLD

Another benefit of the ADC0820's input mechanism is its ability to measure a variety of high speed signals without the help of an external sample-and-hold. In a conventional SAR type converter, regardless of its speed, the input must remain at least $\frac{1}{2}$ LSB stable throughout the conversion process if full accuracy is to be maintained. Consequently, for many high speed signals, this signal must be externally sampled, and held stationary during the conversion.

Sampled-data comparators, by nature of their input switching, already accomplish this function to a large degree (Section 1.2). Although the conversion time for the ADC0820 is 1.5 μ s, the time through which V_{IN} must be $\frac{1}{2}$ LSB stable is much smaller. Since the MS flash ADC uses V_{IN} as its "compare" input and the LS ADC uses V_{IN} as its "zero" input, the ADC0820 only "samples" V_{IN} when WR is low (Sections 1.3 and 2.2). Even though the two flashes are not done simultaneously, the analog signal is measured at one instant. The value of V_{IN} approximately 100 ns after the rising edge of WR (100 ns due to internal logic prop delay) will be the measured value.

Input signals with slew rates typically below 100 mV/ μ s can be converted without error. However, because of the input time constants, and charge injection through the opened comparator input switches, faster signals may cause errors. Still, the ADC0820's loss in accuracy for a given increase in signal slope is far less than what would be witnessed in a conventional successive approximation device. An SAR type converter with a conversion time as fast as 1 μ s would still not be able to measure a 5V 1 kHz sine wave without the aid of an external sample-and-hold. The ADC0820, with no such help, can typically measure 5V, 7 kHz waveforms.

2.0 Analog Considerations

2.1 REFERENCE AND INPUT

The two V_{REF} inputs of the ADC0820 are fully differential and define the zero to full-scale input range of the A to D converter. This allows the designer to easily vary the span of the analog input since this range will be equivalent to the voltage difference between $V_{IN}(+)$ and $V_{IN}(-)$. By reducing $V_{REF}(V_{REF}=V_{REF}(+)-V_{REF}(-))$ to less than 5V, the sensitivity of the converter can be increased (i.e., if $V_{REF}=2V$ then 1 LSB=7.8 mV). The input/reference arrangement also facilitates ratiometric operation and in many cases the chip power supply can be used for transducer power as well as the V_{REF} source.

This reference flexibility lets the input span not only be varied but also offset from zero. The voltage at $V_{REF}(-)$ sets the input level which produces a digital output of all zeroes. Though V_{IN} is not itself differential, the reference design affords nearly differential-input capability for most measurement applications. Figure 13 shows some of the configurations that are possible.

2.2 INPUT CURRENT

Due to the unique conversion techniques employed by the ADC0820, the analog input behaves somewhat differently than in conventional devices. The A/D's sampled-data comparators take varying amounts of input current depending on which cycle the conversion is in.

The equivalent input circuit of the ADC0820 is shown in Figure 14. When a conversion starts (\overline{WR} low, \overline{WR} -RD mode), all input switches close, connecting V_{IN} to thirty-one 1 pF capacitors. Although the two 4-bit flash circuits are not both in their compare cycle at the same time, V_{IN} still sees all input capacitors at once. This is because the MS flash converter is connected to the input during its compare interval and the LS flash is connected to the input during its zeroing phase (Section 1.3). In other words, the LS ADC uses V_{IN} as its zero-phase input.

The input capacitors must charge to the input voltage through the on resistance of the analog switches (about 5 k Ω to 10 k Ω). In addition, about 12 pF of input stray capacitance must also be charged. For large source resistances, the analog input can be modeled as an RC network as shown in Figure 15. As R_S increases, it will take longer for the input capacitance to charge.

In RD mode, the input switches are closed for approximately 800 ns at the start of the conversion. In \overline{WR} -RD mode, the time that the switches are closed to allow this charging is the time that \overline{WR} is low. Since other factors force this time to be at least 600 ns, input time constants of 100 ns can be accommodated without special consideration. Typical total input capacitance values of 45 pF allow R_S to be 1.5 k Ω without lengthening \overline{WR} to give V_{IN} more time to settle.

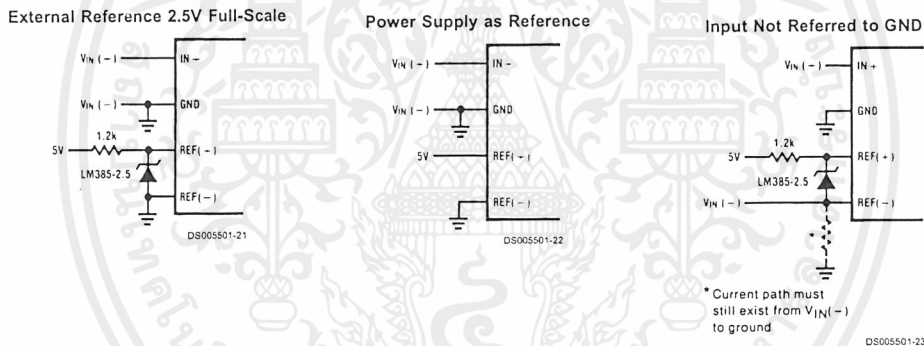
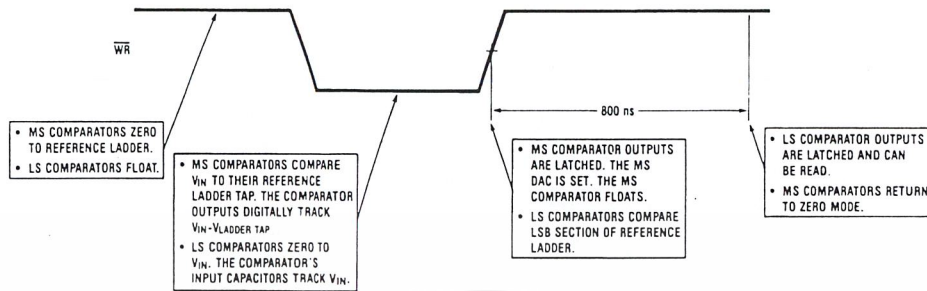


FIGURE 13. Analog Input Options

1.0 Functional Description (Continued)



Note: MS means most significant
LS means least significant

DS005501-20

FIGURE 11. Operating Sequence (WR-RD Mode)

OTHER INTERFACE CONSIDERATIONS

In order to maintain conversion accuracy, \overline{WR} has a maximum width spec of 50 μ s. When the MS flash ADC's sampled-data comparators (Section 1.2) are in comparison mode (\overline{WR} is low), the input capacitors (C, Figure 8) must hold their charge. Switch leakage and inverter bias current can cause errors if the comparator is left in this phase for too long.

Since the MS flash ADC enters its zeroing phase at the end of a conversion (Section 1.3), a new conversion cannot be started until this phase is complete. The minimum spec for this time (t_p , Figures 2, 3, 4, 5) is 500 ns.

1.0 Functional Description (Continued)

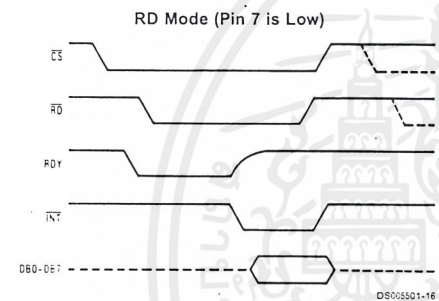
sured while \overline{WR} is low. In RD mode, sampling occurs during the first 800 ns of RD. Because of the input connections to the ADC0820's LS and MS comparators, the converter has the ability to sample V_{IN} at one instant (Section 2.4), despite the fact that two separate 4-bit conversions are being done. More specifically, when \overline{WR} is low the MS flash is in compare mode (connected to V_{IN}), and the LS flash is in zero mode (also connected to V_{IN}). Therefore both flash ADCs sample V_{IN} at the same time.

1.4 DIGITAL INTERFACE

The ADC0820 has two basic interface modes which are selected by strapping the MODE pin high or low.

RD Mode

With the MODE pin grounded, the converter is set to Read mode. In this configuration, a complete conversion is done by pulling \overline{RD} low until output data appears. An \overline{INT} line is provided which goes low at the end of the conversion as well as a RDY output which can be used to signal a processor that the converter is busy or can also serve as a system Transfer Acknowledge signal.



When in RD mode, the comparator phases are internally triggered. At the falling edge of \overline{RD} , the MS flash converter goes from zero to compare mode and the LS ADC's comparators enter their zero cycle. After 800 ns, data from the MS flash is latched and the LS flash ADC enters compare mode. Following another 800 ns, the lower 4 bits are recovered.

WR then RD Mode

With the MODE pin tied high, the A/D will be set up for the WR-RD mode. Here, a conversion is started with the \overline{WR} input; however, there are two options for reading the output data which relate to interface timing. If an interrupt driven scheme is desired, the user can wait for \overline{INT} to go low before reading the conversion result (Figure 10). \overline{INT} will typically go low 800 ns after \overline{WR} 's rising edge. However, if a shorter

conversion time is desired, the processor need not wait for \overline{INT} and can exercise a read after only 600 ns (Figure 9). If this is done, \overline{INT} will immediately go low and data will appear at the outputs.

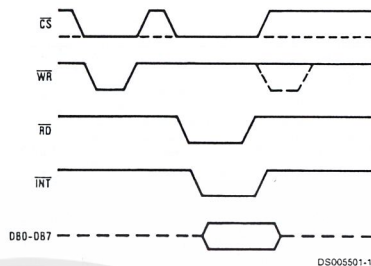


FIGURE 9. WR-RD Mode (Pin 7 is High and $t_{RD} < t_i$)

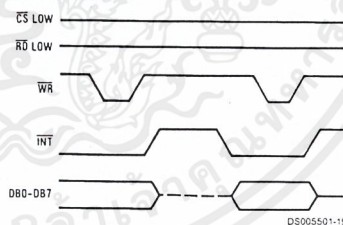


FIGURE 10. WR-RD Mode (Pin 7 is High and $t_{RD} > t_i$)

Stand-Alone

For stand-alone operation in WR-RD mode, \overline{CS} and \overline{RD} can be tied low and a conversion can be started with \overline{WR} . Data will be valid approximately 800 ns following \overline{WR} 's rising edge.

WR-RD Mode (Pin 7 is High) Stand-Alone Operation



1.0 Functional Description (Continued)

ladder for the A/D as well as for generating the DAC signal. The DAC output is actually the tap on the resistor ladder which most closely approximates the analog input. In addition, the "sampled-data" comparators used in the ADC0820 provide the ability to compare the magnitudes of several analog signals simultaneously, without using input summing amplifiers. This is especially useful in the LS flash ADC, where the signal to be converted is an analog difference.

1.2 THE SAMPLED-DATA COMPARATOR

Each comparator in the ADC0820 consists of a CMOS inverter with a capacitively coupled input (Figures 6, 7). Analog switches connect the two comparator inputs to the input capacitor (C) and also connect the inverter's input and output. A comparison requires two cycles, one for zeroing the comparator, and another for making the comparison.

In the first cycle, one input switch and the inverter's feedback switch (Figure 6) are closed. In this interval, C is charged to the connected input (V1) less the inverter's bias voltage (V_B, approximately 1.2V). In the second cycle (Figure 7), these two switches are opened and the other (V2) input's switch is closed. The input capacitor now subtracts its stored voltage from the second input and the difference is amplified by the inverter's open loop gain. The inverter's input (V_B') becomes

$$V_B' - V_B = (V1 - V2) \frac{C}{C + C_S}$$

and the output will go high or low depending on the sign of V_B' - V_B.

The actual circuitry used in the ADC0820 is a simple but important expansion of the basic comparator described above. By adding a second capacitor and another set of switches to the input (Figure 8), the scheme can be expanded to make dual differential comparisons. In this circuit, the feedback switch and one input switch on each capacitor (Z switches) are closed in the zeroing cycle. A comparison is then made

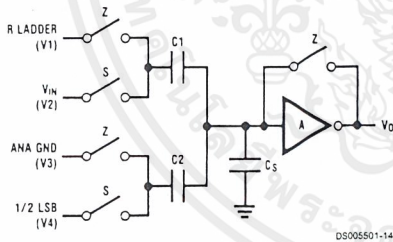


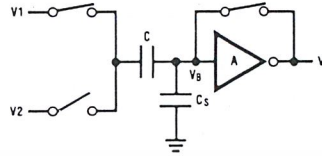
FIGURE 8. ADC0820 Comparator (from MS Flash ADC)

1.3 ARCHITECTURE

In the ADC0820, one bank of 15 comparators is used in each 4-bit flash A/D converter (Figure 12). The MS (most significant) flash ADC also has one additional comparator to detect input overrange. These two sets of comparators operate alternately, with one group in its zeroing cycle while the other is comparing.

When a typical conversion is started, the \overline{WR} line is brought low. At this instant the MS comparators go from zeroing to comparison mode (Figure 11). When \overline{WR} is returned high

by connecting the second input on each capacitor and opening all of the other switches (S switches). The change in voltage at the inverter's input, as a result of the change in charge on each input capacitor, will now depend on both input signal differences.

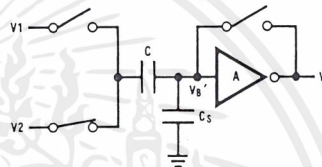


DS005501-12

- V_O = V_B
- V on C = V1 - V_B
- C_S = stray input node capacitor
- V_B = inverter input bias voltage

Zeroing Phase

FIGURE 6. Sampled-Data Comparator



DS005501-13

- V_B' - V_B = (V2 - V1) $\frac{C}{C + C_S}$
- V_O' = $\frac{-A}{C + C_S} [CV2 - CV1]$
- V_O' is dependent on V2 - V1

Compare Phase

FIGURE 7. Sampled-Data Comparator

$$V_O = \frac{-A}{C1 + C2 + C_S} [C1(V2 - V1) + C2(V4 - V3)]$$

$$= \frac{-A}{C1 + C2 + C_S} [\Delta Q_{C1} + \Delta Q_{C2}]$$

DS005501-15

after at least 600 ns, the output from the first set of comparators (the first flash) is decoded and latched. At this point the two 4-bit converters change modes and the LS (least significant) flash ADC enters its compare cycle. No less than 600 ns later, the \overline{RD} line may be pulled low to latch the lower 4 data bits and finish the 8-bit conversion. When \overline{RD} goes low, the flash A/Ds change state once again in preparation for the next conversion.

Figure 11 also outlines how the converter's interface timing relates to its analog input (V_{IN}). In \overline{WR} - \overline{RD} mode, V_{IN} is mea-

Description of Pin Functions

| Pin | Name | Function | Pin | Name | Function |
|-----|-------------------------|---|-----|------------------|--|
| 1 | V_{IN} | Analog input; range $=GND \leq V_{IN} \leq V_{CC}$ | 9 | \overline{INT} | WR-RD Mode \overline{INT} going low indicates that the conversion is completed and the data result is in the output latch. \overline{INT} will go low, ~ 800 ns (the preset internal time out, t_i) after the rising edge of \overline{WR} (see Figure 4); or \overline{INT} will go low after the falling edge of \overline{RD} , if \overline{RD} goes low prior to the 800 ns time out (see Figure 3). \overline{INT} is reset by the rising edge of \overline{RD} or \overline{CS} (see Figures 3, 4). RD Mode \overline{INT} going low indicates that the conversion is completed and the data result is in the output latch. \overline{INT} is reset by the rising edge of \overline{RD} or \overline{CS} (see Figure 2). |
| 2 | DB0 | TRI-STATE data output—bit 0 (LSB) | 10 | GND | Ground |
| 3 | DB1 | TRI-STATE data output—bit 1 | 11 | $V_{REF(-)}$ | The bottom of resistor ladder, voltage range: $GND \leq V_{REF(-)} \leq V_{REF(+)}$ (Note 5) |
| 4 | DB2 | TRI-STATE data output—bit 2 | 12 | $V_{REF(+)}$ | The top of resistor ladder, voltage range: $V_{REF(-)} \leq V_{REF(+)} \leq V_{CC}$ (Note 5) |
| 5 | DB3 | TRI-STATE data output—bit 3 | 13 | \overline{CS} | \overline{CS} must be low in order for the \overline{RD} or \overline{WR} to be recognized by the converter. |
| 6 | \overline{WR} /RDY | WR-RD Mode \overline{WR} : With \overline{CS} low, the conversion is started on the falling edge of \overline{WR} . Approximately 800 ns (the preset internal time out, t_i) after the \overline{WR} rising edge, the result of the conversion will be strobed into the output latch, provided that \overline{RD} does not occur prior to this time out (see Figures 3, 4). RD Mode RDY: This is an open drain output (no internal pull-up device). RDY will go low after the falling edge of \overline{CS} ; RDY will go TRI-STATE when the result of the conversion is strobed into the output latch. It is used to simplify the interface to a microprocessor system (see Figure 2). | 14 | DB4 | TRI-STATE data output—bit 4 |
| 7 | Mode | Mode: Mode selection input—it is internally tied to GND through a 50 μA current source. RD Mode: When mode is low WR-RD Mode: When mode is high | 15 | DB5 | TRI-STATE data output—bit 5 |
| 8 | \overline{RD} | WR-RD Mode With \overline{CS} low, the TRI-STATE data outputs (DB0-DB7) will be activated when \overline{RD} goes low (see Figure 5). \overline{RD} can also be used to increase the speed of the converter by reading data prior to the preset internal time out (t_i , ~ 800 ns). If this is done, the data result transferred to output latch is latched after the falling edge of the \overline{RD} (see Figures 3, 4). RD Mode With \overline{CS} low, the conversion will start with \overline{RD} going low, also \overline{RD} will enable the TRI-STATE data outputs at the completion of the conversion. RDY going TRI-STATE and \overline{INT} going low indicates the completion of the conversion (see Figure 2). | 16 | DB6 | TRI-STATE data output—bit 6 |
| | | | 17 | DB7 | TRI-STATE data output—bit 7 (MSB) |
| | | | 18 | \overline{OFL} | Overflow output—If the analog input is higher than the $V_{REF(+)}$, \overline{OFL} will be low at the end of conversion. It can be used to cascade 2 or more devices to have more resolution (9, 10-bit). This output is always active and does not go into TRI-STATE as DB0-DB7 do. |
| | | | 19 | NC | No connection |
| | | | 20 | V_{CC} | Power supply voltage |

1.0 Functional Description

1.1 GENERAL OPERATION

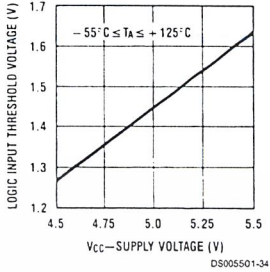
The ADC0820 uses two 4-bit flash A/D converters to make an 8-bit measurement (Figure 1). Each flash ADC is made up of 15 comparators which compare the unknown input to a reference ladder to get a 4-bit result. To take a full 8-bit reading, one flash conversion is done to provide the 4 most significant data bits (via the MS flash ADC). Driven by the 4

MSBs, an internal DAC recreates an analog approximation of the input voltage. This analog signal is then subtracted from the input, and the difference voltage is converted by a second 4-bit flash ADC (the LS ADC), providing the 4 least significant bits of the output data word.

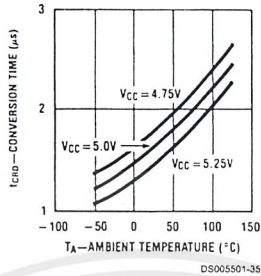
The internal DAC is actually a subsection of the MS flash converter. This is accomplished by using the same resistor

Typical Performance Characteristics

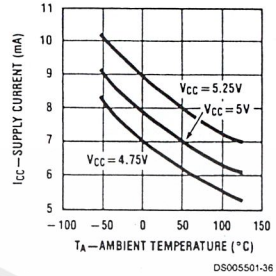
Logic Input Threshold Voltage vs Supply Voltage



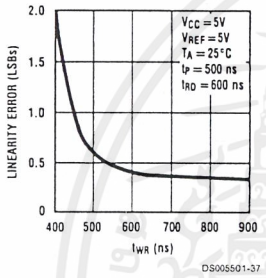
Conversion Time (RD Mode) vs Temperature



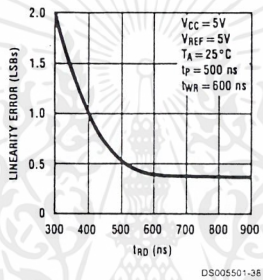
Power Supply Current vs Temperature (not including reference ladder)



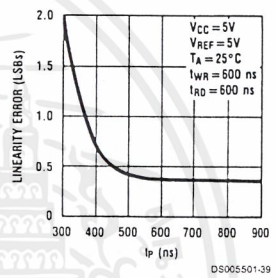
Accuracy vs tWR



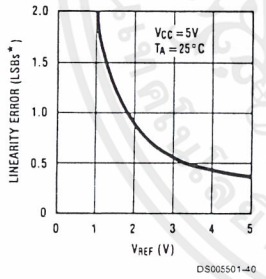
Accuracy vs tRD



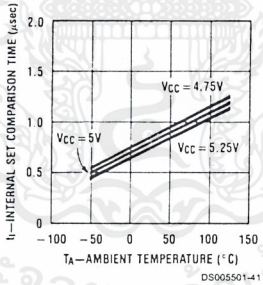
Accuracy vs tp



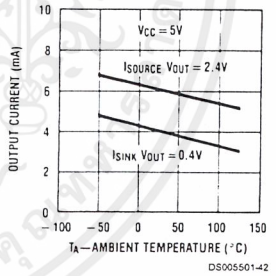
Accuracy vs VREF [VREF = VREF(+) - VREF(-)]



tI, Internal Time Delay vs Temperature



Output Current vs Temperature



$$*1 \text{ LSB} = \frac{V_{REF}}{256}$$

Timing Diagrams (Continued)

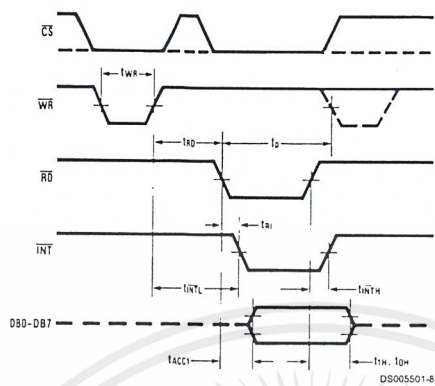


FIGURE 3. WR-RD Mode (Pin 7 is High and $t_{RD} \leq t_i$)

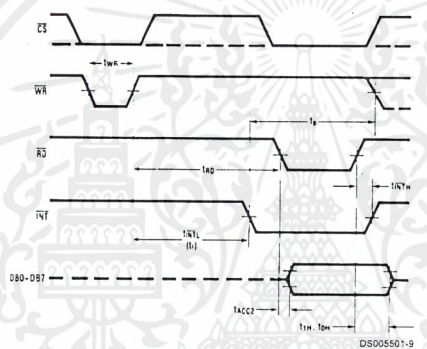


FIGURE 4. WR-RD Mode (Pin 7 is High and $t_{RD} > t_i$)

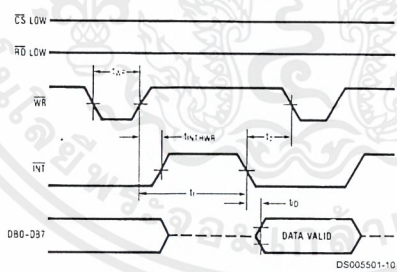
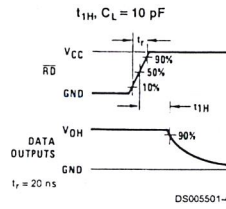
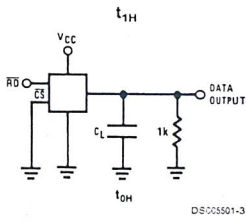


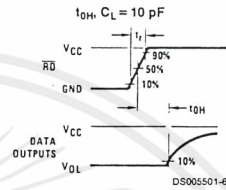
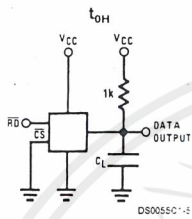
FIGURE 5. WR-RD Mode (Pin 7 is High)
Stand-Alone Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRI-STATE Test Circuits and Waveforms

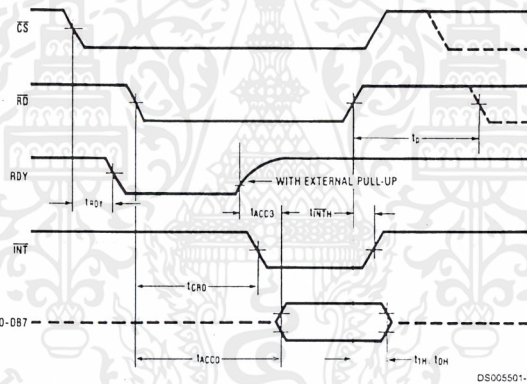


$t_r = 20 \text{ ns}$



$t_r = 20 \text{ ns}$

Timing Diagrams



Note: On power-up the state of INT can be high or low.

FIGURE 2. RD Mode (Pin 7 is Low)

AC Electrical Characteristics (Continued)

The following specifications apply for $V_{CC}=5V$, $t_F=t_r=20$ ns, $V_{REF(+)}=5V$, $V_{REF(-)}=0V$ and $T_A=25^\circ C$ unless otherwise specified.

| Parameter | Conditions | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | Units |
|---|--|-----------------|-----------------------------|-----------------------------|------------|
| t_{ACC2} , Access Time (Delay from Falling Edge of \overline{RD} to Output Valid) | Pin 7 = V_{CC} , $t_{RD} > t_i$; <i>Figure 4</i> $C_L = 15$ pF | 70 | | 120 | ns |
| | $C_L = 100$ pF | 90 | | 150 | ns |
| t_{ACC3} , Access Time (Delay from Rising Edge of RDY to Output Valid) | $R_{PULLUP} = 1k$ and $C_L = 15$ pF | 30 | | | ns |
| t_i , Internal Comparison Time | Pin 7 = V_{CC} ; <i>Figures 4, 5</i> $C_L = 50$ pF | 800 | | 1300 | ns |
| t_{IH} , t_{OH} , TRI-STATE Control (Delay from Rising Edge of \overline{RD} to Hi-Z State) | $R_L = 1k$, $C_L = 10$ pF | 100 | | 200 | ns |
| t_{INTL} , Delay from Rising Edge of \overline{WR} to Falling Edge of \overline{INT} | Pin 7 = V_{CC} , $C_L = 50$ pF $t_{RD} > t_i$; <i>Figure 4</i> $t_{RD} < t_i$; <i>Figure 3</i> | $t_{RD} + 200$ | | t_i | ns |
| | | | | $t_{RD} + 290$ | ns |
| t_{INTH} , Delay from Rising Edge of \overline{RD} to Rising Edge of \overline{INT} | <i>Figures 2, 3, 4</i> $C_L = 50$ pF | 125 | | 225 | ns |
| t_{INTHWR} , Delay from Rising Edge of \overline{WR} to Rising Edge of \overline{INT} | <i>Figure 5</i> , $C_L = 50$ pF | 175 | | 270 | ns |
| t_{RDY} , Delay from \overline{CS} to RDY | <i>Figure 2</i> , $C_L = 50$ pF, Pin 7 = 0 | 50 | | 100 | ns |
| t_{D} , Delay from \overline{INT} to Output Valid | <i>Figure 5</i> | 20 | | 50 | ns |
| t_{RI} , Delay from \overline{RD} to \overline{INT} | Pin 7 = V_{CC} , $t_{RD} < t_i$ <i>Figure 3</i> | 200 | | 290 | ns |
| t_p , Delay from End of Conversion to Next Conversion | <i>Figures 2, 3, 4, 5</i> (Note 4) See Graph | | | 500 | ns |
| Slew Rate, Tracking | | 0.1 | | | V/ μ s |
| C_{VIN} , Analog Input Capacitance | | 45 | | | pF |
| C_{OUT} , Logic Output Capacitance | | 5 | | | pF |
| C_{IN} , Logic Input Capacitance | | 5 | | | pF |

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

Note 2: All voltages are measured with respect to the GND pin, unless otherwise specified.

Note 3: Total unadjusted error includes offset, full-scale, and linearity errors.

Note 4: Accuracy may degrade if t_{WR} or t_{RD} is shorter than the minimum value specified. See Accuracy vs t_{WR} and Accuracy vs t_{RD} graphs.

Note 5: When the input voltage (V_{IN}) at any pin exceeds the power supply rails ($V_{IN} < V^-$ or $V_{IN} > V^+$) the absolute value of current at that pin should be limited to 1 mA or less. The 4 mA package input current limits the number of pins that can exceed the power supply boundaries with a 1 mA current limit to four.

Note 6: Typical values are at $25^\circ C$ and represent most likely parametric norm.

Note 7: Tested limits are guaranteed to National's AOQL (Average Outgoing Quality Level).

Note 8: Design limits are guaranteed but not 100% tested. These limits are not used to calculate outgoing quality levels.

Note 9: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

DC Electrical Characteristics

The following specifications apply for $V_{CC}=5V$, unless otherwise specified. **Boldface limits apply from T_{MIN} to T_{MAX}** ; all other limits $T_A=T_J=25^\circ C$.

| Parameter | Conditions | ADC0820CCJ | | | ADC0820BCN, ADC0820CCN ADC0820BCV, ADC0820BCWM ADC0820CCWM, ADC0820CIWM | | | Limit Units |
|--|---|--------------------|-----------------------------|-----------------------------|---|-----------------------------|-----------------------------|-------------------------------|
| | | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | |
| $V_{IN(1)}$, Logical "1" Input Voltage | $V_{CC}=5.25V$ \overline{CS} , \overline{WR} , \overline{RD} Mode | | 2.0 | | | 2.0 | 2.0 | V |
| $V_{IN(0)}$, Logical "0" Input Voltage | $V_{CC}=4.75V$ \overline{CS} , \overline{WR} , \overline{RD} Mode | | 0.8 | | | 0.8 | 0.8 | V |
| $I_{IN(1)}$, Logical "1" Input Current | $V_{IN(1)}=5V$; \overline{CS} , \overline{RD} $V_{IN(1)}=5V$; \overline{WR} $V_{IN(1)}=5V$; Mode | 0.005 0.1 50 | 1 3 200 | | 0.005 0.1 50 | 0.3 170 | 1 3 200 | μA μA μA |
| $I_{IN(0)}$, Logical "0" Input Current | $V_{IN(0)}=0V$; \overline{CS} , \overline{RD} , \overline{WR} , Mode | -0.005 | -1 | | -0.005 | | -1 | μA |
| $V_{OUT(1)}$, Logical "1" Output Voltage | $V_{CC}=4.75V$, $I_{OUT}=-360 \mu A$; DB0-DB7, \overline{OFL} , \overline{INT} $V_{CC}=4.75V$, $I_{OUT}=-10 \mu A$; DB0-DB7, \overline{OFL} , \overline{INT} | | 2.4 4.5 | | | 2.8 4.6 | 2.4 4.5 | V V |
| $V_{OUT(0)}$, Logical "0" Output Voltage | $V_{CC}=4.75V$, $I_{OUT}=1.6 mA$; DB0-DB7, \overline{OFL} , \overline{INT} , \overline{RDY} | | 0.4 | | | 0.34 | 0.4 | V |
| I_{OUT} , TRI-STATE Output Current | $V_{OUT}=5V$; DB0-DB7, \overline{RDY} $V_{OUT}=0V$; DB0-DB7, \overline{RDY} | 0.1 -0.1 | 3 -3 | | 0.1 -0.1 | 0.3 -0.3 | 3 -3 | μA μA |
| I_{SOURCE} , Output Source Current | $V_{OUT}=0V$; DB0-DB7, \overline{OFL} , \overline{INT} | -12 -9 | -6 -4.0 | | -12 -9 | -7.2 -5.3 | -6 -4.0 | mA mA |
| I_{SINK} , Output Sink Current | $V_{OUT}=5V$; DB0-DB7, \overline{OFL} , \overline{INT} , \overline{RDY} | 14 | 7 | | 14 | 8.4 | 7 | mA |
| I_{CC} , Supply Current | $\overline{CS}=\overline{WR}=\overline{RD}=0$ | 7.5 | 15 | | 7.5 | 13 | 15 | mA |

AC Electrical Characteristics

The following specifications apply for $V_{CC}=5V$, $t_r=t_f=20 ns$, $V_{REF(+)}=5V$, $V_{REF(-)}=0V$ and $T_A=25^\circ C$ unless otherwise specified.

| Parameter | Conditions | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | Units |
|---|--|---|-----------------------------|-----------------------------|---------|
| t_{CRD} , Conversion Time for RD Mode | Pin 7 = 0, Figure 2 | 1.6 | | 2.5 | μs |
| t_{ACC0} , Access Time (Delay from Falling Edge of \overline{RD} to Output Valid) | Pin 7 = 0, Figure 2 | $t_{CRD}+20$ | | $t_{CRD}+50$ | ns |
| t_{CWR-RD} , Conversion Time for WR-RD Mode | Pin 7 = V_{CC} ; $t_{WR} = 600 ns$, $t_{RD}=600 ns$; Figures 3, 4 | | | 1.52 | μs |
| t_{WR} , Write Time | Min | Pin 7 = V_{CC} ; Figures 3, 4 | | 600 | ns |
| | Max | (Note 4) See Graph | 50 | | μs |
| t_{RD} , Read Time | Min | Pin 7 = V_{CC} ; Figures 3, 4 (Note 4) See Graph | | 600 | ns |
| t_{ACC1} , Access Time (Delay from Falling Edge of \overline{RD} to Output Valid) | | Pin 7 = V_{CC} , $t_{RD} < t_r$; Figure 3 $C_L=15 pF$ | 190 | 280 | ns |
| | | $C_L=100 pF$ | 210 | 320 | ns |

Absolute Maximum Ratings (Notes 1, 2)

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

| | |
|---|--------------------------|
| Supply Voltage (V_{CC}) | 10V |
| Logic Control Inputs | -0.2V to $V_{CC} + 0.2V$ |
| Voltage at Other Inputs and Output | -0.2V to $V_{CC} + 0.2V$ |
| Storage Temperature Range | -65°C to +150°C |
| Package Dissipation at $T_A = 25^\circ\text{C}$ | 875 mW |
| Input Current at Any Pin (Note 5) | 1 mA |
| Package Input Current (Note 5) | 4 mA |
| ESD Susceptibility (Note 9) | 1200V |
| Lead Temp. (Soldering, 10 sec.) | |
| Dual-In-Line Package (plastic) | 260°C |

| | |
|--------------------------------|-------|
| Dual-In-Line Package (ceramic) | 300°C |
| Surface Mount Package | |
| Vapor Phase (60 sec.) | 215°C |
| Infrared (15 sec.) | 220°C |

Operating Ratings (Notes 1, 2)

| | |
|--------------------------|---------------------------------|
| Temperature Range | $T_{MIN} \leq T_A \leq T_{MAX}$ |
| ADC0820CCJ | -40°C $\leq T_A \leq$ +85°C |
| ADC0820CIWM | -40°C $\leq T_A \leq$ +85°C |
| ADC0820BCN, ADC0820CCN | 0°C $\leq T_A \leq$ 70°C |
| ADC0820BCV | 0°C $\leq T_A \leq$ 70°C |
| ADC0820BCWM, ADC0820CCWM | 0°C $\leq T_A \leq$ 70°C |
| V_{CC} Range | 4.5V to 8V |

Converter Characteristics

The following specifications apply for RD mode (pin 7=0), $V_{CC}=5V$, $V_{REF(+)}=5V$, and $V_{REF(-)}=GND$ unless otherwise specified. **Boldface limits apply from T_{MIN} to T_{MAX}** ; all other limits $T_A=T_J=25^\circ\text{C}$.

| Parameter | Conditions | ADC0820CCJ | | | ADC0820BCN, ADC0820CCN ADC0820BCV, ADC0820BCWM ADC0820CCWM, ADC0820CIWM | | | Limit Units |
|--------------------------------------|---|-----------------|-----------------------------|-----------------------------|---|-----------------------------|--------------------------------|-------------|
| | | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | Typ (Note 6) | Tested Limit (Note 7) | Design Limit (Note 8) | |
| | | Resolution | | 8 | | | 8 | |
| Total Unadjusted Error (Note 3) | ADC0820BCN, BCWM ADC0820CCJ ADC0820CCN, CCWM, CIWM, ADC0820CCMSA | | ± 1 | | $\pm 1/2$ | $\pm 1/2$ | LSB LSB LSB | |
| Minimum Reference Resistance | | 2.3 | 1.00 | | 2.3 | 1.2 | k Ω | |
| Maximum Reference Resistance | | 2.3 | 6 | | 2.3 | 5.3 | k Ω | |
| Maximum $V_{REF(+)}$ Input Voltage | | | V_{CC} | | V_{CC} | V_{CC} | V | |
| Minimum $V_{REF(-)}$ Input Voltage | | | GND | | GND | GND | V | |
| Minimum $V_{REF(+)}$ Input Voltage | | | $V_{REF(-)}$ | | $V_{REF(-)}$ | $V_{REF(-)}$ | V | |
| Maximum $V_{REF(-)}$ Input Voltage | | | $V_{REF(+)}$ | | $V_{REF(+)}$ | $V_{REF(+)}$ | V | |
| Maximum V_{IN} Input Voltage | | | $V_{CC}+0.1$ | | $V_{CC}+0.1$ | $V_{CC}+0.1$ | V | |
| Minimum V_{IN} Input Voltage | | | GND-0.1 | | GND-0.1 | GND-0.1 | V | |
| Maximum Analog Input Leakage Current | $\overline{CS} = V_{CC}$ $V_{IN} = V_{CC}$ $V_{IN} = GND$ | | 3 | | 0.3 | 3 | μA μA | |
| Power Supply Sensitivity | $V_{CC} = 5V \pm 5\%$ | $\pm 1/16$ | $\pm 1/4$ | | $\pm 1/16$ | $\pm 1/4$ | LSB | |

Connection and Functional Diagrams (Continued)

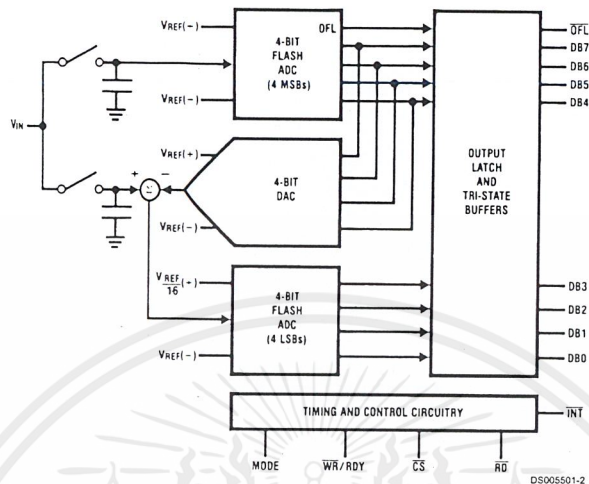


FIGURE 1.

Ordering Information

| Part Number | Total Unadjusted Error | Package | Temperature Range |
|-------------|------------------------|------------------------------|-------------------|
| ADC0820BCV | $\pm \frac{1}{2}$ LSB | V20A—Molded Chip Carrier | 0°C to +70°C |
| ADC0820BCWM | | M20B—Wide Body Small Outline | 0°C to +70°C |
| ADC0820BCN | | N20A—Molded DIP | 0°C to +70°C |
| ADC0820CCJ | ± 1 LSB | J20A—Cerdip | -40°C to +85°C |
| ADC0820CCWM | | M20B—Wide Body Small Outline | 0°C to +70°C |
| ADC0820CIWM | | M20B—Wide Body Small Outline | -40°C to +85°C |
| ADC0820CCN | | N20A—Molded DIP | 0°C to +70°C |

บรรณานุกรม

- กฤษดา ใจเย็น และคณะ. เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่าน
พอร์ตขนาน. กรุงเทพฯ : บริษัทอิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- ปรเมษฐ์ ประณยานันท์ และปิยพงศ์ เผ่าวิช. คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์
MCS 51. กรุงเทพฯ : ห้างหุ้นส่วนจำกัด.เอช.เอ็น.การพิมพ์ 2536
- สมพัฒน์ รุ่งตะวันเรืองศรี. เรียนรู้คอมพิวเตอร์กราฟิก 2มิติด้วยภาษาซี. กรุงเทพฯ : บริษัท ซีเอ็ด
ยูเคชั่นจำกัด
- ธันวา ศรีประโมง. การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม. พิมพ์ครั้งที่ 4. กรุงเทพฯ : โครง
การตำรามหาวิทยาลัยมหานคร.
- สายชล เมฆฉาย และสุรศักดิ์ กิจมงคลชัย. "เครื่องวัดและบันทึกสัญญาณโดยใช้ไมโครคอมพิวเตอร์" ปริญญาานิพนธ์อุตสาหกรรมศาสตร์บัณฑิต สาขาเทคโนโลยีอิเล็กทรอนิกส์ คณะ
วิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537

ประวัติผู้แต่ง



| | |
|------------------------------------|---|
| ชื่อผู้ทำปฏิญานិพนธ์ | นายเจษฎา ถวิล |
| วันเดือนปีเกิด | 7 พฤษภาคม พ.ศ. 2521 |
| สถานที่เกิด | จ. กรุงเทพฯ |
| ภูมิลำเนาเดิม | 241/14 แขวงสามเสนใน เขตพญาไท กรุงเทพฯ |
| ที่อยู่ปัจจุบัน | 241/14 แขวงสามเสนใน เขตพญาไท กรุงเทพฯ |
| โทรศัพท์ | 02-2782856 |
| ประวัติการศึกษา | |
| ประถมศึกษา | โรงเรียนพญาไท |
| มัธยมศึกษาตอนต้น | โรงเรียนวัดสระเกษ |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | โรงเรียนเทคโนโลยีสยาม |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | โรงเรียนเทคโนโลยีสยาม |
| ปริญญาตรี | สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิสุทกรรม คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับรางวัล | - |
| ทุนการศึกษา | - |
| คติพจน์ | ทำให้ดีที่สุด |

ประวัติผู้แต่ง



| | |
|------------------------------------|--|
| ชื่อผู้ทำปฏิญานិพนธ์ | นายพงศ์ศักดิ์ สิงห์ศิริสังกุล |
| วันเดือนปีเกิด | 31 ธันวาคม พ.ศ. 2522 |
| สถานที่เกิด | จังหวัดชลบุรี |
| ภูมิลำเนาเดิม | 193 หมู่ 7 ต.หนองอิรุณ อ.บ้านบึง จ.ชลบุรี 20220 |
| ที่อยู่ปัจจุบัน | 312/45 หมู่ 1 เขตลาดกระบัง กรุงเทพฯ 10520 |
| โทรศัพท์ | 02-3273013 |
| ประวัติการศึกษา | |
| ประถมศึกษา | โรงเรียนบ้านป่อทรางทอง |
| มัธยมศึกษาตอนต้น | โรงเรียนบ้านบึงมณูญวิทยาการ |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | วิทยาลัยเทคนิคชลบุรี |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคชลบุรี |
| ปริญญาตรี | สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับรางวัล | - |
| ทุนการศึกษา | - |
| คติพจน์ | รวดเร็วและรอบคอบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



| | |
|------------------------------------|--|
| ชื่อผู้ทำปฏิญานิพนธ์ | นายวินัย คอกคง |
| วันเดือนปีเกิด | 17 กรกฎาคม พ.ศ. 2521 |
| สถานที่เกิด | จังหวัดอุบลราชธานี |
| ภูมิลำเนาเดิม | 46 หมู่ 7 ต. หนองบ่อ อ. เมือง จ. อุบลราชธานี 34000 |
| ที่อยู่ปัจจุบัน | 397/1 หมู่ 1 เขตลาดกระบัง กรุงเทพฯ 10520 |
| โทรศัพท์ | 02-3268456 ต่อ ห้อง 18 |
| ประวัติการศึกษา | |
| ประถมศึกษา | โรงเรียนบ้านโพงาม |
| มัธยมศึกษาตอนต้น | โรงเรียนหนองบ่อสามัคคีวิทยา |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | วิทยาลัยเทคนิคอุบลราชธานี |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคอุบลราชธานี |
| ปริญญาตรี | สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับรางวัล | - |
| ทุนการศึกษา | - |
| คติพจน์ | ไม่มีคำว่าสายสำหรับการเริ่มต้น |

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์

นายสิทธิโรช โดคเครีอ

วันเดือนปีเกิด

25 พฤษภาคม พ.ศ. 2521

สถานที่เกิด

จังหวัดราชบุรี

ภูมิลำเนาเดิม

23 หมู่ 11 ต. เบิกไพร อ.บ้านโป่ง จ.ราชบุรี 70110

ที่อยู่ปัจจุบัน

378/1 หมู่ 1 เขต ลาดกระบัง กรุงเทพฯ 10520

โทรศัพท์

02-3268326

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนวัดหุบกระติง

มัธยมศึกษาตอนต้น

โรงเรียนรัตนราษฎร์บำรุง

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคราชบุรี

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

วิทยาลัยเทคนิคราชบุรี

ปริญญาตรี

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล

-

ทุนการศึกษา

-

คติพจน์