

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

ปริญญาโท เครื่องวิเคราะห์แถบความถี่

SPECTRUM ANALYZER

- ชื่อนักศึกษา
1. นายฉัตรชัย วงศ์เด่นวงศ์ รหัสประจำตัว 39031305
 2. นายชุมพล จิตรคล รหัสประจำตัว 39031306
 3. นายผดุง ขวัญอินอินทร์ รหัสประจำตัว 39031318
 4. นางสาวอรพิน สัตตบุตร รหัสประจำตัว 39031339

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญาโท

1. อาจารย์วรวิทย์ สมหา
2. อาจารย์อำพล ทองระอา
3. อาจารย์สุรพงษ์ สิริพงศ์ดี



คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์วรวิทย์ สมหา	
2. อาจารย์อำพล ทองระอา	
3. อาจารย์สุรพงษ์ สิริพงศ์ดี	
4. อาจารย์กิตติพงศ์ มะโน	
5. อาจารย์ปิยะ จิตธรรมมาภิรมย์	

วันเดือนปีที่สอบ วันที่ 14 ธันวาคม เวลา 19.00 น. ถึง 20.00 น.

สถานที่สอบ ห้อง ค. 310 คณะครุศาสตร์อุตสาหกรรม

เลขที่.....
เลขทะเบียน..... 30155
วัน, เดือน, ปี..... ๒๕๕๘ ส.ย. 2541



ภาควิชาวิศวกรรม

(ผศ. ดร. ชุมพล เทพหัสดิน ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่.....เดือน.....พ.ศ. ๕๙

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตร
เครื่องวิเคราะห์แถบความถี่
SPECTRUM ANALYZER



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องวิเคราะห์แถบความถี่
SPECTRUM ANALYZER

ผู้จัดทำ

- | | |
|----------------|---------------|
| 1. นายจักรชัย | วงศ์เค่นวงศ์ |
| 2. นายชุมพล | จิตรคต |
| 3. นายศุภ | ขวัญอินอินทร์ |
| 4. นางสาวอรพิน | สัตตบุตร |

อาจารย์ที่ปรึกษา

ลงนาม.....
(อาจารย์วรวิทย์ สมหา)

ลงนาม.....
(อาจารย์อำพล ทองระอา)

ลงนาม.....
(อาจารย์สุรพงษ์ สิริพงศ์ดี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....
(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

ปริญญานิพนธ์

เรื่อง เครื่องวิเคราะห์แถบความถี่
SPECTRUM ANALYZER

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของระบบการประมวลผลสัญญาณเชิงเลข
2. เพื่อออกแบบเครื่องวิเคราะห์แถบความถี่โดยใช้ TMS320C50
3. เพื่อศึกษาการเขียนโปรแกรมบนชิพ TMS320C50 และบนไมโครคอมพิวเตอร์
4. เพื่อสร้างเครื่องวิเคราะห์แถบความถี่ ที่มีย่านความถี่กว้างขึ้นโดยใช้ TMS320C50
5. เพื่อนำเครื่องวิเคราะห์แถบความถี่ไปใช้งานร่วมกับไมโครคอมพิวเตอร์ได้

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำระบบประมวลผลสัญญาณเชิงเลขมาประยุกต์ใช้งานได้
2. สามารถออกแบบเครื่องวิเคราะห์แถบความถี่ โดยใช้ TMS320C50 ได้
3. สามารถสร้างเครื่องวิเคราะห์แถบความถี่ โดยใช้ TMS320C50 ได้
4. สามารถเขียนโปรแกรมควบคุมการทำงานของเครื่องวิเคราะห์แถบความถี่ได้
5. สามารถนำเครื่องวิเคราะห์แถบความถี่ ไปใช้งานร่วมกับไมโครคอมพิวเตอร์ได้

เครื่องวิเคราะห์แถบความถี่

นายฉัตรชัย	วงศ์เด่นวงศ์
นายชุมพล	จิตรกล
นายศकु	ขวัญอันอินทร์
นางสาวอรพิน	สัจตบุตร

อาจารย์ที่ปรึกษา	
อาจารย์วรวิทย์	สมหา
อาจารย์อำพล	ทองระอา
อาจารย์สุรพงษ์	สิริพงศ์ดี
ปีการศึกษา 2540	

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้นำเสนอการออกแบบ และการสร้างเครื่องวิเคราะห์แถบความถี่ โดยใช้ชิพประมวลผลสัญญาณเชิงเลข เบอร์ TMS320C50 (DSP:Digital Signal Processing) เครื่องวิเคราะห์แถบความถี่สามารถวิเคราะห์แถบความถี่ได้ในย่านความถี่(1.33 kHz ถึง 2 MHz) และแสดงผลสัญญาณที่ได้จากการวิเคราะห์ออกทางหน้าจอคอมพิวเตอร์ ผ่านทางพอร์ตขนาน โดยแสดงขนาดของ แร่งคั่น และความถี่ของสัญญาณเป็นกราฟสเปกตรัมและตัวอักษร เพื่อความเที่ยงตรง และความสะดวกในการอ่านค่าต่างๆ พร้อมทั้งฟังก์ชันการใช้งานอื่นๆ เช่น การหยุดภาพ หรือ การบันทึกภาพสัญญาณ เป็นต้น การแสดงผลของสัญญาณจะเป็นแบบเวลาจริง (Real Time)

SPECTRUM ANALYZER

MR.CHATCHAI	WONGDENWONG
MR.CHUMPON	JITDON
MR.PHADUNG	KHANONIN
MISS.ORAPHIN	SUTTABUT

ADVISORS

MR.WORAWIT	SOMHA
MR.AMPHON	THONGRAAR
MR.SURAPHONG	SIRIPHONGDEE

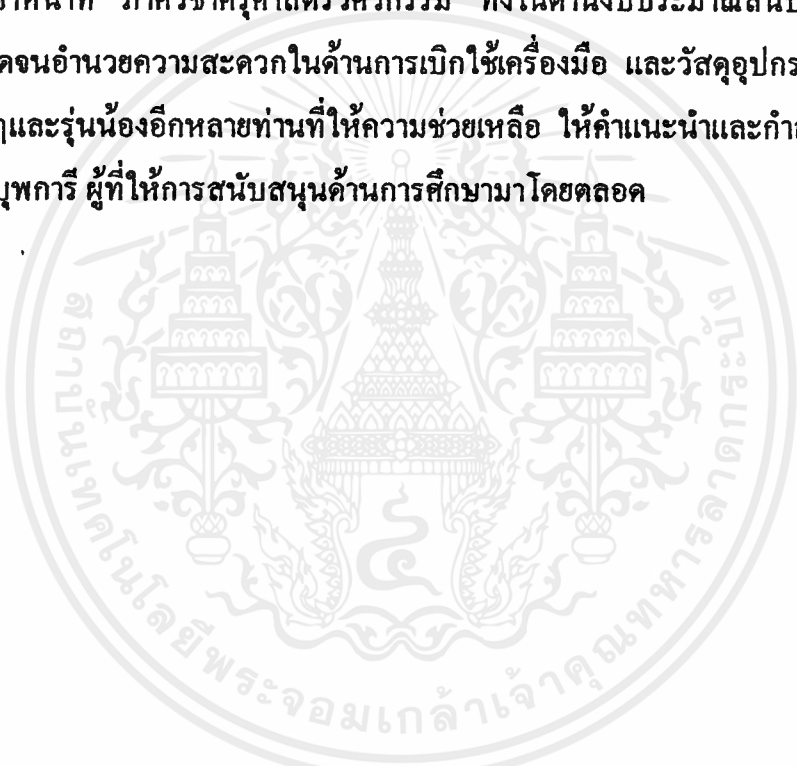
1997

ABSTRACT

This thesis presents a designing and creation of Spectrum Analyzer Using TMS320C50 (DSP:Digital Signal Processor) chip. This Spectrum Analyzer can analyze high frequency spectrum (1.33 kHz ถึง 2 MHz.) and display the analyzed signal on computer with transmission and receive parallel port. Both amplitude and frequency are displayed in spectrum graph and text for accuracy and comfortably in value reading, and others function for using such as Break, Load and Save. Signal is process in real time.

กิตติกรรมประกาศ

การจัดทำปริณิงานิพนธ์นี้สามารถสำเร็จลุล่วงด้วยดี จากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน โดยให้ความร่วมมือกันอย่างเต็มที่ นอกจากนี้ ยังได้รับความกรุณาจาก ท่านอาจารย์ วรวิทย์ สมหา ในด้านแนวความคิด และคำปรึกษา คุณสุวิทย์ ชิบมันตะศิริ ที่เสียสละเวลาให้คำแนะนำ อุปกรณ์และสถานที่ทดลอง นอกจากนี้ยังได้รับการสนับสนุนจากท่านอาจารย์ และเจ้าหน้าที่ ภาควิชาครุศาสตร์วิศวกรรม ทั้งในด้านงบประมาณสนับสนุนการทำโครงการ ตลอดจนอำนวยความสะดวกในด้านการเบิกใช้เครื่องมือ และวัสดุอุปกรณ์ต่างๆ อีกทั้งรุ่นพี่ เพื่อนๆ และรุ่นน้องอีกหลายท่านที่ให้ความช่วยเหลือ ให้คำแนะนำและกำลังใจ และที่สำคัญที่สุดคือ บุพการี ผู้ที่ให้การสนับสนุนด้านการศึกษาโดยตลอด



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปริญญาานิพนธ์	1
1.2 ซึ่คความสามารถของ โครงการงาน	3
1.3 เนื้อหาโดยสังเขป	4
บทที่ 2 ทฤษฎีและหลักการ	6
2.1 บทนำ	6
2.2 ลักษณะ โครงสร้างของ TMS320C50	7
2.3 สถาปัตยกรรมของ TMS320C50 ซึ่พ DSP โดยทั่วไป	8
2.4 ตัวประมวลผลกลาง	8
2.5 ษาของ TMS320C50 DSP	10
2.6 การจัดหน่วยความจำ	10
2.7 สรุปลสถาปัตยกรรมโดยทั่วไป	12
2.8 ทฤษฎีการแปลงสัญญาณ	14
2.9 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	15
2.9.1 A/D แบบเชอร์ไว	15
2.9.2 A/D แบบคูอัลต-สโโลป อินทีเกรต	15
2.9.3 A/D แบบซัคเซสซีฟ-แอฟพรีอกซิเมชัน	16
2.9.4 A/D แบบคอมพาราเตอร์ขนาน	16
2.10 ทฤษฎีการสุ่มข้อมูล	16

สารบัญ(ต่อ)

เรื่อง	หน้า
2.11 ทฤษฎีการสร้างภาพดิจิทัลบนจอคอมพิวเตอร์	16
2.11.1 หลักการสร้างภาพบนจอคอมพิวเตอร์	17
2.11.2 โหมดของจอภาพ	18
2.11.3 วิดีโอแรม	20
2.11.4 โหมดภาพความละเอียดสูง	20
2.12 การแปลงพิกัดฟูรีเยร์	23
2.13 การให้น้ำหนัก	34
2.14 การสื่อสารข้อมูลแบบอนุกรม	35
2.14.1 คุณสมบัติของ RS-232	35
2.14.2 การจักษาสัญญาณของ RS-232	36
2.14.3 การเชื่อมต่อสัญญาณของ RS-232	36
บทที่ 3 หลักการออกแบบ ฮาร์ดแวร์และซอฟต์แวร์	40
3.1 หลักการออกแบบฮาร์ดแวร์	40
3.1.1 วงจรแลตซ์ หรือส่วนการอินเตอร์เฟสกับคอมพิวเตอร์	43
3.1.2 วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล	44
3.2 การออกแบบทางด้านซอฟต์แวร์	46
3.2.1 การออกแบบ และการแปลงซอฟต์แวร์	
วิเคราะห์แถบความถี่	46
3.2.2 การออกแบบซอฟต์แวร์การแสดงผล	48
บทที่ 4 การทดลอง และผลการทดลอง	52
4.1 การทดสอบการทำงานของเครื่องวิเคราะห์แถบความถี่	
โดยใช้ TMS320C50	52
4.1.1 ลำดับขั้นตอนการทดลอง	52
4.1.2 ผลการทดลอง	58

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	59
5.1 บทสรุป	59
5.2 ปัญหาและแนวทางแก้ไข	59
5.2.1 ในส่วนของฮาร์ดแวร์	59
5.2.2 ในส่วนของซอฟต์แวร์	60
5.3 ประโยชน์ที่ได้รับ	60
5.4 แนวทางการพัฒนา	61
ภาคผนวก ก วงจรเครื่องวิเคราะห์แถบความถี่	62
ภาคผนวก ข รายละเอียดของอุปกรณ์ที่ใช้ในโครงการ	69
ภาคผนวก ค ผังการทำงาน และโปรแกรมที่ใช้ในโครงการ	88
ภาคผนวก ง คู่มือการใช้งาน	126
บรรณานุกรม	135
ประวัติผู้แต่ง	136

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 รีจิสเตอร์ภายใน CPU	9
ตารางที่ 2.2 โหมดการแสดงผลภาพ	19
ตารางที่ 2.3 ขาสัญญาณของ RS-232 ทั้งแบบ 9 ขา และ 25 ขา	36
ตารางที่ 2.4 อัตราบอดและช่วงเวลาของแต่ละบิตข้อมูล ในการสื่อสารข้อมูลอนุกรม	38



สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 1.1 ลักษณะการใช้คอมพิวเตอร์ควบคุมการทำงานของบอร์ด DSK	3
รูปที่ 1.2 ผังการทำงานของสเปคตรัมอะนาไลเซอร์โดยใช้ TMS320C50	4
รูปที่ 2.1 ลักษณะภายนอกของ TMS320C50	11
รูปที่ 2.2 การจัดหน่วยความจำของ TMS320C50	12
รูปที่ 2.3 ระบบที่มีการประมวลผลข้อมูลทางดิจิทัล	14
รูปที่ 2.4 ขบวนการประมวลผล	17
รูปที่ 2.5 จอภาพ (Monitors) ตัวอักษรหรือรูปภาพ	17
รูปที่ 2.6 โครงสร้างของสื่อบนจอภาพในโหมด VGA	21
รูปที่ 2.7 โครงสร้างการทำงานของการ์ด SuperVGA	22
รูปที่ 2.8 กราฟการไหล วิธีการคำนวณตามสมการ(2.12)	29
รูปที่ 2.9 หน่วยมีสื่อของการคำนวณตามขั้นตอนวิธีการลดทอนเวลา	31
รูปที่ 2.10 กราฟการไหลของสัญญาณ 8 จุด	32
รูปที่ 2.11 DIT สำหรับ FFT แบบ 8 จุด	33
DIT สำหรับ DFT แบบ 8 จุด	33
รูปที่ 2.12 การสลับตำแหน่งของลำดับ $x(n)$ ด้วยการผันกลับบิต	34
รูปที่ 2.13 การส่งข้อมูลแบบอนุกรมจำนวน 8 บิต	
จะส่งทีละบิตจนครบ 1 ไบต์	35
รูปที่ 2.14 การต่ออุปกรณ์ DTE เข้ากับ DCE	37
รูปที่ 2.15 การต่ออุปกรณ์ DTE เข้ากับ DTE	37
รูปที่ 2.16 รูปแบบของสัญญาณข้อมูลอนุกรม	
ที่ใช้ในการสื่อสารข้อมูลอนุกรม	39
รูปที่ 3.1 ผังการทำงานของสเปคตรัมอะนาไลเซอร์	41
รูปที่ 3.2 ลักษณะของบอร์ด DSK	42
รูปที่ 3.3 บล็อกโคอะแกรมของบอร์ด DSK	42

สารบัญรูปภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.4 วงจรแลตซ์	43
รูปที่ 3.5 ผังการทำงานภายในของไอซี CA3318	44
รูปที่ 3.6 วงจรแปลงสัญญาณแอนาล็อกเป็นสัญญาณดิจิทัล	45
รูปที่ 3.7 วงจรภายในของเครื่องวิเคราะห์แถบความถี่	46
รูปที่ 3.8 ผังการทำงานของโปรแกรมภาษาแอสเซมบลี	47
รูปที่ 3.9 ขั้นตอนการแปลงโปรแกรมภาษาแอสเซมบลี DSK	48
รูปที่ 3.10 ผังการทำงานของโปรแกรมภาษาซี	50
รูปที่ 3.11 ผังการทำงานทั้งหมดของโปรแกรมของเครื่องวิเคราะห์แถบความถี่	51
รูปที่ 4.1 เครื่องวิเคราะห์แถบความถี่โดยใช้ TMS320C50	52
รูปที่ 4.2 การต่อเครื่องวิเคราะห์แถบความถี่ใช้งานร่วมกับคอมพิวเตอร์	53
รูปที่ 4.3 หน้าจอคอมพิวเตอร์	53
รูปที่ 4.4 สัญญาณอินพุตรูปคลื่นไซน์ขนาด 4 Vp-p ที่ความถี่ 1 MHz.	54
รูปที่ 4.5 สเปกตรัมของสัญญาณรูปคลื่นไซน์ ขนาด 4 Vp-p ที่ความถี่ 1 MHz	54
รูปที่ 4.6 สัญญาณอินพุตรูปคลื่นสี่เหลี่ยมขนาด 4 Vp-p ที่ความถี่ 1 MHz.	55
รูปที่ 4.7 สเปกตรัมของสัญญาณรูปคลื่นสี่เหลี่ยม ขนาด 4 Vp-p ที่ความถี่ 1 MHz	56
รูปที่ 4.8 สัญญาณอินพุตรูปคลื่นฟันเลื่อยขนาด 4 Vp-p ที่ความถี่ 1 MHz.	57
รูปที่ 4.9 สเปกตรัมของสัญญาณรูปคลื่นฟันเลื่อย ขนาด 4 Vp-p ที่ความถี่ 1 MHz	57

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญญาประดิษฐ์

ในปัจจุบันความเจริญก้าวหน้าทางเทคโนโลยี ในการผลิตวงจรรวมขนาดใหญ่ มีประสิทธิภาพสูงยิ่งขึ้น โดยเฉพาะสารกึ่งตัวนำประเภท MOS ตลอดจน ข้อดีของการผลิตวงจรรวม VLSI ด้วย วงจรสวิตซ์คาปาซิเตอร์ และประกอบกับวิธีการ ซึ่งเป็นตัวกำหนดแนวทางในการแก้ปัญหา ได้อย่างมีประสิทธิภาพนั้น มีการพัฒนาทางด้านดิจิทัลเข้ามาแทนด้านแอนะล็อก ที่มีใช้อยู่เดิมมากขึ้น ทั้งนี้เพราะเทคโนโลยีทางด้านดิจิทัล ที่นำมาใช้ในการประมวลผลของสัญญาณ มีข้อดีหลายๆ อย่าง เช่น ความเชื่อถือได้, ความถูกต้อง และสามารถทำการประมวลผลสัญญาณได้ด้วยวิธีการทั้งทางด้านฮาร์ดแวร์ และด้านซอฟต์แวร์ ทำให้มีความสะดวกในการเก็บ และถ่ายโอนข้อมูล (Storing and Transferring) และมีความยืดหยุ่นต่อการใช้งานสูง

อุปกรณ์ประมวลผลสัญญาณเชิงเลข (Digital Signal Processing: DSP) มีการประยุกต์ใช้งานได้อย่างกว้างขวาง เช่น

1. ฟิสิกส์, อะแคปติฟฟิสิกส์, อิควอลไลเซอร์ และตัวเลื่อนเฟส
2. มิกเซอร์, มอดูเลเตอร์ และตัวปรับเทียบเฟส
3. วงจรกำเนิดสัญญาณ, ออสซิลเลเตอร์ปรับค่าได้ และแหล่งสัญญาณรบกวน
4. ฟาสต์ฟูเรียร์ทรานส์ฟอร์ม (Fast Fourier Transform)
5. อุปกรณ์ไม่เป็นเชิงเส้น, ลิมิเตอร์ และคอมพาราเตอร์
6. การควบคุม, วงจรควบคุมแบบต่างๆ และเซอร์โว
7. วงจรประมวลผลสัญญาณภาพและเสียงพูด.
8. วงจรวิเคราะห์สเปกตรัม (Spectrum Analyzer)
9. ระบบสื่อสารข้อมูลต่างๆ

การประมวลผลสัญญาณเชิงเลขเป็นขบวนการที่ใช้ความรู้ทางด้านคณิตศาสตร์ขั้นสูง มาอธิบายการประมวลผลสัญญาณต่างๆ ซึ่งมีทั้งแบบเป็นเชิงเส้น และแบบ ไม่เป็นเชิงเส้น เพื่อให้ได้สัญญาณเอาต์พุตตามต้องการ ด้วยกระบวนการทางดิจิทัล หรือเชิงเลข โดยการใช้การประมวลผลจากไมโคร โปรเซสเซอร์ที่ใช้ความเร็วสูง

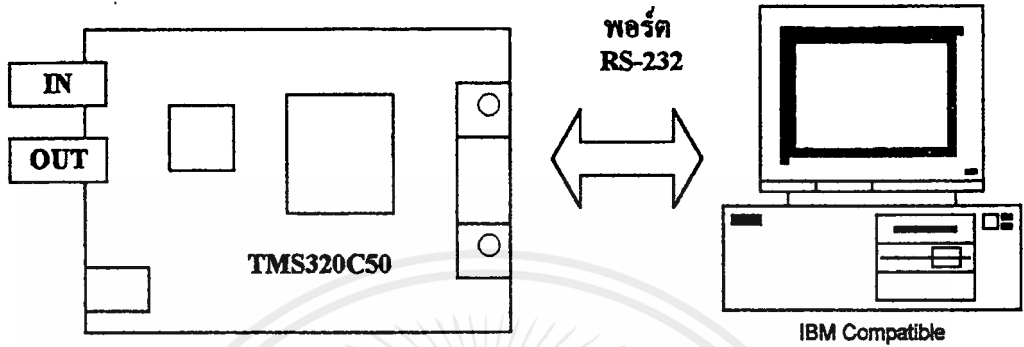
ในอดีต การศึกษาเรื่องของ DSP จะเป็นเรื่องที่ยาก และลำบากมาก เนื่องจากต้องหาตัวชิพ DSP คู่มือ ซอฟต์แวร์แอสเซมเบลอร์และดีบั๊กเกอร์ (Assembler and Debugger) ฯลฯ แต่ในปัจจุบันไม่เป็นเรื่องยาก เมื่อทางบริษัทเท็กซัสอินสตรูเมนต์ เจ้าของชิพ DSP ตระกูล TMS320CXX ได้จัดทำชุดพัฒนาที่สมบูรณ์แบบ มีตัวชิพ คู่มือ และอุปกรณ์ต่างๆ ทั้งหมด พร้อมทั้งซอฟต์แวร์และโปรแกรมตัวอย่าง จัดจำหน่ายในราคาที่ถูกลง เพื่อต้องการให้มีการใช้งานอย่างแพร่หลาย ในปัจจุบันได้มีผู้นำเข้าชุดพัฒนา ระบบประมวลผลสัญญาณเชิงเลข TMS320C50 DSP Start Kit (DSK) ของเท็กซัสอินสตรูเมนต์เข้ามาจำหน่ายให้นักศึกษานักออกแบบ ได้ทดลองศึกษาใช้งานเป็นการพัฒนาชิพ DSP ในประเทศไทย

TMS320C50 ซึ่งใช้สถาปัตยกรรมฮาร์ดแวร์ซึ่งแยกแยะระหว่างโปรแกรม และข้อมูลออกจากกัน จึงทำให้จังหวะการอ่านคำสั่ง และการประมวลผลคำสั่ง (Fetch and Execute) ทำพร้อมกันได้ แต่มีการคิดแปลงจากสถาปัตยกรรมฮาร์ดแวร์ให้เกิดการถ่ายเทข้อมูลระหว่างโปรแกรม และข้อมูลได้ จึงทำให้มีการทำงานได้คล่องตัวยิ่งขึ้น และความเร็วของการทำคำสั่งทางคณิตศาสตร์ สามารถกระทำได้รวดเร็วขึ้น โดยมีเงื่อนไขของชิพ DSP ที่ออกแบบไว้คือ

1. สถาปัตยกรรมเน้นการทำงานด้านฮาร์ดแวร์
2. มีกลไกของฮาร์ดแวร์ การคูณ โดยเฉพาะ
3. มีคำสั่งที่ใช้งานการประมวลผลสัญญาณโดยเฉพาะ
4. ใช้หลักการของไปป์ไลน์นี้

ด้วยหลักการดังกล่าวจึงทำให้การประมวลผลสัญญาณเชิงเลขสามารถใช้ข้อมูลได้เป็นจำนวนมาก และการทำงานส่วนใหญ่ในแต่ละคำสั่งจะเสร็จสิ้นในรอบเดียว

จากชุดพัฒนา DSK สามารถต่อใช้งานแทนระบบแอนาลอก ได้อย่างกว้างขวาง ซึ่งใช้โปรเซสเซอร์ประมวลผลสัญญาณดิจิทัล เบอร์ TMS320C50 พร้อมด้วย ตัวแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล (Analog To Digital A/D) อินพุตที่ต้องการวิเคราะห์จะผ่านวงจรแปลงสัญญาณ A/D จากนั้น TMS320C50 จะวิเคราะห์สัญญาณแถบความถี่ขึ้น แล้วแสดงผลออกทางจอคอมพิวเตอร์ ซึ่งกระบวนการทั้งหมด กระทำด้วยเวลาจริง (Real Time) โดยชุดพัฒนา DSK ได้รับการพัฒนาโปรแกรม แก้ไขทดสอบ และปรับปรุงซอฟต์แวร์ ให้ได้ผลตามต้องการ ซึ่งใช้สายเชื่อมโยงกับคอมพิวเตอร์ผ่านทางพอร์ต RS232 แสดงดังรูปที่ 1.1



รูปที่ 1.1 ลักษณะการใช้คอมพิวเตอร์ควบคุมการทำงานของบอร์ด DSK

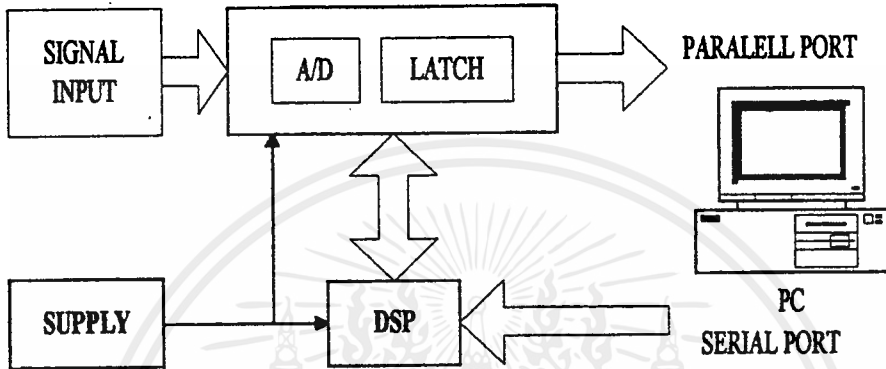
จากการที่ชุดพัฒนา DSK สามารถนำมาประยุกต์ใช้งานร่วมกับส่วนวงจรต่างๆ เป็นเครื่องวิเคราะห์แถบความถี่ของสัญญาณ แสดงผลออกทางจอคอมพิวเตอร์ มีตัวหนังสือแสดงขนาดแรงดัน และความถี่ การที่จะแสดงผลออกทางจอคอมพิวเตอร์ได้นั้น ต้องใช้ พีซี ในการเขียนโปรแกรมควบคุม แสดงดังรูปที่ 1.2

1.2 ขีดความสามารถของโครงการ

การสร้างเครื่องวิเคราะห์แถบความถี่ขึ้นนี้ มีวัตถุประสงค์เพื่อที่จะศึกษา การทำงานของระบบประมวลผลสัญญาณเชิงเลข โดยนำมาประยุกต์ใช้งานจริง เพื่อใช้ในการวิเคราะห์แถบความถี่ของสัญญาณแอนาล็อก และสัญญาณดิจิทัล โดยการเขียนโปรแกรม ควบคุมการทำงานในส่วนต่าง ๆ ทั้งชิพ TMS320C50 และส่วนของ พีซี ที่ใช้ควบคุมการแสดงผล โดยเราสามารถนำไปต่อใช้งานร่วมกับคอมพิวเตอร์ ที่ใช้กันอยู่ทั่วไปได้ โดยไม่ต้องดัดแปลงส่วนใดส่วนหนึ่งของคอมพิวเตอร์

จากการศึกษาค้นคว้า และเก็บรวบรวมข้อมูล ช่วยให้มีแนวความคิดในการออกแบบวงจรในส่วนต่างๆ การดัดแปลง รูปแบบการแสดงผล ที่จะต้องอ่านค่าความถี่ และขนาดของ

แรงดันจากค่านวณเอง มาเป็นระบบที่สามารถ บอกค่าเหล่านี้ ได้ด้วยตัวอักษร และตัวเลข ซึ่งสะดวกต่อการอ่าน และได้ความเที่ยงตรงสูง



รูปที่ 1.2 ผังการทำงานของสเปคตรัมอะนาไลเซอร์ โดยใช้ TMS320C50

1.3 เนื้อหาโดยสังเขป

ในส่วนของเนื้อหา ในปฏิญานิพนธ์นี้ ได้มีการจัดแบ่งเนื้อหาไว้อย่างเป็นสัดส่วน โดยเริ่มจาก

บทที่ 1 บทนำ ซึ่งเป็นเนื้อหาเกี่ยวกับ ความเป็นมา และความสำคัญของปัญหา ที่ต้องทำให้เกิดโครงการนี้ขึ้น ตลอดจนทฤษฎี และหลักการอย่างคร่าวๆ เพื่อให้ผู้อ่านเข้าใจหลักการง่ายๆ ในการออกแบบโครงการ รวมทั้งยังกล่าวถึงวัตถุประสงค์ ขอบเขตและประโยชน์ของการทำปฏิญานิพนธ์ในครั้งนี้

ในส่วนของบทที่ 2 จะเป็นเนื้อหาเกี่ยวกับ ทฤษฎีและหลักการซึ่งได้แสดงรายละเอียดของโครงสร้างทางสถาปัตยกรรมของชิพ TMS320C50 ซึ่งเป็นตัวประมวลผลสัญญาณ

ในส่วนของบทที่ 3 เป็นการออกแบบและการสร้าง ในส่วนนี้เป็นเนื้อหา โดยละเอียด ตั้งแต่ขั้นตอนในการออกแบบวงจรในส่วนต่างๆ การนำส่วนต่างๆ มาอินเตอร์เฟสกัน เพื่อให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ

ในส่วนของบทที่ 4 เป็นผลการทดลอง และทดสอบ ในส่วนนี้เป็นการนำเครื่องวิเคราะห์แถบความถี่ที่สร้างขึ้น มาทดสอบการทำงาน ของเครื่องโดยแบ่งเป็นลำดับขั้นการทดลองลำดับต่างๆ ตลอดจนผลการทดลองที่ได้ ตรงตามวัตถุประสงค์ที่ตั้งไว้หรือไม่

ในส่วนของบทที่ 5 เป็นการสรุป อภิปราย และข้อเสนอแนะ ซึ่งเป็นการสรุปผลเกี่ยวกับความสามารถ ประสิทธิภาพการใช้งานของเครื่องวิเคราะห์แถบความถี่ โดยใช้ชิพ TMS320C50 เป็นตัวประมวลผลพร้อมทั้ง กล่าวถึงปัญหา ที่เกิดขึ้นระหว่างการทำ โครงการ ตลอดจนแนวทางแก้ไขปัญหาที่เกิดขึ้น นอกจากนี้แล้วยังได้ เสนอแนะแนวทางการพัฒนา เครื่องวิเคราะห์แถบความถี่ ให้สามารถนำไปประยุกต์ใช้งานได้ อย่างกว้างขวาง และปรับปรุงให้มีประสิทธิภาพดียิ่งขึ้น

ในภาคผนวกแสดงรายละเอียดของโปรแกรม และรายละเอียดของอุปกรณ์ต่างๆ ที่ใช้จัดทำโครงการ ดังนี้

ภาคผนวก ก วงจรเครื่องวิเคราะห์แถบความถี่

ภาคผนวก ข รายละเอียดของอุปกรณ์ที่ใช้ในโครงการ

ภาคผนวก ค ผังการทำงาน และ โปรแกรมที่ใช้ในโครงการ

ภาคผนวก ง คู่มือการใช้งาน

ปฏิญานิพนธ์เล่มนี้ ได้รวบรวมเนื้อหาที่เกี่ยวข้องไว้บางส่วน หากท่านต้องการรายละเอียดนอกเหนือจากเนื้อหา ในปฏิญานิพนธ์นี้ สามารถค้นคว้าหารายละเอียดได้จากหนังสือที่ได้อ้างอิงไว้ในบรรณานุกรมของปฏิญานิพนธ์เล่มนี้

บทที่ 2

ทฤษฎีและหลักการ

2.1 บทนำ

เครื่องวิเคราะห์แถบความถี่ หรือสเปกตรัมอะนาไลเซอร์ เป็นเครื่องมือวัดชนิดหนึ่งที่มีความจำเป็นใช้กันอย่างมากในงานทดลอง หรืองานวิจัยทั่วไปเนื่องจากสเปกตรัมอะนาไลเซอร์ ที่ใช้กันอยู่นั้น ล้วนเป็นสินค้านำเข้า จากต่างประเทศแทบทั้งสิ้น ทำให้ราคาของเครื่องค่อนข้างสูง หากนำมาใช้ในการทดลอง หรือวิจัยส่วนตัว คงไม่คุ้มค่ากับการลงทุน แต่หากเป็นการนำมาใช้ในสถานศึกษา ห้องทดลอง หรือห้องวิจัย อาจจะคุ้มกับการลงทุน โดยปกติเครื่องสเปกตรัมอะนาไลเซอร์จะมีฟังก์ชันให้เลือกใช้งานมาก ซึ่งเวลาใช้งานจริง อาจใช้เพียงแค่วัดสเปกตรัมของสัญญาณอย่างเดียว จึงไม่คุ้มถ้าหากซื้อมาใช้

จากแนวความคิดในการสร้างเครื่องวิเคราะห์แถบความถี่นั้น จึงนำไมโครโปรเซสเซอร์ มาทำการประมวลผลสัญญาณแบบเวลาจริง และเลือกหน่วยความจำที่มีความเหมาะสมกับการใช้งาน มีพอร์ตอินพุต และพอร์ตเอาต์พุตเพียงพอสำหรับการใช้งาน

ชิพ DSP เป็นไมโครโปรเซสเซอร์ ที่ใช้เฉพาะงานประมวลผลสัญญาณเชิงเลข ในการประยุกต์ใช้ DSP จะเกี่ยวข้องกับค่าของอินพุต เพื่อให้การคำนวณมีประสิทธิภาพ มีความละเอียดสูง และสามารถทำงานกับข้อมูลได้จำนวนมาก ซึ่งในปัจจุบัน มีผู้นำเข้าชุดพัฒนา TMS320C50 หรือ DSP Starter Kit บอร์ด ผลิตโดยบริษัทเท็กซัสอินสตรูเมนต์ เข้ามาในประเทศไทย เพื่อให้ศึกษาและออกแบบทดลองใช้งาน โดยที่บน DSP บอร์ด มีหัวใจหลักที่สำคัญ คือ TMS320C50 ซึ่งเป็นชิพไมโครโปรเซสเซอร์ ที่ใช้งานในด้านการประมวลผลสัญญาณเชิงเลขโดยเฉพาะ และอุปกรณ์เชื่อมต่อภายใน ทำให้มีความสะดวก และประหยัดการใช้งาน โดยเฉพาะการทำงานตามคำสั่งแต่ละคำสั่งจะใช้เวลา 50 นาโนวินาที ต่อ 1 ไบต์ ทำให้สามารถประมวลผลสัญญาณได้ถึง 28.6 ล้านคำสั่งต่อวินาที จึงทำการประมวลผลแบบเวลาจริงได้ทัน เป็นชิพประมวลผลสัญญาณเชิงเลขบนชิพเดี่ยวขนาด 16 บิต แบบไม่คิวดิจิตอล ทำให้เลือกประยุกต์ใช้งานได้อย่างกว้างขวาง

2.2 ลักษณะโครงสร้างของ TMS320C50

1. ไซเคิลในการทำงานของคำสั่งใช้เวลา 35-50 นาโนวินาที
2. ซอร์สโค้ดเข้ากันได้กับ 'C1x และ 'C2x
3. ทำคำสั่งได้ 28.6 ล้านคำสั่งต่อวินาที (MIPS)
4. มีหน่วยความจำข้อมูลภายในชิพ (RAM) 9 กิโลเวิร์ด
5. มีหน่วยความจำโปรแกรม (ROM) 2 กิโลเวิร์ด
6. มีทางเข้าถึงหน่วยความจำข้อมูล 2 ทาง ภายในชิพ (RAM) 1056 เวิร์ด
7. มีพื้นที่หน่วยความจำภายนอก 224 กิโลเวิร์ด
8. หน่วยประมวลผลคณิตศาสตร์ (ALU) ,แอดคิวมูเลเตอร์ (ACC) และ บัฟเฟอร์มี

ขนาด 32 บิต

9. หน่วยตอจิกขนานมีขนาด 16 บิต
10. การคูณเป็นแบบขนานขนาด 16 บิต
11. คูณเลข โดยใช้สัญญาณนาฬิกาหนึ่งไซเคิล
12. มีอ็อกซีตารี่รีจิสเตอร์ จำนวน 8 ตัว
13. เพิ่มฮาร์ดแวร์สแตกซ์ันถึง 8 ระดับ
14. มีตัวเลื่อนบิต (Shifter) แบบขนาน 16 บิต และ 64 บิตในการเลื่อนข้อมูล
15. มีการทวนคำสั่งที่เป็นคำสั่งเดียว และทวนรหัส โปรแกรม หรือข้อมูลเป็นบล็อก
16. มีการจัดการ โปรแกรม หรือข้อมูลไปไว้ในบล็อกหน่วยความจำที่ดีกว่า
17. มีพอร์ตอนุกรมแบบเข้าจังหวะ (Synchronors) 2 ทาง (Full Duplex) สำหรับติดต่อกับ

พอร์ตอนุกรมกับอุปกรณ์อื่น ๆ

18. มีพอร์ตอินพุต เอาต์พุต แบบขนานทั้งหมด 65,536 พอร์ต และมี 16 พอร์ต อยู่ในพื้นที่หน่วยความจำ

19. มีโปรแกรม Wait state เพื่อใช้ในการติดต่อกับอุปกรณ์ หรือหน่วยความจำ ภายนอกที่ทำงานช้ากว่า

20. ปฏิบัติงานแบบ 4 ไปป์ไลน์
21. โหมคการอ้างข้อมูลแบบชี้ตำแหน่ง (Index-addressing)
22. มีตัวกำเนิดสัญญาณนาฬิกาภายในชิพ

23. ใช้เทคโนโลยีแบบ CMOS ใช้ไฟเลี้ยง 5 โวลต์

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24. มี 132 ขา รอบตัวถังสี่ด้าน (Quad Flat Package : QFP)

2.3 สถาปัตยกรรมของ TMS320C50 ชิพ DSP โดยทั่วไป

โครงสร้างส่วนสำคัญของ TMS320C50 DSP ประกอบด้วยส่วนสำคัญสามส่วน คือ

1. หน่วยประมวลผลกลาง (CPU)
2. หน่วยความจำ (Memory)
3. วงจรอินเทอร์เฟซกับอุปกรณ์ภายนอก

TMS320C50 เป็นโปรเซสเซอร์ที่ใช้สำหรับในการประมวลผลสัญญาณเชิงเลข ซึ่งจะมีลักษณะการทำงาน คล้ายคลึงกับ TMS320C25 และ ใช้สถาปัตยกรรมแบบเดียวกัน คือ สถาปัตยกรรมแบบ “ฮาร์ดแวร์” โดยจะแบ่งโครงสร้างของหน่วยความจำออกเป็น 2 ส่วนแยกจากกัน คือ หน่วยความจำโปรแกรม และหน่วยความจำข้อมูล ซึ่งจะทำให้การทำงานต่างๆ เป็นไปด้วยความเร็วสูง นอกจากนี้ยังสามารถทำการโอนย้ายข้อมูลระหว่างกันได้

2.4 ตัวประมวลผลกลาง

การพัฒนาปรับปรุง CPU ‘C5x ยังคงรักษาซอร์สโค้ด (Source Code) ให้เข้ากันได้กับ ‘C1x,’C2x ในขณะที่มีการปรับปรุงสมรรถนะ และความสะดวกในการใช้งาน ในการปรับปรุง มีแอสเซมบลีเตอร์ 32 บิต และเพิ่มการตรวจสอบมีคำสั่งใหม่ในการอินเทอร์เฟซอุปกรณ์ภายนอกสำหรับผู้ใช้งานมากขึ้น มีฟังก์ชัน (Function) ควบคุมอิสระของหน่วยลอจิกแบบขนาน (PLU: Parallel logic Unit) สำหรับการปฏิบัติงานแบบบูลีน (Boolean) และเซต (Set) รีจิสเตอร์ (Register) คอนแทคสวิตช์ (Contact - Switch Register) ในการกำหนด Zero - latency ซึ่ง Contact - Switch สามารถทำการให้บริการตำแหน่ง การอินเทอร์รัพต์ (Interrupt)

TMS320C50 มีหน่วยความจำที่ใช้เก็บข้อมูลหรือตำแหน่ง (Register) ขนาด 1 ไบต์ ภายในชิพ จำนวน 28 ตัว ดังตารางที่ 2.1

ตารางที่ 2.1 รีจิสเตอร์ภายใน 'C50

ชื่อ	ตำแหน่ง 'C50(Dec)	ตำแหน่ง 'C50(Hex)	คำอธิบาย
-	0 - 3	0 - 3	ตำแหน่งที่จองไว้
IMR	4	4	รีจิสเตอร์แสดงสถานะอินเทอร์รัพต์
GREG	5	5	รีจิสเตอร์หน่วยความจำทั่วไป
IFR	6	6	รีจิสเตอร์อินเทอร์แฟล็ก
PMST	7	7	รีจิสเตอร์แสดงสถานะตัวประมวลผล
RPTC	8	8	รีจิสเตอร์ค่านับทวน
BRCR	9	9	รีจิสเตอร์นับทวนแบบบล็อก
PASR	10	A	รีจิสเตอร์ทวนแอดเดรสเริ่มต้นโปรแกรมแบบบล็อก
PAER	11	B	รีจิสเตอร์ทวนแอดเดรสสิ้นสุดโปรแกรมแบบบล็อก
TREG0	12	C	รีจิสเตอร์ชั่วคราวสำหรับการคูณ
TREG1	13	D	รีจิสเตอร์ชั่วคราวสำหรับไดนามิกชิฟต์เคาน์
TREG2	14	E	รีจิสเตอร์ชั่วคราวสำหรับชิบิต ในบิตทดสอบไดนามิก
DBMR	15	F	รีจิสเตอร์ย้ายบิตไดนามิก
AR0	16	10	รีจิสเตอร์ช่วย 0
AR1	17	11	รีจิสเตอร์ช่วย 1
AR2	18	12	รีจิสเตอร์ช่วย 2
AR3	19	13	รีจิสเตอร์ช่วย 3
AR4	20	14	รีจิสเตอร์ช่วย 4
AR5	21	15	รีจิสเตอร์ช่วย 5
AR6	22	16	รีจิสเตอร์ช่วย 6
AR7	23	17	รีจิสเตอร์ช่วย 7
INDX	24	18	รีจิสเตอร์อินเดกซ์
ARCR	25	19	รีจิสเตอร์เปรียบเทียบ รีจิสเตอร์ช่วย
CBSR1	26	1A	รีจิสเตอร์แอดเดรสเริ่มต้น บัฟเฟอร์ทมนวน 1
CBER1	27	1B	รีจิสเตอร์แอดเดรสสิ้นสุด บัฟเฟอร์ทมนวน 1
CBSR2	28	1C	รีจิสเตอร์แอดเดรสเริ่มต้น บัฟเฟอร์ทมนวน 2
CBER2	29	1D	รีจิสเตอร์แอดเดรสสิ้นสุด บัฟเฟอร์ทมนวน 2

ตารางที่ 2.1 (ต่อ) รีจิสเตอร์ภายใน 'C50

ชื่อ	ตำแหน่ง 'C50(Dec)	ตำแหน่ง 'C50(Hex)	คำอธิบาย
CBCR	30	1E	รีจิสเตอร์ควบคุม บัฟเฟอร์หมุนวน
BMAR	31	1F	รีจิสเตอร์เคลื่อนย้ายกลุ่มแอดเดรส
-	32-79	20-4F	รีจิสเตอร์สำหรับหน่วยความจำ

ทำให้ TMS320C50 มีลักษณะพิเศษ เนื่องจากมีหน่วยความจำรอม (ROM) ภายในขนาด 2k x 16 บิต สามารถโปรแกรม (EPROM) เป็นหน่วยความจำที่ใช้ในการบูต (Boot) หน่วยความจำรอมภายนอก ด้วยโปรแกรมจากภายนอก ซึ่งเป็นโปรแกรมความเร็วสูงที่อยู่ภายใน หรือจาก SRAM ภายนอก ที่สามารถเลือกรีเซ็ต (Reset) หน่วยความจำรอมได้โดยการให้ ขา MP/MC' มีสถานะเป็น Low

2.5 ขาของ TMS320C50 DSP

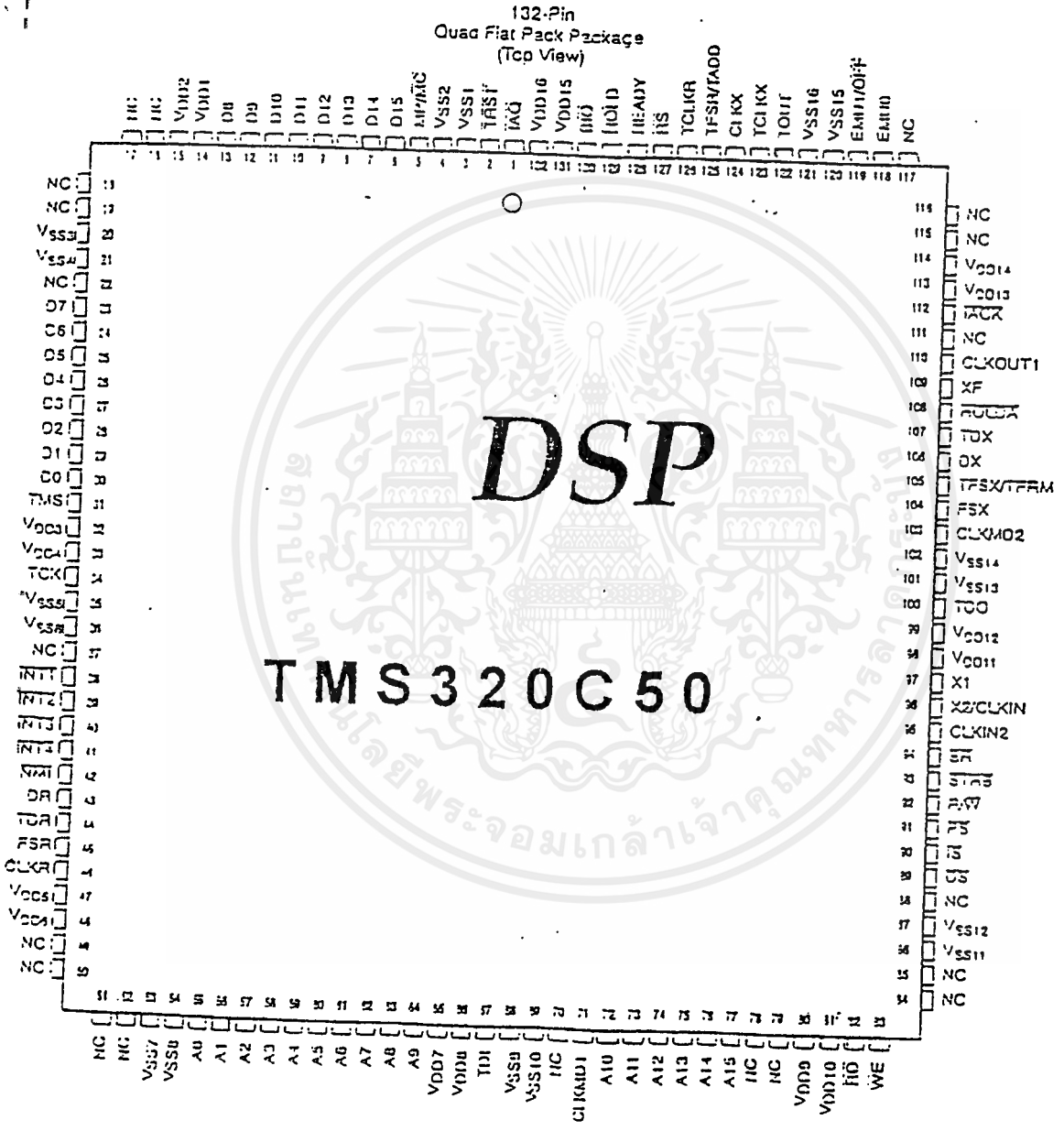
ชิปประมวลสัญญาณเชิงเลข เบอร์ TMS320C50 จะมีลักษณะเป็นแพ็คเกจ มีขาต่างๆ 132 ขา รอบแพ็คเกจทั้งสี่ด้าน (Quad Flat Pack : QFP) ซึ่งแต่ละขามีความสัมพันธ์กัน แสดงดังรูปที่ 2.1

2.6 การจัดหน่วยความจำ

ในตัว TMS320C50 มีหน่วยความจำความเร็วสูงขนาด 1,056 เวิร์ด โดยที่แต่ละเวิร์ดมีขนาด 16 บิต และมีรอมภายในขนาด 256 เวิร์ด สามารถอ้างอิงหน่วยความจำภายนอกได้ถึง 224 กิโลเวิร์ด แบ่งเป็นหน่วยความจำข้อมูล, หน่วยความจำโปรแกรม และหน่วยความจำอินพุต เอาต์พุต อย่างละ 64 กิโลเวิร์ด นอกนั้นอีก 32 กิโลเวิร์ด สำหรับใช้งานทั่วไป ส่วน ALU ,แอดคิวิตูเลเตอร์ และแอดคิวิตูเลเตอร์บัฟเฟอร์มีขนาด 32 บิต ที่สามารถคูณเลขขนาด 32 บิต ได้ภายใน 1 แมกซิมัซไคเคิล ทั้งนี้เนื่องจากตัว ALU มีวงจรถูกแบบ 16 x 16 แบ่งเป็นอินพุต 16 แชนแนล เอาต์พุต 16 แชนแนล วงจรติดต่อหน่วยความจำร่วม (Global Data Memory Interface) แสดงดังรูปที่ 2.2

ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pinout



รูปที่ 2.1 ลักษณะภายนอกของ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program		Program		Data	
Hex		Hex		Hex	
0000	Interrupt and Reversed External	0000	Interrupt and Reserved On chip	0000	Memory-Maped Registers
002F		002F		005F	
0030	External	0030	on-chip ROM	0060	On-Chip DARAM B
07FF		07FF		007F	
0800	On-Chip SARAM (RAM=1) External (RAM=0)	0800	On-Chip SARAM (RAM=1) External (RAM=0)	0080	Reserved
2BFF		2BFF		02FF	
2C00	External	2C00	External	0300	On-Chip DARAM B
FDFE		FDFE		04FF	
FE00	On-Chip DARAM B0(CNF=1) External(CNF=0)	FE00		0500	Reserved
FFFF		FFFF		07FF	
				0800	On-Chip SARAM'1' External '0'
				2BBF	
				2C00	External
				FFFF	

MP/MC=1
Microprocessor Mode

MP/MC=0
Microprocessor

รูปที่ 2.2 การจัดหน่วยความจำของ TMS320C50

ตัว TMS320C50 สามารถต่อใช้งานเพียงตัวเดียวหรือจะต่อใช้งานขนานกันหลายตัวที่เรียกว่าเป็นระบบ Multi - Processor System ก็ได้

ในการใช้งานกับหน่วยความจำภายนอก ไม่ว่าจะเป็นอีพรอมหรือรอม ตัวชิพจะทำงานได้ช้ามาก จึงได้เน้นให้ใช้หน่วยความจำแบบภายในชิพเป็นที่เก็บโปรแกรมแทน ในการใช้งานให้ความเร็วไหลดโปรแกรมจากหน่วยความจำภายนอกมาเก็บในชิพ แล้วจึงรันโปรแกรมทำงานโดย TMS320C50 ได้เพิ่มคำสั่ง “Block Transfer” ที่ใช้ในการไหลดข้อมูลจากหน่วยความจำภายนอกมาที่ละบล็อกล แล้วมาเก็บไว้ในแรมของชิพ วิธีการนี้จะช่วยให้การทำงานเร็วขึ้น

2.7 สรุปสถาปัตยกรรม โดยทั่วไป

TMS320C50 เป็นตัวประมวลผลสัญญาณทางดิจิทัล ที่มีสมรรถนะสูงมาก ซึ่งโดยทั่วไป ได้ถูกออกแบบให้มีโครงสร้างเหมือนกับ ‘C2x ทางด้านโครงสร้างใช้เทคโนโลยีแบบ “ฮาร์วาร์ด”

มีสมรรถนะ ในการประมวลผลสูงสุด โดยแยกหน่วยความจำออกจากกัน เป็นหน่วย-
ความจำโปรแกรม และหน่วยความจำข้อมูล สำหรับการเอ็กซ์คิวต์คำสั่งได้ด้วยความเร็วสูงสุด
คำสั่งจะรองรับการส่งข้อมูลระหว่างสองระยะเวลา

TMS320C50 ใช้ปฏิบัติการคำนวณแบบ 2' s Complement ขนาด 32 บิต ของหน่วย
คำนวณแบบคณิตศาสตร์และลอจิก และแอคคิวมูเลเตอร์

ALU จุดประสงค์ทั่วไปใช้เป็นหน่วยคำนวณมีขนาด 16 บิตเวิร์ด จากหน่วยความจำ
โครงสร้างคำสั่ง หรือผลของการคูณขนาด32บิต เป็นการเพิ่มการปฏิบัติงานเกี่ยวกับคณิตศาสตร์
ทำให้ ALU สามารถปฏิบัติงานแบบบูลีนได้ โดยมีแอคคิวมูเลเตอร์เป็นตัวเก็บผลลัพธ์ที่ได้จาก
การคำนวณของ ALU และเป็นอินพุตให้กับ ALU อีกด้วย

แอคคิวมูเลเตอร์มีขนาด 32 บิต แบ่งออกเป็นไบต์ระดับสูง (บิตที่ 31 ถึง บิต 16) และ
ตำแหน่งไบต์ระดับต่ำ (บิต 15 ถึง บิต 0) คำสั่งในการเก็บข้อมูลจะอยู่ภายในเงื่อนไข สำหรับการ
เก็บที่ตำแหน่งระดับสูงหรือ ระดับต่ำในหน่วยความจำของแอคคิวมูเลเตอร์ ในการเก็บข้อมูล
ชั่วคราวจะมีแอคคิวมูเลเตอร์ขนาด 32 บิต

วิธีการบวกของ ALU จะมีหน่วยลอจิกแบบขนาน PLU จะเป็นตัวจัดการเกี่ยวกับบิตที่
ความเร็วสูงมาก และทำให้ไม่มีความสลับซับซ้อน ในการกำหนดบิต การทดสอบความต้องการ
ในการควบคุม และการปฏิบัติการของรีจิสเตอร์สถานะ

การดำเนินการของตัวคูณมีขนาด 16 x 16 บิต แบบ 2's Complement ผลของการคูณจะ
มีขนาด 32 บิต คำสั่งเดียวในหนึ่งไซเคิล ตัวคูณพื้นฐานจะมีอยู่ 3 ตัว คือ

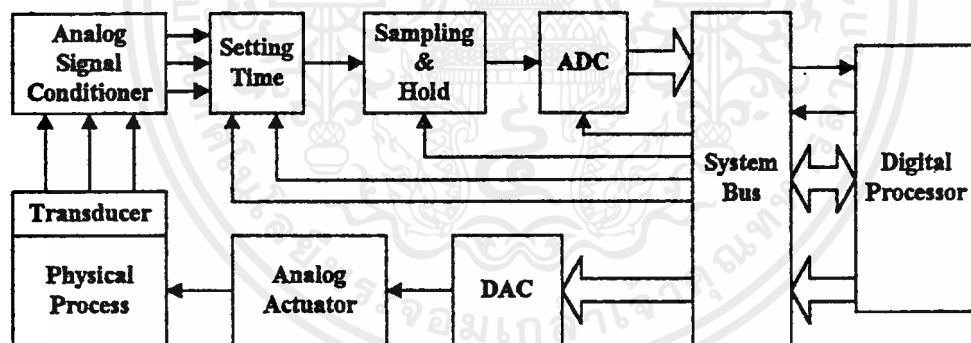
1. Multiplier Array เป็นตัวคูณแบบอาร์เรย์
2. PREG (Product Register) เป็นรีจิสเตอร์ใช้ในการเก็บผลคูณมีขนาด 32 บิต ซึ่งค่า
ของตัวคูณนำมาจากหน่วยความจำข้อมูล และนำมาจากหน่วยความจำโปรแกรม ตัวคูณภายใน
ชิพจะมีความเร็วมาก ทำให้มีประสิทธิภาพในการจัดการ การปฏิบัติงานที่เป็นปัจจัยพื้นฐานของ
DSP เช่น การหมุนวน การเกี่ยวพันกัน และการยอมให้คำสั่งหรือโปรแกรม มีข้อมูล หรือสภาพ
ตามเงื่อนไข

3. TREG0 (Temporary Register) เป็นรีจิสเตอร์เก็บผลคูณชั่วคราว ขนาด 16 บิต

2.8 ทฤษฎีการแปลงสัญญาณ (Data Acquisition and Conversion)

ในอดีตรูปแบบของสัญญาณไฟฟ้าโดยมากมักจะอยู่ในรูปสัญญาณแอนาลอก การนำเอาสัญญาณไฟฟ้ามาประมวลผล เพื่อให้เกิดรูปแบบที่ต้องการนั้น ต้องใช้อุปกรณ์ทางแอนาลอก แต่ปัจจุบันนี้เทคโนโลยีทางด้านดิจิทัลก้าวหน้าไปมาก ทำให้การประมวลผลสัญญาณทางดิจิทัล สามารถทำได้อย่างรวดเร็ว และมีประสิทธิภาพ

ดังนั้นการแปลงรูปแบบสัญญาณ (Conversion) จึงมีความจำเป็นในการแปลงสัญญาณแอนาลอกที่มีอยู่แล้ว ให้เป็นสัญญาณดิจิทัล โดยอุปกรณ์การแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล และจะถูกประมวลผลโดยตัวประมวลสัญญาณดิจิทัล เช่น คอมพิวเตอร์ เป็นต้น จากผลลัพธ์ที่ได้ อาจถูกนำมาแสดงผลโดยตรงเลข หรืออาจถูกแปลงให้อยู่ในรูปของสัญญาณแอนาลอกที่ใช้งานได้ การที่จะแปลงสัญญาณแอนาลอกเป็นดิจิทัลได้นั้น สามารถทำได้โดยใช้อุปกรณ์แปลงสัญญาณดิจิทัลเป็นสัญญาณแอนาลอก สำหรับระบบที่มีการประมวลผลสัญญาณข้อมูลทางดิจิทัล แสดงดังรูปที่ 2.3



รูปที่ 2.3 ระบบที่มีการประมวลผลข้อมูลทางดิจิทัล

จากรูปที่ 2.3 การเปลี่ยนแปลงทางกายภาพในลักษณะใดๆ ก็ตาม เช่น อุณหภูมิ ความดัน ความเร็ว จะถูกเปลี่ยนให้มาเป็นสัญญาณไฟฟ้าแบบแอนาลอก โดยทรานสดิวเซอร์ ทฤษฎีการสุ่มที่มีรูปแบบเหมาะสมกับลักษณะทางกายภาพนั้นๆ จากนั้นสัญญาณทางไฟฟ้า ก็จะถูกปรับให้อยู่ในรูปแบบ และขนาดที่เหมาะสมก่อน โดยวงจรต่าง ๆ เช่น วงจรขยาย หรือวงจรกรองสัญญาณ เป็นต้น วงจรแซมปลิงแอนด์โฮลด์ จะสุ่มขนาดของสัญญาณแอนาลอกมา แล้วจะทำการโฮลด์สัญญาณนั้นไว้ชั่วขณะ โดยไม่จำเป็นต้องใช้ วงจร ADC แล้วข้อมูลทางดิจิทัล จะถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งต่อไปยังบัลลูนของระบบ จากนั้นตัวโปรเซสเซอร์ จะทำการประมวลผลข้อมูลกลับมา เพื่อควบคุมกิจการทางกายภาพของระบบ โดยผ่านตัวกระทำทางกล (Analog Actuator)

2.9 วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล

การติดต่อระหว่างมนุษย์ส่วนมากจะใช้สัญญาณต่อเนื่อง (Analog) เป็นสัญญาณติดต่อกัน แต่ว่าการทำงานของระบบคอมพิวเตอร์จะใช้สัญญาณเป็นช่วงดิจิทัลเป็นสัญญาณในการทำงาน ดังนั้นถ้าเราต้องการที่จะให้ คอมพิวเตอร์ช่วยเราทำงานแล้ว เราจึงต้องเปลี่ยนสัญญาณแอนาลอกที่เราใช้อยู่ ให้ไปเป็นสัญญาณดิจิทัล เพื่อให้คอมพิวเตอร์ หรือเครื่องประมวลสัญญาณดิจิทัลรับรู้ได้ เมื่อประมวลสัญญาณเสร็จ ก็จะส่งข้อมูลออกมา เป็นสัญญาณดิจิทัล ซึ่งเป็นเรื่องยุ่งยาก ที่เราจะเข้าใจในข้อมูลนั้น ดังนั้นเราจึงเปลี่ยนข้อมูลที่เป็นสัญญาณดิจิทัลให้เป็นสัญญาณแอนาลอก เพื่อให้มนุษย์เข้าใจในข้อมูลนั้น ดังนั้นพอจะสรุปได้ว่าการเปลี่ยนสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล และการเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณแอนาลอก ซึ่งเป็นการประสานโลกของคอมพิวเตอร์เข้ากับโลกของมนุษย์ เพื่อให้มนุษย์ได้ใช้คอมพิวเตอร์ได้อย่างสมบูรณ์ยิ่งขึ้น การเปลี่ยนสัญญาณแอนาลอกไปเป็นดิจิทัล เรียกว่า Analog To Digital Conversion (ADC) หรือ เรียกย่อๆ ว่า A TO D หรือ A/D ในทำนองเดียวกัน การเปลี่ยนสัญญาณดิจิทัลไปเป็นสัญญาณแอนาลอก เรียกว่า Digital To Analog Conversion (DAC) เรียกย่อๆว่า D TO A หรือ D/A

การเปลี่ยนสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล A/D มีขบวนการอยู่หลายวิธี แต่ละวิธีมีข้อดี และข้อเสียรวมทั้งราคาที่แตกต่างกันไป จึงขอกล่าววิธีแปลงสัญญาณอย่างคร่าว ๆ ซึ่งมีหลักการดังนี้

2.9.1 A/D แบบเซอร์โว (Servo)

เป็นแบบที่ง่ายและราคาถูกที่สุด แต่การทำงานไม่เที่ยงตรง อันเนื่องมาจาก อุณหภูมิ เวลา และความไวในการรับสัญญาณอินพุต

2.9.2 A/D แบบดูอัล-สโลป อินทิเกรต (Dual-Slope Intergrator)

เป็นแบบที่มีความละเอียดถูกต้องสูง ราคาค่อนข้างแพง ทางด้านอุณหภูมิมีผลน้อย แต่มีความเร็วในการทำงานต่ำ

2.9.3 A/D แบบซัคเซสซีฟ-แอฟพร็อกซิเมชัน (Successive-Approximation)

เป็นแบบที่ใช้งานกันทั่วไป และเป็นแบบซึ่งทำงานได้อย่างรวดเร็ว มีความถูกต้องสูง แต่การสร้างวงจรนั้นต้องใช้ D/A

2.9.4 A/D แบบคอมพาราเตอร์ขนาน (Parallel-Comparator)

เป็นแบบที่มีความเร็วในการทำงานสูงแบบหนึ่ง แต่ราคาก็สูงที่สุดในบรรดา A/D ทุกชนิด หลักการทำงานไม่ยุ่งยากซับซ้อน แต่ต้องใช้ส่วนประกอบอื่นค่อนข้างมาก

2.10 ทฤษฎีของการสุ่มข้อมูล (Sampling)

ในการสุ่มข้อมูลนั้น สัญญาณแอนาลอกจะถูกสุ่มเป็นระยะคงที่ กลุ่มของสัญญาณที่สุ่มจะแทนด้วยความเร็วสูง ซึ่งเกิดจากการตัดต่อสัญญาณแอนาลอกด้วยระยะเวลาอันสั้น ผลของการสุ่มด้วยความเร็วจะเหมือนกับการคูณขบวนสัญญาณพัลส์ กับสัญญาณแอนาลอก ซึ่งจะได้สัญญาณที่มอดูเลต (Modulate) ระหว่างขบวนพัลส์กับสัญญาณแอนาลอก

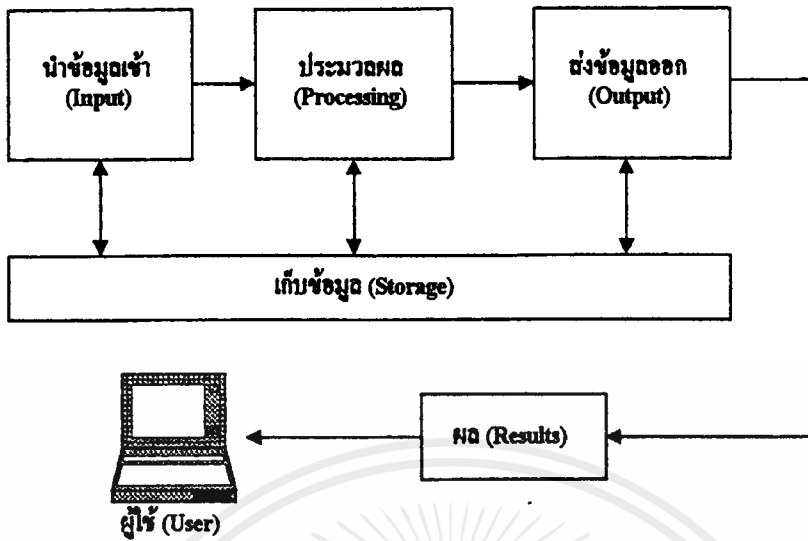
อัตราการสุ่มสัญญาณ หรือความถี่ของการสุ่มสัญญาณควรมีค่าเท่าใด ที่ข้อมูล ที่สุ่มได้นี้จะเป็นตัวแทนที่ดีของสัญญาณต่อเนื่องนั้น คำตอบคือขึ้นอยู่กับความถี่สัญญาณแอนาลอก และจากทฤษฎีของการสุ่ม (Nyquist Theorem) กล่าวไว้ว่า “ถ้าสัญญาณต่อเนื่องที่มีความถี่ฮาร์โมนิก (Harmonic Frequency) ไม่เกินความถี่พินคาเมนทอลแล้วสัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาเช่นเดิม โดยไม่สูญเสียรายละเอียด หรือผิดเพี้ยนไป ถ้าอัตราการสุ่มมากกว่า $2f$ ”

2.11 ทฤษฎีการสร้างภาพดิจิทัลบนจอคอมพิวเตอร์

คอมพิวเตอร์ จัดเป็นเครื่องมือที่ใช้กันอย่างกว้างขวาง ทั้งด้านไฟฟ้า อิเล็กทรอนิกส์ รวมทั้งทางด้านวิทยาศาสตร์และด้านธุรกิจด้วย ข้อดีของคอมพิวเตอร์ คือ สามารถประมวลผลและเก็บข้อมูลได้เป็นจำนวนมาก และสามารถสั่งให้ทำงานต่าง ๆ ได้

การประมวลผลของคอมพิวเตอร์มีดังนี้

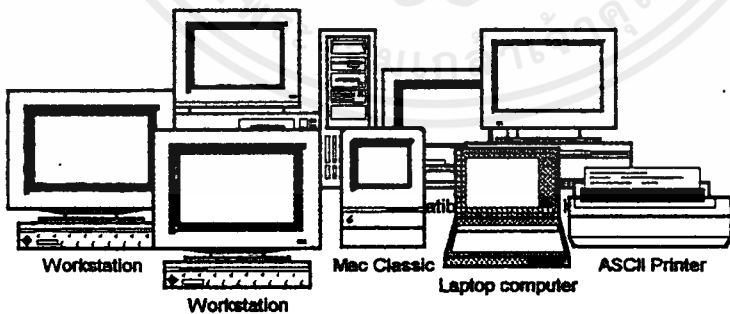
1. การนำข้อมูลเข้า
2. การประมวลผล
3. การเก็บข้อมูล
4. การส่งข้อมูลออก



รูปที่ 2.4 ขบวนการประมวลผล

2.11.1 หลักการสร้างภาพบนจอคอมพิวเตอร์

จอภาพ (Monitor) เป็นอุปกรณ์ที่ใช้ส่งข้อมูลออกชนิดหนึ่ง ซึ่งบางครั้ง เรียกว่า Screen หรือ Video Display มีลักษณะคล้าย ๆ กับจอโทรทัศน์ เป็นส่วนที่ใช้แสดงตัวอักษร หรือภาพให้ ผู้ใช้งานมองเห็นได้ ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 จอภาพ (monitors) ตัวอักษร หรือรูปภาพ

มีคำอีกหนึ่งคำที่ใช้แทนความหมายว่าจอภาพ คือ หลอดภาพ (Cathode Ray Tube) หรือ CRT เป็นหลอดสุญญากาศ มีความหมาย เช่นเดียวกับคำว่า หน่วยแสดงผลภาพ (VDU) ซึ่งย่อมาจาก Visual Display Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอภาพจะแตกต่างกันที่ความละเอียด (resolution) ซึ่งหมายถึงจำนวนจุดเล็กๆ ที่ประกอบกันเป็นตัวอักษร หรือ จำนวนช่องเล็กๆ ที่กำหนดภาพบนจอภาพ จอภาพที่ความละเอียดต่ำสุดมักเป็นพวกจอภาพสีเดียว (monochrome) ที่แสดงตัวอักษร หรือภาพได้สีเดียวบนพื้นสีดำ หรือเทา โดยตัวอักษรอาจสีเขียว หรือเหลืองอำพัน จอภาพที่มีความละเอียดมากขึ้น ได้แก่ จอภาพสี RGB ย่อมาจาก Red Green และ Blue เป็นจอภาพที่ให้กำเนิดสัญญาณ สีแดง เขียว และสีน้ำเงิน นำมาผสมกันจนกระทั่งได้ภาพสีที่สดใส จอภาพสีที่มีความละเอียดสูงกว่า RGB คือ จอภาพสี EGA ย่อมาจาก Enhanced Graphics Adapter เป็นระบบแสดงภาพสีที่มีความละเอียดสูงกว่าปกติ ประกอบด้วยแผงวงจรเชื่อมต่อ และจอภาพแบบอีจีเอ แสดงข้อมูลแบบตัวอักษรได้ 40 x 25 บรรทัด หรือ 80 x 25 บรรทัด แสดงกราฟฟิกส์ได้พร้อมกัน 16 สี ด้วยความละเอียดขนาด 640 x 350 จุด หรือ 320 x 200 จุด แต่จอภาพสีที่มีความละเอียดสูงกว่าอีกก็คือ จอภาพสี VGA ย่อมาจากคำว่า Video Graphics Array ซึ่งถือว่าเป็นระบบแสดงภาพบนจอภาพสีที่มีความละเอียดสูงกว่าปกติ ประกอบด้วยแผงวงจรเชื่อมต่อ และจอภาพแบบวีจีเอ แสดงข้อมูลแบบตัวอักษรได้ 40 x 25 บรรทัด หรือ 80 x 25 บรรทัด แสดงกราฟฟิกส์ได้ 2 แบบ ถ้าแสดงพร้อมกัน 16 สี ใน 256 สี ด้วยความละเอียดขนาด 640 x 480 จุด หรือแสดง 256 สี ด้วยความละเอียด 320x 200 จุด มีการพัฒนาการ์ดให้มีความละเอียด ในการแสดงภาพสูงขึ้น เช่น การ์ด SupperVGA (SVGA) ที่มีความละเอียดสูงถึง 1024 x 768 จุด ที่ 256 สี และการ์ด Extended VGA (XGA) ที่สามารถแสดงสีได้ถึง 32,768 สี 65,536 สีและ 16.7 ล้านสีที่ความละเอียด 640 x 480จุด

จอภาพแต่ละชนิด ใช้คู่กับแผงวงจร (Circuit Card) ที่ติดตั้งอยู่ภายในตัวเครื่องคอมพิวเตอร์ แผงวงจรต่างๆ ที่ใส่เข้าไปในคอมพิวเตอร์ ก็เพื่อทำให้คอมพิวเตอร์สามารถทำงานได้มากขึ้น เช่น แผงวงจรกราฟฟิกส์ ทำให้คอมพิวเตอร์สามารถทำกราฟได้ หรือบางครั้งก็ใส่แผงวงจรเข้าไปเพื่อเพิ่มหน่วยความจำ

จุด (dots) ต่างๆ ที่สร้างภาพ หรือตัวอักษรบนจอภาพเราเรียกว่า ช่องกำเนิดภาพ (pixels) ซึ่งเป็นจุดเล็กที่สุดของแสง ในการกำเนิดภาพบนจอภาพ ถ้าช่องกำเนิดภาพ ยิ่งเล็กลงความละเอียดของภาพจะยิ่งมากขึ้น

2.11.2 โหมดของจอภาพ

เราอาจแบ่งการแสดงผลบนจอภาพ ตามกรรมวิธีในการแสดงผลได้สองชนิดคือ

1. โหมดตัวอักษร หรือโหมดอักขระ (Text Mode)

ในโหมดนี้รหัสของตัวอักษร และรูปแบบการแสดงผลตัวอักษรซึ่งเรียกว่า แอดตริบิวต์

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้ภายในเท่านั้น การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Attribute) จะเก็บเรียงกันไปในวิดีโอแรม (VideoRAM) ซึ่งวางตัวอยู่ในการ์ดแสดงผล รหัสจะถูกนำไปใช้ในการเปิดตารางภาพของตัวอักษร (Character Generators) ภาพที่ได้จากการเปิดตารางจะถูกนำไปแสดงบนจอภาพตามรูปแบบการแสดงที่กำหนด ในโหมดนี้ เราจะเห็นว่าบนจอภาพ จะถูกแบ่งพื้นที่ออกเป็นส่วนย่อยๆ ในแต่ละส่วนจะทำหน้าที่ แสดงภาพของตัวอักษรหนึ่งตัว เช่น ในโหมด 80 x 25 ตัวอักษร ก็จะมีพื้นที่สำหรับแสดงภาพตัวอักษรได้ 80 คอลัมน์ (จำนวนตัวอักษรในแนวนอน) และ 25 บรรทัด (จำนวนตัวอักษรในแนวตั้ง)

2. โหมดภาพ (Graphic Mode)

ในโหมดนี้ ข้อมูลที่อยู่ภายในวิดีโอแรมจะหมายถึง การแสดงจุดแต่ละจุดบนจอภาพ โดยตรง กรรมวิธีการแสดงจุดนี้มีสองลักษณะย่อย คือ

1. ใช้การเปิดตารางค่าสี ข้อมูลที่อยู่ภายในวิดีโอแรมจะถูกนำไปเปิดตารางค่าสี เพื่อให้ได้ค่าความสว่างของแม่สีแดง สีเขียว และสีน้ำเงินของจุดภาพแต่ละจุด
2. ข้อมูลที่อยู่ภายในวิดีโอแรมเป็นค่าแม่สี ข้อมูลที่อยู่ภายในวิดีโอแรมจะหมายถึง ค่าความสว่างของแม่สีโดยตรง ดังนั้น ในวิธีนี้จะต้องใช้พื้นที่หลายบิตของข้อมูล สำหรับแทนจุดหนึ่งจุดบนจอภาพ เพื่อให้ได้ภาพที่มีความสมจริง

การแสดงผลบนเครื่องคอมพิวเตอร์พีซี จะแบ่งการทำงานออกเป็นโหมด ในแต่ละหมายเลขของโหมด ก็คือการแสดงในโหมดตัวอักษร หรือโหมดภาพตามจำนวนสี และจำนวนจุดต่างๆ กัน โหมดภาพมาตรฐานบนเครื่องพีซีมีดังนี้

ตารางที่ 2.2 โหมดการแสดงผล

หมายเลขโหมด	ชนิดของโหมด	จำนวนสีที่แสดงได้ในคราวเดียว	ความละเอียด (ตัวอักษร, จุด)	การ์ดแสดงผลที่ทำงาน
0,1	ตัวอักษร	16	40x50	CGA, EGA, VGA
2,3	ตัวอักษร	16	80x25	CGA, EGA, VGA
4,5	ภาพ	4	320x200	CGA, EGA, VGA
6	ภาพ	2	640x200	CGA, EGA, VGA
7	ตัวอักษร	2	80x25	Hercules, VGA
13	ภาพ	16	320x200	CGA, EGA, VGA
14	ภาพ	16	640x200	CGA, EGA, VGA

ตารางที่ 2.2(ต่อ) โหมดการแสดงผล

หมายเลขโหมด	ชนิดของโหมด	จำนวนสีที่แสดง ได้ในคราวหนึ่ง	ความละเอียด (ตัวอักษร, จุด)	การ์ดแสดงผลที่ ทำงาน
15	ภาพ	2	640x350	VGA
16	ภาพ	16	640x350	EGA, VGA
17	ภาพ	2	640x840	VGA
18	ภาพ	16	640x480	VGA
19	ภาพ	256	320x200	VGA

2.11.3 วิดีโอแรม (VideoRAM)

เป็นหน่วยความจำที่ใช้เก็บข้อมูล สำหรับแสดงบนจอภาพ หน่วยความจำส่วนนี้จะอยู่บนการ์ดแสดงผล ข้อมูลที่ใช้แสดงอาจเป็นรหัสแทน ตัวอักษร และรหัสบอกสถานะการแสดงผล หรืออาจเป็นข้อมูล จุด ภาพขึ้นอยู่กับโหมดการแสดงผลขณะนั้น ตำแหน่งหน่วยความจำเริ่มต้นบนพีซี (ตำแหน่งที่พีซีจะเข้าถึง) ของวิดีโอแรมที่ควรทราบมี ดังนี้คือ

- โหมด 7 มีตำแหน่งวิดีโอแรมเริ่มต้นที่ B000:0000
- โหมด 0 ถึง 6 มีตำแหน่งวิดีโอแรมเริ่มต้นที่ B800:0000
- โหมด 13 ถึง 19 มีตำแหน่งวิดีโอแรมเริ่มต้นที่ A000:0000

เราสามารถคำนวณหาตำแหน่งในหน่วยความจำ ที่ตรงกับตำแหน่งตัวอักษรบนจอถ่มณ์ x บรรทัด y ได้ดังสูตร

$$\text{ค่าออฟเซตของตำแหน่งตัวอักษร} = (x+y * \text{จำนวนคอลัมน์ต่อบรรทัด}) * 2$$

$$\text{ค่าออฟเซตของตำแหน่งรหัสควบคุมการแสดงผล} = (x+y * \text{จำนวนคอลัมน์ต่อบรรทัด}) * 2 + 1$$

2.11.4 โหมดภาพความละเอียดสูง

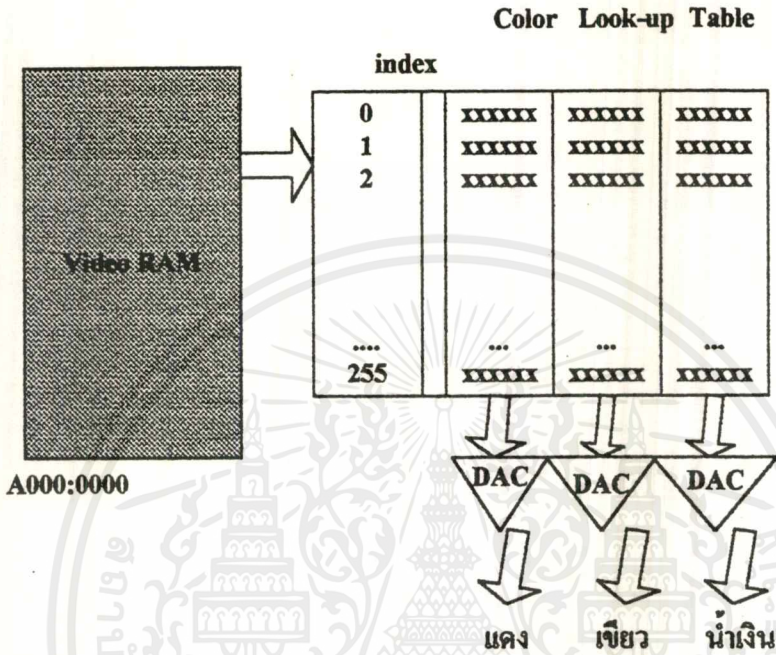
ในโหมดภาพVGAมีโครงสร้างการจัดการหน่วยความจำและการแสดงสีแสงดังรูปที่ 2.6

ในวิดีโอแรมบนการ์ดแสดงผล จะเก็บข้อมูลที่ใช้แสดงอยู่ในรูปของค่าหมายเลขแม่สี

ของตารางเทียบสี(Color Look-up Table) หน่วยความจำในวิดีโอแรมบนการ์ด VGA มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีอยู่ 256 กิโลไบต์ แต่จะใช้ในโฮมคนี้เพียง 64 กิโลไบต์ มีจุดเริ่มต้นที่ตำแหน่ง A000:0000 ใช้ข้อมูลหนึ่งไบต์แทนจุดหนึ่งไบต์แทนจุดหนึ่งจุด



รูปที่ 2.6 โครงสร้างของสีบนจอภาพในโฮมค VGA

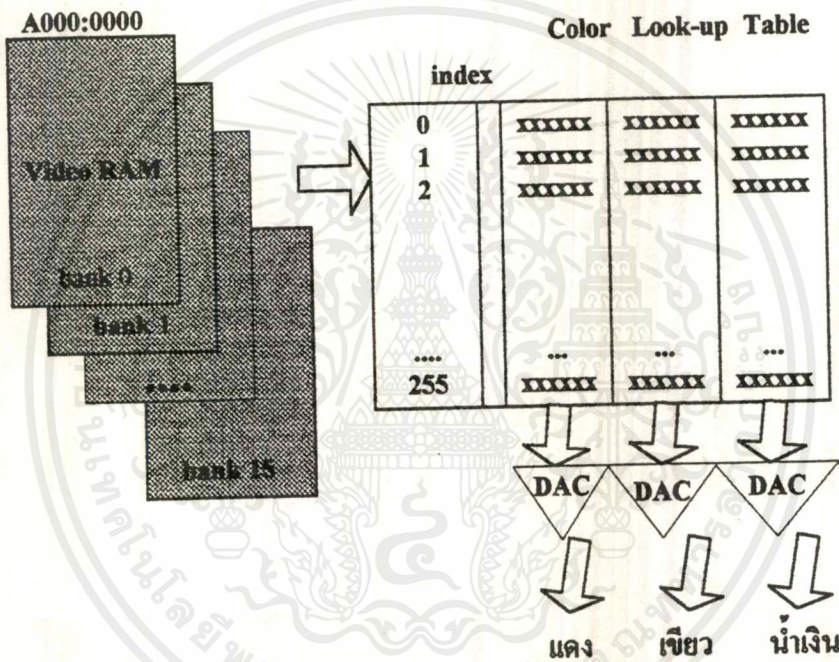
สูตรหาตำแหน่งหน่วยความจำที่ใช้เก็บค่าหมายเลขสี ของจุด(x,y) คือ

$$\text{ค่าออฟเซตของตำแหน่งจุดภาพ} = x+(y*320)$$

ในการแสดงภาพของการ์ดแสดงผล จะใช้ข้อมูลจากวิดีโอแรมมาเปิดตารางเทียบสีซึ่งมีอยู่ 256 เรคอร์ดจะประกอบไปด้วยค่าแม่สีแดง สีเขียว และสีน้ำเงิน มีขนาดค่าละ 6 บิต ดังนั้นการ์ดจึงสามารถสร้างได้ทั้งหมด 262,144 สี แต่เนื่องจากมีตารางเพียง 256 เรคอร์ด และในหนึ่งจุดภาพจะใช้พื้นที่เก็บหนึ่งไบต์ จึงทำให้การ์ดรุ่นนี้สามารถแสดงผลได้คราวละ 256 สีเท่านั้น แม่สีที่ได้จากตารางเทียบสีจะถูกแปลงให้เป็นสัญญาณแอนาลอก โดยวงจรแปลงสัญญาณจากดิจิตอลเป็นแอนาลอก (Digital to Analog Converter - DAC) เพื่อส่งให้จอภาพแสดงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแสดงของการ์ด SVGA มีความซับซ้อนขึ้น โดยซีพียู จะเห็นหน่วยความจำ วิดีโอแรม มีหน่วยความจำซ่อนอยู่ในตำแหน่ง A000 0000 ถึง A000:FFFF อยู่ตั้งแต่ 4 แบนด์ ถึง 16 แบนด์ ในแต่ละแบนด์ของหน่วยความจำมีขนาด 64 กิโลไบต์ ซีพียู จะติดต่อกับหน่วยความจำในแบนด์ใดๆ ได้โดยการใส่ค่าแบนด์ ที่ต้องการติดต่อลงในรีจิสเตอร์สำหรับเลือกแบนด์บนการ์ดแสดงผล ซึ่งซีพียูจะติดต่อผ่านทางพอร์ต ดังนั้น ในกรณีที่มีหน่วยความจำอยู่บนการ์ด 16 แบนด์ นั้นหมายถึง วิดีโอแรมจะมีขนาด 1 เมกะไบต์



รูปที่ 2.7 โครงสร้างการทำงานของการ์ด SuperVGA

การเข้าถึงตำแหน่งข้อมูลของข้อมูลจอในจุด (x,y) บนแบนด์ที่กำหนด มีสูตรดังนี้

$$\text{หมายเลขแบนด์} = (x + (y * \text{จำนวนจุดที่แสดงได้ในแกน } x)) / 0x10000L$$

$$\text{ค่าออฟเซตของหน่วยความจำ} = (x + (y * \text{จำนวนจุดที่แสดงได้ในแกน } x)) \text{ AND } 0xffff$$

บวกตัวเลขมาก ความเร็วในการคำนวณ DFT จึงขึ้นอยู่กับความเร็วและจำนวนครั้งในการคูณตัวเลขเป็นสำคัญ

วิธีในการคำนวณ DFT ให้รวดเร็วยิ่งขึ้น มีชื่อเรียกว่า การแปลงฟูรีเยอร์ (FFT) ซึ่งเป็นผลงานของ กูลีย์ (J.W. Cooley) กับ ทูคีย์ (J.W. Tukey) ที่เสนอไว้ในปี ค. ศ. 1965 ซึ่งทำให้เกิดการพัฒนาต่อมา การคำนวณโดยใช้วิธี FFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้งเท่านั้น หรือ จำนวนครั้ง ในการคูณตัวเลขลดลงไปถึง $N / (\log_2 N)$ เท่า ผลคืออีกประการหนึ่งก็คือ ทำให้การสร้างวงจรเฉพาะ เพื่อการคำนวณ DFT ทำได้ง่าย และคำนวณได้เร็วขึ้น เป็นผลให้การประมวลสัญญาณเชิงเลข สามารถนำไปประยุกต์ ใช้กับระบบเวลาจริงได้ อย่างไรก็ตามการแปลงฟูรีเยอร์เป็นเพียงวิธีการในการคำนวณ ที่ช่วยให้การคำนวณ DFT ซึ่งเป็นการแปลงฟูรีเยอร์ได้รวดเร็วยิ่งขึ้น

โดยทั่วไปสามารถแบ่งการแปลงฟูรีเยอร์ ได้เป็นชนิดใหญ่ๆ 2 ชนิด คือ ชนิดลดทอนทางเวลา (Decimation in Time หรือ DIT) และชนิดลดทอนทางความถี่ (Decimation in Frequency หรือ DIF)

หลักการเบื้องต้นของ FFT

จากสมการการแปลง DFT สำหรับลำดับ $x(m)$ ที่ยาว N จุด สามารถนิยามได้ดังนี้คือ

$$X(k) = \sum_{m=0}^{N-1} x(m) \cdot W^{mk} \quad ; \quad k, m = 0, 1, \dots, N-1 \quad (2.1)$$

โดยจำนวนเชิงซ้อน $W = \exp(-j2\pi / N)$ ในทางวิชาอิเล็กทรอนิกส์ ลำดับ $x(m)$ มักจะเกี่ยวข้องกับสัญญาณใน Time domain ส่วน $X(k)$ เกี่ยวข้องกับสัญญาณใน Frequency domain โดเมนความถี่ หรือเรียกว่า Spectrum สเปกตรัมของสัญญาณ สำหรับการเขียนสมการ (2.1) นั้นการคำนวณหา DFT จะต้องมีการคูณ $1 / N$ ประกอบอยู่ด้วยเสมอ แต่ในที่นี้ละไว้เพื่อสะดวกในการอธิบาย

ตัวอย่างเช่น พิจารณาสำหรับกรณี $N = 4$ จะเขียนสมการได้ดังนี้

$$X(0) = x(0)W^0 + x(1)W^0 + x(2)W^0 + x(3)W^0$$

$$X(1) = x(0)W^0 + x(1)W^1 + x(2)W^2 + x(3)W^3$$

$$X(2) = x(0)W^0 + x(1)W^2 + x(2)W^4 + x(3)W^6$$

$$X(3) = x(0)W^0 + x(1)W^3 + x(2)W^6 + x(3)W^9 \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่รวบรวมไว้สำหรับใช้ภายในห้องเรียนเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านอื่นๆ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ (2.2) สามารถเขียนให้อยู่ในรูปของสมการ matrix ได้คือ

$$\{X\} = \{A\} * \{x\}$$

โดยที่ $\{X\}$ และ $\{x\}$ เป็นเวกเตอร์แนวตั้ง (column vector) ที่ประกอบด้วยลำดับ $X(k)$ และ $x(m)$ ตามลำดับจำนวน N ลำดับ และ $\{A\}$ เป็นเมทริกซ์จัตุรัส (square matrix) ขนาด $N * N$ ที่มีสมาชิกเป็นจำนวนเชิงซ้อน W^{mk}

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (2.3)$$

การคำนวณจากสมการ (2.3) โดยตรงสังเกตเห็นได้ว่า ต้องใช้จำนวนเชิงซ้อน $4 * 4$ หรือ 16 ครั้ง และต้องทำการบวกจำนวนเชิงซ้อนอีก $N(N-1) = 4(4-1) = 12$ ครั้ง

หลักการสำคัญประการหนึ่งของ Fast Fourier Transform หรือ FFT ที่ลดจำนวนครั้งในการคูณจำนวนเชิงซ้อนได้ โดยอาศัยคุณสมบัติของความเป็นคาบของจำนวนเชิงซ้อน W คือ

$$W^{mk} = W^{[mk \bmod (n)]} \quad (2.4)$$

ซึ่ง $[mk \bmod (n)]$ หมายถึง ส่วนที่เหลือหลังจากการหารพจน์ mk ด้วย N โดยอาศัยคุณสมบัติความเป็นคาบนี้ ทำให้สมการ (2.3) อาจเขียนได้เป็น

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & x(0) \\ 1 & W^1 & W^2 & W^3 & x(1) \\ 1 & W^2 & W^0 & W^2 & x(2) \\ 1 & W^3 & W^2 & W^1 & x(3) \end{bmatrix} \quad (2.5)$$

จากคุณสมบัติความเป็นคาบ ทำให้เราสามารถแยกตัวประกอบ ของเมตริกซ์ $\{A\}$ ออกเป็นเมตริกซ์ย่อยหลายเมตริกซ์คู่กัน และสมาชิกภายในเมตริกซ์ย่อยให้มีค่าเป็น “0” มากที่สุด วิธีการแยกตัวประกอบไม่ อาจจะทำโดยตรงจาก $\{A\}$ แต่จะมีการสลับตำแหน่งหรือจัดกลุ่มของเมตริกซ์ (อาจเป็นการสลับสมาชิกตามแนวตั้ง หรือสมาชิกตามแนวนอนอย่างใดอย่างหนึ่ง และการสลับทั้งสองแบบจะให้ผลต่างกัน) ด้วยวิธีการของ การผันกลับบิต (bit reversed) และเมตริกซ์ หลังจัดการสลับแล้วจะนำมาแยกตัวประกอบอีกครั้งหนึ่ง

ในที่นี้จะขออธิบาย วิธีการแยกตัวประกอบ ด้วยการเขียนแทนครรรชนี k และ m จะมีค่าได้เพียง 0, 1, 2 และ 3 เท่านั้น เพราะฉะนั้นสามารถแทนตัวเลข 4ฐานสิบได้ด้วยตัวเลขฐานสอง 2 หลัก คือ

$$k = (k_1, k_2), \quad m = (m_1, m_0) \quad (2.6)$$

โดยที่ $k_1, k_2, m_1,$ และ m_0 เป็นเลขโดดที่มีค่าได้แค่ 0 หรือ 1 เท่านั้น ครรรชนีใน สมการ (2.6) นี้สามารถเขียนได้ใหม่เป็น

$$k = 2k_1 + k_0, \quad m = 2m_1 + m_0 \quad (2.7)$$

เมื่อนำค่าครรรชนี k และ m นี้ไปแทนลงในสมการ (2.1) ทำให้ได้

$$X(k_1, k_0) = \sum_{m_1=0}^1 \sum_{m_0=0}^1 X(m_1, m_0) \cdot W^{(2m_1+m_0) \cdot (2k_1+k_0)} \quad (2.8)$$

จากคุณสมบัติความเป็นคาบของ W ทำให้ได้ความสัมพันธ์

$$\begin{aligned} W^{(2m_1+m_0) \cdot (2k_1+k_0)} &= W^{(2k_1+k_0)2m_1} \cdot W^{(2k_1+k_0)m_0} \\ &= W^{4m_1k_1} \cdot W^{2m_1k_0} \cdot W^{(2k_1+k_0)m_0} \\ &= W^{2m_1k_0} \cdot W^{(2k_1+k_0)m_0} \end{aligned} \quad (2.9)$$

โดยที่ $W^{4m_1k_1} = 1$ เพราะฉะนั้นเขียนสมการ (2.8) ได้ใหม่เป็น

$$\begin{aligned} X(k_1, k_0) &= \sum_{m_0=0}^1 \left\{ \sum_{m_1=0}^1 X(m_1, m_0) \cdot W^{(2m_1k_0)} \right\} W^{(2k_1+k_0)m_0} \\ &= \sum_{m_0=0}^1 \{ X_1(k, m) \} \cdot W^{(2k+k_0)m_0} \end{aligned} \quad (2.10)$$

โดยการสมมติให้ตัวแปร $x_1(k_0, m_0)$ เป็นการคำนวณระหว่างกลางผลจากสมการ (2.8) อาจเขียนเป็นสมการเมทริกซ์ได้คือ

$$\begin{matrix} (k_1, k_0) \\ \begin{bmatrix} X(0,0) \\ X(1,0) \\ X(0,1) \\ X(1,1) \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{matrix} (m_1, m_0) \\ \begin{bmatrix} X(0,0) \\ X(0,1) \\ X(1,0) \\ X(1,1) \end{bmatrix} \end{matrix} \quad (2.11)$$

โดยที่ผลจากการคำนวณระหว่างกลาง และผลลัพธ์สามารถหาได้ตามลำดับ

$$\begin{matrix} (k_1, k_0) \\ \begin{bmatrix} X(0,0) \\ X(0,1) \\ X(1,0) \\ X(1,1) \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{matrix} (m_1, m_0) \\ \begin{bmatrix} X(0,0) \\ X(0,1) \\ X(1,0) \\ X(1,1) \end{bmatrix} \end{matrix} \quad (2.12)$$

และค่า DFT ของลำดับสัญญาณจะเป็น

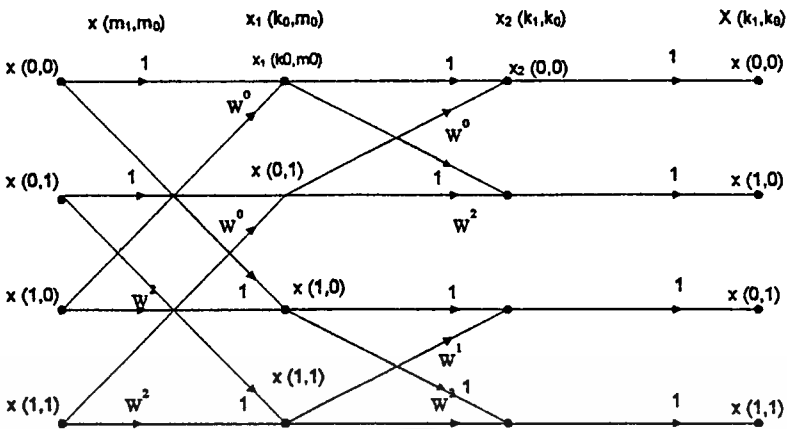
$$\begin{matrix} (k_1, k_0) \\ \begin{bmatrix} X(0,0) \\ X(1,0) \\ X(0,1) \\ X(1,1) \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{matrix} (m_1, m_0) \\ \begin{bmatrix} X(0,0) \\ X(0,1) \\ X(1,0) \\ X(1,1) \end{bmatrix} \end{matrix} \quad (2.13)$$

ผลที่ได้จากสมการ (2.11) ซึ่งเป็นสมการแยกตัวประกอบของ $\{A\}$ นี้จะเห็นว่า สมาชิกตามแนว
นอนของเมทริกซ์ที่ได้จากการแยกตัวประกอบ 2 ตัวเท่านั้น ที่มีค่าไม่เป็นศูนย์และในสองตัวนี้มี
สมาชิกตัวหนึ่งมีค่าเป็นหนึ่งเสมอ ส่วนอีกตัวหนึ่งก็จะเป็นจำนวนเชิงซ้อน แต่ถ้าพิจารณาการ
คูณเมทริกซ์ย่อยแต่ละเมทริกซ์ของ (2.12) จะเห็นว่าต้องการการคูณจำนวนเชิงซ้อนเพียง $N = 4$
โดยที่ N อยู่ในรูปของสองยกกำลังใดๆ ($N = 2$) และถ้าลองเขียนสมการ (2.8) สำหรับกรณีนี้ จะ
เห็นว่าสามารถเขียนเป็นสมการย่อยได้ $\log_2 N = \log_2 2 = 2$ สมการ คือ (2.12) และ (2.13)
เพราะฉะนั้นด้วยวิธีการนี้จำนวนครั้งของการคูณจำนวนเชิงซ้อนจึงมี เท่ากับ $N \log_2 N = 8$ ครั้ง
และบวกจำนวนเชิงซ้อน 8 ครั้ง ซึ่งจำนวนนี้น้อยกว่า กรณีคำนวณโดยตรง ซึ่งใช้จำนวนเชิงซ้อน
16 ครั้ง และบวกจำนวนเชิงซ้อนอีก 12 ครั้ง ดังนั้นจะเห็นว่า การแยกการคำนวณ DFT เป็น
เมทริกซ์ย่อยดังสมการ (2.11) จะทำให้จำนวนครั้งในการคูณจำนวนเชิงซ้อนน้อยลง จำนวน
เชิงซ้อนจะลดลงไปได้อีกดังนี้

$$\begin{aligned}x_1(0,0) &= x(0,0) + W^0 \cdot x(1,0) = x(0,0) + W^0 \cdot x(1,0) \\x_1(1,0) &= x(0,0) + W^2 \cdot x(1,0) = x(0,0) - W^0 \cdot x(1,0)\end{aligned}$$

ผลจากคุณสมบัติของจำนวนเชิงซ้อน $W^2 = -W^0$ ทำให้การคำนวณ $x_1(0,0)$ และ $x_1(0,1)$
ต้องการการคูณจำนวนเชิงซ้อนเพียงครั้งเดียวเท่านั้น ซึ่งทำได้โดยการคำนวณพจน์ $W^0 \cdot x(1,0)$
ก่อน แล้วนำไปบวก และลบกับพจน์ $x(0,0)$ เพื่อให้ได้ลำดับ $x_1(0,0)$ และ $x_1(1,0)$ ตามลำดับ
อย่างไรก็ตามสำหรับกรณีทั่วไป อาจกล่าวได้ว่าจำนวนครั้งในการคำนวณ DFT ขนาด N จุด
โดยใช้ FFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้ง

ลำดับการคำนวณของ FFT ตามสมการ (2.9) เป็นผลงานที่เสนอโดย กูลีซี และ ทุคซี
ต่อมาก็ได้มีผู้เสนอวิธีการคำนวณ FFT แบบอื่นขึ้นมา ผลงานเหล่านี้หลักการใหญ่แล้วเหมือน
กันต่างกันเพียงวิธีการ ในรายละเอียดย่อเท่านั้น ในรูป 2.8 เป็นกราฟการไหลที่เขียนวิธีการ
คำนวณของ FFT ตามสมการ (2.12) โดยที่หัวลูกศรชี้ทิศทางการคำนวณ ส่วนตัวอักษรกำกับ
เป็นตัวคูณ ค่าของสัญญาณที่ต้นทางของลูกศรนั้น และที่บัพ หรือ ปม (Node) เป็นการรวม หรือ
บวกกันของสัญญาณ ส่วน $x_1(k_0, m_0)$ แทนลำดับการคำนวณระหว่างกลาง และ $X(k_1, k_0)$
เป็นค่า DFT ของลำดับสัญญาณ



รูปที่ 2.8 กราฟการไหล วิธีการคำนวณตามสมการ (2.12)

ขั้นตอนวิธีการลดทอนทางเวลา (Decimation in time หรือ DIT)

วิธีการที่เสนอโดย คูลิย์ และ ทูคีย์ ซึ่งได้กล่าวมาแล้วนั้น ความเป็นจริงเป็นการแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา $x(m)$ ที่มีขนาด N จุด ออกเป็นสองลำดับสัญญาณที่มีความยาว $N/2$ จุดเท่ากัน ทั้งสองลำดับนี้ให้ชื่อเรียกว่า ลำดับสัญญาณคู่ และ ลำดับสัญญาณคี่ โดยที่ลำดับสัญญาณคู่ เกิดจากการเอาลำดับในตำแหน่งเป็นเลขคู่มาเรียงกัน ที่เหลือก็เป็นลำดับสัญญาณคี่ ดังนั้นถ้าเรานิยามให้ $x_2(m)$ เป็นลำดับคู่และลำดับคี่เป็น $x_0(m)$ ตามลำดับ เพราะฉะนั้น

$$\begin{aligned} x_2(m) &= x(2m) & ; m &= 0, 1, \dots, (N/2) - 1 \\ x_0(m) &= x(2m+1) & ; m &= 0, 1, \dots, (N/2) - 1 \end{aligned} \tag{2.14}$$

ด้วยการจัดแบ่งเช่นนี้ ถ้าให้ W_N แทนค่า W ของลำดับยาว N จุด ทำให้การคำนวณการแปลง DFT ของลำดับสัญญาณ $x(m)$ ที่ยาว N จุดเขียนใหม่ได้เป็น

$$\begin{aligned} X(k) &= \sum_{m=0}^{N-1} x_2(m) (W_N)^{km} + \sum_{m=0}^{N-1} x_0(m) (W_N)^{km} \\ &= \sum_{m=0}^{(N/2)-1} x(2m) (W_N)^{2km} + \sum_{m=0}^{(N/2)-1} x(2m+1) (W_N)^{(2m+1)k} \end{aligned} \tag{2.15}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่เขียนให้พจน์ $(W_N)^2$ เป็น

$$(W_N)^2 = \{ \exp (j2\pi/N^2) \} = \exp (j2\pi/N/2) = W_{N/2}$$

ซึ่ง $W_{N/2}$ หรือค่า W ของลำดับยาว $N/2$ จุด เพราะฉะนั้น (2.15) จัดพจน์ใหม่ได้ คือ

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} x_2(m) (W_{N/2})^{km} + (W_N)^k \sum_{m=0}^{(N/2)-1} x_0(m) (W_{N/2})^{km} \\ X(k) &= X_1(k) + (W_N)^k X_2(k) \end{aligned} \quad (2.16)$$

โดยที่ $X_1(k)$ และ $X_2(k)$ แทนผลการแปลง DFT ขนาด $N/2$ จุดของลำดับ $x_2(m)$ และ $x_0(m)$ ตามลำดับ สมการที่ (2.16) แสดงให้เห็นว่าการคำนวณ DFT ขนาด N จุด สามารถแบ่งคำนวณย่อยออกเป็น การคำนวณ DFT ขนาด $N/2$ จุดสองอันดับได้ และข้อสำคัญคือ การคูณจำนวนเชิงซ้อน จะลดลงเหลือ $2(N/2)^2 = N^2 / 2$ ครั้ง ซึ่งจะเห็นว่า ลดเวลาการคำนวณลงไปถึง 50 เปอร์เซ็นต์ โดยอาศัยหลักการเดียวกัน ถ้าเราแบ่งทอนลำดับ $x_2(m)$ และ $x_0(m)$ ออกเป็นลำดับคู่และลำดับคี่ลงไปอีกตามลำดับ จนในที่สุดเหลือเป็นลำดับขนาด 2 จุด หรือกล่าวได้ว่าการคำนวณการแปลง DFT ขนาด N จุด ทำได้โดยการคำนวณการแปลง DFT ขนาด 2 จุด จำนวน $N/2$ ภาคด้วยกัน ข้อสังเกตที่สำคัญก็คือการขอยเพื่อลำดับ $x(m)$ ออกเป็นที่ละครั้งจนเหลือการคำนวณ DFT ขนาด 2 จุดนี้ สำหรับ ขนาด N ลำดับ จะทำการแบ่งออกได้ $\log_2 N$ ครั้ง

ปัญหาต่อมาก็คือ เมื่อแบ่งย่อยลงไปแล้ว จะเอาการแปลง DFT ขนาด 2 จุด จำนวน $N/2$ ภาคนี้มาประกอบกัน หรือรวมกันเพื่อให้เป็นการคำนวณ DFT ขนาด N จุดได้อย่างไร เพราะถ้าหากนำมาประกอบกันอย่างไม่มีการเรียงลำดับ ผลการคำนวณ DFT ที่ได้ก็จะมีค่า ผิดพลาดไป ตัวอย่าง เช่น การนำเอา DFT ขนาด $N/2$ จุดมาต่อกันโดยตรงตามสมการ (2.17) ย่อมมีข้อผิดพลาดเพราะว่าทั้ง $X_1(k)$ และ $X_2(k)$ เป็น DFT ขนาด $N/2$ จุดที่นิยามเฉพาะสำหรับช่วง $0 \leq k \leq N/2$ เท่านั้น การนำมาประกอบกันสำหรับลำดับสัญญาณ N ลำดับ อย่างถูกต้อง เราต้องทำการนิยามค่าของสมการ (2.16) สำหรับค่า $k > N/2$ ด้วยซึ่งทำได้โดยเขียน

$$\begin{aligned} X_0(k) &= X_1(k) + (W_N)^2 X_2(k) & ; 0 \leq k \leq (N/2) - 1 \\ &= X_1(k-N/2) + (W_N)^k X_2(k-N/2) & ; N/2 \leq k \leq N-1 \end{aligned} \quad (2.17)$$

พจน์ $(W_N)^k$ ในสมการ (2.17) มีชื่อเรียกว่า ตัวประกอบหมุน (twiddle factor หรือ phase หรือ rotation factor) นี้เป็นส่วนสำคัญที่จะรวม DFT ขนาด 2 จุด หรือ DFT ขนาด $N/2$ จุด ที่จะนำมาประกอบกันเข้าเป็น DFT ขนาด N จุดได้เหมือนเดิม จากสมการ (2.17) เขียนในรูปทั่วไปได้คือ

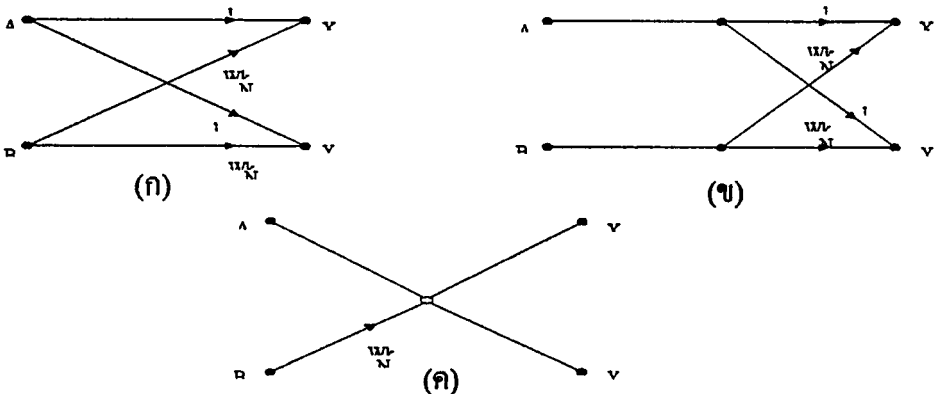
$$X_1(k) = X_{1-1}(k) + (W_N)^k X_{1-1}(k + N/2^{-1}) \tag{2.18}$$

$$X_1(k+N/2) = X_{1-1}(k) - (W_N)^k X_{1-1}(k + N/2^{-1}) \tag{2.19}$$

สมการที่ได้นี้เป็นหลักการสำคัญสำหรับ FFT แบบนี้ ในการคำนวณหา DFT ของลำดับคู่หนึ่ง จะประกอบด้วยลำดับ $X(k)$ ในสมการ (2.18) และลำดับ $X(k)$ ใน (2.19) ซึ่งห่างออกไปจากลำดับ $X(k)$ ใน (2.18) ไป $N/2$ จุด โดยใช้การคูณจำนวนเชิงซ้อนเพียงครั้งเดียวเท่านั้น จากผลนี้ เราสามารถนำไปสร้างหน่วยคำนวณที่มีชื่อเรียกว่า หน่วยผีเสื้อ (butterfly unit) โดยหน่วยคำนวณนี้มีข้อมูลเข้าสองข้อมูลคือ A และ B และให้ข้อมูลออกเป็น X และ Y เป็น

$$\begin{aligned} X &= A + (W_k)^k \\ Y &= A - (W_k)^k \end{aligned} \tag{2.20}$$

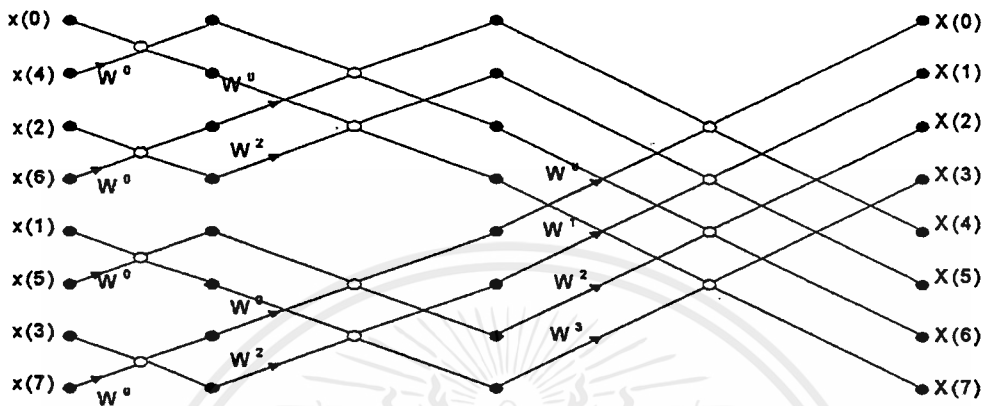
โดยที่การทำงานของหน่วยผีเสื้อนี้ สามารถเขียนอธิบายแทนด้วย กราฟการไหล (signal flow graph) แสดงดังรูป 2.8



รูปที่ 2.9 หน่วยผีเสื้อของการคำนวณตามขั้นตอนวิธีลดทอนทางเวลา

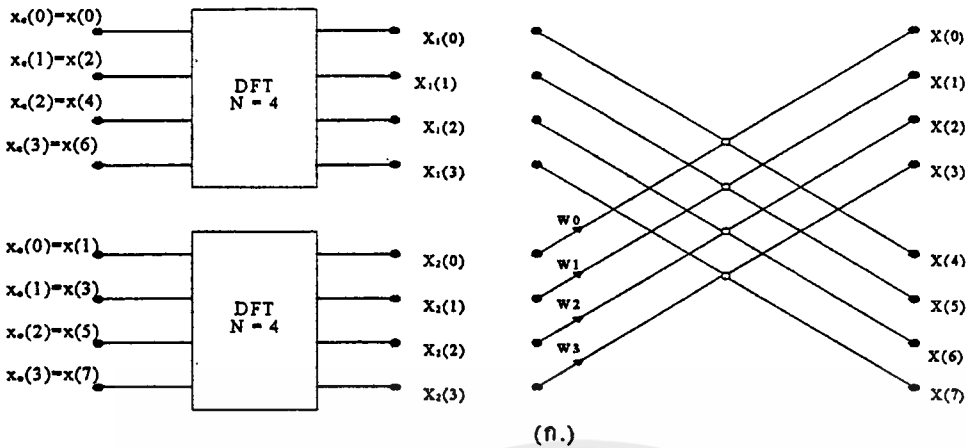
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปแสดงตัวอย่างการคำนวณ DFT แบบ 8 จุด โดยอาศัยวิธีการที่อธิบายมาแล้ว ขั้นตอนในการคำนวณได้แสดงเป็นลำดับด้วยกราฟการไหลของสัญญาณดังรูป

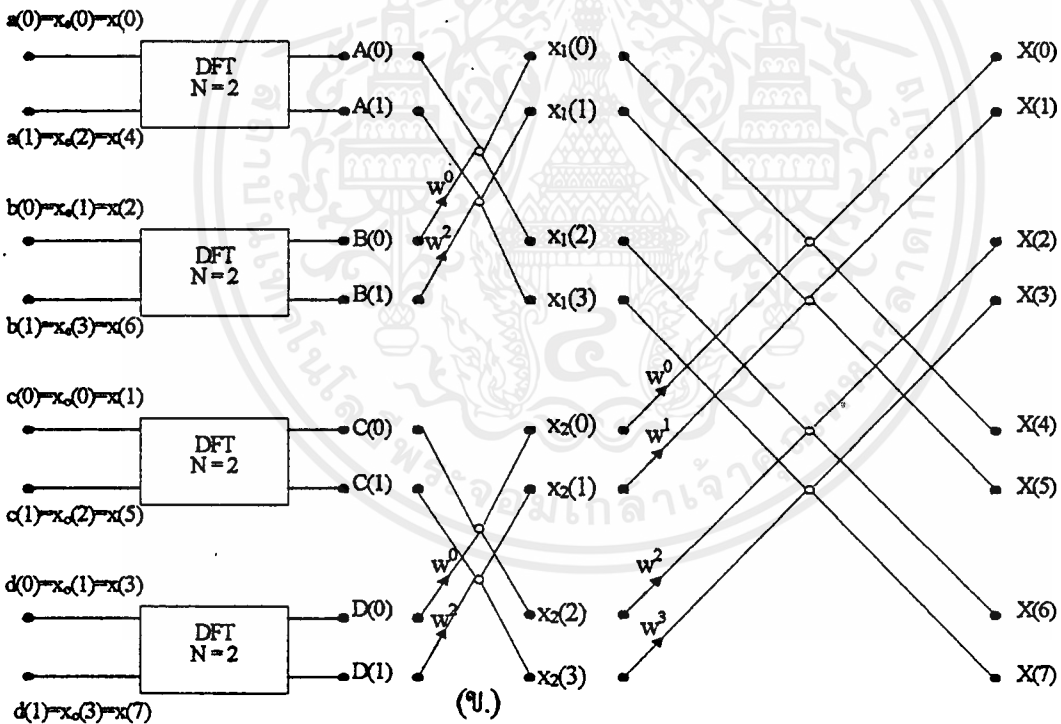


รูปที่ 2.10 กราฟการไหลของสัญญาณ 8 จุด

ในขั้นแรกลำดับของสัญญาณเข้า $x(n)$ จะถูกแบ่งออกเป็น ลำดับคู่ $X_E(m)$ และ ลำดับคี่ $X_O(m)$ อย่างละสี่ลำดับ ดังรูป (2.11ก) ต่อจากนั้นลำดับ $X_E(m)$ และ $X_O(m)$ ก็จะถูกแบ่งย่อยลงไปอีกดังรูป (2.11ข) ซึ่งในที่สุดการคำนวณ DFT ขนาด 8 จุด ก็จะถูกแบ่งย่อยออกเป็น การคำนวณ DFT ขนาด 2 จุด 4 ภาคด้วยกัน ดังรูป (2.10) ซึ่งเป็นกราฟการไหลของสัญญาณ สำหรับการคำนวณ DFT ขนาด 8 จุดอย่างสมบูรณ์ สังเกตว่าในกรณีนี้ $N = 8$ ทำให้ $\log_2 8 = 3$ การคำนวณจึงแบ่งเป็น 3 ภาคย่อยแต่ละภาคประกอบด้วยหน่วยผีเสื้ออยู่ $N/2 = 4$ หน่วย และหน่วยผีเสื้อแต่ละหน่วย ใช้ในการคูณเชิงซ้อน 1 ครั้ง เพราะฉะนั้นการคำนวณทั้งหมดมีการคูณตัวเลขเท่ากับ $(8/2) \log_2 8 = 12$ ครั้ง เมื่อเปรียบเทียบกับ การคำนวณ ปกติที่ต้องมีการคูณกันถึง $8 \cdot 8 = 64$ ครั้ง



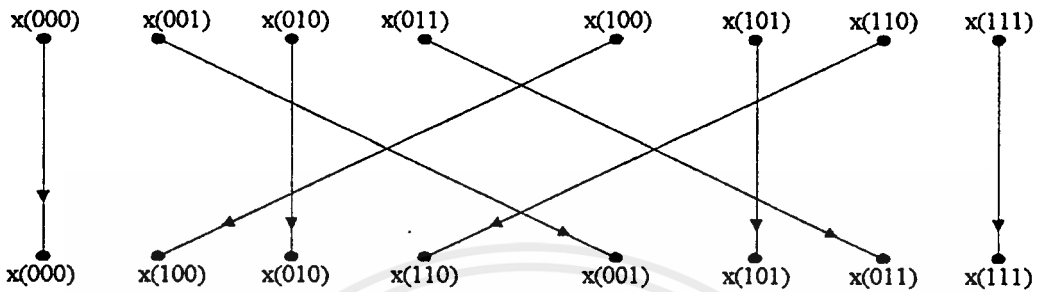
รูปที่ 2.11 วิธีแบบ DIT สำหรับ FFT แบบ 8 จุด



รูปที่ 2.11 (ต่อ) วิธีแบบ DIT สำหรับ DFT แบบ 8 จุด

ตามรูป (2.10) มีข้อนำสังเกตคือ ลำดับสัญญาณเข้า $x(n)$ ไม่ได้ถูกจัดเรียงอย่างต่อเนื่องหรือตามธรรมชาติ แต่มีการสลับตำแหน่งกันอย่างมีหลักเกณฑ์ เป็นไปตามวิธีการที่เรียกว่า การผันกลับบิต คือถ้าเราแทนคอร์ด n ของลำดับ $x(n)$ ด้วยเลขฐานสอง เช่นในกรณีนี้ $N = 8$ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็ต้องแทนด้วยเลขฐานสอง 3 บิต จากนั้นการจัดลำดับ $x(n)$ ใหม่จะได้จากการผันกลับบิตของเลขฐานสองที่แทนครรรชนี n ดังรูป (2.12)



รูปที่ 2.12 แสดงการสลับตำแหน่งของลำดับ $x(n)$ ด้วยการผันกลับบิต

วิธีการของ FFT แบบนี้เป็นกรลดเวลาการคำนวณ โดยการตัดทอนลำดับใน Time domain หรือ $x(n)$ ออกเป็นกลุ่มย่อย โดยที่แต่ละกลุ่มประกอบด้วยลำดับ $x(n)$ สองลำดับที่เป็น dual node pair กัน การจัดกลุ่มนี้คล้ายกับการสุ่มตัวอย่างลำดับเดิมอีกครั้งหนึ่งด้วยอัตราการสุ่มตัวอย่างที่ต่ำกว่า และถ้าหากเราถือว่าแต่ละกลุ่มลำดับข้อมูลใหม่ที่จัดทำขึ้นมา ต่างเป็นลำดับข้อมูลชุดหนึ่งแล้ว ก็เท่ากับว่าเราได้ตัดทอนลำดับในโดเมนเวลาลงไป เป็นกลุ่มลำดับข้อมูลย่อยหลายลำดับ ดังนั้นวิธีการคำนวณแบบนี้จึงมีชื่อเรียกว่า การลดทอนทางเวลา

2.13 การให้นำหนัก

รูปแบบของการเฉลี่ย โดยปกติมีอยู่หลายวิธีเพื่อให้ได้เส้นของพารามิเตอร์เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้อง ทางเลือกหนึ่งที่ใช้คือวินโดว์ วินโดว์มีอยู่หลายแบบ แต่ที่จะกล่าวถึงคือ แฮนมิงวินโดว์ ซึ่งมีสมการ ดังนี้

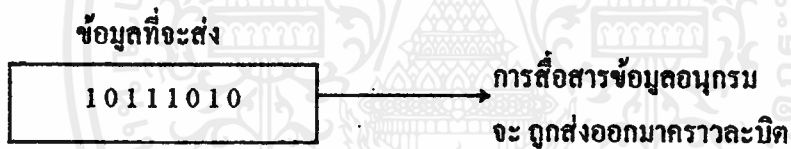
$$W(n) = 0.54 - 0.46 \cos(2\pi n/N-1); 0 \leq n \leq N-1 \quad (2.21)$$

$$= 0 \quad ; n > N-1 \quad (2.22)$$

การกำหนดหน้าตาแบบแฮมมิงนี้ จะช่วยเพิ่มความลาดของส่วนต้น และปลายของ สัญญาณ เพื่อให้ส่วนกลางมีน้ำหนักมากกว่าส่วนอื่นๆ เมื่อนำไปคำนวณด้วยฟาสต์ฟูเรียร์แล้ว จะได้ค่าที่ถูกต้องมากที่สุด

2.14 การสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอนุกรม เป็นการรับและส่งข้อมูลคราวละ 1 บิต เป็นลำดับจนสิ้นสุดกลุ่มข้อมูล การสื่อสารแบบนี้ แตกต่างจากการสื่อสารแบบขนาน เนื่องจากการส่งข้อมูลแบบขนานจะโอนย้ายข้อมูลพร้อมกัน จึงต้องใช้จำนวนเส้นของสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรม ต้องการเส้นสัญญาณเพียง 2 หรือ 3 เส้นเท่านั้น ดังนั้น การสื่อสารแบบขนานจึงไม่เหมาะในการสื่อสารกับอุปกรณ์ภายนอกที่เป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก ดังรูปที่ 2.13



รูปที่ 2.13 การส่งข้อมูลแบบอนุกรม จำนวน 8 บิต จะส่งทีละบิตจนครบ 1 ไบต์

พอร์ตสื่อสารข้อมูลอนุกรม RS-232

2.14.1 คุณสมบัติของ RS-232

1. อัตราการรับส่งข้อมูล 0-20,000 บิตต่อวินาที
2. ระดับแรงดันเอาต์พุตสูงสุดในภาวะไม่มีโหลด
 - -25 โวลต์ (ลอจิก "1")
 - +25 โวลต์ (ลอจิก "0")
3. ระดับแรงดันเอาต์พุตสำหรับโหลด 3-7 กิโลโห์ม
 - ลอจิก "1" -15 โวลต์ (7 กิโลโห์ม)
 - ลอจิก "1" -5 โวลต์ (3 กิโลโห์ม)
 - ลอจิก "0" +15 โวลต์ (7 กิโลโห์ม)
 - ลอจิก "0" +5 โวลต์ (3 กิโลโห์ม)

4. กระแสเอาต์พุต เมื่อลัดวงจร สูงสุด 500 มิลลิแอมป์
5. เอาต์พุตอิมพีแดนซ์เมื่อไม่จ่ายไฟเลี้ยง ต่ำสุด 300 โอห์ม
6. สลั้วเรตทางเอาต์พุตสูงสุด 30 โวลต์ต่อไมโครวินาที
7. ความต้านทานอินพุตของภาครับ

สูงสุด 7 กิโลโอห์ม

ต่ำสุด 3 กิโลโอห์ม

8. ค่าความจุอินพุตของภาครับ สูงสุด 2,500 พิโกฟารัด
9. ย่านแรงดันอินพุตของภาครับ -25 โวลต์ ถึง +25 โวลต์

2.14.2 การจัดขาสัญญาณของ RS-232

มีด้วยกัน 2 แบบคือ แบบ 9 ขา และ 25 ขา ดังนี้

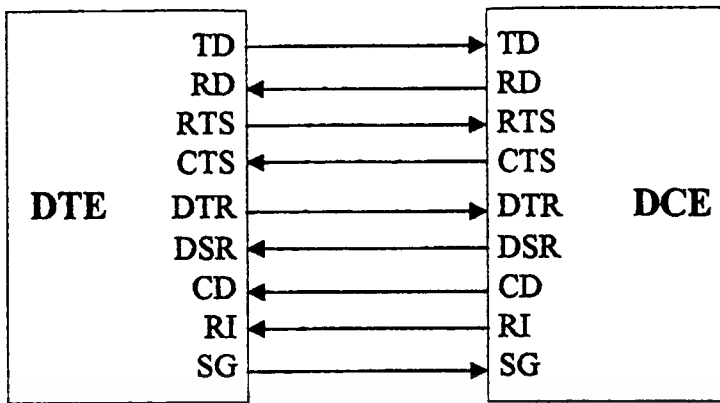
ตารางที่ 2.3 ขาสัญญาณของ RS-232 ทั้งแบบ 9 และ 25 ขา

ชื่อสัญญาณ	หมายเลขขาในแบบ 9 ขา	หมายเลขขาในแบบ 25 ขา
TD Transmitted Data	3	2
RD Received Data	2	3
RTS Request to Send	7	4
CTS Clear to Send	8	5
DSR Data Set Ready	6	6
SG Signal Ground	5	7
CD Carrier Detect	1	8
DTR Data Terminal Ready	4	20
RI Ring indicator	9	22

2.14.3 การเชื่อมต่อสัญญาณของ RS-232

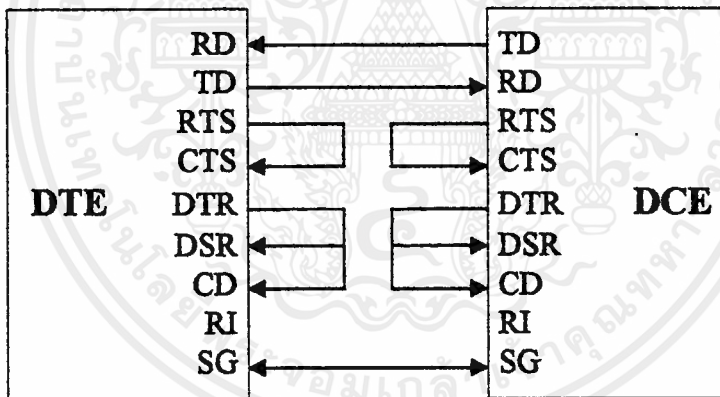
มีลักษณะการเชื่อมต่อ 2 แบบด้วยกัน แบบแรกเป็นการเชื่อมต่อกันระหว่างอุปกรณ์ DTE (Data Terminal Equipment) เช่น คอมพิวเตอร์ กับอุปกรณ์ DCE (Data Circuit Terminal Equipment) เช่น โมเด็ม ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การต่ออุปกรณ์ DTE เข้ากับ DCE

แบบที่สองเป็นการต่อระหว่างอุปกรณ์ DTE ด้วยกันเองทั้งสองฝ่าย หรือเชื่อมต่อคอมพิวเตอร์ 2 ตัวเข้าด้วยกัน เพื่อส่งผ่านถ่ายเทข้อมูล มีรูปแบบการต่อดังรูปที่ 2.15



รูปที่ 2.15 การต่ออุปกรณ์ DTE เข้ากับ DTE

1. ระยะเวลาของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารแบบอนุกรมเป็นการรับส่งข้อมูลเป็นกลุ่มของบิตข้อมูล (Bit Stream) ทีละบิต ดังนั้นจึงต้องพิจารณาถึงความเร็วในการรับส่งข้อมูลเป็นอันดับแรก โดยทั่วไปความเร็วจะระบุกันในหน่วยที่เรียกว่า อัตราบอด (Baud Rate) คือ เป็นหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที ตามค่ามาตรฐาน

ตารางที่ 2.4 อัตราบอดและช่วงเวลาของแต่ละบิตข้อมูลในการสื่อสารข้อมูลอนุกรม

อัตราบอด	ช่วงเวลาของแต่ละบิต
110	9.091 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	833 μ s
2400	417 μ s
4800	208 μ s
9600	104 μ s
19200	52.08 μ s

2. รูปแบบของการส่งข้อมูลอนุกรม

วิธีการที่จะทำให้ข้อมูลในการสื่อสารอนุกรมถูกต้องมากยิ่งขึ้นนั้น จะใช้วิธีการเพิ่มบิตข้อมูลบางอย่างรวมไปกับการส่งข้อมูลจริง คือ

2.1 บิตเริ่มต้น (Star Bit)

มีหน้าที่สำหรับบอกให้ฮาร์ดแวร์ทราบว่าถึงตำแหน่งเริ่มต้นของบิตข้อมูลกลุ่มใหม่แล้ว เพื่อที่จะปรับจังหวะของสัญญาณรับข้อมูลให้ตรงกัน ดังนั้นบิตเริ่มต้นนี้ จึงถูกเพิ่มเข้าไปก่อนมีการส่งข้อมูลจริง โดยทั่วไปแล้วค่าของบิตเริ่มต้นมักจะมีระดับลอจิกตรงข้ามกับระดับลอจิกของสายส่งข้อมูลเมื่อไม่มีการส่งข้อมูล (Idle State)

2.2 บิตพาริตี (Parity Bit)

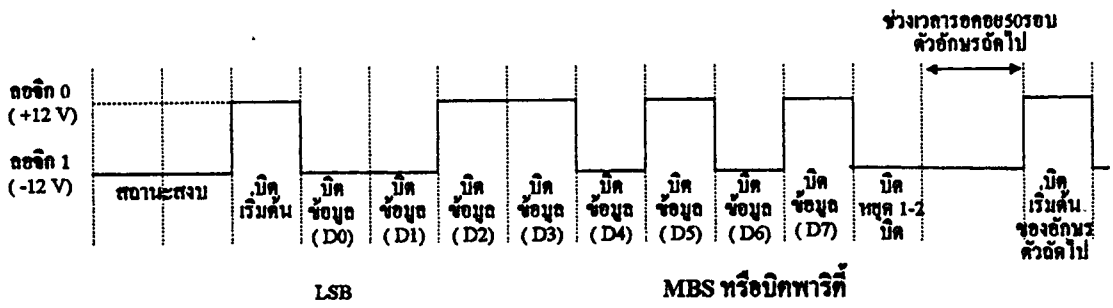
มีหน้าที่เพื่อตรวจสอบความผิดพลาดของข้อมูล โดยจะนำไปต่อท้ายบิตข้อมูล ค่าของบิตนี้ ขึ้นอยู่กับจำนวนของบิตข้อมูลที่เป็น 1 ซึ่งพาริตีนี้จะเป็นได้ 2 ลักษณะ คือ พาริตีคู่ (Even Parity) และ พาริตีคี่ (Odd Parity)

2.3 บิตหยุด (Stop Bit)

บิตหยุด เป็นบิตที่เพิ่มเข้าไปเพื่อบอกขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตหยุดนี้จะมีมากกว่า 1 บิตก็ได้ คือ 1 บิต, 1 บิต ครึ่ง และ 2 บิต ดังนั้นในกรณีของการส่งข้อมูล 8 บิต

ผังรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 รูปแบบของสัญญาณข้อมูลอนุกรมที่ใช้ในการสื่อสารข้อมูลอนุกรม



บทที่ 3

หลักการออกแบบฮาร์ดแวร์และซอฟต์แวร์

ในการออกแบบและสร้างเครื่องวิเคราะห์แถบความถี่โดยใช้ TMS320C50 ได้แบ่งการออกแบบเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของ ฮาร์ดแวร์ และส่วนของซอฟต์แวร์ ซึ่งในแต่ละส่วนมีการแบ่งการทำงาน แยกออกเป็นส่วนย่อยๆ เพื่อความสะดวกในการออกแบบ และการสร้าง ตลอดจนการตรวจเช็ค และทดสอบการทำงาน

ในส่วนของซอฟต์แวร์ มีการใช้ภาษา 2 ภาษา ในการเขียนซอฟต์แวร์ คือ โปรแกรมภาษาซี และโปรแกรมภาษาแอสเซมบลี (Assembly) โดยภาษาซี ให้เขียนโปรแกรมการสร้างภาพกราฟฟิกส์ ที่จะแสดงออกทางจอคอมพิวเตอร์เพื่อบอก ขนาดของแรงดัน และ ความถี่ของสัญญาณที่ทำการวัด ในส่วนของโปรแกรมควบคุมบอร์ด DSP ซึ่งมีชิพ DSP เป็นตัวประมวลผลนั้น ใช้โปรแกรมภาษาแอสเซมบลีเขียนโปรแกรมในการควบคุม

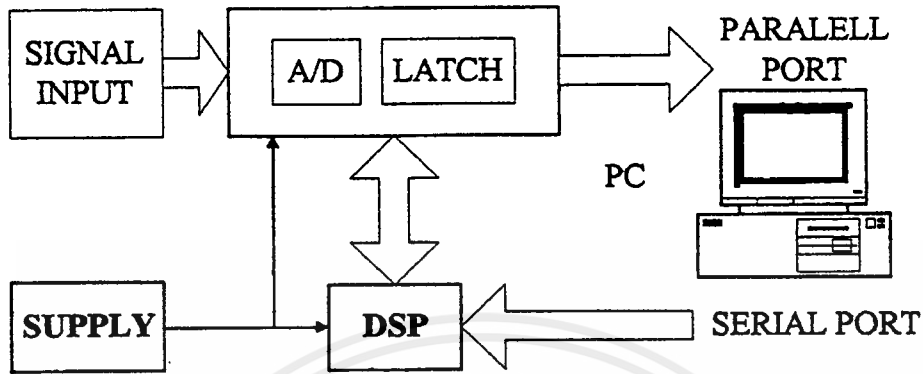
3.1 หลักการออกแบบฮาร์ดแวร์

การออกแบบทางด้านฮาร์ดแวร์นั้น ได้ออกแบบเป็นแนวความคิดแล้วเขียนเป็นผังการทำงานก่อนหลังจากนั้นจึงศึกษาหาข้อมูล จึงได้ผังการทำงานที่สมบูรณ์ แสดงดังรูปที่ 3.1

จากผังการทำงาน สเปคตรัมอะนาลิเซอร์ สามารถแบ่งส่วนต่างๆ ของฮาร์ดแวร์ ออกเป็น 3 ส่วน คือ คอมพิวเตอร์ บอร์ด DSP และส่วนอินเตอร์เฟซ

ฮาร์ดแวร์ของเครื่องที่ใช้วิเคราะห์สเปคตรัมได้เลือกใช้ชิพ DSP เบอร์ TMS320C50 ซึ่งมีจำหน่ายรวมเป็นบอร์ด ซึ่งเรียกว่าบอร์ด DSP Starter Kit เรียกชื่อว่า บอร์ด DSP รูปร่างภายนอกของบอร์ด DSP แสดงดังรูปที่ 3.2

อุปกรณ์โดยส่วนใหญ่ เป็นแบบติดตั้งบนผิวหน้าลงแผ่นวงจรพิมพ์ จะมีเพียงตัวเก็บประจุค่าสูง และไอซีเรกกูเลเตอร์ ที่เป็นแบบธรรมดาบนบอร์ด DSP จะมีชิพที่เป็นหัวใจหลักๆ อยู่ 2 ตัว คือ TMS320C50 ซึ่งเป็น DSP โปรเซสเซอร์ และชิพแปลงสัญญาณจาก ดิจิตอลเป็นแอนาล็อก เบอร์ TLC32040CFN



รูปที่ 3.1 ฟังก์การทำงานของสเปคตรัมอะนาไลเซอร์

บอร์ดมีขนาด 2.5 x 3.5 นิ้ว มีคอนเน็คเตอร์ชนิด D ตัวเมียเพื่อใช้ในการเชื่อมต่อกับคอมพิวเตอร์ ผ่านทางพอร์ตอนุกรม RS232 1 ชุด มีแจ๊ค RCA สำหรับการรับ และการส่งสัญญาณแอนาล็อก ซึ่งมีลักษณะดังนี้

1. มีความเร็วสูงถึง 40 MHz.
2. มีหน่วยความจำ RAM ในชิพ ขนาด 10 กิโลเวิร์ด
3. ประมวลผลข้อมูลได้สูงถึง 28 ล้านคำสั่งต่อวินาที
4. มีชิพแอนาล็อกอินเตอร์เฟสเบอร์ TLC32040 ขนาด 14 บิต เป็นทั้ง A/D และ D/A
5. มีแจ๊ค RCA สองแจ๊ค เป็นทั้งอินพุต และเอาต์พุตโดยต่อกับอุปกรณ์ ไมค์โครโฟน ลำโพง หรืออุปกรณ์ แอนาล็อกอื่นๆ ได้โดยตรง
6. มีพอร์ต RS-232 อนุกรมสำหรับต่อกับคอมพิวเตอร์
7. ใช้แหล่งจ่าย 9 VAC
8. ตัวบอร์ดมีขนาดเล็กกระทัดรัด แสดงดังรูปที่ 3.2

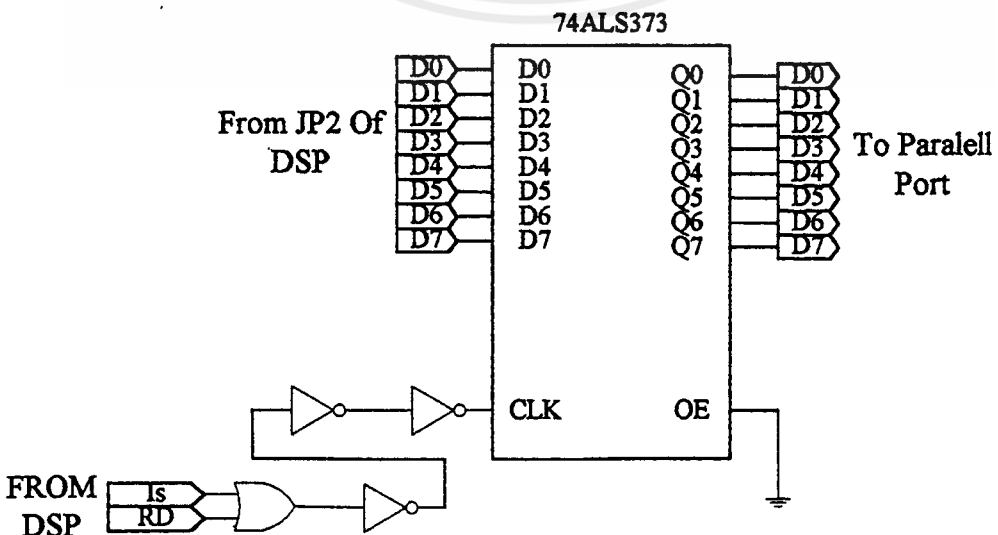
ขนาด และความถี่ของสัญญาณออกมาเป็นค่าตัวเลข และออกแบบให้มีการส่งข้อมูลในลักษณะ ขนานส่งไปยังพอร์ตขนานของคอมพิวเตอร์ แล้วไปแสดงผลที่จอ

ในส่วนของอินเตอร์เฟซ จะเป็นการอินเตอร์เฟซกับ A/D และคอมพิวเตอร์ โดยจะกล่าว แยกเป็นส่วนๆ ดังนี้

3.1.1 วงจรแลตซ์ หรือส่วนการอินเตอร์เฟซกับคอมพิวเตอร์

วงจรแลตซ์จะทำหน้าที่คงสถานะของข้อมูลเดิมไว้ จนกว่าควบคุมของวงจรจะได้รับการกระตุ้นให้แอสติฟ ข้อมูลที่ค้างไว้จึงจะเปลี่ยนแปลงตามอินพุต และจะค้างสถานะนั้นตลอด ไปจนกว่าควบคุมจะได้รับการกระตุ้นให้แอสติฟอีกครั้ง ในที่นี้จะใช้วงจรแลตซ์นี้ทำหน้าที่ ในการคงสถานะของข้อมูลที่ผ่านการประมวลผล จากชิพ DSP แล้ว ซึ่งจะส่งให้เครื่อง คอมพิวเตอร์ทางพอร์ตขนาน เพื่อให้เกิดความสัมพันธ์กันระหว่างการรับ และการส่งข้อมูลของ อุปกรณ์ทั้ง 2 ที่เชื่อมต่อกันนอกจากนี้ยังทำหน้าที่ในการแยกบิตข้อมูลที่ใช้ในการรับส่งข้อมูล ระหว่างชิพ DSP กับวงจร A/D และการรับส่งข้อมูลระหว่างชิพ DSP กับ คอมพิวเตอร์

วงจรแลตซ์ ในที่นี้ใช้ไอซี TTL เบอร์ 74LS373 ซึ่งภายใน จะประกอบไปด้วย D FlipFlop 8 ตัว โดยมีควบคุม 2 ขา ซึ่งจะแอสติฟที่ขอบขาขึ้นของสัญญาณควบคุม ที่ได้รับจาก การตีโค้ดสัญญาณการส่งข้อมูลออกทางเอาต์พุตของชิพ DSP (สัญญาณ IS และ WR) ซึ่ง สัญญาณนี้จะถูกส่งออกมา เมื่อ DSP ทำการประมวลผลสัญญาณเสร็จใน 1 ครั้ง และต้องการจะ ส่งข้อมูลนี้ให้กับคอมพิวเตอร์



รูปที่ 3.4 วงจรแลตซ์

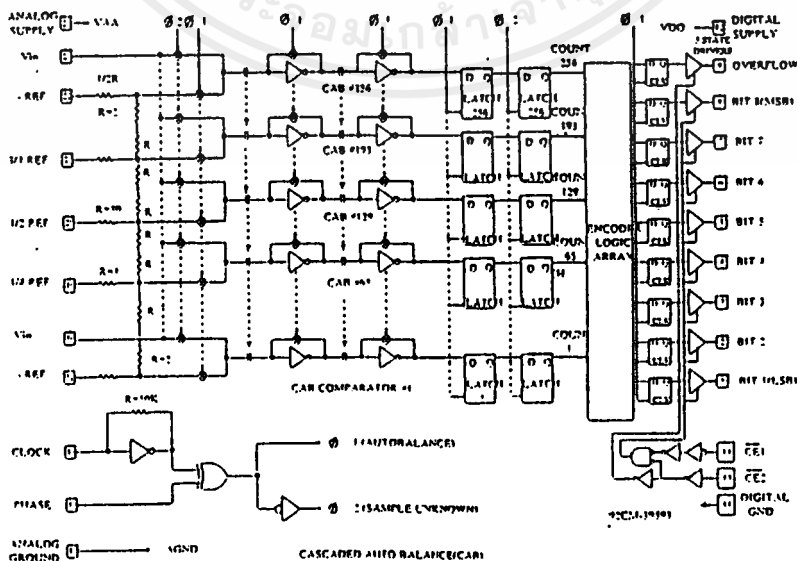
3.1.2 วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล

เป็นวงจรที่ใช้ในการสุ่มตัวอย่างสัญญาณอินพุตที่ต้องการวิเคราะห์ โดยเป็นวงจรสุ่มสัญญาณแบบขนานขนาด 8 บิต โดยใช้ความถี่ในการสุ่มสัญญาณ 4 MHz. ซึ่งจากทฤษฎีการสุ่มสัญญาณแล้ว จะทำให้สามารถวิเคราะห์ตัวอย่างสัญญาณความถี่ได้สูงสุด 2 MHz.

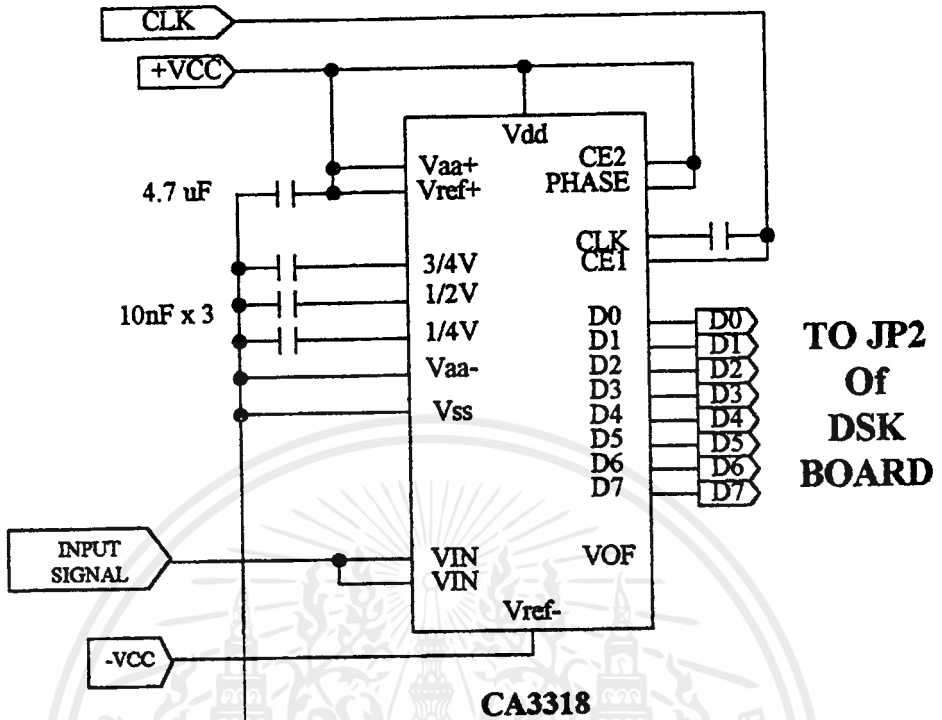
วงจรนี้จะใช้ไอซี A/D เบอร์ CA3318C ซึ่งเป็นไอซี ที่มีหน้าที่แปลงสัญญาณแอนาลอกให้เป็นสัญญาณดิจิทัล เพื่อนำไปประมวลผลทางคณิตศาสตร์โดยมีขนาด 8 บิต สามารถสุ่มตัวอย่างสัญญาณได้ 256 ระดับ คือ 00H - FFH

คุณสมบัติไอซี CA 3318C

1. ใช้เทคโนโลยี CMOS with SOS Speed
2. ใช้เทคนิคการแปลงข้อมูลแบบขนาน
3. อัตราการแปลงข้อมูล 15 Msps ที่ 5 V.
4. ให้สัญญาณเอาต์พุตขนาด 8 Bits
5. ใช้แหล่งจ่ายไฟชุดเดียว 4 ถึง 6.5 V.
6. แยกระบบกราวด์ของสัญญาณแอนาลอกกับสัญญาณดิจิทัลออกจากกันเด็ดขาด
7. กำลังงานสูญเสีย 200 mW.
8. แรงดันอินพุตอยู่ในช่วง 0 ถึง 6.4 V.
9. สัญญาณนาฬิกา 15 MHz

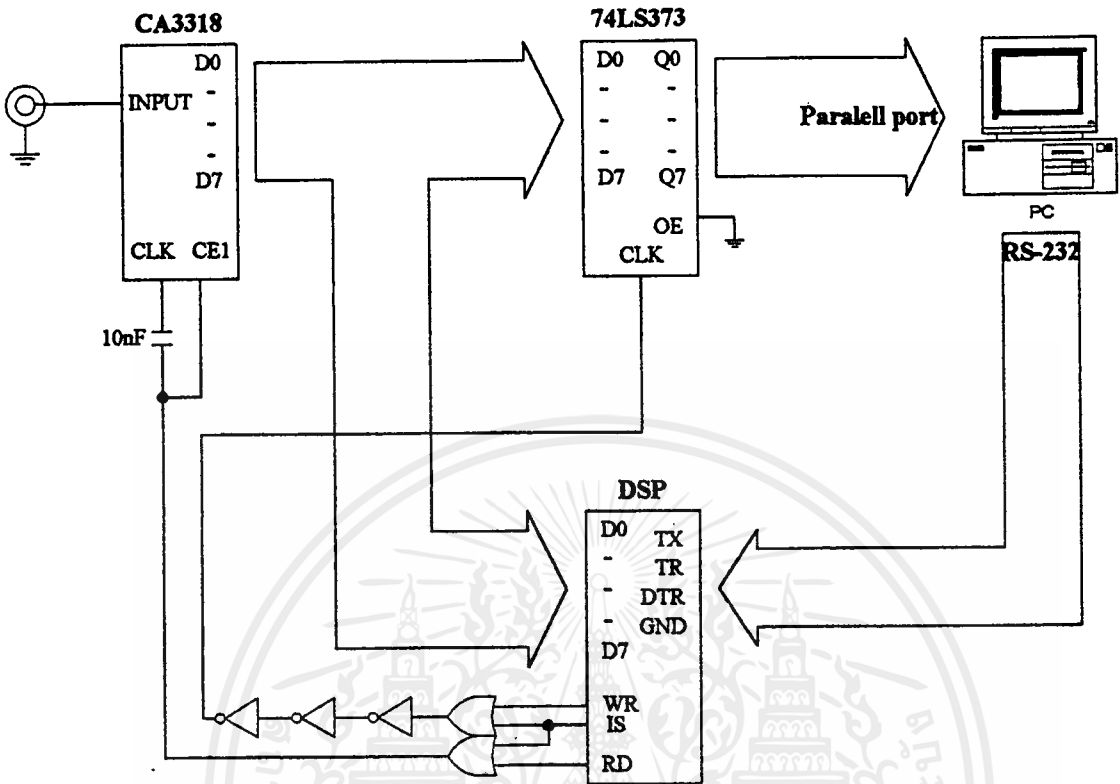


เอกสารนี้เป็นเอกสารที่สงวน **รูปที่ 3.5** ผังการทำงานภายในของไอซี CA3318C. หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล

วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัลนี้ จะใช้อัตราการสุ่ม (Sampling - Rate) ที่ 4 MHz. ซึ่งกำหนดได้จาก clock ที่ป้อนให้กับขา clk ของ IC โดยสัญญาณ clock ที่ป้อนให้ นี้ นามาจากการ OR กัน ของสัญญาณ การติดต่อกับอุปกรณ์อินพุต/เอาต์พุต (IS) และสัญญาณการอ่าน (RD) ของ DSP ซึ่งจะแอกทีฟ พร้อมกันเมื่อ DSP ต้องการอ่านข้อมูลจากอุปกรณ์อินพุต ในอัตรา 0.25 ไมโครวินาทีต่อครั้ง ทำให้ได้ความถี่ของการอ่านข้อมูล หรือสัญญาณที่นำไปเห็นสัญญาณ clock ของไอซี A/D นี้ เท่ากับ 4 MHz. เป็นผลให้สามารถสุ่มตัวอย่างสัญญาณอินพุตที่ความถี่สูงสุด 2 MHz. เอาต์พุตที่ได้จะอยู่ในรูปของสัญญาณดิจิทัลขนาด 8 บิต ส่งให้กับ DSP เพื่อนำไปประมวลผลต่อไป



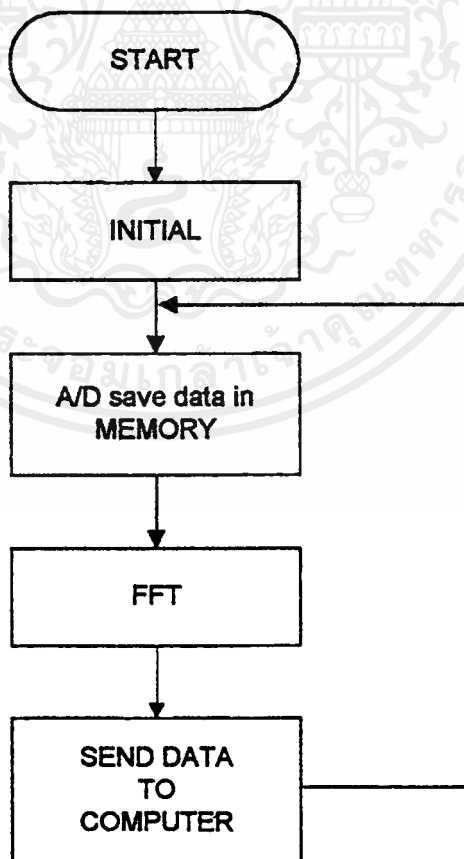
รูปที่ 3.7 วงจรภายในของเครื่องวิเคราะห์แถบความถี่

3.2 การออกแบบทางด้านซอฟต์แวร์

3.2.1 การออกแบบ และการแปลงซอฟต์แวร์วิเคราะห์แถบความถี่

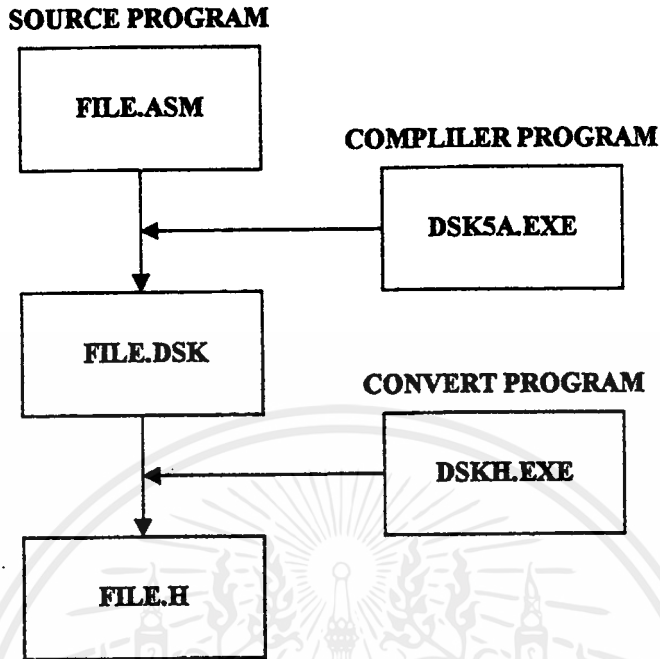
การออกแบบซอฟต์แวร์ การวิเคราะห์แถบความถี่นี้ จะใช้โปรแกรมแอสเซมบลีและตีบ์เกอร์ของชิพ DSP ในการทำการเขียนโปรแกรม โดยได้ออกแบบลำดับขั้นตอนการทำงานของโปรแกรมดังแผนผัง แสดงดังรูป 3.8 โดยเมื่อเริ่มการทำงาน จะเป็นการกำหนดสถานะเริ่มต้น ของระบบเพื่อให้ทุกส่วนพร้อมที่จะทำงาน เครื่องจะทำการสุ่มสัญญาณข้อมูล โดย A/D เข้ามาเก็บไว้ในหน่วยความจำภายในจนครบ 1 Kbyte จากนั้นจะนำข้อมูลที่ได้ออกมาจำนวนโดยใช้ FFT (Fast Fourier Transform) ได้ผลลัพธ์ออกมาเป็นขนาดของสัญญาณความถี่ต่างๆ แล้วจึงส่งค่าผลลัพธ์ที่ได้นี้ ให้กับคอมพิวเตอร์ทางพอร์ตขนาน (พอร์ต พรีนเตอร์) เพื่อทำการแสดงผลเป็นรูปสัญญาณต่างๆ ซึ่งขั้นตอนทั้งหมดที่กล่าวมา เป็นหน้าที่ของ DSP ที่จะทำการประมวลผล และส่งงานส่วนต่างๆ ให้ทำงานสัมพันธ์กัน และมีความเร็วที่สูงมากๆ โดยขึ้นอยู่กับวิธีการเขียนโปรแกรม การเลือกใช้คำสั่ง และการศึกษาการทำงานของ DSP ไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบซอฟต์แวร์เริ่มต้นจากการเขียนโปรแกรม File.ASM ซึ่งเป็นโปรแกรมวิเคราะห์แถบความถี่ โดยการใช้โปรแกรมภาษาแอสเซมบลีของ DSK บอร์ด แล้วทำการคอมไพล์โดยแอสเซมเบลอร์ของ DSK ด้วยเพิ่มข้อมูล DSK5A ได้เพิ่มข้อมูลใหม่คือ File.DSK และ File.LST โดยใช้ดีบักเกอร์ของ DSK ในการเริ่มต้นทำโปรแกรม File.DSK ทำการทดลองและแก้ไขโปรแกรมจนได้โปรแกรมที่สมบูรณ์ แล้วจึงทำการแปลงเพิ่มข้อมูลที่ได้ให้เป็นเพิ่มข้อมูล File.H โดยใช้เพิ่มข้อมูล DSKHEXE ในการแปลงข้อมูลเพื่อใช้ในการแยกส่วนของโปรแกรมและส่วนของข้อมูล เพื่อส่งข้อมูลผ่านจากโปรแกรมภาษาซี ไปยังบอร์ด DSK เขียนโปรแกรมแสดงผลทางหน้าจอคอมพิวเตอร์ด้วยโปรแกรมภาษาซี และฟังก์ชันการใช้งานต่างๆ รวมทั้งการรับส่งข้อมูล ผ่านทางพอร์ตขนาน และพอร์ตอนุกรมโดย พอร์ตอนุกรมนี้จะใช้ในการโหลดโปรแกรมให้กับบอร์ดDSK



รูปที่ 3.8 ผังการทำงานของโปรแกรมภาษาแอสเซมบลี

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ขั้นตอนแปลงโปรแกรมแอสเซมบลี DSK

3.2.2 การออกแบบซอฟต์แวร์การแสดงผล

จากแผนผังการทำงาน ในรูปที่ 3.10 เมื่อเริ่มต้นการทำงานโปรแกรม จะทำการส่งโปรแกรมการทำงานให้บอร์ด DSK เพื่อให้ DSP ทำการ INITIAL ตัวเอง และทำการเคลียร์ข้อมูลในหน่วยความจำ และแสดงหน้าจอเป็นแถบความถี่ และแรงดัน พร้อมทั้งจะแสดงรูปภาพ จากนั้นรอรับทำการสั่งงานต่างๆ ทางคีย์บอร์ด โดยจะทำงานตามฟังก์ชันต่างๆ เช่น

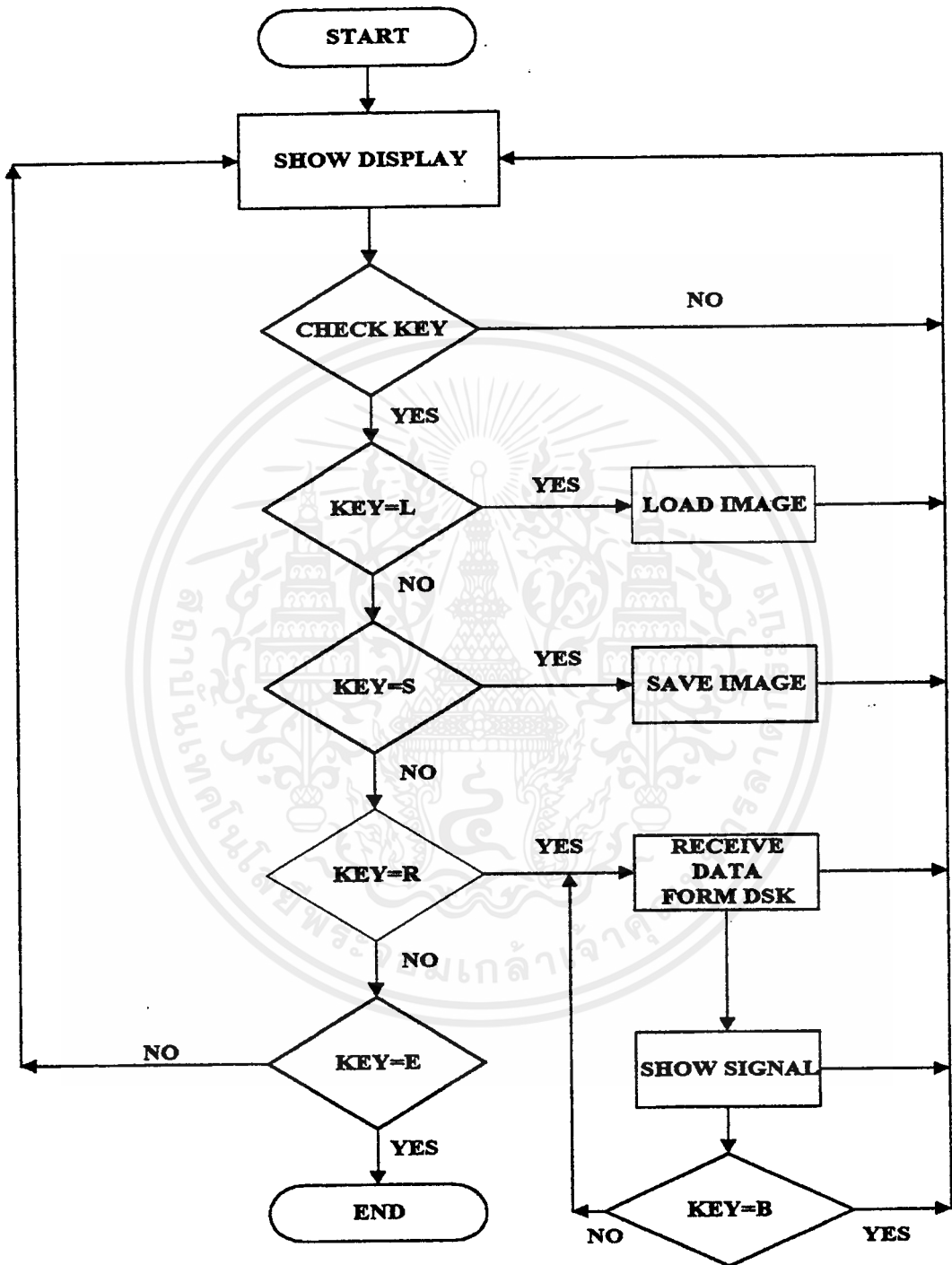
1. ฟังก์ชันการเรียกคืนภาพ (Load หรือคีย์ L) จะทำการนำภาพของสเปกตรัมที่เคยทำการบันทึกไว้มาแสดงผล เพื่อใช้ในการอ้างอิง หรือใช้ในการวิเคราะห์เปรียบเทียบกับสัญญาณอื่นๆ
2. ฟังก์ชันการบันทึกภาพ (Save คีย์ S) ใช้ในการ SAVE หน้าจอที่ผู้ใช้ต้องการเก็บไว้ โดยจะมีฟังก์ชันที่ใช้ในการโหลด สัญญาณภาพที่เก็บไว้ขึ้นมาวิเคราะห์โดยสะดวก
3. ฟังก์ชันการแสดงผล (Run หรือคีย์ R) จะทำงานในลักษณะต่อเนื่อง คือ รับข้อมูลผ่านเข้ามาทาง พอร์ตนาน แล้วทำการแปลงข้อมูลเป็นแท่งสเปกตรัม คำนวณขนาดของแอมพลิจูด และคำนวณหาค่าความถี่ แล้วส่งไปแสดงผลทางหน้าจอคอมพิวเตอร์ จากนั้นก็จะทำการตรวจ

เช็คคีย์ Break หรือฟังก์ชันการหยุดภาพ หากไม่มีการเรียกใช้ฟังก์ชันนี้ ก็จะกลับไปรับข้อมูลทางพอร์ตขั้วนาบ อีกครั้งเพื่อนำมาแสดงค่าใหม่อีกครั้งหนึ่ง

4. ฟังก์ชันการหยุดภาพ (Break หรือคีย์ B) ในฟังก์ชันนี้จะเป็นการหยุดหน้าจอไว้ เพื่อความสะดวกในการวิเคราะห์สัญญาณ เพื่อใช้ในการตรวจสอบความถี่ และขนาดของแรงดันที่ตำแหน่งต่างๆ

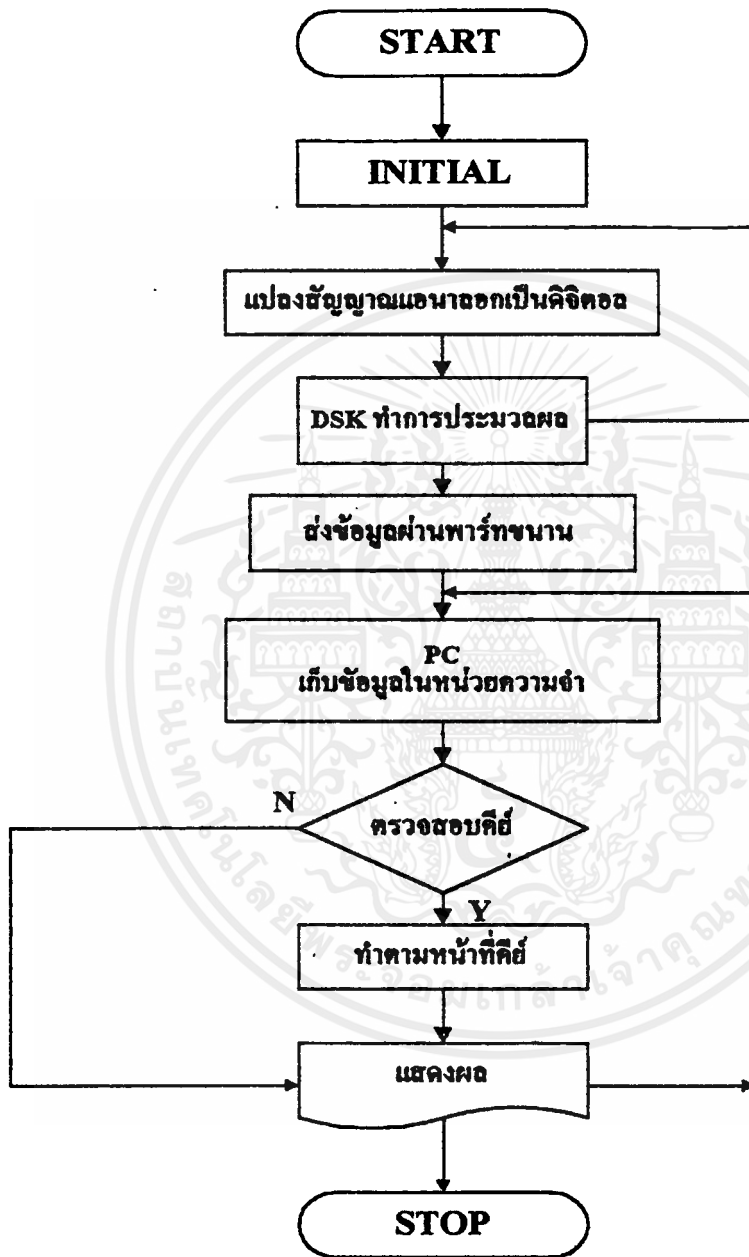
5. ฟังก์ชันการออกจากโปรแกรม (Exit หรือคีย์ E) เป็นฟังก์ชันที่ใช้ในการออกจากโปรแกรม (จบการทำงาน)

จากรูปที่ 3.11 เป็นผังการทำงานทั้งหมดของซอฟต์แวร์ที่ใช้ในการควบคุมการทำงานของเครื่องวิเคราะห์แถบความถี่ โดยเป็นลำดับขั้นตอนทั้งหมดของโปรแกรมภาษาแอสเซมบลี และโปรแกรมภาษาซี ที่เขียนขึ้น และใช้งานร่วมกัน ดังที่ได้กล่าวไว้ในขั้นตอน ส่วนซอฟต์แวร์ควบคุมการทำงานได้แสดงไว้ในภาคผนวก



รูปที่ 3.10 ผังการทำงานของโปรแกรมภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของงานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ผังการทำงานทั้งหมดของโปรแกรมของเครื่องวิเคราะห์แถบความถี่

บทที่ 4

การทดลอง และผลการทดลอง

การทดลองเครื่องวิเคราะห์แถบความถี่ โดยการทดสอบการวัดสัญญาณในรูปแบบของสัญญาณขนาดต่างๆ จากแหล่งกำเนิดสัญญาณความถี่ (ฟังก์ชันเจนเนอเรเตอร์)

4.1 การทดสอบการทำงานของเครื่องวิเคราะห์แถบความถี่โดยใช้ TMS320C50

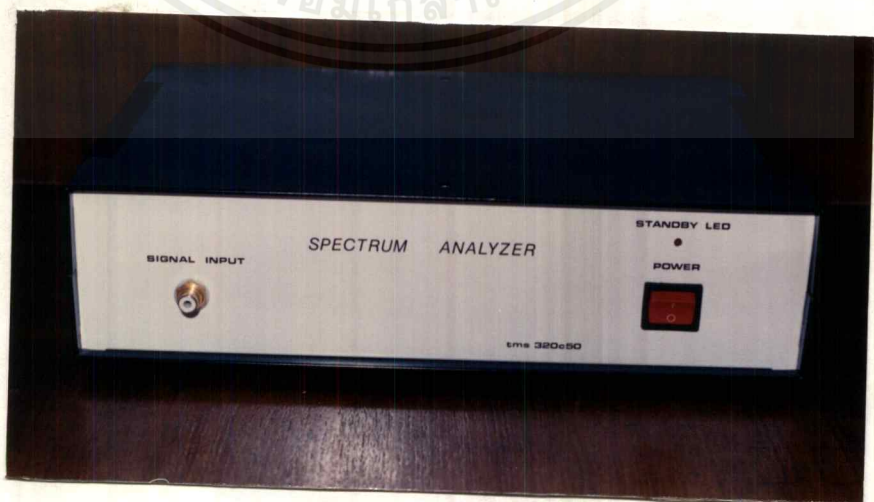
4.1.1 ลำดับขั้นตอนการทดลอง

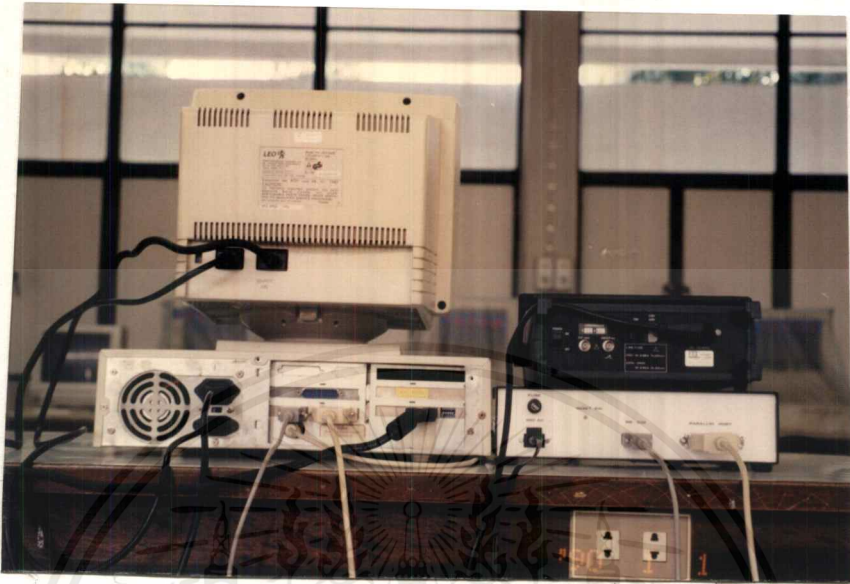
1. ทำการเชื่อมต่อเครื่องวิเคราะห์แถบความถี่ เข้ากับพอร์ตขนาน และพอร์ตอนุกรมของคอมพิวเตอร์ และทำการเรียกโปรแกรม SPECT

2. ป้อนสัญญาณอินพุตรูปคลื่นไซน์ (Sine Wave) ขนาด 4 Vp-p ที่ความถี่ 1 MHz. เข้ากับ Input ของเครื่องวิเคราะห์แถบความถี่

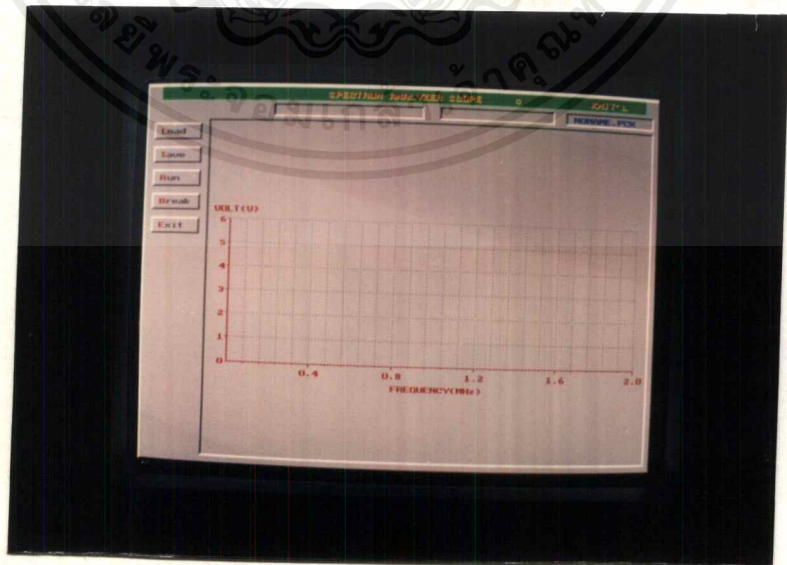
3. เปรียบเทียบขนาดของแรงดัน และความถี่ ที่อ่านจาก หน้าจอคอมพิวเตอร์ กับ ออสซิลโลสโคป สัญญาณอินพุตที่ป้อนจากฟังก์ชันเจนเนอเรเตอร์

4. สังเกต รูปคลื่นสเปกตรัมของสัญญาณอินพุตที่ป้อนแสดงดังรูปที่ 4.4



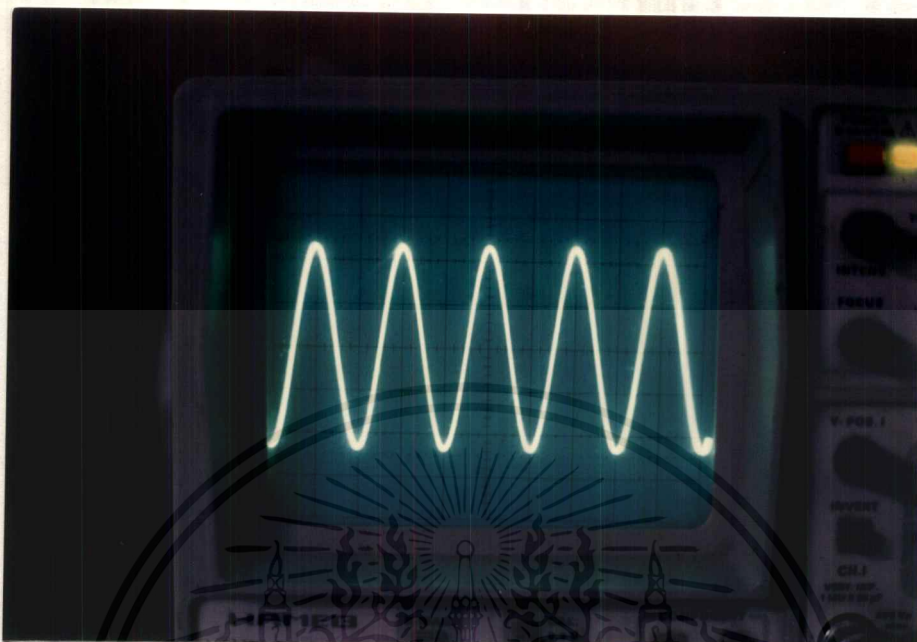


รูปที่ 4.2 การต่อเครื่องวิเคราะห์แถบความถี่ใช้งานร่วมกับคอมพิวเตอร์

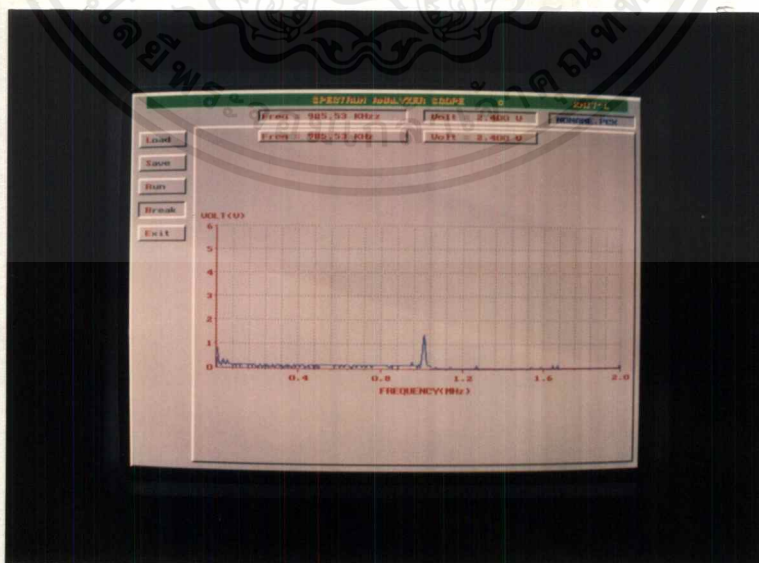


รูปที่ 4.3 หน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนที่นำเอกสารนี้ไปใช้ประโยชน์ในการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สัญญาณอินพุตรูปคลื่นไซน์ ขนาด 4 Vp-p ที่ความถี่ 1 MHz.



รูปที่ 4.5 สเปกตรัมของสัญญาณรูปคลื่นไซน์ ขนาด 4 Vp-p ที่ความถี่ 1 MHz

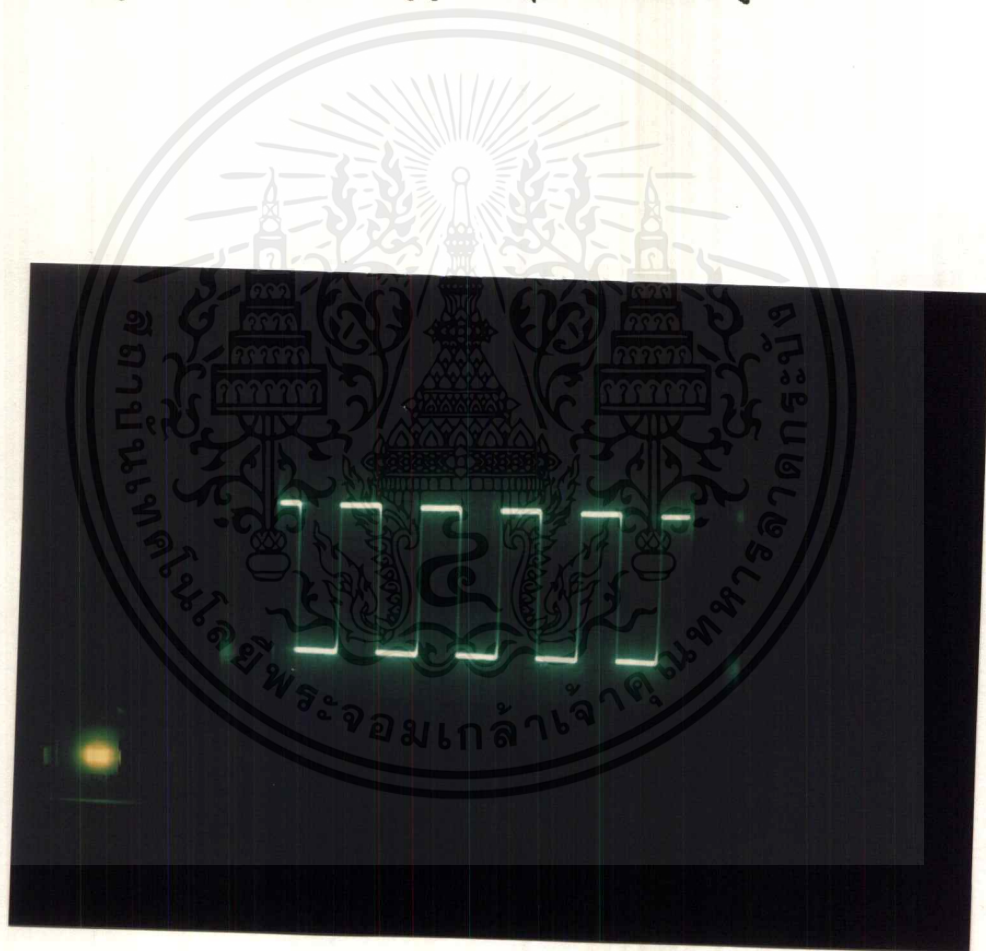
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทดลองเปลี่ยนขนาดของแรงดัน และความถี่ของสัญญาณอินพุต ที่ป้อนทำการ ทดลองซ้ำตามข้อ 2 และ 3 อีกครั้ง สังเกตการเปลี่ยนแปลงของรูปคลื่นสเปกตรัม

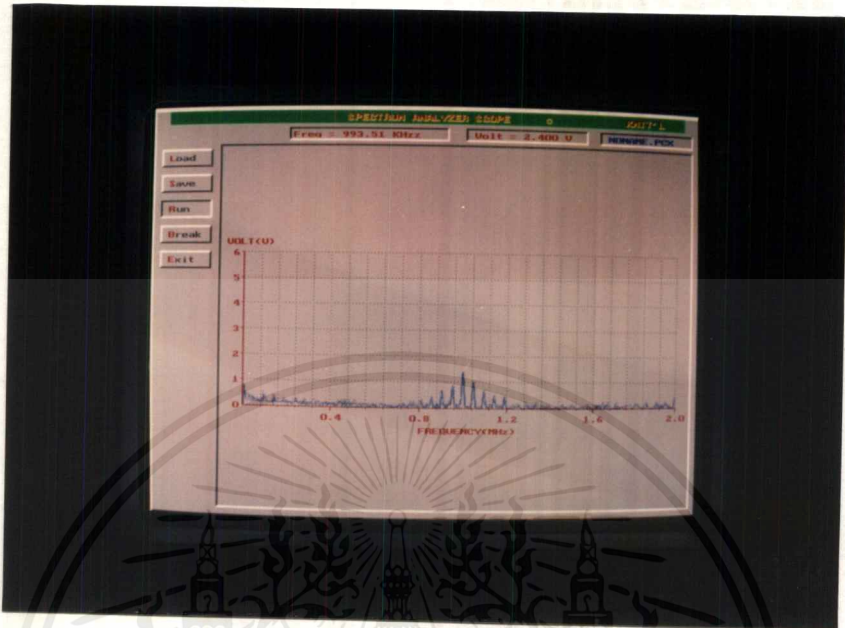
5. เปลี่ยนรูปคลื่นสัญญาณอินพุต จากรูปคลื่นไซน์ เป็นรูปคลื่นสี่เหลี่ยม (Square Wave) ขนาด 4 Vp-p ที่ความถี่ 1 MHz.

6. เปรียบเทียบขนาดของแรงดัน และความถี่ ที่อ่านจากหน้าจอกอมพิวเตอรื กับ ออสซิลโลสโคป สัญญาณอินพุตที่ป้อนจากฟังก์ชันเจนเนอเรเตอร์

7. สังเกตรูปคลื่นสเปกตรัมของสัญญาณอินพุตที่ป้อนแสดงดังรูปที่ 4.6

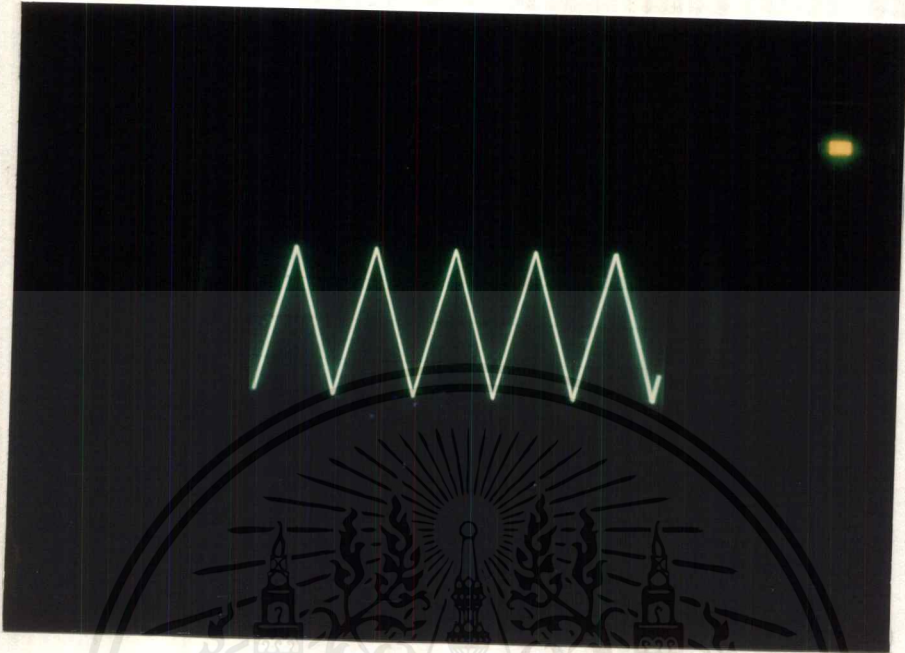


รูปที่ 4.6 สัญญาณอินพุตรูปคลื่นสี่เหลี่ยม ขนาด 4 Vp-p ที่ความถี่ 1 MHz.

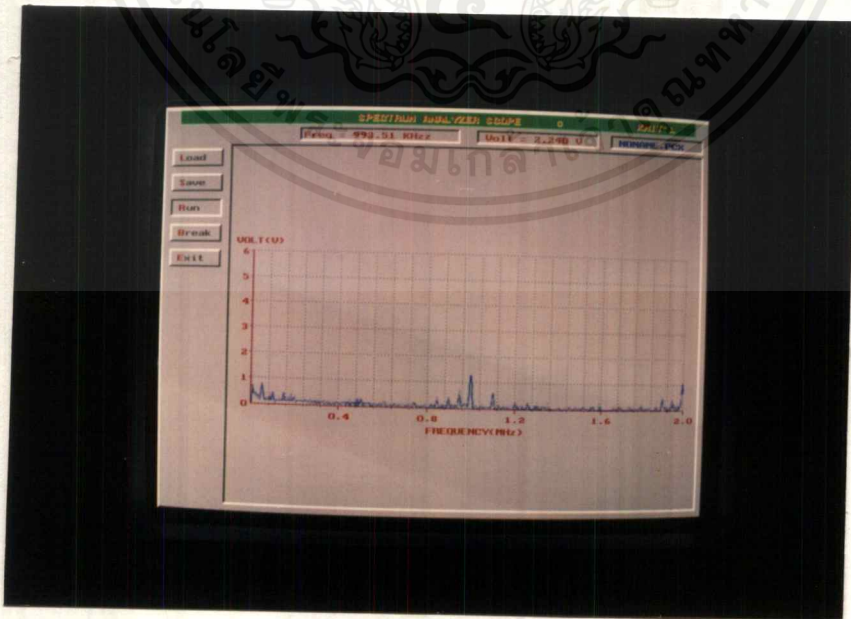


รูปที่ 4.7 สเปกตรัมของสัญญาณรูปคลื่นสี่เหลี่ยม ขนาด 4 Vp-p ที่ความถี่ 1 MHz.

8. ทดลองเปลี่ยนขนาดของแรงดัน และความถี่ของสัญญาณอินพุต ที่ป้อน ทำการทดลองซ้ำ ตามข้อ 6 และ 7 อีกครั้ง สังเกตการเปลี่ยนแปลงของรูปคลื่นสเปกตรัม
9. เปลี่ยนรูปคลื่นสัญญาณอินพุตจากรูปคลื่นสี่เหลี่ยมเป็นรูปคลื่นฟันเลื่อย(Saw Wave) ขนาด 4 Vp-p ที่ความถี่ 1 MHz.
10. เปรียบเทียบขนาดของแรงดันและความถี่ที่อ่านจากหน้าจอคอมพิวเตอร์ กับออสซิลโลสโคป สัญญาณอินพุตที่ป้อนจากฟังก์ชันเจนเนอเรเตอร์
11. สังเกตรูปคลื่นสเปกตรัมของสัญญาณอินพุตที่ป้อน แสดงดังรูปที่ 4.8



รูปที่ 4.8 สัญญาณอินพุตรูปคลื่นฟันเลื่อย ขนาด 4 Vp-p ที่ความถี่ 1 MHz.



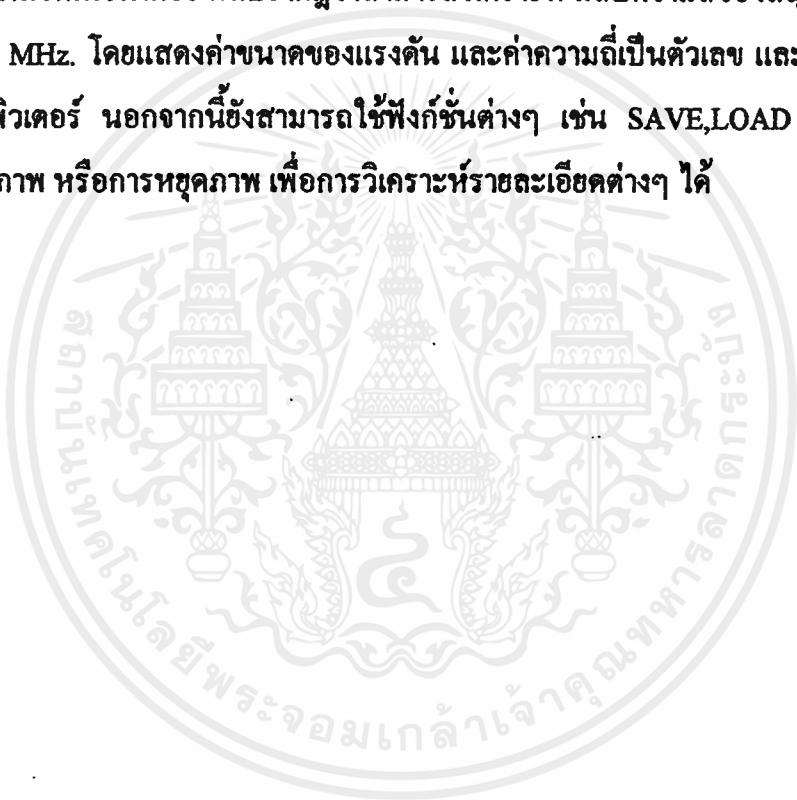
รูปที่ 4.9 สเปกตรัมของสัญญาณรูปคลื่นฟันเลื่อย ขนาด 4 Vp-p ที่ความถี่ 1 MHz.

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ซึ่งใช้โดยสถาบันด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ทดลองเปลี่ยนขนาดแรงดัน และความถี่ ของสัญญาณอินพุตที่ป้อนทำการทดลองซ้ำตามข้อ 10 และ 11 อีกครั้ง สังเกตการเปลี่ยนแปลงของรูปคลื่นสเปกตรัม

4.1.2 ผลการทดลอง

จากการทดลอง การทำงานของเครื่องวิเคราะห์แถบความถี่ โดยใช้ TMS320C50 โดยการป้อนสัญญาณรูปคลื่นไซน์ รูปคลื่นสี่เหลี่ยม และรูปคลื่นฟันเลื่อยที่ขนาด และความถี่ต่างๆ กัน จากฟังก์ชันเจนเนอเรเตอร์ ผลปรากฏว่าสามารถวิเคราะห์ แถบความถี่ของสัญญาณในย่าน 1.33 kHz - 2 MHz. โดยแสดงค่าขนาดของแรงดัน และค่าความถี่เป็นตัวเลข และตัวอักษรบนหน้าจอคอมพิวเตอร์ นอกจากนี้ยังสามารถใช้ฟังก์ชันต่างๆ เช่น SAVE, LOAD ภาพหน้าจอคอมพิวเตอร์ภาพ หรือการหยุดภาพ เพื่อการวิเคราะห์รายละเอียดต่างๆ ได้



บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

5.1 บทสรุป

เครื่องวิเคราะห์แถบความถี่นี้ ออกแบบ และสร้างขึ้นบนทฤษฎี และหลักการของการประมวลผลสัญญาณเชิงเลขซึ่งได้นำเอาไมโครโปรเซสเซอร์ประมวลผลสัญญาณเชิงเลขมาใช้ในการประมวลผลโดยขอบเขตที่วางไว้ในขั้นต้นคือสามารถวัดความถี่ ในย่าน 0-10 MHz. และแสดงผลเป็นตัวเลขบนจอขนาดได้

จากการที่ได้ทำการศึกษา และออกแบบสร้างเครื่องวิเคราะห์แถบความถี่ เป็นผลให้ได้เครื่องวิเคราะห์แถบความถี่ที่สามารถทำการวัดแถบความถี่ได้ในย่าน 1.33 kHz. ถึง 2 MHz. โดยแสดงผลที่หน้าจอกอมพิวเตอร์ ผ่านทางพอร์ตขนาน เครื่องวิเคราะห์แถบความถี่นี้สามารถแสดงผลความถี่ และขนาดของสัญญาณ ออกมาในรูปของตัวเลขบนหน้าจอกอมพิวเตอร์ ได้ อย่างที่ตรง อีกทั้งยังมีฟังก์ชันการใช้งานต่างๆ เช่น การหยุดภาพบนหน้าจอ เพื่อการอ่านค่า การบันทึกภาพ และการเรียกคืน (Load) ทำให้มีความสะดวกในการวิเคราะห์สัญญาณ และการใช้งานเป็นอย่างมาก

ข้อดีของเครื่องวิเคราะห์แถบความถี่ คือ สามารถนำคอมพิวเตอร์มาประยุกต์ใช้งาน และมีราคาถูกกว่าเครื่องที่มีขายตามท้องตลาด แต่ประสิทธิภาพสามารถนำมาทดแทนกันได้

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ในส่วนของฮาร์ดแวร์

ปัญหา การทำงานของวงจรไม่เป็นไปตามความต้องการ

แนวทางการแก้ไข ทำการออกแบบ และจำลองการทำงานให้ได้ผลที่พอใจก่อนการลงมือสร้างจริง

ปัญหา วงจรมีขนาดค่อนข้างใหญ่ ทำให้ต้องใช้ไฟเลี้ยงหลายชุด

แนวทางการแก้ไข ออกแบบวงจรให้มีขนาดเล็กที่สุด และเลือกใช้ไอซี ที่กินพลังงานต่ำ แต่มีความเร็วในการทำงานที่สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหา จังหวะการทำงานของอุปกรณ์ แต่ละตัวไม่สัมพันธ์กัน เนื่องจากโครงสร้างพื้นฐานของอุปกรณ์แต่ละชนิด เช่นในการใช้งานกับความถี่ต่างๆ อัตราการหน่วงเวลาในการทำงานในแต่ละจังหวะ จะมีผลอย่างมาก

แนวทางการแก้ไข ศึกษาถึงโคอะแกรม การทำงานของอุปกรณ์นั้นๆ อย่างลึกซึ้ง และเลือกใช้อุปกรณ์ที่เหมาะสม

ปัญหา สัญญาณรบกวน ที่เกิดจากการเชื่อมต่อจุดต่างๆ

แนวทางการแก้ไข ออกแบบลวดวงจรพิมพ์ ให้เรียบร้อย รวมทั้งการใช้สายเชื่อมต่อ ในจุดต่างๆ ควรให้สั้น และน้อยที่สุด

ปัญหา อุปกรณ์ที่ใช้บางตัวมีราคาแพง และหายาก

แนวทางการแก้ไข เลือกใช้อุปกรณ์ที่เหมาะสม และสามารถทำได้ตามท้องตลาด หรืออาจติดต่อสอบถามทางจำหน่าย เพื่อทำการสั่งซื้อล่วงหน้า

5.2.2 ในส่วนของซอฟต์แวร์

ปัญหา การศึกษาคำสั่งและการเขียนโปรแกรมภาษาแอสเซมบลี ของชิพ DSP ก่อนข้างใช้เวลานาน เนื่องจากไม่เคยศึกษามาก่อน

แนวทางการแก้ไข ศึกษาและขอคำแนะนำจากอาจารย์ และรุ่นพี่ นอกจากนี้ต้องอาศัยการฝึกเขียนบ๊อช ๆ รวมทั้งศึกษาจากคู่มือ และตัวอย่างโปรแกรมให้มากขึ้น

ปัญหา การเขียนโปรแกรมแสดงผลทางหน้าจอคอมพิวเตอร์ ไม่เป็นเวลาจริง

แนวทางการแก้ปัญหา เลือกใช้คำสั่ง และขั้นตอนการทำงานของโปรแกรมที่ใช้เวลาประมวลผลน้อยที่สุด พร้อมทั้งเขียนโปรแกรมให้สั้นที่สุด

ปัญหา การเขียนโปรแกรมฟังก์ชันการใช้งานมีความยุ่งยากซับซ้อน

แนวทางการแก้ปัญหา ศึกษาเทคนิคการเขียนโปรแกรมใหม่ๆ หรือลดทอนฟังก์ชันที่มีปัญหาลงเล็กน้อย

5.3 ประโยชน์ที่ได้รับ

1. ทำให้ได้รับความรู้ ความเข้าใจ ระบบการทำงานของเครื่องวิเคราะห์แถบความถี่
2. ทำให้เกิดทักษะทางด้านการทำงาน ทั้งทางด้านทฤษฎี และปฏิบัติ
3. ทำให้รู้จักการแก้ปัญหา และเกิดความคิดริเริ่มสร้างสรรค์ในการทำงาน

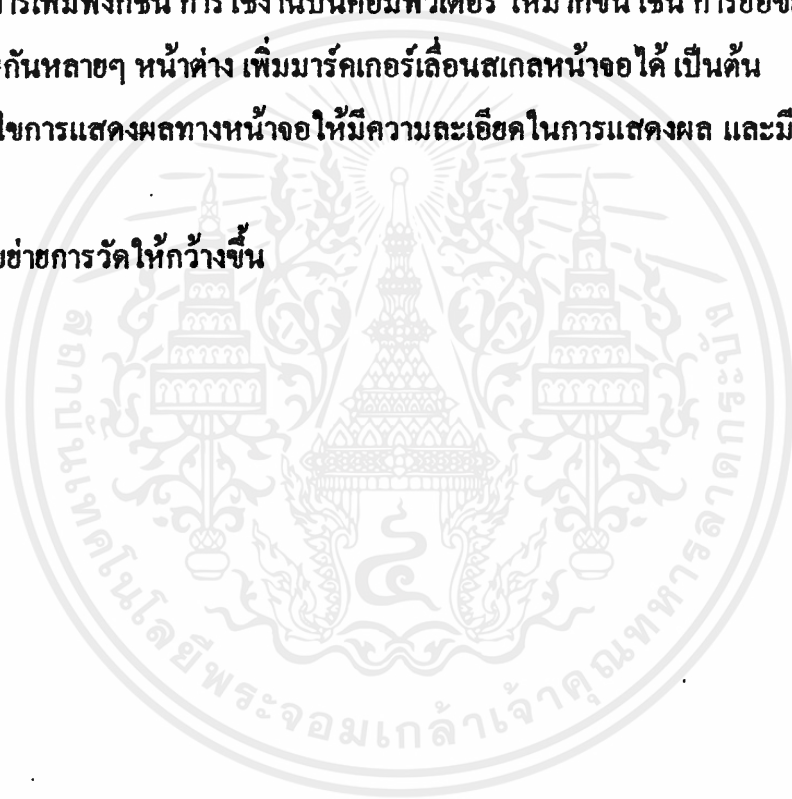
4. ทำให้ได้รับประสบการณ์ในการทำงานร่วมกันเป็นกลุ่ม

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เพื่อให้เป็นแนวทางในการศึกษาพัฒนาในรุ่นต่อไป
6. เพื่อนำโครงการนี้ไปใช้สอบในรายวิชา โครงการสร้างอุปกรณ์เพื่อการสอนได้

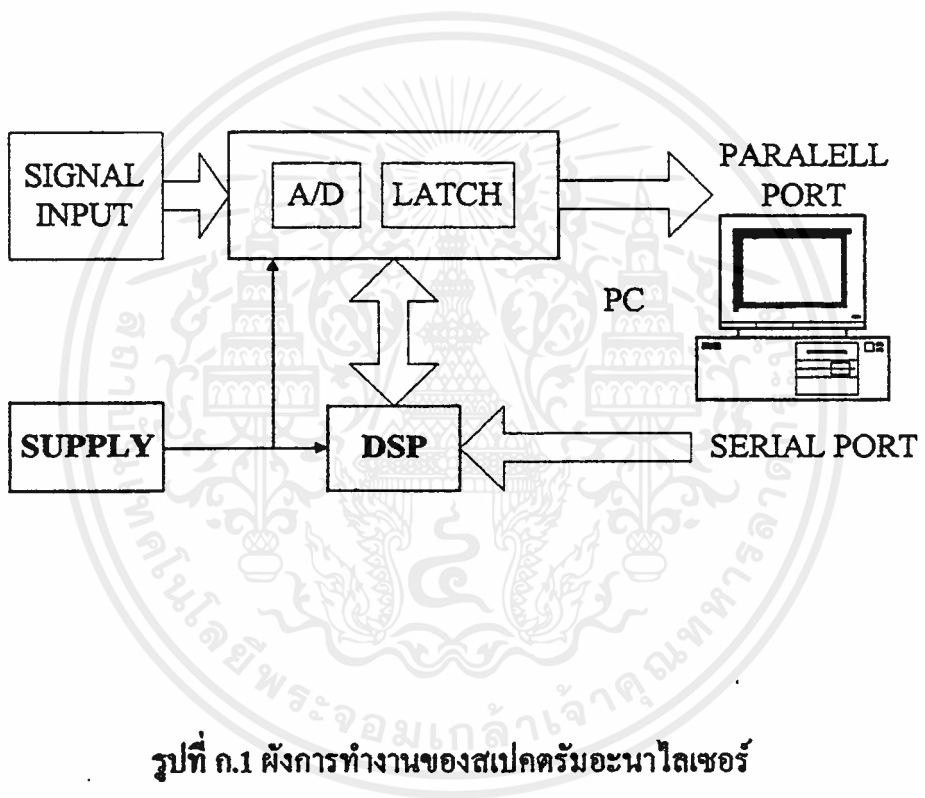
5.4 แนวทางการพัฒนา

1. ควรปรับปรุงแก้ไข ในส่วนของวงจร A/D ให้มีอัตราการสุ่มสัญญาณที่สูงขึ้น โดยใช้ ไอซี A/D ที่มีความเร็วสูง เพื่อเป็นการขยายการวัดให้กว้างขึ้น
2. ทำการเพิ่มฟังก์ชัน การใช้งานบนคอมพิวเตอร์ ให้มากขึ้น เช่น การช้อขยายภาพ หรือ การใช้งานร่วมกันหลายๆ หน้าต่าง เพิ่มมาร์คเกอร์เลื่อนสเกลหน้าจอได้ เป็นต้น
3. แก้ไขการแสดงผลทางหน้าจอให้มีความละเอียดในการแสดงผล และมีความสวยงามยิ่งขึ้น
4. ขยายขยายการวัดให้กว้างขึ้น

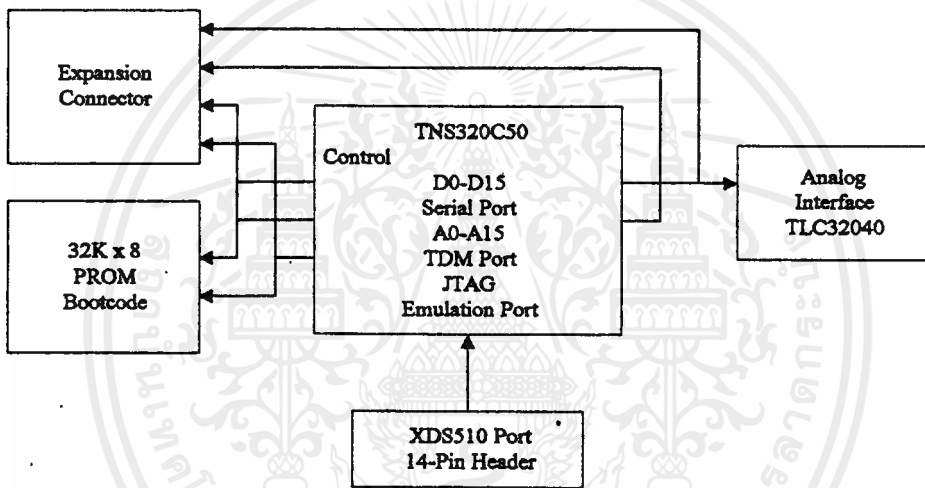




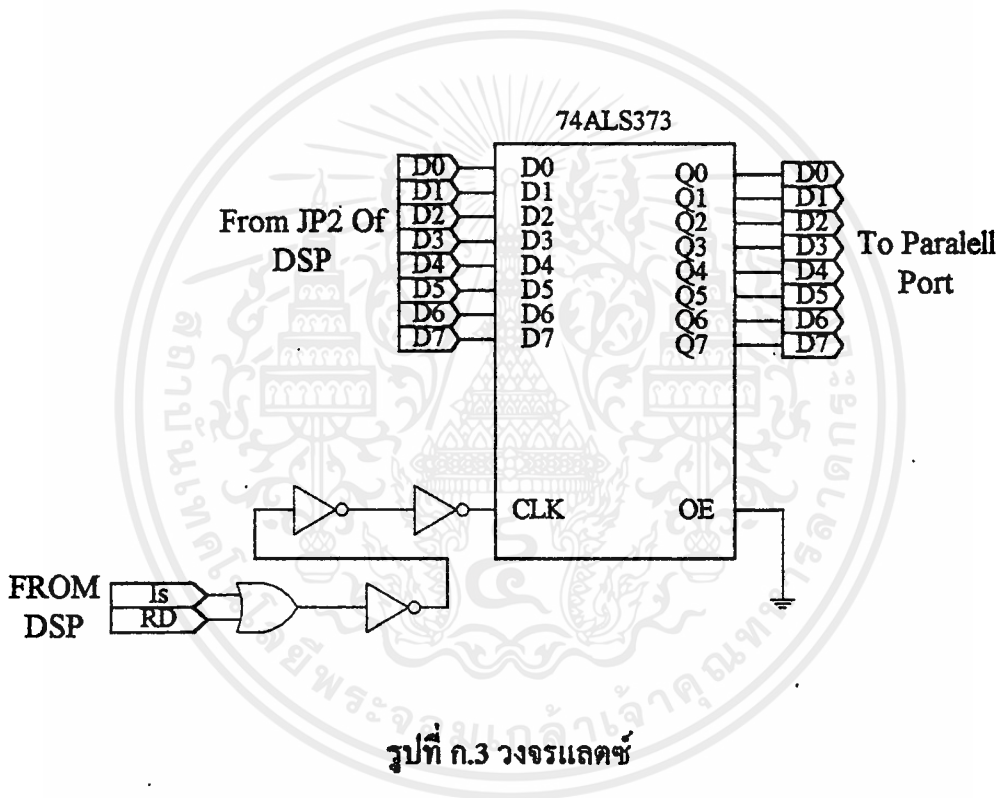
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

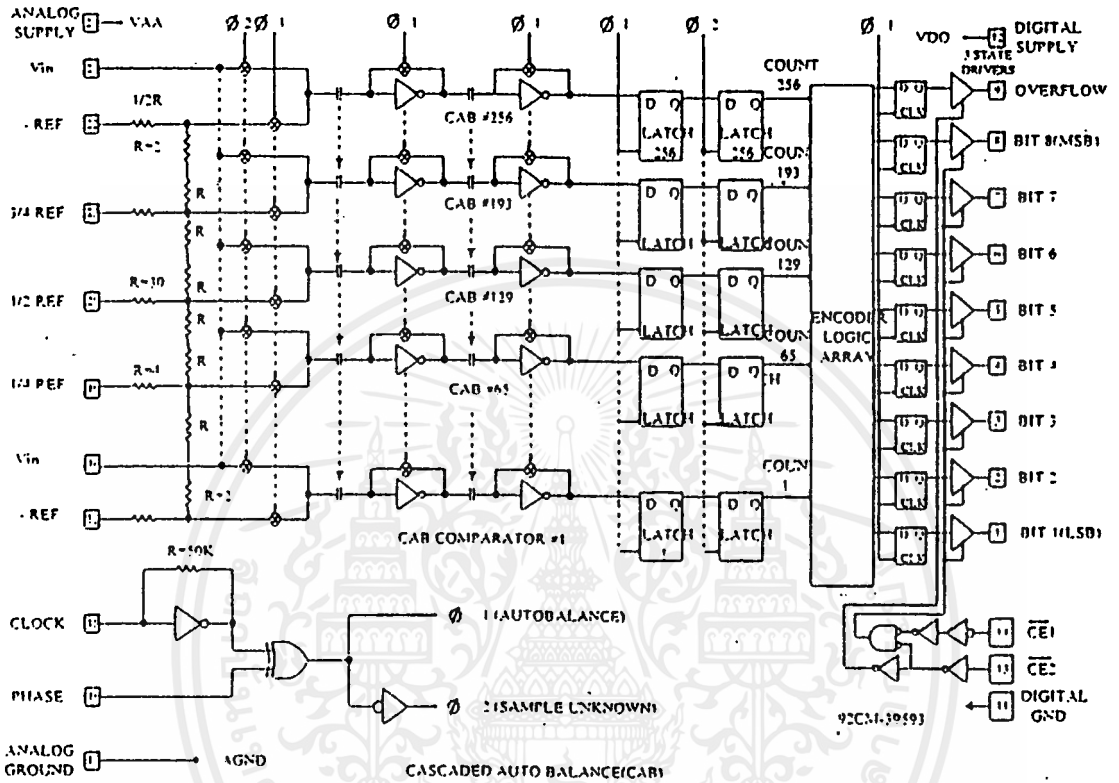


รูปที่ ก.1 ฟังก์ชันการทำงานของสเปคตรัมอะนาไลเซอร์

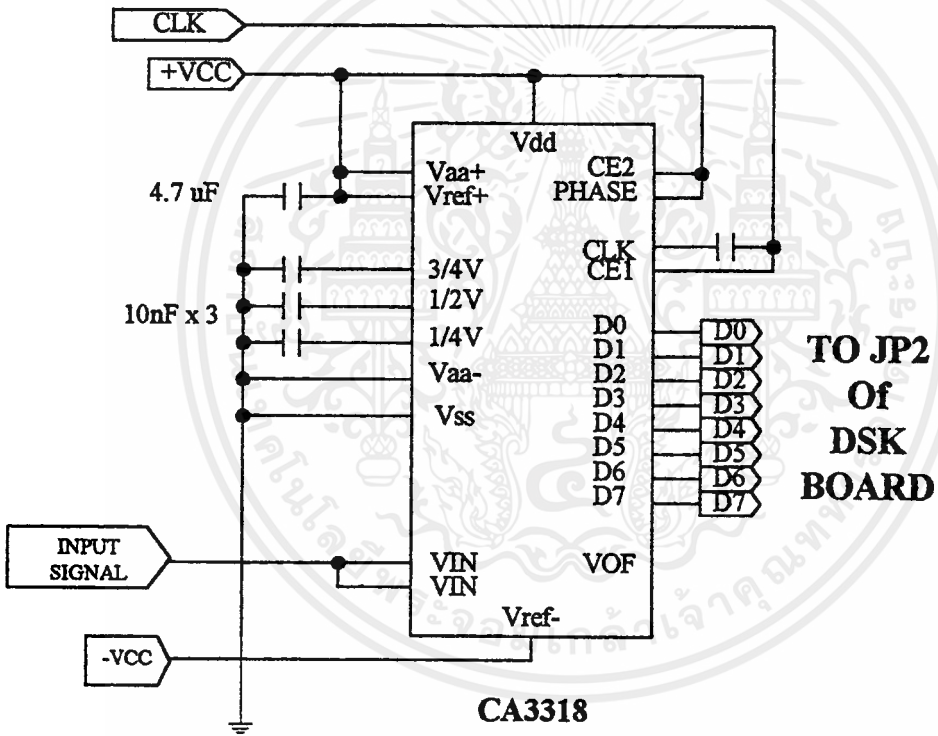


รูปที่ ก.2 บล็อกไดอะแกรมของบอร์ด DSK

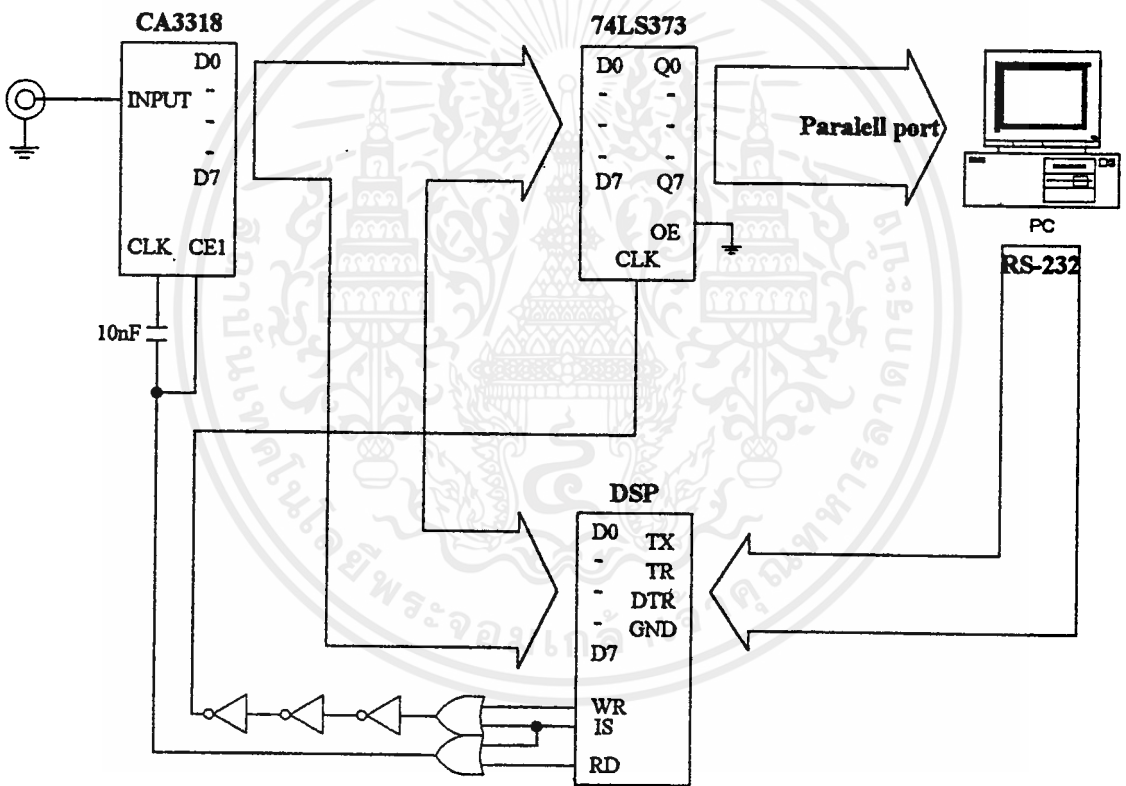




รูปที่ ก.4 ผังการทำงานภายในของไอซี CA3318



รูปที่ ก.5 วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล



รูปที่ ก.6 วงจรภายในของเครื่องวิเคราะห์แถบความถี่

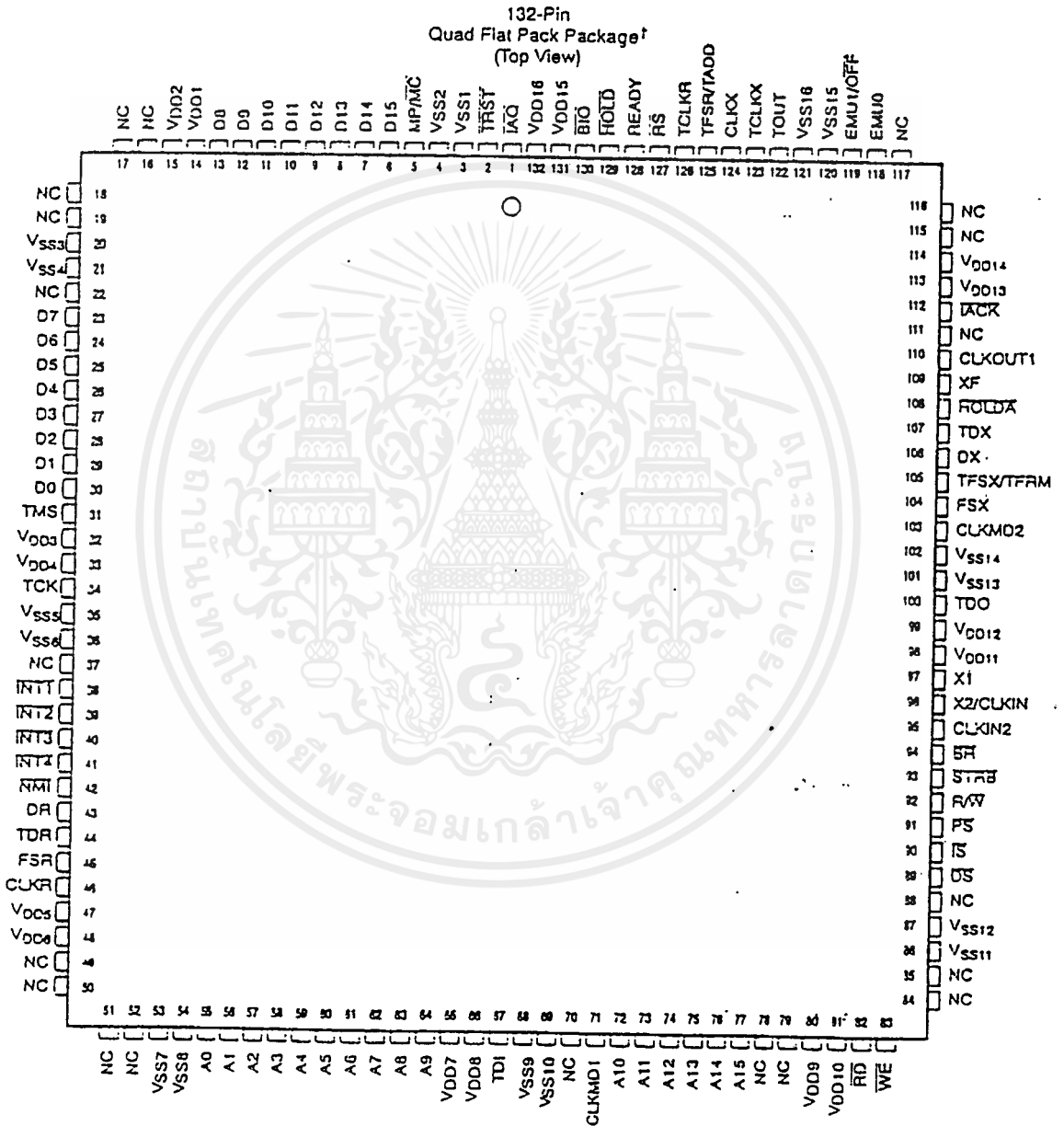


ภาคผนวก ข
รายละเอียดของอุปกรณ์ที่ใช้ในโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.1 Pinout and Signal Descriptions

Figure A-1. TMS320C5x Pinout



† See Pin Assignments, Table A-1 (page A-3) for location and description of all pins. The 'C50, 'C51, and 'C53 are packaged in 132-pin plastic QFP in production.
 Note: NC = No connect. (These pins are reserved.)

Table A-1. TMS320C5x Pin Assignments

Pin	Name	Type	Description
1	IAQ	O/Z	Instruction Acquisition
2	TRST	I	JTAG Test Reset
3	V _{SS}	Supply	Ground
4	V _{SS}	Supply	Ground
5	MP/MC	I	Microprocessor/Microcomputer
6	D15 (MSB)	I/O/Z	Parallel Data Port, High-Byte (8 pins)
7	D14	I/O/Z	
8	D13	I/O/Z	
9	D12	I/O/Z	
10	D11	I/O/Z	
11	D10	I/O/Z	
12	D9	I/O/Z	
13	D8	I/O/Z	
14	V _{DD}	Supply	+5 V
15	V _{DD}	Supply	+5 V
16	NC [†]		Reserved
17	NC [†]		Reserved
18	NC [†]		Reserved
19	NC [†]		Reserved
20	V _{SS}	Supply	Ground
21	V _{SS}	Supply	Ground
22	NC [†]		Reserved
23	D7	I/O/Z	Parallel Data Port, Low-Byte (8 pins)
24	D6	I/O/Z	
25	D5	I/O/Z	
26	D4	I/O/Z	
27	D3	I/O/Z	
28	D2	I/O/Z	
29	D1	I/O/Z	
30	D0 (LSB)	I/O/Z	
31	TMS	I	JTAG Test Mode
32	V _{DD}	Supply	+5 V
33	V _{DD}	Supply	+5 V
34	TCK	I	JTAG Test Clock

† NC = No connect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table A-1. TMS320C5x Pin Assignments (Continued)

Pin	Name	Type	Description
35	V _{SS}	Supply	Ground
36	V _{SS}	Supply	Ground
37	NC†		Reserved
38	INT1	I	Interrupt #1
39	INT2	I	Interrupt #2
40	INT3	I	Interrupt #3
41	INT4	I	Interrupt #4
42	NMI	I	Nonmaskable Interrupt
43	DR	I	Serial Port 1 Data Receive
44	TDR	I	Serial Port 2 Data Receive
45	FSR	I	Serial Port 1 Receiver Frame Sync
46	CLKR	I	Serial Port 1 Receiver Clock
47	V _{DD}	Supply	+5 V
48	V _{DD}	Supply	+5 V
49	NC†		Reserved
50	NC†		Reserved
51	NC†		Reserved
52	NC†		Reserved
53	V _{SS}	Supply	Ground
54	V _{SS}	Supply	Ground
55	A0 (LSB)	I/O/Z	Parallel Port Address Bus (10 pins)
56	A1	I/O/Z	
57	A2	I/O/Z	
58	A3	I/O/Z	
59	A4	I/O/Z	
60	A5	I/O/Z	
61	A6	I/O/Z	
62	A7	I/O/Z	
63	A8	I/O/Z	
64	A9	I/O/Z	
65	V _{DD}	Supply	+5 V
66	V _{DD}	Supply	+5 V
67	TDI	I	JTAG Scan Input

† NC = No connect

Table A-1. TMS320C5x Pin Assignments (Continued)

Pin	Name	Type	Description
68	V _{SS}	Supply	Ground
69	V _{SS}	Supply	Ground
70	NC [†]		Reserved
71	CLKMD1	I	Clock Mode Pin 1
72	A10	I/O/Z	Parallel Port Address Bus (6 pins)
73	A11	I/O/Z	
74	A12	I/O/Z	
75	A13	I/O/Z	
76	A14	I/O/Z	
77	A15	I/O/Z	
78	NC [†]		Reserved
79	NC [†]		Reserved
80	V _{DD}	Supply	+5 V
81	V _{DD}	Supply	+5 V
82	RD	O/Z	Read Enable
83	WE	O/Z	Write Enable
84	NC [†]		Reserved
85	NC [†]		Reserved
86	V _{SS}	Supply	Ground
87	V _{SS}	Supply	Ground
88	NC [†]		Reserved
89	DS	O/Z	Data Space Select
90	IS	O/Z	I/O Space Select
91	PS	O/Z	Program Space Select
92	R/W	I/O/Z	Read/Write
93	STRB	I/O/Z	External Parallel Access Active
94	BR	I/O/Z	Bus Request
95	CLKIN2	I	Divide-by-One Clock Input
96	X2/CLKIN	I	Divide-by-Two Clock Input
97	X1	O	Oscillator Output
98	V _{DD}	Supply	+5 V
99	V _{DD}	Supply	+5 V
100	TDO	O/Z	JTAG Scan Output

† NC = No connect

Pinout and Signal Descriptions

Table A-1. TMS320C5x Pins (Concluded)

Pin	Name	Type	Description
101	V _{SS}	Supply	Ground
102	V _{SS}	Supply	Ground
103	CLKMD2	I	Clock Mode Pin 2
104	FSX	I/O/Z	Serial Port 1 Transmitter Frame Sync
105	TFSX/TFRM	I/O/Z	Serial Port 2 Transmitter Frame Sync
106	DX	O/Z	Serial Port 1 Transmitter Output
107	TDX	O/Z	Serial Port 2 Transmitter Output
108	HOLDA	O/Z	Hold Acknowledge
109	XF	O/Z	External Flag
110	CLKOUT1	O/Z	Machine Clock Output
111	NC [†]		Reserved
112	IACK	O/Z	Interrupt Acknowledge
113	V _{DD}	Supply	+5 V
114	V _{DD}	Supply	+5 V
115	NC [†]		Reserved
116	NC [†]		Reserved
117	NC [†]		Reserved
118	EMU0	I/O/Z	Emulator Interrupt 0
119	EMU1/OF [†]	I/O/Z	Emulator Interrupt 1
120	V _{SS}	Supply	Ground
121	V _{SS}	Supply	Ground
122	TOUT	O/Z	Timer Output
123	TCLKX	I/O/Z	Serial Port 2 Transmitter Clock
124	CLKX	I/O/Z	Serial Port 1 Transmitter Clock
125	TFSR/TADD	I/O/Z	Serial Port 2 Receive Frame/Address
126	TCLKR	I	Serial Port 2 Receiver Clock
127	RS	I	Device Reset
128	READY	I	External Access Ready to Complete
129	HOLD	I	Request Access of Local Memory
130	BIT	I	Bit I/O Pin
131	V _{DD}	Supply	+5 V
132	V _{DD}	Supply	+5 V

[†] NC = No connect

Electrical Specifications

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CA3318C

CMOS Video Speed
8-Bit Flash A/D Converter

December 1993

Features

- CMOS Low Power with SOS Speed (150mW Typ.)
- Parallel Conversion Technique
- 15MHz Sampling Rate (67ns Conversion Time)
- 8-Bit Latched Tri-State Output with Overflow Bit
- ± 1 LSB Accuracy (Typ.)
- Single Supply Voltage (4V to 7.5V)
- 2 Units in Series Allow 9-Bit Output
- 2 Units in Parallel Allow 30MHz Sampling Rate

Applications

- TV Video Digitizing (Industrial/Security/Broadcast)
- High-Speed A/D Conversion
- Ultrasound Signature Analysis
- Transient Signal Analysis
- High Energy Physics Research
- High Speed Oscilloscope Storage/Display
- General Purpose Hybrid ADCs
- Optical Character Recognition
- Radar Pulse Analysis
- Motion Signature Analysis
- μ P Data Acquisition Systems

Description

The CA3318C is a CMOS parallel (FLASH) analog-to-digital converter designed for applications demanding both low power consumption and high speed digitization.

The CA3318 operates over a wide full scale input voltage range of 4V up to 7.5V with maximum power consumption depending upon the clock frequency selected. When operated from a 5V supply at a clock frequency of 15MHz, the typical power consumption of the CA3318 is 150mW.

The intrinsic high conversion rate makes the CA3318 ideally suited for digitizing high speed signals. The overflow bit makes possible the connection of two or more CA3318s in series to increase the resolution of the conversion system. A series connection of two CA3318s may be used to produce a 9-bit high speed converter. Operation of two CA3318s in parallel doubles the conversion speed (i.e., increases the sampling rate from 15MHz to 30MHz).

256 paralleled auto balanced voltage comparators measure the input voltage with respect to a known reference to produce the parallel bit outputs in the CA3318.

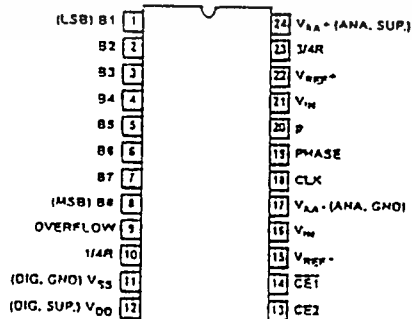
255 comparators are required to quantize all input voltage levels in this 8-bit converter, and the additional comparator is required for the overflow bit.

Ordering Information

PART NUMBER	LINEARITY (INL)	SAMPLING RATE	TEMPERATURE RANGE	PACKAGE
CA3318CE	± 1.5 LSB	15MHz (67ns)	-40°C to +85°C	24 Lead Plastic DIP
CA3318CM	± 1.5 LSB	15MHz (67ns)	-40°C to +85°C	24 Lead Plastic SOIC
CA3318CO	± 1.5 LSB	15MHz (67ns)	-40°C to +85°C	24 Lead Ceramic DIP

Pinout

CA3318C (PDIP, CDIP, SOIC)
TOP VIEW



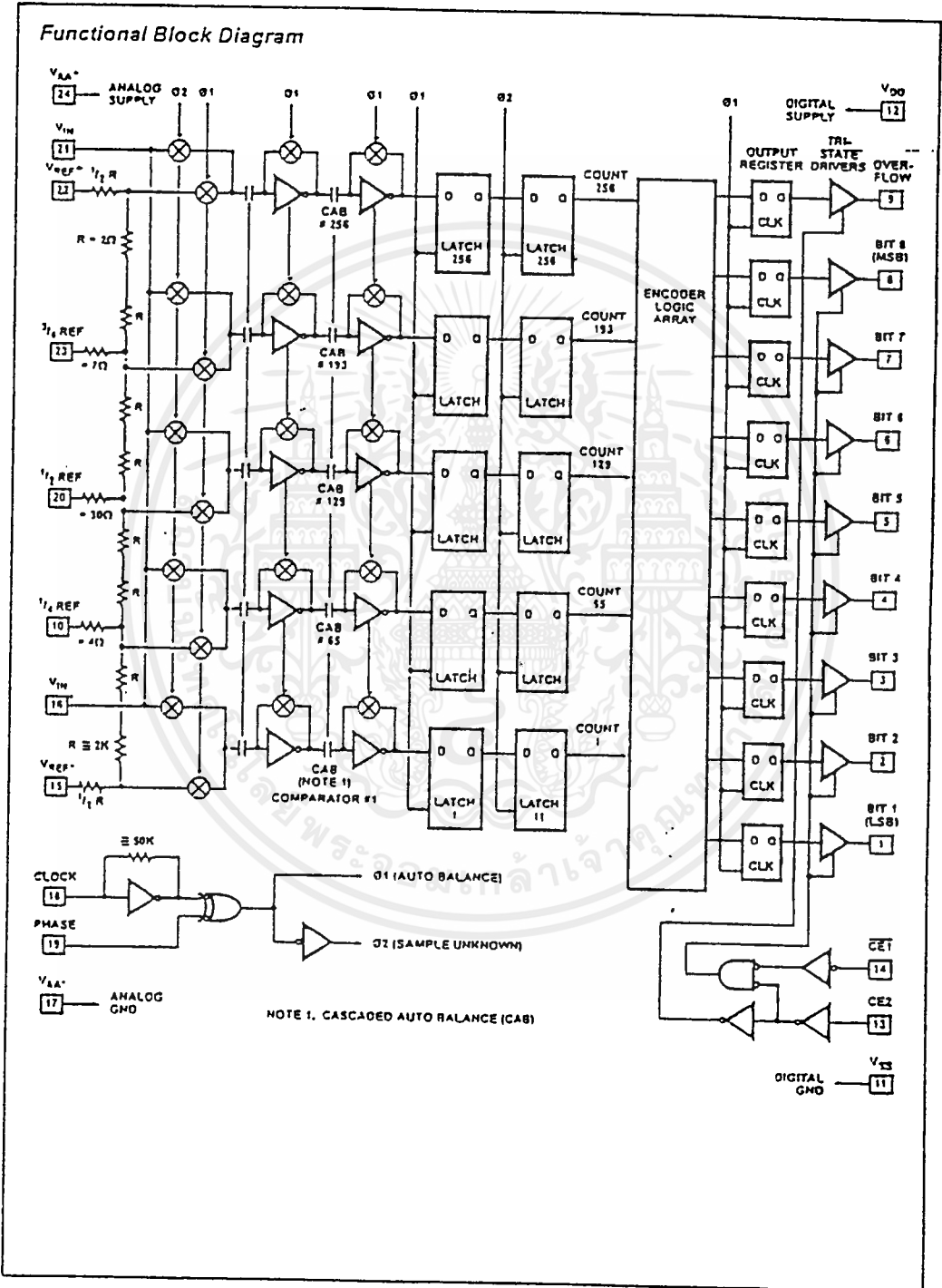
CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper ESD handling procedures.
Copyright © Harris Corporation 1993

File Number 3103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CA3318C

Functional Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specifications CA3318C

Absolute Maximum Ratings

DC Supply Voltage Range (V_{DD} or V_{AA}) -0.5V to +8V
(Referenced to V_{SS} or V_{AA} Terminal, Whichever is More Negative)

Input Voltage Range

CE2 and $\overline{CE1}$ V_{AA} - 0.5V to V_{DD} + 0.5V

Clock, Phase, V_{REF+} , $1/2$ Ref. V_{AA} - 0.5V to V_{AA} + 0.5V

Clock, Phase, V_{REF-} , $1/4$ Ref. V_{SS} - 0.5V to V_{DD} + 0.5V

V_{IN} $1/4$ REF, V_{REF+} V_{AA} - 0.5V to V_{AA} + 7.5V

Output Voltage Range V_{SS} - 0.5V to V_{DD} + 0.5V

Bits 1-8, Overflow (Outputs Off)

DC Input Current ± 20 mA

Clock, Phase, $\overline{CE1}$, CE2, V_{IN} , Bits 1-8, Overflow

Operating Voltage Range (V_{DD} or V_{AA}) 4V Min to 7.5V Max

Recommended V_{AA} Operating Range $V_{DD} \pm 1$ V

Recommended V_{AA} Operating Range $V_{SS} \pm 1$ V

Storage Temperature Range -55°C to +150°C

Lead Temperature (Soldering 10s) +265°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Thermal Information

Thermal Resistance

Ceramic DIP Package θ_{JA} 58°C/W θ_{JC} 11°C/W

Plastic DIP Package 50°C/W

Plastic SOIC Package 75°C/W

Maximum Power Dissipation 0.67W

Operating Temperature Range (T_A) -40°C to +85°C

Junction Temperature

Ceramic Package +175°C

Plastic Package +150°C

Electrical Specifications At +25°C, $V_{AA} = V_{DD} = 5$ V, $V_{REF+} = 4$ V, $V_{REF-} = V_{AA} - V_{SS}$, CLK = 15MHz
All Reference Points Adjusted, Unless Otherwise Specified.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
SYSTEM PERFORMANCE					
Resolution		8	-	-	Bits
Integral Linearity Error		-	-	± 1.5	LSB
Differential Linearity Error		-	-	+1, -0.8	LSB
Offset Error, Unadjusted	$V_{IN} = V_{REF+} - 1/2$ LSB	-0.5	4.5	6.4	LSB
Gain Error Unadjusted	$V_{IN} = V_{REF+} - 1/2$ LSB	-1.5	0	1.5	LSB
DYNAMIC CHARACTERISTICS					
Maximum Input Bandwidth	(Note 1) CA3318C	2.5	5.0	-	MHz
Maximum Conversion Speed	CLK = Square Wave	15	17	-	MSPS
Signal to Noise Ratio (SNR) $\frac{\text{RMS Signal}}{\text{RMS Noise}}$	$F_S = 15$ MHz, $f_{IN} = 100$ kHz	-	47	-	dB
	$F_S = 15$ MHz, $f_{IN} = 4$ MHz	-	43	-	dB
Signal to Noise Ratio (SINAD) $\frac{\text{RMS Signal}}{\text{RMS Noise} + \text{Distortion}}$	$F_S = 15$ MHz, $f_{IN} = 100$ kHz	-	45	-	dB
	$F_S = 15$ MHz, $f_{IN} = 4$ MHz	-	35	-	dB
Total Harmonic Distortion, THD	$F_S = 15$ MHz, $f_{IN} = 100$ kHz	-	-46	-	dBc
	$F_S = 15$ MHz, $f_{IN} = 4$ MHz	-	-36	-	dBc
Effective Number of Bits (ENOB)	$F_S = 15$ MHz, $f_{IN} = 100$ kHz	-	7.2	-	Bits
	$F_S = 15$ MHz, $f_{IN} = 4$ MHz	-	5.5	-	Bits
Differential Gain Error	Unadjusted	-	2	-	%
Differential Phase Error	Unadjusted	-	1	-	%
ANALOG INPUTS					
Full Scale Range, V_{IN} and $(V_{REF+}) - (V_{REF-})$	Notes 2, 4	4	-	7	V
Input Capacitance, V_{IN}		-	30	-	pF
Input Current, V_{IN} (See Text)	$V_{IN} = 5.0$ V, $V_{REF+} = 5.0$ V	-	-	3.5	mA
REFERENCE INPUTS					
Ladder Impedance		270	500	800	Ω

Specifications CA3318C

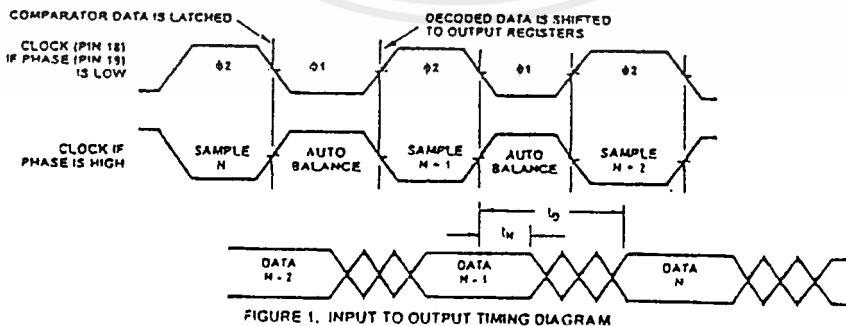
Electrical Specifications At -25°C , $V_{AA} = V_{DD} = 5\text{V}$, $V_{REF} = 0.4\text{V}$, $V_{REF} = V_{AA} = V_{SS}$, CLK = 15MHz.
All Reference Points Adjusted, Unless Otherwise Specified. (Continued)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
DIGITAL INPUTS					
Low Level Input Voltage, V_{OL} $\overline{\text{CE1}}, \text{CE2}$	Note 4	-	-	$0.2V_{DD}$	V
Phase, CLK	Note 4	-	-	$0.2V_{AA}$	V
High Level Input Voltage, V_{IH} $\overline{\text{CE1}}, \text{CE2}$	Note 4	$0.7V_{DD}$	-	-	V
Phase, CLK	Note 4	$0.7V_{AA}$	-	-	V
Input Leakage Current, I_I (Except CLK Input)	Note 3	-	± 0.2	± 5	μA
Input Capacitance, C_I		-	3	-	pF
DIGITAL OUTPUTS					
Output Low (Sink) Current	$V_O = 0.4\text{V}$	4	10	-	mA
Output High (Source) Current	$V_O = 4.5\text{V}$	-4	-8	-	mA
Tri-State Output Off-State Leakage Current, I_{OZ}		-	± 0.2	± 5	μA
Output Capacitance, C_O		-	4	-	pF
TIMING CHARACTERISTICS					
Auto Balance Time (t_1)		33	-	∞	ns
Sample Time (t_2)	Note 4	25	-	500	ns
Aperture Delay		-	15	-	ns
Aperture Jitter		-	100	-	ps
Data Valid Time, T_D	Note 4	-	50	55	ns
Data Hold Time, T_H	Note 4	25	40	-	ns
Output Enable Time, T_{EN}		-	18	-	ns
Output Disable Time, T_{DIS}		-	18	-	ns
POWER SUPPLY CHARACTERISTICS					
Device Current ($I_{DD} + I_A$) (Excludes I_{REF})	Continuous Conversion (Note 4)	-	30	50	mA
	Auto Balance ($\phi 1$)	-	30	60	mA

NOTES.

- A full scale sine wave input of greater than $F_{CLK}/2$ or the specified input bandwidth (whichever is less) may cause an erroneous code. The -3dB bandwidth for frequency response purposes is greater than 30MHz.
- V_{IN} (Full Scale) or V_{REF} should not exceed $V_{AA} + 1.5\text{V}$ for accuracy.
- The clock input is a CMOS inverter with a 50k Ω feedback resistor and may be AC coupled with 1V $_{pk}$ minimum source.
- Parameter not tested, but guaranteed by design or characterization.

Timing Waveforms



CA3318C

Timing Waveforms (Continued)

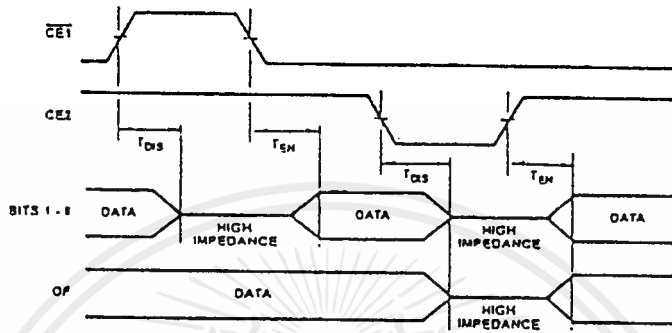


FIGURE 2. OUTPUT ENABLE TIMING DIAGRAM

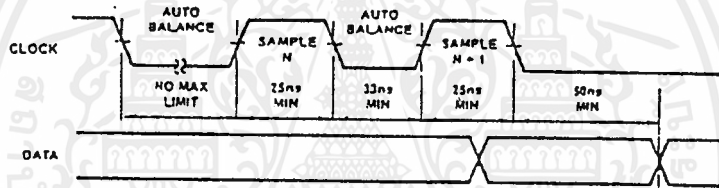


FIGURE 3A. STANDBY IN INDEFINITE AUTO BALANCE (SHOWN WITH PHASE = LOW)

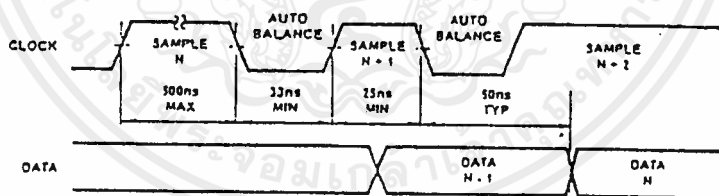


FIGURE 3B. STANDBY IN SAMPLE (SHOWN WITH PHASE = LOW)

FIGURE 3. PULSE MODE OPERATION

CA3318C

Typical Performance Curves

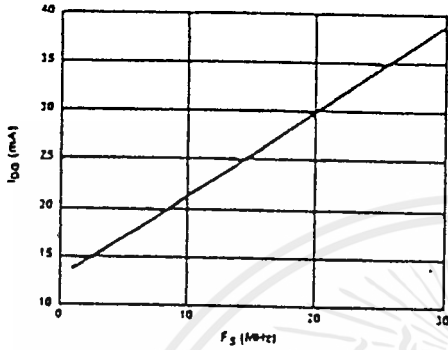


FIGURE 4. DEVICE CURRENT vs SAMPLE FREQUENCY

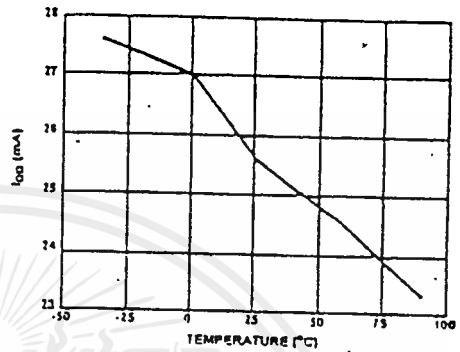


FIGURE 5. DEVICE CURRENT vs TEMPERATURE

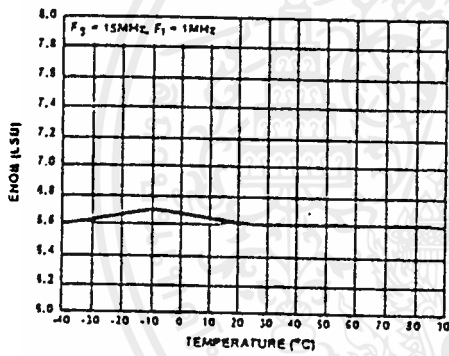


FIGURE 6. ENOB vs TEMPERATURE

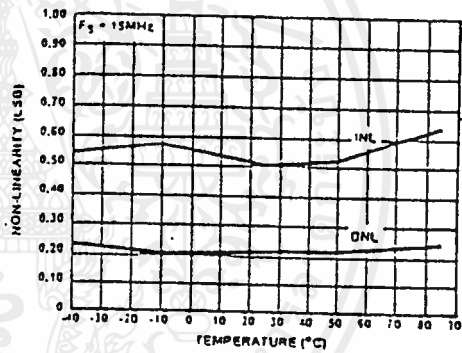


FIGURE 7. NON-LINEARITY vs TEMPERATURE

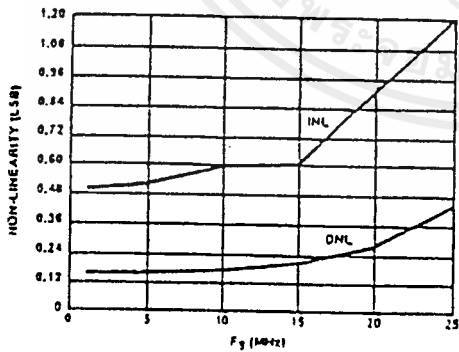


FIGURE 8. NON-LINEARITY vs SAMPLE FREQUENCY

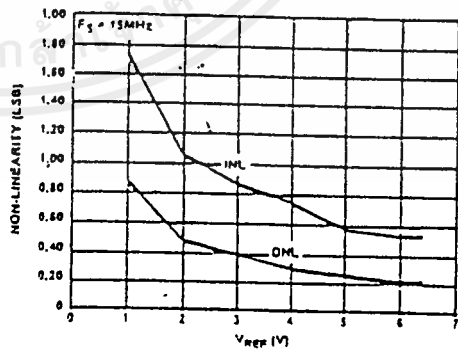


FIGURE 9. NON-LINEARITY vs REFERENCE VOLTAGE

CA3318C

Typical Performance Curves (Continued)

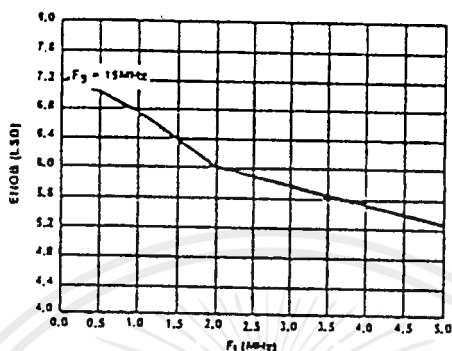


FIGURE 10. ENOB vs INPUT FREQUENCY

Pin Descriptions

PIN	NAME	DESCRIPTION
1	B1	Bit 1 (LSB)
2	B2	Bit 2
3	B3	Bit 3
4	B4	Bit 4
5	B5	Bit 5
6	B6	Bit 6
7	B7	Bit 7
8	B8	Bit 8 (MSB)
9	OF	Overflow
10	V _{1R}	Reference Ladder V ₁ Point
11	V _{SS}	Digital Ground
12	V _{DD}	Digital Power Supply, +5V
13	CE2	Tri-State Output Enable Input, Active Low. See Truth Table.
14	CE1	Tri-State Output Enable Input Active High. See Truth Table.
15	V _{REF-}	Reference Voltage Negative Input
16	V _{IN}	Analog Signal Input
17	V _{AA-}	Analog Ground
18	CLK	Clock Input
19	PHASE	Sample clock phase control input. When PHASE is low, "Sample Unknown" occurs when the clock is low and "Auto Balance" occurs when the clock is high (see text).
20	V _{1R}	Reference Ladder Midpoint
21	V _{IN}	Analog Signal Input
22	V _{REF+}	Reference Voltage Positive Input
23	V _{1R}	Reference Ladder V ₁ Point
24	V _{AA+}	Analog Power Supply, +5V

CHIP ENABLE TRUTH TABLE

CE1	CE2	B1 - B8	OF
0	1	Valid	Valid
1	1	Tri-State	Valid
X	0	Tri-State	Tri-State

X = Don't Care

Theory of Operation

A sequential parallel technique is used by the CA3318 converter to obtain its high speed operation. The sequence consists of the "Auto-Balance" phase, φ₁, and the "Sample Unknown" phase, φ₂. (Refer to the circuit diagram.) Each conversion takes one clock cycle*. With the phase control (pin 19) high, the "Auto-Balance" (φ₁) occurs during the high period of the clock cycle, and the "Sample Unknown" (φ₂) occurs during the low period of the clock cycle.

* The device requires only a single phase clock. The terminology of φ₁ and φ₂ refers to the high and low periods of the same clock.

During the "Auto-Balance" phase, a transmission switch is used to connect each of the first set of 256 commutating capacitors to their associated ladder reference tap. Those tap voltages will be as follows:

$$V_{TAP}(N) = \left\{ \frac{N}{256} V_{REF} \right\} - \left\{ \frac{1}{512} \right\} V_{REF}$$

$$= \left\{ \left(2N - 1 \right) / 512 \right\} V_{REF}$$

Where:

- V_{TAP}(n) = reference ladder tap voltage at point n.
- V_{REF} = voltage across V_{REF+} to V_{REF-}
- N = tap number (1 through 256)

The other side of these capacitors are connected to single-stage amplifiers whose outputs are shorted to their inputs by switches. This balances the amplifiers at their intrinsic tap points, which is approximately (V_{AA+} + V_{AA-})/2. The first set of capacitors now charges to their associated tap voltages.

CA3318C

At the same time a second set of commutating capacitors and amplifiers is also auto-balanced. The balancing of the second-stage amplifier at its intrinsic trip point removes any tracking differences between the first and second amplifier stages. The cascaded auto-balance (CAB) technique, used here, increases comparator sensitivity and temperature tracking.

In the "Sample Unknown" phase, all ladder tap switches and comparator shunting switches are opened. At the same time V_{IN} is switched to the first set of commutating capacitors. Since the other end of the capacitors are now looking into an effectively open circuit, any input voltage that differs from the previous tap voltage will appear as a voltage shift at the comparator amplifiers. All comparators that had tap voltages greater than V_{IN} will go to a "high" state at their outputs. All comparators that had tap voltages lower than V_{IN} will go to a "low" state.

The status of all these comparator amplifiers is AC coupled through the second-stage comparator and stored at the end of this phase (62) by a latching amplifier stage. The latch feeds a second latching stage, triggered at the end of 61. This delay allows comparators extra settling time. The status of the comparators is decoded by a 256 to 9-bit decoder array, and the results are clocked into a storage register at the end of the next 62.

A 3-stage buffer is used at the output of the 9 storage registers which are controlled by two chip-enable signals. CE1 will independently disable B1 through B6 when it is in a high state. CE2 will independently disable B1 through B8 and the OF buffers when it is in the low state.

To facilitate usage of this device, a phase control input is provided which can effectively complement the clock as it enters the chip.

Continuous-Clock Operation

One complete conversion cycle can be traced through the CA3318 via the following steps. (Refer to timing diagram.) With the phase control in a "low" state, the rising edge of the clock input will start a "sample" phase. During this entire "high" state of the clock, the comparators will track the input voltage and the first-stage latches will track the comparator outputs. At the falling edge of the clock, all 256 comparator outputs are captured by the 256 latches. This ends the "sample" phase and starts the "auto-balance" phase for the comparators. During this "low" state of the clock, the output of the latches settles and is captured by a second row of latches when the clock returns high. The second-stage latch output propagates through the decode array, and a 9-bit code appears at the D inputs of the output registers. On the next falling edge of the clock, this 9-bit code is shifted into the output registers and appears with time delay t_D as valid data at the output of the tri-state drivers. This also marks the end of the next "sample" phase, thereby repeating the conversion process for this next cycle.

Pulse-Mode Operation

The CA3318 needs two of the same polarity clock edges to complete a conversion cycle: If, for instance, a negative going clock edge ends sample "N", then data "N" will appear after the next negative going edge. Because of this requirement, and because there is a maximum sample time of 500ns (due to capacitor droop), most pulse or intermittent sample applications will require double clock pulsing.

If an indefinite standby state is desired, standby should be in auto-balance, and the operation would be as in Figure 3A.

If the standby state is known to last less than 500ns and lowest average power is desired, then operation could be as in Figure 3B.

Increased Accuracy

In most cases the accuracy of the CA3318 should be sufficient without any adjustments. In applications where accuracy is of utmost importance, five adjustments can be made to obtain better accuracy, i.e., offset trim; gain trim; and $1/4$, $1/2$ and $3/4$ point trim.

Offset Trim

In general, offset correction can be done in the preamp circuitry by introducing a dc shift to V_{IN} or by the offset trim of the op amp. When this is not possible the V_{REF} input can be adjusted to produce an offset trim. The theoretical input voltage to produce the first transition is $1/2$ LSB. The equation is as follows:

$$V_{IN} \text{ (0 to 1 transition)} = \frac{1}{2} \text{ LSB} = \frac{1}{2} (V_{REF}/256) \\ = V_{REF}/512$$

If V_{IN} for the first transition is less than the theoretical, then a single-turn 50 Ω pot connected between V_{REF} and ground will accomplish the adjustment. Set V_{IN} to $1/2$ LSB and trim the pot until the 0-to-1 transition occurs.

If V_{IN} for the first transition is greater than the theoretical, then the 50 Ω pot should be connected between V_{REF} and a negative voltage of about 2 LSB's. The trim procedure is as stated previously.

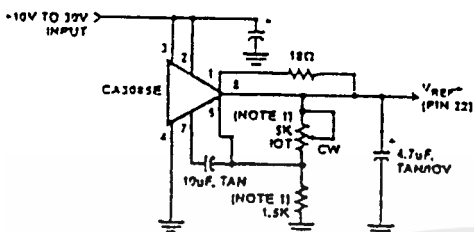
Gain Trim

In general, the gain trim can also be done in the preamp circuitry by introducing a gain adjustment for the op amp. When this is not possible, then a gain adjustment circuit should be made to adjust the reference voltage. To perform this trim, V_{IN} should be set to the 255 to overflow transition. That voltage is $1/2$ LSB less than $V_{REF} +$ and is calculated as follows:

$$V_{IN} \text{ (255 to 256 transition)} = V_{REF} - V_{REF}/512 \\ = V_{REF}(511/512)$$

To perform the gain trim, first do the offset trim and then apply the required V_{IN} for the 255 to overflow transition. Now adjust $V_{REF} +$ until that transition occurs on the outputs.

CA3318C

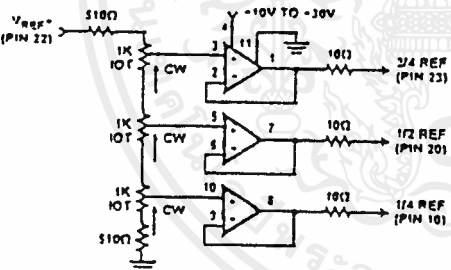


NOTE: Bypass V_{REF+} to analog GND near A/D with 0.1μF ceramic cap. Parts noted should have low temperature drift.

FIGURE 11. TYPICAL VOLTAGE REFERENCE SOURCE FOR DRIVING V_{REF+} INPUT

$1/4$ Point Trims

The $1/4$, $1/2$ and $3/4$ points on the reference ladder are brought out for linearity adjusting or if the user wishes to create a nonlinear transfer function. The $1/4$ points can be driven by the reference drivers shown (Figure 12) or by 2-K pots connected between V_{REF+} and V_{REF-} . The $1/2$ (mid-) point should be set first by applying an input of $257/512 \times (V_{REF})$ and adjusting for an output changing from 128 to 129. Similarly the $1/4$ and $3/4$ points can be set with inputs of $129/512$ and $385/512 \times (V_{REF})$ and adjusting for counts of 192 to 193 and 64 to 65. (Note that the points are actually $1/4$, $1/2$ and $3/4$ of full scale +1 LSB.)



NOTES:

1. All Op Amps = $2/4$ CA324E
2. Bypass all reference points to analog ground near A/D with 0.1μF ceramic caps.
3. Adjust V_{REF+} first, then $1/2$, $1/4$, and $3/4$ points.

FIGURE 12. TYPICAL $1/4$ POINT DRIVERS FOR ADJUSTING LINEARITY (USE FOR MAXIMUM LINEARITY)

9-Bit Resolution

To obtain 9-bit resolution, two CA3318's can be wired together. Necessary ingredients include an open-ended ladder network, an overflow indicator, tri-state outputs, and chip-enable controls—all of which are available on the CA3318.

The first step for connecting a 9-bit circuit is to totem-pole the ladder networks, as illustrated in Figure 13. Since the absolute resistance value of each ladder may vary, external trim of the mid-reference voltage may be required.

The overflow output of the lower device now becomes the ninth bit. When it goes high, all counts must come from the upper device. When it goes low, all counts must come from the lower device. This is done simply by connecting the lower overflow signal to the $\overline{CE1}$ control of the lower A/D converter and the CE2 control of the upper A/D converter. The tri-state outputs of the two devices (bits 1 through 8) are now connected in parallel to complete the circuitry. The complete circuit for a 9-bit A/D converter is shown in Figure 14.

Grounding/Bypassing

The analog and digital supply grounds of a system should be kept separate and only connected at the A/D. This keeps digital ground noise out of the analog data to be converted. Reference drivers, input amps, reference lops, and the V_{AA} supply should be bypassed at the A/D to the analog side of the ground. See Figure 15 for a block diagram of this concept. All capacitors shown should be low impedance 0.1μF ceramics and should be mounted as close to the A/D as possible. If V_{AA+} is derived from V_{DD} , a small (10Ω) resistor or inductor and additional filtering (4.7μF tantalum) may be used to keep digital noise out of the analog system.

Input Loading

The CA3318 outputs a current pulse to the V_{IN} terminal at the start of every sample period. This is due to capacitor charging and switch feedthrough and varies with input voltage and sampling rate. The signal source must be capable of recovering from the pulse before the end of the sample period to guarantee a valid signal for the A/D to convert. Suitable high speed amplifiers include the HA-5033, HA-2542; and CA3450. Figure 16 is an example of an amplifier which recovers fast enough for sampling at 15MHz.

Output Loading

The CMOS digital output stage, although capable of driving large loads, will reflect these loads into the local ground. It is recommended that a local CMOS buffer such as CD74HC541 E be used to isolate capacitive loads.

Definitions

Dynamic Performance Definitions

Fast Fourier Transform (FFT) techniques are used to evaluate the dynamic performance of the converter. A low distortion sine wave is applied to the input, it is sampled, and the output is stored in RAM. The data is then transformed into the frequency domain with a 4095 point FFT and analyzed to evaluate the dynamic performance of the A/D. The sine wave input to the part is -0.5dB down from fullscale for all these tests.

CA3318C

Signal-to-Noise (SNR)

SNR is the measured RMS signal to RMS noise at a specified input and sampling frequency. The noise is the RMS sum of all of the spectral components except the fundamental and the first five harmonics.

Signal-to-Noise + Distortion Ratio (SINAD)

SINAD is the measured RMS signal to RMS sum of all other spectral components below the Nyquist frequency excluding DC.

Effective Number of Bits (ENOB)

The effective number of bits (ENOB) is derived from the SINAD data. ENOB is calculated from:

$$ENOB = (SINAD - 1.76 - V_{CORR})/6.02$$

where: $V_{CORR} = 0.5dB$

Total Harmonic Distortion (THD)

THD is the ratio of the RMS sum of the first 5 harmonic components to the RMS value of the measured input signal.

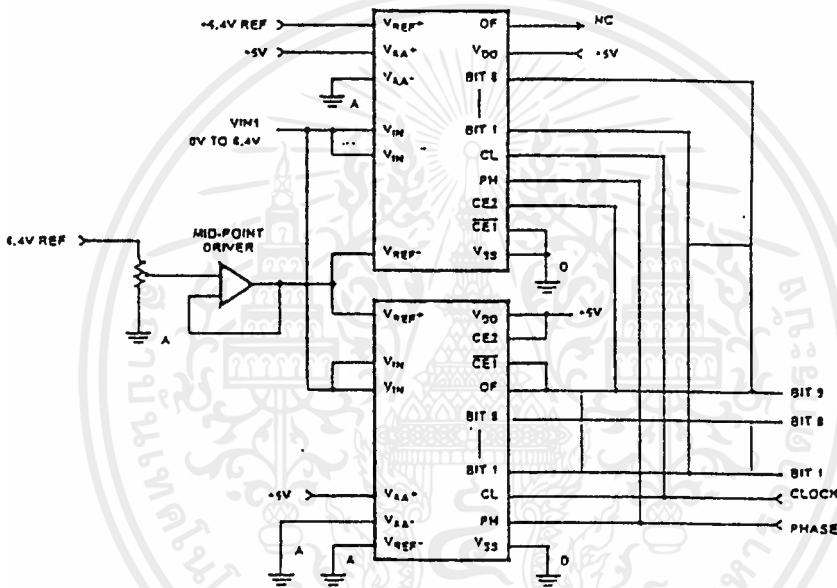


FIGURE 13. USING TWO CA3318s FOR 9-BIT RESOLUTION

CA3318C

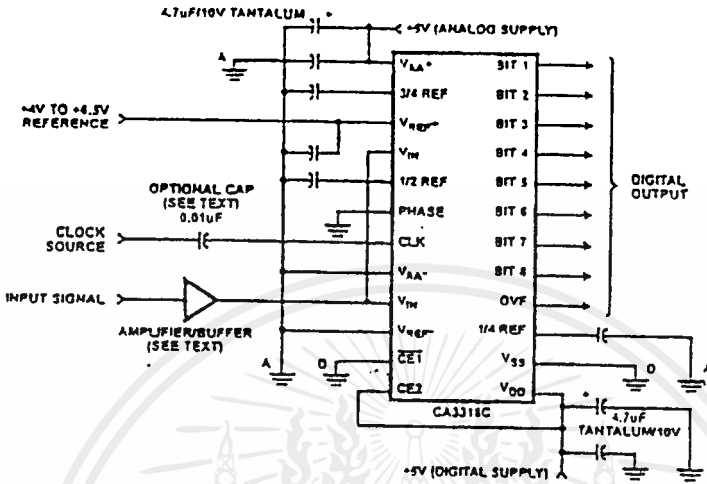


FIGURE 14. TYPICAL CIRCUIT CONFIGURATION FOR THE CA3318 WITH NO LINEARITY ADJUST

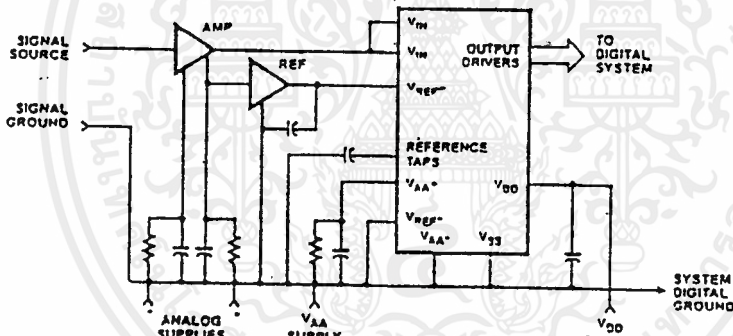


FIGURE 15. TYPICAL SYSTEM GROUNDING/BYPASSING

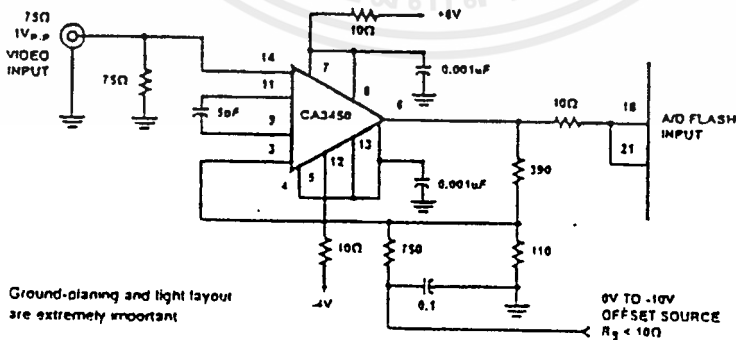


FIGURE 16. TYPICAL HIGH BANDWIDTH AMPLIFIER FOR DRIVING THE CA3318

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CA3318C

TABLE 1. OUTPUT CODE TABLE

CODE DESCRIPTION	INPUT VOLTAGE (NOTE 1)		BINARY OUTPUT CODE									DECIMAL COUNT
	V _{REF} 5.40V (V)	V _{REF} 5.12V (V)	OF	MSB 88	87	86	85	84	83	82	LSB 81	
Zero	0.00	0.00	0	0	0	0	0	0	0	0	0	0
1 LSB	0.025	0.02	0	0	0	0	0	0	0	0	0	1
2 LSB	0.05	0.04	0	0	0	0	0	0	0	0	1	0
.
.
1/4 Full Scale	1.60	1.28	0	0	1	0	0	0	0	0	0	0
.
.
1/2 Full Scale - 1 LSB	3.175	2.54	0	0	1	1	1	1	1	1	1	1
1/2 Full Scale	3.20	2.56	0	1	0	0	0	0	0	0	0	0
1/2 Full Scale + 1 LSB	3.225	2.58	0	1	0	0	0	0	0	0	0	1
.
.
3/4 Full Scale	4.80	3.84	0	1	1	0	0	0	0	0	0	0
.
.
Full Scale - 1 LSB	5.35	5.08	0	1	1	1	1	1	1	1	1	0
Full Scale	5.375	5.10	0	1	1	1	1	1	1	1	1	1
Over Flow	6.40	5.12	1	1	1	1	1	1	1	1	1	1

NOTE: 1. The voltages listed above are the ideal centers of each output code shown as a function of its associated reference voltage.

Reducing Power

Most power is consumed while in the auto-balance state. When operating at lower than 15MHz clock speed, power can be reduced by stretching the sample (ϕ) time. The constraints are a minimum balance time (ϕ) of 33ns, and a maximum sample time of 500ns. Longer sample times cause droop in the auto-balance capacitors. Power can also be reduced in the reference string by switching the reference on only during auto-balance.

Clock Input

The Clock and Phase inputs feed buffers referenced to V_{AA+} and V_{AA-}. Phase should be tied to one of these two potentials, while the clock (if DC coupled) should be driven at least from 0.2 to 0.7 x (V_{AA+} - V_{AA-}). The clock may also be AC coupled with at least a 1 V_{p,p} swing. This allows TTL drive levels or 5V CMOS levels when V_{AA+} is greater than 5V.

54374/74374 Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

	Schottky TTL		High-Speed TTL		Low-Power Schottky TTL		Standard TTL		Low-Power TTL	
	Device Type	Package	Device Type	Package	Device Type	Package	Device Type	Package	Device Type	Package
	C:P:M:ICF		C:P:M:CF		C:P:M:ICF		C:P:M:ICF		C:P:M:ICF	
TTL	SN54374 SN74374	DIP-16			SN54LS374 SN74LS374	DIP-16				
FAIRCHILD										
MOTOROLA										
N.S.C.										
PHILIPS										
SIGNETICS										
SIEMENS										
FUJITSU										
HITACHI										
MITSUBISHI										
NEC										
TOSHIBA										

Electrical Characteristics SN54LS374/SN74LS374

absolute maximum ratings over operating free-air temperature range

Supply voltage, V _{CC}	1 V	Operating power	SN54LS374	-55°C to +125°C
Input voltage	1 V	Temperature range	SN74LS374	0°C to +70°C
		Storage temperature range		-55°C to +150°C

recommended operating conditions

PARAMETER	SN54LS374		SN74LS374		UNITS
	MIN	NOM	MAX	MIN	
Supply voltage, V _{CC}	5.0	5.0	5.5	5.0	V
Low-level output voltage, V _{OL}			-0.5		V
High-level output current, I _{OH}			16		mA
Width of noise shaping pulse, t _w	High	15	15		ns
	Low	15	15		ns
Data hold time, t _{hold}		0	2		ns
Setup time, t _{setup}		20	20		ns
Operating storage temperature, T _s	0	125	0	20	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

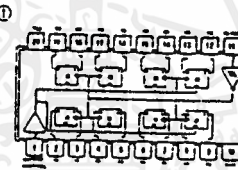
PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
V _{OH}	High-level output voltage			0.1	V
V _{OL}	Low-level output voltage			-0.5	V
I _{OH}	High-level output current	V _{CC} = MAX, I _{OL} = 16mA		16	mA
I _{OL}	Low-level output current	V _{CC} = MIN, V _{OL} = 0.1V, I _{OH} = 16mA	0.35	0.1	mA
I _{OHZ}	Off-state output current, 0-gate voltage applied	V _{CC} = MAX, V _{OL} = 0V, I _{OL} = 16mA		20	µA
I _{OLZ}	Off-state output current, 1-gate voltage applied	V _{CC} = MAX, V _{OL} = 0V, I _{OH} = 16mA		-20	µA
I _I	Input current at maximum input voltage	V _{CC} = MAX, V _I = 1V		1.1	mA
I _{IS}	Input current at low-level input voltage	V _{CC} = MAX, V _I = 0.1V		20	µA
I _{IS}	Input current at high-level input voltage	V _{CC} = MAX, V _I = 1V		-1.1	mA
I _{OS}	Short-circuit output current	V _{CC} = MAX		30	mA
I _{CC}	Supply current	V _{CC} = MAX, Output driven at 1V LS374		21	mA

switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNITS
t _{prop}				15	16		ns
t _{PLH}	0	Any 0	C _L = 15pF, R _L = 4kΩ, 3σ Max (1 unit)		15	21	ns
t _{PLZ}	0	Any 0			20	21	ns
t _{PHL}	Any 1	Any 0			21	21	ns
t _{PHZ}	Any 1	Any 0			12	15	ns
t _{tr}	Control	Any 0			14	21	ns

For conditions given in MIN or MAX, use the appropriate value specified under recommended operating conditions.
 1. All timing values are at V_{CC} = 5V, T_A = 25°C.
 2. Test setup: the output should be driven at 1 mA and duration of the short circuit should not exceed one second.

Pin Assignment (Top View)



SN54LS374 LR SN74LS374 J2, M1
 SN54LS374 LR SN74LS374 J2, M1

LS374, LS374 FUNCTION TABLE

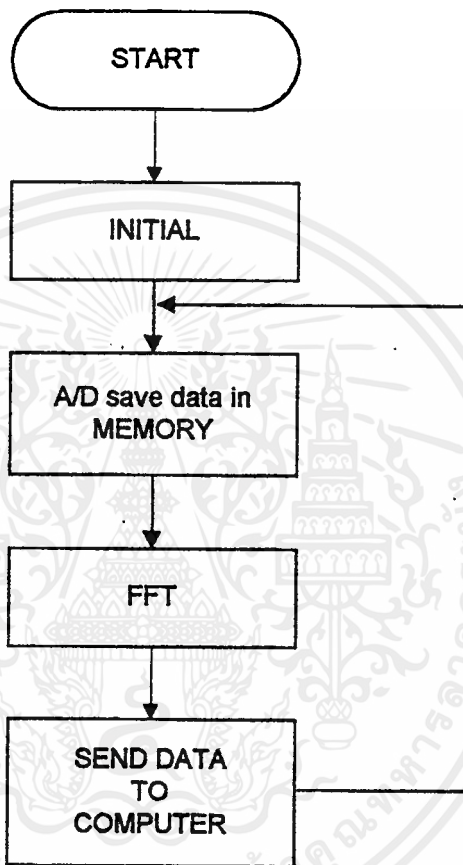
OUTPUT CONTROL	CLOCK	0	OUTPUT
L	1	H	H
L	1	L	L
L	0	X	Q _n
H	0	X	Q

NOTE: 1. Maximum clock frequency is listed with all inputs held at 1. 2. See also outputs and waveforms on page 3-11. 3. See maximum output frequency. 4. L_{OH} = propagation delay time, high-to-low-level output. 5. L_{OL} = propagation delay time, low-to-low-level output. 6. P_{OH} = output delay time to high level. 7. P_{OL} = output delay time to low level. 8. P_{HL} = output delay time from high level. 9. P_{LL} = output delay time from low level.

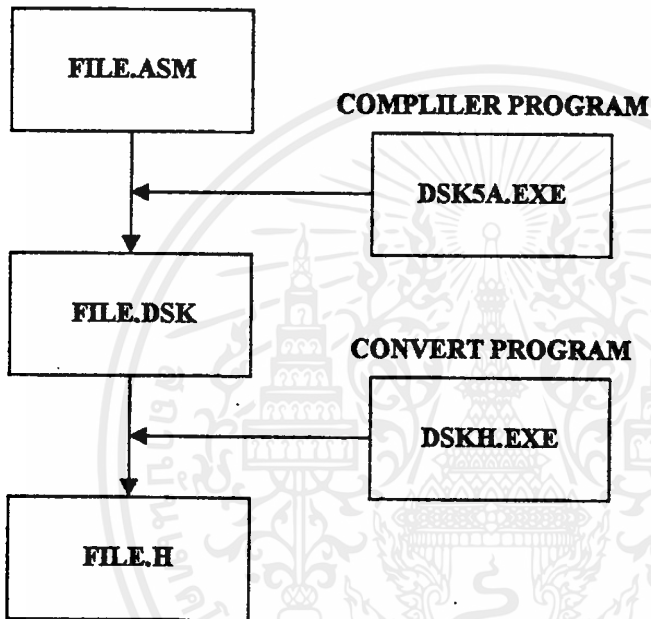
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



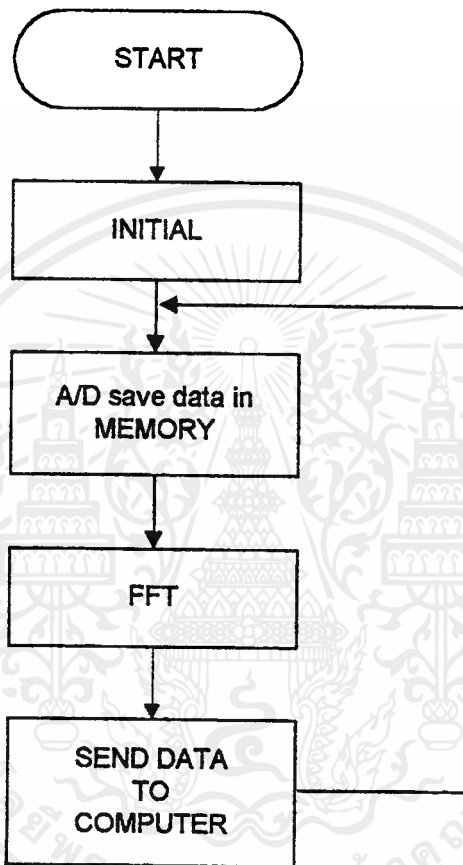
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



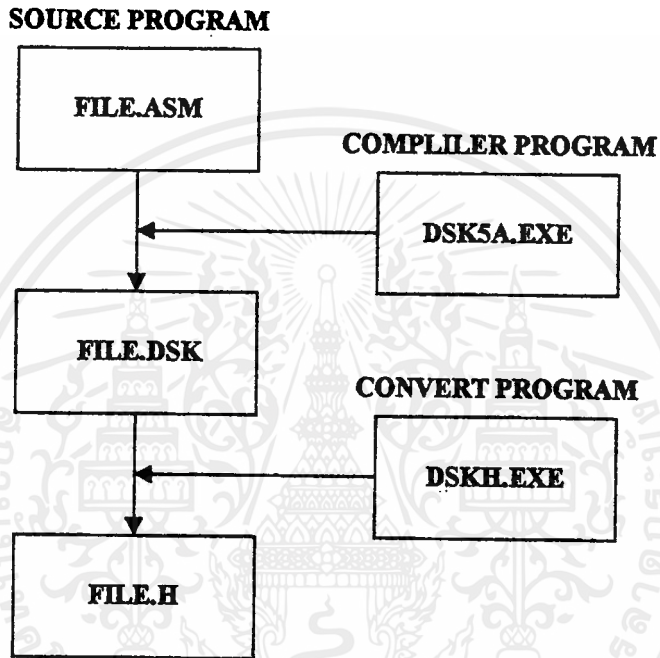
รูปที่ ค.1 ผังการทำงานของโปรแกรมภาษาแอสเซมบลี

SOURCE PROGRAM

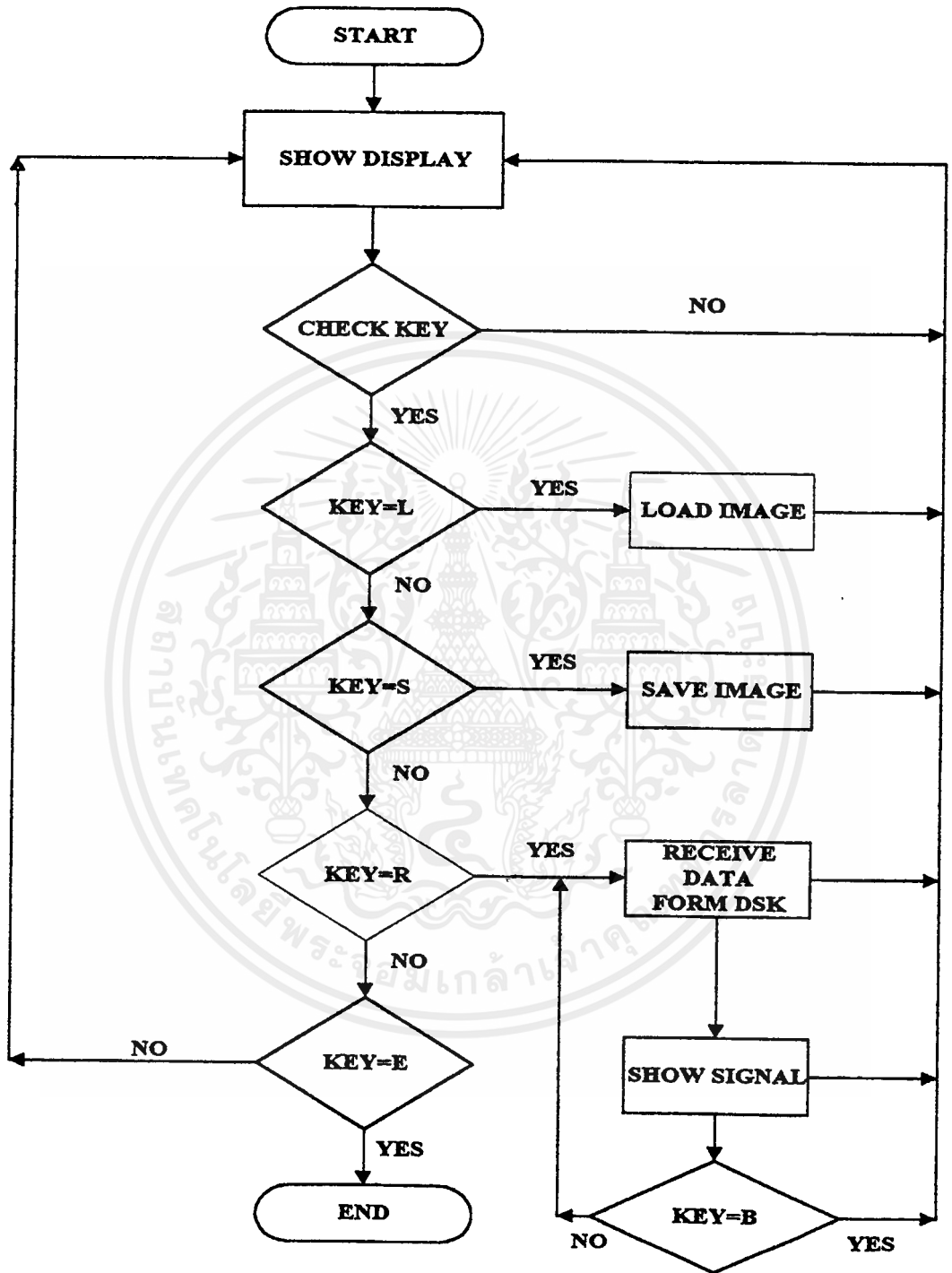
รูปที่ ค.2 ขั้นตอนแปลงโปรแกรมแอสเซมบลี DSK



รูปที่ ค.1 ผังการทำงานของโปรแกรมภาษาแอสเซมบลี

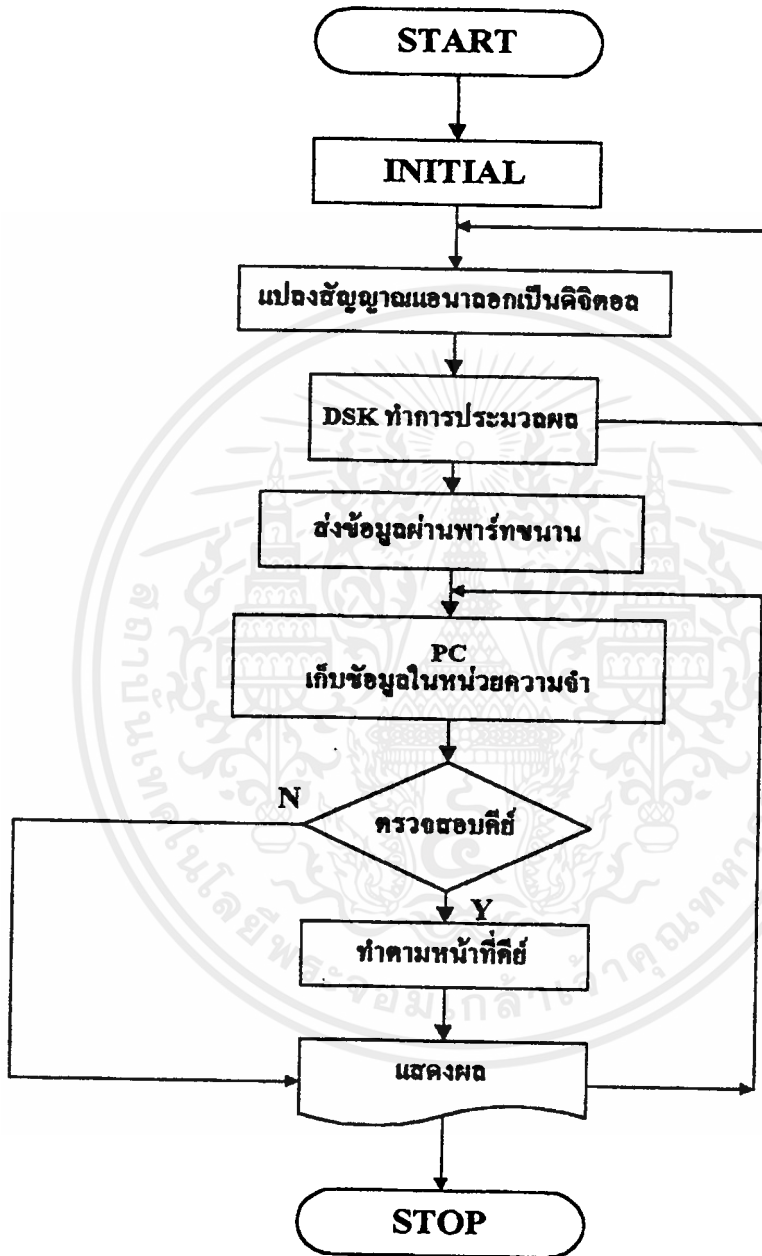


รูปที่ ค.2 ขั้นตอนแปลง โปรแกรมแอสเซมบลี DSK



รูปที่ ค.3 ผังการทำงานของโปรแกรมภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.4 ผังการทำงานทั้งหมดของโปรแกรมของเครื่องวิเคราะห์แถบความถี่

```
***** PROGRAM DISPLAY *****
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <alloc.h>
```

```
#include <dos.h>
```

```
#include <math.h>
```

```
#include <graphics.h>
```

```
#include <process.h>
```

```
#include <pcx.c>
```

```
#include <vga1.c>
```

```
#define DATAPORT 0X378
```

```
#define STATUSPORT 0X379
```

```
#define CONTROLPORT 0X37A
```

```
#define SM 0x80
```

```
#define CM 0x0B
```

```
#define ESC 0x1b
```

```
#define UP 0x48
```

```
#define DOWN 0x50
```

```
#define LIFT 0x4b
```

```
#define RIGHT 0x4d
```

```
#define A_UP 0x98 //ALT & UP
```

```
#define A_DOWN 0xa0 //ALT & DOWN
```

```
#define A_LIFT 0x9b //ALT & LIFT
```

```
#define A_RIGHT 0x9d //ALT & RIGHT
```

```
int conv(int ch);
```

```
void sport(int port, char ch);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char rport(int port);
char check_stat(int port);
void port_init(int port,char code);
void readdsk(char s);
void Run();
void Initialize(void);
void Vtable(int sx,int sy,int ex,int ey,int wx);
void Htable(int sx,int sy,int ex,int ey,int wy);
void Lline(int sx,int sy,int ex,int ey);
void Xline(int sx,int sy,int lx,int ly);
void Hline(int sx,int ex,int yy);
void Vline(int xx,int sy,int ey);
void HlineT(int sx,int ex,int yy);
void VlineT(int xx,int sy,int ey);
void Send_pg0;
void W_text0;
void W_name0;
void Menu0;
void D_box(int sx,int sy,int ex,int ey,char s);
void Select(void);
void Load0;
void Save(void);
void Still0;
char Exit(void);
void S0;

int x,y,nx,ny,nn;

```

```

int bl[2][600],bh[2][600];

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int ch,c,i,j,k,m,in;
```

```
int Hz=0,Hzb;
```

```
float Maxv=0,Maxvb;
```

```
char *fname,*te;
```

```
FILE *fp;
```

```
***** MAIM PROGRAM *****
```

```
main(int argc,char *argv[])
```

```
{
```

```
clrscr();
```

```
printf(" %s ",argv[1]);
```

```
if(!(fp=fopen(argv[1],"rb"))) {
```

```
printf("cannot open input file \n");
```

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

```
Send_pg();
```

```
fclose(fp);
```

```
Initialize();
```

```
setfillstyle(SOLID_FILL,LIGHTGRAY);
```

```
bar(0,0,639,479);
```

```
Menu();
```

```
D_box(80,45,630,470,2); //box scope
```

```
Lline(109,170,610,351);
```

```
Hline(110,610,350);
```

```
Vline(110,170,350);
```

```
Vtable(110,170,610,350,20);
```

```
Htable(110,170,610,350,30);
```

```
W_name();
```

```
outport(CONTROLPORT,0xff);
```

```

do{
    Select();
}while(ch!=ESC);
closegraph();
return 0;
}

```

```

int conv(int ch)
{ ch = toupper(ch);
  if(ch>=65) { ch = ((ch - 65) + 10); }
  else { ch = ch - 48; }
  return ch;
}

```

```

void sport(int port,char ch)
{ union REGS r;
  r.x.dx = port;
  r.h.al = ch;
  r.h.ah = 1;
  int86(0x14,&r,&r);
  if(r.h.ah & 128)
  { printf("send error detected in serial port...!");
    exit(1); }
}

```

```

char rport(int port)
{ union REGS r;

```

```

  r.x.dx = port;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

r.h.ah = 2;
int86(0x14,&r,&r);
return r.h.al;
}

```

```

char check_stat(int port)

```

```

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

```

```

void port_init(int port,char code)

```

```

{ union REGS r;
  r.x.dx = port;
  r.h.ah = 0;
  r.h.al = code;
  int86(0x14,&r,&r);
}

```

```

void Initialize(void)

```

```

{ int gdriver = DETECT,gmode,errorcode;
  errorcode = registerbgidriver(EGAVGA_driver);
  if(errorcode < 0)
  { printf("Graphics error: %s\n",grapherrormsg(errorcode));
    exit(1);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

initgraph(&gdriver,&gmode,"");
errorcode = graphresult();
if(errorcode != grOk)
{ printf("Graphics error : %s\n",grapherrormsg(errorcode));
  printf("Press any key to halt");
  getch(); exit(1);
}
setgraphmode(VGA);
}

void Vtable(int sx,int sy,int ex,int ey,int wx)
{ setcolor(DARKGRAY);
  setlinestyle(USERBIT_LINE,0X1111,1);
  for(i=sx;i<=ex;i+=wx)
    line(i,sy,i,ey);
}

void Htable(int sx,int sy,int ex,int ey,int wy)
{ setcolor(DARKGRAY);
  setlinestyle(USERBIT_LINE,0X1111,1);
  for(i=sy;i<=ey;i+=wy)
    line(sx,i,ex,i);
}

void Lline(int sx,int sy,int ex,int ey)
{ setcolor(RED);
  line(sx,sy,sx,ey);
  line(sx,ey,ex,ey);
}

```

```

}

void Xline(int sx,int sy,int lx,int ly)
{ setcolor(RED);
  line(sx,sy,sx,sy+ly);
  line(sx,sy+(ly/2),sx+lx,sy+(ly/2));
}

```

```

void Hline(int sx,int ex,int yy)
{ int h=2;
  for(i=20;i<(ex-sx);i+=20)
    line(sx+i,yy,sx+i,yy+h);
  for(i=1;i<6;i++) {
    sprintf(te,"%1.1f",i*1.6);
    outtextxy(sx+(i*100)-10,yy+10,te);
    line(sx+(i*100),yy,sx+(i*100),yy+h+3);
  }
}

```

```

void Vline(int xx,int sy,int ey)
{ int w=3;
  for(i=0;i<(ey-sy);i+=30)
    line(xx,sy+i,xx-w,sy+i);
  for(i=0;i<7;i++) {
    sprintf(te,"%d",6-i);
    outtextxy(xx-12,sy+(i*30)-2,te);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void HlineT(int sx,int ex,int yy)
{ int h=2;
  float i;
  for(i=20;i<(ex-sx);i+=20)
  line(sx+i,yy,sx+i,yy+h);
  for(i=1;i<6;i++) {
    sprintf(te,"%0.1f", (250/i));
    outtextxy(sx+(i*100)-18,yy+10,te);
    line(sx+(i*100),yy,sx+(i*100),yy+h+3);
  }
}

```

```

void VlineT(int xx,int sy,int ey)
{ int w=3;
  for(i=0;i<(ey-sy);i+=10)
  line(xx,sy+i,xx-w,sy+i);
  for(i=0;i<4;i++) {
    sprintf(te," %d",6-(2*i));
    outtextxy(xx-20,sy+(i*20)-2,te);
  }
  for(i=1;i<4;i++) {
    sprintf(te,"-%d",2*i);
    outtextxy(xx-20,sy+(i*20)+58,te);
  }
}

```

```

void W_text()

```

```

{ setcolor(LIGHTGRAY);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sprintf(te, "      ");
outtextxy(165,30,te);
sprintf(te, "      ");
outtextxy(370,30,te);
setcolor(RED);
Hzb = Hz;
Maxvb = Maxv;
sprintf(te, "Freq = %.2f KHz ",Hz*1333);
outtextxy(165,30,te);
sprintf(te, "Volt = %.3f V", (Maxv*0.033333));
outtextxy(370,30,te);
}

void W_name()
{ setcolor(RED);
  outtextxy(90,155,"VOLT(V)");
  settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
  outtextxy(310,375,"FREQUENCY(MHz)");
}

void D_box(int sx,int sy,int ex,int ey,char s)
{ char co1,co2;           //s = 1 = UP OR s = 2 = DOWN
  if(s==1) { co1 = 15; co2 = 8; }
  else { co1 = 8; co2 = 15; }
  setcolor(co1);
  rectangle(sx,sy,ex,ey);
  rectangle(sx+1,sy+1,ex-1,ey-1);
  setcolor(co2);

```

```

line(sx,ey,ex,ey);
line(sx+1,ey-1,ex,ey-1);
line(ex,sy,ex,ey);
line(ex-1,sy+1,ex-1,ey);
}

```

```
void Menu()
```

```

{ setbkcolor(LIGHTGRAY);
  setfillstyle(SOLID_FILL,WHITE);
  bar(0,0,639,3);
  setcolor(10);
  rectangle(19,4,619,20);
  setfillstyle(SOLID_FILL,GREEN);
  bar(20,5,618,19);
  settxtstyle(DEFAULT_FONT,HORIZ_DIR,0);
  setcolor(YELLOW);
  outtextxy(226,10,"SPECTRUM ANALYZER SCOPE           KMITL");
  outtextxy(227,10,"SPECTRUM ANALYZER SCOPE           KMITL");
  setcolor(RED);
  outtextxy(226,9,"SPECTRUM ANALYZER SCOPE           KMITL");
  setcolor(WHITE);
  line(0,3,0,479);      line(1,3,1,478);
  setcolor(DARKGRAY);
  line(639,0,639,479);  line(638,1,638,478);
  line(2,478,639,478);  line(1,479,639,479);
  setcolor(RED);
  outtextxy(20,57,"L");      outtextxy(20,87,"S");
  outtextxy(20,117,"R");     outtextxy(20,147,"B");

```

```

outtextxy(20,177,"E");
setcolor(DARKGRAY);
outtextxy(27,57,"oad");      outtextxy(27,87,"ave");
outtextxy(29,117,"un");      outtextxy(28,147,"reak");
outtextxy(28,177,"xit");

D_box(10,50,70,70,1);      //box load
D_box(10,80,70,100,1);     //box save
D_box(10,110,70,130,1);    //box run
D_box(10,140,70,160,1);    //box break
D_box(10,170,70,190,1);    //box exit

setttextstyle(DEFAULT_FONT,HORIZ_DIR,0);
D_box(160,25,340,40,2);    //box Hz
D_box(360,25,500,40,2);    //box Volt
D_box(515,25,630,40,2);    //box name.pcx
setcolor(BLUE);
outtextxy(525,30,"NONAME.PCX");
}

```

```

void Send_pg0
{ int port=0;
  port_init(port,0xe7);
  ch = 0x80;
  do{
    sport(port,ch);
  }while(rport(port)!=0x1b);

```

```
c = 0;ch = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    c = getc(fp);
    if(c==0x0d) {
        c = getc(fp);
        c = getc(fp);
    }
    c = conv(c)<<4;
    ch = getc(fp);
    ch = conv(ch);
    ch = (ch | c);
    if(!feof(fp)) break;
    printf("\t%X",ch);
    sport(port,ch);
    delay(2);
    if(kbhit()) exit(1);
    printf(":%X",rport(port)&0x0ff);
}while(!feof(fp));
ch = 0x05;
sport(port,ch);
printf("\n:%X",rport(port)&0x0ff);
ch = 0x0A;
delay(2);
sport(port,ch);
printf(":%X",rport(port)&0x0ff);
ch = 0x00;
delay(2);
sport(port,ch);
printf(":%X",rport(port)&0x0ff);

```

```

ch = 0x80;
delay(2);
sport(port,ch);
printf("\n\t\t*****()\n");
delay(3);
}

```

```

void Select(void)
{
if(kbhit())
{
setlinestyle(USERBIT_LINE,0X0FFFF,1);
ch = toupper(getch());
switch(ch)
{
case 'L': { D_box(10,140,70,160,1);
            D_box(10,50,70,70,2);
            Load();
            D_box(10,50,70,70,1);
            break; }
case 'S': { D_box(10,140,70,160,1);
            D_box(10,80,70,100,2);
            Save();
            D_box(10,80,70,100,1);
            break; }
case 'R': { setfillstyle(SOLID_FILL,LIGHTGRAY);
            bar(159,50,501,65);
            D_box(80,45,630,470,2); //box scope

```

```

D_box(10,140,70,160,1); //box run
D_box(10,110,70,130,2); //box break
Run();
Still();
break; }

case 'E': { D_box(10,140,70,160,1);
            D_box(10,170,70,190,2);
            if(Exit()) ch = ESC;
            else { ch = 'N'; D_box(10,170,70,190,1); }
            break; }
}
}
}
void Still()
{ int x,y,xx,yy,px=0,py=0,chw;
  void far *imagex,*imagey;
  unsigned sizex,sizey;
  x = 610; //250-Maxv;
  y = 70; //Hz+110;
  xx = x;
  yy = y;
  D_box(80,45,630,470,1); //box scope
  D_box(10,110,70,130,1);
  D_box(10,140,70,160,2); //box run
  D_box(160,50,340,65,1); //box Hz
  D_box(360,50,500,65,1); //box Volt
  setcolor(RED);
  sprintf(tc,"Freq = %.2f KHz",Hzb*1333);

```

```

outtextbxy(165,55,te);
sprintf(te,"Volt = %.3f V",(Maxvb*0.033333));
outtextbxy(370,55,te);
setlinestyle(USERBIT_LINE,0X0FFFF,1);
sizey=imagesize(110,yy,610,yy);
imagey=malloc(sizey);
getimage(110,yy,610,yy,imagey);
sizex=imagesize(xx,70,xx,250);
imagex=malloc(sizex);
getimage(xx,70,xx,250,imagex);
do{
  if(kbhit())
  {
    ch = toupper(getch()); //bioskey(0);
    switch(ch)
    {
      // alt & arrow = speeds
      case UP: { if(y<71) ; else { yy=y; y--; py = 1; } break; }
      case DOWN: { if(y>249) ; else { yy=y; y++; py = 1; } break; }
      case LIFT: { if(x<111) ; else { xx = x; x--; px = 1; } break; }
      case RIGHT: { if(x>609) ; else { xx = x; x++; px = 1; } break; }
      case A_UP: { if(y<75) ; else { yy=y; y-=5; py = 1; } break; }
      case A_DOWN: { if(y>245) ; else { yy=y; y+=5; py = 1; } break; }
      case A_LIFT: { if(x<115) ; else { xx = x; x-=5; px = 1; } break; }
      case A_RIGHT: { if(x>605) ; else { xx = x; x+=5; px = 1; } break; }
      case 'C': { putimage(x,70,imagex,COPY_PUT);
                  putimage(110,y,imagey,COPY_PUT); }
      case 'L':
      case 'S':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'R':
case 'E': { ungetch(ch); chh = 'Q'; D_box(10,140,70,160,1); break; }
}
if(py==1) {
    putimage(110,yy,imagey,COPY_PUT);
    putimage(x,70,imagex,COPY_PUT);
    getimage(110,y,610,y,imagey);
    line(110,y,610,y);
    line(x,70,x,250);
    py = 2;
    Maxvb = 250 - y;
    setcolor(LIGHTGRAY);
    sprintf(te, " ");
    outtextxy(370,55,te);
    setcolor(RED);
    sprintf(te, "Volt = %.3f V", (Maxvb*0.033333));
    outtextxy(370,55,te);
}
if(px==1) {
    putimage(xx,70,imagex,COPY_PUT);
    putimage(110,y,imagey,COPY_PUT);
    getimage(x,70,x,250,imagex);
    line(110,y,610,y);
    line(x,70,x,250);
    px = 2;
    Hzb = x - 110;
    setcolor(LIGHTGRAY);
    sprintf(te, " ");

```

```

    outtextxy(165,55,te);

    setcolor(RED);

    sprintf(te,"Freq = %.2f KHz",Hzb*1333);

    outtextxy(165,55,te);
}
}
}while(chh!='Q');
}

void Load()
{ FILE *fn;
  int sx=100,sy=150,ex=555,ey=221;
  void far *image;
  unsigned size;
  char filename[13],string[13],che;
  size=gimagesize(sx,sy,ex,ey);
  image=malloc(size);
  getimage(sx,sy,ex,ey,image);
  setfillstyle(SOLID_FILL,DARKGRAY);
  bar(sx+11,sy+11,ex-5,ey-1);
  setfillstyle(SOLID_FILL,WHITE);
  rectangle(sx+1,sy+1,ex-15,ey-56);
  bar(sx+1,sy+1,ex-15,ey-56);
  setcolor(BLUE);
  outtextxy(sx+100,sy+4," ** Load Pic File (*.PCX) **");
  setfillstyle(SOLID_FILL,LIGHTGRAY);
  bar(sx+1,sy+16,ex-15,ey-11);

  setcolor(WHITE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rectangle(sx+1,sy+1,ex-15,ey-56);
rectangle(sx+1,sy+1,ex-15,ey-11);
setcolor(RED);
outtextxy(sx+16,sy+31,"Load File Name : ");
for(i=0;i<12;i++) {
    ch = toupper(getch());
    if(ch==0x0d) break;
    sprintf(fn,"%c",ch);
    outtextxy(250+(i*9),181,fn);
    filename[i] = ch;
}
fname = filename;
if(!strchr(fname,'.')) { /* is there an ext ? */
    strcpy(string,fname); /* copy filename to buffer */
    fname = string; /* FileName now points at buffer */
    strcat(fname,".pcx"); /* if not add .gif ext */
}
putimage(sx,sy,image,COPY_PUT);
che = DisplayPCXFile(fname,80,45,0);
free(image);
if(che);
else {
    setcolor(LIGHTGRAY);
    outtextxy(525,30,"");
    setcolor(BLUE);
    outtextxy(525,30,fname);
}
getimage(sx,sy,ex,ey,image);

```

```

setfillstyle(SOLID_FILL,DARKGRAY);
bar(sx+11,sy+11,ex-5,ey-1);
setfillstyle(SOLID_FILL,WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
bar(sx+1,sy+1,ex-15,ey-56);
setcolor(BLUE);
outtextxy(sx+100,sy+4," ** Load Pic File (*.PCX) **");
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(sx+1,sy+16,ex-15,ey-11);
setcolor(WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
rectangle(sx+1,sy+1,ex-15,ey-11);
setcolor(BLUE);
if(chc==0) printf(fn,"** Load File : %s OK Please and KEY**",fname);
else { setcolor(RED);
    printf(fn," ** Coule Not Load File : Please and KEY **"); }
outtextxy(sx+16,sy+31 ,fn);
getch();
putimage(sx,sy,image,COPY_PUT);
return 0;
}

```

```
void Save(void)
```

```
{ FILE *fn;
```

```
int sx=100,sy=150,ex=555,ey=221;
```

```
void far *image;
```

```
unsigned size;
```

```
char filename[13],string[13],che;
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

size=gimagesize(sx,sy,ex,ey);
image=malloc(size);
getimage(sx,sy,ex,ey,image);
setfillstyle(SOLID_FILL,DARKGRAY);
bar(sx+11,sy+11,ex-5,ey-1);
setfillstyle(SOLID_FILL,WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
bar(sx+1,sy+1,ex-15,ey-56);
setcolor(BLUE);
outtextxy(sx+100,sy+4," ** Save Pic File (*.PCX) **");
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(sx+1,sy+16,ex-15,ey-11);
setcolor(WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
rectangle(sx+1,sy+1,ex-15,ey-11);
setcolor(RED);
outtextxy(sx+16,sy+31,"Save File Name : ");
for(i=0;i<12;i++) {
    ch = toupper(getch());
    if(ch==0x0d) break;
    sprintf(fn,"%c",ch);
    outtextxy(250+(i*9),181,fn);
    filename[i] = ch;
}
fname = filename;
if (!strchr(fname,0x0d)) { /* is there an ext ? */
    strcpy(string,fname); /* copy filename to buffer */
    fname = string; /* FileName now points at buffer */
}

```

```

    strcat(fname, ".pcx");    /* if not add .gif ext */
}

putimage(sx,sy,image,COPY_PUT);
che = WritePCXFile(fname,80,45,630,470);
free(image);
if(che) ;
else {
    setcolor(LIGHTGRAY);
    outtextxy(525,30,"");
    setcolor(BLUE);
    outtextxy(525,30,fname);
}
setfillstyle(SOLID_FILL,DARKGRAY);
bar(sx+11,sy+11,ex-5,ey-1);
setfillstyle(SOLID_FILL,WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
bar(sx+1,sy+1,ex-15,ey-56);
setcolor(BLUE);
outtextxy(sx+100,sy+4," ** Save Pic File (*.PCX) **");
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(sx+1,sy+16,ex-15,ey-11);
setcolor(WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
rectangle(sx+1,sy+1,ex-15,ey-11);
setcolor(BLUE);
if(che==0) printf(fn,"** Save File : %s OK Please and KEY **",fname);
else { gsetcolor(RED);

printf(fn," ** Coule Not Write File :%d: Please and KEY **",che); }

```

```

outtextxy(sx+16,sy+31 ,fn);
getch();
putimage(sx,sy,image,COPY_PUT);
return 0;
}

```

```

void readdsk(char s)

```

```

{ int hd,ld,hdh,ldh,dl[2][3100],dh[2][3100];

```

```

do{

```

```

    outport(CONTROLPORT,0xff);

```

```

        ldh = (inp(DATAPORT)&0x07);

```

```

        hdh = (inp(DATAPORT)&0x0f8);

```

```

        dh[s][0] = (hdh|ldh);

```

```

}while(dh[s][0]!=0x7f);

```

```

do{

```

```

    outport(CONTROLPORT,0xff);

```

```

        ldh = (inp(DATAPORT)&0x07);

```

```

        hdh = (inp(DATAPORT)&0x0f8);

```

```

        dh[s][0] = (hdh|ldh);

```

```

}while(dh[s][0]==0x7f);

```

```

for(i=0;i<1500;i++){

```

```

    outport(CONTROLPORT,0xff);

```

```

    ld = (inp(DATAPORT)&0x07);

```

```

    hd = (inp(DATAPORT)&0x0f8);

```

```

    ldh = (inp(DATAPORT)&0x07);

```

```

    hdh = (inp(DATAPORT)&0x0f8);

```

```

    dl[s][i] = ((hd|ld)^0X80)>>1;

```

```

    dh[s][i] = ((hdh|ldh)^0x80)>>1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
outport(DATAPORT,0X00);
outport(CONTROLPORT,0x04);
dh[s][i] = 0;
k = 0;
Hz = 0;
Maxv = 0;
for(j=0;j<1500;j+=3) {

    if(dh[s][k]>Maxv) {
        Hz = k;
        Maxv = dh[s][k]; }
    k++;
    bh[s][k] = (dh[s][j+1]+dh[s][j+2]+dh[s][j+3])/3;
}
}

void Run()
{ int cc=500,rx=110,bas1=350,bas2=423;
  bl[0][0] = 0x40; bl[1][0] = 0x40;
  bh[0][0] = 0; bh[0][0] = 0;
  readsk(0);
  for(k=0;k<cc;k++) { for(j=0;j<100;j++);
    setcolor(LIGHTGRAY);
    line(k+rx,bas1,k+rx,bas1-181);
    setcolor(BLUE);
    line(k+rx,bas1-bh[0][k],k+rx+1,bas1-bh[0][k+1]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    readdsk(1);
    W_text();
    Vtable(110,170,610,350,20);
    Htable(110,170,610,350,30);
    setlinestyle(USERBIT_LINE,0X0FFFF,1);
    for(k=0;k<cc;k++) { for(j=0;j<100;j++) ;
        setcolor(LIGHTGRAY);
        line(k+rx,bas1-bh[0][k],k+rx+1,bas1-bh[0][k+1]);
        setcolor(BLUE);
        line(k+rx,bas1-bh[1][k],k+rx+1,bas1-bh[1][k+1]);
    }
    readdsk(0);
    for(k=0;k<cc;k++) { for(j=0;j<100;j++) ;
        setcolor(LIGHTGRAY);
        line(k+rx,bas1-bh[1][k],k+rx+1,bas1-bh[1][k+1]);
        setcolor(BLUE);
        line(k+rx,bas1-bh[0][k],k+rx+1,bas1-bh[0][k+1]);
    }
    if(kbhit()) ch = toupper(getch());
}while(ch!='B');
}

```

```
char Exit(void)
```

```
{ int sx=100,sy=150,ex=555,ey=221;
```

```
void far *image;
```

```
unsigned size;
```

```
size=imagesize(sx,sy,ex,ey);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

image=malloc(size);
getimage(sx,sy,ex,ey,image);
setfillstyle(SOLID_FILL,WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
bar(sx+1,sy+1,ex-15,ey-56);
setcolor(BLUE);
outtextxy(sx+145,sy+4," ** Exit Program **");
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(sx+1,sy+16,ex-15,ey-11);
setfillstyle(SOLID_FILL,DARKGRAY);
bar(sx+11,sy+62,ex-5,ey-1);
bar(sx+442,sy+11,ex-5,ey-11);
setcolor(WHITE);
rectangle(sx+1,sy+1,ex-15,ey-56);
rectangle(sx+1,sy+1,ex-15,ey-11);
setcolor(RED);
outtextxy(sx+16,sy+31,"Do you want to exit(Y/N) : ");
moveto(sx+61,sy+31);
ch = toupper(getche());
if(ch=='Y') return 1;
else {
    putimage(sx,sy,image,COPY_PUT);
    free(image);
    return 0;
}
}

```

-----PROGRAM SPECTRUM ANALYZER-----

; This is the source code for the DSK spectrum analyzer. ;

; Assemble using the DSK5A assembler and execute the DSK ;

; loader to load and run the application on your PC. ;

.mmregs

FFT_S .set 1024

FFT_S-1 .set 1023

FFT_S/2 .set 512

(FFT_S/2)-1 .set 511

CO .set 70h

TA .set 6 ; Auxin ---+ +--- Loopback

RA .set 6 ; Synch --+ | | +- BP Filter

TAp .set 1 ; | | | |

RAp .set 1 ; +-----+-----+

TB .set 18 ;|00 00 G1 G0 | SY AX LB BP|

RB .set 18 ; +-----+-----+

.ds 0f00h

TEMP .word 0 ;location of TEMPorary storage

ACCU_lo .word 0

ACCU_hi .word 0

STAT1 .word 0 ;STAT1 storage

* TMS32C05X INITIALIZATION *

* This routine initializes the C5x registers, internal RAM and *

* external RAM from xxxx to FFFF *

```

        .ps 0a00h

        .entry

start:   setc INTM           ; Disable interrupts

        ldp #0             ; Set data page pointer

        splk #830h,PMST    ; 9K on-chip RAM as Data, No ROM

        lacl #0            ; Set Wait State Control Register

        samm CWSR         ; for 0 waits in pgm & data memory

        samm PDWSR

```

*** initialize and reset serial port**

```

        splk #20h,TCR

        splk #1,PRD

        mar *,AR0

        lacl #08h          ; set FSM bit for FSX/FSR per frame

        samm spc           ; Configure for 16 bit mode with

        lacl #0C8h        ; external CLKX, reset tx and rx

        samm spc

        lamm drr           ; clear first int

        lacc #0080h

        sach dxr           ; clear first int

        saci GREG          ; Pulse AIC reset by setting it low

        lar AR0,#0FFFFFFh

        rpt #10000         ; and taking it high after 1000 cycles

        lacc *,0,AR0       ; (.5ms at 50ns)

        sach GREG

        setc SXM

        setc OVM

```

```

;-----
lar AR7,#0 ; Buffer initially filled
lacl #030h ; RINT
samm IMR
ldp #TEMP
FFT: lar AR0,#FFT_S/2
mar *,AR0 ; start FFT with AR0=FFTSize
new_stg: lar AR1,#_D_base ; AR1 is the TOP BFLY address
lar AR2,#_D_base ; AR2 is the BOT BFLY address
lar AR3,#_T_base+1 ; AR3 is the TWiddle pointer
lar AR4,#FFT_S/2 ; AR4 counts DFT blocks
b n_DFT2,*,AR1
DFT: mar *BR0+,AR5 ; complete circular buffer for TW's
lar AR5,#1 ; set up DFT loop with *BR0+/BANZ
mar *BR0+,AR1 ; using 1 cuts *BR0+ loop in half!
;-----
; AR1=Top AR2=Bottom AR3=TWiddle
;-----
BFLY: lacc *,14,AR2 ; (imag1+imag2)/4
add *,14,AR1
sach *+,1,AR2 ; store TOP imag
sub *,15 ; (imag1-imag2)/2
sach *+,1,AR1 ; store BOT imag
lacc *,14,AR2 ; (real1+real2)/4
add *,14,AR1
sach *+,1,AR2 ; store TOP real
sub *,15 ; (real1-real2)/2
sach *,1,AR5 ; store BOT real

```

```

    banz OK,*BR0+,AR3 ; If at DFT end quit early
    mar  *+,AR2        ; clean up TW base (xxx0000+1)
    mar  *+            ; modify BOTom DATA pointer
    mar  *0+
    mar  *0+,AR1
n_DFT2: mar  *0+        ; modify the TOP pointer
    mar  *0+,AR4
    banz DFT,*0-,AR3 ; dec DFT block count AR4 by OFFSET
    mar  *,AR0
    mar  *BR0+
    banz new_stg,*    ; if OFFSET was 1, now cleared
b      endFFT
;-----
OK    lt  *-,AR2        ; TREG=TWR  *NOTE* Twiddles are Q15
    mpy *-,            ; PREG=REAL*TWR
    ltp  *+,AR3        ; TREG=IMAG  ACCU=REAL*TWR
    mpy *               ; PREG=IMAG*TWI  AR2=R AR3=I
    lts  *+,AR2        ; TREG=TWI  ACCU=REAL*TWR-IMAG*TWI
    mpy *               ; PREG=REAL*TWI
    sach *-,1
    ltp  *,AR3         ; TREG=IMAG  ACCU=REAL*TWI
    mpy *BR0+,AR2     ; PREG=IMAG*TWR
    apac ; ACCU=IMAG*TWR+REAL*TWI
    sach *+,1
    b    BFLY,*+,AR1 ;
;-----
endFFT: mar  *,AR2        ; Transform REAL & IMAG to log magnitude

```

```

    lar  AR2,#_D_base ; AR3=FFT data pointer

```

```

lar   AR3,#FFT_S-1 ; AR5=FFT loop counter
more_MAG  sqra  *+          ; PREG=IMAG^2
ltp   *          ; TREG=REAL ACCU=IMAG^2
mpy   *,AR1      ; PREG=REAL^2
apac          ; ACCU=REAL^2+IMAG^2
lar   AR1,#31    ; NORMalize the accumulator
rpt   #30        ; use for other types of conversion
norm  *-
nop
nop
mar   *,AR2
bcnd  sig_NZ,NEQ ; if zero must return 0
lar   AR1,#0
sig_NZ  sach  *,2          ; clear explicit 1.0 from mantissa
lacl  *          ; load into accumulator and
and   #0FF80h   ; clear LSB's for AIC
sar   AR1,*     ;append the exponent (AR5)
add   *,16
xor   #020h,15 ;change to 2's compliment
rpt   #3        ;jam result to top of ACCU
sfl
sach  *+,7,AR3
banz  more_MAG,*-,AR2 ;keep going until all done
;-----

```

```

BITREV:  lar   AR0,#FFT_S ;Now perform Output bit reversal
lar   AR1,#_D_base ;by moving the magnitude, which
lar   AR2,#_D_base+1 ;is in the REAL slots, into the
lar   AR3,#FFT_S-1 ;IMAG slots of the FFT data array

```

```

more_BR:   lacc  *+           ;load the magnitude
           mar  *+,AR1
           sac1 *BR0+,0,AR3 ;move it to an open IMAG slot
           banz more_BR,*-,AR2 ;more data to move?
           ;-----
           lar  AR3,#IN_buf

MOVE_IO:   mar  *,AR6           ; wait until buffer is full
           lacc *,8,ar6
           sach *,0,AR6
           OUT  *,#0FFF0h
           RPT  #0420h
           NOP
           mar  *,AR3           ; point to data input buffer
           lacc *+,0,ar6
           and  #00ffh
           sub  #007fh,0,AR6
           sac1 *+,0,AR7       ;store data from input buffer

           banz MOVE_IO,*-,AR2 ;(AR7 is decremented by ISR)
           lar  AR3,#_D_base   ;AR3=FFT data pointer
           lar  AR4,#_B_base   ;AR4=BUFF data pointer
           lar  AR5,#(FFT_S/2)-1 ;AR5=FFT loop counter
           lar  AR6,#_B_base   ;AR6=ISR BUFF data pointer
           SPLK #07Fh,CO
           OUT  CO,0fff0h

```

```
mar  *,AR7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lar   AR7,#IN_buf
rpt   #1024
in    *+,0fff1h
mar   *,AR2

lar   AR2,#_T_base+1 ; AR2=WIN data pointer
lar   AR7,#FFT_S-1   ; AR7=ISR BUFF loop counter
;-----
lar   AR0,#FFT_S/2   ; Use twiddle table for raised
more_IO: lacc #04000h,1   ; cosine window
add   *BR0+,0,AR4
sfr
sac1  TEMP
lt    TEMP           ;TREG=WIN
mpy   *,AR3         ;PREG=IN*WIN
lacl  *,AR4         ;ACCU=magnitude (put in Buffer)
sac1  *+,0,AR3
sach  *+            ;IMAG=0
pac
sach  *+,1,AR5      ;REAL=IN (windowed buffer)
clrc  INTM          ;1st BUFF posn clr so enable INT's
banz  more_IO,*-,AR2

lar   AR5,#(FFT_S/2)-1 ;AR5=FFT loop counter
more_IO2 lacc #04000h,1 ; cosine window
add   *BR0-,0,AR4
sfr
sac1  TEMP
lt    TEMP           ;TREG=IN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mpy  *,AR3      ; PREG=IN*WIN
lacl  *,AR4      ; ACCU=magnitude (put in Buffer)
sac1  *+,0,AR3
sach  *+         ; IMAG=0
pac
sach  *+,1,AR5   ; REAL=IN (windowed buffer)
banz  more_IO2,*-,AR2
b      FFT

```

```

.ds 01000h
.listoff
.include "dsk_twid.asm" ; NOTE: Internal RAM block B2 is used
.liston

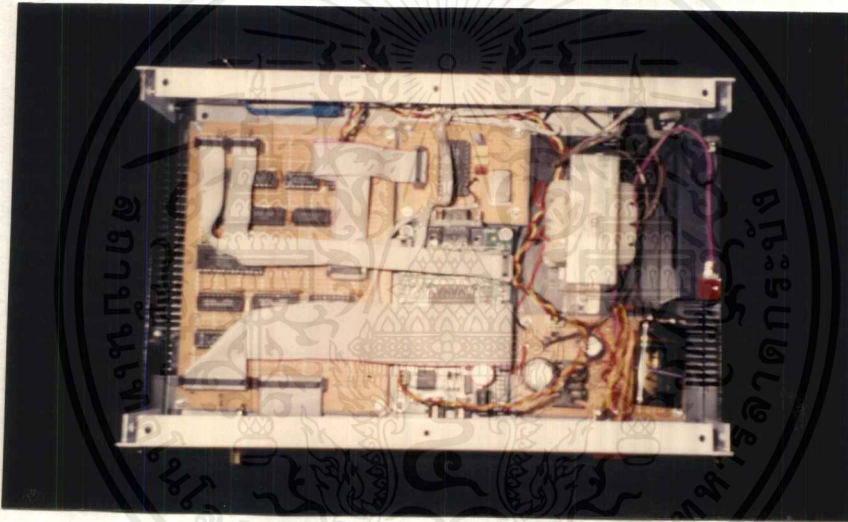
```



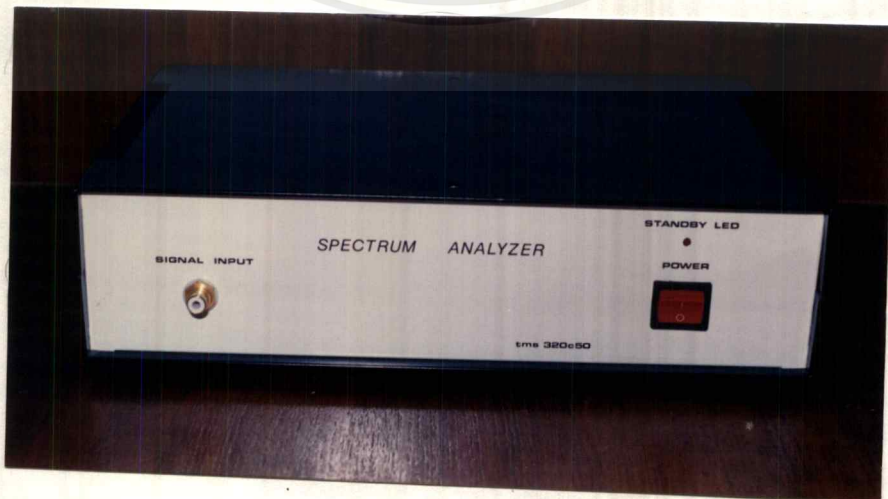
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวิเคราะห์แถบความถี่

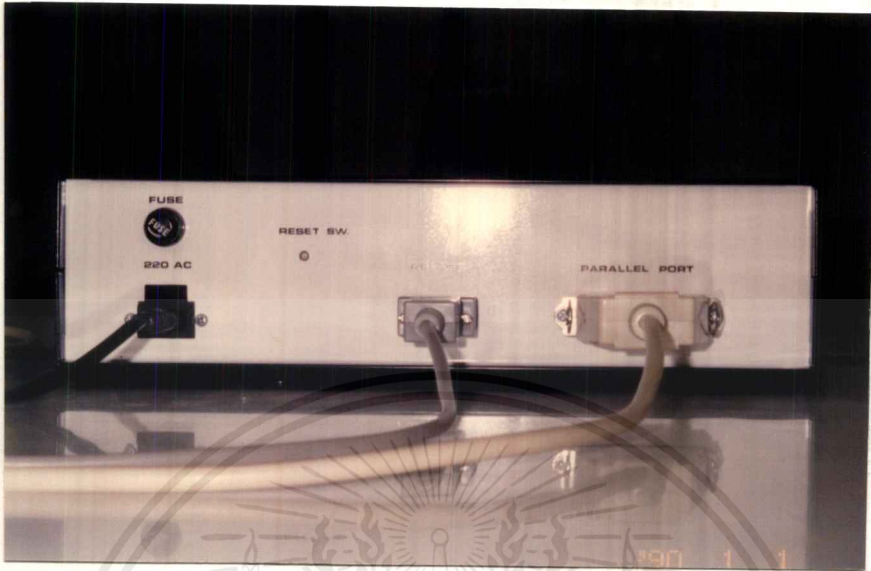
เครื่องวิเคราะห์แถบความถี่นี้ ใช้ในการวิเคราะห์ความถี่ ที่เป็นองค์ประกอบสัญญาณ อินพุต ซึ่งเป็นสัญญาณแอนะล็อก ซึ่งสามารถวิเคราะห์แถบความถี่ได้ในย่าน 1.33 kHz - 2 MHz. โดยแสดงผลออกทางหน้าจอคอมพิวเตอร์ ซึ่งสื่อสารกันผ่านทางพอร์ตขนาน ทำให้ได้การทำงานที่เป็นเวลาจริง และเที่ยงตรง อีกทั้งยังสะดวกต่อการอ่านค่าของแรงดัน และความถี่ด้วย เนื่องจากแสดงผลได้ทั้งรูปภาพสเปกตรัม และตัวเลข โครงสร้าง และส่วนประกอบต่างๆ ของเครื่อง



รูปที่ 1 โครงสร้างภายนอกของเครื่อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2 ส่วนประกอบด้านหน้าของเครื่อง ญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 ส่วนประกอบด้านหลังของเครื่อง

ส่วนประกอบต่างๆ

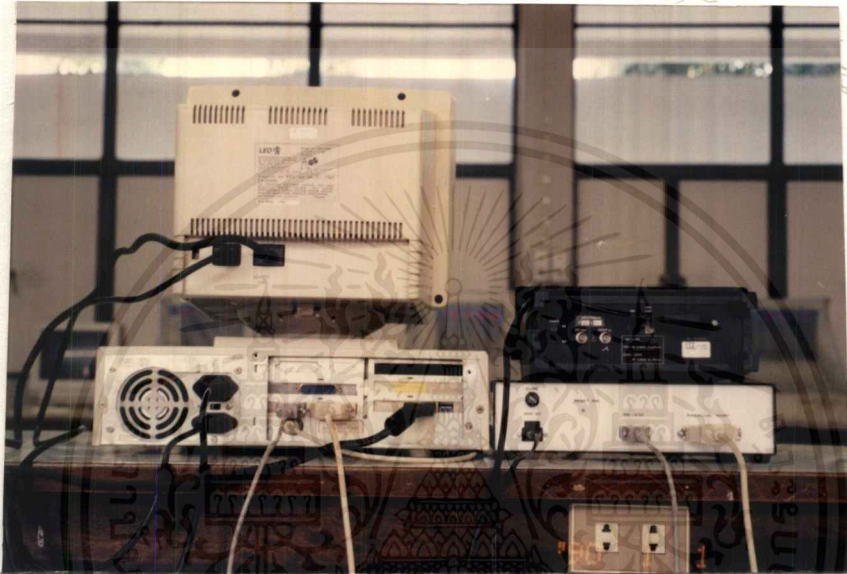
1. สวิตช์ POWER
2. LED แสดงสถานะการทำงานของเครื่อง
3. ขั้วต่อสัญญาณอินพุต
4. ขั้วต่อ DB9 สำหรับเชื่อมต่อกับคอมพิวเตอร์ทางพอร์ต RS 232
5. ขั้วต่อพอร์ตขนาน สำหรับเชื่อมต่อทางพอร์ตพริ้นเตอร์
6. สวิตช์รีเซตเครื่อง
7. พิวส์
8. ขั้วต่อสายไฟ AC 220 V

การใช้งานเครื่องวิเคราะห์แถบความถี่

การเชื่อมต่อเครื่องวิเคราะห์แถบความถี่เข้ากับเครื่องคอมพิวเตอร์ แสดงดังรูปที่ 4

การใช้งานเครื่องวิเคราะห์แถบความถี่นี้จะใช้งานร่วมกับคอมพิวเตอร์พีซีทั่วไป โดยต่อตัวกันผ่านทางพอร์ตขนาน และพอร์ตอนุกรม ดังนั้นจึงต้องทำการเชื่อมต่อ อุปกรณ์ทั้ง 2 เข้าด้วยกัน โดยใช้สายเชื่อมต่อทางพอร์ตขนาน หรือสายพริ้นเตอร์ ทั่วๆ ไป ในการเชื่อมต่อระหว่าง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตขนานของทั้ง 2 เครื่อง จากนั้นทำการเชื่อมต่อพอร์ตอนุกรม (พอร์ต RS232) ของเครื่อง ทั้ง 2 เข้าด้วยกัน โดยใช้สายเชื่อมต่อพอร์ต RS232 ที่ติดมากับเครื่อง ต่อสายไฟ AC 220 V เข้ากับเครื่องวิเคราะห์แถบความถี่ และต่อสายสัญญาณที่ต้องการจะวิเคราะห์เข้าทางขั้วต่ออินพุตของเครื่อง ทำการตรวจเช็คการเชื่อมต่อทุกจุดให้ถูกต้อง และเรียบร้อย



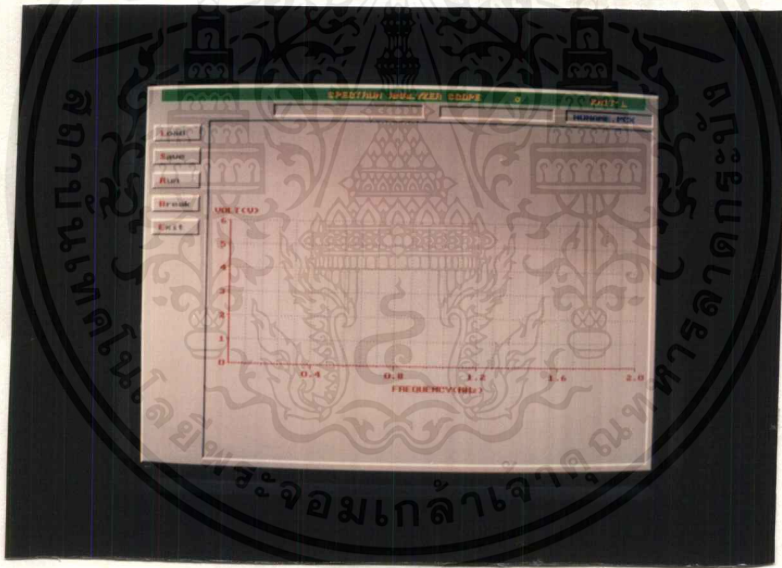
รูปที่ 4 การต่ออุปกรณ์วิเคราะห์แถบความถี่



รูปที่ 5 รูปแสดงการป้อนสัญญาณอินพุตที่ต้องการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำการเชื่อมต่อพอร์ตต่างๆ ของเครื่องวิเคราะห์แถบความถี่นี้เข้ากับเครื่องคอมพิวเตอร์เป็นที่เรียบร้อยแล้ว ขั้นตอนต่อไป จะเป็นการใช้งาน โดยการทำการเปิดสวิตช์ POWER ของเครื่อง ในตอนนี้ LED แสดงสถานะของเครื่องจะติดขึ้น แสดงถึงการพร้อมทำงานของเครื่องวิเคราะห์แถบความถี่ จากนั้นทำการรันโปรแกรมวิเคราะห์แถบความถี่ บนเครื่องคอมพิวเตอร์ที่เซตระบบไว้เป็นที่เรียบร้อยแล้ว โดยพิมพ์ SPEC บนเครื่องคอมพิวเตอร์ เครื่องจะทำการรันโปรแกรมแสดงผลหน้าจอ และจะส่งข้อมูลโปรแกรม วิเคราะห์แถบความถี่ให้กับเครื่องวิเคราะห์แถบความถี่ กดสวิตช์ รีเซต 1 ครั้ง เพื่อเซตให้ เครื่องวิเคราะห์แถบความถี่นี้พร้อมทำงานจริงๆ หลังจากนั้นที่หน้าจอคอมพิวเตอร์ จะทำการโหลดข้อมูลไปยังเครื่องวิเคราะห์แถบความถี่ และเข้าสู่หน้าจอการแสดงผล



รูปที่ 6 หน้าจอเริ่มต้น

จากหน้าจอคอมพิวเตอร์ที่แสดงให้เห็น จะเห็นว่า มีส่วนประกอบต่างๆ คือ

1. ส่วนของ TITLE
2. ส่วนของการแสดงขนาดของแรงดัน และความถี่
3. ส่วนของชื่อไฟล์ นามสกุล .PCX
4. ปุ่มฟังก์ชันต่างๆ ได้แก่

- ฟังก์ชัน LOAD

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน SAVE
- ฟังก์ชัน RUN
- ฟังก์ชัน BREAK
- ฟังก์ชัน EXIT

5. ส่วนแสดงผล

- ส่วนแสดงผลสเปคตรัม

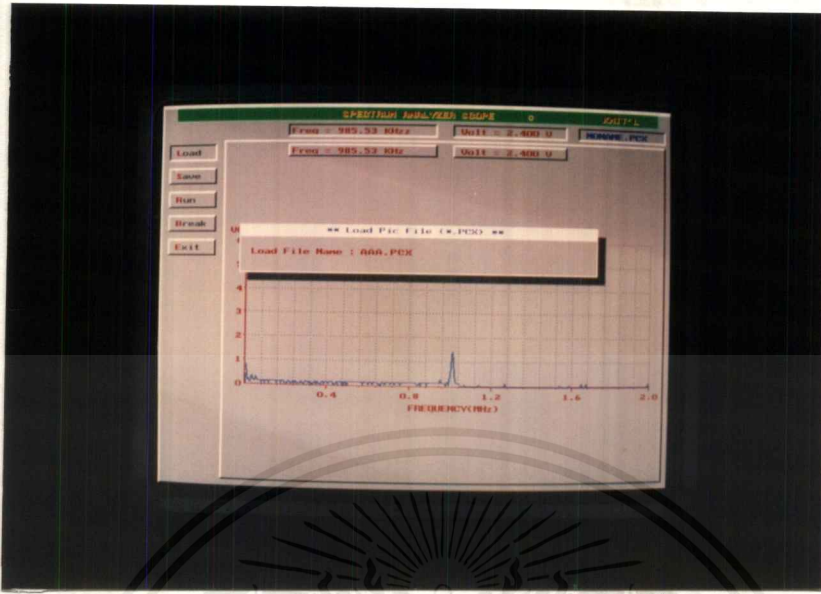
ขั้นตอนการใช้งาน ผู้ใช้สามารถเลือกใช้ฟังก์ชันการทำงานต่างๆ ของเครื่องได้โดยการกดปุ่มตัวอักษร ตัวแรกของชื่อปุ่มต่างๆ ได้แก่ L, S, R, B และ E ซึ่งฟังก์ชันการทำงานต่างๆ ของเครื่องแต่ละฟังก์ชัน มีรายละเอียดการใช้งานดังนี้

1. ฟังก์ชัน LOAD

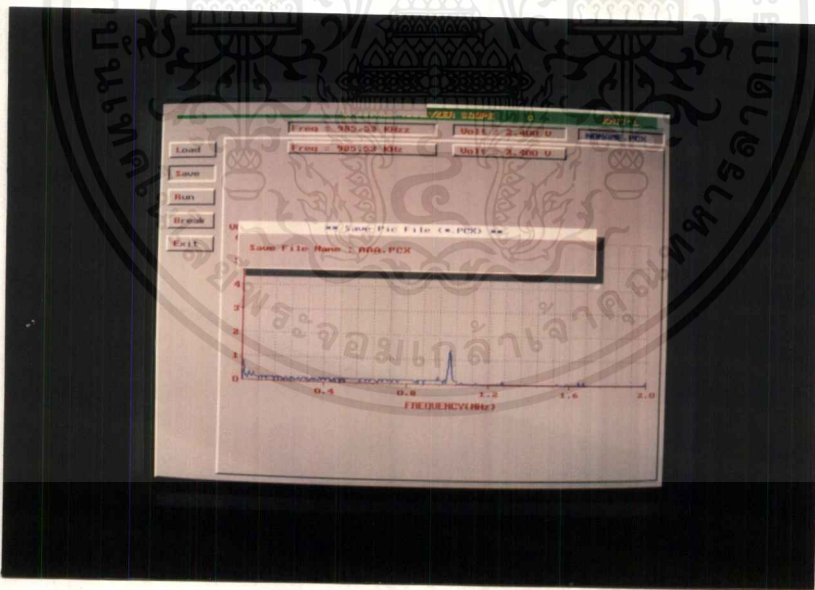
เป็นฟังก์ชันที่ใช้ในการเรียกไฟล์ของภาพการวิเคราะห์สัญญาณที่ได้ทำการบันทึกไว้ด้วย ฟังก์ชัน SAVE กลับมาแสดงที่หน้าจออีกครั้ง เพื่อประโยชน์ในการวิเคราะห์ หรือเปรียบเทียบสัญญาณต่างๆ การใช้งานฟังก์ชัน LOAD ทำได้โดยกดคีย์ "L" เพื่อทำการ LOAD ไฟล์ในฟังก์ชันนี้จะให้ผู้ใช้ป้อนชื่อไฟล์ที่ต้องการเรียกกลับมาแสดงผล ซึ่งต้องมีนามสกุล .PCX คอมพิวเตอร์จะทำการดึงภาพที่ต้องการขึ้นมา แสดงผลที่หน้าจอ พร้อมทั้งขนาดแรงดัน และความถี่ของสัญญาณในภาพด้วย

2. ฟังก์ชัน SAVE

ฟังก์ชันนี้จะใช้ในการบันทึกภาพของการวิเคราะห์สัญญาณ ที่ต้องการเก็บไว้ เพื่อการศึกษาในเวลาอื่น โดยทำการกดคีย์ "S" เพื่อทำการบันทึกภาพสัญญาณ ซึ่งการบันทึกภาพนี้ จะต้องทำการแสดงผลของการวิเคราะห์สัญญาณก่อน แล้วทำการหยุดภาพไว้ด้วยฟังก์ชัน Break แล้วจึงจะทำการบันทึกภาพที่ต้องการได้ โดยเครื่องจะให้ผู้ใช้ทำการป้อนชื่อของไฟล์ภาพที่จะบันทึก โดยจะบันทึกเป็นไฟล์นามสกุล .PCX เครื่องจะทำการบันทึกภาพหน้าจอในขณะที่ทำการ Break นั้น ทั้งหมด รวมทั้งขนาดของแรงดัน และความถี่ ขณะนั้นด้วย



รูปที่ 7 เมื่อใช้ฟังก์ชัน LOAD



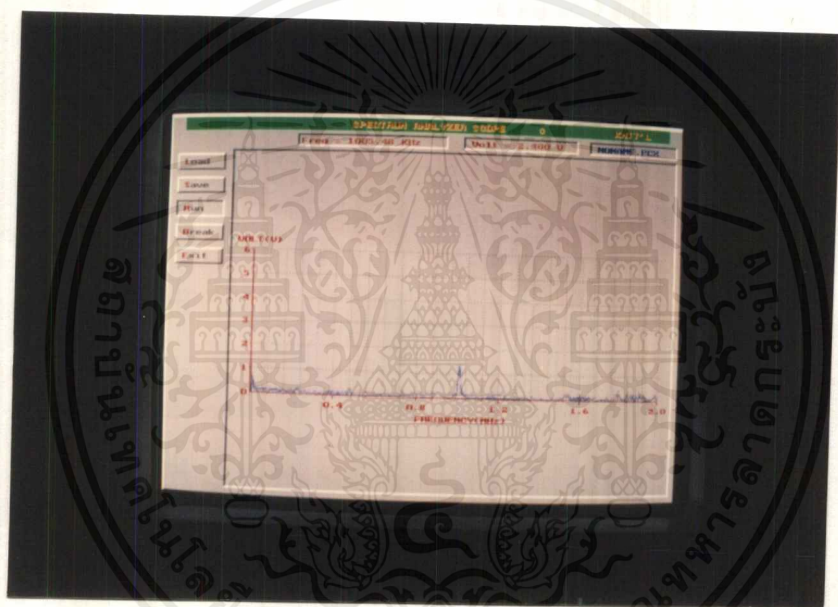
รูปที่ 8 เมื่อใช้ฟังก์ชัน SAVE

3. ฟังก์ชัน RUN

เป็นฟังก์ชัน ที่ใช้ในการแสดงผลการวิเคราะห์สัญญาณที่ป้อนเข้ามาทางอินพุต โดยการทำการกดคีย์ “R” เครื่องจะทำการวิเคราะห์สัญญาณอินพุตที่ได้รับเข้ามา ทางข้อต่อสัญญาณ และเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่อผู้ดูแลให้พิมพ์เอกสารนี้เป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแสดงผลทางหน้าจอคอมพิวเตอร์ โดยจะแสดงผลการวิเคราะห์เป็นกราฟสเปกตรัมของสัญญาณ ในแกนชุดบน โดยแกน Y จะเป็นแกนของขนาดแรงดัน ของสัญญาณอินพุต มีหน่วยเป็นโวลต์ แกน X จะเป็นแกนของเวลา มีหน่วยเป็นเมกะเฮิรต์

ในส่วนบนของจอภาพจะเป็นการแสดงขนาดของแรงดัน และความถี่ของสัญญาณที่ได้ทำการวิเคราะห์ออกมาเป็นตัวเลข ซึ่งสะดวกต่อการอ่านค่า และเที่ยงตรง



รูปที่ 9 เมื่อใช้ฟังก์ชัน RUN

4. ฟังก์ชัน BREAK

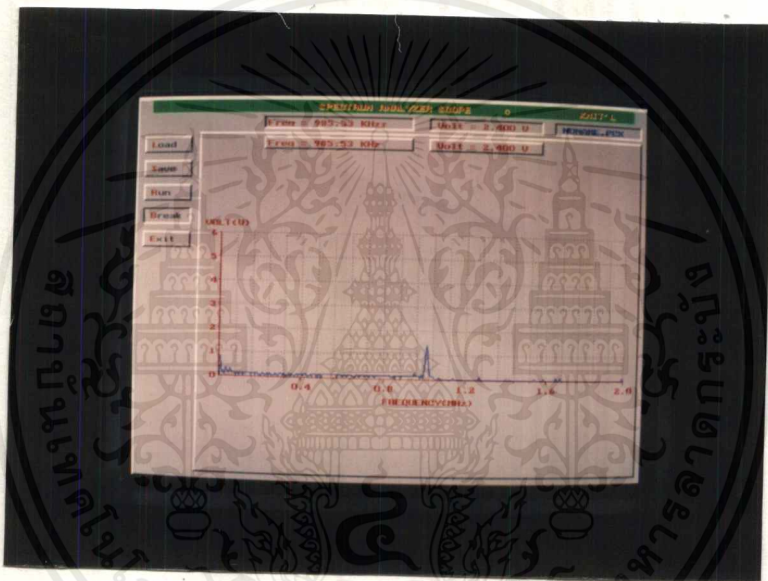
ฟังก์ชันการ Break จะใช้ในการหยุดภาพ การแสดงผลการวิเคราะห์สัญญาณที่ปรากฏอยู่บนหน้าจอ ให้อยู่นี้ เพื่ออ่านค่าข้อมูลต่างๆ ที่ได้จากการวิเคราะห์สัญญาณ หรือใช้เปรียบเทียบกับสัญญาณอินพุตที่เข้ามา ซึ่งแสดงอยู่ด้านล่างของจอภาพ

ฟังก์ชัน Break นี้เป็นฟังก์ชันสำคัญที่ต้องเรียกใช้ ร่วมกับการใช้งานฟังก์ชันอื่นๆ เช่น ฟังก์ชันการบันทึกภาพ หรือฟังก์ชันการจบการทำงาน การใช้งานฟังก์ชัน Break นี้ทำได้โดยการกดคีย์ "B"

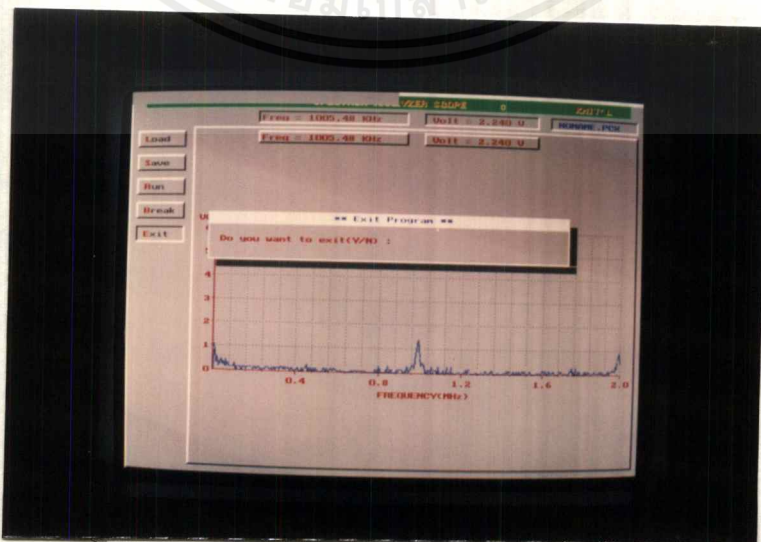
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ฟังก์ชัน EXIT

ฟังก์ชัน EXIT คือฟังก์ชันที่ใช้ในการจบการทำงาน และออกไปสู่ระบบของคอมพิวเตอร์ การใช้งานฟังก์ชันนี้ทำได้โดยการเรียกใช้ฟังก์ชัน Break ก่อน 1 ครั้ง (กดคีย์ "B") จากนั้นให้เรียกใช้ฟังก์ชัน Exit โดยการกดคีย์ "E" เครื่องจะทำการสอบถามความต้องการอีก 1 ครั้ง หากต้องการออกจากการทำงานให้ตอบ โดยการ กดคีย์ "Y" เครื่องทำการออกจากรโปรแกรมไปสู่ระบบของเครื่อง หากต้องการจะทำการวิเคราะห์สัญญาณต่อ ให้ทำการกดคีย์ "N" เพื่อกลับเข้าสู่การวิเคราะห์ความถี่ของสัญญาณอีกครั้ง



รูปที่ 10 เมื่อใช้ฟังก์ชัน BREAK



รูปที่ 11 เมื่อใช้ฟังก์ชัน EXIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ชิน ภู่วรรณ , “ การประมวลผลสัญญาณดิจิทัล ” วารสารเคมีคอนคัคเตอร์
อิเล็กทรอนิกส์,ฉบับที่ 77 (เมษายน 2530) : หน้า 156-172

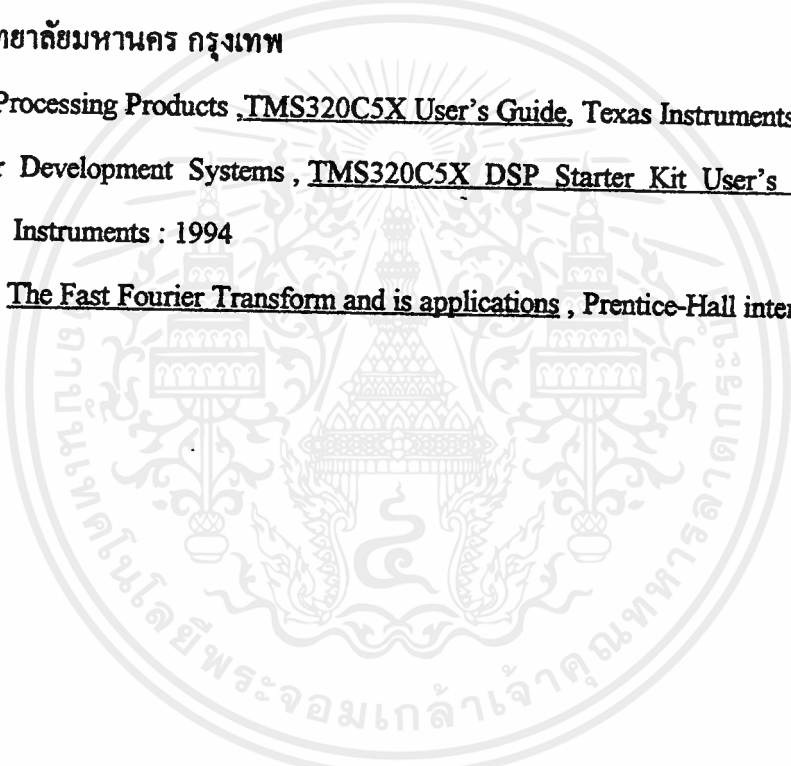
โอภาส ศิริครษิตถาวร , “บอร์ดDSP Starter Kit”วารสารเคมีคอนคัคเตอร์อิเล็กทรอนิกส์,
ฉบับที่ 148 (พฤษภาคม 2538) : หน้า 49-55

ธันวา ศรีประโมง การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม พิมพ์ครั้งที่ 4 : โครงการตำรา
มหาวิทยาลัยมหานคร กรุงเทพฯ

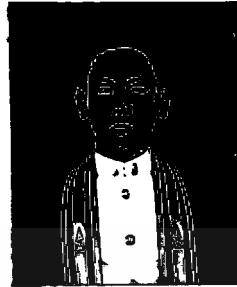
Digital Signal Processing Products ,TMS320C5X User's Guide, Texas Instruments : 1993

Microprocessor Development Systems ,TMS320C5X DSP Starter Kit User's Guide,
Texas Instruments : 1994

O.E.Brigham., The Fast Fourier Transform and is applications , Prentice-Hall international Inc.,
1988



ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาโท

นายจักรชัย วงศ์ไคนวงศ์

สถานที่เกิด

สมุทรปราการ

ภูมิลำเนาเดิม

216 ม.3 ต. โศกปีบ อ. ศรีมโหสถ

โทรศัพท์

จ. ปราจีนบุรี 25190

ที่อยู่ปัจจุบัน

ชลบุรี

โทรศัพท์

-

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนนพคุณวิทยา

มัธยมศึกษาตอนต้น

โรงเรียนมัธยมวัดใหม่กรงทอง

ประกาศนียบัตรวิชาชีพ (ปวช.)

วิทยาลัยเทคนิคชลบุรี

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

วิทยาลัยเทคนิคสัตหีบ

ปริญญาตรี

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับ

-

กิตติคุณ

-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร

สถานที่เกิด

ภูมิลำเนาเดิม

โทรศัพท์

ที่อยู่ปัจจุบัน

ประวัติการศึกษา

ประถมศึกษา

มัธยมศึกษาตอนต้น

ประกาศนียบัตรวิชาชีพ (ปวช.)

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

ปริญญาตรี

ผลงานที่ได้รับ

คตินิพนธ์

นายชุมพล จิตรกุล

ระยอง

5 ม.3 ต.เนินพระ อ.เมือง

จ.ระยอง 21000

(038) 807396

ระยอง

โรงเรียนอนุบาลระยอง

โรงเรียนระยองวิทยาคม

วิทยาลัยเทคนิคระยอง

วิทยาลัยเทคนิคสัตหีบ

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

-

-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

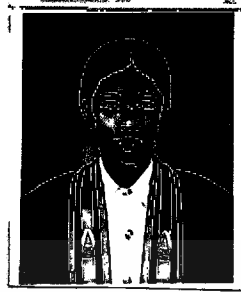
ประวัติผู้แต่ง



ชื่อผู้ทำปริยญาณิพนธ์	นายชอุง ขวัญอันอินทร์
สถานที่เกิด	พิจิตร
ภูมิลำเนาเดิม	พิจิตร
โทรศัพท์	-
ที่อยู่ปัจจุบัน	พิจิตร
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเทศบาลบ้านปากทาง
มัธยมศึกษาตอนต้น	โรงเรียนพิจิตรพิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคพิจิตร
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคพิจิตร
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
คตินพจน์	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นางสาวอรพิน สัตตบุตร
สถานที่เกิด	สุพรรณบุรี
ภูมิลำเนาเดิม	สุพรรณบุรี
โทรศัพท์	(044) 857052
ที่อยู่ปัจจุบัน	43 ม. 1 ต. vencebaek อ.เทพสถิต จ. ชัยภูมิ 36230
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนชุมชนบ้านvencebaek
มัธยมศึกษาตอนต้น	โรงเรียนชัยบาดาลวิทยา
มัธยมศึกษาตอนปลาย	โรงเรียนบุญเหลือวิทยานุสรณ์
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขต ภาคตะวันออกเฉียงเหนือ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
คติพจน์	ฝันให้ไกล แล้วไปให้ถึง