

คอมพิวเตอร์ออสซิลโลสโคป
COMPUTER OSCILLOSCOPE



โดย

นายบุญนันท์ เกียงเอีย

นายอดงกรณ์ ศศิธรวรรณกุล

ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 36891
วัน, เดือน, ปี..... 29 ส.ค. 2543

ขอสงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ได้เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ออสซิลโลสโคป
COMPUTER OSCILLOSCOPE

โดย

นายบุญอนันต์ เกียงเอี้ย 40013176

นายอลงกรณ์ ศศิธรวรรณกุล 40013199



ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ออสซิลโลสโคป

COMPUTER OSCILLOSCOPE

1. นาย บุญอนันต์ เกียงเอี้ย รหัส 40013176
2. นายอลงกรณ์ ศศิธรวรรณกุล รหัส 40013199

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2542

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง คอมพิวเตอร์ออสซิลโลสโคป

ผู้จัดทำ

1. นายบุญอนันต์ เกียงเอีย รหัส 40013176
2. นายอลงกรณ์ ศศิธรวรรณกุล รหัส 40013199

.....อาจารย์ที่ปรึกษา
(รศ.ดร.มนัส สังวรศิลป์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

คอมพิวเตอร์ออสซิลโลสโคปที่นำเสนอนี้เป็นการนำคอมพิวเตอร์มาประยุกต์ใช้งานแทนเตอเรจออสซิลโลสโคปซึ่งใช้วัดและบันทึกรูปสัญญาณต่างๆ ในการนำสัญญาณเข้าสู่คอมพิวเตอร์จะใช้ตัวกลางซึ่งทำหน้าที่แปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัลก่อน จากนั้นจึงนำสัญญาณดิจิทัลที่ได้ไปทำการประมวลผลและแสดงผลออกทางจอภาพ ประโยชน์ที่ได้จากการนำคอมพิวเตอร์มาประยุกต์ใช้งานในลักษณะนี้คือการประหยัดต้นทุนในการซื้อเตอเรจออสซิลโลสโคปซึ่งมีราคาแพงได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Computer Oscilloscope, presented in this project, uses the computer to measure and record the signal. In sending input signal to the computer, it uses a media for converting the analog signal to digital signal. Afterward, the digital signal is processed and displayed to a monitor. The benefit from this project is saving a cost, spend for the expensive Storage Oscilloscope.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	X
บทที่ 1 บทนำ	1
บทที่ 2 Data Acquisition and Conversion	2
2.1 ทฤษฎีการ Sampling	3
2.2 Analog to Digital Converter	7
บทที่ 3 หลักการของดิจิตอลสโตเรจอสซิลโลสโคป	10
3.1 การทำงานของดิจิตอลสโตเรจอสซิลโลสโคป (DSO)	10
3.2 ขอบเขตการวัด	11
3.3 การสุ่มตัวอย่างของดิจิตอลสโตเรจอสซิลโลสโคป	18
3.3.1 การสุ่มตัวอย่างแบบเวลาจริง (Real Time or One-shot)	18
3.3.2 การสุ่มตัวอย่างแบบเวลาเสมือน (Equivalent Time)	18
3.4 ขั้นตอนการสุ่มตัวอย่าง	21
3.5 การเก็บข้อมูลของดิจิตอลสโตเรจอสซิลโลสโคป	22
3.6 ปัญหาพื้นฐานของดิจิตอลสโตเรจอสซิลโลสโคป	22
3.7 ความละเอียดถูกต้องแม่นยำ	23
3.8 สัญญาณชั่วขณะ (Transient Capture)	24
3.9 แบนวิธและอัตราการสุ่มตัวอย่าง (Bandwidth and Sample rate)	26
3.10 ประโยชน์ของหน่วยความจำที่ยาวในดิจิตอลสโตเรจอสซิลโลสโคป	27
3.10.1 การเก็บรายละเอียด	28
3.10.2 การจับสัญญาณกิลิทาร์ดาว	29
3.10.3 การแก้ปัญหาเรื่องความถี่และเวลา	29
3.10.4 ความเชื่อถือได้ในการจับสัญญาณที่ไม่สามารถทำนายได้	29
3.10.5 ไม่มีการสูญเสียเวลาระหว่างการรับข้อมูล	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 4 ทฤษฎีฟาสต์ฟูเรียร์ทรานฟอร์ม (FFT)	31
4.1 ทฤษฎีดีสครีตฟูเรียร์ทรานฟอร์ม (Discrete Fourier Transform)	31
4.2 ทฤษฎีฟาสต์ฟูเรียร์ทรานฟอร์มและวิธีการแบบบัคเตอร์ฟลาย	32
4.3 การคำนวณโดยใช้คอมพิวเตอร์	35
4.4 การใช้วินโดว์	42
4.5 กราฟการแปลงฟูเรียร์แบบเร็ว	45
บทที่ 5 สัญญาณต่างๆบนสล็อตของ IBM/PC	46
5.1 รายละเอียดเกี่ยวกับสัญญาณต่างๆ	46
5.2 บัสของแหล่งจ่ายไฟของระบบ	54
บทที่ 6 หลักการออกแบบวงจรและโปรแกรมควบคุมการทำงาน	55
6.1 วงจรลดทอนสัญญาณ (Attenuator)	56
6.2 วงจรบัฟเฟอร์ (Buffer)	57
6.3 วงจรขยายสัญญาณแนวตั้ง (Vertical Amplifier)	58
6.4 วงจรปรับระดับสัญญาณ (DC Adjust)	59
6.5 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter)	60
6.6 วงจรกำเนิดสัญญาณนาฬิกาหลายความถี่ (Multi Frequency Clock Generator)	61
6.7 วงจรถอดรหัสตำแหน่งพอร์ทและวงจรเชื่อมต่อ	63
6.8 วงจรนับตำแหน่งหน่วยความจำ	64
6.9 วงจรหน่วยความจำ	65
6.10 โปรแกรมควบคุมการทำงาน	66
6.11 หลักการเขียน โปรแกรมควบคุมการทำงาน	67
6.12 การเขียนโปรแกรม Interface กับ ISA-Bus	68
6.13 การควบคุมและการใช้งาน	70
บทที่ 7 การทดลอง	72
7.1 การทดลองวงจรลดทอนสัญญาณ	72
7.2 การทดลองวงจรบัฟเฟอร์และวงจรสวิตช์เลือกสัญญาณ	73
7.3 การทดลองวงจรขยายสัญญาณ	73
7.4 การทดลองวงจรกำเนิดสัญญาณนาฬิกาหลายความถี่	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
7.5 การทดลองวงจรถอดรหัสตำแหน่งพอร์ท	75
7.6 การทดลองวงจรนับตำแหน่งหน่วยความจำ	75
7.7 การทดลองวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	76
7.8 การทดลองวัดและวิเคราะห์สัญญาณ	77
7.8.1 ทดลองวัดสัญญาณที่ความถี่ต่างๆ	78
- ผลการวัดสัญญาณชาวน์	78
- ผลการวัดสัญญาณสี่เหลี่ยม	82
- ผลการวัดสัญญาณสามเหลี่ยม	85
7.8.2 ทดลองวิเคราะห์สเปกตรัมของสัญญาณที่ความถี่ต่างๆ	88
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณชาวน์	88
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณสี่เหลี่ยม	92
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณสามเหลี่ยม	96
บทที่ 8 สรุปผลการทดลอง	99
8.1 บทสรุปและวิจารณ์	99
8.2 ปัญหาที่พบในการทำโครงการ	99
8.3 การแก้ไขปัญหา	100
8.4 ประโยชน์ที่ได้รับจากการทำโครงการ	100
8.5 แนวทางในการพัฒนาต่อ	101
8.6 คุณสมบัติของโครงการนี้	101
ภาคผนวก ก. โปรแกรม Digital Storage Oscilloscope	
ภาคผนวก ข. DataSheet	
หนังสืออ้างอิง	

สารบัญรูป

	หน้า
รูปที่ 2.1 ระบบควบคุมที่มีการประมวลผลข้อมูลทางดิจิทัล	2
รูปที่ 2.2 วิธีการพื้นฐานของ ADC	3
รูปที่ 2.3 ทราานเฟอร์ฟังก์ชันของคอมพาราเตอร์	3
รูปที่ 2.4 แสดงการผิดพลาดจากการวัดใน Aperture Time	4
รูปที่ 2.5 แสดง Spectrum ของสัญญาณอนาลอกที่จะถูกสุ่ม	5
รูปที่ 2.6 หลังการสุ่มเกิด Frequencu Folding	5
รูปที่ 2.7 การเกิด Alias Frequency จากการสุ่มด้วยความถี่ที่ต่ำกว่า 2 เท่าของสัญญาณ อนาลอก	6
รูปที่ 2.8 แสดงการต่อวงจร Parallel Comparator A/D Convertor	7
รูปที่ 2.10 วงจรเปลี่ยนสัญญาณ เหนือคี่ แบบสโลปเดี่ยว	8
รูปที่ 3.1 แสดงบล็อกไดอะแกรมของดิจิทัลสตอเรจอสซิลโลสโคป (DSO)	11
รูปที่ 3.2 แสดงลักษณะของการเกิดเอเลียสซิ่ง	13
รูปที่ 3.3 ทฤษฎีการสอดแทรก	14
รูปที่ 3.4 แสดงการจัดช่วงขอบขาขึ้นซึ่งค่าคลาดเคลื่อนที่เกิดขึ้นอยู่กับตำแหน่งในการสุ่ม	17
รูปที่ 3.5 เทคนิคการสุ่มตัวอย่างแบบต่างๆของดิจิทัลสตอเรจอสซิลโลสโคป	19
รูปที่ 3.6 แสดงให้เห็นถึงผลที่ออกมาจากการสุ่มตัวอย่างสัญญาณอินพุตแบบเป็นลำดับ	20
รูปที่ 3.7 การควอนไทซ์ซึ่งโดยใช่วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลคอนเวอร์เตอร์ เปลี่ยนแรงดันอนาลอกเป็นเลขฐานสองที่ระดับต่างๆ	21
รูปที่ 3.8 บล็อกไดอะแกรมของ RIS ดิจิไทเซอร์	25
รูปที่ 3.9 บล็อกไดอะแกรมของดิจิไทเซอร์การสุ่มตัวอย่าง	26
รูปที่ 3.10 แสดงรูปคลื่นดิจิทัลสตอเรจอสซิลโลสโคปที่มีหน่วยความจำ 2.5 M	27
รูปที่ 3.11 แสดงรูปคลื่นดิจิทัลสตอเรจอสซิลโลสโคปที่มีหน่วยความจำ 50 k	28
รูปที่ 3.12 รูปสัญญาณเบิร์สต่อเนื่อง	30
รูปที่ 3.13 แสดงสัญญาณที่มีช่วงเวลาระหว่างสัญญาณเบิร์สที่ยาวกว่า	30
รูปที่ 4.1 ดิสกรีตฟูเรียร์ทรานฟอร์ม	31
รูปที่ 4.2 (a) การหาค่า 8 จุด ฟาสต์ฟูเรียร์ทรานฟอร์ม	34
รูปที่ 4.2 (b) การหาค่า 8 จุด ฟาสต์ฟูเรียร์ทรานฟอร์ม	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 4.2 (c) หลักการคำนวณโดยใช้บัตเตอร์ฟลายเดียว	35
รูปที่ 4.3 (a) อินพุต 16 จุด ฟาสต์ฟูเรียร์	36
รูปที่ 4.3 (b) เอาท์พุทของ 16 จุด ฟาสต์ฟูเรียร์	37
รูปที่ 4.3 (c) การสุ่มข้อมูลต่างๆจากฟังก์ชัน $\cos(3t) + \sin(10t)$	38
รูปที่ 4.3 (d) ฟังก์ชันไซน์, โคไซน์และผลลัพธ์ที่ได้จากฟูเรียร์ทรานฟอร์ม	38
รูปที่ 4.3 (e) วิธีการหาฟาสต์ฟูเรียร์ทรานฟอร์ม 32 จุด	39
รูปที่ 4.3 (f) การพล็อตค่าต่างๆบนแกนความถี่	41
รูปที่ 4.4 การเกิดโหลบข้าง	42
รูปที่ 4.5 แสดงฟังก์ชันถ่วงน้ำหนักหรือวินโดว์ของ FFT	43
รูปที่ 5.1 ขาสัญญาณต่างๆของ ISA Bus	47
รูปที่ 6.1 แสดงบล็อกไดอะแกรมของดิจิตอลออสซิลโลสโคป	55
รูปที่ 6.2 วงจรลดทอนสัญญาณ	56
รูปที่ 6.3 วงจรบัฟเฟอร์	57
รูปที่ 6.4 วงจรขยาย	58
รูปที่ 6.5 วงจรระดับสัญญาณกระแสตรง	59
รูปที่ 6.6 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	60
รูปที่ 6.7 วงจรกำเนิดสัญญาณนาฬิกาหลายความถี่	61
รูปที่ 6.8 วงจรถอดรหัสตำแหน่งพอร์ท	63
รูปที่ 6.9 วงจรเชื่อมต่อพอร์ท ISA กับวงจรต่างๆ	63
รูปที่ 6.10 วงจรนับตำแหน่งหน่วยความจำ	64
รูปที่ 6.11 วงจรหน่วยความจำ	65
รูปที่ 6.12 โปรแกรมควบคุมการทำงานของการ์ด	66
รูปที่ 6.13 แสดงผังงานของโปรแกรมควบคุม	67
รูปที่ 6.14 แสดงผังงานของการควบคุมฮาร์ดแวร์ เมื่อทำการวัดสัญญาณ	70
รูปที่ 7.1 แสดง โปรแกรมหลักในการใช้งาน	77
รูปที่ 7.2 ผลการวัดสัญญาณชายน์ความถี่ 500 Hz	78
รูปที่ 7.3 ผลการวัดสัญญาณชายน์ความถี่ 4 kHz	79
รูปที่ 7.4 ผลการวัดสัญญาณชายน์ความถี่ 20 kHz	79
รูปที่ 7.5 ผลการวัดสัญญาณชายน์ความถี่ 50 kHz	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 7.6 ผลการวัดสัญญาณชายน์ความถี่ 100 kHz	80
รูปที่ 7.7 ผลการวัดสัญญาณชายน์ความถี่ 300 kHz	81
รูปที่ 7.8 ผลการวัดสัญญาณชายน์ความถี่ 500 kHz	81
รูปที่ 7.9 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 500 Hz	82
รูปที่ 7.10 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 4 kHz	83
รูปที่ 7.11 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 20 kHz	83
รูปที่ 7.12 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 50 kHz	84
รูปที่ 7.13 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 100 kHz	84
รูปที่ 7.14 ผลการวัดสัญญาณสามเหลี่ยมความถี่ 500 Hz	85
รูปที่ 7.15 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 4 kHz	86
รูปที่ 7.16 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 20 kHz	86
รูปที่ 7.17 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 50 kHz	87
รูปที่ 7.18 ผลการวัดสัญญาณสี่เหลี่ยมความถี่ 100 kHz	87
รูปที่ 7.19 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 4 kHz	88
รูปที่ 7.20 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 20 kHz	89
รูปที่ 7.21 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 50 kHz	89
รูปที่ 7.22 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 100 kHz	90
รูปที่ 7.23 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 300 kHz	90
รูปที่ 7.24 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 500 kHz	91
รูปที่ 7.25 ผลการวิเคราะห์สัญญาณชายน์ความถี่ 1 MHz	91
รูปที่ 7.26 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 4 kHz	92
รูปที่ 7.27 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 20 kHz	93
รูปที่ 7.28 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 50 kHz	93
รูปที่ 7.29 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 100 kHz	94
รูปที่ 7.30 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 300 kHz	94
รูปที่ 7.31 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 500 kHz	95
รูปที่ 7.32 ผลการวิเคราะห์สัญญาณสี่เหลี่ยมความถี่ 1 MHz	95
รูปที่ 7.33 ผลการวิเคราะห์สัญญาณสามเหลี่ยมความถี่ 4 kHz	96
รูปที่ 7.34 ผลการวิเคราะห์สัญญาณสามเหลี่ยมความถี่ 20 kHz	97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

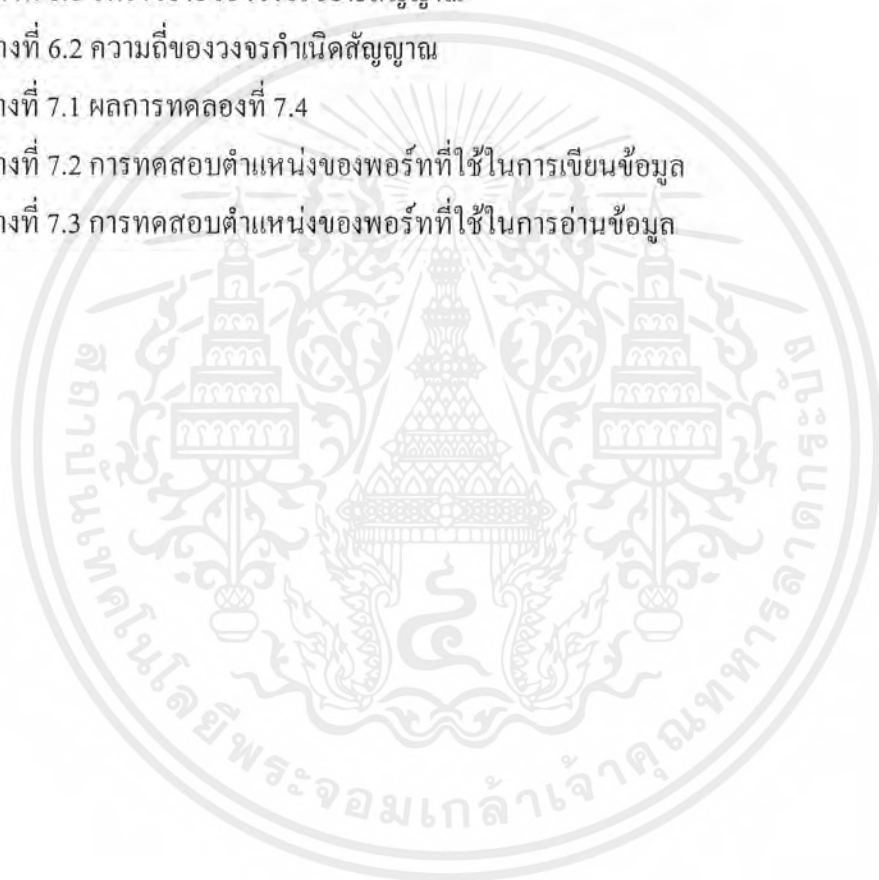
	หน้า
รูปที่ 7.35 ผลการวิเคราะห์สัญญาณสามเหลี่ยมความถี่ 50 kHz	97
รูปที่ 7.36 ผลการวิเคราะห์สัญญาณสามเหลี่ยมความถี่ 100 kHz	98



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 4.1 เปรียบเทียบจำนวนการคำนวณระหว่างคิสทริสฟูเรียร์ทรานฟอร์ม และฟาสต์ฟูเรียร์ทรานฟอร์ม	36
ตารางที่ 4.2 ข้อมูลต่างๆที่ได้จากการคำนวณ โดยวิธีฟาสต์ฟูเรียร์ทรานฟอร์ม	40
ตารางที่ 6.1 อัตรายายของวงจรมายสัญญาณ	58
ตารางที่ 6.2 ความถี่ของวงจรมายสัญญาณ	62
ตารางที่ 7.1 ผลการทดลองที่ 7.4	74
ตารางที่ 7.2 การทดสอบตำแหน่งของพอร์ทที่ใช้ในการเขียนข้อมูล	75
ตารางที่ 7.3 การทดสอบตำแหน่งของพอร์ทที่ใช้ในการอ่านข้อมูล	75



บทที่ 1

บทนำ

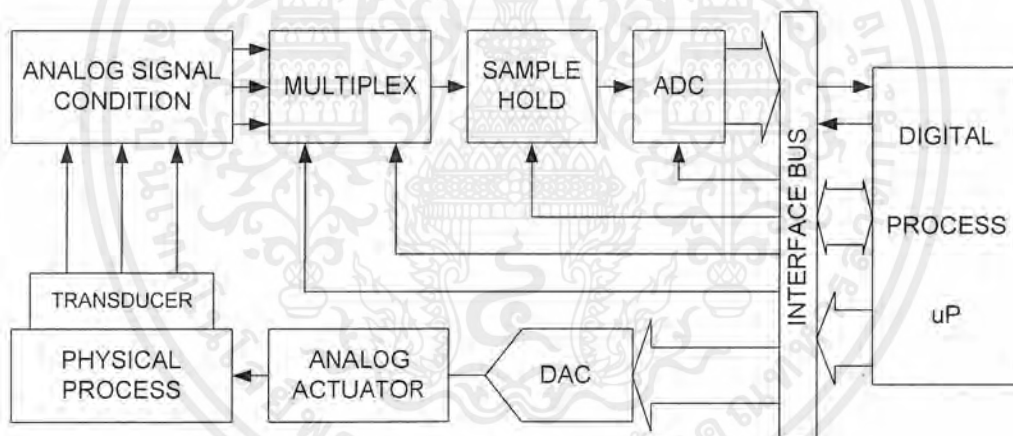
ปัจจุบันเครื่อง Oscilloscope ได้แพร่หลายในวงการอิเล็กทรอนิกส์เป็นอย่างมาก ไม่ว่าจะเป็นการวิเคราะห์ งานซ่อม หรือการเก็บข้อมูลบางอย่างไว้อ้างอิง จะยังมีอื่นๆอีกมากมายในปัจจุบัน เครื่อง Oscilloscope รุ่นใหม่ที่เก็บสัญญาณหรือบันทึกข้อมูล พร้อมกับยังมี Function พิเศษต่างๆ มากมาย ซึ่งก็จะมีค่าสูงมากและในโรงงานอุตสาหกรรมได้นำเอาคอมพิวเตอร์เข้ามาควบคุมเครื่องจักรเพื่อลดการผิดพลาด ลดเวลาการสูญเสียต่างๆ และยังสามารถเพิ่มประสิทธิภาพการทำงานได้สูงขึ้น จากข้อดีของ Oscilloscope และ คอมพิวเตอร์นั้น จึงเป็นที่เริ่มของโครงการนี้ที่จะสร้างการ์ดใช้งานร่วมกับคอมพิวเตอร์ให้เป็น Oscilloscope แสดงผลหน้าจอที่ คอมพิวเตอร์โดยได้ตัวบาง Function ของ Oscilloscope ออกไป เพราะบาง Function ไม่ได้ถูกใช้งานนำไปบันทึกสัญญาณและวิเคราะห์ เช่น นำไปบันทึกสัญญาณของเครื่องตรวจสอบ IC โดยการตรวจสอบสัญญาณบางสัญญาณว่าเครื่องตรวจสอบ IC สามารถผลิตสัญญาณโดยที่ยังอยู่ใน spec เพื่อที่จะได้ผลผลิตที่มีคุณภาพหรือป้องกันของเสีย หรือนำไปใช้ SPC (Statistic Process Control) ทำให้เกิดประโยชน์สูงสุดในการใช้คอมพิวเตอร์ ส่วนเครื่อง Oscilloscope ก็นำไปใช้อย่างอื่นๆ ซึ่งก็จะเป็นการลดค่าใช้จ่ายในการที่จะซื้อ Oscilloscope หลายตัวมาใช้งานซึ่งจะไม่คุ้ม

ฉะนั้น จึงได้มีการสร้างเครื่องมือชนิดหนึ่งขึ้นมา เรียกว่า PC BASE STORAGE Oscilloscope เพื่อแก้ไขปัญหาในจุดนี้ และมีข้อดีคือใช้งานร่วมกับเครื่องคอมพิวเตอร์ทั่วไปที่มีใช้กันอย่างแพร่หลาย เพียงแต่นำ CARD ของ HARDWARE เข้าไปเสียบ SLOT ของเครื่องคอมพิวเตอร์ และมี SOFTWARE เป็นตัวควบคุมให้เครื่องคอมพิวเตอร์ทำหน้าที่เป็น Digital Storage Oscilloscope โดยแสดงผลออกทางจอภาพ เครื่องพิมพ์ พร้อมทั้งบันทึกข้อมูลลงบนแผ่น Diskette เช่นเดียวกับเครื่องที่มีขายตามท้องตลาดทั่วไป

บทที่ 2

Data Acquisition and Conversion

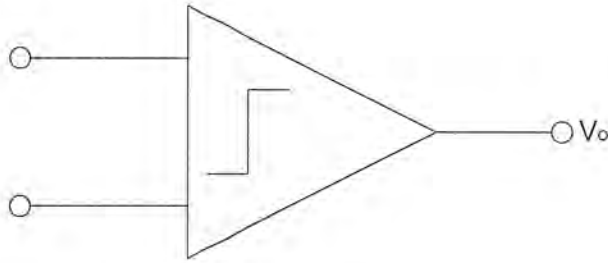
รูปร่างสัญญาณทางไฟฟ้าที่เราพบเห็นและคุ้นเคยในชีวิตประจำวันจะอยู่ในรูปแบบของสัญญาณที่ต่อเนื่องกันหรือเรียกว่าสัญญาณอนาลอก ซึ่งแต่เดิมการนำเอาสัญญาณไฟฟ้างกล่าวมาประมวลผล (Process) ใช้ในรูปแบบที่มีประโยชน์จะกระทำในแบบอนาลอกนั่นเอง แต่เมื่อมีเทคนิคการประมวลผลสัญญาณทางดิจิทัลได้รับการพัฒนาขึ้นมาเนื่องจากพบว่าในรูปแบบดิจิทัล การประมวลผล, เก็บ, สื่อสาร และแสดง กระทำได้ง่ายและมีประสิทธิภาพมากกว่า ดังนั้นการเปลี่ยนรูปแบบของสัญญาณ (Conversion) จึงได้มีความจำเป็นขึ้น จากสัญญาณอนาลอกที่มีอยู่ตามธรรมชาติ ถูกเปลี่ยนมาเป็นสัญญาณดิจิทัล โดย Analog to Digital Converter (ADC) และประมวลผลโดยตัวประมวลผลทางดิจิทัล (Digital Processor) เช่น คอมพิวเตอร์ จากนั้นจะถูกนำมาแสดง หรือ ถูกเปลี่ยนกลับมาอยู่ในรูปแบบอนาลอก และใช้งานได้ง่ายกว่าโดยใช้ Digital to Analog Converter (DAC) รูปที่ 2.1 แสดงระบบควบคุมโดยการประมวลผลข้อมูลในระบบดิจิทัล



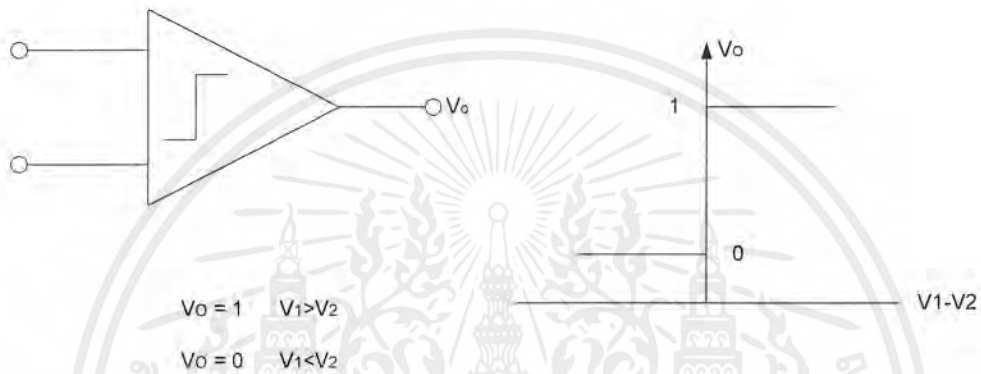
รูปที่ 2.1 ระบบควบคุมที่มีการประมวลผลข้อมูลทางดิจิทัล

วิธีการแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบง่าย ๆ แสดงในรูป 2.2 โดยใช้หลักการของวงจรคอมพาราเตอร์ แรงดันอินพุตที่ไม่ทราบค่า V_x จะต่อเข้ากับขาอินพุตขาหนึ่งของอนาลอกคอมพาราเตอร์ และแรงดันอ้างอิงที่ขนาดแปรตามเวลา V_R ต่อเข้ากับอีกอินพุตหนึ่งของคอมพาราเตอร์ ลักษณะของทรานเฟอร์ฟังก์ชันของคอมพาราเตอร์ แสดงในรูป 2.3 ถ้าแรงดันอินพุต V_1 มากกว่า V_2 แล้วแรงดันเอาต์พุตจะเป็น “1” ถ้าอินพุต V_1 น้อยกว่า V_2 แล้วเอาต์พุตจะเป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 วิธีการพื้นฐานของ ADC



รูปที่ 2.3 ทรานสเฟอ์ฟังก์ชันของคอมพาราเตอร์

วงจร Analog to Digital Converter (ADC) ที่ใช้งานทั่วไปมีหลายชนิด เช่น Counter type ADC, Tracking ADC, Integrating ADC, Successive Approximation ADC หรือ Parallel (Flash) ADC เป็นต้น

2.1 ทฤษฎีการ Sampling

ในการแปลงสัญญาณอนาลอก(Analog) เป็นดิจิตอล(Digital) นั้น ADC จะต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวขึ้นอยู่กับตัวแปร เช่น

1. ความละเอียดของการเปลี่ยนสัญญาณ
2. เทคนิคการแปลงสัญญาณ
3. ความเร็วในการทำงานของอุปกรณ์ร่วมอื่นๆ

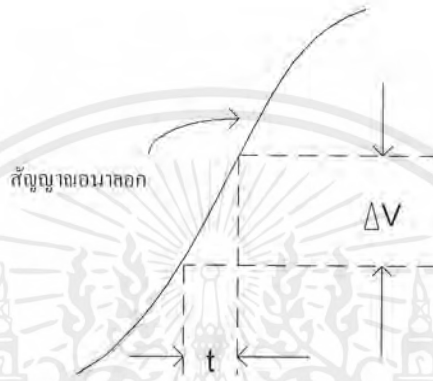
ความเร็วของการแปลงสัญญาณนี้จำเป็นสำหรับการใช้งานเฉพาะอย่างและความแม่นยำที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Aperture Time : คือช่วงเวลาในการแปลงสัญญาณ ซึ่งคำว่า Aperture Time โดยทั่วไปหมายถึง ช่วงเวลาที่เกิดความไม่แน่นอนในการวัดและผลก็คือเกิดความผิดพลาด (Error) ต่อค่าที่วัดได้

ในรูปที่ 2.4 สัญญาณอนาล็อก $V(t)$ มีอัตราการเปลี่ยนแปลง dV/dt ช่วง Aperture Time t_a ดังนั้นช่วงการเปลี่ยนแปลงสัญญาณอนาล็อกจะเท่ากับ ΔV โดย

$$\Delta V = t_a \frac{dV(t)}{dt} \dots (1.1)$$



รูปที่ 2.4 แสดงการผิดพลาดจากการวัดใน Aperture Time

ดังนั้นหากเวลาที่ ADC ใช้ในการเปลี่ยนสัญญาณในเวลา t_a นี้รหัสของสัญญาณดิจิทัลที่ได้ อาจจะตรงกับขนาดของสัญญาณอนาล็อกค่าใดค่าหนึ่งในช่วงนี้ และส่วนอื่นๆที่เหลือคือความผิดพลาดที่เกิดขึ้น ซึ่งแน่นอนในบางครั้งเป็นไปได้ที่รหัสของสัญญาณดิจิทัลจะตรงกับค่าของสัญญาณอนาล็อกที่ถูกต้อง

Sample and Hold และ Aperture error

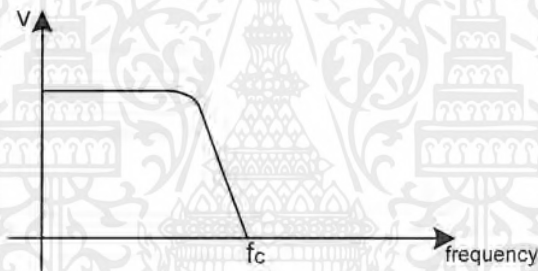
วงจร Sample and Hold จะทำการสุ่มสัญญาณอินพุต และนำสัญญาณนั้นมาเก็บ (Hold) ไว้ในเวลาหนึ่ง ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นในตัวเก็บประจุที่รั่วไหลต่ำต้งนั้น ในเมื่อแรงดันอินพุตสามารถคงอยู่ได้นานพอ ทำให้ ADC ไม่จำเป็นต้องมีเวลาในการแปลง (Conversion Time) อย่างรวดเร็วนัก Aperture Time ของ Sample and Hold คือเวลาตั้งแต่เริ่มสุ่มสัญญาณจนตัวเก็บประจุมีค่าแรงดันจนถึงค่าที่สุ่มซึ่งสำหรับ Sample and Hold แล้ว Aperture Time ขึ้นอยู่กับ Bandwidth และ Switching Time ของอุปกรณ์แอกทีฟที่ใช้ในวงจร ซึ่งหาและสร้างได้ง่ายและราคาถูกกว่าการสร้าง ADC ความเร็วสูง

มีปัญหาว่าอัตราการสุ่มสัญญาณนั้นควรจะมีค่าเท่าใดที่จะไม่ทำให้ข้อมูลสูญหายไปเมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเช่นเดิม อันนี้ขึ้นอยู่กับค่าความถี่ของสัญญาณอนาล็อก และทฤษฎีการสุ่มที่กล่าวไว้ว่า

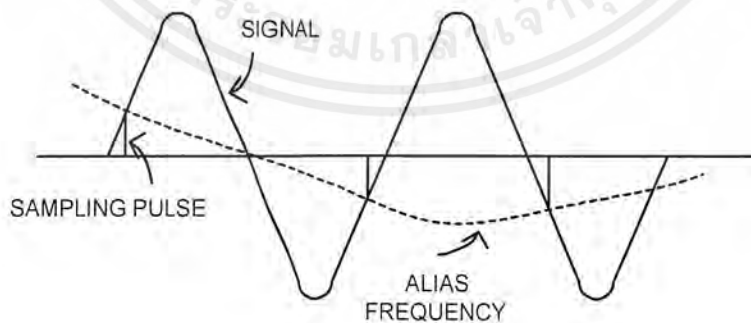
“ถ้าสัญญาณต่อเนื่องซึ่งมีความถี่และฮาร์โมนิกส์ ไม่เกิน f_c แล้วสัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาได้อย่างเดิมโดยไม่สูญเสยรายละเอียดหรือเพี้ยนไปถ้าอัตราการสุ่มไม่น้อยกว่า $2f_c$ ต่อวินาที”

Frequency folding and Aliasing

จากทฤษฎีการสุ่มสามารถอธิบายลักษณะรูปสเปกตรัม (Spectrum) ของสัญญาณในรูปที่ 2.5 แสดงให้เห็นสเปกตรัมของสัญญาณที่ถูกสุ่มซึ่งแบนด์วิดท์ไม่เกินกว่า f_c ในขณะที่สัญญาณนี้จะถูกสุ่มด้วยความถี่ f_s ขบวนการมอดูเลชัน (Modulation) จะทำให้แถบสเปกตรัมของสัญญาณสุ่มขยายกว้างออกจาก f_s เป็น $2f_s, 3f_s, \dots$ ได้ดังรูปที่ 2.6



รูปที่ 2.5 แสดง Spectrum ของสัญญาณอนาล็อกที่จะถูกสุ่ม



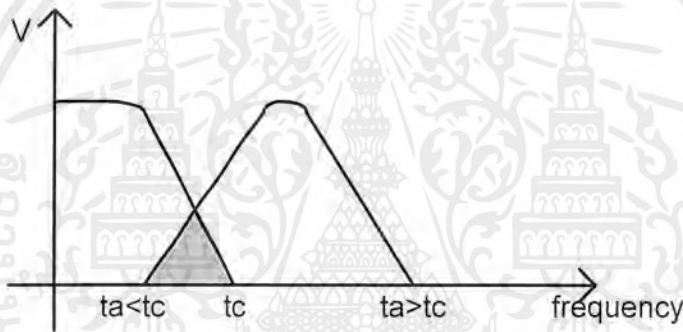
รูปที่ 2.6 หลังจากการสุ่มเกิด Frequency folding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าความถี่ของสัญญาณสุ่ม f_s ไม่สูงพอหลังจากการสุ่มสัญญาณสเปกตรัมบางส่วนของ f_s จะมาซ้อนทับกับสเปกตรัมของสัญญาณ ซึ่งเรียกว่า Frequency folding หากเป็นเช่นนี้ก็จะทำให้เกิดความเพี้ยนแก่สัญญาณอนาลอกจากการซ้อนทับกันของสเปกตรัม เมื่อสัญญาณถูกเปลี่ยนกลับให้อยู่ในรูปเดิม และถ้าเลื่อนความถี่ของการสุ่มสูงขึ้นจน โอกาสการซ้อนทับกันของสเปกตรัมหมดไป $(f_s - f_c) = f_c$ จะทำให้การเปลี่ยนกลับของสัญญาณหลังจากถูกสุ่มก็ยังคงเหมือนเดิมได้

จากที่กล่าวมาแสดงการสนับสนุนทฤษฎีการสุ่มที่ว่าให้ $f_s < 2f_c$ นั่นคือการกำจัดการซ้อนทับกันของสเปกตรัม ได้สองวิธีคือ

- 1) ใช้อัตราการสุ่มที่สูงพอ
- 2) การทำการกรองความถี่ของสัญญาณอนาลอก ก่อนการสุ่มเพื่อให้ Bandwidth ไม่เกินไปกว่า $f_s/2$



รูปที่ 2.7 การเกิด Alias Frequency จากการสุ่มด้วยความถี่ต่ำกว่า 2 เท่า ของสัญญาณอนาลอก

ในทางปฏิบัติแล้วจะยังคงเกิด Frequency folding ได้เสมอจากส่วนฮาร์โมนิคส์ของสัญญาณ รวมทั้งสเปกตรัมของสัญญาณรบกวนที่ยังคงอยู่แม้ว่าจะทำการกรองความถี่ก่อนหน้ามาแล้วก็ตามการกำจัดการซ้อนทับของสเปกตรัมนี้วิธีที่ได้ผลคือพยายามให้การสุ่มสัญญาณเป็นไปอย่างรวดเร็วมากที่สุด

ผลของการใช้อัตราการสุ่มที่ไม่เหมาะสมอีกประการหนึ่งเกิดขึ้น ดังรูปที่ 2.7 เรียกว่า Alias Frequency ซึ่งเกิดกับสัญญาณที่เปลี่ยนกลับมาเช่นเดิมหลังจากถูกสุ่มแล้ว

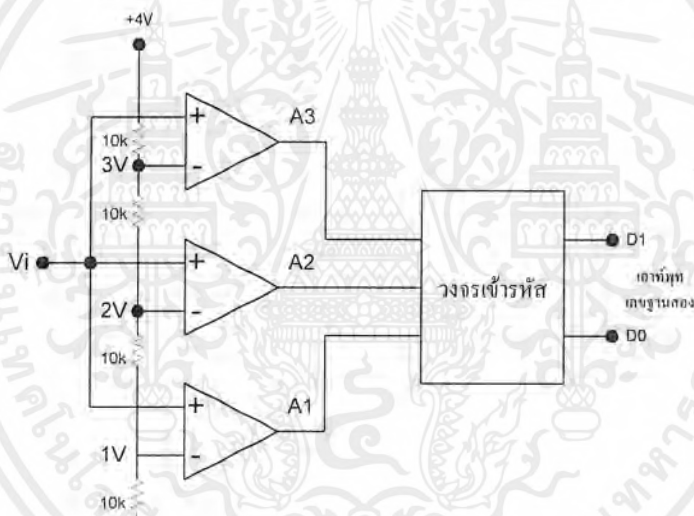
2.2 Analog to Digital Converter

วงจรเอชดีแบบใช้วงจรเปรียบเทียบ หรือแบบ “แฟลช”

(Parallel Comparator Simultaneous “Flash” A/D Converter)

สำหรับการแปลงสัญญาณที่ต้องการความเร็วสูงมากๆ เช่น การแปลงสัญญาณภาพ โทรทัศน์, เรดาห์ จำเป็นต้องใช้ ADC แบบพิเศษ ที่เรียกว่า Flash (Parallel) ADC ซึ่งจะแสดงบล็อก ไดอะแกรมดังรูปที่ 2.8

หลักการทำงานก็คือ จะใช้คอมพาราเตอร์ทำการเปรียบเทียบสัญญาณอนาล็อกอินพุตกับ แรงดันอ้างอิงที่แบ่งแรงดันให้สอดคล้องกับรหัสดิจิทัล โดยใช้ตัวต้านทาน และแปลงเอาท์พุทจาก คอมพาราเตอร์ให้ตรงกับรหัสดิจิทัล ซึ่งจะเห็นว่าอุปกรณ์ทางด้านความเร็วจะถูกจำกัดเพียง Propagation time ของคอมพาราเตอร์เท่านั้น แต่อุปสรรคที่สำคัญต่อการพัฒนาวงจรบนชิพไอซี คือ วงจรนี้ต้องการคอมพาราเตอร์ จำนวน $2^n - 1$ ตัว ซึ่งเป็นจำนวนที่มากพอสมควร



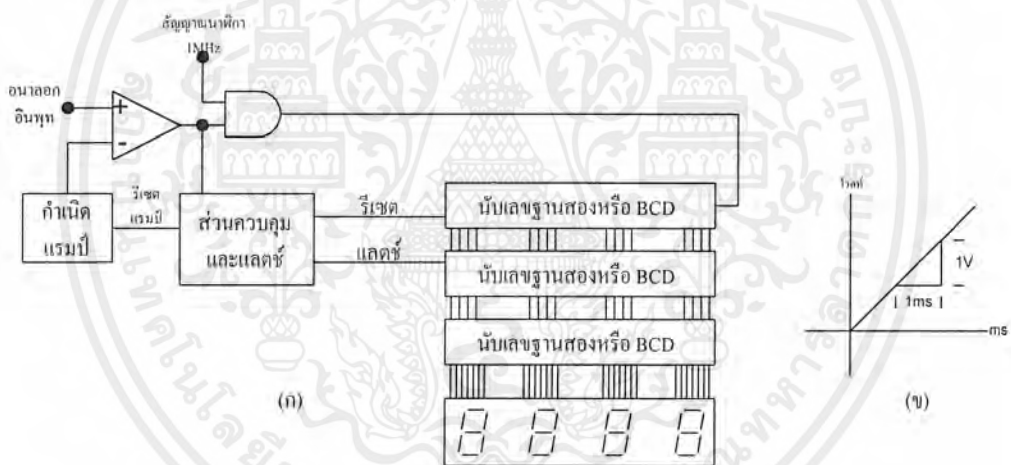
รูปที่ 2.8 แสดงการต่อวงจร Parallel Comparator A/D Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันอินพุท V_{in} (โวลต์)	เอาต์พุทของวงจรเปรียบเทียบ			เอาต์พุทเลขฐานสอง	
	A1	A2	A3	D1	D0
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

ตารางที่ 2.1 ความสัมพันธ์ระหว่างแรงดันอินพุท
ที่เป็นอนาลอกกับเอาต์พุทที่เป็นดิจิทัล

วงจรเอทูดิแบบสโลปเดียว หรือแบบแรมป์
(Single Ramp หรือ Single Slope A/D Converter)



รูปที่ 2.10 วงจรเปลี่ยนสัญญาณเอทูดิแบบสโลปเดียว

(ก) แสดงบล็อกไดอะแกรม (ข) ความชันของสัญญาณแรมป์

เมื่อเริ่มทำการเปลี่ยนสัญญาณ สัญญาณแรมป์และวงจรมับจะถูกรีเซตให้เป็น 0 แรงดันอนาลอกถูกป้อนเข้าไปยังวงจรเปรียบเทียบทางขาอินพุทแบบไม่กลับ เมื่อแรงดันอินพุทที่ขานี้เป็นบวกมากกว่าที่ขาอินพุทแบบกลับ วงจรเปรียบเทียบก็จะให้เอาต์พุทเป็นระดับ “HIGH” ทำให้แอนด์เกตปล่อยสัญญาณนาฬิกาผ่านไปยังวงจรมับได้ และทำให้เริ่มเกิดสัญญาณแรมป์

สัญญาณแรมป์มีแรงดันเป็นบวกเพิ่มขึ้นเรื่อยๆจนมากกว่าระดับแรงดันอินพุท เอาท์พุทของวงจรถ่ายเทียบก็ตกลงมาเป็นระดับ “LOW” ปิดแอนด์เกตไม่มีสัญญาณผ่านไปให้วงจรถ่ายเทียบจะหยุดนับและเก็บค่าที่วงจรถ่าย (Latch) จากนั้นจะทำการรีเซตวงจรถ่ายและวงจรถ่ายกำเนิดสัญญาณแรมป์

สมมติให้สัญญาณนาฬิกามีความถี่ 1MHz , วงจรถ่าย BCD 4 หลัก, แรงดันอินพุท V_{in} 2 โวลท์, สัญญาณแรมป์มีความชัน 1 V/ms ดังแสดงในรูปที่ 2.10

จากจุดเริ่มต้นจนถึงแรงดันสูงสุด (2 โวลท์) สัญญาณแรมป์ใช้เวลา 2 ms หลังจากนั้นจึงปิดสัญญาณนาฬิกาที่ส่งไปให้วงจรถ่าย

ในช่วง 2 ms นี้มีการส่งพัลส์ไปให้วงจรถ่ายทำการนับ 200 ลูก เอาท์พุทของวงจรถ่ายเทียบที่มีระดับ “HIGH” เป็นการส่งสัญญาณให้วงจรถ่ายส่งค่าที่นับได้ไปยังภาคแสดงผลและเติมจุดทศนิยมที่ตำแหน่งที่เหมาะสมของตัวแสดงผลให้เป็นค่า 2,000 ที่แรงดันอินพุท 2 โวลท์

วงจรถ่ายนี้เป็นหลักการเบื้องต้นของดิจิตอลโวลท์มิเตอร์ ซึ่งถ้าใช้วงจรถ่ายเลขฐานสองแทนแบบ BCD เอาท์พุทก็จะอ่านได้เป็นค่าเลขฐานสองโดยตรง

บทที่ 3

หลักการของดิจิตอลสโตเรจอสซิลโลสโคป

ดิจิตอลสโตเรจอสซิลโลสโคป (Digital Storage Oscilloscope) หรือเรียกเป็นชื่อย่อว่า DSO ได้รับการออกแบบให้มีขีดความสามารถในการตอบสนองความต้องการของผู้ใช้ ซึ่งไม่สามารถทำได้ในอนาลอกสโคปแบบเดิมๆ ที่มีอยู่ อาทิเช่น การเก็บข้อมูลของสัญญาณที่วัดได้ เพื่อการนำมาวิเคราะห์ในภายหลัง การวิเคราะห์รูปคลื่นอย่างละเอียดโดยการซูมรูปคลื่นส่วนที่ต้องการ การจับสัญญาณที่เกิดขึ้นเพียงชั่วขณะหนึ่ง การทำปริทริกเกอร์ (Pre-Trigger) และการตรวจจับสัญญาณที่มีการเปลี่ยนแปลงทางความถี่อย่างรวดเร็วหรือซ้ำเกินไป ซึ่งไม่สามารถตรวจจับได้ด้วยอนาลอกสโคปแบบปกติ

จุดสำคัญของสโคปแบบนี้อยู่ที่วิธีการในการเก็บข้อมูลและวิธีการทางดิจิตอลที่นำมาใช้ โดยปกติแล้วข้อมูลที่ถูกนำมาเก็บไว้นั้นสามารถนำไปบันทึกไว้ในเครื่องคอมพิวเตอร์ได้และยังสามารถต่อพ่วงพล็อตเตอร์หรือพรินเตอร์ได้ด้วย

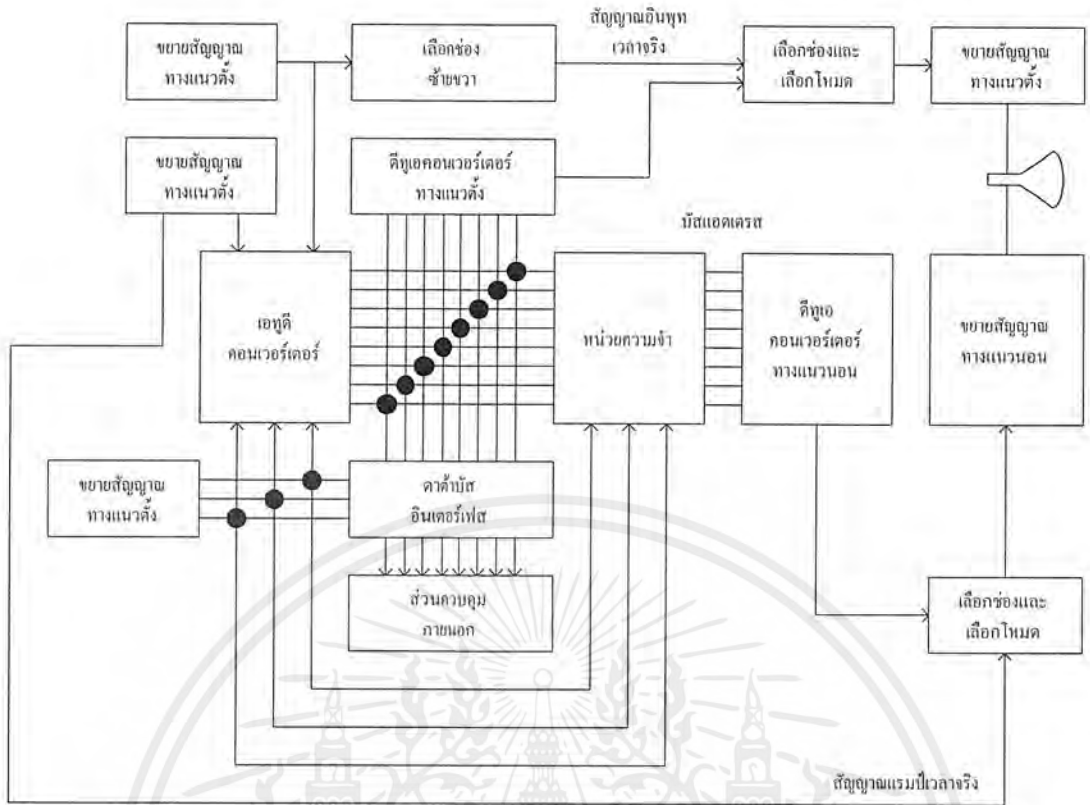
3.1 การทำงานของดิจิตอลสโตเรจอสซิลโลสโคป(DSO)

สัญญาณอนาลอกและสัญญาณดิจิตอลนั้นมีการนำมาเปรียบเทียบกันอยู่เสมอ ในเรื่องของสโคปก็เช่นเดียวกัน สโตเรจอสโคปนั้นได้ถูกประดิษฐ์คิดขึ้นมาทั้งแบบอนาลอกและแบบดิจิตอล สำหรับอนาลอกสโตเรจอสโคปนั้นจะมีวิธีการเก็บบันทึกรูปคลื่นแบบอนาลอก นั่นคือ การใช้หลอดสโตเรจ (Storage CRT) ซึ่งหลอดที่ว่านี้มีความสามารถทำให้สัญญาณที่เกิดขึ้นเพียงชั่วขณะแสดงผลบนจอได้เป็นเวลา 1 ชั่วโมงหรือมากกว่า

ส่วน DSO หรือดิจิตอลสโตเรจอสโคปจะมีวิธีการเก็บบันทึกรูปคลื่นด้วยระบบดิจิตอล ข้อมูลจะถูกบันทึกไว้ในหน่วยความจำแบบดิจิตอล ซึ่งจะเห็นว่าเมื่อเทียบกับแบบดิจิตอลแล้ว แบบอนาลอกจะเก็บรูปคลื่นด้วยอุปกรณ์ แต่แบบดิจิตอลจะเก็บรูปคลื่นด้วยวงจร นี่คือนี่ที่แตกต่างกัน

ส่วน DSO จะมีกระบวนการในการทำงานอยู่ 3 ช่วงใหญ่ๆคือ ช่วงของการสุ่มตัวอย่างและการแปลงสัญญาณเป็นดิจิตอล, ช่วงของการเก็บข้อมูล และสุดท้ายก็คือ ช่วงของการแสดงผลค่าต่างๆ

เริ่มต้นด้วยการสุ่มตัวอย่างสัญญาณที่เข้ามาเพื่อให้ได้จำนวนของจุดบนสัญญาณเหล่านี้ จากนั้นจะเป็นขั้นตอนการนำค่าที่ได้จากการสุ่มตัวอย่างมาแปลงเป็นสัญญาณดิจิตอลด้วยวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter) ซึ่งหากเป็นวงจรที่ใช้สัญญาณดิจิตอล 8 บิต ก็จะได้ระดับของสัญญาณที่แปลงออกมา 2^8 หรือ 256 ระดับ



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของดิจิทัลสต่อเรจออกสซิคโลสโคป (DSO)

ขั้นตอนที่กล่าวมาจะถูกควบคุมความเร็วในการทำงานด้วยฐานเวลา (time base) เดียวกัน ซึ่งเป็นฐานเวลาแบบดิจิทัลที่มีความเที่ยงตรงสูงมาก และจะใช้เป็นเวลาอ้างอิงในการทำงานของ วงจร ซึ่งโดยปกติแล้วค่าฐานเวลาจะมีหน่วยเป็น MS/s หรือจำนวนจุดต่อวินาที ค่าที่แปลงเป็น สัญญาณดิจิทัลแล้วจะถูกเก็บไว้ในหน่วยความจำ

สำหรับการแสดงผลของดิจิทัลสต่อเรจสโคป จะอาศัยการทำงานแบบอนาลอก ดังนั้นในช่วงของการแสดงผลจึงต้องแปลงสัญญาณดิจิทัลไปเป็นสัญญาณอนาลอกเสียก่อน เพื่อให้ได้ สัญญาณควบคุมลำโพงอิเล็กทรอนิกส์ทั้งแนวตั้งและแนวนอน ในรูปที่ 3.1 เป็นรายละเอียดพื้นฐานของ ดิจิทัลสต่อเรจออกสซิคโลสโคป

3.2 ขอบเขตการวัด

ทุกสิ่งทุกอย่างย่อมมีขีดจำกัดในตัวของมันเอง เมื่อเรานำสิ่งนั้นมาใช้งานเกินความสามารถ ของมัน ผลที่ได้ก็จะไม่ถูกต้องสมบูรณ์ ดิจิทัลสต่อเรจออกสซิคโลสโคปก็เช่นเดียวกัน เมื่อดิจิตอล สต่อเรจออกสซิคโลสโคปถูกสร้างขึ้นมาจากด้วยขบวนการทางดิจิทัล ดังนั้นขีดความสามารถของมันจึง ถูกจำกัดด้วยวิธีการทางดิจิทัล ในอนาลอกสโคปหรือดิจิทัลสโคปทั่วไป จะเห็นว่าผู้ผลิตกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการวัดเป็นค่าความถี่สูงสุดที่เครื่องสามารถทำการวัดได้ เช่น 20 MHz, 60MHz, 100MHz, 150 MHz เป็นต้น ซึ่งเวลาซื้อก็ต้องไตร่ตรองพิจารณาถึงขีดจำกัดอันนี้

สำหรับดิจิตอลสโตนเรจออกสซิลโลสโคปนั้น ขอบเขตการวัดนอกจากสามารถบอกเป็นค่าความถี่สูงสุดที่เครื่องสามารถทำการวัดได้แล้ว ยังต้องบอกค่าอัตราการสุมตัวอย่างข้อมูลควบคู่กันไปด้วย โดยปกติแล้ว จะมีค่าเป็นล้านครั้งต่อวินาที (MS/s) เช่น 20 MS/s 20 MHz, 40MS/s 100 MHz, 100MS/s 100MHz, 200 MS/s 70 MHz เป็นต้น ค่าเหล่านี้จะบอกอะไรเราบ้าง

เริ่มด้วยอัตราการสุมตัวอย่างข้อมูลจะชี้ให้เห็นถึงความละเอียดและคุณภาพของภาพที่ปรากฏบนจอ ซึ่งประกอบขึ้นมาจากจุดที่ได้จากการสุมตัวอย่างบนรูปคลื่นที่วัด ถ้าจำนวนจุดที่ได้มีจำนวนมากภาพที่ได้ก็就会有ความถูกต้องสมบูรณ์มาก การสูญเสียจุดจากการสุมตัวอย่างไปเพียงเล็กน้อยในบางรูปคลื่น ก็อาจเป็นการสูญเสียรายละเอียดของรูปคลื่นนั้นไปได้ ถ้าอัตราการสุมตัวอย่างมีค่ามาก นั่นคือ ดิจิตอลสโตนเรจออกสซิลโลสโคปจะมีความละเอียดของรูปคลื่นสูงมากด้วย ส่วนความถี่ที่บอกมานั้นก็ยังเป็นขอบเขตการวัดความถี่สูงสุดของดิจิตอลสโตนเรจออกสซิลโลสโคป ในลักษณะเดียวกับอนาลอกสโคป

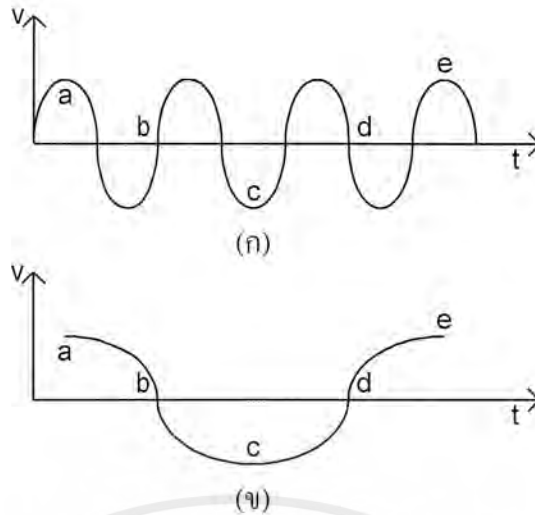
ถ้าหากเราต้องการจะรู้ว่ารูปคลื่นที่ปรากฏบนจอภาพนั้นมีความละเอียดแค่ไหน ก็สามารถทำได้ด้วยการคำนวณแบบง่ายๆตามสูตรที่ว่า

$$\text{จำนวนจุดต่อหนึ่งรูปคลื่น} = \frac{\text{จำนวนการสุมตัวอย่างของดิจิตอลสโตนเรจออกสซิลโลสโคป}}{\text{ค่าความถี่ที่ทำการวัด}}$$

ตัวอย่างเช่น ดิจิตอลสโตนเรจออกสซิลโลสโคปมีอัตราการสุมตัวอย่าง 50 MS/s ใช้วัดสัญญาณที่มีความถี่ 500kHz ดังนั้นจะได้ว่า

$$\text{จำนวนจุดต่อหนึ่งรูปคลื่น} = 50 \text{ MS/s} / 500 \text{ kHz} = 100$$

จะเห็นว่าภาพที่แสดงผลบนจอภาพนั้น ในหนึ่งรูปคลื่นจะเกิดการเรียงกันของจุดจำนวน 100 จุด แต่หากความถี่ที่ใช้วัดเปลี่ยนเป็น 10 MHz ที่อัตราการสุมตัวอย่างเดิม ความละเอียดของภาพในหนึ่งรูปคลื่นจะมีเพียง 5 จุด ต่อรูปคลื่นเท่านั้น ซึ่งจากทั้ง 2 กรณี ทำให้เราสรุปได้ว่า เมื่อนำมาวัดค่าสัญญาณที่มีความถี่สูงๆแล้ว จะให้ภาพที่มีความละเอียดชัดเจนได้น้อยลง



รูปที่ 3.2 แสดงลักษณะของการเกิดเอเลียสซิ่ง
 (ก) สัญญาณอินพุตเมื่อมีการสุ่มตัวอย่างต่ำเกินไป
 (ข) เมื่อแสดงผลเกิดเป็นสัญญาณเอเลียส

ปัญหาอย่างหนึ่งซึ่งเกิดขึ้นกับดิจิทัลสโตเรจอสซิทโลสโคป คือ การที่สัญญาณที่จะทำการวัดมีความถี่สูงมากๆ เมื่อมีการสุ่มตัวอย่างสัญญาณดังกล่าว แล้วนำมาสร้างให้ปรากฏบนจออีกครั้ง ผลที่ได้บนจอจะเป็นภาพที่ต่างไปจากสัญญาณที่วัดจริง เรียกว่า สัญญาณเอเลียส (Alias Signal) อันเกิดจากอัตราการสุ่มต่ำกว่าความถี่สัญญาณที่วัด โดยดูได้จาก ตัวอย่างในรูปที่ 3.2 ในการใช้งาน ควรพยายามหลีกเลี่ยงอย่างให้เกิดสัญญาณเอเลียสขึ้น เพราะว่าสัญญาณเอเลียสเป็นรูปคลื่นที่มีความถี่ต่างไปจากรูปคลื่นจริงอย่างมาก ซึ่งไม่สามารถนำไปวิเคราะห์ได้

การทำให้รูปคลื่นที่ปรากฏบนจอมีความสมบูรณ์ถูกต้องนั้น จำนวนจุดที่ได้จากการสุ่มตัวอย่างต้องมีมากกว่า 2 เท่าของความถี่ที่วัดนั้นใน 1 รูปคลื่น และเมื่อทำการตรวจวัดขอบขาขึ้นและขอบขาลงของพัลส์จะต้องได้จำนวนพัลส์ที่สุ่มตัวอย่างมากกว่า 10 ครั้งต่อ 1 รูปคลื่น นอกจากการสุ่มตัวอย่างที่ต้องการความเร็วในการทำงานพอสมควรแล้วยังต้องพิจารณาอีกด้วยว่าวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลนั้น สามารถทำงานได้ด้วยความเร็วขนาดไหนซึ่งเราจะพิจารณากันในหัวข้อต่อไป

สิ่งที่จะทำให้เรารู้ถึงขอบเขตการวัดของดิจิทัลสโตเรจอสซิทโลสโคป อีกอย่างหนึ่งก็คือ ค่าความถี่สูงสุดของสัญญาณที่ดิจิทัลสโตเรจอสซิทโลสโคป สามารถทำการเก็บข้อมูลได้ ซึ่งเราเรียกว่าค่า USB (Useful Storage Bandwidth) USB นี้จะมีค่าเป็นเท่าไรนั้นขึ้นอยู่กับความสามารถในการสุ่มตัวอย่างของดิจิทัลสโตเรจอสซิทโลสโคป ด้วย ซึ่งค่า USB สามารถหาได้จากสูตร

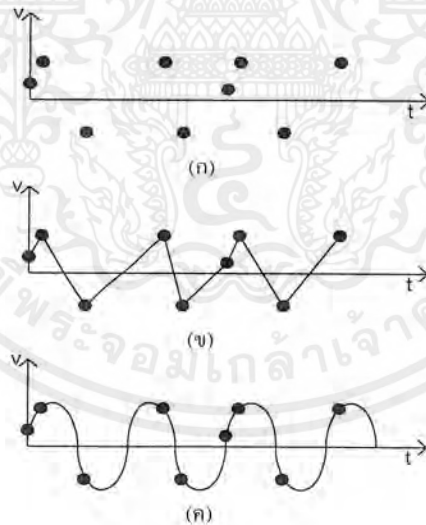
$$\text{USB} = \text{อัตราการสุ่มตัวอย่างสูงสุด (MS/s)} / 25$$

ตัวเลข 25 เป็นจำนวนจุดที่ใช้ในการสร้างภาพบนจอใน 1 รูปคลื่น ซึ่งเป็นค่าที่ใช้โดยทั่วไป

เนื่องจากดิจิทัลสตอเรจอสซิลโลสโคป ไม่สามารถแสดงรายละเอียดของสัญญาณที่มีความถี่สูงกว่าขีดจำกัดได้มากพอ ดังนั้นจึงได้มีการคิดค้นวิธีการเพิ่มรายละเอียดของสัญญาณ โดยการสร้างสัญญาณสุ่มตัวอย่างสอดแทรกลงไปในการสุ่มตัวอย่างแบบปกติ ผลที่ได้คือ จะมีจุดเสริมเข้าไปในรูปคลื่นเดิม ภาพที่ได้จะมีความต่อเนื่องสมำเสมอมากขึ้นและเรียกววิธีการนี้ว่า “ทฤษฎีการสอดแทรก” (Interpolation Methods) ซึ่งแสดงตัวอย่างการสอดแทรกสัญญาณในรูปที่ 3.3

ในรูปที่ 3.3 (ก) จะเป็นการแสดงให้เห็นถึงการสร้างภาพให้ปรากฏบนจอโดยการเรียงจุดที่ได้จากการสุ่มตัวอย่างโดยปกติจะได้ประมาณ 25 จุดต่อรูปคลื่น ส่วนรูปที่ 3.3 (ข) เป็นการลากเส้นเชื่อมต่อระหว่างจุดที่ได้มา ทำให้ได้ภาพที่เป็นเส้นตรงที่เชื่อมต่อกัน 10 เส้นต่อรูปคลื่น ซึ่งวิธีการนี้เรียกว่า การสอดแทรกแบบลิเนียร์ (Linear Interpolation) หรือ การแสดงผลแบบเวกเตอร์ (Vector Display) ดังนั้น เมื่อเป็นการแสดงผลของรูปคลื่นชาชนี้ ค่า USB จะเป็น

$$\text{USB} = \text{อัตราการสุ่มตัวอย่างสูงสุด (MS/s)} / 10$$



รูปที่ 3.3 ทฤษฎีการสอดแทรก

- (ก) การแสดงผลแบบจุด
- (ข) การสอดแทรกด้วยวิธีเวกเตอร์
- (ค) การสอดแทรกแบบชาชนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรูปที่ 3.3 (ค) เป็นการสอดแทรกโดยการนำเอาฟังก์ชันซายน์มาคำนวณลากเส้นเชื่อมต่อ ระหว่างจุด (Sine Interpolation) ซึ่งวิธีการนี้จะใช้จุดเพียง 2.5 จุดต่อรูปคลื่นเท่านั้น ในการแสดงผล ดังนั้นค่า USB จะเป็น

$$USB = \text{อัตราการสุ่มตัวอย่างสูงสุด (MS/s)} / 2.5$$

ซึ่งทำให้ค่า USB มีค่ามากขึ้นเท่ากับว่าขอบเขตของดิจิตอลสต่อเรจออกสซิดโลสโคป ก็ถูกขยายมากไปด้วย

ดิจิตอลสต่อเรจออกสซิดโลสโคปกับรูปคลื่นพัลส์

หากเราจะพูดถึงรูปคลื่นพัลส์แล้ว ส่วนที่ความสำคัญมากต่อรูปคลื่นชนิดนี้ก็คือ ช่วงเวลาขอบขาขึ้นและขอบขาลงของพัลส์ ดังนั้นเครื่องมือที่จะใช้วัดรูปคลื่นเหล่านี้จะต้องสามารถวัดช่วงขอบขาขึ้นและขอบขาลงได้โดยมีความผิดพลาดที่น้อยที่สุด ในอนาลอกสโคปสามารถคำนวณหาช่วงเวลาขอบขาขึ้น (t_r : rise time) ของพัลส์ได้ดังนี้

$$t_r (\text{ns}) = 0.35 / \text{แบนด์วิดท์ (MHz)}$$

จากสูตรดังกล่าวหากอนาลอกสโคปมีย่านการวัด 100 MHz จะได้ช่วงเวลาขอบขาขึ้นของพัลส์เท่ากับ 3.5 ns นี่คือการของอนาลอกสโคป ส่วนในดิจิตอลสโคปช่วงเวลาขอบขาขึ้นที่วัดได้จะมีการเปลี่ยนแปลงอยู่ระหว่าง 0.8 ถึง 1.6 เท่า ของช่วงเวลาการสุ่มตัวอย่างหากเราคิดเอาช่วงเวลาขอบขาขึ้นเป็น 10 ถึง 90 เปอร์เซ็นต์ของแอมพลิจูดขอบพัลส์จะได้

$$t_r = 0.8 * 2 (\text{ช่วงเวลาการสุ่มตัวอย่าง})$$

หรือ

$$t_r = 1.6 / (\text{อัตราการสุ่มตัวอย่าง})$$

เพื่อให้การคำนวณมีการคลาดเคลื่อนน้อยลง สูตรการหาช่วงเวลาขอบขาขึ้นที่นำไปใช้งานจริง ($U_{tr} = \text{Use Rise Time}$) จึงใช้ค่า 1.6 ตลอด ดังนี้

$$U_{tr} = 1.6 / (\text{อัตราการสุ่มตัวอย่าง})$$

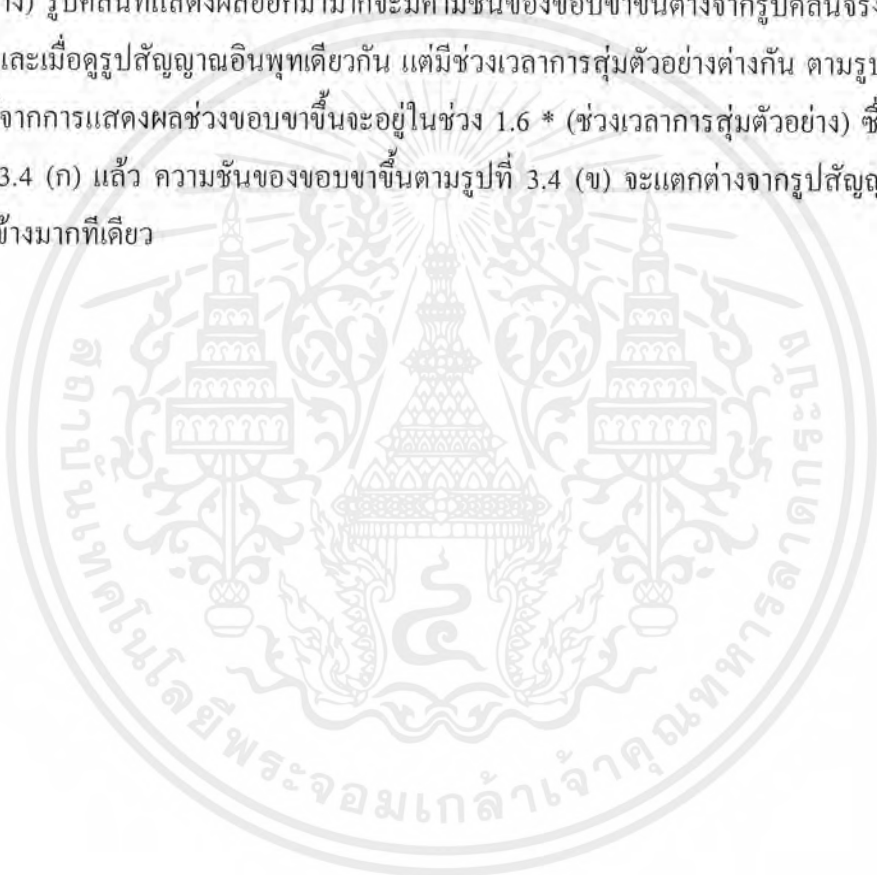
หรือ

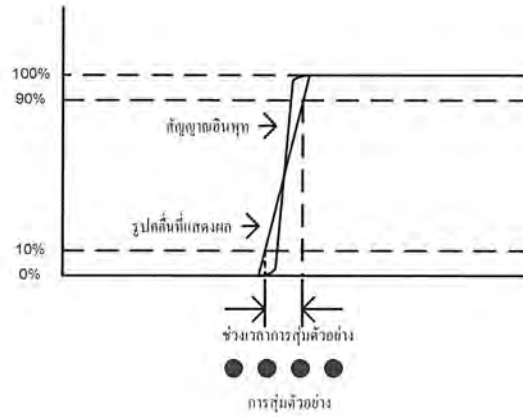
$$U_{tr} = (\text{ช่วงเวลาการสุ่มตัวอย่างต่ำสุด}) * 1.6$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

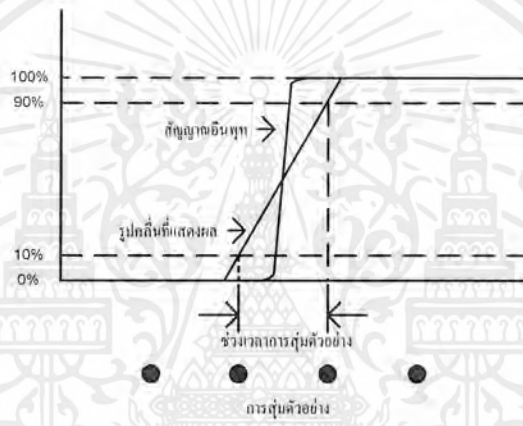
สำหรับดิจิตอลสต่อเรจอสซิลโลสโคปแล้ว ในการหาช่วงขอบขาขึ้นที่สามารถจะวัดได้จะ มีการคำนวณเช่นเดียวกับดิจิตอลสต่อเรจอสซิลโลสโคป เช่นถ้า มีอัตราการสุ่มตัวอย่างเป็น 100MS/s ก็จะมี การสุ่มตัวอย่างเป็น 0.01 μ s และ $t_{DSO} = 0.01 \mu s * 1.6 = 16 \text{ ns}$ นั่นคือ ดิจิตอลสต่อเรจอสซิลโล สโคปเครื่องนี้จะสามารถแสดงผลช่วงขอบขาขึ้นในการวัดได้ภายในเวลา 16 ns เมื่อคิดช่วงขอบขา ขึ้นเป็น 10 ถึง 90 เปอร์เซ็นต์ของแอมพลิจูด

จะเห็นได้ว่าความสามารถในการแสดงผลรูปคลื่นพัลส์นั้นจะขึ้นอยู่กับช่วงเวลาในการสุ่ม ตัวอย่างของ ดิจิตอลสต่อเรจอสซิลโลสโคปเอง รูปที่ 3.4 (ก) จะเป็นการแสดงให้เห็นถึงช่วงเวลา ขอบขาขึ้นที่อยู่ระหว่างช่วงการสุ่มตัวอย่าง โดยรูปคลื่นที่แสดงผลจะอยู่ในช่วง 0.8 * (ช่วงเวลาการ สุ่มตัวอย่าง) รูปคลื่นที่แสดงผลออกมามากจะมีความชันของขอบขาขึ้นต่างจากรูปคลื่นจริงอยู่บ้างพอ สมควร และเมื่อดูรูปสัญญาณอินพุตเดียวกัน แต่มีช่วงเวลาการสุ่มตัวอย่างต่างกัน ตามรูปที่ 3.4 (ข) ภาพที่ได้จากการแสดงผลช่วงขอบขาขึ้นจะอยู่ในช่วง 1.6 * (ช่วงเวลาการสุ่มตัวอย่าง) ซึ่งถ้าเปรียบ กับรูปที่ 3.4 (ก) แล้ว ความชันของขอบขาขึ้นตามรูปที่ 3.4 (ข) จะแตกต่างจากรูปสัญญาณอินพุต จริงค่อนข้างมากทีเดียว





(ก)



(ข)

รูปที่ 3.4 แสดงการจัดช่วงขอบขาขึ้น ซึ่งค่าคลาดเคลื่อนที่เกิดขึ้นอยู่กับตำแหน่งในการสุ่ม
 (ก) ช่วงขอบขาขึ้นอยู่ระหว่าง 0.85
 (ข) ช่วงขอบขาขึ้นอยู่ระหว่าง 0.85

จากที่ทราบในตอนต้นว่า การจัดช่วงขอบขาขึ้นและขอบขาลงของพัลส์จะต้องทำการสุ่มตัวอย่างมากกว่า 10 ครั้ง ดังนั้น เราสามารถหาช่วงเวลาที่ต้องใช้ในการวัดขอบขาขึ้นได้ โดยหารอัตราการสุ่มตัวอย่างด้วย 10 แล้วกลับเศษเป็นส่วนดังนี้

$$\text{Resolution} = 10 / (\text{อัตราการสุ่มตัวอย่าง})$$

ถ้าเป็นดิจิตอลสต่อเรจอสซิคัลสโคป ที่มีอัตราการสุ่มตัวอย่าง 100MS/s จะมีการสุ่มตัวอย่างทุกๆ 10ns (1/100 MS/s) และจะใช้การทำงานกับขอบขาขึ้นเป็นเวลา 100 ns แล้วถ้าหากว่าพัลส์ที่เข้ามามีความถี่สูงมาก เราก็ย่อมต้องการดิจิตอลสต่อเรจอสซิคัลสโคป

ที่มีความเร็วในการสุ่มตัวอย่างสูงขึ้นด้วย เช่น 500 MS/s ซึ่งจะมีการสุ่มตัวอย่างทุกๆ 2 ns และสามารถวัดช่วงขอบขาขึ้นที่อยู่ระหว่างช่วงการสุ่มตัวอย่างโดยใช้เวลา 1.17 ns

จะเห็นว่าดิจิตอลสต่อเรจอสซิคัลสโคปตอบสนองการวัดรูปคลื่นพัลส์ได้ไม่ก่อยจะดีเท่าอนาลอกสโคป และเราไม่สามารถนำค่า U_r เป็นค่าซึ่งแสดงถึงขอบเขตจำกัดของดิจิตอลสต่อเรจอสซิคัลสโคปก็ได้

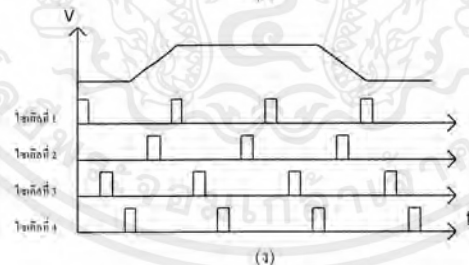
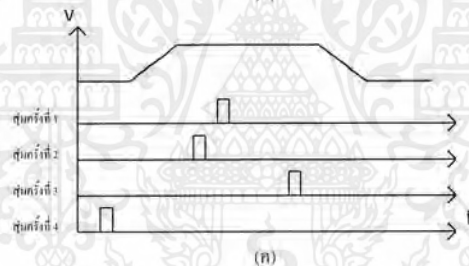
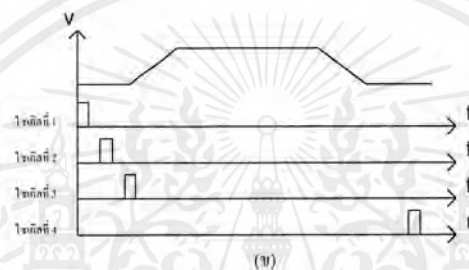
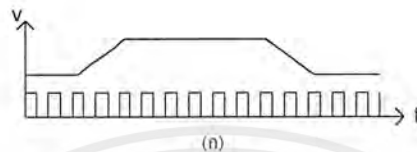
3.3 การสุ่มตัวอย่างของดิจิตอลสต่อเรจอสซิคัลสโคป

ดิจิตอลสต่อเรจอสซิคัลสโคป ใช้การสุ่มตัวอย่างด้วยกันสองแบบ คือ

3.3.1 การสุ่มตัวอย่างแบบเวลาจริง (Real Time or One-shot) จะเกิดการสุ่มตัวอย่างขึ้นทันทีที่มีการจับสัญญาณอินพุตได้ โดยผลที่ได้จะมาจากรูปคลื่นจริงๆ และจะมีความเหมือนกับรูปคลื่นจริงมาก ส่วนมากจะใช้กับสัญญาณที่เกิดขึ้นเพียงชั่วขณะหรือใช้ในการสุ่มตัวอย่างบางส่วนของสัญญาณที่ต้องการวิเคราะห์เป็นพิเศษ ดังนั้นความเร็วของการสุ่มตัวอย่างแบบเวลาจริงนี้จึงต้องสูงพอที่จะทำงานในกรณีดังกล่าวได้ การสุ่มตัวอย่างแบบนี้จะทำงานแบบผ่านครั้งเดียวคือ จะมีการสุ่มตัวอย่างจุดเดียวในแต่ละครั้งของการทำงาน จึงสามารถนำไปตรวจจับสัญญาณที่เกิดขึ้นเพียงชั่วขณะเช่น แรงไฟกระชาก, สัญญาณควบคุมเครื่องกล, สัญญาณในลักษณะทางชีววิทยา ฯลฯ ซึ่งจะปรากฏให้สามารถวัดได้ในระยะเวลาที่สั้นมาก การจะใช้วิธีสุ่มตัวอย่างจึงต้องใช้แบบที่เมื่อเริ่มสุ่มและสิ้นสุดการสุ่มต้องได้รูปสัญญาณที่เกิดขึ้น รูปที่ 3.5 (ก) จะเห็นการสุ่มตัวอย่างแบบเวลาจริงที่ใช้ในการวัดรูปคลื่นพัลส์หนึ่งลูก

3.3.2 การสุ่มตัวอย่างแบบเวลาเสมือน (Equivalent Time) การสุ่มตัวอย่างแบบนี้โดยมากจะใช้กับสัญญาณที่เกิดขึ้นซ้ำๆกันหรือใช้ตรวจจับบางส่วนของสัญญาณที่เกิดขึ้นซ้ำๆกัน การสุ่มตัวอย่างแบบเวลาเสมือนสามารถแบ่งออกเป็น 2 ชนิดย่อยๆอีก คือ การสุ่มชนิดเป็นลำดับ (Sequential Sampling) และการสุ่มชนิดไม่เป็นลำดับ (Random Sampling)

ในการสุ่มตัวอย่างชนิดลำดับจะเป็นดังรูปที่ 3.5 (ข) ซึ่งการสุ่มตัวอย่างชนิดนี้ในการทำงานหนึ่งครั้งหรือหนึ่งไซเคิลจะต้องได้จุดบนรูปคลื่น 1 จุด แล้วนำไปเก็บไว้ในหน่วยความจำ จนกระทั่งหน่วยความจำเต็ม เมื่อนำไปแสดงผลก็จะเป็นไปตามรูปที่ 3.6 ส่วนการสุ่มตัวอย่างแบบเวลาเสมือนชนิดไม่เป็นลำดับนั้นจะเป็นดังรูป 3.5 (ค) ซึ่งการสุ่มตัวอย่างแบบนี้จะใช้เมื่อมีการทำปริทริกเกอร์ (pre-trigger) เพื่อจุดรูปคลื่นที่อยู่หลังจุดทริก ซึ่งการสุ่มตัวอย่างแบบเป็นลำดับจะไม่สามารถทำได้

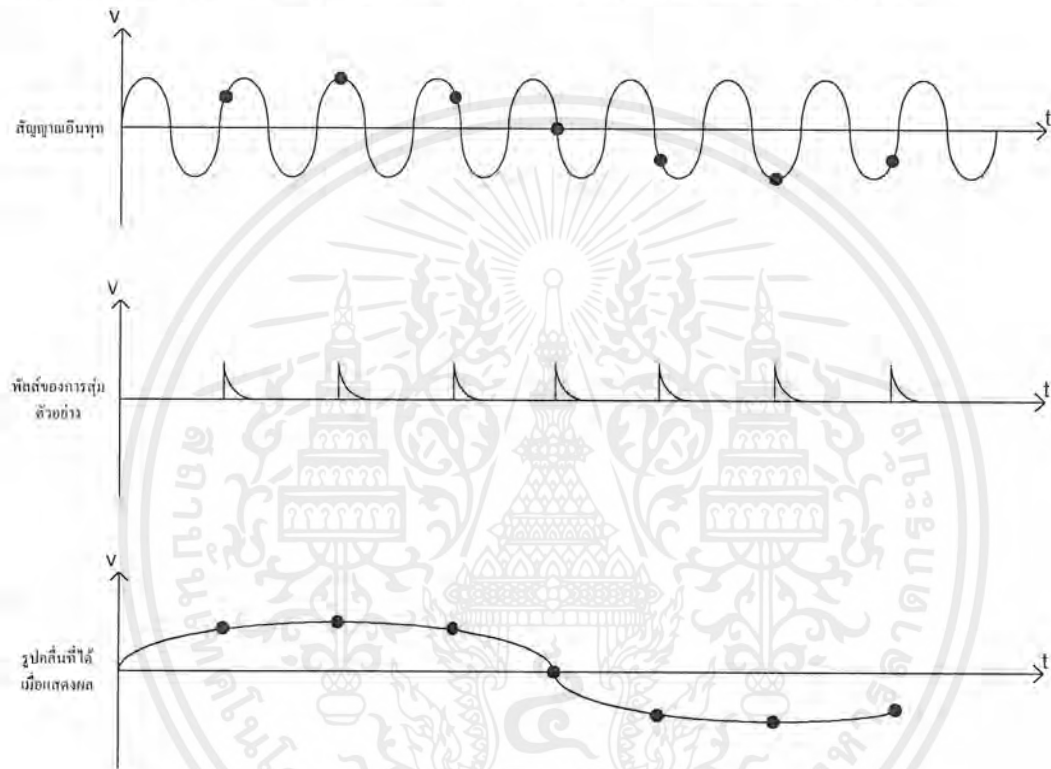


- รูปที่ 3.5 เทคนิคการสุ่มตัวอย่างแบบต่างๆของดิจิตอลสตอเรจออสซิลโลสโคป
- (ก) การสุ่ม แบบเวลาจริง
 - (ข) การสุ่มแบบเวลาเสมือนชนิดเป็นลำดับ
 - (ค) การสุ่มแบบเวลาเสมือนชนิดไม่เป็นลำดับ โดยใช้จุดเดียวในการทำงานหนึ่งครั้ง
 - (ง) การสุ่มแบบเวลาเสมือนชนิดไม่เป็นลำดับ โดยใช้หลายจุดในการทำงานหนึ่งครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทำงานหนึ่งไซเคิลจะได้จุดบนรูปคลื่น 1 จุดเช่นกัน แต่จะเป็นจุดบนส่วนใดของรูปคลื่นก็ได้ โดยจะเข้าไปเรื่อยๆจนหน่วยความจำเต็ม

ในรูปที่ 3.5 (ง) จะเป็นการสุ่มตัวอย่างแบบไม่เป็นลำดับเช่นกัน แต่จะต่างกันตรงที่ในการทำงานหนึ่งไซเคิลนั้นจะได้จุดหลายจุดบนรูปคลื่น และในการทำงานครั้งต่อไปก็จะได้จุดหลายจุด แต่จะเป็นจุดที่ไม่ซ้ำจุดเดิม (จุดใดก็ได้) การสุ่มตัวอย่างในรูปนี้จะนำไปใช้ประโยชน์เมื่อมีการวิเคราะห์รูปคลื่น โดยการนำข้อมูลที่ได้จากการสุ่มตัวอย่างแบบนี้มาเปรียบเทียบกับรูปคลื่นอื่นๆ โดยเฉพาะอย่างยิ่งการวิเคราะห์รูปคลื่นในส่วนที่เป็นการทรักในลักษณะต่างๆกัน



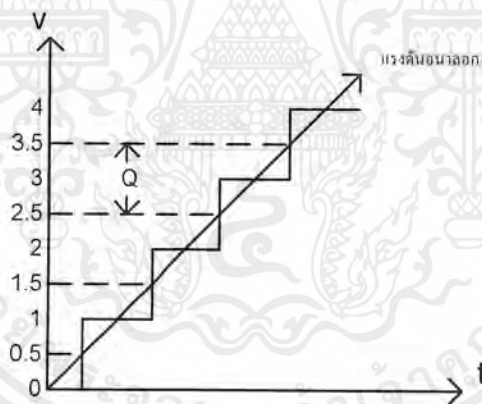
รูปที่ 3.6 แสดงให้เห็นถึงผลที่ออกมาจากการสุ่มตัวอย่างสัญญาณอินพุตแบบเป็นลำดับ

สำหรับดิจิตอลสโตเรจอสซิลโลสโคป ที่มีการใช้การแสดงผลแบบอนาลอก ซึ่งระบุความสามารถในการวัดไว้ เช่น 100 MHz ความหมายของ 100 MHz นี้ก็คือ ความสามารถในการวัดสัญญาณความถี่ 100 MHz แล้วดิจิตอลสโตเรจอสซิลโลสโคปจะแสดงผลในแบบอนาลอกได้ทันที 100 MHz เฉพาะสัญญาณที่เกิดขึ้นซ้ำๆกัน แต่หากเป็นสัญญาณชั่วขณะเกิดขึ้นแล้วใช้การสุ่มตัวอย่างแบบเวลาจริงแล้ว ดิจิตอลสโตเรจอสซิลโลสโคปนี้จะสามารถวัดสัญญาณชั่วขณะได้เพียง 10 MHz เท่านั้น

เมื่อคิดเทียบเป็นอัตราส่วนระหว่างการทำงานกับสัญญาณปกติและการทำงานกับสัญญาณชั่วขณะของดิจิตอลสต่อเรจอสซึลโลสโคป ก็จะได้เป็น 10:1 และสามารถเพิ่มเป็น 5:1 หรือ 2.5:1 ได้ ซึ่งหมายถึง ดิจิตอลสต่อเรจอสซึลโลสโคปจะต้องมีการใช้วิธีการสต่อแทรกอย่างมากเลยทีเดียว สาเหตุที่เกิดกรณีดังกล่าวขึ้นนั้นมาจากการที่ความสามารถในการสุ่มตัวอย่างถูกกำหนดด้วยสัญญาณนาฬิกาภายใน แต่สัญญาณที่เข้าวัดจะเป็นการทำงานแสดงผลแบบอนาลอก ซึ่งขอบเขตของความถี่จะกว้างมาก นี่คือการนำข้อดีของอนาลอกมาใช้

3.4 ขั้นตอนการสุ่มตัวอย่าง

เมื่อเราได้ทำการสุ่มตัวอย่างไปแล้ว ค่าที่ได้จะไม่ทำการแปลงเป็นสัญญาณดิจิตอลในทันที โดยทั้งนี้เนื่องมาจากค่าที่มาจากการสุ่มตัวอย่างนั้นเป็นค่าที่มีระดับแตกต่างกันอย่างมากมา ในทางปฏิบัติเราไม่สามารถแปลงเป็นสัญญาณดิจิตอลได้มากขนาดนั้น จึงจำเป็นต้องแบ่งค่าที่ได้มาจากการสุ่มตัวอย่างออกเป็นกลุ่มๆ เราเรียกว่า ควอนไทซิง (Quantizing) ซึ่งการควอนไทซิงนี้จะเป็นการนำค่าจากการสุ่มตัวอย่างที่มีค่าใกล้เคียงกันมากอยู่ในกลุ่มเดียวกัน ต่อจากนั้นจะนำค่าตัวเลขมาใช้แทนกลุ่มสัญญาณดังกล่าว



รูปที่ 3.7 การควอนไทซิง โดยใช้วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลคอนเวอร์เตอร์ เปลี่ยนแรงดันอนาลอกเป็นเลขฐานสองที่ระดับต่างๆ

การแบ่งกลุ่มสัญญาณจะมีลักษณะเป็นขั้นบันได ตามรูปที่ 3.7 โดยให้มีการเปลี่ยนแปลงของค่าที่ได้จากการสุ่มตัวอย่างอยู่ระหว่าง 0-4 โวลต์ และจัดแบ่งได้ 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 อีกครั้งเพื่อทำเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากวิธีการควอนไทซ์ดังกล่าวมีโอการสที่จะเกิดค่าคลาดเคลื่อนได้ อันเนื่องมาจากสัญญาณที่จะเกิดค่าคลาดเคลื่อนได้ อันเนื่องมาจากสัญญาณรบกวนที่เข้ามาในระบบทำให้ค่าที่ได้ในการจัดกลุ่มผิดไปจากความเป็นจริงได้ และเนื่องจากการควอนไทซ์ซึ่งนั้นประกอบจากวงจรดิจิทัลอิเล็กทรอนิกส์ ดังนั้นจึงเกิดความไม่แน่นอนของผลที่ได้อยู่บ้าง โดยความไม่แน่นอนเหล่านี้จะแสดงค่าบวก ลบ ของบิตสุดท้าย

เมื่อผ่าน ADC แล้ว นอกจากบิตสุดท้ายจะเป็นค่าของความไม่แน่นอนแล้ว บิตอื่นๆก็จะเป็นค่าผิดพลาดที่เกิดจากสัญญาณรบกวน ได้ด้วยเหมือนกันแต่สัญญาณดิจิทัลที่ได้ส่วนใหญ่ก็ยังมี ความเที่ยงตรงอยู่บ้าง

3.5 การเก็บข้อมูลของดิจิทัลสต่อเรจอสซิลโลสโคป

ความสามารถของดิจิทัลสต่อเรจอสซิลโลสโคป ที่สำคัญมากอย่างหนึ่งก็คือ ความสามารถในการเก็บข้อมูล ซึ่งสำหรับดิจิทัลสต่อเรจอสซิลโลสโคปแล้วหน่วยความจำที่สามารถรองรับข้อมูลได้นั้นก็มีมากมายเช่นเดียวกับคอมพิวเตอร์ทั้งหลาย ดังนั้น จึงเป็นที่แน่นอนว่าหน่วยความจำที่ใช้ก็ย่อมมี 2 แบบ คือ RAM (Random Access Memory) และ ROM (Read Only Memory)

สำหรับ RAM สามารถเก็บข้อมูลที่เป็นปัจจุบันได้อยู่ตลอดเวลา ในขณะที่ ROM นั้นจะเป็นหน่วยความจำแบบถาวร ไม่สามารถบันทึกข้อมูลเพิ่มเติมได้ ส่วนมากจะใช้ในการแสดงคุณสมบัติอุปกรณ์หรือวงจรต่างๆรวมทั้งการแก้ไขปัญหาในลักษณะต่างๆกัน

การเก็บข้อมูลนั้นยังสามารถบันทึกลงบนแผ่นดิสเก็ตต์, เทปแม่เหล็ก หรือแม้กระทั่งการพ่วงเข้ากับคอมพิวเตอร์เพื่อถ่ายโอนข้อมูลได้ด้วย

สำหรับการเก็บข้อมูลแบบ RAM นั้น ปกติจะมีทั้งแบบ CMOS, NMOS หรือแม้แต่ ECL (Emitter Coupled bipolar Logic) ซึ่งแบบต่างๆเหล่านั้นนั้นเป็นหน่วยความจำ RAM ซึ่งอยู่ภายในดิจิทัลสต่อเรจอสซิลโลสโคป และมีขนาดหน่วยความจำที่จำกัด สำหรับหน่วยความจำขนาดใหญ่และมีความยาวข้อมูลมากๆนั้นก็จะต้องมีการแก้ไขปรับปรุงให้ทันสมัยอยู่ตลอดเวลาด้วยและขนาดหน่วยความจำที่ดิจิทัลสต่อเรจอสซิลโลสโคปใช้ จะอยู่ในช่วง 4k ถึง 32k นอกจากนี้ในปัจจุบันยังมีหน่วยความจำแบบแผ่นการ์ด (RAM card) อีกด้วย

3.6 ปัญหาพื้นฐานของดิจิทัลสต่อเรจอสซิลโลสโคป

ปัญหาที่มักจะเกิดขึ้นบ่อยๆกับอนาล็อกสโคปและดิจิทัลสโคปแบบเก่าๆก็คือ การสั่นไหวของภาพทางแนวนอน (Horizontal Jitter) อาการดังกล่าวจะทำให้ภาพที่ได้จากการวัดเกิดอาการพร่ามัว เนื่องจากมีการสั่นของภาพ และจะเกิดขึ้นกับรูปคลื่นขายน้ไปจนถึงรูปคลื่นพัลส์ด้วย เราจึงคำนวณให้มีค่าเป็น $\pm 1/2$ ของช่วงเวลาระหว่างการสุ่มตัวอย่าง

3.7 ความละเอียดถูกต้องแม่นยำ

ความละเอียดถูกต้องแม่นยำเป็นสิ่งที่สำคัญอย่างมากๆสำหรับเครื่องวัดทุกชนิด DSO ก็เช่นเดียวกัน เมื่อพูดถึงความละเอียดทางด้านแนวตั้งย่อมาจะหมายถึง ความสามารถในการแยกแยะระหว่างสัญญาณซึ่งอยู่ติดๆกัน

โดยปกติแล้วความละเอียดทางด้านแนวตั้งของดิจิตอลสโคปเรจิสเตอร์ลอจิกโคปจะถูกกำหนดด้วยจำนวนเลขฐานสองที่เป็นเอาต์พุตของ ADC ตัวอย่างเช่น ADC ให้เอาต์พุตออกมา 8 บิตก็就会有ความละเอียดได้ถึง 256 (2^8) ตำแหน่ง หรือคิดเป็น 0.391% ($1/256$) เพื่อให้ชัดเจนยิ่งขึ้นนอกจากเราต้องรู้จำนวนบิตของ ADC แล้วเราจะต้องรู้ว่าถ้าเราตั้งค่าโวลต์ต่อช่องไว้ที่ค่าหนึ่งแล้วเราสามารถอ่านค่าความละเอียดออกมาเป็นเท่าไร ตัวอย่างเช่น ถ้าเราตั้งค่าโวลต์ต่อช่องไว้ที่ 50mV/div เมื่ออ่านค่าเดิมนำจอซึ่งมี 8 ช่อง ก็จะอ่านค่าได้ทั้ง 400 mV และถ้า ADC ใช้ 8 บิต เราจะได้ความละเอียดของแนวตั้งทั้งสิ้น 256 ตำแหน่ง ซึ่งแต่ละช่วงระหว่างตำแหน่งจะมีค่าเป็น

$$400 * 10^{-3} / 256 = 1.5625 \text{ mV}$$

นั่นคือ 1 ตำแหน่งจะแทนเป็นระดับแรงดันได้ 1.5625 mV ในทำนองเดียวกันหาก ADC ให้เอาต์พุตออกมา 4 บิต ความละเอียดทางด้านตั้งก็จะได้เป็น 16 ตำแหน่งและตั้งค่าโวลต์ต่อช่องไว้ที่ค่าเดิมนั้นสำหรับ 1 ตำแหน่งก็จะมีค่า 25 mV ($0.4/16$)

ดังนั้นจำนวนบิตของ ADC และมาตราส่วนทางแนวตั้งที่ตั้งค่าไว้จะเป็นตัวกำหนดความสามารถในการแยกแยะรายละเอียดต่างๆของรูปคลื่น เมื่อรายละเอียดต่างๆของการแสดงผลมีความสมบูรณ์นั้นย่อมาหมายความว่าไม่มีเครื่องมือวัดที่จะมีความถูกต้องแม่นยำกว่า

ต่อไปเราจะมาพิจารณาถึงความละเอียดทางด้านแนวนอนกันบ้างสำหรับความละเอียดทางด้านแนวนอนนั้นเป็นค่าที่เกี่ยวข้องกับคาบเวลาที่ถูกลบไว้ในหน่วยความจำแบบดิจิตอล หากสัญญาณถูกลบไว้ในข้อมูล 1024 เวิร์ด ก็จะได้ค่าความละเอียดของด้านแนวนอนเป็น $1/1024$ หรือคิดเป็น 0.098%

เราลองมาเปรียบเทียบกันดีกว่าสำหรับหัวข้อนี้ที่อนาล็อกและดิจิตอลเป็นอย่างไร ถ้าเป็นอนาล็อกสโคปความละเอียดของภาพที่ได้จะขึ้นอยู่กับพื้นที่ผิวด้านหน้าจอ CRT และขนาดรูปร่างของลำอิเล็กตรอนที่ใช้ในการเขียน รวมไปถึงความสามารถในการทำงานของวงจรขยายสัญญาณทางแนวตั้งและแนวนอนซึ่งกลายเป็นปัจจัยอันหนึ่งซึ่งกลายเป็นขีดจำกัดอย่างมากสำหรับอนาล็อกสโคปด้วย

แต่สำหรับดิจิตอลสโคปความละเอียดทางด้านตั้งจะขึ้นอยู่กับจำนวนบิตที่ ADC ใช้ และในเรื่องของความถูกต้องแม่นยำ อนาล็อกจะคล้ายกับดิจิตอลในบางกรณี ซึ่งโดยปกติค่าความคลาดเคลื่อนทางแนวตั้งจะไม่เกิน $\pm 2-4\%$ และทางแนวนอนจะไม่เกิน $\pm 1-3\%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ดีดิจิทัลออสโคปมีข้อได้เปรียบเรื่องความถูกต้องแม่นยำเพิ่มขึ้นอีกเล็กน้อย ด้วยลักษณะของภาพที่เรียกว่า frozen ที่ปรากฏบนจอ CRT และโดยการใช้ตัวชี้ค่าที่เราเรียกกันว่า cursors ซึ่งมีการชี้ค่าได้แม่นยำมากทีเดียว สิ่งเสริมเข้ามานี้ทำให้ดิจิทัลออสโคปสามารถลบจุดด้อยด้านความแม่นยำบางเรื่องลงไปได้มากทีเดียว

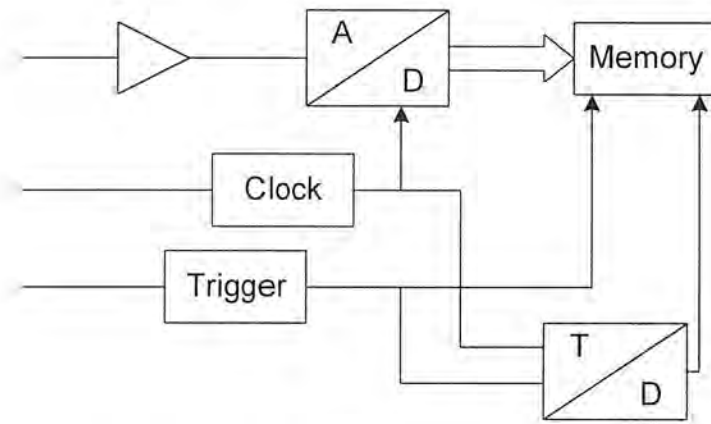
สำหรับเรื่องของฐานเวลานั้นดิจิทัลออสโคปจะไม่ใช่วงจรกำเนิดฐานเวลาแบบลิเนียร์สวีป (linear sweep) แต่จะใช้วงจรกำเนิดฐานเวลาแบบคริสตอลแทน เนื่องจากฐานเวลาที่ได้จากแบบคริสตอลจะมีค่าความคลาดเคลื่อนน้อยกว่า $\pm 0.01\%$

เมื่อมาถึงจุดนี้เราจะเห็นว่าความแม่นยำด้านแวนอนจะเกี่ยวข้องกับสัญญาณนาฬิกา, ขนาดของหน่วยความจำ, และตัวชี้ค่า จะเห็นว่าความแม่นยำด้านแวนอนจะต้องอาศัยองค์ประกอบเยอะมากทีเดียว ดังนั้นเพื่อให้เกิดค่าความแม่นยำทางแวนอนสูงสุดสำหรับดิจิทัลออสโคป เราจะต้องทำการวัดสัญญาณนั้นๆ ซ้ำๆ กัน และต้องลดทอนค่าคลาดเคลื่อนที่เกิดจากการเลือกใช้ตัวชี้ค่าไม่เหมาะสมลงด้วย

3.8 สัญญาณชั่วขณะ (Transient Capture)

สโคปแบบอนาล็อกส่วนใหญ่ไม่สามารถแสดงค่าชั่วขณะที่เกิดขึ้นได้ ในทางตรงข้ามสโคปแบบดิจิทัลถูกออกแบบมาเพื่อแสดงค่าชั่วขณะ โดยมีโครงสร้างของดิจิทัลเซออร์ 3 แบบสำหรับทรานเซียนดิิจิตเซออร์ และ RIS (Random Interleaved Sampling) สามารถตรวจจับสัญญาณชั่วขณะและบันทึกข้อมูลรูปคลื่นปริทริกเกอร์ (Pre-trigger) ได้ ในขณะที่แซมปลิงดิิจิตเซออร์ไม่สามารถทำงานได้ ดิจิตเซออร์ทั้ง 3 ชนิด สามารถบันทึกสัญญาณที่เกิดขึ้นซ้ำๆ หลายครั้งได้

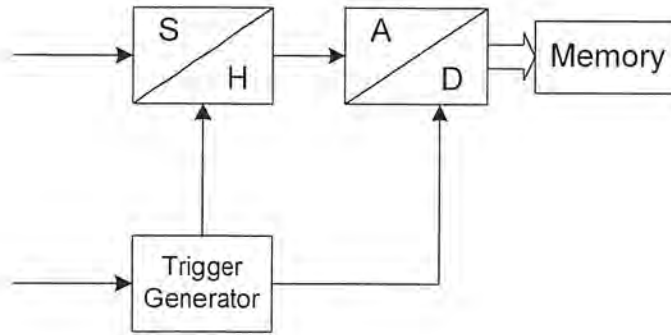
ในทรานเซียนดิิจิตเซออร์ มีอุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล และหน่วยความจำรูปคลื่น อุปกรณ์เปลี่ยนสัญญาณอนาล็อกเป็นดิจิทัลจะเปลี่ยนสัญญาณต่อเนื่องให้อยู่ในรูปตัวเลข และส่งสัญญาณที่สุ่มได้ไปยังหน่วยความจำแบบ circular addressing หลังจากใช้หน่วยความจำตำแหน่งสุดท้าย ระบบจะเขียนข้อมูลทับอันเดิม โดยเริ่มเขียนทับที่หน่วยความจำเริ่มต้น หลังจากทริกเกอร์ถูกผลิตขึ้น หน่วยความจำจะยังคงทำงานต่อไปโดยเป็นค่าที่ถูกเลือกใช้ของตัวอย่างโพสทริกเกอร์ หลังจากนั้นอุปกรณ์เปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลจะหยุดป้อนข้อมูลสู่หน่วยความจำ ถ้าผู้ใช้เลือกใช้ข้อมูลปริทริกเกอร์ 100% แล้วอุปกรณ์เปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลจะหยุดส่งข้อมูลทันทีที่ทริกเกอร์มาถึง ถ้าผู้ใช้เลือกใช้โพสทริกเกอร์ 100% ระบบจะเติมข้อมูลทุกแอดเดรสของหน่วยความจำมากกว่า 1 ครั้งแล้วหยุด หน่วยความจำจะมีข้อมูลรูปคลื่นที่เกิดขึ้นหลังทริกเกอร์



รูปที่ 3.8 บล็อกไดอะแกรมของ RIS ดิจิทัลเซออร์

RIS ดิจิทัลเซออร์ ประกอบด้วยทรานเซียนดิจิตัลเซออร์ที่เพิ่มโหมดการสอดแทรก สำหรับทริกเกอร์แต่ละอัน RIS ดิจิทัลเซออร์จะบันทึกหุคของจุดตัวอย่างรูปคลื่น จุดตัวอย่างในดิจิตัลเซออร์มาจากการได้มาของการเพิ่มทริกเกอร์ เพื่อสร้างรายละเอียดของรูปร่างสัญญาณต้นกำเนิด เนื่องจากดิจิตัลเซออร์ไม่มีทางรู้ได้ว่ามีสัญญาณทริกเกอร์แล้ว สัญญาณนาฬิกาและจุดทริกเกอร์จึงไม่สอดคล้องกัน ดังนั้นเวลาระหว่างทริกเกอร์และทุกๆสัญญาณนาฬิกาเปลี่ยนแปลงอย่างสุ่มจากรูปคลื่นที่ได้มา โครงสร้างของ RIS ใช้ TDC ในการวัดความสัมพันธ์และความแม่นยำของการได้มาของรูปคลื่น TDC มีการแก้ปัญหาล่วงเวลามากกว่าการเว้นช่วงของเวลาของตัวอย่าง ดังนั้น RIS จะช่วยแสดงผลให้เห็นรายละเอียดที่ทรานเซียนดิจิตัลเซออร์อย่างเฉยพลาดไป RIS สามารถเลือกการบันทึกปริทริกเกอร์เช่นเดียวกับทรานเซียนดิจิตัลเซออร์

ดิจิตัลเซออร์การสุ่มตัวอย่าง (Sampling Digitizer) ที่มีประสิทธิภาพประกอบด้วยส่วนหัวของการสุ่มตัวอย่าง, อุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิตอล, หน่วยความจำรูปคลื่นและวงจรสัญญาณเวลา ส่วนหัวของการสุ่มตัวอย่างจะเก็บค่าศักดาไฟฟ้าและค่าคงที่ไว้ ขณะที่อุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิตอลทำการแปลงเป็นตัวเลข ดิจิตัลเซออร์การสุ่มตัวอย่างจะได้ 1 ตัวอย่างต่อทริกเกอร์ สำหรับแต่ละทริกเกอร์ตามลำดับ วงจรสัญญาณเวลาจะหน่วงเวลาจากทริกเกอร์ถึงจุดตัวอย่าง เช่น อัตราการสุ่ม 1GS/S จุดตัวอย่างแรกจะอยู่ที่จุดทริกเกอร์ จุดที่ 2 หน่วงไป 1 นาโนวินาที จุดที่ 3 หน่วงไป 2 นาโนวินาที เนื่องจากจุดตัวอย่างถูกหน่วงจากจุดทริกเกอร์ ดังนั้นดิจิตัลเซออร์การสุ่มตัวอย่างไม่สามารถบันทึกข้อมูลที่มีก่อนทริกเกอร์ได้



รูปที่ 3.9 บล็อกไดอะแกรมของดิจิตอลเซอร์การสุ่มตัวอย่าง

3.9 แบนวิดท์และอัตราการสุ่มตัวอย่าง (Bandwidth and Sample rate)

แบนวิดท์เป็นส่วนสำคัญที่ต้องระบุสำหรับดิจิตอลเซอร์เช่นเดียวกับสโคปแบบอนาล็อก การขยายข้อมูลที่ป้อนเข้าของดิจิตอลเซอร์และการกรองเป็นตัวตัดสินกำหนดแบนวิดท์ ขอบสัญญาณพัลส์ที่เร็วและจุดยอดของรูปคลื่นที่ลึบปัด ประกอบด้วยส่วนประกอบสัญญาณความถี่สูงเพื่อที่จะบันทึกยอดและขอบของสัญญาณได้ถูกต้องแม่นยำ ดิจิตอลเซอร์ต้องมีแบนวิดท์ที่เพียงพอเพื่อจะผ่านสัญญาณความถี่สูงที่มีการลดทอนต่ำ

ต้องมีแบนวิดท์เท่าใด เพื่อแสดงการขยายจุดยอดสัญญาณอย่างถูกต้องแม่นยำ แบนวิดท์ของดิจิตอลเซอร์ควรมีมากกว่าแบนวิดท์ของสัญญาณ ดังนั้นสิ่งแรกที่ต้องพิจารณาแบนวิดท์ของสัญญาณ โดยการประมาณค่าเวลาที่เพิ่มขึ้นเร็วกว่า สมมติผลตอบสนองต่อระบบมีโพลเดียว แบนวิดท์ของสัญญาณจะเป็นดังนี้

แบนวิดท์ของสัญญาณ 0.35/(10%-90% rise time)

แบนวิดท์ของดิจิตอลเซอร์แสดงค่าซึ่งถูกลดทอนลง 3 เดซิเบล การลดทอนนี้เกิดขึ้นอย่างค่อยเป็นค่อยไป เริ่มจากความถี่ต่ำกว่า ดังนั้นต้องเลือกดิจิตอลเซอร์ซึ่งมีแบนวิดท์สูงกว่าของสัญญาณ ผลกระทบต่ออัตราการสุ่มตัวอย่างของแบนวิดท์ที่ใช้

อัตราการสุ่มตัวอย่างของดิจิตอลเซอร์สามารถลดแบนวิดท์ที่ใช้ เพื่อให้แน่ใจว่าการสุ่มตัวอย่างที่ใช้เพียงพอ เราเริ่มมา 4 ตัวอย่างรอบกับการแทรก $\sin x / x$ หรือ 10 ตัวอย่างต่อรอบกับการแทรกเส้นตรง ถ้าเป็นสัญญาณทรานเซียนพิจารณารายละเอียดอัตราการสุ่มตัวอย่าง ซึ่งเกิดข้อผิดพลาด (Single Shot) ถ้าเกิดซ้ำๆหลายครั้งสามารถใช้อัตราการสุ่มที่เท่าๆกันอย่างรวดเร็ว

ดิจิตอลเซอร์ในอุดมคติที่ไม่มีสัญญาณรบกวนและให้สัญญาณจำกัด หลักการในควิสต์ (Nyquist Theorem) ต้องได้อย่างน้อย 2 ตัวอย่างต่อแต่ละลูกคลื่น อาจกล่าวอีกอย่างได้ว่าความถี่ข้อมูลป้อนเข้าที่สูงที่สุดห้ามเกินครึ่งหนึ่งของอัตราการสุ่มตัวอย่าง ขั้นตอนการสอดแทรก $\sin x / x$ สามารถจำลองสัญญาณข้อมูลป้อนเข้าในดิจิตอลเซอร์อย่างแม่นยำถูกต้องเหมาะสม ขั้นตอน $\sin x / x$ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมาะสมกับส่วนโค้งระหว่างจุดตัวอย่างเพื่อแสดงรูปคลื่นที่เรียบ มีข้อเสียการสอดแทรก $\sin x / x$ สามารถขยายสัญญาณรบกวน เนื่องจากสัญญาณรบกวนมีอยู่ในสัญญาณจริงและในดิจิทัลไซเซอร์ ดังนั้น $\sin x / x$ ควรจะใช้อย่างระมัดระวัง โดยเฉพาะการสุ่มตัวอย่างเพียง 2 ตัวอย่างต่อรูปคลื่น

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

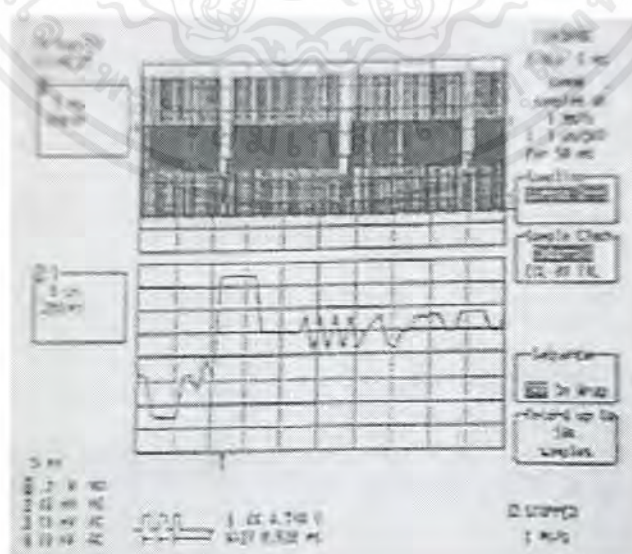
สำหรับการแทนรูปคลื่นที่แม่นยำกว่า ดิจิไลเซอร์ควรบันทึกอย่างน้อย 4 จุดตัวอย่างต่อรูปคลื่นของส่วนประกอบของคลื่นไซน์ที่มีความถี่สูงสุด จุดตัวอย่างที่เพิ่มเติมเพิ่มอัตราส่วนของสัญญาณต่อสัญญาณรบกวน สำหรับการสอดแทรก $\sin x / x$

การสอดแทรกเส้นตรงสามารถแสดงรูปคลื่นที่ถูกต้องแม่นยำโดยปราศจากการขยายสัญญาณรบกวน เพื่อผลที่ดีจะต้องการอย่างน้อย 10 ตัวอย่างต่อรูปคลื่น

3.10 ประโยชน์ของหน่วยความจำที่ยาวในดิจิตอลอสซิลโลสโคป

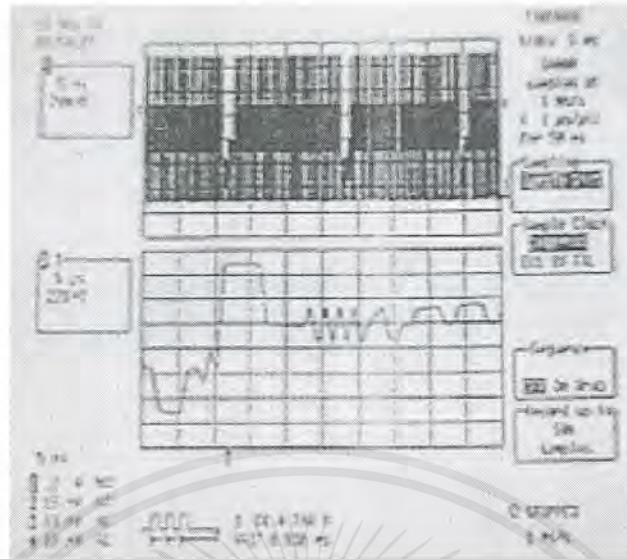
การเพิ่มความยาวของหน่วยความจำใน DSO นำมาซึ่งข้อได้เปรียบหลายอย่าง

1. ไม่สูญเสียข้อมูลของรูปคลื่น เนื่องจากอัตราการสุ่มรูปคลื่นที่มีประสิทธิภาพสูง
2. จับสัญญาณกลิทช์ได้โดยไม่ทำให้รูปคลื่นผิดรูปหรือบิดเบือนไป
3. ความละเอียดเรื่องความถี่และเวลาที่ดีกว่า
4. การจับที่เชื่อถือได้ของเหตุการณ์ซึ่งไม่สามารถทำนายได้ในเวลานั้น
5. ไม่เสียเวลาระหว่างเหตุการณ์แต่ละเหตุการณ์ที่ได้รับ



รูปที่ 3. 10 แสดงรูปคลื่นดิจิตอลอสซิลโลสโคปที่มีหน่วยความจำ 2.5 M

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 11 แสดงรูปคลื่นดิจิทัลออสซิลโลสโคปที่มีหน่วยความจำ 50 k

3.10.1 การเก็บรายละเอียด

รูปที่ 3.10 และ 3.11 แสดงรูปคลื่นเดียวกันโดยสโคปที่ต่างกัน อันแรกมีหน่วยความจำ 2.5 M และอันที่สอง 50 k การตัดสินใจที่ดีกว่าของสโคปที่มีหน่วยความจำมากกว่าสามารถเห็นได้จากการเปรียบเทียบส่วนขยายของรูปคลื่นในรูปที่ 3.10 กับการขยายในรูปที่ 3.11 ของสโคปที่มีหน่วยความจำน้อยกว่า สโคปที่มีหน่วยความจำมากกว่าจะแสดงรูปคลื่นที่ไม่บิดเบือน โดยตัวอย่างที่ชัดเจน

จากตัวอย่างแสดงให้เห็นผลของความยาวที่บันทึกบนอัตราการสุ่มตัวอย่าง สโคปทั้งสองแสดง 50 msec ของข้อมูล ดังนั้น 50 k สโคปจะมีความเร็ว 1 MS/s และ 2.5 M สโคปจะมีความเร็ว 50MS/s

ดังนั้นอัตราการสุ่มตัวอย่างเป็นหน้าที่โดยตรงของความยาวของหน่วยความจำ(จะเป็นจริงถึงขีดจำกัดของอัตราการสุ่มตัวอย่างสูงสุดของสโคป) ดังนั้นหน่วยความจำที่ยาวกว่าจะรักษาแบนวิดธ์ไว้มากกว่าการตั้งเวลาต่อช่วง โดยปราศจากอะไรที่ครึ่งๆกลางๆกับอัตราการสุ่มตัวอย่างที่ต่ำกว่า ถ้าสโคป 2 อัน มีความสามารถอัตราการสุ่มตัวอย่างเหมือนกัน สำหรับรูปคลื่นสั้นๆ ดิจิตอลสตรอเรียออสซิลโลสโคปที่มีหน่วยความจำยาวกว่าสามารถเก็บข้อมูลรูปคลื่นได้ยาวกว่าและให้แบนวิดธ์มากกว่าสำหรับสัญญาณที่ยาวกว่า

3.10.2 การจับสัญญาณกัลติฟัถวาร

สโคปแสดงข้อมูลโดยวิธีที่ต่างกัน บางอันแสดงเฉพาะส่วนเล็กๆของหน่วยความจำบนจอ หรือเลื่อนตำแหน่งบนจอเพื่อแสดงข้อมูลที่เหลือ การแสดงจุดที่วัดทั้งหมดบนจอโดยวิธีที่รูปคลื่นที่สามารถแสดงร่วมกัน นี่เป็นการใช้วิธีทำให้ร่วมกันและรับรองกัลติฟัถ การแสดงเป็นส่วนน้อย $1/8000000$ ของรูปคลื่นที่แสดงจะถูกจับอย่างแม่นยำสม่ำเสมอ และแสดงผลเปรียบเทียบกับสโคปซึ่งเชื่อถือได้ในการค้นพบจุดยอดเพื่อจับกัลติฟัถ ข้อมูลการค้นพบจุดยอดอาจจะมีประโยชน์ในการพิจารณา แต่มันจะให้ผลที่ไม่แน่นอน เพราะตลอดเวลาข้อมูลจะมีการเบี่ยง

ดิจิทัลสตรอเรจออกสซิดัล สโคปบางอันจะมีส่วนสำหรับการค้นหาจุดยอดซึ่งอุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิทัล จะมีอัตราการสุ่มตัวอย่างเฉพาะจุดสูงสุดและต่ำสุดของค่าสัญญาณ และจะถูกนำไปเก็บไว้ในหน่วยความจำ เราไม่สามารถรู้เวลาที่将会เกิดจุดยอด

3.10.3 การแก้ปัญหาเรื่องความถี่และเวลา

เปรียบเทียบสโคปในรูป 3.10 และ 3.11 สโคปอันแรกจะมีจุดตามแนวขวางมากกว่า 50 ครั้ง นั่นคือมีแฟลตเตอร์ 50:1 การแก้ปัญหาทางแกน X ที่ดีจะปรับปรุงความแม่นยำถูกต้องของการวัดความสัมพันธ์ที่เวลาขณะใดๆ และผลนี้ก็จะช่วยปรับปรุงการแสดงผลทางด้านความถี่ด้วย เพราะว่าจำนวนของจุดที่แสดงตามแบบ FFT จะเท่ากับจำนวนจุดที่บันทึกต้นแบบ

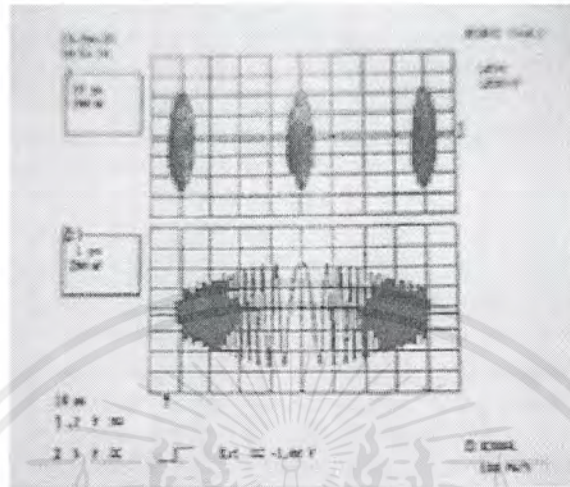
3.10.4 ความเชื่อถือได้ในการจับสัญญาณที่ไม่สามารถทำนายได้

เหตุการณ์ที่เกิดขึ้นบางเหตุการณ์ไม่สามารถทำนายได้ซึ่งเป็นการยากที่จะทริกเกอร์ได้วิธีที่ง่ายคือใช้สโคปที่มีหน่วยความจำยาว ขบวนการพัลส์จะถูกจับและขยาย มีการทำความเข้าใจกับความผิดพลาดที่เกิดขึ้นและได้ใช้สมาร์ททริกเกอร์กับเหตุการณ์เฉพาะที่เกิดขึ้นนี้

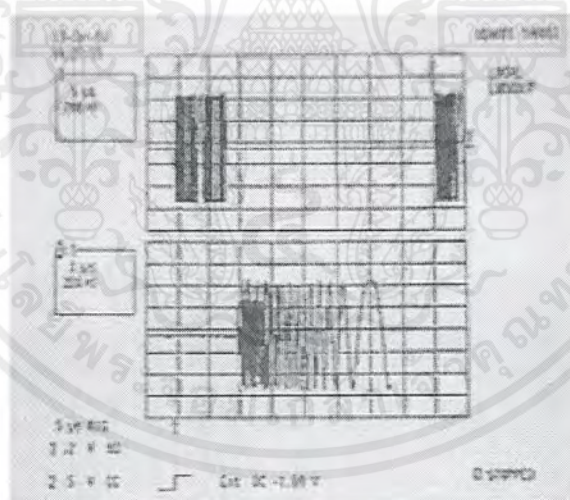
3.10.5 ไม่มีการสูญเสียเวลาระหว่างการรับข้อมูล

ช่วงเวลาระหว่างการได้มาจะมีช่วงเวลาที่แน่นอน ระหว่างช่วงเวลานี้สโคปมีขบวนการที่แสดงเป็นปกติ เวลานี้ทำให้เกิดปัญหาเมื่อได้รับเหตุการณ์ต่อเนื่องเข้ามา จากตัวอย่างเป็นสัญญาณเบิร์สต่อเนื่องตามรูปที่ 3.12 เพื่อความแม่นยำในการแสดงผลที่ต้องการอัตราการสุ่มตัวอย่างสัญญาณสูงสัมพันธ์กับช่วงระยะเวลา

รูปที่ 3.13 แสดงสัญญาณที่มีช่วงเวลาระหว่างสัญญาณเบิร์ตที่ยาวกว่า ซึ่งจะใช้วิธีการแบ่งหน่วยความจำออกเป็นส่วนย่อยๆ ทำให้สามารถลดเวลาเหลือน้อยกว่า 100 ไมโครวินาที ดังนั้นสัญญาณเบิร์ตที่แสดงถูกเก็บเป็น 50 ส่วน ส่วนละ 1 กิโลไบต์



รูปที่ 3. 12 รูปสัญญาณเบิร์ตต่อเนื่อง



รูปที่ 3. 13 แสดงสัญญาณที่มีช่วงเวลาระหว่างสัญญาณเบิร์ตที่ยาวกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

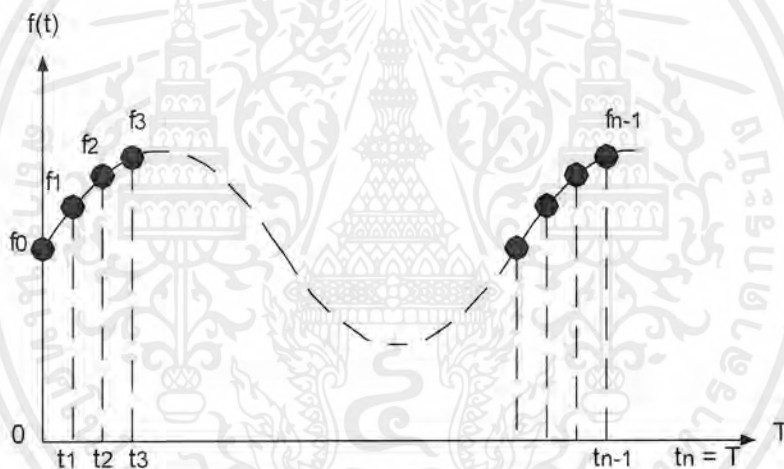
บทที่ 4

ทฤษฎีฟาสต์ฟูเรียร์ทรานฟอร์ม (FFT)

ทฤษฎีฟาสต์ฟูเรียร์ทรานฟอร์มเป็นการคำนวณฟูเรียร์เพื่อให้สามารถคำนวณได้เร็วขึ้นโดยใช้คอมพิวเตอร์เป็นตัวควบคุมการคำนวณทั้งหมด ในการที่จะทำความเข้าใจเกี่ยวกับฟาสต์ฟูเรียร์ทรานฟอร์มจะต้องศึกษาทฤษฎีดิสครีตฟูเรียร์ทรานฟอร์มเสียก่อน

4.1 ทฤษฎีดิสครีตฟูเรียร์ทรานฟอร์ม (The Discrete Fourier Transform)

ดิสครีตฟูเรียร์ทรานฟอร์มเป็นการแปลงฟูเรียร์แบบไม่ต่อเนื่อง โดยข้อมูลที่เรารวบรวมเข้ามาจะต้องเริ่มจากค่า "0" จนถึงค่า "T" ดังรูปที่ 4.1 โดยให้ค่า N เป็นจำนวนการสุ่มข้อมูลทั้งหมด โดยมีค่า t_n เป็นเวลาในการสุ่มข้อมูล และค่า f_n เป็นค่าของฟังก์ชันต่อเนื่องตามการสุ่มของ t_n ใดๆ



รูปที่ 4.1 ดิสครีตฟูเรียร์ทรานฟอร์ม

โดยข้อมูลจะอยู่ในตำแหน่ง $n = 0, 1, 2, 3, \dots, N-1$ สามารถเขียนสมการดิสครีตฟูเรียร์ทรานฟอร์มได้ดังนี้

$$F_k = \sum f_n e^{-jk\omega n} \quad \text{โดยค่า } k = 0 \text{ ถึง } N-1 \quad (1)$$

ดังนั้นจะได้ค่าอินเวิรตฟูเรียร์ทรานฟอร์มดังนี้

$$f_n = (1/N) * \sum F_n e^{jk\omega n} \quad \text{โดยค่า } n = 0 \text{ ถึง } N-1 \quad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่า $\omega_0 = 2\pi/N$ และการแปลงค่าดิซครีตฟูเรียร์ทรานฟอร์มต้องทำการคำนวณเป็นจำนวน N^2 โดยข้อมูลต้องอยู่ในรูปของจำนวนเชิงซ้อนเพื่อให้คอมพิวเตอร์คำนวณได้ง่าย จากสมการ (1) สามารถใช้ทฤษฎีของออยเลอร์ (Euler's) มาคำนวณจากสูตรดังนี้

$$e^{\pm ia} = \cos a \pm j \sin a$$

แทนสูตรของออยเลอร์ลงในสมการ (1) และ (2) จะได้

$$F_k = \sum [f_n \cos(k\omega_0 n) - j f_n \sin(k\omega_0 n)] \quad \text{โดยค่า } k = 0 \text{ ถึง } N-1 \quad (3)$$

$$f_k = \sum [F_n \cos(k\omega_0 n) - j F_n \sin(k\omega_0 n)] \quad \text{โดยค่า } n = 0 \text{ ถึง } N-1 \quad (4)$$

จากสมการ (3) และ (4) สามารถนำไปเขียนโปรแกรมเพื่อใช้ในการคำนวณหาค่าฟูเรียร์ได้ต่อไป

4.2 ทฤษฎีฟาสต์ฟูเรียร์ทรานฟอร์มและวิธีการแบบบัตเตอร์ฟลาย

ฟาสต์ฟูเรียร์ทรานฟอร์มเป็นการพัฒนามาจากดิซครีตฟูเรียร์ทรานฟอร์มเพื่อให้มีการคำนวณเร็วขึ้นซึ่งดิซครีตฟูเรียร์ทรานฟอร์มต้องการข้อมูลที่ใช้ในการคำนวณ N^2 แต่ฟาสต์ฟูเรียร์ทรานฟอร์มจะใช้ข้อมูลในการคำนวณประมาณ $N \log_2 N$

ฟาสต์ฟูเรียร์ทรานฟอร์มจะคำนวณเร็วกว่าดิซครีตฟูเรียร์ทรานฟอร์มประมาณ 10 เท่า โดยจะใช้วิธีการแบบบัตเตอร์ฟลายหรือวิธีการของคูเลย์คูคีย์ (Cooley-Tukey algorithm)

โดยสมมุติให้ $N = 2^M$ โดย M เป็นค่าจริงใดๆจะได้

$$F_k = X(k), f_n = x(n) \quad \text{จะได้สมการใหม่ดังนี้}$$

$$X(k) = \sum x(n) e^{-j(2\pi/N)nk} \quad \text{โดยค่า } k = 0 \text{ ถึง } N-1 \quad (5)$$

ถ้าให้ $e^{-j(2\pi/N)nk} = W^{nk}$ ได้สมการใหม่ดังนี้

$$X(k) = \sum x(n) W^{nk} \quad \text{โดยค่า } n = 0 \text{ ถึง } N-1 \quad (6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งสมการ (6) ได้เป็น 2 ส่วนดังนี้

$$X(k) = \sum x(2n) W_N^{nk} + \sum x(n+1) W_N^{(2n+1)k} \quad \text{โดยค่า } n = 0 \text{ ถึง } N/2-1 \quad (7)$$

N แทนขนาดความยาวของการสุ่มตัวอย่างและแทนสมการทั้งสองด้วยสมการคู่และสมการคี่ได้สมการใหม่คือ

$$X(k) = \sum x_{ev}(n) W_{N/2}^{nk} + \sum x_{od}(n) W_{N/2}^{nk} \quad \text{โดยค่า } n = 0 \text{ ถึง } N/2-1 \quad (8)$$

$$\text{หรือ} \quad X(k) = X_{ev}(k) + W_{N/2}^k X_{od}(k) \quad (9)$$

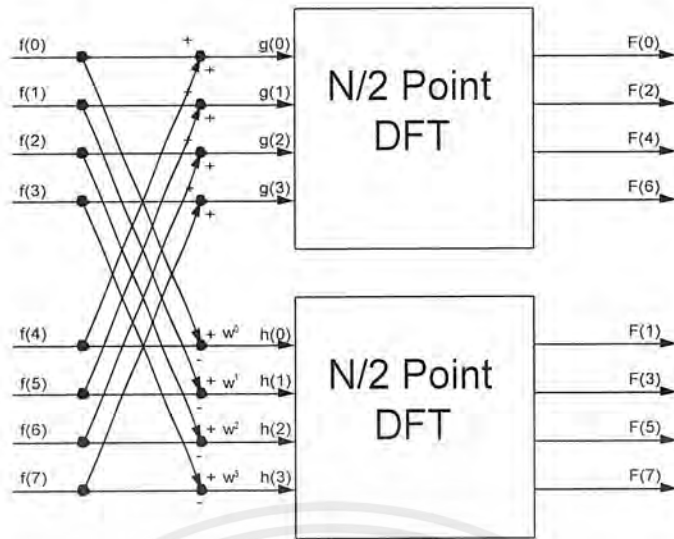
ในการนี้จะใช้ $N/2$ จุดเท่านั้นเพื่อคำนวณหาค่า $X(k)$ โดย k ต้องมีค่าตั้งแต่ 0 ถึง $N-1$ อย่างไม่รู้ก็ตามคุณสมบัติของสมการคู่และสมการคี่หาได้จาก

$$X_{cc}(k) = X_{ev}(k-N/2) \rightarrow \text{for } N/2 \leq k \leq (N-1) \quad (10)$$

ผลลัพธ์ของการแปลงฟูเรียร์แบบไม่ต่อเนื่องสามารถทำซ้ำๆ จนกระทั่งเหลือข้อมูลแค่ตัวเดียวโดยต้องมีข้อมูลอินพุต 2 ตัวเพื่อใช้ในการคำนวณตลอดเวลา

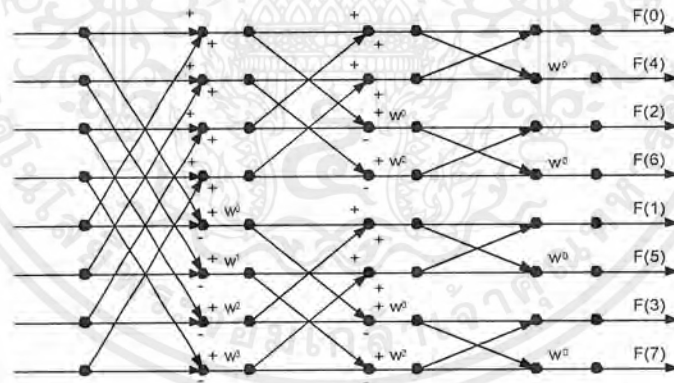
$$\begin{aligned} \Lambda(k) &= \lambda(0) + \lambda(1) e^{-j2\pi k/2} \\ &= \lambda(0) + \lambda(1) \quad \text{for all, for } k \text{ even, for } k \text{ odd ตามลำดับ} \\ &= \lambda(0) - \lambda(1) \end{aligned}$$

สำหรับการหาค่าฟาสต์ฟูเรียร์ทรานฟอร์มจะใช้แค่ 2 จุดในการคำนวณหาค่าของฟาสต์ฟูเรียร์ทรานฟอร์มที่สมบูรณ์จะต้องมีการคูณโดยใช้แฟลคเตอร์ที่เหมาะสมคือ ค่า “ W ” โดยค่า W จะต้องเริ่มจาก W^0 to $W^{N/2-1}$ ในรูปที่ 2 แสดงกราฟของฟาสต์ฟูเรียร์ทรานฟอร์ม 8 จุด โดยต้องคำนวณหาค่าต่างๆตามวิธีการของบัตเตอร์ฟลายให้เหมาะสม



รูปที่ 4.2 (a) การหาค่า 8 จุด ฟาสต์ฟูเรียร์ทรานฟอร์ม

สำหรับการเริ่มต้นคำนวณต้องทราบค่าของ N ที่จะใช้ในการคำนวณ จากรูปที่ 4.2 จะใช้ค่า $N = 8$ และจะได้ค่า g_n และ h_n ที่คำนวณได้เป็นจำนวน $N/2$ จุด โดยมีค่า W^m มาคูณกับค่า h_n ดังรูปที่ 4.1 และจะคำนวณในลำดับต่อไปดังรูปที่ 4.2 (b)



รูปที่ 4.2 (b) การหาค่า 8 จุด ฟาสต์ฟูเรียร์ทรานฟอร์ม

จากรูปที่ 4.2 (b) เมื่อกำหนดช่วงแรกเสร็จจะกำหนดช่วงที่ 2 ต่อโดยมีวิธีการกำหนดดังรูป และจะใช้ข้อมูลในการคำนวณทั้งสิ้นประมาณ $N \log_2(N)$ สามารถหาจำนวนของบิตเตอร์ฟลายได้ ดังนี้

$$\text{จำนวนของบิตเตอร์ฟลาย} = N/2 \log_2(N)$$

$$\text{โดย} \quad N/2 = \text{จำนวนแถวของบิตเตอร์ฟลาย (มี 2 อินพุต)}$$

$$\log_2(N) = \text{จำนวนหลักของบิตเตอร์ฟลาย}$$

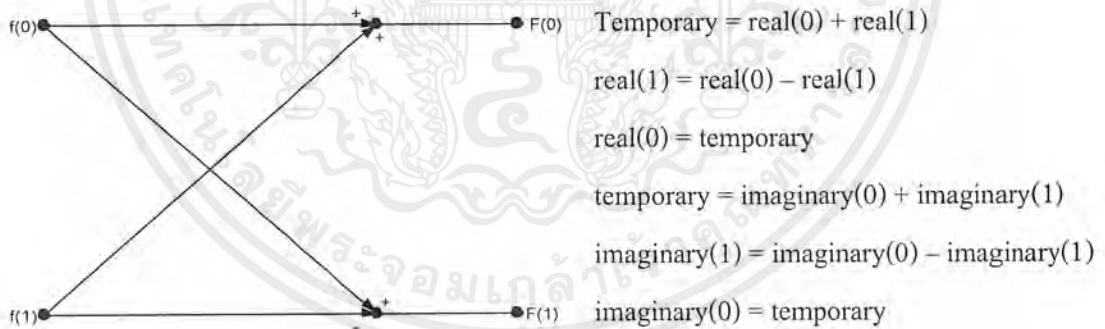
แต่ละบิตเตอร์ฟลายต้องมี 2 อินพุตเท่านั้น

4.3 การคำนวณโดยใช้คอมพิวเตอร์

ในการคำนวณโดยใช้คอมพิวเตอร์จะต้องทำการแปลงฟังก์ชันเอ็กโปเนนเชียลให้เป็น ฟังก์ชันตรีโกณมิติจากสูตร

$$e^{\pm ia} = \cos a \pm i \sin a$$

เนื่องจากโปรแกรมที่ใช้เขียนไม่รู้จักฟังก์ชันเอ็กโปเนนเชียล ในการคำนวณจะยกตัวอย่าง มาแค่บิตเตอร์ฟลายเดียว และจะต้องมีการเก็บค่าไว้ชั่วคราว ดังรูปที่ 4.2(c)



รูปที่ 4.2 (c) หลักการคำนวณ โดยใช้บิตเตอร์ฟลายเดียว

จากรูปที่ 4.2(c) จะได้ค่า

$$F(0) = f(0) + f(1)$$

$$F(1) = f(0) - f(1)$$

โดยข้อมูลที่ได้อาจจะมีทั้งค่าจริงและค่าจินตภาพ

ตัวอย่างการหาค่าฟาสต์ฟูเรียร์ทรานฟอร์ม

ตัวอย่างที่ 1 เป็นการหาค่า 16 จุดฟาสต์ฟูเรียร์ทรานฟอร์ม โดยมีการสุ่มข้อมูล 16 จุดและมีค่าสัญญาณดังนี้

$$x(n) = \cos [2\pi(4n/16)]$$

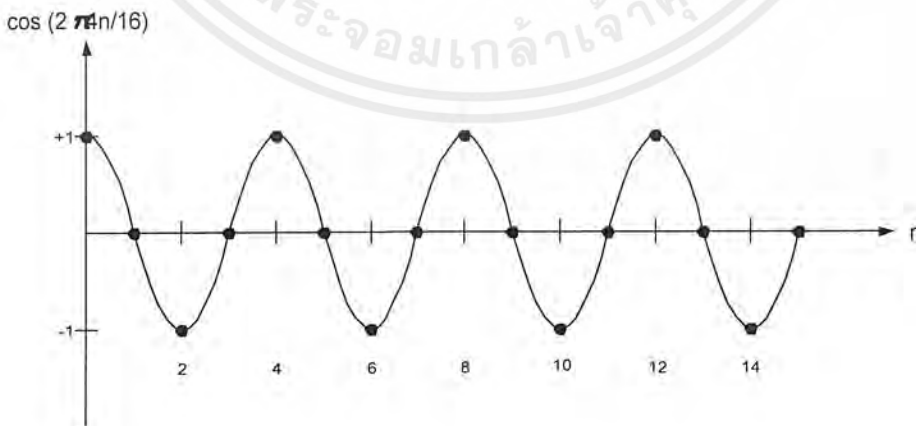
จะเปรียบเทียบจำนวนบิตเตอร์ฟลายแบบดิสครีตฟูเรียร์ทรานฟอร์มและฟาสต์ฟูเรียร์ทรานฟอร์ม ที่ใช้ในการคำนวณ

Length of Transform (N)	DFT Operation (N ²)	FFt Operation N log ₂ (N)
8	64	24
16	256	64
32	1024	160
64	4096	384
128	16384	896

ตารางที่ 4.1

เปรียบเทียบจำนวนการคำนวณระหว่างดิสครีตฟูเรียร์ทรานฟอร์มและฟาสต์ฟูเรียร์ทรานฟอร์ม

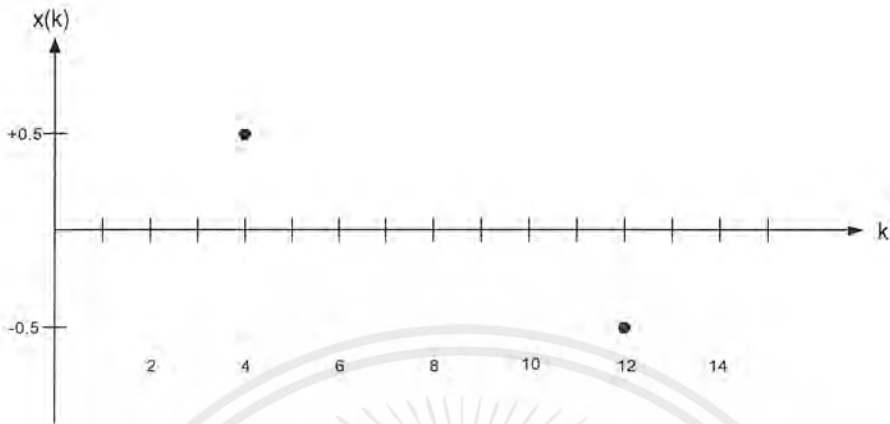
จากตัวอย่างมีค่าแอมพลิจูด “1” และสัญญาณเป็นค่าจริง ส่วนค่าจินตภาพเป็น “0” การสุ่มข้อมูลได้จากรูปที่ 4.3(a) ซึ่งมีจำนวนข้อมูล 16 จุด ตั้งแต่ x₀ – x₁₅



รูปที่ 4.3 (a) อินพุต 16 จุดฟาสต์ฟูเรียร์ทรานฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่าเอาต์พุตที่ได้จะ โขไว้ในรูปที่ 4.3(b) ซึ่งมีค่าอยู่ที่ $k = 4$ โดยมีขนาดแอมพลิจูด 0.5 และที่ $k = 12$ มีขนาดแอมพลิจูด -0.5



รูปที่ 4.3 (b) เอาต์พุตของ 16 จุด ฟาสต์ฟูเรียร์ทรานฟอร์ม

สำหรับสัญญาณที่ได้ในรูปของความถี่โดยใช้ฟูเรียร์ทรานฟอร์มคิด จะได้

$$X(f) = \int_{-\infty}^{\infty} \exp[j2\pi f_0 t] e^{-j2\pi f t} dt \quad \text{เมื่อ } t = -\infty \text{ ถึง } \infty$$

แปลงให้อยู่ในรูปเอ็กโปเนนเชียลได้

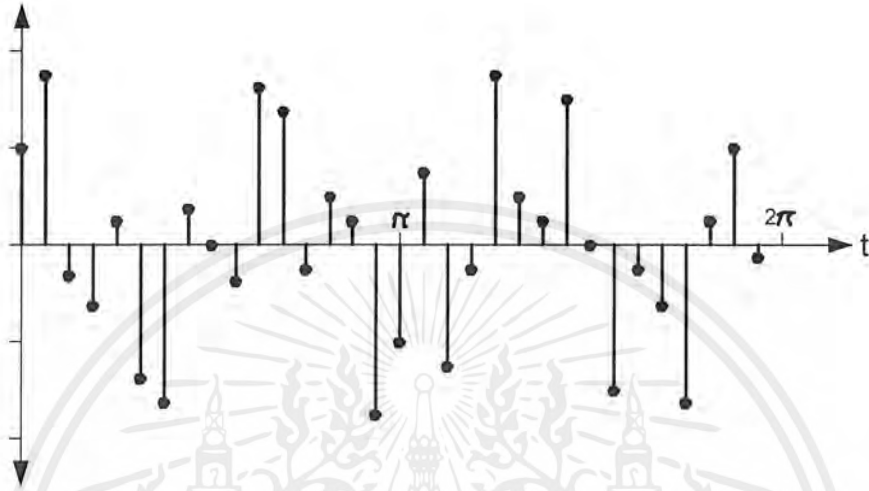
$$X(f) = 1/2 \int_{-\infty}^{\infty} \exp[j2\pi(f_0 - f)t] dt - 1/2 \int_{-\infty}^{\infty} \exp[-j2\pi(f_0 + f)t] dt \quad \text{เมื่อ } t = -\infty \text{ ถึง } \infty$$

ผลลัพธ์จะได้ที่ $k = 4$ มีขนาดแอมพลิจูด 0.5 และที่ $k = 12$ มีขนาดแอมพลิจูด -0.5 โดยที่ $k = 4$ จะมีค่า $f = -f_0$ ตามรูปที่ 4.3(b) เช่นเดียวกัน

ตัวอย่างที่ 2 จงหาค่าความถี่จากสัญญาณต่อไปนี้

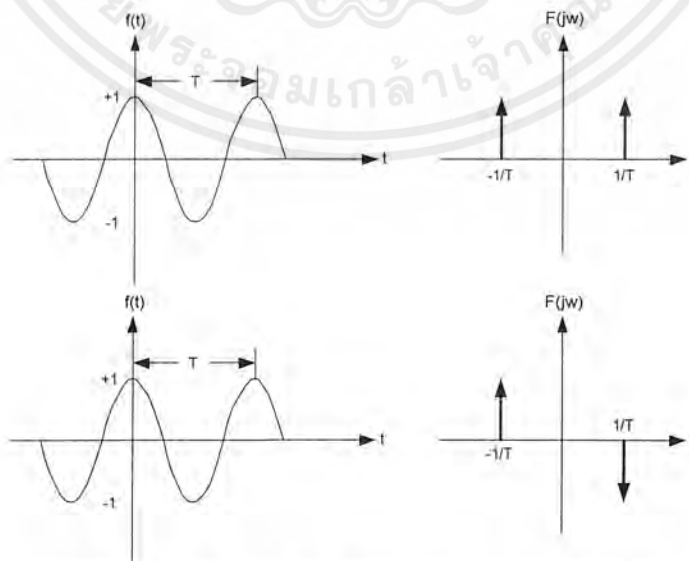
$$f(t) = \cos(3t) + \sin(10t)$$

มีการสุ่มข้อมูลมา 32 จุด เวลาที่ใช้ในการสุ่มข้อมูลทั้งหมด = $2\pi/32$ โดยมรการสุ่มข้อมูล ดังรูปที่ 4.3(c)



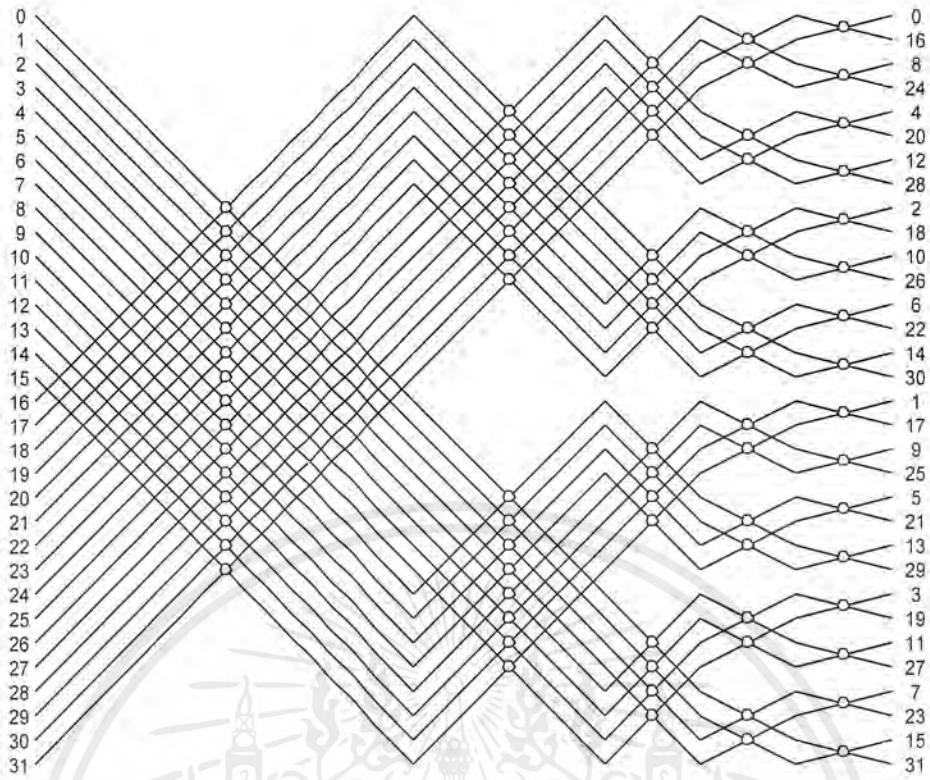
รูปที่ 4.3 (c) การสุ่มข้อมูลต่างๆจากฟังก์ชัน $\cos(3t) + \sin(10t)$

เมื่อทำการวิเคราะห์โดยใช้วิธีฟาสต์ฟูเรียร์ทรานฟอร์มจะได้ค่า cosine มีค่าเป็นบวกอยู่ที่ตำแหน่ง $+1/T$ โดยเป็นค่าจริงอยู่บนแกนความถี่และฟังก์ชัน sine จะได้ค่าบวกและลบอยู่ที่ตำแหน่ง $-1/T$ และ $1/T$ โดยจะเป็นค่าจินตภาพอยู่บนแกนความถี่ดังรูปที่ 4.3(d)



รูปที่ 4.3 (d) ฟังก์ชันไซน์, โคไซน์และผลลัพธ์ที่ได้จากฟูเรียร์ทรานฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 (e) วิธีการหาฟาสต์ฟูเรียร์ทรานฟอร์ม 32 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดใช้โดยวิธีฟูเรียร์ทรานฟอร์มจะได้ข้อมูลต่างๆดังรูปที่ 4.3 (f)

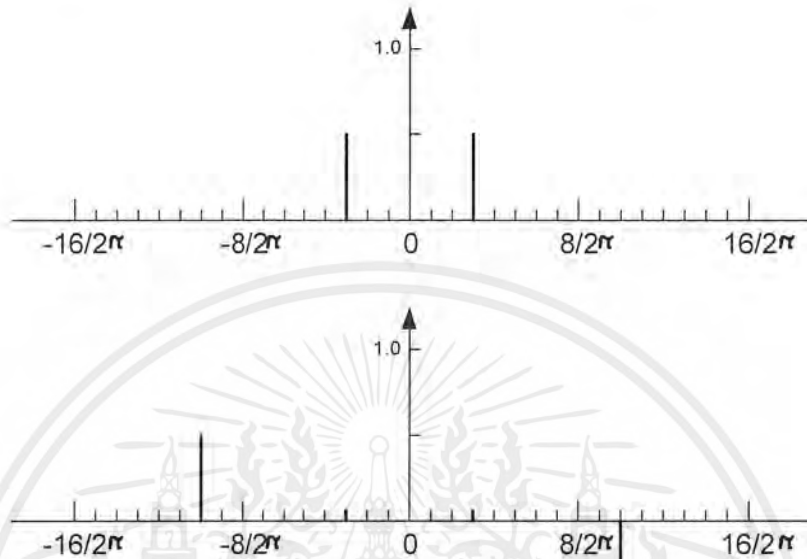
Index	F(t)	Real	Imaginary	frequency
0	1.000	-0.000	0.000	0
1	1.755	-0.000	0.000	$1/2\pi$
2	-0.324	0.000	-0.000	$2/2\pi$
3	-0.578	0.500	0.000	$3/2\pi$
4	0.0293	-0.000	-0.000	$4/2\pi$
5	-1.363	-0.000	-0.000	$5/2\pi$
6	-1.631	-0.000	-0.000	$6/2\pi$
7	0.368	-0.000	-0.000	$7/2\pi$
8	-0.000	-0.000	-0.000	$8/2\pi$
9	-0.368	-0.000	-0.000	$9/2\pi$
10	1.631	0.000	-0.500	$10/2\pi$
11	1.363	-0.000	0.000	$11/2\pi$
12	-1.293	-0.000	0.000	$12/2\pi$
13	0.578	-0.000	0.000	$13/2\pi$
14	0.324	-0.000	0.000	$14/2\pi$
15	-1.755	-0.0000	0.000	$15/2\pi$
16	-1.000	-0.000	0.000	$-16/2\pi$ (Nyquist Frequency)
17	0.092	-0.000	-0.000	$-15/2\pi$
18	-1.060	-0.000	-0.000	$-14/2\pi$
19	-0.188	-0.000	-0.000	$-13/2\pi$
20	1.707	-0.000	-0.000	$-12/2\pi$
21	0.598	-0.000	-0.000	$-11/2\pi$
22	0.217	0.000	0.500	$-10/2\pi$
23	1.479	-0.000	0.000	$-9/2\pi$
24	0.000	-0.000	0.000	$-8/2\pi$
25	-1.479	-0.000	0.000	$-7/2\pi$
26	-0.217	-0.000	0.000	$-6/2\pi$
27	-0.598	-0.000	0.000	$-5/2\pi$
28	-1.707	-0.000	0.000	$-4/2\pi$
29	0.188	0.500	-0.000	$-3/2\pi$
30	1.090	0.000	0.000	$-2/2\pi$
31	-0.092	-0.000	-0.000	$-1/2\pi$

ตารางที่ 4.2

ข้อมูลต่างๆที่ได้จากการคำนวณโดยวิธีฟาสต์ฟูเรียร์ทรานฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อมูลที่ได้ ที่ตำแหน่ง $k = 16 - 32$ จะได้ค่าความถี่เป็นค่าทางด้านลบ และที่ตำแหน่ง $k = 0 - 15$ จะเป็นค่าตำแหน่งทางด้านบวก โดยผลลัพธ์ที่ได้จะแสดงในรูปที่ 4.3(g) โดยมีค่าจริงเป็นบวกที่ $f = \pm 3/2\pi$ และค่าจินตภาพเป็นค่าบวกและค่าลบที่ $f = -10/2\pi$ และ $f = 10/2\pi$ ตามลำดับ



รูปที่ 4.3 (f) การพล็อตค่าต่างๆบนแกนความถี่

ในการหาค่าฟาสต์ฟูเรียร์ทรานส์ฟอร์มโดยใช้วิธีการแบบบัตเตอร์ฟลายนั้นจะใช้วิธีการหาค่าต่างๆตามรูปที่ 4.2 โดยจะต้องรู้ว่าคู่สมมูลญาณมาที่จุดและต้องกาค่า W มาให้ครบทุกค่าและวิธีนี้ต้องการอินพุตแค่ 2 อินพุต โดยค่าอินพุตที่ได้จะแบ่งเป็น อัปเปอร์อินพุต(Upper Input) และ โลเวอร์อินพุต(Lower Input) สมมุติให้มีบัตเตอร์ฟลายแค่ตัวเดียวจะได้

$$\text{อัปเปอร์อินพุต} = \text{อัปเปอร์อินพุต} + \text{โลเวอร์อินพุต}$$

$$\text{โลเวอร์อินพุต} = W(\text{อัปเปอร์อินพุต} - \text{โลเวอร์อินพุต})$$

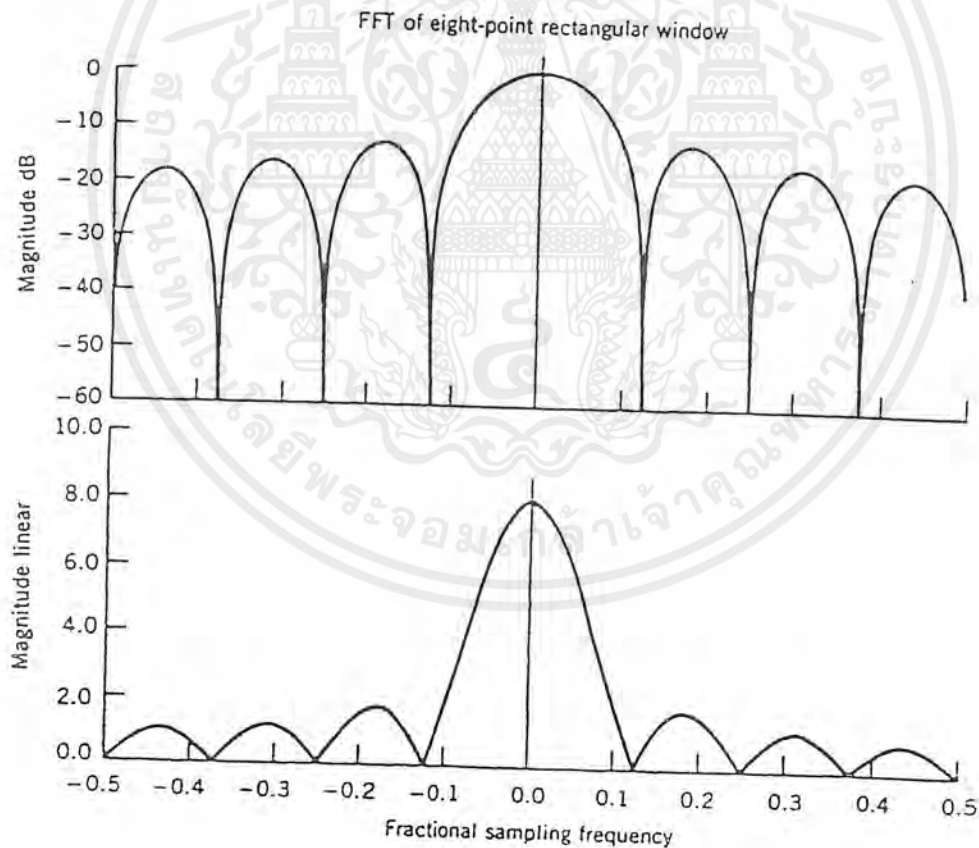
ถ้ามีบัตเตอร์ฟลายหลายตัวจะมีการคำนวณที่ยุ่งยากกว่านี้ โดยในการคำนวณจะทำเป็นขั้นตอนตามรูปที่ 4.2(a), 4.2(b), 4.2(c)

4.4 การใช้วินโดว์

ผลตอบสนองความถี่ จากการแปลง DFT จะเกิดผลความผิดเพี้ยนจากการเลือกช่องความถี่, การเกิดโหลบข้างเคียง, สกอลดอปปลิงลอส (Scalloping loss) และแบนวิธของสัญญาณรบกวน เสมือน (equipvalent noise bandwidth) ซึ่งเป็นคุณสมบัติภายในของการแปลงฟูเรียร์แบบไม่ต่อเนื่อง วิธีปรับปรุงจะทำโดยการใช้วินโดว์ คือการถ่วงน้ำหนักข้อมูลก่อนนำมาแปลงฟูเรียร์แบบเร็วตามสมการ

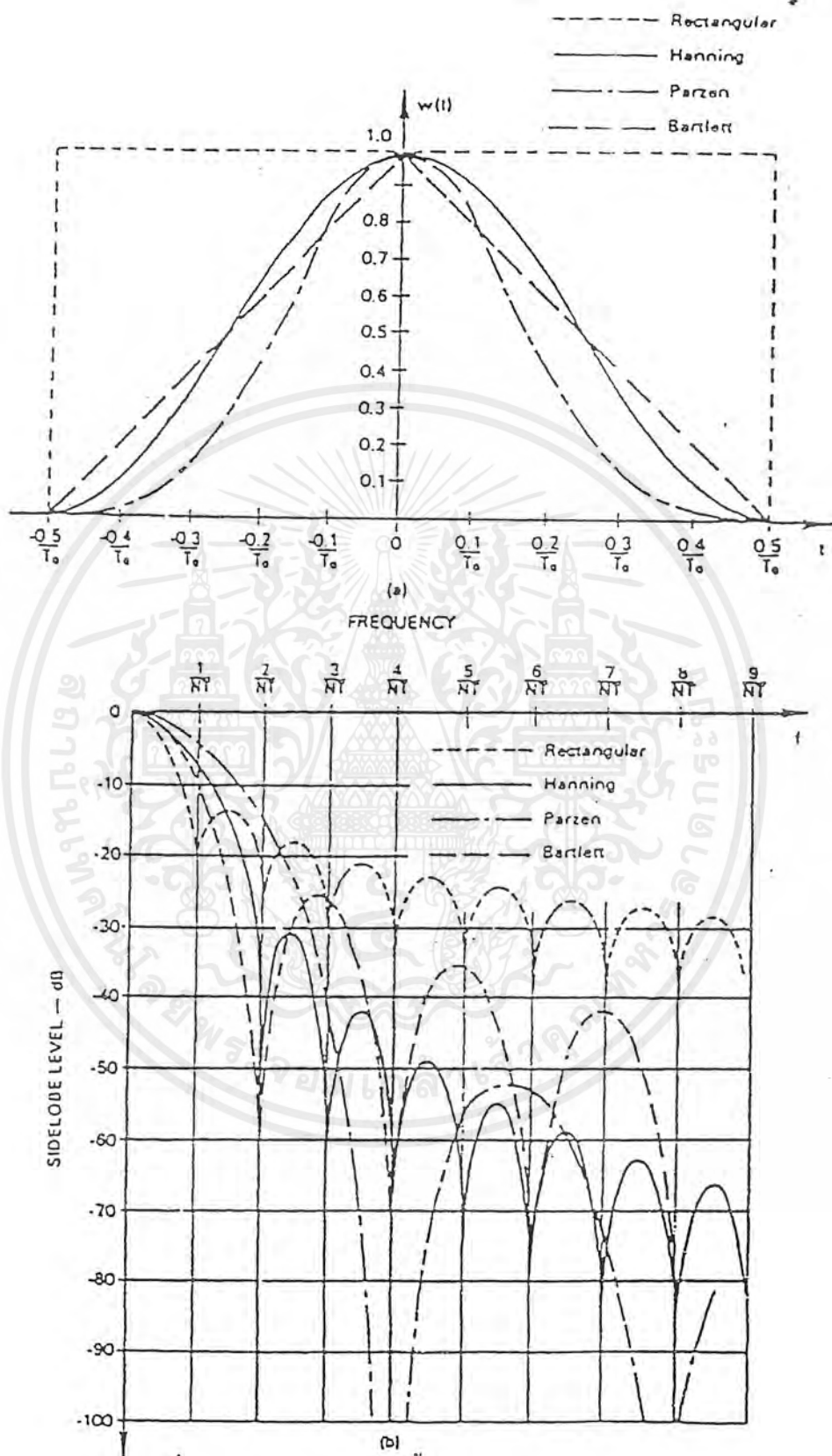
$$Y(k) = 1/N \sum_{n=0}^{N-1} a(n) x(n) W_N^{nk} \quad \text{โดย } n=0 \text{ ถึง } N-1 \quad \dots(11)$$

การแปลงฟูเรียร์แบบไม่ต่อเนื่องที่ไม่ได้ทำการถ่วงน้ำหนักคือให้ค่า $a(n) = 1$ ซึ่งเราเรียกฟังก์ชันถ่วงน้ำหนักที่เหลี่ยม (Rectangular weighting function)



รูปที่ 4.4 การเกิดโหลบข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงฟังก์ชันถ่วงน้ำหนักหรือวินโดว์ของ FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม เราสุ่มสัญญาณขาเข้า ไปคูณด้วยฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม เพื่อจำกัดจำนวนของจุดสุ่มเพื่อให้เหลือเท่ากับ N ในที่นี้การตัดทอนของโดเมนเวลา (Time Domain) ก็คือ การถ่วงน้ำหนักของข้อมูลโดยการคูณด้วยฟังก์ชันสี่เหลี่ยมเรารู้ว่าฟังก์ชันในโดเมนความถี่ (Frequency Domain) ของฟังก์ชันสี่เหลี่ยมคือฟังก์ชัน $(\sin f)/f$ ดังนั้นการแปลงโดเมนความถี่ของฟังก์ชันเวลาที่ตัดทอนแล้วจึงเป็นการทำคอนโวลูชัน (Convolution) ของฟังก์ชันสี่เหลี่ยม ทำให้ผลที่ได้มีองค์ประกอบของความถี่เพิ่มขึ้นมาคือ โทลบข้าง (side lobe) ของฟังก์ชันสี่เหลี่ยมองค์ประกอบที่เพิ่มขึ้นมานี้เราเรียกส่วนรั่วไหล (leakage) ทำให้อิมพัลส์ของความถี่ของฟังก์ชันรั่วออกไปที่ลอนข้างของฟังก์ชัน $(\sin f)/f$

เพื่อลดการรั่วไหลเราจำเป็นต้องใช้วิธีการตัดทอนโดเมนเวลาหรือฟังก์ชันถ่วงน้ำหนักที่มีลักษณะข้างในโดเมนความถี่เล็กกว่าฟังก์ชันสี่เหลี่ยม ลอนข้างที่เล็กกว่านี้ทำให้การรั่วไหลของผลลัพธ์การแปลงฟูเรียร์แบบเร็วมีค่าน้อยลง

ฟังก์ชันที่นำมาถ่วงน้ำหนักจะต้องมีคุณสมบัติ คือ

1. ให้โทลบหลักของการตอบสนองความถี่แคบ แต่ภายในโทลบหลักสามารถครอบคลุมพลังของสัญญาณทั้งหมดของโทลบหลักไว้ให้มากที่สุด
2. ให้ผลการตอบสนองความถี่ของวินโดว์มีค่าขนาดของโทลบข้างน้อย และมีขนาดลดลงอย่างรวดเร็วเมื่อความถี่สูงขึ้น

3.

ฟังก์ชันถ่วงน้ำหนักหรือวินโดว์ที่ใช้คือ

1. วินโดว์แบบฮานมีสมการคือ

$$a(n) = 0.5 + 0.5\cos(2\pi n/N-1) \quad \text{เมื่อ } |n| \leq (N-1)/2$$

$$= 0 \quad \text{เมื่อ } n \text{ กรณีอื่นๆ}$$

2. วินโดว์แบบแฮมมิงมีสมการคือ

$$a(n) = 0.54 + 0.46\cos(2\pi n/N-1)$$

$$= 0 \quad \text{เมื่อ } |n| \leq (N-1)/2$$

$$\text{เมื่อ } n \text{ กรณีอื่นๆ}$$

3. วินโดว์แบบเบล็กแมนมีสมการคือ

$$a(n) = 0.42 + 0.5\cos(2\pi n/N-1) + 0.8\cos(4\pi n/(N-1))$$

$$= 0 \quad \text{เมื่อ } |n| \leq (N-1)/2$$

$$\text{เมื่อ } n \text{ กรณีอื่นๆ}$$

4.5 กราฟการแปลงฟูรีเยร์แบบเร็ว

ผลลัพธ์จากการแปลงฟูรีเยร์แบบเร็วแต่ละความถี่จะมีระยะห่าง $f_0 = 1/NT$ ดังนั้นจุดสูงสุดในโดเมนจึงมีค่าตั้งแต่ $0/NT, 1/NT, 2/NT, \dots, (N/2)/NT$ สำหรับด้านความถี่ด้านความถี่บวก ระยะห่างของความถี่ $f_0 = 1/NT$ เป็นเทอมที่แสดงความละเอียดของการแปลงฟูรีเยร์แบบเร็วซึ่งเราอาจลดระยะห่างของแต่ละความถี่ได้โดยการเพิ่มจำนวนข้อมูล N ถ้าค่าของ N เพิ่มขึ้นเป็นสองเท่าระยะห่างของความถี่ก็จะลดลงสองเท่าเช่นกัน



บทที่ 5

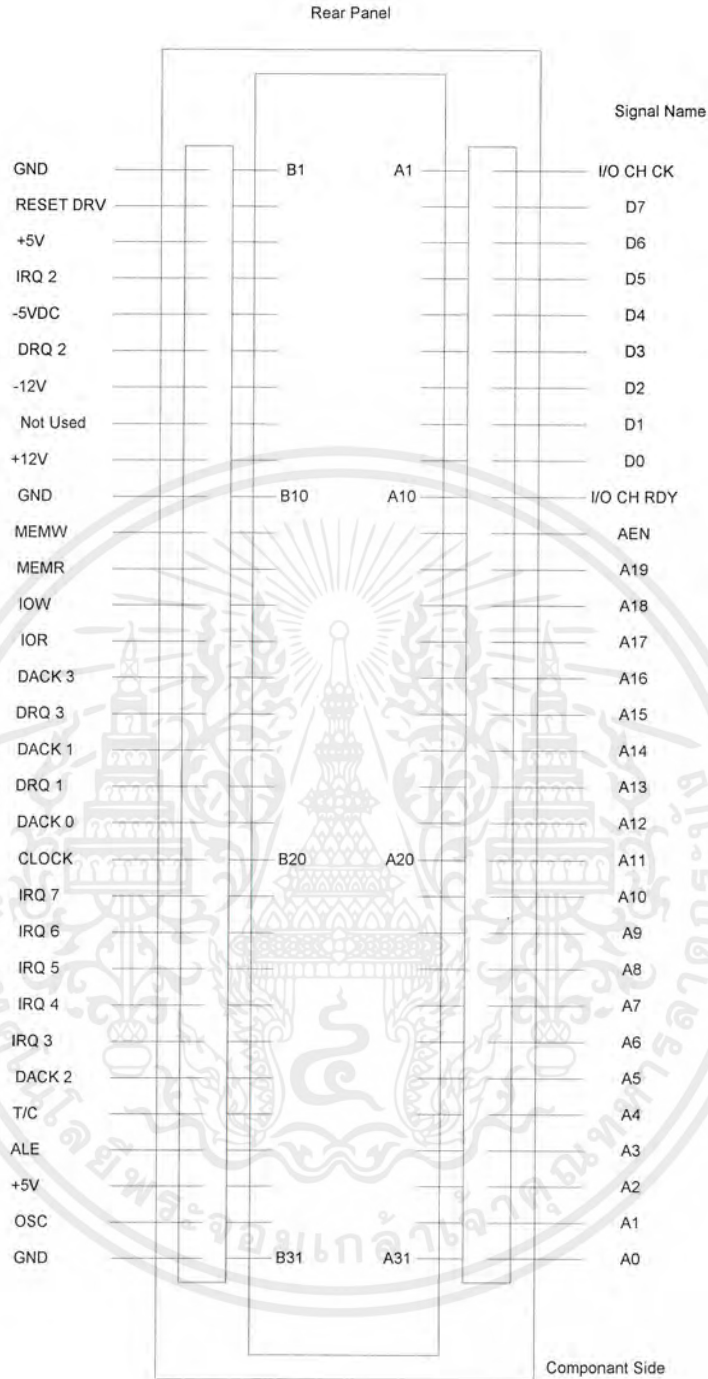
สัญญาณต่างๆบนสล็อตของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรรินเตอร์เฟสเข้าไปในภายหลังได้ โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ด (Main Board) สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต (สำหรับใน IBM PC /xt จะมี 8 สล็อต) ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็นสองข้างๆละ 31 ขา ส่วนการเรียกขาของสล็อตเหล่านี้จะขึ้นอยู่กับว่าขาผู้นั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อต โดยขาที่อยู่ทางด้านซ้ายของสล็อต จะเรียกโดยใช้ตัวอักษร B นำหน้าเลขตำแหน่งของขา เช่น B16 ก็คือขาทางด้านซ้ายของสล็อต ขาที่ 16 (นับจากทางด้านซ้ายของเครื่อง) ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกโดยใช้อักษร A นำหน้าเลขตำแหน่งของขา เช่น ขา A24 ก็คือขาทางด้านขวาของสล็อตขาที่ 24 (นับจากทางด้านซ้ายของเครื่อง)

แต่ละขาของสล็อตเหล่านี้จะเชื่อมต่อกับเส้นสัญญาณต่างๆบนเมนบอร์ด ทำให้การสร้างวงจรรินเตอร์เฟสกับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus), บัสข้อมูล (Data Bus), บิวควบคุมสำหรับการเขียน/อ่านข้อมูลจากหน่วยความจำ หรือพอร์ต I/O, เส้นสัญญาณสำหรับการขออินเตอร์เฟส, เส้นสัญญาณสำหรับการขอ DMA, สัญญาณฐานเวลา (Timing Signal) ต่างๆที่ใช้ในระบบ, เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHCK)

5.1 รายละเอียดเกี่ยวกับสัญญาณต่างๆ

OSC (Oscillator ; ขา B30) : ขานี้เป็นเอาต์พุตที่เชื่อมต่อกับสัญญาณคล็อกที่มีค่าความถี่สูงสุดบนเมนบอร์ด คือ 14.31818 MHz ซึ่งมีคาบเวลาประมาณ 70 ns และมี Duty Cycle ๖ ช่วงเวลาใน 1 คาบที่สัญญาณคล็อกมีลอจิกเป็น 1 หารด้วยคาบเวลาทั้งหมด) ประมาณ 50% สัญญาณคล็อกอื่นๆของระบบ เช่น คล็อกที่ป้อนให้กับ 8088 หรือ ชิพซีพพอร์ทต่างๆนั้นจะถูกสร้างขึ้นโดยการหารสัญญาณนี้ อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงในการใช้งานสัญญาณ OSC ก็คือ สัญญาณนี้จะไม่ Synchronize กับสัญญาณอื่นๆบนบัสของระบบ ดังนั้นจึงไม่ควรที่จะนำสัญญาณจากขา OSC นี้ไปใช้เป็นสัญญาณคล็อกสำหรับวงจรรายนอกอื่นๆที่ทำงานร่วมกับระบบ



รูปที่ 5.1 ขาสัญญาณต่างๆของ ISA Bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLK (Clock ; ขา B20) : ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งต่อกับสัญญาณคล็อกที่ถูกสร้างขึ้น โดยการหารสัญญาณ OSC ด้วย 3 ทำให้ได้ความถี่ประมาณ 4.77 MHz (14.31818 MHz/3) หรือ มีช่วงเวลาใน 1 คาบ (ช่วงเวลาของคล็อก 1 ลูก) เท่ากับ 210 ns (1/4.77 MHz) สำหรับค่า Duty Cycle ของสัญญาณนี้จะมีค่าประมาณ 1/3 คือ ใน 1 คาบจะมีช่วงเวลาที่เป็นลอจิก 1 เท่ากับ 1/3 ของคาบเวลาทั้งหมด หรือประมาณ 70 ns และช่วงเวลาที่เป็นลอจิก 0 เท่ากับ 2/3 ของคาบเวลาทั้งหมด หรือประมาณ 140 ns สัญญาณนี้เป็นสัญญาณที่ถูกใช้เป็นตัวคล็อกของระบบ

RESET DRV (ขา B2) : ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งจะแอกทีฟ (ลอจิก 1) ในช่วงที่เรารับเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอกทีฟไปจนกว่าระบบต่างๆ ภายใน IBM/PC จะพร้อมที่จะทำงานได้ จากนั้นสัญญาณนี้ก็เปลี่ยนกลับเป็นลอจิก 0 นอกจากนี้ในระหว่างการทำงานของ IBM/PC ถ้าระดับแรงดันของแหล่งจ่ายไฟลดลง สัญญาณนี้ก็จะถูกทำให้แอกทีฟเช่นกัน โดยทั่วไปแล้วสัญญาณนี้จะถูกนำไปใช้ในการรีเซ็ตวงจรอินเทอร์เฟซหรืออุปกรณ์ I/O ต่างๆ ในช่วงที่เริ่มจ่ายไฟให้กับระบบ ซึ่งจะเป็นการทำให้วงจรหรืออุปกรณ์เหล่านั้นถูกปรับให้อยู่ในสถานะที่แน่นอนก่อนที่จะเริ่มต้นการทำงานในระบบ (สถานะนี้เป็นสถานะที่เราทราบ และต้องการให้วงจรทำงานในขณะที่ระบบถูกรีเซ็ต)

A0-A19 (Address Bus ; ขา A31-A12) : ขาสัญญาณทั้ง 20 ขานี้เป็นเอาต์พุต ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำหรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อกับ โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (Least Significant Bit) และ A19 นี้จะมีนัยสำคัญสูงสุด (Most Significant Bit) สำหรับค่าแอดเดรสบนบัสแอดเดรส A0-A19 นี้ จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน/เขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA-Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ)

จะเห็นได้ว่าจำนวนเส้นแอดเดรสจะมีอยู่ 20 เส้น ซึ่งสามารถที่จะอ้างแอดเดรสของหน่วยความจำได้ถึง 1 Mbyte แต่อย่างไรก็ตามจะมีแอดเดรสบางแอดเดรสที่ถูกใช้งานโดย IBM/PC อยู่ก่อนแล้ว คือ แอดเดรสของหน่วยความจำ RAM บนเมนบอร์ดที่ถูกใช้โดยระบบ จำนวน 64 Kbyte (สำหรับ IBM PC/XT จะเป็นจำนวน 256 Kbyte) และแอดเดรสสำหรับหน่วยความจำ ROM อีก 48 Kbyte ซึ่งถูกจัดในช่วงแอดเดรสบนสุดใน 1 Mbyte คือ 0FC00H จนถึง 0FFFFFFH (สำหรับ IBM PC/XT จะเป็น 64 Kbyte)

สำหรับการอ้างแอดเดรสของพอร์ต I/O นั้น จะใช้เส้นแอดเดรสเพียง 16 เส้น คือ A0-A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ตได้ 64 K พอร์ต โดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอดเดรสที่เหลือ คือ A16-A19 นั้นจะไม่ถูกใช้งาน อย่างไรก็ตามใน IBM/PC จะใช้เส้นแอดเดรสในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างแอดเดรสของพอร์ตเพียง 10 เส้น คือ จาก A0-A9 และค่าแอดเดรสที่ใช้งานจะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

D0-D7 (Data Bus ; ขา A9-A2) : ขาสัญญานี้จะเป็นแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ต I/O กับ IBM/PC โดยบิต D0 จะมีนัยสำคัญต่ำสุดและบิต D7 จะมีนัยสำคัญสูงสุด

สำหรับบัสไซเคิลของการเขียนข้อมูลที่สร้างโดย 8088 นั้น ข้อมูลจะถูกส่งออกมาบนบัสข้อมูล ก่อนที่สัญญาณ IOW (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ต) หรือ MEMW (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก 0 เป็น 1 (ขอบขาขึ้น) ซึ่งโดยทั่วไปขอบขาขึ้นของสัญญาณ IOW หรือ MEMW นี้ จะถูกใช้เพื่อสั่งให้พอร์ต I/O หรือหน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลไปเก็บไว้

สำหรับในบัสไซเคิลการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ต I/O หรือหน่วยความจำที่ถูกอ้างถึงจะต้องส่งข้อมูลออกมาบนบัสข้อมูล ก่อนที่ IOR (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ต) หรือ MEMR (ในกรณีที่ต้องการอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจากลอจิก 0 เป็นลอจิก 1 (ขอบขาขึ้น)

ALE (Address Latch Enable ; ขา B28) : ขาสัญญานี้เป็นขาเอาต์พุตที่ 8088 Bus Controller สร้างขึ้นเพื่อใช้สำหรับแสดงการเริ่มต้นของบัสไซเคิล และแสดงให้อุปกรณ์ภายนอกทราบว่าแอดเดรสที่ 8088 ต้องการจะติดต่อด้วยนั้นถูกส่งออกมาบนบัสแอดเดรสแล้ว โดยที่สัญญาณ ALE นี้จะเปลี่ยนจากลอจิก 1 เป็น 0 เมื่อค่าแอดเดรสที่ต้องการส่งออกมาบนบัสข้อมูลเรียบร้อยแล้ว ดังนั้นขอบขาลงของสัญญาณ ALE นี้จะถูกใช้ในการแลทช์ค่าแอดเดรสจากบัสแอดเดรส/ข้อมูล Address/Data Bus ; AD0-AD7) ของ 8088 ทำให้สามารถแยกค่าแอดเดรส (A0-A19) และข้อมูล (A0-A7) ออกจากกันได้ อย่างไรก็ตามสัญญาณ ALE จะแอกทีฟเฉพาะในบัสไซเคิลที่สร้างโดย 8088 เท่านั้น โดยจะไม่แอกทีฟในระหว่างขบวนการ DMA

I/OCHCK (I/O Channel Chack ; ขา A1) : ขาสัญญานี้เป็นอินพุตที่ใช้ในการแสดงความผิดพลาดเกี่ยวกับพาร์ตี ที่เกิดขึ้นในการทำงานของวงจรรีจิสเตอร์เฟสหรืออุปกรณ์ I/O เมื่อขานี้ได้รับลอจิก 0 จะทำให้ 8088 ถูกอินเทอร์รัพท์แบบ Non-Maskable (NMI) อย่างไรก็ตามเราสามารถที่จะกำหนดให้วงจรรายในของ IBM/PC ทำการขออินเทอร์รัพท์ (เมื่อได้รับสัญญาณ I/O CHCK) หรือไม่ได้ โดยการกำหนดลอจิกของบิตข้อมูลของพอร์ตที่ควบคุมการขออินเทอร์รัพท์แบบ NMI คือ บิต D7 ของพอร์ต 00A0H ในกรณีที่บิต D7 ของพอร์ต 00A0H ถูกเซตเป็น 1 ก็จะให้วงจรราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกขอนินเตอร์รัพท์แบบ NMI ได้ (Enable) แต่ถ่าบิท D7 ของพอร์ท 00A0H ถูกเซ็ทเป็น 0 ก็จะมีการ ดิสเอเบิล (Disable) การขอนินเตอร์รัพท์แบบ NMI ดังนี้

Enable : ใช้คำสั่ง OUT ส่งข้อมูล 80H ไปยังพอร์ท 00A0H

Disable : ใช้คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

และเนื่องจากที่มีอุปกรณ์อื่นที่สามารถขอนินเตอร์รัพท์แบบ NMI ได้อีก ดังนั้นซอฟต์แวร์ที่ใช้งานจะต้องสามารถตรวจสอบว่าการขอนินเตอร์รัพท์นั้นเกิดขึ้นจากแหล่งใดได้ด้วย

I/O CHRDY (I/O Channel Ready ; ขา A10) : ขาสัญญาณนี้เป็นอินพุทที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อุปกรณ์ I/O หรือหน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น ไม่สามารถทำงานทันตามช่วงเวลาปกติของบัสไซเคิลนั้นๆได้(ช่วงเวลาของบัสไซเคิลที่เกี่ยวกับหน่วยความจำใช้ช่วงเวลาเท่ากับช่วงเวลาคlocks 4 ลูก หรือ 840 ns ในขณะที่บัสไซเคิลที่เกี่ยวกับ I/O จะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 5 ลูก หรือ 1.05 μ s)

เมื่ออุปกรณ์ I/O หรือหน่วยความจำต้องการที่จะเพิ่มช่วงเวลาในบัสไซเคิลให้นานขึ้นอีกนั้น จะสามารถทำได้โดยการป้อนลอจิก 0 ให้กับขา I/O CHRDY ในช่วงที่ I/O หรือหน่วยความจำที่ถูกกำหนดนั้น ได้รับสัญญาณจากการตีโค้ดแอดเดรส และสัญญาณ MEMR, MEMW, IOR หรือ IOW แอคทีฟ

IRQ2-IRQ7 (Interrupt Request 2-7 ; ขา B4 และ B25-B21) : ขาสัญญาณทั้ง 6 นี้เป็นขาอินพุทที่ใช้สำหรับการขอนินเตอร์รัพท์จาก 8088 โดยสัญญาณเหล่านี้จะต่อเข้ากับ 8259A บนเมนบอร์ดโดยตรง โปรแกรมในส่วนของ BIOS ของ IBM/PC จะทำให้การโปรแกรม 8259A ให้ IRQ2 มีลำดับความสำคัญสูงสุด (Highest Priority) และ IRQ7 มีลำดับความสำคัญที่ต่ำสุด ในกรณีที่มีการขอนินเตอร์รัพท์เกิดขึ้น คือระดับลอจิกที่ขา IRQ ขาใดขาหนึ่งถูกเปลี่ยนจากลอจิก 0 เป็นลอจิก 1 (ขอบขาขึ้น) 8259A ก็จะทำการส่งสัญญาณ INT ให้กับ 8088 เพื่อทำการขอนินเตอร์รัพท์

สิ่งสำคัญในการขอนินเตอร์รัพท์ผ่านทาง IRQ2-IRQ7 นี้ ก็คืออุปกรณ์ที่จะทำการขอนินเตอร์รัพท์โดยผ่านทาง IRQ ขาใดก็จะต้องรักษาระดับสัญญาณที่ขา IRQ นั้น ให้แอคทีฟ (ลอจิก 1) อยู่จนกว่าจะได้รับสัญญาณ INTA (Interrupt Acknowledge) จาก 8088 เสียก่อน ถ้าไม่เช่นนั้น การขอนินเตอร์รัพท์จะถูกยกเลิก และอินเตอร์รัพท์ Level 7 (IRQ7) ก็จะถูกสร้างขึ้นโดยอัตโนมัติไม่ว่าการขอนินเตอร์รัพท์ที่ถูกยกเลิกนั้นจะเป็นการขอนินเตอร์รัพท์ใน Level หรือ ขาใด

แต่อย่างไรก็ตามสัญญาณ INTA นี้จะไม่ถูกต่อออกมาที่ขาของสล็อตด้วย ดังนั้นโปรแกรมที่ทำการตอบสนองต่อการขอนินเตอร์รัพท์ (Interrupt Service Routine) จะต้องทำการรีเซ็ตสัญญาณ IRQ เอง โดยใช้คำสั่ง OUT ไปยังพอร์ท I/O ที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IOR (I/O Read ; ขา B14) : ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก 0 ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลที่อ่านข้อมูลจากพอร์ต I/O เพื่อให้พอร์ต I/O ที่มีแอดเดรสที่ตรงกับที่แอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูลโดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอขาขึ้นของสัญญาณ IOR ประมาณ 30 ns เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง สำหรับขบวนการ DMA 8237A-5 DMA Controller จะทำการสร้างสัญญาณ IOR เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำ (แทนที่จะเป็นแอดเดรสของพอร์ต I/O) ที่พอร์ต I/O ที่ขอ DMA ต้องการจะนำข้อมูลไปเก็บ การที่พอร์ตใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้น จะต้องอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนด เช่นกรณีที่สัญญาณ DACK1 แอกทีฟก็แสดงว่าพอร์ต I/O ที่จะต้องส่งข้อมูลออกมา ก็คือพอร์ต I/O ที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) เป็นต้น

IOW (I/O Write ; ขา B13) : ขาสัญญาณนี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ต I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้น รับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ได้อย่างไรก็ตามเนื่องจากในช่วงเวลาที่สัญญาณ IOW นี้แอกทีฟ (ลอจิก “0”) นั้นข้อมูลบนบัสแอดเดรสอาจจะยังไม่สมบูรณ์ ดังนั้นในการออกแบบจึงควรใช้ขอขาขึ้นของสัญญาณ IOW แทนขอขาลงในการทำให้พอร์ต I/O ที่เกี่ยวข้องรับข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลเสร็จเสียก่อน สำหรับในขบวนการ DMA นั้น DMA – Controller จะทำการสร้างสัญญาณ IOW เอง โดยให้ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำที่พอร์ต I/O ที่ขอ DMA ต้องการจะอ่านข้อมูล

MEMW (Memory Write ; ขา B11) : ขานี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่ง 8288 Bus Controller สร้างขึ้นในระหว่างบัสไซเคิลในการเขียนข้อมูลลงในหน่วยความจำของ 8088 สัญญาณ MEMW นี้จะถูกส่งออกมาเพื่อให้หน่วยความจำที่แอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ โดยทั่วไปหน่วยความจำจะรับข้อมูลในช่วงขอขาขึ้นของสัญญาณ MEMW

สำหรับในระหว่างขบวนการ DMA นั้น 8237A-5 DMA – Controller จะทำการควบคุมบัสต่างๆของระบบแทน 8088 และสัญญาณ MEMW จะถูกใช้ในบัสไซเคิลของการเขียนข้อมูลลงในหน่วยความจำ (ข้อมูลถูกส่งจากอุปกรณ์ I/O ไปให้กับหน่วยความจำ)

MEMR (Memory Read ; ขา B12) : ขานี้เป็นขาเอาต์พุตจาก 8288 ซึ่งสัญญาณนี้จะแอกทีฟ (ลอจิก “0”) ในระหว่างบัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำของ 8088 เพื่อให้หน่วยความจำที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการส่งข้อมูลออกมาบนบัสข้อมูล โดยหน่วยความจำนั้นจะต้องส่งข้อมูลออกมาในช่วงเวลา 30 nanosec ก่อนที่สัญญาณ MEMW จะกลับเป็นลอจิก “1” ทั้งนี้ก็เพื่อให้ 8088 ได้รับข้อมูลที่ถูกต้อง

สำหรับในระหว่างขบวนการ DMA นั้น DMA – Controller จะควบคุมบัสต่างๆของระบบแทน 8088 และสัญญาณ MEMR จะถูกใช้ในบัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำ (ข้อมูลถูกส่งจากหน่วยความจำไปให้กับอุปกรณ์ I/O)

DRQ 1-DRQ3 (DMA Request 1-3 ; ขาB18, B6 และ ขาB16) : ขาสัญญาณทั้งสามนี้เป็นสัญญาณอินพุตแอกทีฟที่ลอจิก “1” ซึ่งอุปกรณ์ภายนอกจะใช้ในการขอ DMA จากระบบ โดยการป้อนระดับสัญญาณลอจิก “1” ให้กับขา DRQ ขาใดขาหนึ่ง (ขา DRQ ทั้งสามนี้จะต่อเข้ากับ DRQ 1-DRQ3 ของ 8237A-5)

เมื่อ 8237A-5 ได้รับสัญญาณนี้แล้วก็จะตรวจสอบว่ามีการขอ DMA ในแชนแนลที่มีลำดับความสำคัญ (Priority) สูงกว่าหรือไม่ ถ้าไม่มีก็จะทำการขอ DMA จาก 8088 และตอบรับการขอ DMA จากอุปกรณ์ภายนอก (สัญญาณ DACK ของแชนแนลที่ขอ DMA จะแอกทีฟ) แต่ถ้ามี 8237A-5 ก็จะทำการขอ DMA ให้กับแชนแนลที่มีลำดับความสำคัญสูงกว่าแล้วจึงทำการขอ DMA ให้กับแชนแนลที่มีลำดับความสำคัญต่ำกว่า ภายใน ROM BIOS ของ IBM/PC จะโปรแกรม 8237A-5 ให้ DRQ1 มีลำดับความสำคัญสูงสุดและ DRQ3 มีลำดับความสำคัญต่ำสุด ดังนั้นถ้ามีการขอ DMA ของอุปกรณ์ภายนอกผ่านทางแชนแนลที่ 1 (DRQ1) และแชนแนลที่ 2 (DRQ2) 8237A-5 ก็จะทำการขอ DMA ให้กับแชนแนลที่ 1 ก่อน จากนั้นเมื่อเสร็จขบวนการ DMA ของแชนแนลที่ 1 แล้ว จึงทำการขอ DMA ให้กับแชนแนลที่ 2

อย่างไรก็ตาม 8237A-5 ยังมีแชนแนลสำหรับการขอ DMA อยู่อีก 1 แชนแนล คือแชนแนลที่ 0 (DRQ0) ซึ่งในความเป็นจริงแล้วแชนแนลนี้จะมีลำดับความสำคัญที่สูงกว่าแชนแนลที่ 1 แต่จะไม่ถูกต่อออกมาของสล็อต เนื่องจาก IBM/PC จะใช้แชนแนลที่ 0 นี้ ในการรีเฟรชหน่วยความจำที่เป็น dynamic RAM

ในการขอ DMA นั้นสัญญาณ DRQ นี้ จะต้องแอกทีฟอยู่ในช่วงระยะเวลาหนึ่งเท่านั้น ถ้าสัญญาณนี้แอกทีฟอยู่นานเกินไป จะทำให้เกิดขบวนการขอ DMA ขึ้นมากกว่า 1 ขบวนการได้สำหรับวงจรที่ขอ DMA โดยทั่วไปจะใช้สัญญาณตอบรับการขอ DMA หรือสัญญาณ DACK ของแชนแนลที่ขอ DMA นั้น ในการรีเซ็ตสัญญาณ DRQ เช่นอุปกรณ์ภายนอกที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) ก็จะคอยตรวจสอบการตอบรับการขอ DMA จากสัญญาณ DACK ของแชนแนล

ที่ 1 (DACK1) เมื่อได้รับสัญญาณจาก DACK1 แล้ว ก็จะรีเซ็ตสัญญาณ DRQ1 (เปลี่ยนจากลอจิก “1” เป็น “0”)

DACK0 – DACK3 (DMA Acknowledge 0-3 ; ขา B19, B17, B26 และ B15) : สัญญาณทั้ง 4 นี้เป็นเอาต์พุตแอกทีฟที่ลอจิก “0” ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้วงจรภายนอกที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์ I/O ที่ขอ DMA กับหน่วยความจำเกิดขึ้นได้โดยตรง (คือไม่ต้องผ่าน 8088) โดยสัญญาณ DACK นี้จะแอกทีฟในแชนแนลใดก็ขึ้นอยู่กับว่าขบวนการ DMA ที่จะเกิดขึ้นนั้น เป็นการตอบสนองต่อการขอ DMA ในแชนแนลใด เช่น ถ้าขบวนการขอ DMA ที่เกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ในแชนแนลที่ 2 (DRQ2) สัญญาณ DACK2 ก็จะมีแอกทีฟ เป็นต้น

ดังที่ได้กล่าวแล้วว่าสัญญาณ DRQ0 นั้น จะไม่ถูกต่อออกมายังขาของสล็อต ดังนั้นวงจรอินเทอร์เฟซจึงไม่สามารถจะขอ DMA ผ่านทางแชนแนล 0 ได้ แต่สัญญาณ DACK0 จะถูกต่อออกมายังสล็อตด้วย (ขา B19) ทั้งนี้ก็เพื่อที่จะแสดงให้วงจรอินเทอร์เฟซต่างๆทราบว่าขบวนการ DMA ที่เกิดขึ้นในช่วงเวลาที่ DACK0 แอกทีฟนั้น เป็นขบวนการที่ใช้สำหรับการรีเฟรชหน่วยความจำที่เป็น Dynamic RAM ซึ่งวงจรอินเทอร์เฟซที่ใช้หน่วยความจำประเภทนี้สามารถจะนำไปใช้ในการรีเฟรช Dynamic RAM ที่อยู่ในวงจรได้

โดยที่การรีเฟรชหน่วยความจำนั้นจะต้องเกิดขึ้นในทุกๆ 15.12 μsec หรือทุกๆ 72 คล็อก ดังนั้นสัญญาณ DACK0 นี้ก็จะแอกทีฟในทุกๆ 15.12 μsec ด้วย

AEN (Address Enable ; ขา A11) : สัญญาณนี้เป็นเอาต์พุตที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอกทีฟ (ลอจิก “1”) นั้น เป็นบัสไซเคิลของขบวนการ DMA

สำหรับเมนบอร์ดของ IBM/PC นั้น จะใช้สัญญาณนี้ในการดิสเอเบิล (Disable) 8288 Bus Controller และจะใช้ดิสเอเบิลพอร์ท I/O ต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้นนี้ ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส และจะทำให้สัญญาณ IOR หรือ IOW แอกทีฟด้วยดังนั้นถ้าไม่ทำการดิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็อาจจะทำให้พอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรส (ซึ่งเป็นแอดเดรสของหน่วยความจำ) นั้นทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูล ทำให้เกิดความผิดพลาดขึ้นได้

T/C (Terminal Count ; ขา B27) : สัญญาณนี้จะถูกสร้างขึ้นจากการนำเอาสัญญาณเอาต์พุตที่ขา EOP ของ 8237A-5 มากลับลอจิก (โดยใช้เกท Inverter) ทำให้สัญญาณ T/C นี้แอกทีฟที่ลอจิก “1”

สำหรับสัญญาณนี้จะแอกทีฟเมื่อจำนวนไบต์ในการส่งผ่านข้อมูลของขบวนการ DMA ในแชนแนลใดแชนแนลหนึ่ง ครบตามจำนวนที่กำหนดไว้ โดยทั่วไปแล้วสัญญาณที่จะถูกใช้ในการสิ้นสุดขบวนการ DMA ที่ทำการส่งผ่านข้อมูลเป็นบล็อก เนื่องจากสัญญาณนี้จะแอกทีฟโดยไม่แสดงว่าเป็นของแชนแนลใด ดังนั้นจึงต้องทำการนำสัญญาณ T/C นี้ผ่านเกท Inverter แล้วนำไป OR กับสัญญาณ DACK เพื่อให้สามารถทราบได้ว่า สัญญาณ T/C ที่เกิดขึ้นนั้นเป็นสัญญาณของแชนแนลใด สำหรับใน แชนแนลที่ 0 นั้น สัญญาณ T/C จะแอกทีฟในช่วงเวลาที่คงที่คือ ทุกๆ 990.804 millisec ซึ่งก็คือช่วงเวลาในการรีเฟรชหน่วยความจำขนาด 64 Kbyte นั้นเอง

5.2 บัซของแหล่งจ่ายไฟของระบบ

+5Vdc (ขา B3 และ B29) : ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulate) $\pm 5\%$ คืออยู่ในช่วง +4.75 ถึง +5.25 Vdc

+12Vdc (ขา B9) : ขานี้ต่อกับแหล่งจ่ายไฟ DC +12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulate) $\pm 5\%$ คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

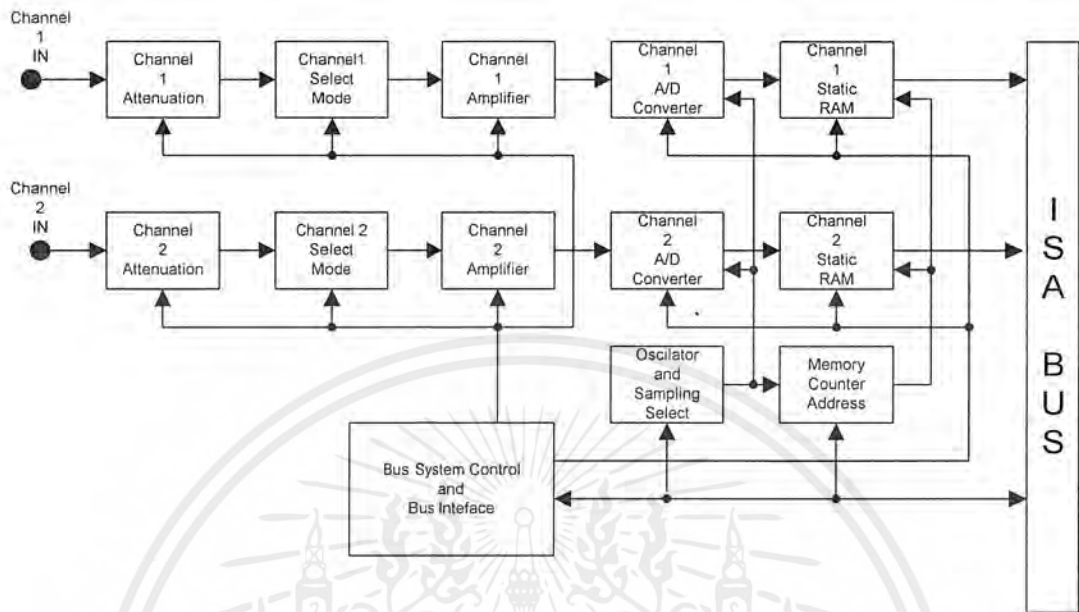
-5Vdc (ขา B5) : ขานี้ต่อกับแหล่งจ่ายไฟ DC -5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulate) $\pm 10\%$ คืออยู่ในช่วง -5.5 ถึง -4.5 Vdc

-12Vdc (ขา B7) : ขานี้ต่อกับแหล่งจ่ายไฟ DC -12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulate) $\pm 10\%$ คืออยู่ในช่วง -13.2 ถึง -10.8 Vdc

GND (ขา B1, B10 และ B31) : ขาทั้งสามนี้จะต่อเข้ากับกราวด์ (Ground) ของระบบ

บทที่ 6

หลักการออกแบบวงจรและโปรแกรมควบคุมการทำงาน

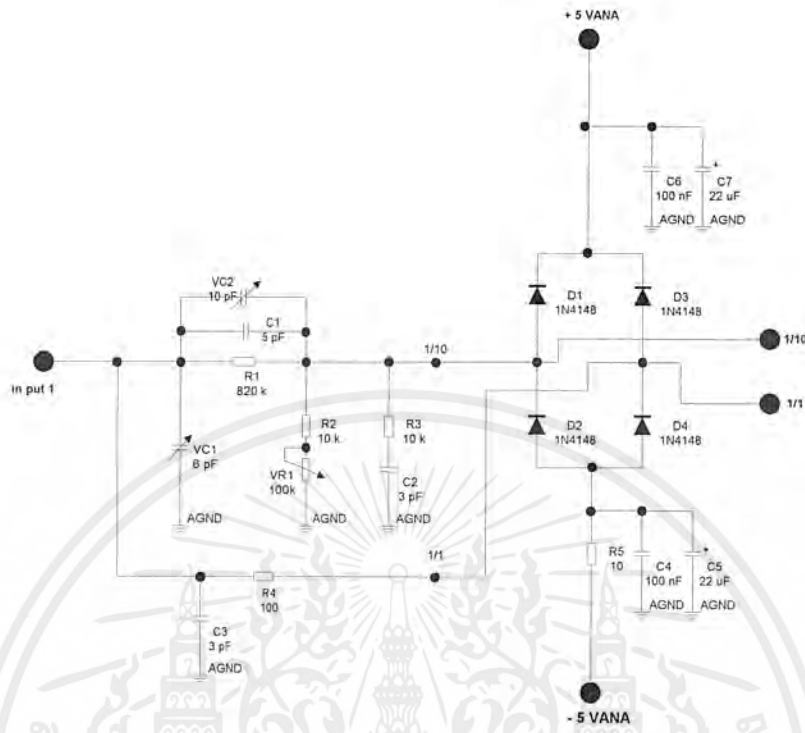


รูปที่ 6.1 แสดงบล็อกโคอะแกรมของดิจิตอลสโคปเรจิสซิงชิฟต์ รีจิสเตอร์

จากบล็อกโคอะแกรม สัญญาณอินพุตที่เข้ามาของแต่ละช่อง จะถูกส่งเข้าสู่ วงจร Attenuator เพื่อลดทอนแรงดันสูงๆ ให้มีค่าขนาดสัญญาณลดลง วงจร Select Mode จะทำการเลือกที่จะทำการวัดใน Mode AC, DC หรือ Ground สัญญาณที่ผ่านการ Select Mode จะถูกส่งเข้าไปให้วงจร Amplifier เพื่อขยายสัญญาณให้มีขนาดเพียงพอที่จะทำการ Sampling ได้ หลังจากผ่านวงจร Amplifier แล้ว ก็จะถูกส่งเข้าสู่ วงจร A/D Converter ข้อมูลที่ได้จากการ Converter จะถูกส่งเข้าไปเก็บไว้ในหน่วยความจำ

ส่วนบล็อกโคอะแกรม Oscillator and sampling จะทำหน้าที่ผลิตความถี่ และเลือกความถี่ที่จะใช้ในการ sampling ส่วน Counter Address Memory จะนำความถี่ที่ใช้ในการ sampling มาเป็น Counter ให้แก่แอดเดรสของ SRAM ส่วน Bus System Control & Bus Interface จะทำหน้าที่ควบคุมส่วนต่างๆ และเชื่อมต่อกับคอมพิวเตอร์ต่อไป

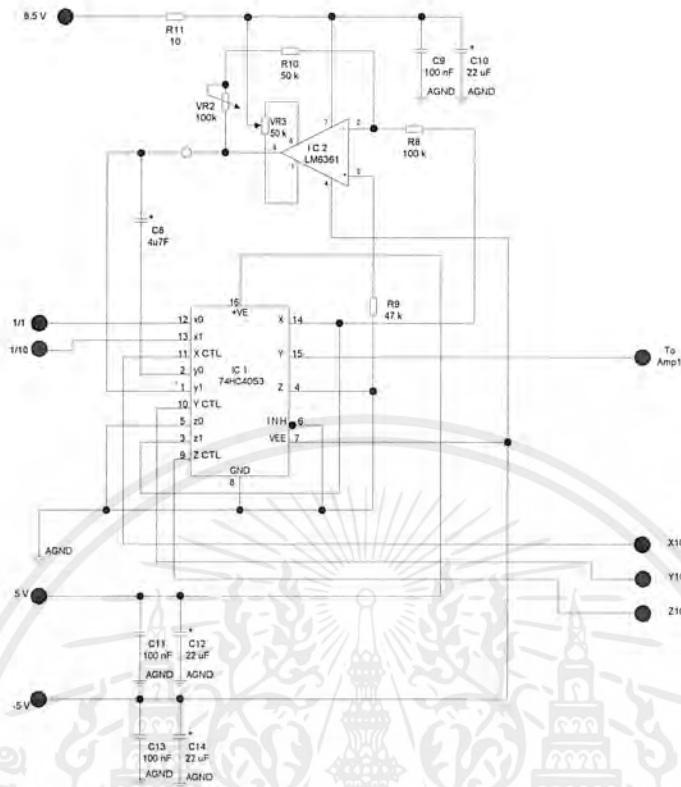
6.1 วงจรลดทอนสัญญาณ (Attenuator)



รูปที่ 6.2 วงจรลดทอนสัญญาณ

เนื่องจากสัญญาณที่เข้ามามีขนาดที่แตกต่างกัน ดังนั้นจะต้องปรับระดับสัญญาณให้เหมาะสมก่อนที่จะส่งไปยังภาคอื่นๆต่อ ซึ่งการลดทอนสัญญาณจะใช้หลักการแบ่งแรงดัน (Voltage Divider) และการลดทอนสัญญาณจะต้องเป็นอิสระกับความถี่ ซึ่งจะต้องใช้วงจรลดทอนแบบชดเชยความถี่ (Compensated Attenuator) ซึ่งจะเป็นวงจร อาซีโวลท์เตจดีไวเดอร์ (RC Voltage Divider) ในวงจรมีการลดทอนทั้งหมด 2 ระดับ คือ 1/1 และ 1/10

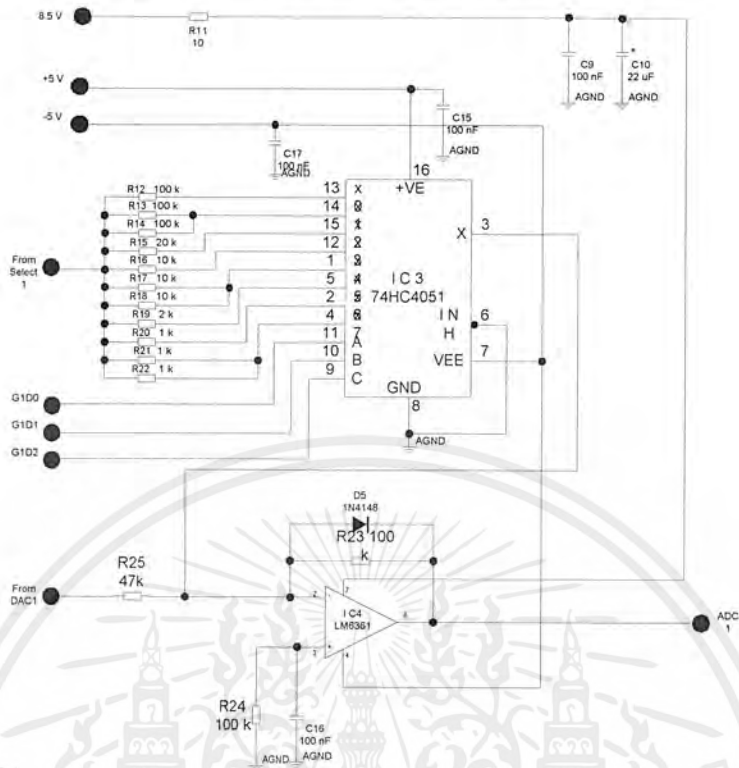
6.2 วงจรบัฟเฟอร์ (Buffer)



รูปที่ 6.3 วงจรบัฟเฟอร์

สัญญาณที่ผ่านการลดทอนแบบชดเชยความถี่แล้ว จะถูกเลือกด้วยอิเล็กทรอนิกส์สวิตช์ และนำสัญญาณที่ได้ผ่านวงจรบัฟเฟอร์ เพื่อเป็นการส่งผ่านสัญญาณจากภาควจรลดทอนซึ่งมีเอาต์พุตอิมพีแดนซ์สูงไปยังภาควจรขยาย โดยใช้อปแอมป์เป็นบัฟเฟอร์ต่อแบบ Inverting และมี VR_3 ใช้ในการปรับออฟเซตให้แก่อปแอมป์

6.3 วงจรขยายสัญญาณแนวตั้ง (Vertical Amplifier)



รูปที่ 6.4 วงจรขยายสัญญาณ

ในการขยายสัญญาณจะมีการขยายทั้งหมด 8 ระดับ ดังตาราง

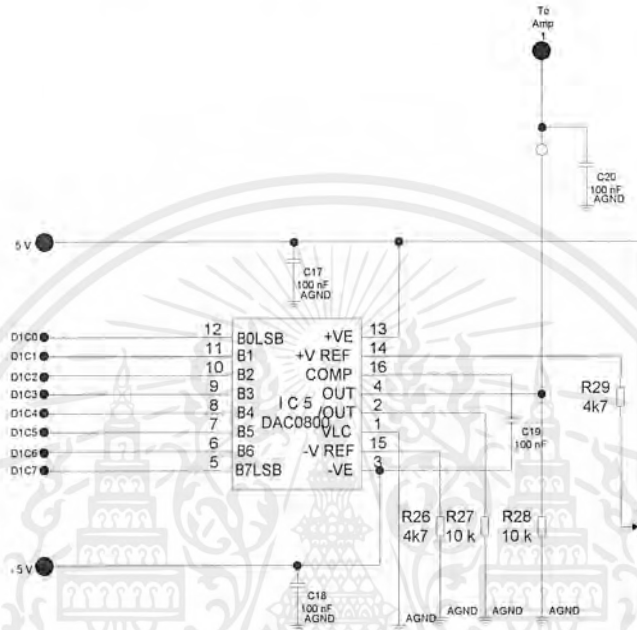
C	B	A	อัตราขยาย (เท่า)
0	0	0	1
0	0	1	2
0	1	0	5
0	1	1	10
1	0	0	20
1	0	1	50
1	1	0	100
1	1	1	200

ตาราง 6.1 อัตราขยายของวงจขยายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้วงจรขยายแบบกลับเฟส (Inverting Amplifier) ในการเลือกอัตราขยายจะใช้ไอซีเบอร์ 74HC4051 ซึ่งเป็นมัลติเพลกซ์ ส่วนไดโอด D5 ใช้ในการจำกัดสัญญาณช่วงบวกไม่ให้ถูกขยายออกออปแอมป์ได้

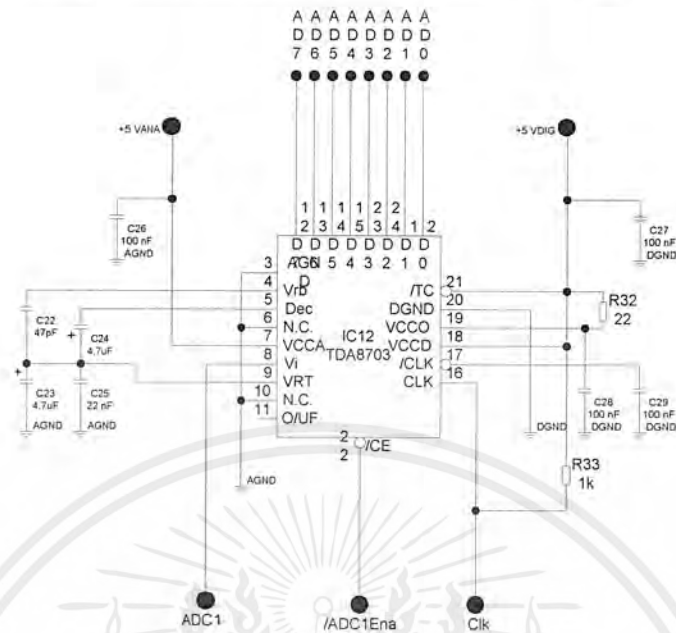
6.4 วงจรยกระดับสัญญาณ (DC Adjust)



รูปที่ 6.5 วงจรยกระดับสัญญาณกระแสตรง

จะใช้ในการยกระดับสัญญาณให้อยู่ในช่วงที่สามารถนำสัญญาณอนาล็อกที่ได้ไปแปลงเป็นสัญญาณดิจิทัลได้

6.5 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Analog to Digital Converter)

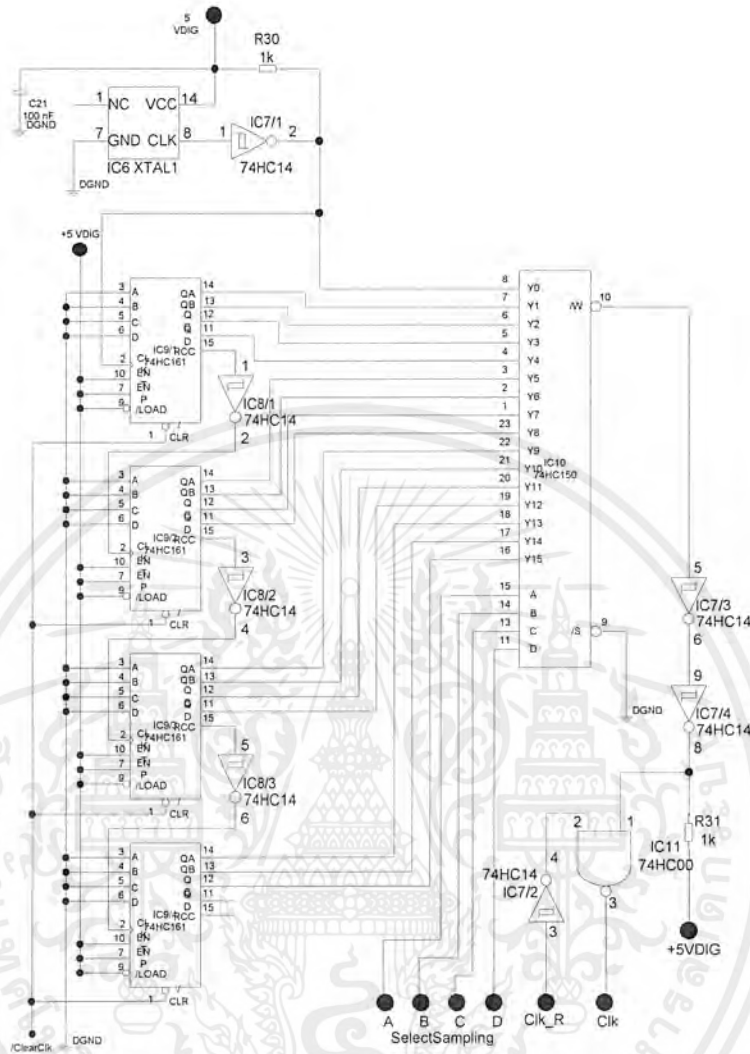


รูปที่ 6.6 วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล

วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล จะใช้ไอซีเบอร์ TDA 8703 ซึ่งเป็นแบบเฟลชมี
ความละเอียด 8 บิต และมีอัตราการสุ่มสูงสุดที่ 40 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 วงจรกำเนิดสัญญาณพิกหลายความถี่ (Multi Frequency Clock Generator)



รูปที่ 6.7 วงจรกำเนิดสัญญาณพิกหลายความถี่

วงจรนี้ทำหน้าที่ในการสร้างสัญญาณพิกสำหรับการใช้ในการทำงานของวงจรทั้งหมดในวงจรนี้จะประกอบด้วย ออสซิลเลเตอร์ความถี่ 40 MHz วงจรนับใช้ไอซีเบอร์ 74HC161 4ตัว ใช้เป็นวงจรหารความถี่เพื่อให้เกิดความถี่ต่างๆซึ่งในวงจรนี้จะได้ความถี่ 16 ความถี่ด้วยกัน ดังตารางและใช้ไอซีเบอร์ 74HC150 ในการเลือกความถี่ไปใช้งาน

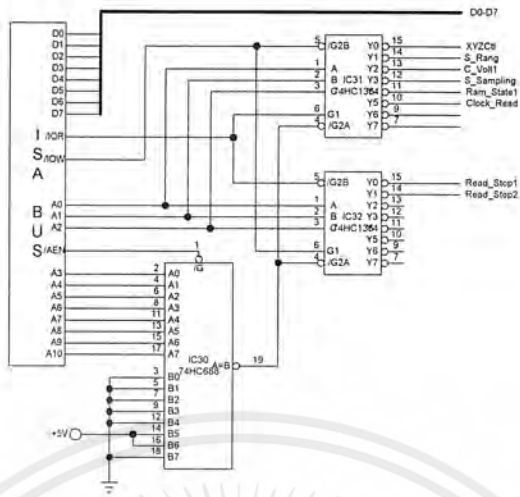
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D	C	B	A	ความถี่ (Hz)
0	0	0	0	40 M
0	0	0	1	20 M
0	0	1	0	10 M
0	0	1	1	5 M
0	1	0	0	2.5 M
0	1	0	1	1.25 M
0	1	1	0	625 k
0	1	1	1	312.5 k
1	0	0	0	156.25 k
1	0	0	1	78.125 k
1	0	1	0	39.062 k
1	0	1	1	19.53 k
1	1	0	0	9.76 k
1	1	0	1	4.88 k
1	1	1	0	2.44 k
1	1	1	1	1.22 k

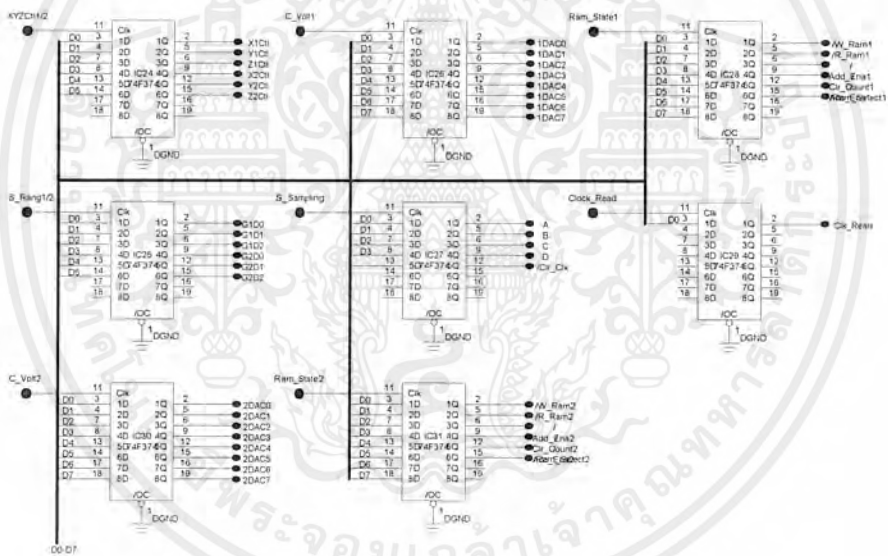
ตารางที่ 6.2 ความถี่ของวงจรถ่ายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 วงจรถอดรหัสตำแหน่งพอร์ต และวงจรเชื่อมต่อพอร์ต



รูปที่ 6.8 วงจรถอดรหัสตำแหน่งพอร์ต

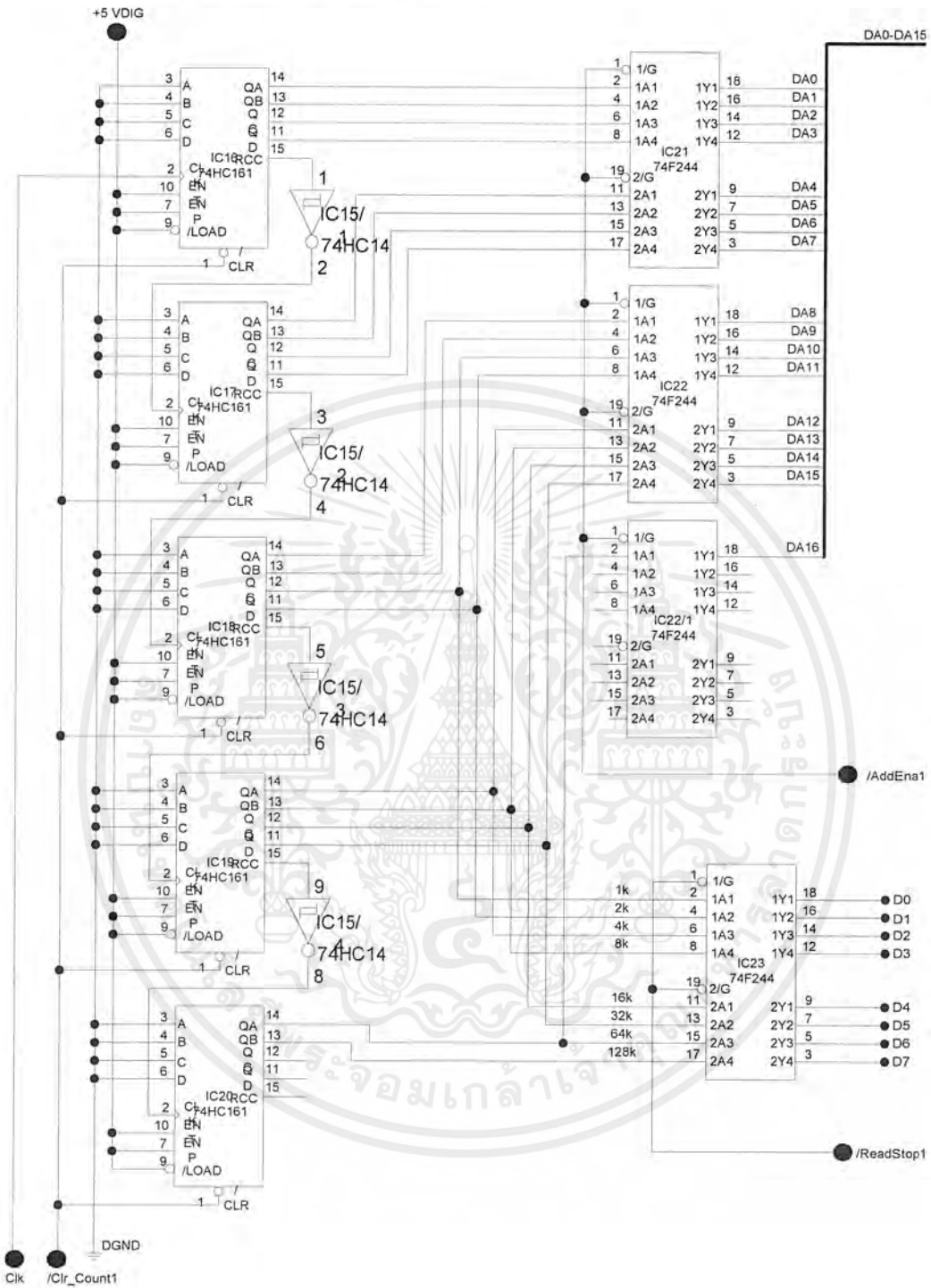


รูปที่ 6.9 วงจรเชื่อมต่อพอร์ต ISA กับวงจรต่างๆ

เป็นวงจรเปรียบเทียบตำแหน่งพอร์ตที่สล็อต ISA กับตำแหน่งพอร์ตที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 วงจรนับตำแหน่งหน่วยความจำ



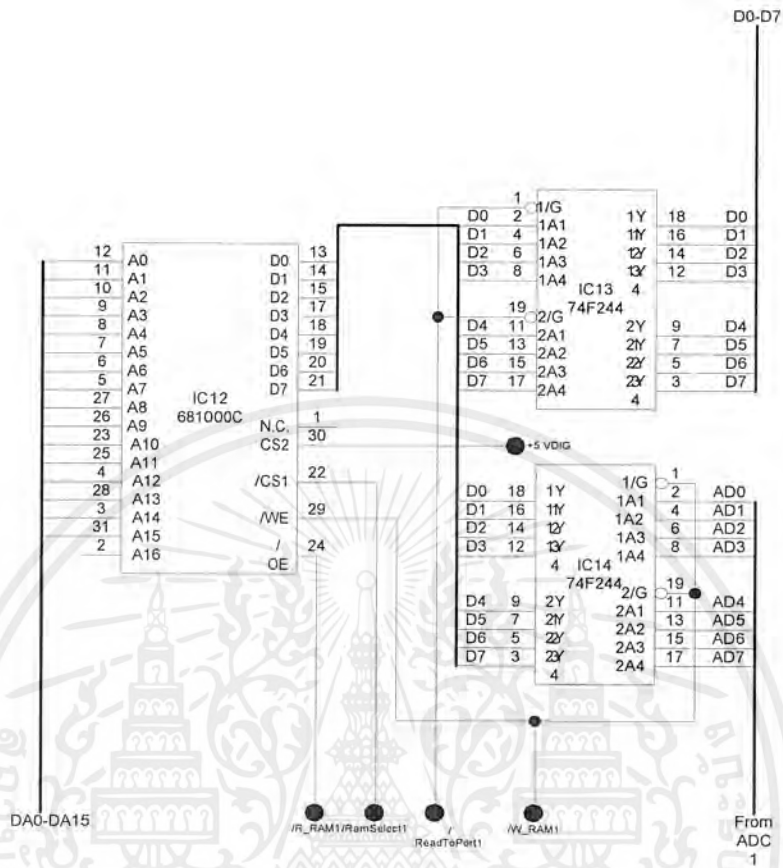
รูปที่ 6.10 วงจรนับตำแหน่งหน่วยความจำ

เป็นวงจรถับตำแหน่งหน่วยความจำ โดยที่เราสามารถเลือกได้ว่าจะใช้หน่วยความจำเท่า

ไหนในการเก็บข้อมูล ในการนับจะใช้ไอซีเบอร์ 74HC161 เป็นตัวนับ ซึ่งจะนับได้สูงสุด 128 kbyte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.9 วงจรหน่วยความจำ

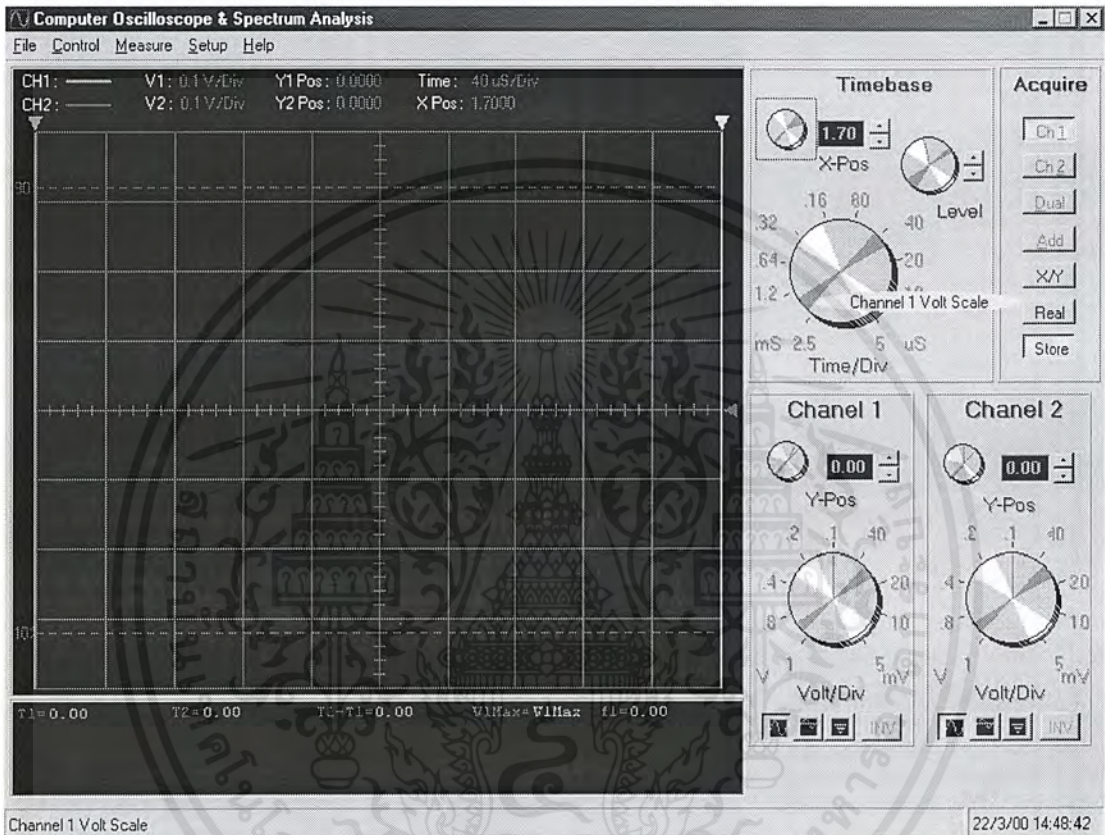


รูปที่ 6.11 วงจรหน่วยความจำ

วงจรหน่วยความจำจะใช้ในการเก็บข้อมูลที่ได้จากการแปลงสัญญาณอนาลอกเป็นดิจิทัลแล้วนำมาเก็บไว้ในหน่วยความจำ ซึ่งหน่วยความจำในวงจรนี้ใช้ไอซีเบอร์ 681000C ซึ่งเป็น SDRAM ขนาด 128 kbyte แต่จะใช้เพียงแค่ 64 kbyte เท่านั้น

6.10 โปรแกรมควบคุมการทำงาน

คอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมควบคุมการทำงานนั้น ใช้ Delphi 4.0 เพราะผู้เขียนมีพื้นฐานการเขียนโปรแกรมมาจาก Pascal และเมื่อมาใช้ Delphi จึงเริ่มต้นใช้งานได้โดยไม่ต้องยุ่งยาก อีกทั้ง Delphi เป็นโปรแกรมที่มีความยืดหยุ่นทางโครงสร้าง และใช้เวลาในการคอมไพล์น้อยกว่าโปรแกรมอื่นๆ



รูปที่ 6.12 โปรแกรมควบคุมการทำงานของการ์ด

จากรูปที่ 6.12 เป็นโปรแกรมควบคุมการทำงานของการ์ด เราสามารถควบคุมการทำงานได้จากการเลือกคอนโทรลที่หน้าจอหลัก หรือจะเลือกการควบคุมจากเมนูก็ได้ คอนโทรลต่างๆ จะมีหน้าที่ๆแตกต่างกัน ซึ่งแต่ละคอนโทรลจะมีชื่อบอกไว้ เมื่อเลื่อนเมาท์เข้าไปในคอนโทรล จะมีชื่อแสดงให้เห็นขึ้นมา และแถบข้างล่างสุดจะมีค่าบอกหน้าที่ของคอนโทรลอันนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.11 หลักการเขียนโปรแกรมควบคุมการทำงาน หลักการเขียนโปรแกรมควบคุมการทำงานของการ์ดสามารถสรุปเป็นแผนงานได้ดังนี้



รูปที่ 6.13 แสดงผังงานของโปรแกรมควบคุม

เมื่อเริ่มต้นโหลดโปรแกรม Computer Oscilloscope and Spectrum Analysis โปรแกรมจะทำการโหลดค่าซึ่งเป็นค่าสถานะเริ่มต้นของสโคป เช่น Volt/Div, Time/Div, Xpos, Ypos ฯลฯ ออกมาเก็บไว้ และแสดงผลออกมาที่ตัวควบคุมต่างๆ จากนั้นจะรอรับคำสั่งจากผู้ใช้งานและปฏิบัติตามคำสั่งต่างๆ ซึ่งจะอธิบายในหัวข้อต่อไป และเมื่อผู้ใช้งานทำการปิดโปรแกรม ตัวโปรแกรมจะทำการเก็บค่าสถานะต่างๆของสโคปเก็บไว้ในไฟล์ PCScope.ini เพื่อจะนำมาใช้เป็นค่าสถานะเริ่มต้น เมื่อทำการเปิดโปรแกรมมาใช้งานในครั้งต่อไป

6.12 การเขียนโปรแกรม Interface กับ ISA-Bus

โครงการานนี้ใช้บัสอินเตอร์เฟสเป็นแบบ ISA-Bus การเขียนโปรแกรมเพื่อติดต่อกับ ISA-Bus โดยใช้ Delphi 4.0 สามารถทำได้หลายวิธี แต่ที่ใช้ในโครงการานนี้จะใช้วิธีการเขียนโปรแกรมแอสเซมบลี(Assembly) ใน Delphi เพื่อติดต่อกับระบบ ISA-Bus ซึ่งมีซอร์สโค้ดและวิธีใช้งานดังนี้

โปรแกรมสำหรับส่งค่าข้อมูลออกพอร์ตซึ่งค่าข้อมูลเป็นแบบ 8 bit

```
procedure OutPort(Address,Data:Word);Assembly;
```

```
begin
```

```
asm
```

```
push dx
```

```
push ax
```

```
mov ax,data
```

```
mov dx,address
```

```
out dx,ax
```

```
pop ax
```

```
pop dx
```

```
end;
```

```
end;
```

วิธีการใช้งาน เช่น ต้องการส่งข้อมูล 80H ซึ่งเป็นข้อมูล 8 bit ออกที่ พอร์ต 300H

```
Var
```

```
AddressData:Word;
```

```
DataSend:Byte;
```

```
Begin
```

```
AddressData := $300;
```

```
DataSend := $80;
```

```
OutPort(AddressData,DataSend);
```

```
End;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับอ่านค่าข้อมูลเข้ามาจากพอร์ตซึ่งค่าข้อมูลเป็นแบบ 8 bit

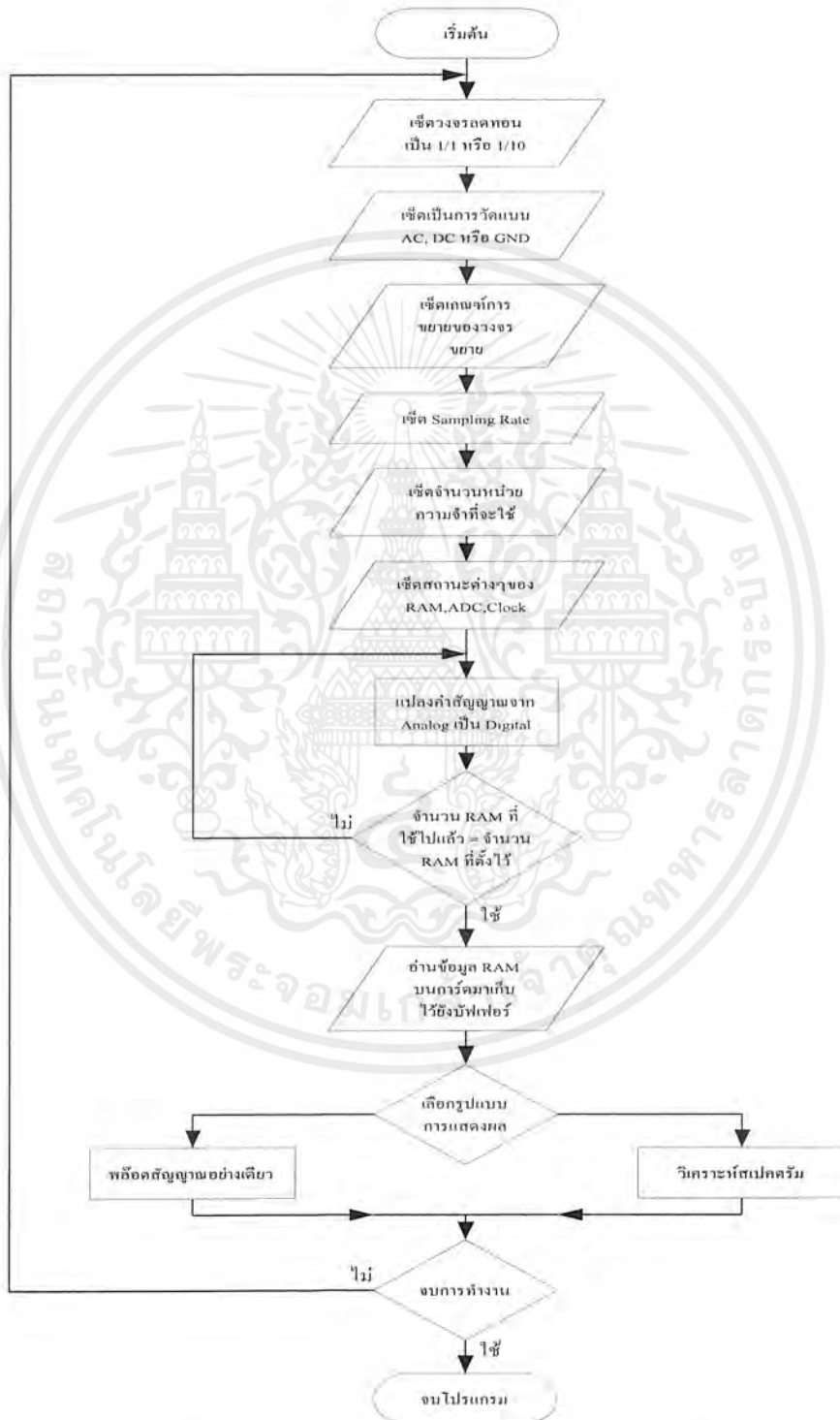
```
function InPort(Address:Word):byte;Assembly;
begin
  asm
    push dx
    mov dx,Address
    in al,dx
    mov result,al
    pop dx
  end;
end;
```

วิธีใช้งาน เช่น ต้องการรับค่าข้อมูล 8 bit จากพอร์ต 300H มาเก็บไว้ใน Data ซึ่ง Data เป็นตัวแปรชนิด byte

```
Var
  AddressData:Word;
  Data:Byte;
Begin
  AddressData := $300;
  Data := InPort(AddressData);
End;
```

6.13 การควบคุมและการทำงานของฮาร์ดแวร์

ในส่วนของการควบคุมฮาร์ดแวร์, การเซ็ตค่าต่างๆ เช่น ค่าสถานะ, ค่าข้อมูลควบคุม ชนิดของการวัดตลอดจนการอ่านค่าข้อมูลเก็บไว้ในหน่วยความจำ การแสดงผลสัญญาณ และการวิเคราะห์ สเปกตรัมสามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 6.14 แสดงผังงานของการควบคุมฮาร์ดแวร์ เมื่อทำการวัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลสามารถแสดงผลได้ 2 แบบ คือแสดงผลเป็นรูปสัญญาณที่วัดได้ หรือแสดงผล
ข้อมูลที่ได้จากการวิเคราะห์สเปกตรัม และสามารถนำค่าที่วัดหรือวิเคราะห์ได้ แปลงออกมาเก็บไว้
เป็น bmp File เพื่อนำไปเก็บไว้ หรือพิมพ์สัญญาณที่วัดได้ออกยังเครื่องพิมพ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทดลอง

วัตถุประสงค์

1. เพื่อเป็นการทดสอบการทำงานของวงจรส่วนต่างๆ
2. เพื่อเป็นการปรับปรุงและแก้ไขข้อบกพร่องของวงจรต่างๆ
3. สร้างฮาร์ดแวร์ และ ซอฟต์แวร์ สำหรับใช้ในการประยุกต์คอมพิวเตอร์ให้สามารถทำงานเป็นสตอเรจออสซิลโลสโคป (Storage Oscilloscope) ได้
4. เปรียบเทียบการทำงานของเครื่องที่สร้างขึ้นมากับเครื่องที่มีขายในท้องตลาด

จะแบ่งการทดลองเป็น 3 ส่วน คือ

- การทดลองวงจรส่วนต่างๆ
- การติดต่อระหว่างการ์ดกับคอมพิวเตอร์
- การวัดสัญญาณเชิงความถี่และวิเคราะห์สเปกตรัมโดยแสดงผลการทดลองมาด้วย

เพื่อให้ง่ายในการสร้างเป็นวงจรใช้งานจริงๆจึงต้องทำการทดลองวงจรที่คิดขึ้นมาว่าสามารถทำงานได้ตามที่ออกแบบไว้หรือไม่ โดยการทดลองวงจรในแบบทดลองวงจรที่ละวงจรและวัดสัญญาณต่างๆ โดยเริ่มด้วยวงจรลดทอน, วงจรบัฟเฟอร์, วงจรยกระดับสัญญาณ, วงจรกำเนิดสัญญาณนาฬิกาหลายความถี่, วงจรถอดรหัส, วงจรนับ, วงจรหน่วยความจำ และ วงจรเอทูดิ

ลำดับขั้นการทดลอง

7.1 การทดลองวงจรลดทอนสัญญาณ

7.1.1 ต่อวงจรดังรูปที่ 6.2 ซึ่งเป็นวงจรลดทอนสัญญาณและป้องกันแรงดันอินพุทเกิน $12 V_{pp}$

7.1.2 ป้อนแรงดัน +5 โวลต์, -5 โวลต์ และ กราวด์นี้ให้แก่วงจร

7.1.3 ป้อนแรงดันดีซี 10 โวลต์ที่อินพุท แล้วทำการปรับค่า VR1 เพื่อให้แรงดันที่ 1/10 มีค่าเป็น 0.1 โวลต์ดีซี

7.1.4 ป้อนสัญญาณ $10 V_{pp}$ ความถี่ 500 KHz แล้วทำการปรับค่า VC2 เพื่อให้แรงดันที่ 1/10 มีค่า $0.1 V_{pp}$

สรุปผลการทดลองที่ 7.1

จากการทดลองวงจรลทอนแรงดันและป้องกันแรงดันเกิน $12 V_{pp}$ VR1 จะใช้ในการปรับค่าการลทอนแรงดันดิซี ให้เป็น 1/10 เท่า ส่วน VC2 จะใช้ในการปรับค่าการลทอนสัญญาณที่ความถี่ต่างๆ ให้เป็น 1/10 เท่า ส่วนไดโอด D1-D4 จะใช้ในการป้องกันระดับแรงดันที่จุด 1/10 กับจุด 1/1 ไม่ให้เกิน $12 V_{pp}$ สัญญาณที่วัดได้ที่จุด 1/10 จะเป็นสัญญาณที่ผ่านการลทอนสัญญาณลง 1/10 เท่า และที่จุด 1/1 จะเป็นสัญญาณที่ไม่ได้ผ่านการลทอนสัญญาณ R4 จะใช้ในการกลับปลี้งสัญญาณจากอินพุทไปยังวงจรป้องกันแรงดันเกิน

7.2 การทดลองวงจรมัลติเพลอร์และวงจรวัดซ์เลือกสัญญาณ

7.2.1 ต่อดังรูปที่ 6.3 ซึ่งเป็นวงจรมัลติเพลอร์และวงจรวัดซ์เลือกสัญญาณ

7.2.2 ป้อนไฟเลี้ยงให้แก่วงจร

7.2.3 ทำการปรับแรงดันออฟเซ็ทของออปแอมป์ โดยการต่อขา 14 ของไอซี 74HC4053 ลงกราวด์ แล้วทำการปรับ VR3 ให้มีค่าแรงดันที่ขา 6 ของ LM6361 มีค่าเป็น 0 โวลท์

สรุปผลการทดลองที่ 7.2

จากการทดลองจะไม่สามารถปรับแรงดันออฟเซ็ทเป็น 0 โวลท์ได้ จะได้แรงดันดิซีค่าหนึ่งขา Xcd จะเป็นขาควมคุมการเลือกสัญญาณ ถ้า Xcd เป็น "1" จะเลือกสัญญาณจาก 1/10 ออกที่ขา X และถ้า Xcd เป็น "0" จะเลือกสัญญาณจาก 1/1 ออกที่ขา X ส่วน Ycd จะใช้ในการเลือกวัดแบบ เอซี คัปปลี้ง หรือ ดิซีคัปปลี้ง ถ้า Ycd เป็น "1" จะเป็นการวัดแบบดิซีคัปปลี้ง จะให้สัญญาณที่มีม็องค์ประกอบกระแสตรงผ่านเข้ามาด้วย และถ้า Ycd เป็น "0" จะเป็นการวัดแบบเอซีคัปปลี้ง จะให้สัญญาณที่มีม็องค์ประกอบกระแสสลับผ่านเข้ามาได้อย่างเดียว ส่วน Zcd ถ้า เป็น "1" จะนำสัญญาณต่อลงกราวด์ การทำงานของวงจรในส่วนนี้ไม่มีปัญหาแต่อย่างใด

7.3 การทดลองวงจรยกระดับสัญญาณ

7.3.1 ต่อดังรูปที่ 6.5 ซึ่งเป็นวงจรยกระดับสัญญาณ

7.3.2 ป้อนแรงดันไฟเลี้ยง +5 โวลท์ และ -5 โวลท์ กับกราวด์ให้แก่วงจร

สรุปผลการทดลองที่ 7.3

แรงดันสูงสุดที่ขา 4 ของไอซี DAC 0800 จะขึ้นอยู่กับแรงดัน V_{ref} และเราสามารถปรับค่าแรงดันที่ขา 4 ของไอซี DAC 0800 ได้ด้วยการเปลี่ยนแปลงข้อมูลที่ขา 5 ถึง ขา 12 ซึ่งเป็นข้อมูลแบบ 8 bit แรงดันที่ได้จาก DAC 0800 จะใช้ในการยกระดับสัญญาณที่เราจะทำการวัดให้อยู่ในช่วงที่สามารถแปลงเป็นสัญญาณดิจิตอลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 การทดลองวงจรกำเนิดสัญญาณนาฬิกาหลายความถี่

7.4.1 ต่ วงจรดังรูปที่ 6.7 ซึ่งเป็นวงจรกำเนิดสัญญาณนาฬิกาหลายความถี่

7.4.2 จ่ายไฟ +5 โวลต์ และกราวด์ให้แก่วงจร

7.4.3 วัดสัญญาณเอาต์พุตที่ IC9/1, IC9/2, IC9/3 และ IC9/4 จะได้ความถี่สัญญาณนาฬิกาตามต้องการ

7.4.4 ใช้ Clk_Read เป็น “0” แล้วทำการทดลองและวัดสัญญาณเอาต์พุตที่ Clk บันทึกผลการทดลอง

D	C	B	A	ความถี่ (Hz)
0	0	0	0	40 M
0	0	0	1	20 M
0	0	1	0	10 M
0	0	1	1	5 M
0	1	0	0	2.5 M
0	1	0	1	1.25 M
0	1	1	0	625 k
0	1	1	1	312.5 k
1	0	0	0	156.25 k
1	0	0	1	78.125 k
1	0	1	0	39.062 k
1	0	1	1	19.53 k
1	1	0	0	9.76 k
1	1	0	1	4.88 k
1	1	1	0	2.44 k
1	1	1	1	1.22 k

ตารางที่ 7.1 ผลการทดลองที่ 7.4

สรุปผลการทดลองที่ 7.4

จากการทดลองวงจรกำเนิดสัญญาณนาฬิกาสามารถผลิตสัญญาณนาฬิกาได้ทั้งหมด 16 ความถี่ โดยความถี่สูงสุดประมาณ 40 MHz และต่ำสุดประมาณ 1.22 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 การทดลองวงจรถอดรหัสตำแหน่งพอร์ต

7.5.1 ตัวอย่างรูปที่ 6.8 ซึ่งเป็นวงจรถอดรหัสตำแหน่งพอร์ต

IOW	IOR	ABN	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Logic "0"
0	1	0	0	1	1	0	0	0	0	0	0	0	0	Y0
0	1	0	0	1	1	0	0	0	0	0	0	0	1	Y1
0	1	0	0	1	1	0	0	0	0	0	0	1	0	Y2
0	1	0	0	1	1	0	0	0	0	0	0	1	1	Y3
0	1	0	0	1	1	0	0	0	0	0	1	0	0	Y4
0	1	0	0	1	1	0	0	0	0	0	1	0	1	Y5
0	1	0	0	1	1	0	0	0	0	0	1	1	0	Y6
0	1	0	0	1	1	0	0	0	0	0	1	1	1	Y7

ตารางที่ 7.2 การทดสอบตำแหน่งของพอร์ตที่ใช้ในการเขียนข้อมูล

7.5.4 ป้อนค่าต่างๆตามตารางเพื่อดูว่า IC 32 ให้เอาท์พุทมาอย่างไร

IOW	IOR	ABN	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Logic "0"
1	0	0	0	1	1	0	0	0	0	0	0	0	0	Y0
1	0	0	0	1	1	0	0	0	0	0	0	0	1	Y1
1	0	0	0	1	1	0	0	0	0	0	0	1	0	Y2
1	0	0	0	1	1	0	0	0	0	0	0	1	1	Y3
1	0	0	0	1	1	0	0	0	0	0	1	0	0	Y4
1	0	0	0	1	1	0	0	0	0	0	1	0	1	Y5
1	0	0	0	1	1	0	0	0	0	0	1	1	0	Y6
1	0	0	0	1	1	0	0	0	0	0	1	1	1	Y7

ตารางที่ 7.3 การทดสอบตำแหน่งของพอร์ตที่ใช้ในการอ่านข้อมูล

สรุปผลการทดลองที่ 7.5

จากการทดลองวงจรถอดรหัสตำแหน่งพอร์ตของการอ่านและเขียนข้อมูลได้ตามที่ต้องการ และสามารถนำไปใช้งานได้

7.6 การทดลองวงจรนับตำแหน่งหน่วยความจำ

7.6.1 ตัวอย่างรูปที่ 6.10 ซึ่งเป็นวงจรนับตำแหน่งหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6.2 ป้อนแหล่งจ่ายไฟ +5 โวลต์ และกราวด์ให้แก่วงจร

7.6.3 ทำให้ขา CLR ของ IC16-IC20 เป็น “0” จะ ได้ข้อมูลที่ $Q_A - Q_D$ ของ IC16-IC20 เป็น “0” ทั้งหมด

7.6.4 ทำให้ /CLR_Count1 เป็น logic “1”

7.6.5 ป้อนสัญญาณนาฬิกา 1 ลูก ให้แก่ CLR จะ ได้ Q_A ของ IC16 เป็น “1”

7.6.6 ป้อนสัญญาณนาฬิกาความถี่ 1kHz – 1MHz เข้ามาที่ CLR

7.6.7 จับเอาที่พู่ที่ขา $Q_A - Q_D$ ของ IC16-IC20 จะ ได้สัญญาณที่ความถี่ต่างๆหาร 2^{n+1} เมื่อ n เป็น Side Bit

สรุปผลการทดลองที่ 7.6

จากการทดลองข้างต้นสามารถควบคุมวงจรมอบให้มีการนับตั้งแต่แอดเดรส 0000H ถึง FFFFH และเราสามารถตรวจสอบได้ว่าขณะที่ทำการนับอยู่นั้นกำลังนับถึงแอดเดรสเท่าไร โดยการตรวจสอบจาก IC23

7.7 การทดลองวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

การทดลองหัวข้อนี้จะทดลองแก้ป้อนแรงดันต่ำสุดและแรงดันสูงสุดที่ไอซีสามารถแปลงเป็นข้อมูลดิจิตอลได้

7.7.1 ต่อวงจรดังรูปที่ 6.6 ซึ่งเป็นวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

7.7.2 ป้อนแหล่งจ่ายไฟ +5 โวลต์ และกราวด์ให้แก่วงจร

7.7.3 ป้อนสัญญาณนาฬิกาความถี่ 1kHz ให้แก่ขา 16 ของ IC 12

7.7.4 ให้ /ADCIEna เป็น “0”

7.7.5 ป้อนแรงดันดีซี 2.4 โวลต์ ให้แก่ขา 8 ของ IC 12

7.7.6 เพิ่มค่าแรงดันดีซี ขึ้นเรื่อยๆจนค่าเอาต์พุต D0 – D7 มีค่าเป็น FFH จะ ได้แรงดันดีซีที่อินพุตประมาณ 3.25 โวลต์

7.7.7 ลดค่าแรงดันอินพุตลงจนทำให้ D0 – D7 มีค่าเป็น 00H ซึ่งจะได้แรงดันที่อินพุตประมาณ 1.5 โวลต์

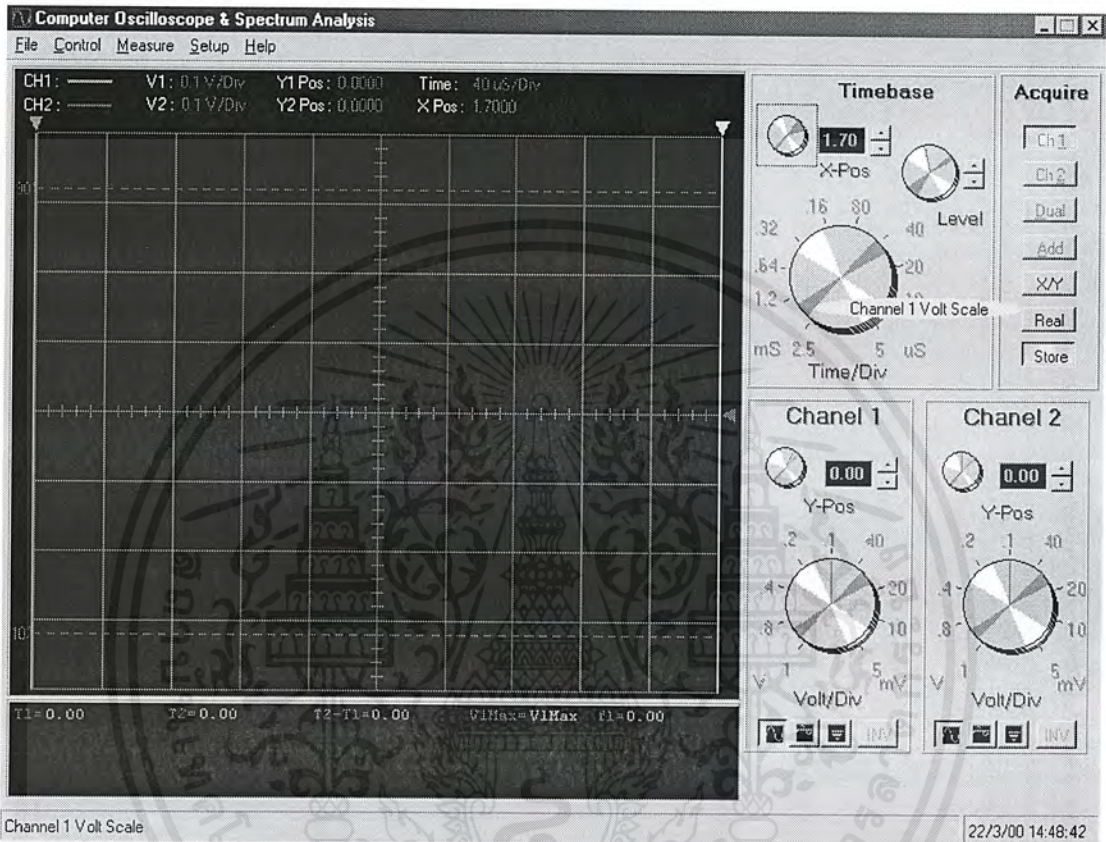
สรุปผลการทดลองที่ 7.7

จากการทดลองเป็นการหาค่าแรงดันสูงสุดและต่ำสุดที่ไอซี 12 สามารถแปลงค่าออกมาเป็นข้อมูลดิจิตอลได้ ซึ่งถ้าแรงดันอินพุตมีค่ามากกว่า 3.25 โวลต์ ค่าเอาต์พุตที่ได้จะเป็น FFH และถ้าแรงดันอินพุตมีค่าต่ำกว่า 1.5 โวลต์จะได้เอาต์พุตเป็น 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8 การทดลองวัดและวิเคราะห์สัญญาณ

จะใช้โปรแกรมในภาคผนวกพร้อมกับวงจรที่สร้างขึ้นมาทำการวัดสัญญาณที่ความถี่ต่างๆ เมื่อทำการรัน โปรแกรมจากไฟล์ PCScope.EXE จะปรากฏดังภาพ



รูปที่ 7.1 แสดงโปรแกรมหลักในการใช้งาน

การทดลองจะแบ่งเป็น 2 ส่วน คือ

7.8.1 ทดลองวัดสัญญาณที่ความถี่ต่างๆ

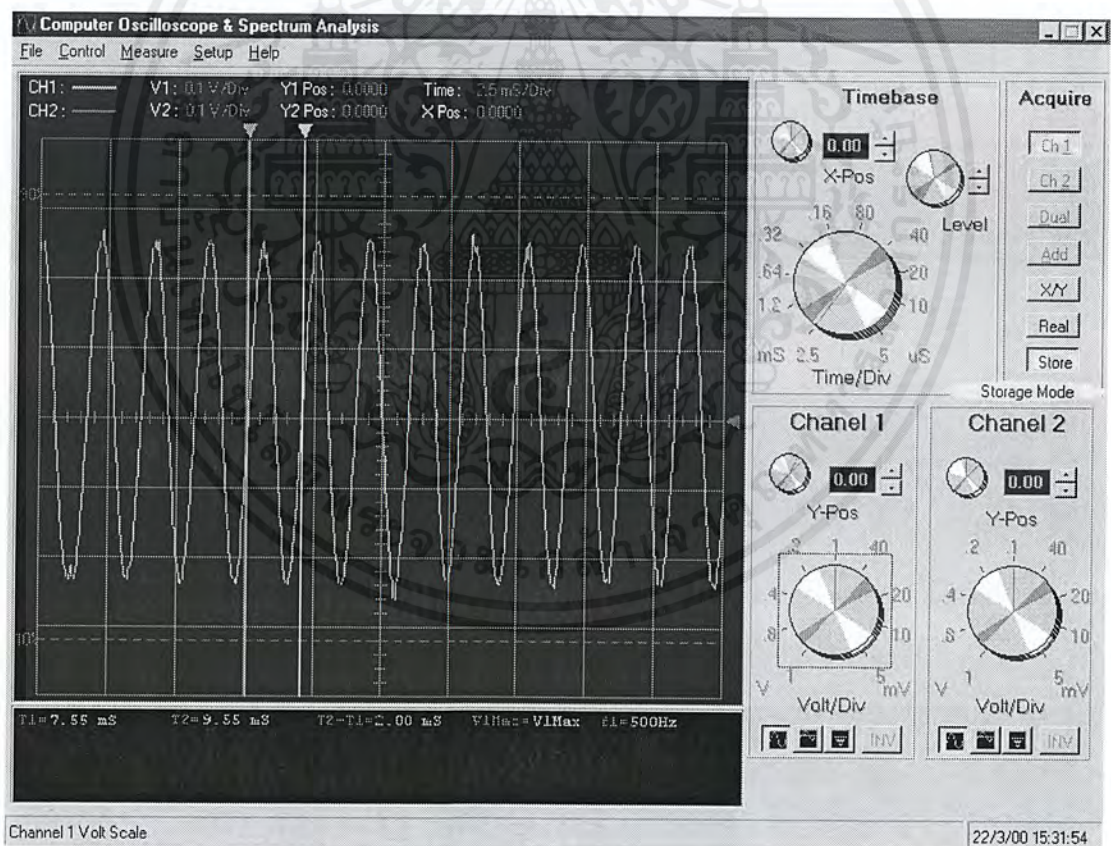
7.8.2 ทดลองวิเคราะห์สเปกตรัมของสัญญาณที่ความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.1 ทดลองวัดสัญญาณที่มีความถี่ต่างๆ

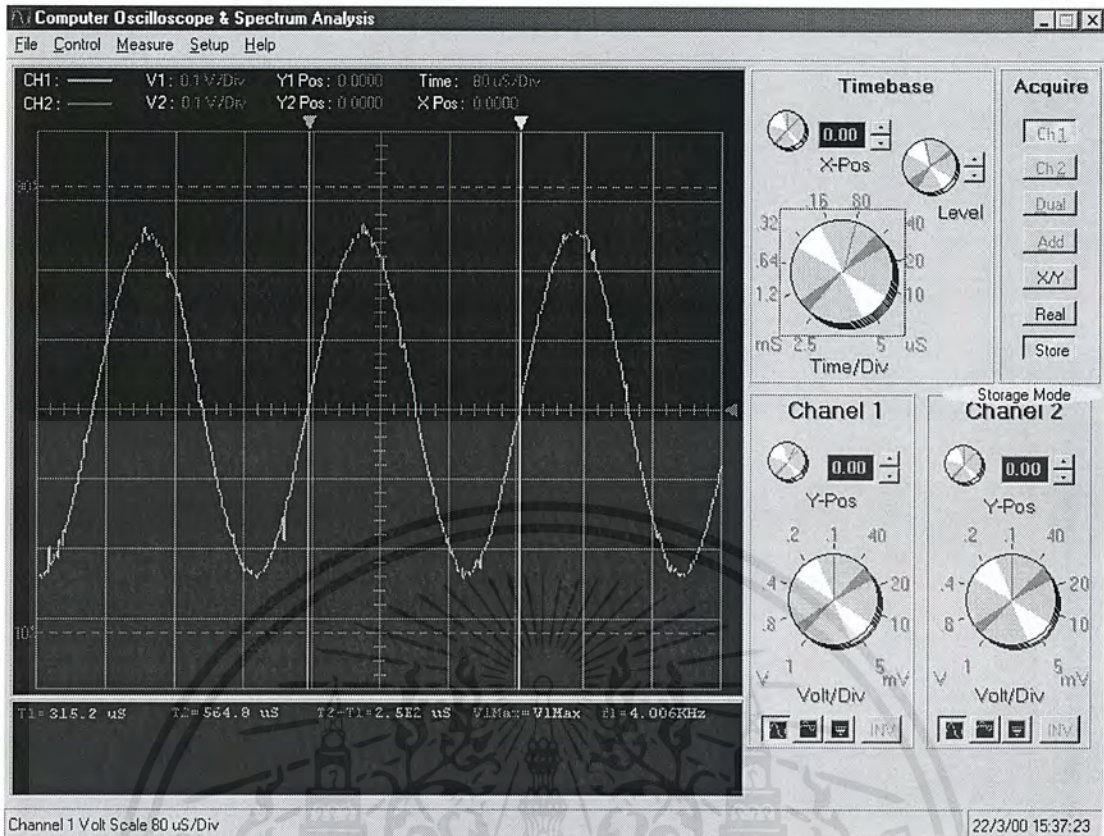
- ผลการทดลองวัดสัญญาณ sine

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการการ Sampling
รูปที่ 7.2	500	0.5	128 kbyte	5 MHz
รูปที่ 7.3	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.4	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.5	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.6	100 k	0.5	4 kbyte	10 MHz
รูปที่ 7.7	300 k	0.5	4 kbyte	10 MHz
รูปที่ 7.8	500 k	0.5	4 kbyte	10 MHz

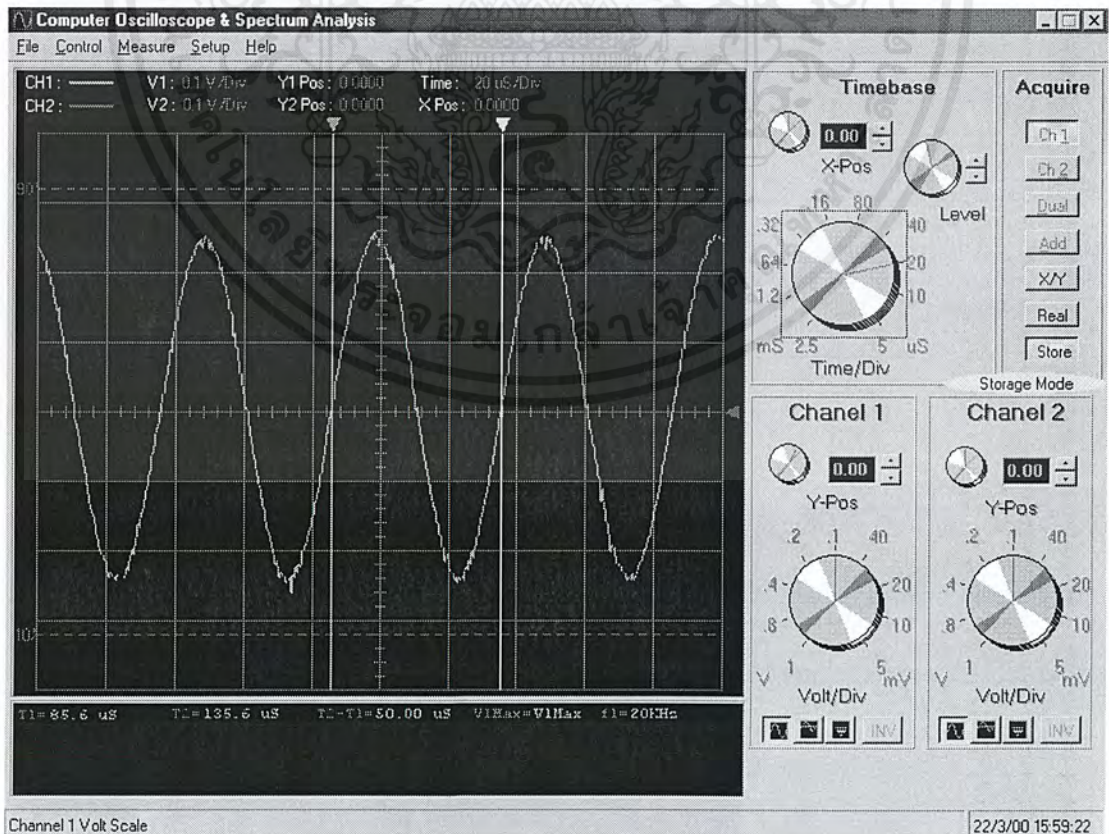


รูปที่ 7.2 ผลการวัดสัญญาณซายน์ความถี่ 500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

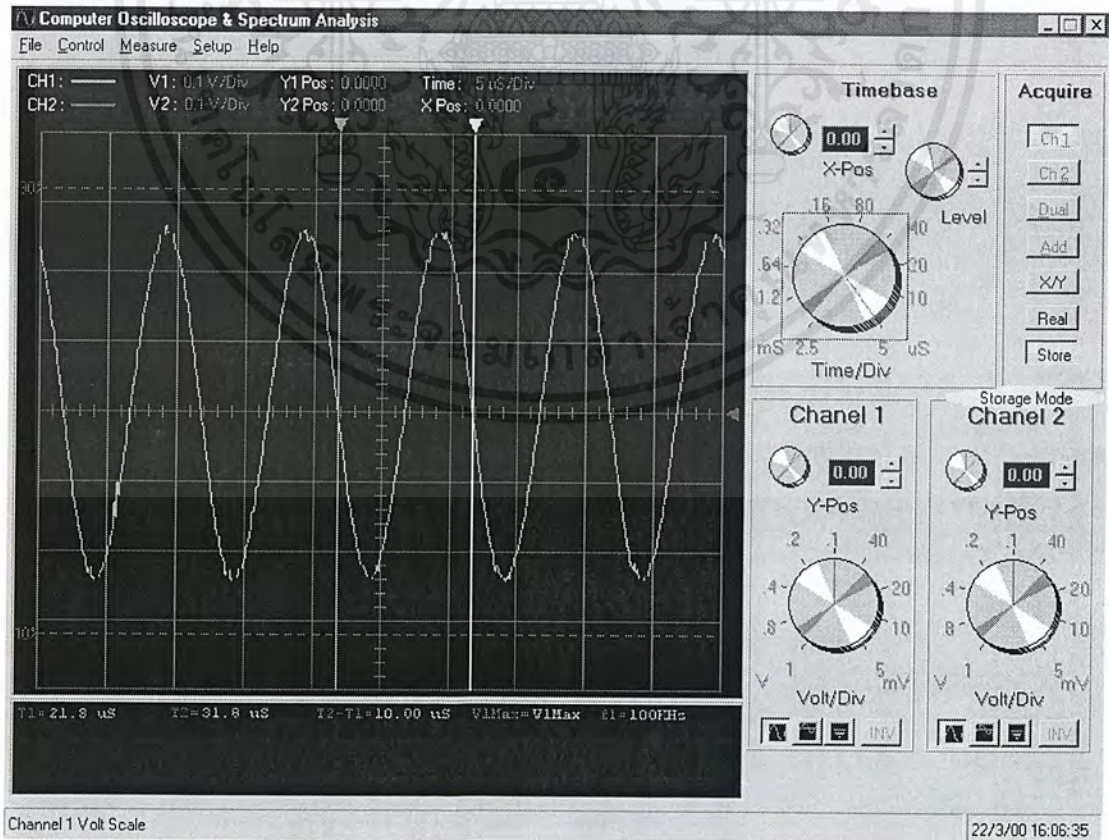
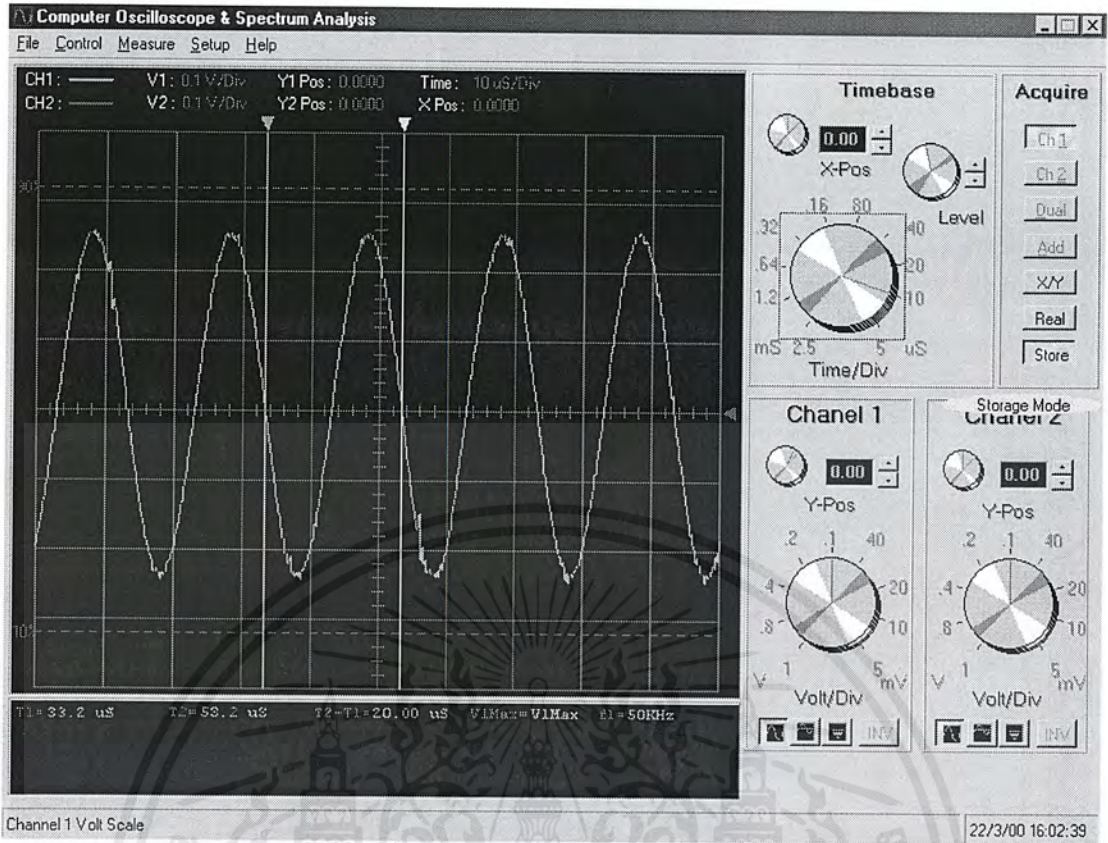


รูปที่ 7.3 ผลการวัดสัญญาณไซน์ความถี่ 4 kHz

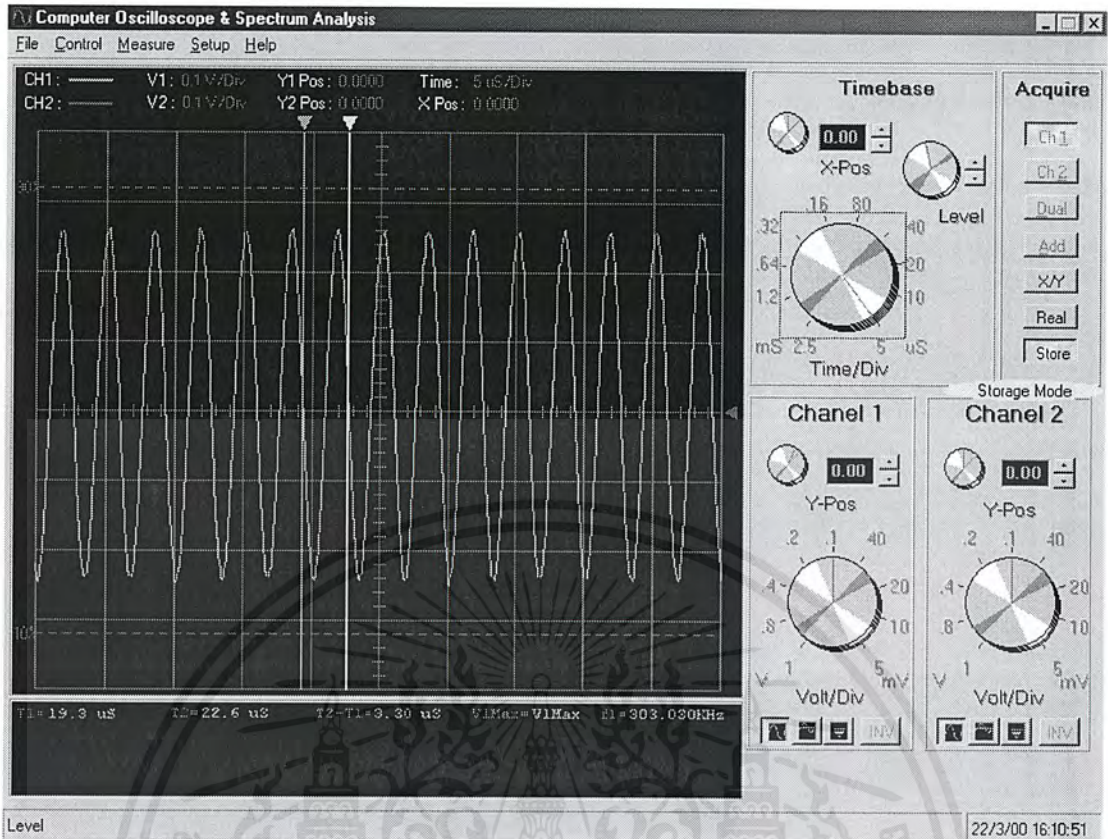


รูปที่ 7.4 ผลการวัดสัญญาณไซน์ความถี่ 20 kHz

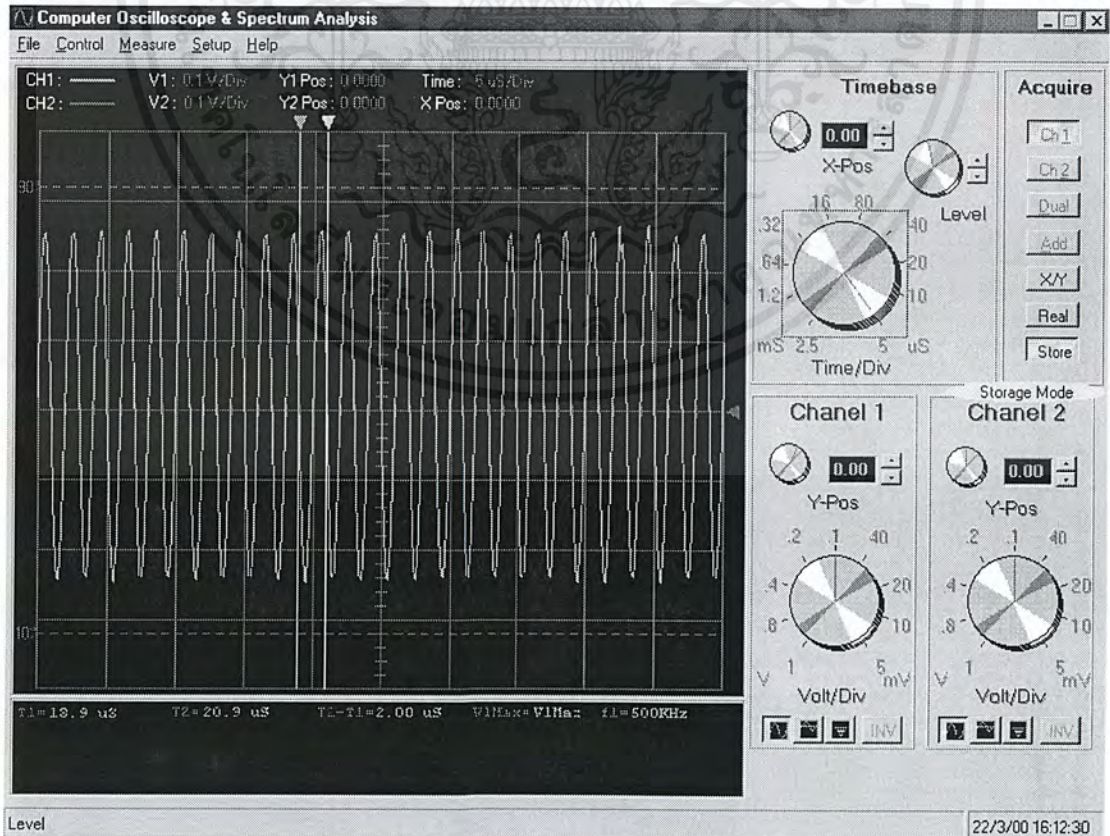
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.7 ผลการวัดสัญญาณชานน์ความถี่ 300 kHz

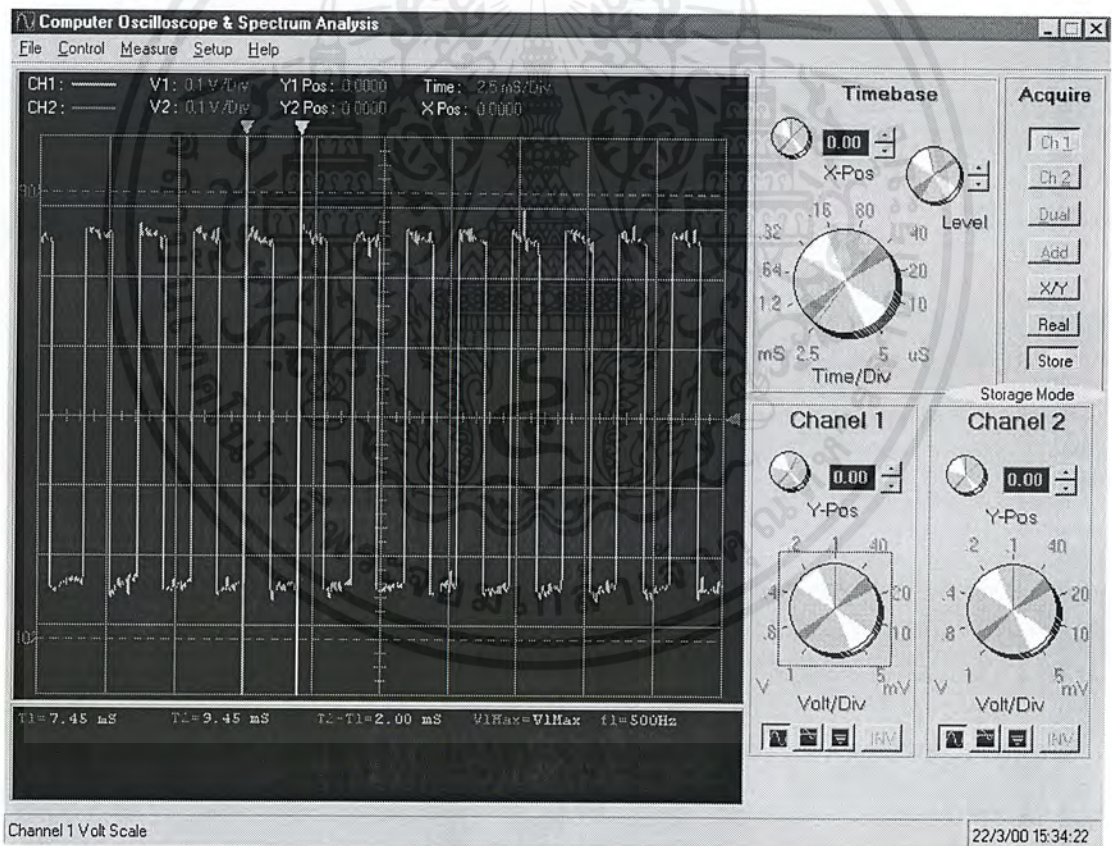


รูปที่ 7.8 ผลการวัดสัญญาณชานน์ความถี่ 500 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

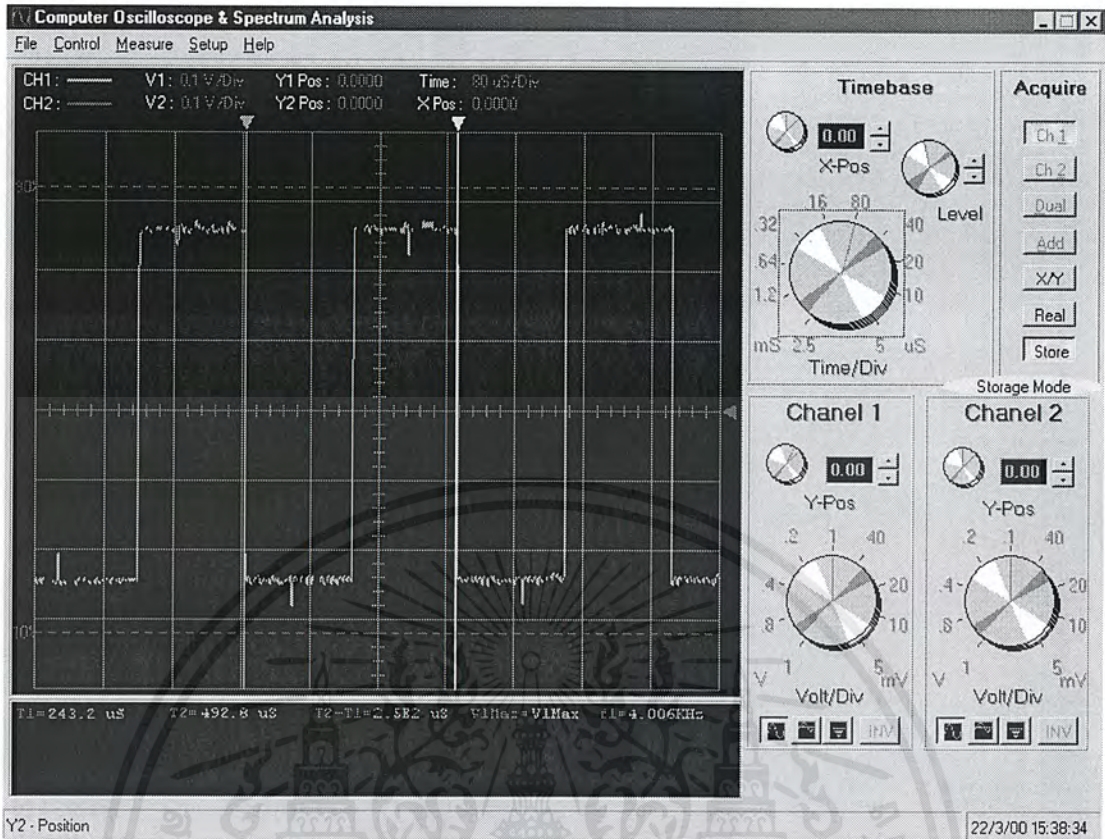
- ผลการทดลองวัดสัญญาณ square

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการ Sampling
รูปที่ 7.9	500	0.5	128 kbyte	5 MHz
รูปที่ 7.10	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.11	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.12	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.13	100 k	0.5	4 kbyte	10 MHz

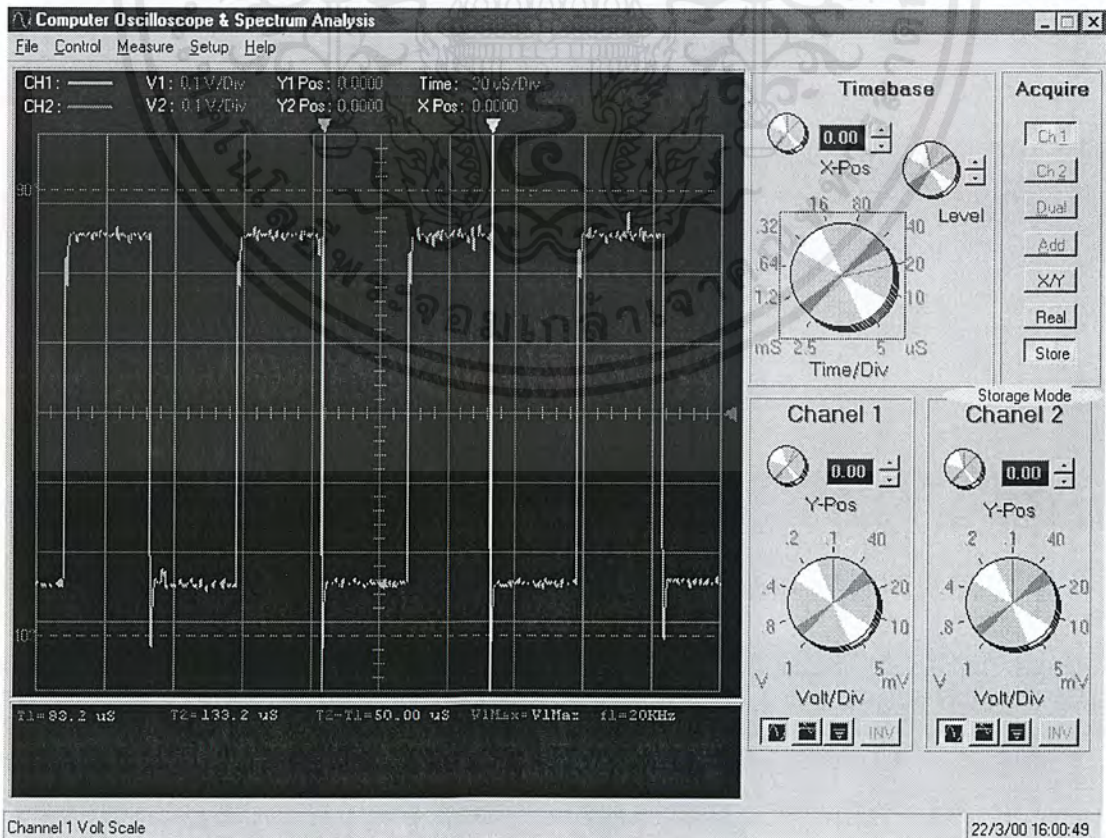


รูปที่ 7.9 ผลการวัดสัญญาณ square ความถี่ 500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

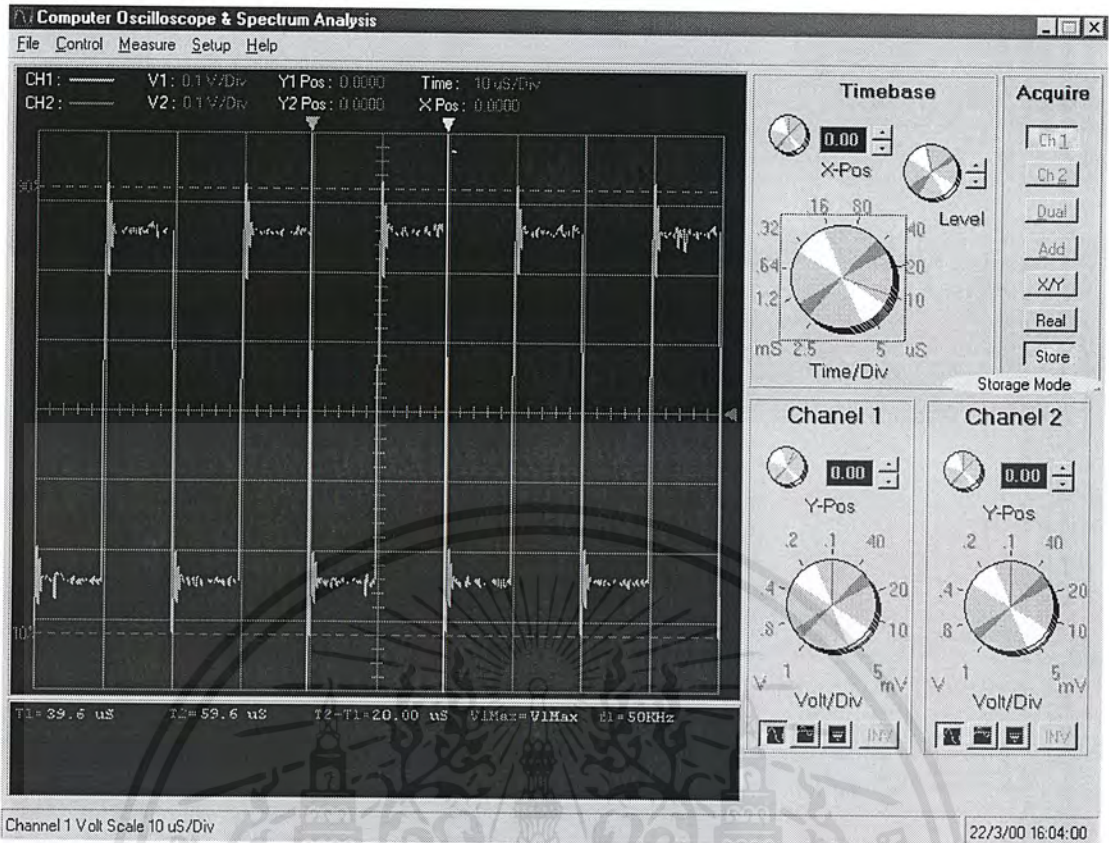


รูปที่ 7.10 ผลการวัดสัญญาณ square ความถี่ 4 kHz

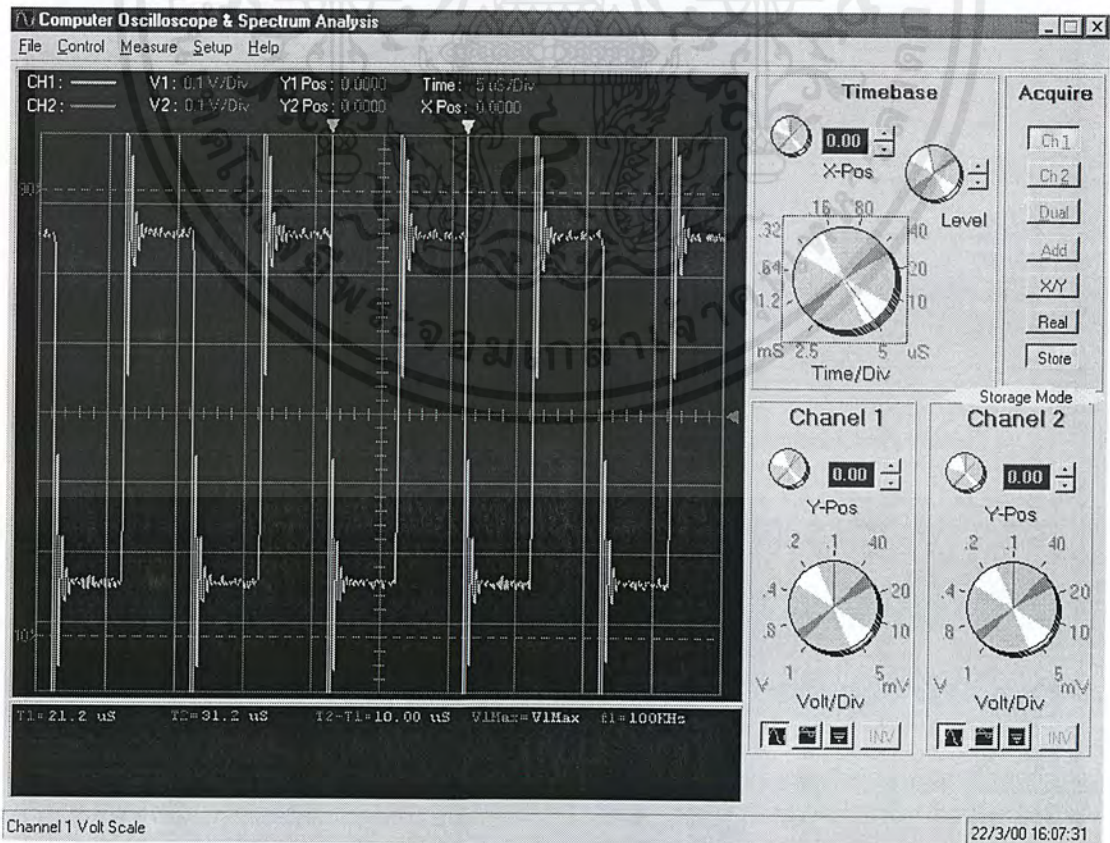


รูปที่ 7.11 ผลการวัดสัญญาณ square ความถี่ 20 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.12 ผลการวัดสัญญาณ square ความถี่ 50 kHz

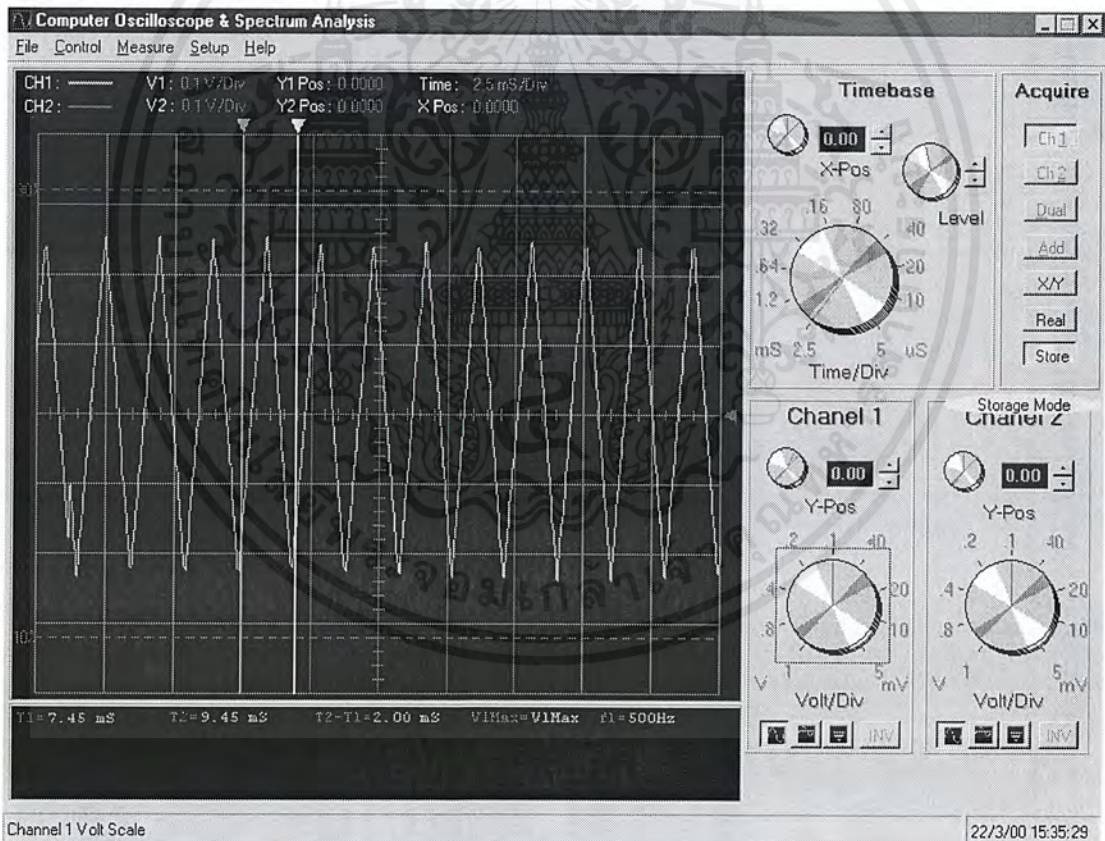


รูปที่ 7.13 ผลการวัดสัญญาณ square ความถี่ 100 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

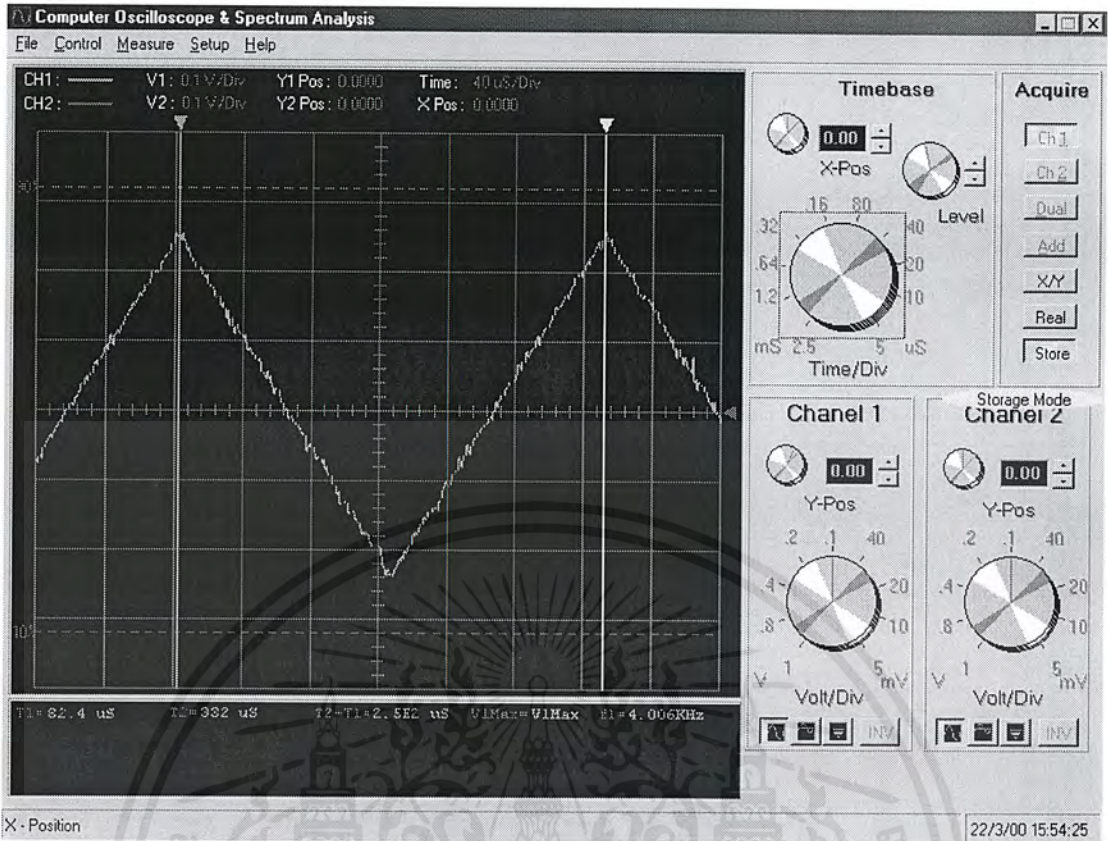
- ผลการทดลองวัดสัญญาณ triangle

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการ Sampling
รูปที่ 7.14	500	0.5	128 kbyte	5 MHz
รูปที่ 7.15	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.16	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.17	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.18	100 k	0.5	4 kbyte	10 MHz

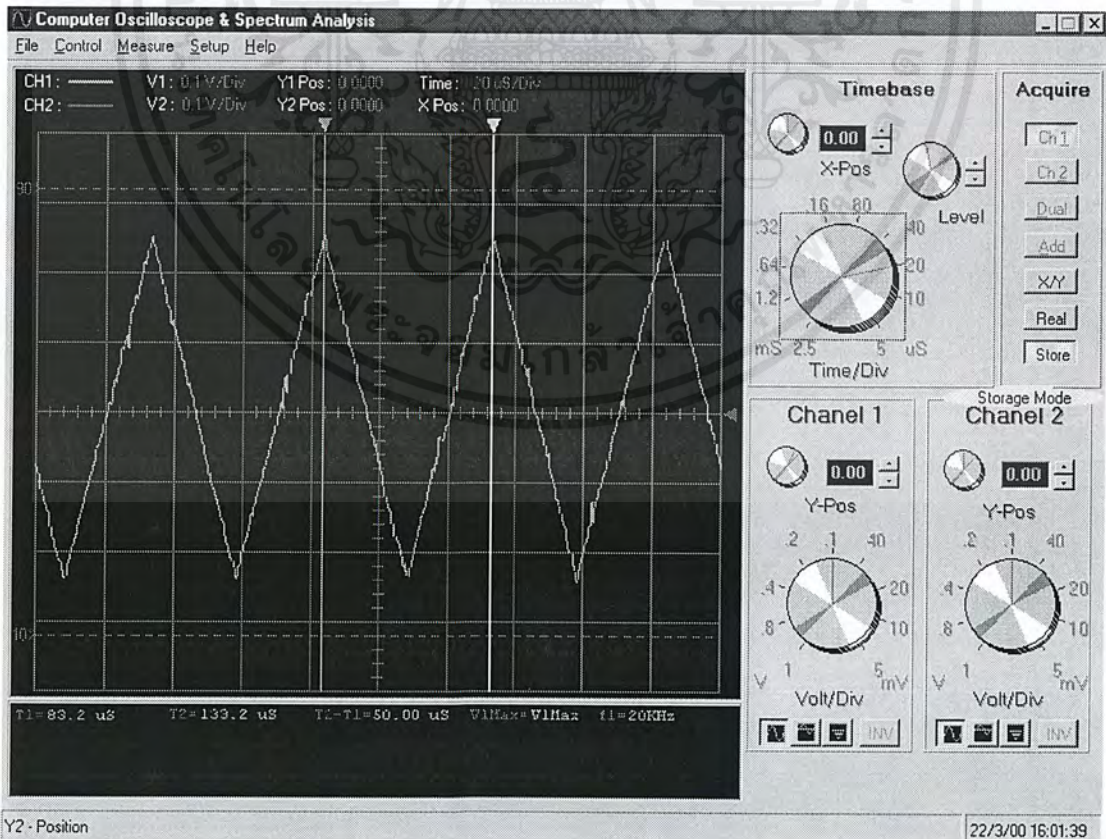


รูปที่ 7.14 ผลการวัดสัญญาณ triangle ความถี่ 500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

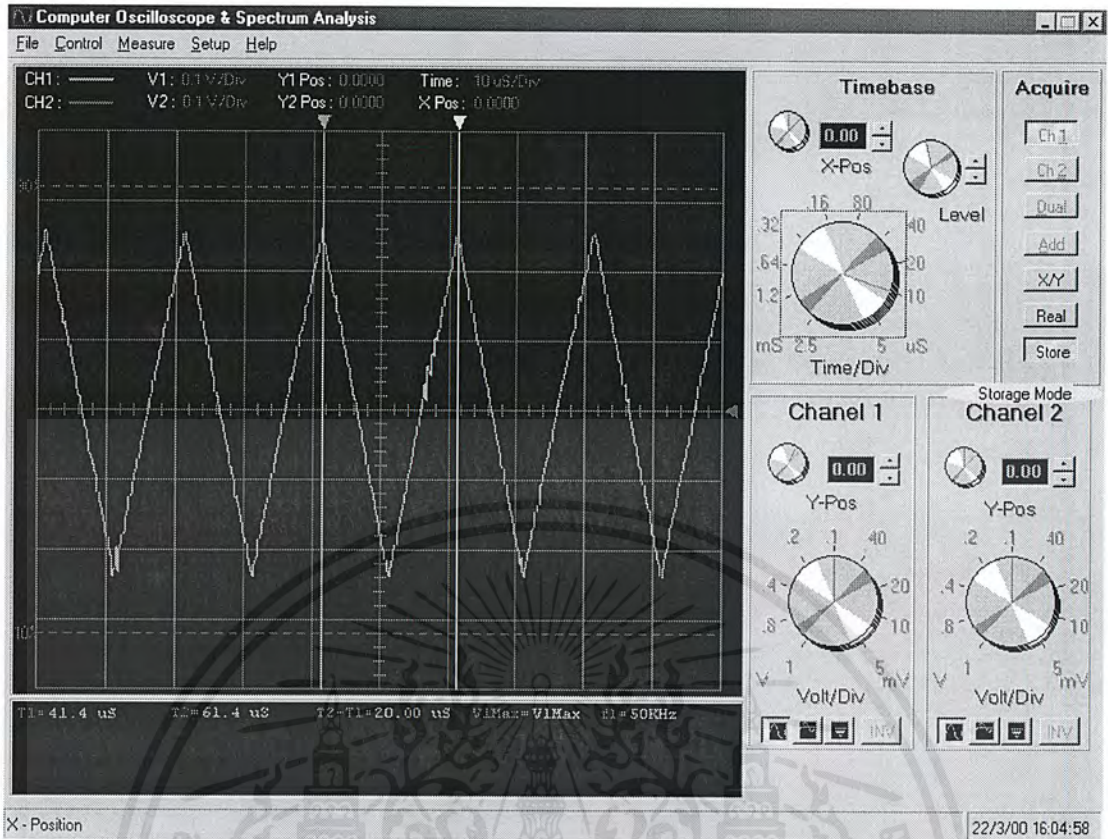


รูปที่ 7.15 ผลการวัดสัญญาณ triangle ความถี่ 4 kHz

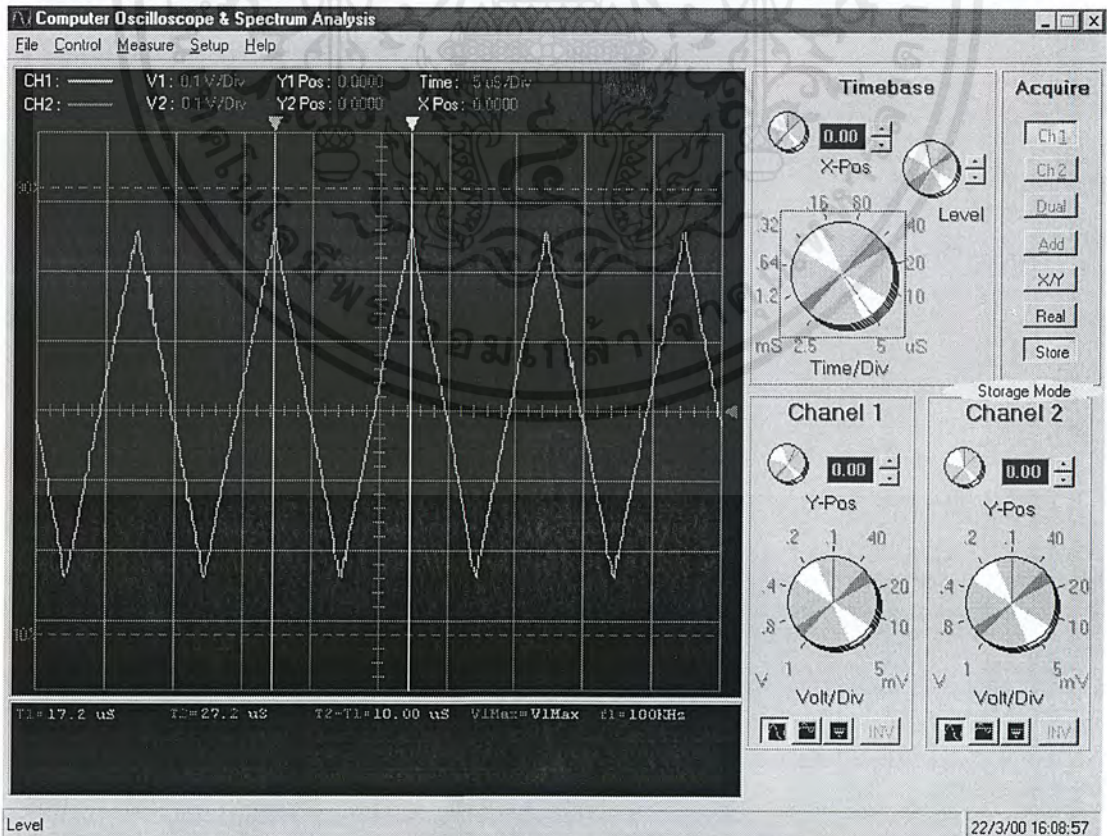


รูปที่ 7.16 ผลการวัดสัญญาณ triangle ความถี่ 20 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.17 ผลการวัดสัญญาณ triangle ความถี่ 50 kHz



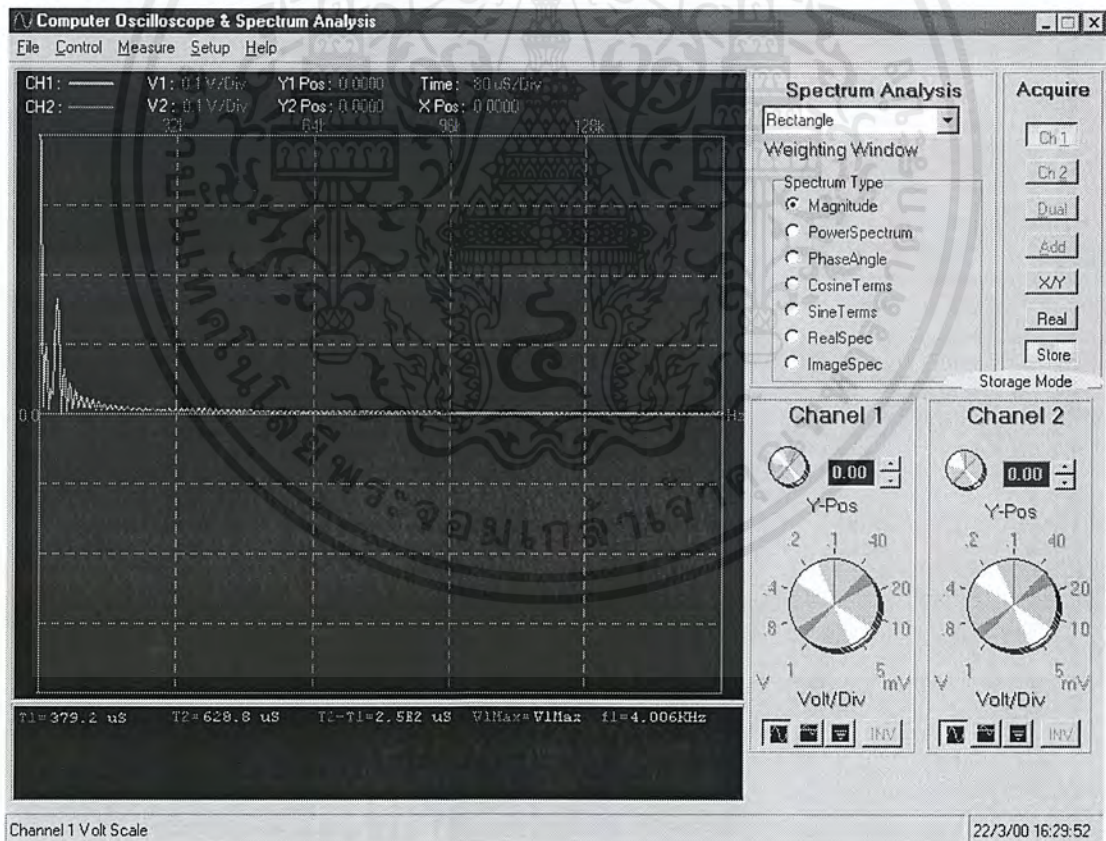
รูปที่ 7.18 ผลการวัดสัญญาณ triangle ความถี่ 100 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.2 ทดลองวิเคราะห์สเปกตรัมของสัญญาณที่มีความถี่ต่างๆ

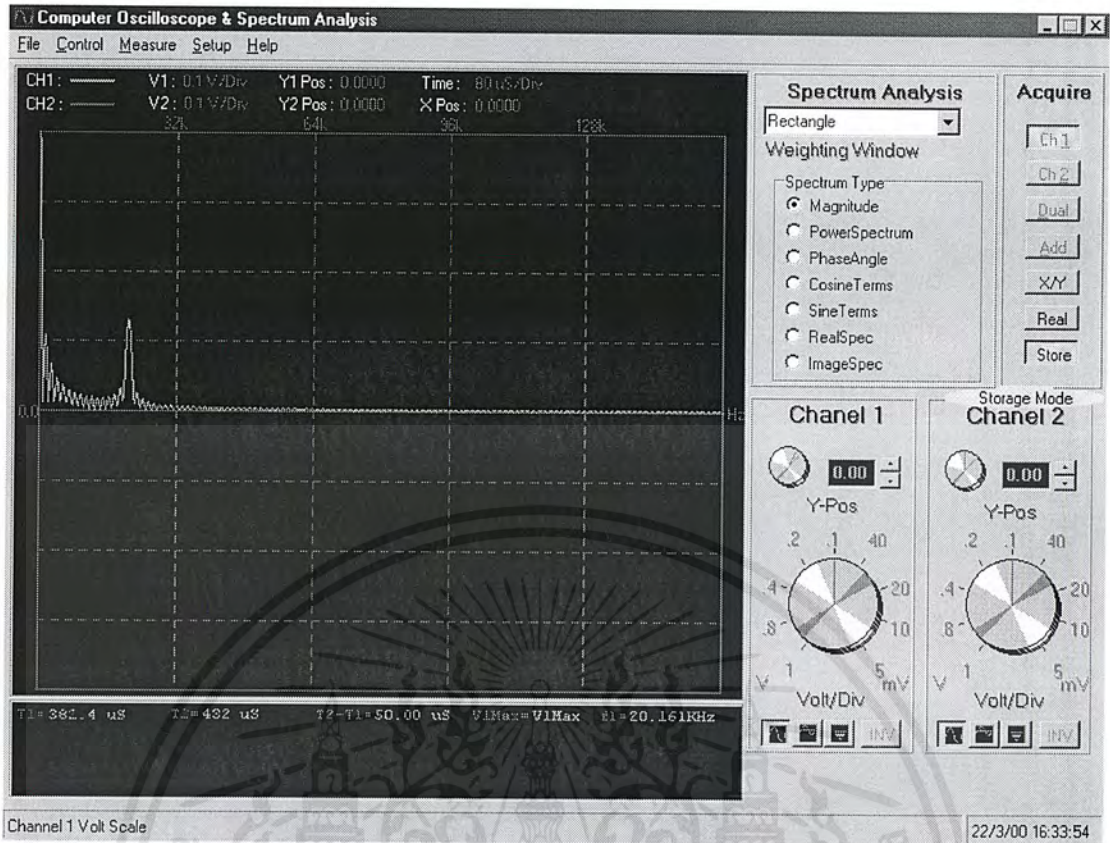
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณ sine

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการ Sampling
รูปที่ 7.19	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.20	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.21	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.22	100 k	0.5	4 kbyte	10 MHz
รูปที่ 7.23	300 k	0.5	4 kbyte	10 MHz
รูปที่ 7.24	500 k	0.5	4 kbyte	10 MHz
รูปที่ 7.25	1 M	0.5	4 kbyte	10 MHz

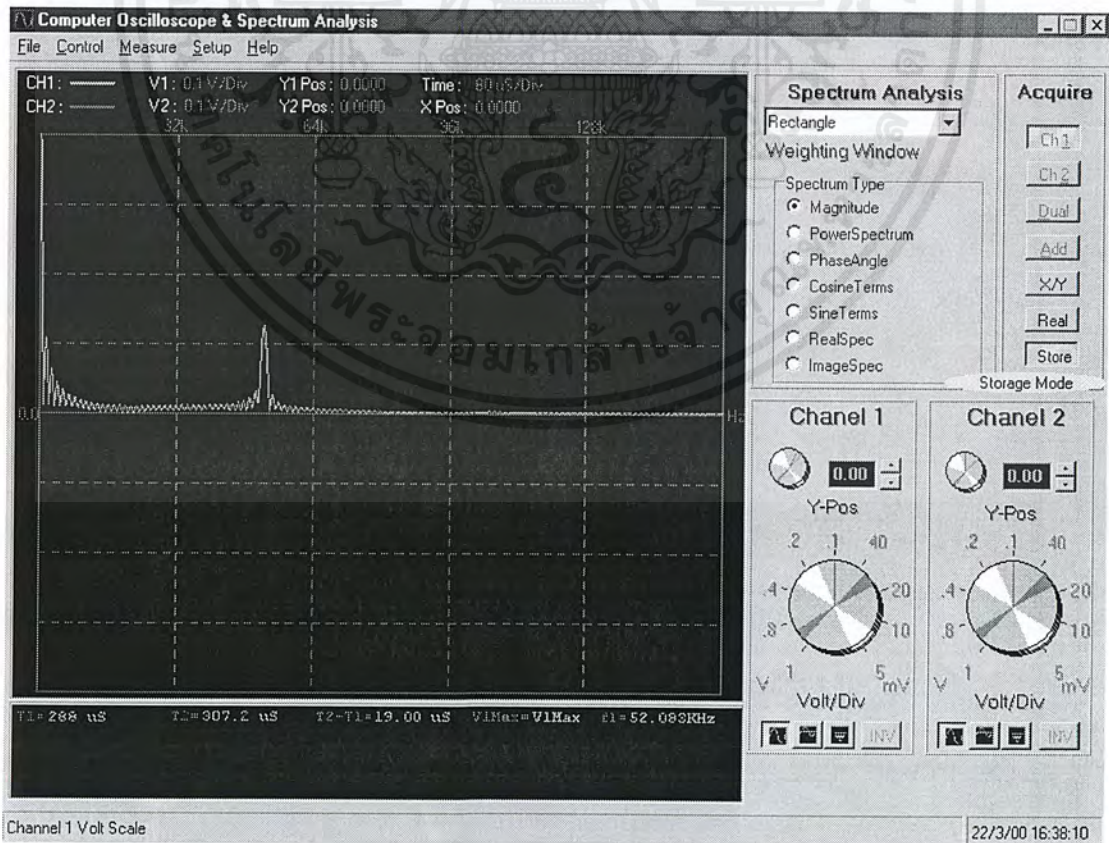


รูปที่ 7.19 ผลการวิเคราะห์สัญญาณไซน์ความถี่ 4 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

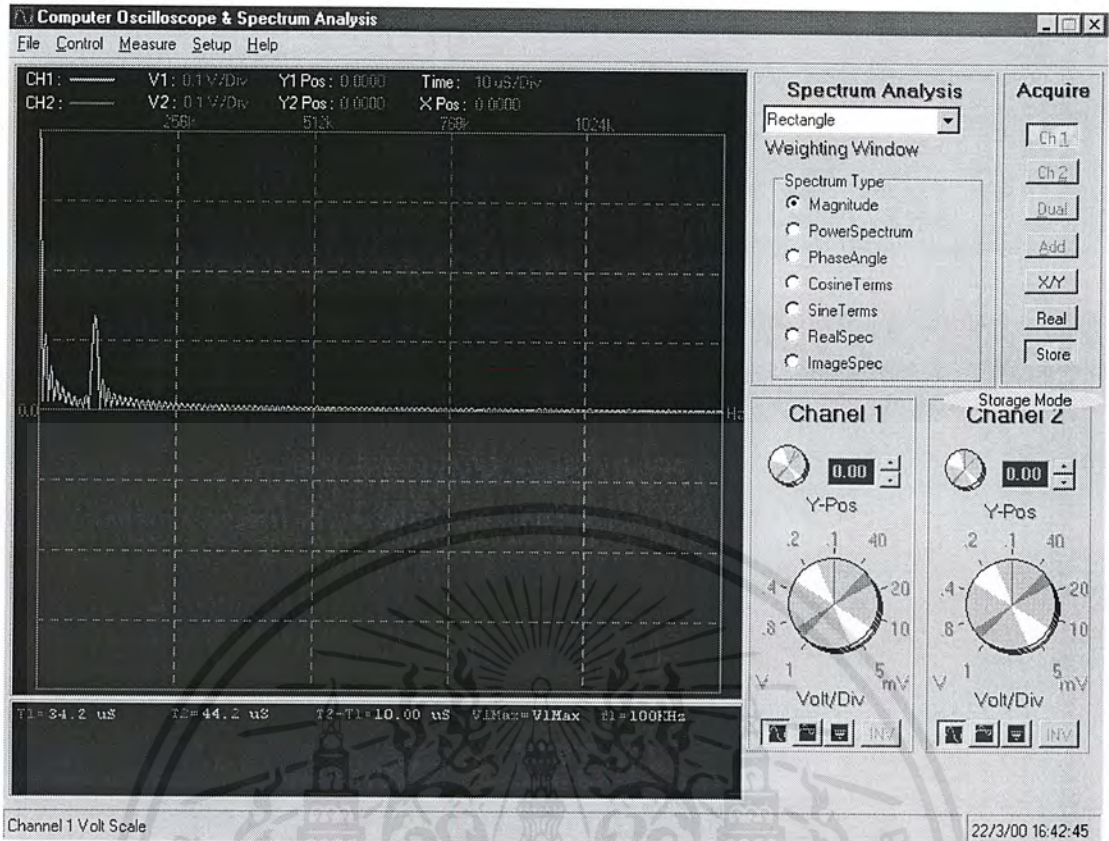


รูปที่ 7.20 ผลการวิเคราะห์สัญญาณความถี่ 20 kHz

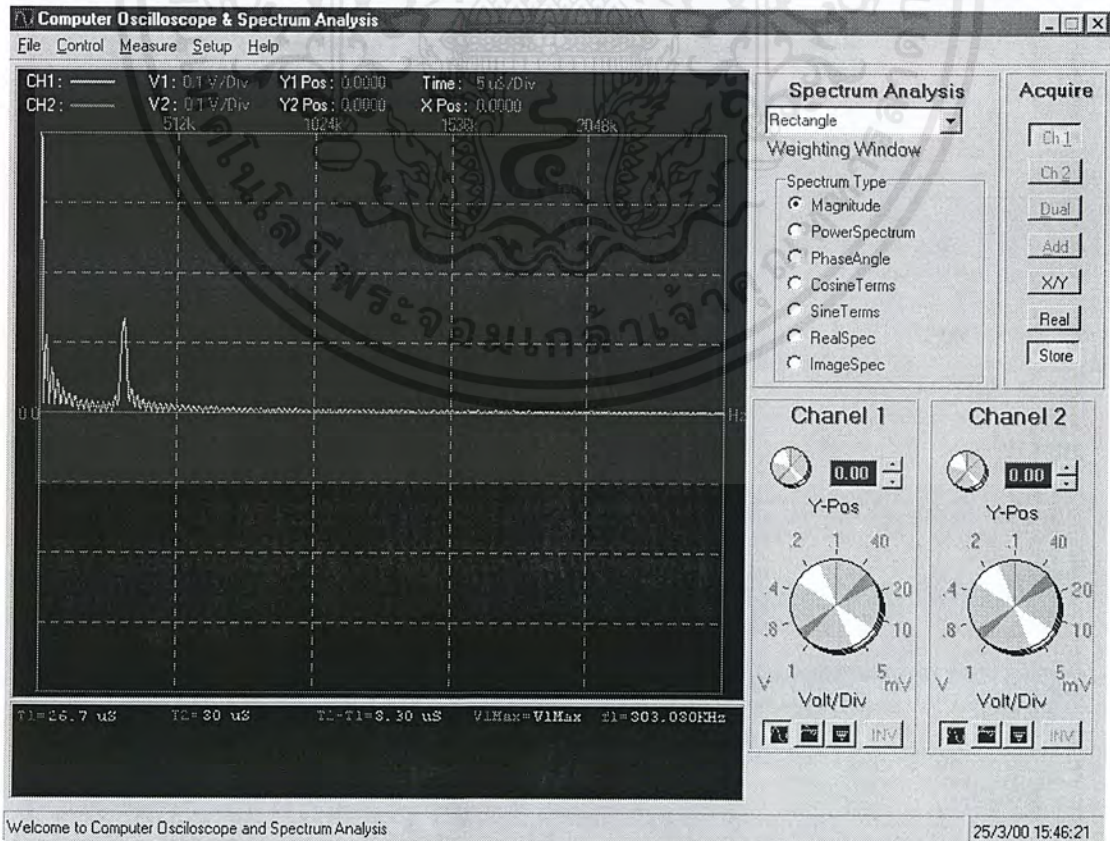


รูปที่ 7.21 ผลการวิเคราะห์สัญญาณความถี่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

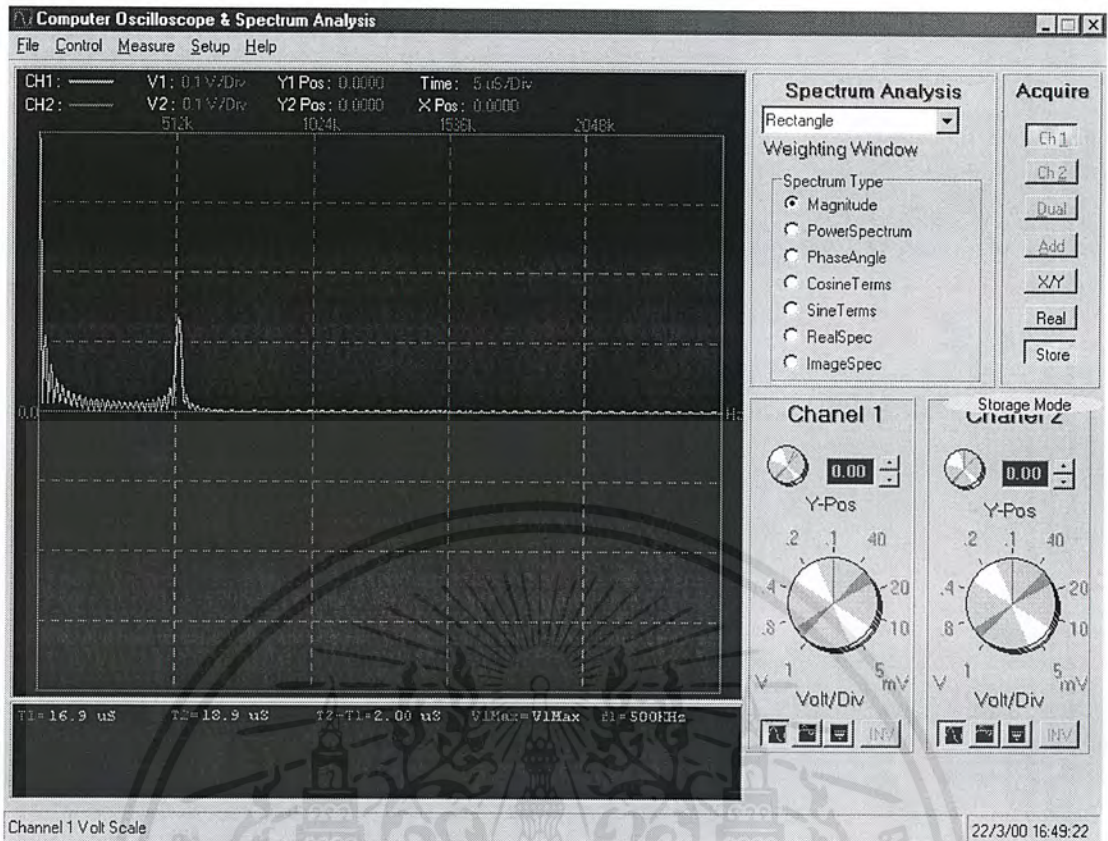


รูปที่ 7.22 ผลการวิเคราะห์สัญญาณความถี่ 100 kHz

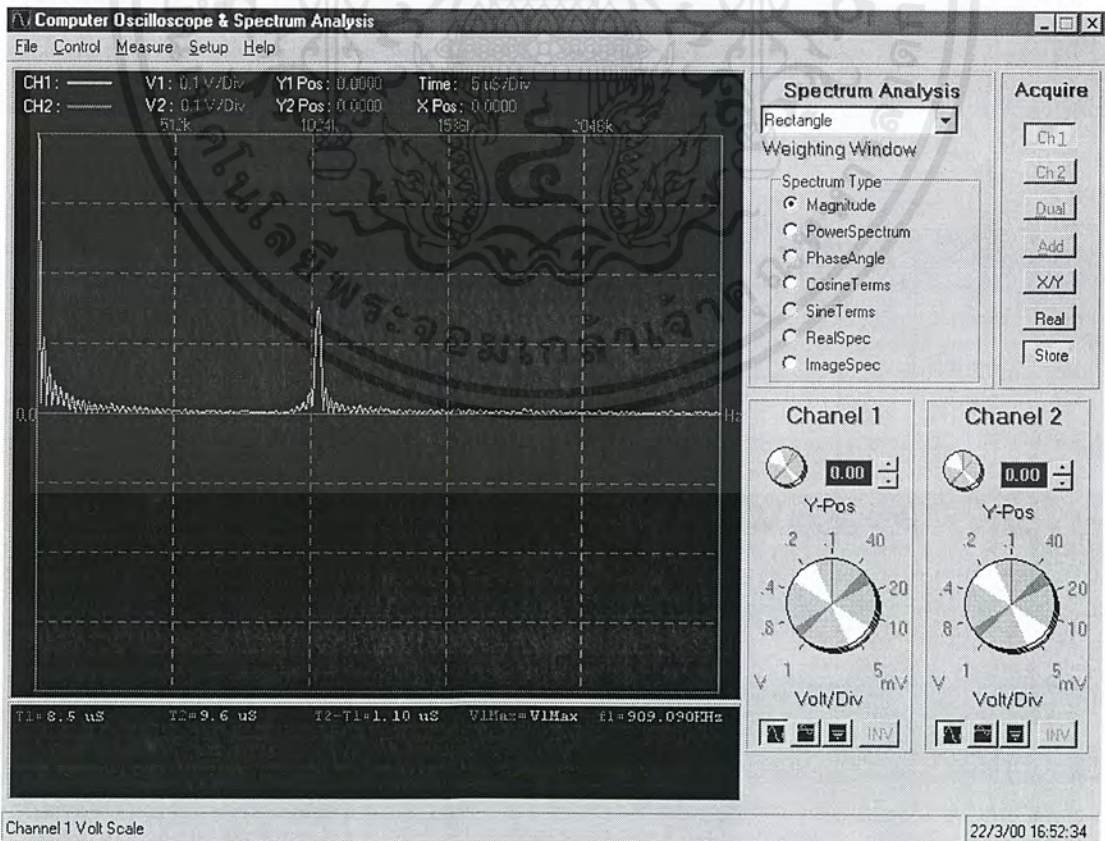


รูปที่ 7.23 ผลการวิเคราะห์สัญญาณความถี่ 300 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.24 ผลการวิเคราะห์สัญญาณชาน์ความถี่ 500 kHz

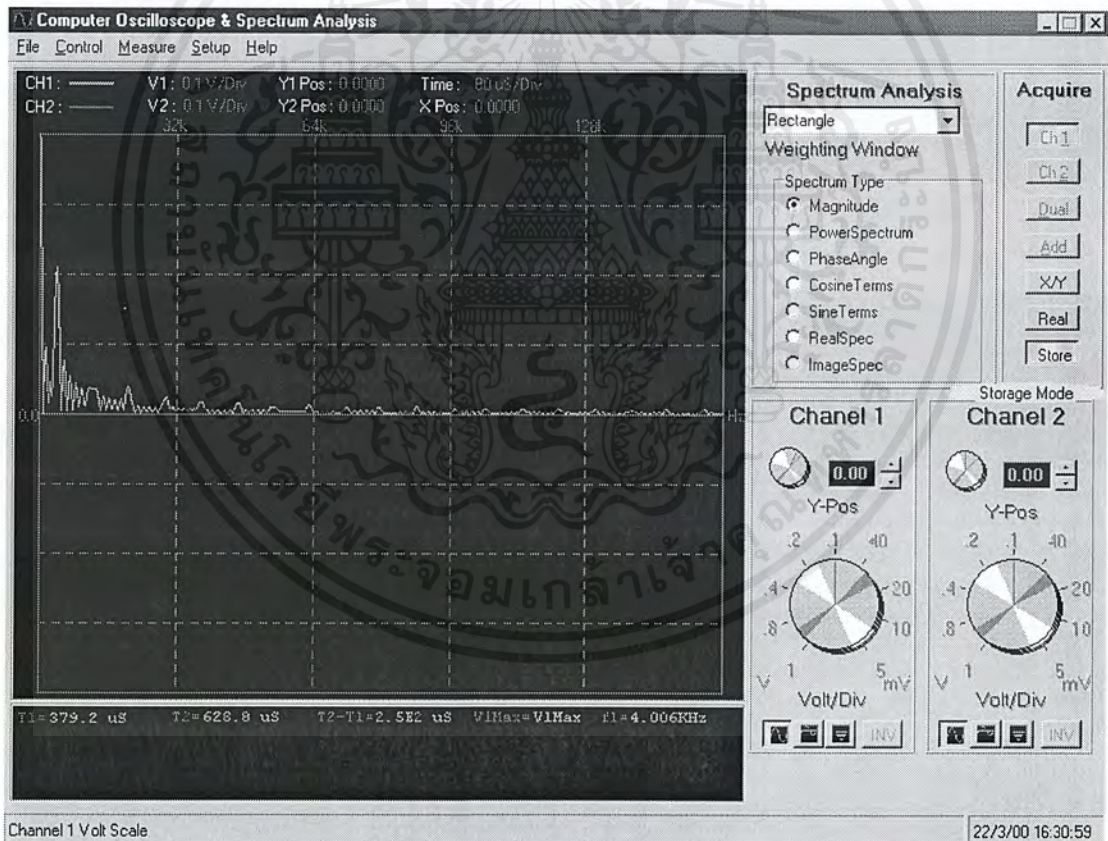


รูปที่ 7.25 ผลการวิเคราะห์สัญญาณชาน์ความถี่ 1 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

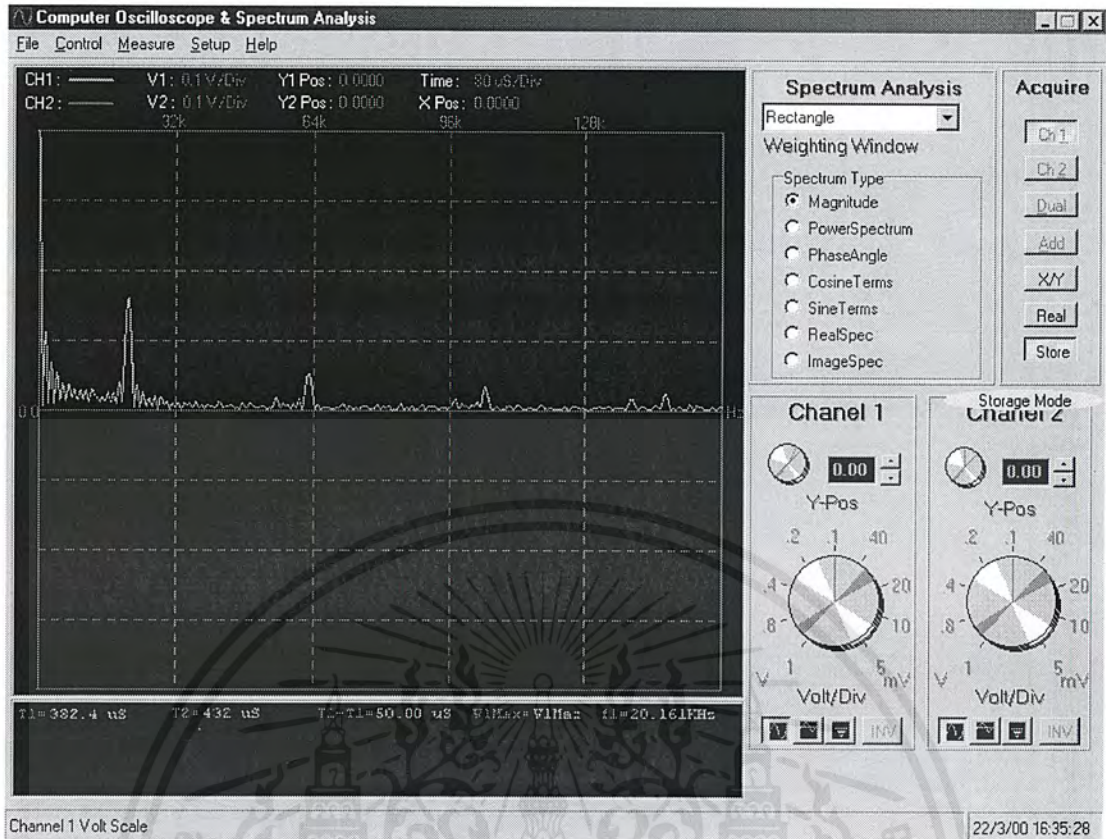
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณ square

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการการ Sampling
รูปที่ 7.26	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.27	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.28	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.29	100 k	0.5	4 kbyte	10 MHz
รูปที่ 7.30	300 k	0.5	4 kbyte	10 MHz
รูปที่ 7.31	500 k	0.5	4 kbyte	10 MHz
รูปที่ 7.32	1 M	0.5	4 kbyte	10 MHz

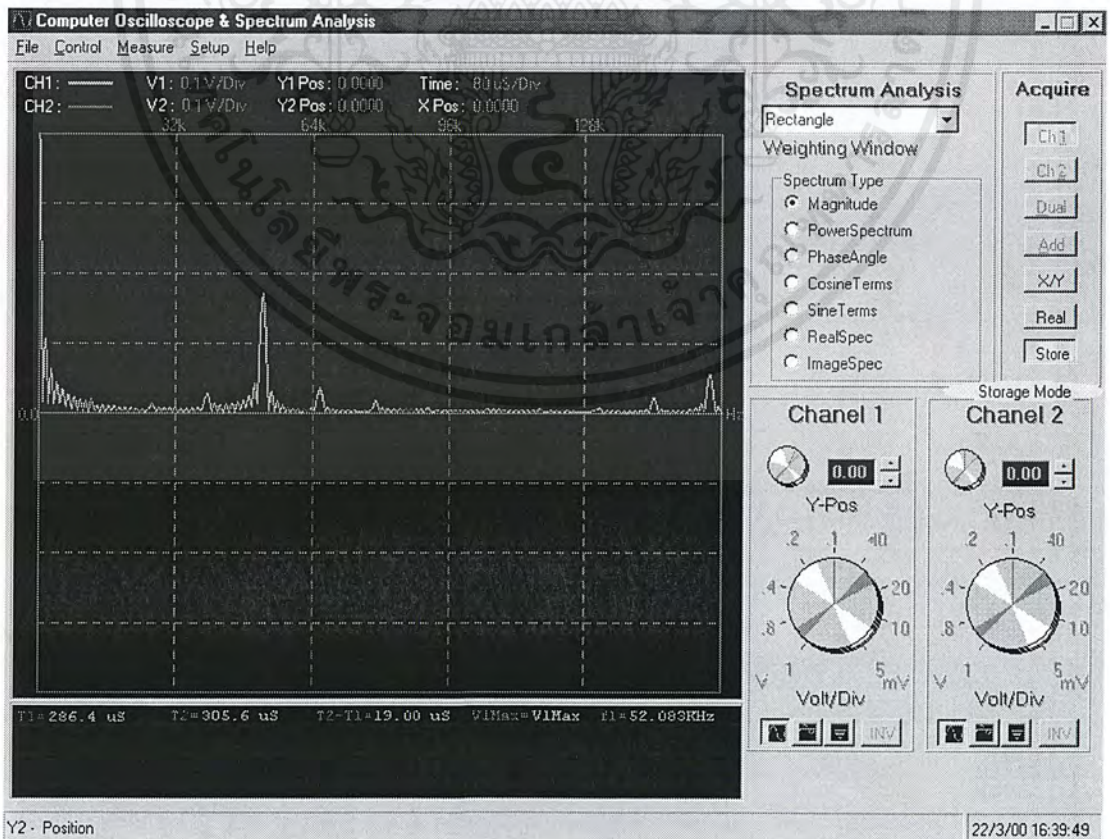


รูปที่ 7.26 ผลการวิเคราะห์สัญญาณ square ความถี่ 4 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

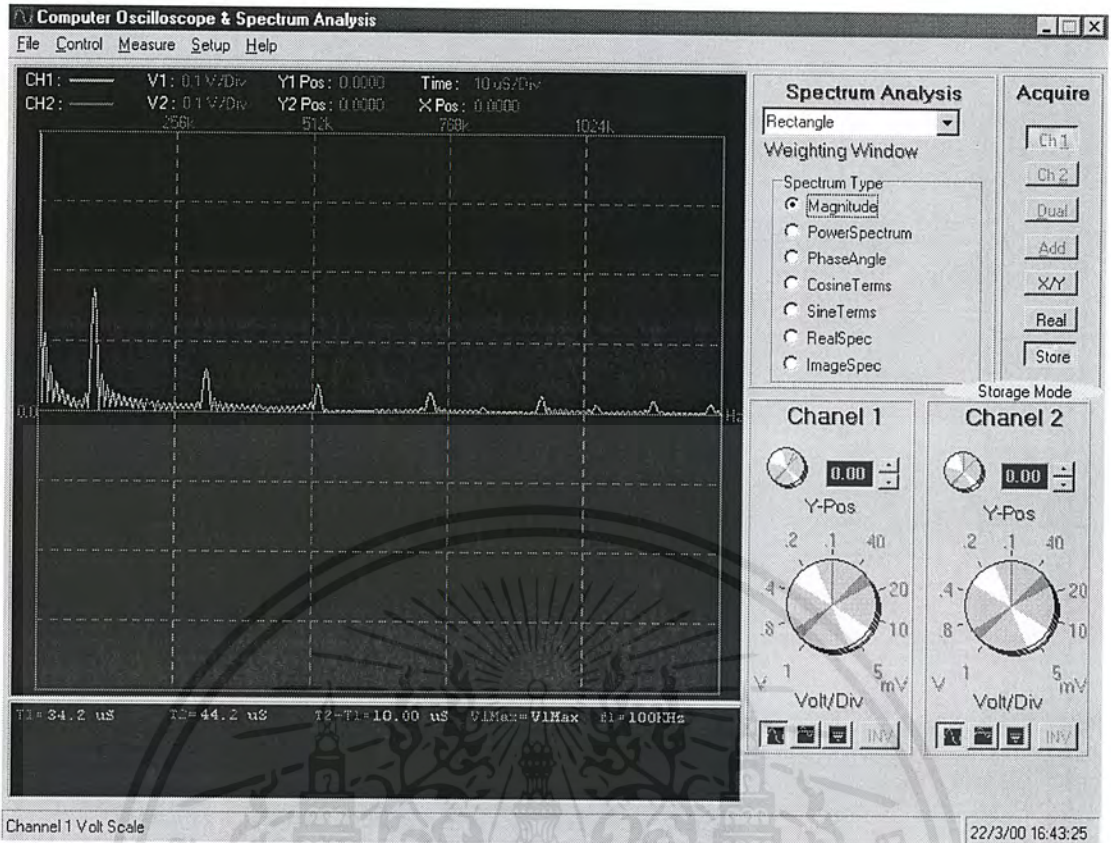


รูปที่ 7.27 ผลการวิเคราะห์สัญญาณ square ความถี่ 20 kHz

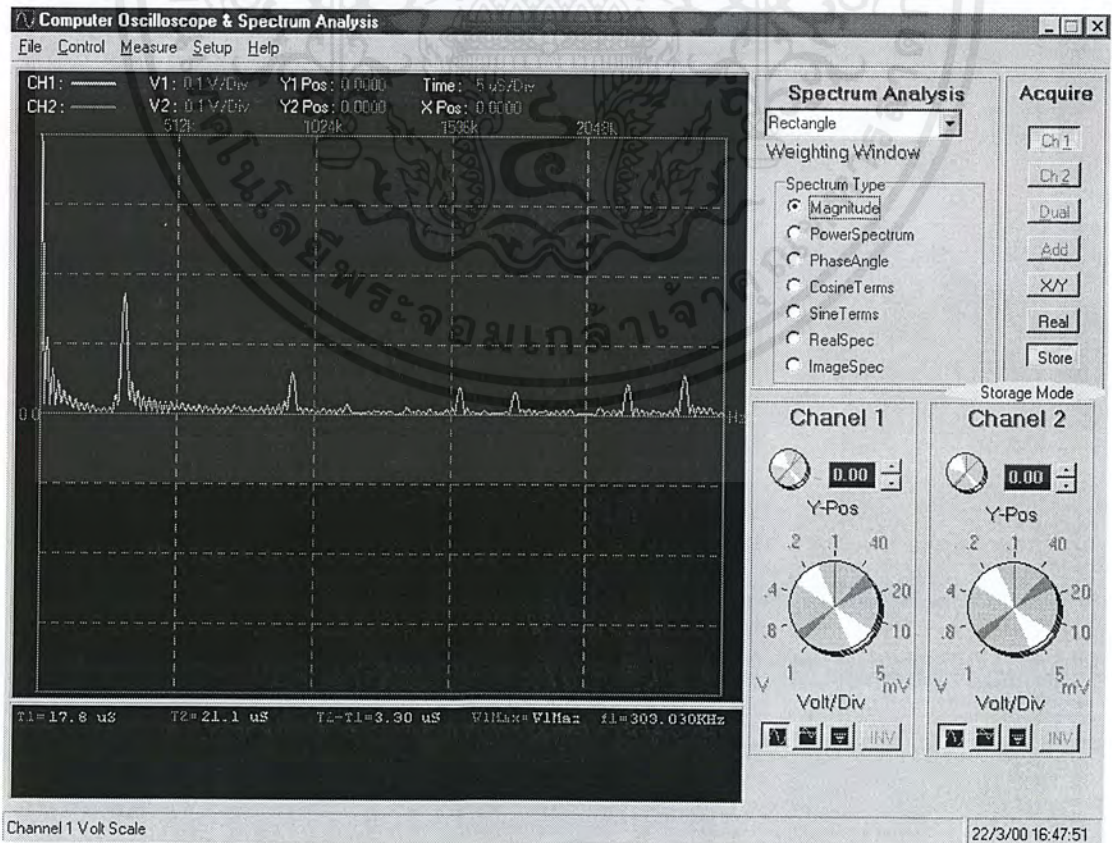


รูปที่ 7.28 ผลการวิเคราะห์สัญญาณ square ความถี่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

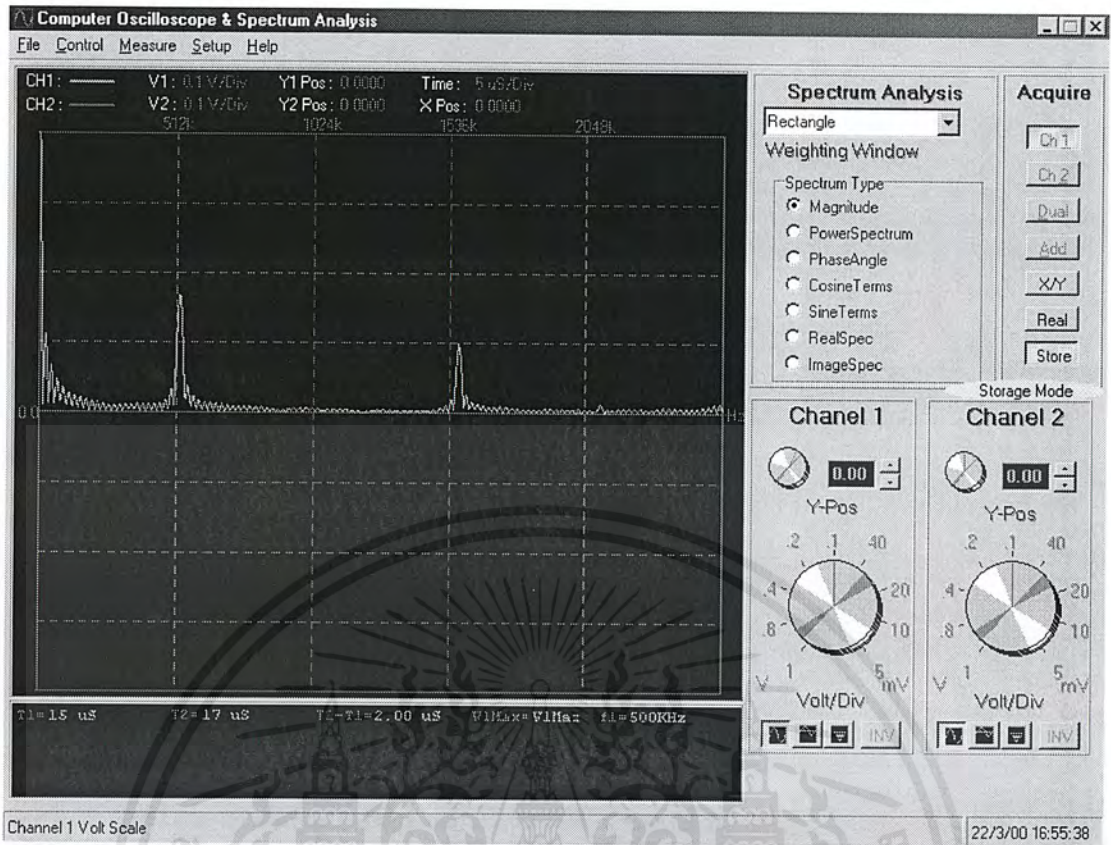


รูปที่ 7.29 ผลการวิเคราะห์สัญญาณ square ความถี่ 100 kHz

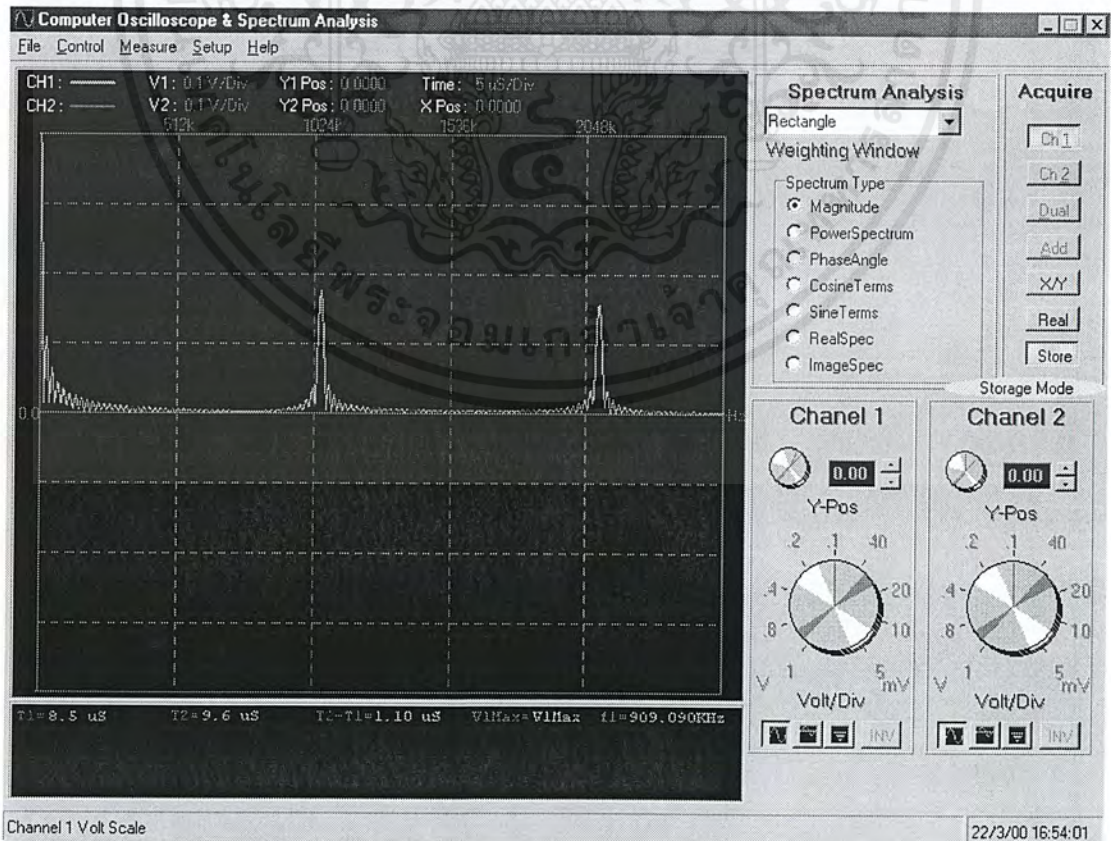


รูปที่ 7.30 ผลการวิเคราะห์สัญญาณ square ความถี่ 300 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.31 ผลการวิเคราะห์สัญญาณ square ความถี่ 500 kHz

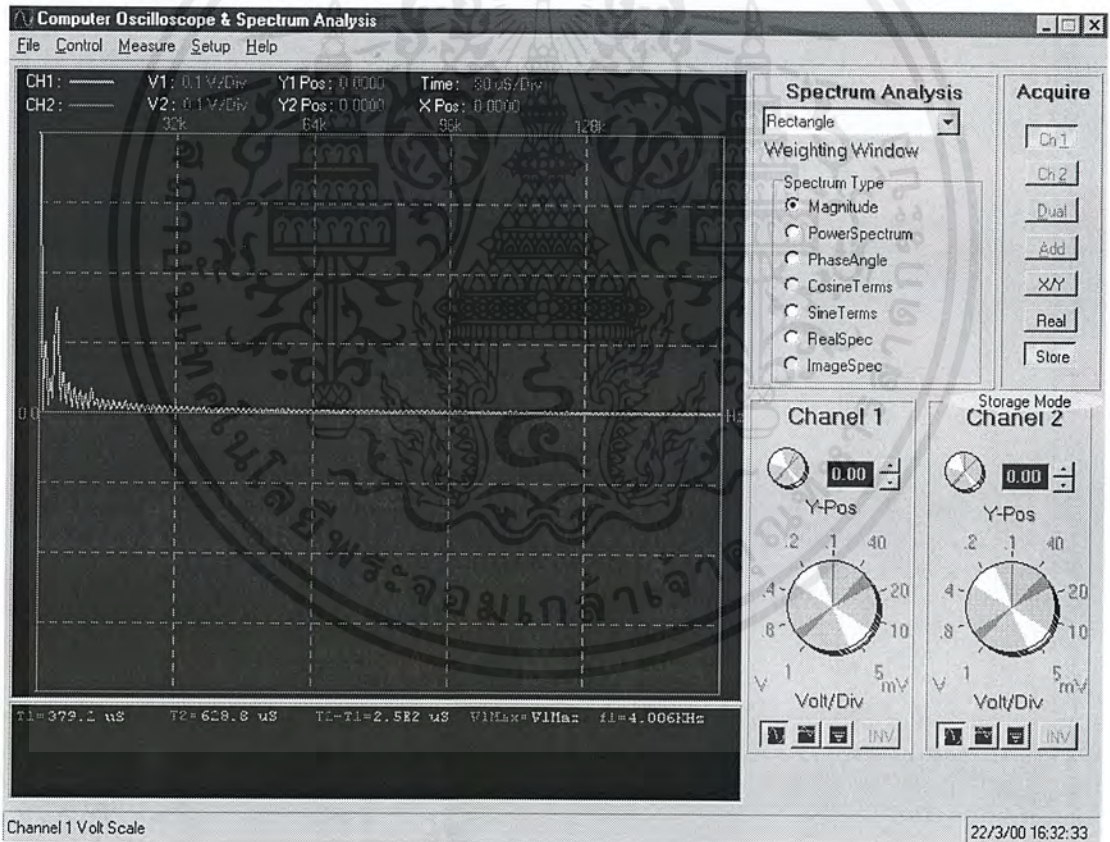


รูปที่ 7.32 ผลการวิเคราะห์สัญญาณ square ความถี่ 1 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

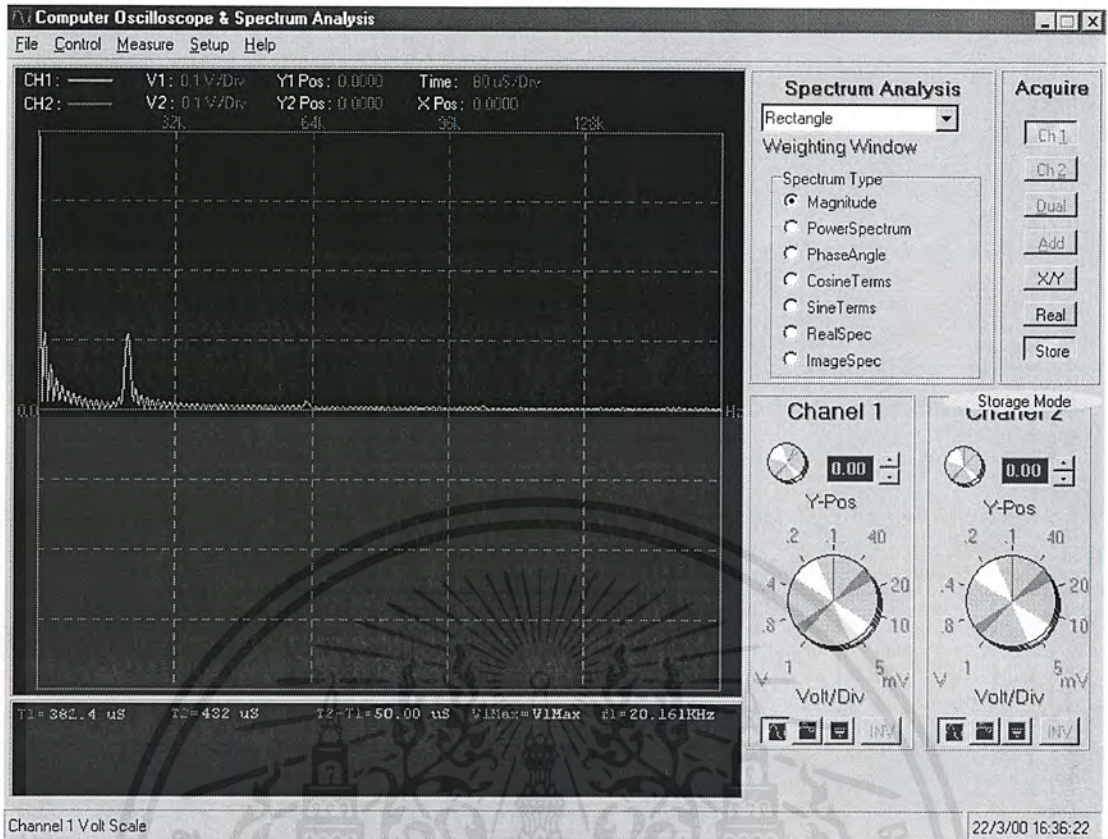
- ผลการทดลองวิเคราะห์สเปกตรัมของสัญญาณ triangle

ผลการทดลอง	ความถี่ (Hz)	ขนาดสัญญาณ(Vpp)	ขนาดหน่วยความจำที่ใช้	อัตราการ Sampling
รูปที่ 7.33	4 k	0.5	8 kbyte	625 kHz
รูปที่ 7.34	20 k	0.5	8 kbyte	2.5 MHz
รูปที่ 7.35	50 k	0.5	4 kbyte	5 MHz
รูปที่ 7.36	100 k	0.5	4 kbyte	10 MHz

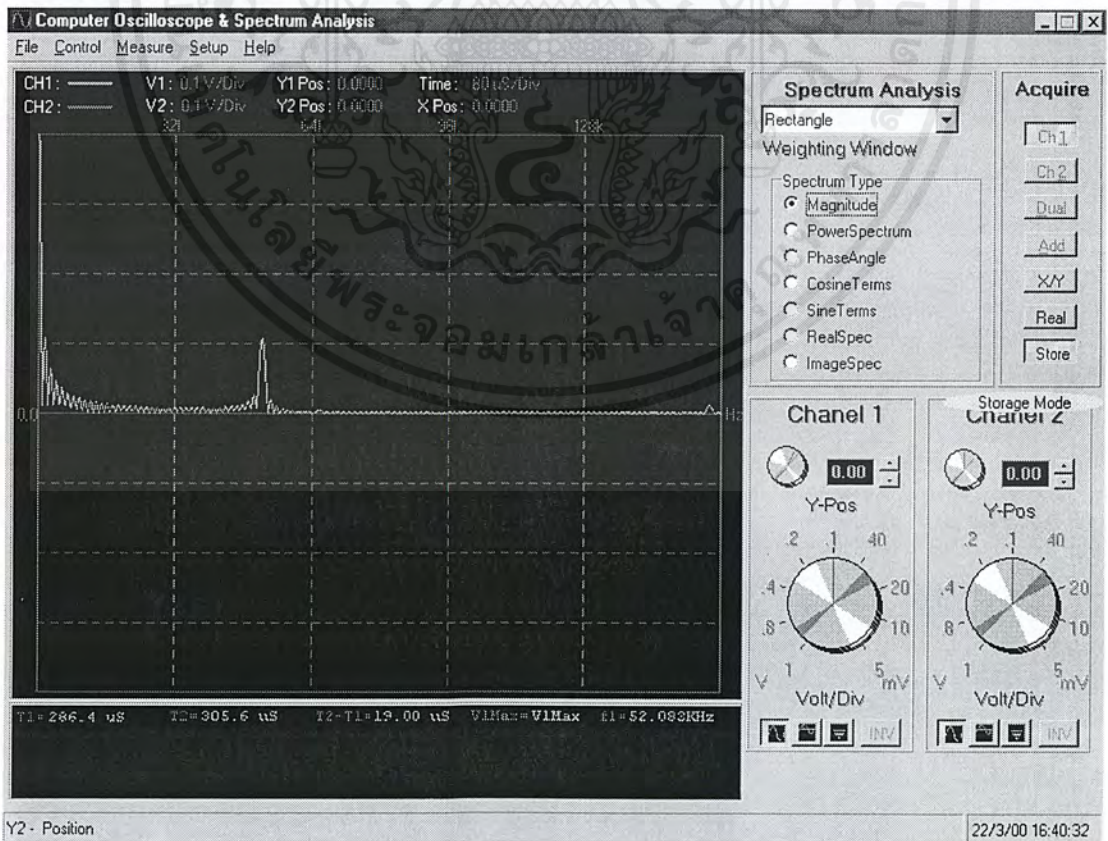


รูปที่ 7.33 ผลการวิเคราะห์สัญญาณ triangle ความถี่ 4 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

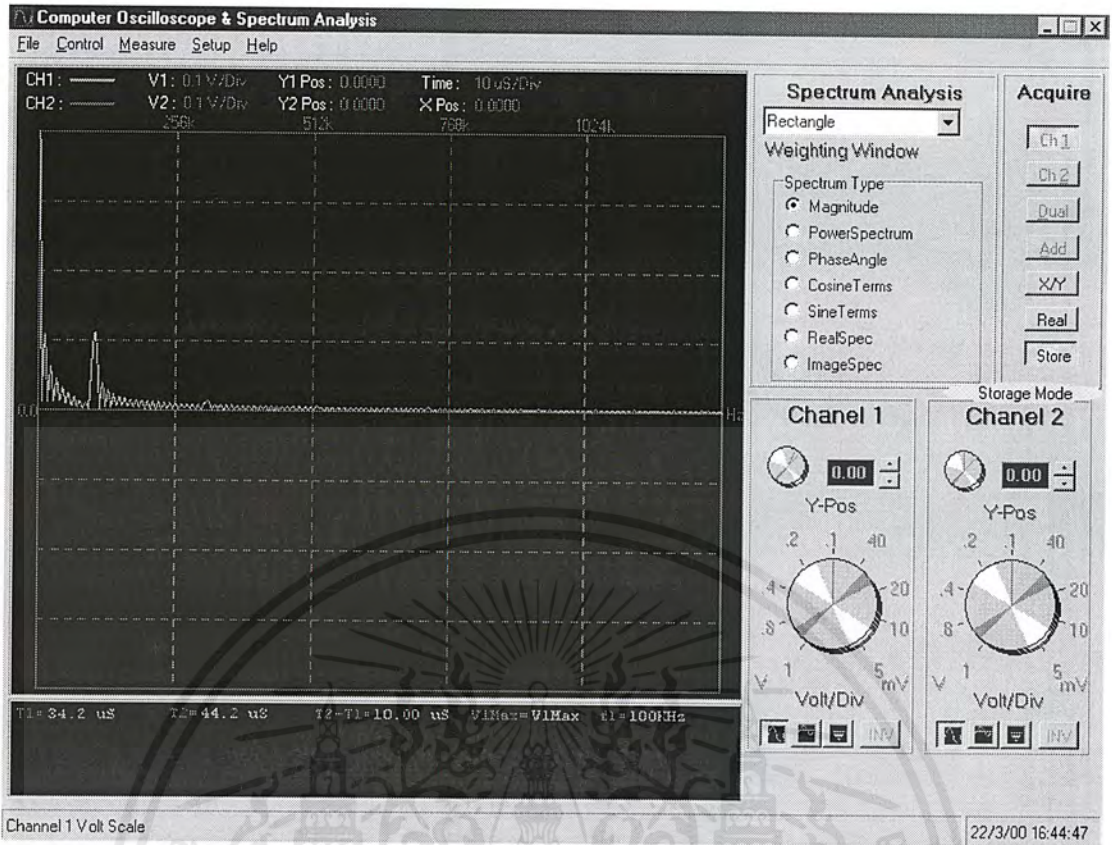


รูปที่ 7.34 ผลการวิเคราะห์สัญญาณ triangle ความถี่ 20 kHz



รูปที่ 7.35 ผลการวิเคราะห์สัญญาณ triangle ความถี่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.36 ผลการวิเคราะห์สัญญาณ triangle ความถี่ 100 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปผลการทดลอง

8.1 บทสรุปและวิจารณ์

จากผลการทดลองวงจรในส่วนต่างๆในบทที่ 7 ทั้งวงจรในส่วนของภาคอนาล็อก และวงจรในส่วนของภาคดิจิทัล วงจรสามารถทำงานได้ค่อนข้างดี การปรับระดับทริกเกอร์ ในโครงงานนี้จะไม่ใช่ฮาร์ดแวร์ทริกเกอร์ แต่จะใช้ซอฟต์แวร์ทริกเกอร์แทน ข้อดีของซอฟต์แวร์ทริกเกอร์คือ สัญญาณที่แสดงผลจะนิ่งกว่าฮาร์ดแวร์ทริกเกอร์ แต่ก็มีข้อเสียตรงที่การแสดงผลจะช้าลงเพราะต้องแบ่งเวลาส่วนหนึ่งไปประมวลผลในส่วนของการรักษาระดับทริกเกอร์ไว้ ข้อมูลที่ได้จากการแปลงอนาล็อกเป็นดิจิทัล จะถูกเก็บลงในหน่วยความจำ และตัวโปรแกรมจะอ่านค่าข้อมูลขึ้นมาแสดงผล ในส่วนของการแปลงข้อมูลของวงจรถอนาล็อกเป็นดิจิทัล แปลงข้อมูลได้ค่อนข้างแม่นยำ ข้อมูลที่ผิดพลาดส่วนใหญ่จะเกิดจากสัญญาณรบกวนที่เกิดขึ้นในภาคของวงจรถอนาล็อก

ในส่วนของโปรแกรมวัดสัญญาณและแสดงผลที่คอมพิวเตอร์เมื่อใช้วัดความถี่ต่างๆการแสดงผลจะค่อนข้างช้า และมีโอกาสที่จะผิดพลาดได้ แต่ที่ความถี่สูง(ประมาณ 5kHz – 500 kHz) การทำงานโดยรวมไม่ค่อยมีปัญหา ในส่วนของการวิเคราะห์สเปกตรัม ก่อนจะเข้าฟังก์ชันวิเคราะห์สเปกตรัม เพื่อให้ได้ผลการวิเคราะห์สเปกตรัมที่ถูกต้องแม่นยำ ควรปรับ Time/Div ให้มีสัญญาณปรากฏที่หน้าจออย่างน้อย 5 คาบเวลาขึ้นไป

8.2 ปัญหาที่พบในการทำโครงงาน

1. อุปกรณ์บางตัวหายาก เช่น IC เบอร์ TDA 8703 แบบ DIP 24 ขา มีขายเพียงร้านเดียวเท่านั้น ที่ร้าน EPS บ้านหม้อ และ IC เบอร์ LM 6361 หาซื้อไม่ได้จะต้องให้ร้าน กล้วยงษ์ สั่งซื้อให้
2. เนื่องจากวงจรที่ออกแบบมา ทำให้เกิดปัญหาในบางครั้ง ทางด้านผลการทดลองที่ผิดบ้าง ถูกบ้าง กว่าที่จะได้วงจรที่สมบูรณ์แบบ และใช้งานได้จริง จะต้องทำการออกแบบวงจรและทดลองใหม่หลายครั้งทำให้เสียเวลาไปมาก ก่อนที่จะออกแบบแผ่นวงจรพิมพ์ก็ได้ ทำการสร้างวงจรโดยใช้สาย Wire ระหว่างอุปกรณ์แต่ละตัว ซึ่งเมื่อนำไปใช้งานวงจรที่ได้จากการ Wire ให้สัญญาณรบกวนออกมาน้อย จึงเริ่มทำการออกแบบแผ่นวงจรพิมพ์ด้วย ปริ๊น PCB มาทดลองกลับปรากฏว่ามีสัญญาณรบกวนมากกว่าวงจรที่ Wire ไว้ จึงตัดสินใจใช้วงจรที่ได้จากการ Wire ทำผลการทดลองดังบทที่ 7
3. ในการออกแบบวงจรเราใช้แผ่นวงจรอนกประสงค์ ทำให้เกิดการผิดพลาดบางอย่างขึ้นได้
4. เกิดการรบกวนจากสัญญาณภายในเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ต้องใช้เวลาในการเขียน โปรแกรมค่อนข้างมากเพราะต้องศึกษาการเขียน โปรแกรม Delphi 4.0 เพิ่มเติม

8.3 การแก้ไขปัญหา

ในส่วนของฮาร์ดแวร์ เนื่องจากใช้การ Wire lab ซึ่งทำให้เกิดการรบกวนกันขึ้นจึงแก้ปัญหา ดังนี้

1. วางอุปกรณ์ให้ชิดกันมากที่สุด เพื่อลดระยะทางของสายให้สั้นลง ทำให้การรบกวนลดลงไปได้บ้าง
2. ในการวางสาย Wire lab ควรจัดวางให้เป็นระเบียบมากที่สุดเท่าที่จะทำได้
3. สำหรับ IC TTL มีการต่อตัวเก็บประจุค่า 0.1 μF คร่อมระหว่าง Vcc กับ Gnd ทุกตัว
4. ควรแยกไฟเลี้ยงที่จะจ่ายให้กับวงจรอนาลอกและวงจรดิจิทัลออกจากกัน
5. ในเครื่องคอมพิวเตอร์มีการออสซิเลทของอุปกรณ์บางตัว ซึ่งจะมารบกวนสัญญาณอนาลอกของภาคอินพุทของเราได้ ดังนั้นที่จุดต่อสัญญาณอินพุทที่เข้ามายังวงจรควรใช้สาย SHIELD เพื่อที่จะลดสัญญาณรบกวนให้น้อยที่สุดเท่าที่จะทำได้

ในส่วนของซอฟต์แวร์พยายามค้นคว้าหาตำราหรือเอกสารที่เกี่ยวข้องกับเทคนิคการเขียนโปรแกรม Delphi 4.0 ซึ่งถ้าจะหาข้อมูลจากอินเทอร์เน็ต ก็สามารถเข้าไปหาข้อมูลได้ที่ www.thaiDelphi.com ซึ่งจะมี Link ไปยัง WebSite อื่นที่เกี่ยวข้องกับการเขียน Delphi อีกมากมาย

8.4 ประโยชน์ที่ได้รับจากการทำโครงการ

1. ได้เครื่อง Digital Storage Oscilloscope & Spectrum Analysis ที่มีราคาถูกโดยใช้ต้นทุนประมาณ 2,500 บาท ในขณะที่ถ้าเราซื้อชนิดที่มีขายอยู่ในท้องตลาดจะมีราคาหลายหมื่นบาท และเครื่องที่เราสร้างขึ้นมาก็มีความสามารถในการวัดและวิเคราะห์สัญญาณได้เช่นเดียวกัน
2. ฮาร์ดแวร์ที่สร้างขึ้นสามารถใช้ร่วมกับคอมพิวเตอร์ PC ที่เรามีใช้กันอยู่อย่างแพร่หลายทำให้สะดวกในการใช้งานเป็นอย่างยิ่ง อีกทั้งยังจะทำให้เครื่องคอมพิวเตอร์ใช้งานได้อย่างกว้างขวางขึ้น
3. เครื่องที่เราสร้างขึ้นมาใช้งานแทน Storage Oscilloscope และ Spectrum Analysis ได้
4. โปรแกรมที่เขียนขึ้นมาสำหรับควบคุมการทำงาน สามารถใช้ได้กับระบบปฏิบัติการ Windows 32 bit ขึ้นไปได้
5. ได้รับความรู้กว้างขวางขึ้นอีกมาก ในการทำโครงการนี้ทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์
6. นักศึกษารุ่นหลังจะได้มีโอกาสศึกษาและพัฒนาให้ดีขึ้นได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.5 แนวทางในการพัฒนาต่อ

1. พัฒนาในส่วนของฮาร์ดแวร์ที่สามารถวัดสัญญาณทางด้าน Input ที่มีค่าแรงดันสูงขึ้น
ไปอีก
2. พัฒนาให้สามารถวัดสัญญาณที่มีความถี่สูงขึ้นกว่านี้ได้
3. พัฒนาให้สามารถวัดสัญญาณได้หลายๆ ช่องมากขึ้น

8.6 คุณสมบัติของโครงการนี้

1. ขนาดของสัญญาณอินพุต 20 mVpp – 8 Vpp
2. ความถี่ของสัญญาณอินพุต 100Hz – 500 kHz และสามารถวิเคราะห์สเปกตรัมได้ถึง
ความถี่ 1 MHz
3. สามารถวิเคราะห์ Magnitude Spectrum, Power Spectrum, Phase Spectrum, Real
Spectrum และ Imaginary Spectrum ได้
4. สามารถพริ้นต์ รูปสัญญาณที่ได้จากการวัดหรือที่ได้จากการวิเคราะห์สเปกตรัมได้
5. สามารถเก็บข้อมูลที่วัดหรือวิเคราะห์ลงฮาร์ดดิส หรือฟลอปปีดิสได้

The seal of the National Library of Thailand is a circular emblem. It features a central sunburst with a sun disk, flanked by two traditional Thai stupas. Below the sunburst is a decorative base with a central flame-like motif. The entire emblem is surrounded by a circular border containing Thai text. The text at the top reads 'สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง' and the text at the bottom reads 'หอสมุดแห่งชาติ'.

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit PCScope;

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ComCtrls, ExtCtrls, Menus, StdCtrls, Buttons, ImgList, zPanel, Math,
  ExtDigs, IniFiles, MsysStd, zLed, zonzoff, Fourier, printers;

type
TScope = class(TForm)
  MainMenu1: TMainMenu;
  File1: TMenuItem;
  Help1: TMenuItem;
  Panel1: TPanel;
  StatusBar1: TStatusBar;
  Bevel1: TBevel;
  Control1: TMenuItem;
  Setup1: TMenuItem;
  Measure1: TMenuItem;
  Label2: TLabel;
  Label1: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Panel2: TPanel;
  Label11: TLabel;
  Label14: TLabel;
  Label22: TLabel;
  Label25: TLabel;
  Label27: TLabel;
  ST1: TLabel;
  ST2: TLabel;
  DT: TLabel;
  VIMax: TLabel;
  F1: TLabel;
  Open1: TMenuItem;
  N1: TMenuItem;
  Save1: TMenuItem;
  Saveas1: TMenuItem;
  N2: TMenuItem;
  PrintSetup1: TMenuItem;
  Print1: TMenuItem;
  N3: TMenuItem;
  Export1: TMenuItem;
  Import1: TMenuItem;
  N4: TMenuItem;
  Exit1: TMenuItem;
  Shape1: TShape;
  Shape2: TShape;
  Image1: TImage;
  Mark2: TPanel;
  Mark1: TPanel;
  Image2: TImage;
  DateTime: TTimer;
  P00: TLabel;
  P10: TLabel;
  Report1: TMenuItem;
  N5: TMenuItem;

  Time1: TMenuItem;
  N6: TMenuItem;
  Channel11: TMenuItem;
  Channel21: TMenuItem;
  N7: TMenuItem;
  Trigger1: TMenuItem;
  Channel1Only: TMenuItem;
  Channel2Only1: TMenuItem;
  All1: TMenuItem;
  N8: TMenuItem;
  SetGrid1: TMenuItem;
  SetScreen1: TMenuItem;
  N9: TMenuItem;
  SetFont1: TMenuItem;
  ThaiFont1: TMenuItem;
  EnglishFont1: TMenuItem;
  Cont1: TMenuItem;
  Index1: TMenuItem;
  N10: TMenuItem;
  About1: TMenuItem;
  TimeBase: TPanel;
  XPos1: TzKnob;
  Ch1Panel: TPanel;
  Y1Pos: TzKnob;
  Ch1Volt: TzKnob;
  Ch2Panel: TPanel;
  Y2Pos: TzKnob;
  Inv1: TSpeedButton;
  Inv2: TSpeedButton;
  Inv11: TSpeedButton;
  Inv22: TSpeedButton;
  AC1: TSpeedButton;
  DC1: TSpeedButton;
  GND1: TSpeedButton;
  AC2: TSpeedButton;
  DC2: TSpeedButton;
  GND2: TSpeedButton;
  UpDown1: TUpDown;
  UpDown2: TUpDown;
  Image3: TImage;
  DPos2: TEdit;
  DXPos: TEdit;
  UpDown3: TUpDown;
  DPos1: TEdit;
  Panel3: TPanel;
  XY1: TSpeedButton;
  Real: TSpeedButton;
  Store: TSpeedButton;
  XY11: TSpeedButton;
  Ch1: TSpeedButton;
  Ch2: TSpeedButton;
  Dual: TSpeedButton;
  Add: TSpeedButton;
  ChkRam: TTimer;
  Label20: TLabel;
  Plot: TImage;
  LV1: TLabel;
  LV2: TLabel;
  LY1: TLabel;
  LY2: TLabel;
  LT: TLabel;
  LX: TLabel;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LevelIV: TzKnob;
 UpDown4: TUpDown;
 AnalysisSpectrum1: TMenuItem;
 SavePictureDialog1: TSavePictureDialog;
 Discreat1: TMenuItem;
 Continues1: TMenuItem;
 OpenPictureDialog1: TOpenPictureDialog;
 z3DLabel1: Tz3DLabel;
 z3DLabel2: Tz3DLabel;
 z3DLabel3: Tz3DLabel;
 z3DLabel4: Tz3DLabel;
 z3DLabel5: Tz3DLabel;
 z3DLabel6: Tz3DLabel;
 z3DLabel7: Tz3DLabel;
 z3DLabel8: Tz3DLabel;
 z3DLabel9: Tz3DLabel;
 z3DLabel10: Tz3DLabel;
 z3DLabel11: Tz3DLabel;
 z3DLabel12: Tz3DLabel;
 z3DLabel13: Tz3DLabel;
 z3DLabel14: Tz3DLabel;
 z3DLabel15: Tz3DLabel;
 Ch2Voll: TzKnob;
 z3DLabel16: Tz3DLabel;
 z3DLabel17: Tz3DLabel;
 z3DLabel18: Tz3DLabel;
 z3DLabel19: Tz3DLabel;
 z3DLabel20: Tz3DLabel;
 z3DLabel21: Tz3DLabel;
 z3DLabel22: Tz3DLabel;
 z3DLabel23: Tz3DLabel;
 z3DLabel24: Tz3DLabel;
 z3DLabel25: Tz3DLabel;
 z3DLabel26: Tz3DLabel;
 z3DLabel27: Tz3DLabel;
 z3DLabel28: Tz3DLabel;
 z3DLabel29: Tz3DLabel;
 z3DLabel30: Tz3DLabel;
 z3DLabel31: Tz3DLabel;
 z3DLabel32: Tz3DLabel;
 ColorDialog1: TColorDialog;
 Channel12: TMenuItem;
 Channel22: TMenuItem;
 SetGrid2: TMenuItem;
 N1Volt1: TMenuItem;
 N8Volt1: TMenuItem;
 N04Volt1: TMenuItem;
 N02Volt1: TMenuItem;
 N01Volt1: TMenuItem;
 N40mVolt1: TMenuItem;
 N20mVolt1: TMenuItem;
 N10mVolt1: TMenuItem;
 N5mVolt1: TMenuItem;
 N1Volt2: TMenuItem;
 N08Volt2: TMenuItem;
 N04Volt2: TMenuItem;
 N02Volt2: TMenuItem;
 N01Volt2: TMenuItem;
 N40mVolt2: TMenuItem;
 N20mVolt2: TMenuItem;
 N10mVolt2: TMenuItem;
 N5mVolt2: TMenuItem;

Time: TzKnob;
 z3DLabel33: Tz3DLabel;
 z3DLabel34: Tz3DLabel;
 z3DLabel35: Tz3DLabel;
 z3DLabel36: Tz3DLabel;
 z3DLabel37: Tz3DLabel;
 z3DLabel38: Tz3DLabel;
 z3DLabel39: Tz3DLabel;
 z3DLabel40: Tz3DLabel;
 z3DLabel41: Tz3DLabel;
 z3DLabel42: Tz3DLabel;
 z3DLabel43: Tz3DLabel;
 z3DLabel44: Tz3DLabel;
 On1: TMenuItem;
 SpectrumAnalysis: TPanel;
 z3DLabel45: Tz3DLabel;
 SpectrumType: TRadioGroup;
 Windows: TComboBox;
 FFT1: TFastFourier;
 PnnterSetupDialog1: TPrinterSetupDialog;
 PrintDialog1: TPrintDialog;
 PHz: TLabel;
 Center: TLabel;
 S1: TLabel;
 S2: TLabel;
 S3: TLabel;
 S4: TLabel;
 procedure FormCreate(Sender: TObject);
 procedure Image2MouseMove(Sender: TObject; Shift: TShiftState;
 X, Y: Integer);
 procedure Image2MouseDown(Sender: TObject; Button:
 TMouseButton;
 Shift: TShiftState; X, Y: Integer);
 procedure Image2MouseUp(Sender: TObject; Button:
 TMouseButton;
 Shift: TShiftState; X, Y: Integer);
 procedure DateTimeTimer(Sender: TObject);
 procedure ThaiFont1Click(Sender: TObject);
 procedure XY1Click(Sender: TObject);
 procedure Inv1Click(Sender: TObject);
 procedure Inv2Click(Sender: TObject);
 procedure UpDown1Changing(Sender: TObject; var AllowChange:
 Boolean);
 procedure Y1PosChange(Sender: TObject);
 procedure UpDown2Changing(Sender: TObject; var AllowChange:
 Boolean);
 procedure Y2PosChange(Sender: TObject);
 procedure ChkRamTimer(Sender: TObject);
 procedure AC1Click(Sender: TObject);
 procedure DC1Click(Sender: TObject);
 procedure RealClick(Sender: TObject);
 procedure StoreClick(Sender: TObject);
 procedure XPosTChange(Sender: TObject);
 procedure UpDown3Changing(Sender: TObject; var AllowChange:
 Boolean);
 procedure LevelIVChange(Sender: TObject);
 procedure UpDown4Changing(Sender: TObject; var AllowChange:
 Boolean);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure EnglishFont1Click(Sender: TObject);
procedure Export1Click(Sender: TObject);
procedure AnalysisSpectrum1Click(Sender: TObject);
procedure Cont1Click(Sender: TObject);
procedure Index1Click(Sender: TObject);
procedure Discreat1Click(Sender: TObject);
procedure Continues1Click(Sender: TObject);
procedure Import1Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Ch1VoltChange(Sender: TObject);
procedure Ch2VoltChange(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Channel12Click(Sender: TObject);
procedure Channel22Click(Sender: TObject);
procedure SelGrid2Click(Sender: TObject);
procedure TimeChange(Sender: TObject);
procedure On1Click(Sender: TObject);
procedure WindowsChange(Sender: TObject);
procedure SpectrumTypeClick(Sender: TObject);
procedure PrintSetup1Click(Sender: TObject);
procedure Print1Click(Sender: TObject);
procedure AC1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure DC1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure GND1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure AC2Click(Sender: TObject);
procedure DC2Click(Sender: TObject);
procedure GND2Click(Sender: TObject);
procedure About1Click(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
procedure PaintMonitor;
procedure PaintMark(ox1,ox2:Integer);
procedure ProcessMark(ox1,ox2:Integer);
procedure PaintPosY(Pos1,Pos2:Integer);
procedure DisplaySignal;
procedure DisplaySpectrum;
procedure DisplayGnd;
procedure LoadInitial;
procedure Channel1Chang;
procedure Channel2Chang;
procedure TimebaseChang;
procedure CallTime1(Position:Integer);
procedure CallTime2(Position:Integer);
procedure DisplayForPrintSignal;
procedure DisplayForPrintSpectrum;
procedure CalVolt(XMark,Mark:Integer);
end;
Const
ScreenLeft=16;
ScreenTop=45;
GainPort=$306;
AltPort=$304;
SamplingPort=$303;
ClockReadPort=$305;
//ClamPort=$306;
RamStatePort=$307;

ReadToPort=$302;
ReadStopPort=$303;

R1K=$2; //0000 0010
R2K=$4; //0000 0100
R4K=$8; //0000 1000
R8K=$10; //0001 0000
R16K=$20; //0010 0000
R32K=$40; //0100 0000
R64K=$82; //1000 0010
R128K=$1; //0000 0001
PixelGndInvAc=66;
PixelGndInvDc=110;
PixelGndNorAc=329;
PixelGndNorDc=290;

var
Scope: TScope;
Old_ox1,Old_ox2,Origin1,Origin2,MarkSelect,cl,Old_Y1,Old_Y2;
x1,x2:Integer;
Data:Array [0..131072] of byte;
DataBuff:Array [0..1024] of byte;
SamplingRate,RamState,AltClf,GainClf,LevelTrig:byte;
DivRam,MonitorColor,Color1,Color2,GndNor,GndInv,DPos,YPos1,YPos2,XPos,RamE,RBuff,NTemp:integer;
GndC1,Run:Boolean;
MonitorName,TimeScale:String;
TB,I1,I2,DT12,Frq,Rate:Extended;
XMark1,XMark2,Temp:integer;
implementation
uses PrintForm,PGSkin;
{$R 'DFM'}
procedure OutPort(Address,D:word):Assembler;
begin
asm
push dx
push ax
mov ax,d
mov dx,address
out dx,ax
pop ax
pop dx
end;
end;

function InPort(Address:word):byte;assembler;
begin
asm
push dx
push ax
mov dx,address
in al,dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov result,al
pop ax
pop dx
end;
end;

procedure TScope.LoadInitial;
var IniFile:TIniFile;
    LevelTrig,TimeBase,VoltDiv,ColorS,YPos,XPos:integer;
    FontShow,ScreenShow,Mode:String;
    TrigOn,Invert:Boolean;

begin
    IniFile:=TIniFile.Create('PCScope.ini');
    // Channel 1 Load parameter from PCScope.ini
    with IniFile do
        begin
            VoltDiv:=ReadInteger('Channel 1','VoltDiv',0);
            ColorS:=ReadInteger('Channel 1','Color',0);
            YPos:=ReadInteger('Channel 1','YPos',0);
            Mode:=ReadString('Channel 1','Mode','');
            Invert:=ReadBool('Channel 1','Invert',False);
            end;

            Ch1Volt.Value:=VoltDiv;
            Channel1Chang;
            Color1:=ColorS;
            Shape1.Brush.Color:=Color1;
            Y1Pos.Value:=YPos;
            if Mode='AC' then AC1.Down:=True;
            if Mode='DC' then DC1.Down:=True;
            if Mode='GND' then GND1.Down:=True;

            if Invert then
                begin
                    //Invert plot
                    Inv1.Tag:=1;
                    Inv1.Down:=True;
                    //Inv11.Down:=True;
                end
            else
                begin
                    //Noninvert plot
                    Inv1.Tag:=0;
                    Inv1.Down:=False;
                    //Inv11.Down:=False;
                end;

            // Channel 2 Load parameter from PCScope.ini
            with IniFile do
                begin
                    VoltDiv:=ReadInteger('Channel 2','VoltDiv',0);
                    ColorS:=ReadInteger('Channel 2','Color',0);
                    YPos:=ReadInteger('Channel 2','YPos',0);
                    Mode:=ReadString('Channel 2','Mode','');
                    Invert:=ReadBool('Channel 2','Invert',False);
                    end;

                    Ch2Volt.Value:=VoltDiv;
                    Channel2Chang;
                    //Color1:=ColorS;
                    Y2Pos.Value:=YPos;

                    if Mode='AC' then AC2.Down:=True;
                    if Mode='DC' then DC2.Down:=True;
                    if Mode='GND' then GND2.Down:=True;

                    // Invert then
                    //begin
                    // //Invert plot
                    // Inv2.Tag:=1;
                    // Inv2.Down:=True;
                    // //Inv22.Down:=True;
                    // end
                    // else
                    // begin
                    // //Noninvert plot
                    // Inv2.Tag:=0;
                    // Inv2.Down:=False;
                    // //Inv22.Down:=False;
                    // end;

                    //Load Time Setting From INI File
                    with IniFile do
                        begin
                            TimeBase:=ReadInteger('TimeBase','TimeDiv',0);
                            LevelTrig:=ReadInteger('TimeBase','Level',0);
                            XPos:=ReadInteger('TimeBase','XPos',0);
                            end;

                            Time.Value:=TimeBase;
                            TimebaseChang;
                            LevelV.Value:=LevelTrig;
                            XPosT.Value:=XPos;

                    //Setup Parameter Load From INI File
                    with IniFile do
                        begin
                            MonitorColor:=ReadInteger('Setup','MonitorColor',0);
                            MonitorName:=ReadString('Setup','MonitorFile','');
                            FontShow:=ReadString('Setup','Font','');
                            ScreenShow:=ReadString('Setup','Screen','');
                            TrigOn:=ReadBool('Setup','TrigOn',False);

                            P90.Font.Color:=MonitorColor;
                            P10.Font.Color:=MonitorColor;
                            if FontShow='English' then
                                begin
                                    EnglishFont1.Click;
                                end
                            else
                                begin
                                    ThaiFont1.Click;
                                end;

                            if ScreenShow='Discreat' then
                                begin
                                    Discreat1.Click;
                                end
                            else
                                begin
                                    Continues1.Click;
                                end;

                            On1.Checked:=TrigOn;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

end;

procedure TScope.Channel1Chang;
begin
  case Ch1Volt.Value of
  1:begin
    LV1.Caption:=1 V/Div;
    GainCtl:=1./1;
    AttCtl:=1;
    Rate:=3.9;
    if not Inv1.Down then
      DPos:=-466
    else
      DPos:=466;
    end;
  2:begin
    LV1.Caption:=0.8 V/Div;
    GainCtl:=2./1
    AttCtl:=1;
    Rate:=2.1;
    if not Inv1.Down then
      DPos:=-145
    else
      DPos:=145;
    end;
  3:begin
    LV1.Caption:=0.4 V/Div;
    GainCtl:=3./2
    AttCtl:=1;
    Rate:=2.2;
    if not Inv1.Down then
      DPos:=-127
    else
      DPos:=127;
    end;
  4:begin
    LV1.Caption:=0.2 V/Div;
    GainCtl:=0;
    AttCtl:=0;
    Rate:=1.8;
    if not Inv1.Down then
      DPos:=-100
    else
      DPos:=100;
    end;
  5:begin
    LV1.Caption:=0.1 V/Div;
    GainCtl:=1;
    AttCtl:=0;
    Rate:=1.8;
    if not Inv1.Down then
      DPos:=-100
    else
      DPos:=100;
    end;
  6:begin
    LV1.Caption:=40 mV/Div;
    GainCtl:=2;
    AttCtl:=0;
    Rate:=1.9;
    if not Inv1.Down then
      DPos:=-115
    else
      DPos:=115;
    end;
  7:begin
    LV1.Caption:=20 mV/Div;
    GainCtl:=3;
    AttCtl:=0;
    Rate:=1.9;
    if not Inv1.Down then
      DPos:=-110
    else
      DPos:=110;
    end;
  8:begin
    LV1.Caption:=10 mV/Div;
    GainCtl:=4;
    AttCtl:=0;
    Rate:=2;
    if not Inv1.Down then
      DPos:=-125
    else
      DPos:=125;
    end;
  9:begin
    LV1.Caption:=5 mV/Div;
    GainCtl:=5;
    AttCtl:=0;
    Rate:=2.4;
    if not Inv1.Down then
      DPos:=-180
    else
      DPos:=180;
    end;
  end;
end;

procedure TScope.Channel2Chang;
begin
  case Ch2Volt.Value of
  1:begin
    LV2.Caption:=1 V/Div;
    end;
  2:begin
    LV2.Caption:=0.8 V/Div;
    end;
  3:begin
    LV2.Caption:=0.4 V/Div;
    end;
  4:begin
    LV2.Caption:=0.2 V/Div;
    end;
  5:begin
    LV2.Caption:=0.1 V/Div;
    end;
  6:begin
    LV2.Caption:=40 mV/Div;
    end;
  7:begin
    LV2.Caption:=20 mV/Div;
    end;
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

8.begin
    LV2.Caption:='10 mV/Div';
end;
9.begin
    LV2.Caption:='5 mV/Div';
end;
end;
end;

procedure TScope.TimebaseChang;
begin
case Time.Value of
1:begin
//Set Spectrum Rang
S1.Caption:='1k';
S2.Caption:='2k';
S3.Caption:='3k';
S4.Caption:='4k';
//Set Time Base
LT.Caption:='2.5 mS/Div';
RamE:=R128K;
RBuf:=128*1024;
DivRam:=256;
SamPlingRate:=3;
ChkRam.Interval:=400;
TimeScale:=' mS';
TB:=2.5;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
2:begin
//Set Spectrum Rang
S1.Caption:='2k';
S2.Caption:='4k';
S3.Caption:='6k';
S4.Caption:='8k';
//Set Time Base
LT.Caption:='1.2 mS/Div';
RamE:=R128K;
RBuf:=128*1024;
DivRam:=256;
SamPlingRate:=2;
ChkRam.Interval:=400;
TimeScale:=' mS';
TB:=1.2;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
3:begin
//Set Spectrum Rang
S1.Caption:='4k';
S2.Caption:='8k';
S3.Caption:='12k';
S4.Caption:='16k';
//Set Time Base
LT.Caption:='0.64 mS/Div';
RamE:=R128K;
RBuf:=128*1024;
DivRam:=128;
SamPlingRate:=2;
ChkRam.Interval:=400;
TimeScale:=' mS';
TB:=0.64;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
4:begin
//Set Spectrum Rang
S1.Caption:='8k';
S2.Caption:='16k';
S3.Caption:='24k';
S4.Caption:='32k';
//Set Time Base
LT.Caption:='0.32 mS/Div';
RamE:=R128K;
RBuf:=128*1024;
DivRam:=64;
SamPlingRate:=2;
ChkRam.Interval:=400;
TimeScale:=' mS';
TB:=0.32;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
5:begin
//Set Spectrum Rang
S1.Caption:='16k';
S2.Caption:='32k';
S3.Caption:='48k';
S4.Caption:='64k';
//Set Time Base
LT.Caption:='0.16 mS/Div';
RamE:=R16K;
RBuf:=16*1024;
DivRam:=32;
SamPlingRate:=2;
ChkRam.Interval:=300;
TimeScale:=' mS';
TB:=0.16;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
6:begin
//Set Spectrum Rang
S1.Caption:='32k';
S2.Caption:='64k';
S3.Caption:='96k';
S4.Caption:='128k';
//Set Time Base
LT.Caption:='80 uS/Div';
RamE:=R8K;
RBuf:=8*1024;
DivRam:=1;
SamPlingRate:=8;
ChkRam.Interval:=300;
TimeScale:=' uS';
TB:=80;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
7:begin
//Set Spectrum Rang
S1.Caption:='64k';
S2.Caption:='128k';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

S3.Caption:='192k';
S4.Caption:='256k';
//Set Time Base
LT.Caption:='40 uS/Div';
RamE:=R8K;
RBuf:=8*1024;
DivRam:=1;
SamPlingRate:=5;
ChkRam.Interval:=300;
TimeScale:=' uS';
TB:=40;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
8:begin
//Set Spectrum Rang
S1.Caption:='128k';
S2.Caption:='256k';
S3.Caption:='384k';
S4.Caption:='512k';
//Set Time Base
LT.Caption:='20 uS/Div';
RamE:=R8K;
RBuf:=8*1024;
DivRam:=1;
SamPlingRate:=4;
ChkRam.Interval:=300;
TimeScale:=' uS';
TB:=20;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
9:begin
//Set Spectrum Rang
S1.Caption:='256k';
S2.Caption:='512k';
S3.Caption:='768k';
S4.Caption:='1024k';
//Set Time Base
LT.Caption:='10 uS/Div';
RamE:=R4K;
RBuf:=4*1024;
DivRam:=1;
SamPlingRate:=3;
ChkRam.Interval:=300;
TimeScale:=' uS';
TB:=10;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
10:begin
//Set Spectrum Rang
S1.Caption:='512k';
S2.Caption:='1024k';
S3.Caption:='1536k';
S4.Caption:='2048k';
//Set Time Base
LT.Caption:='5 uS/Div';
RamE:=R4K;
RBuf:=4*1024;
DivRam:=1;
SamPlingRate:=2;
ChkRam.Interval:=300;
TimeScale:=' uS';
TB:=5;
// ChkRam.Interval:=StrToInt(Edit5.Text);
// Label9.Caption:=IntToStr(ChkRam.Interval);
end;
end;
procedure TScope.CallTime1(Position:Integer);
var DataTemp:String;
begin
if (Position>=10) and (Position<=510) then
begin
t1:=TB*(Position-10)*2/100;
DT12:=t2-t1;
if DT12<>0 then Frq:=(1/DT12)*1000;
ST1.Caption:=FloatToStr(t1)+TimeScale;
DT.Caption:=FloatToStr(DT12, #Number, 1, 2)+TimeScale;
if TimeScale=' uS' then
begin
DataTemp:=Copy(FloatToStr(Frq), 1, Pos(' ', floatToStr(Frq))+3);
F1.Caption:=DataTemp+' KHz';
end;
if TimeScale=' mS' then
begin
DataTemp:=Copy(FloatToStr(Frq), 1, Pos(' ', floatToStr(Frq))+3);
F1.Caption:=DataTemp+' Hz';
end;
end;
end;
end;
procedure TScope.CallTime2(Position:Integer);
var DataTemp:String;
begin
if (Position>=10) and (Position<=510) then
begin
t2:=TB*(Position-10)*2/100;
DT12:=t2-t1;
if DT12<>0 then Frq:=(1/DT12)*1000;
ST2.Caption:=FloatToStr(t2)+TimeScale;
DT.Caption:=FloatToStr(DT12, #Number, 1, 2)+TimeScale;
if TimeScale=' uS' then
begin
DataTemp:=Copy(FloatToStr(Frq), 1, Pos(' ', floatToStr(Frq))+3);
F1.Caption:=DataTemp+' KHz';
end;
if TimeScale=' mS' then
begin
DataTemp:=Copy(FloatToStr(Frq), 1, Pos(' ', floatToStr(Frq))+3);
F1.Caption:=DataTemp+' Hz';
end;
end;
end;
end;
end;
procedure TScope.PaintMonitor;
var x,y:integer;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Paint col and row line
// Paint row
for y:=0 to 9 do
begin
for x:=0 to 500 do
begin
Plot.Canvas.Pixels[x,y*50]:=MonitorColor;
end;
end;
//Paint col
for x:=0 to 11 do
begin
for y:=0 to 400 do
begin
Plot.Canvas.Pixels[x*50,y]:=MonitorColor;
end;
end;
//Paint Gnde
x:=0;
while x<=490 do
begin
for y:=-4 to 4 do Plot.Canvas.Pixels[x,200+y]:=MonitorColor;
x:=x+10;
end;
y:=0;
while y<=400 do
begin
for x:=-4 to 4 do Plot.Canvas.Pixels[250+x,y]:=MonitorColor;
y:=y+10;
end;
//Paint 90% and 10% Line
x:=0;y:=0;
while x<500 do
begin
for y:=0 to 5 do
begin
Plot.Canvas.Pixels[x+y,(P10.Top)360]:=MonitorColor;
Plot.Canvas.Pixels[x+y,(P90.Top)40]:=MonitorColor;
end;
x:=x+10;
end;
Plot.Picture.SaveToFile(MonitorName)
P90.Font.Color:=MonitorColor;
P10.Font.Color:=MonitorColor;
end;

procedure TScope.PaintMark(ox1,ox2:integer);
var x,y:integer;
begin
//clear Old Mark 1
for y:=0 to 10 do
begin
for x:=0 to y do
Image2.Canvas.Pixels[ScreenLeft-6+Old_ox1-Round(y/2)+x,9-
y]:=clBlack;
end;

//new Mark 1
for y:=0 to 10 do
begin
for x:=0 to y do
Image2.Canvas.Pixels[ScreenLeft-6+ox1-Round(y/2)+x,9-
y]:=clRed;
end;

//clear Old Mark 2
for y:=0 to 10 do
begin
for x:=0 to y do
Image2.Canvas.Pixels[ScreenLeft-6+Old_ox2-Round(y/2)+x,9-
y]:=clBlack;
end;

//new Mark 2
for y:=0 to 10 do
begin
for x:=0 to y do
Image2.Canvas.Pixels[ScreenLeft-6+ox2-Round(y/2)+x,9-
y]:=clYellow;
end;
Mark1.Left:=ScreenLeft+ox1+2;
Origin1:=ox1+5;
Old_ox1:=ox1;
Mark2.Left:=ScreenLeft+ox2+2;
Origin2:=ox2+5;
Old_ox2:=ox2;
//Label16.Caption:='Origin1 '+IntToStr(Origin1);
end;

procedure TScope.PaintPosY(Pos1,Pos2:integer);
var x,y:integer;
begin
//clear Old Mark
for x:=0 to 10 do
begin
for y:=0 to x do
Image3.Canvas.Pixels[x,10+200-Round(x/2)+y-
Old_Y1]:=clBlack;
end;

//new Mark
for x:=0 to 10 do
begin
for y:=0 to x do
Image3.Canvas.Pixels[x,10+200-Round(x/2)+y-Pos1]:=clGreen;
end;

//clear Old Mark
for x:=0 to 10 do
begin
for y:=0 to x do
Image3.Canvas.Pixels[x,10+200-Round(x/2)+y-
Old_Y2]:=clBlack;
end;

//new Mark
for x:=0 to 10 do
begin
for y:=0 to x do
Image3.Canvas.Pixels[x,10+200-Round(x/2)+y-Pos2]:=clGreen;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
Old_Y1:=Pos1;
Old_Y2:=Pos2;
end;

procedure TScope.ProcessMark(o1,o2:(integer);
begin
if o1<Screen.Left-7 then
begin
o1:=-1;
end
else
begin
if o1>500+Screen.Left-6 then
begin
o1:=500;
end
else
begin
o1:=o1-10;
end;
end;

if o2<Screen.Left-7 then
begin
o2:=-1;
end
else
begin
if o2>500+Screen.Left-6 then
begin
o2:=500;
end
else
begin
o2:=o2-10;
end;
end;
PaintMark(o1,o2);
end;

procedure TScope.DisplaySignal;
var Exit,Smooth:Boolean;
n,yo:Integer;
DataTemp:byte;
begin
//Rate:=StrToFloat(Edit5.Text);
Plot.Picture.LoadFromFile('Monitor.bmp');
NTemp:=0;
Exit:=False;
NTemp:=10;

if Qn1.Checked then
begin
Repeat
DataTemp:=Data[NTemp];
NTemp:=NTemp+1;

if NTemp<RBuf then
begin
if (DataTemp=LevelTrig) and (Data[NTemp+1]>=LevelTrig) and
(Data[NTemp+2]>=LevelTrig) and
(Data[NTemp+3]>=LevelTrig) and
(Data[NTemp+4]>=LevelTrig) then
begin
NTemp:=NTemp;
Exit:=True;
end;
end
else
begin
Exit:=True;
end;
Until Exit=True;
end
else
begin
NTemp:=200;
end;

//NTemp:=StrToInt(Edit4.Text);

if (NTemp+(500*DivRam))<=RBuf then
begin
n:=NTemp;
While n<=(NTemp+(500*DivRam)) do
begin
case Inv1.Tag of
0:yo:=GndNor-Round(Rate*data[n])-(YPos1+DPos); //Normal
Plot
1:yo:=GndInv+Round(Rate*data[n])-(YPos1-DPos); //Invert plot
end;

if Continues1.Checked then
begin
Plot.Canvas.MoveTo((n-NTemp) div DivRam+XPos,yo);
//DataBuff[n-NTemp]:=yo;
Plot.Canvas.Pen.Color:=Color1;
case Inv1.Tag of
0:yo:=GndNor-Round(Rate*data[n+DivRam])-(YPos1+DPos);
//Normal Plot
1:yo:=GndInv+Round(Rate*data[n+DivRam])-(YPos1-DPos);
//Invert plot
end;
end;
Plot.Canvas.LineTo((n-NTemp) div DivRam+XPos,yo);
end
else
begin
Plot.Canvas.Pixels[(n-NTemp) div DivRam+XPos,yo]:=Color1;
end;
//DataBuff[n-NTemp]:=yo;
n:=n+DivRam;
end;

end;
Label20.Caption:=IntToStr(n);

end;

procedure TScope.DisplayForPrintSpectrum;
var n,nfft:integer;
y0,y1:Double;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

1.y0:=GndInv+Round(Rate*data[n+DivRam])-(YPos1-DPos);
//invert plot
end;
PrnForm.BWMon.Canvas.LineTo((n-NTemp) div
DivRam+XPos,y0);
end
else
begin
PrnForm.BWMon.Canvas.Pixels[(n-NTemp) div
DivRam+XPos,y0]:=clBlack;
end;
n:=n+DivRam;
end;
end;
end;

procedure TScope.DisplaySpectrum;
var n,nfft:integer;
y0,y1:Double;
begin
FFT1.ClearImag;
FFT1.SpectrumSize:=1024;
Plot.Picture.LoadFromFile('SpecMon.bmp');
NTemp:=200;
if (NTemp+(500*DivRam))<=RBuf then
begin
n:=NTemp;
While n<=(NTemp+(500*DivRam)) do
begin
n:=n+DivRam;
nfft:=(n-NTemp+DivRam) Div 2;
FFT1.RealSpec[nfft]:=4*data[n+DivRam];
end;
end;
FFT1.Transform;
for nfft:=1 to 500[250]503 do
begin
case SpectrumType.ItemIndex of
0 : begin
y0 := FFT1.Magnitude[nfft*-2];
y1 := FFT1.Magnitude[nfft+1];
end;
1 : begin
y0 := 0.00002*FFT1.PowerSpec[nfft];
y1 := 0.00002*FFT1.PowerSpec[nfft+1];
end;
2 : begin
y0 := 20*FFT1.Phase[nfft];
y1 := 20*FFT1.Phase[nfft+1];
end;
3 : begin
y0 := FFT1.FourSerCosCoeff[nfft];
y1 := FFT1.FourSerCosCoeff[nfft+1];
end;
4 : begin
y0 := FFT1.FourSerSinCoeff[nfft];
y1 := FFT1.FourSerSinCoeff[nfft+1];
end;
5 : begin
y0 := 0.004*FFT1.RealSpec[nfft+1];
y1 := 0.004*FFT1.RealSpec[nfft+2];

```

```

end;
b := begin
y0 := 0.004*FFT1.ImagSpec[nfft+1];
y1 := 0.004*FFT1.ImagSpec[nfft+2];
end;
end;

if Continues1.Checked then
begin
Plot.Canvas.Pen.Color:=Color1;
//Plot Positive Frequency
Plot.Canvas.MoveTo(nfft,Round(200-y0));
Plot.Canvas.LineTo(nfft+1,Round(200-y1));
end
else
begin
//Plot Positive Frequency
Plot.Canvas.Pixels[nfft,Round(200-y0)]:=Color1;
end;
end;
end;
end;
end;

procedure TScope.DisplayGnd;
var n,y0:integer;
begin
Plot.Picture.LoadFromFile('Monitor.bmp');
for n:=50 to 550 do
begin
y0:=320-(data[n]-YPos1);
Plot.Canvas.Pixels[n-50+XPos,y0]:=clRed;
end;
Label20.Caption:=IntToStr(Temp);
end;

procedure TScope.CalVolt(XMark,Mark:integer);
begin
end;

procedure TScope.FormCreate(Sender:TObject);
begin
Application.HintColor:=clYellow;
Application.HintPause:=300;
Sleep(2000);
SplashScreen.Hide;

Mark1.Top:=48;Mark2.Top:=48;Mark1.Height:=400;Mark2.Height:=400;
0;
Old_ox1:=0;Old_ox2:=500;Origin1:=5;Origin2:=506;
x1:=-1;x2:=510;
P90.Left:=2;P10.Left:=2;P90.Top:=80;P10.Top:=400;
PaintMark(-1.500);
PaintPosY(0,0);
GndC1:=False;
YPos1:=0;
YPos2:=0;
//XPos:=0;
RamState:=S0;
//0110 1011v=2.464 v
//VoltClam:=S6B;
Plot.Height:=401;Plot.Width:=691;
Plot.Top:=ScreenTop+2;Plot.Left:=ScreenLeft+2;
LY1.Caption:=DPos1.Text;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LY2.Caption:=DPos2.Text;
LX.Caption:=DXPos2.Text;
Temp:=0;
Sleep(50);
LevelTrig:=$80;
Run:=False;
//RunTime.Enabled:=false;
TimeBase.Height:=227;
TimeBase.Width:=178;
TimeBase.Top:=6;
TimeBase.Left:=542;

Discreat1.Checked:=True;
Continues1.Checked:=False;

GndNor:=PixelGndNorAc;
GndInv:=PixelGndInvAc;

AttCl:=0;
GainCl:=0;
//GainCl:=StrToInt(Edit2.Text);
OutPort(GainPort,GainPort); //0000 0000B
OutPort(AttPort,AttCl); //0000 0010B

LoadInitial;

end;

procedure TScope.Image2MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
if Cl=1 then
begin
case MarkSelect of
1:begin
x1:=x;
XMark1:=x1;
ProcessMark(x1,x2);
CallTime1(x1);
//Label9.Caption:=IntToStr(x);
CalVolt(XMark1,1);
end;
2:begin
x2:=x;
XMark2:=x2;
ProcessMark(x1,x2);
CallTime2(x2);
//Label10.Caption:=IntToStr(x2);
CalVolt(XMark2,2);
end;
end;
end;
end;

procedure TScope.Image2MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
Cl:=1;
if (x>Origin1) and (x<(Origin1+15)) then
begin
MarkSelect:=1;
end
else
begin
if (x>Origin2) and (x<(Origin2+15)) then
begin
MarkSelect:=2;
end
else
begin
MarkSelect:=0;
end;
end;
end;

procedure TScope.Image2MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
Cl:=0;
end;

procedure TScope.DateTimeTimer(Sender: TObject);
begin
StatusBar1.Panels.Items[1].Text:=DateTimeToStr(Now);
end;

procedure TScope.ThaiFont1Click(Sender: TObject);
begin
ThaiFont1.Checked:=True;
EnglishFont1.Checked:=False;
Scope.Caption:= 'คอมพิวเตอร์ไทยและวิศวกรรมศาสตร์';
File1.Caption:='&ไฟล์';
Open1.Caption:='&เปิด';
Save1.Caption:='&บันทึกเพิ่ม';
Saveas1.Caption:='บันทึกเพิ่มเป็น...';
PrintSetup1.Caption:='ปรับ&แต่งก่อนพิมพ์';
Print1.Caption:='&พิมพ์';
Report1.Caption:='&รายงาน';
Export1.Caption:='&นำออกไปเป็น...';
Import1.Caption:='&นำเข้ามาเป็น...';
Exit1.Caption:='&ออกจากโปรแกรม';

Control1.Caption:='&ควบคุม';
Time1.Caption:='&เวลา';
Channel11.Caption:='ช่องที่ &1';
Channel21.Caption:='ช่องที่ &2';
Trigger1.Caption:='&ม&ตรึง';
On1.Caption:='&ทำงาน';

Measure1.Caption:='&วัด';
Channel1Only.Caption:='ช่องที่ &1 อย่างเดียว';
Channel2Only1.Caption:='ช่องที่ &2 อย่างเดียว';
All1.Caption:='&ทุกช่อง';
AnalysisSpectrum1.Caption:='วิเคราะห์&สเปกตรัม';

Setup1.Caption:='&ปรับแต่ง';
SetGnd2.Caption:='ปรับ&แต่ง&กร&บ';
SetGnd1.Caption:='ปรับ&แต่ง&ส&ร&ง&ร&บ';
Channel12.Caption:='ช่องที่ &1';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Channel22.Caption:='ช่องที่ 2';
SetScreen1.Caption:='ปรับแต่งแฉกในกราฟ';
Discreat1.Caption:='พัลส์เป็นจุด';
Continues1.Caption:='พัลส์เป็นเส้นต่อเนื่อง';
SetFont1.Caption:='ปรับแฟอนต์อักษร';
ThaiFont1.Caption:='ภาษาไทย';
EnglishFont1.Caption:='ภาษาอังกฤษ';

Help1.Caption:='&ช่วยเหลือ';
Cont1.Caption:='&เนื้อหา';
Index1.Caption:='&ดัชนี';
About1.Caption:='&เกี่ยวกับโปรแกรม';

end;

procedure TScope.EnglishFont1Click(Sender: TObject);
begin
  ThaiFont1.Checked:=False;
  EnglishFont1.Checked:=True;
  Scope.Caption:='Computer Oscilloscope & Spectrum Analysis';

  File1.Caption:='&File';
  Open1.Caption:='&Open';
  Save1.Caption:='&Save';
  Saves1.Caption:='Save &as...';
  PrintSetup1.Caption:='Print &Setup...';
  Print1.Caption:='&Print';
  Report1.Caption:='R&eport';
  Export1.Caption:='Expo&rt...';
  Import1.Caption:='&Import...';
  Exit1.Caption:='E&xit';

  Control1.Caption:='&Control';
  Time1.Caption:='&Time';
  Channel11.Caption:='Channel &1';
  Channel21.Caption:='Channel &2';
  Trigger1.Caption:='Tri&gger';
  On1.Caption:='&On';

  Measure1.Caption:='&Measure';
  Channel1Only1.Caption:='Channel&1 Only';
  Channel2Only1.Caption:='Channel&2 Only';
  All1.Caption:='&All Channel';
  AnalysisSpectrum1.Caption:='Analysis &Spectrum';

  Setup1.Caption:='&Setup';
  SetGrid2.Caption:='Setup &Grid';
  SetGrid1.Caption:='Setup &Color Signal';
  Channel12.Caption:='Channel &1';
  Channel22.Caption:='Channel &2';
  SetScreen1.Caption:='Setup &Screen';
  Discreat1.Caption:='Dis&creat';
  Continues1.Caption:='&Continues';
  SetFont1.Caption:='Setup &Font';
  ThaiFont1.Caption:='&Thai Font';
  EnglishFont1.Caption:='&English Font';

  Help1.Caption:='&Help';
  Cont1.Caption:='&Content';
  Index1.Caption:='&Index';
  About1.Caption:='&About...';

end;

procedure TScope.XY1Click(Sender: TObject);
begin
  if xy1.Tag=0 then
    begin
      xy1.Tag:=1;
      xy11.Down:=False;
    end
  else
    begin
      xy1.Tag:=0;
      xy11.Down:=True;
    end;
end;

procedure TScope.Inv1Click(Sender: TObject);
begin
  if Inv1.Tag=0 then
    begin
      //invert plot
      Inv1.Tag:=1;
      Inv11.Down:=False;
      //StatusBar1.Panels.Items[0].Text:='invert mode';
    end
  else
    begin
      //Noninvert plot
      Inv1.Tag:=0;
      Inv11.Down:=True;
      //StatusBar1.Panels.Items[0].Text:='Non invert mode';
    end;
  DisplaySignal;
end;

procedure TScope.Inv2Click(Sender: TObject);
begin
  if Inv2.Tag=0 then
    begin
      //Invert Plot
      Inv2.Tag:=1;
      Inv22.Down:=False;
      //StatusBar1.Panels.Items[0].Text:='invert mode';
    end
  else
    begin
      //non Invert Plot
      Inv2.Tag:=0;
      Inv22.Down:=True;
      //StatusBar1.Panels.Items[0].Text:='Non invert mode';
    end;
end;

procedure TScope.UpDown1Changing(Sender: TObject);
var AllowChange: Boolean;
begin
  Y1Pos.Value:=UpDown1.Position;
  PaintPosY(UpDown1.Position,UpDown2.Position);
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

YPos1:=UpDown1.Position;
DPos1.Text:=FloatToStrF(YPos1/50,ffNumber,1,4);
LY1.Caption:=DPos1.Text;
//StatusBar1.Panels.Items[0].Text:=UpDown1.Hint+' '+LY1.Caption;
//UpDateY(YPos1);
DisplaySignal;
end;

procedure TScope.Y1PosChange(Sender: TObject);
begin
  UpDown1.Position:=Y1Pos.Value;
  PaintPosY(Y1Pos.Value,Y2Pos.Value);
  YPos1:=Y1Pos.Value;
  DPos1.Text:=FloatToStrF(YPos1/50,ffNumber,1,4);
  LY1.Caption:=(DPos1.Text);
  //StatusBar1.Panels.Items[0].Text:=Y1Pos.Hint+' '+LY1.Caption;
  //UpDateY(YPos1);
  DisplaySignal;
end;

procedure TScope.UpDown2Changing(Sender: TObject);
var AllowChange: Boolean;
begin
  Y2Pos.Value:=UpDown2.Position;
  PaintPosY(Y1Pos.Value,Y2Pos.Value);
  YPos2:=UpDown2.Position;
  DPos2.Text:=FloatToStrF(YPos2/50,ffNumber,1,4);
  LY2.Caption:=Dpos2.Text;
  //StatusBar1.Panels.Items[0].Text:=UpDown2.Hint+' '+LY2.Caption;
end;

procedure TScope.Y2PosChange(Sender: TObject);
begin
  UpDown2.Position:=Y2Pos.Value;
  PaintPosY(UpDown1.Position,UpDown2.Position);
  YPos2:=Y2Pos.Value;
  DPos2.Text:=FloatToStrF(YPos2/50,ffNumber,1,4);
  LY2.Caption:=DPos2.Text;
  //StatusBar1.Panels.Items[0].Text:=Y2Pos.Hint+' '+LY2.Caption;
end;

procedure TScope.ChkRamTimer(Sender: TObject);
var yo,n,r,DataTemp:Integer;
Exit:Boolean;
begin
  OutPort(ClockReadPort,0);
  // set Attenuation and Gain control
  OutPort(AltPort,AltCtl);

  OutPort(GainPort,GainCtl);

  //set sampling clock
  //RamState:=$E7; //1110 0111 clr counter & clr clk
  RamState:=$E7;
  OutPort(RamStatePort,RamState);

  OutPort(SamplingPort,SamPngRate); //0000 0010 10MHz

  // RamState:=$1A; //0001 1010 /ram_select,adc_ena,add_ena,/w
  RamState:=$1A;
  OutPort(RamStatePort,RamState);
  OutPort(AltPort,AltCtl);

  n:=0;
  Repeat
    n:=InPort(ReadStopPort);
  // Label35.Caption:=FloatToStr(n);
  n:=n and RamE ;
  Until n=RamE;
  Label20.Caption:=FloatToStr(n);
  RamState:=$E7; //1110 0111 clr counter & clr clk
  OutPort(RamStatePort,RamState);

  OutPort(SamplingPort,7);

  RamState:=$29; //0010 1001 /ram_select,clr_clock/add_ena,/r
  OutPort(RamStatePort,RamState);
  for r:=1 to 2 do
  begin
    for n:=1 to RBuf do
    begin
      OutPort(ClockReadPort,1);
      OutPort(ClockReadPort,0);
      data[n]:=InPort(ReadToPort);
    end;
  end;
  if TimeBase.Visible then
    DisplaySignal
  else
    DisplaySpectrum;
  end;

procedure TScope.AC1Click(Sender: TObject);
begin
  AltCtl:=0;
  //PixelGndInv:=71;
  GndNor:=PixelGndNorAc;
  GndInv:=PixelGndInvAc;
end;

procedure TScope.DC1Click(Sender: TObject);
begin
  AltCtl:=AltCtl or 2;
  //PixelGndInv:=110;
  GndNor:=PixelGndNorDc;
  GndInv:=PixelGndInvDc;
end;

procedure TScope.RealClick(Sender: TObject);
begin
  ChkRam.Enabled:=True;
end;

procedure TScope.StoreClick(Sender: TObject);
begin
  ChkRam.Enabled:=False;
end;

procedure TScope.XPosTChange(Sender: TObject);
begin
  XPos:=XPos1.Value;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UpDown3.Position:=XPos;
DXPos.Text:=FloatToStrF(XPos/50,fnNumber,1,4);
LX.Caption:=DXPos.Text;
//StatusBar1.Panels.Items[0].Text:=XPosT.Hint+' '*LX.Caption;
//UpDateX(XPos);
DisplaySignal;
end;

procedure TScope.UpDown3Changing(Sender: TObject);
var AllowChange: Boolean;
begin
XPos:=UpDown3.Position;
XPosT.Value:=XPos;
DXPos.Text:=FloatToStrF(XPos/50,fnNumber,1,4);
LX.Caption:=DXPos.Text;
//StatusBar1.Panels.Items[0].Text:=UpDown3.Hint+' '*LX.Caption;
//UpDateX(XPos);
DisplaySignal;
end;

procedure TScope.LevelVChange(Sender: TObject);
begin
LevelTrig:=LevelV.Value;
UpDown4.Position:=LevelTrig;
//StatusBar1.Panels.Items[0].Text:=LevelV.Hint+' '*IntToStr
(LevelTrig);
end;

procedure TScope.UpDown4Changing(Sender: TObject);
var AllowChange: Boolean;
begin
LevelTrig:=UpDown4.Position;
LevelV.Value:=LevelTrig;
//StatusBar1.Panels.Items[0].Text:=UpDown4.Hint+' '*IntToStr
(LevelTrig);
end;

procedure TScope.Export1Click(Sender: TObject);
begin
if SavePictureDialog1.Execute then
Plot.Picture.SaveToFile(SavePictureDialog1.FileName);
end;

procedure TScope.AnalysisSpectrum1Click(Sender: TObject);
begin
if TimeBase.Visible then
begin
P90.Visible:=False;
P10.Visible:=False;
TimeBase.Visible:=False;
Ch1Panel.Enabled:=False;
Ch2Panel.Enabled:=False;
Image2.Visible:=False;
Mark1.Visible:=False;
Mark2.Visible:=False;
Center.Visible:=True;
PHz.Visible:=True;
S1.Visible:=True;
S2.Visible:=True;
end;
end;

procedure TScope.Cont1Click(Sender: TObject);
begin
Application.HelpCommand(HELP_FINDER, 0);
end;

procedure TScope.Index1Click(Sender: TObject);
begin
Application.HelpCommand(HELP_CONTENTS,0);
end;

procedure TScope.Discreat1Click(Sender: TObject);
begin
Discreat1.Checked:=True;
Continues1.Checked:=False;
if AnalysisSpectrum1.Checked then
DisplaySpectrum
else
DisplaySignal;
end;

procedure TScope.Continues1Click(Sender: TObject);
begin
S3.Visible:=True;
S4.Visible:=True;
AnalysisSpectrum1.Checked:=True;
SpectrumAnalysis.Visible:=True;

SpectrumAnalysis.Height:=227;
SpectrumAnalysis.Width:=178;
SpectrumAnalysis.Top:=5;
SpectrumAnalysis.Left:=542;
DisplaySpectrum;
end
else
begin
AnalysisSpectrum1.Checked:=False;
SpectrumAnalysis.Visible:=False;

P90.Visible:=True;
P10.Visible:=True;
TimeBase.Visible:=True;
Ch1Panel.Enabled:=True;
Ch2Panel.Enabled:=True;
Image2.Visible:=True;
Mark1.Visible:=True;
Mark2.Visible:=False;
Center.Visible:=False;
PHz.Visible:=False;
S1.Visible:=False;
S2.Visible:=False;
S3.Visible:=False;
S4.Visible:=False;

TimeBase.Height:=227;
TimeBase.Width:=178;
TimeBase.Top:=5;
TimeBase.Left:=542;
DisplaySignal;
end;
end;

end;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Discreat1.Checked:=False;
Continues1.Checked:=True;
if AnalysisSpectrum1.Checked then
  DisplaySpectrum;
else
  DisplaySignal;
end;

procedure TScope.Import1Click(Sender: TObject);
begin
  if OpenPictureDialog1.Execute Then
    PlotPicture.LoadFromFile(OpenPictureDialog1.FileName);
end;

procedure TScope.Edit2Change(Sender: TObject);
begin
  //GainCtl:=StrToInt(Edit2.Text);
  //Label37.Caption:=IntToStr(GainCtl);
end;

procedure TScope.Edit3Change(Sender: TObject);
begin
  //AttCtl:=StrToInt(Edit3.Text);
  //Label37.Caption:=IntToStr(AttCtl);
end;

procedure TScope.Exit1Click(Sender: TObject);
begin
  if MessageDlg('Exiting the Computer Oscilloscope & Spectrum
Analysis', mtInformation,
  [mbYes, mbNo], 0) = mrYes then
    Close;
end;

procedure TScope.Ch1VoltChange(Sender: TObject);
begin
  Channel1.Chang;
  //StatusBar1.Panels.Items[0].Text:=Ch1Volt.Hint+' '*LV1.Caption;
end;

procedure TScope.Ch2VoltChange(Sender: TObject);
begin
  Channel2.Chang;
  //StatusBar1.Panels.Items[0].Text:=Ch2.Hint+' '*LV2.Caption;
end;

procedure TScope.FormClose(Sender: TObject; var Action:
TCloseAction);
var IniFile:TIniFile;
begin
  IniFile:=TIniFile.Create('PCScope.ini');
  with IniFile do
    begin
      //Save Configuration to INI File Channel 1
      WriteInteger('Channel 1','Volt/Div',Ch1Volt.Value);
      WriteInteger('Channel 1','Color',Color1);
      WriteInteger('Channel 1','YPos',Y1Pos.Value);
      if AC1.Down then WriteString('Channel 1','Mode','AC');
      if DC1.Down then WriteString('Channel 1','Mode','DC');
      if GND1.Down then WriteString('Channel 1','Mode','GN');
      if Inv1.Down then
        begin
          WriteBool('Channel 1','Invert',True);
        end
      else
        begin
          WriteBool('Channel 1','Invert',False);
        end;
      //Save Configuration to INI File Channel 2
      WriteInteger('Channel 2','Volt/Div',Ch2Volt.Value);
      WriteInteger('Channel 2','Color',Color2);
      WriteInteger('Channel 2','YPos',Y2Pos.Value);
      if AC2.Down then WriteString('Channel 2','Mode','AC');
      if DC2.Down then WriteString('Channel 2','Mode','DC');
      if GND2.Down then WriteString('Channel 2','Mode','GN');
      if Inv2.Down then
        begin
          WriteBool('Channel 2','Invert',True);
        end
      else
        begin
          WriteBool('Channel 2','Invert',False);
        end;
      //Save Configuration to INI File Time
      WriteInteger('TimeBase','Time/Div',Time.Value);
      WriteInteger('TimeBase','Level',LevelV.Value);
      WriteInteger('TimeBase','XPos',XPosT.Value);
      //Save Configuration to INI File Setup
      WriteInteger('Setup','MonitorColor',MonitorColor);
      WriteString('Setup','MonitorFile',MonitorName);
      if EnglishFont1.Checked then
        begin
          WriteString('Setup','Font','English');
        end
      else
        begin
          WriteString('Setup','Font','Thai');
        end;
      if Discreat1.Checked then
        begin
          IniFile.WriteString('Setup','Screen','Discreat');
        end
      else
        begin
          IniFile.WriteString('Setup','Screen','Continues');
        end;
      if On1.Checked then
        begin
          IniFile.WriteBool('Setup','TrigOn',True);
        end
      else
        begin
          IniFile.WriteBool('Setup','TrigOn',False);
        end;
    end;
end;

procedure TScope.Channel12Click(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  if ColorDialog1.Execute then
    begin
      Color1:=ColorDialog1.Color;
      Shape1.Brush.Color:=Color1;
    end;
  end;

procedure TScope.Channel22Click(Sender: TObject);
begin
  if ColorDialog1.Execute then
    begin
      Color2:=ColorDialog1.Color;
      Shape2.Brush.Color:=Color2;
    end;
  end;

procedure TScope.SetGrid2Click(Sender: TObject);
begin
  Store.Down:=True;
  Store.Click;
  if ColorDialog1.Execute then
    begin
      Plot.Picture.LoadFromFile(MonitorName);
      MonitorColor:=ColorDialog1.Color;
      PaintMonitor;
    end;
  Real.Down:=True;
  Real.Click;
  end;

procedure TScope.TimeChange(Sender: TObject);
begin
  Timebasechang;
  //StatusBar1.Panels.Items[0].Text:=Time Hint+' '+LT.Caption;
  end;

procedure TScope.On1Click(Sender: TObject);
begin
  if On1.Checked then
    begin
      On1.Checked:=False;
    end
  else
    begin
      On1.Checked:=True;
    end;
  end;

procedure TScope.WindowsChange(Sender: TObject);
begin
  if Windows.Text='Rectangle' then
    FFT1.WeightingWindow:=fwRectangle;
  if Windows.Text='Triangle' then FFT1.WeightingWindow:=fwTriangle;
  if Windows.Text='Blackman' then
    FFT1.WeightingWindow:=fwBlackman;
  if Windows.Text='Cos2' then FFT1.WeightingWindow:=fwCos2;
  if Windows.Text='Gaussian' then FFT1.WeightingWindow:=fwGauss;
  if Windows.Text='Hamming' then
    FFT1.WeightingWindow:=fwHamming;
  DisplaySpectrum;
end;

end;

procedure TScope.SpectrumTypeClick(Sender: TObject);
begin
  DisplaySpectrum;
end;

procedure TScope.PrintSetup1Click(Sender: TObject);
begin
  PrinterSetupDialog1.Execute;
end;

procedure TScope.Print1Click(Sender: TObject);
var
  AspectRatio: Single;
  OutputWidth, OutputHeight: Single;
begin
  ChkRam.Enabled:=False;
  Store.Down:=True;
  if AnalysisSpectrum1.Checked then
    begin
      DisplayForPrintSpectrum;
    end
  else
    begin
      DisplayForPrintSignal;
    end;
  PrintForm.Show;
  if not PrintDialog1.Execute then Exit;
  Printer.BeginDoc;
  try
    OutputWidth := PrintForm.BWMon.Picture.Width;
    OutputHeight := PrintForm.BWMon.Picture.Height;
    AspectRatio := OutputWidth / OutputHeight;
    if (OutputWidth < Printer.PageWidth) and
      (OutputHeight < Printer.PageHeight) then
      begin
        if OutputWidth < OutputHeight then
          begin
            OutputHeight := Printer.PageHeight;
            OutputWidth := OutputHeight * AspectRatio;
          end
        else
          begin
            OutputWidth := Printer.PageWidth;
            OutputHeight := OutputWidth / AspectRatio;
          end
        end;
    if OutputWidth > Printer.PageWidth then
      begin
        OutputWidth := Printer.PageWidth;
        OutputHeight := OutputWidth / AspectRatio;
      end;
    if OutputHeight > Printer.PageHeight then
      begin
        OutputHeight := Printer.PageHeight;
        OutputWidth := OutputHeight * AspectRatio;
      end;
  except
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Printer.Canvas.StretchDraw(Rect(0,0,
Trunc(OutputWidth), Trunc(OutputHeight)),
PrinForm.BWMon.Picture.Graphic);
finally
Printer.EndDoc;
end;
end;

procedure TScope.AC1MouseMove(Sender: TObject; Shift:
TShiftState; X,
Y: Integer);
begin
//StatusBar1.Panels.Items[0].Text:=AC1.Hint;

end;

procedure TScope.DC1MouseMove(Sender: TObject; Shift:
TShiftState; X,
Y: Integer);
begin
//StatusBar1.Panels.Items[0].Text:=DC1.Hint;

end;

procedure TScope.GND1MouseMove(Sender: TObject; Shift:
TShiftState; X,
Y: Integer);
begin
//StatusBar1.Panels.Items[0].Text:=GND1.Hint;

end;

procedure TScope.AC2Click(Sender: TObject);
begin
//StatusBar1.Panels.Items[0].Text:=Ac2.Hint;

end;

procedure TScope.DC2Click(Sender: TObject);
begin
//StatusBar1.Panels.Items[0].Text:=DC2.Hint;

end;

procedure TScope.GND2Click(Sender: TObject);
begin
//StatusBar1.Panels.Items[0].Text:=Gnd2.Hint;

end;

end;

procedure TScope.About1Click(Sender: TObject);
begin
SplashScreen.BorderStyle:=bsDialog;
SplashScreen.Height:=330;
SplashScreen.ShowModal;

end;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET



TDA8703 8-bit high-speed analog-to-digital converter

Product specification
Supersedes data of April 1993
File under Integrated Circuits, IC02

1996 Aug 26

8-bit high-speed analog-to-digital converter

TDA8703

FEATURES

- 8-bit resolution
- Sampling rate up to 40 MHz
- High signal-to-noise ratio over a large analog input frequency range (7.1 effective bits at 4.43 MHz full-scale input)
- Binary or two's complement 3-state TTL outputs
- Overflow/underflow 3-state TTL output
- TTL compatible digital inputs
- Low-level AC clock input signal allowed
- Internal reference voltage generator
- Power dissipation only 290 mW (typical)
- Low analog input capacitance, no buffer amplifier required
- No sample-and-hold circuit required.

APPLICATIONS

- General purpose high-speed analog-to-digital conversion
- Digital TV, IDTV
- Subscriber TV decoder
- Satellite TV decoders
- Digital VCR.

GENERAL DESCRIPTION

The TDA8703 is an 8-bit high-speed Analog-to-Digital Converter (ADC) for video and other applications. It converts the analog input signal into 8-bit binary-coded digital words at a maximum sampling rate of 40 MHz. All digital inputs and outputs are TTL compatible, although a low-level AC clock input signal is allowed.

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
TDA8703	DIP24	plastic dual in-line package; 24 leads (600 mil)	SOT101-1
TDA8703T	SO24	plastic small outline package; 24 leads; body width 7.5 mm	SOT137-1

8-bit high-speed analog-to-digital converter

TDA8703

QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{CCA}	analog supply voltage		4.5	5.0	5.5	V
V_{CCD}	digital supply voltage		4.5	5.0	5.5	V
V_{CCO}	output stages supply voltage		4.2	5.0	5.5	V
I_{CCA}	analog supply current		–	28	36	mA
I_{CCD}	digital supply current		–	19	25	mA
I_{CCO}	output stages supply current		–	11	14	mA
ILE	DC integral linearity error		–	–	± 1	LSB
DLE	DC differential linearity error		–	–	$\pm 1/2$	LSB
AILE	AC integral linearity error	note 1	–	–	± 2	LSB
B	–3 dB bandwidth	note 2; $f_{CLK} = 40$ MHz	–	19.5	–	MHz
f_{CLK}/f_{CLK}	maximum conversion rate	note 3	40	–	–	MHz
P_{tot}	total power dissipation		–	290	415	mW

Notes

- Full-scale sine wave ($f_i = 4.4$ MHz; f_{CLK} ; $f_{CLK} = 27$ MHz).
- The –3 dB bandwidth is determined by the 3 dB reduction in the reconstructed output (full-scale signal at input).
- The circuit has two clock inputs CLK and \overline{CLK} . There are four modes of operation:
 - TTL (mode 1); \overline{CLK} decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
 - TTL (mode 2); CLK decoupled to DGND by a capacitor. \overline{CLK} input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
 - AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the \overline{CLK} input with such a signal, sampling takes place on the HIGH-to-LOW transition.
 - If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.

8-bit high-speed analog-to-digital converter

TDA8703

BLOCK DIAGRAM

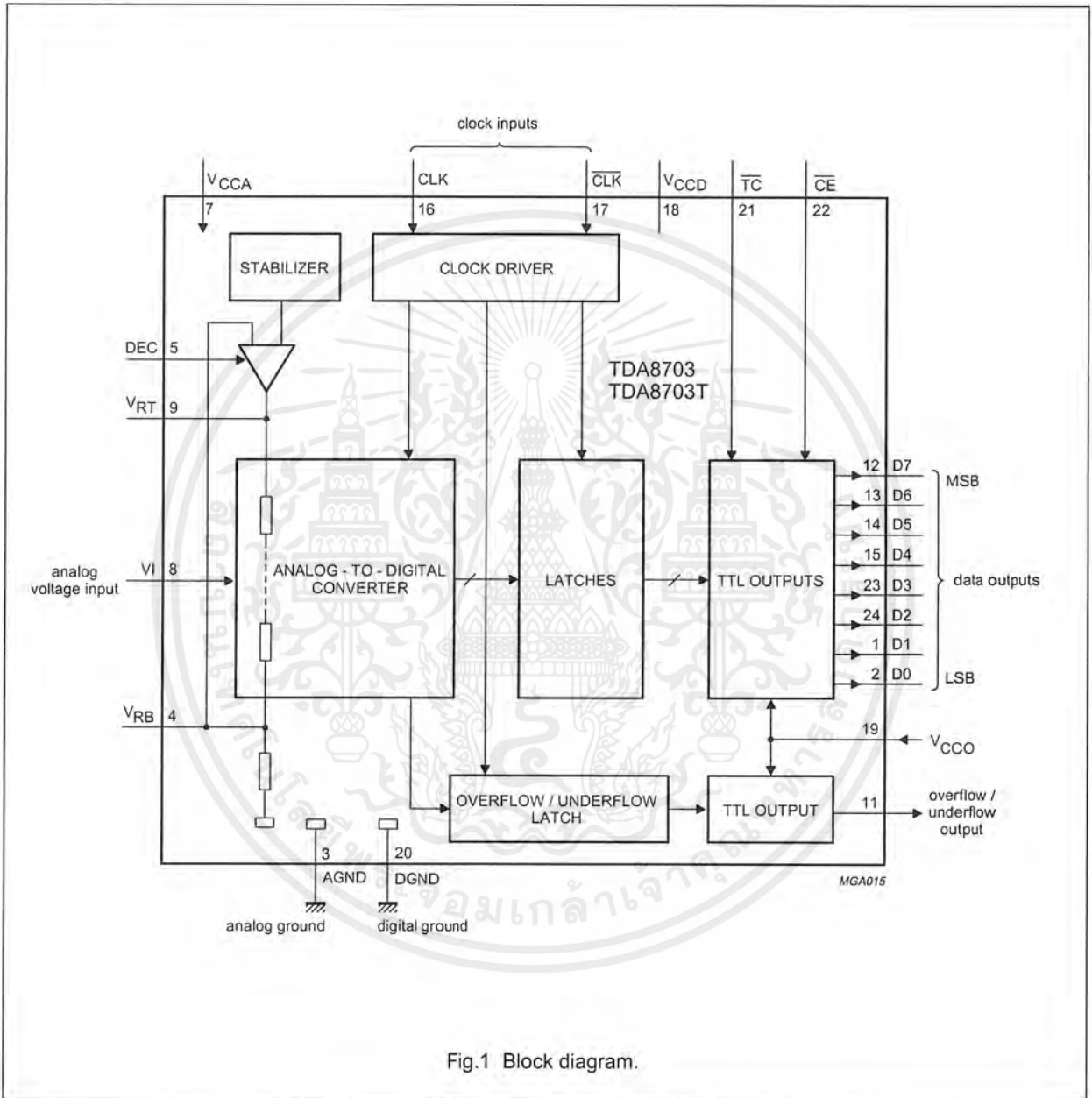


Fig.1 Block diagram.

8-bit high-speed analog-to-digital converter

TDA8703

PINNING

SYMBOL	PIN	DESCRIPTION
D1	1	data output; bit 1
D0	2	data output; bit 0 (LSB)
AGND	3	analog ground
V _{RB}	4	reference voltage bottom (decoupling)
DEC	5	decoupling input (internal stabilization loop decoupling)
n.c.	6	not connected
V _{CCA}	7	positive supply voltage for analog circuits (+5 V)
V _I	8	analog voltage input
V _{RT}	9	reference voltage top (decoupling)
n.c.	10	not connected
O/UF	11	overflow/underflow data output
D7	12	data output; bit 7 (MSB)
D6	13	data output; bit 6
D5	14	data output; bit 5
D4	15	data output; bit 4
CLK	16	clock input
CLK	17	complementary clock input
V _{CCD}	18	positive supply voltage for digital circuits (+5 V)
V _{CCO}	19	positive supply voltage for output stages (+5 V)
DGND	20	digital ground
TC	21	input for two's complement output (TTL level input, active LOW)
CE	22	chip enable input (TTL level input, active LOW)
D3	23	data output; bit 3
D2	24	data output; bit 2

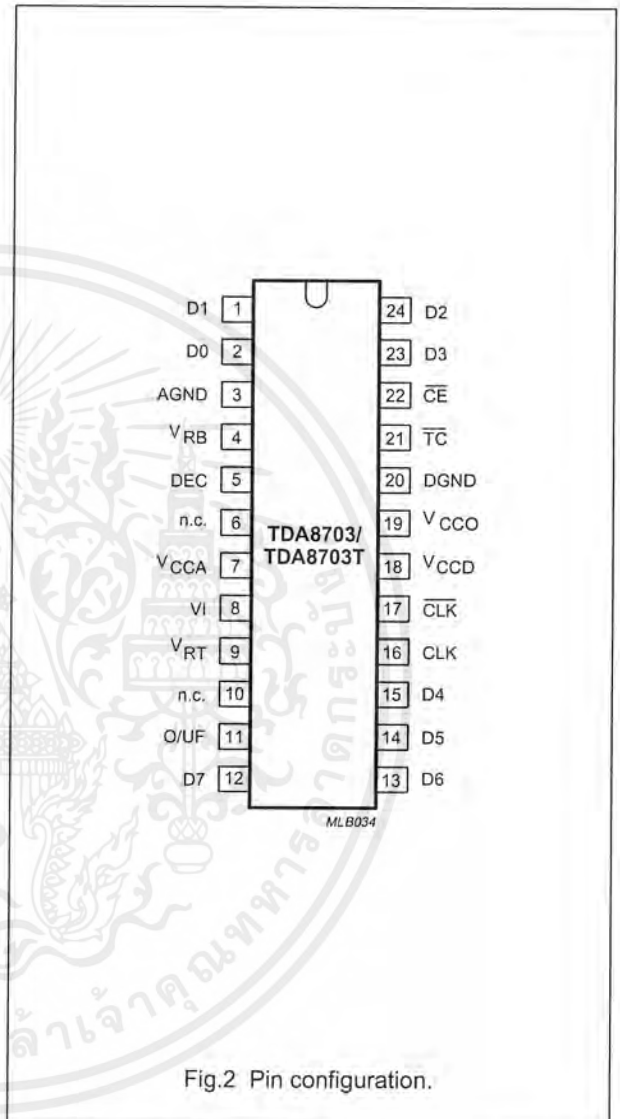


Fig.2 Pin configuration.

8-bit high-speed analog-to-digital converter

TDA8703

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CCA}	analog supply voltage		-0.3	+7.0	V
V_{CCD}	digital supply voltage		-0.3	+7.0	V
V_{CCO}	output stages supply voltage		-0.3	+7.0	V
$V_{CCA} - V_{CCD}$	supply voltage differences		-1.0	+1.0	V
$V_{CCO} - V_{CCD}$	supply voltage differences		-1.0	+1.0	V
$V_{CCA} - V_{CCO}$	supply voltage differences		-1.0	+1.0	V
V_{VI}	input voltage range	referenced to AGND	-0.3	+7.0	V
$V_{CLK}/V_{\overline{CLK}}$	AC input voltage for switching (peak-to-peak value)	note 1; referenced to DGND	-	2.0	V
I_O	output current		-	+10	mA
T_{stg}	storage temperature		-55	+150	°C
T_{amb}	operating ambient temperature		0	+70	°C
T_j	junction temperature		-	+125	°C

Notes

- The circuit has two clock inputs CLK and \overline{CLK} . There are four modes of operation:
 - TTL (mode 1); \overline{CLK} decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
 - TTL (mode 2); CLK decoupled to DGND by a capacitor. \overline{CLK} input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
 - AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the \overline{CLK} input with such a signal, sampling takes place on the HIGH-to-LOW transition.
 - If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.

HANDLING

Inputs and outputs are protected against electrostatic discharges in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling integrated circuits.

THERMAL RESISTANCE

SYMBOL	PARAMETER	VALUE	UNIT
$R_{th\ j-a}$	from junction to ambient in free air		
	SOT101-1	55	K/W
	SOT137-1	75	K/W

8-bit high-speed analog-to-digital converter

TDA8703

CHARACTERISTICS

$V_{CCA} = V_7 - V_3 = 4.5 \text{ V to } 5.5 \text{ V}$; $V_{CCD} = V_{18} - V_{20} = 4.5 \text{ V to } 5.5 \text{ V}$; $V_{CCO} = V_{19} - V_{20} = 4.5 \text{ V to } 5.5 \text{ V}$; AGND and DGND shorted together; $V_{CCA} - V_{CCD} = -0.5 \text{ V to } +0.5 \text{ V}$; $V_{CCO} - V_{CCD} = -0.5 \text{ V to } +0.5 \text{ V}$;
 $V_{CCA} - V_{CCD} = -0.5 \text{ V to } +0.5 \text{ V}$; $T_{amb} = 0 \text{ }^\circ\text{C to } +70 \text{ }^\circ\text{C}$; unless otherwise specified (typical values measured at $V_{CCA} = V_{CCD} = V_{CCO} = 5 \text{ V}$ and $T_{amb} = 25 \text{ }^\circ\text{C}$).

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{CCA}	analog supply voltage		4.5	5.0	5.5	V
V_{CCD}	digital supply voltage		4.5	5.0	5.5	V
V_{CCO}	output stages supply voltage		4.2	5.0	5.5	V
I_{CCA}	analog supply current		–	28	36	mA
I_{CCD}	digital supply current		–	19	25	mA
I_{CCO}	output stage supply current	all outputs LOW	–	11	14	mA
Inputs						
CLOCK INPUT $\overline{\text{CLK}}$ AND CLK (note 1; REFERENCED TO DGND)						
V_{iL}	LOW level input voltage		0	–	0.8	V
V_{iH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{iL}	LOW level input current	$V_{\overline{\text{CLK}}}/V_{\overline{\text{CLK}}} = 0.4 \text{ V}$	–400	–	–	μA
I_{iH}	HIGH level input current	$V_{\overline{\text{CLK}}}/V_{\overline{\text{CLK}}} = 0.4 \text{ V}$	–	–	100	μA
		$V_{\overline{\text{CLK}}}/V_{\overline{\text{CLK}}} = V_{CCD}$	–	–	300	μA
Z_i	input impedance	$f_{\overline{\text{CLK}}}/f_{\overline{\text{CLK}}} = 10 \text{ MHz}$	–	4	–	$\text{k}\Omega$
C_i	input capacitance	$f_{\overline{\text{CLK}}}/f_{\overline{\text{CLK}}} = 10 \text{ MHz}$	–	4.5	–	pF
$V_{\overline{\text{CLK}}} - V_{\overline{\text{CLK}}}$	AC input voltage for switching (peak-to-peak value)	note 1; DC level = 1.5 V	0.5	–	2.0	V
$\overline{\text{TC}}$ AND $\overline{\text{CE}}$ (REFERENCED TO DGND)						
V_{iL}	LOW level input voltage		0	–	0.8	V
V_{iH}	HIGH level input voltage		2.0	–	V_{CCD}	V
I_{iL}	LOW level input current	$V_{iL} = 0.4 \text{ V}$	–400	–	–	μA
I_{iH}	HIGH level input current	$V_{iH} = 2.7 \text{ V}$	–	–	20	μA
VI (ANALOG INPUT VOLTAGE REFERENCED TO AGND)						
$V_{VI(B)}$	input voltage (bottom)		1.33	1.41	1.48	V
$V_{VI(0)}$	input voltage	output code = 0	1.455	1.55	1.635	V
$V_{OS(B)}$	offset voltage (bottom)	$V_{VI(0)} - V_{VI(B)}$	0.125	–	0.155	V
$V_{VI(T)}$	input voltage (top)		3.2	3.36	3.5	V
$V_{VI(255)}$	input voltage	output code = 255	3.115	3.26	3.385	V
$V_{OS(T)}$	offset voltage (top)	$V_{VI(T)} - V_{VI(255)}$	0.085	–	0.115	V
$V_{VI(p-p)}$	input voltage amplitude (peak-to-peak value)		1.66	1.71	1.75	V
I_{iL}	LOW level input current	$V_{VI} = 1.4 \text{ V}$	–	0	–	μA
I_{iH}	HIGH level input current	$V_{VI} = 3.6 \text{ V}$	60	120	180	μA
Z_i	input impedance	$f_i = 1 \text{ MHz}$	–	10	–	$\text{k}\Omega$
C_i	input capacitance	$f_i = 1 \text{ MHz}$	–	14	–	pF

8-bit high-speed analog-to-digital converter

TDA8703

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Reference resistance						
R_{ref}	reference resistance	V_{RT} to V_{RB}	–	220	–	Ω
Outputs						
DIGITAL OUTPUTS (D7 - D0) (REFERENCED TO DGND)						
V_{OL}	LOW level output voltage	$I_O = 1$ mA	0	–	0.4	V
V_{OH}	HIGH level output voltage	$I_O = -0.4$ mA	2.7	–	V_{CCD}	V
I_{OZ}	output current in 3-state mode	0.4 V < V_O < V_{CCD}	–20	–	+20	μ A
Switching characteristics (note 2; see Fig.3)						
f_{CLK}/f_{CLK}	maximum clock frequency		40	–	–	MHz
Analog signal processing ($f_{CLK} = 40$ MHz)						
B	–3 dB bandwidth	note 3	–	19.5	–	MHz
G_d	differential gain	note 4	–	0.6	–	%
ϕ_d	differential phase	note 4	–	0.8	–	deg
f_1	fundamental harmonics (full-scale)	$f_i = 4.43$ MHz	–	–	0	dB
f_{all}	harmonics (full-scale), all components	$f_i = 4.43$ MHz	–	–55	–	dB
SVRR1	supply voltage ripple rejection	note 5	–	–28	–25	dB
SVRR2	supply voltage ripple rejection	note 5	–	1	2.5	%/V
Transfer function						
ILE	DC integral linearity error		–	–	± 1	LSB
DLE	DC differential linearity error		–	–	$\pm 1/2$	LSB
AILE	AC integral linearity error	note 6	–	–	± 2	LSB
EB	effective bits	$f_i = 4.43$ MHz	–	7.1	–	bits
Timing (note 7; see Figs 3 to 6; $f_{CLK} = 40$ MHz)						
t_{dS}	sampling delay		–	–	2	ns
t_{HD}	output hold time		6	–	–	ns
t_{dLH}	output delay time	LOW-to-HIGH transition	–	8	10	ns
t_{dHL}	output delay time	HIGH-to-LOW transition	–	16	20	ns
t_{dZH}	3-state output delay times	enable-to-HIGH	–	19	25	ns
t_{dZL}	3-state output delay times	enable-to-LOW	–	16	20	ns
t_{dHZ}	3-state output delay times	disable-to-HIGH	–	14	20	ns
t_{dLZ}	3-state output delay times	disable-to-LOW	–	9	12	ns

8-bit high-speed analog-to-digital converter

TDA8703

Notes

1. The circuit has two clock inputs CLK and $\overline{\text{CLK}}$. There are four modes of operation:
 - a) TTL (mode 1); $\overline{\text{CLK}}$ decoupled to DGND by a capacitor. CLK input is TTL threshold voltage of 1.5 V and sampling on the LOW-to-HIGH transition of the input clock signal.
 - b) TTL (mode 2); CLK decoupled to DGND by a capacitor. $\overline{\text{CLK}}$ input is TTL threshold voltage of 1.5 V and sampling on the HIGH-to-LOW transition of the input clock signal.
 - c) AC drive modes (modes 3 and 4); When driving the CLK input directly and with any AC signal of 0.5 V (peak-to-peak value) imposed on a DC level of 1.5 V, sampling takes place on the LOW-to-HIGH transition of the clock signal. When driving the $\overline{\text{CLK}}$ input with such a signal, sampling takes place on the HIGH-to-LOW transition.
 - d) If one of the clock inputs is not driven, then it is recommended to decouple this input to DGND with a 100 nF capacitor.
2. In addition to a good layout of the digital and analog ground, it is recommended that the rise and fall times of the clock must not be less than 2 ns.
3. The -3 dB bandwidth is determined by the 3 dB reduction in the reconstructed output (full-scale signal at the input).
4. Low frequency ramp signal ($V_{VI(p-p)} = 1.8$ V and $f_i = 15$ kHz) combined with a sinewave input voltage ($V_{VI(p-p)} = 0.5$ V, $f_i = 4.43$ MHz) at the input.
5. Supply voltage ripple rejection:
 - a) SVRR1; variation of the input voltage producing output code 127 for supply voltage variation of 1 V:

$$\text{SVRR1} = 20 \log (\Delta V_{VI(127)} / \Delta V_{CCA})$$
 - b) SVRR2; relative variation of the full-scale range of analog input for a supply voltage variation of 1 V:

$$\text{SVRR2} = \{ \Delta (V_{VI(0)} - V_{VI(255)}) / (V_{VI(0)} - V_{VI(255)}) \} \div \Delta V_{CCA}$$
6. Full-scale sinewave ($f_i = 4.4$ MHz; $f_{\text{CLK}}; f_{\overline{\text{CLK}}} = 27$ MHz).
7. Output data acquisition:
 - a) Output data is available after the maximum delay of t_{dHL} and t_{dLH} .

8-bit high-speed analog-to-digital converter

TDA8703

Table 1 Output coding and input voltage (referenced to AGND; typical values)

STEP	$V_{VI(p-p)}$	O/UF	BINARY OUTPUT BITS								TWO'S COMPLEMENT OUTPUT BITS							
			D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Underflow	<1.55	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1.55	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	—	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
.
.
254	.	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0
255	3.26	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Overflow	>3.26	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Table 2 Mode selection

\overline{TC}	\overline{CE}	D7-D0	O/UF
X ⁽¹⁾	1	high impedance	high impedance
0	0	active; two's complement	active
1	0	active; binary	active

Note

- 1. X = don't care.

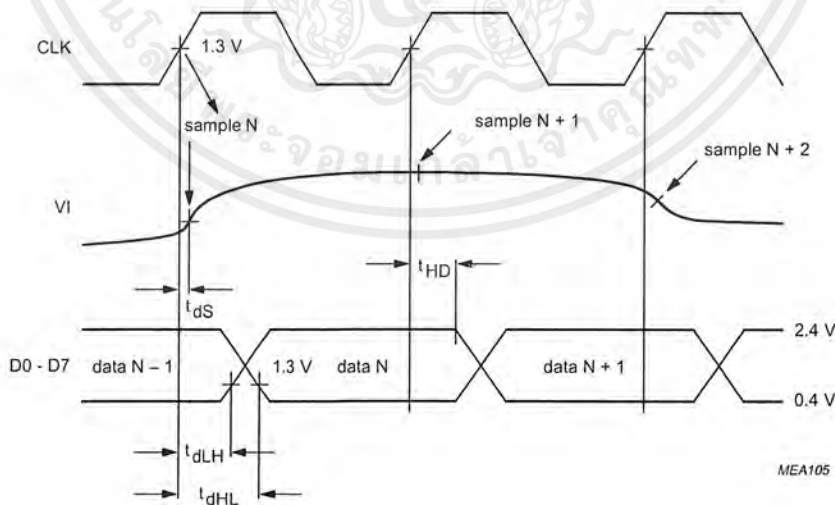


Fig.3 Timing diagram.

8-bit high-speed analog-to-digital converter

TDA8703

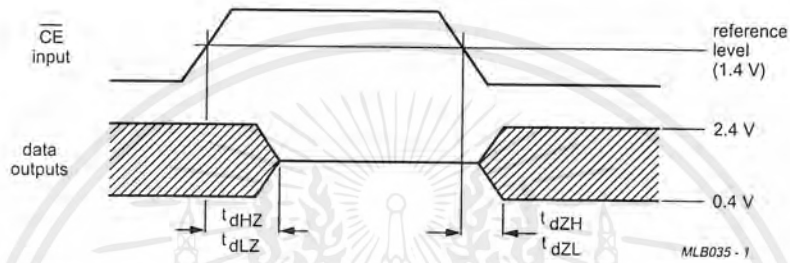


Fig.4 3-state delay timing diagram.

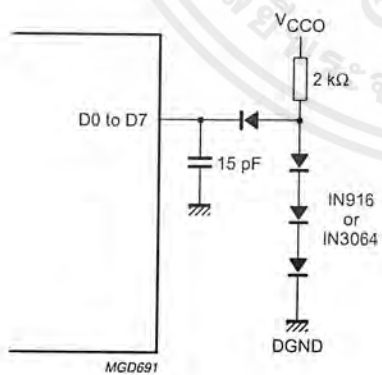


Fig.5 Load circuit for timing measurement; data outputs ($\overline{CE} = \text{LOW}$).

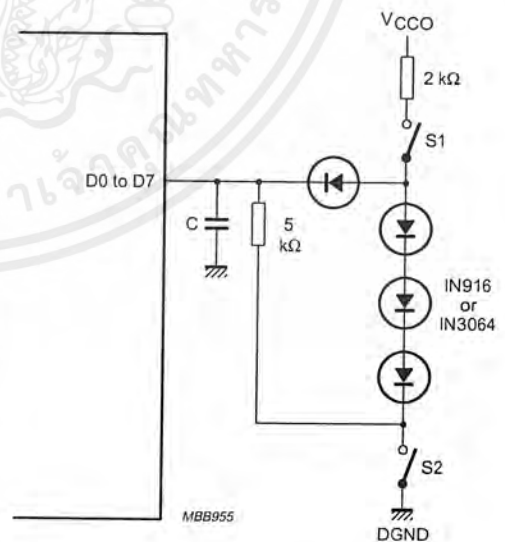


Fig.6 Load circuit for timing measurement; 3-state outputs (\overline{CE} : $f_i = 1 \text{ MHz}$; $V_{VI} = 3 \text{ V}$); see Table 3.

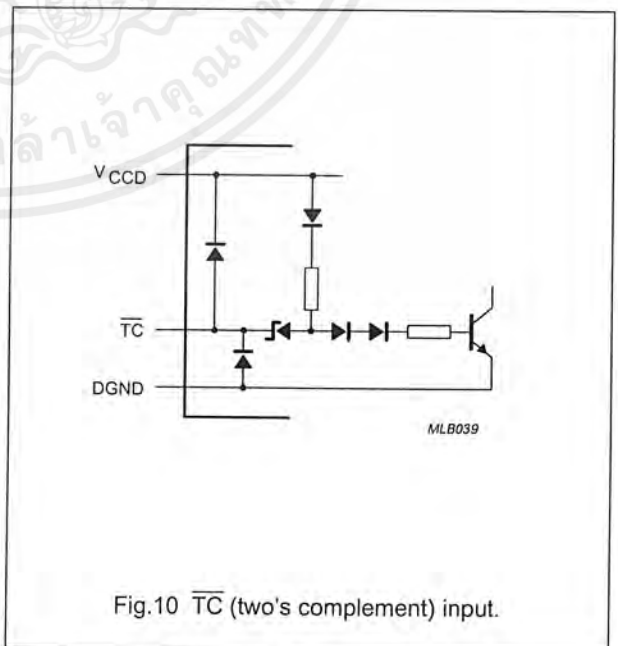
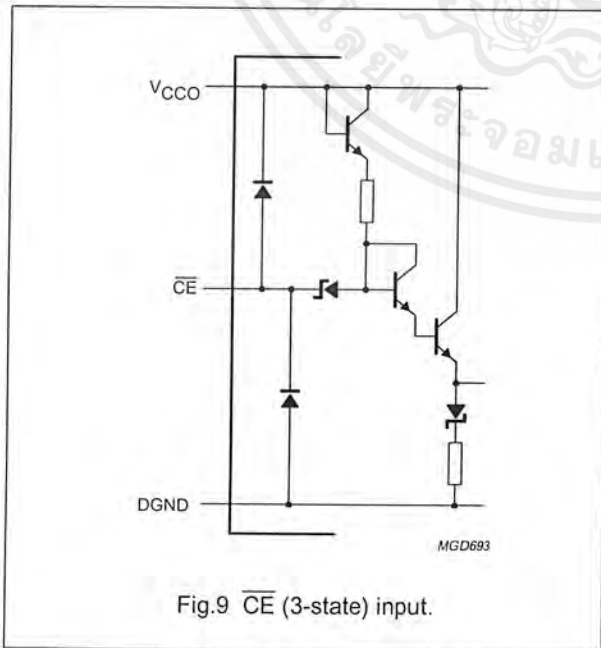
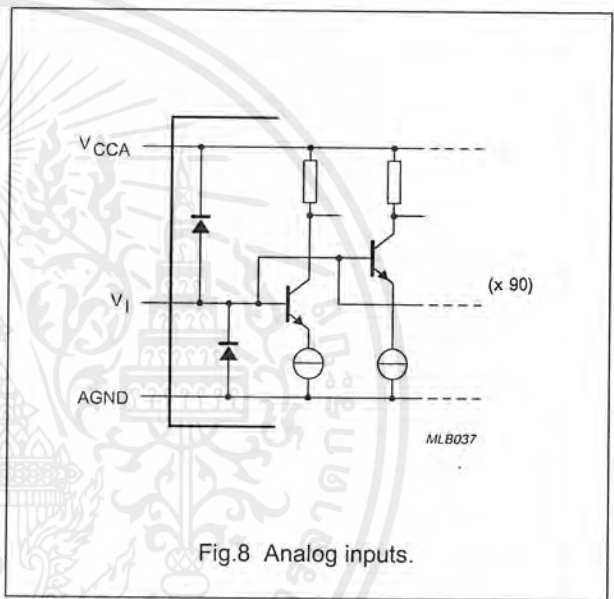
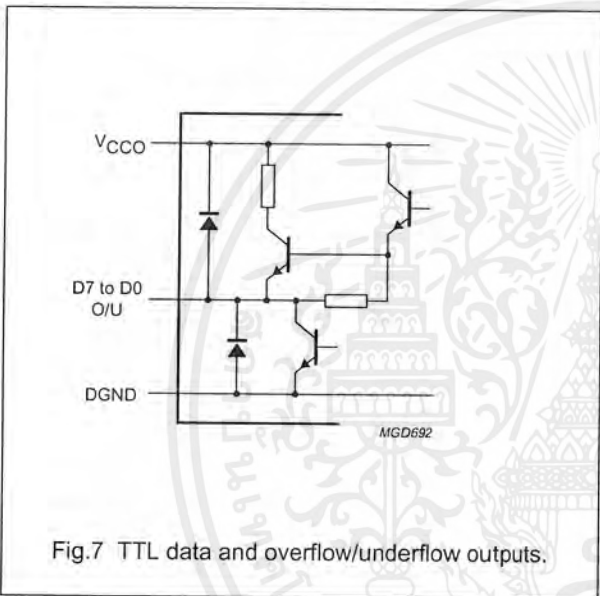
8-bit high-speed analog-to-digital converter

TDA8703

Table 3 Mode selection

TIMING MEASUREMENT	SWITCH S1	SWITCH S2	CAPACITOR
t_{dZH}	open	closed	15 pF
t_{dZL}	closed	open	15 pF
t_{dHZ}	closed	closed	5 pF
t_{dLZ}	closed	closed	5 pF

INTERNAL PIN CONFIGURATIONS



8-bit high-speed analog-to-digital converter

TDA8703

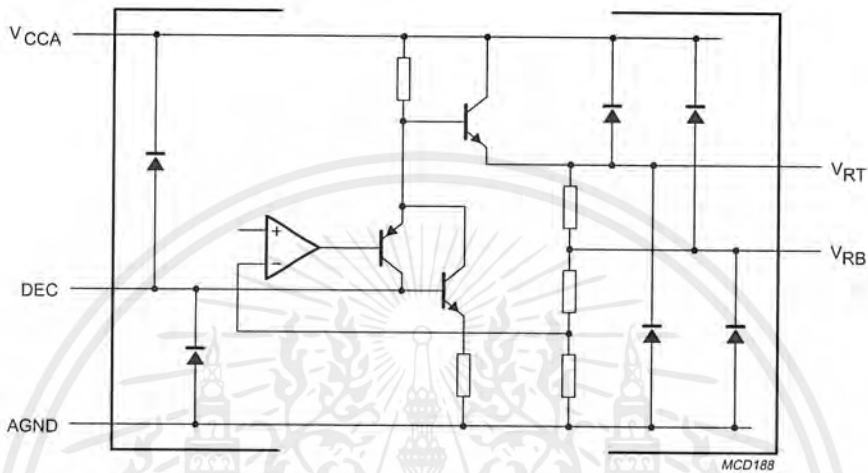


Fig.11 V_{RB}, V_{RT} and DEC.

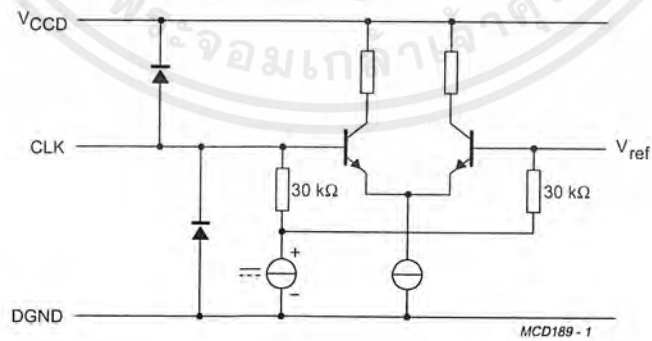


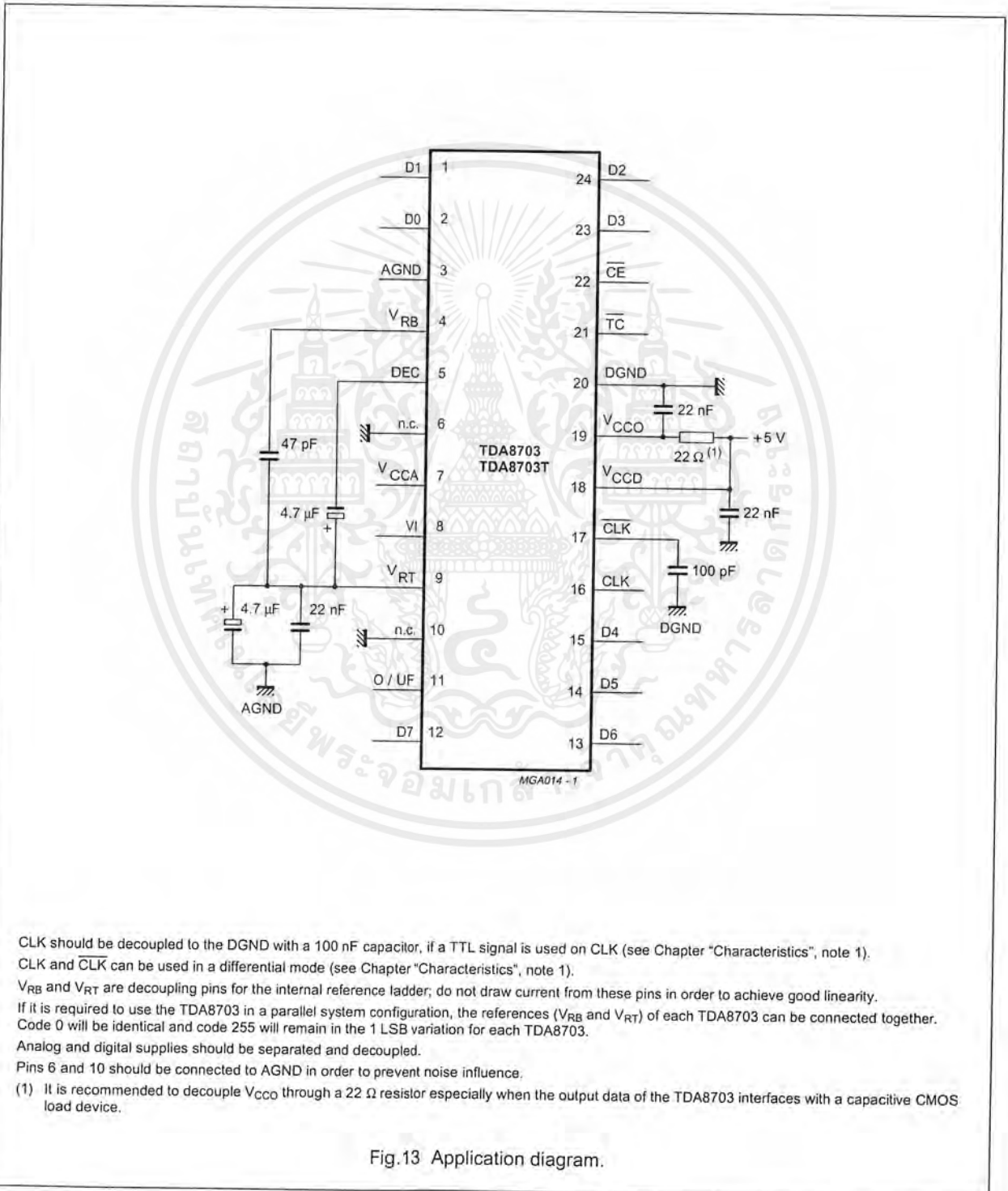
Fig.12 CLK and $\overline{\text{CLK}}$ inputs.

8-bit high-speed analog-to-digital converter

TDA8703

APPLICATION INFORMATION

Additional application information will be supplied upon request (please quote number FTV/8901).



CLK should be decoupled to the DGND with a 100 nF capacitor, if a TTL signal is used on CLK (see Chapter "Characteristics", note 1). CLK and $\overline{\text{CLK}}$ can be used in a differential mode (see Chapter "Characteristics", note 1).

V_{RB} and V_{RT} are decoupling pins for the internal reference ladder; do not draw current from these pins in order to achieve good linearity.

If it is required to use the TDA8703 in a parallel system configuration, the references (V_{RB} and V_{RT}) of each TDA8703 can be connected together. Code 0 will be identical and code 255 will remain in the 1 LSB variation for each TDA8703.

Analog and digital supplies should be separated and decoupled.

Pins 6 and 10 should be connected to AGND in order to prevent noise influence.

(1) It is recommended to decouple V_{CCO} through a 22 Ω resistor especially when the output data of the TDA8703 interfaces with a capacitive CMOS load device.

Fig.13 Application diagram.

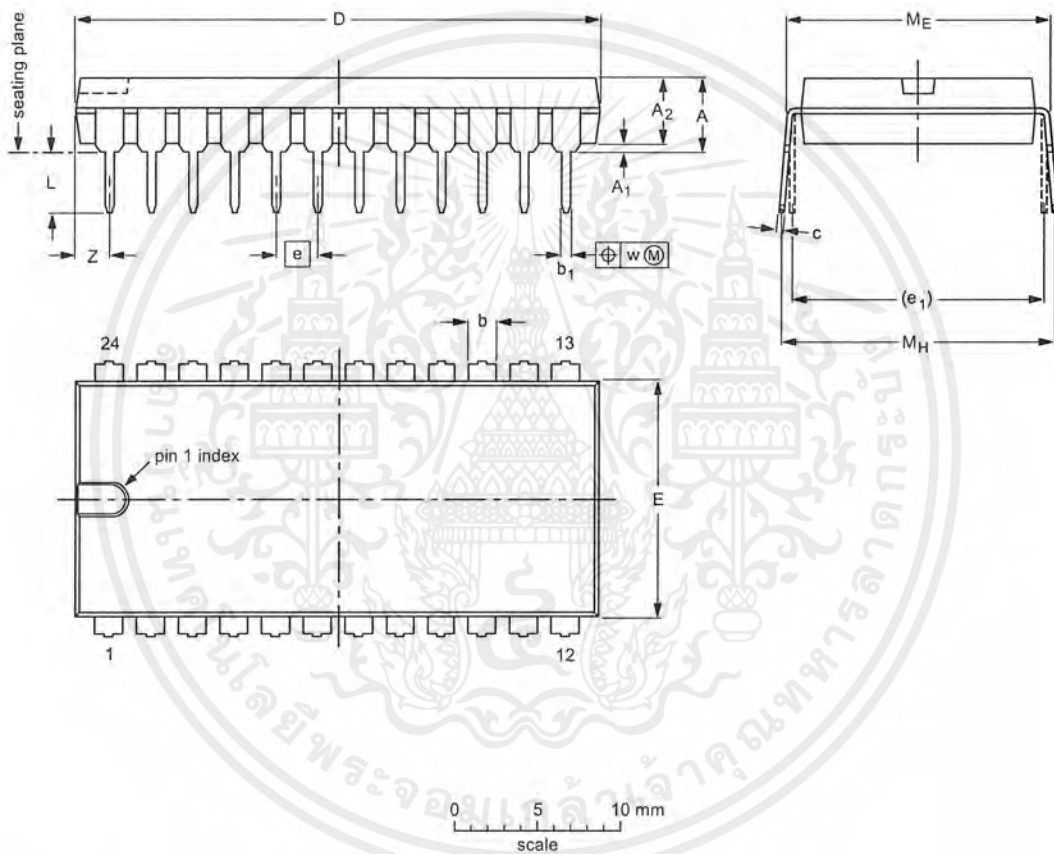
8-bit high-speed analog-to-digital converter

TDA8703

PACKAGE OUTLINES

DIP24: plastic dual in-line package; 24 leads (600 mil)

SOT101-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A ₁ min.	A ₂ max.	b	b ₁	c	D ⁽¹⁾	E ⁽¹⁾	e	e ₁	L	M _E	M _H	w	Z ⁽¹⁾ max.
mm	5.1	0.51	4.0	1.7 1.3	0.53 0.38	0.32 0.23	32.0 31.4	14.1 13.7	2.54	15.24	3.9 3.4	15.80 15.24	17.15 15.90	0.25	2.2
inches	0.20	0.020	0.16	0.066 0.051	0.021 0.015	0.013 0.009	1.26 1.24	0.56 0.54	0.10	0.60	0.15 0.13	0.62 0.60	0.68 0.63	0.01	0.087

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

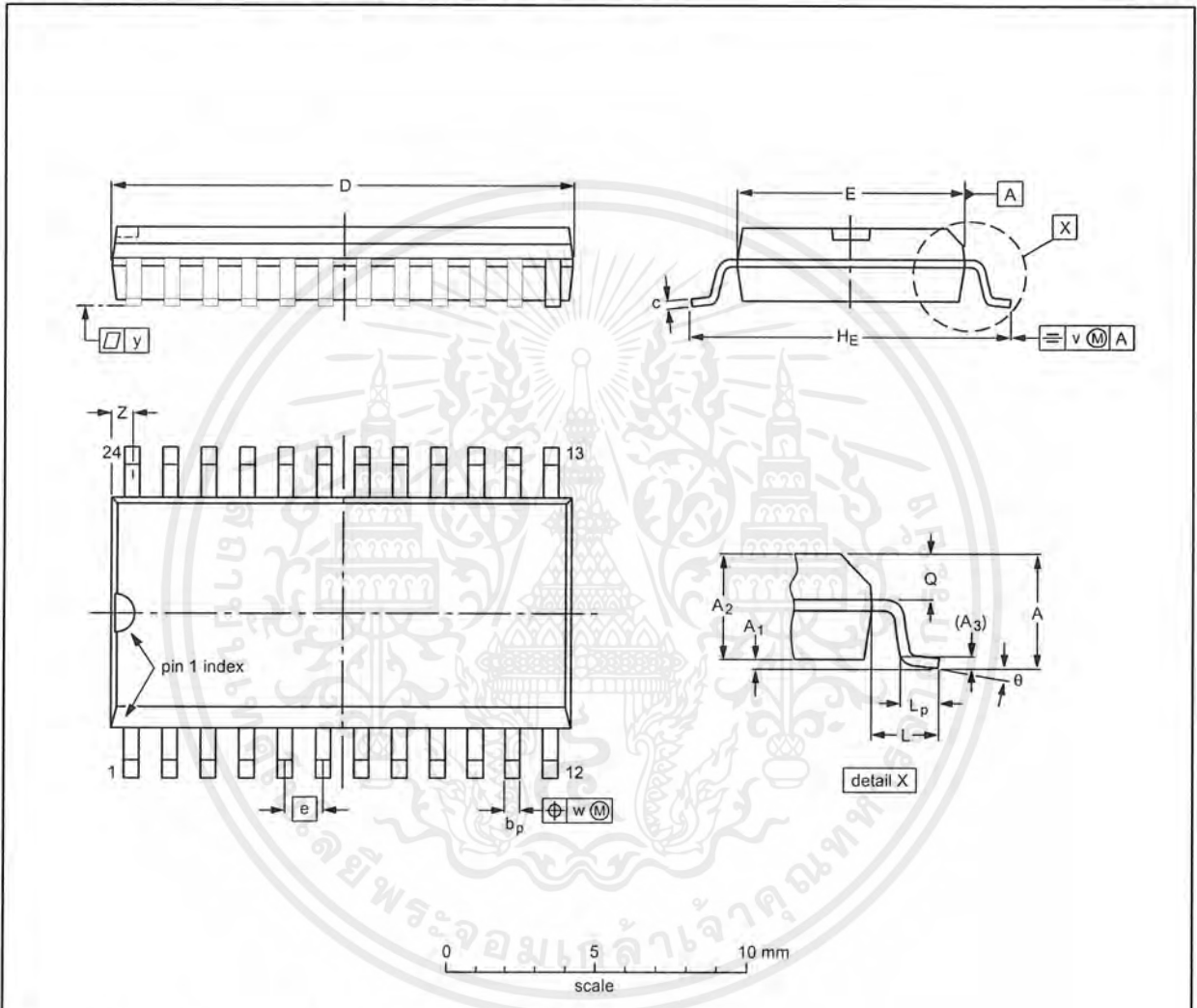
OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT101-1	051G02	MO-015AD			92-11-17 95-01-23

8-bit high-speed analog-to-digital converter

TDA8703

SO24: plastic small outline package; 24 leads; body width 7.5 mm

SOT137-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽¹⁾	e	H _E	L	L _p	Q	v	w	y	Z ⁽¹⁾	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	15.6 15.2	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.61 0.60	0.30 0.29	0.050	0.42 0.39	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT137-1	075E05	MS-013AD			92-11-17 95-01-24

8-bit high-speed analog-to-digital converter

TDA8703

SOLDERING**Introduction**

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our "IC Package Databook" (order code 9398 652 90011).

DIP**SOLDERING BY DIPPING OR BY WAVE**

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg\ max}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

REPAIRING SOLDERED JOINTS

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.

SO**REFLOW SOLDERING**

Reflow soldering techniques are suitable for all SO packages.

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 °C.

WAVE SOLDERING

Wave soldering techniques can be used for all SO packages if the following conditions are observed:

- A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.
- The longitudinal axis of the package footprint must be parallel to the solder flow.
- The package footprint must incorporate solder thieves at the downstream end.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 °C within 6 seconds. Typical dwell time is 4 seconds at 250 °C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

REPAIRING SOLDERED JOINTS

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

8-bit high-speed analog-to-digital converter

TDA8703

DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

เอกสารอ้างอิง

- ก. เอกสารอ้างอิงที่เป็นวารสารภาษาอังกฤษ จัดเรียงตามลำดับดังนี้
John Becker , "PC_SCOPE INTERFACE" Everyday Electronics, October
1991 ,pp 620-625
- ข. เอกสารอ้างอิงที่เป็นวารสารภาษาไทย จัดเรียงตามลำดับดังนี้
เปรมจิตร วิสุทธีศิริ, "พื้นฐานวงจรเอพูดี,ดีทูเอ ตอน 2 วงจรแปลงอนาลอกเป็นดิจิตอล",
เซมิคอนดักเตอร์ อิเล็กทรอนิกส์ ฉบับที่ 103
ชานินทร์ ถาวรศาสนวงศ์ "บทที่ 4 สัญญาณต่างๆบนสล็อตของ IBM/PC",
การอินเทอร์เฟส IBM/PC, มีนาคม 2536
รศ.กฤษฎา วิสุทธิรานนท์ ผศ.ประยูร เขียววัฒนา, "หลักการงานและเทคนิคการใช้งาน
ออสซิลโลสโคป", ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย, 158 หน้า
2521