

เครื่องวิเคราะห์ฮาร์โมนิกส่ของแหล่งจ่ายไฟ

Harmonic Analyzer

โดย

นาย กมล

เขมะรังษี

นางสาว กรรณิการ์

ศรีสันติโรจน์

นาย ชาญเรือง

เหลื่องนิมิตรมาศ



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรสุตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

ปริญญาานิพนธ์ปีการศึกษา 2536

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวิเคราะห์สเปกตรัมของแหล่งจ่ายไฟ(Power Spectrum Analyzer)

ผู้จัดทำ

- | | | |
|--------------------|-----------------|----------|
| 1. นายกมล | เขมะรังษี | 33100003 |
| 2. นางสาวกรรณิการ์ | ศรีสันติโรจน์ | 33100007 |
| 3. นายชาญเรือง | เหลืองนิมิตรมาศ | 33100087 |



.....

ศ.ดร.วัลลภ สุระกำพลธร

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวิเคราะห์ฮาร์โมนิกส์ของแหล่งจ่ายไฟ

POWER SPECTRUM ANALYZER

โดย นายกมล เชมะรังษี
นางสาวกรรณิการ์ ศรีสันติโรจน์
นายชาญเรือง เหลืองนิมิตรมาศ
อาจารย์ที่ปรึกษา
ศ.ดร. วัลลภ สุระคำพลธร

บทคัดย่อ

เครื่องมือวิเคราะห์ฮาร์โมนิกส์ ทำการประมวลผลบนเครื่องไมโครคอมพิวเตอร์โดยใช้หลักการแปลงฟาสต์ฟูริเยอร์ด้วยวิธีการลดทอนทางความถี่เพื่อวิเคราะห์และตรวจสอบสัญญาณของศักดาไฟฟ้าและกระแสจากแหล่งจ่ายไฟได้ถึงฮาร์โมนิกส์ที่ 63 ของความถี่มูลฐาน 50 เฮิรตซ์ และควบคุมการสุ่มตัวอย่าง พร้อมทั้งส่งข้อมูลเข้ายังพอร์ทอนุกรมของไมโครคอมพิวเตอร์โดยใช้ระบบไมโครคอนโทรลเลอร์ 8031

สัญญาณเชิงอุปมานของศักดาไฟฟ้าและกระแส จะถูกสุ่มตัวอย่างด้วยความถี่ 6400 เฮิรตซ์และแปลงเป็นสัญญาณเชิงเลขโดยวงจรรวมของการแปลงสัญญาณเชิงอุปมานเป็นเชิงเลขซึ่งมีขนาด 12 บิต เพื่อให้ได้สัญญาณที่มีความละเอียดสูงแล้วทำการเก็บข้อมูลที่ไค้จากการสุ่มตัวอย่างลงในหน่วยความจำสำรอง ณ เวลาต่าง ๆ และนำข้อมูลส่งผ่านทางพอร์ทอนุกรมให้แก่ไมโครคอมพิวเตอร์ทำการวิเคราะห์โดยใช้โปรแกรมการแปลงฟาสต์ฟูริเยอร์ในการคำนวณหาค่าฮาร์โมนิกส์ค้างของสัญญาณพร้อมทั้งนำผลที่ได้แสดงในรูปกราฟของสเปกตรัม

ABSTRACT

Power Spectrum Analyzer(Harmonic Analyzer) is applied with microcomputer which uses Fast Fourier Transform Calculation by Decimation in Frequency Algorithm. Voltage and current signals are analyzed for the first 63 harmonics of 50 Hz fundamental frequency. Sampling signal and communication via serial port of microcomputer are controlled by microcontroller.

Analog signals of voltage and current will be sampled with 6400 Hz and converted to Digital signal by 12 bits high accuracy analog to digital converter, then stored in RAM at various time, and sent through the serial port to the microcomputer to analyze and calculate harmonics by FFT in spectrum graph.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อ		
บทที่ 1	บทนำ	1
	องค์ประกอบการทำงาน	2
บทที่ 2	ทฤษฎีการแปลงฟาสท์ฟูรีเยอร์	3
	วิธีการหาอัลกอริทึมพื้นฐาน	4
	การคำนวณการแปลงฟูรีเยอร์โดยการลดทอนทางเวลา	5
	การคำนวณการแปลงฟูรีเยอร์โดยการลดทอนทางความถี่	10
	ผลที่เกิดจากการทำ Window สัญญาณ	14
	การรบกวนในระบบไฟฟ้า	19
บทที่ 3	ฮาร์ดแวร์เครื่องวิเคราะห์ฮาร์โมนิกส์	22
	วงจรตัวตรวจจับสัญญาณ	25
	วงจรสุ่มตัวอย่างสัญญาณและคงค่า	26
	วงจรการแปลงสัญญาณ ADC	28
	วงจรกำเนิดสัญญาณความถี่ในการสุ่มตัวอย่าง	32
	วงจรรีจิสเตอร์ระดับแรงดันอ้างอิง	33
	วงจรสร้างสัญญาณโมโนสเตเบิล มัลติไวเบรเตอร์	33
	วงจรหน่วยความจำชั่วคราว	34
	การทำงานของระบบควบคุมการเก็บข้อมูล	34
บทที่ 4	โปรแกรมการทำงาน	38
	อธิบายโปรแกรมการคำนวณ และ แสดงผลฮาร์โมนิกส์	38
	Flowchart	
บทที่ 5	สรุปผลการทดลองและวิเคราะห์	
ภาคผนวก	ก. วงจรของวงจรตรวจจับสัญญาณ	
	ข. วงจรของระบบแหล่งจ่ายไฟ	
	ค. วงจรของบอร์ดเก็บข้อมูล	
บรรณานุกรม		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

สัญญาณอิเล็กทรอนิกส์ที่เราพบเห็นกันโดยทั่วไปมักจะอยู่ในโดเมนเวลา ซึ่งสามารถแสดงลักษณะของสัญญาณได้โดยใช้ข้อซซิลโลสโคป ทำให้ทราบถึงขนาดของสัญญาณที่เวลาต่าง ๆ กันได้ แต่ถ้าเราต้องการทราบถึงขนาดของสัญญาณที่ความถี่แตกต่างกัน คงจะเป็นการไม่สะดวกที่จะสร้างวงจรกรอง (filter) เพื่อกรองเอาความถี่ที่ต้องการทราบขนาดมาทีละความถี่

ในอดีตรวิวิธีการวัดสัญญาณฮาร์โมนิกส์จะต้องใช้โวลต์มิเตอร์และวงจรกรองผ่านแถบความถี่ (Band pass filter BPF) การวัดก็ทำการปรับหาความถี่ที่ต้องการและทำการอ่านค่าความต่างศักย์ที่ได้และค่าที่ได้จะนำมาทำการสร้างกราฟ นอกจากนี้ก็ต้องทำการให้คนคอยติดตามการวัดตลอดเวลาซึ่งผลที่ได้มีข้อจำกัดทางด้านความละเอียดและไม่สามารถวัดค่าเฟสได้ ต่อมาได้เริ่มมีการใช้เครื่องวิเคราะห์สเปกตรัมของสัญญาณ และเครื่องวัดสัญญาณรบกวนเกิดขึ้น แต่วิธีใหม่ก็ยังคงต้องมีเครื่องออสซิลโลสโคป (Oscilloscope) และ เครื่องวัดแบบตัวเลข (Digital Meter) ผลที่ได้จะได้ข้อมูลที่มีความละเอียดมากยิ่งขึ้น แต่กระนั้นก็ตามก็ต้องใช้ค่าใช้จ่ายสูงมาก ดังนั้น เครื่องวิเคราะห์สัญญาณฮาร์โมนิกส์ (Harmonic Analyzer) จึงถูกสร้างเพื่อช่วยให้การวิเคราะห์สัญญาณในโดเมนความถี่ทำได้สะดวกขึ้น โดยมีการใช้เทคนิคของระบบประมวลผลสัญญาณเชิงตัวเลข (Digital signal processing) อาศัยพื้นฐานที่สำคัญของการแปลงฟูริเยร์เต็มหน่วย (Discrete Fourier Transform ,DFT) โดยมีการแปลงฟาสต์ฟูริเยร์ (Fast Fourier Transform ,FFT) เป็นเครื่องมือที่ช่วยให้การคำนวณทำได้รวดเร็วขึ้น ในโครงการนี้เป็นสร้างอุปกรณ์พื้นฐานในการวิเคราะห์สัญญาณดังกล่าวโดยประยุกต์ใช้งานระบบควบคุมไมโครคอนโทรลเลอร์เพื่อทำการควบคุมการทำงาน

การศึกษาการออกแบบเครื่องวิเคราะห์สัญญาณฮาร์โมนิกส์ด้วยระบบเชิงเลข (Digital Harmonic Analyzer) นั้นจะมุ่งเน้นให้เป็นระบบตรวจสอบคุณภาพของแหล่งจ่ายไฟฟ้าโดยทำการวิเคราะห์สัญญาณไฟฟ้าบ้าน หรือสัญญาณจากระบบป้องกันไฟฟ้าขาดช่วง Uninterrupted Power Supply (UPS) ซึ่งเป็นสัญญาณซายน์ (Sinusoidal) ที่มีความถี่มูลฐาน (Fundamental Frequency) 50 Hz โดยจะสามารถคำนวณผลได้ถึงฮาร์โมนิกส์ที่ 63 (The first 63 harmonics of 50 Hz) ของความถี่มูลฐาน ของความต่างศักย์และกระแสไฟฟ้า นอกจากนี้จะให้ทำการแสดงผลการคำนวณหาสัญญาณฮาร์โมนิกส์บนเครื่องไมโครคอมพิวเตอร์โดยมีโปรแกรมแยกต่างหากจากโปรแกรมระบบควบคุม

ระบบของเครื่องวิเคราะห์สัญญาณฮาร์โมนิกส์นี้มีลักษณะคือเป็นระบบรับข้อมูล (Data acquisition) และ มีการประมวลผลแยกต่างหากบนเครื่องไมโครคอมพิวเตอร์

องค์ประกอบการทำงานแบ่งออกเป็น 2 ส่วนหลัก ดังนี้

1. ส่วนสัญญาณเชิงอุปมาน (Analog Section) ซึ่งเป็นส่วนของระบบรับข้อมูล (Data acquisition) ประกอบด้วยอุปกรณ์ที่ใช้ในการรับสัญญาณเชิงอุปมานจากภายนอกผ่านส่วนตรวจจับทางกระแสไฟฟ้าและศักย์ไฟฟ้า (Current and Voltage Sensors) การทำมัลติเพล็กซ์ (Multiplexing) นำสัญญาณมาทำการสุ่มตัวอย่างด้วยวงจรสุ่มและคงค่าสัญญาณ (Sample-and-Hold) ด้วยความถี่ 6400 Hz ซึ่งคิดเป็น 2 เท่าของความถี่ฮาร์โมนิกส์ที่ 63 ของความถี่มูลฐาน 50Hz(Fundamental frequency) แล้วนำไปแปลงเป็นสัญญาณดิจิทัลที่มีขนาด 12 บิต ด้วยตัวแปลงสัญญาณเชิงอุปมานเป็นเชิงเลข (Analog to Digital Converter ,ADC) เพื่อให้ได้ความละเอียดของสัญญาณสูงสุด โดยที่มีฐานเวลาในการแปลงสัญญาณเป็นคริสตัล (Crystal Oscillator)

2. ส่วนสัญญาณเชิงตัวเลข (Digital Section) ประกอบไปด้วยการเก็บข้อมูล,การสื่อสารข้อมูล, การประมวลผลข้อมูล สำหรับการเก็บข้อมูลที่มาจากตัวแปลงสัญญาณ (ADC) จะเก็บไว้ในหน่วยความจำชั่วคราว (Static RAM) โดยที่มีระบบไมโครคอนโทรลเลอร์8031 ทำการควบคุมการเก็บข้อมูลผ่านทางพอร์ทขยาย 8255 และมีการบันทึกข้อมูลของเวลาตามเวลาที่แท้จริง (Real Time Clock) จากนั้นนำข้อมูลรายละเอียดของการทำการสุ่มสัญญาณ นำไปเก็บไว้ในหน่วยความจำสำรองของระบบไมโครคอนโทรลเลอร์ ในส่วนของการสื่อสารข้อมูลเมื่อต้องการนำข้อมูลไปประมวลผลระบบไมโครคอนโทรลเลอร์จะทำการอ่านข้อมูลในหน่วยความจำชั่วคราวส่งโดยการสื่อสารผ่านทางพอร์ทอนุกรมมาตรฐาน RS-232 (Serial Port) เพื่อนำข้อมูลส่งต่อไปยังเครื่องไมโครคอมพิวเตอร์ ระบบไมโครคอนโทรลเลอร์จะต้องมีส่วนของโปรแกรมระบบ ซึ่งทำการควบคุมการทำงานทั้งหมด เมื่อข้อมูลส่งผ่านมาถึงเครื่องไมโครคอมพิวเตอร์จะมีโปรแกรมทำการรับข้อมูลเพื่อนำไปบันทึกเก็บไว้ในฮาร์ดดิสก์และโปรแกรมของการประมวลผลทำการแปลงข้อมูลในโดเมนเวลาเป็นข้อมูลในโดเมนความถี่ด้วยอัลกอริทึมของการแปลงฟาสต์ฟูริเยอร์ FFT และแสดงผลในรูปของกราฟสเปกตรัมทางหน้าจอคอมพิวเตอร์

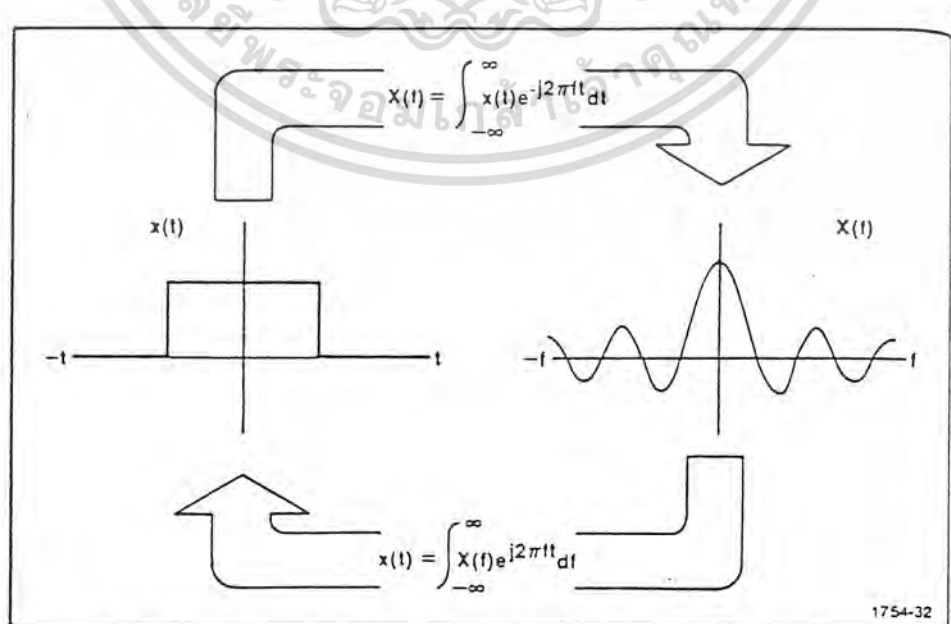


Figure 3-17 The Fourier transform has an inverse.

เอกสารนี้เป็นเอกสารที่สงวนไว้ว่าห้ามมิให้ผู้อื่นใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีการแปลงฟาสต์ฟูรีเยอร์

Fast Fourier Transform FFT

วิทยาศาสตร์และวิศวกรรมศาสตร์ได้มีการพัฒนาขึ้นอย่างมากเนื่องจาก การพัฒนาของความเร็ว ในการคำนวณที่สูงขึ้นและอัลกอริทึมที่มีประสิทธิภาพ ในทางสาขาของการประมวลผลของสัญญาณ (Signal Processing) ก็เช่นกัน ได้เปลี่ยนแปลงไปอย่างมากด้วยวิวัฒนาการของสัญญาณเชิงเลขเมื่อมีการ ได้เริ่มพัฒนาอัลกอริทึมที่ทำให้การคำนวณผลของการแปลงฟูรีเยอร์เต็มหน่วย(Discrete Fourier Transform) มีความรวดเร็วขึ้น

จากหลักการของ การแปลงฟูรีเยอร์เต็มหน่วย (Discrete Fourier Transform) ทำให้เกิดการ ประยุกต์ใช้งานของการแปลงฟูรีเยอร์เต็มหน่วย (DFT)ทางด้านการวิเคราะห์ สเปกตรัม (Spectrum) ของ สัญญาณ[3](ซึ่งทำให้เป็นประโยชน์ในการพัฒนาโครงการนี้) นอกจากนี้ยังมีการทำConvolution และ Correlation และอื่น ๆ อีกมาก ด้วยเหตุผลที่ว่า วิธีการคำนวณโดยตรงซึ่งจะต้องใช้เวลาในการคำนวณ มากและต้องการเครื่องคำนวณที่มีความเร็วสูง ก็ได้ทำให้มีผู้เสนอ อัลกอริทึมที่มีประโยชน์มากในการ ประมวลผลทางด้านสัญญาณ (Signal Processing) โดยทำให้เกิดวิธีการที่สำคัญเพื่อที่จะคำนวณการแปลง ฟูรีเยอร์เต็มหน่วยได้เร็วกว่าการคำนวณธรรมดาซึ่งผู้ที่ริเริ่มคือ James Cooley และ John W. Tukey ในปี ค.ศ.1965 และในปีต่อมา W.M. Gentle Man และ G.Sande ก็ได้เสนอวิธีการคำนวณที่มีประสิทธิ ภาพ และมีลักษณะคล้ายคลึงกับของ Cooley และ Tukey เราเรียกรวมทั้งสองดังกล่าวว่าการแปลงฟาสต์ฟูรีเยอร์ (Fast Fourier Transform ,FFT) จากการคิดค้นดังกล่าว ทำให้สามารถทำการประมวลผลการแปลง ฟูรีเยอร์เต็มหน่วย (DFT) ในระบบเวลาจริง Real Time ได้ในหลาย ๆ งาน และเมื่อ FFTs ถูกใช้ในการ ประมวลผลของ Convolution และ Correlation ก็ถูกเรียกว่า Fast Convolution และ Fast Correlation

จากการประยุกต์ใช้งานการแปลงฟาสต์ฟูรีเยอร์ ในการวิเคราะห์สัญญาณในโดเมนความถี่โดยใช้ คอมพิวเตอร์ช่วยนั้น ถ้าเราใช้การแปลงฟูรีเยอร์มาคำนวณโดยตรง ก็จะเป็นการคำนวณหาค่าอินทิกรัล ซึ่ง จะใช้เวลานาน และไม่แน่ว่าจะมีการลู่เข้าของคำตอบหรือไม่ ดังนั้นการแปลงฟูรีเยอร์เต็มหน่วย (Discrete Fourier Transform)จึงถูกนำมาใช้ซึ่งจะให้ผลการคำนวณเหมือนกันกับการแปลงฟูรีเยอร์สำหรับสัญญาณ ที่เป็นคาบ

ด้วยวิธีของการแปลงฟูรีเยอร์เต็มหน่วย DFT สัญญาณในโดเมนเวลา $x(t)$ จะถูกสุ่มสัญญาณ ด้วยความถี่อย่างน้อย 2 เท่า ของความถี่สูงสุดที่ต้องการทราบค่าสเปกตรัม ค่าความถี่ในการสุ่มตัวอย่าง (ตามทฤษฎีการสุ่มตัวอย่างของ แซนนอน นั้นการสุ่มตัวอย่างสัญญาณได้ถูกต้องก็ต่อเมื่อทราบค่าความถี่ ปฏิบัติงานของสัญญาณ) ที่เป็นสองเท่าของความถี่ตัวอย่าง ค่าความถี่นี้เรียกว่า ความถี่ไนควิสต์ (Nyquist

frequency) และคาบเวลาที่ได้เรียกว่า ช่วงเวลาสุ่มตัวอย่างไนควิสต์ (Nyquist interval) [1] โดยใน 1 คาบของ $x(t)$ จะถูกสุ่มออกเป็น N ครั้ง โดย $x(n)$ คือสัญญาณที่ได้จากการสุ่ม เมื่อนำค่า $x(n)$ ไปทำการคำนวณตามสมการของการแปลงจะได้ $X(k)$ ซึ่งหมายถึง ขนาดของสัญญาณที่ความถี่หนึ่ง ๆ

จากสมการข้อกำหนดของ DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)nk} \quad , k = 0,1,2,\dots,N-1 \quad (1)$$

โดยสมมติให้ข้อมูลที่เข้ามาคือ $x(n)$ อาจมีค่าเป็นจำนวนเชิงซ้อน (COMPLEX Number) จะพบว่าในค่าของ k แต่ละค่าในสมการข้างบนต้องการการคูณจำนวนเชิงซ้อน N ครั้ง โดยจะมีค่า k จำนวน N ค่าที่จะพิจารณา สำหรับการแปลงฟูริเยอร์เต็มหน่วย DFT ต้องการการคูณจำนวนเชิงซ้อน $N*N = N^2$ ครั้ง ดังนั้นจุดมุ่งหมายก็คือการพัฒนาวิธีการคำนวณ DFT ด้วย Fast Fourier Transform FFT

การคำนวณ DFT ของลำดับสัญญาณที่ยาว N ลำดับนั้น คอมพิวเตอร์ จะต้องทำการคูณจำนวนเชิงซ้อน $N*N$ ครั้ง และ บวกจำนวนเชิงซ้อนอีก $N*(N-1)$ ครั้ง โดยปกติมักจะใช้จำนวนครั้งในการคูณที่จะต้องทำในการคำนวณการแปลงฟูริเยอร์เต็มหน่วย เพื่อบอกให้ทราบถึงความซับซ้อนในการคำนวณซึ่งการคูณโดยใช้คอมพิวเตอร์จะใช้เวลาานกว่าการบวกมาก และเพื่อลดจำนวนครั้งในการคูณลงการแปลงฟาสต์ฟูริเยอร์ Fast Fourier Transform จึงถูกนำมาใช้ ซึ่งจะช่วยให้การแปลงเร็วขึ้น

ต่อไปนี้จะขอแสดงการแบ่งและหาวิธี ซึ่ง DFT ที่เราต้องการหาผลลัพธ์จะถูกแบ่งย่อยลงเป็นลำดับชั้นย่อย ๆ ซึ่งเป็นการคำนวณของ DFT ที่มีประสิทธิภาพในการคำนวณมากขึ้น ถึงแม้ว่า FFT จะย่อมาจาก Fast Fourier Transform แต่ตัว FFT เองไม่ใช้การแปลงฟูริเยอร์แท้จริงแล้วเป็นเพียงวิธีการ หรือ ลำดับการ (Algorithm) ในการคำนวณ เพื่อช่วยให้การคำนวณ DFT ซึ่งเป็น การแปลงฟูริเยอร์ทำได้รวดเร็วยิ่งขึ้น

อัลกอริทึมของการแปลงฟาสต์ฟูริเยอร์ FFT ที่ถูกพัฒนาขึ้นมากมาย โดยมีพื้นฐานมาจาก The Cooley-Tukey Algorithm หรือ The Sande-Tukey Algorithm โดยที่ The Cooley-Tukey Algorithm เป็น FFT ชนิดลดทอนทางเวลาหรือ Decimation in time (DIT) ในขณะที่ The Sande-Tukey Algorithm เป็นชนิดลดทอนทางความถี่ หรือ Decimation in frequency (DIF) โดยจะแสดงการคำนวณ FFT โดยอัลกอริทึมของ the Sande-Tukey ไว้ดังนี้

วิธีการหา Algorithm พื้นฐาน

การคำนวณการแปลงฟูริเยอร์เต็มหน่วย DFT เมื่อ $N = 2, 4, 8$ หรือ เป็นจำนวนกำลังของเลข 2 และเพื่อที่จะทำสัญลักษณ์ให้ง่ายขึ้นเราเขียนการแปลงฟูริเยอร์ DFT ในรูปของสมการดังนี้

$$X(k) = \sum_{n=0}^{N-1} x(n) [W_N]^{nk} \quad , k = 0,1,2,\dots,N-1 \quad (2)$$

เมื่อ $W_N = e^{-j(2\pi/N)}$ ผลคูณของ nk เป็นเลขกำลังของ W_N และ $[W_N]^{nk}$ ถูกเรียกว่า weighting factor บางทีเรียกว่า ตัวประกอบหมุน Twiddle factor หรือ Phase หรือ Rotation factor ซึ่งต่อไปจะเขียนเพียงง่าย ๆ คือ W_N^{nk}

การคำนวณการแปลงฟูริเยอร์โดยการลดทอนทางเวลา

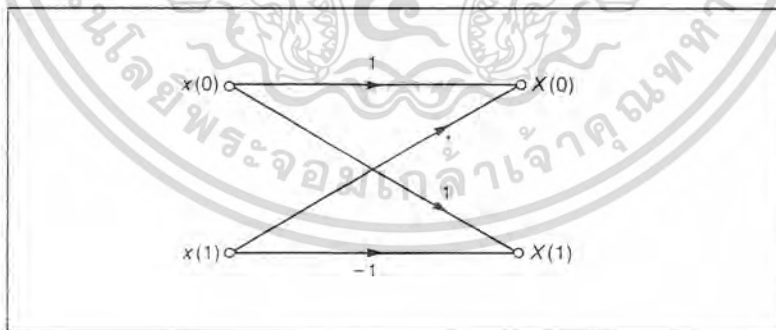
พิจารณาการคำนวณ DFT แบบ 2 จุด โดยขั้นตอนลดทอนทางเวลาจะได้ว่า

$$X(k) = \sum_{n=0}^1 x(n) W_N^{nk} = x(0)W_2^{0k} + x(1)W_2^{1k} \quad , k = 0,1 \quad (3)$$

ซึ่งเมื่อ $W_2 = e^{-j(2\pi/2)} = e^{-j\pi} = -1$ ดังนั้นเมื่อ $n = 0$ และ 1 จะได้ $W_2^{0k} = (-1)^{0k} = 1$ และ $W_2^{1k} = (-1)^k$

$$\begin{aligned} X(0) &= x(0)*1 + x(1)*1 = x(0) + x(1) \\ X(1) &= x(0)*1 + x(1)*(-1) = x(0) - x(1) \end{aligned} \quad (4)$$

สังเกตได้ว่าในกรณีนี้จะไม่มีการคูณเลขเชิงซ้อน และการคำนวณนี้จะสามารถแทนได้ด้วย Signal Flow Graph เมื่อ Input แทนด้วยโหนด $x(0)$ และ $x(1)$ และ Output แทนด้วยโหนดของ $X(0)$ และ $X(1)$ ส่วนเส้นซึ่งลากระหว่างจุดจะมี Gain กำหนดอยู่ที่ตัวคูณ ซึ่งในที่นี้ได้แก่ $+1$ หรือ -1 โดยโครงสร้างดังกล่าวเรียกว่า หน่วยผีเสื้อ (Butterfly) โดยจะแสดงในรูปที่ 2.1



พิจารณาการคำนวณ DFT แบบ 4 จุด โดยขั้นตอนลดทอนทางเวลาจะได้ว่า

$$X(k) = \sum_{n=0}^3 x(n) W_4^{nk} \quad , k = 0,1,2,3 \quad (5)$$

$$= x(0)W_4^{0k} + x(1)W_4^{1k} + x(2)W_4^{2k} + x(3)W_4^{3k}$$

จากเทอมแรกในสมการ (5)

$$W_4^{0k} = e^{-j(2\pi/4)0k} = e^{-j(2\pi/2)0k} = W_2^0 \quad (6)$$

และในเทอมที่สามในสมการ (5)

$$W_4^{2k} = e^{-j(2\pi/4)2k} = e^{-j(2\pi/2)k} = W_2^k \quad (7)$$

ดังนั้นทำให้เรารวมกลุ่มของเทอมแรกและเทอมที่สามในสมการ (5) จะได้

$$x(0)W_4^{0k} + x(2)W_4^{2k} = x_1(0)W_2^{0k} + x_1(1)W_2^k \quad (8)$$

ดังนั้นอาจเขียนได้เป็นลำดับใหม่ได้ว่า

$$x_1(n) = x(2n) \quad ; n = 0,1 \quad (9)$$

ซึ่งบรรทัดดัชนีเลขคู่ของข้อมูลตัวอย่าง ในลำดับแรกสุด(Input) เมื่อเปรียบเทียบสมการ(8),(9)

จะได้
$$X_1(k) = x_1(0)W_2^{0k} + x_1(1)W_2^{1k} \quad (10)$$

ซึ่งก็คือ DFTs แบบ 2 จุด ของลำดับ $x_1(n)$ ในสมการ (7) พิจารณาสองเทอมที่เหลือในการแปลงแบบ 4 จุด จากสมการที่ (5) จะได้

$$x(1)W_4^{1k} + x(3)W_4^{3k} = W_4^{1k} [x(1)W_4^{0k} + x(3)W_4^{2k}] \quad (11)$$

และเมื่อ $W_4^{2k} = W_2^k$ จากสมการ (7)

$$\begin{aligned} x(1)W_4^{1k} + x(3)W_4^{3k} &= W_4^{1k} [x_2(0)W_2^{0k} + x_2(1)W_2^k] \\ &= W_4^{1k} X_2(k) \end{aligned} \quad (12)$$

และก็เช่นเดิมทำให้เราได้ลำดับใหม่ว่า

$$x_2(n) = x(2n+1) \quad ; n = 0,1 \quad (13)$$

เทอม $X_2(k)$ ในสมการที่ (12) ก็คือการแปลงฟูริเยร์เต็มหน่วย DFT แบบ 2 จุด ของ $x_2(n)$ เมื่อรวม

สมการ (10) กับ (12) ทำให้ได้การแปลงฟูริเยร์เต็มหน่วย DFT แบบ 4 จุด ของ $x(n) = x_1(n) + x_2(n)$

คือ

$$X(k) = x_1(0)W_2^{0k} + x_1(1)W_2^k + W_4^{1k} [x_2(0)W_2^{0k} + x_2(1)W_2^k] \quad (14)$$

หรือ

$$X(k) = X_1(k) + W_4^{1k} X_2(k) \quad ; k = 0,1,2,3 \quad (15)$$

หากจะสรุปเป็นคำกล่าวจะได้ว่า

$$\text{DFT 4จุด ของ } x(n) = \text{DFT 2 จุด ของ } x_1(n) + W_4^{1k} [\text{DFT 2จุด ของ } x_2(n)] \quad (16)$$

แม้ว่าเราจะเขียน $X(k)$ ในเทอมของ DFT 2 จุด 2 ชุด ผลที่ได้ก็คือ DFT 4 จุด และจะสังเกตได้ว่าเนื่องจากคุณสมบัติความเป็นคาบ (Periodicity) ของ DFT 2 จุด จะได้

$$X_1(k+2) = X_1(k) \quad \text{และ} \quad X_2(k+2) = X_2(k) \quad ; k = 0, 1 \quad (17)$$

เมื่อเขียน $X(k)$ ในเทอมของการแปลงฟูริเยร์เต็มหน่วย DFT 2 จุดจะได้จากสมการ (15) เป็น

$$X(0) = X_1(0) + W_4^0 X_2(0) = X_1(0) + X_2(0) \quad (18)$$

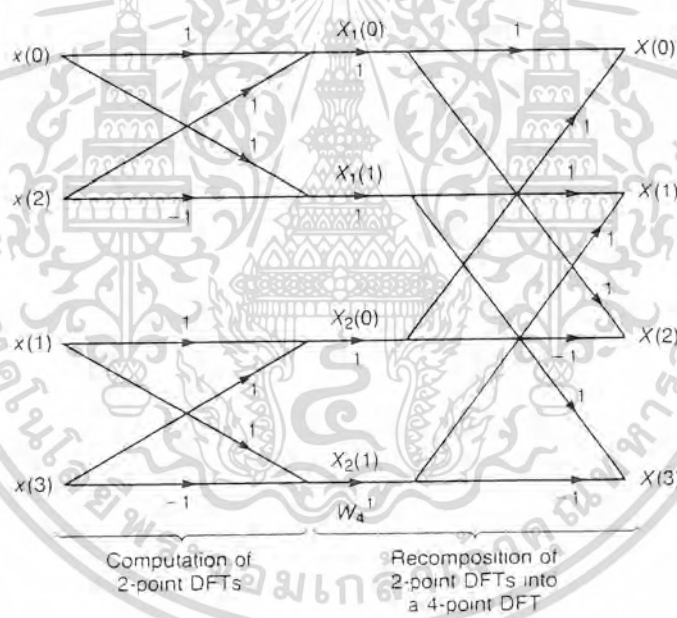
$$X(1) = X_1(1) + W_4^1 X_2(1) \quad (19)$$

$$X(2) = X_1(2) + W_4^2 X_2(2) = X_1(0) - X_2(0) \quad (20)$$

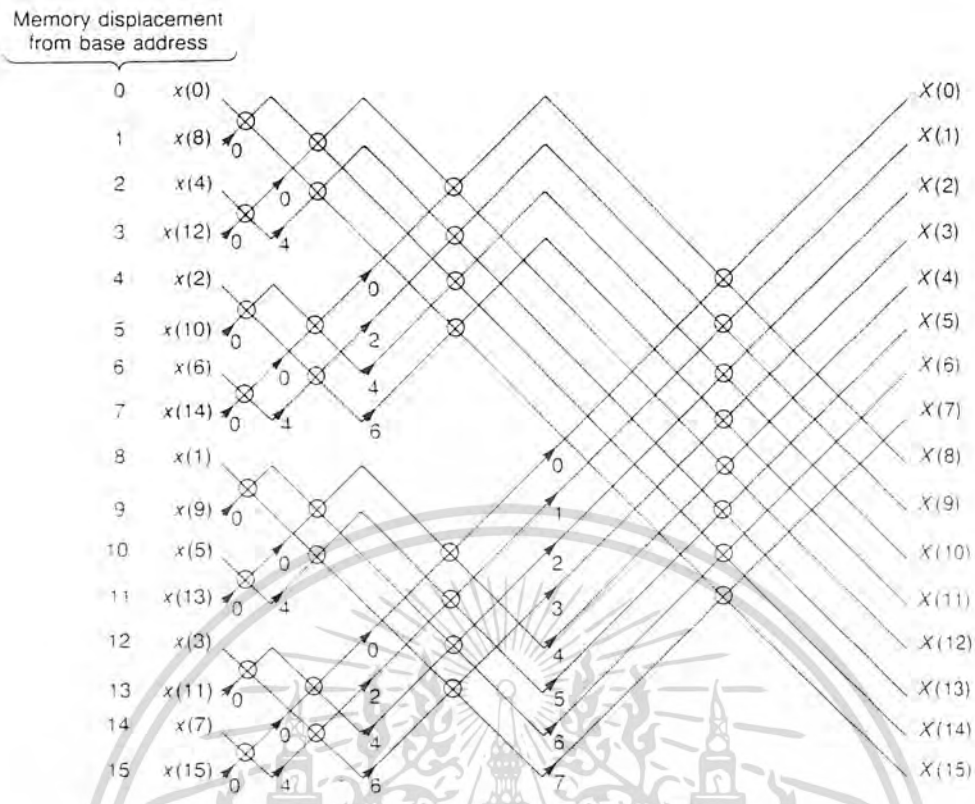
ซึ่งเราได้ใช้คุณสมบัติความเป็นคาบ (Periodicity) ของ $X_1(k)$ และ $X_2(k)$ เพื่ออธิบายโดยใช้สมการที่ (17) และจากความจริงที่ว่า $W_4^2 = -1$ จะได้

$$\begin{aligned} X(3) &= X_1(3) + W_4^3 X_2(3) = X_1(3) + W_4^1 W_4^2 X_2(3) \\ &= X_1(1) - W_4^1 X_2(1) \end{aligned} \quad (21)$$

ผลของสมการ (17) ถึง (21) ได้ใช้คุณสมบัติของความเป็นคาบ (Periodicity) กับ $X_1(k)$ และ $X_2(k)$ โดยความรู้ที่ว่า $W_4^2 = -1$ เรียกสมการทั้ง 4 นี้ว่า Recomposition Equation ซึ่งสามารถเขียนอธิบายได้ด้วย Signal Flow Graph ดังรูปข้างล่างที่ 2.2 และในรูปที่ 2.3 หน้าถัดไปเป็นการแสดงถึงการคำนวณ FFT DIT แบบ 16 จุด



รูปที่ 2.3 แสดงการ Decomposition DFT 4 จุด



รูปที่ 2.3 แสดง FFT การลดทอนทางเวลา 16 จุดที่มีลำดับเข้าแบบผกผันบิต

ผลที่เราได้จากการกระทำขั้นตอนดังกล่าวข้างต้น (Algorithm) ก็คือจำนวนครั้งในการคำนวณเมื่อเปรียบเทียบกับ การคำนวณการแปลงฟูริเยอร์เต็มหน่วยจากข้อกำหนดมาตรฐานจะเห็นว่า การคำนวณจะถูกแบ่งออกเป็น หลาย ๆ ขั้นตอน (Stages) จากการวิเคราะห์ข้างต้นทำให้เราพบว่าจะมีขั้นตอนหนึ่งที่ต้องคำนวณ คือ DFT 2 จุด $N/2$ จุด และมีขั้นตอนของการ Recomposition ซึ่งมี R ทำให้เกิด

$$N/2^R = 2 \quad (22)$$

หากทำการใส่ \log_2 จำนวน 2 เข้าไปทั้งสองข้างของสมการข้างต้นจะได้ว่า

$$\log_2 N - \log_2 2^R = \log_2 2 \quad (23)$$

หรือ $\log_2 N - R = 1$ และ $R = \log_2 N - 1$ (24)

ผลที่ได้เป็นจำนวนขั้นตอน (Stages) ในการคำนวณ เมื่อนำมารวมกับอีกหนึ่งขั้นในการคำนวณ DFT 2 จุด จะได้จำนวนขั้นตอนการคำนวณทั้งหมดที่ต้องการคือ $R + 1$ หรือ

$$\text{จำนวนขั้น} = \log_2 N \quad (25)$$

ในแต่ละขั้นตอนจะมีการคำนวณของหน่วยผีเสื้อ (Butterfly) $N/2$ ครั้งและเมื่อมีจำนวนขั้น $\log_2 N$ ขั้น จะได้จำนวนครั้งในการคำนวณหน่วยผีเสื้อ (Butterfly) ทั้งหมดคือ

$$\text{จำนวนหน่วยผีเสื้อทั้งหมด} = (\text{จำนวนหน่วยผีเสื้อต่อขั้นตอน}) * (\text{จำนวนของขั้นตอน}) \quad (26)$$

$$\text{Total Butterfly} = N/2 * \log_2 N \quad ; \text{ เมื่อ } N \text{ เป็นเลขกำลังของ } 2 \quad (27)$$



โดยปกติในแต่ละหน่วยผีเสื้อ (Butterfly) จะต้องมีการคูณเชิงซ้อนและการทำการบวก 2 ครั้ง (ปกติเป็นการบวก 1 ครั้ง และการลบ 1 ครั้ง) ดังนั้นจำนวนการคูณเชิงซ้อนที่จำเป็นต้องกระทำในการคำนวณ DFT โดยใช้ DIT FFT อัลกอริทึมการคูณเลขเชิงซ้อน $N/2 \log_2 N$ ครั้ง ซึ่งจากที่ทราบว่า DFT ตามข้อกำหนดต้องทำการคูณจำนวนเชิงซ้อน N^2 ครั้ง ฉะนั้นจะพบว่า FFT จะให้การคำนวณที่เร็วกว่า DFT มาตรฐานดังตารางที่ 2.1 แสดง จำนวนครั้งการคูณเลขเชิงซ้อนสำหรับ DFT และ FFT ฐาน 2

N	Standard DFT	FFT	Ratio DFT/FFT
2	4	1	4.0
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.3
128	16,384	448	36.6
256	65,536	1024	64.0
512	262,144	2304	113.8
1024	1,048,576	5120	204.8

ตารางที่ 2.1 แสดง จำนวนครั้งการคูณเลขเชิงซ้อนสำหรับ DFT และ FFT ฐาน 2 จากตารางชี้ให้เห็นว่ายิ่ง N ยิ่งมากจำนวนครั้งในการคำนวณยิ่งแตกต่างกันมากขึ้น

การเขียนโปรแกรมคอมพิวเตอร์เพื่อการทำตามขั้นตอนวิธีการลดทอนเวลา (DIT) จำเป็นที่เราจะต้องหาสูตรในการคำนวณ ตัวประกอบหมุน W และต้องทราบว่าจะนำข้อมูลคู่ใดในลำดับมาทำการคำนวณในหน่วยผีเสื้อ หากพิจารณาในรูปที่ 2.3 จะพบว่าเลขยกกำลังของตัวประกอบหมุนมีคุณสมบัติของความเป็นคาบและในทศ ๗ คาบ เริ่มต้นด้วยค่าเท่ากับ 0 ถ้ากำหนดให้เลขของจำนวนขั้นตอนของหน่วยผีเสื้อ เริ่มด้วยขั้นตอนทางซ้ายสุดเป็นขั้นที่ 1, 2, ... ถึง $\log_2 N$ ขั้น จะพบว่า การเพิ่มของเลขยกกำลังในกลุ่มของตัวประกอบหมุนที่ขั้นที่ L th คือ $N/2^L$ หากสมมุติว่าลำดับข้อมูลเข้าในแต่ละขั้นตอนมีที่เก็บ ω ตำแหน่งเริ่มต้นเดียวกันจะได้ว่าตำแหน่งถัดไปจะเก็บข้อมูลในลำดับการคำนวณถัดไปนอกจากนี้ยังพบว่าในขั้นตอนที่ L th ความกว้างของหน่วยผีเสื้อคือ $2^{(L-1)}$

ในการจัดลำดับของข้อมูลเข้า (Input Sequence) ไม่ใช่การจัดลำดับธรรมดาแต่เป็นการจัดลำดับเข้าของข้อมูลในรูปของ การผันกลับบิต (Bit reversing) ตัวอย่างของการกำหนดลำดับข้อมูลเข้าเช่น $x(8) = x(1000_2)$ จะต้องทำการวางไว้ ณ ตำแหน่งที่ (0001_2) คือตำแหน่ง 1 เพื่อความเข้าใจลองพิจารณาในตารางที่ 2.2

Sequence element	Binary address	Bit-reversed address	Desired displacement from base address
x(0)	0000	0000	0
x(1)	0001	1000	8
x(2)	0010	0100	4
x(3)	0011	1100	12
x(4)	0100	0010	2
x(5)	0101	1010	10
x(6)	0110	0110	6
x(7)	0111	1110	14
x(8)	1000	0001	1
x(9)	1001	1001	9
x(10)	1010	0101	5
x(11)	1011	1101	13
x(12)	1100	0011	3
x(13)	1101	1011	11
x(14)	1110	0111	7
x(15)	1111	1111	15

ตารางที่ 2.2 แสดงการผกผันกลับบิต

การคำนวณการแปลงฟูริเยอร์โดยการลดทอนทางความถี่

วิธีการคำนวณหา DFT อีกวิธีหนึ่งก็คือการลดทอนทางความถี่ Decomposition หรือ Decimation in Frequency DIF ซึ่งมีแนวความคิดคล้ายคลึงกันกับการลดทอนทางเวลา Decomposition หรือ Decimation In Time DIT และทำให้ได้ผลของจำนวนครั้งในการคูณผลลงในส่วนนี้เราจะอธิบาย การหา DIF เพื่อเปรียบเทียบกับ DIT

พิจารณาการคำนวณ DFT แบบ 2 จุด โดยขั้นตอนลดทอนทางความถี่จะได้ว่าจากสมการข้อกำหนดของ DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad ; k = 0, 1, \dots, N-1 \quad (28)$$

ในขั้นตอนการลดทอนทางเวลา เราได้แบ่งส่วนของลำดับในโดเมนเวลาเป็นลำดับย่อย ๆ 2 ลำดับซึ่งมีดัชนีเป็นเลขคู่และเลขคี่ แต่ในทางกลับกันในวิธีของการลดทอนทางความถี่เราจะทำการแบ่งลำดับเป็นลำดับย่อย ๆ ซึ่งประกอบด้วยส่วนของลำดับครึ่งแรกและส่วนของลำดับครึ่งหลังของลำดับ x(n) จากการทำดังกล่าวทำให้สามารถเขียนสมการ (28) ได้ใหม่ว่า

$$X(k) = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \quad ; k = 0, 1, \dots, N-1 \quad (29)$$

เมื่อทำการพิจารณาในเทอมที่ 2 จะสามารถเขียนได้เป็น

$$\begin{aligned} \sum_{n=N/2}^{N-1} x(n)W_N^{nk} &= \sum_{n=0}^{N/2-1} x(n+N/2)W_N^{(n+N/2)k} \\ &= \sum_{n=0}^{N/2-1} x(n+N/2)W_N^{(N/2)k}W_N^{nk} \end{aligned} \quad (30)$$

และจาก $W_N^{(N/2)k} = e^{-j(2\pi/N)(n/2)k} = e^{-j\pi k} = (-1)^k$ (31)

ดังนั้น $X(k)$ จากสมการ (29) เขียนได้เป็น

$$X(k) = \sum_{n=0}^{N/2-1} [x(n) + (-1)^k x(n+N/2)] W_N^{nk}; \quad k = 0, 1, \dots, N-1 \quad (32)$$

แม้ว่าสมการ (32) จะมีลักษณะเหมือน DFT $N/2$ จุด แต่จริงๆ แล้วไม่ใช่ เพราะว่าการถ่วงน้ำหนักด้วยแฟกเตอร์ของ W_N^{nk} มากกว่า $W_{N/2}^{nk}$ และแนวทางสำหรับการพิจารณาขั้นต่อไปก็คือเทอมของ $(-1)^k$ ในเทอมที่สองของผลรวม เมื่อค่าของ k เป็นเลขคู่ เทอมที่สองจะเป็น 1 ขณะที่ค่าของ k เป็นเลขคี่ค่าของเทอมที่สองจะเป็น -1 ถ้าหากพิจารณาค่าของ k แยกจากกันหรือนั่นก็คือการ Decompose ในแกนความถี่ ดังนั้นอัลกอริทึมของ DIF สำหรับค่า k เลขคู่เขียนได้เป็น

$$X(2k) = \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)] W_N^{2kn}; \quad k = 0, 1, \dots, N/2-1 \quad (33)$$

และสำหรับค่า k เลขคี่เราจะได้

$$X(2k+1) = \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_N^{(2k+1)n}; \quad k = 0, 1, \dots, N/2-1 \quad (34)$$

จากการพิจารณาพบว่า

$$W_N^{2nk} = e^{-j(2\pi/N)2nk} = e^{-j(2\pi/N/2)nk} = W_{N/2}^{nk} \quad (35)$$

และ $W_N^{(2k+1)n} = W_N^n W_N^{2kn}$ (36)

แต่เมื่อใช้สมการ (35) จะได้ว่า

$$W_N^{(2k+1)n} = W_N^n W_{N/2}^{nk} \quad (37)$$

ดังนั้น เราสามารถเขียน สมการ (33) จะได้ว่า

$$X(2k) = \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)] W_{N/2}^{nk} ; k = 0, 1, \dots, N/2-1 \quad (38)$$

หรือเขียนเป็นคำกล่าวได้ว่า

$$X(2k) = \text{DFT } N/2 \text{ จุด ของ } [x(n) + x(n+N/2)] \quad (39)$$

ดังนั้นในวิธีเดียวกันโดยใช้สมการ (36) เราจะเขียนสมการ (34) ได้ว่า

$$X(2k+1) = \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_{N/2}^{nk} ; k = 0, 1, \dots, N/2-1 \quad (40)$$

หรือเขียนเป็นคำกล่าวได้ว่า

$$X(2k+1) = \text{DFT } N/2 \text{ จุด ของ } [x(n) - x(n+N/2)] W_{N/2}^{nk} \quad (41)$$

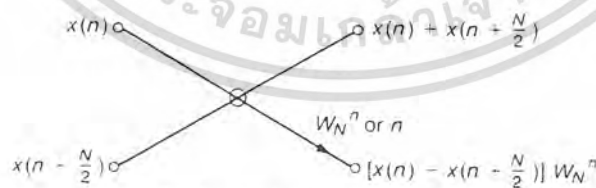
หากพิจารณา Signal Flow Graph ของสมการทั้ง (38) และ (40) เราจะได้ว่าลำดับที่ใช้ในการคำนวณคือ

$$x(n) + x(n+N/2) \quad \text{และ} \\ [x(n) - x(n+N/2)] W_{N/2}^{nk}$$

ในสัญลักษณ์ของ Signal Flow Graph ในรูปแบบ 2 จุดแสดงดังรูปที่ 2.3 ข้างล่าง และสามารถเขียนให้อยู่ในรูปของ 16 จุดได้ดังรูปที่ 2.4

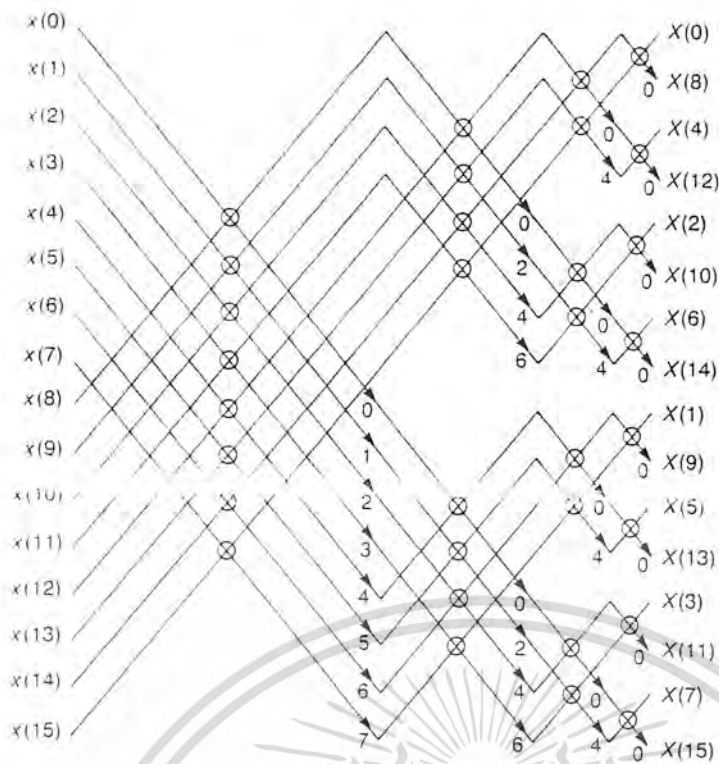


(a) Signal flow graph for DIF butterfly



(b) Shorthand notation for DIF butterfly

รูปที่ 2.3 แสดงหน่วยผีเสื้อของการลดทอนทางความถี่



รูปที่ 2.4 แสดงการคำนวณการลดทอนทางความถี่เวลาของ ของ FFT 16 จุด

การเปรียบเทียบการแปลงฟูริเยอร์โดยใช้ การลดทอนทางเวลาและ การลดทอนทางความถี่ แบบ 16 จุด จะได้ว่าลักษณะโครงสร้างของหน่วยสี่เหลี่ยมในวิธีการลดทอนทางความถี่ เป็นการพิจารณาได้ว่าจะมี $\log_2 N$ ขั้นตอน ของหน่วยสี่เหลี่ยมและแต่ละขั้นจะประกอบด้วยการคำนวณเชิงซ้อน $N/2$ หน่วยสี่เหลี่ยม จะสังเกตได้ว่า DIT butterfly algorithm ต้องการการคูณโดยตัวประกอบหมุน W (Weighting Factor) ก่อนที่จะมีการบวกและการลบ ในขณะที่ DIF butterfly algorithm จะมีการบวกและลบก่อนที่จะมีการคูณด้วยตัวประกอบหมุนดังกล่าว และไม่ว่าจะด้วยวิธีใดในทั้ง 2 วิธี จะได้ว่าอัลกอริทึมของจำนวนครั้งในการคูณจำนวนเชิงซ้อนคือ

$$T_B = N/2 \log_2 N \quad \text{สำหรับ } N \text{ เป็นเลขกำลังของ } 2$$

และเช่นเดียวกันกับผลของ DIT พบว่าในรูป DIF 16 จุด จะมีกำลังของตัวประกอบหมุน W มีคุณสมบัติความเป็นคาบ และในทุก ๆ คาบ เลขยกกำลังเริ่มต้นที่ 0 และจำนวนขั้นตอนของหน่วยสี่เหลี่ยม (Butterfly) จากซ้ายไปขวาเริ่มด้วยขั้นตอนที่ 1 และคำนวณไปจนถึงขั้นตอนที่ $\log_2 N$ เมื่อสังเกตจะทราบว่า การเพิ่มเลขกำลังของ W ณ ขั้นตอนที่ L คือ $2^{(L-1)}$ สำหรับการแยกหน่วยของจุดซึ่งอยู่ในโครงสร้างหน่วยสี่เหลี่ยม n ขั้นตอนที่ L คือ $N/2^L$

จากการสังเกตจะรู้ว่าการเพิ่มของกำลังและการแยกกลุ่มของจุด ของการลดทอนทางความถี่ได้เปลี่ยนไปเมื่อเปรียบเทียบกับลดทอนทางเวลา ซึ่งในความเป็นจริง การลดทอนทั้งสองมีคุณสมบัติของความถี่ (Duality) ซึ่งโดยพิจารณาจากลำดับของข้อมูลเข้ามีการเรียงในการลดทอนทางความถี่และจะมีข้อมูลออกเป็นลำดับแบบผกกลับบิต ในขณะที่ตรงกันข้ามกับการลดทอนทางเวลาที่ ข้อมูลเข้าของ

การลดทอนทางเวลามีลักษณะผันกลับบิท และได้ข้อมูลออกมีการเรียง
ผลที่เกิดจากการทำ window สัญญาณ

ในการวิเคราะห์สัญญาณโดยการแปลงฟูรีเยอร์เต็มหน่วยนั้น สัญญาณข้อมูลเข้าที่นำมาใช้จะมี
จำนวนจำกัด ซึ่งหมายถึงการที่เรา Window ให้สัญญาณข้อมูลที่มีอยู่จำนวนมากเข้ามาทำการคำนวณเพียง
บางส่วนเท่านั้น และการ Window ในลักษณะนี้เรียกว่า Rectangle window

จากคุณสมบัติของการแปลงฟูรีเยอร์เต็มหน่วย สัญญาณข้อมูลเข้าที่ผ่านการสุ่มเข้ามาจะถูกสมมติ
ว่ามีลักษณะเป็น periodic ซึ่งมีคาบเวลาเท่ากับเวลาที่ใช้ทำการ Sampling สัญญาณ 1 ชุด หรือเท่า
กับความกว้างของ Window นั่นเองดังแสดงในรูปที่ 2.5

รูปที่ 2.5 แสดงการ window สัญญาณ

จากรูปที่ 2.5 จะเห็นว่า การ window สัญญาณควรจะให้เฟสของสัญญาณในตอนต้นของการ
Window เท่ากันกับเฟสที่ปลายของ Window มิฉะนั้นสัญญาณที่วิเคราะห์จะเกิดความไม่ต่อเนื่อง ซึ่งจะ
ทำให้เกิด Spectrum ของความถี่อื่นๆที่ไม่ใช่อันที่ประกอบ
ความถี่ของสัญญาณจริงเกิดเป็น Leakage ขึ้น

ในการแก้ปัญหาความไม่ต่อเนื่องนี้สามารถแก้ไขโดยการ Window สัญญาณให้มีค่าลดลงมีส่วน
ปลายทั้ง 2 ข้างของ Window ซึ่งจะช่วยให้สัญญาณของ Window ถัดไปมีความต่อเนื่องกันมากขึ้น

ในโดเมนเวลาสัญญาณข้อมูลเข้าถูกคูณกับ Window ที่เลือกใช้ก่อนนำมาแปลงฟูรีเยอร์ ซึ่งเมื่อ
วิเคราะห์ในโดเมนความถี่ การ Window สัญญาณจะนำ Spectrum ของสัญญาณ input มา Convolution
กันกับ Spectrum ของ Window ดังนั้นในการเลือกใช้ Window จึงต้องคำนึงถึงคุณสมบัติของ Spectrum
ของ Window นั้นๆ ซึ่งแสดงในตารางที่

Spectrum ของ window แต่ละชนิดจะประกอบด้วยขนาดของ main lobe และ side lobe ที่
แตกต่างกันออกไป ซึ่งในการเลือก Window ควรเลือกให้มี side lobe ต่ำๆ ค่า main lobe สูง และมี
bandwidth แคบๆ จากตารางจะพบว่า Window แบบ Rectangular มีค่า main lobe และ bandwidth แคบ
แต่ก็มี side lobe สูงด้วยซึ่ง side lobe นี้เองที่ทำให้ leakage เกิดได้สูงด้วย

จะสังเกตได้ว่าในตารางจะเรียงจากวินโดว์ที่มี side lobe และ main lobe สูงไปยัง window ที่มี
side lobe และ main lobe ต่ำ ส่วน bandwidth จะเรียงกลับกัน ยกเว้นแต่ Hamming window ซึ่งแม้ว่าจะ
มี side lobe ต่ำกว่า window ที่อยู่ข้างบน แต่ก็มี main lobe ที่สูงและมี bandwidth ที่ต่ำกว่า ซึ่งคุณสมบัติ
ดังกล่าวเหมาะที่จะนำมาใช้สำหรับการ Window โดยทั่วไป

สำหรับในกรณีที่เราทราบคาบของสัญญาณที่แน่นอนเราก็สามารถทำการ window ให้สัญญาณ มี
ความต่อเนื่องกันได้ซึ่งจะทำให้ Leakage มีค่าต่ำ ดังนั้น Rectangle window จึงเป็น window ที่เหมาะสม
เนื่องจากมี main lobe ที่สูงและมี bandwidth ต่ำ

ตารางที่ 2.3 แสดงคุณสมบัติของวินโดว์ชนิดต่างๆ

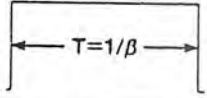
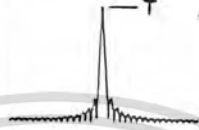

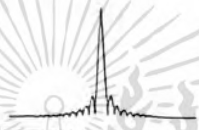


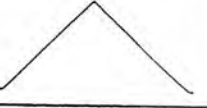
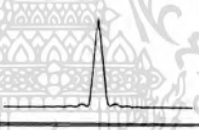



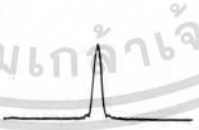

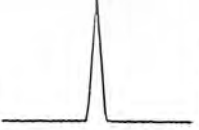
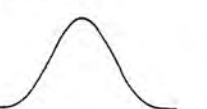



Unity Amplitude Window	Shape Equation	Frequency Domain Magnitude	Major Lobe Height	Highest Side Lobe (dB)	Bandwidth (3 dB)	Theoretical Roll-Off (dB/Octave)
Rectangle 	$A=1$ for $t=0$ to T		T	-13.2	0.86β	6
Extended Cosine Bell 	$A=0.5(1-\cos 2\pi 5t/T)$ for $t=0$ to $T/10$ and $t=9T/10$ to T $A=1$ for $t=T/10$ to $9T/10$		$0.9 T$	-13.5	0.95β	18 (beyond 5β)
Half Cycle Sine 	$A=\sin 2\pi 0.5t/T$ for $t=0$ to T		$0.64 T$	-22.4	1.15β	12
Triangle 	$A=2t/T$ for $t=0$ to $T/2$ $A=-2t/T + 2$ for $t=T/2$ to T		$0.5 T$	-26.7	1.27β	12
Cosine ² (Hanning) 	$A=0.5(1-\cos 2\pi t/T)$ for $t=0$ to T		$0.5 T$	-31.6	1.39β	18
Half Cycle Sine ³ 	$A=\sin^3 2\pi 0.5t/T$ for $t=0$ to T		$0.42 T$	-39.5	1.61β	24
Hamming 	$A=0.08 + 0.46(1-\cos 2\pi t/T)$ for $t=0$ to T		$0.54 T$	-41.9	1.26β	6 (beyond 5β)
Cosine ⁴ 	$A=(0.5(1-\cos 2\pi t/T))^2$ for $t=0$ to T		$0.36 T$	-46.9	1.79β	30
Parzen 	$A=1-6(2t/T-1)^2+6(2t/T-1)^3$ for $t=T/4$ to $3T/4$ $A=2(1-2t/T-1)^3$ for $t=0$ to $T/4$ and $t=3T/4$ to T		$0.37 T$	-53.2	1.81β	24

Table 7-1 Some common data windows and their frequency-domain parameters.

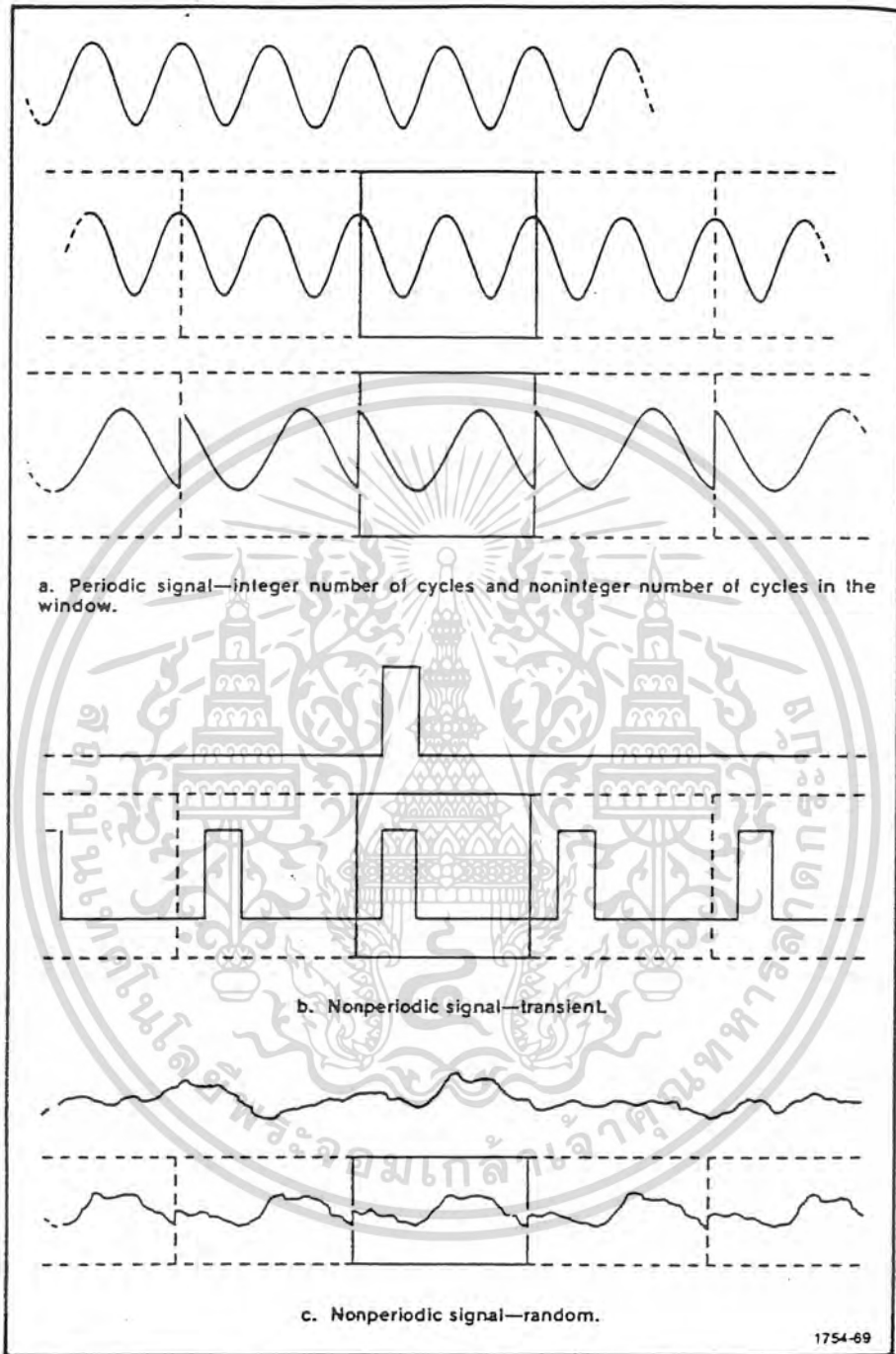


Figure 7-2 The type of waveform doesn't change the FFT's point of view. Periodicity is always assumed. However, you can change your point of view, according to the type of signal, in interpreting FFT results.

รูปที่ 2.5 แสดงการ Window สัญญาณ

การเกิด Folding Spectrum

ในการ Sampling สัญญาณจะถูกกำหนดไว้ว่าจะต้องทำการ Sampling อย่างน้อย 2 เท่าของความถี่สัญญาณ ซึ่งเมื่อทำการแปลงฟูริเยอร์จะสามารถแสดงสเปกตรัมของสัญญาณได้สูงสุดเท่ากับความถี่ Nyquist แต่ในกรณีที่สัญญาณที่ทำการ Sampling มีความถี่สูงกว่าความถี่ Nyquist จะต้องทำการกรองออกเสียก่อนมิฉะนั้นสเปกตรัมของความถี่ดังกล่าวจะมาปรากฏอยู่ใน Range ของความถี่ Nyquist เนื่องจากสัญญาณความถี่สูงดังกล่าวจะถูก Sampling แต่เนื่องจาก Sampling มีความถี่ต่ำจึงมีโอกาสที่การ Sampling สัญญาณความถี่สูงนั้นไปตรงกันกับการ Sampling สัญญาณความถี่ที่ต่ำได้ดังรูปที่ 2.6

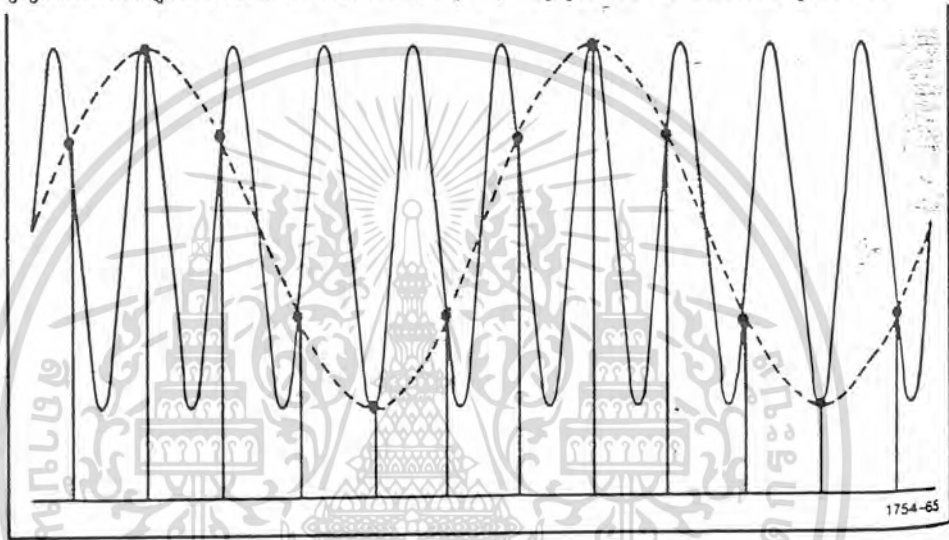


Figure 6-14 Insufficient sampling of a high-frequency component results in a low-frequency alias.

รูปที่ 2.6 แสดงการ Sampling สัญญาณที่มีความถี่สูงกว่า Nyquist Frequency ซึ่งในโดเมนความถี่ Spectrum ของสัญญาณที่ความถี่สูงกว่า Nyquist Frequency จะ Folder มาใน Spectrum ของสัญญาณที่มีความถี่ต่ำกว่า Nyquist Frequency

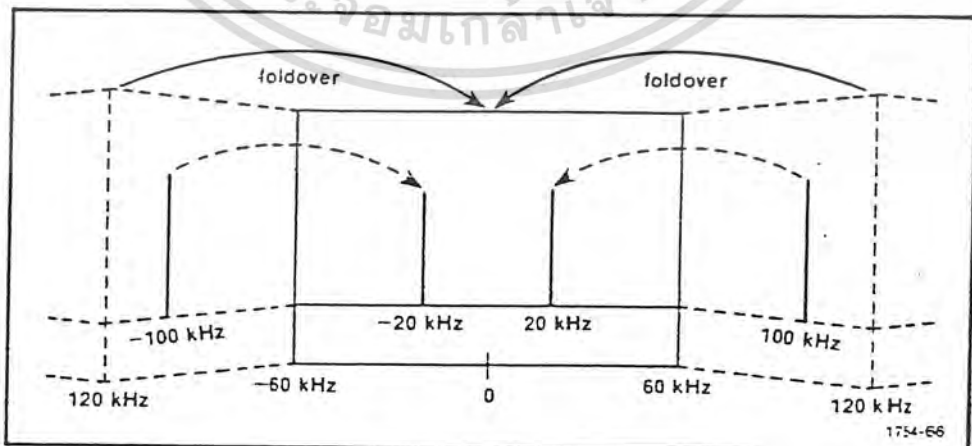


Figure 6-15 When the sampling rate is 60 kHz, a 100-kHz component is folded down to become a 20-kHz alias.

รูปที่ 2.7 แสดง การเกิด Folding Spectrum

จากรูปแสดงให้เห็นว่าในกรณีที่ Nyquist Frequency มีค่าเท่ากับ 60 KHZ สัญญาณ ความถี่ 100 KHZ จะถูก Foldover มาที่ความถี่ 20 KHZ ดังนั้นก่อนการ Sampling สัญญาณทุกครั้งจึงควรจะกรองสัญญาณที่มีความถี่สูงกว่าความถี่ Nyquist ออกเสียก่อน



การรบกวนในระบบไฟฟ้า

การรบกวนในระบบไฟฟ้า สามารถเกิดขึ้นได้ทั่วไปซึ่งอาจจะเกิดจากต้นทาง เช่น แหล่งจ่ายไฟ หรือปลายทางจากโหลด เช่น โหลดบางชนิด ก็สามารถรบกวนกันเอง การรบกวนสามารถผ่านสื่อเชื่อมโยง(Connection) หรือแพร่กระจาย(Radiation) ไปยังโหลด (Victem) ที่เปราะบางทำให้เสียหายได้ โหลดที่มักกล่าวถึงมักเป็นพวก อิเล็กทรอนิกส์ เช่น คอมพิวเตอร์, เครื่องมือสื่อสาร, เครื่องมือแพทย์ ฯลฯ

ประเภทของการรบกวนทางไฟฟ้า

1. Brownout คือ ปรากฏการณ์ของแรงดันเกิดเปลี่ยนแปลงทันทีโดย Amplitude ผิดไปจากปกติ 10-100% และเวลาจาก ≤ 10 ms จนเป็นวินาที
ความเสียหาย
 - อุปกรณ์คอมพิวเตอร์อาจจะหยุดทำงาน เนื่องจากวงจรตรวจจับแรงดันของคอมพิวเตอร์ทำงานผิดพลาด
 - hardware เสีย
 - ข้อมูลเสีย
2. Impulse หรือ Spike คือ ปรากฏการณ์ที่แรงดันเปลี่ยนแปลงอย่างรวดเร็วและรุนแรงเป็น KV ในระยะเวลาที่สั้นมากเป็น μsec ตัวอย่างเช่น IEC 60 Standard impulse จะมี rise time จาก 0-90% ของ PEAK ภายใน $1.2 \mu\text{sec}$ ที่ PEAK Amplitude 6KV
ความเสียหาย
 - ข้อมูลบน disk หรือ tape เสียหาย
 - การประมวลผลผิดพลาด (processing error)
 - อุปกรณ์เสียหายถาวร (permanent damage)
3. Harmonic คือ ปรากฏการณ์ของคลื่นรบกวนที่มีความถี่เป็นจำนวนเท่าของความถี่มูลฐาน ซึ่งสอดแทรกไปบนคลื่น sine wave ทำให้เกิดความเพี้ยน (distortion) ทั้งแรงดันและกระแส
ความเสียหาย
 - ทำให้ต้อง oversize ขนาดของ hardware
 - ทำให้เกิด over heat ซึ่งมีผลกับฉนวนในระบบไฟฟ้า
 - ทำให้เกิด resonance ใน capacitor
4. High frequency noise คือ ปรากฏการณ์ของคลื่นรบกวน ที่แทรกมาบนคลื่นแรงดัน เป็น

คลื่นที่มีความถี่สูง แต่มีค่า Amplitude ต่ำ ช่วง 0.5 millivolt จนถึง 30 volt

ความเสียหาย

- เกิด malfunction ของ data transfer
- เกิด power supply เสีย

5. Frequency variation

คือ ปรากฏการณ์ที่แหล่งจ่ายไฟ มีความถี่ไม่คงที่ เช่น ความถี่ไฟฟ้า จาก Generator

ความเสียหาย

- disk หรือ tape เสียหาย (chash)
- ข้อมูลบน disk หรือ tape เสียหายระหว่างการบันทึก

6. Blockout

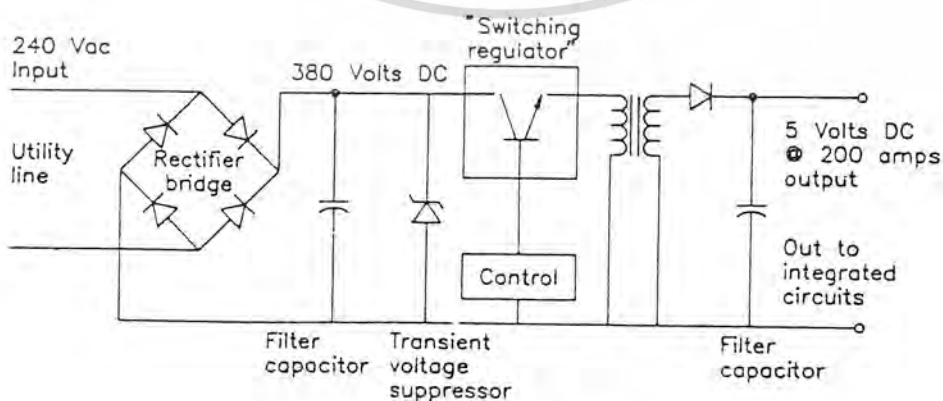
คือ ปรากฏการณ์ที่แรงดันขาดหายไป มีตั้งแต่เวลา ≤ 10 ms (micro-break) จนถึงชั่วโมง(ไฟดับ)

ความเสียหาย

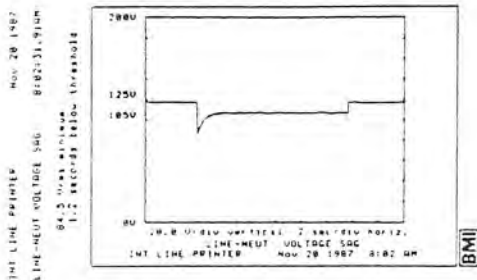
- ระบบคอมพิวเตอร์หยุด
- ข้อมูลเสียหาย (lose)
- เสียเวลาและ operating expense เพิ่ม

สาเหตุของการรบกวนกำลังไฟฟ้าต่ออุปกรณ์อิเล็กทรอนิกส์

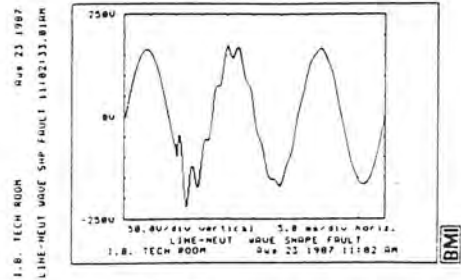
การอธิบายความเปราะบาง(sensitive) ของอุปกรณ์อิเล็กทรอนิกส์ต่อความผันแปรของกำลังไฟฟ้านั้นเป็นสิ่งที่ยาก การวิเคราะห์ power supply ที่เกี่ยวข้องกับคอมพิวเตอร์ จะช่วยให้เข้าใจถึงสาเหตุการเกิดความผิดปกติได้ดีกว่า ในอุปกรณ์อิเล็กทรอนิกส์ ทั้งขนาดใหญ่และขนาดเล็กจะต้องแปลงกระแสสลับ input เป็นแรงดันต่ำกระแสตรง โดยใช้วงจรอิเล็กทรอนิกส์ซึ่งประกอบไปด้วยส่วนวงจร rectifier, filter และ regulation power supply ดังแสดงในรูปที่ 2.8 แสดงวงจร Power supply



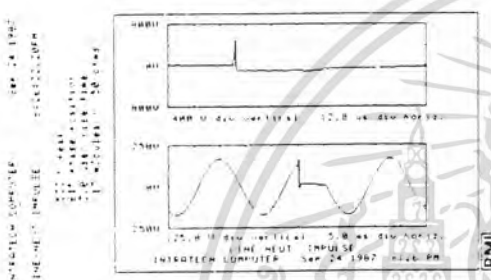
Typical computer power supply (1000 watt rating)



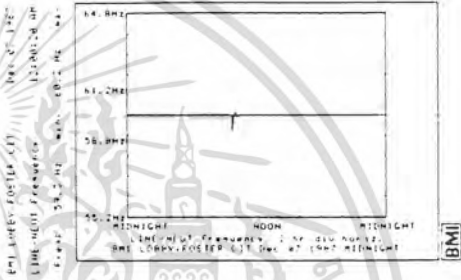
Rms/Vdc



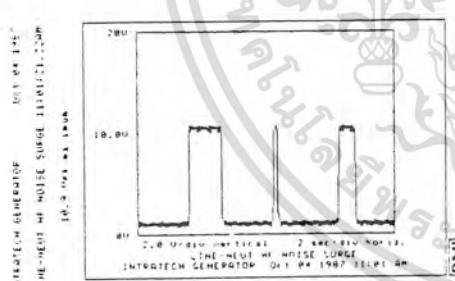
Waveshape faults



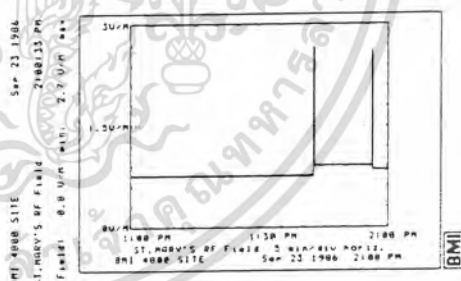
Impulses



Frequency



High-frequency Noise



Radio Frequency Interference

HARM	PCT	PHASE	HARM	PCT	PHASE
FUND	100.0%	0°	2nd	0.6%	125°
3rd	72.5%	209°	4th	0.6%	101°
5th	22.0%	190°	6th	0.5%	127°
7th	24.5%	119°	8th	0.4%	121°
9th	2.0%	1°	10th	0.4%	200°
11th	6.9%	49°	12th	0.5%	102°
13th	6.5%	207°	14th	0.4%	124°
15th	4.6%	95°	16th	0.7%	201°
17th	0.3%	129°	18th	0.5%	91°
19th	1.2%	68°	20th	0.3%	221°
21st	2.1%	79°	22nd	0.1%	50°
23rd	4.4%	84°	24th	0.2%	121°
25th	2.1%	75°	26th	0.2%	122°
27th	0.9%	12°	28th	0.8%	100°
29th	0.5%	149°	30th	0.9%	130°
31st	0.3%	249°	32nd	1.2%	100°
33rd	0.6%	47°			
000	84.8%		EVEN	2.1%	
140:	84.9%				

Total Harmonic Distortion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปกราฟฟิก

แสดงการรบกวนทางไฟฟ้า
สามารถตรวจวัดได้จาก
power profiler

โดยทั่วไปแล้วไม่มีองค์ประกอบย่อย (component) ของ switching power supply ที่สามารถทนต่อแรงดันที่สูงหลายร้อย Volt ได้ยกเว้นตัวกำจัดแรงดันเปลี่ยนแปลงชั่วขณะ (transient voltage suppressor) เป็นองค์ประกอบเดียวของ Commercial Switching power supply ที่สามารถทนต่อแรงดันที่เกิน 200-300V จากแรงดันพิกัดได้ องค์ประกอบนี้ถูกออกแบบให้ช่วยตรวจจับแรงดันสูง (spike) ก่อนจะผ่านมายัง regulator และเข้าไปในวงจรที่ไว (sensitive circuit) แต่ตัว Suppressors ต้องการเวลาจำกัดในการทำงานการ spike ของแรงดันไฟฟ้าในช่วงเวลาสั้นและถ้าจะผ่านเข้าไปในระบบและทำความเสียหายได้

การที่ไม่มีระบบป้องกันในช่วง power supply input กับ utility lines นั้นผลที่เกิดขึ้นอย่างเห็นได้ชัดคือ เมื่อเกิดฟ้าผ่า หรือ crossed line จะทำให้เกิด input spike voltage ขนาดใหญ่ ซึ่งจะไม่เป็นผลดีต่ออุปกรณ์อิเล็กทรอนิกส์ โดยทั่วไปผู้ผลิตได้ออกแบบให้ใช้กับ power supply ที่สามารถกำจัดได้ 100% ของ input transient ที่เป็นไปได้ในเวลา 100% ผลดังกล่าวนี้ผู้ผลิตทั้งหมดต้องคำนวณและตัดสินใจเกี่ยวกับระดับการป้องกันด้วย



บทที่ 3

Harmonic Analyzer Hardware.

สำหรับระบบการทำงานของ Harmonic Analyzer มีหลักการเบื้องต้นจากการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)[1] โดยการประยุกต์ใช้ระบบไมโครโพรเซสเซอร์ในส่วนการทำงานของ Hardware ซึ่งจะประกอบด้วยระบบรับข้อมูล (Data acquisition) และส่วนของระบบควบคุมการรับข้อมูล (Controller) สำหรับการคำนวณผลของการแปลงฟาสต์ฟูริเยอร์จะทำบนเครื่องไมโครคอมพิวเตอร์ ฉะนั้นผลการแปลงก็จะเป็นระบบเวลาจริง(Real Time)

จากการออกแบบเพื่อทดสอบการทำงานในเบื้องต้นของระบบรับข้อมูล โดยแบ่งส่วนของการทำงานเพื่อทำการรับสัญญาณทางตัวตรวจจับ(Sensor)ซึ่งได้แก่ตัวตรวจจับความต่างศักย์(Voltage) และตัวตรวจจับทางกระแส(Current) โดยสัญญาณที่ได้จะอยู่ในรูปของสัญญาณความต่างศักย์ในทั้งสองแบบ เมื่อได้สัญญาณอินพุตแล้วก็นำสัญญาณเชิงอุปมานที่สนใจ (Analog Signal) มาทำการสุ่ม ตัวอย่างโดยตัวสุ่มตัวอย่างและคงค่า (Sampling and Hold) จากนั้นก็นำสัญญาณที่ได้ไปทำการแปลงเป็นสัญญาณเชิงเลข (Digital Signal) แล้วนำสัญญาณที่ได้ไปเก็บไว้ในหน่วยความจำชั่วคราว(RAM) เพื่อรอการนำไปทำการประมวลผลเชิงเลขด้วยเครื่องไมโครคอมพิวเตอร์อีกทีหนึ่ง

สำหรับในส่วนของการควบคุมการเก็บข้อมูลเราได้ใช้ระบบไมโครคอนโทรลเลอร์ขนาดข้อมูล 8 บิต ในการรับสัญญาณการแปลงสมบูรณ์(Conversion Complete) จากตัวแปลงสัญญาณเชิงอุปมานเป็นเชิงเลข เพื่อทำการกำหนดตำแหน่งของข้อมูลที่จะทำการเก็บลงในหน่วยความจำสำรองและคอยตรวจสอบตำแหน่ง ที่ใช้ในการเก็บข้อมูล นอกจากนี้ระบบควบคุมดังกล่าวได้ทำการเพิ่มวงจรรวมของระบบเวลา(Real time clock) สำหรับใช้ในการตั้งเวลาที่จะทำการสุ่มตัวอย่าง และหน้าที่ท้ายสุดคือทำการส่งข้อมูลที่ทำการเก็บไว้ในหน่วยความจำสำรองผ่านทางพอร์ทสื่อสารอนุกรม (Serial port) ไปยังโปรแกรมการแปลงฟาสต์ฟูริเยอร์บนระบบไมโครคอมพิวเตอร์

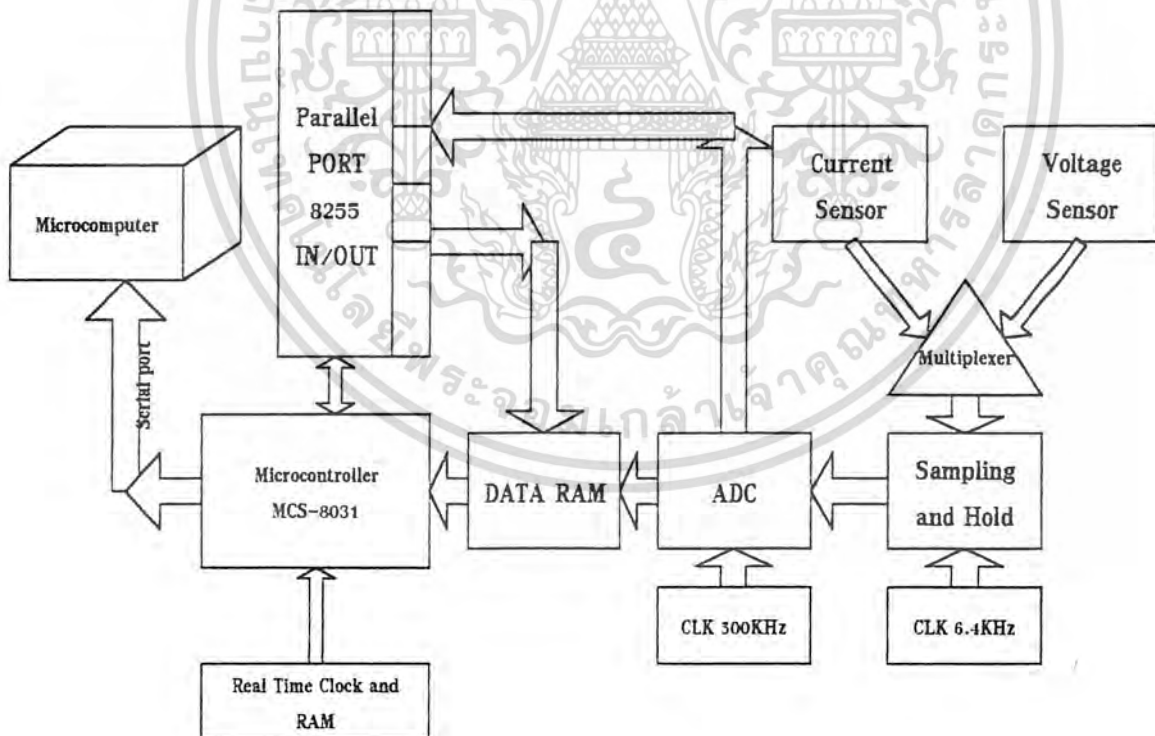
จากการทำงานข้างต้นทำให้จำเป็นต้องทราบถึงการใช้งานวงจรและการทำงานต่าง ๆ ดังต่อไปนี้ เพื่อนำไปประกอบการออกแบบเครื่องวิเคราะห์สัญญาณฮาร์โมนิกส์

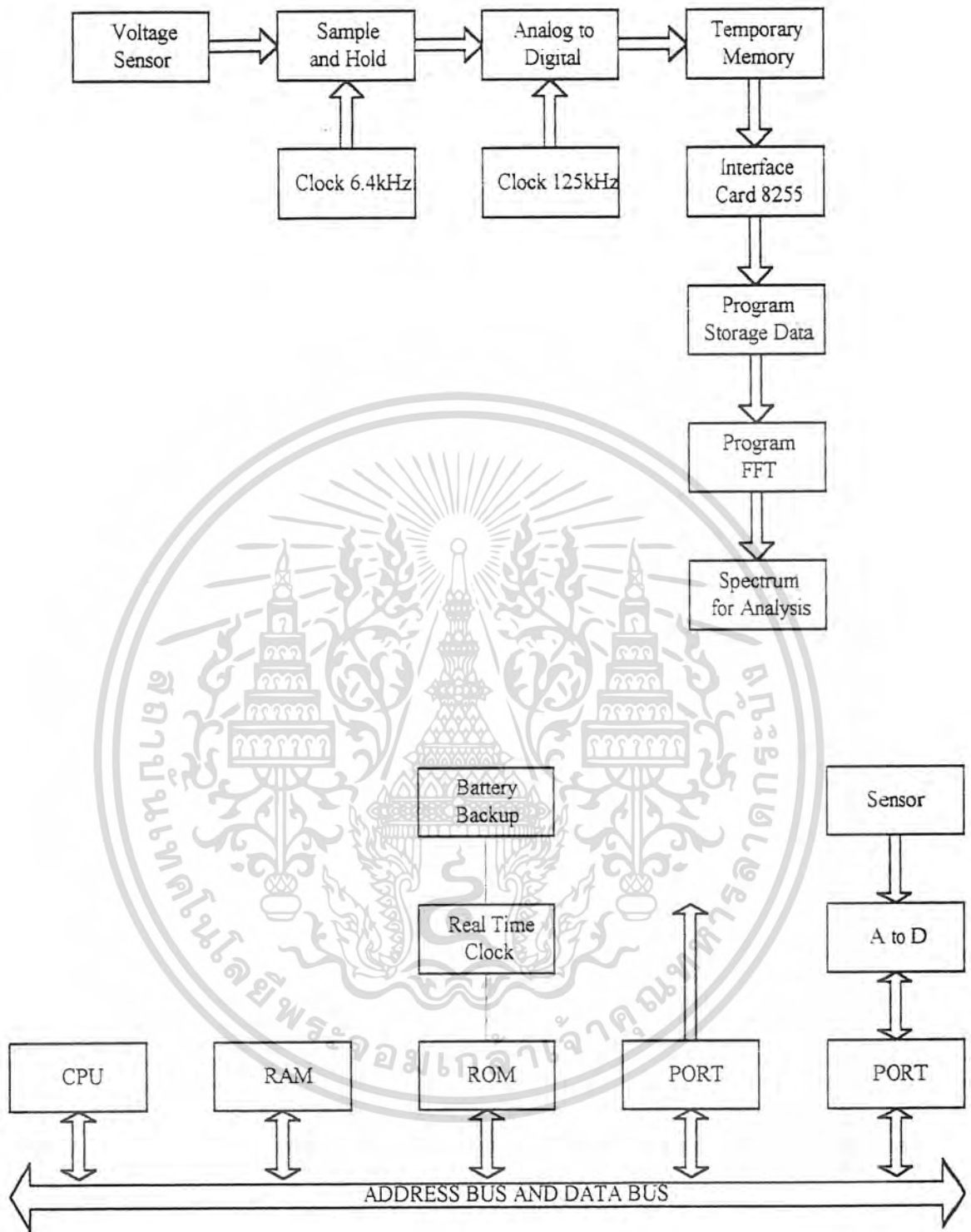
1. วงจรตัวตรวจจับสัญญาณเชิงอุปมาน ในส่วนความต่างศักย์และในส่วนกระแส (Sensor)และการมัลติเพล็กซ์ (Multiplexing)
2. วงจรการสุ่มตัวอย่างสัญญาณและคงค่า (Sample and Hold)
3. วงจรการแปลงสัญญาณอุปมานเป็นสัญญาณเชิงเลข (Analog to Digital Converter)

4. วงจรกำเนิดสัญญาณความถี่ในการสุ่มตัวอย่าง (Sampling Frequency Base Clock)
5. วงจรรักษาระดับแรงดันอ้างอิง (Voltage Reference)
6. วงจรสร้างสัญญาณโมโนสเตเบิล มัลติไวเบเรเตอร์ (Monostble Multivibrator)
7. วงจรหน่วยความจำชั่วคราว (Static Random Access Memory)
8. การทำงานของระบบควบคุมการเก็บข้อมูล (Microcontroller)
9. ผังแสดงการทำงานของสัญญาณนาฬิกาและสัญญาณความถี่ในการสุ่มตัวอย่าง

ส่วนของการประมวลผลเพื่อให้ได้สเปกตรัมของฮาร์โมนิกส์ที่ต้องการ ได้จัดทำโปรแกรมทำการเก็บข้อมูลที่ได้อมาทำการแปลงสัญญาณในโดเมนเวลาให้เปลี่ยนเป็นสัญญาณในโดเมนเชิงความถี่ เพื่อที่จะได้ทำการวิเคราะห์คุณสมบัติของสัญญาณตัวอย่างที่สนใจ ซึ่งอยู่ในส่วนของโปรแกรมประยุกต์บนระบบไมโครคอมพิวเตอร์

สำหรับผังแสดงการทำงานอย่างคร่าวๆ ของวงจรต่าง ๆ ได้แสดงไว้ในหน้าต่อไปโดยจะแสดงถึงส่วนของการทำงานทั้งหมดและหน้าถัดไปจะเป็นการอธิบายการทำงานของทำงานวงจรข้างต้น





บล็อกไอซีแถมแสดงระบบการทำงานโดยรวม จากรูปข้างบนเป็นขั้นตอนการพัฒนาเบื้องต้นของเทอมตันจะพบว่าได้มีการเพิ่มความถี่ในการทำงานของวงจรแปลง ADC และวิธีการเชื่อมต่อกับเครื่องไมโครคอมพิวเตอร์ และในส่วนของวงจรตรวจจับมีการเพิ่มวงจรตรวจจับกระแสขึ้นมา

1. วงจรตัวตรวจจับสัญญาณเชิงอนุพันธ์ในส่วนความต่างศักย์และในส่วนกระแส (Sensor) และการมัลติเพล็กซ์ (Multiplexing)

วงจรตัวตรวจจับในที่นี้เนื่องจากเราต้องการวัดตัวอย่างในสัญญาณระบบจ่ายไฟฟ้ากระแสสลับ ซึ่งเราสามารถตรวจวัดได้จากแหล่งจ่ายไฟไม่ว่าจะเป็นไฟฟ้ากระแสสลับตามบ้านที่อยู่อาศัย หรือที่ได้จากเครื่องจ่ายไฟฟ้าพวก Stabilizer หรือ Uninterrupted Power supply

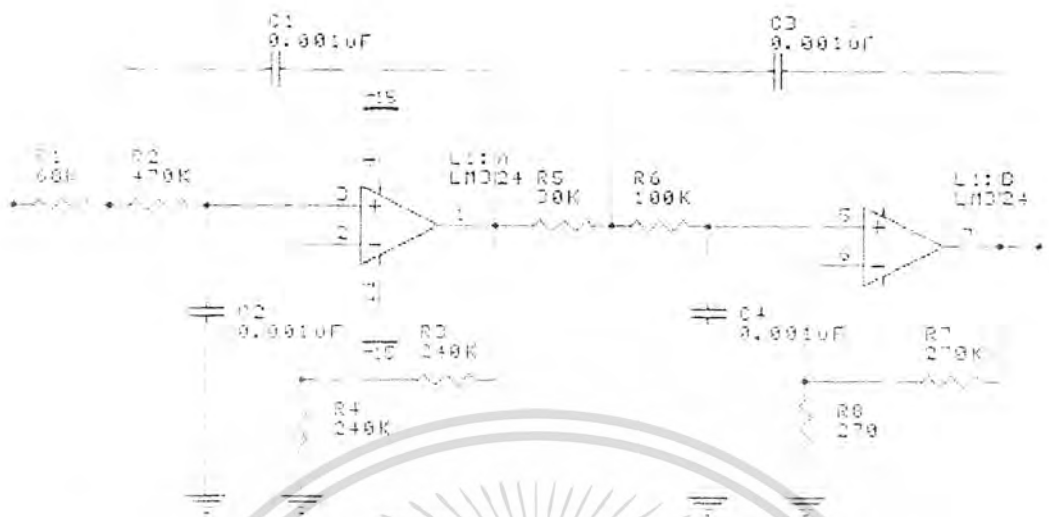
จึงได้ทำการออกแบบตัวตรวจจับอย่างง่ายของศักดาไฟฟ้าด้วยชุดของตัวต้านทานที่มีค่าความต้านทานสูงที่ทำการลดทอนความต่างศักย์ลงมาเป็นความต่างศักย์ที่เราต้องการและผ่านตัว Buffer OPAMP ดังแสดงในวงจรในรูปที่ 3.1 ข้างล่าง



รูปที่ 3.1 แสดงวงจรตรวจจับสัญญาณความต่างศักย์อย่างง่าย

ส่วนตัวตรวจจับอย่างง่ายของกระแสไฟฟ้า จะสร้างจาก Current transformer ที่เป็นขดลวดที่มีแกนกลางกลวง โดยให้กระแสไฟฟ้าผ่านในแกนของมันก็จะเกิดกระแสไฟฟ้าไหลในแกนของขดลวด โดยอาศัยหลักการเหนี่ยวนำของสนามแม่เหล็กไฟฟ้า และเมื่อนำ load มาต่อที่ ขดลวด จะสามารถวัดความต่างศักย์ที่ตกคร่อม load นำมาเป็นสัญญาณ input ได้ จากนั้นสัญญาณ input จะผ่านวงจร buffer และวงจรขยายเพื่อให้มีขนาดที่เหมาะสมในการนำไปแปลง A/D ซึ่งเพื่อให้สามารถใช้จำนวนบิตที่มีอยู่ของ A/D ได้อย่างเต็มที่ในการขยายสัญญาณของกระแสจึง กำหนด ให้มี Gain ที่แตกต่างกันสำหรับค่า I ที่ต่างกัน

สัญญาณความต่างศักย์และกระแสที่ได้จะต้องผ่านไปยังวงจร Low pass filter ที่มี cut off ที่ 3200 Hz เพื่อที่จะไม่ให้ Spectrum สัญญาณที่ความถี่สูงเข้าไปกวน Spectrum ของสัญญาณที่ทำการวิเคราะห์ ในส่วนที่เลือกใช้ Low pass filter แบบ butterworth ที่มี order = 4 ดังรูปที่ 3.2 สัญญาณจาก filter ทั้ง 2 สัญญาณจะถูก multiplex ด้วยความถี่เท่ากับ 6.4 KHz ซึ่งมีความถี่เป็นครึ่งหนึ่งของความถี่ที่ใช้ในการ Sampling วงจรรวมที่ใช้เป็น Multiplexer คือเบอร์ 4053



รูปที่ 3.2 แสดงวงจรกรองผ่านความถี่ต่ำ อันดับที่ 4 ที่ 3200 Hz

2. วงจรทำการสุ่มตัวอย่างสัญญาณและคงค่า (Sample and Hold)

สำหรับวงจรสุ่มตัวอย่างและคงค่า Sample and Hold อย่างง่ายสามารถทำได้โดยใช้วงจรรวมเบอร์ LF398 ซึ่งเป็น Monolithic Sample and Hold Circuits ซึ่งใช้เทคโนโลยีของ BI-FET เพื่อที่จะได้ความเที่ยงตรง(Accuracy)ของสัญญาณกระแสตรง (DC) พร้อมทั้งสามารถประยุกต์ให้ทำงานเก็บข้อมูลที่ความถี่สูงและมีอัตราส่วนสัญญาณที่ลดลงต่อเวลาที่ค้างค่าสัญญาณไว้ต่ำ (Low Droop Rate) นอกจากนี้ยังสามารถใช้ได้ทั้งกับอุปกรณ์ประเภท TTL, CMOS และ PMOS ด้วย

ขาสัญญาณในวงจรต้องทำการต่อวงจรให้เป็นดังนี้

Voltage +	+15 V	เข้าที่ขา 1
Voltage -	-15 V	เข้าที่ขา 4
Analog INPUT	+/- 5 V	เข้าที่ขา 3
Sampling Signal	0 - +5V	เข้าที่ขา 8
Hold Capacitor	0.01 uF	เข้าที่ขา 6 และ ขา 7
Gnd		เข้าที่ขา 7

คุณสมบัติของ LF398

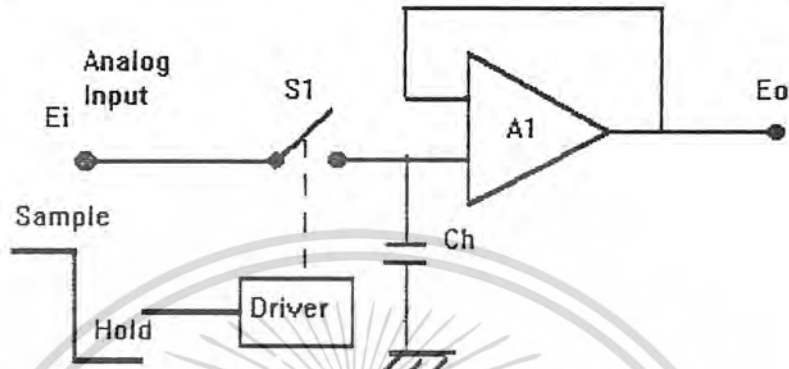
0.002% Gain accuracy

Low output noise in hold mode

Less than 10 uSec acquisition time

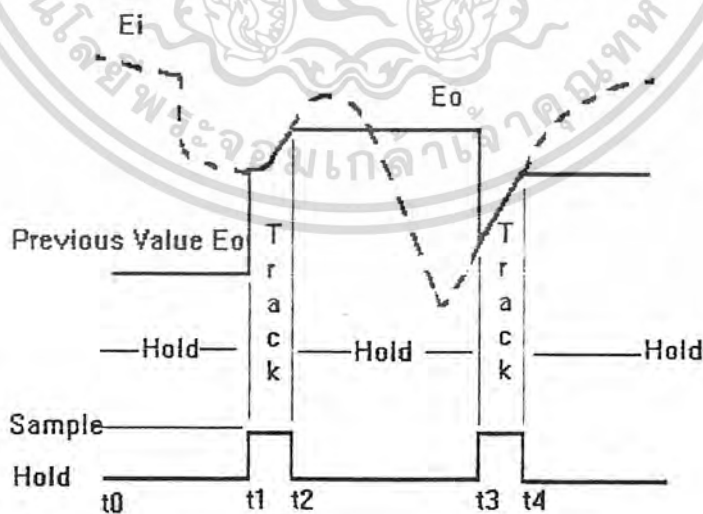
วงจรสุ่มค่าสัญญาณและคงค่าไว้เป็นวงจรที่ประยุกต์มาจากวงจรของอนาล็อกสวิตช์ และวงจรคงค่า วงจรนี้จะสุ่มตัวอย่างสัญญาณอุปมาน(Analog signal) ข้อมูลเข้า(input)ที่เป็นความต่างศักย์และนำมาคง

ค่าไว้ที่จุดสัญญาณออก(output) อุปกรณ์นี้มิใช้ในการแปลงข้อมูล เนื่องจากวงจรแปลงสัญญาณเชิงอนุพันธ์เป็นเชิงเลข(ADC)ไม่สามารถยอมรับการเปลี่ยนแปลงมาก ๆ ของสัญญาณอนุพันธ์ (Analog) ที่เข้าไปได้ในเวลาที่กำลังทำการแปลงข้อมูล (Conversion Time) ดังนั้นการแก้ปัญหาจึงต้องใช้วงจรสุ่มและคงค่าไว้ให้คงที่ภายในช่วงเวลาที่กำลังแปลงข้อมูล การทำงานของ วงจรดังกล่าวเป็นดังรูปที่ 3.3 และ 3.4



รูปที่ 3.3 แสดงวงจรอย่างง่ายของวงจรสุ่มสัญญาณ

การทำงานของวงจรสุ่มค่าและคงค่าดังในรูปที่ 3.4 เส้นประแสดงถึงสัญญาณอนุพันธ์ขาเข้า E_i ขณะที่เส้นทึบแสดงสัญญาณที่ขาออกของวงจร E_o จากเวลา t_0 ถึง t_1 ค่าที่เอาท์พุทจะเป็นค่าของระดับสัญญาณที่ทำการสุ่มในช่วงก่อนหน้านี้ แต่ที่เวลา t_1 เส้นสัญญาณสุ่มเป็นสถานะ HIGH ดังนั้นสวิทช์ S1 ในรูปที่ 3.3 จะปิดและจะจ่ายสัญญาณอนุพันธ์ให้แก่คาปาซิเตอร์ เมื่ออยู่ในช่วงเวลาที่สัญญาณสุ่มเป็นสถานะ HIGH คาปาซิเตอร์จะค่าศักย์ไฟฟ้าที่เอาท์พุทจะ ตาม (Track) สัญญาณอนุพันธ์ที่อินพุท และเมื่อสัญญาณสุ่มเป็นสถานะ LOW อีกครั้ง (ที่เวลา t_2) ค่าศักย์ไฟฟ้าที่เอาท์พุทจะรักษาค่าเดิมของสัญญาณไว้ให้เท่ากับก่อนที่สัญญาณจะเป็น LOW และเมื่อสัญญาณสุ่มเป็น HIGH อีกครั้งการทำงานก็จะวนอยู่ในลักษณะนี้



รูปที่ 3.4 แสดงช่วงเวลาในการทำงานของวงจร Sample and Hold

ปัญหาที่จะต้องคำนึงถึงในระหว่างการสุ่มตัวอย่างคือความผิดพลาดในช่วงเวลาการสุ่มค่า นั่นก็คือช่วงเวลาที่เรียกว่า Settling Time ซึ่งคือเวลาที่ใช้สำหรับเอาท์พุทรักษาค่าสุดท้ายเมื่อสัญญาณสุ่มจ่ายให้ นอกจากนี้ในช่วงที่จะนำค่าตัวอย่างไปเก็บก็มีความผิดพลาดที่อาจเกิดขึ้นได้ในหัวข้อต่อไปนี้ Aperture Time, Switching Transients, Settling time และ sample-to-hold offset. สำหรับหัวข้อดังกล่าวจะไม่ขอกล่าวในรายละเอียด

รูปที่ 3.5 ข้างล่างแสดงวงจรที่ใช้งานจริงของ Sample and Hold



3. วงจรแปลงสัญญาณอนุมาณ เป็นสัญญาณเชิงเลข (Analog to Digital Converter)

เนื่องจากจุดประสงค์ของการทำงานคือต้องการที่จะวิเคราะห์สัญญาณที่มีความละเอียดสูง นั่นคือมีความสามารถแสดง Harmonic ของสัญญาณที่ซ่อนอยู่ได้มากถึง 64 Harmonics และต้องการให้ได้ความถูกต้องของข้อมูลมาก จึงพิจารณาใช้ IC A/D Converter ที่มีความละเอียด 12-Bit ในการสร้างส่วนนี้ และเนื่องจากสัญญาณที่เราพิจารณาเป็นสัญญาณที่มีความถี่ไม่สูงมากนักเราจึงใช้ Analog to Digital Converter ที่มีความเร็วในการแปลงข้อมูลไม่สูงมากนักและเนื่องจากในประเทศไทยมี IC ADC ที่มีจำนวน Bit Resolution ที่สูงขายอยู่เพียงเบอร์ ADC1210 ที่สามารถหาซื้อได้ง่ายที่สุด นอกจากนี้ IC ADC ที่มีจำนวนบิตมาก ๆ และมีความเร็วสูง ก็มีราคาที่สูงมาก

คุณสมบัติของ IC ADC 1210 12-Bit CMOS A/D Converters

เป็นวงจร Analog to Digital ที่สิ้นเปลืองพลังงานต่ำ, มีความเร็วปานกลาง, เป็น Analog to Digital 12บิตแบบที่เรียกว่า "Successive Approximation." ภายในประกอบด้วยวงจรทางตรรกะของ Successive Approximation, CMOS Analog Switch, ความต้านทานละเอียดสูงแบบ Laser Trimmed Film R-2R คู่กันแบบ Ladder เป็นวงจร Digital to Analog Converter ภายใน และ FET Input Comparator.

- เป็น ADC 12 Bit Resolution and 12 bit Accuracy
- 100 uSec 12 bit , 30 uSec 10 bit Conversion Rate

- CMOS Compatible Output
- สามารถใช้ Clock ที่ความถี่ 500 kHz สูงสุด

หลักการการทำงานของ ADC เนื่องจากเป็นแบบ Successive Approximation การแปลงข้อมูลจากสัญญาณเชิงอนุมาณเป็นสัญญาณเชิงเลขนั้นเป็นการแปลงข้อมูลหนึ่งบิตซึ่งเกิดจากการที่เปรียบเทียบสัญญาณที่ขาสัญญาณออกของ DAC(ภายในADC1210) กับสัญญาณที่ไม่ทราบค่าที่ขาสัญญาณเข้าของADCเมื่อขาสัญญาณ START เปลี่ยนสถานะเป็น LOW จะทำให้รีจิสเตอร์ที่เก็บข้อมูลบิตต่าง ๆ ภายในทำการ Reset ในขณะที่เดียวกับที่สัญญาณCLOCK เปลี่ยนจาก LOWเป็นHIGH MSB Q11 ถูกเซตเป็นสถานะ LOW และบิตอื่น ๆ จะถูกเซตเป็นสถานะ HIGH และรีจิสเตอร์จะคงค่าอยู่ในสถานะนี้จนกระทั่งสัญญาณขาให้เริ่มการแปลง SC(Start Conversion) เปลี่ยนเป็นสถานะ HIGH ซึ่งเมื่อสัญญาณ START เปลี่ยนเป็นสถานะ HIGH จากนั้นการแปลงข้อมูลก็จะเริ่มเมื่อสถานะของ CLOCK เปลี่ยนจากสถานะ LOW เป็น HIGH Q11 จะทำการเก็บค่าของการเปรียบเทียบสัญญาณของDigital to Analog กับ สัญญาณที่ไม่ทราบภายนอก จากขาสัญญาณออกของตัว Comparatorที่ขา 23 ซึ่งเป็น Comparator Output ถ้าเป็น HIGH Q11 ก็จะเป็น HIGH ด้วย ถ้าเป็น LOW Q11 ก็ยังคงค่าเดิมไว้และในขณะที่เดียวกันบิตถัดไป Q10 จะถูกเซตเป็น LOW ส่วนบิตอื่น ๆ จะยังคงไม่เปลี่ยนเป็นHIGHขั้นตอนดังกล่าวนี้จะดำเนินต่อเนื่องไปเรื่อย ๆ จนกระทั่งถึงบิตสุดท้าย LSB Q0 เมื่อการแปลงเสร็จสิ้นก็จะทำให้ขาสัญญาณ Conversion Complete (CC) เปลี่ยนสถานะเป็น LOW เสร็จสิ้นข้อมูลหนึ่งชุดก็จะใช้สัญญาณนาฬิกาจำนวน 13ลูก โดย ลูก ต่อการเปรียบเทียบ 1 บิตและอีก 1 บิตสำหรับการเริ่มการทำงาน

ข้อมูลที่ได้ออกจากสัญญาณออก Q11-Q0 เป็นค่าสัญญาณเชิงเลข(Digital Data) ที่เป็นเลขฐานสองแบบส่วนเติมเต็ม(Complement) แทนค่าของสัญญาณเชิงอนุมาณ(Analog Signal)โดย Q11เป็นข้อมูลบิตนัยสำคัญสูงสุด(MSB) และ Q0 แทนบิตนัยสำคัญต่ำสุด(LSB) และค่าของข้อมูลดังกล่าวจะคงค่าอยู่ในสถานะดังกล่าวจนกระทั่ง สัญญาณที่ขา Start Conversion เป็น LOW อีกครั้งหนึ่ง ADC จึงจะเริ่มทำการแปลงค่าอีกครั้ง

วงจรที่ต้องนำมาประกอบการทำงานของ ADC ก็คือวงจรที่ให้ค่าไฟเลี้ยงแก่อุปกรณ์ ADC และให้ค่าศักดาอ้างอิง(Reference Voltage) วงจรดังกล่าวมีความสำคัญมากดังจะพิจารณาในหัวข้อที่ 5 ต่อไป

จากวงจร ADC ที่เราทำการต่อจะเป็นแบบ Complementary Logic ที่มีอินพุตอยู่ในช่วงของ +5.18 V ถึงประมาณ -4.99 V (จากการปรับ Offset) ทำให้ได้สัญญาณ Logic 1 ที่ค่าประมาณ 0.5 V และได้สัญญาณ Logic 0 ที่ค่าประมาณ 10 V ที่เอาท์พุทบิต นอกจากนี้ในการต่อเอาท์พุทADC กับ IC จำพวก TTL จำเป็นจะต้องทำการแปลงระดับสัญญาณที่เป็นเนื่องจากสัญญาณไฟที่ระดับ Logic 0 มีค่าถึง 10 V ดังนั้นในที่นี้จะใช้ IC เบอร์ 74C914 ที่เป็น Output Hex Buffer Inverter ที่มีคุณสมบัติคล้ายกับ IC ตระกูล CMOS ที่สามารถจ่ายไฟเลี้ยงได้หลายค่าในช่วง 5 ถึง 10 V ซึ่งสามารถใช้ไฟเลี้ยง 5 V และ

จ่ายเอาต์พุตให้กับ IC ทั้ง TTL และ CMOS ได้

ส่วนที่สำคัญอีกส่วนก็คือสัญญาณนาฬิกาของ ADC ที่ใช้เป็นค่าที่ใช้ในการทำงานเนื่องจาก ADC 1210 นี้เราสามารถให้สัญญาณได้สูงสุดถึง 500 kHz แต่ในที่นี้เราจะใช้ค่าที่ไม่สูงมากนัก ในการทำงานเนื่องจากสัญญาณที่เราสนใจไม่มีความถี่สูงมากนักนั่นคือ 50 Hz และเพื่อที่จะกำหนดค่าของสัญญาณนาฬิกา จึงต้องพิจารณาถึงช่วงเวลาที่ทำการสุ่มตัวอย่างจากที่เราทราบทฤษฎีการสุ่มตัวอย่าง (Sampling Theory) ของ แซนนอน (Shannon) ที่กล่าวไว้ว่าหากมีสัญญาณเชิงอนุพันธ์ที่มีแถบความถี่ปฏิบัติงานเป็น f_0 แล้ว จะสามารถทำการสุ่มตัวอย่าง โดยที่สัญญาณที่ได้ไม่สูญเสียข้อมูลที่สำคัญไปก็คือเมื่อความถี่ในการสุ่มตัวอย่าง f_s มีค่ามากกว่าหรือ เท่ากับสองเท่าของความถี่ f_0 หรือ $f_s \geq 2f_0$ โดยที่ค่าความถี่ f_s ที่เท่ากับ $2f_0$ นี้เรียกว่าความถี่ในควิสต์ (Nyquist frequency) ดังนั้นในโครงงานนี้จะสามารถคิดความถี่ในควิสต์ได้เท่ากับ $50 * 2 = 100$ Hz แต่เราจะต้องทำการหาข้อมูลที่มีถึง 64 Harmonics ที่ซ่อนอยู่ ดังนั้นจะได้ว่าความถี่ที่เราจะใช้คือ $100 * 64 = 6.4$ kHz หรือคือช่วงเวลาเท่ากับ 156.25 μ Sec

สัญญาณดังกล่าวจะจ่ายให้กับวงจรสุ่มตัวอย่างและคงค่า (Sample and Hold) และเป็นตัวบอกช่วงเวลาที่จะใช้ได้มากที่สุดในการใช้เวลาแปลงข้อมูลอย่างคร่าว ๆ (หา Conversion Time) และจากที่ได้ทราบแล้วว่าข้อมูลหนึ่งค่าต้องใช้สัญญาณนาฬิกา 13 ลูก โดยให้สัญญาณนาฬิกาที่ความถี่เท่ากับ 125 kHz หรือ 8 μ Sec ดังนั้นเราจะใช้เวลาในการแปลงข้อมูลเท่ากับ $8 \mu\text{Sec} * 13 = 104 \mu\text{Sec}$

ในการเชื่อมต่อสัญญาณนาฬิกาและสัญญาณให้เริ่มการแปลงข้อมูล (Start conversion) ซึ่งมาจากสัญญาณเอาต์พุตของ IC ทรานซิสเตอร์ TTL ซึ่งมีค่าของสัญญาณเอาต์พุตน้อยคือที่ logic 1 เป็นค่าประมาณ 2.7 V ซึ่งไม่สามารถจะต่อกับวงจร ADC นี้ได้เราจึงทำการยกระดับสัญญาณที่จะเข้า ADC ด้วยการต่อ Inverter และ Transistor เพื่อทำการยกระดับศักย์ไฟฟ้าให้สูงขึ้น

สำหรับตารางข้างล่างนี้เป็นข้อมูลของขาสัญญาณของ ADC 1210

ขาเลขที่	สัญญาณ	หน้าที่
1-12	สัญญาณ Q11-Q0	ข้อมูลที่เชิงเลขที่ออกมาเป็นข้อมูลแบบขนาน 12 bit เลขฐานสองแบบส่วนเติมเต็ม แทนสัญญาณเชิงอนุพันธ์ที่ให้เข้าไป ข้อมูลจะมาเมื่อสัญญาณ Conversion Complete LOW
13	SC	ระดับสัญญาณที่ GND และ +Vref สัญญาณเริ่มการแปลงเป็นสัญญาณเอาต์พุต ระดับสัญญาณที่ GND และ +Vref
14	CC	สัญญาณ Conversion Complete เป็นสัญญาณแสดงถึงสถานะการทำงานเมื่อเป็น HIGH คือกำลังทำการแปลงข้อมูล แต่ LOW แสดงถึง

		การแปลงเสร็จสมบูรณ์
		ระดับสัญญาณที่ GND และ +Vref
18,19	AnalogIN	เป็นขาสัญญาณเชิงอุปมานที่เป็นอินพุท
20	V-	แหล่งจ่ายไฟเลี้ยงค่าลบสำหรับ Analog Comparator
		ระดับสัญญาณที่ GND หรือ -20 V
21	GND	สัญญาณ Ground ทั้ง Analog และ Digital
22	V+(Vref)	แหล่งจ่ายไฟเลี้ยงค่าบวกและเป็นระดับสัญญาณที่สูงที่สุดสำหรับ Logic อินพุทและเอาต์พุท
23	CO	สัญญาณ Comparator Output
24	Cp	สัญญาณนาฬิกา ที่ทำให้การทำงานแบบ Successive Approximation ระดับสัญญาณที่ GND และ V+

ความผิดพลาดที่เกิดขึ้นใน Analog to Digital Converter

Quantization Uncertainty เป็นผลมาจากความละเอียดในการแปลงข้อมูลซึ่งสัญญาณเชิงอุปมานของความต่างศักย์ (Analog Voltage) จะถูกกำหนดให้แทนด้วยค่าเชิงเลข (Digital) ค่าหนึ่ง ๆ ที่มันจะสามารถเกิดความผิดพลาดได้แม้ในตัวอุปกรณ์ ADC ที่ดีที่สุด นั่นก็คือ ในการที่ค่าศักดาขาเข้า ตกไปอยู่ในช่วงระหว่างค่าของค่าที่ถูกกำหนดที่ขาสัญญาณออก (Output Code) ที่กำหนด Voltage Input ไว้ 2 ค่า เช่น V_{in} เป็น 0.5 V ส่วนค่า Output Code นั้นได้กำหนดไว้ให้กับค่า 1 V กับ 2 V นั่นคือมีค่าไม่แน่นอนของ $\pm 1/2$ LSB การแก้ไขปัญหของค่าไม่แน่นอนของสัญญาณดังกล่าวก็ทำได้โดยการเพิ่มจำนวนความละเอียดของบิต (Bit Resolution) ให้แทนค่าสัญญาณ Analog Input ได้มากขึ้น

Linearity Error คือค่าเบี่ยงเบนมากที่สุดจากเส้นตรงซึ่งผ่านจุดท้ายสุดของ ADC Transfer Characteristic ซึ่งมันจะสามารถวัดได้จากการปรับ Zero และ Full Scale Error สำหรับ Linearity นี้เป็นคุณสมบัติภายในของอุปกรณ์ ไม่สามารถปรับได้จากภายนอก

Zero Scale Error (or Offset) คือการวัดของความแตกต่างระหว่างสัญญาณออกทางทฤษฎีกับทางปฏิบัติของ ADC เมื่อความต่างศักดาสัญญาณเข้าเป็น ศูนย์ (Zero) จะทำให้เกิดการเลื่อนของ กราฟคุณสมบัติ Transfer Characteristic ไปทางซ้ายหรือขวาตามค่า Absissa นั่นคือค่าศักดาในความเป็นจริงจะผิดพลาดไป

Full Scale Error (or Gain Error) คือการวัดของความแตกต่างระหว่างสัญญาณขาออกของADC ทางทฤษฎีกับทางปฏิบัติสำหรับค่าความต่างศักย์ที่อินพุตเป็นค่าเต็ม (Full Scale)

4. วงจรกำเนิดสัญญาณความถี่ในการสุ่ม (Sampling Frequency Base Clock)

วงจรที่ให้กำเนิดสัญญาณนาฬิกาในที่นี้เราใช้วงจร Crystal-Controlled Oscillator ในการสร้างสัญญาณนาฬิกาที่มีความถี่ 4 MHz สำหรับเป็นสัญญาณนาฬิกาการทำงานของวงจร ADC ส่วนสัญญาณที่มีความถี่ 4.9152 MHz ใช้เป็นฐานเวลาให้กับความถี่ในการสุ่มตัวอย่าง

รูปที่ 3.6 ข้างล่างแสดงวงจรของ Crystal-Controlled Oscillator



สำหรับสัญญาณนาฬิกาของวงจร ADC ได้กำหนดให้ใช้ที่ความถี่ 500 kHz ดังนั้นจึงต้องทำการหารความถี่ที่ 4 MHz ลงไปด้วยค่า 8 เท่า ในที่นี้จะใช้ IC เบอร์ 74LS93 ในการหารความถี่ต่อจากนั้น เพราะว่าสัญญาณที่ได้ยังมีค่าระดับสัญญาณแค่ +5 V กับ GND จึงจำเป็นต้องทำการเพิ่มระดับสัญญาณขึ้นไปให้เท่ากับ +10 V กับ GND เพื่อที่จะทำให่วงจร ADC ทำงานได้อย่างถูกต้อง

ในส่วนของสัญญาณความถี่ในการสุ่มตัวอย่าง เนื่องจากเราได้ทำการหาค่าในการสุ่มตัวอย่างไว้เท่ากับ 6.4 kHz จากฐานเวลาดังกล่าวเราทำการหารด้วยจำนวน 768 เท่า ในที่นี้เราใช้วงจรของ Counter ที่สามารถหารได้ด้วยเลข 12 จำนวนหนึ่งตัว และ Counter ที่หารได้ด้วยเลข 8 จำนวน 2 ตัวมาต่อกัน ได้เป็น $12 * 8 * 8 = 768$ ดังนั้นในที่สุดเมื่อเอาสัญญาณฐานเวลาที่ความถี่ 4.9152 MHz มาเข้าวงจรหารความถี่ดังกล่าวก็จะได้ความถี่ในการสุ่มตัวอย่างที่ต้องการ ความถี่ดังกล่าวนี้จะถูกนำไปใช้ดังนี้

1. เป็นความถี่ที่จะถูกนำไปสร้างให้เป็นสัญญาณ Mono Stable เพื่อใช้กับวงจร Sample and Hold
2. เป็นความถี่ที่จะถูกนำไปทำการ Invert ด้วยตัว Inverter Buffer และทำการปรับระดับสัญญาณให้เป็นที่ระดับ +10 V กับ GND เพื่อเป็นสัญญาณ Start Conversion (SC) ให้กับ ADC

สำหรับ IC ที่ใช้ในการหารความถี่เราใช้เบอร์ 74LS92 (divided by 12) และ 74LS93 (divided by 16)

5. วงจรรักษาระดับแรงดันอ้างอิง (Voltage Reference)

เนื่องจากการใช้งาน วงจรแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลขต้องมีระดับแรงดันอ้างอิงที่มีคุณภาพดีเพราะหากมีปัญหาอาจทำให้การแปลงข้อมูลผิดพลาดได้ เนื่องจากว่าค่าศักดาที่ได้จะต้องมีค่าคงที่มากคือมีค่าเปลี่ยนแปลงน้อยมากต่อปัจจัยต่างๆเช่น การเปลี่ยนแปลงของอุณหภูมิในการจัดหาอุปกรณ์เพื่อให้ได้ความแม่นยำและถูกต้องจริง ๆ เราจะต้องใช้ ZENER ที่มีการชดเชยอุณหภูมิ (Temperature Compensate) และควรจะใช้ LOWDRIFT OPAMP แต่ในที่นี้เราใช้แค่ ZENER Diode ธรรมดา 6.8 V , OPAMP 741 และความต้านทานที่มีค่าเปอร์เซ็นต์ความผิดพลาดเป็น 1 % จากวงจรดังกล่าวทำให้ได้ค่าศักดาอ้างอิงเป็นค่าในช่วง 9.96 V ถึง 10.00 V

รูปที่ 3.7 วงจรข้างล่างแสดงวงจรที่ทำหน้าที่เป็น Reference Voltage



6. วงจรสร้างสัญญาณโมโนสเตเบิล มัลติไวเบเรเตอร์ (Monostable Multivibrator)

วงจรมัลติไวเบเรเตอร์ทำหน้าที่จ่ายสัญญาณพัลส์หนึ่ง ลูกคลื่นให้กับวงจรภายนอกเมื่อมีสัญญาณทริกซ์เข้ามาที่ขาสัญญาณเข้าของวงจร จากการทำงานดังกล่าวเราจำเป็นต้องใช้งานสัญญาณในลักษณะดังกล่าวเพื่อจุดประสงค์ดังนี้

1. ทำหน้าที่สร้างสัญญาณเริ่มการทำงานของวงจรแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข(ADC) ในที่นี้ก็เพื่อสร้างสัญญาณการเริ่มทำการแปลงให้ตรงกันกับข้อมูลที่เข้ามาเพราะจำเป็นต้องใช้สัญญาณเดียวกันการจ่ายให้วงจรส่วนตัวอย่างและคงค่า

2. ทำหน้าที่สร้างสัญญาณเพื่อควบคุมการทำงานของวงจร Dฟลิปฟล็อป(Latch)คงค่าสัญญาณที่ออกมาจากตัวแปลงสัญญาณ(ADC)เมื่อเวลาที่มีการแปลงข้อมูลเสร็จสมบูรณ์

ความกว้างของสัญญาณ โมโนสเตเบิล ก็มีผลต่อการทำงานของวงจรเช่นเดียวกัน โดยเฉพาะเมื่อ

เวลาที่ สัญญาณนาฬิกาของตัวแปลงสัญญาณ(ADC)มีความเร็วสูง ๆ ความกว้างของพัลส์ในการส่งสัญญาณเริ่มการทำงานก็จะต้องมีความแคบเพียงพอด้วย(หรือมีความถี่สูงพอ)

หมายเหตุ การสร้างสัญญาณ Mono Stable เราใช้ IC เบอร์ 4528 ในการสร้างสัญญาณและที่เราใช้ตามตารางการทำงานเราใช้ที่ค่าความต้านทาน 1 Kohms และ ตัวเก็บประจุ 1000pF

7. วงจรหน่วยความจำชั่วคราว (Static Random Access Memory)

ในระบบการเก็บข้อมูลเราได้กำหนดให้มีขนาดของข้อมูลเป็น 12 บิตตั้งนั้นจากการออกแบบจึงจำเป็นต้องใช้หน่วยความจำขนาด 8 บิต 2 ชุดมาต่อขนาดกันให้สามารถทำการเก็บข้อมูลได้โดย ในเครื่องมีนี้เราได้ใช้ Static RAM เบอร์ 6264 ในการเก็บข้อมูลได้ ข้อมูลสูงสุด 8K ตำแหน่ง ตำแหน่งละ 12 บิต นอกจากนี้หน่วยความจำชั่วคราว RAM ในอุปกรณ์ดังกล่าวได้ทำการต่อวงจรเพิ่มเพื่อทำให้สามารถเก็บข้อมูลที่ถูกรบกวนอย่างไว้ได้โดยมีการต่อแหล่งจ่ายไฟสำรอง Battery Backup ไว้ด้วยโดยการควบคุมขา CE1 ของวงจร Static RAM ไม่ให้เป็นค่าทางตรรกเป็น 0 พิจารณาการทำงานของ SRAM 6264 ดังนี้

CE1	CE2	OE	WE	A0- A12	Mode	Data I/O	Device current
H	X	X	X	X	N/S	Z	STDB
X	L	X	X	X	N/S	Z	STDB
L	H	L	H	STA	RD	D/O	ACTV
L	H	X	L	STA	WR	D/I	ACTV
L	H	H	H	STA	O/P/D	Z	ACTV

หมายเหตุ N/S = Not Select, Z = High impedance, STDB = Stand by, ACTV = Active, STA = Stable, O/P/D = Output disable

8. การทำงานของระบบควบคุมการเก็บข้อมูล (Microcontroller)

เครื่องวิเคราะห์สัญญาณฮาร์โมนิกส์ดังกล่าวมีระบบควบคุมการเก็บข้อมูลคือไมโครคอนโทรลเลอร์บอร์ด เบอร์ 8031 ซึ่งได้ถูกออกแบบให้ ทำหน้าที่หลัก ๆ ดังต่อไปนี้

- 8.1 ทำหน้าที่ควบคุมการเก็บข้อมูล
- 8.2 ทำหน้าที่อ่านข้อมูลและส่งข้อมูลผ่านทางพอร์ทอนุกรม
- 8.3 ทำหน้าที่ตรวจสอบเวลาจริง

รายละเอียดของระบบไมโครคอนโทรลเลอร์ 8031

การควบคุมผ่านไมโครคอนโทรลเลอร์ซึ่งเป็นอุปกรณ์ทางไมโครคอมพิวเตอร์ ที่มีความสามารถสูงและมีความนิยมสูงมาก เนื่องจากมีความสามารถและความคล่องตัวในการทำงานสูงในโครงการนี้เราใช้ไมโครคอนโทรลเลอร์เบอร์ 8031 ในตระกูลเดียวกันกับ MCS-51 โดยใช้ร่วมกับบอร์ดขยายพอร์ท 8255 และ

บอร์ดขยาย RTC ซึ่งมีหน่วยความจำสำรอง(RAM Backup)ในการเก็บข้อมูล

คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์ V-31

CPU 80C31 (40pin-dip)

CLOCK 11.0592 MHz

MEMORY 8 KB 27C64

PORT 8 BITS (PORT1)

4 BITS (INT1,INT0,T1,T0)

1 SERIAL PORT (STANDART RS232)

CONNECTOR 16 PINS PORT1 BUS (WITH /INT1,/INT0/T1,/T0)

40 PINS MCS-51 SYSTEM BUS

3 PINS RS-232

2 PINS 5V DC

คุณสมบัติของบอร์ดขยายพอร์ต 8255

PORT INPUT/OUTPUT 8255

CONNECTOR 26 PINS I/O 24 BITS

คุณสมบัติของบอร์ดขยาย RTC และ หน่วยความจำสำรอง

RTC REAL TIME CLOCK 6264

RAM 6264

MEMORY BACKUP MAX691

BATTERY BACKUP 3V LITHIUM CR2016

การควบคุมทั้งหมดของไมโครคอนโทรลเลอร์กระทำผ่านพอร์ตขยาย 8255 และพอร์ตหมายเลขที่ 1 และ บางส่วนของพอร์ตหมายเลขที่ 3 คือไปนี้จะเป็นการแสดงหน้าที่ของขาควบคุมต่าง ๆ

พอร์ตขยาย 8255 พอร์ต A ทำหน้าที่รับข้อมูล

พอร์ตขยาย 8255 พอร์ต C บน ทำหน้าที่รับข้อมูล

พอร์ตขยาย 8255 พอร์ต B ทำหน้าที่กำหนดตำแหน่งข้อมูลในหน่วยความจำชั่วคราว

พอร์ตขยาย 8255 พอร์ต C ล่าง ทำหน้าที่เช่นเดียวกับพอร์ต B

พอร์ตหมายเลขหนึ่งและหมายเลขสามบางบิตควบคุมการทำงานของบอร์ดเก็บข้อมูล

PORT1	Port1.0	Read Write control
	Port1.1	Chip Select RAM
	Port1.2	Decode Address RAM Bit 12

Port1.3	Check Voltage/Current
Port1.4	Switch Status
Port1.5	Output Latch Control
Port1.6	Switch Status
Port1.7	Select data 128 points
INT0	Conversion Complete
T0	Select data 256 points
T1	Select data 512 points

อธิบายการทำงานของหน้าที่ต่าง ๆ

8.1 หน้าที่ควบคุมการเก็บข้อมูล เป็นตัวจัดการกับตำแหน่งของข้อมูลที่มีขนาด 12 บิตที่ได้จากการแปลงสัญญาณเชิง อุปมานเป็นเชิงเลข(ADC) โดยทำการรับสัญญาณ การแปลงข้อมูลเสร็จ (Conversion Complete) จากตัวแปลงสัญญาณในการแปลงข้อมูลแต่ละ12บิต เข้าทางขาอินเทอร์พท์0 (INT0) จากนั้นตัวไมโครคอนโทรลเลอร์ก็จะทำการนับเป็นตำแหน่งของข้อมูลที่หน่วยความจำต่อการอินเทอร์พท์หนึ่งครั้งและ ในหนึ่งครั้งนั้นจะส่งผลการ นับออกเป็นตำแหน่งของข้อมูลบนหน่วยความจำชั่วคราวผ่านทางพอร์ทขยาย 8255

ในขั้นตอนการเก็บข้อมูลก็ต้องเริ่มต้นด้วยการทราบจำนวนจุดที่จะต้องเก็บในหนึ่งครั้ง วันเวลาที่ทำการเก็บข้อมูล เมื่อทราบรายละเอียดต่าง ๆ ดังนี้แล้ว ก็ทำการอ่านสัญญาณจากวงจร บัลลิสติก เซอร์ที่ตัวตรวจจับว่าข้อมูลที่จะเก็บจุดแรกจะต้องเป็นความต่างศักย์ไฟฟ้าของสัญญาณตัวอย่างจุดต่อไป เป็นค่าของกระแส แล้วจากนั้นก็เริ่มทำการเก็บข้อมูลจนครบตำแหน่งจำนวนจุดที่ต้องการ และทำการบันทึกลงตำแหน่งของหน่วยความจำชั่วคราวไว้ในหน่วยความจำสำรองของบอร์ดขยาย

ในการเก็บครั้งต่อ ๆ ไปก็ทำการเก็บข้อมูลต่อเนื่องไปกับตำแหน่งสุดท้ายที่เคยทำการเก็บข้อมูล จนกว่าจะมีคำสั่งให้เคลียร์หน่วยความจำ

8.2 หน้าที่อ่านข้อมูลและส่งข้อมูลทางพอร์ทอนุกรม เมื่อมีคำสั่งให้ทำการเลือกอ่านข้อมูลเพื่อนำไปประมวลผลบนเครื่องไมโครคอมพิวเตอร์ ระบบไมโครคอนโทรลเลอร์จะทำการอ่านค่าของจำนวนจุด วันที่ เวลา ของข้อมูลแต่ละจุดก่อนและทำการส่งข้อมูลดังกล่าวออกไปก่อนแล้วจึงค่อยทำการส่งข้อมูลของสัญญาณที่ถูกสุ่มตามไป เมื่อครบหนึ่งจุดก็จะส่งค่าของจำนวนจุด วันที่และเวลาของจุดต่อไปและตามด้วยข้อมูลเป็นเช่นนี้ไปเรื่อย ๆ จนถึงตำแหน่งที่เก็บข้อมูลจุดสุดท้าย

การส่งข้อมูลทางพอร์ทอนุกรมก็จะทำการส่งข้อมูลด้วยความเร็ว 9600 BPS ด้วยขนาดข้อมูล 8 บิต ไม่มีพาริตีบิต และบิตหยุด 1 บิต

8.3 หน้าที่ตรวจสอบเวลาจริง เนื่องจากบอร์ดขยายที่ใช้มีวงจรรวมเบอร์ 6264 ซึ่งทำหน้าที่เป็น

วงจรระบบเวลาจริง (Real Time Clock) ดังนั้นเราจึงสามารถทราบเวลาในการทำงานได้ เมื่อต้องการที่จะเก็บข้อมูล ดังนั้นเราสามารถตั้งให้ทำการเก็บข้อมูลทุก ชั่วโมงได้

9. ผังแสดงการทำงานของสัญญาณนาฬิกาและสัญญาณความถี่ในการสุ่มตัวอย่าง

รูปต่อไปนี้จะแสดงสัญญาณการทำงานของวงจรทำการเก็บข้อมูลตัวอย่าง สัญญาณนาฬิกาที่มีความสำคัญต่าง ๆ ในวงจร คือ

1. สัญญาณความถี่พื้นฐานที่เหมาะสม หรือความถี่ Nyquist frequency
2. สัญญาณที่ควบคุมการสุ่มตัวอย่าง เป็นสัญญาณ Mono Stable ให้กับ LF398
3. สัญญาณที่ควบคุมการทำงานของวงจร ADC ให้เริ่มทำการแปลงข้อมูล
4. สัญญาณที่แสดงการแปลงข้อมูลของ ADC ที่สำเร็จเรียบร้อย
5. สัญญาณแสดงการควบคุมการ Latch ค่าข้อมูล



บทที่ 4

โปรแกรมการทำงาน

ส่วนของซอฟต์แวร์ในโครงงานนี้ แบ่งออกเป็น 2 ส่วน ได้แก่

1. โปรแกรมควบคุมการทำงานของระบบเก็บข้อมูลซึ่งควบคุมโดยไมโครคอนโทรลเลอร์
2. โปรแกรมการคำนวณ และแสดงผลของฮาร์โมนิกส์

อธิบายโปรแกรมการคำนวณ และแสดงผลฮาร์โมนิกส์

โปรแกรมคำนวณการแปลงสัญญาณและแสดงผล

โปรแกรมในส่วนนี้เป็นส่วนที่จะติดต่อโดยตรงกับผู้ใช้ดังนั้นจึงเขียนขึ้นให้สามารถนำไปใช้งานได้โดยง่าย เมื่อเรียกโปรแกรมจะมี menu ให้ผู้ใช้เลือกเพื่อที่จะทำงานตามที่คนต้องการ ภาษาที่ใช้เขียนใช้ภาษา C และอยู่ในโหมด กราฟฟิก

ในโปรแกรมหลัก main() จะทำการเรียกหน้าจอซึ่งแสดง menu ให้เลือกเพื่อที่จะเข้าไปยัง function ย่อยๆ และสามารถ switch ไปมาระหว่าง menu และ function ที่ทำงานอยู่ได้ function หลักๆประกอบด้วย

function input-data ซึ่งจะรับข้อมูลที่ได้อจากการ Sampling มาทำการคำนวณ โดยจะเรียกใช้ function off และเลือกใช้ window ที่กำหนดให้ จากนั้นก็จะเก็บผลการแปลงไว้ในชื่อ file ที่ตั้งชื่อให้ตามต้องการแต่เพื่อให้การเก็บข้อมูลมีระบบจึงทำการเก็บแยกสัญญาณคึกคาและกระแสเป็น 2 file โดยให้ file ของสัญญาณคึกคาไฟฟ้ามี่ชื่อเป็น *****.VOL** และ file ของกระแสไฟฟ้ามี่ชื่อเป็น *****.CUR**

function option ใช้ในการกำหนดข้อมูลต่างๆที่ใช้ในการแปลงสัญญาณ เช่น Sampling rate ชนิดของ window range ของกระแสที่ใช้

function graph ใช้แสดงผลของข้อมูลในรูปกราฟซึ่งสามารถเลือกชั่วโมงที่ต้องการนำมาแสดงทีละชั่วโมง หรือจะแสดงทุกชั่วโมงก็ได้

function text ใช้แสดงผลของข้อมูลเป็นตัวเลข ซึ่งจะคำนวณและแสดงค่าของ harmonic distortion ของสัญญาณด้วย

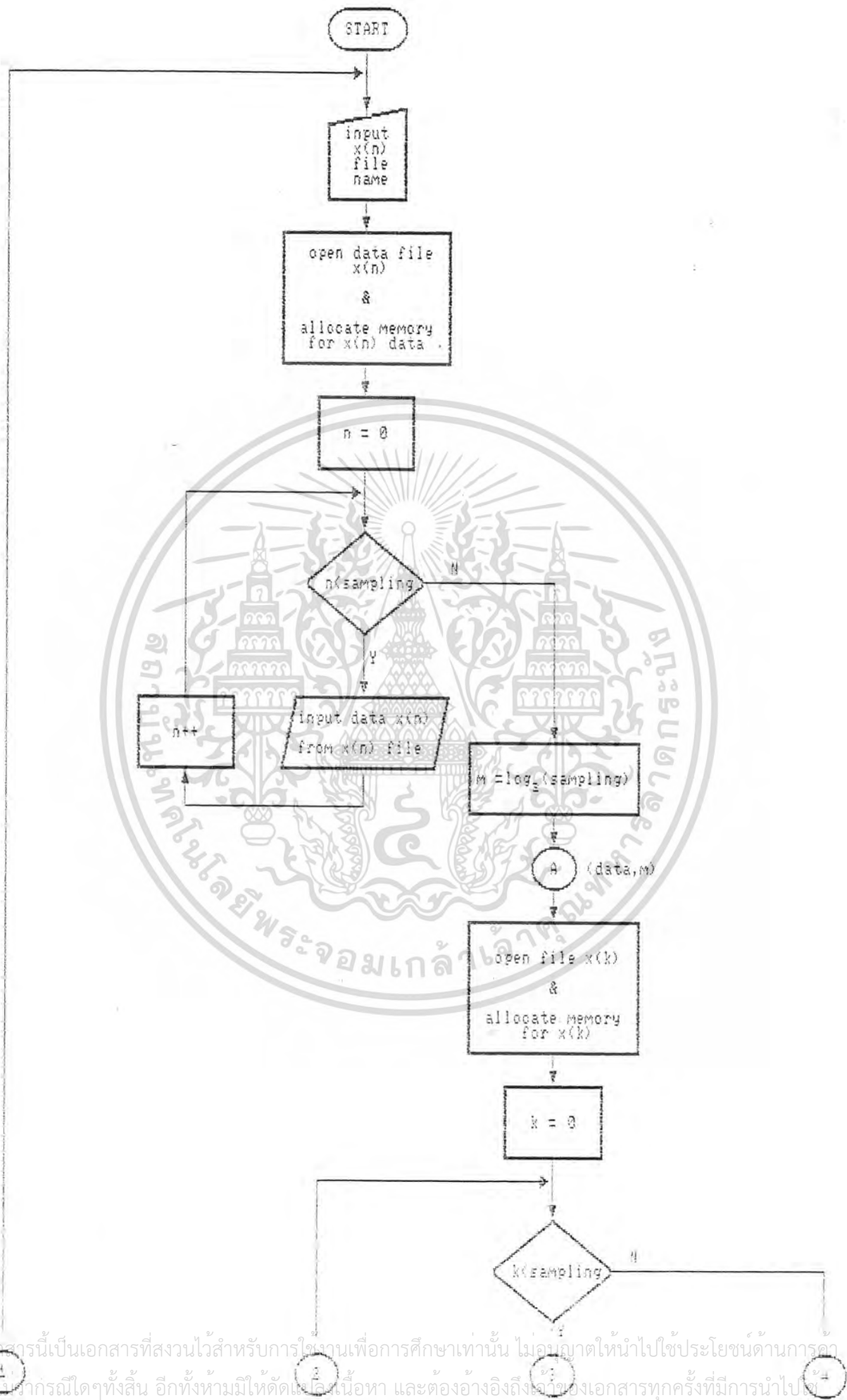
function make_new file ใช้สำหรับกรณีที่ต้องการเปลี่ยนแปลงข้อกำหนดของ file ข้อมูลเดิม และสร้างเป็น file ใหม่

ในส่วนของฟังก์ชัน FFT ใช้ FFT แบบ Decimation in Frequency หรือ DIF ซึ่งจะทำการคูณตัวประกอบหลังจากการบวกหรือลบข้อมูลเข้าและลำดับข้อมูลออกจะถูกสลับตำแหน่งกับแบบผกกลับ บิต การทำงานของฟังก์ชัน FFT เริ่มจากการ หาค่าของตัวประกอบหมุน (W_N^j) ทั้งหมดที่ต้องใช้ใน

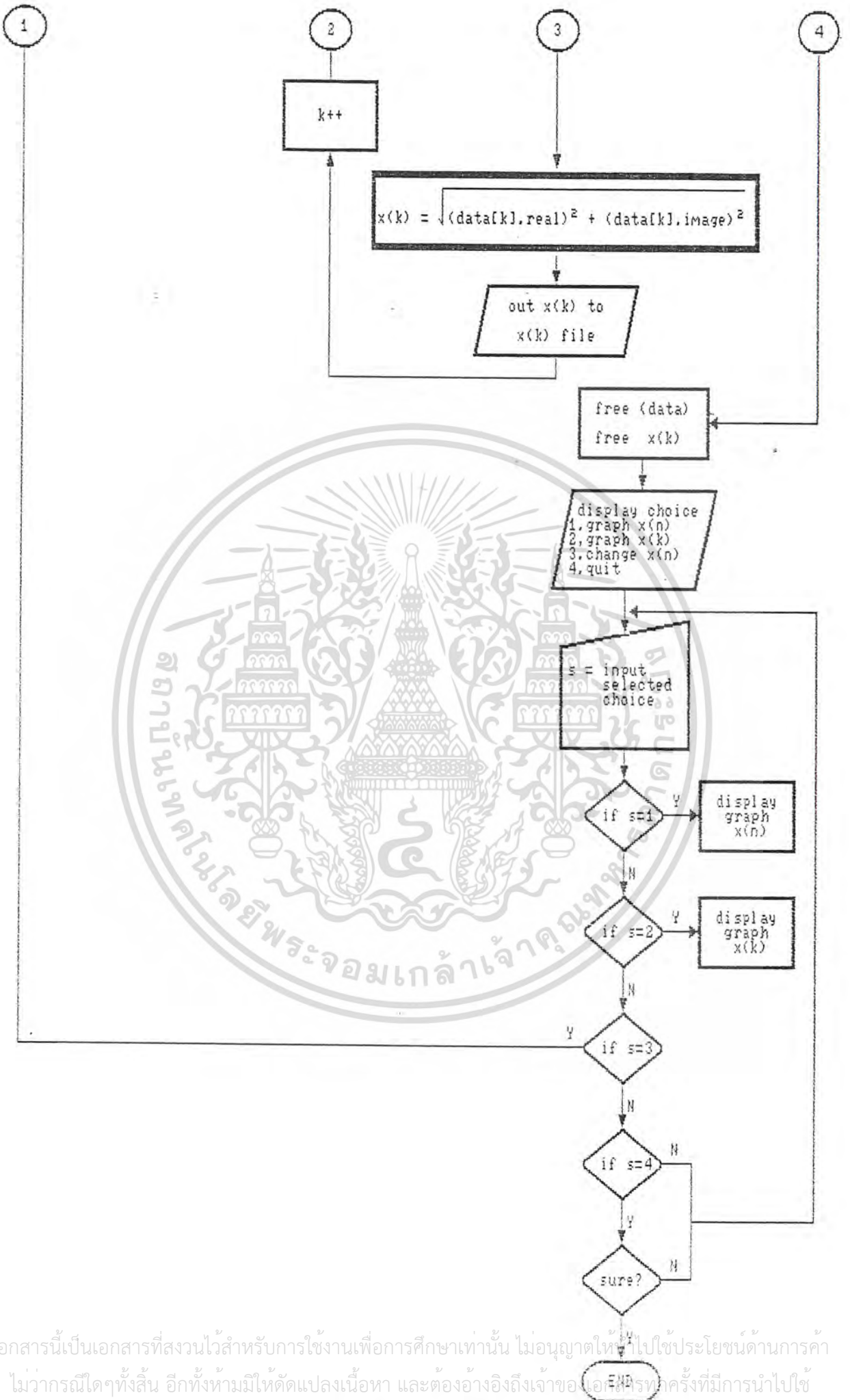
การคำนวณซึ่งจะทำเฉพาะในครั้งแรกของการเรียกใช้โปรแกรมเท่านั้นและจะเก็บค่าไว้สำหรับการคำนวณครั้งต่อ ๆ ไป จากนั้นจะทำการคำนวณหาค่า $X(k)$ ในลำดับที่ผลบวกของ $X(n)$ ไม่ต้องคูณด้วยตัวประกอบหมุนก่อน (ตัวประกอบหมุนเป็น w^0) เพื่อประหยัดเวลา แล้วจึงหาค่า $X(k)$ ในลำดับที่ต้องคูณด้วย ตัวประกอบ หมุนค่าต่างๆ โดยที่ค่า $X(k)$ ที่ได้ยังไม่ใช่ $X(k)$ ที่เป็นผลการแปลง แต่เป็นเพียง ผลการคำนวณ ระหว่าง กลาง ชั้นที่ 1 ซึ่งจะถูกแบ่งย่อยและใช้เป็นลำดับข้อมูลเพื่อ นำไปคำนวณหาผลการแปลงระหว่างกลาง ชั้นต่อไป ด้วยวิธีซ้ำเดิม ซึ่งจะทำการทั้งหมด $\log_2 128 = 7$ รอบแล้ว $X(k)$ ที่ได้ในตอนสุดท้ายก็คือผลการแปลง แต่จะมี ลำดับที่สลับตำแหน่งกัน จึงต้องทำการจัดเรียงใหม่ ซึ่งกระทำในส่วนท้ายของ function

ต่อจากนี้ไปจะเป็นการแสดงโปรแกรมการทำงานที่ใช้ทั้งหมดและ Flowchart ประกอบการทำงาน

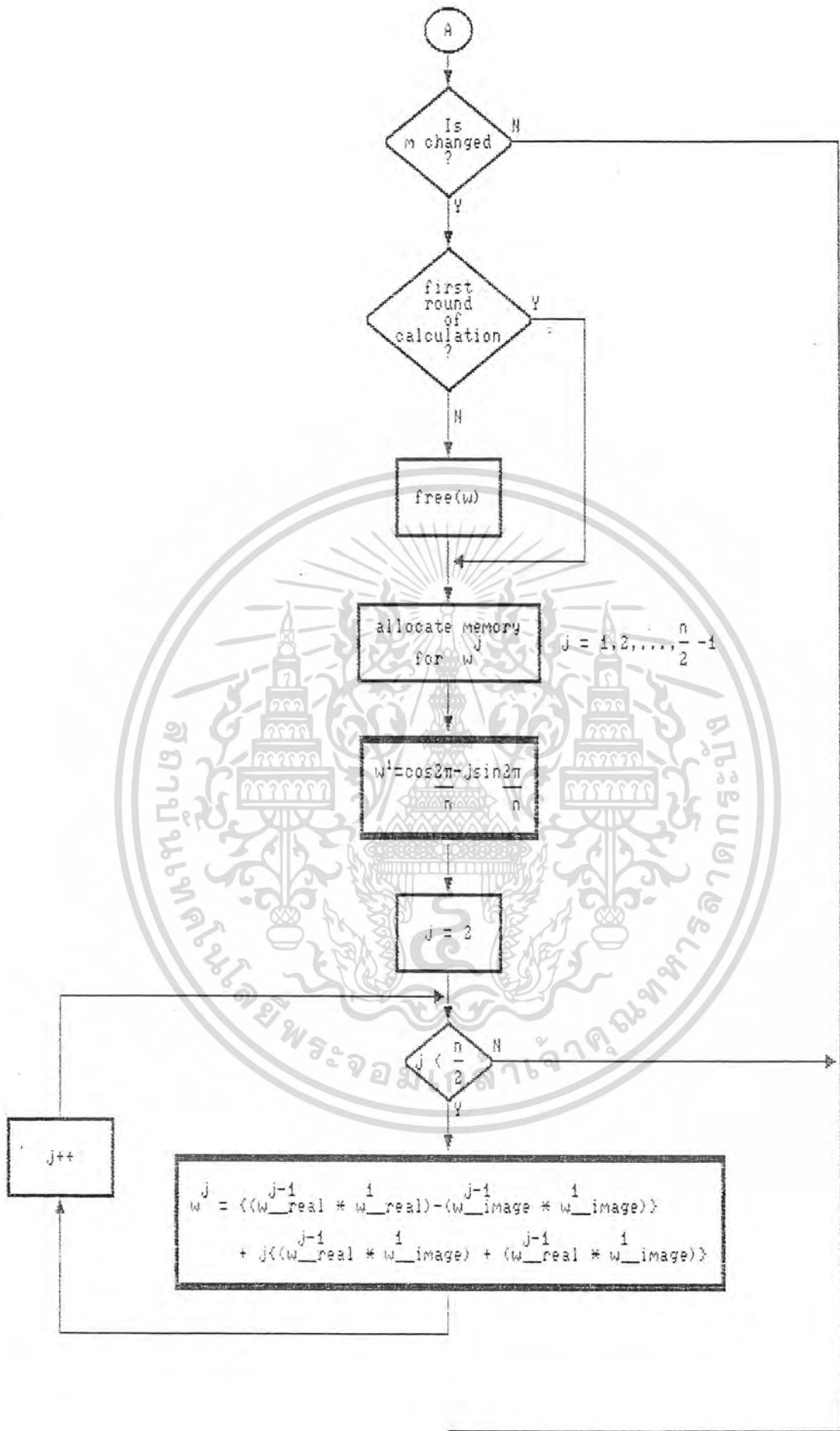




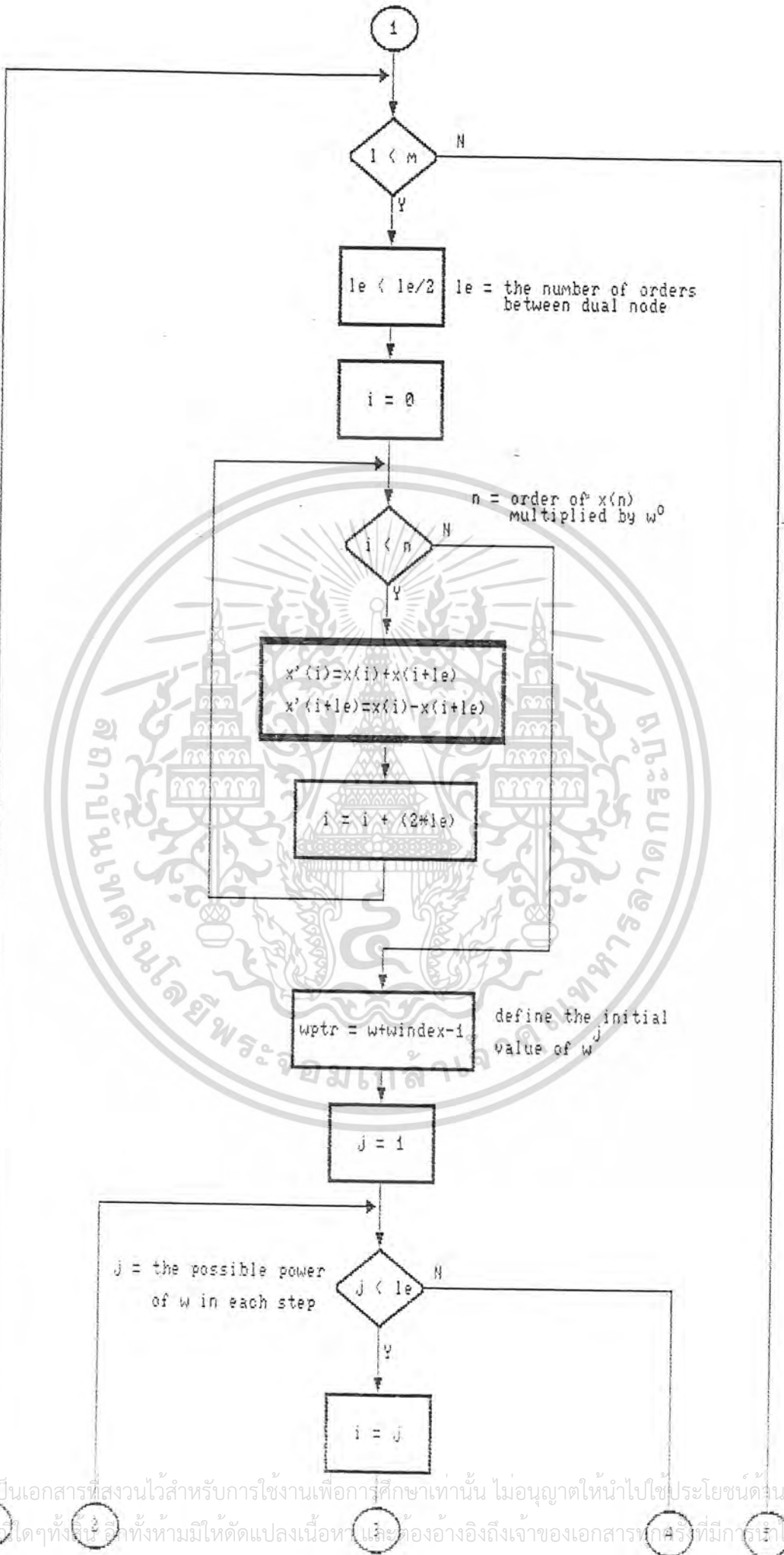
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆทั้งที่ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้



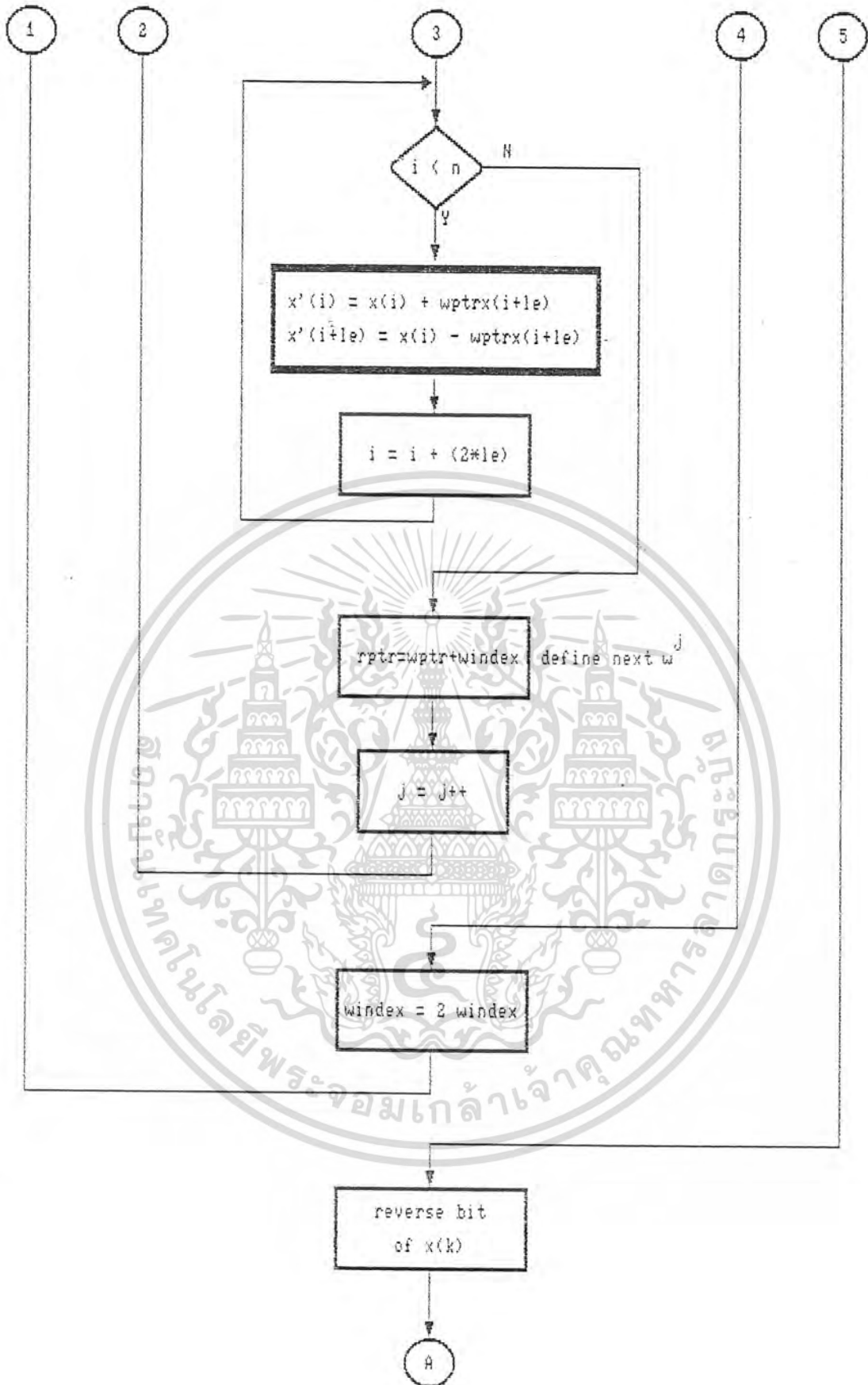
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ผู้ใช้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่หวังกำไรใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void fft(COMPLEX *x,int m)
{
    static COMPLEX *w;      /* used to store the w complex array */
    static int mstore=0;    /* stores m for future reference */
    static int n=1;        /* length of fft stored for future */

    COMPLEX u,temp,tm;
    COMPLEX *xi,*xip,*xj,*wptr;
    int i,j,k,l,le,windex;

    double arg,w_real,w_imag,wrecur_real,wrecur_imag,wtemp_real;

    if(m!=mstore){
        /* free previously allocated storage and set new m */
        if(mstore!=0) free(w);
        mstore=m;
        if(m==0) return; /* if m=0 then done */
        /* n = 2**m = fft length */
        n=1<<m;
        le=n/2;

        /* allocate the storage for w */
        w=(COMPLEX *)calloc(le-1,sizeof(COMPLEX));
        if(!w){
            fprintf(stderr,"NOT ENOUGH MEMORY FOR W VARIABLE");
            exit(1);
        }

        /* calculate the w values recursively */
        arg = 4.0*atan(1.0)/le; /* PI/le calculation */
        wrecur_real = w_real = cos(arg);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
wrecur_imag = w_imag = -sin(arg);
```

```
xj = w;
```

```
for (j=1;j<le;j++){
```

```
    xj->real = (float)wrecur_real;
```

```
    xj->imag = (float)wrecur_imag;
```

```
    xj++;
```

```
    wtemp_real = wrecur_real*w_real-wrecur_imag*w_imag;
```

```
    wrecur_imag = wrecur_real*w_imag+wrecur_imag*w_real;
```

```
    wrecur_real = wtemp_real;
```

```
}
```

```
}
```

```
/* start fft */
```

```
le = n;
```

```
windex = 1;
```

```
for(l=0;l<m;l++) {
```

```
    le = le/2;
```

```
/*first iteration with no multiplies */
```

```
for(i=0;i<n;i=i+2*le) {
```

```
    xi = x+i;
```

```
    xip = xi+le;
```

```
    temp.real = xi->real + xip->real;
```

```
    temp.imag = xi->imag + xip->imag;
```

```
    xip->real = xi->real - xip->real;
```

```
    xip->imag = xi->imag - xip->imag;
```

```
    *xi = temp;
```

```
}
```

```
/* remaining iterations use stored w */
```

```
wptr = w + windex -1;
```

```
for(j=1;j<le;j++){
```

```
    u=*wptr;
```

```
    for(i=j;i<n;i=i+2*le) {
```

```
        xi=x+i;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

xip=xi+le;
temp.real = xi->real + xip->real;
temp.imag = xi->imag + xip->imag;
tm.real = xi->real - xip->real;
tm.imag = xi->imag - xip->imag;
xip->real = tm.real*u.real - tm.imag*u.imag;
xip->imag = tm.real*u.imag + tm.imag*u.real;
*xi = temp;
    }
    wptr = wptr + windex;
}
windex = 2*windex;
}
/* rearrange data by bit reversing */
j = 0;
for(i=1;i<(n-1);i++) {
    k = n/2;
    while(k <= j) {
        j = j-k;
        k = k/2;
    }
    j = j+k;
    if(i<j) {
        xi = x+i;
        xj = x+j;
        temp = *xj;
        *xj = *xi;
        *xi = temp;
    }
}
;
/* rearrange data from 0-pi to (-pi)-pi */
for(i=0;i<(n/2);i++) {
    xi=x+i;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

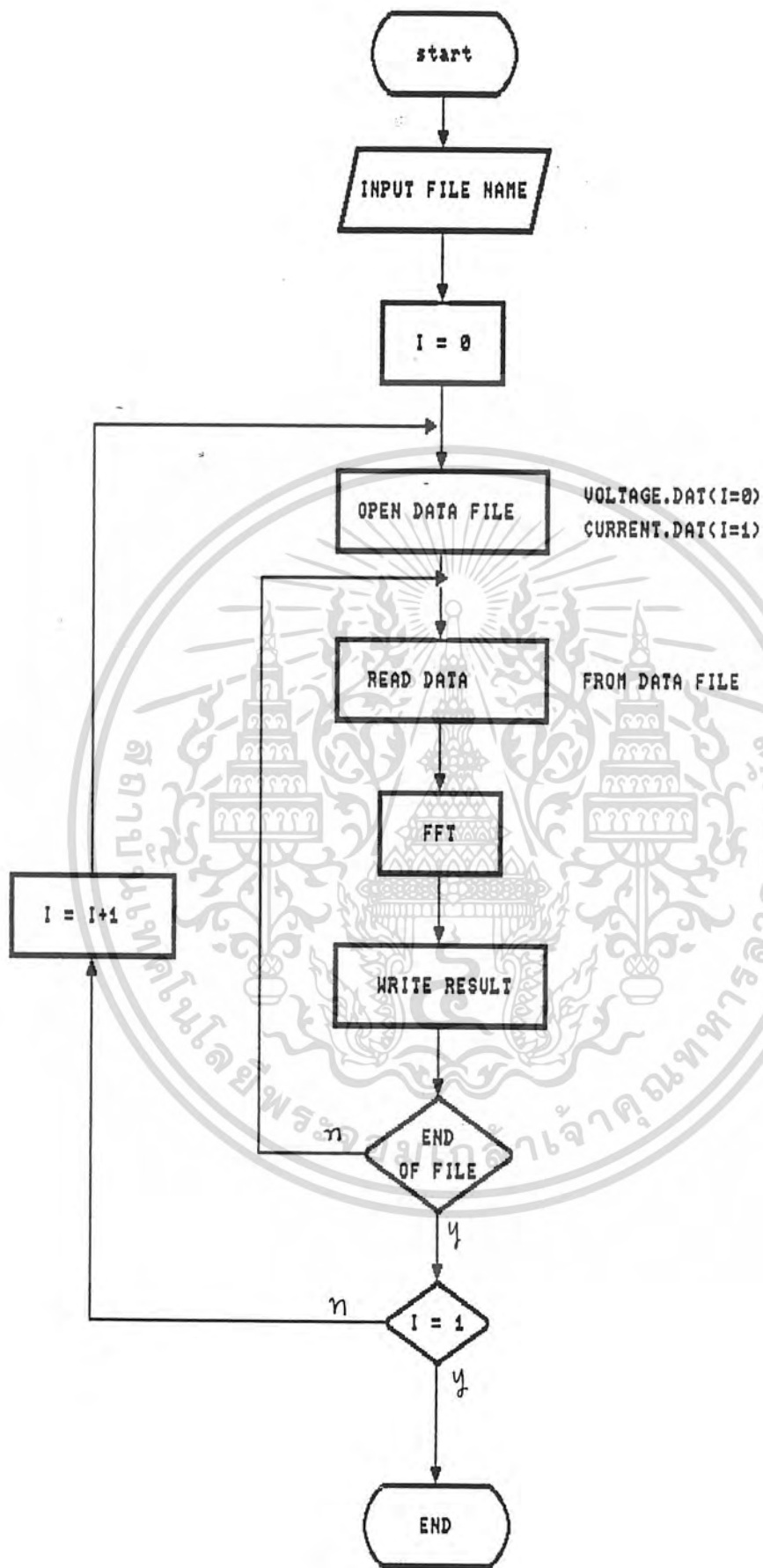
```
xj=xi+n/2;  
temp=*xi;  
*xi=*xj;  
*xj=temp;
```

```
}
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void input_data()
{
    int  x1, y1, x2, y2,
    x, y,
    'count,no_data[10],done,
    m,n,i=1;
    char  fname[30],date[10],time[10],
    temp_vol_fname[30],temp_cur_fname[30];
    void  far *input_data_box;
    FILE  *fp_in, *fpc_in, *fpv_in,
    *fp_out, *fpc_out, *fpv_out;
    float *xknag,dat,temp,arg,eoh=9999;
    COMPLEX *data,*dummy;

/* INPUT FILE NAME */
    in_fname(fname);
    if(fname[0]=='\0')
        return;

    x1=midx()-150;
    y1=220;
    x2=midx()+160;
    y2=250;

/* OPEN VOLTAGE DATA FILE */
    fpv_in=fopen("voltage.dat","rb"); /*mon_f*/
    if(!fpv_in)
    {
        warn_message("Couldn't open input data file INPUT DATA PROCESS ERROR");
        return;
    }

/* OPEN CURRENT DATA FILE */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fpc_in=fopen("current.dat","rb");          /*mon_f*/
if(!fpc_in)
{
    warn_message("Couldn't open input data file INPUT DATA PROCESS ERROR");
    return;
}

/* OPEN VOLTAGE XN&XK FILE */
sprintf(temp_vol_fname,"%s.VOL",fname);
fpv_out=fopen(temp_vol_fname,"wb");
if(!fpv_out)
{
    printf("Couldn't open file %s.vol (wb)",temp_vol_fname);
    return ;
}

/* OPEN CURRENT XN&XK FILE */
sprintf(temp_cur_fname,"%s.CUR",fname);
fpc_out=fopen(temp_cur_fname,"wb");
if(!fpc_out)
{
    printf("Couldn't open file %s.volt (wb)",temp_cur_fname);
    return ;
}

setcolor(WHITE);
input_data_box=save_image(x1,y1,x2,y2);
draw_fill_rectangle(x1,y1,x2,y2,RED);
outtextxy((x1+x2)/2-30,y1+10,"WAITING !");

/* RUN CURRENT & VOLTAGE */
for(i=1;i<=2;i++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(i==1) { fp_out=fpv_out; fp_in=fpv_in;}
else     { fp_out=fpc_out; fp_in=fpc_in;}
hour=0;
done=0;

/* ALLOCATE XN & XK FILE FOR HEAD DETAIL */

fprintf(fp_out,"%d\n",hour);
fprintf(fp_out,"%f\n",set_samp_rate);
fprintf(fp_out,"%d\n",set_samp_no);
fprintf(fp_out,"%s\n",set_win);

x1=midx()-160;
y1=220;
x2=midx()+160;
y2=250;

/* READ DATA FROM DATA FILE WRITE INTO CURRENT&VOLTAGE FILE */
/* CAL FFT WRITE INTO XN FILE */
fscanf(fp_in,"%f\n",&dat);
fprintf(fp_out,"%f\n",ech);

do
(
draw_fill_rectangle(x1,y1,x3,y2,RED);
x=(x1+x2)/2-150;
y=y1+10;
if(i==1) gprintf(&x,&y,"CALCULATING VOLTAGE SIGNAL HOUR : %d",hour+1);
if(i==2) gprintf(&x,&y,"CALCULATING CURRENT SIGNAL HOUR : %d",hour+1);

fscanf(fp_in,"%s\n",&date);
fscanf(fp_in,"%s\n",&time);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count=0;
fprintf(fp_out,"%s\n",date);
fprintf(fp_out,"%s\n",time);

/* ALLOCATE MEMORY FOR XN&XK DATA */
data=(COMPLEX*)calloc(set_samp_no,sizeof(COMPLEX));
if(!data)
{
    gprintxy(100,300,"NOT ENOUGH MEMOLY FOR DATA VARIABLE");
    exit(1);
}

/* BEGIN TRANSFORM */

if(strcmp(set_win,"RECTANGLE")) {
    if(strcmp(set_win,"TRIANGLE"))
        arg = 2.0/(set_samp_no-1);
    else arg = 2.0*4*atan(1.0)/(set_samp_no-1);
}
dummy=data;

/* READ AND WINDOW DATA */
do
{
    fscanf(fp_in,"%f\n",&dat);
    if(dat==9999)
        dat=dat+1-1;
    if((count<set_samp_no)&&((int)dat!=(int)eoh))
    {

/* SCALE CURRENT */
        if(i==2)
            dat=dat*scale_current;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fprintf(fp_out,"%f\n",dat);

/* WINDOW */
/* HANNING */
if(!strcmp(set_win,"HANNING"))
    dat=dat*(0.5-0.5*cos(arg*count));

/* HAMMING */
else if(!strcmp(set_win,"HAMMING"))
    dat=dat*(0.54-0.46*cos(arg*count));

/* TRIANGLE */
else if(!strcmp(set_win,"TRIANGLE")) {
    if(count<=(set_samp_no-1)/2)
dat=dat*(count*arg);
    else
dat=dat*(2.0-count*arg);
}

/* BLACKMAN */
else if(!strcmp(set_win,"BLACKMAN"))
    dat=dat*(0.42-0.5*cos(arg*count)-0.08*cos(2.0*arg*count));

/* BLACKMAN-HARRIS */
else if(!strcmp(set_win,"BLK-HARRIS"))
    dat=dat*(0.35875-0.48829*cos(arg*count)+0.14128*cos(2*arg*count)-0.01168*cos(3*ar

data->real=dat;
data++;
count++;
}
}while((dat!=eoh)&&!feof(fp_in));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

no_data[hour]=count;

/* IF SAMPING NO. NOT CORRECT SET NEW SAMPING NO. */
if(count<set_samp_no)
{
    if(count>=512) count=512;
    else if(count>=256) count=256;
    else if(count>=128) count=128;
    else count=0;
}

    samp_no=count;

hour++;
data=dummy;

/* FIND m */
m=log((double)count)/log(2.0);

/* FFT */
fft(data,m);

/* ALLOCATE MEMORY FOR XKMAG DATA */
xkmag=(float*)calloc(count,sizeof(float));
if(!xkmag)
{
    gprintxy(100,300,"NOT ENOUGH MEMOLY FOR XKMAG VARIABLE");
    return;
}

/* CAL MAGNITUDE */
fprintf(fp_out,"%f\n",eoh);
for(n=0;n<count;n++)
{
    temp= data[n].real*data[n].real;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp+= data[n].imag*data[n].imag;
xkmag[n]= sqrt((double)temp)/count;
fprintf(fp_out,"%f\n",xkmag[n]);
}
fprintf(fp_out,"%f\n",eoh);

free(data);
free(xkmag);

```

```

if(!feof(fp_in)) done=1;
    } while (!done);

```

```

/* WRITE DETAIL IN XN & XK FILE */

```

```

rewind(fp_out);
fprintf(fp_out,"%d\n",hour);
fprintf(fp_out,"%f\n",set_samp_rate);
fprintf(fp_out,"%d\n",samp_no);
fprintf(fp_out,"%s\n",set_win);
}

```

```

setcolor(WHITE);

```

```

setfillstyle(1, GREEN);

```

```

bar(x1,y1,x2,y2);

```

```

outtextxy((x1+x2)/2-30,y1+10,"READY 1");

```

```

delay(500);

```

```

putimage(x1,y1,input_data_box,0);

```

```

farfree(input_data_box);

```

```

fclose(fp_in);

```

```

fclose(fpv_in);

```

```

fclose(fpc_in);

```

```

fclose(fp_out);

```

```

fclose(fpv_out);

```

```

fclose(fpc_out);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

struct frame_struct {
    int block[4];
} frame[24];

```

```

char graph(FILE *fp,int first)

```

```

{
    char cha;
    static int xn=1;
    union inkey {
        char ch[2];
        int i;
    } buf;

    if(first)
    {
        setfillstyle(SOLID_FILL,LIGHTCYAN);
        bar(0,60,640,480);

        setfillstyle(SOLID_FILL,WHITE);
        bar(0,getmaxy()-20,getmaxx(),getmaxy());

        setcolor(RED);
        outtextxy(10,getmaxy()-15," <F1>          <RETURN>");
        setcolor(CYAN);
        outtextxy(10,getmaxy()-15,"          Switch Domain          Select");

        all_graph(fp,xn);
    }

    for(;;) {
        buf.i=bioskey(0);
        switch(buf.i) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case LEFT_ARROW : if(hour_select==0+1) break;
    normal_frame(hour_select);
hour_select=hour_select-1;
inverse_frame();
break;

    case RIGHT_ARROW: if(hour_select==(hour-1+1)) break;
normal_frame();
hour_select=hour_select+1;
inverse_frame();
break;

    case RETURN      : cha=each_graph(fp,1);
return cha;

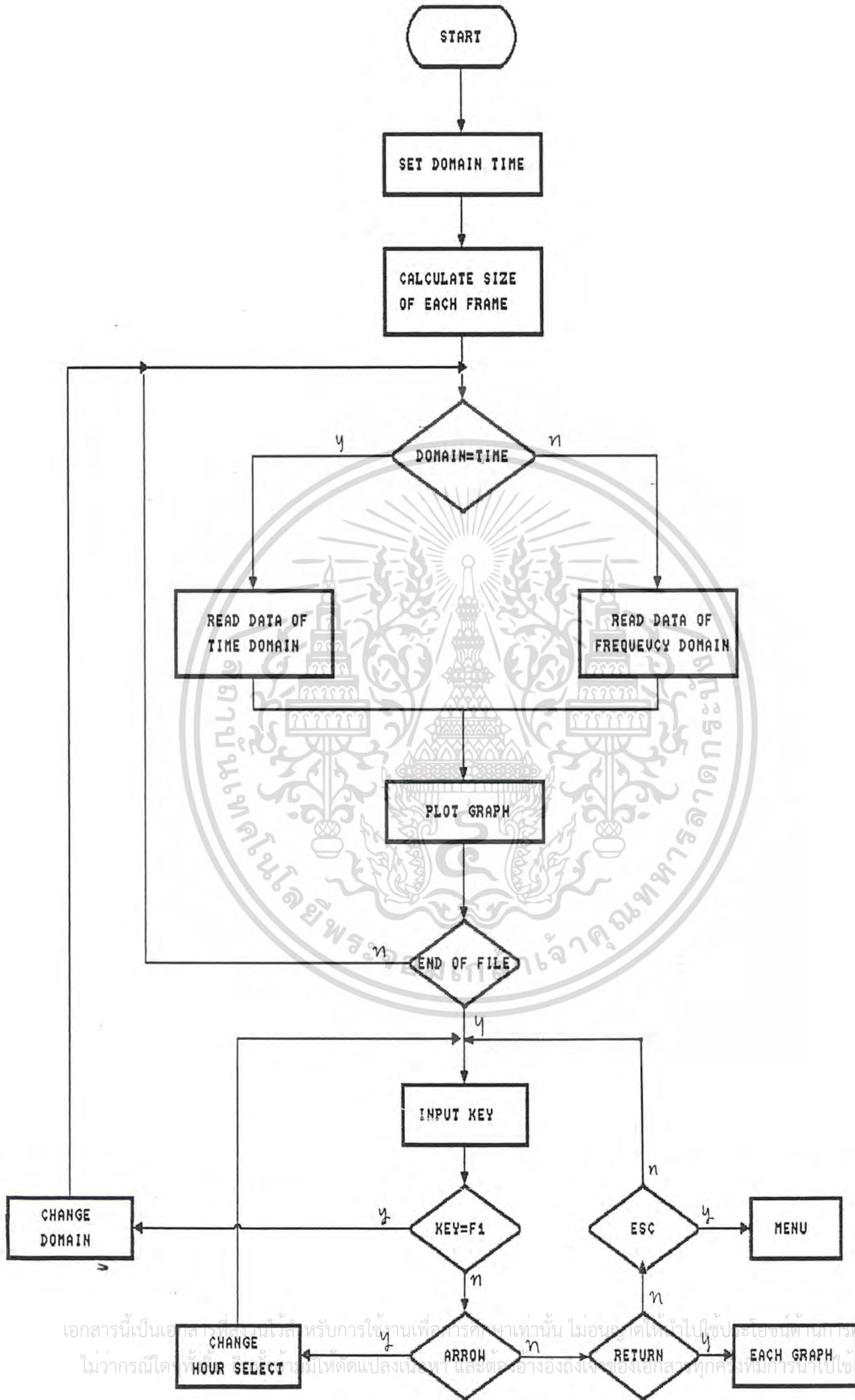
    case F1          : if(xn==1) xn=0; else xn=1;
all_graph(fp,xn);
hour_select=0+1;
break;

    case ESC         : return 'm';
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FUNCTION ALL GRAPH

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดที่เอกสารนี้ทำให้เกิดการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

```
void all_graph(FILE *fp,int xn)
```

```
{
```

```
    int i,j,
```

```
    w,h,
```

```
    xl,y1,x2,y2,
```

```
    x,y,
```

```
    y_axis,
```

```
    count,
```

```
    samp_no.
```

```
    frame_no,
```

```
    col,row,
```

```
    last_frame:
```

```
    float dat,*data,*dummy,max,min,n,samp_rate;
```

```
    char date[10].time[10].win[10];
```

```
/* DISPLAY DETAIL OF DATA SIGNAL */
```

```
x=15;y=65:
```

```
setcolor(CHAR_COLOR);
```

```
if(fp==fp1) {
```

```
    gprintf(&x,&y,"FILE NAME      : %s",vol_fname);
```

```
    y=y+15;
```

```
    gprintf(&x,&y,"SIGNAL      : VOLTAGE DOMAIN      :");
```

```
}
```

```
if(fp==fp2) {
```

```
    gprintf(&x,&y,"FILE NAME      : %s",cur_fname);
```

```
    y=y+15;
```

```
    gprintf(&x,&y,"SIGNAL      : CURRENT DOMAIN      :");
```

```
}
```

```
setcolor(LIGHTCYAN);
```

```
if(xn==1)
```

```
{
```

```
    gprintf(&x,&y,"
```

```
FREQUENCY" );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    setcolor(CHAR_COLOR);
    fprintf(&x,&y,"
    TIME");
}

else if(xn==0)
{
    fprintf(&x,&y,"
    TIME");
    setcolor(CHAR_COLOR);
    fprintf(&x,&y,"
    FREQUENCY");
}

rewind(fp);
fscanf(fp,"%d",&hour);
fscanf(fp,"%f",&samp_rate);
fscanf(fp,"%d",&samp_no);
fscanf(fp,"%s",win);
x=15;y=y+15;
fprintf(&x,&y,"SAMPLING RATE : %.1f kHz SAMPLING NUMBER : %d",samp_rate,samp_no);
x=15;y=y+15;
fprintf(&x,&y,"WINDOW : %s",win);

/* CALCULATE SIZE OF FRAME */
if(hour>30) { col=6;row=6; }
else if(hour>24) { col=6;row=5; }
else if(hour>20) { col=6;row=4; }
else if(hour>16) { col=5;row=4; }
else if(hour>12) { col=4;row=4; }
else if(hour>9) { col=4;row=3; }
else if(hour>6) { col=3;row=3; }
else if(hour>4) { col=3;row=2; }
else if(hour>2) { col=2;row=2; }
else if(hour>0) { col=2;row=1; }
else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

warn_message("INPUT FILE FORMAT IS NOT RIGHT");
exit(0);
}

x1 = 15;
y1 = y+20;
w = (getmaxx()-30)/col;
h = (getmaxy()-y1-30)/row;
x2 = x1+w;
y2 = y1+h;

for(j=0;j<=col-1;j++) {
    for(i=j;i<row*col;i=i+col) {
frame[i].block[0]=x1;
frame[i].block[2]=x2;
}

    x1=x2;
    x2=x1+w;
}
j=col-1;
for(i=0;i<row*col;i=i+1) {
    frame[i].block[1]=y1;
    frame[i].block[3]=y2;
    if(i>=j) {
y1=y2;
y2=y1+h;
j=j+col;
    }
}
}

```

```

/* DRAW FRAME */

```

```

setfillstyle(SOLID_FILL,WHITE);
setcolor(CYAN);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setlinestyle(0,0,3);
for(j=0;j<hour;j++) {
    bar( frame[j].block[0],frame[j].block[1],
frame[j].block[2],frame[j].block[3] );
    rectangle( frame[j].block[0],frame[j].block[1],
frame[j].block[2],frame[j].block[3] );
}

```

```

data= (float*)calloc(samp_no,sizeof(float));
if(!data) {
    printf("Memory not enough\n");
    exit(1);
}

```

```

/* READ DATA */

```

```

fscanf(fp,"%f",&dat);
for(j=0;j<hour;j++){
    count=0;
    max=0;
    min=0;

    setcolor(CYAN);
    fscanf(fp,"%s",date);
    outtextxy( frame[j].block[0]-7,frame[j].block[2]-7,date);
    fscanf(fp,"%s",time);
    outtextxy( frame[j].block[0]+7,frame[j].block[1]+15,time);
    dummy=dat;
}

```

```

/* PLOT GRAPH */

```

```

if(n==1) {
    y_axis = (int)((frame[j].block[3]+h-25)/2);
}
if(n==0) i

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y_axis = (int)(frame[j].block[3]-5);
}

setcolor(CYAN);
setlinestyle(DOTTED_LINE,0,1);
line(frame[j].block[0]+5,y_axis,frame[j].block[2]-5,y_axis);

if(xn==0) for(n=0;n<=samp_no;n++) { fscanf(fp,"%f",&dat); }
fscanf(fp,"%f",&dat);

while((dat!=9999)&&(count<samp_no)) {
if(dat>max) max=dat;
if(dat<min) min=dat;
*data=dat;
data++;
count++;
if(count==20)
count=count+1-1;
fscanf(fp,"%f",&dat);
}
data=dummy;

setcolor(LIGHTMAGENTA);
setlinestyle(0,0,1);
for(n=0;n<count;n++) {
x = ((n/count)*(w-10))+frame[j].block[0]+5;
if(xn==1) {
y = y_axis-(*data/(max-min))*((h-25)/2);
if(n!=0) lineto(x,y);
else moveto(x,y);
}
}
if(xn==0) {
y = y_axis-(*data/(max-min))*(h-30);
line(x,frame[j].block[3]-5,x,y);
}

```

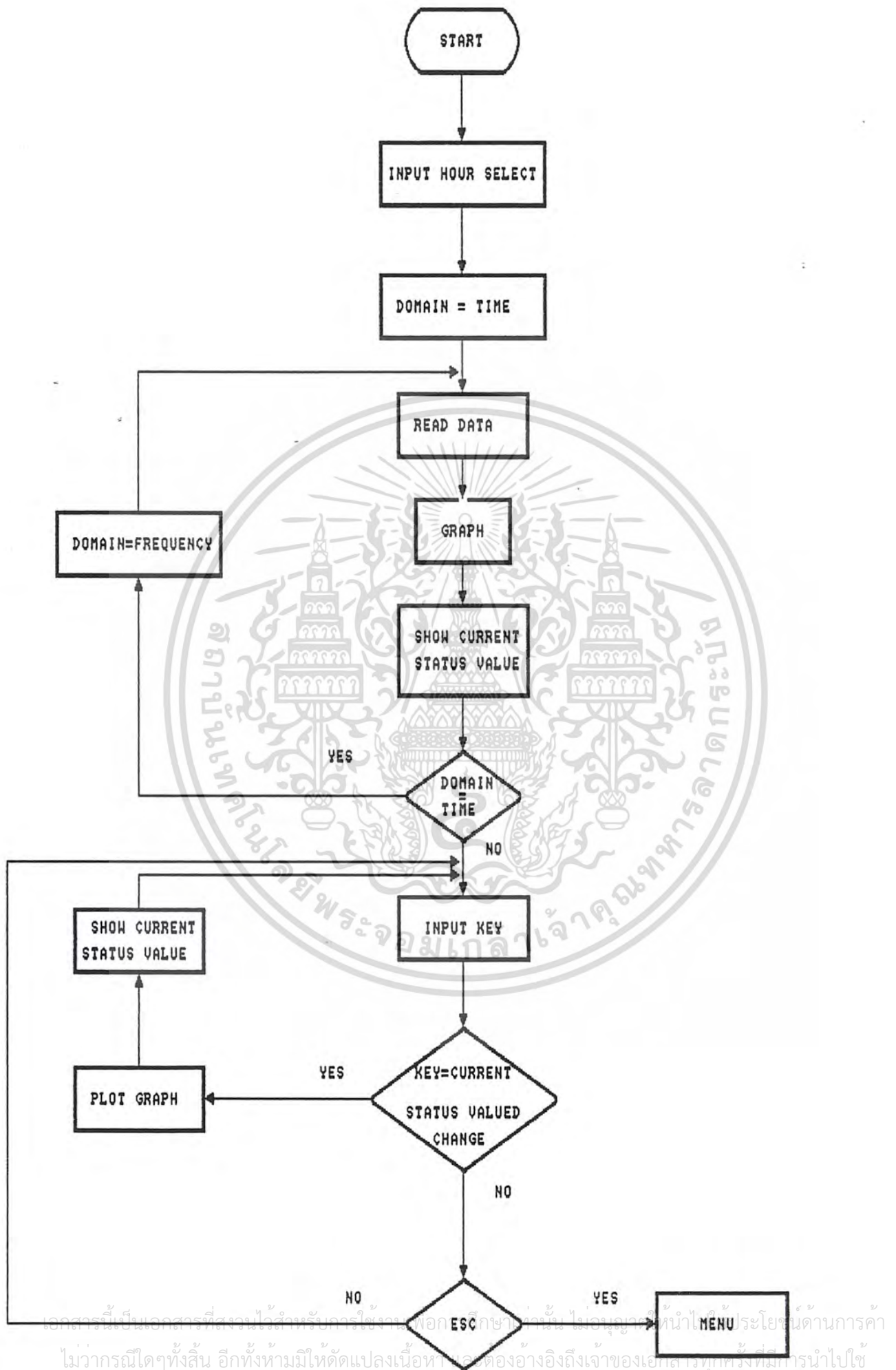


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
data++;  
    }  
    data=dummy;  
    if(xn==1) for(n=0;n<=samp_no;n++) { fscanf(fp,"%f",&dat); }  
}  
free(data);  
inverse_frame();  
return 1;  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

struct graph_struct {
    float *dummy;
    float scale_xaxis,scale_yaxis.
min,max:
    int   frame_x1,frame_y1,frame_x2,frame_y2.
step,
no_now,no_left,no_right.
center,shift;
};

```

```

char each_graph(FILE *fp,int first)

```

```

{
    int   n,i,count,
x,y,
hour;
    float *data[2],dat;

static int xn=0;
double tem;
char date[10],time[10],win[10];
static struct graph_struct graph[2];
char diamond[8]={0x10,0x38,0x7C,0xFE,0x7C,0x10,0x00};

```

```

setfillpattern(diamond,YELLOW);

```

```

/* FIRST CALL */

```

```

if(first)

```

```

{
    setfillstyle(SOLID_FILL,LIGHTCYAN);
bar(0,60,getmaxx(),getmaxy());
setfillstyle(SOLID_FILL,WHITE);
bar(0,getmaxy()-20,getmaxx(),getmaxy());
setcolor(RED);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

outtextxy(10,getmaxy()-15,"<F1>

<F3-F4>

<F2>

<PgUp-PgDw>

setcolor(CYAN);

outtextxy(10,getmaxy()-15,"

Switch

Scale X-Y

All Graph

St

```
/* READ DETAIL OF DATA FILE */
```

```
rewind(fp);
```

```
fscanf(fp,"%d",&hour);
```

```
fscanf(fp,"%f",&samp_rate);
```

```
fscanf(fp,"%d",&samp_no);
```

```
fscanf(fp,"%s",win);
```

```
rsamp_rate=samp_rate*1000;
```

```
/* SET DEFAULT */
```

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

```
graph[xn].center=samp_no/2;
```

```
graph[xn].no_now=samp_no/2;
```

```
graph[xn].no_right=samp_no;
```

```
graph[xn].no_left=0;
```

```
graph[xn].scale_xaxis=samp_no/3;
```

```
graph[xn].scale_yaxis=100;
```

```
graph[xn].step=1;
```

```
}
```

```
xn=0;
```

```
/* GO TO SELECTED HOUR */
```

```
read_freedata(fp,samp_no);
```

```
fscanf(fp,"%s",date);
```

```
fscanf(fp,"%s",time);
```

```
/* DISPLAY DETAIL OF DATA FILE */
```

```
setcolor(CHAR_COLOR);
```

```
x=15;y=65;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(fp==fp1) {
    fprintf(&x,&y,"FILE NAME      : %s",vol_fname);
    y=y+15;
    fprintf(&x,&y,"SIGNAL        : VOLTAGE");
}
if(fp==fp2) {
    fprintf(&x,&y,"FILE NAME      : %s",vol_fname);
    y=y+15;
    fprintf(&x,&y,"SIGNAL        : CURRENT");
}
y=y+15;
fprintf(&x,&y,"SAMPLING RATE : %.1fKHz      SAMPLING NUMBER : %d",samp_rate,samp_no);
y=y+15;
fprintf(&x,&y,"WINDOW          : %s",win);
y=y+15;
fprintf(&x,&y,"DATE : %s    TIME : %s",date,time);
x=60;
y=y+30;
fprintf(&x,&y,"DCMAIN          : DCMAIN");
y=y+15;
fprintf(&x,&y,"SAMPLING NO.   : SAMPLING NO.   HARMONIC :");
y=y+15;
fprintf(&x,&y,"TIME           : TIME           FREQUENCY :");
y=y+15;
fprintf(&x,&y,"AMPLITUDE     : AMPLITUDE");

graph[0].frame_y1=y+25;
graph[1].frame_y1=y+25;

```

```

/* DISPLAY SCALE AND STEP OF TRACK */

```

```

setcolor(CHAR_COLOR);

```

```

w=15;y=getmaxy()-50;

```

```

fprintf(&x,&y,"SCALE  M:          S/DI= V:          "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x=x+textwidth("SCALE X: ");
gprintf(&x,&y,"% .2f",value(1,'x',graph[1].scale_xaxis));
x=x+textwidth("      ms/Div Y: ");
gprintf(&x,&y,"% .2f",value(1,'y',graph[1].scale_yaxis));

x=getmaxx()/2+40;
gprintf(&x,&y,"X:      Hz/Div Y:      %/Div"):
x=x+textwidth("X: ");
gprintf(&x,&y,"% .0f",value(0,'x',graph[0].scale_xaxis));
x=x+textwidth("      Hz/Div Y: ");
gprintf(&x,&y,"% .2f",value(0,'y',graph[0].scale_yaxis));

chstep(1,1):
chstep(0,1):

/* READ DATA */
for(n=1;n>=0;n--)
{
  data[n]= (float)*acos(100/samp_nc+1*sizeof(float));
  if(!data[n])
    printf("Memory not enough\n");
  exit(1);
}

graph[n].dummy=data[n];

fscanf(fp,"%f",&dat):
count=0:
graph[n].max=0:graph[n].min=0:
while((dat!='9999')&&(count<samp_nc))
if(dat>graph[n].max) graph[n].max=dat:
if(dat<graph[n].min) graph[n].min=dat:
data[n]=dat:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data[n]++;
count++;
fscanf(fp,"%f",&dat);
    }
    if(n==0) *data[n]=*(graph[n].dummy);
    else *data[n]=9999;
    data[n]=graph[n].dummy;
}
xn=1;
/* PLOT GRAPH */
    show_text(1,"on",graph[1]);
    show_text(0,"off",graph[0]);
    show_graph(1,&graph[1]);
    track("track".1,graph[1]);
    show_graph(0,&graph[0]);
    track("track".0,graph[0]);
} /* if */
/* CONTINUOUS RUN */
else
{
    rewind(fp);
    fscanf(fp,"%d",&hour);
    fscanf(fp,"%f",&samp_rate);
    fscanf(fp,"%d",&samp_no);
    fscanf(fp,"%s",win);
    read_freedata(fp,samp_no);

    fscanf(fp,"%s",date);
    fscanf(fp,"%s",time);

    while(n>0;n--)

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
data[n]= (float*)calloc(samp_no+1,sizeof(float));
if(!data[n]) {
printf("Memory not enough\n");
exit(1);
}

graph[n].dummy=data[n];

fscanf(fp,"%f",&dat);
count=0;
graph[n].max=0;graph[n].min=0;
while((dat!=9999)&&(count<samp_no)) {
if(dat>graph[n].max) graph[n].max=dat;
if(dat<graph[n].min) graph[n].min=dat;
*data[n]=dat;
data[n]++;
count++;
fscanf(fp,"%f",&dat);
if(n==0) *data[n]=*(graph[n].dummy);
else *data[n]=9999;
data[n]=graph[n].dummy;
}
track("track",1,graph[1]);
track("track",0,graph[0]);
} /* else */

highlight_domain(graph[xn]);

/* INPUT HOTKEY */
for(;;) {
n=bioskey(0);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(n) {
    case RIGHT_ARROW:track("untrack",xn,graph[xn]);
graph[xn].no_now=graph[xn].no_now+graph[xn].step;
if(graph[xn].no_now>samp_no)
    graph[xn].no_now=samp_no;
if(graph[xn].no_now>graph[xn].no_right)
{
    graph[xn].shift=graph[xn].no_now-graph[xn].no_right;
    graph[xn].center+=graph[xn].shift;
    graph[xn].no_right+=graph[xn].shift;
    graph[xn].no_left+=graph[xn].shift;
    show_graph(xn,&graph[xn]);
}
track("track",xn,graph[xn]);
show_text(xn,"on",graph[xn]);
break;
    case LEFT_ARROW :track("untrack",xn,graph[xn]);
graph[xn].no_now-=graph[xn].step;
if(graph[xn].no_now<0)
{
    graph[xn].no_now=0;
}
if(graph[xn].no_now<graph[xn].no_left)
{
    graph[xn].shift=graph[xn].no_left-graph[xn].no_now;
    graph[xn].center-=graph[xn].shift;
    graph[xn].no_right-=graph[xn].shift;
    graph[xn].no_left-=graph[xn].shift;
    show_graph(xn,&graph[xn]);
}
track("track",xn,graph[xn]);
show_text(xn,"on",graph[xn]);
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case F4 : track("free",0,graph[0]);
        track("free",1,graph[1]);
        free(data[0]);
        free(data[1]);
        return 'g';
case F1 : normal_domain(graph[xn]);
        show_text(xn,"off",graph[xn]);
        if(xn==0) xn=1;
else xn=0;
        highlight_domain(graph[xn]);
        show_text(xn,"on",graph[xn]);
        break;
case F2 : chscale_xaxis(xn,&graph[xn].scale_xaxis);
        graph[xn].no_right=graph[xn].center+graph[xn].scale_xaxis*4;
        graph[xn].no_left=graph[xn].center-graph[xn].scale_xaxis*4;
        if(graph[xn].no_now>graph[xn].no_right)
graph[xn].no_now=graph[xn].no_right;
else if(graph[xn].no_now<graph[xn].no_left)
graph[xn].no_now=graph[xn].no_left;
        show_graph(xn,&graph[xn]);
        track("track",xn,graph[xn]);
        show_text(xn,"on",graph[xn]);
        break;
case F3 : chscale_yaxis(xn,&graph[xn].scale_yaxis);
        show_graph(xn,&graph[xn]);
        track("track",xn,graph[xn]);
        show_text(xn,"on",graph[xn]);
        break;
case PgUp: graph[xn].step=graph[xn].step*2;
if(graph[xn].step>=samp_no) graph[xn].step=samp_no;
chstep(xn,graph[xn].step);
break;
case PgDw: graph[xn].step=graph[xn].step/2;

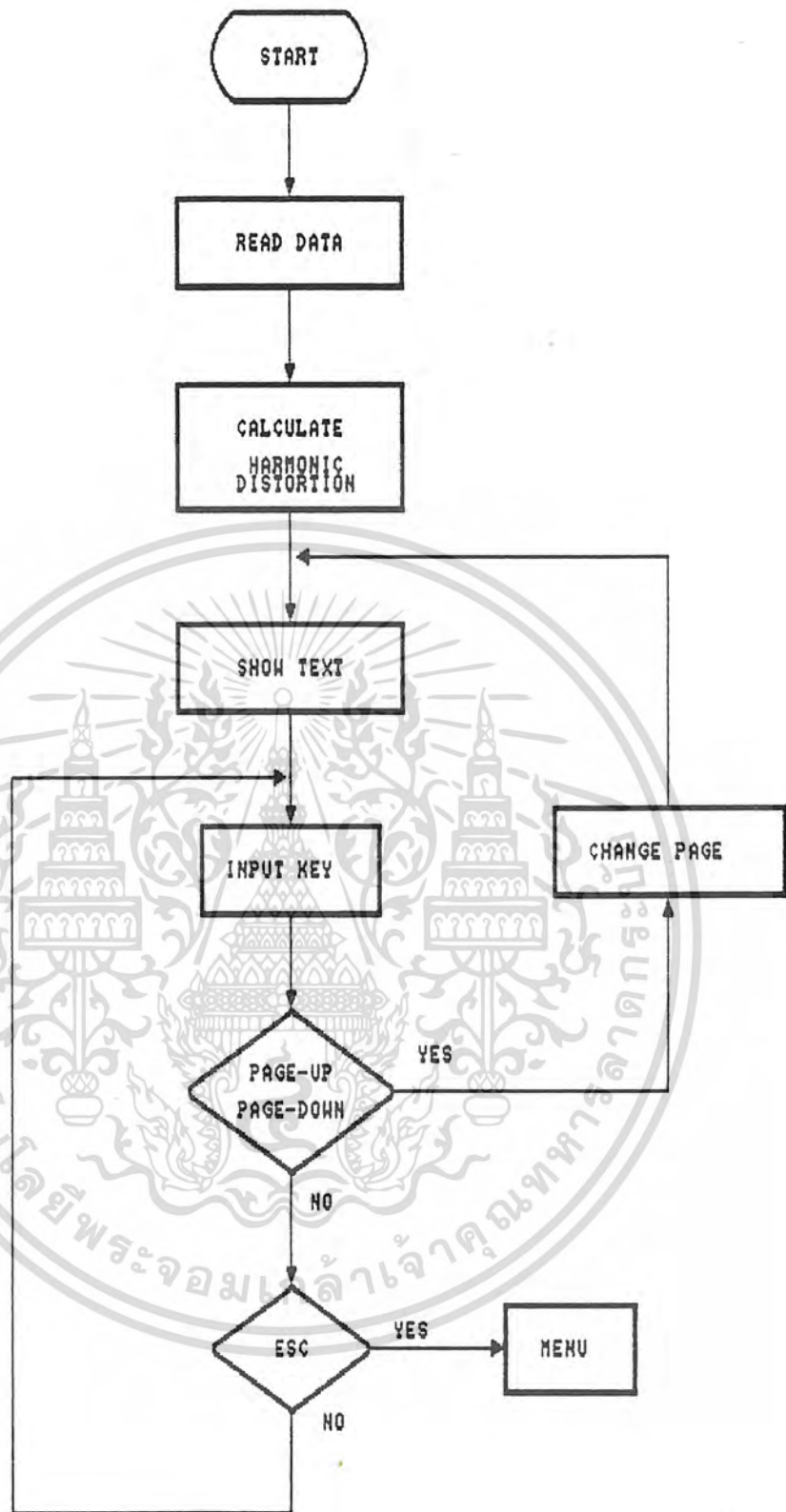
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(graph[xn].step<1) graph[xn].step=1:
chstep(xn,graph[xn].step);
break;
    case ESC: track("free".0,graph[0]);
        track("free",i,graph[1]);
        free(data[0]);
        free(data[1]);
        return 'e';
    }
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FUNCTION TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char text(FILE *fp,int first)
{
    int i,n,samp_no,
    last_page,
    no_line=20,
    no_data:
    float har_dis,har.cof:
    static int page_now=0;
    float *dummy,*data,dat,samp_rate.
max=0,thd=0,rms=0;
    char win[10],date[10],time[10];

```

```

/* READ DETAIL OF DATA FILE */

```

```

rewind(fp);
fscanf(fp,"%d/n",&hour);
fscanf(fp,"%f/n",&samp_rate);
fscanf(fp,"%d/n",&samp_no);
fscanf(fp,"%s/n",win);

```

```

/* GO TO READ HOUR SELECTED DATA */

```

```

read_freedata(fp,samp_no);
fscanf(fp,"%s/n",date);
fscanf(fp,"%s/n",time);

```

```

for(n=1;n<=(samp_no+1);n++)

```

```

fscanf(fp,"%f/n",&dat);

```

```

/* ALLOCATION MEMOLY FOR DATA */

```

```

data=(float*)calloc(samp_no,sizeof(float));

```

```

if (!data) {

```

```

printf("Memory not enough!");

```

```

exit(1);

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* READ DATA */
dummy=data;
for(n=1;n<=samp_no;n++) {
    fscanf(fp,"%f",&dat);
    if(dat>max)
        max=dat;
    *data=dat;
    data++;
}
data=dummy;

```

```

/* FIND HARMONIC DISTORTION */

```

```

no_data=samp_no/2+1;
cof=samp_rate*1000/samp_no;
*(data+samp_no)=*data;
for(i=0;i<no_data;i++) {
    har_dis = *(data+samp_no/2)/max;
    *data = har_dis*100;
    har=cof*(i)/FUNDAMENTAL;
    if(!(ceil(har-floor(har))&&har!=0))
thd += har_dis*har_dis;
    data++;
}
rms = sqrt(1+thd*(thd));
thd = sqrt(thd)*100;
data=dummy;

```

```

/* DISPLAY GRAPH */

```

```

if(first) {
    page_now=0;
    edit(date,time,thd,rms);
    write_data(page_now,data,samp_rate,samp_no);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
last_page=samp_no/(no_line*4);
```

```
/* INPUT HOT KEY */
```

```
for(;;) {
```

```
  i=bioskey(0);
```

```
  switch(i) {
```

```
    case PgUp : if(page_now!=0) {
```

```
      page_now-=1;
```

```
      write_data(page_now,data,samp_rate,samp_no);
```

```
    }
```

```
    break;
```

```
    case PgDw : if(page_now!=last_page) {
```

```
      page_now+=1;
```

```
      write_data(page_now,data,samp_rate,samp_no);
```

```
    }
```

```
    break;
```

```
    case ESC : free(data);
```

```
    return 'T';
```

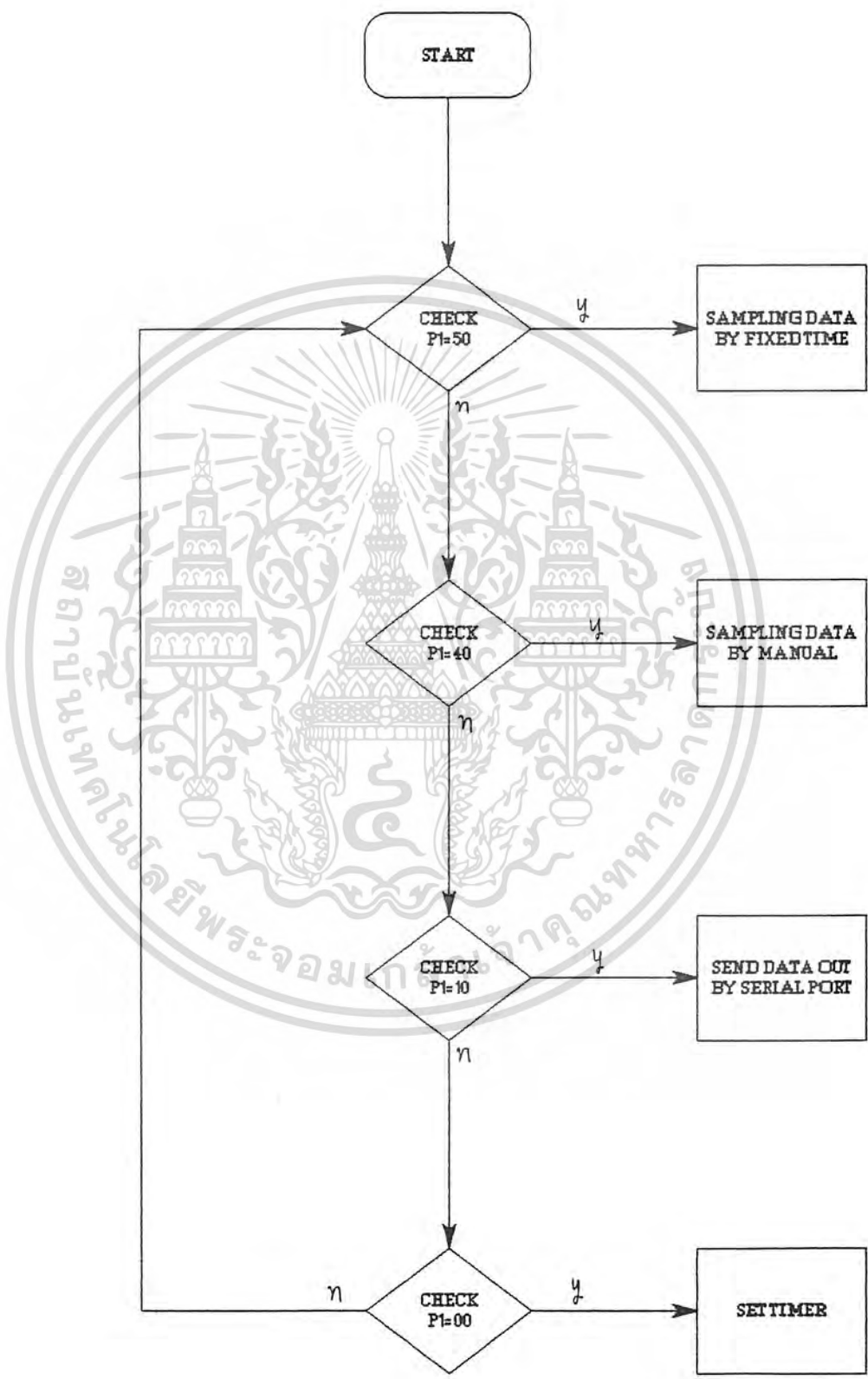
```
  }
```

```
}
```

```
}
```

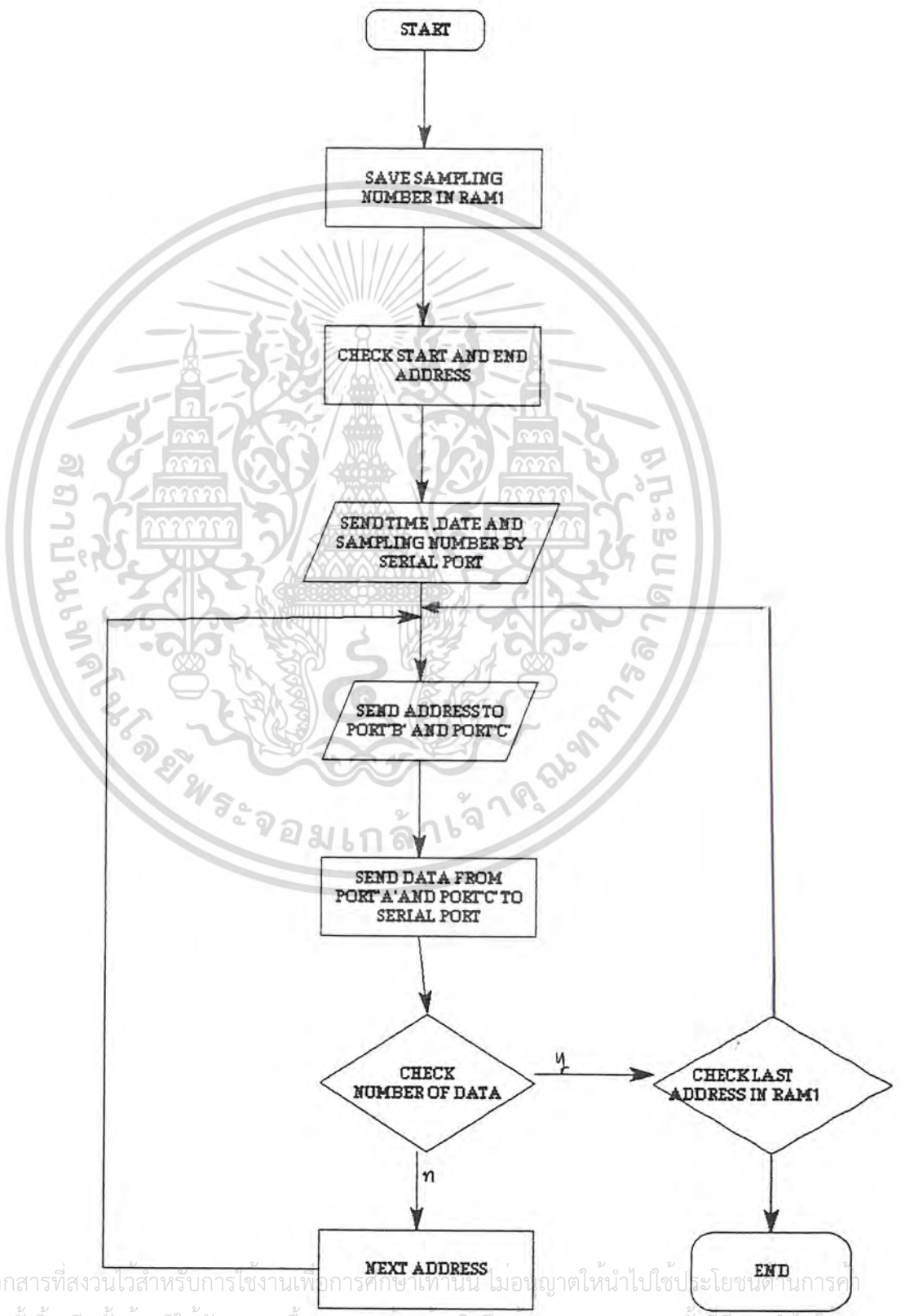


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

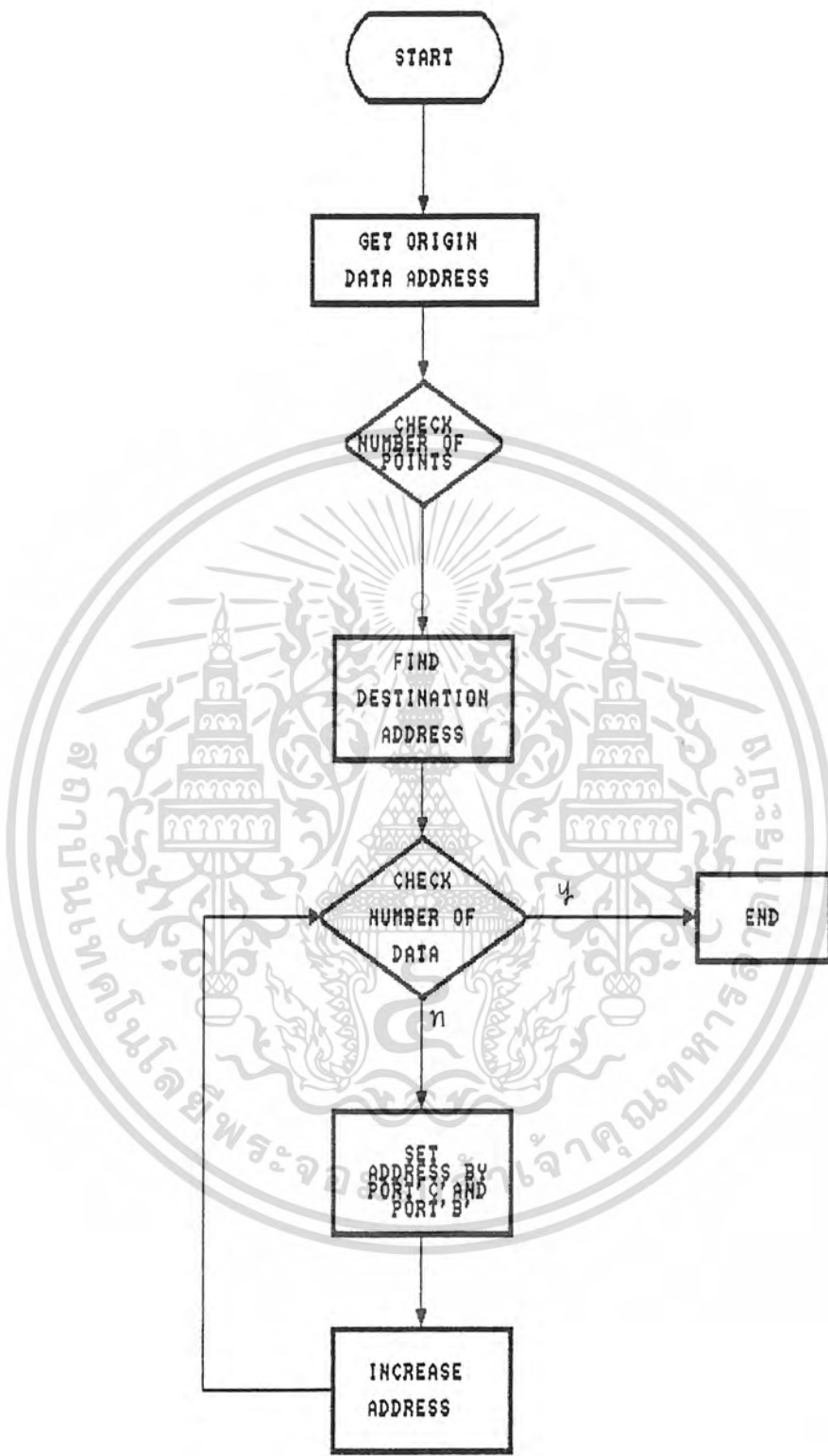


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SAMPLING DATA BY FIXED TIME



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FUNCTION KEEP DATA IN RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETBUF EQU 50H
PA EQU 0FE00H
PB EQU 0FE01H
PC1 EQU 0FE02H

```

```

ORG 0000H
LJMP START
ORG 0003H
LJMP SEINT0
ORG 0100H

```

```

START: LCALL DELAY
MOV SP,#35H
MOV A,P1

```

```

SCKTIM: CJNE A,#50H,SUSINT
SJMP TWORK

```

```

SUSINT: CJNE A,#10H,SSERIAL
LJMP WORK

```

```

SSERIAL: SJMP START
CJNE A,#40H,STIME
LJMP SERIAL4

```

```

STIME: SJMP START
CJNE A,#00H,SCKTIM
LJMP TIME
SJMP START

```

```

;##### TWORK #####

```

```

TWORK: LCALL RDTIME
MOV A,R1
CLR C
SUBB A,R3
JZ TWORK
MOV A,R4
MOV R1,A

```

```

;##### WORK #####

```

```

WORK: MOV DPTR,#0FE03H ;0FE03
MOV A,#98H
MOVX @DPTR,A
CLR C

```

```

S2: MOV DPTR,#9F00H
MOVX A,@DPTR
JZ S0
SETB P1.2
SJMP S1

```

```

S0: CLR P1.2

```

```

S1: MOV DPTR,#9F01H
MOVX A,@DPTR
ANL A,#0FH
MOV R7,A
MOV DPTR,#9F02H
MOVX A,@DPTR

```

```

MOV R6,A
MOV A,R7
JB P1.7,Y1
MOV R0,#01H
ADD A,R0

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การนำเอกสารนี้ไปใช้ในการเรียนการสอนโดยไม่ได้รับอนุญาตให้ไปใช้ประโยชน์ด้านการค้า
 YO: ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Y1:      SJMP      Y4
         JB       P3.4,Y2
         MOV      R0,#02H
         ADD      A,R0
         SJMP      Y4
Y2:      JB       P3.5,Y0
         MOV      R0,#04H
         ADD      A,R0
Y4:      MOV      R1,A
         JNB      P1.2,Y5

Y5:      JB       P1.3,$
CKIV1:   JNB      P1.3,$
         MOV      DPTR,#9F50H
         MOVX     A,@DPTR
         MOV      45H,A
         MOV      DPTR,#9F51H
         MOVX     A,@DPTR
         MOV      46H,A
         MOV      DPL,45H
         MOV      DPH,46H
         MOV      A,R0
         MOVX     @DPTR,A           ; KEEP NUMBER OF POINTS
         INC      DPTR
         PUSH     DPH
         PUSH     DPL
         LCALL    RDTIME
         POP      DPL
         POP      DPH
         MOV      A,R2
         MOVX     @DPTR,A           ; KEEP HOURS
         INC      DPTR
         MOV      A,R3
         MOVX     @DPTR,A           ; KEEP MINUTES
         INC      DPTR
         PUSH     DPH
         PUSH     DPL
         LCALL    RDDATE
         POP      DPL
         POP      DPH
         MOV      A,R2
         MOVX     @DPTR,A           ; KEEP DATES
         INC      DPTR
         MOV      A,R3
         MOVX     @DPTR,A           ; KEEP MONTH
         INC      DPTR
         MOV      A,R4
         MOVX     @DPTR,A           ; KEEP YEAR
         CLR      P1.0
         CLR      P1.1
         CLR      C
         CLR      P1.5

         SETB     PX0
         SETB     IT0
         SETB     EX0
         SETB     EA

```

MAIN:

```

MOV      A,R1
XRL      A,R7
JNZ      MAIN
CLR      EX0
SETB     P1.0
SETB     P1.1
SETB     P1.5
MOV      DPTR,#9F00H           ; KEEP ADD OF P1.2

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 MON: ไม่วารณี่ใดๆทั้งสิ้น อีกทั้งยังห้ามนำไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JNB      P1.2,KEEP
MOV      A,#0FFH
MOVX    @DPTR,A
SJMP    KEEP
KEEP:    MOV      A,#00H
MOVX    @DPTR,A
KEEPPC: MOV      DPTR,#9F01H      ; KEEP ADD OF PC
MOV      A,R7
MOVX    @DPTR,A
KEEPPB: MOV      DPTR,#9F02H      ; KEEP ADD OF PB
MOV      A,R6
MOVX    @DPTR,A
PAR:    MOV      DPTR,#9F50H      ; KEEP ADD POSITION FIRST
MOVX    A,@DPTR
ADD     A,#06H
MOV     R0,A
JNC    PAR1
MOV     DPTR,#9F51H
MOVX    A,@DPTR
ADD     A,#01H
MOVX    @DPTR,A
PAR1:   MOV      A,R0
MOV     DPTR,#9F50H
MOVX    @DPTR,A
MOV     A,P1
ANL    A,#50H
CJNE   A,#50H,HTIME
LJMP   TWORK
HTIME:  SJMP    $
;***** SEINTO *****
SEINTO: JC      PU
MOV     DPTR,#0FE02H      ;0FE02
MOV     A,R7
MOVX    @DPTR,A
MOV     DPTR,#0FE01H      ;0FE01
MOV     A,R6
MOVX    @DPTR,A
ADD     A,#01H
MOV     R6,A
MOV     A,R0
RETI
PU:     CLR     AC
CLR     C
MOV     A,R7
ADD     A,#01H
MOV     R7,A
JNB    AC,NOI
SETB   P1.2
NOI:    MOV     DPTR,#0FE02H      ;0FE02
MOV     A,R7
MOVX    @DPTR,A
MOV     DPTR,#0FE01H      ;0FE01
MOV     A,R6
MOVX    @DPTR,A
INC     R6
MOV     A,R0
RETI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
;##### CLRRAM #####

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLRRAM: MOV     DPTR,#8000H
CYC:    MOV     A,#00H
MOVX    @DPTR,A

```

```

INC DPTR
MOV A,DPH
XRL A,#0A0H
JNZ CYC
MOV A,#80H
MOV DPTR,#9F51H
MOVX @DPTR,A
RET

```

```

;##### SERIAL #####

```

SERIAL4:

```

CLR EA
CLR EX0
MOV TMOD,#26H
MOV SCON,#52H
MOV TH1,#0FDH
SETB TR1
SETB P1.0
CLR P1.1
SETB P1.5
MOV SP,#35H
MOV DPTR,#0FE03H
MOV A,#98H
MOVX @DPTR,A
CLR P1.2
MOV R6,#00H
MOV R7,#00H
MOV DPTR,#9F40H
MOV A,#00H
MOVX @DPTR,A

```

```

;0FE03H

```

CKF:

```

MOV DPTR,#9F40H
MOVX A,@DPTR
ADD A,#06H
MOV R5,A
MOV DPTR,#9F50H
MOVX A,@DPTR
ADD A,#06H
XRL A,R5
JNZ CKF1
LJMP FIN1

```

CKF1:

```

MOV A,R5
SUBB A,#06H
MOV DPTR,#9F40H
MOVX A,@DPTR
MOV DPL,A
MOV DPH,#80H
MOVX A,@DPTR
MOV R0,A
LCALL SBYTEH ; SENT NUMBER OF POINTS
MOV A,R7
ADD A,R0
MOV R4,A
INC DPTR
MOVX A,@DPTR
LCALL SBYTEH ; SENT HOURS
INC DPTR
MOVX A,@DPTR
LCALL SBYTEH ; SENT MINUTES
INC DPTR
MOVX A,@DPTR
LCALL SBYTEH ; SENT DATES
INC DPTR
MOVX A,@DPTR
LCALL SBYTEH ; SENT MONTHS
INC DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดเป็นข้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX    A,@DPTR
LCALL   SBYTE           ; SENT YEARS

MAIN1:
MOV     DPTR,#PC1
MOV     A,R7
MOVX    @DPTR,A
MOV     DPTR,#PB
MOV     A,R6
MOVX    @DPTR,A
MOV     DPTR,#PC1
MOVX    A,@DPTR
ANL     A,#0F0H
SWAP    A
LCALL   SBYTE
MOV     DPTR,#PA
MOVX    A,@DPTR
LCALL   SBYTE
MOV     A,R6
ADD     A,#01H
MOV     R6,A
JC      AUGM
LJMP    MAIN1

AUGM:
CLR     C
CLR     AC
MOV     A,R7
ADD     A,#01H
JNB     AC,AUGM1
SETB    P1.2
AUGM1:
MOV     R7,A
XRL    A,R4
JZ      FIN
LJMP    MAIN1

FIN:
MOV     A,#'+ '
LCALL   SBYTE
MOV     A,#'+ '
LCALL   SBYTE
MOV     DPTR,#9F40H
MOVX    A,@DPTR
ADD     A,#06H
MOV     R5,A
MOVX    @DPTR,A
MOV     DPTR,#9F50H
MOVX    A,@DPTR
ADD     A,#06H
XRL    A,R5
JZ      FIN1
LJMP    CKF

FIN1:
MOV     A,#'* '
LCALL   SBYTE
MOV     A,#'+ '
LCALL   SBYTE
MOV     A,#'* '
LCALL   SBYTE
MOV     DPTR,#9F40H
MOV     A,#00H
MOVX    @DPTR,A           ; GO TO ADD ORG
SETB    P1.0
SETB    P1.1
SETB    P1.5
MML:
JB      P3.3,MM
LCALL   CLRRAM
MM:
SJMP    MML

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 MML: ใ้บริการ P1.5 เพื่อการศึกษาเท่านั้น ไม่ควรนำเอกสารไปใช้โดยไม่ได้รับอนุญาต
 MM: ใ้บริการ P3.3, MM และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 MM: SJMP MML

;##### TIME #####

```
TIME:      MOV      SP,#35H
           LCALL   DELAY
           LCALL   RTCINT

           MOV     TMOD,#20H
           MOV     SCON,#52H
           MOV     TH1,#0FDH      ;9600
           SETB   TR1
           CLR    EX0

           LCALL  DELAY
           MOV    DPTR,#TABLE2
           LCALL  SBLOCK
           MOV    DPTR,#TABLE3
           LCALL  SBLOCK
```

```
TIME1:     JB      RI,TIMES      ;CHECK RECEIVE SERIAL
TIME2:     LCALL  RDTIME
           MOV    A,R1
           CLR   C
           SUBB  A,R4
           JZ    TIME2
           MOV   A,R4
           MOV   R1,A
           MOV   DPTR,#TABLE4
           LCALL SBLOCK
           MOV   A,R3
           PUSH  ACC
           MOV   A,R2
           LCALL SBYTEH
           MOV   A,#':'
           LCALL SBYTE
           POP   ACC
           LCALL SBYTEH
           MOV   A,#':'
           LCALL SBYTE
           MOV   A,R4
           LCALL SBYTEH
           LCALL RDDATE
           MOV   DPTR,#TABLE5
           LCALL SBLOCK
           MOV   A,R3
           PUSH  ACC
           MOV   A,R2
           LCALL SBYTEH
           MOV   A,#'/'
           LCALL SBYTE
           POP   ACC
           LCALL SBYTEH
           MOV   A,#'/'
           LCALL SBYTE
           MOV   A,R4
           LCALL SBYTEH
           LCALL SLF
           SJMP  TIME1
```

```
TIMES:     LCALL  RBYTE
           CJNE  A,#'T',TIME1
           LCALL SETUP
           MOV   DPTR,#TABLE1
           LCALL SBLOCK
           SJMP  TIME1
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในการใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังห้ามนำไปตัดแต่งหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TABLE1:      DB      'END OF RTC SETUP THANK YOU...',ODH
TABLE2:      DB      'REAL TIME CLOCK SERVICE...',ODH
TABLE3:      DB      'TO CHANGE TIME PRESS KEY "T"',ODH
TABLE4:      DB      '          TIME..',000H
TABLE5:      DB      '          DATE..',000H

```

```

;***** RDTIME SUB *****

```

```

RDTIME:      MOV      DPTR,#0FE8DH
             MOVX     A,@DPTR
             SETB     ACC.0          ;HOLD=1
             MOVX     @DPTR,A
             MOVX     A,@DPTR       ;READ BUSY
             JNB      ACC.1,RDTIME1
             CLR      ACC.0          ;HOLD=0
             MOVX     @DPTR,A
             SJMP     RDTIME

```

```

RDTIME1:     MOV      DPTR,#0FE80H    ;SECOND
             MOVX     A,@DPTR
             ANL      A,#0FH
             MOV      R0,A
             MOV      DPTR,#0FE81H
             MOVX     A,@DPTR
             ANL      A,#07H
             SWAP     A
             ORL      A,R0
             MOV      R4,A
             MOV      DPTR,#0FE82H    ;MINUTE
             MOVX     A,@DPTR
             ANL      A,#0FH
             MOV      R0,A
             MOV      DPTR,#0FE83H
             MOVX     A,@DPTR
             ANL      A,#07H
             SWAP     A
             ORL      A,R0
             MOV      R3,A
             MOV      DPTR,#0FE84H    ;HOUR
             MOVX     A,@DPTR
             ANL      A,#0FH
             MOV      R0,A
             MOV      DPTR,#0FE85H
             MOVX     A,@DPTR
             ANL      A,#03H          ; 24 HOUR
             SWAP     A
             ORL      A,R0
             MOV      R2,A
             MOV      DPTR,#0FE8DH
             MOVX     A,@DPTR
             CLR      ACC.0          ;HOLD=0
             MOVX     @DPTR,A
             RET

```

```

;***** RDDATE SUB *****

```

```

RDDATE:      MOV      DPTR,#0FE8DH
             MOVX     A,@DPTR
             SETB     ACC.0          ;HOLD=1
             MOVX     @DPTR,A
             MOVX     A,@DPTR       ;READ BUSY
             JNB      ACC.1,RDDATE1
             CLR      ACC.0          ;HOLD=0

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งให้ตัดและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOVX @DPTR, A
SJMP RDDATE
```

```
RDDATE1: MOV DPTR, #0FE86H ; DAY
MOVX A, @DPTR
ANL A, #0FH
MOV R0, A
MOV DPTR, #0FE87H
MOVX A, @DPTR
ANL A, #03H
SWAP A
ORL A, R0
MOV R2, A
```

```
MOV DPTR, #0FE88H ; MONTH
MOVX A, @DPTR
ANL A, #0FH
MOV R0, A
MOV DPTR, #0FE89H
MOVX A, @DPTR
ANL A, #01H
SWAP A
ORL A, R0
MOV R3, A
```

```
MOV DPTR, #0FE8AH ; YEAR
MOVX A, @DPTR
ANL A, #0FH
MOV R0, A
MOV DPTR, #0FE8BH
MOVX A, @DPTR
ANL A, #0FH
SWAP A
ORL A, R0
MOV R4, A
```

```
MOV DPTR, #0FE8CH ; WEEK
MOVX A, @DPTR
ANL A, #07H
MOV R5, A
```

```
MOV DPTR, #0FE8DH
MOVX A, @DPTR
CLR ACC.0 ; HOLD=0
MOVX @DPTR, A
RET
```

```
;***** DELAY *****
```

```
DELAY: MOV R0, #0FFH
DELAY0: MOV R1, #0FFH
DELAY1: DJNZ R1, DELAY1
DJNZ R0, DELAY0
RET
```

```
;***** SETUP *****
```

```
SETUP: MOV DPTR, #SETX1 ; HOUR
LCALL SBLOCK
LCALL RBYTEH
MOV SETBUF, A
LCALL SBYTEH
LCALL SLE
MOV DPTR, #SETX2 ; MINUTE
LCALL SBLOCK
LCALL RBYTEH
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังห้ามนำไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     SETBUF+1,A
LCALL  SBYTEH
LCALL  SLF

```

```

MOV     DPTR,#SETX3
LCALL  SBLOCK
LCALL  RBYTEH
MOV     55H,A
LCALL  SBYTEH
LCALL  SLF
MOV     DPTR,#SETX4
LCALL  SBLOCK
LCALL  RBYTEH
MOV     56H,A
LCALL  SBYTEH
LCALL  SLF
MOV     DPTR,#SETX5
LCALL  SBLOCK
LCALL  RBYTEH
MOV     57H,A
LCALL  SBYTEH
LCALL  SLF

```

```

MOV     R2,SETBUF           ; HOUR
MOV     R3,SETBUF+1       ; MINUTE
MOV     R4,#00H           ; SECOND
LCALL  WRTIME

MOV     R2,55H             ; DATE
MOV     R3,56H             ; MONTH
MOV     R4,57H             ; YEAR
LCALL  WRDATE
RET

```

```

SETX1:  DB 'ENTER SETTING HOUR (XX) :',0000
SETX2:  DB 'ENTER SETTING MINUTE (XX) :',0000
SETX3:  DB 'ENTER SETTING DATE (XX) :',0000
SETX4:  DB 'ENTER SETTING MONTH(XX) :',0000
SETX5:  DB 'ENTER SETTING YEAR (XX) :',0000
;***** RTCINT SUB *****

```

```

RTCINT:  MOV     DPTR,#0FE8FH
MOV     A,#00000011B      ; STOP=1, RESET=1
MOVX   @DPTR,A
ORL    A,#00000100B      ; 24 HOUR
MOVX   @DPTR,A
MOV     DPTR,#0FE8EH
MOV     A,#00000100B      ; SET STND 1SEC
MOVX   @DPTR,A
MOV     DPTR,#0FE8DH
MOV     A,#00000000B      ; HOLD=0
MOVX   @DPTR,A

MOV     DPTR,#0FE8FH
MOVX   A,@DPTR
ANL    A,#00000100B      ; STOP=0, RESET=0
MOVX   @DPTR,A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
;***** WRTIME SUB *****

```

WRTIME:  MOV     DPTR,#0FE8DH
MOVX   A,@DPTR

```

```

SETB ACC.0 ;HOLD=1
MOVX @DPTR,A
MOVX A,@DPTR ;READ BUSY
JNB ACC.1,WRTIME1
CLR ACC.0 ;HOLD=0
MOVX @DPTR,A
SJMP WRTIME

WRTIME1: MOV DPTR,#0FE80H ;SECOND
MOV A,R4
ANL A,#0FH
MOVX @DPTR,A
MOV DPTR,#0FE81H
MOV A,R4
ANL A,#70H
SWAP A
MOVX @DPTR,A

MOV DPTR,#0FE82H ;MINUTE
MOV A,R3
ANL A,#0FH
MOVX @DPTR,A
MOV DPTR,#0FE83H
MOV A,R3
ANL A,#70H
SWAP A
MOVX @DPTR,A

MOV DPTR,#0FE84H ;HOUR
MOV A,R2
ANL A,#0FH
MOVX @DPTR,A
MOV DPTR,#0FE85H
MOV A,R2
ANL A,#30H ; 24 HOUR
SWAP A
MOVX @DPTR,A

MOV DPTR,#0FE8DH
MOVX A,@DPTR
CLR ACC.0 ;HOLD=0
MOVX @DPTR,A
RET

```

;***** WRDATE SUB *****

```

WRDATE: MOV DPTR,#0FE8DH
MOVX A,@DPTR
SETB ACC.0 ;HOLD=1
MOVX @DPTR,A
MOVX A,@DPTR ;READ BUSY
JNB ACC.1,WRDATE1
CLR ACC.0 ;HOLD=0
MOVX @DPTR,A
SJMP WRDATE

```

```

WRDATE1: MOV DPTR,#0FE86H ;DAY
MOV A,R2
ANL A,#0FH
MOVX @DPTR,A
MOV DPTR,#0FE87H
MOV A,R2
ANL A,#30H
SWAP A
MOVX @DPTR,A

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังให้ตัดค่าใช้จ่ายอื่นใด และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     DPTR,#0FE88H      ;MONTH
MOV     A,R3
ANL     A,#0FH
MOVX    @DPTR,A
MOV     DPTR,#0FE89H
MOV     A,R3
ANL     A,#10H
SWAP    A
MOVX    @DPTR,A

```

```

MOV     DPTR,#0FE8AH      ;YEAR
MOV     A,R4
ANL     A,#0FH
MOVX    @DPTR,A
MOV     DPTR,#0FE8BH
MOV     A,R4
ANL     A,#0F0H
SWAP    A
MOVX    @DPTR,A

```

```

MOV     DPTR,#0FE8DH
MOVX    A,@DPTR
CLR     ACC.0              ;HOLD=0
MOVX    @DPTR,A
RET

```

```

; ***** SBYTE SUB ***** 45
; SEND BYTE
; IN = A
; REG = NO

```

```

SBYTE:   JNB     TI,$          ;WAIT FOR SEND OK
         CLR     TI
         MOV     SBUF,A
         RET

```

```

; ***** SBYTEH SUB ***** 46
; SEND 2 BYTE HEX FROM A
; IN = A
; REG = A,R2,R3

```

```

SBYTEH:  LCALL    HTOA
         MOV     A,R2
         LCALL    SBYTE
         MOV     A,R3
         LCALL    SBYTE
         RET

```

```

SBYTEHC: XCH     A,R7          ;R7=CHECKSUM
         ADD     A,R7
         XCH     A,R7
         LCALL    SBYTEH
         RET

```

```

; ***** SBLOCK SUB ***** 48
; SEND BLOCK
; IN = DPTR ROM-ADDRESS (END BY 0 OR 0DH)
; OUT = DPTR (NEXT)
; REG = A,DPTR

```

```

SBLOCK:  CLR     A            ;รับค่าใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
         MOVX   A,@A+DPTR
         CJNE  A,#0,SBLOCK1
         INC   DPTR          ;NEXT ADDRESS
         RET                ;EXIT BY 0

```

```

SBLOCK1:      CJNE   A,#0DH,SBLOCK2
               LCALL  SLF           ;CR/LF
               INC    DPTR          ;NEXT ADDRESS
               RET                ;EXIT BY ODH

SBLOCK2:      LCALL  SBYTE
               INC    DPTR
               SJMP  SBLOCK

```

```

; ***** SLF SUB ***** 50
; PRINT CR/LF
; REG = A

```

```

SLF:          MOV     A,#0DH
               LCALL  SBYTE
               MOV     A,#0AH
               LCALL  SBYTE
               RET

```

```

; ***** RBYTE SUB ***** 51
; OUT = A
; REG = A

```

```

RBYTE:        JNB     RI,$           ;WAIT FOR RECEIVE OK
               CLR     RI
               MOV     A,SBUF
               RET

```

```

; ***** RBYTEH SUB ***** 52
; READ HEX-ASCII FROM SERIAL (2 BYTE)
; OUT = A
; REG = A,R2,R3

```

```

RBYTEH:       LCALL  RBYTE
               MOV   R2,A
               LCALL RBYTE
               MOV   R3,A
               LCALL ATOH
               RET

```

```

RBYTEHC:      LCALL  RBYTEH           ;R7=CHECKSUM
               XCH   A,R7
               ADD   A,R7
               XCH   A,R7
               RET

```

```

; ***** HTOA SUB ***** 60
; CONVERT HEX TO ASCII
; IN = A
; OUT = R2,R3
; REG = A,R2,R3

```

```

HTOA:         PUSH   ACC
               SWAP  A
               LCALL HTOAS
               MOV   R2,A
               POP   ACC
               LCALL HTOAS
               MOV   R3,A
               RET

```

```

HTOAS:        ANL   A,#0FH
               CJNE A,#0AH,$+3
               JNC  HTOAS1
               ORL  A,#30H

```

```
HTOAS1:      RET
              SUBB   A,#9
              ORL    A,#40H
              RET
```

```
; ***** ATOH SUB ***** 61
; ASCII TO HEX CONVERT
; IN  = R2,R3   30H,41H
; OUT = A       0AH
; REG = A,R2
```

```
ATOH:        MOV     A,R2
              LCALL  ATOHS
              SWAP   A
              MOV    R2,A
              MOV    A,R3
              LCALL  ATOHS
              ORL    A,R2
              RET
```

```
ATOHS:       CJNE   A,#'A',S+3
              JC     ATOHS1
              ADD    A,#9
ATOHS1:      ANL    A,#0FH
              RET
```

```
END
```



สรุปผลการทดลองและวิเคราะห์

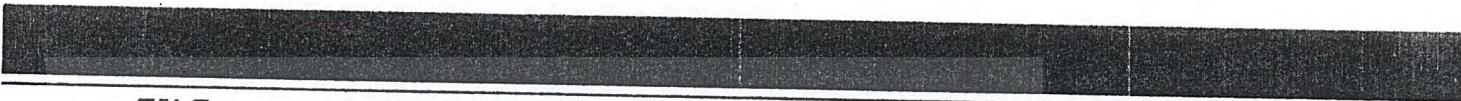
การแซมปลิงสัญญาณ A/D ลงแรมอาจเกิดปัญหา เนื่องจากการค้างค่าไว้ และอาจมีปัญหาเนื่องจากการแปลงข้อมูลของวงจรแปลงสัญญาณเอชดีจะมีผลทำให้ข้อมูลผิดพลาดได้

ในส่วนของระบบไมโครคอนโทรลเลอร์สามารถควบคุมให้มีการตั้งเวลาและทำการเก็บข้อมูลเมื่อถึงเวลาที่กำหนดไว้ ไมโครคอนโทรลเลอร์สามารถแยกมัลติเพล็กซ์ระหว่างแรงดันกับกระแส และสามารถควบคุมการเขียนและอ่านข้อมูลจากแรมได้ จากนั้นนำข้อมูลที่ถูกเก็บภายในแรมส่งเข้าพอร์ตอนุกรมของคอมพิวเตอร์ได้ แต่ในส่วนของข้อมูลที่เข้าเมื่อได้ทำการแปลงสัญญาณที่ได้รับมาแล้วเกิดความผิดพลาดทางข้อมูลจึงไม่สามารถนำมาใช้เป็นข้อมูล ในโปรแกรมของการทำ FFT และแสดงผลได้

ดังนั้นในการทดสอบโปรแกรมการแปลงสัญญาณ FFT จึงทำการจำลองสัญญาณขึ้นในรูปแบบที่แตกต่างกัน และทำการวิเคราะห์สัญญาณที่ได้จากการแปลงโปรแกรมกับสัญญาณทางทฤษฎี แสดงให้เห็นว่าการทำงานของโปรแกรมเป็นไปอย่างถูกต้อง ดังรูปที่ 5.1 แสดงสัญญาณที่จำลองขึ้นเป็นสัญญาณไฟสลับ 50 เฮิร์ตที่มีสัญญาณฮาร์โมนิกส์ที่ 10 รวมอยู่ และเมื่อทำการแปลงเป็นโดเมนความถี่ก็สามารถแสดงให้เห็นถึง ฮาร์โมนิกส์ได้อย่างชัดเจน

ในส่วนของ การ window ได้ทำการจำลองสัญญาณขึ้นเพื่อทดสอบว่า window ชนิดใดเหมาะสมกับสัญญาณหรือข้อมูลแบบไหน การ window จะช่วยให้ทราบถึงค่าที่ถูกต้องมีมากเพียงใด โดยการทดลองและจำลองสัญญาณ sine 50 Hz และทำการ window แบบสี่เหลี่ยมและแบบแฮมมิง (Hamming) แสดงดังรูปที่ 5.2 และรูปที่ 5.3 ซึ่งแสดงให้เห็นว่า Spectrum ของ window แบบแฮมมิงมีความถี่อื่นนอกเหนือจากความถี่ 50 Hz กำเนิดขึ้นซึ่งเกิดจากการที่ Spectrum window แบบแฮมมิงมี bandwidth สูงกว่าแบบ rectangle นั่นเอง ในสัญญาณ sine 45 Hz ใน window แบบแฮมมิงจะมี sidelobe น้อยกว่า window แบบสี่เหลี่ยมดังรูป 5.4 และ 5.5

รูปที่ 5.6 แสดงรูปสัญญาณต่างๆ ไปที่เก็บได้ จะพบว่า มีฮาร์โมนิกส์ต่างๆ ซ่อนอยู่



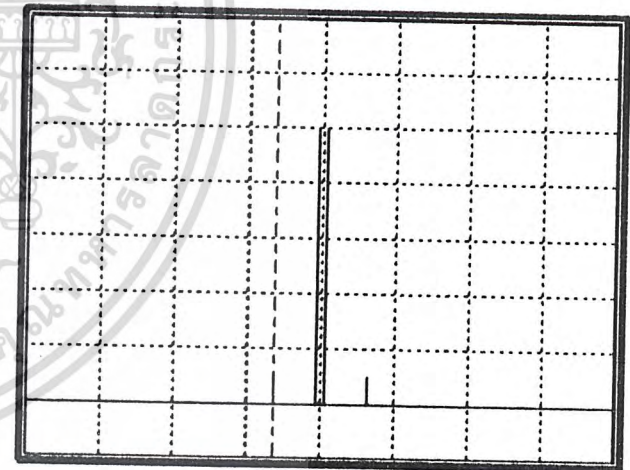
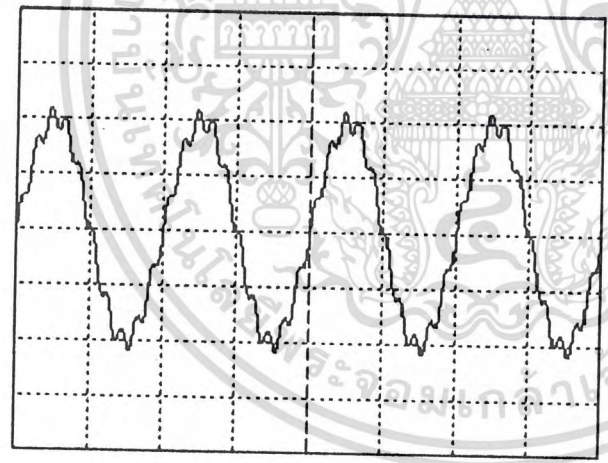
FILE	SHOW GRAPH	SHOW TEXT	OPTION
------	------------	-----------	--------

FILE NAME : SNOISE1.UOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 512
 WINDOW : RECTANGLE
 DATE : 29/12/36 TIME : 12:00

DOMAIN : TIME
 SAMPING NO. : 256
 TIME : 40.00 ms
 AMPLITUDE : -0.00

DOMAIN : FREQUENCY
 HARMONIC : 10
 FREQUENCY: 500.00 Hz
 AMPLITUDE: 10.0000

5.1



SCALE X: 10.00ms/Div Y: 100.00V/Div
 STEP : 1

X: 800 Hz/Div Y: 20.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu



FILE

SHOW GRAPH

SHOW TEXT

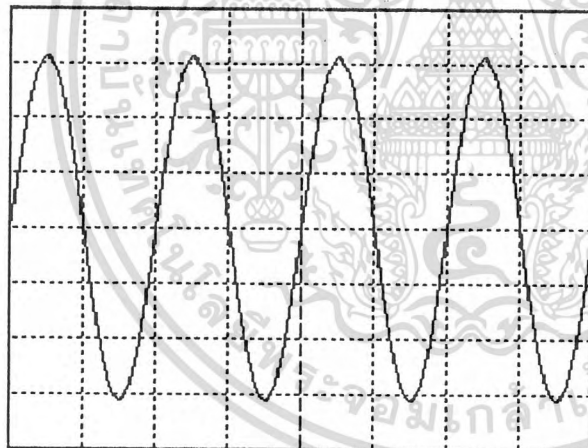
OPTION

FILE NAME : SIN50.VOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 512
 WINDOW : RECTANGLE
 DATE : 29/12/36 TIME : 1:00

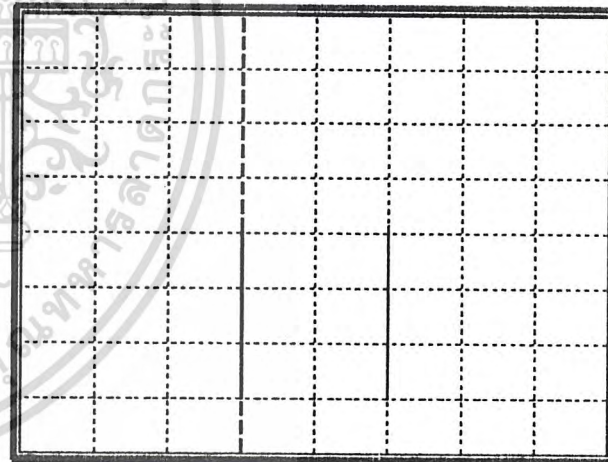
DOMAIN : TIME
 SAMPING NO. : 256
 TIME : 40.00 ms
 AMPLITUDE : -0.00

DOMAIN : FREQUENCY
 HARMONIC : 1 (FUNDAMENTAL)
 FREQUENCY: 50.00 Hz
 AMPLITUDE: 155.5635

slm 5.2



SCALE X: 10.00ms/Div Y: 100.00V/Div
 STEP : 1



X: 50 Hz/Div Y: 50.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu



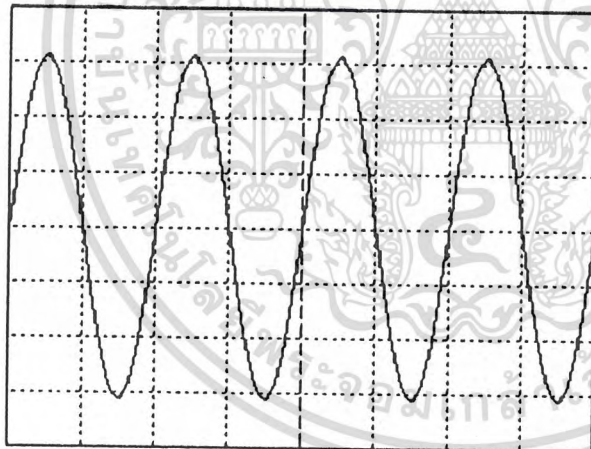
FILE SHOW GRAPH SHOW TEXT OPTION

FILE NAME : SINHAM.UOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 512
 WINDOW : HAMMING
 DATE : 29/12/36 TIME : 1:00

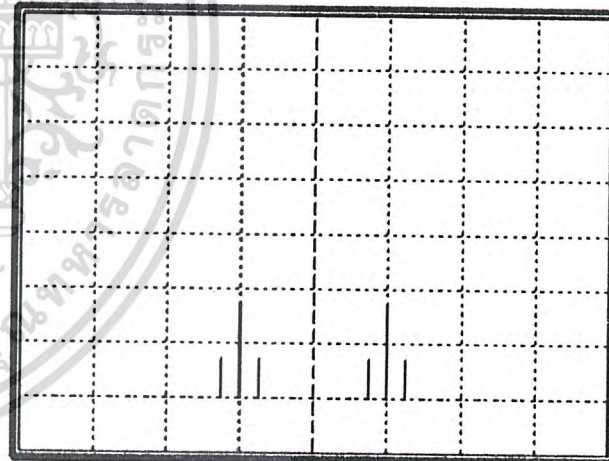
DOMAIN : TIME
 SAMPING NO. : 256
 TIME : 40.00 ms
 AMPLITUDE : -0.00

DOMAIN : FREQUENCY
 HARMONIC : DC
 FREQUENCY: 0.00 Hz
 AMPLITUDE: 0.0005

รูป 5.3

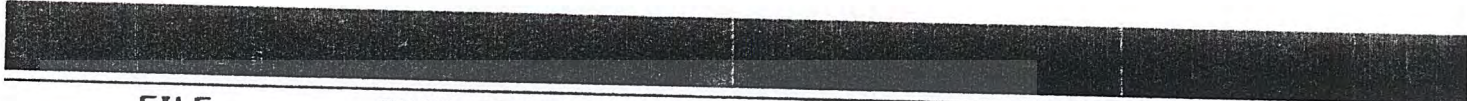


SCALE X: 10.00ms/Div Y: 100.00V/Div
 STEP : 1



X: 50 Hz/Div Y: 50.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu



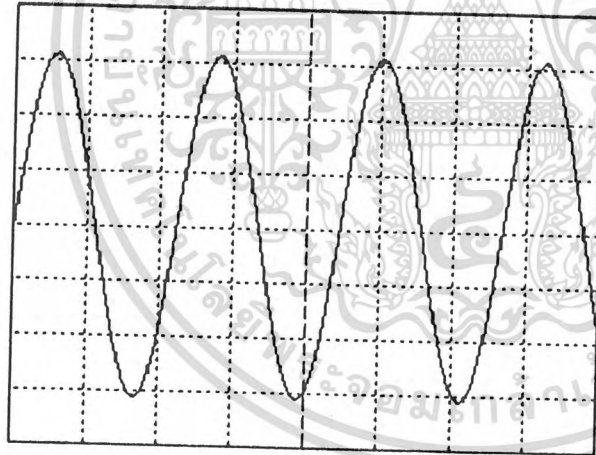
FILE SHOW GRAPH SHOW TEXT OPTION

FILE NAME : SIN45.VOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 512
 WINDOW : RECTANGLE
 DATE : 29/12/36 TIME : 1:00

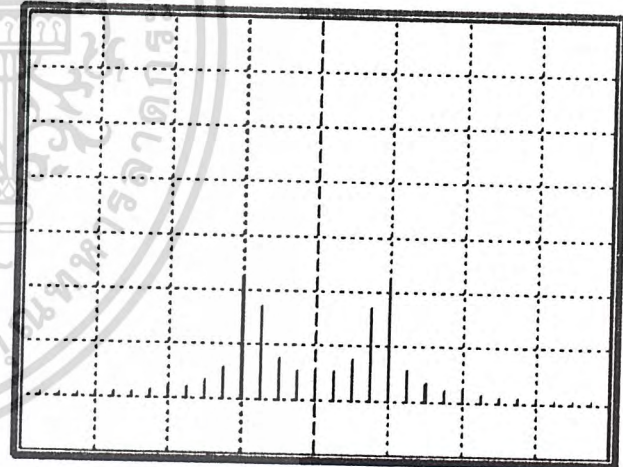
DOMAIN : TIME
 SAMPING NO. : 256
 TIME : 40.00 ms
 AMPLITUDE : -295.90

DOMAIN : FREQUENCY
 HARMONIC : DC
 FREQUENCY: 0.00 Hz
 AMPLITUDE: 25.0573

5.4



SCALE X: 10.00ms/Div Y: 100.00V/Div
 STEP : 1



X: 50 Hz/Div Y: 50.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu

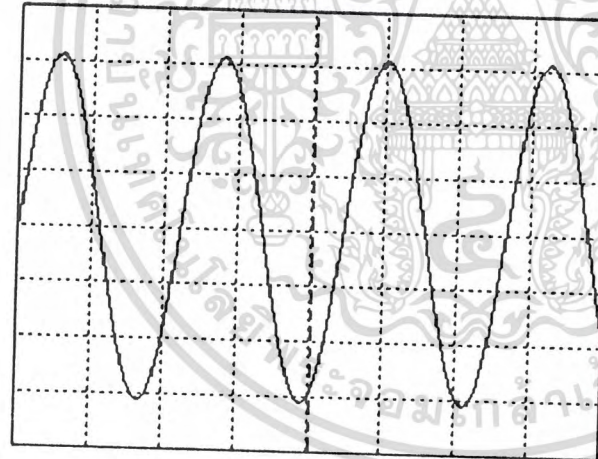
FILE SHOW GRAPH SHOW TEXT OPTION

FILE NAME : SIN45HAM.UOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 512
 WINDOW : HAMMING
 DATE : 29/12/36 TIME : 1:00

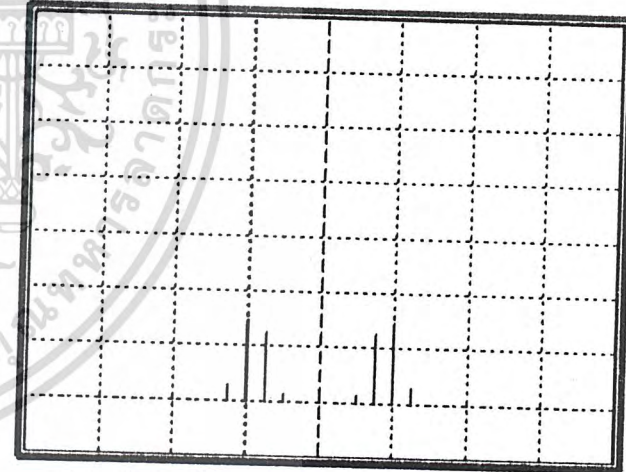
DOMAIN : TIME
 SAMPING NO. : 257
 TIME : 40.16 ms
 AMPLITUDE : -291.36

DOMAIN : FREQUENCY
 HARMONIC : DC
 FREQUENCY: 0.00 Hz
 AMPLITUDE: 1.0301

5.5



SCALE X: 10.00ms/Div Y: 100.00V/Div
 STEP : 1



X: 50 Hz/Div Y: 50.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu



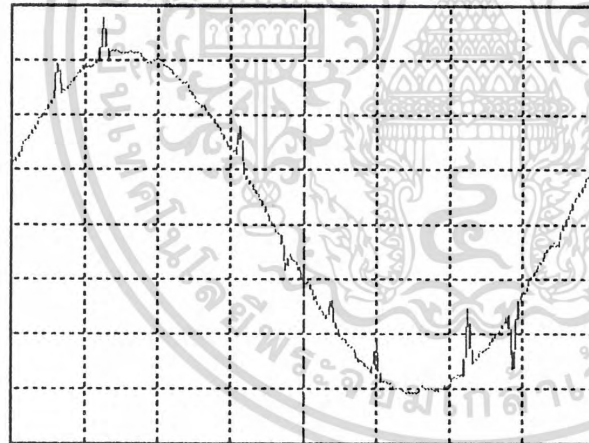
FILE SHOW GRAPH SHOW TEXT OPTION

FILE NAME : OLD.VOL
 SIGNAL : VOLTAGE
 SAMPING RATE : 6.4KHz SAMPING NUMBER : 128
 WINDOW : RECTANGLE
 DATE : 29/12/36 TIME : 12:00

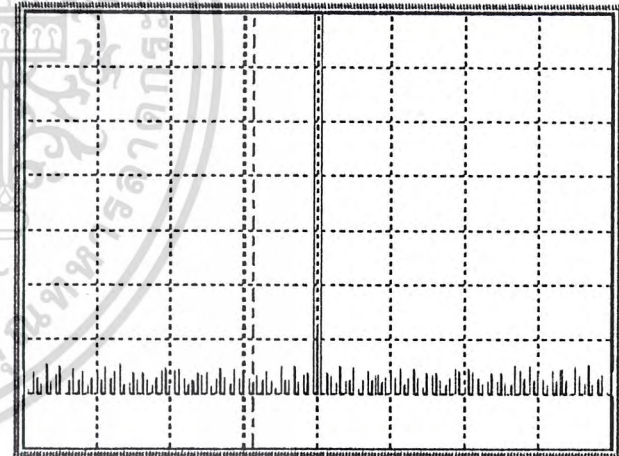
DOMAIN : TIME
 SAMPING NO. : 64
 TIME : 10.00 ms
 AMPLITUDE : -103.76

DOMAIN : FREQUENCY
 HARMONIC : 14
 FREQUENCY : 700.00 Hz
 AMPLITUDE : 1.0579

5.6

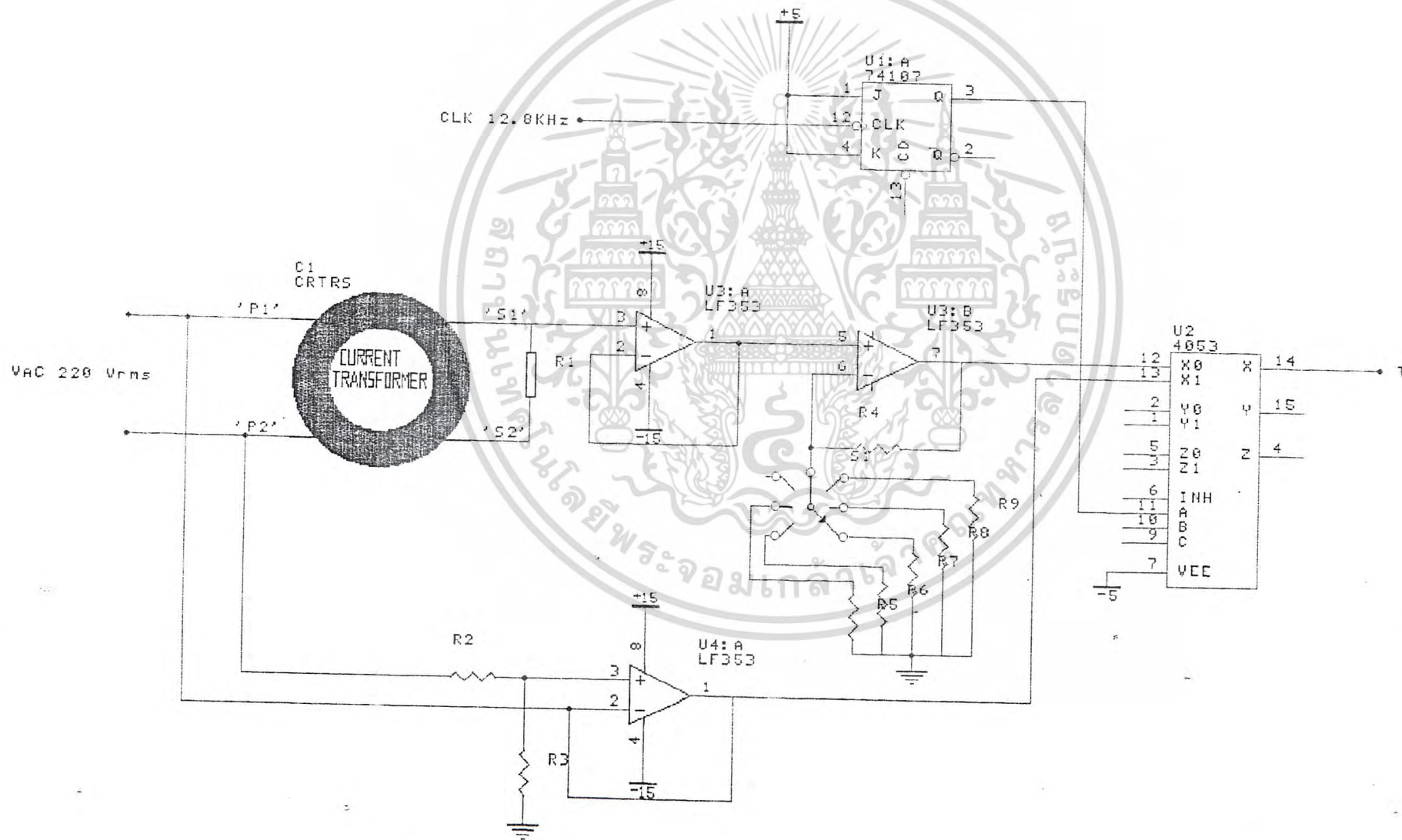


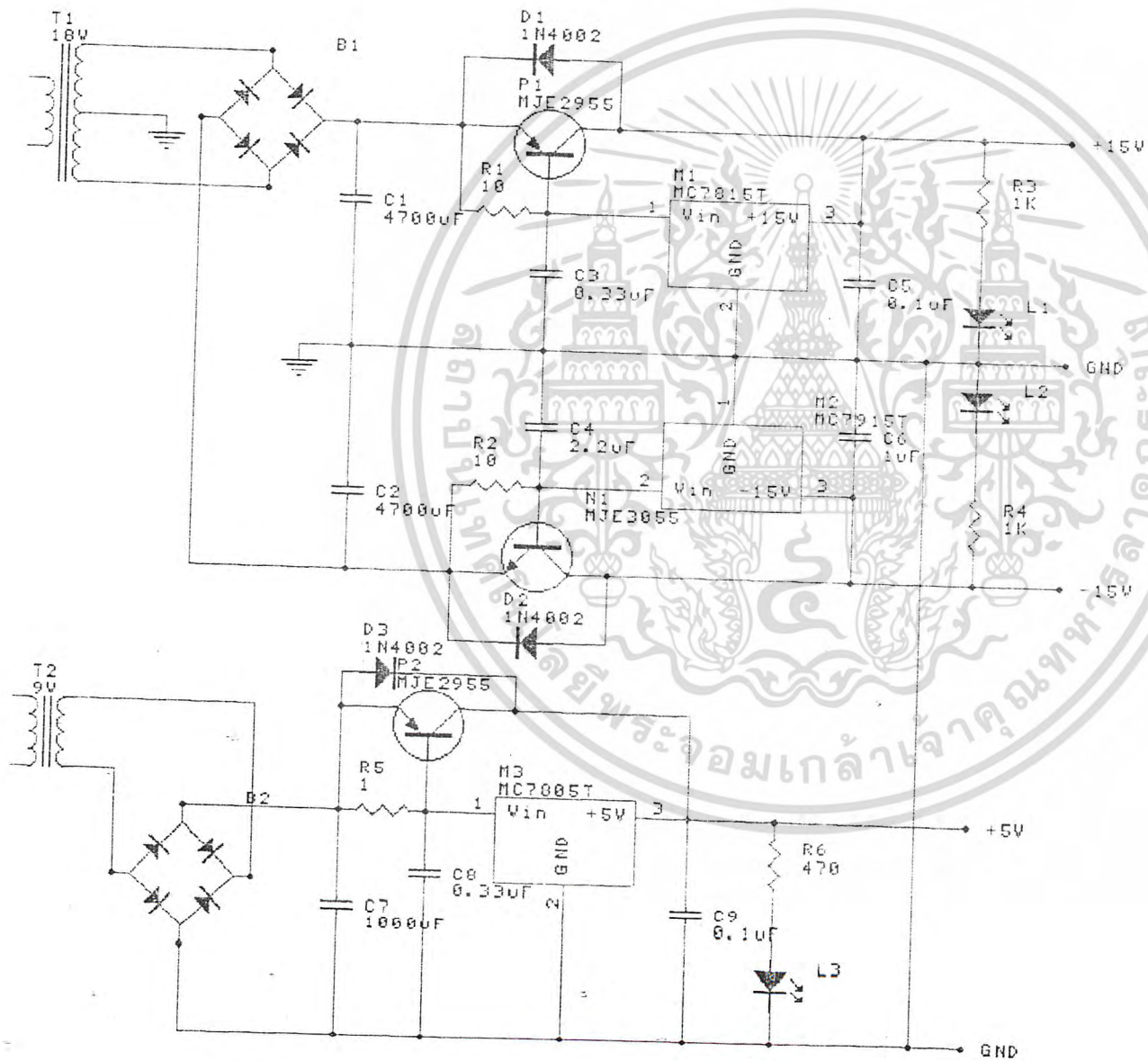
SCALE X: 2.50 ms/Div Y: 100.00V/Div
 STEP : 1

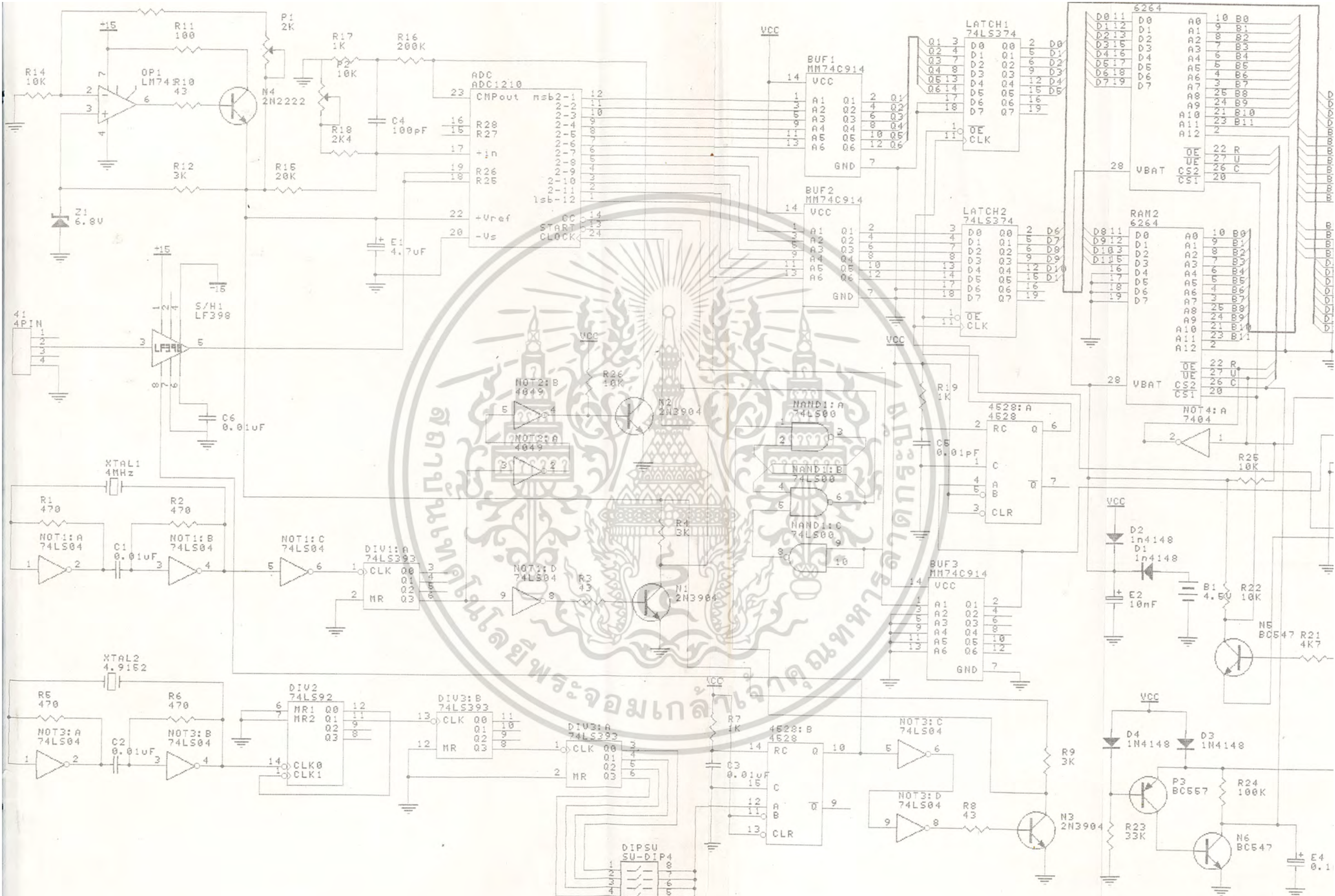


X: 800 Hz/Div Y: 5.00 %/Div
 STEP : 1

<F1>Switch <F2-F3>Scale X-Y <F4>All Graph <PgUp-PgDw>StepUp-Down <ESC>Menu







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่อนุญาตให้นำไปเผยแพร่ หรือใช้เพื่อวัตถุประสงค์อื่นใด และสงวนลิขสิทธิ์ในเอกสารฉบับนี้

บรรณานุกรม

ก. เอกสารอ้างอิงที่เป็นวารสารภาษาอังกฤษ

1. A.Norman Mortensen and Gary L. Johnson, "A Power System Digital Harmonic Analyzer", IEEE TRANSACTIONS ON INSTRUMENT AND MEASUREMENT, VOL. 37, NO. 4, DECEMBER 1988.

ข. เอกสารอ้างอิงที่เป็นหนังสือภาษาไทยและภาษาอังกฤษ

1. ศ.ดร. วิถกต สุระกำพลธร, "การประมวลผลสัญญาณเชิงเลข", บริษัท ไคนาทริน จก. 348 หน้า, 2532. [1]
2. Robert W. Ramirez, "The FFT Fundamentals and Concepts", Prentice-Hall, Inc., 177 Pages, 1985. [2]
3. Robert D. Strum, Donald E. Kirt, "Discrete systems and Digital Signal Processing", Addison Wesley Publishing Company, [3]
4. Paul M. Embree and Bruce Kimble, "C Language Algorithms for Digital Signal Processing", Prentice-Hall International, Inc., 456 Pages.