ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดทดลองการประมวลสัญญาณเชิงเลขด้วย DSK TMS320C31

Demonstration of Digital Signal Processing by DSK TMS320C31

ชื่อนักศึกษา

1. นายจักรกฤษณ์	หมายมั่น	รหัสประจำตัว	44035395
2. นายณัฐวัฒน์	ทองนิรันดร์	รหัสประจำตัว	44035402
3. นายพิทยา	ถวิลไพร	รหัสประจำตัว	44035411

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา อาจารย์สุระชัย พิมพ์สาลี

อาจารย์ที่ปรึกษาร่วม อาจารย์อำพล ทองระอา

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์สุระชัย	พิมพ์สาลี	.....
2. อาจารย์อำพล	ทองระอา	.....
3. อาจารย์ปิยะ	ศุภวราสุวัฒน์	.....
4. อาจารย์วรวิทย์	สมหา	.....
5. อาจารย์อมรชัย	ชัยชนะ	.....

วัน/เดือน/ปีที่สอบ วันศุกร์ที่ 15 พฤศจิกายน พ.ศ. 2545 เวลา 11.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 20 เดือน พ.ย. พ.ศ. 2545

<BT4503022>

ชุดทดลองการประมวลสัญญาณเชิงเลขด้วย DSK TMS320C31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31

DEMONSTRATION OF DIGITAL SIGNAL PROCESSING BY DSK TMS320C31



TO48320

นายจักรกฤษณ์ หมายมั่น
นายณัฐวัฒน์ ทองนิรันดร์
นายพิทยา ถวิลไพร

เลขหมู่.....
เลขทะเบียน 48320
วัน, เดือน, ปี 10 ต.ค. 2546

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31

DEMONSTRATION OF DIGITAL SIGNAL PROCESSING BY DSK TMS320C31

วัตถุประสงค์

1) เพื่อศึกษาระบบการทำงานและหลักการเชื่อมต่อของบอร์ด DSK TMS320C31
2) เพื่อออกแบบชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31
3) เพื่อสร้างชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วยบอร์ด DSK TMS320C31
4) เพื่อทดสอบชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วยบอร์ด DSK TMS320C31
5) เพื่อนำบอร์ด DSK TMS320C31 ที่เพิ่มประสิทธิภาพไปใช้ประกอบการเรียนรู้กับผู้ที่สนใจ

ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้ทักษะจากการใช้บอร์ด DSK TMS320C31
2) ได้รับความรู้จากการออกแบบชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31
3) ได้ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31
4) ได้ผลการทดสอบของ ชุดทดลองการประมวลสัญญาณเชิงเลขด้วย DSK TMS320C31
5) ได้นำชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 ไปใช้ประกอบการเรียนรู้กับผู้ที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31	
นักศึกษา	นายจักรกฤษณ์	หมายมั่น
	นายณัฐวัฒน์	ทองนิรันดร์
	นายพิทยา	ถวิลไพร
อาจารย์ที่ปรึกษา	อาจารย์สุระชัย	พิมพ์สาลี
อาจารย์ที่ปรึกษาร่วม	อาจารย์อำพล	ทองระอา
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรรม	
ปีการศึกษา	2545	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอเกี่ยวกับ ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 และมีการเพิ่มประสิทธิภาพของบอร์ด DSK TMS320C31 โดยมีการเพิ่มหน่วยความจำ ตัวแปลงสัญญาณจากแอนะลอกเป็นดิจิตอลและตัวแปลงสัญญาณดิจิตอลเป็นแอนะลอก ในชุดทดลองมีใบงานเกี่ยวกับการทดลองการประมวลผลสัญญาณเชิงเลขอยู่ 6 ใบงาน ประกอบด้วย การใช้งานบอร์ดเบื้องต้น, การใช้ชุดคำสั่งแบบง่าย, การทดสอบการเชื่อมต่อหน่วยความจำภายนอก, การสร้างรูปคลื่นไซน์ (sine wave), Fast Fourier Transform (FFT) จึงเหมาะกับผู้ที่สนใจเกี่ยวกับการประมวลผลสัญญาณเชิงเลขเป็นอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	Demonstration of Digital Signal Processing by DSKTMS320C31	
Students	Mr.Chakkrit	Maiman
	Mr.Nattawat	Thongnirun
	Mr.Pittaya	Thawinprai
Advisor	Mr.Surachai	Pimsalee
Co - Advisor	Mr.Amphon	Thongra-ar
Education level	Bachelor of Science in Industrial Education	
Program in	Industrial Instrument Technology	
Academic year	2002	

ABSTRACT

This thesis presents about Demostration of Digital Signal Processing TMS320C31 and increase of efficiency of DSK TMS320C31 board by the memory unit increase,Analog to Digital and Digital to Analog. There are six reports of Digital Signal Processing experiments which consist of first using DSK TMS320C31 board experiment, using easy instruction set experiment , memory unit test experiment ,sine wave inventing experiment ,fast fourier tranform the efficicent increasing board and FIR FILLTER. This thesis is suitable for the people who are interested in Digital Signal Processing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ลุล่วงไปด้วยดี เนื่องจากความร่วมมือของสมาชิกภายในกลุ่มและการสนับสนุนทางด้านอุปกรณ์จากทางภาควิชาครุศาสตร์วิศวกรรม ขอขอบพระคุณ อาจารย์อำพล ทองระอา และคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความรู้ คำแนะนำ และแนวทางการแก้ปัญหาต่างๆให้ลุล่วงไปด้วยดี ขอขอบคุณ คุณอัปสร รังสรรค์ ที่ให้การจัดทำปริญญานิพนธ์ลุล่วงไปด้วยดี สุดท้ายนี้สิ่งที่ระลึกอยู่เสมอคือ บิดา มารดา และครูอาจารย์ ตั้งแต่อดีตจนถึงปัจจุบันที่เป็นผู้ให้การสนับสนุนด้านการศึกษา และเป็นผู้ให้กำลังใจด้วยดีตลอดมา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาพอสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	3
2.1 ตัวประมวลผลสัญญาณเชิงเลข	3
2.1.1 สถาปัตยกรรมและหน่วยความจำ	3
2.1.2 รีจีสเตอร์	5
2.1.3 การติดต่ออินพุตและเอาต์พุตของ DSK	7
2.1.4 วงจรเชื่อมต่อแอนะลอก (Analog Interface Circuit; AIC)	7
2.1.5 การติดต่อระหว่าง PC กับ TMS320C31	10
2.1.6 ซอฟต์แวร์ Development Tools	13
2.1.7 คอมไพล์ภาษาซี (Compiling C Code)	14
2.2 อัตราการสุ่มตัวอย่าง (sampling)	16
2.3 หน่วยความจำ	18
2.3.1 การขยายจำนวนบิตของหน่วยความจำ	20
2.3.2 การขยายจำนวนหน่วยความจำ	21
2.3.3 การจัดสรรพื้นที่ในหน่วยความจำ	23
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	25
3.1 ส่วนของฮาร์ดแวร์	25
3.1.1 การต่อหน่วยความจำภายนอก (ROM/RAM)	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.2 การเชื่อมต่อตัวแปลงสัญญาณจากแอนะลอกเป็นดิจิตอล (A/D) และตัวแปลงสัญญาณดิจิตอลเป็นแอนะลอก (D/A)	27
3.2 ส่วนของซอฟต์แวร์	30
3.2.1 การสร้างคลื่นไซน์ (Sine Wave)	29
3.2.2 การแปลงฟูเรียร์โดย Fast Fourier Transfrom (FFT)	30
3.2.3 การเชื่อมต่อหน่วยความจำภายนอก	33
3.2.4 การใช้งานบอร์ด DSK TMS320C31 เบื้องต้น	33
3.2.5 การใช้ชุดคำสั่งแบบง่าย	33
3.2.6 วงจรกรองแบบเอฟไออาร์ (FIR Filter)	33
บทที่ 4 การทดลองและผลการทดลอง	35
4.1 การกำเนิดสัญญาณไซน์ (Sine Wave)	35
4.2 การทดลองแปลงฟูเรียร์โดยวิธี Fast Fourier Transfrom (FFT)	37
4.3 การทดลองการเชื่อมต่อหน่วยความจำภายนอก	38
4.4 การทดลองการใช้งานบอร์ด DSK TMS320C31 เบื้องต้น	39
4.5 การทดลองการใช้ชุดคำสั่งแบบง่าย	41
4.6 การทดลองเอฟไออาร์ฟิลเตอร์ (FIR Filter)	42
บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไข และการพัฒนา	43
5.1 บทสรุป	43
5.2 ปัญหาและแนวทางการแก้ไข	43
5.3 แนวทางในการพัฒนา	44
ภาคผนวก ก เครื่องต้นแบบ	45
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	47
ภาคผนวก ค ผังการทำงานและโปรแกรม	53
ภาคผนวก ง รายการอุปกรณ์	67
ภาคผนวก จ ใบงานการทดลอง	69
ภาคผนวก ฉ เฉลยใบงานการทดลอง	106
ภาคผนวก ช รายละเอียดและคุณสมบัติของอุปกรณ์	144

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณนุกรม 155
ประวัติผู้แต่ง 156



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ขาต่างๆ ของชิพ AIC	9
ตารางที่ 2.2 การแปลงรหัส 1 จาก 4	22
ตารางที่ ง.1 รายการอุปกรณ์ที่ใช้ในวงจรการเชื่อมต่อแรมภายนอก	68
ตารางที่ ง.2 รายการอุปกรณ์ที่ใช้ในวงจรการเชื่อมต่อ A/D และ D/A	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 สถาปัตยกรรม TMS320C31	4
รูปที่ 2.2 ตำแหน่งหน่วยความจำ	6
รูปที่ 2.3 อินพุตและเอาต์พุตของระบบ	7
รูปที่ 2.4 ตัวถังชิพ (AIC)	8
รูปที่ 2.5 การทำงานภายในของ AIC ชิพ	8
รูปที่ 2.6 การเชื่อมต่อ Parallel Port เข้าไปใน DSK บอร์ด	11
รูปที่ 2.7 การแก้ไข PATH Statements	12
รูปที่ 2.8 พื้นฐานหน้าจอ Debugger	12
รูปที่ 2.9 ส่วนประกอบ TMS320C3x/C4x เกี่ยวกับซอฟแวร์ Development	13
รูปที่ 2.10 การใช้ C Compiler	14
รูปที่ 2.11 การใช้คำสั่ง Cl30 ในการคอมไพล์ภาษาซี	15
รูปที่ 2.12 หลังจากการคอมไพล์ภาษาซี	15
รูปที่ 2.13 การสุ่มข้อมูลสัญญาณแอนะลอก	16
รูปที่ 2.14 ผลของการ Sampling	17
รูปที่ 2.15 การแบ่งประเภทของหน่วยความจำ	18
รูปที่ 2.16 สแตติกแรม 256 x 4 บิต สองตัวต่อให้มีขนาด 256 x 8 บิต	20
รูปที่ 2.17 การต่อแรม 256 x 4 บิต เพื่อให้ได้ขนาด 1024 x 4 บิต	21
รูปที่ 2.18 การต่อเกต NAND และ NOT เป็นหน่วยแปลรหัส 1 จาก 4	22
รูปที่ 2.19 การใช้สัญญาณ CS ช่วยเลือกแผนผังแรมหรือรอม	23
รูปที่ 2.20 วงจรถอดรหัส	24
รูปที่ 2.21 การจัดหน่วยความจำ 4k x 8บิต ลดในเนื้อที่ทั้งหมด 64k	24
รูปที่ 3.1 ตำแหน่งที่ต่อหน่วยความจำ	26
รูปที่ 3.2 การต่อวงจรหน่วย ความจำภายนอก (ROM/RAM)	27
รูปที่ 3.3 การต่อตัวแปลงสัญญาณจากแอนะลอกเป็นดิจิตอล (A/D)	28
รูปที่ 3.4 ตัวแปลงสัญญาณจากดิจิตอล เป็นสัญญาณแอนะลอก (D/A)	29
รูปที่ 3.5 การรวมวงจรถอดรหัส (Decoder)	29
รูปที่ 3.6 วิธีการลดทอนเวลา	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.1 ผลการทดลองการกำเนิดสัญญาณไซน์ แบบจำนวน N = 4 จุด	35
รูปที่ 4.2 ผลการทดลองการกำเนิดสัญญาณไซน์ แบบจำนวน N = 4 จุด	36
รูปที่ 4.3 ผลการทดลองการแปลงฟูเรียร์โดยวิธี Fast Fourier Transform (FFT)	37
รูปที่ 4.4 ผลการทดลองการรันโปรแกรม SIN RAM dsk ผ่านหน่วยความภายจำนอก	39
รูปที่ 4.5 หน้าจอ Debugger	40
รูปที่ 4.6 ผลการทดลองการรันโปรแกรม Sin4p.asm	40
รูปที่ 4.7 ผลการทดลองการรันโปรแกรม FIR Filter	42
รูปที่ ก.1 ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 ด้านนอก	46
รูปที่ ก.2 ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 ด้านใน	46
รูปที่ ข.1 วงจรเชื่อมต่อแรมภายนอก	48
รูปที่ ข.2 วงจรเชื่อมต่อA/D และ D/A	49
รูปที่ ข.3 แผ่นวงจรพิมพ์วงจรแรมภายนอกด้านบน	50
รูปที่ ข.4 แผ่นวงจรพิมพ์วงจรแรมภายนอกด้านล่าง	51
รูปที่ ข.5 แผ่นวงจรพิมพ์วงจร A/D และ D/A	52
รูปที่ ค.1 ผังการทำงานของโปรแกรมการกำเนิดคลื่นไซน์	54
รูปที่ ค.2 ผังการทำงานของโปรแกรม Fast Fourier Transform (FFT)	55
รูปที่ ค.3 ผังการทำงานของโปรแกรมฟิลเตอร์ แบบ FIR	56
รูปที่ ค.4 โปรแกรม Sin4p.asm	57
รูปที่ ค.5 โปรแกรม MATRIX.ASM	58
รูปที่ ค.6 โปรแกรม SIN RAM.ASM	59
รูปที่ ค.7 โปรแกรม AICCOM31.ASM	62
รูปที่ ค.8 โปรแกรม Fast Fourier Transform (FFT)	65
รูปที่ ค.9 โปรแกรม LP11SIM ASM	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ปัจจุบันนี้วิทยาการทางด้านคอมพิวเตอร์ได้เจริญก้าวหน้ามากและก้าวไปอย่างรวดเร็วโดยเฉพาะเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) จะมีประสิทธิภาพสูงกว่าแต่ก่อนมาก ทั้งด้านความจุของหน่วย ความจำและความเร็วในการประมวลผล แต่ราคากลับต่ำลงเรื่อยๆ ทำให้การศึกษาเรื่องของการประมวลสัญญาณไฟฟ้าดิจิตอลนั้น สามารถกระทำได้ง่ายขึ้น เพราะสามารถนำคอมพิวเตอร์มาประมวลผลได้ซึ่งสามาถตัดข้อเสียของระบบประมวลผลแอนะลอกเรื่องการเปลี่ยนแปลงค่าของอุปกรณ์ ตามอุณหภูมิไปได้ และหากเป็นการกระทำกับสัญญาณที่มีความถี่สูงๆ ปัจจุบันก็มีอุปกรณ์ต่อเชื่อมกับคอมพิวเตอร์เพื่อประมวลผลสัญญาณโดย ฮาร์ดแวร์ ซึ่งจะมีความเร็วสูงมากและได้มีการพัฒนาความเร็วสูงไปเรื่อยๆ เช่น แผงวงจรการประมวลผล (DSP Board) TMS320C31 ของบริษัท Texas Instruments เป็นต้น

ในส่วนของโครงงานนี้จะนำเสนอเกี่ยวกับ การศึกษาการประมวลสัญญาณเชิงเลข (DSP) และได้เพิ่มประสิทธิภาพของบอร์ด DSP TMS320C31 ให้มีประสิทธิภาพมากขึ้น เพื่อให้เหมาะกับการศึกษาพร้อมทั้งยังมีใบงานให้ได้ทดลองปฏิบัติ เพราะหากผู้ที่ศึกษาในเรื่องของการประมวลผลสัญญาณเชิงเลขไม่มีการปฏิบัติหรือนำทฤษฎีที่ได้จากการศึกษา ไปใช้ทดลองว่าจะสามารถเป็นอย่างที่ผู้ศึกษาเข้าใจหรือไม่ อาจทำให้เกิดปัญหาเกี่ยวกับการเรียนรู้ของผู้ที่ศึกษา ผู้ที่ศึกษาอาจจะไม่สามารถนึกภาพหรือจินตนาการเกี่ยวกับการประมวลผลสัญญาณเชิงเลขได้เลย

1.2 ขีดความสามารถของโครงงาน

1) มีแผงวงจรขยายความสามารถการทำงานร่วมกับบอร์ด DSK TMS320C31
2) มีแผงวงจรขยายมีหน่วยความจำ ROM ไม่น้อยกว่า 64 k และ RAM ไม่น้อยกว่า 128 k
3) บนแผงวงจรขยายมี A/D และ D/A ที่มีการ Sampling Rate ไม่ต่ำกว่า 1 MHz
4) มีใบงานเพื่อใช้ทดลองการทำงานของชุดทดลองไม่ต่ำกว่า 6 ใบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาดังนี้, ตัวประมวลผลเชิงเลข, Sampling, หน่วยความจำ

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงเนื้อหาที่เกี่ยวกับวงจรต่างๆ และโปรแกรมที่ใช้ในโครงงานนี้

บทที่ 4 การทดลองและผลการทดลอง ผลการทดลองของซอฟต์แวร์และผลการทดลองของฮาร์ดแวร์

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนาเป็นขั้นตอนการสรุปผล ปัญหาที่เกิดขึ้น, แนวทางการแก้ไข และแนวทางในการพัฒนาโครงงานให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์

ภาคผนวก ค ผังการทำงานและโปรแกรม

ภาคผนวก ง รายการอุปกรณ์

ภาคผนวก จ ใบงานการทดลอง

ภาคผนวก ฉ เฉลยใบงานการทดลอง

ภาคผนวก ช รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ตัวประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)

ใช้บอร์ด DSK (DSP Starter Kit) TMS320C31 เป็นไมโครโปรเซสเซอร์ที่ต้องใช้กับชุด คำสั่งพิเศษและสถาปัตยกรรมเหมาะกับการประมวลผลสัญญาณเชิงเลข

คุณสมบัติของบอร์ด DSK

1) TMS320C31 เป็นตัวประมวลผลแบบจุดทศนิยม(Floating-Point)จะประกอบไปด้วย AIC ย่อมาจาก Analog Interface Circuit Chip ซึ่งมีไว้สำหรับการทำโปรแกรม ADC (ตัวเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก) กับ DAC (ตัวเปลี่ยนสัญญาณแอนาะลอกเป็นสัญญาณดิจิตอล) และ อินพุตและเอาต์พุต ฟิลเตอร์, สัญญาณทั้งหมดบนชิพ

2) บอร์ด DSK จัดต่อพอร์ตขนาน (เครื่องพิมพ์) เพื่อติดต่อกับคอมพิวเตอร์

3) สามารถใช้งานร่วมกับออสซิลโลสโคบ, เครื่องกำเนิดสัญญาณ, ลำโพงและ Signal/Spectrum Analyzer ได้

4) สามารถโปรแกรมด้วยภาษาแอสเซมบลี้และภาษาซีได้

5) ใช้กับแหล่งจ่ายไฟตรง 7.5-12 V

2.1.1 สถาปัตยกรรมและหน่วยความจำ TMS320C31

TMS320C31 มีหน่วยความจำของเวิร์ด (Word) เท่ากับ 2k (32 บิต) จากหน่วยความจำภายในและมี 16 ล้านเวิร์ด สำหรับตำแหน่งหน่วยความจำภายนอก ซึ่งประกอบไปด้วย ส่วนของโปรแกรม, ส่วนของข้อมูลอินพุตและเอาต์พุต

TMS320C31 เป็นสมาชิกในตระกูล C3X เป็น Floating-Point (จุดทศนิยม) มีขนาด 32 บิต โดยมีหน่วยความจำสามารถแยกได้ดังนี้ ส่วนโปรแกรม Boot-Loader (ส่วนของการเริ่มต้นโปรแกรม), หน่วยความจำภายใน 2k (RAM) ในสถาปัตยกรรมมาสารถแบ่งได้ดังนี้

1) Arithmetic Logic Unit (ALU)

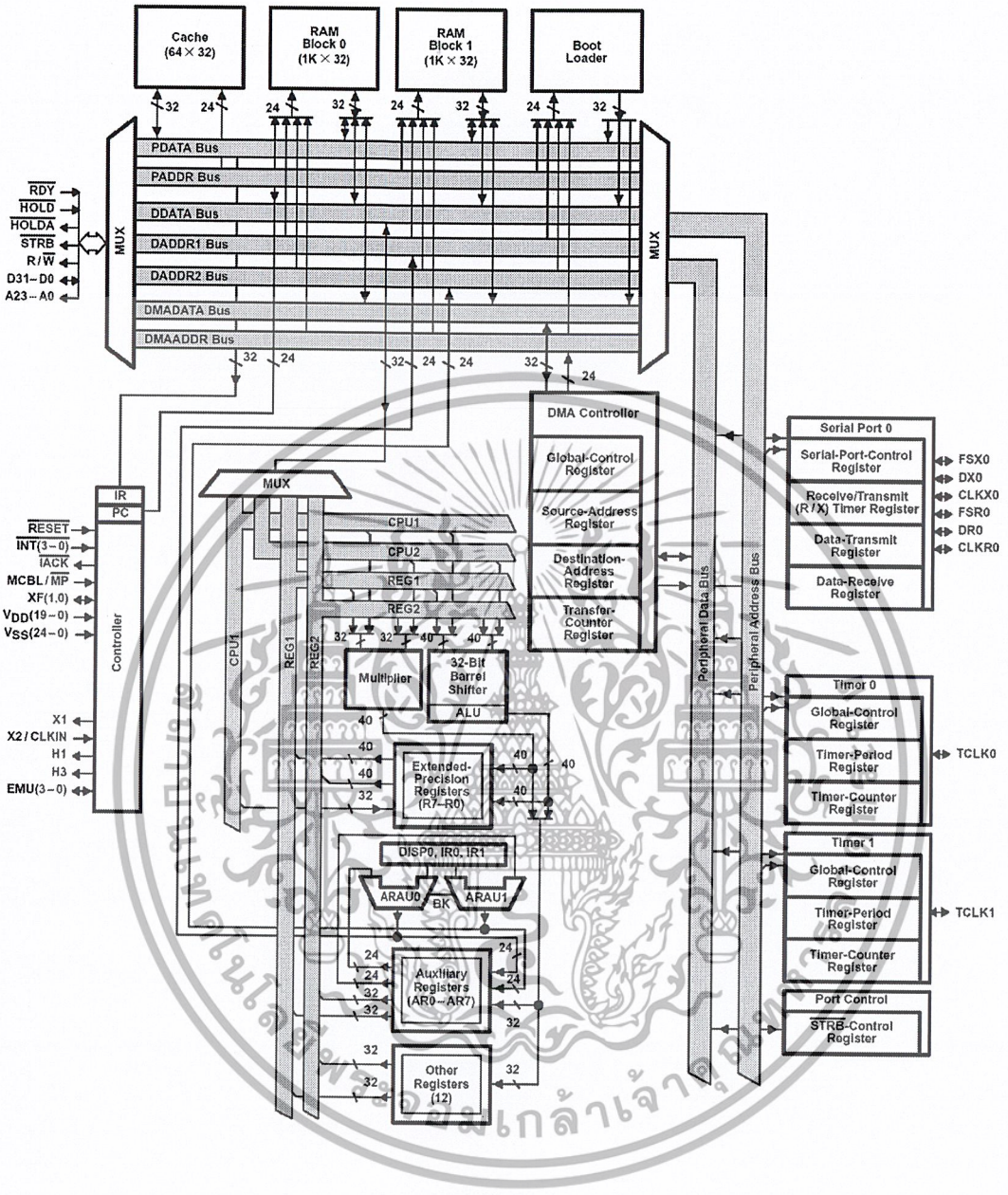
2) 32-Bit Barrel Shifter

3) Internal Buses (CPU1/CPU2 and REG1/REG2)

4) Auxiliary Register Arithmetic units (ARAUs)

5) CPU Register File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 สถาปัตยกรรม TMS320C31

จำนวนเต็ม (Fixed-Point) มีขนาด 24 บิตและค่าทศนิยม (Floating-Point) มีขนาด 32 บิต สำหรับ Floating-Point และ Fixed-Point จะมีความเร็ว 33 ns ต่อคำสั่งหนึ่ง Cycle, การคูณและ ALUจะคำนวณใน Cycle เดียว เมื่อไรมีการกระทำ Multiplier โดย Floating-Point Multiplicati Inputs มีขนาด 32 บิต แต่ผลลัพธ์มีขนาด 40 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 รีจีสเตอร์

1) R0-R7มีทั้งหมด 8 ตัว เป็นรีจีสเตอร์ขนาด 40 บิต สามารถเก็บผลลัพธ์ที่เป็นจำนวนเต็ม 32 บิต และสำหรับตัวเลขทศนิยมเก็บ 40 บิต

2) Ar0-AR7 เป็นรีจีสเตอร์ช่วย

3) IR0 และ IR1 ใช้สำหรับชี้ตำแหน่งหน่วยความจำ

4) ST ใช้บอกสถานะของ CPU

5) SP เป็น Stack Pointer

6) BK (Specify the Book)

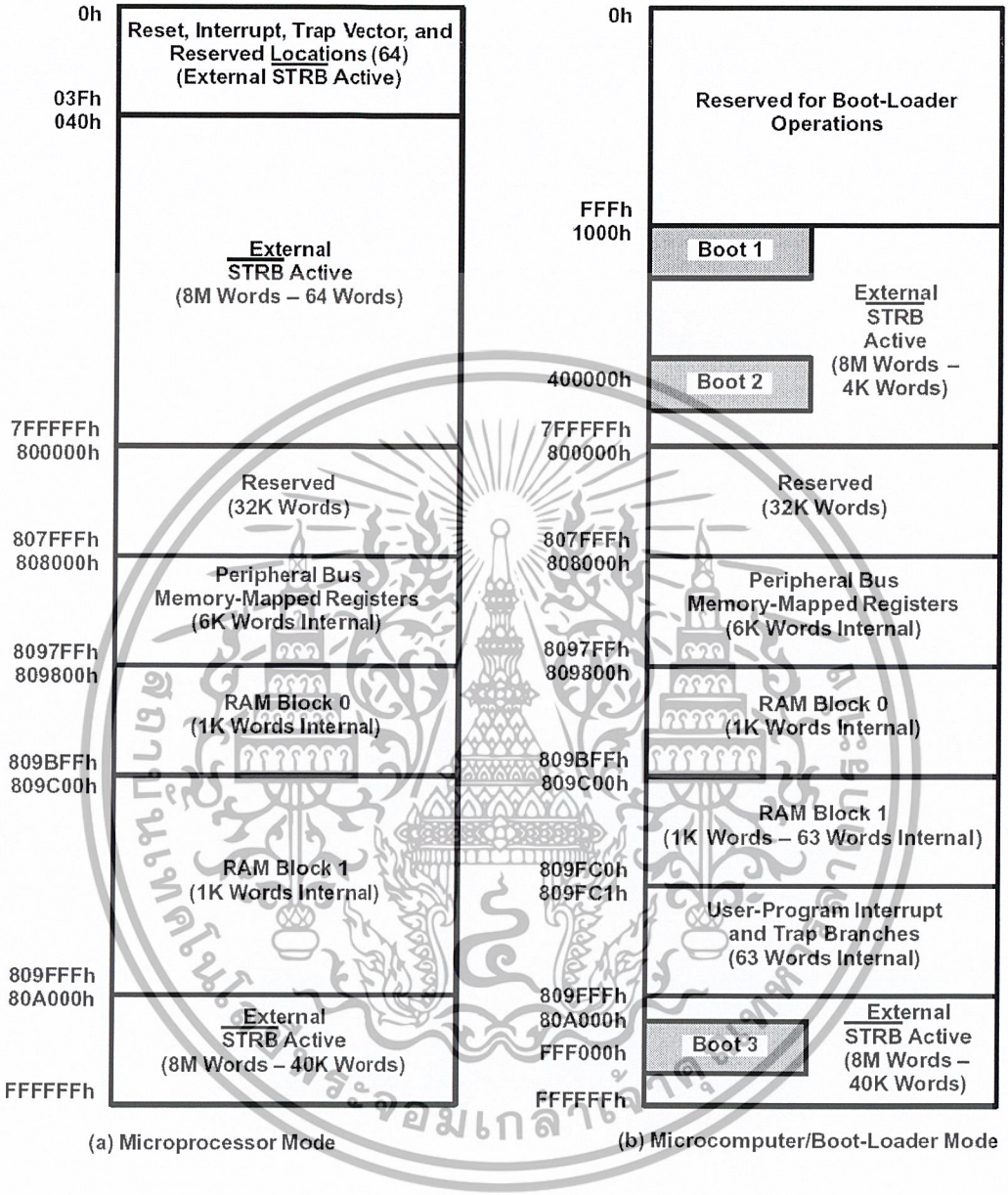
7) IE, IF, IOF คือ อินเตอร์รัพท์Enable, อินเตอร์รัพท์ Flag, I/O Flag(IOF)

8) RC (Repeat Count) ใช้เกี่ยวกับ Time

9) RS และ RE จะประกอบไปด้วยตำแหน่งเริ่มต้นกับตำแหน่งสุดท้าย

10) PC (Program Counter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

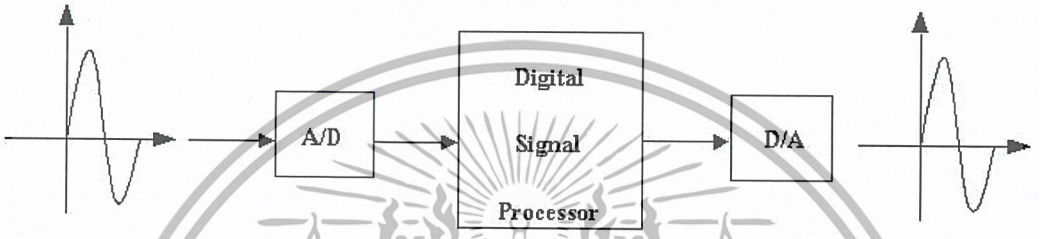


รูปที่ 2.2 ตำแหน่งหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การติดต่ออินพุตและเอาต์พุตของ DSK ขั้นพื้นฐาน

การติดต่ออินพุตและเอาต์พุตของ DSK ขั้นพื้นฐานแสดงในรูปที่ 2.2 ซึ่งประกอบไปด้วย แอนะลอกอินพุตและแอนะลอกเอาต์พุต ในส่วนอินพุตคือAntialiasing Filter ซึ่งจะมี A/D คือวงจรแปลงแอนะลอกเป็นดิจิตอล แล้วจึงนำสัญญาณที่ได้ไปประมวลผล โดยวงจรประมวลผลสัญญาณเชิงเลข ซึ่งระดับสูงสุดของสัญญาณอินพุตจะถูกจำกัดด้วยตัว A/D ดังแสดงในรูปที่ 2.1



รูปที่ 2.3 อินพุตและเอาต์พุตของระบบ

หลังจากสัญญาณผ่านการประมวลแล้ว ผลลัพธ์ที่ต้องการจะต้องผ่าน D/A คือวงจรแปลงดิจิตอลเป็นแอนะลอก

2.1.4 AIC (Analog Interface Circuit)

มีคุณสมบัติ ดังนี้

1) มีขนาด 14 บิต Dynamic Range ADC และ DAC

2) ADC และ DAC มีอัตราการสุ่ม 19,200 Samples ต่อวินาที

3) มี Switched-capacitor Antialiasing เป็น Input Filter และ Output Filter

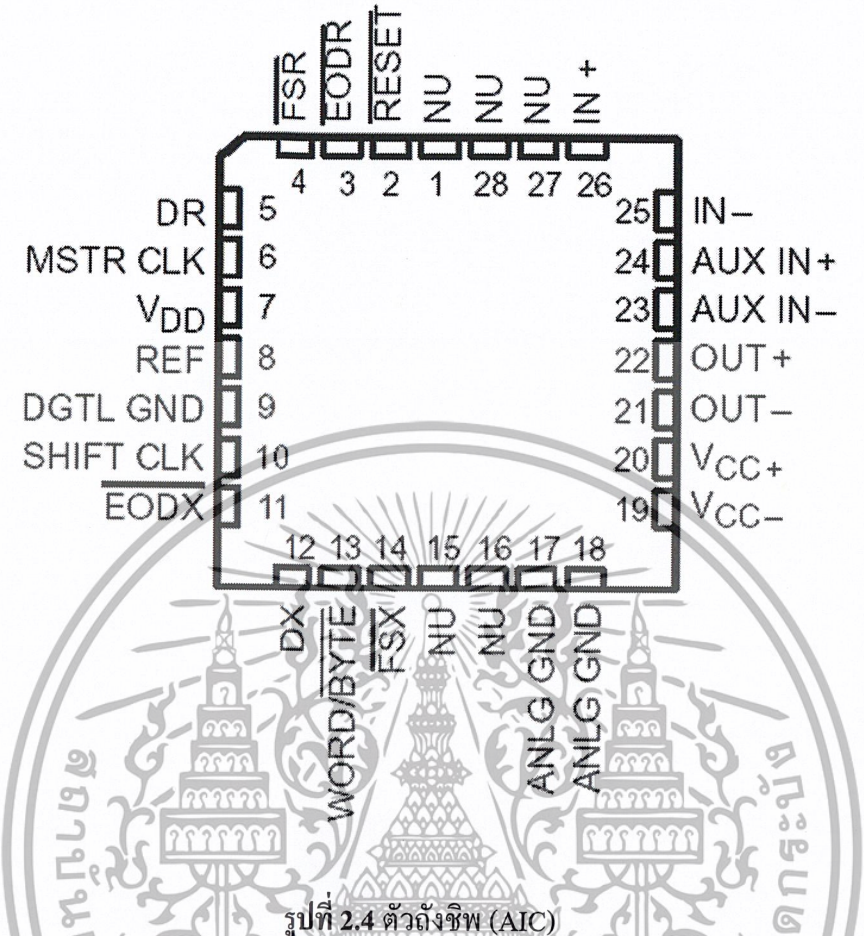
4) เป็นชุด Serial Port สำหรับต่อโดยตรง เพื่อนำไปประมวลผลสัญญาณเชิงเลข

5)มี Synchronous และ Asynchronous ADC และ DAC Convert

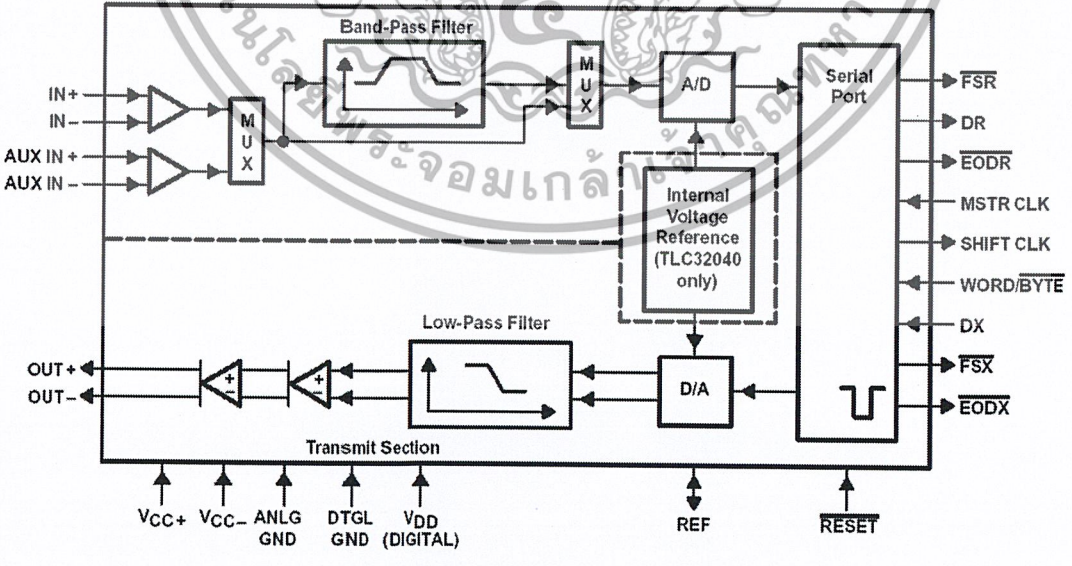
6) เป็นชุด Port ส่วนติดต่อ SN74299

7) เป็นเทคโนโลยี CMOS ซึ่งแสดงในรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวถังชิพ (AIC)



รูปที่ 2.5 การทำงานภายในของ AIC ชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ขาต่างๆของชิพ AIC

ชื่อ	หน้าที่
ANLG GND	อยู่ที่ ขา 17, 18 เป็น Analog Ground อยู่ภายใน Analog Circuits จะไม่เชื่อมต่อกับ DGTL GND
AUX IN+	อยู่ที่ขา 24 เป็น Noninverting Auxiliary Analog Input สามารถเปลี่ยนค่าในการกรองความถี่แบบ Brand Pass และ A/D
AUX IN-	อยู่ที่ขา 23 Inverting Auxiliary Analog Input (อยู่ข้างบน AUX IN +)
DGTL GND	อยู่ที่ขา 9 Digital Ground อยู่ภายใน Logic Circuits ไม่เชื่อมต่อกับ ANLG GND
DR	อยู่ที่ขา 5 เป็นเอาต์พุต DR ถูกใช้เพื่อส่งไปข้อมูลจาก ADC ไปยัง AIC
DX	อยู่ที่ขา 12 เป็นอินพุต DX ถูกใช้เพื่อรับข้อมูลจาก DAC Input และหน่วยควบคุม ไปยัง AIC ทาง port
EODR	อยู่ที่ขา 3 เป็นเอาต์พุต เพื่อแสดงการจบของข้อมูลและเป็นชุด port ของ Timing Diagrams ระหว่างโหมดคำ
EODR	ไบต์ต่ำหลังไบต์แรกเป็น Transmitted จาก AIC สู่ TMS320C31 เป็นชุด Port และ Kept เป็น Logic Low กระทั่งไบต์ที่สองได้เป็น Transmitted TMS320C31 เพื่อบอกความแตกต่างระหว่างไบต์สอง ซึ่งก็คือ วินาที
EODR	จะปรากฏ หลัง Secondary ของการสื่อสาร
EODX	อยู่ที่ขา 11 เป็นเอาต์พุตเป็นการแสดงว่าจบของข้อมูลที่ส่งไปยังชุด Port การ Timing Diagram ระหว่าง โหมดคำที่ Timer, EODX คือ ไบต์ต่ำ
SR 4	เป็นเอาต์พุต เป็นเฟรม sync ที่ใช้รับข้อมูลในชุด Transmission โหมด ซึ่งถูกอ้างถึงในคำ FRS คือ Held เป็น Logic Low ระหว่างบิต เมื่อเกิดการ Transmission เมื่อไร FRS เป็นไบต์ต่ำ TMS320C31 เมื่อชุด Port เริ่มต้นการรับบิตจาก AIC
FFSX	อยู่ที่ขา 14 เป็นเอาต์พุตเฟรม sync ส่งไป เมื่อไรที่ FSX เป็นไบต์ต่ำเมื่อนั้นชุด Port เริ่มต้นการส่งไปกลับสู่ AIC
DX	โหมดคำสั่งชุด Transmission ซึ่งถูกอ้างถึงในคำ FSX คือ Held ในบิตต่ำ
IN +	อยู่ที่ขา 26 เป็นอินพุต Non Inverting Input เป็นตัวขยาย
IN -	อยู่ที่ขา 25 เป็นอินพุต Inverting Input เป็นตัวขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ) ขาต่างๆ ของชิพ AIC

ชื่อ	หน้าที่
MS TR CLK	อยู่ที่ขา 6 เป็นอินพุต Master Clock เมื่อถูกใช้ Key Logic Signals ที่ AIC Switched-Capacitor, A/D และ D/A ค่าเวลาที่ตั้งไว้ภายใน Diagram จะให้เอาต์พุตเป็นสัญญาณ โดยความถี่สัญญาณขึ้นอยู่กับการโปรแกรม
OUT +	อยู่ที่ขา 22 เป็นเอาต์พุต Non-Inverting ส่งออกสัญญาณเป็น Analog ส่งออก Power Amplifier
OUT -	อยู่ที่ขา 21 เป็นเอาต์พุต Inverting ส่งออกสัญญาณเป็น Analog ส่ง Power Amplifier
REF	อยู่ที่ขา 8 เป็นแรงดันอ้างอิงสำหรับ TLC32040 สำหรับ TLC32040 และ TLC32041 แรงดันภายนอก
V +	อยู่ที่ขา 20 เป็น Positive Analog จ่ายแรงดัน 5V ถึง +,- 5%
V -	อยู่ที่ 19 เป็น Analog จ่ายแรงดัน –5 ถึง +,-5%

2.1.5 การติดต่อระหว่าง PC (Personal Computer) กับ TMS320C31

การติดต่อระหว่างคอมพิวเตอร์กับ TMS320C31 จะเริ่มต้นจากการ Load Kernel จากคอมพิวเตอร์ ผ่านพอร์ตขนาน ก่อนที่ Kernel จะ Load TMS320C31 ต้องการการรีเซ็ตจากคอมพิวเตอร์ ผ่านพอร์ตขนาน โดยในการ Boot (การเริ่มต้น) จะโหลดจากตำแหน่งเริ่มต้น 0xFFF000

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับ แผงวงจรDSK ต้องใช้การเชื่อมต่อพอร์ตขนาน (เครื่องพิมพ์)

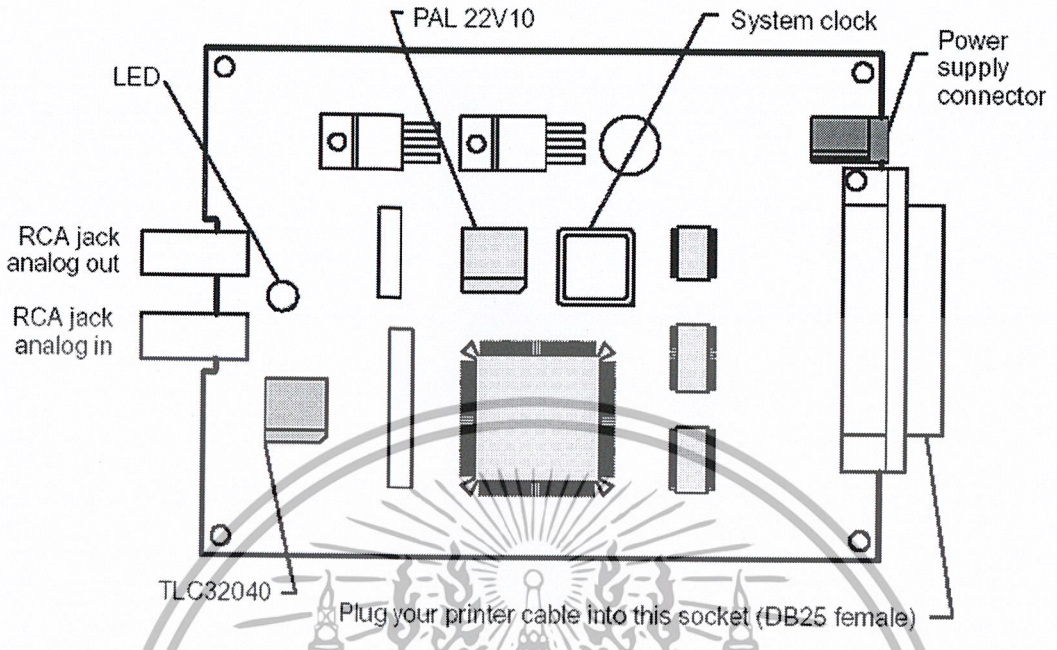
ขั้นตอน 1

1) ปิดสวิตซ์เครื่องคอมพิวเตอร์

2) เชื่อมต่อ Parallel Port (เครื่องคอมพิวเตอร์) สู่ Parallel Communication Port (DSKบอร์ด)

3) ต่อแหล่งจ่ายไฟตรง 7 – 12 Vdc หรือ 6 – 9 Vac

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การเชื่อมต่อ Parallel Port เข้าไปใน DSK บอร์ด

4) เสียบแหล่งจ่ายเข้าที่รูเสียบ (jack)

5) เปิดเครื่องคอมพิวเตอร์

6) LED เกิดแสงสว่างทั้งสีแดงหรือสีเขียว

ขั้นตอน 2 : การติดตั้ง DSK ซอฟต์แวร์

ส่วนนี้อธิบายการติดตั้ง Debugger ซอฟต์แวร์

1) สร้างไดเร็กทอรี่ ชื่อ Dsktools เอกสารจะบรรจุ อยู่ใน DSK Assembler และซอฟต์แวร์ Debugger

2) ทำการ Copy โปรแกรม

ขั้นตอน 3 : แก้ไขแฟ้ม Config.sys

เมื่อไรมีการใช้ Debugger ต้องเกิดการแก้ไข Config.sys และเพิ่ม file

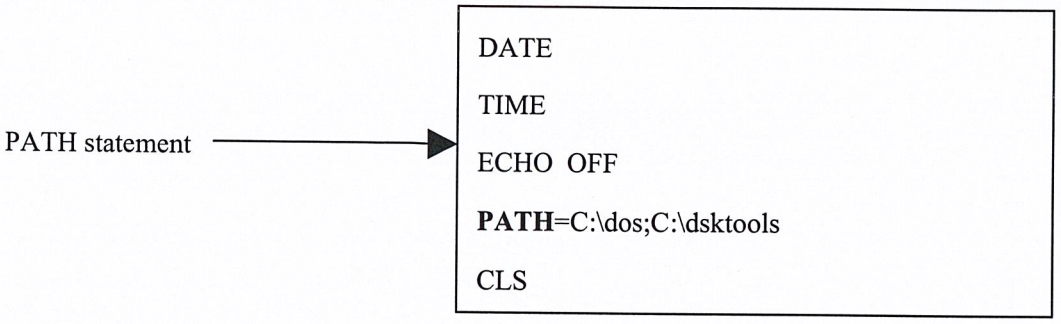
ขั้นตอน 4 : การแก้ไข PATH Statement

ทำเพื่อให้แน่ใจ Debugger และ Assembler เป็นการอ้างจาก Directory ในคอมพิวเตอร์ต้องแก้ไข PATH statement เพื่อค้นหาไดเร็กทอรี่ของ Dsktools เช่น

PATH = C:\ dsktools; pathname2 ; pathname3

การติดตั้ง DSK Environment บน DOS Command (ตัวอย่าง autoexec.bat File)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การแก้ไข PATH Statement

ถ้าแก้ไข autoexec.bat

autoexe

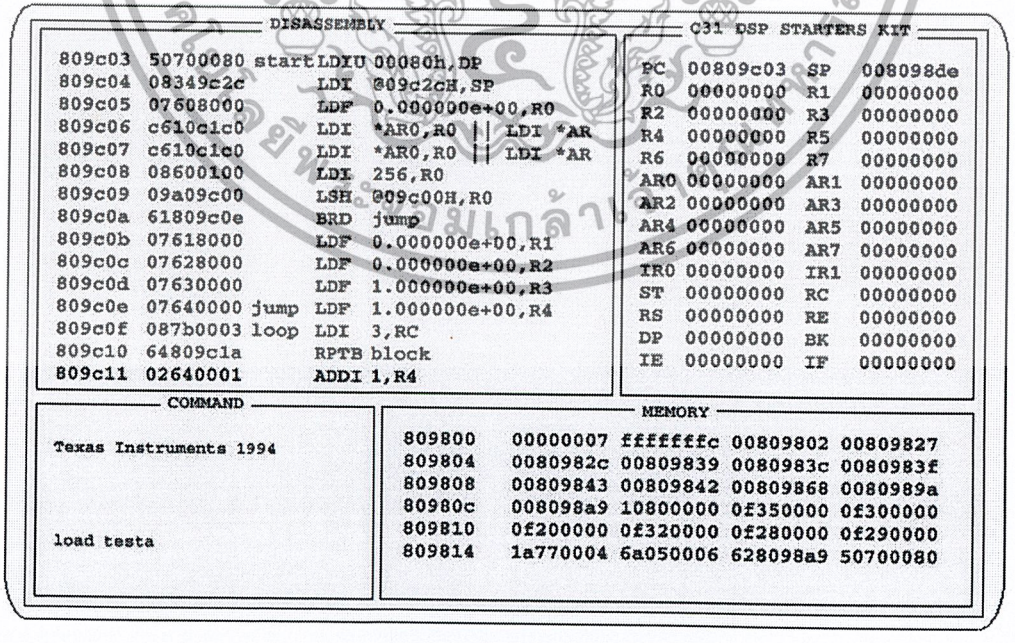
ขั้นตอน 5 : ทดสอบการติดตั้ง

เพื่อให้แน่ใจในการติดตั้ง DSK บอร์ด, Assembler, และ Debugger, ใส่คำสั่งต่อไปนี้ที่ระบบพร้อมเริ่มต้น DSK

debugger:

dsk3d

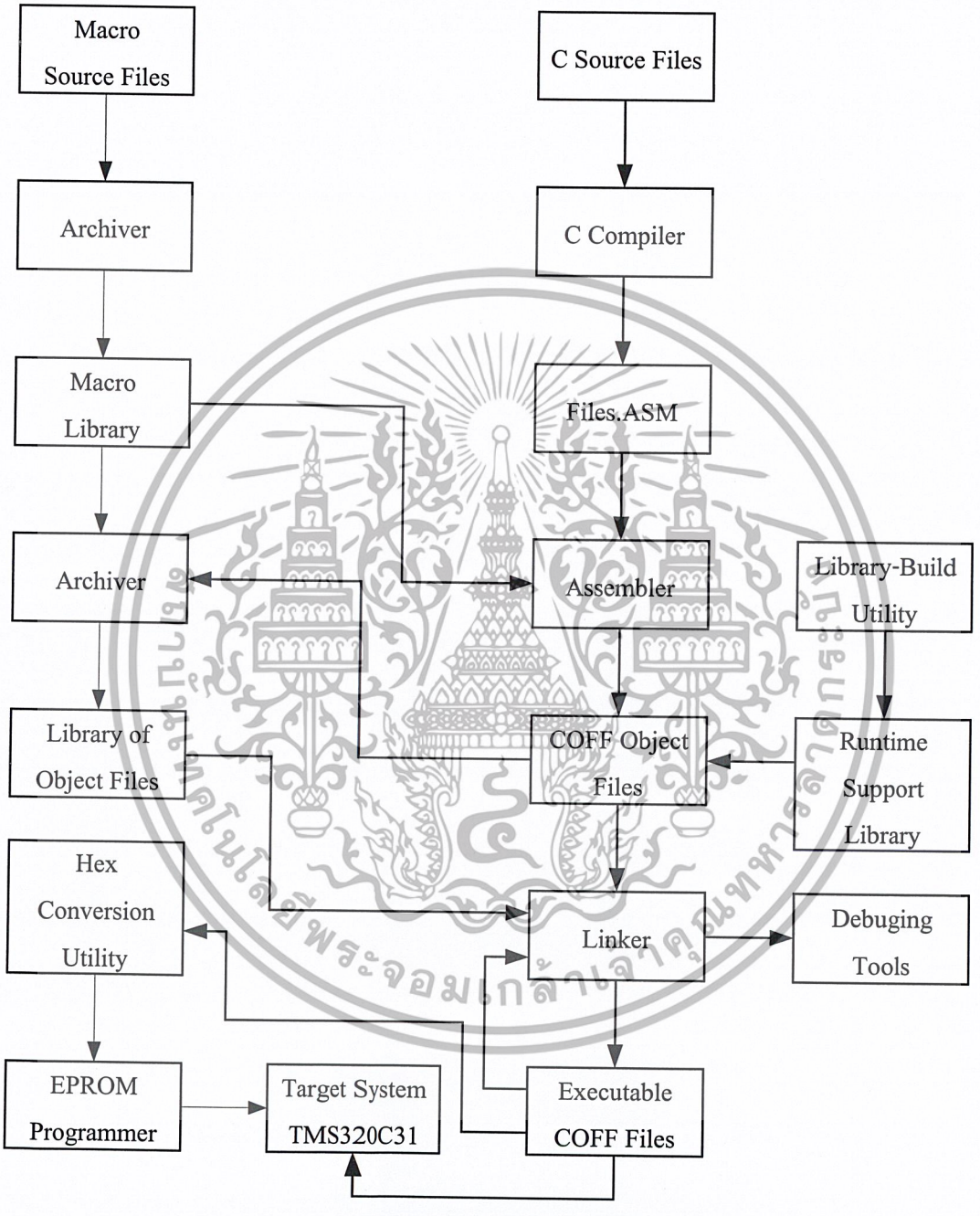
หลังจากใส่คำสั่ง dsk3d จะเห็นหน้าจอ ดังรูปที่ 2.8



รูปที่ 2.8 พื้นฐานหน้าจอ Debugger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ซอฟต์แวร์ Development Tools



รูปที่ 2.9 ส่วนประกอบ TMS320C3x/C4x เกี่ยวกับซอฟต์แวร์ Development

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือ ที่แสดงในรูปที่ 2.9

1) C Compiler จะเป็นข้อมูลรหัสซึ่งจะใช้กับ TMS320C3x หรือ TMS320C4x หรือภาษา Assembly โปรแกรม Shell (cl30), Optimizer (opt30), Interlist (clist) จะอยู่ในชุด C Compiler

2) โปรแกรม Shell เป็นโปรแกรมช่วยให้ตรวจสอบโดยอัตโนมัติ, assemble และ Link Source Modules

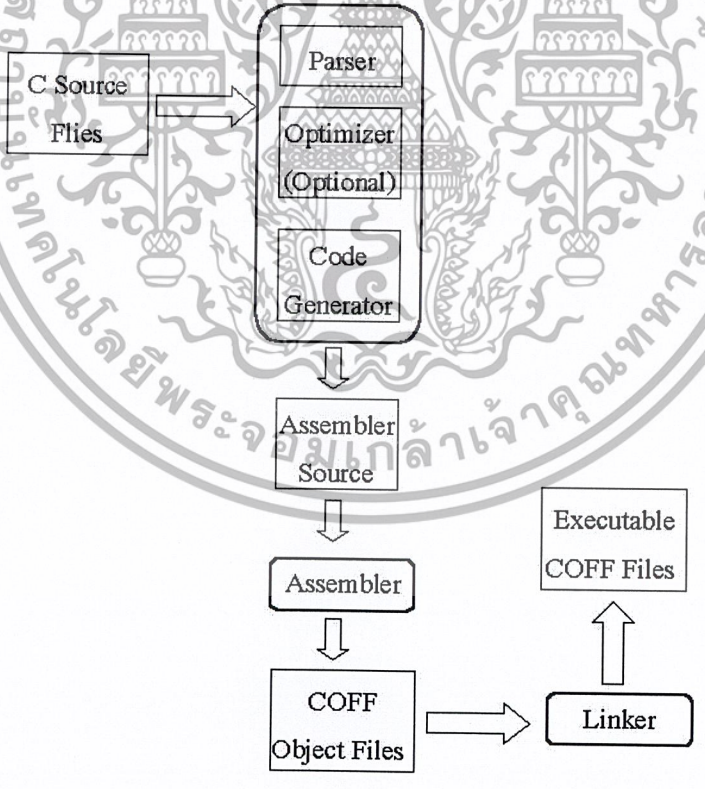
3) Optimizer แก้ไขปรับปรุงโปรแกรมภาษา C

4) Interlist เครื่องมือ Interlist C จะแปลงข้อมูลประโยคไปเป็นภาษา Assembly

5) Assembler จะแปล Assembly ภาษาข้อมูลเข้าไปในเครื่อง

2.1.7 การคอมไพล์ภาษาซี (Compiling C Code)

cl30 เป็นโปรแกรม shell คือเครื่องมือใช้ Compile, Assemble และ ข้อกำหนดเพิ่มเติม การเชื่อมต่อ Compiler จะประกอบด้วย Parser, Optimizer และ Generator Code, Assembler ซึ่งทำให้เกิด แฟ้ม COFF



รูปที่ 2.10 การใช้ C Compiler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Linker สามารถ Invoked (อ้าง) เป็นส่วนของกระบวนการ Larger หรือสามารถ Compiler และ Assembler Invoking ของ C Compiler เพื่อใช้ในการ Compiler ใส่

```
Cl30 [ -option] [filenames] [ -z [link_options] [object files]]
```

รูปที่ 2.11 การใช้คำสั่ง Cl30 ในการคอมไพล์ภาษาซี

Cl30 คือ คำสั่งในการ Invokes ของ Compiler

Options คือ ข้อกำหนดที่มีผลต่อวิธี Compiler ของกระบวนการ Input Files

- z คือ ข้อกำหนดใช้ Linker เป็น หน่วยควบคุมข้อกำหนดทางเชื่อมต่อ เช่น

> Cl30 symtab

เมื่อ Enter แล้วจะเกิดการ Compiler จะพบข้อความด้านล่าง

```
[symtab]
TMS320C3x/4x ANSI C Compiler Version x.xx
Copyright (c) 1987-1997 , Texas Instruments Incorporated
"symtab.c":== > main
"symtab.c":== > lookup
TMS320C3x/4x ANSI C Codegen Version x.xx
Copyright (c) 1987-1997 , Texas Instrument Incorporated
"symtab.c":== > main
"symtab.c":== > lookup
TMS320C3x/4x COFF Assembler Version x.xx
Copyright (c) 1987-1997 , Texas Instrument Incorporated
PASS 1
PASS 2
No Errors. No Warnings
```

รูปที่ 2.12 หลังจากการคอมไพล์ภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การสุ่มตัวอย่างข้อมูล

เมื่อต้องการแปลงสัญญาณแอนะลอกเปลี่ยนเป็นสัญญาณดิจิตอลเพื่อประมวลผลเราก็จะสามารถทำได้โดยการสุ่มตัวอย่างหรือตัดสัญญาณนั้นออกเป็นส่วนๆ และเมื่อกระทำต่อเนื่องก็จะเป็นเส้นตั้งเรียงกันไป ดังนั้นเราจะได้สัญญาณดิจิตอลเรียงเป็นตัวเลขติดต่อกันไป ปกติเราจะเรียกสัญญาณเหล่านี้ว่า Sequence ใช้สัญลักษณ์เป็น x[n] โดย n เป็นเหมือน เลขบอกตำแหน่งของข้อมูล (เส้นตั้ง) แต่ละตัวใน Sequence สิ่งที่สำคัญมากที่สุดคือเราจะสุ่มตัวอย่างสัญญาณด้วยความถี่เท่าใดจึงจะเหมาะสม (หรือจำนวนเส้นตั้งเท่าใดในหนึ่งวินาที)



(ก) การสุ่มข้อมูล ด้วยความถี่ต่ำ (ข) การสุ่มข้อมูล ด้วยความถี่สูง

(ค) ผล Output หลัง Sampling

รูปที่2.13 การสุ่มข้อมูลสัญญาณแอนะลอก

จากการศึกษาถึงผลของการสุ่มสัญญาณไฟฟ้าโดยทฤษฎีของ Shannon'sfamous sampling และได้ข้อสรุปออกมาว่า สัญญาณควรจะถูกสุ่มอย่างน้อยด้วยความถี่สุ่มประมาณ สองเท่าของความถี่สูงสุดที่เราจะทำการประมวล หากเราให้สัญญาณอินพุตเป็น f1 และ ความถี่สัญญาณสุ่มเป็น fs เราก็จะได้สมการง่ายๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

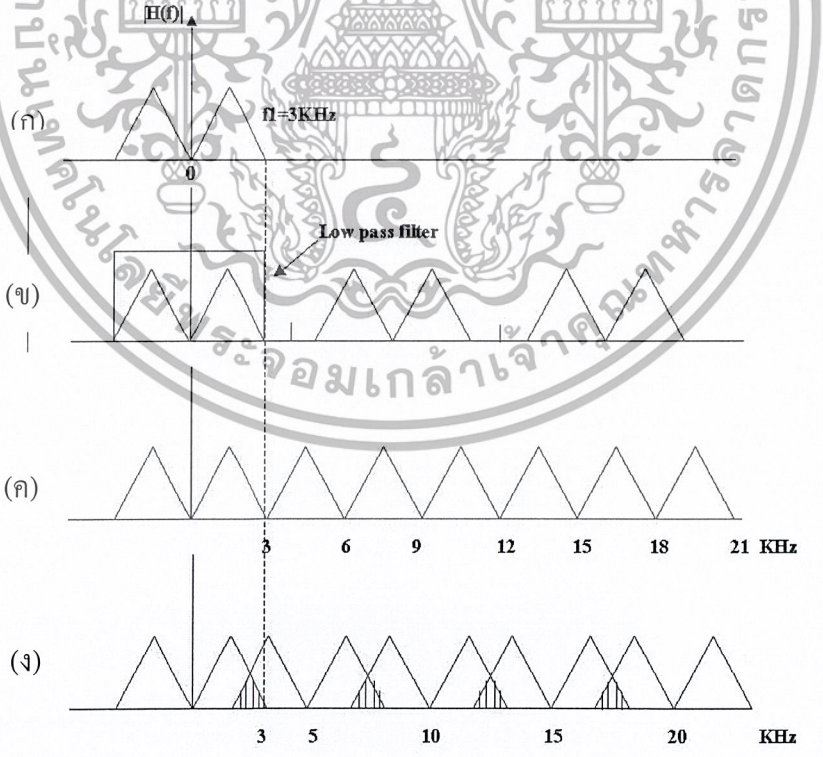
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$$fs = \frac{1}{2fl} \tag{2.1}$$

ดังนั้นที่ความถี่ 3 kHz เราจึงควรจะสุ่มสัญญาณนี้ด้วยความถี่อย่างน้อย 6 kHz หรือ ขนาดเท่ากับ 2 Sample/sec และถ้าหากเราให้เวลาของการ Sample แทนค่าด้วย T (Sample Interval) เราก็จะเขียนสมการง่ายๆ ได้อีกคือ

$$T = \frac{1}{2fl} \tag{2.2}$$

ซึ่งทำให้แน่ใจได้ว่าจะสามารถทำ D/A กลับสัญญาณคืนสู่แอนะลอกได้ (ในการปฏิบัติกับสัญญาณจริงๆ เรามักจะเพิ่มสัญญาณาสุ่มเป็น 2-3 เท่าของความถี่ที่สุ่ม) ในรูปที่ 2.14 แสดงถึงความถี่ของสัญญาณอินพุต (20-3kHz) และหากเราสุ่มด้วย ความถี่ 8 kHz ฉะนั้นความถี่ก็จะซ้ำตัวเองที่ 8 kHz, 16 kHz และเพิ่มต่อไปเรื่อยๆ ในหลักการเราต้องการจะ ประมวลผลเพียงสัญญาณดิจิตอลเป็นแอนะลอกเพื่อให้ได้ดังในรูป 2.14(ก) นั้น จึงสามารถทำได้โดยใช้ Low Pass ฟิลเตอร์มากรองสัญญาณเฉพาะช่วงที่ต้องการเท่านั้น โดยฟิลเตอร์ควรจำกัดสัญญาณที่ความถี่ 5 kHz ขึ้นไป



รูปที่ 2.14 ผลของการ Sampling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

48320

หากลดความถี่สุ่มลงจนถึงความถี่ต่ำสุดที่ Sampling Theory กำหนดจากตัวอย่างนี้เราลดลงมาที่ 6 kHz ผลคือ เราจะเห็นว่าความถี่จะเริ่มซ้ำที่ความถี่ 6 kHz , 12 kHz และต่อไปเรื่อย สัญญาณยังคงสามารถจะแปลงคืนมาเป็นแอนะลอกหรือกระทำ D/A ได้

หากเราลดความถี่ลงต่อไปอีกความถี่ของความถี่ก็จะขยับเข้ามาอีกจากผลของความถี่สุ่มที่ลดลง ทำให้เกิดการ Overlap ของความถี่ ดังแสดงในรูปที่ 2.14(ง) นั่นคือเรา จะไม่สามารถทำการแปลงสัญญาณนี้คืนมาสู่แอนะลอกได้ และโดยปกติแล้วหากเราสุ่มด้วยความถี่ตามทฤษฎี คือ 2f ในทางปฏิบัติเราไม่สามารถคืนสัญญาณได้เพราะในทางปฏิบัติเรา จะไม่สามารถออกแบบ Low Pass Filter ที่มีลักษณะของการ Cut off Frequency ได้อย่างในรูปที่ 2.14 (ข)

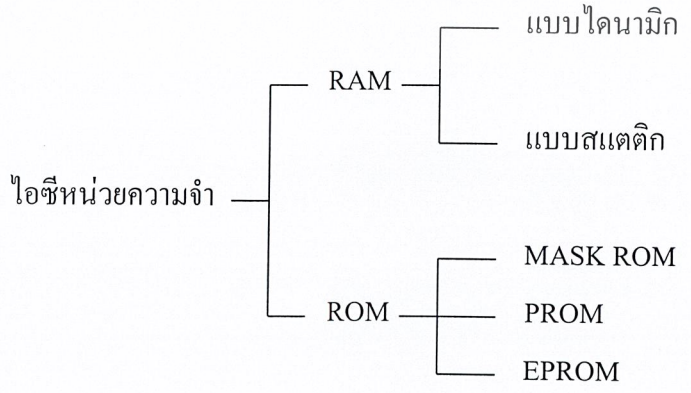
ดังนั้นจึงควรอย่างยิ่งที่ต้องสุ่มความถี่ ด้วยความถี่ที่สูงกว่าทฤษฎีกำหนดเพื่อประกันว่าสัญญาณจะสามารถคืนกลับมาเป็นแอนะลอกได้อย่างเดิมแน่นอน

2.3 หน่วยความจำ

หน่วยความจำในระยะแรกจะเป็นแหวนแม่เหล็ก (Core Memory) เป็นกระบอกแม่เหล็ก (Magnetic Drum) ซึ่งเป็นอุปกรณ์แม่เหล็กไฟฟ้าและเชิงกลไฟฟ้า ทำให้การติดต่อกับ CPU ทำได้ยาก วงจรซับซ้อนมาก การทำหน่วยความจำเป็นไอซีได้จึงเป็นเรื่องสำคัญทำให้วงจรในคอมพิวเตอร์ง่ายลงมาก

เมื่อเทคโนโลยีในการทำ MOS ไอซีเจริญขึ้น ก็สามารถทำไอซีที่มีหน่วยความจำที่มีความจุสูงขึ้น สามารถใช้งานในทางปฏิบัติได้ดีขึ้น จึงทำให้ไมโครคอมพิวเตอร์มีขนาดเล็กลงและสามารถใช้งานได้ดีเยี่ยม

ถ้าจะแบ่งชนิดของไอซีหน่วยความจำออกตามคุณสมบัติและการใช้งานจะสามารถแบ่งออกได้ดังนี้



รูปที่ 2.15 การแบ่งประเภทของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) RAM ย่อมาจาก Random Access Memory เป็นหน่วยความจำชนิดที่สามารถใช้เขียนข้อมูลและอ่านข้อมูลออกมาได้อย่างอิสระ RAM ยังมีทั้งชนิดไดนามิกแรมและสแตติกแรม แบบไดนามิกจะต้องมีการเขียนข้อมูลซ้ำอยู่ตลอด เพราะข้อมูลภายในเก็บได้ชั่วคราว แบบสแตติกไม่จำเป็นต้องเขียนข้อมูลซ้ำ ข้อมูลภายในจะคงค่าอยู่ตลอดเวลาที่ยังมีแหล่งจ่ายไฟอยู่

2) ROM ย่อมาจาก Read Only Memory ซึ่งก็คือ อ่านได้อย่างเดียวนั่นเอง เมื่อเราเขียนข้อมูลเข้าไปใน ROM แล้ว ข้อมูลจะไม่สูญหาย แม้เราจะหยุดจ่ายไฟเลี้ยงไอซีก็ตาม ในการใช้งานเราจะอ่านข้อมูลที่เก็บไว้ภายใน ROM อย่างเดียว

2.1) MASK ROM เป็น ROM ชนิดที่เขียนข้อมูลมาแล้วจากโรงงาน ในขั้นตอนการผลิตจะทำหน้ากากซึ่งมีอยู่เรียบร้อยแล้วถ่ายลงไปบนไอซี

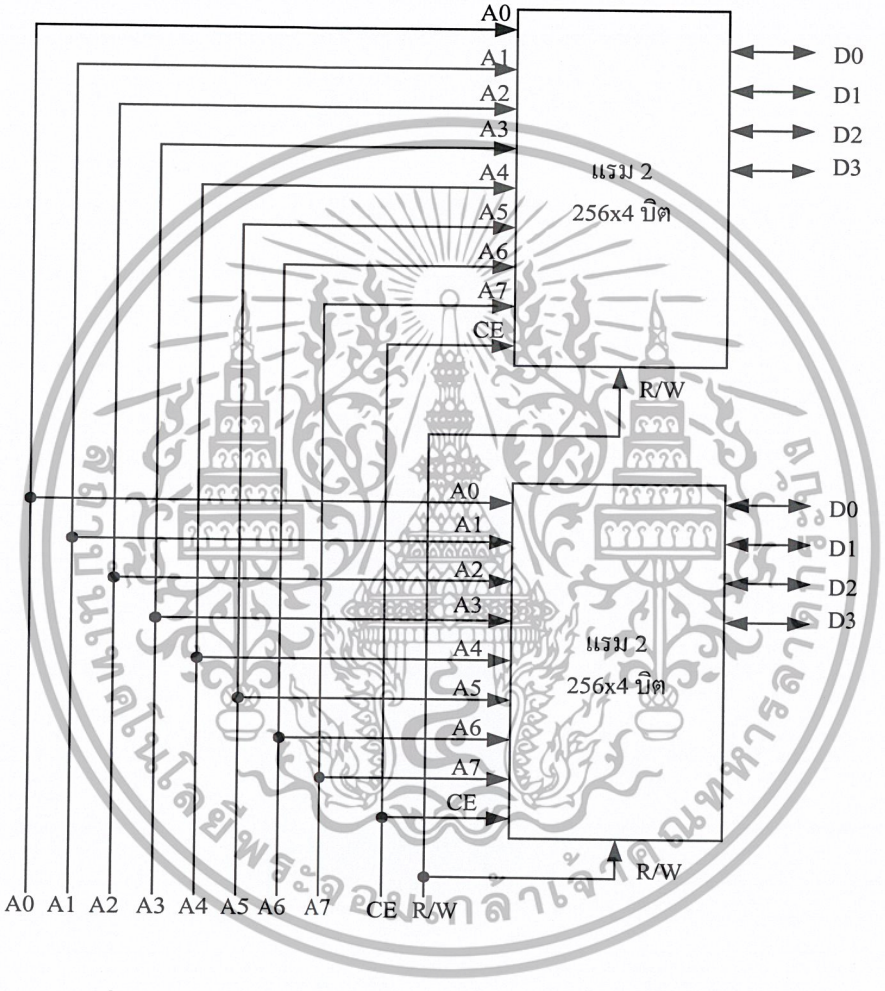
2.2) PROM ย่อมาจาก Programmable ROM เป็น ROM ที่เราสามารถเขียนข้อมูลลงไปได้โดยใช้เครื่องโปรแกรม ROM แต่การเขียนข้อมูลจะทำได้เพียงครั้งเดียว จะลบข้อมูลแล้วเขียนใหม่ไม่ได้อีก

2.3) EPROM ย่อมาจาก Erasable Programmable ROM เป็น ROM ชนิดที่เราคุ้นเคยกันมากที่สุด เราสามารถเขียนข้อมูลลงไปได้โดยใช้เครื่องโปรแกรม ROM และสามารถลบข้อมูลออกได้โดยการฉายแสงอัลตราไวโอเลตผ่านแว่นกระจกด้านบนของไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การขยายจำนวนบิตของหน่วยความจำ

ในการออกแบบจำนวนบิตในหน่วยความจำของไอซีอาจจะไม่เพียงพอแก่การใช้งานกล่าวคือ ถ้าจะต้องใช้ 8 บิต แต่ไอซีหน่วยความจำอาจมีเพียง 4 บิต เพราะฉะนั้นจึงต้องใช้ไอซีหน่วยความจำหลายตัวมาต่อรวมกันเพื่อให้ได้ขนาด 8 บิต ตามที่ต้องการ

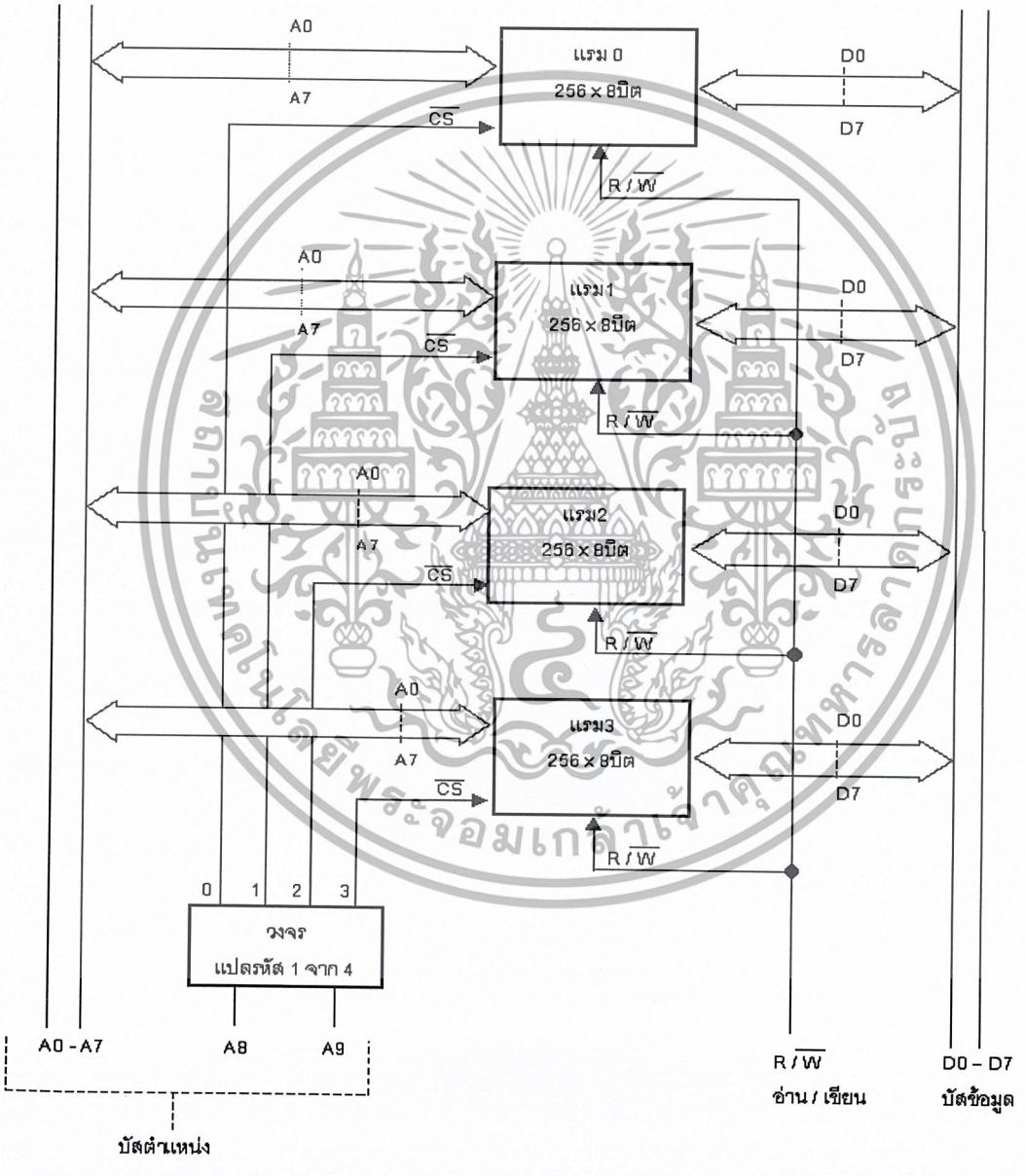


รูปที่ 2.16 สแตติกแรม 256x4 บิต สองตัวต่อให้มีขนาด 256x8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การขยายจำนวนหน่วยความจำ

หน่วยความจำขนาด 256 x 8 บิตหากมีความต้องการความจุมากกว่า 256 ตำแหน่ง จะต้องใช้ไอซีที่มีหน่วยความจำเพิ่มขึ้นอีก โดยต่อเข้าด้วยกัน ดังแสดงในรูปที่ 2.17 เพื่อป้องกันการอ่านหรือเขียนพร้อมกันทั้งสี่ตัว ต้องใช้เส้นสัญญาณเลือกไอซี ($\overline{CS}$) สัญญาณตำแหน่ง A8 และ A9 จะป้อนให้แก่หน่วยแปลรหัส เพื่อเลือกชิ้นไอซี



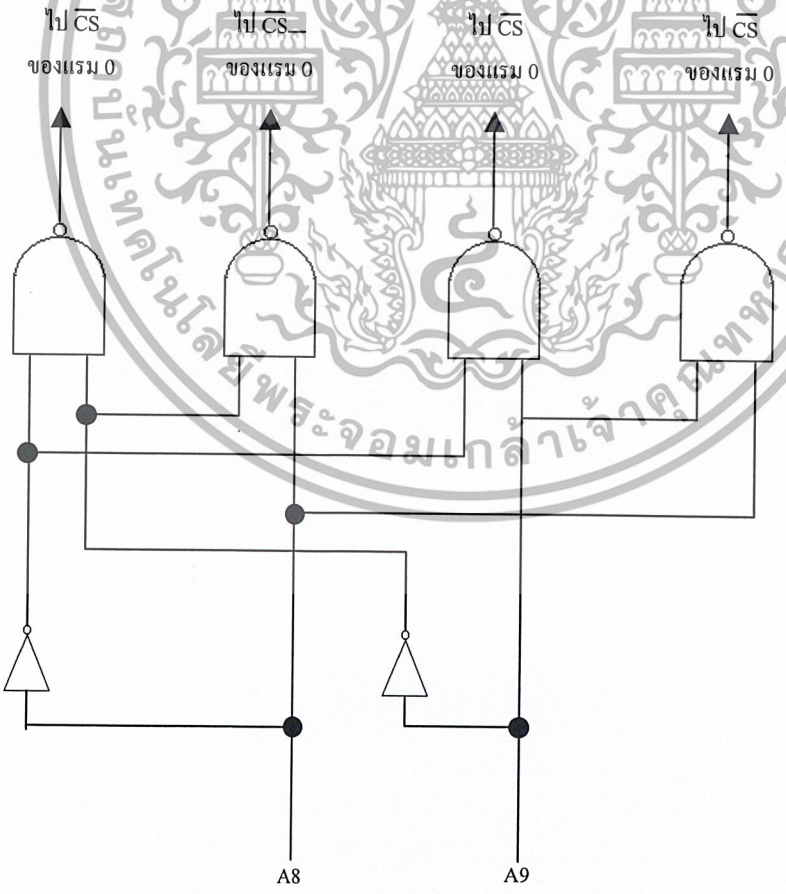
รูปที่ 2.17 การต่อแรม 256x 8 บิตเพื่อให้ได้ขนาด 1024x8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงการแปลงรหัส เพื่อเลือกชิ้นไอซี และรูปที่ 2.18 แสดงการใช้เกต NAND และ NOT ต่อเป็นวงจรภายในของหน่วยแปลรหัส

ตารางที่ 2.2 การแปลรหัส 1 จาก 4

บัสตำแหน่ง		สัญญาณออก จากหน่วยแปลรหัส				หมายเหตุ
A9	A8	3	2	1	0	
0	0	1	1	1	0	แรม 0 ถูกเลือก
0	1	1	1	0	1	แรม 1 ถูกเลือก
1	0	1	0	1	1	แรม 2 ถูกเลือก
1	1	0	1	1	1	แรม 3 ถูกเลือก



รูปที่ 2.18 การต่อเกต NAND และ NOT เป็นหน่วยแปลรหัส 1 จาก 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การจัดสรรพื้นที่ในหน่วยความจำ

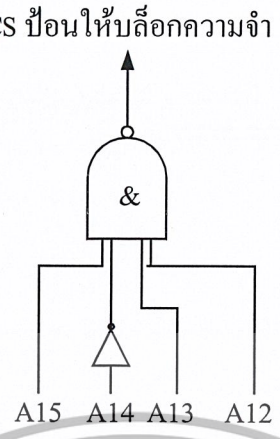
หากเราต้องการจะจัดบล็อกของแรมหรือรอมลงในบริเวณใดบริเวณหนึ่งของ 64 k ที่มีอยู่ เราจะทำได้โดยการแปลงรหัสตำแหน่งจากบัสตำแหน่ง 16 บิต ให้สอดคล้องกับบล็อกนั้นๆ ตัวอย่างเช่น เรามีบล็อกความจำขนาด 4k x 8 บิต ซึ่งต้องการที่จะต่อเข้าเป็นส่วนหนึ่งของเนื้อที่ 64k ที่มีอยู่ ดังแสดงในรูปที่ 2.19 หน่วยความจำ 4k x 8 บิต ต้องใช้บัสตำแหน่ง 12 เส้น A0- A11 เพราะ 2^{12} = 4096 (4k) หากเขียนเป็นเลขฐานสิบหก หมายถึงตำแหน่งจาก 000 ถึง FFF

เนื่องจากบัสตำแหน่งทั้งหมดมี 16 เส้น จึงจำเป็นต้องใช้บัสตำแหน่ง A12 – A15 ป้อนให้หน่วยแปลรหัสเพื่อนำสัญญาณออกจากหน่วยแปลรหัสนี้ไปเลือกบล็อกด้วย โดยใช้เส้นสัญญาณ $\overline{CS}$ ดังแสดงในรูปที่ 2.20



รูปที่ 2.19 การใช้สัญญาณ CS ช่วยเลือกบล็อกแรมหรือรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 วงจรถอดรหัส



รูปที่ 2.21 การจัดหน่วยความจำ 4k x 8 บิต ลงในเนื้อที่ทั้งหมด 64k

สมมติว่าบล็อกความจำ 4K x 8 บิตนี้ต้องการจะจัดให้อยู่ในเนื้อที่ระหว่าง B000 ถึง BFFF ดังนั้นสัญญาณ $\overline{CS}$ ในรูปที่ 2.19 จะต้องมีค่าระดับ 0 เมื่อบัสA15 – A12 มีค่าไบนารี 1011 รูปที่ 2.20 แสดงหน่วยแปลรหัสทำหน้าที่ดังกล่าว รูปที่ 2.21 แสดงการแปลรหัสจากบัสตำแหน่ง A12 – A15 เพื่อบ่งแต่ละบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

โครงงานชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 บอร์ด DSK TMS320C31 มีจุดเด่นหลายประการคือ เป็นตัวประมวลผลที่มีความเร็วสูง สามารถคำนวณทางคณิตศาสตร์ในระดับทศนิยม สามารถประมวลผลสัญญาณที่เวลาจริง (Real Time) และยังสามารถพัฒนาโปรแกรมบน PC (Personal Computer) แล้วส่งไปยังบอร์ด DSK TMS320C31 จึงสามารถแก้ไขและพัฒนาโปรแกรมได้อย่างสะดวกและใช้เวลาไม่มาก ซึ่งโครงงานนี้ประกอบไปด้วยส่วนของฮาร์ดแวร์ และส่วนของซอฟต์แวร์ ส่วนของฮาร์ดแวร์ประกอบไปด้วยวงจรการต่อหน่วยความจำภายนอก (ROM/RAM) วงจรตัวสุ่มตัวอย่างข้อมูล (A/D, D/A) ส่วนของซอฟต์แวร์ประกอบไปด้วยโปรแกรมที่ใช้ในใบงานทั้ง 6 ใบงาน

3.1 ส่วนของฮาร์ดแวร์

3.1.1 การต่อหน่วยความจำภายนอก (ROM/RAM)

ในการต่อหน่วยความจำภายนอก (ROM/RAM) จำเป็นต้องใช้ขาสัญญาณของ DSK TMS320C31 ดังนี้

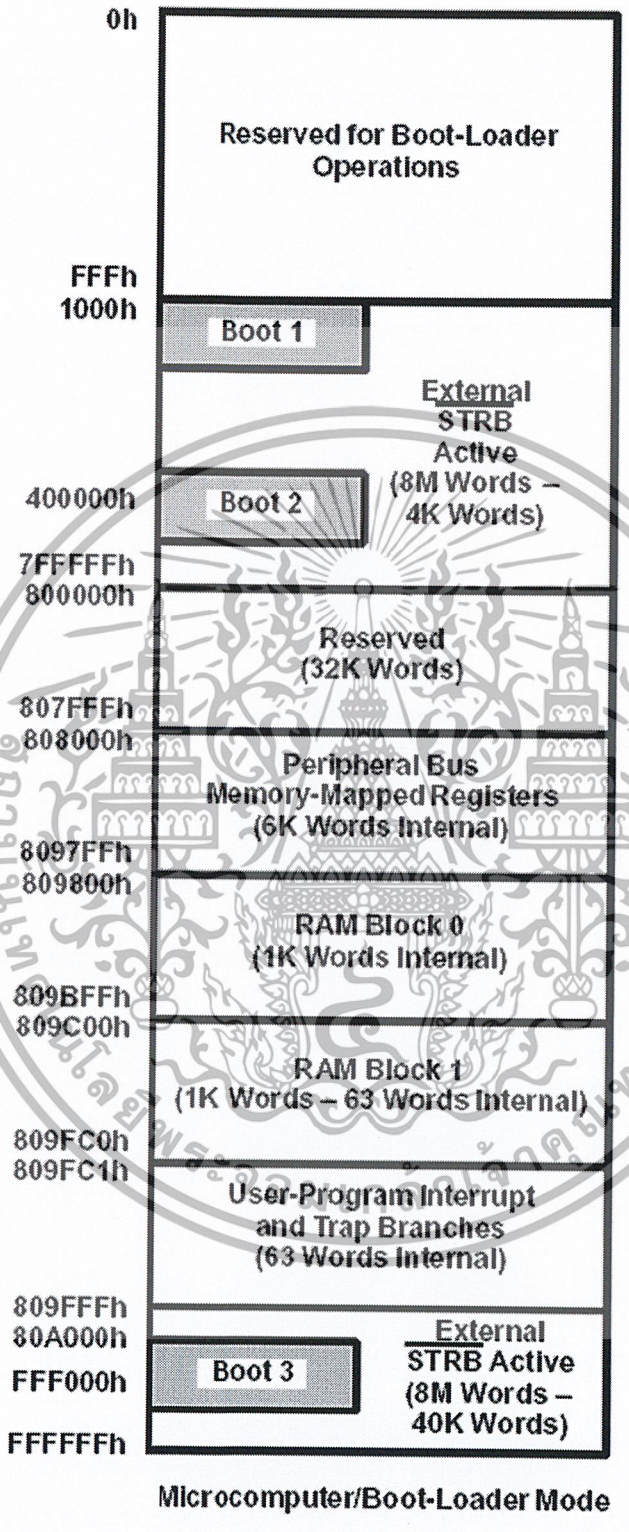
1) บัสข้อมูล (Data Bus) ขนาด 32 บิต

2) บัสตำแหน่ง (Address Bus) ขนาด 24 บิต

3) ขาสัญญาณควบคุม STRB, R/W

การต่อหน่วยความจำข้อมูลภายนอก (RAM) จะต้องเลือกตำแหน่งหน่วยความจำที่สงวนไว้ดังรูปที่ 3.1 ในโครงงานเลือกตำแหน่งที่ 200000H โดยขนาดหน่วยความจำข้อมูล 128k และหน่วยความจำโปรแกรมภายนอก (ROM) เลือก ใช้ตำแหน่งที่ 1000H โดยขนาดหน่วยความจำโปรแกรมภายนอก 64k

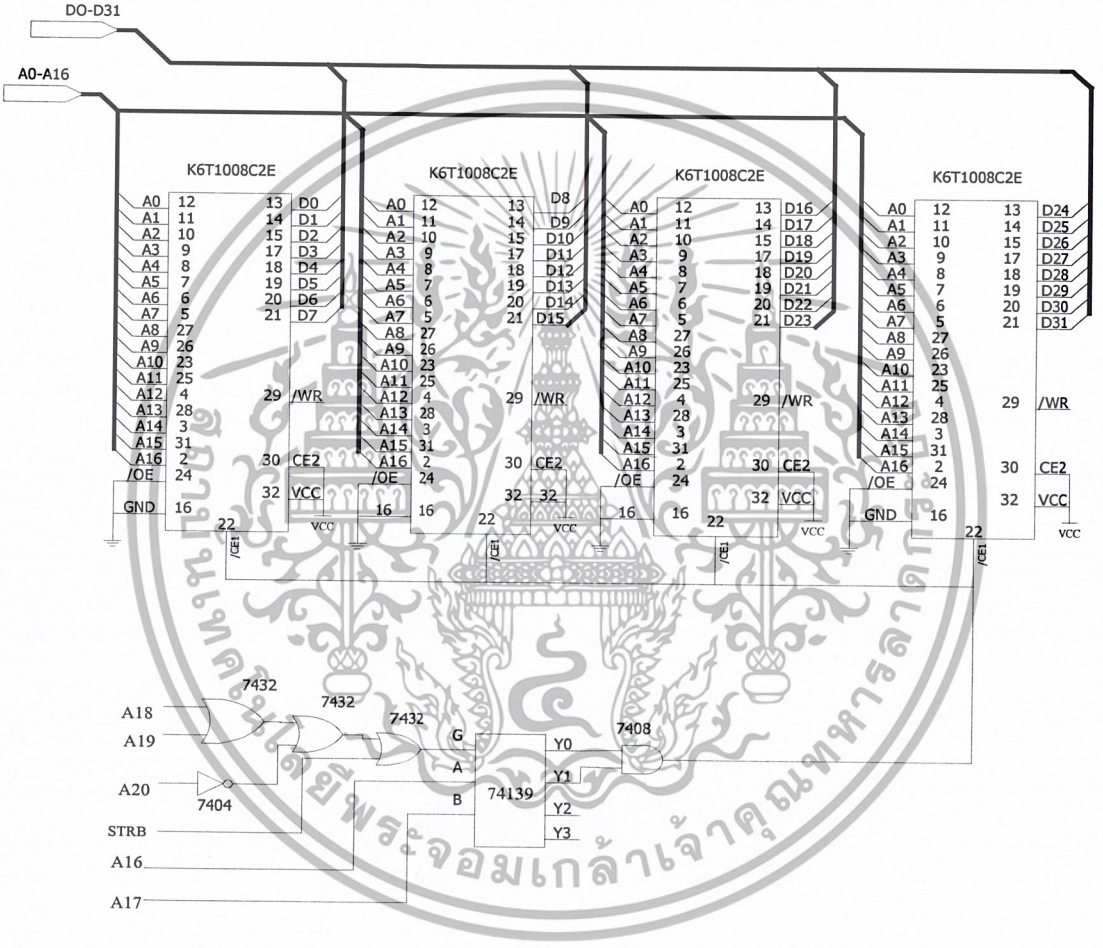
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ตำแหน่งที่ต่อหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหน่วยความจำข้อมูลภายนอก (RAM) ใช้ไอซีเบอร์ K6T1008C2E มีขนาด 8 บิต 128k Byte จึงต้องใช้ 4 ตัว โดยตัวที่ 1 ต่อ D0-D7 , โดยตัวที่ 2 ต่อ D8-D15 , โดยตัวที่ 3 ต่อ D16-D23 , โดยตัวที่ 4 ต่อ D24-D31 ใช้ไอซีเบอร์ 74739 ในการถอดรหัสที่ตำแหน่ง 200000H ส่วนหน่วยความจำโปรแกรมภายนอก (ROM) ใช้ไอซีเบอร์ 27C512 มีขนาด 8 บิต 64k Byte ใช้ 1 ตัว ต่อเข้ากับ D0-D7 ใช้ไอซีเบอร์ 74139 ในการถอดรหัสตำแหน่งที่ 1000H



รูปที่ 3.2 การต่อวงจรหน่วยความจำภายนอก (ROM/RAM)

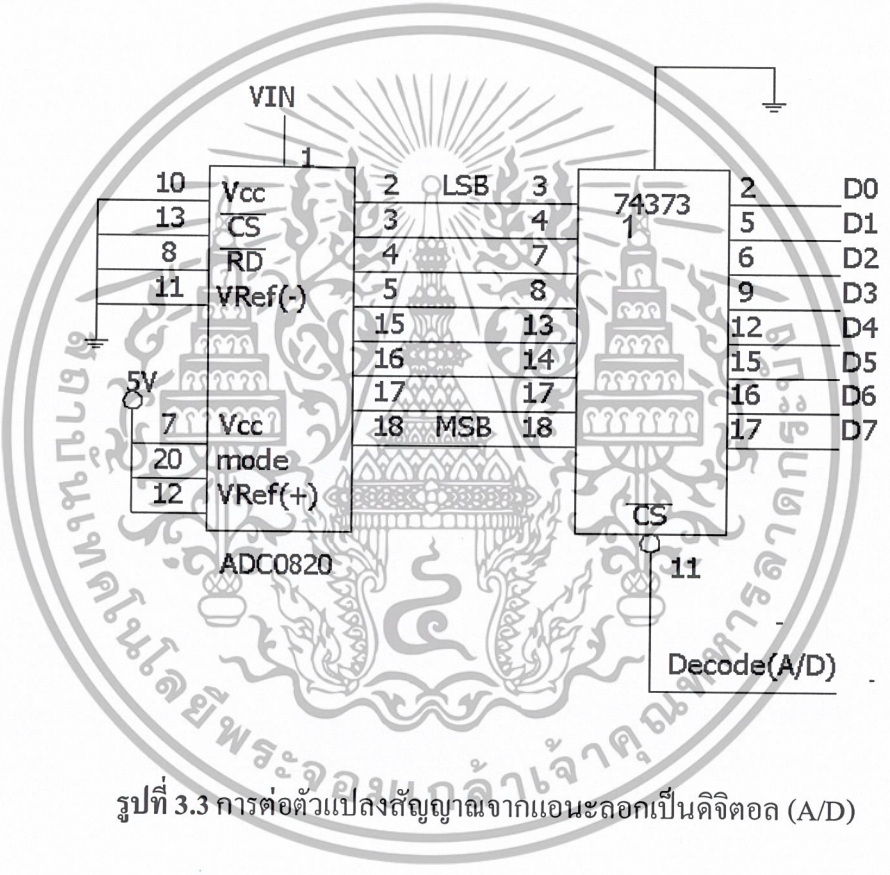
3.1.2 การเชื่อมต่อตัวแปลงสัญญาณจากแอนะลอกเป็นสัญญาณดิจิตอล (A/D) และตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก (D/A)

ในการเชื่อมต่อตัวแปลงสัญญาณจากแอนะลอกเป็นดิจิตอลและตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก จำเป็นต้องใช้ขาสัญญาณของ DSP TMS320C31 จะใช้ขาต่างของ DSP TMS320C31 คล้ายกับการต่อหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

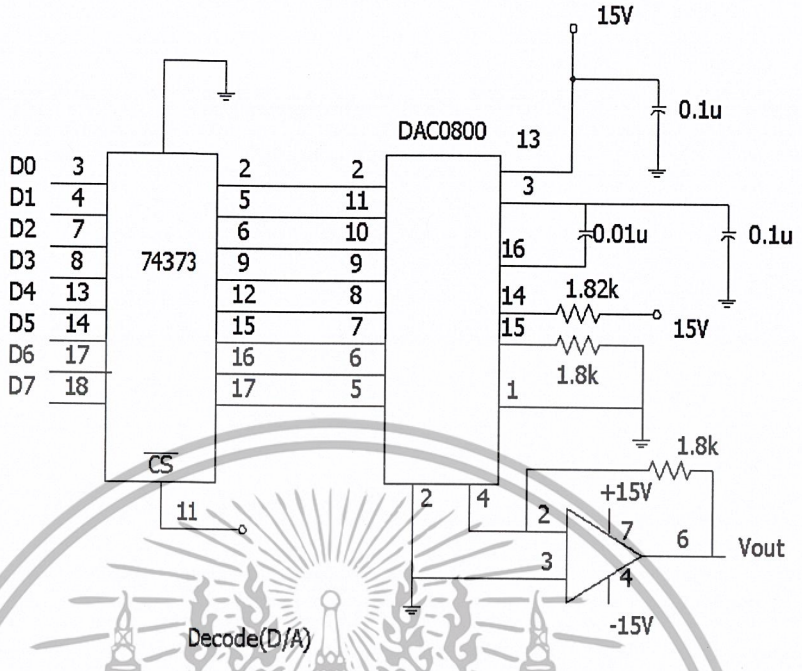
การเชื่อมต่อตัวแปลงสัญญาณจากสัญญาณแอนะลอกเป็นดิจิตอล (A/D) จะต้องเลือกตำแหน่งหน่วยความจำที่สงวนไว้ดังรูปที่ 3.1 เพื่อไม่ให้ซ้ำกับตำแหน่งของหน่วยความจำจึงเลือกตำแหน่งที่ 500000H ส่วนการเชื่อมต่อตัวแปลงสัญญาณจากดิจิตอลเป็นสัญญาณแอนะลอก (D/A) เลือกตำแหน่งที่ 600000H

สำหรับตัวแปลงสัญญาณจากแอนะลอกเป็นสัญญาณดิจิตอล (A/D) ใช้ไอซีเบอร์ ADC0820 มีขนาด 8 บิต โดยจะสุ่มตัวอย่างข้อมูล (Sampling) 8บิตล่างของข้อมูลที่เข้ามา ส่วนตัวแปลสัญญาณจากดิจิตอลเป็นสัญญาณแอนะลอก (D/A) ใช้ไอซีเบอร์ DAC0808

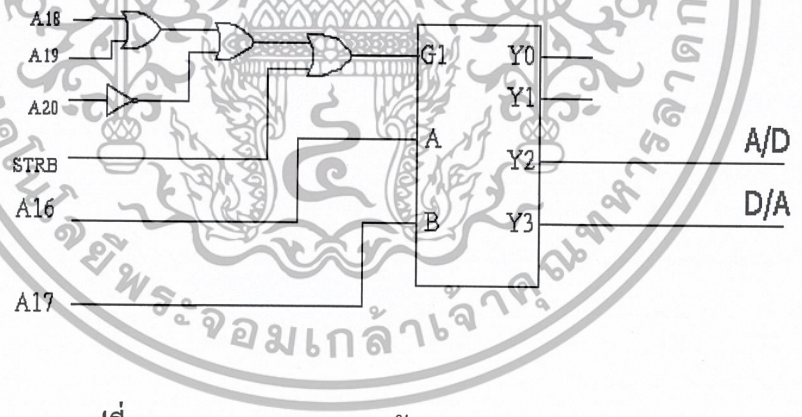


รูปที่ 3.3 การต่อตัวแปลงสัญญาณจากแอนะลอกเป็นดิจิตอล (A/D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ตัวแปลงสัญญาณจากดิจิตอลเป็นสัญญาณแอนะลอก



รูปที่ 3.5 การรวมวงจรถอดรหัส (Decoder)

3.2 ส่วนของซอฟต์แวร์

3.2.1 การสร้างคลื่นไซน์ (Sine Wave)

การออกแบบใบงานรูปคลื่นไซน์ (Sine wave) โดยใช้ TMS320C31 นั้นในการกำหนดความจะอาศัยหลักการสุ่มความถี่ (Sampling frequency) และการกำหนดจำนวนจุด (N) โดยที่

$$F = \frac{Fs}{N} \qquad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F = ความถี่เอาต์พุต

Fs = ความถี่สุ่ม

N = จำนวนจุด

จากหลักการของการมองค่าจากตารางซึ่งจำเป็นจะต้องคำนวณหาค่าไซน์ (sin) ของแต่ละจุด จากนั้นทำการสเกลค่าด้วย 1,000 และนำค่าที่สเกลแล้วไปเก็บในหน่วยความจำ จากนั้นเขียนโปรแกรมอ่านค่าจากหน่วยความจำ ส่งออกไปยัง D/A ของบอร์ด DSK

ตัวอย่าง

$N = 4$ จุด Sampling 8k

ดังนั้น $F = \frac{8000}{4}$

$F = 2k$

หาค่าไซน์ (sin) แต่ละจุดจาก

$A = \frac{360}{N}$

เมื่อ $N = 4$

$A = 90°$

$Sin\,1\,A = 1000$

$Sin\,2\,A = 0$

$Sin\,3\,A = -1000$

$Sin\,4\,A = 0$

นำค่าที่ได้เก็บไว้ในหน่วยความจำ แล้วเขียนโปรแกรมดึงค่าจากหน่วยความจำ ส่งไปยัง D/A ในกรณีทีต้องการความถี่สูงกว่านี้ สามารถทำได้ 2 วิธีคือ เปลี่ยนความถี่ Sampling และเปลี่ยนค่า N

3.2.2 การแปลงฟูเรียร์โดย Fast Fourier Transform (FFT)

การแปลงฟูเรียร์ (Fourier Transforms) เป็นเครื่องมือที่มีความสำคัญที่ใช้กันบ่อย ๆ ในระบบปฏิบัติงาน ทางสัญญาณดิจิตอล โดยจุดมุ่งหมายของการ Transform ก็คือเป็นการแปลงให้ทราบว่าจาก Time Domain เป็น Frequency domain หรือการแปลงกลับจาก Frequency domain เป็น Time Domain ผลจากการคำนวณจะเป็นเครื่องมือของ Fourier Transform ทำให้ทราบค่า ของ Fast Fourier Transform (FFT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fast Fourier Transform (FFT) เป็นวิธีการที่ใช้ในการคำนวณ Discrete Fourier Fourier Transforms (DFT) ให้เร็วขึ้นหรือให้มีลักษณะที่เป็นเรียลไทม์ขึ้น

ที่มาของ FFT โดยอาศัยหลักการของ DFT วิธีการเสนอโดย คูลีย์ และทูคีย์ ซึ่งได้กล่าวมาแล้วนั้น ความเป็นจริง การแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา x(m) ที่มีขนาด N จุด ออกเป็นสองลำดับสัญญาณที่มีความยาว N/2 จุดเท่ากัน ทั้งสองลำดับนี้หน้าชื่อเรียกว่า ลำดับสัญญาณคู่ และลำดับสัญญาณคี่โดยที่ลำดับสัญญาณคู่ เกิดจากการการเอาลำดับในตำแหน่งเป็นเลขคู่มาเรียงที่เหลือก็เป็นลำดับสัญญาณคี่ ดังนั้นถ้าเรานิยามให้ $x_2(m)$ เป็นลำดับคู่และลำดับที่ $x_0(m)$ ตามลำดับเพราะฉะนั้น

$$\begin{aligned} x_2(m) &= x(2m) && ; m = 0,1,\ldots,(N/2)-1 \\ x_0(m) &= x(2m+1) && ; m = 0,1,\ldots,(N/2)-1 \end{aligned} \qquad (3.2)$$

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

$$X(k) = \sum_{M=0}^{N-1} x_2(m)(W_N)^{km} + \sum_{M=0}^{N-1} x_0(m)(W_N)^{km}$$

$$= \sum_{M=0}^{(N-2)-1} x(2m)(W_N)^{2km} + \sum_{M=0}^{(N-2)-1} x(2m+1)(W_N)^{(2m+1)k} \qquad (3.3)$$

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 ของลำดับความยาว N/2 จุด เพราะฉะนั้น จัดพจน์ใหม่ได้ คือ

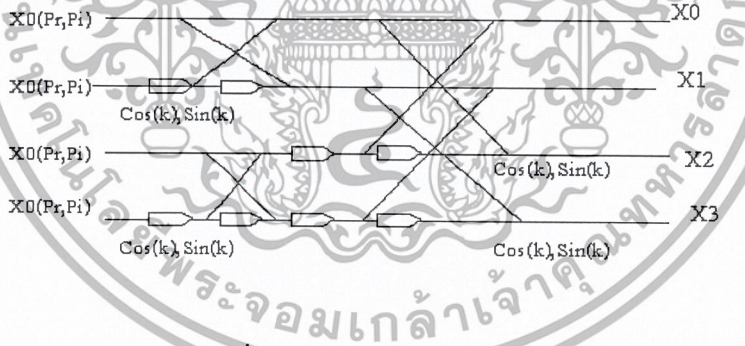
$$X(k) = \sum_{M=0}^{(N-1)-1} x_2(m)(W_{N/2})^{km} + \sum_{M=0}^{(N-2)-1} x0(m)(W_{N/2})^{km}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X(k) = X_1(k)+(W_n)^k X_2(k) \quad (3.4)$$

โดยที่ $x_1(k)$ และ $X_2(k)$ แทนผลการแปลง DFT ขนาด N/2 จุดของลำดับ $X_2(m)$ และX_0 (ตามลำดับ แสดงให้เห็นว่าการคำนวณ DFT ขนาด N จุดสามารถแบ่งคำนวณย่อยออกเป็น การ คำนวณ DFT ขนาด N/2 จุดสองอันดับได้และข้อสำคัญคือ การคูณจำนวนเชิงซ้อนจะลดลงเหลือ $2(N/2)^2=N^2/2$ ครั้ง ซึ่งจะเห็นว่าลดการคำนวณไปถึง 50 เปอร์เซ็นต์ โดยอาศัยหลักเดียวกัน ถ้าเราแบ่งทอนลำดับ $X_2(m)$ และ $X_0(m)$ ออกเป็นลำดับคู่และลำดับคี่ลงไปอีกตามลำดับในที่สุดเหลือเป็นลำดับขนาด 2 จุด หรือกล่าวได้ว่า การคำนวณการแปลง DFT ขนาด N จุดทำการซอยเพื่อลำดับ x(n0 ออกเป็นทีละครึ่งจนเหลือการคำนวณ DFT ขนาด 2 จุด โดยที่ Fast Fourier Transforms (FFT) สามารถแบ่งออกได้ 2 ชนิดใหญ่ๆ คือ ชนิดทอนเวลา (Decimation In Time หรือ DIT) และ ชนิดลดทอนความถี่ (Decimation In frequency หรือ DIF)

ส่วนของการทดลองในงานนี้จะใช้วิธีลดทอนเวลาดังรูปที่ 1 โดยใช้สมการที่ (3.5) และสมการที่ (3.6) ในการคำนวณ



รูปที่ 3.6 วิธีการลดทอนเวลา

$$\Pr n = [\Pr + QrCos(k) + QiSin(k)] + j[Pi + QiCos(k) - Sin(k)] \quad (3.5)$$

$$Qm = [\Pr - QrCos(k) - QiSin(k)] + j[Pi - OiCos(k) + Sin(k)] \quad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การเชื่อมต่อหน่วยความจำภายนอก

หน่วยความจำในระยะแรกจะเป็นแหวนแม่เหล็ก (Core Memory) เป็นกระบอกแม่เหล็ก (Magnetic Drum) ซึ่งเป็น อุปกรณ์แม่เหล็กไฟฟ้าและเชิงกลไฟฟ้า ทำให้การติดต่อกับ CPU ทำได้ยาก วงจรซับซ้อนมาก การทำหน่วยความจำเป็นไอซีได้จึงเป็นเรื่องสำคัญทำให้วงจรในคอมพิวเตอร์ง่ายลงมาก

เมื่อเทคโนโลยีในการทำ MOS ไอซีเจริญขึ้น ก็สามารถทำไอซีที่มีหน่วยความจำที่มีความจุสูงขึ้น สามารถใช้งานในทางปฏิบัติได้ดีขึ้น จึงทำให้ไมโครคอมพิวเตอร์มีขนาดเล็กลงและสามารถใช้งานได้ดีเยี่ยม

3.2.4 การใช้งานแผงวงจร DSK TMS320C31 เบื้องต้น

บอร์ด DSK TMS320C31 สามารถใช้ภาษาในการเขียนอยู่ถึง2 ภาษา จึงต้องมีการคอมไพล์ให้เป็นภาษาเครื่องเพื่อให้บอร์ดDSKทำงานตามโปรแกรมที่ป้อนให้กับบอร์ด DSK TMS320C31 การเขียนโปรแกรมจะเขียนได้จากคอมพิวเตอร์ โดยใช้หน้าต่างของDOS จึงทำให้มีคำสั่งที่ยุ่งยากพอสมควร แต่สามารถเข้าใจง่าย

3.2.5 การใช้ชุดคำสั่งแบบง่าย

ในการศึกษาชุดคำสั่งนั้น ต้องเริ่มต้นจากโปรแกรมง่ายๆ เพื่อที่จะได้เข้าใจหลักการของชุดคำสั่งได้อย่างรวดเร็ว ในใบงานนี้จึงขอยกตัวอย่างการใช้โปรแกรมการคูณแมททริก ดังนี้

โดยการใช้โปรแกรม การคูณแมทริกขนาด(3x3)*(3x1)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} (1x1) + (2x2) + (3x3) \\ (4x1) + (5x2) + (6x3) \\ (7x1) + (8x2) + (9x3) \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 32 \\ 50 \end{bmatrix} \qquad \text{Ans}$$

3.2.6 วงจรกรองแบบเอฟไออาร์(FIR Filter)

สมการที่(1)เป็นสมการที่ใช้ออกแบบ

$$y(n) = \sum_{k=0}^{M} b_x x(n-k) \qquad (3.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าสมการนี้เป็นสมการของดิจิตอลฟิลเตอร์ที่มี a = 0 โดย b จะมีจำนวนเท่ากับจำนวนอิมพัลส์เรสอนส์ ของฟิลเตอร์ ตามที่ได้ออกแบบไว้

จากสมการของวงจรกรองเอฟไออาร์ จะเห็นได้ว่าเป็นรูปแบบของสมการคอนโวลูชัน (Convolution Equation) ดังสมการที่(2)

$$y(n) = \sum_{k=0}^{M} \text{h(n-k) } xz^{-1} \tag{3.8}$$

การหาสัมประสิทธิ์ของ FIR Fillter

$$y(n) = \begin{cases} \frac{1}{\pi n}(\sin \omega 2n - \sin \omega 1n) & ; n \neq 0 \\ \frac{1}{\pi}(\omega 1 - \omega 2); n = 0 \end{cases} \tag{3.9}$$

เมื่อเราได้ค่า h(n) แต่ละตัว จากนั้นต้องทำการคูณด้วยค่า Windows Hamming จากสมการ

$$\omega(\text{n}) = \begin{cases} 0.5 + 0.5\cos(\frac{\pi n}{1 + (M/2)}) & ; -M/2 \leq n \leq M/2 \\ 0; |n| > M/2 \end{cases}$$

$$\text{h(n)} = \text{r(n)} * \omega(\text{n}) \tag{3.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

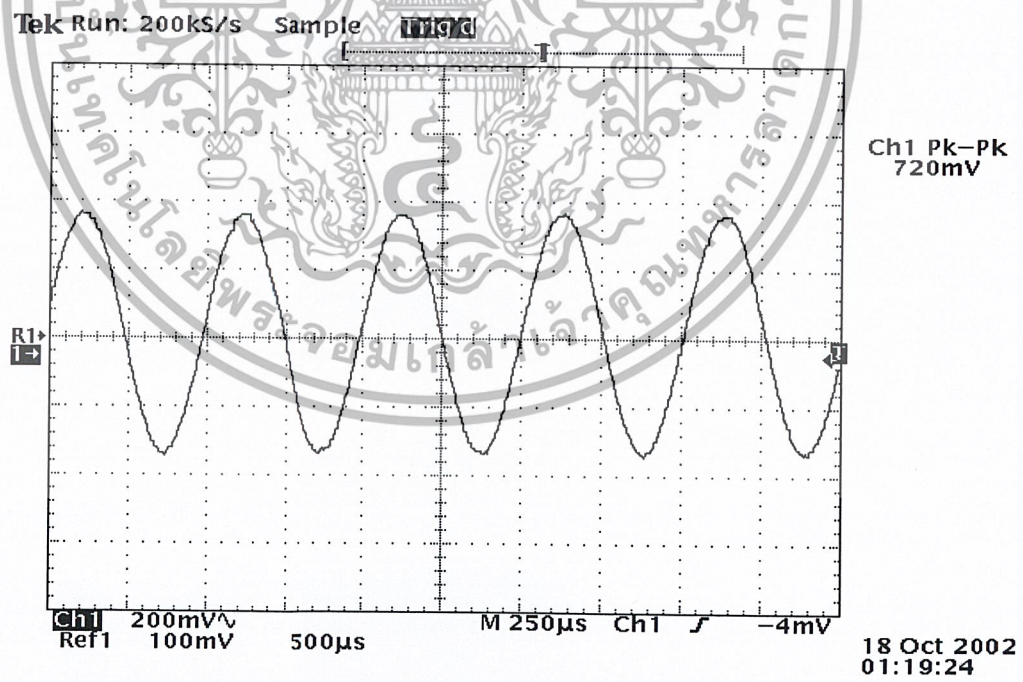
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงผลการทดลองต่างๆในโครงงาน อาทิ เช่น การทดลองสร้างคลื่นไซน์ การทดลองการเชื่อมต่อหน่วยความจำ การทดลองการแปลงฟูเรียร์โดยวิธีการ Fast Fourier transform [FFT] เป็นต้น

4.1 การกำเนิดสัญญาณไซน์ (Sine Wave)

4.1.1 ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์
2) ป้อนโปรแกรมกำเนิดคลื่นไซน์
3) คอมไพล์และรันโปรแกรม
4) บันทึกผลการทดลอง



Volt/Div= 200mv Time/Div= 250us f= 2 kHz

รูปที่ 4.1 ผลการทดลองการกำเนิดสัญญาณไซน์แบบจำนวน N = 4 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ทดลองเปลี่ยนค่า ในบรรทัดที่ 9 ของโปรแกรม Sine4p.asm เป็น 8 จุด โดยใช้ค่าดังนี้

$$\frac{360}{8} = 45°$$

ฉะนั้นระยะห่างของจุดจะเท่ากับ 45° นำค่าองศาประจำจุดต่างมาอยู่ในรูปของ Sine y (y คือ ค่าขององศาประจำจุดต่างๆ) แล้วนำมาคูณด้วย 1,000 เนื่องจาก TMS320C31 เป็น Fired Point

จุดที่ 1 เท่ากับ 0

จุดที่ 2 เท่ากับ 707

จุดที่ 3 เท่ากับ 1000

จุดที่ 4 เท่ากับ 707

จุดที่ 5 เท่ากับ 0

จุดที่ 6 เท่ากับ -707

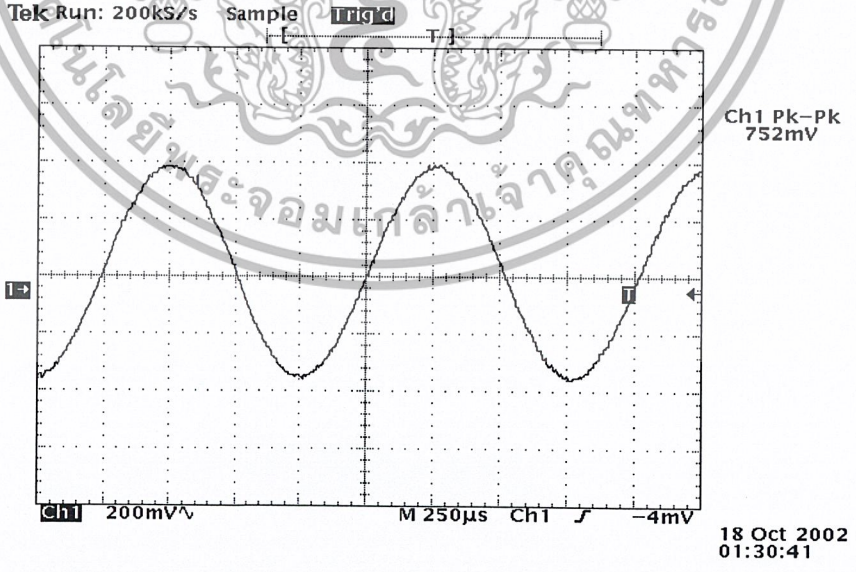
จุดที่ 7 เท่ากับ -1000

จุดที่ 8 เท่ากับ -707

6) นำค่าไปใส่ในโปรแกรม Sine4p.asm ในบรรทัดที่ 9 และเปลี่ยนค่า Length ในบรรทัดที่ 10 เป็น 8 เพื่อทำการวน 8 รอบ

7) ทำการคอมไพล์และรันโปรแกรม

8) บันทึกผลการทดลอง



Volt/Div= 200mv Time/Div= 250us f= 1 kHz

รูปที่ 4.2 ผลการทดลองการกำเนิดสัญญาณไซน์แบบจำนวน N = 8 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

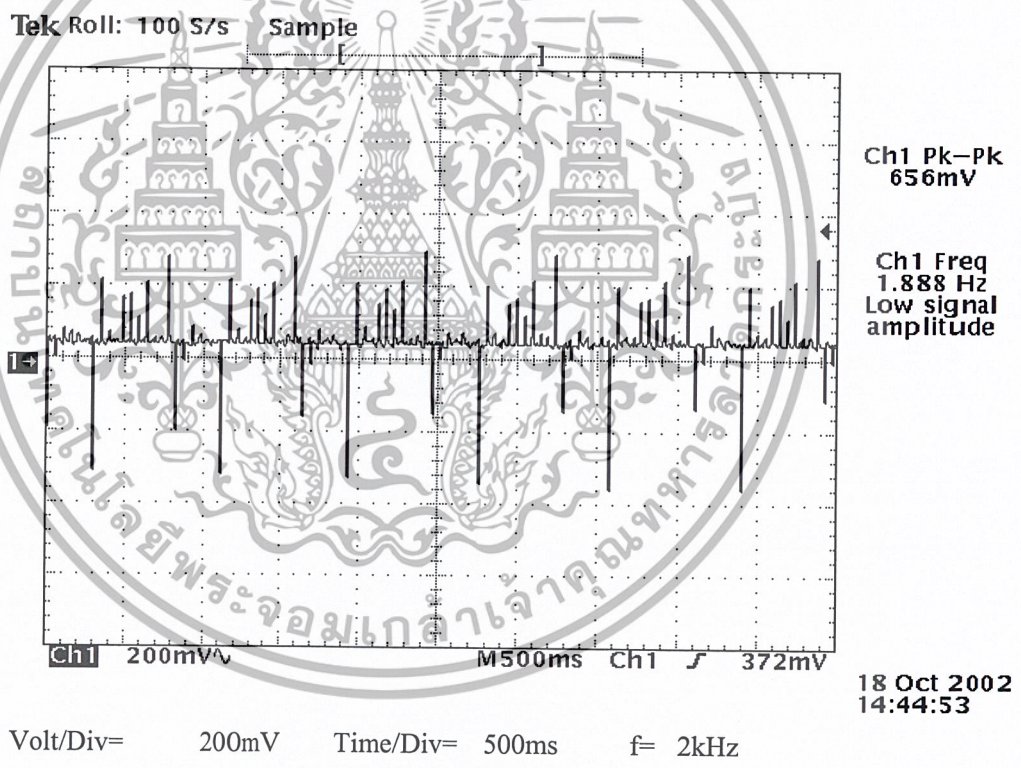
4.1.2 สรุปผลการทดลอง

ความถี่(f)จะขึ้นอยู่กับจำนวน N และความถี่สุ่ม(Fs) คือ N จะแปลผกผันกับความถี่(f)และ และความถี่สุ่ม(Fs) จะแปลผันกับความถี่(f)

4.2 การทดลองการแปลงฟูเรียร์โดยวิธี Fast Fourier Transform [FFT]

4.2.1 ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองเข้ากับอุปกรณ์
2) ป้อนโปรแกรม Fast Fourier Transfrom
3) Sine wave ให้กับบอร์ด DSK TMS320C31 บันทึกผลการทดลอง



Volt/Div= 200mV Time/Div= 500ms f= 2kHz

รูปที่ 4.3 ผลการทดลองการแปลงฟูเรียร์โดยวิธี Fast Fourier Transform [FFT]

4.2.2 สรุปผลการทดลอง

จากการรันโปรแกรม Fast Fourier Transform (FFT) จะทำให้ทราบถึงวิธีการเขียนโปรแกรมและการแสดงผลของข้อมูลที่ตำแหน่ง Data Memory ที่กำหนดไว้ในโปรแกรมและสามารถทำการเปลี่ยนค่าอินพุตที่เข้ามายัง FFT โดยไม่ต้องใช้วิธีการคำนวณทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการเชื่อมต่อหน่วยความจำภายนอก

4.3.1ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์

2) การทดสอบหน่วยความจำภายนอก โดยการกดคำสั่งdsk3d <Enter>

3) พิมพ์คำสั่ง mem 0x100000 เนื่องจากหน่วยความจำภายนอกต่ออยู่ที่ตำแหน่งแอดเดรส 100000H

4) ที่ตำแหน่ง 100000H จะแสดงข้อมูลขณะนั้นอยู่สังเกตค่า บันทึกผลการทดลอง

100000H 0000

5) กดALT+m หรือ กดF1 เพื่อดูคำอธิบายจากนั้นทำการป้อนค่า โดยใช้คำสั่ง mm ใน Debuger เช่น mm 0x100000 จะเป็นการใส่ค่าแอดเดรสที่ 100000H

6)ใส่ค่าดังต่อไปนี้ ตำแหน่ง 100000H ใส่ FFFF, ตำแหน่งที่ 100002H ใส่ FC ,ตำแหน่งที 100005H ใส่ 9999, ตำแหน่งที่ 100009.ใส่ 77C9 บันทึกผลการทดลอง

100000H เท่ากับ FFFF

100002H เท่ากับ FC

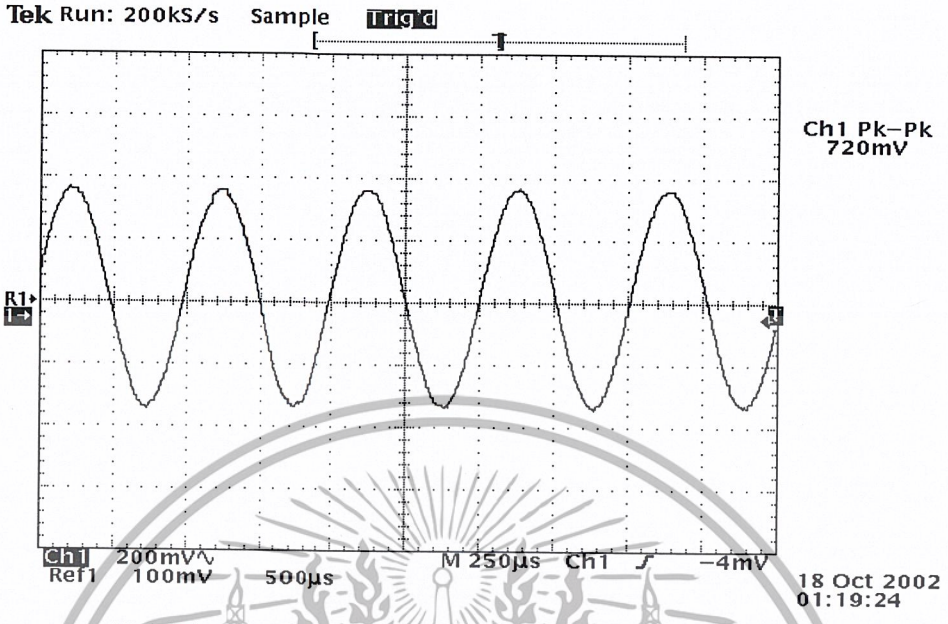
100005H เท่ากับ 9999

100009H เท่ากับ 77C9

7) โหลดโปรแกรม SIN_RAM.dsk สังเกตว่าจะเป็นตำแหน่งของหน่วยความจำภายนอก คือ ตำแหน่ง 100000H

8) บันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Volt/Div= 200mv Time/Div= 250us f= 2 kHz

รูปที่ 4.4 ผลการทดลองการรันโปรแกรม SIN_RAM.dsk ผ่านหน่วยความจำภายนอก

4.3.2 สรุปผลการทดลอง

จากการทดลองจะเห็นได้ว่าเมื่อต้องการเพิ่มหน่วยความจำให้กับบอร์ด DSK TMS320C31 จะต้องมีการทดสอบหน่วยความจำนั้นตามใบงานข้างต้น

4.4 การทดลองการใช้งานบอร์ด DSK TMS320C31 เบื้องต้น

4.4.1 ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์

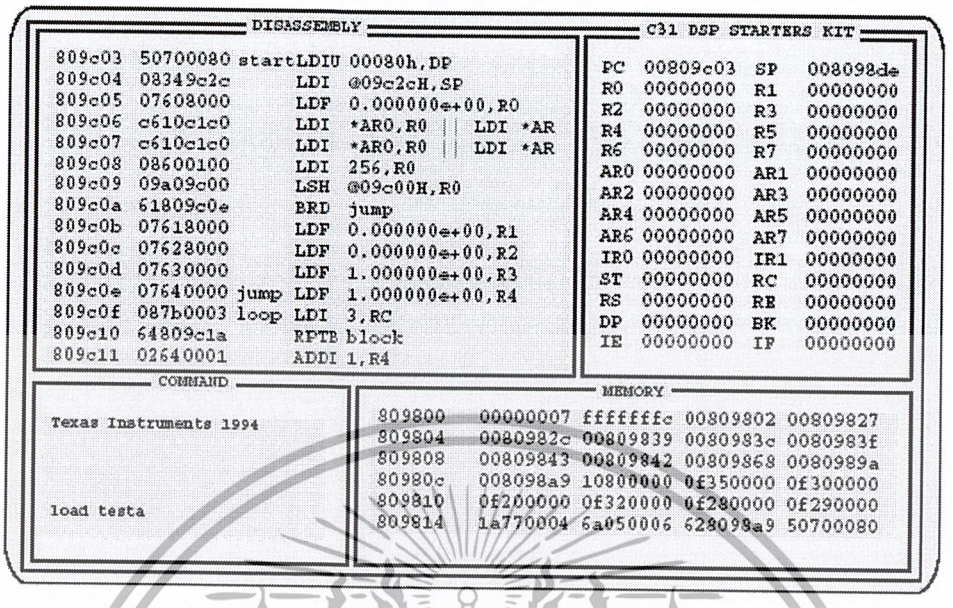
2) เข้าไปที่ Dsktoos ใช้คำสั่ง DSK3A ตามด้วยชื่อไฟล์นามสกุล .asmในใบงานนี้ใช้ไฟล์ชื่อ sine4p.asm (DSK3A เป็นการแปลงไฟล์ DSK เพื่อใช้โหลดลง DSK TMS320C31)

3) ใส่คำสั่ง DSK 3d เพื่อเข้าสู่หน้าต่าง Debuger ดังรูปที่ 4.5

4) ทำการ Reset บอร์ด DSK TMS320C31 โดยใช้คำสั่ง resetในหน้าต่าง Debuger

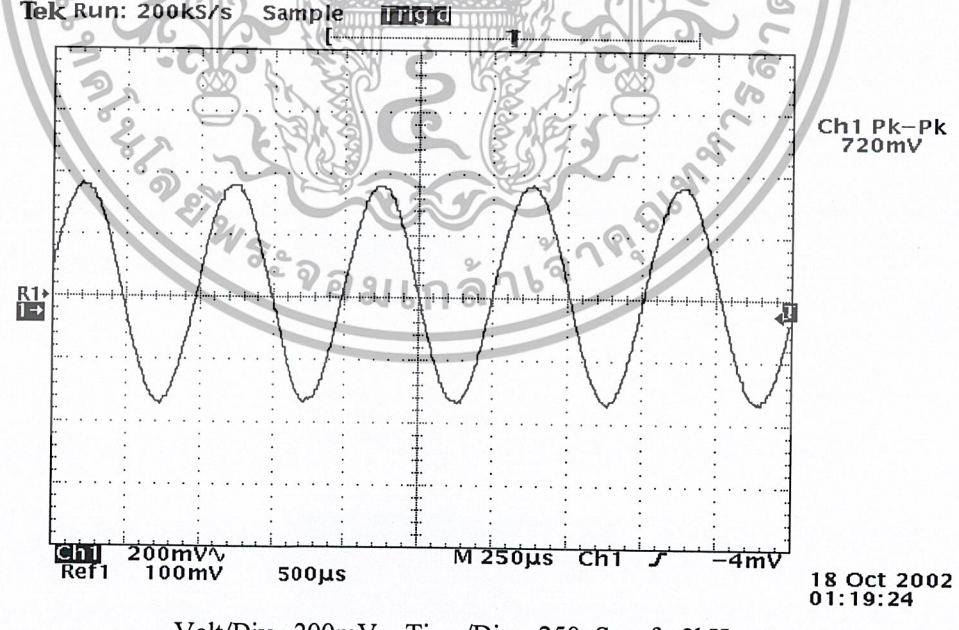
5) ใช้คำสั่ง Load ชื่อ .dsk (จากข้อ2) ดังนี้ Load sine4p.dsk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 หน้าจอ Debugger

6) ใส่คำสั่ง RUN เพื่อรัน โปรแกรม หรือกด F5 บันทึกผลการทดลอง



Volt/Div= 200mV Time/Div= 250uS f= 2kHz

รูปที่ 4.6 ผลการทดลองการรัน โปรแกรม Sin4p.asm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 สรุปผลการทดลอง

จากการทดลองรันโปรแกรมทำให้ทราบคำสั่งในการใช้งานเบื้องต้นของ Debugger ของบอร์ด DSK TMS320C31 บนหน้าต่าง Debugger

4.5 การทดลองการใช้ชุดคำสั่งแบบง่าย

4.5.1 ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองเข้ากับอุปกรณ์

2) ป้อนโปรแกรมการคูณแมทริกซ์

3) ทำการคอมไพล์ โปรแกรมตามใบงานที่ 1

4) ทำการโหลดโปรแกรมในหน้าต่าง Debugger โปรแกรมจะเริ่มที่ตำแหน่ง 809900H และข้อมูลจะเริ่มที่ตำแหน่ง 809c00H ในบรรทัดที่ 5 และ 6 จะอยู่ที่ตำแหน่งเริ่มต้นของตำแหน่งข้อมูล ดังนั้น คำตอบจะเริ่มที่ตำแหน่ง 809c0eH เราต้องการดูค่าที่ตำแหน่ง 809c0eH โดยการใช้คำสั่ง memd 0x809c0e (เป็นคำสั่งที่ใช้ดูเลขจำนวนเต็ม)

5) กด F8 เพื่อทำการรันทีละสเต็ป ทำการกดไปเรื่อยๆ สังเกตการเปลี่ยนแปลงค่าที่ตำแหน่ง 809c0eH

6) ทำการบันทึกค่า ที่อยู่ในตำแหน่งดังนี้

809c0eH = 14

809c0fH = 32

809c10H = 50

4.5.2 สรุปผลการทดลอง

จากการทดลองทำให้ทราบชุดคำสั่งอย่างง่าย เช่น LDI 2,AR4 เป็นการเก็บค่า 2ไว้ในรีจีสเตอร์ AR4

4.6 การทดลองเอฟไออาร์ฟิลเตอร์ (FIR Filter)

4.6.1 ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองเข้ากับอุปกรณ์

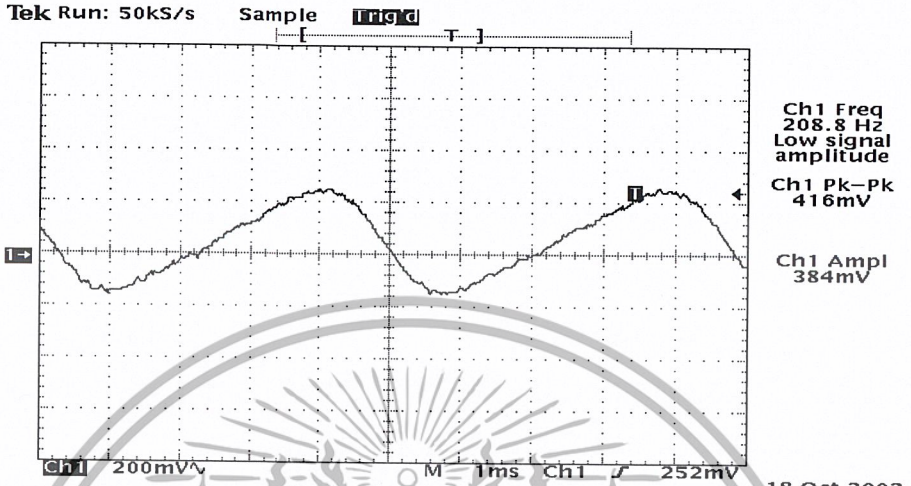
2) ป้อนโปรแกรมเอฟไออาร์ฟิลเตอร์ (Lp11sim.asm)

3) ปรับความถี่อินพุตที่ 200Hz 3Vp-p

4) คอมไพล์และรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) บันทึกผลการทดลอง



Volt/DIV = 200mV TIME/DIV = 1 ms f = 200Hz

รูปที่ 4.7 ผลการทดลองการรันโปรแกรม FIR Filter

4.6.2 สรุปผลการทดลอง

จากการทดลองใช้ชิพ TMS320C31 มาประยุกต์ใช้เป็นวงจรกรองความถี่ต่ำผ่านชนิด FIR ซึ่งเป็นการสะดวกเพราะสามารถสร้างจากซอฟต์แวร์แทนการต่อวงจรที่ยุ่งยากและซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางการแก้ไขและพัฒนา

5.1 บทสรุป

ชุดทดลองการประมวลผลสัญญาณเชิงเลข สร้างขึ้นเพื่อเป็นสื่อการศึกษาสำหรับผู้ที่สนใจเกี่ยวกับการประมวลผลสัญญาณเชิงเลขในโครงงานนี้ได้ใช้ชิพตัวประมวลผลสัญญาณเชิงเลขเบอร์ TMS320C31 ในการประมวลผลสัญญาณเชิงเลขซึ่งขอบเขตของโครงงานนี้ คือ มีใบงานการทดลองไม่น้อยกว่า 6 ใบงาน ซึ่งในโครงงานนี้มีใบงานการทดลองทั้งหมด 6 ใบงาน ได้แก่

1) การใช้งานบอร์ด DSK TMS320C31 เบื้องต้น
2) การใช้ชุดคำสั่ง
3) การเชื่อมต่อหน่วยความจำ
4) การสร้างคลื่นไซน์
5) การแปลงฟูเรียร์โดยวิธี Fast Fourier Transfrom (FFT)
6) เอฟไออาร์

5.2 ปัญหาและแนวทางการแก้ไข

1) **ปัญหา** การศึกษาและการเขียนโปรแกรมควบคุมการทำงานของ DSK เป็นไปได้ช้าเนื่องจากความยุ่งยากและซับซ้อนของคำสั่ง

แนวทางแก้ไข ต้องใช้ระยะเวลาในการศึกษาด้านซอฟต์แวร์มาก

2) **ปัญหา** การทดลองเบื้องต้นโดยการต่อวงจรลงบนโฟโต้บอร์ดไม่มีความแน่นอน

แนวทางแก้ไข ต่อลงบนแผ่นพิมพ์วงจรอเนกประสงค์แทนการต่อลงบนโฟโต้บอร์ด

3) **ปัญหา** การจัดซื้ออุปกรณ์บางตัวทำได้ยากเนื่องจากไม่สามารถหาซื้อได้

แนวทางแก้ไข ลดขนาดของอุปกรณ์บางตัวลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางในการพัฒนา

1) สร้างและออกแบบใบงานเพิ่มมากขึ้น เพื่อเป็นแนวทางในการศึกษาวิชาการประมวลผลสัญญาณเชิงเลขให้เข้าใจมากยิ่งขึ้น

2) ควรทดลองออกแบบโปรแกรมที่มีการทำงานในลักษณะเดิม แต่ใช้จำนวนคำสั่งน้อยกว่าเดิม

3) ต่อหน่วยความจำเพิ่มขึ้น เพื่องานที่ต้องการหน่วยความจำมากๆ เช่น การบีบอัดข้อมูลเสียง เป็นต้น

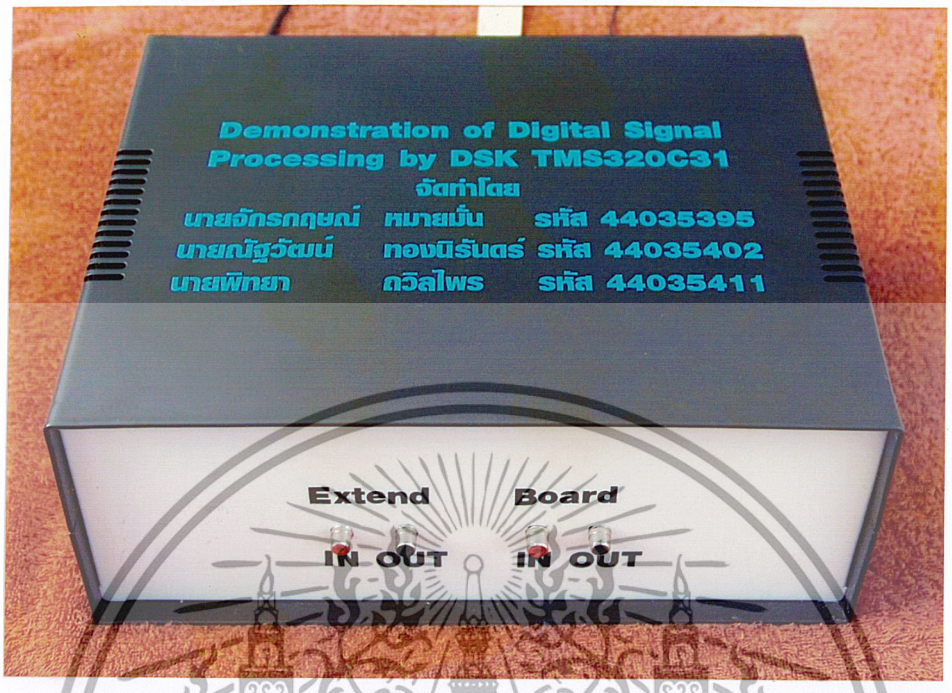


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

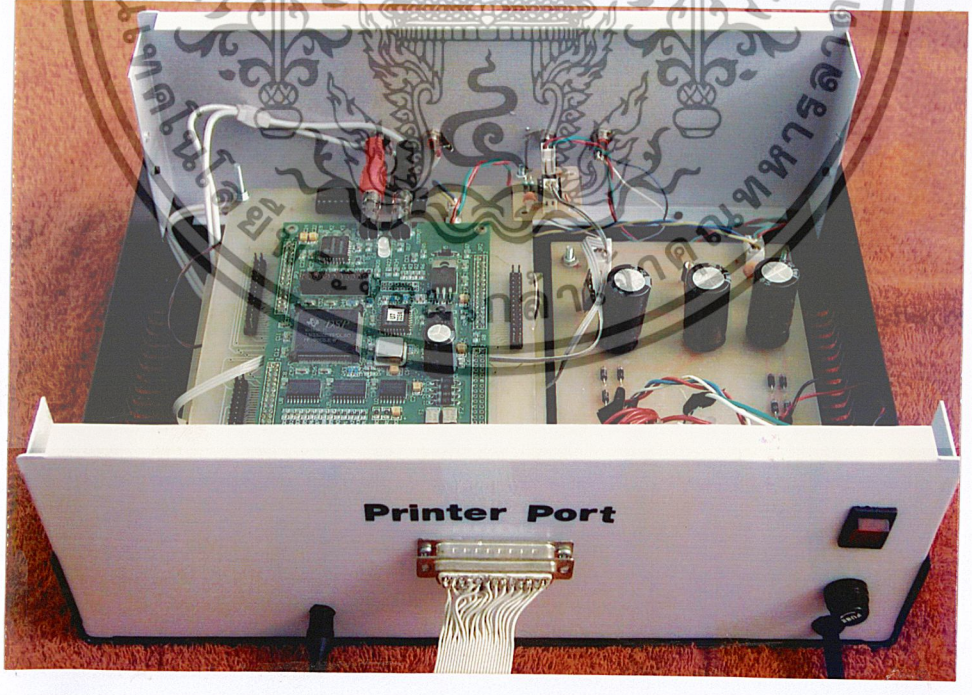


ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 ด้านนอก



รูปที่ ก.2 ชุดทดลองการประมวลผลสัญญาณเชิงเลขด้วย DSK TMS320C31 ด้านใน

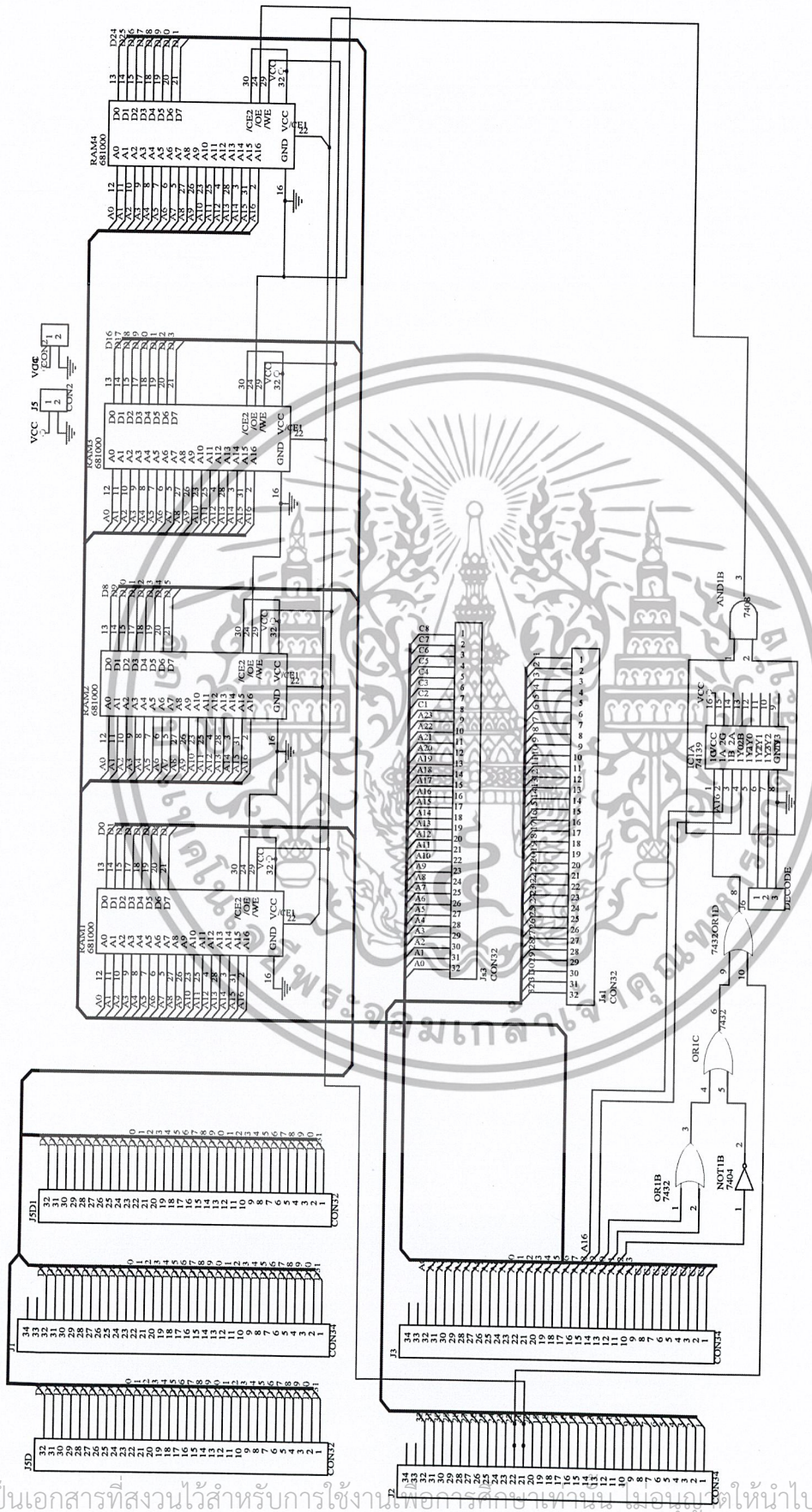
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

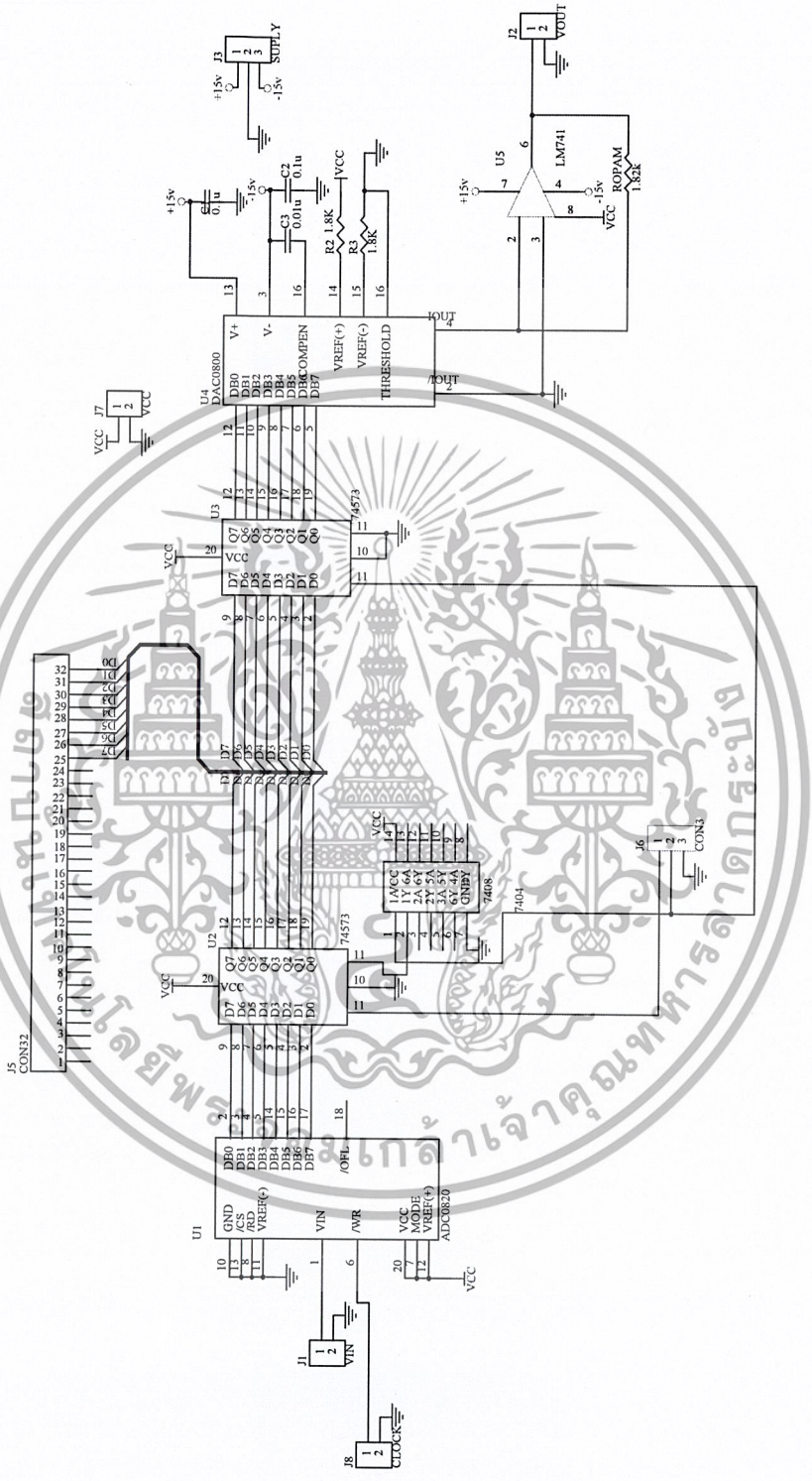
วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



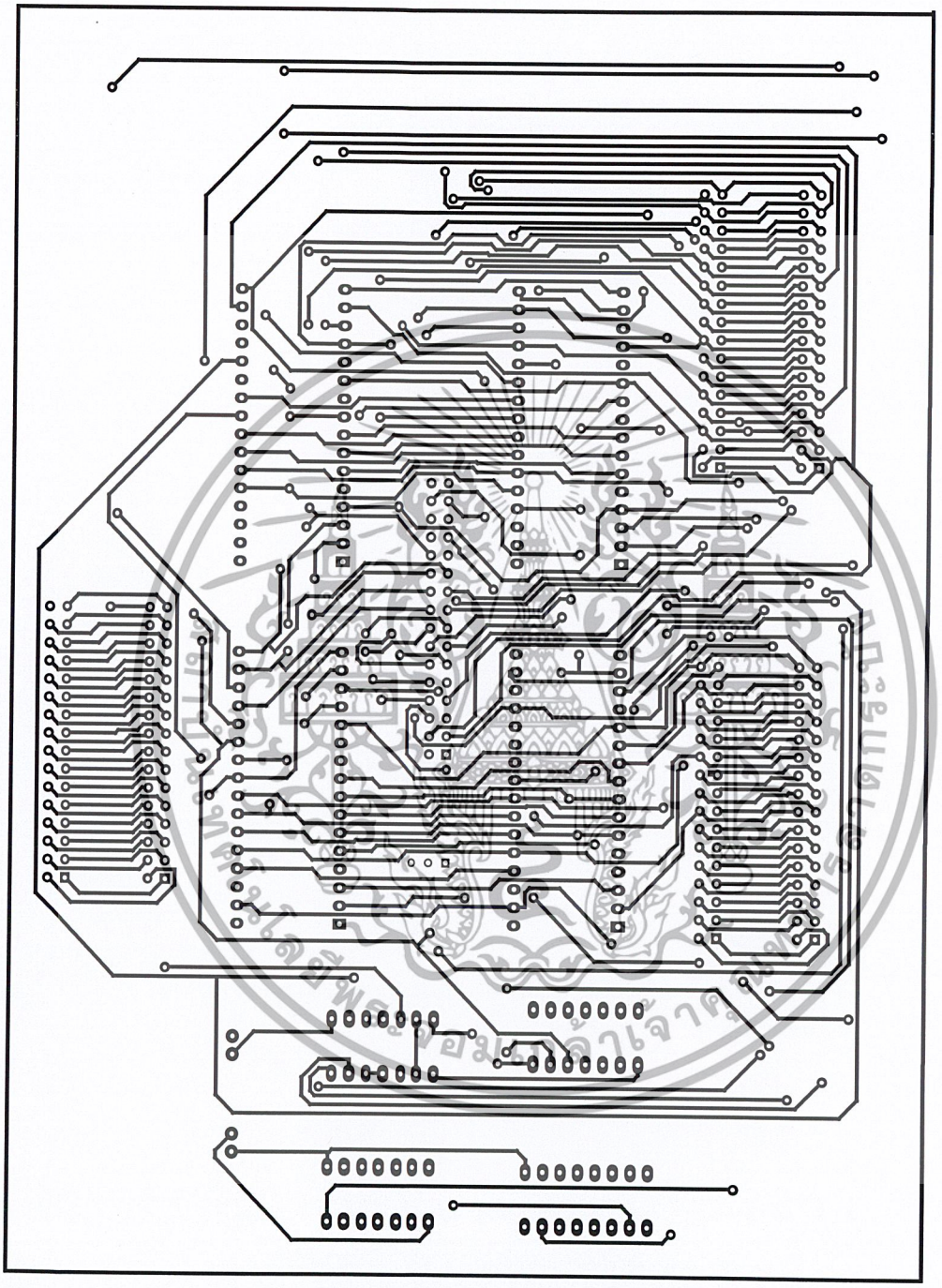
รูปที่ ข.1 วงจรเชื่อมต่อแรมภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



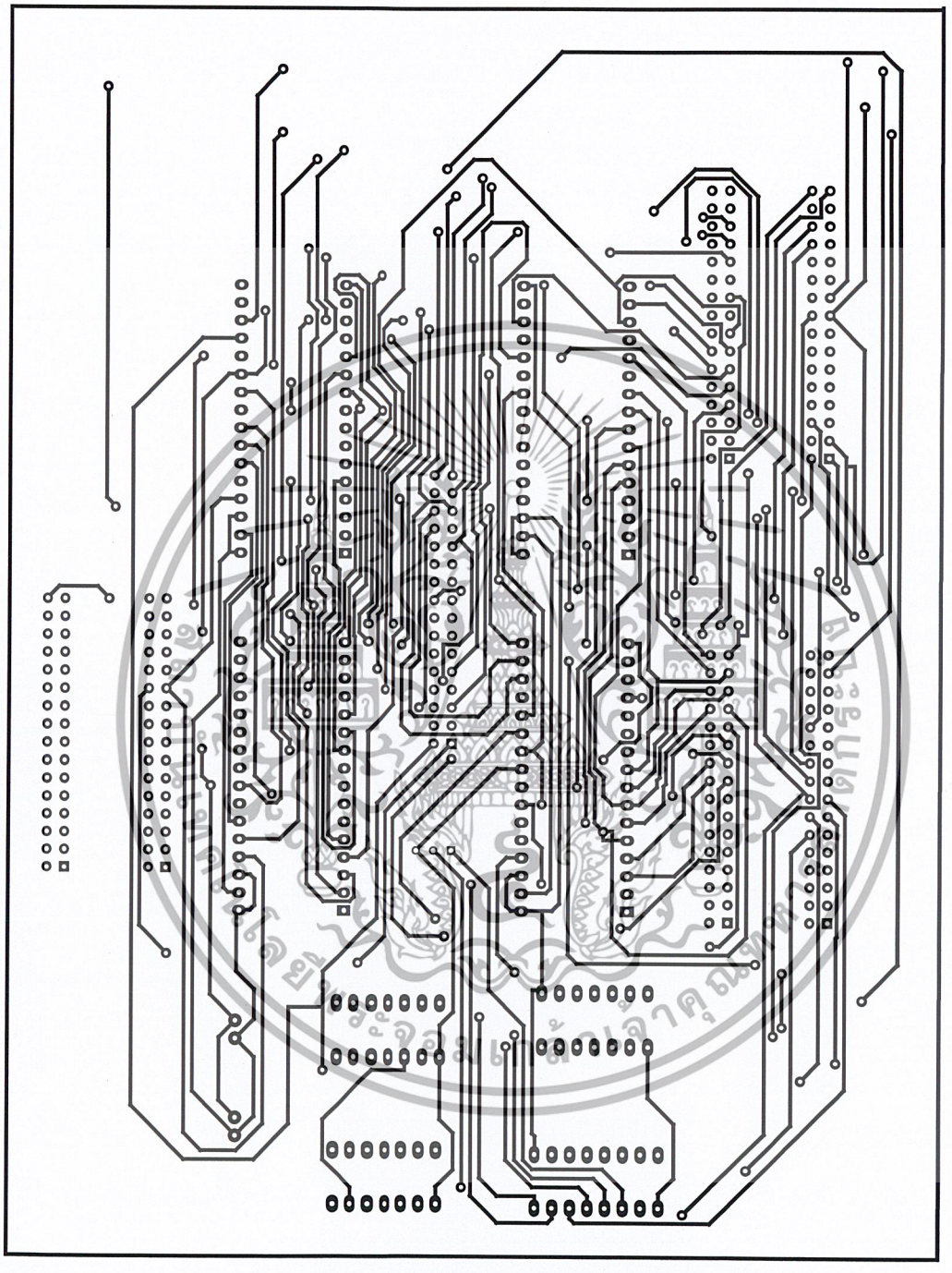
รูปที่ ข.2 วงจรการเชื่อมต่อ A/D และ D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



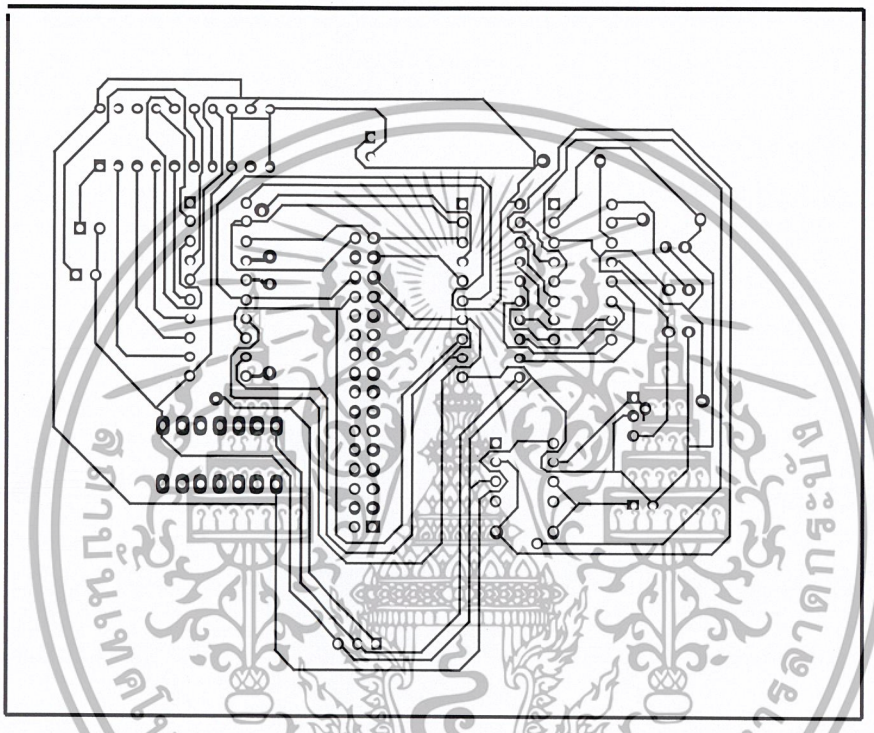
รูปที่ ข.3 แผ่นวงจรพิมพ์วงจรแรงเกาะผนอกด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แผ่นวงจรพิมพ์ของเครื่องมือวัดแรงดันภายนอก ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



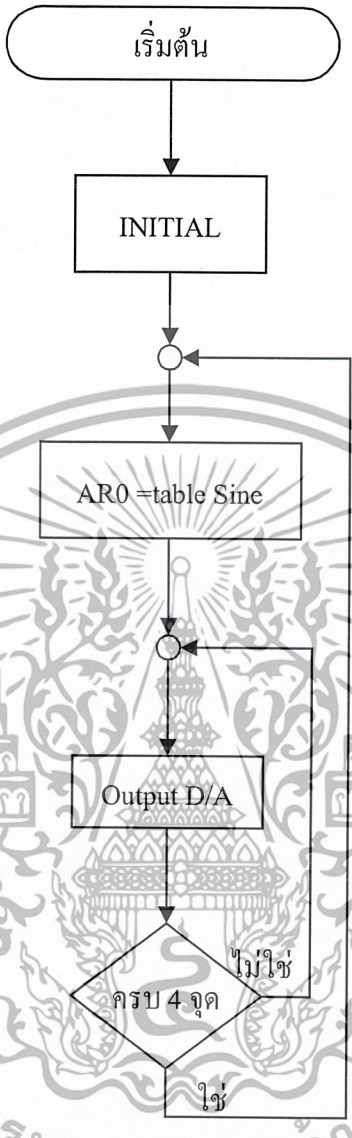
รูปที่ ข.5 แผ่นพิมพ์วงจร A/D และ D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



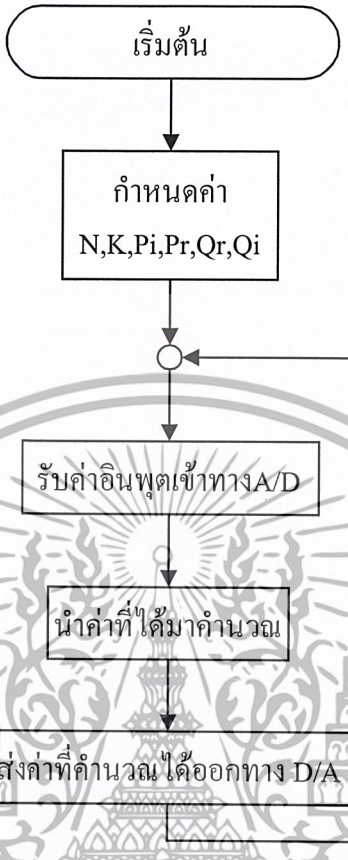
ภาคผนวก ค
ผังการทำงานและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



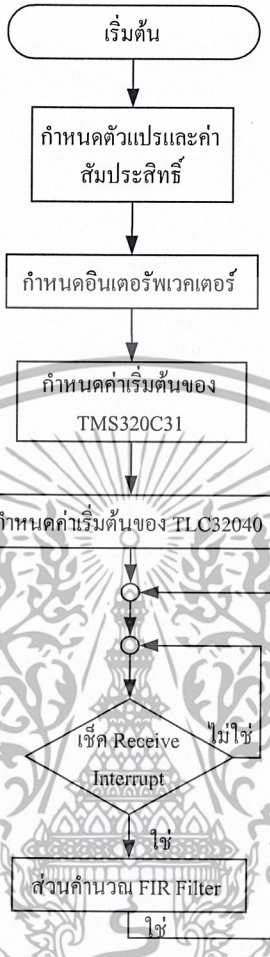
รูปที่ ค.1 ผังการทำงานของโปรแกรมกำเนิดคลื่นไซน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.2 ผังการทำงานของโปรแกรม Fast Fourier Transform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.3 ผังการทำงานของโปรแกรมฟิลเตอร์แบบ FIR

```
;SINE4P.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start   ".text",0x809900          ;starting address for text
        .start   ".data",0x809C00          ;start address for data
        .include "AICCOM31.ASM"    ;AIC communication routines
        .data                              ;data section
AICSEC  .word    162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word  SINE_VAL          ;address of sine values
        .brstart "SINE_BUFF",16    ;align sine table
SINE_VAL    .word   0,1000,0,-1000          ;sine values
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LENGTH      .set   4      ;length of circular buffer
            .entry BEGIN        ;start of code
            .text               ;text section
BEGIN   CALL   AICSET     ;initialize AIC
            LDI    LENGTH,BK  ;BK = size of buffer
            LDI    @SINE_ADDR,AR1    ;AR1 = addr of sine values
LOOP        LDI    *AR1++%,R7        ;R7 = table value
            CALL   AICIO_P           ;call AICIO for output
            BR     LOOP              ;loop back
            .end                     ;end
```

รูปที่ ค.4 โปรแกรม Sine4p.asm

```
;MATRIX.ASM - MATRIX/VECTOR MULTIPLICATIO
; (3x3)x(3x1) = (3x1)
        .start ".data",0x809C00          ;starting address for data
        .start ".text",0x809900          ;starting address for text
        .data                            ;data section
A       .float 1,2,3,4,5,6,7,8,9         ;values for matrix A
B       .float 1,2,3                     ;values for matrix B
A_ADDR  .word  A                         ;starting address of matrix A
B_ADDR  .word  B                         ;starting address of matrix B
OUT_ADDR word  $                         ;output (current) address
        .entry BEGIN                     ;start of code
        .text                            ;text section
BEGIN       LDP   A_ADDR                        ;init to data page 128
        LDI   @A_ADDR,AR0                ;AR0=starting address of A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LDI   @B_ADDR,AR1      ;AR1=starting address of B
      LDI   @OUT_ADDR,AR2   ;AR2= output address
      LDI   3,R4            ;R4 used as LOOPI counter
LOOPI LDF   0,R0            ;initialize R0=0
      LDI   2,AR4           ;AR4 used as LOOPJ counter
LOOPJ MPYF3 *AR0++,*AR1++,R1   ;R1=A[I,J]*B[J]
      ADDF3 R1,R0,R0        ;accumulate in R0
      DB    AR4,LOOPJ       ;decrement AR4.Branch until AR4<0
      FIX   R0,R2           ;convert R0 from float to integer
      STI   R2,*AR2++       ;store integer output in memory
      LDI   @B_ADDR,AR1     ;AR1=starting address of matrix B
      SUBI  1,R4            ;decrement R4
      BNZ   LOOPI           ;branch while R4 is not zero
      BR    $               ;branch to current addr (itself)
```

รูปที่ ค.5 โปรแกรม MATRIX.ASM

```
;SINE_RAM.ASM - GENERATES A SINE USING ONLY 4 POINTS
      .start  ".text",0x809900      ;starting address for text
      .start  ".data",0x100000  ;<NEW DATA ADDRESS>
      .include "AICCOM31.ASM"   ;AIC communication routines
      .data                     ;data section
AICSEC      .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word   SINE_VAL      ;address of sine values
      .brstart "SINE_BUFF",16   ;align sine table
SINE_VAL    .word   0,1000,0,-1000          ;sine values
LENGTH      .set    4                       ;length of circular buffer
      .entry  BEGIN             ;start of code
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .text                          ;text section
BEGIN   CALL    AICSET                 ;initialize AIC
LDI     LENGTH,BK       ;BK = size of buffer
LDI     @SINE_ADDR,AR1  ;AR1 = addr of sine values
LOOP    LDI     *AR1++%,R7  ;R7 = table value
CALL    AICIO_P         ;call AICIO for output
BR      LOOP            ;loop back
.end                    ;end
```

รูปที่ ค.6 โปรแกรม SINE_RAM.ASM

```
;*AICCOM31.ASM - AIC COMMUNICATION ROUTINES –
;POLLING OR INTERRUPT
        .data           ;assemble into data section
PBASE   .word   808000h     ;peripheral base address
SETSP   .word   0E970300h   ;serial port set-up data
ATABLE  .word   AICSEC      ;SP0 AIC init table address
        .text           ;assemble into text section
AICSET  PUSH    AR0         ;save AR0
        PUSH    AR1         ;save AR1
        PUSH    R0          ;save R0
        PUSH    R1          ;save R1
        LDI     @PBASE,AR0  ;AR0 -> 808000h
        LDI     1,R0        ;timer CLK=H1/2*(AIC master CLK)
        STI     R0,*+AR0(28h)  ;timer period reg(TCLK0=6.25 MHZ)
        LDI     03C1h,R0    ;init timer global register
        STI     R0,*+AR0(20h)  ;reset timer
        LDI     62h,IOF     ;AIC reset = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LDI    @ATABLE,AR1    ;AR1 -> AIC init data
RPTS   99             ;repeat next instr 100 times
NOP                   ;keep IOF low for a while
LDI    131h,R0        ;X & R port control register data
STI    R0,*+AR0(42h)  ;FSX/DX/CLKX=SP operational pins
STI    R0,*+AR0(43h)  ;FSR/DR/CLKR=SP operational pins
LDI    @SETSP,R0      ;RESET->SP:16 bits,ext clks,std mode
STI    R0,*+AR0(40h)  ;FSX=output&INT enable SP global reg
LDI    0,R0           ;R0 = 0
STI    R0,*+AR0(48h)  ;clear serial port XMIT register
OR     06h,IOF        ;bring AIC out of reset
LDI    03h,RC         ;RC=3 to transmit 4 values
RPTB   SECEND         ;repeat 4 data transmit of sec com
CALL   TWAIT          ;wait for data transmit
LDI    03h,R0         ;value for secondary XMIT request
STI    R0,*+AR0(48h)  ;secondary XMIT request to AIC
CALL   TWAIT          ;wait for data transmit
LDI    *AR1++(1),R0   ;AR1 -> next AIC init data
SECEND STI    R0,*+AR0(48h)  ;DTR = current AIC data
POP    R1             ;restore R1
POP    R0             ;restore R0
POP    AR1            ;restore AR1
POP    AR0            ;restore AR0
RETS                  ;return from subroutine
AICSET_I  ;-CONFIG FOR INTERRUPT
CALL   AICSET         ;call AICSET routine
LDI    0h,IF          ;clear IF register
OR     10h,IE         ;enable EXINT0 CPU interrupt
OR     2000h,ST       ;global interrupt enable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RETS            ;return from subroutine
    ;------TRANSMIT WAIT ROUTINE-----
TWAIT   PUSH   AR0      ;save AR0
        PUSH   R0       ;save R0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
TW1     LDI    *+AR0(40h),R0  ;R0=content of SP global control reg
        AND    02h,R0       ;see if transmit buffer is ready
        BZ     TW1          ;if not ready, try again
        POP    R0           ;restore R0
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine
    ;-------------------AIC TRANSFER ROUTINE-------------
AICIO_I  LDI   R7,R6        ;copy output to modify for AIC
        LSH    2,R6         ;two LSB must=0 for primary AIC comm
IO      PUSH   AR0          ;save AR0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
        STI    R6,*+AR0(48h)  ;DTR = next data for AIC D/A
        LDI    *+AR0(4Ch),R6  ;R6 = DRR data from AIC A/D
        LSH    16,R6        ;left shift for sign extension
        ASH    -18,R6       ;right shift keeping sign
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine
    ;--------------------AIC POLLING ROUTINE----------------
AICIO_P  CALL  TWAIT        ;wait for data to be transferred
        CALL   AICIO_I      ;call AIC transfer routine
        RETS                ;return from subroutine
SW_IO   PUSH   AR0          ;save AR0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
        LDI    R7,R6        ;copy output to modify for AIC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LSH    2,R6          ;prepare for secondary AIC com
        OR     03h,R6        ;set two LSB for secondary com
        CALL   TWAIT         ;wait for data to be transferred
        CALL   IO            ;call AIC transfer routine
        CALL   TWAIT         ;wait for data to be transferred
        STI    R1,*+AR0(48h) ;DTR = next data for AIC control
        POP    AR0           ;restore AR0
        RETS                 ;return from subroutine
;SUBROUTINES FOR PRIMARY OR AUXILIARY INPUT
IOPRI   PUSH   R1            ;save R1
        LDI    063h,R1       ;load secondary com data into R1
        CALL   SW_IO         ;call IO routine to switch inputs
        POP    R1            ;restore R1
        RETS                 ;return from subroutine
IOAUX   PUSH   R1            ;save R1
        LDI    073h,R1       ;load secondary com data into R1
        CALL   SW_IO         ;call IO routine to switch inputs
        POP    R1            ;restore R1
        RETS                 ;return from subroutine
```

รูปที่ ค.7 โปรแกรม AICCOM31.ASM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*FFT.C - REAL-VALUED FFT WITH 128 POINTS. CALLS FFT_RL.ASM/

#include "math.h"                    /*standard library func  */

#include "aiccomc.c"                 /*AIC comm routines      */

#define N 128                        /*size of FFT            */

#define M 7                          /*number of stages       */

volatile int index = 0;              /*input_output index     */

float *IO_buffer, *data, *temp;      /*--> array buffers      */

int AICSEC[4] = {0x162C,0x1,0x3872,0x67};  /*AIC config data  */

extern void fft_rl(int, int, float *);     /*fft function protype  */

void c_int05()                       /*interrupt handler func */

{

 PBASE[0x48] = ((int)(IO_buffer[index])) << 2;      /*output data*/

IO_buffer[index] = (float)(PBASE[0x4C] << 16 >> 18);  /*input data */

 if (++index >= N) index = 0;        /*increment index, reset = N*/

}

main()

{

 int loop;                   /* declare variable       */

 float real, img;            /* declare variables      */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AICSET_I();                          /*config AIC for interrupt  */

IO_buffer = (float *) calloc(N, sizeof(float)); /*input_out buffer  */

data = (float *) calloc(N, sizeof(float));     /* fft data buffer       */

while (1)                            ;* create endless loop      */

{

fft_rl(N, M, (float *)data);         /*call FFT function         */

 data[0] = sqrt(data[0]*data[0])/N;          /*magnitude of X(0) */

 for (loop = 1; loop < N/2; loop++)    /*calculate X(1)..X(N/2-1)  */

{

real = data[loop];                           /*real part  */

 img = data[N-loop];                         /*imaginary part   */

 data[loop] = sqrt(real*real+img*img)/N;   /*find magnitude   */

}

 data[N/2] = sqrt(data[N/2]*data[N/2])/N;  /*magnitude of X(N/2)    */

 for (loop = N/2+1; loop < N; loop++)      /*X(N/2+1).. X(N-1)*/

  data[loop] = data[N-loop];                          /*use symetry  */

 while (index);                  /      *wait till IO_buffer empty */

emp = data;                          *temp => data buffer       */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data = IO_buffer;                /*IO_buffer-->data buffer   */

IO_buffer = temp;                /*data buffer->new IO_buffer*/

IO_buffer[0] = -2048;            /*sync pulse,negative spike */

}

}
```

รูปที่ ค.8 โปรแกรม Fast Fourier Transform

```
;LP11SIM.ASM - FIR LOWPASS FILTER WITH 11 COEFF FOR SIMULATION
        .start  ".text",0x809900   ;        where text begins
        .start  ".data",0x809C00 ;where data begins
        .data                              ;data section
IN_ADDR     .word   INB                    ;starting address for input
OUT_ADDR    .word   OUTB                   ;starting address for output
XB_ADDR     word    XN+LENGTH-1    ;bottom address of circ buffer
HN_ADDR     .word   COEFF          ;starting addr of coefficients
COEFF       .float  0          ;H10
        .float  0.0468, 0.1009, 0.1514, 0.1872, 0.2
        .float  0.1872, 0.1514, 0.1009, 0.0468
H0      .float  0                          ;H0
LENGTH  .set    H0-COEFF+1     ;# of coefficients
INB     .float  10000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
OUTB    .float  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
        .brstart "XN_BUFF",16          ;align samples buffer
XN      .sect   "XN_BUFF"              ;section for input samples
        .loop   LENGTH                 ;buffer size for samples
        .float  0.0                    ;initialize samples to zero
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .endloop                        ;end of loop
        .entry  BEGIN                   ;start of code
        .text
BEGIN       LDP     @0x809800   ;init to data page 128
        LDI     LENGTH,BK               ;size of circular buffer
        LDI     @XB_ADDR,AR1            ;last sample address ->AR1
        LDI     @IN_ADDR,AR2            ;input address      -->AR2
LDI     @OUT_ADDR,AR3       ;output address     -->AR3
FILT    LDI     LENGTH-1,AR4            ;length in AR4
LOOP    LDF     *AR2++,R3               ;input new sample
STF     R3,*AR1++%                      ;store newest sample
        LDI     @HN_ADDR,AR0            ;AR0 points to H(N-1)
        LDF     0,R0                    ;init R0
        LDF     0,R2                    ;init R2
        RPTS    LENGTH-1                ;repeat LENGTH-1 times
        MPYF3   *AR0++,*AR1++%,R0 ;R0 = HN*XN
        ADDF3   R0,R2,R2                ;accumulation in R2
        DBNZD   AR4,LOOP                ;delayed branch until AR4<0
        ADDF    R0,R2                   ;last mult result accumulated
        FIX     R2,R7                   ;convert float R2 to integer R7
        STI     R7,*AR3++               ;store into output buffer
WAIT    BR      WAIT                    ;wait
        .end                    ;end
```

รูปที่ ค.9 โปรแกรม LP11SIM.ASM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ง
รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.วงจรเชื่อมต่อแรมภายนอก

1) IC เบอร์ K6T1008E	1 ตัว
2) IC เบอร์ 7404	1 ตัว
3) IC เบอร์ 74139	1 ตัว
4) IC เบอร์ 7408	1 ตัว
5) IC เบอร์ 7432	1 ตัว
6) คอนเนคเตอร์ 32 ขา	7 ตัว
7) คอนเนคเตอร์ 2 ขา	2 ตัว

ตารางที่ ง.1 รายการอุปกรณ์ที่ใช้ในวงจรการเชื่อมต่อแรมภายนอก

2. วงจรการเชื่อมต่อ A/D และ D/A

1) ตัวต้านทาน 1.8 กิโลโอห์ม	3 ตัว
2) คาปาซิสเตอร์ 0.1 ไมโครฟารัด	2 ตัว
3) คาปาซิสเตอร์ 0.01 ไมโครฟารัด	1 ตัว
4) IC เบอร์ 7404	1 ตัว
5) IC เบอร์ 74573	1 ตัว
6) IC เบอร์ ADC0820	1 ตัว
7) IC เบอร์ DAC0800	1 ตัว
8) IC เบอร์ 741	1 ตัว

ตารางที่ ง.2 รายการอุปกรณ์ที่ใช้ในวงจรการเชื่อม A/D และ D/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1
การใช้งานบอร์ด DSK TMS320C31 เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

1) เพื่อให้นักศึกษาสามารถใช้งานบอร์ด DSK TMS320C31 เบื้องต้นได้อย่างถูกต้อง

ทฤษฎี

แผงวงจร DSK TMS320C31 สามารถใช้ภาษาในการเขียนอยู่ถึง 2 ภาษา จึงต้องมีการคอมไพล์ให้เป็นภาษาเครื่องเพื่อให้บอร์ด DSK ทำงานตามโปรแกรมที่ป้อนให้กับบอร์ด DSK TMS320C31 การเขียนโปรแกรมจะเขียนได้จากคอมพิวเตอร์ โดยใช้หน้าต่างของ DOS จึงทำให้มีคำสั่งที่ยุ่งยากพอสมควร แต่สามารถเข้าใจง่าย

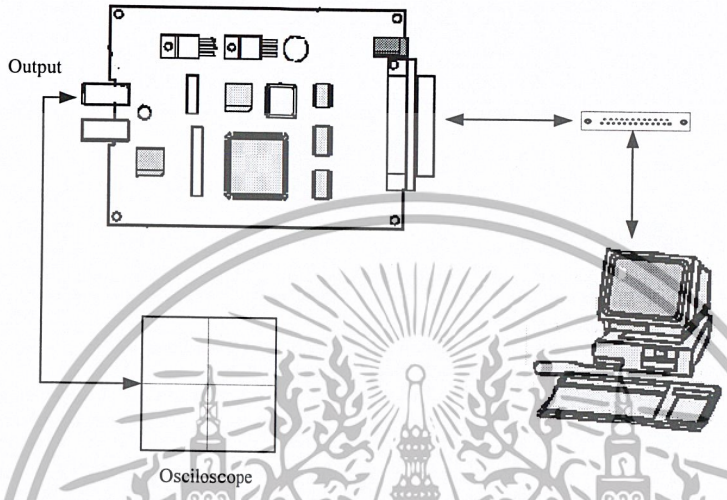
อุปกรณ์

1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 เครื่อง
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง
4) แผ่นดิสก์โปรแกรมDSK	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ จ.1



รูปที่ จ.1 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) เข้าไปที่ DSKTOOL ใช้คำสั่งDSK 3a ตามด้วยชื่อไฟล์นามสกุล .asmในใบงานใช้ชื่อไฟล์ชื่อ sine4p.asm (DSK3aเป็นการแปลงไฟล์ DSK เพื่อใช้โหลดลงDSK TMS320C31)

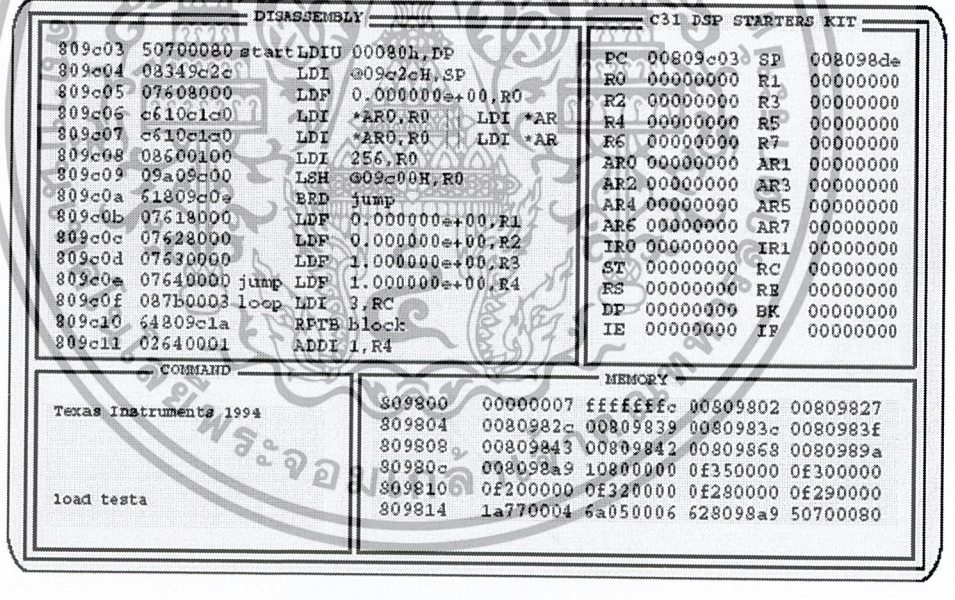
```
;SINE4P.ASM - GENERATES A SINE USING ONLY 4 POINTS
                .start  ".text",0x809900          ;starting address for text
                .start  ".data",0x809C00          ;start address for data
                .include "AICCOM31.ASM"   ;AIC communication routines
                .data                             ;data section
AICSEC          .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word   SINE_VAL          ;address of sine values
                .brstart "SINE_BUFF",16   ;align sine table
SINE_VAL        .word   0,1000,0,-1000            ;sine values
LENGTH          .set    4       ;length of circular buffer
                .entry  BEGIN             ;start of code
                .text                     ;text section
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN    CALL   AICSET      ;initialize AIC
         LDI    LENGTH,BK   ;BK = size of buffer
         LDI    @SINE_ADDR,AR1    ;AR1 = addr of sine values
LOOP     LDI    *AR1++%,R7        ;R7 = table value
         CALL   AICIO_P           ;call AICIO for output
         BR     LOOP              ;loop back
         .end          ;end
```

รูปที่ จ.2 โปรแกรม Sine4p.asm

3) ใส่คำสั่ง DSK 3d เพื่อเข้าสู่หน้าต่าง debager ดังรูปที่ จ.2



รูปที่ จ.3 หน้าต่าง Debuger

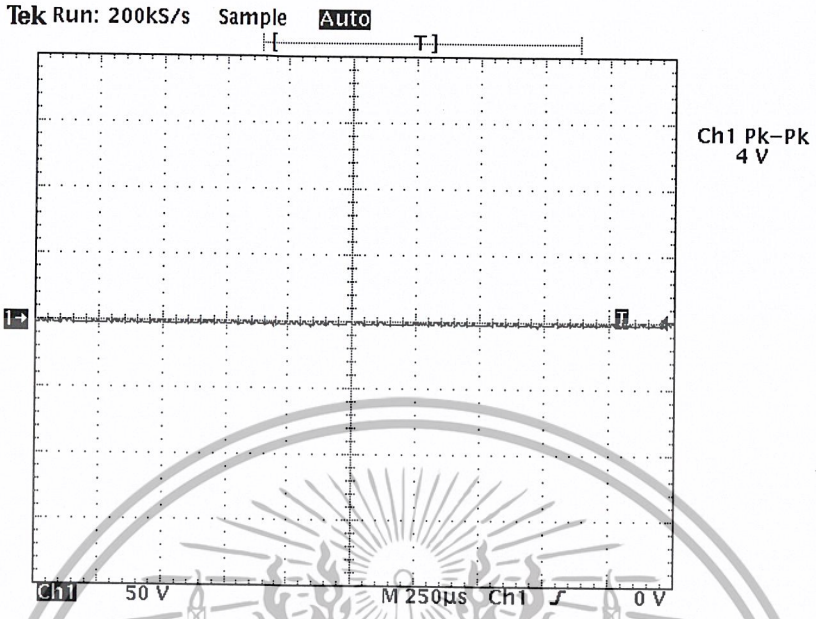
4) ทำการ reset บอร์ด DSK TMS320C31 โดยใช้คำสั่ง resetในหน้าต่างdebuger

5) ใช้คำสั่งLoad ชื่อ .dsk (จากข้อ2) ดังนี้ Load sine4p.dsk

6) ใส่คำสั่ง RUN เพื่อรันโปรแกรม หรือกด F5

7) วาดรูปผลการทดลองจากสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Volt/Div=_____ Time/Div=_____ f=_____ Hz

สรุปผลการทดลอง

.....

.....

คำถาม

ถ้าต้องการโหลดไฟล์ต้องทำอย่างไรจงอธิบาย ?

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2
การใช้ชุดคำสั่งแบบง่าย

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถอธิบายการใช้ชุดคำสั่งแบบง่ายได้
2. เพื่อให้นักศึกษาสามารถใช้โปรแกรมการคูณแมททริกได้ถูกต้อง

ทฤษฎี

ในการศึกษาชุดคำสั่งนั้น ต้องเริ่มต้นจากโปรแกรมง่ายๆ เพื่อที่จะได้เข้าใจหลักการของชุดคำสั่งได้อย่างรวดเร็ว ในใบงานนี้จึงขอยกตัวอย่างการใช้โปรแกรมการคูณแมททริก ดังนี้

โดยการใช้โปรแกรม การคูณแมทริกขนาด(3x3)*(3x1)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} x \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} (1x1) + (2x2) + (3x3) \\ (4x1) + (5x2) + (6x3) \\ (7x1) + (8x2) + (9x3) \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 32 \\ 50 \end{bmatrix}$$ Ans

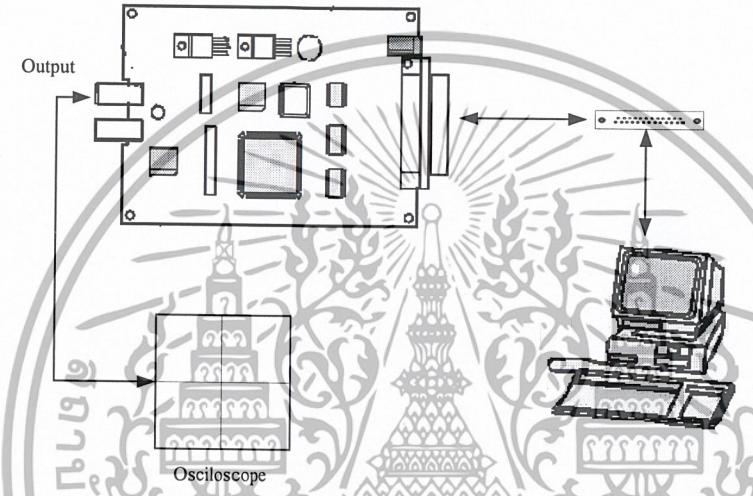
อุปกรณ์

1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) เครื่องคอมพิวเตอร์	1 เครื่อง
3) แผ่นดิสก์โปรแกรม DSK	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูป



รูปที่ จ.4 การต่อชุดทดลองกับอุปกรณ์

2) ป้อนโปรแกรมดังนี้

```
;MATRIX.ASM - MATRIX/VECTOR MULTIPLICATIO
; (3x3)x(3x1) = (3x1)
        .start ".data",0x809C00          ;starting address for data
        .start ".text",0x809900          ;starting address for text
        .data                            ;data section
A               .float 1,2,3,4,5,6,7,8,9 ;values for matrix A
B               .float 1,2,3             ;values for matrix B
A_ADDR          .word  A                 ;starting address of matrix A
B_ADDR          .word  B                 ;starting address of matrix B
OUT_ADDR        word $                   ;output (current) address
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .entry BEGIN                    ;start of code
        .text                           ;text section
BEGIN       LDP   A_ADDR                    ;init to data page 128
        LDI   @A_ADDR,AR0           ;AR0=starting address of A
        LDI   @B_ADDR,AR1       ;AR1=starting address of B
        LDI   @OUT_ADDR,AR2     ;AR2= output address
        LDI   3,R4                  ;R4 used as LOOPI counter
LOOPI       LDF   0,R0                  ;initialize R0=0
        LDI   2,AR4                 ;AR4 used as LOOPJ counter
LOOPJ       MPYF3  *AR0++,*AR1++,R1     ;R1=A[I,J]*B[J]
        ADDF3  R1,R0,R0         ;accumulate in R0
        DB    AR4,LOOPJ         ;decrement AR4.Branch until AR4<0
        FIX   R0,R2             ;convert R0 from float to integer
        STI   R2,*AR2++             ;store integer output in memory
        LDI   @B_ADDR,AR1       ;AR1=starting address of matrix B
        SUBI  1,R4                  ;decrement R4
        BNZ   LOOPI             ;branch while R4 is not zero
        BR    $                 ;branch to current addr (itself)
```

รูปที่ จ.5 โปรแกรม MATRIX.ASM

3) ทำการคอมไพล์ โปรแกรมตามใบงานที่ 1

4) ทำการโหลดโปรแกรมในหน้าต่าง Debugger โปรแกรมจะเริ่มที่ตำแหน่ง 809900H และข้อมูลจะเริ่มที่ตำแหน่ง 809C00H ในบรรทัดที่ 5 และ 6จะอยู่ที่ตำแหน่งเริ่มต้นของตำแหน่งข้อมูล ดังนั้น คำตอบจะเริ่มที่ตำแหน่ง 809c0eH เราต้องการดูค่าที่ตำแหน่ง 809c0eH โดยการใช้คำสั่ง memd 0x809c0e (เป็นคำสั่งที่ใช้ดูเลขจำนวนเต็ม)

5) กด F8 เพื่อทำการรันทีละสเต็ป ทำการกดไปเรื่อยๆ สังเกตการเปลี่ยนแปลงค่าที่ตำแหน่ง 809c0eH

6) ทำการบันทึกค่า ที่อยู่ในตำแหน่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

809c0eH = ...

809c0fH = ...

809c10H = ...

สรุปผลการทดลอง

.....

.....

.....

คำถามท้ายการทดลอง

ถ้าต้องการเปลี่ยนสมการเป็น $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \times \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$ จะต้องเพิ่มโปรแกรมส่วนใด ?

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
การทดสอบการเชื่อมต่อหน่วยความจำภายนอก

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถบอกหลักการเชื่อมต่อหน่วยความจำภายนอกได้ถูกต้อง

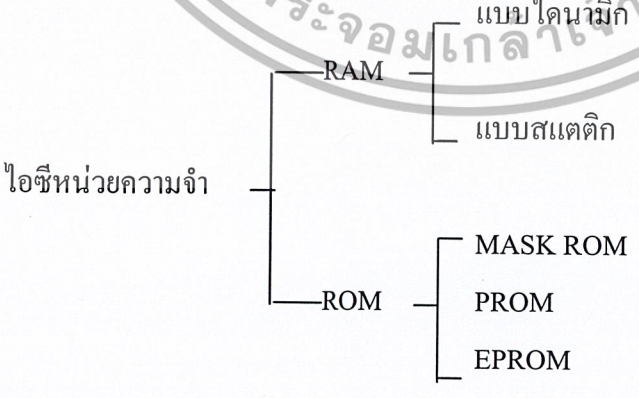
2.เพื่อให้นักศึกษาสามารถอธิบายหลักการการติดต่อกับหน่วยความจำภายนอกได้อย่างถูกต้อง

ทฤษฎี

หน่วยความจำในระยะแรกจะเป็นแหวนแม่เหล็ก (core memory) เป็นกระบอกแม่เหล็ก (magnetic drum) ซึ่งเป็นอุปกรณ์แม่เหล็กไฟฟ้าและเชิงกลไฟฟ้า ทำให้การติดต่อกับ CPU ทำได้ยาก วงจรซับซ้อนมาก การทำหน่วยความจำเป็นไอซีได้จึงเป็นเรื่องสำคัญทำให้วงจรในคอมพิวเตอร์ง่ายลงมาก

เมื่อเทคโนโลยีในการทำ MOS ไอซีเจริญขึ้น ก็สามารถทำไอซีที่มีหน่วยความจำที่มีความจุสูงขึ้น สามารถใช้งานในทางปฏิบัติได้ดีขึ้น จึงทำให้ไมโครคอมพิวเตอร์มีขนาดเล็กลงและสามารถใช้งานได้ดีเยี่ยม

ถ้าแบ่งชนิดของไอซีหน่วยไอซีหน่วยความจำออกตามคุณสมบัติและการใช้งานจะสามารถแบ่งออกได้ดังนี้



รูปที่ จ.6 แสดงการแบ่งประเภทของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

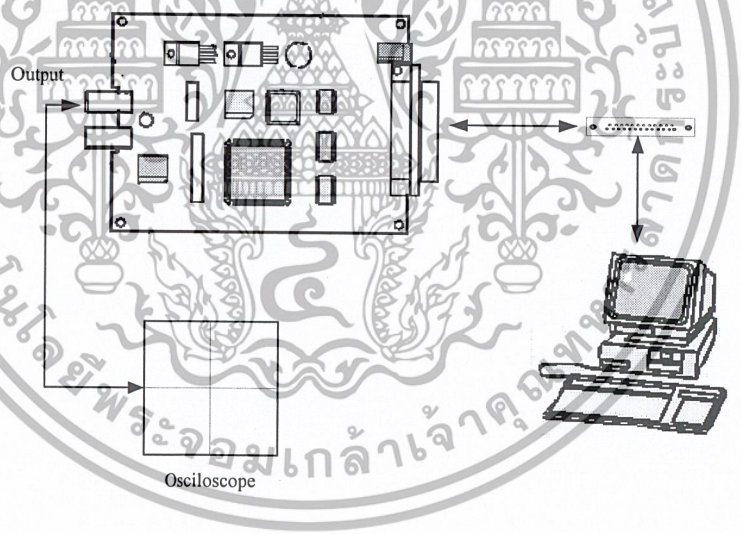
การต่อหน่วยความจำภายนอกต้องมีการต่อวงจรถอดรหัส โดยใช้ขาแอดเดรสมาถอดรหัส เพื่อจะไปเป็นตัวเลือกตัวไอซี ในการบันทึกข้อมูล

อุปกรณ์

1) ชุดทดลอง การประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง
4) แผ่นดิสก์โปรแกรมDSK	1 แผ่น

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ จ.6



รูปที่ จ.7 การต่อชุดทดลองกับอุปกรณ์

2) การทดสอบหน่วยความจำภายนอก โดยการกดคำสั่งdsk3d <Enter>

3) พิมพ์คำสั่ง mem 0x100000 เนื่อจากหน่วยความจำภายนอกต่ออยู่ที่ตำแหน่งแอดเดรส 100000H

4) ที่ตำแหน่ง 100000H จะแสดงข้อมูลขณะนั้นอยู่สังเกตค่า บันทึกผลการทดลอง

100000H _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) กดALT+m หรือ กดF1 เพื่อดูคำอธิบายจากนั้นทำการป้อนค่า โดยใช้คำสั่ง mm ใน Debuger เช่น mm 0x100000 จะเป็นการใส่ค่าแอดเดรสที่ 100000H

6) ใส่ค่าดังต่อไปนี้ ตำแหน่ง 100000H ใส่ FFFF, ตำแหน่งที่ 100002H ใส่ FC ,ตำแหน่งที่ 100005H ใส่ 9999, ตำแหน่งที่ 100009.ใส่ 77C9 บันทึกผลการทดลอง

100000H _____

100002H _____

100005H _____

100009H _____

7) โหลดโปรแกรม SIN_RAM.DSK สังเกตว่าจะเป็นตำแหน่งของหน่วยความจำภายนอก คือ ตำแหน่ง 100000H

```
;SINE_RAM.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start  ".text",0x809900          ;starting address for text
        .start  ".data",0x100000  ;<NEW DATA ADDRESS>
        .include "AICCOM31.ASM"    ;AIC communication routines
        .data                              ;data section
AICSEC  .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word  SINE_VAL          ;address of sine values
        .brstart "SINE_BUFF",16    ;align sine table
SINE_VAL .word   0,1000,0,-1000            ;sine values
LENGTH  .set    4                  ;length of circular buffer
        .entry  BEGIN              ;start of code
        .text                              ;text section
BEGIN   CALL    AICSET                     ;initialize AIC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LDI     LENGTH,BK        ;BK = size of buffer
LDI     @SINE_ADDR,AR1   ;AR1 = addr of sine values
LOOP            LDI     *AR1++%,R7    ;R7 = table value
CALL    AICIO_P          ;call AICIO for output
BR      LOOP             ;loop back
.end                     ;end
```

รูปที่ จ.8 โปรแกรม SINE_RAM.ASM

8) บันทึกผลการทดลอง



Volt/Div _____ Time/Div_____ f = _____Hz

สรุปผลการทดลอง

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

ถ้าหน่วยความจำภายนอกอยู่ที่ตำแหน่ง 700000H จะติดต่อกับหน่วยความจำได้อย่างไร จงอธิบาย ?



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
การสร้างรูปคลื่นไซน์(Sine Wave)

จุดประสงค์เชิงพฤติกรรม

1.เพื่อให้นักศึกษาสามารถเขียนโปรแกรมบน DSK TMS320C31 ได้

2.เพื่อให้นักศึกษาสามารถบอกวิธีการสร้างรูปคลื่นไซน์จาก DSKTMS320C31ได้

ทฤษฎี

Sine Generator

การใช้งาน DSK TMS320C31 ในการผลิตสัญญาณไซน์เวฟ (Sine Wave)มีลักษณะดังนี้

คลื่นรูปไซน์ (Sine Wave) ที่สร้างโดย DSK TMS320C31 นั้น ในการกำหนดความถี่จะอาศัยหลักการสุ่มตัวอย่างความถี่ (Sampling Frequency) และการกำหนดจุด (N) โดยใช้สมการที่ (1)

$$F = \frac{Fs}{N}$$

F = ความถี่เอาต์พุต

Fs = ความถี่สุ่ม (Sampling Frequency)

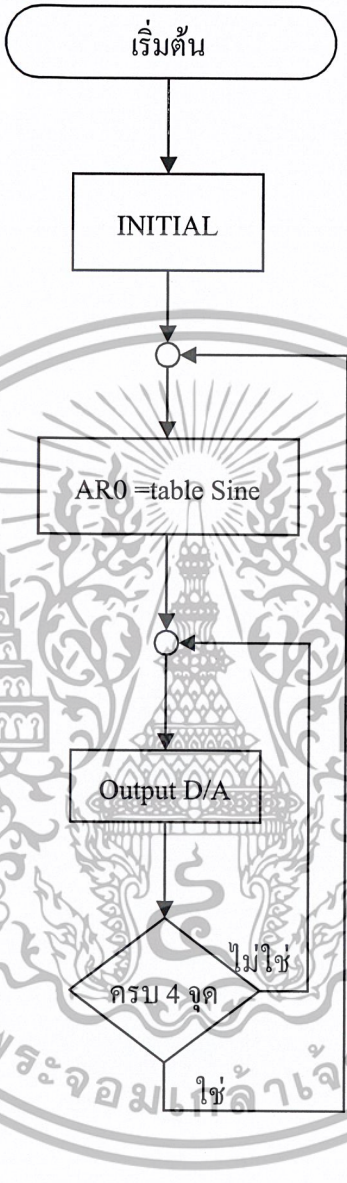
N = จำนวนจุดในการสุ่มสัญญาณ 1 คาบเวลา

จากหลักการของการมองค่าจากตาราง ซึ่งจำเป็นจะต้องคำนวณหาค่าไซน์ (Sine) ของแต่ละจุดหลังจากนั้นทำการสเกลค่าด้วยการคูณ 1,000 และนำค่าที่สเกลแล้วไปเก็บไว้ในหน่วยความจำ จากนั้นทำการเขียนโปรแกรมอ่านค่าจากหน่วยความจำ ส่งออกไปยัง D/A ของบอร์ด DSK TMS32CC31

ตัวอย่าเช่น ถ้าความถี่สุ่ม (Fs) มีค่า = 8k Hz จำนวนจุดในการสุ่มค่าสัญญาณ 1 คาบเวลา = 4 จุด ฉะนั้นความถี่เอาต์พุต = $\frac{8\,kHz}{4} = 2\,kHz$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังการทำงาน



รูปที่ จ.9 ผังการทำงานของโปรแกรมกำเนิดรูปคลื่นไซน์

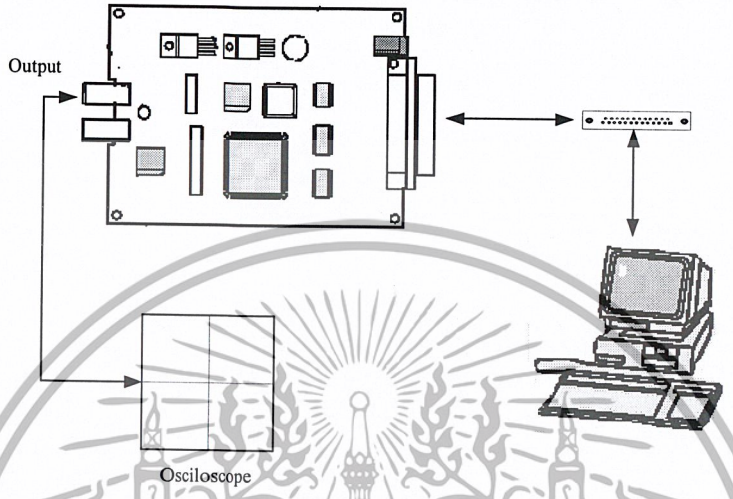
อุปกรณ์

1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง
4) แผ่นดิสก์โปรแกรม DSKTOOL	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ จ.10



รูปที่ จ.10 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อน โปรแกรมดังนี้

```
;*AICCOM31.ASM - AIC COMMUNICATION ROUTINES –
;POLLING OR INTERRUPT
        .data           ;assemble into data section
PBASE   .word   808000h     ;peripheral base address
SETSP   .word   0E970300h   ;serial port set-up data
ATABLE  .word   AICSEC      ;SP0 AIC init table address
        .text           ;assemble into text section
AICSET  PUSH    AR0         ;save AR0
        PUSH    AR1         ;save AR1
        PUSH    R0          ;save R0
        PUSH    R1          ;save R1
        LDI     @PBASE,AR0  ;AR0 -> 808000h
        LDI     1,R0        ;timer CLK=H1/2*(AIC master CLK)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        STI    R0,*+AR0(28h)  ;timer period reg(TCLK0=6.25 MHZ)
        LDI    03C1h,R0       ;init timer global register
        STI    R0,*+AR0(20h)  ;reset timer
        LDI    62h,IOF        ;AIC reset = 0
        LDI    @ATABLE,AR1    ;AR1 -> AIC init data
        RPTS   99             ;repeat next instr 100 times
        NOP                   ;keep IOF low for a while
        LDI    131h,R0        ;X & R port control register data
        STI    R0,*+AR0(42h)  ;FSX/DX/CLKX=SP operational pins
        STI    R0,*+AR0(43h)  ;FSR/DR/CLKR=SP operational pins
        LDI    @SETSP,R0      ;RESET->SP:16 bits,ext clks,std mode
        STI    R0,*+AR0(40h)  ;FSX=output&INT enable SP global reg
        LDI    0,R0           ;R0 = 0
        STI    R0,*+AR0(48h)  ;clear serial port XMIT register
        OR     06h,IOF        ;bring AIC out of reset
        LDI    03h,RC         ;RC=3 to transmit 4 values
        RPTB   SECEND         ;repeat 4 data transmit of sec com
        CALL   TWAIT          ;wait for data transmit
        LDI    03h,R0         ;value for secondary XMIT request
        STI    R0,*+AR0(48h)  ;secondary XMIT request to AIC
        CALL   TWAIT          ;wait for data transmit
        LDI    *AR1++(1),R0   ;AR1 -> next AIC init data
SECEND  STI    R0,*+AR0(48h)  ;DTR = current AIC data
        POP    R1             ;restore R1
        POP    R0             ;restore R0
        POP    AR1            ;restore AR1
        POP    AR0            ;restore AR0
        RETS                  ;return from subroutine
        AICSET_I  ;-CONFIG FOR INTERRUPT
        CALL   AICSET         ;call AICSET routine
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LDI    0h,IF        ;clear IF register
        OR     10h,IE       ;enable EXINT0 CPU interrupt
        OR     2000h,ST     ;global interrupt enable
        RETS                ;return from subroutine
;------TRANSMIT WAIT ROUTINE-----
TWAIT   PUSH   AR0          ;save AR0
        PUSH   R0           ;save R0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
TW1     LDI    *+AR0(40h),R0 ;R0=content of SP global control reg
        AND    02h,R0       ;see if transmit buffer is ready
        BZ     TW1          ;if not ready, try again
        POP    R0           ;restore R0
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine
;-----------------AIC TRANSFER ROUTINE-----------
AICIO_I LDI    R7,R6        ;copy output to modify for AIC
        LSH    2,R6         ;two LSB must=0 for primary AIC comm
IO      PUSH   AR0          ;save AR0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
        STI    R6,*+AR0(48h) ;DTR = next data for AIC D/A
        LDI    *+AR0(4Ch),R6 ;R6 = DRR data from AIC A/D
        LSH    16,R6        ;left shift for sign extension
        ASH    -18,R6       ;right shift keeping sign
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine

;--------------------AIC POLLING ROUTINE----------------
AICIO_P CALL   TWAIT        ;wait for data to be transferred
        CALL   AICIO_I      ;call AIC transfer routine
        RETS                ;return from subroutine
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW_IO    PUSH   AR0          ;save AR0
LDI   @PBASE,AR0    ;AR0 -> 0808000h
LDI   R7,R6         ;copy output to modify for AIC
LSH   2,R6          ;prepare for secondary AIC com
OR    03h,R6        ;set two LSB for secondary com
CALL  TWAIT         ;wait for data to be transferred
CALL  IO            ;call AIC transfer routine
CALL  TWAIT         ;wait for data to be transferred
STI   R1,*+AR0(48h) ;DTR = next data for AIC control
POP   AR0           ;restore AR0
RETS                ;return from subroutine
;SUBROUTINES FOR PRIMARY OR AUXILIARY INPUT
IOPRI    PUSH   R1          ;save R1
LDI   063h,R1       ;load secondary com data into R1
CALL  SW_IO         ;call IO routine to switch inputs
POP   R1            ;restore R1
RETS                ;return from subroutine
IOAUX    PUSH   R1          ;save R1
LDI   073h,R1       ;load secondary com data into R1
CALL  SW_IO         ;call IO routine to switch inputs
POP   R1            ;restore R1
RETS                ;return from subroutine
```

รูปที่ จ.11 โปรแกรม AICCOM31.ASM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ทำการบันทึกเป็นไฟล์ชื่อ AICCOM31.ASM หลังจากนั้นทำการเขียนโปรแกรมต่อไปนี้

```
;SINE4P.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start  ".text",0x809900
        .start  ".data",0x809C00
        .include "AICCOM31.ASM"     ;AIC communication routines
        .data                       ;data section
AICSEC      .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR   .word   SINE_VAL        ;address of sine values
        .brstart "SINE_BUFF",16     ;align sine table
SINE_VAL    .word   0,1000,0,-1000  ;sine values
LENGTH      .set    4               ;length of circular buffer
        .entry  BEGIN               ;start of code
        .text                       ;text section
BEGIN       CALL    AICSET          ;initialize AIC
        LDI     LENGTH,BK           ;BK = size of buffer
        LDI     @SINE_ADDR,AR1      ;AR1 = addr of sine values
LOOP        LDI     *AR1++%,R7      ;R7 = table value
        CALL    AICIO_P             ;call AICIO for output
        BR      LOOP                ;loop back
        .end                        ;end
```

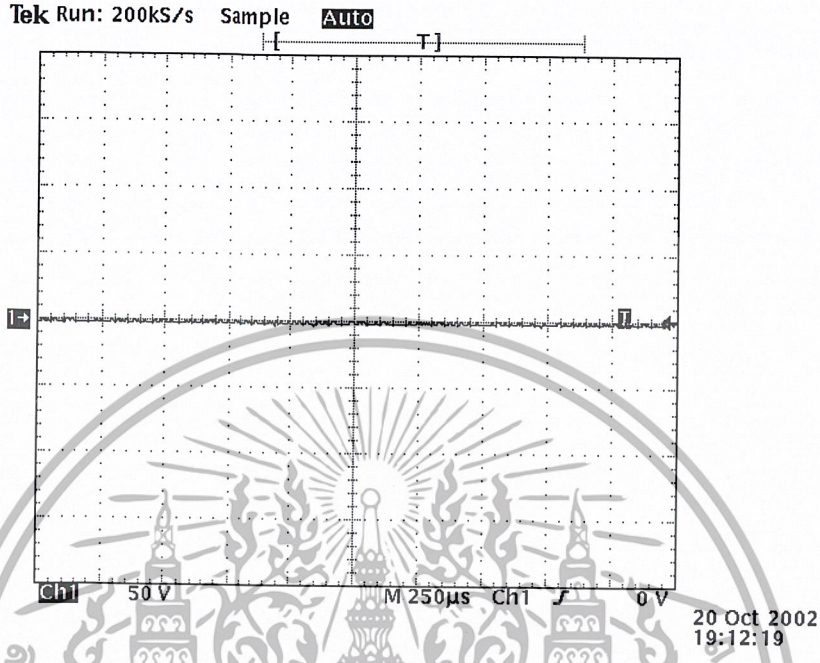
รูปที่ จ.12 โปรแกรม SINE4P.ASM

ทำการบันทึกเป็น ชื่อ SINE4P.ASM

4) คอมไพล์และรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) วาดรูปสัญญาณ



Volt/Div=_____ Time/Div=_____ f=_____Hz

6) ทดลองเปลี่ยนค่า ในบรรทัดที่ 9 ของโปรแกรม SINE4P.ASM เป็น 8 จุด โดยใช้ค่าดังนี้

$$\frac{360}{8} = 45°$$

ฉะนั้นระยะห่างของจุดจะเท่ากับ 45° นำค่าองศาประจำจุดต่างมาอยู่ในรูปของ Sine y (y คือค่าขององศาประจำจุดต่างๆ) แล้วนำมาคูณด้วย 1,000 เนื่องจาก TMS320C31 เป็น Fired point

จุดที่ 1 .....

จุดที่ 2 .....

จุดที่ 3 .....

จุดที่ 4 .....

จุดที่ 5 .....

จุดที่ 6 .....

จุดที่ 7 .....

จุดที่ 8 .....

7) นำค่าไปใส่ในโปรแกรม SINE4P.ASM ในบรรทัดที่ 9 และเปลี่ยนค่า LENGTH ในบรรทัดที่ 10 เป็น 8 เพื่อทำการวน 8 รอบ

8) ทำการคอมไพล์และรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) วาดรูปสัญญาณ

สรุปผลการทดลอง

.....

.....

.....

คำถามท้ายการทดลอง

ถ้าเปลี่ยน N = 10 จุด ความถี่สุ่มเป็น 20k Hz จะได้ความถี่เอาต์พุตเท่าใด ?



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5

Fast Fourier Transform (FFT)

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถเขียนโปรแกรมคำนวนหาค่า Fast Fourier Transform ได้
2. เพื่อให้นักศึกษาสามารถอธิบายวิธีการพร็อตสเป็คตรัม จากค่าของ Fast Fourier Transform ได้
3. เพื่อให้นักศึกษาสามารถใช้งานโปรแกรม DSK Debugger ในการรันโปรแกรม Fast Fourier Transform ได้
4. เพื่อให้นักศึกษาสามารถเปรียบเทียบวิธีการคำนวณค่า FFT กับ DFT ได้

ทฤษฎี

การแปลงฟูเรียร์ (fourier transforms) เป็นเครื่องมือที่มีความสำคัญที่ใช้กันบ่อย ๆ ในระบบปฏิบัตงาน ทางสัญญาณดิจิตอล โดยจุดมุ่งหามยของการ Transfrom ก็คือเป็นการแปลงให้ทราบว่าจาก Time domain เป็น Frequency domain หรือการแปลงกลับจาก Frequency domain เป็น Time domain ผลจากการคำนวณจะเป็นเครื่องมือของ Fourier Tranfrom ทำให้ทราบค่า ของ Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) เป็นวิธีการที่ใช้ในการคำนวณ Discrete Fourier Fourier Transforms (DFT) ให้เร็วขึ้นหรือให้มีลักษณะที่เป็นเรียลไทม์ขึ้น

ที่มาของ FFT โดยอาศัยหลักการของ DFT

วิธีการเสนอโดย คูลีย์ และทูคีย์ ซึ่งได้กล่าวมาแล้วนั้น ความเป็นจริง การแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา x(m) ที่มีขนาด N จุด ออกเป็นสองลำดับสัญญาณที่มีความยาว N/2 จุดเท่ากัน ทั้งสองลำดับนี้หน้าชื่อเรียกว่า ลำดับสัญญาณคู่ และลำดับสัญญาณคี่โดยที่ลำดับสัญญาณคู่ เกดจากการการเอาลำดับในตำแหน่งเป็นเลขคู่มาเรียงที่เหลือก็เป็นลำดับสัญญาณคี่ ดังนั้นถ้าเรานิยามให้ x_2 (m) เป็นลำดับคู่และลำดับที่ x_0(m) ตามลำดับเพราะฉะนั้น

$$x_2(m) = x(2m) \quad ; m = 0,1,\ldots,(N/2)-1$$

$$x_0(m) = x(2m+1) \quad ; m = 0,1,\ldots,(N/2)-1 \qquad (5.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$X(k) = \sum_{M=0}^{N-1} x_2(m)(W_N)^{km} + \sum_{M=0}^{N-1} x_0(m)(W_N)^{km}$$

$$= \sum_{M=0}^{(N-2)-1} x(2m)(W_N)^{2km} + \sum_{M=0}^{(N-2)-1} x(2m+1)(W_N)^{(2m+1)k} \qquad (5.2)$$

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 ของลำดับความยาว N/2 จุด เพราะฉะนั้น จัดพจน์ใหม่ได้ คือ

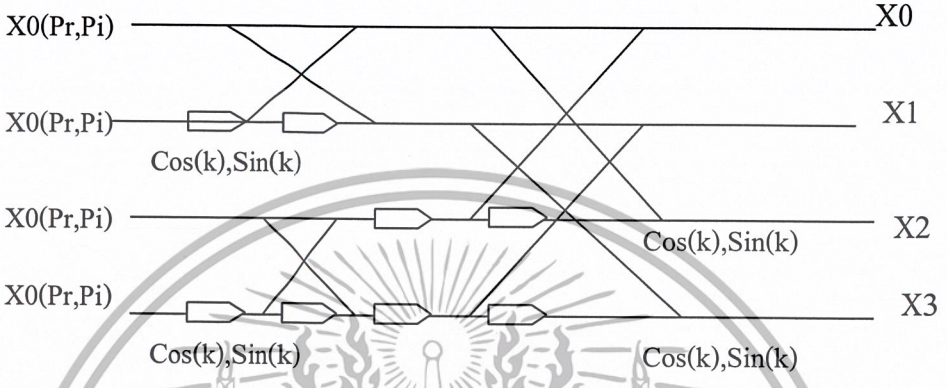
$$X(k) = \sum_{M=0}^{(N-1)-1} x_2(m)(W_{N/2})^{km} + \sum_{M=0}^{(N-2)-1} x0(m)(W_{N/2})^{km}$$

$$X(k) = X_1(k) + (W_n)^k X_2(k) \qquad (5.3)$$

โดยที่ $x_1(k)$ และ $X_2(k)$ แทนผลการแปลง DFT ขนาด N/2 จุดของลำดับ $X_2(m)$ และ X_0 (ตามลำดับ แสดงให้เห็นว่าการคำนวณ DFT ขนาด N จุดสามารถแบ่งคำนวณย่อยออกเป็น การ คำนวณ DFT ขนาด N/2 จุดสองอันดับได้และข้อสำคัญคือ การคูณจำนวนเชิงซ้อนจะลดลงเหลือ $2(N/2)^2 = N^2/2$ ครั้ง ซึ่งจะเห็นว่าลดการคำนวณไปถึง 50 เปอร์เซ็นต์ โดยอาศัยหลักเดียวกัน ถ้าเราแบ่งทอนลำดับ $X_2(m)$ และ $X_0(m)$ ออกเป็นลำดับคู่และลำดับคี่ลงไปอีกตามลำดับในที่สุดเหลือเป็นลำดับขนาด 2 จุด หรือกล่าวได้ว่า การคำนวณการแปลง DFT ขนาด N จุดทำการซอยเพื่อลำดับ x(n0 ออกเป็นทีละครึ่งจนเหลือการคำนวณ DFT ขนาด 2 จุด โดยที่ Fast Fourier Transforms (FFT) สามารถแบ่งออกได้ 2 ชนิดใหญ่ๆ คือ ชนิดทอนเวลาและชนิดลดทอนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการทดลองใบงานนี้จะใช้วิธีลดทอนเวลาดังรูปที่ จ.13 โดยใช้สมการที่ (5.3) และสมการที่ (5.4) ในการคำนวณ

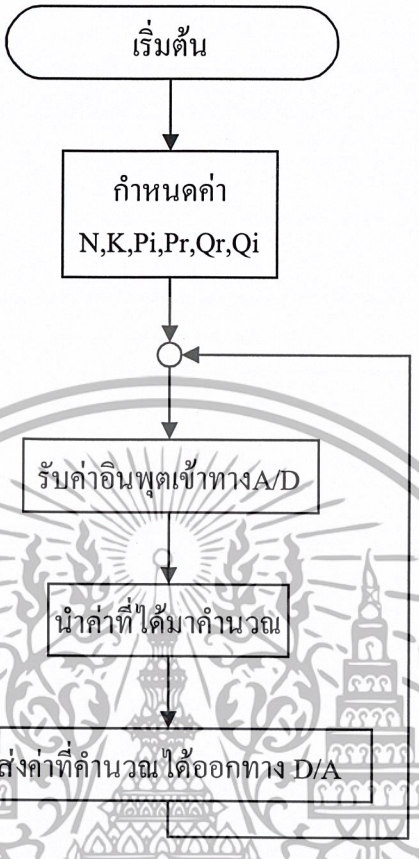


รูปที่ จ.13

$$\Pr n = [\Pr + QrCos(k) + QiSin(K)] + j[Pi + QiCos(k) - Sin(k)] \quad (5.3)$$

$$Qm = [\Pr - QrCos(k) - QiSin(k)] + j[Pi - QiCos(k) + Sin(k)] \quad (5.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.14 ผังการทำงานของโปรแกรม Fast Fourier Transform

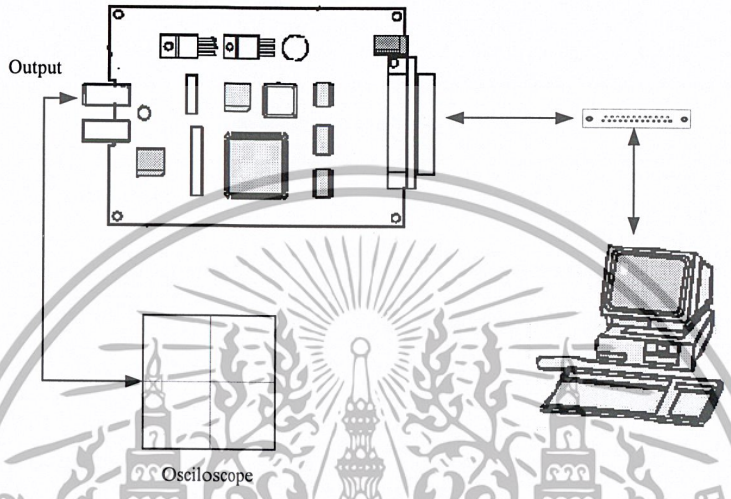
อุปกรณ์

1) ชุดฝึกปฏิบัติการการประมวลผลสัญญาณเชิงเลขด้วย TMS320C31	1 ชุด
2) เครื่องคอมพิวเตอร์	1 เครื่อง
3) โปรแกรม Dsktool	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูป



รูปที่ จ.15 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อนโปรแกรมดังนี้

```
/*FFT.C - REAL-VALUED FFT WITH 128 POINTS. CALLS FFT_RL.ASM/

#include "math.h"                    /*standard library func  */

#include "aiccomc.c"                 /*AIC comm routines      */

#define N 128                        /*size of FFT            */

#define M 7                          /*number of stages       */

volatile int index = 0;              /*input_output index     */

float *IO_buffer, *data, *temp;      /*--> array buffers      */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int AICSEC[4] = {0x162C,0x1,0x3872,0x67};  /*AIC config data       */

extern void fft_rl(int, int, float *);          /*fft function protype   */

void c_int05()                                  /*interrupt handler func */

{

 PBASE[0x48] = ((int)(IO_buffer[index])) << 2;        /*output data*/

IO_buffer[index] = (float)(PBASE[0x4C] << 16 >> 18);  /*input data */

 if (++index >= N) index = 0;          /*increment index, reset = N*/

}

main()

{

 int loop;                              /* declare variable       */

 float real, img;                       /* declare variables      */

AICSET_I();                             /*config AIC for interrupt  */

IO_buffer = (float *) calloc(N, sizeof(float)); /*input_out buffer  */

data = (float *) calloc(N, sizeof(float));     /* fft data buffer      */

while (1)                               /* create endless loop     */

{

fft_rl(N, M, (float *)data);            /*call FFT function       */

  data[0] = sqrt(data[0]*data[0])/N;          /*magnitude of X(0) */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 for (loop = 1; loop < N/2; loop++)    /*calculate X(1)..X(N/2-1)  */

{

  real = data[loop];                          /*real part  */

img = data[N-loop];                           /*imaginary part   */

data[loop] = sqrt(real*real+img*img)/N;   /*find magnitude   */

}

 data[N/2] = sqrt(data[N/2]*data[N/2])/N;  /*magnitude of X(N/2)    */

 for (loop = N/2+1; loop < N; loop++)      /*X(N/2+1).. X(N-1)*/

  data[loop] = data[N-loop];                          /*use symetry  */

 while (index);                 /      *wait till IO_buffer empty */

emp = data;                              *temp => data buffer       */

 data = IO_buffer;                 /*IO_buffer-->data buffer   */

 IO_buffer = temp;                 /*data buffer->new IO_buffer*/

 IO_buffer[0] = -2048;                 /*sync pulse,negative spike */

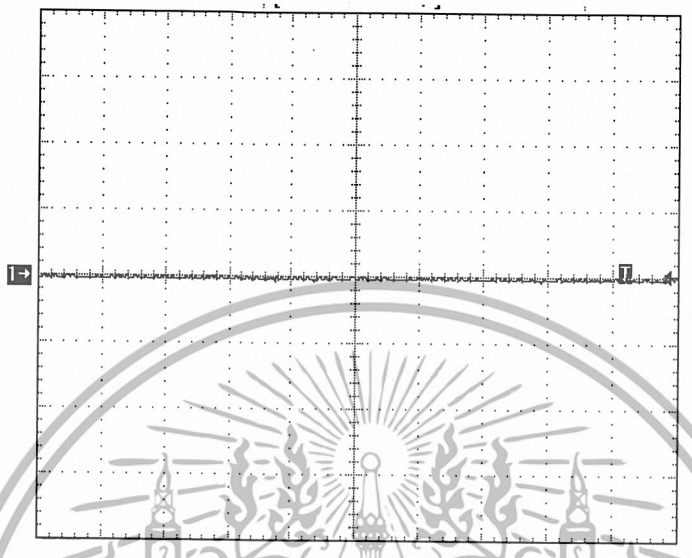
}

}
```

รูปที่ จ.16 โปรแกรม Fast Fourier Transform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ป้อนคลื่น Sine Wave ให้กับบอร์ด DSK TMS320C31 บันทึกผลการทดลอง



Volt/Div=_____ Time/Div=_____ f=_____Hz

สรุปผลการทดลอง

.....

.....

.....

คำถามท้ายการทดลอง

จงอธิบายหลักการของวิธีการพร๊อต สเป็กตรัม ของ Fast Fourier Tranforms ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่6

เอฟไออาร์ฟิลเตอร์ (FIR Filter)

จุดประสงค์เชิงพฤติกรรม

1.เพื่อให้นักศึกษาสามารถอธิบายหลักการทำงานของฟิลเตอร์แบบเอฟไออาร์ได้

2.เพื่อให้นักศึกษาสามารถบอกวิธีกาหาค่าสัมประสิทธิ์ของ เอฟไออาร์ได้

3.เพื่อให้นักศึกษาสามารถเขียนโครงสร้างและเขียนไดอะแกรม FIR filter ได้

ทฤษฎี

วงจรกรองแบบเอฟไออาร์ (FIR Filter)

สมการที่ (1) เป็นสมการที่ใช้ออกแบบ

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad (6.1)$$

จะเห็นได้ว่าสมการนี้เป็นสมการของดิจิตอลฟิลเตอร์ที่มี a =0 โดย b จะมีจำนวนเท่ากับจำนวนอิมพัลส์เรสปอนส์ ของฟิลเตอร์ ตามที่ได้ออกแบบไว้

จากสมการของวงจรกรองเอฟไออาร์ จะเห็นได้ว่าเป็นรูปแบบของสมการคอนโวลูชัน (Convolution Equation)ดังสมการที่(2)

$$y(n) = \sum_{k=0}^{M} h(n-k)xz^{-1} \qquad (6.2)$$

การหาสัมประสิทธิ์ของ FIR Fillter

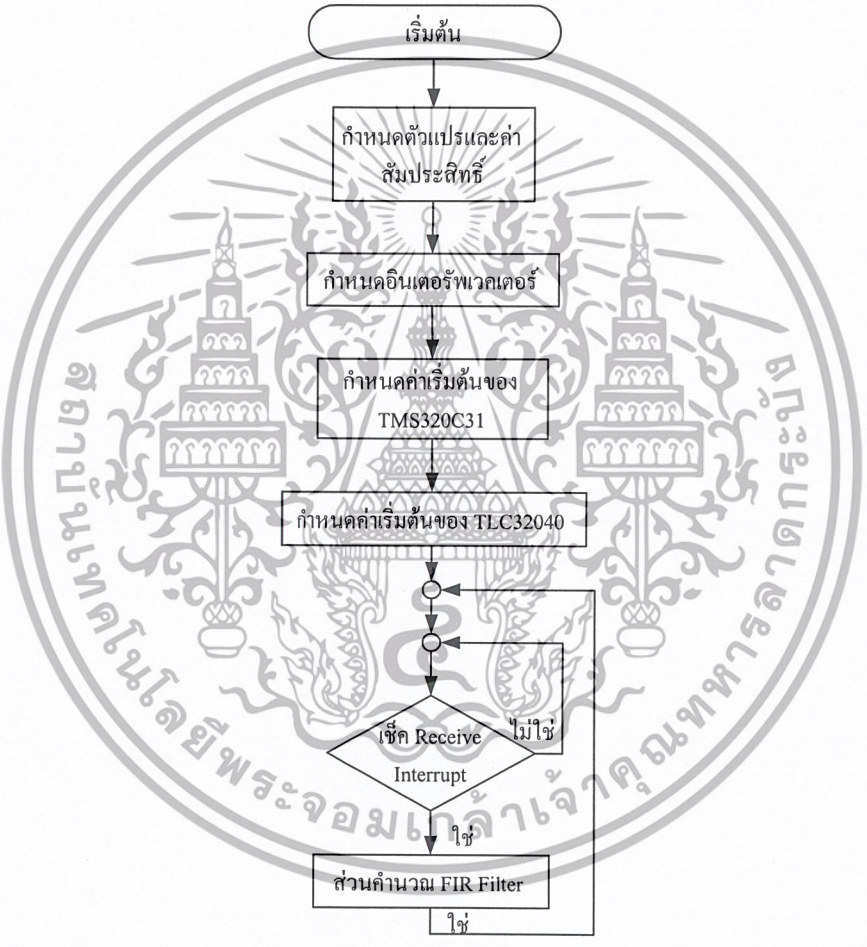
$$y(n) = \begin{cases} \frac{1}{\pi n} \quad (\sin\omega 2n - \sin\omega 1n) \quad ; n \neq 0 \\ \frac{1}{\pi} \quad (\omega 1 - \omega 2); n = 0 \end{cases}$$

เมื่อเราได้ค่า h(n) แต่ละตัว จากนั้นต้องทำการคูณด้วยค่า Windows Hamming จากสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\omega(\mathrm{n}) = \begin{cases} 0.5 + 0.5\cos\left(\dfrac{\pi n}{1 + (M/2)}\right) & ;-M/2 \le n \le M/2 \\ 0; & ;|n| > M/2 \end{cases}$$

$$h(n) = r(n) * \omega(n)$$



รูปที่ จ.17 ผังการทำงานโปรแกรมฟิลเตอร์แบบ FIR

อุปกรณ์

1) ชุดฝึกปฏิบัติการการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) ซิกแนลเจนเนอร์เรเตอร์	1 เครื่อง

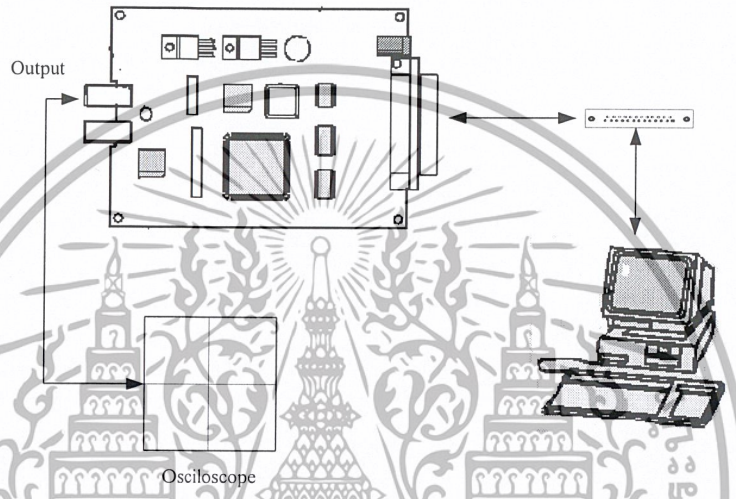
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เครื่องคอมพิวเตอร์ 1 เครื่อง

5) แผ่นดิสก์โปรแกรม DSK 1 แผ่น

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ จ.18



รูปที่ จ.18 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อนโปรแกรมดังนี้ โปรแกรม LP 11 SIM.ASM

```
;LP11SIM.ASM - FIR LOWPASS FILTER WITH 11 COEFF FOR SIMULATION
        .start  ".text",0x809900   ;        where text begins
        .start  ".data",0x809C00  ;where data begins
        .data                               ;data section
IN_ADDR     .word   INB             ;starting address for input
OUT_ADDR    .word   OUTB            ;starting address for output
XB_ADDR     word    XN+LENGTH-1     ;bottom address of circ buffer
HN_ADDR     .word   COEFF           ;starting addr of coefficients
COEFF       .float  0           ;H10
        .float  0.0468, 0.1009, 0.1514, 0.1872, 0.2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .float  0.1872, 0.1514, 0.1009, 0.0468
H0      .float  0                       ;H0
LENGTH  .set    H0-COEFF+1      ;# of coefficients
INB     .float  10000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
OUTB    .float  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
        .brstart "XN_BUFF",16           ;align samples buffer
XN      .sect   "XN_BUFF"               ;section for input samples
        .loop   LENGTH                  ;buffer size for samples
        .float  0.0                     ;initialize samples to zero
        .endloop                        ;end of loop
        .entry  BEGIN                   ;start of code
        .text
BEGIN       LDP     @0x809800       ;init to data page 128
        LDI     LENGTH,BK               ;size of circular buffer
        LDI     @XB_ADDR,AR1            ;last sample address ->AR1
        LDI     @IN_ADDR,AR2            ;input address      -->AR2
LDI     @OUT_ADDR,AR3       ;output address     -->AR3
FILT    LDI     LENGTH-1,AR4            ;length in AR4
LOOP    LDF     *AR2++,R3               ;input new sample
STF     R3,*AR1++%                      ;store newest sample
        LDI     @HN_ADDR,AR0            ;AR0 points to H(N-1)
        LDF     0,R0                    ;init R0
        LDF     0,R2                    ;init R2
        RPTS    LENGTH-1                ;repeat LENGTH-1 times
        MPYF3   *AR0++,*AR1++%,R0 ;R0 = HN*XN
        ADDF3   R0,R2,R2                ;accumulation in R2
        DBNZD   AR4,LOOP                ;delayed branch until AR4<0
        ADDF    R0,R2                   ;last mult result accumulated
        FIX     R2,R7                   ;convert float R2 to integer R7
        STI     R7,*AR3++               ;store into output buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

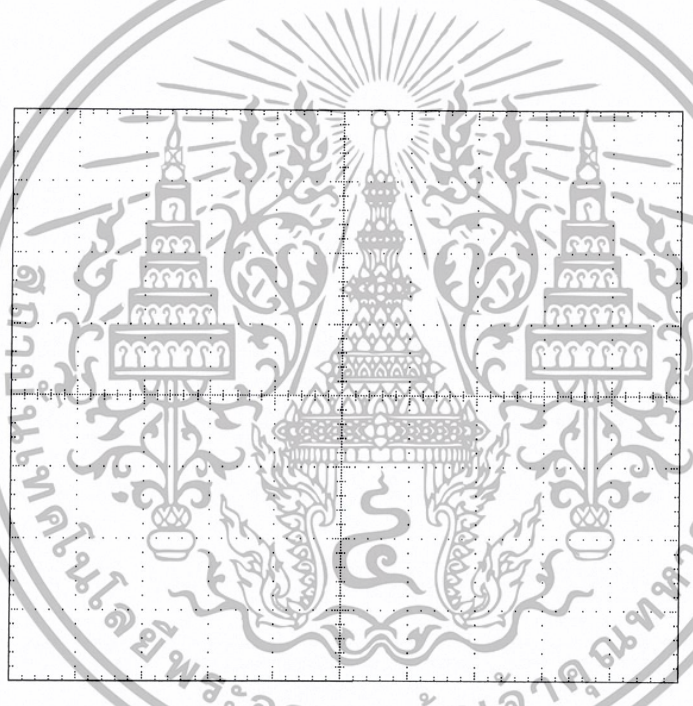
```
WAIT    BR      WAIT                    ;wait
        .end                            ;end
```

รูปที่ จ.19 โปรแกรม LP11SIM.ASM

3) ปรับความถี่อินพุตที่ 200 Hz 3Vp-p

4) คอมไพล์และรันโปรแกรม

5) บันทึกผลการทดลอง



Volt/DIV = _____ TIME/DIV = _____ f =_____Hz

สรุปผลการทดลอง

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

จงบอกข้อดีข้อเสียของวงจรกรองเชิงเลขแบบเอฟไออาร์ ?



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ฉ
เฉลยใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1
การใช้งานบอร์ด DSK TMS320C31 เบื้องต้น

จุดประสงค์เชิงพฤติกรรม

1) เพื่อให้นักศึกษาสามารถใช้งานบอร์ด DSK TMS320C31 เบื้องต้นได้อย่างถูกต้อง

ทฤษฎี

แผงวงจร DSK TMS320C31 สามารถใช้ภาษาในการเขียนอยู่ถึง 2 ภาษา จึงต้องมีการคอมไพล์ให้เป็นภาษาเครื่องเพื่อให้บอร์ดDSKทำงานตามโปรแกรมที่ป้อนให้กับบอร์ด DSK TMS320C31 การเขียนโปรแกรมจะเขียนได้จากคอมพิวเตอร์ โดยใช้หน้าต่างของDOS จึงทำให้มีคำสั่งที่ยุ่งยากพอสมควร แต่สามารถเข้าใจง่าย

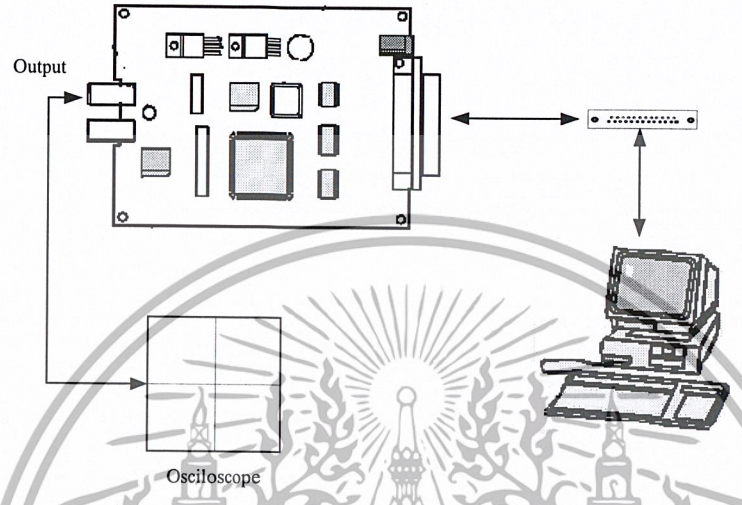
อุปกรณ์

1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง
4) แผ่นดิสก์โปรแกรมDSK	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ ฉ.1



รูปที่ ฉ.1 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) เข้าไปที่ DSKTOOL ใช้คำสั่ง DSK3A ตามด้วยชื่อไฟล์นามสกุล .asm ในใบงานใช้ชื่อไฟล์ชื่อ Sine4p.asm (DSK3aเป็นการแปลงไฟล์ DSK เพื่อใช้โหลดลงDSK TMS320C31)

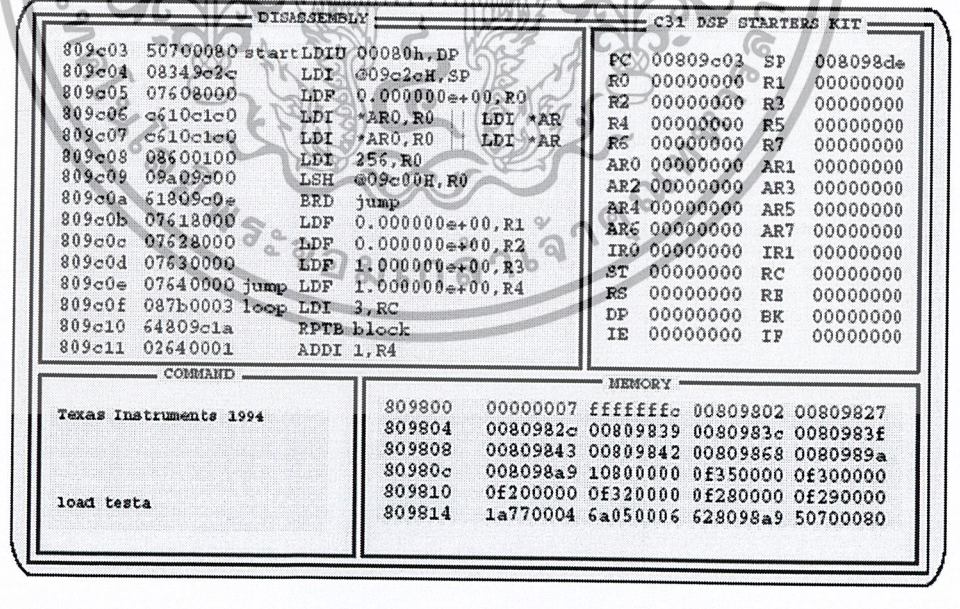
```
;SINE4P.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start ".text",0x809900          ;starting address for text
        .start ".data",0x809C00          ;start address for data
        .include "AICCOM31.ASM"  ;AIC communication routines
        .data                            ;data section
AICSEC  .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word  SINE_VAL        ;address of sine values
        .brstart "SINE_BUFF",16  ;align sine table
SINE_VAL  .word  0,1000,0,-1000          ;sine values
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LENGTH      .set   4      ;length of circular buffer
            .entry BEGIN         ;start of code
            .text                ;text section
BEGIN   CALL   AICSET    ;initialize AIC
            LDI    LENGTH,BK  ;BK = size of buffer
            LDI    @SINE_ADDR,AR1    ;AR1 = addr of sine values
LOOP        LDI    *AR1++%,R7        ;R7 = table value
            CALL   AICIO_P           ;call AICIO for output
            BR     LOOP              ;loop back
            .end          ;end
```

รูปที่ ฉ.2 โปรแกรม Sine4p.asm

3) ใส่คำสั่ง DSK3D เพื่อเข้าสู่หน้าต่าง Debugger ดังรูปที่ ฉ.2



รูปที่ ฉ.3 หน้าต่าง Debugger

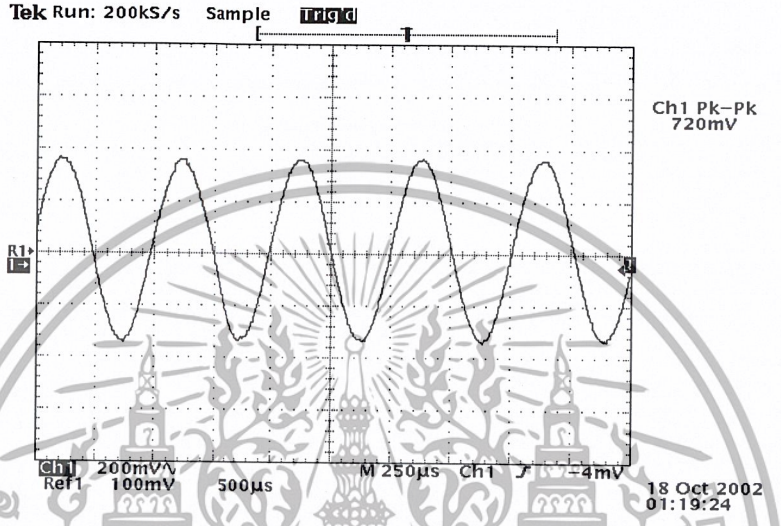
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการ Reset บอร์ด DSK TMS320C31 โดยใช้คำสั่ง Resetในหน้าต่าง Debugger

5) ใช้คำสั่งLoad ชื่อ .dsk (จากข้อ2) ดังนี้ Load Sine4p.dsk

6) ใส่คำสั่ง RUN เพื่อรันโปรแกรม หรือกด F5

7) วาดรูปผลการทดลองจากสโคป



Volt/Div= 200mV Time/Div= 250uS f= 2kHz

สรุปผลการทดลอง

จากการทดลองรันโปรแกรมทำให้ทราบคำสั่งในการใช้งานเบื้องต้นของ Debugger ของบอร์ด DSK TMS320C31 บนหน้าต่าง Debugger

คำถาม

ถ้าต้องการโหลดไฟล์ต้องทำอย่างไรจงอธิบาย ?

ทำการพิมพ์ LOAD ตามด้วยชื่อที่มีนามสกุลเป็น .DSK เช่นดังตัวอย่าง LOAD SINE4P.DSK <ENTER> หน้าต่างของ Debugger จะแสดงตำแหน่งเริ่มต้นของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2
การใช้ชุดคำสั่งแบบง่าย

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถอธิบายการใช้ชุดคำสั่งแบบง่ายได้
2. เพื่อให้นักศึกษาสามารถใช้โปรแกรมการคูณเมทริกซ์ได้ถูกต้อง

ทฤษฎี

ในการศึกษาชุดคำสั่งนั้น ต้องเริ่มต้นจากโปรแกรมง่ายๆ เพื่อที่จะได้เข้าใจหลักการของชุดคำสั่งได้อย่างรวดเร็ว ในใบงานนี้จึงขอยกตัวอย่างการใช้โปรแกรมการคูณเมทริกซ์ ดังนี้

โดยการใช้โปรแกรม การคูณเมทริกซ์ขนาด(3x3)*(3x1)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} x \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} (1x1) + (2x2) + (3x3) \\ (4x1) + (5x2) + (6x3) \\ (7x1) + (8x2) + (9x3) \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 32 \\ 50 \end{bmatrix} \quad \text{Ans}$$

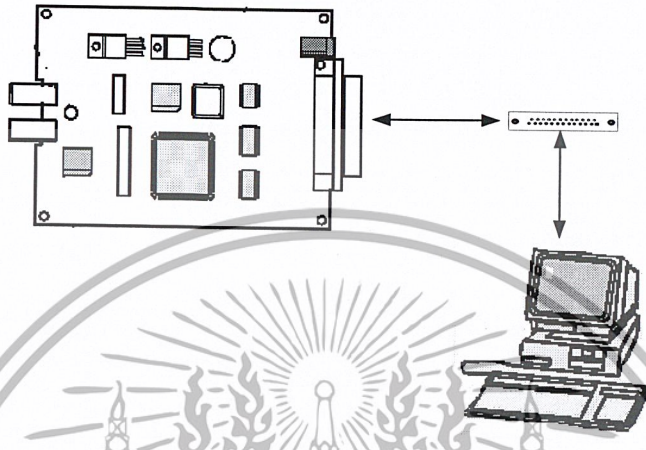
อุปกรณ์

1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) เครื่องคอมพิวเตอร์	1 เครื่อง
3) แผ่นดิสก์โปรแกรม DSK	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูป



รูปที่ ฉ.4 การต่อชุดทดลองกับอุปกรณ์

2) ป้อนโปรแกรมดังนี้

```
;MATRIX.ASM - MATRIX/VECTOR MULTIPLICATIO
; (3x3)x(3x1) = (3x1)
        .start ".data",0x809C00          ;starting address for data
        .start ".text",0x809900          ;starting address for text
        .data                            ;data section
A               .float 1,2,3,4,5,6,7,8,9 ;values for matrix A
B               .float 1,2,3             ;values for matrix B
A_ADDR          .word  A                 ;starting address of matrix A
B_ADDR          .word  B                 ;starting address of matrix B
OUT_ADDR        word $                   ;output (current) address
        .entry BEGIN                     ;start of code
        .text                            ;text section
BEGIN           LDP    A_ADDR                    ;init to data page 128
        LDI    @A_ADDR,AR0               ;AR0=starting address of A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LDI   @B_ADDR,AR1      ;AR1=starting address of B
        LDI   @OUT_ADDR,AR2    ;AR2= output address
        LDI   3,R4             ;R4 used as LOOPI counter
LOOPI   LDF   0,R0             ;initialize R0=0
        LDI   2,AR4            ;AR4 used as LOOPJ counter
LOOPJ   MPYF3 *AR0++,*AR1++,R1 ;R1=A[I,J]*B[J]
        ADDF3 R1,R0,R0         ;accumulate in R0
        DB    AR4,LOOPJ        ;decrement AR4.Branch until AR4<0
        FIX   R0,R2            ;convert R0 from float to integer
        STI   R2,*AR2++        ;store integer output in memory
        LDI   @B_ADDR,AR1      ;AR1=starting address of matrix B
        SUBI  1,R4             ;decrement R4
        BNZ   LOOPI            ;branch while R4 is not zero
        BR    $                ;branch to current addr (itself)
```

รูปที่ ฉ.5 โปรแกรม MATRIX.ASM

3) ทำการคอมไพล์ โปรแกรมตามใบงานที่ 1

4) ทำการโหลดโปรแกรมในหน้าต่าง Debugger โปรแกรมจะเริ่มที่ตำแหน่ง 809900H และข้อมูลจะเริ่มที่ตำแหน่ง 809C00H ในบรรทัดที่ 5 และ 6จะอยู่ที่ตำแหน่งเริ่มต้นของตำแหน่งข้อมูล ดังนั้น คำตอบจะเริ่มที่ตำแหน่ง 809c0eH เราต้องการดูค่าที่ตำแหน่ง 809c0eH โดยการใช้คำสั่ง memd 0x809c0e (เป็นคำสั่งที่ใช้ดูเลขจำนวนเต็ม)

5) กด F8 เพื่อทำการรันทีละสเต็ป ทำการกดไปเรื่อยๆ สังเกตการเปลี่ยนแปลงค่าที่ตำแหน่ง809c0eH

6) ทำการบันทึกค่า ที่อยู่ในตำแหน่งดังนี้

809c0eH = 14

809c0fH = 32

809c10H = 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลองทำให้ทราบชุดคำสั่งอย่างง่าย เช่น LDI 2,AR4 เป็นการเก็บค่า 2ไว้ในรีจีสเตอร์ AR4

คำถามท้ายการทดลอง

ถ้าต้องการเปลี่ยนสมการเป็น $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} x \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$ จะต้องเพิ่มโปรแกรมส่วนใด ?

```
;MATRIXNEW.ASM - MATRIX/VECTOR MULTIPLICATION (3x3)x(3x2) = (3x2)
        .start ".data",0x809C00          ;starting address for data
        .start ".text",0x809900          ;starting address for text
        .data                            ;data section
A       .float 1,2,3,4,5,6,7,8,9         ;values for matrix A
B       .float 1,2,3                     ;values for matrix 1,2,3
C       .float 4,5,6                     ;NEW VALUES TO MATRIX
A_ADDR  .word  A                         ;starting address of matrix A
B_ADDR  .word  B                         ;starting address of matrix B
C_ADDR  .word  C                         ;NEW ADDRESS FOR 4,5,6
OUT_ADDR word  $                         ;output (current) address
        .entry BEGIN                     ;start of code
        .text                            ;text section
BEGIN   LDP    A_ADDR                    ;init to data page 128
        LDI    @A_ADDR,AR0               ;AR0=starting address of A
        LDI    @B_ADDR,AR1               ;AR1=starting address of 1,2,3
        LDI    @C_ADDR,AR3               ;AR3=STARTING ADD OF 4,5,6
        LDI    @OUT_ADDR,AR2             ;AR2= output address
        LDI    3,R4                      ;R4 used as LOOPI counter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOPI     LDF   0,R0          ;initialize R0=0
          LDI   2,AR4         ;AR4 used as LOOPJ counter
LOOPJ     MPYF3 *AR0++,*AR1++,R1 ;R1=A[I,J]*B[J]
          ADDF3 R1,R0,R0      ;accumulate in R0
          DB    AR4,LOOPJ     ;decrement AR4.Branch until AR4<0
          FIX   R0,R2         ;convert R0 from float to integer
          STI   R2,*AR2++     ;store integer output in memory
          LDI   @B_ADDR,AR1   ;AR1=starting address of matrix B
          SUBI  1,R4          ;decrement R4
          BNZ   LOOPI         branch while R4 is not zero
          LDP   A_ADDR
          LDI   @A_ADDR,AR0
          LDI   @C_ADDR,AR3
          LDF   0,R1          ;initialize R1=0
          LDI   3,R4          ;R4 used as LOOPCI counter
LOOPCI    LDF   0,R0          ;initialize R0=0
          LDI   2,AR4         ;AR4 used as LOOPJC counter
LOOPJC    MPYF3 *AR0++,*AR3++,R1   ;R1=A[I,J]*B[J]
          ADDF3 R1,R0,R0      ;accumulate in R0
          DB    AR4,LOOPJC    ;decrement AR4.Branch until AR4<0
          FIX   R0,R2         ;convert R0 from float to integer
          STI   R2,*AR2++     ;store integer output in memory
          LDI   @B_ADDR,AR1   ;AR1=starting address of matrix B
          SUBI  1,R4          ;decrement R4
          BNZ   LOOPCI        ;branch while R4 is not zero
          BR    $             ;branch to current addr (itself)
```

รูปที่ ฉ.6 โปรแกรม MATRIXNEW.ASM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
การทดสอบการเชื่อมต่อหน่วยความจำภายนอก

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถบอกหลักการเชื่อมต่อหน่วยความจำภายนอกได้ถูกต้อง

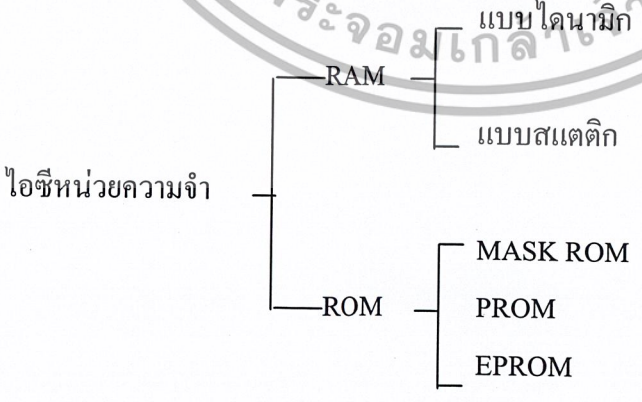
2.เพื่อให้นักศึกษาสามารถอธิบายหลักการการติดต่อกับหน่วยความจำภายนอกได้อย่างถูกต้อง

ทฤษฎี

หน่วยความจำในระยะแรกจะเป็นแหวนแม่เหล็ก (Core Memory) เป็นกระบอกแม่เหล็ก (Magnetic Drum) ซึ่งเป็นอุปกรณ์แม่เหล็กไฟฟ้าและเชิงกลไฟฟ้า ทำให้การติดต่อกับ CPU ทำได้ยาก วงจรซับซ้อนมาก การทำหน่วยความจำเป็นไอซีได้จึงเป็นเรื่องสำคัญทำให้วงจรในคอมพิวเตอร์ง่ายลงมาก

เมื่อเทคโนโลยีในการทำ MOS ไอซีเจริญขึ้น ก็สามารถทำไอซีที่มีหน่วยความจำที่มีความจุสูงขึ้น สามารถใช้งานในทางปฏิบัติได้ดีขึ้น จึงทำให้ไมโครคอมพิวเตอร์มีขนาดเล็กลงและสามารถใช้งานได้ดีเยี่ยม

ถ้าแบ่งชนิดของไอซีหน่วยไอซีหน่วยความจำออกตามคุณสมบัติและการใช้งานจะสามารถแบ่งออกได้ดังนี้



รูปที่ ฉ.7 การแบ่งประเภทของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

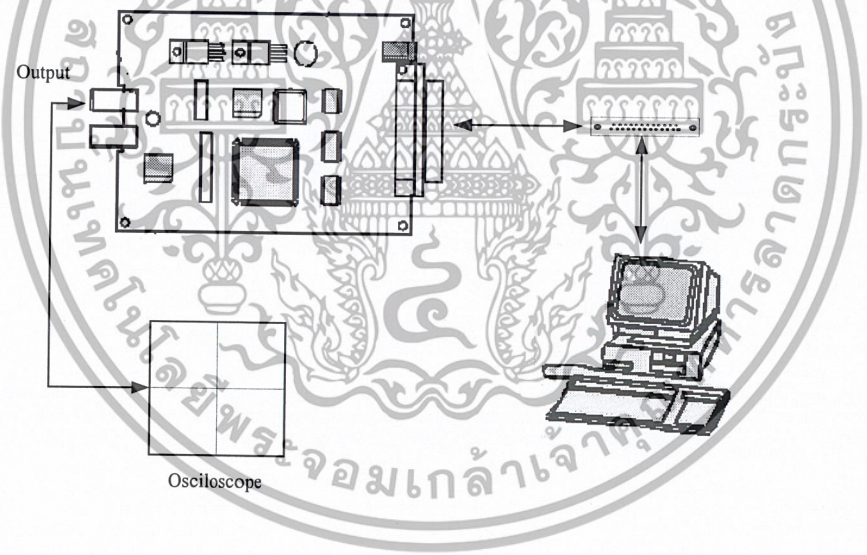
การต่อหน่วยความจำภายนอกต้องมีการต่อวงจรถอดรหัส โดยใช้ขาแอดเดรสมาถอดรหัส เพื่อจะไปเป็นตัวเลือกตัวไอซี ในการบันทึกข้อมูล

อุปกรณ์

1) ชุดทดลอง การประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง
4) แผ่นดิสก์โปรแกรมDSK	1 แผ่น

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ ฉ.8



รูปที่ ฉ.8 การต่อชุดทดลองกับอุปกรณ์

2) การทดสอบหน่วยความจำภายนอก โดยการกดคำสั่งdsk3d <Enter>

3) พิมพ์คำสั่ง mem 0x100000 เนื่อจากหน่วยความจำภายนอกต่ออยู่ที่ตำแหน่งแอดเดรส 100000H

4) ที่ตำแหน่ง 100000H จะแสดงข้อมูลขณะนั้นอยู่สังเกตค่า บันทึกผลการทดลอง

100000H 0000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) กดALT+m หรือ กดF1 เพื่อดูคำอธิบายจากนั้นทำการป้อนค่า โดยใช้คำสั่ง mm ใน Debuger เช่น mm 0x100000 จะเป็นการใส่ค่าแอดเดรสที่ 100000H

6) ใส่ค่าดังต่อไปนี้ ตำแหน่ง 100000H ใส่ FFFF, ตำแหน่งที่ 100002H ใส่ FC ,ตำแหน่งที่ 100005H ใส่ 9999, ตำแหน่งที่ 100009.ใส่ 77C9 บันทึกผลการทดลอง

100000H เท่ากับ FFFF

100002H เท่ากับ FC

100005H เท่ากับ 9999

100009H เท่ากับ 77C9

7) โหลดโปรแกรม SIN_RAM.DSK สังเกตว่าจะเป็นตำแหน่งของหน่วยความจำภายนอก คือ ตำแหน่ง 100000H

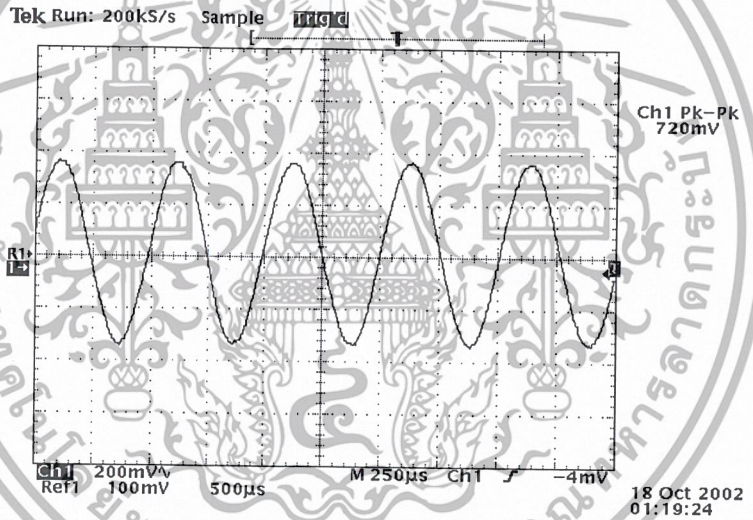
```
;SINE_RAM.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start   ".text",0x809900          ;starting address for text
        .start   ".data",0x100000  ;<NEW DATA ADDRESS>
        .include "AICCOM31.ASM"    ;AIC communication routines
        .data                              ;data section
AICSEC  .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR .word  SINE_VAL          ;address of sine values
        .brstart "SINE_BUFF",16    ;align sine table
SINE_VAL .word   0,1000,0,-1000            ;sine values
LENGTH  .set    4                  ;length of circular buffer
        .entry  BEGIN              ;start of code
        .text                              ;text section
BEGIN   CALL    AICSET                     ;initialize AIC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LDI     LENGTH,BK          ;BK = size of buffer
LDI     @SINE_ADDR,AR1     ;AR1 = addr of sine values
LOOP            LDI     *AR1++%,R7     ;R7 = table value
CALL    AICIO_P            ;call AICIO for output
BR      LOOP               ;loop back
.end                       ;end
```

รูปที่ ฉ.9 โปรแกรม SINE_RAM.ASM

8) บันทึกผลการทดลอง



Volt/Div 200mv Time/Div 250us f = 720 Hz

สรุปผลการทดลอง

จากการทดลองจะเห็นได้ว่าเมื่อต้องการเพิ่มหน่วยความจำให้กับบอร์ด DSK TMS 320C31 จะต้องมีการทดสอบหน่วยความจำนั้นตามใบงานข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

ถ้าหน่วยความจำภายนอกอยู่ที่ตำแหน่ง 700000H จะติดต่อกับหน่วยความจำได้อย่างไร จงอธิบาย ?

ตอบ ถ้าจะป้อนโปรแกรมต้องเริ่มที่ตำแหน่ง 700000H และถ้าจะป้อนข้อมูลก็ต้องเริ่มที่ตำแหน่ง 700000H เช่นกัน และจะสิ้นสุดเท่ากับขนาดของหน่วยความจำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
การสร้างรูปคลื่นไซน์ (Sine Wave)

จุดประสงค์เชิงพฤติกรรม

1.เพื่อให้นักศึกษาสามารถเขียนโปรแกรมบน DSK TMS320C31 ได้

2.เพื่อให้นักศึกษาสามารถบอกวิธีการสร้างรูปคลื่นไซน์จาก DSKTMS320C31ได้

ทฤษฎี

Sine Generator

การใช้งาน DSK TMS320C31 ในการผลิตสัญญาณไซน์เวฟ (Sine Wave) มีลักษณะดังนี้

คลื่นรูปไซน์ (Sine Wave) ที่สร้างโดย DSK TMS320C31 นั้น ในการกำหนดความถี่จะอาศัยหลักการสุ่มตัวอย่างความถี่ (Sampling Frequency) และการกำหนดจุด (N) โดยใช้สมการที่ (1)

$$F = \frac{Fs}{N}$$

F = ความถี่เอาต์พุต

Fs = ความถี่สุ่ม(Sampling Frequency)

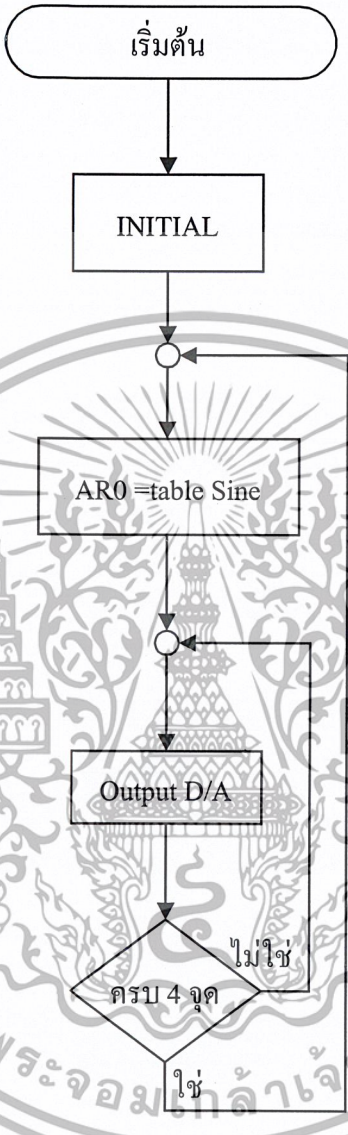
N = จำนวนจุดในการสุ่มสัญญาณ 1 คาบเวลา

จากหลักการของการมองค่าจากตาราง ซึ่งจำเป็นจะต้องคำนวณหาค่าไซน์ (Sine) ของแต่ละจุดหลังจากนั้นทำการสเกลค่าด้วยการคูณ 1,000 และนำค่าที่สเกลแล้วไปเก็บไว้ในหน่วยความจำ จากนั้นทำการเขียนโปรแกรมอ่านค่าจากหน่วยความจำ ส่งออกไปยัง D/A ของบอร์ด DSK TMS32CC31

ตัวอย่าเช่น ถ้าความถี่สุ่ม (Fs) มีค่า = 8 k Hz จำนวนจุดในการสุ่มค่าสัญญาณ 1 คาบเวลา = 4 จุด ฉะนั้นความถี่เอาต์พุต = $\frac{8\,kHz}{4} = 2\,kHz$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังการทำงาน



รูปที่ ฉ.10 ผังการทำงานของโปรแกรมกำเนิดรูปคลื่นไซน์

อุปกรณ์

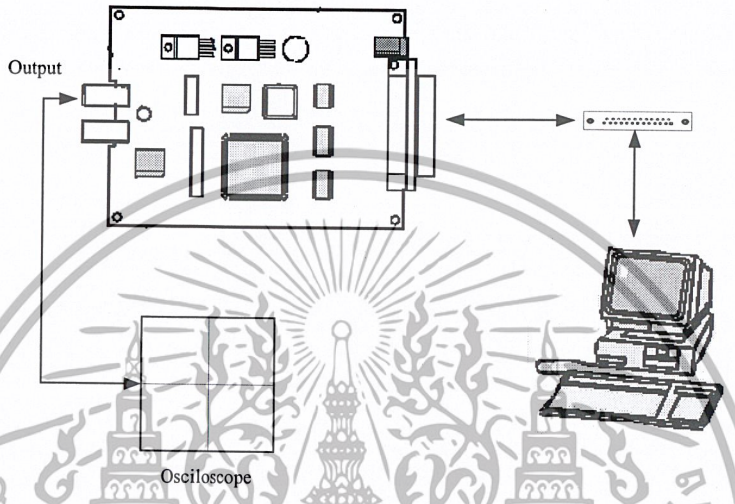
1) ชุดทดลองการประมวลผลสัญญาณเชิงเลข	1 ชุด
2) สโคป	1 เครื่อง
3) เครื่องคอมพิวเตอร์	1 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) แผ่นดิสก์โปรแกรม DSKTOOL 1 แผ่น

ลำดับขั้นตอนการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ ฉ.11



รูปที่ ฉ.11 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อนโปรแกรมดังนี้

```
;*AICCOM31.ASM - AIC COMMUNICATION ROUTINES –
;POLLING OR INTERRUPT
        .data           ;assemble into data section
PBASE   .word   808000h     ;peripheral base address
SETSP   .word   0E970300h   ;serial port set-up data
ATABLE  .word   AICSEC      ;SP0 AIC init table address
        .text           ;assemble into text section
AICSET  PUSH    AR0         ;save AR0
        PUSH    AR1         ;save AR1
        PUSH    R0          ;save R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PUSH   R1          ;save R1
LDI    @PBASE,AR0    ;AR0 -> 808000h
LDI    1,R0        ;timer CLK=H1/2*(AIC master CLK)
STI    R0,*+AR0(28h)  ;timer period reg(TCLK0=6.25 MHZ)
LDI    03C1h,R0     ;init timer global register
STI    R0,*+AR0(20h)  ;reset timer
LDI    62h,IOF      ;AIC reset = 0
LDI    @ATABLE,AR1   ;AR1 -> AIC init data
RPTS   99          ;repeat next instr 100 times
NOP                ;keep IOF low for a while
LDI    131h,R0      ;X & R port control register data
STI    R0,*+AR0(42h)  ;FSX/DX/CLKX=SP operational pins
STI    R0,*+AR0(43h)  ;FSR/DR/CLKR=SP operational pins
LDI    @SETSP,R0    ;RESET->SP:16 bits,ext clks,std mode
STI    R0,*+AR0(40h)  ;FSX=output&INT enable SP global reg
LDI    0,R0         ;R0 = 0
STI    R0,*+AR0(48h)  ;clear serial port XMIT register
OR     06h,IOF      ;bring AIC out of reset
LDI    03h,RC       ;RC=3 to transmit 4 values
RPTB   SECEND       ;repeat 4 data transmit of sec com
CALL   TWAIT        ;wait for data transmit
LDI    03h,R0       ;value for secondary XMIT request
STI    R0,*+AR0(48h)  ;secondary XMIT request to AIC
CALL   TWAIT        ;wait for data transmit
LDI    *AR1++(1),R0  ;AR1 -> next AIC init data
SECEND STI    R0,*+AR0(48h)  ;DTR = current AIC data
POP    R1          ;restore R1
POP    R0          ;restore R0
POP    AR1         ;restore AR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine
       AICSET_I   ;-CONFIG FOR INTERRUPT
        CALL   AICSET       ;call AICSET routine
        LDI    0h,IF        ;clear IF register
        OR     10h,IE       ;enable EXINT0 CPU interrupt
        OR     2000h,ST     ;global interrupt enable
        RETS                ;return from subroutine
    ;------TRANSMIT WAIT ROUTINE-----
TWAIT   PUSH   AR0          ;save AR0
        PUSH   R0           ;save R0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
TW1     LDI    *+AR0(40h),R0  ;R0=content of SP global control reg
        AND    02h,R0       ;see if transmit buffer is ready
        BZ     TW1          ;if not ready, try again
        POP    R0           ;restore R0
        POP    AR0          ;restore AR0
        RETS                ;return from subroutine
    ;--------------------AIC TRANSFER ROUTINE-------------
    AICIO_I   LDI    R7,R6      ;copy output to modify for AIC
        LSH    2,R6         ;two LSB must=0 for primary AIC comm
    IO          PUSH   AR0        ;save AR0
        LDI    @PBASE,AR0   ;AR0 -> 0808000h
        STI    R6,*+AR0(48h)  ;DTR = next data for AIC D/A
        LDI    *+AR0(4Ch),R6  ;R6 = DRR data from AIC A/D
        LSH    16,R6        ;left shift for sign extension
        ASH    -18,R6       ;right shift keeping sign
        POP    AR0          ;restore AR0
       RETS                 ;return from subroutine
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;--------------------AIC POLLING ROUTINE----------------
AICIO_P   CALL   TWAIT        ;wait for data to be transferred
          CALL   AICIO_I      ;call AIC transfer routine
          RETS                ;return from subroutine
SW_IO     PUSH   AR0          ;save AR0
          LDI    @PBASE,AR0   ;AR0 -> 0808000h
          LDI    R7,R6        ;copy output to modify for AIC
          LSH    2,R6         ;prepare for secondary AIC com
          OR     03h,R6       ;set two LSB for secondary com
          CALL   TWAIT        ;wait for data to be transferred
          CALL   IO           ;call AIC transfer routine
          CALL   TWAIT        ;wait for data to be transferred
          STI    R1,*+AR0(48h) ;DTR = next data for AIC control
          POP    AR0          ;restore AR0
          RETS                ;return from subroutine
;SUBROUTINES FOR PRIMARY OR AUXILIARY INPUT
IOPRI     PUSH   R1           ;save R1
          LDI    063h,R1      ;load secondary com data into R1
          CALL   SW_IO        ;call IO routine to switch inputs
          POP    R1           ;restore R1
          RETS                ;return from subroutine
IOAUX     PUSH   R1           ;save R1
          LDI    073h,R1      ;load secondary com data into R1
          CALL   SW_IO        ;call IO routine to switch inputs
          POP    R1           ;restore R1
          RETS                ;return from subroutine
```

รูปที่ ฉ.12 โปรแกรม AICCOM31.ASM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ทำการบันทึกเป็นไฟล์ชื่อ AICCOM31.ASM หลังจากนั้นทำการเขียนโปรแกรมต่อไปนี้

```
;SINE4P.ASM - GENERATES A SINE USING ONLY 4 POINTS
        .start  ".text",0x809900
        .start  ".data",0x809C00
        .include "AICCOM31.ASM"     ;AIC communication routines
        .data                       ;data section
AICSEC          .word   162Ch,1h,4892h,67h ;Fs = 8 kHz
SINE_ADDR       .word   SINE_VAL        ;address of sine values
        .brstart "SINE_BUFF",16     ;align sine table
SINE_VAL        .word   0,1000,0,-1000  ;sine values
LENGTH          .set    4               ;length of circular buffer
        .entry  BEGIN               ;start of code
        .text                       ;text section
BEGIN           CALL    AICSET          ;initialize AIC
        LDI     LENGTH,BK           ;BK = size of buffer
        LDI     @SINE_ADDR,AR1      ;AR1 = addr of sine values
LOOP            LDI     *AR1++%,R7      ;R7 = table value
        CALL    AICIO_P             ;call AICIO for output
        BR      LOOP                ;loop back
        .end                        ;end
```

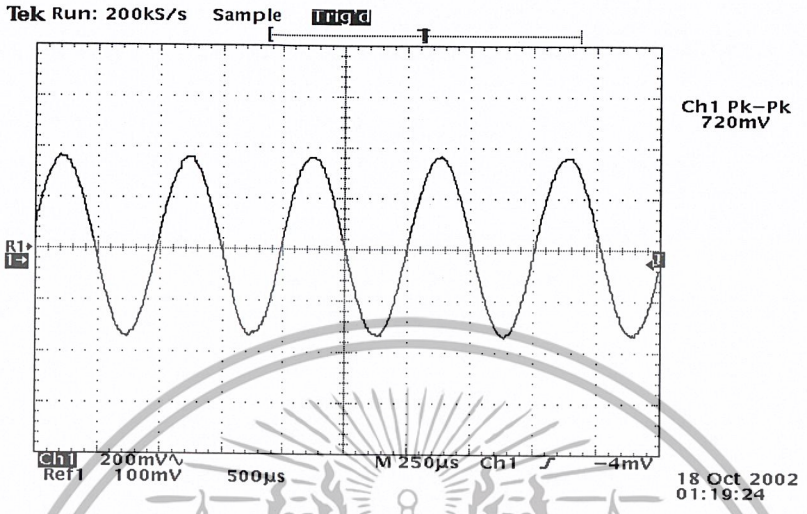
รูปที่ ฉ.13 โปรแกรม SINE4P.ASM

ทำการบันทึกเป็น ชื่อ SINE4P.ASM

4) คอมไพล์และรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) วาดรูปสัญญาณ



Volt/Div= 200mv Time/Div= 250us f= 2 kHz

6) ทดลองเปลี่ยนค่า ในบรรทัดที่ 9 ของโปรแกรม SINE4P.ASM เป็น 8 จุด โดยใช้ค่าดังนี้

$$\frac{360}{4} = 45°$$

ฉะนั้นระยะห่างของจุดจะเท่ากับ 45° นำค่าองศาประจำจุดต่างมาอยู่ในรูปของ sine y (y คือค่าขององศาประจำจุดต่างๆ) แล้วนำมาคูณด้วย 1,000 เนื่องจาก TMS320C31 เป็น Fired Point

จุดที่ 1 เท่ากับ 0

จุดที่ 2 เท่ากับ 707

จุดที่ 3 เท่ากับ 1000

จุดที่ 4 เท่ากับ 707

จุดที่ 5 เท่ากับ 0

จุดที่ 6 เท่ากับ -707

จุดที่ 7 เท่ากับ -1000

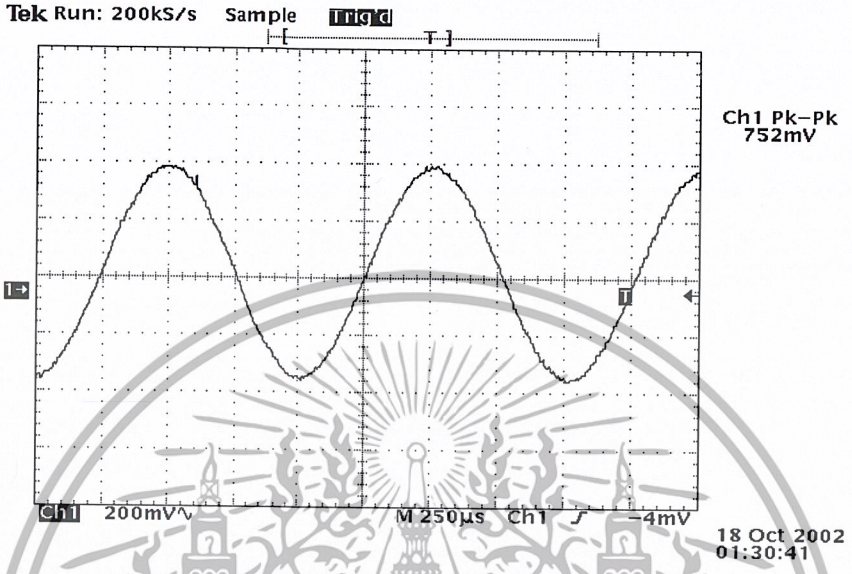
จุดที่ 8 เท่ากับ -707

7) นำค่าไปใส่ในโปรแกรม SINE4P.ASM ในบรรทัดที่ 9 และเปลี่ยนค่า LENGTH ในบรรทัดที่ 10 เป็น 8 เพื่อทำการวน 8 รอบ

8) ทำการคอมไพล์และรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) วาดรูปสัญญาณ



Volt/Div= 200mv Time/Div= 250us f= 1 kHz

สรุปผลการทดลอง

ความถี่(f)จะขึ้นอยู่กับจำนวน N และความถี่สุ่ม (Fs) คือ N จะแปลผกผันกับความถี่ (f) และ และความถี่สุ่ม (Fs) จะแปลผันกับความถี่ (f)

คำถามท้ายการทดลอง

ถ้าเปลี่ยน N = 10 จุด ความถี่สุ่มเป็น 20 k Hz จะได้ความถี่เอาต์พุตเท่าใด ?

ตอบ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5

Fast Fourier Transform (FFT)

จุดประสงค์เชิงพฤติกรรม

1. เพื่อให้นักศึกษาสามารถเขียนโปรแกรมคำนวณหาค่า Fast Fourier Transform ได้
2. เพื่อให้นักศึกษาสามารถอธิบายวิธีการพร็อตสเป็คตรัม จากค่าของ Fast Fourier Transform ได้
3. เพื่อให้นักศึกษาสามารถใช้งานโปรแกรม DSK Debugger ในการรันโปรแกรม Fast Fourier Transform ได้
4. เพื่อให้นักศึกษาสามารถเปรียบเทียบวิธีการคำนวณค่า FFT กับ DFT ได้

ทฤษฎี

การแปลงฟูเรียร์ (Fourier Transforms) เป็นเครื่องมือที่มีความสำคัญที่ใช้กันบ่อย ๆ ในระบบปฏิบัติงาน ทางสัญญาณดิจิตอล โดยจุดมุ่งหมายของการ Transfrom ก็คือเป็นการแปลงให้ทราบว่าจาก Time domain เป็น Frequency domain หรือการแปลงกลับจาก Frequency domain เป็น Time domain ผลจากการคำนวณจะเป็นเครื่องมือของ Fourier Transfrom ทำให้ทราบค่า ของ Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) เป็นวิธีการที่ใช้ในการคำนวณ Discrete Fourier Fourier Transforms (DFT) ให้เร็วขึ้นหรือให้มีลักษณะที่เป็นเรียลไทม์ขึ้น

ที่มาของ FFT โดยอาศัยหลักการของ DFT

วิธีการเสนอโดย คูลีย์ และทูคีย์ ซึ่งได้กล่าวมาแล้วนั้น ความเป็นจริง การแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา x(m) ที่มีขนาด N จุด ออกเป็นสองลำดับสัญญาณที่มีความยาว N/2 จุดเท่ากัน ทั้งสองลำดับนี้หน้าชื่อเรียกว่า ลำดับสัญญาณคู่ และลำดับสัญญาณคี่โดยที่ลำดับสัญญาณคู่ เกดจากการการเอาลำดับในตำแหน่งเป็นเลขคู่มาเรียงที่เหลือก็เป็นลำดับสัญญาณคี่ ดังนั้นถ้าเรานิยามให้ $x_2(m)$ เป็นลำดับคู่และลำดับที่ $x_0(m)$ ตามลำดับเพราะฉะนั้น

$$x_2(m) = x(2m) \quad ; m = 0,1,\ldots,(N/2)-1$$

$$x_0(m) = x(2m+1) \quad ; m = 0,1,\ldots,(N/2)-1 \qquad (5.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$X(k) = \sum_{M=0}^{N-1} x_2(m)(W_N)^{km} + \sum_{M=0}^{N-1} x_0(m)(W_N)^{km}$$

$$= \sum_{M=0}^{(N-2)-1} x(2m)(W_N)^{2km} + \sum_{M=0}^{(N-2)-1} x(2m+1)(W_N)^{(2m+1)k} \qquad (5.2)$$

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 ของลำดับความยาว N/2 จุด เพราะฉะนั้น จัดพจน์ใหม่ได้ คือ

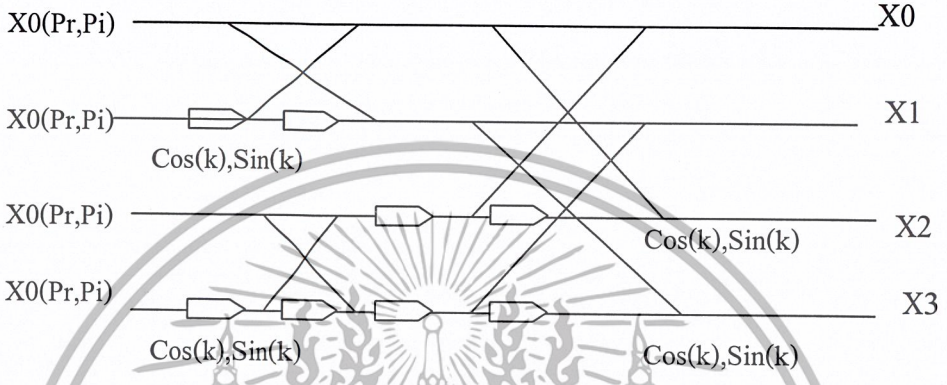
$$X(k) = \sum_{M=0}^{(N-1)-1} x_2(m)(W_{N/2})^{km} + \sum_{M=0}^{(N-2)-1} x0(m)(W_{N/2})^{km}$$

$$X(k) = X_1(k) + (W_n)^k X_2(k) \qquad (5.3)$$

โดยที่ $x_1(k)$ และ $X_2(k)$ แทนผลการแปลง DFT ขนาด N/2 จุดของลำดับ $X_2(m)$ และ X_0 ตามลำดับ แสดงให้เห็นว่าการคำนวณ DFT ขนาด N จุดสามารถแบ่งคำนวณย่อยออกเป็น การคำนวณ DFT ขนาด N/2 จุดสองอันดับได้และข้อสำคัญคือ การคูณจำนวนเชิงซ้อนจะลดลงเหลือ $2(N/2)^2 = N^2/2$ ครั้ง ซึ่งจะเห็นว่าลดการคำนวณไปถึง 50 เปอร์เซ็นต์ โดยอาศัยหลักเดียวกัน ถ้าเราแบ่งทอนลำดับ $X_2(m)$ และ $X_0(m)$ ออกเป็นลำดับคู่และลำดับคี่ลงไปอีกตามลำดับในที่สุดเหลือเป็นลำดับขนาด 2 จุด หรือกล่าวได้ว่า การคำนวณการแปลง DFT ขนาด N จุดทำการซอยเพื่อลำดับ xn0 ออกเป็นทีละครึ่งจนเหลือการคำนวณ DFT ขนาด 2 จุด โดยที่ Fast Fourier Transforms (FFT) สามารถแบ่งออกได้ 2 ชนิดใหญ่ๆ คือ ชนิดทอนเวลาและชนิดลดทอนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการทดลองใบงานนี้จะใช้วิธีลดทอนเวลาดังรูปที่ 1 โดยใช้สมการที่ (5.4) และสมการที่ (5.5) ในการคำนวณ

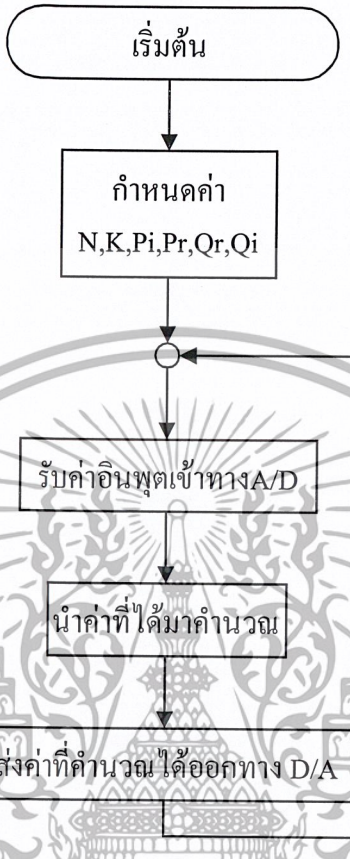


รูปที่ ฉ.14

$$\Pr n = [\Pr + QrCos(k) + QiSin(k)] + j[Pi + QiCos(k) - Sin(k)] \quad (5.4)$$

$$Qm = [\Pr - QrCos(k) - QiSin(k)] + j[Pi - QiCos(k) + Sin(k)] \quad (5.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ฉ.15 ผังการทำงานของโปรแกรม Fast Fourier Transform

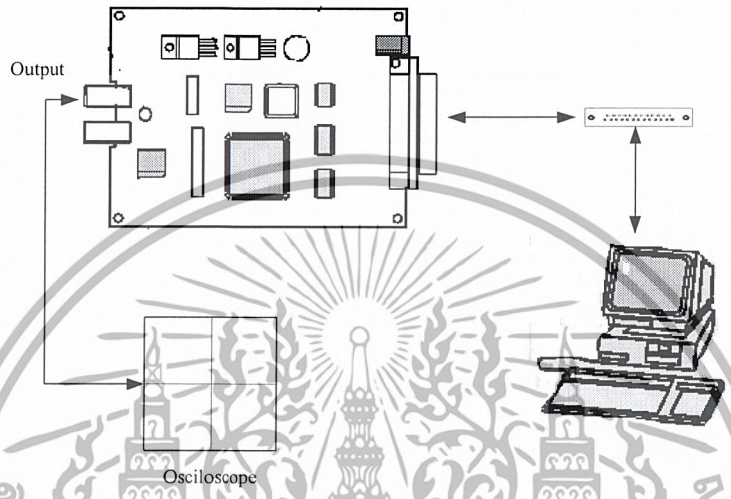
อุปกรณ์

1) ชุดฝึกปฏิบัติการการประมวลผลสัญญาณเชิงเลขด้วย TMS320C31	1 ชุด
2) เครื่องคอมพิวเตอร์	1 เครื่อง
3)โปรแกรม Dsktool	1 แผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูป



รูปที่ ฉ.16 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อนโปรแกรมดังนี้

```
/*FFT.C - REAL-VALUED FFT WITH 128 POINTS. CALLS FFT_RL.ASM/

#include "math.h"                    /*standard library func  */

#include "aiccomc.c"                 /*AIC comm routines      */

#define N 128                        /*size of FFT            */

#define M 7                          /*number of stages       */

volatile int index = 0;              /*input_output index     */

float *IO_buffer, *data, *temp;      /*--> array buffers      */

int AICSEC[4] = {0x162C,0x1,0x3872,0x67};   /*AIC config data       */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
extern void fft_rl(int, int, float *);          /*fft function protype   */

void c_int05()                                  /*interrupt handler func */

{

 PBASE[0x48] = ((int)(IO_buffer[index])) << 2;        /*output data*/

IO_buffer[index] = (float)(PBASE[0x4C] << 16 >> 18);   /*input data */

 if (++index >= N) index = 0;          /*increment index, reset = N*/

}

main()

{

 int loop;                              /* declare variable       */

 float real, img;                       /* declare variables      */

AICSET_I();                             /*config AIC for interrupt  */

IO_buffer = (float *) calloc(N, sizeof(float)); /*input_out buffer  */

data = (float *) calloc(N, sizeof(float));   /* fft data buffer       */

while (1)                               ;* create endless loop      */

{

fft_rl(N, M, (float *)data);            /*call FFT function        */

 data[0] = sqrt(data[0]*data[0])/N;           /*magnitude of X(0) */

 for (loop = 1; loop < N/2; loop++)     /*calculate X(1)..X(N/2-1)  */
```

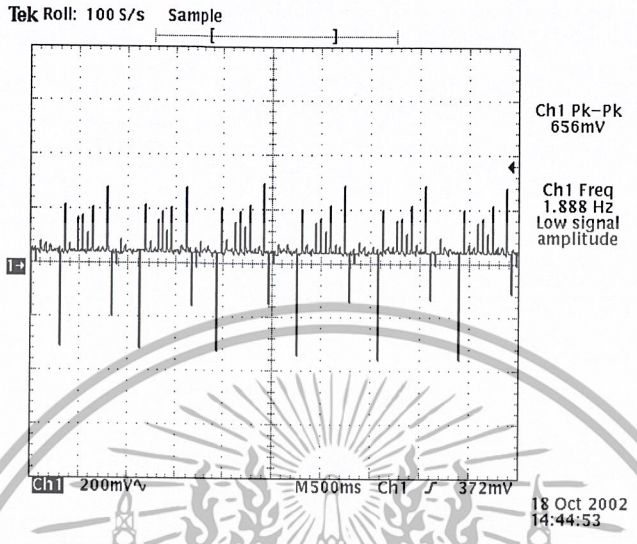
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  real = data[loop];                    /*real part  */
 img = data[N-loop];                    /*imaginary part   */
 data[loop] = sqrt(real*real+img*img)/N;   /*find magnitude   */
 }
  data[N/2] = sqrt(data[N/2]*data[N/2])/N;  /*magnitude of X(N/2)    */
  for (loop = N/2+1; loop < N; loop++)      /*X(N/2+1).. X(N-1)*/
   data[loop] = data[N-loop];                    /*use symetry  */
  while (index);              /*wait till IO_buffer empty */
emp = data;                   *temp => data buffer       */
 data = IO_buffer;            /*IO_buffer-->data buffer   */
IO_buffer = temp;             /*data buffer->new IO_buffer*/
 IO_buffer[0] = -2048;        /*sync pulse,negative spike */
}
}
```

รูปที่ ฉ.17 โปรแกรม Fast Fourier Transform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ป้อนคลื่น Sine Wave ให้กับบอร์ด DSK TMS320C31 บันทึกผลการทดลอง



Volt/Div= 200mV Time/Div= 500ms f= 2kHz

สรุปผลการทดลอง

จากการรันโปรแกรม Fast Fourier Transform (FFT) จะทำให้ทราบถึงวิธีการเขียนโปรแกรมและการแสดงผลของข้อมูลที่ตำแหน่ง Data Memory ที่กำหนดไว้ในโปรแกรมและสามารถทำการเปลี่ยนค่าอินพุตที่เข้ามายัง FFT โดยไม่ต้องใช้วิธีการคำนวณทางทฤษฎี

คำถามท้ายการทดลอง

จงอธิบายหลักการของวิธีการพร็อตสเป็กตรัม ของ Fast Fourier Transforms ?

ตอบ ผลที่ได้เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่6

เอฟไออาร์ฟิลเตอร์ (FIR Filter)

จุดประสงค์เชิงพฤติกรรม

1.เพื่อให้นักศึกษาสามารถอธิบายหลักการทำงานของฟิลเตอร์แบบเอฟไออาร์ได้

2.เพื่อให้นักศึกษาสามารถบอกวิธีกาหาค่าสัมประสิทธิ์ของ เอฟไออาร์ได้

3.เพื่อให้นักศึกษาสามารถเขียนโครงสร้างและเขียนไดอะแกรม FIR Filter ได้

ทฤษฎี

วงจรกรองแบบเอฟไออาร์ (FIR Filter)

สมการที่ (1) เป็นสมการที่ใช้ออกแบบ

$$y(n) = \sum_{k=0}^{M} b_x x(n-k) \qquad (6.1)$$

จะเห็นได้ว่าสมการนี้เป็นสมการของดิจิตอลฟิลเตอร์ที่มี a =0 โดย b จะมีจำนวนเท่ากับจำนวนอิมพัลส์เรสปอนส์ ของฟิลเตอร์ ตามที่ได้ออกแบบไว้

จากสมการของวงจรกรองเอฟไออาร์ จะเห็นได้ว่าเป็นรูปแบบของสมการคอนโวลูชัน (Convolution Equation) ดังสมการที่ (2)

$$y(n) = \sum_{k=0}^{M} h(n-k)\ xz^{-1} \qquad (6.2)$$

การหาสัมประสิทธิ์ของ FIR Fillter

$$y(n) = \begin{cases} \frac{1}{\pi n}\ (\sin \omega 2n - \sin \omega 1n) & ; n \neq 0 \\ \frac{1}{\pi}\ (\omega 1 - \omega 2); n = 0 \end{cases}$$

เมื่อเราได้ค่า h(n) แต่ละตัว จากนั้นต้องทำการคูณด้วยค่า Windows hamming จากสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\omega(\mathrm{n}) = \begin{cases} 0.5 + 0.5\cos\left(\dfrac{\pi n}{1 + (M/2)}\right) & ;-M/2 \le n \le M/2 \\ 0; \mid n \mid > M/2 \end{cases}$$

$$h(n) = r(n) * w(n)$$



รูปที่ ฉ.18 ผังการทำงานโปรแกรมฟิลเตอร์แบบ FIR

อุปกรณ์

1) ชุดฝึกปฏิบัติการการประมวลผลสัญญาณเชิงเลข 1 ชุด

2) สโคป 1 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

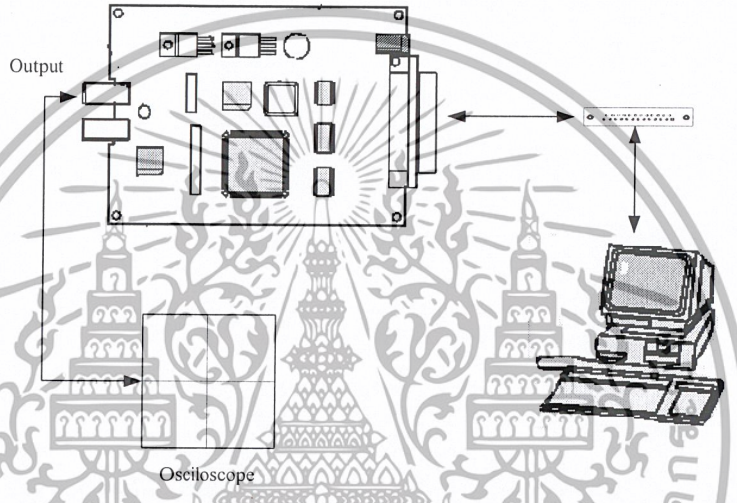
3) ซิกแนลเจนเนอร์เรเตอร์ 1 เครื่อง

4) เครื่องคอมพิวเตอร์ 1 เครื่อง

5) ผ่นดิสก์โปรแกรม DSK 1 แผ่น

ลำดับขั้นการทดลอง

1) ต่อชุดทดลองกับอุปกรณ์ดังรูปที่ ฉ.19



รูปที่ ฉ.19 การเชื่อมต่ออุปกรณ์กับชุดทดลอง

2) ป้อนโปรแกรมดังนี้ โปรแกรม LP 11 SIM.ASM

```
;LP11SIM.ASM - FIR LOWPASS FILTER WITH 11 COEFF FOR SIMULATION
        .start  ".text",0x809900    ;       where text begins
        .start  ".data",0x809C00 ;where data begins
        .data                               ;data section
IN_ADDR     .word   INB             ;starting address for input
OUT_ADDR    .word   OUTB            ;starting address for output
XB_ADDR     word    XN+LENGTH-1     ;bottom address of circ buffer
HN_ADDR     .word   COEFF           ;starting addr of coefficients
COEFF       .float  0           ;H10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .float  0.0468, 0.1009, 0.1514, 0.1872, 0.2
        .float  0.1872, 0.1514, 0.1009, 0.0468
H0      .float  0                       ;H0
LENGTH  .set    H0-COEFF+1      ;# of coefficients
INB     .float  10000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
OUTB    .float  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
        .brstart "XN_BUFF",16           ;align samples buffer
XN      .sect   "XN_BUFF"               ;section for input samples
        .loop   LENGTH                  ;buffer size for samples
        .float  0.0                     ;initialize samples to zero
        .endloop                        ;end of loop
        .entry  BEGIN                   ;start of code
        .text
BEGIN       LDP     @0x809800       ;init to data page 128
        LDI     LENGTH,BK               ;size of circular buffer
        LDI     @XB_ADDR,AR1            ;last sample address ->AR1
        LDI     @IN_ADDR,AR2            ;input address      -->AR2
        LDI     @OUT_ADDR,AR3           ;output address     -->AR3
FILT    LDI     LENGTH-1,AR4            ;length in AR4
LOOP    LDF     *AR2++,R3               ;input new sample
STF     R3,*AR1++%                      ;store newest sample
        LDI     @HN_ADDR,AR0            ;AR0 points to H(N-1)
        LDF     0,R0                    ;init R0
        LDF     0,R2                    ;init R2
        RPTS    LENGTH-1                ;repeat LENGTH-1 times
        MPYF3   *AR0++,*AR1++%,R0 ;R0 = HN*XN
        ADDF3   R0,R2,R2                ;accumulation in R2
        DBNZD   AR4,LOOP                ;delayed branch until AR4<0
        ADDF    R0,R2                   ;last mult result accumulated
        FIX     R2,R7                   ;convert float R2 to integer R7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

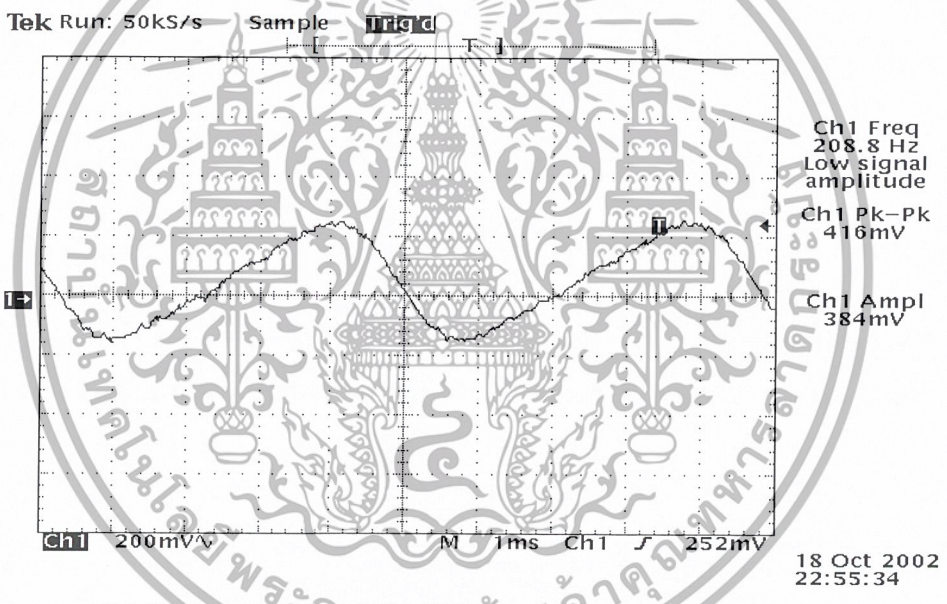
```
        STI    R7,*AR3++              ;store into output buffer
WAIT    BR     WAIT                   ;wait
        .end                  ;end
```

รูปที่ ฉ.20 โปรแกรม LP11SIM.ASM

3) ปรับความถี่อินพุตที่ 200 Hz 3Vp-p

4) คอมไพล์และรันโปรแกรม

5) บันทึกผลการทดลอง



Volt/DIV = 200mV TIME/DIV = 1 ms f = 200Hz

สรุปผลการทดลอง

จากการทดลองใช้ชิพ TMS320C31 มาประยุกต์ใช้เป็นวงจรกรองความถี่ต่ำผ่านชนิด FIR ซึ่งเป็นการสะดวกเพราะสามารถสร้างจากซอฟฟ์แวร์แทนการต่อวงจรที่ยุ่งยากและซับซ้อน

คำถามท้ายการทดลอง

จงบอกข้อดีข้อเสียของวงจรกรองเชิงเลขแบบเอฟไออาร์ ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบ ข้อดี สามารถออกแบบให้มีการตอบสนองความถี่ของเฟส มีความเป็นเชิงเส้นได้โดยแท้จริงและวงจรกรองประเภท FIR จะมีเสถียรภาพเสมอ เนื่องจากโครงสร้างปราศจากส่วนป้อนกลับ

ข้อเสีย จะต้องใช้จำนวนลำดับ (order) มากทำให้เปลืองหน่วยความจำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ช
รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Document Title

128Kx8 bit Low Power CMOS Static RAM

Revision History

Revision No.	History	Draft Data	Remark
0.0	Design target	October 12, 1998	Preliminary
1.0	Finalize - Improve twp form 55ns to 50ns for 70ns product. - Remove 55ns speed bin from industrial product.	August 30, 1999	Final
1.01	Errata correction	December 1, 1999	
2.0	Revise	February 14, 2000	Final
3.0	Revise - Add 55ns parts to industrial products.	March 3, 2000	Final



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K6T1008C2E Family CMOS SRAM

128Kx8 bit Low Power CMOS Static RAM

FEATURES

- Process Technology: TFT
- Organization: 128Kx8
- Power Supply Voltage: 4.5~5.5V
- Low Data Retention Voltage: 2V(Min)
- Three state output and TTL Compatible
- Package Type: 32-DIP-600, 32-SOP-525, 32-TSOP1-0820F/R

GENERAL DESCRIPTION

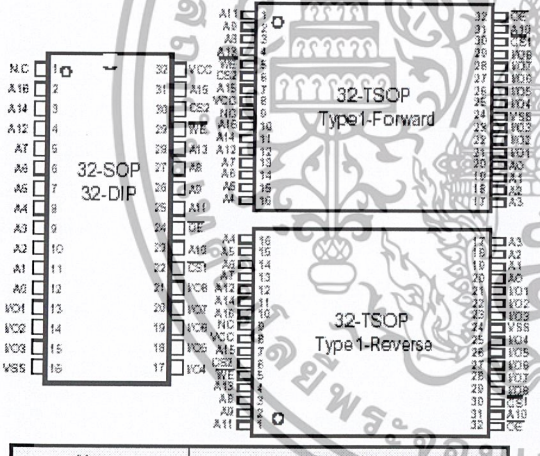
The K6T1008C2E families are fabricated by SAMSUNG's advanced CMOS process technology. The families support various operating temperature ranges and have various package types for user flexibility of system design. The families also support low data retention voltage for battery back-up operation with low data retention current.

PRODUCT FAMILY

Product Family	Operating Temperature	Vcc Range	Speed	Power Dissipation		PKG Type
				Standby (I_{SB1}, Max)	Operating (I_{CC2}, Max)	
K6T1008C2E-L	Commercial(0~70°C)	4.5~5.5V	55[1]/70ns	50μA	50mA	32-DIP-600, 32-SOP-525 32-TSOP1-0820F/R
K6T1008C2E-B				10μA		
K6T1008C2E-P	Industrial(-40~85°C)			50μA		32-SOP -525 32-TSOP1-0820F/R
K6T1008C2E-F				15μA		

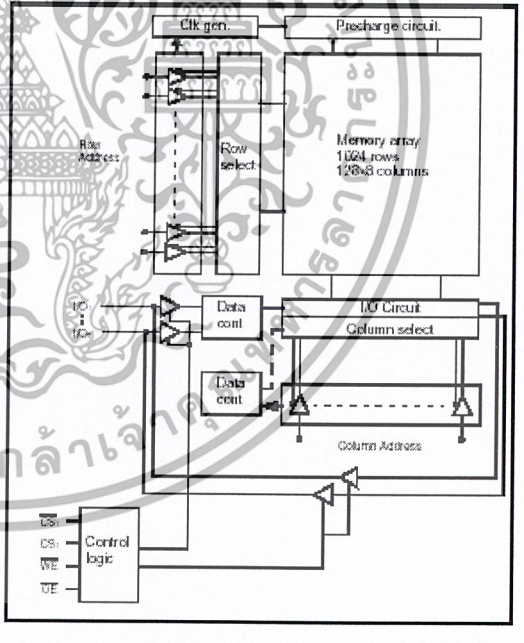
1. The parameters are tested with 50pF test load

PIN DESCRIPTION



Name	Function
$\overline{CS_1}$, CS_2	Chip Select Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
I/O_1~I/O_8	Data Inputs/Outputs
A_0~A_{16}	Address Inputs
Vcc	Power
Vss	Ground
N.C.	No Connection

FUNCTIONAL BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT LIST

Commercial Temperature Products(0~70°C)		Industrial Temperature Products(-40~85°C)	
Part Name	Function	Part Name	Function
K6T1008C2E-DL55	32-DIP, 55ns, Low Power	K6T1008C2E-GP55	32-SOP, 55ns, Low Power
K6T1008C2E-DL70	32-DIP, 70ns, Low Power	K6T1008C2E-GP70	32-SOP, 70ns, Low Power
K6T1008C2E-DB55	32-DIP, 55ns, Low Low Power	K6T1008C2E-GF55	32-SOP, 55ns, Low Low Power
K6T1008C2E-DB70	32-DIP, 70ns, Low Low Power	K6T1008C2E-GF70	32-SOP, 70ns, Low Low Power
K6T1008C2E-GL55	32-SOP, 55ns, Low Power	K6T1008C2E-TF55	32-TSOP F, 55ns, Low Low Power
K6T1008C2E-GL70	32-SOP, 70ns, Low Power	K6T1008C2E-TF70	32-TSOP F, 70ns, Low Low Power
K6T1008C2E-GB55	32-SOP, 55ns, Low Low Power	K6T1008C2E-RF55	32-TSOP R, 55ns, Low Low Power
K6T1008C2E-GB70	32-SOP, 70ns, Low Low Power	K6T1008C2E-RF70	32-TSOP R, 70ns, Low Low Power
K6T1008C2E-TB55	32-TSOP F, 55ns, Low Low Power		
K6T1008C2E-TB70	32-TSOP F, 70ns, Low Low Power		
K6T1008C2E-RB55	32-TSOP R, 55ns, Low Low Power		
K6T1008C2E-RB70	32-TSOP R, 70ns, Low Low Power		

FUNCTIONAL DESCRIPTION

$\overline{CS_1}$	CS_2	$\overline{OE}$	$\overline{WE}$	I/O	Mode	Power
H	X[1]	X[1]	X[1]	High-Z	Deselected	Standby
X[1]	L	X[1]	X[1]	High-Z	Deselected	Standby
L	H	H	H	High-Z	Output Disabled	Active
L	H	L	H	Dout	Read	Active
L	H	X[1]	L	Din	Write	Active

1. X means don't care (Must be in high or low states)

ABSOLUTE MAXIMUM RATINGS[1]

Item	Symbol	Ratings	Unit	Remark
Voltage on any pin relative to Vss	V_{IN},V_{OUT}	-0.5 to 7.0	V	-
Voltage on Vcc supply relative to Vss	V_{CC}	-0.5 to 7.0	V	-
Power Dissipation	P_D	1.0	W	-
Storage temperature	T_{STG}	-65 to 150	°C	-
Operating Temperature	T_A	0 to 70	°C	K6T1008C2E-L/-B
		-40 to 85	°C	K6T1008C2E-P/-F

1. Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation should be restricted to recommended operating condition. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRODUCT LIST

Commercial Temperature Products(0~70°C)		Industrial Temperature Products(-40~85°C)	
Part Name	Function	Part Name	Function
K6T1008C2E-DL55	32-DIP, 55ns, Low Power	K6T1008C2E-GP55	32-SOP, 55ns, Low Power
K6T1008C2E-DL70	32-DIP, 70ns, Low Power	K6T1008C2E-GP70	32-SOP, 70ns, Low Power
K6T1008C2E-DB55	32-DIP, 55ns, Low Low Power	K6T1008C2E-GF55	32-SOP, 55ns, Low Low Power
K6T1008C2E-DB70	32-DIP, 70ns, Low Low Power	K6T1008C2E-GF70	32-SOP, 70ns, Low Low Power
K6T1008C2E-GL55	32-SOP, 55ns, Low Power	K6T1008C2E-TF55	32-TSOP F, 55ns, Low Low Power
K6T1008C2E-GL70	32-SOP, 70ns, Low Power	K6T1008C2E-TF70	32-TSOP F, 70ns, Low Low Power
K6T1008C2E-GB55	32-SOP, 55ns, Low Low Power	K6T1008C2E-RF55	32-TSOP R, 55ns, Low Low Power
K6T1008C2E-GB70	32-SOP, 70ns, Low Low Power	K6T1008C2E-RF70	32-TSOP R, 70ns, Low Low Power
K6T1008C2E-TB55	32-TSOP F, 55ns, Low Low Power		
K6T1008C2E-TB70	32-TSOP F, 70ns, Low Low Power		
K6T1008C2E-RB55	32-TSOP R, 55ns, Low Low Power		
K6T1008C2E-RB70	32-TSOP R, 70ns, Low Low Power		

FUNCTIONAL DESCRIPTION

$\overline{CS_1}$	CS_2	$\overline{OE}$	$\overline{WE}$	I/O	Mode	Power
H	X[1]	X[1]	X[1]	High-Z	Deselected	Standby
X[1]	L	X[1]	X[1]	High-Z	Deselected	Standby
L	H	H	H	High-Z	Output Disabled	Active
L	H	L	H	Dout	Read	Active
L	H	X[1]	L	Din	Write	Active

1. X means don't care (Must be in high or low states)

ABSOLUTE MAXIMUM RATINGS[1]

Item	Symbol	Ratings	Unit	Remark
Voltage on any pin relative to Vss	V_{IN},V_{OUT}	-0.5 to 7.0	V	-
Voltage on Vcc supply relative to Vss	V_{CC}	-0.5 to 7.0	V	-
Power Dissipation	P_D	1.0	W	-
Storage temperature	T_{STG}	-65 to 150	°C	-
Operating Temperature	T_A	0 to 70	°C	K6T1008C2E-L/-B
		-40 to 85	°C	K6T1008C2E-P/-F

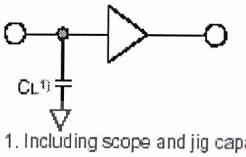
1. Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation should be restricted to recommended operating condition. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC OPERATING CONDITIONS

TEST CONDITIONS(Test Load and Input/Output Reference)

Input pulse level: 0.8 to 2.4V
Input rising and falling time: 5ns
Input and output reference voltage: 1.5V
Output load(see right): C_L=100pF+1TTL
C_L=50pF+1TTL



1. Including scope and jig capacitance

AC CHARACTERISTICS (V_{CC}=4.5~5.5V, Commercial Product: T_A=0 to 70°C, Industrial Product: T_A=-40 to 85°C)

Parameter List		Symbol	Speed Bins				Units
			55ns		70ns		
			Min	Max	Min	Max	
Read	Read Cycle Time	t_{RC}	55	-	70	-	ns
	Address Access Time	t_{AA}	-	55	-	70	ns
	Chip Select to Output	t_{CO}	-	55	-	70	ns
	Output Enable to Valid Output	t_{OE}	-	25	-	35	ns
	Chip Select to Low-Z Output	t_{LZ}	10	-	10	-	ns
	Output Enable to Low-Z Output	t_{OLZ}	5	-	5	-	ns
	Chip Disable to High-Z Output	t_{HZ}	0	20	0	25	ns
	Output Disable to High-Z Output	t_{OHZ}	0	20	0	25	ns
	Output Hold from Address Change	t_{OH}	10	-	10	-	ns
Write	Write Cycle Time	t_{WC}	55	-	70	-	ns
	Chip Select to End of Write	t_{CW}	45	-	60	-	ns
	Address Set-up Time	t_{AS}	0	-	0	-	ns
	Address Valid to End of Write	t_{AW}	45	-	60	-	ns
	Write Pulse Width	t_{WP}	40	-	50	-	ns
	Write Recovery Time	t_{WR}	0	-	0	-	ns
	Write to Output High-Z	t_{WHZ}	0	20	0	25	ns
	Data to Write Time Overlap	t_{DW}	20	-	25	-	ns
	Data Hold from Write Time	t_{DH}	0	-	0	-	ns
	End Write to Output Low-Z	t_{OW}	5	-	5	-	ns

DATA RETENTION CHARACTERISTICS

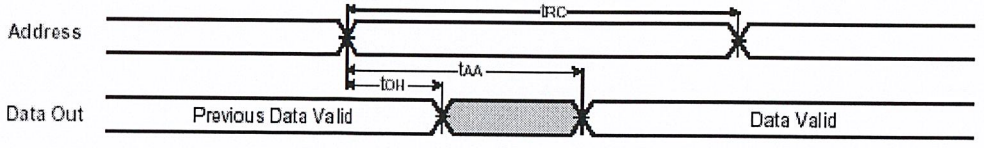
Item	Symbol	Test Condition		Min	Typ	Max	Unit
Vcc for data retention	V_{DR}	$\overline{CS_1} \geq V_{CC}-0.2V$ 1)		2.0	-	5.5	V
Data retention current	I_{DR}	V_{CC}=3.0V, $\overline{CS_1} \geq V_{CC}-0.2V$ 1)	K6T1008C2E-L	-	-	20	µA
			K6T1008C2E-B	-	-	10	
			K6T1008C2E-P	-	-	25	
			K6T1008C2F-F	-	-	10	
Data retention set-up time	t_{SDR}	See data retention waveform		0	-	-	ms
Recovery time	t_{RDR}			5	-	-	

1. $\overline{CS_1} \geq V_{CC}-0.2V$, $CS_2 \geq V_{CC}-0.2V$($\overline{CS_1}$ controlled) or $CS_2 \leq 0.2V$(CS_2 controlled)

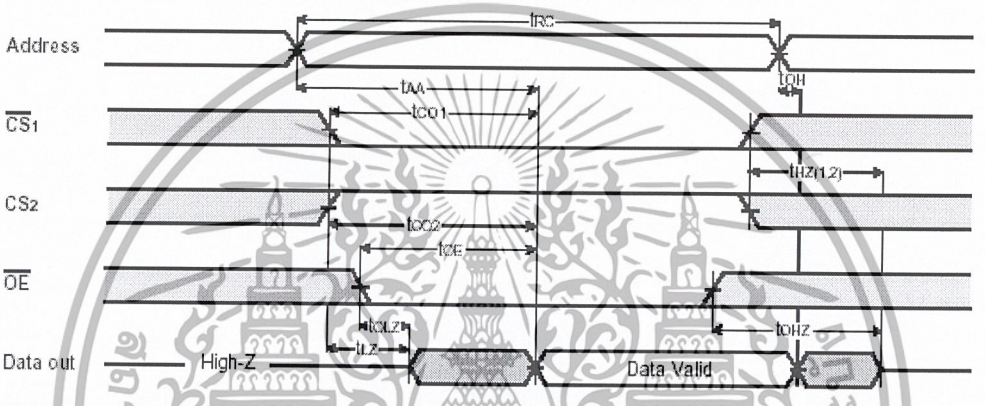
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMMING DIAGRAMS

TIMING WAVEFORM OF READ CYCLE(1) (Address Controlled, $\overline{CS1}$=$\overline{OE}$=V_{IL}, CS2=$\overline{WE}$=V_{IH})



TIMING WAVEFORM OF READ CYCLE(2) ($\overline{WE}$=V_{IH})

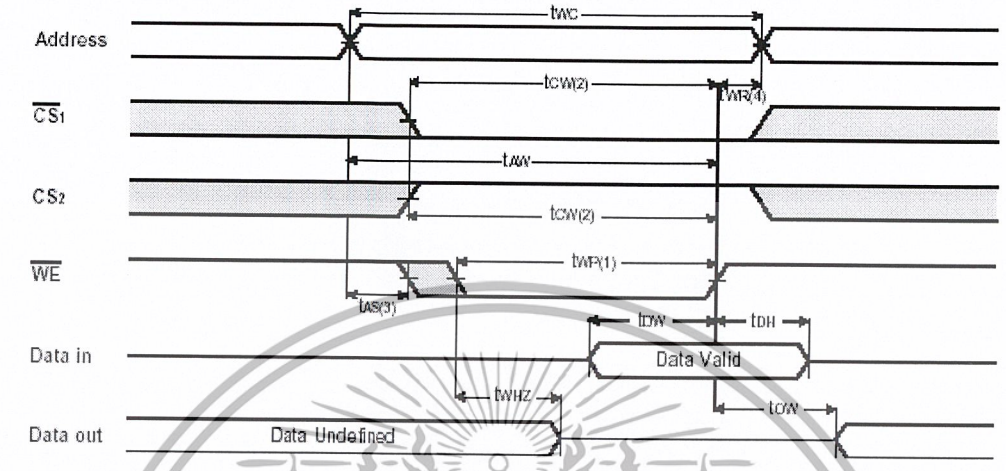


NOTES (READ CYCLE)

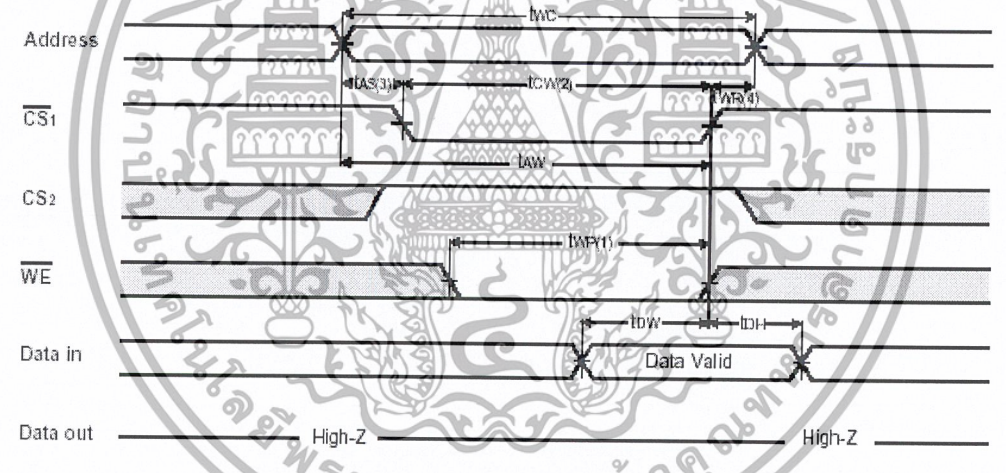
1. t_{HZ} and t_{OHZ} are defined as the time at which the outputs achieve the open circuit conditions and are not referenced to output voltage levels.
2. At any given temperature and voltage condition, t_{HZ}(Max.) is less than t_{LZ}(Min.) both for a given device and from device to device interconnection.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING WAVEFORM OF WRITE CYCLE(1) ($\overline{WE}$ Controlled)

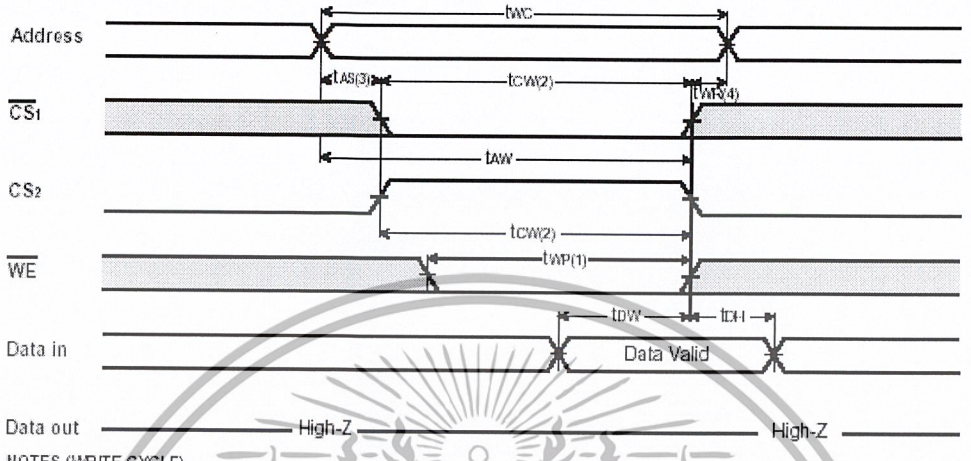


TIMING WAVEFORM OF WRITE CYCLE(2) ($\overline{CS_1}$ Controlled)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING WAVEFORM OF WRITE CYCLE(3) (CS_2 Controlled)

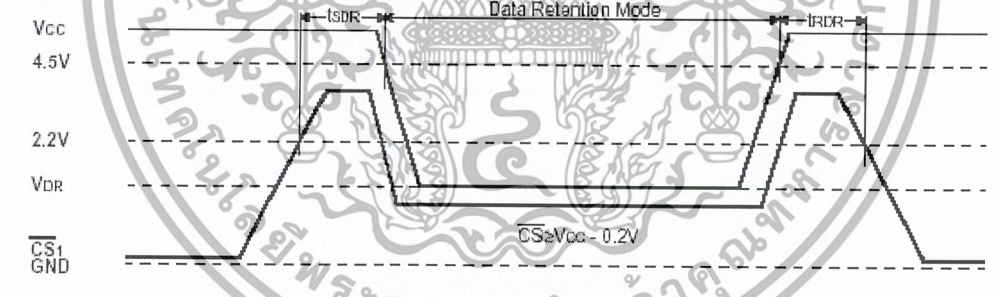


NOTES (WRITE CYCLE)

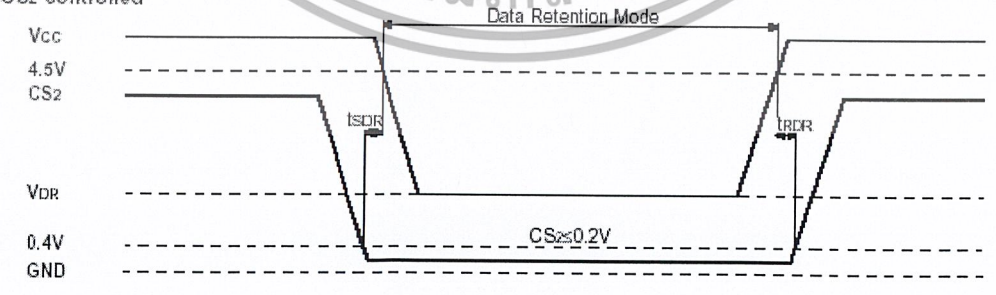
1. A write occurs during the overlap of a low $\overline{CS_1}$, a high CS_2 and a low $\overline{WE}$. A write begins at the latest transition among $\overline{CS_1}$ goes low, CS_2 going high and $\overline{WE}$ going low; A write end at the earliest transition among $\overline{CS_1}$ going high, CS_2 going low and $\overline{WE}$ going high, t_{WP} is measured from the begining of write to the end of write.
2. t_{CW} is measured from the $\overline{CS_1}$ going low or CS_2 going high to the end of write.
3. t_{AS} is measured from the address valid to the beginning of write.
4. t_{WR} is measured from the end of write to the address change. t_{WR1} applied in case a write ends as $\overline{CS_1}$ or $\overline{WE}$ going high t_{WR2} applied in case a write ends as CS_2 going to low.

DATA RETENTION WAVE FORM

$\overline{CS_1}$ controlled



CS_2 controlled

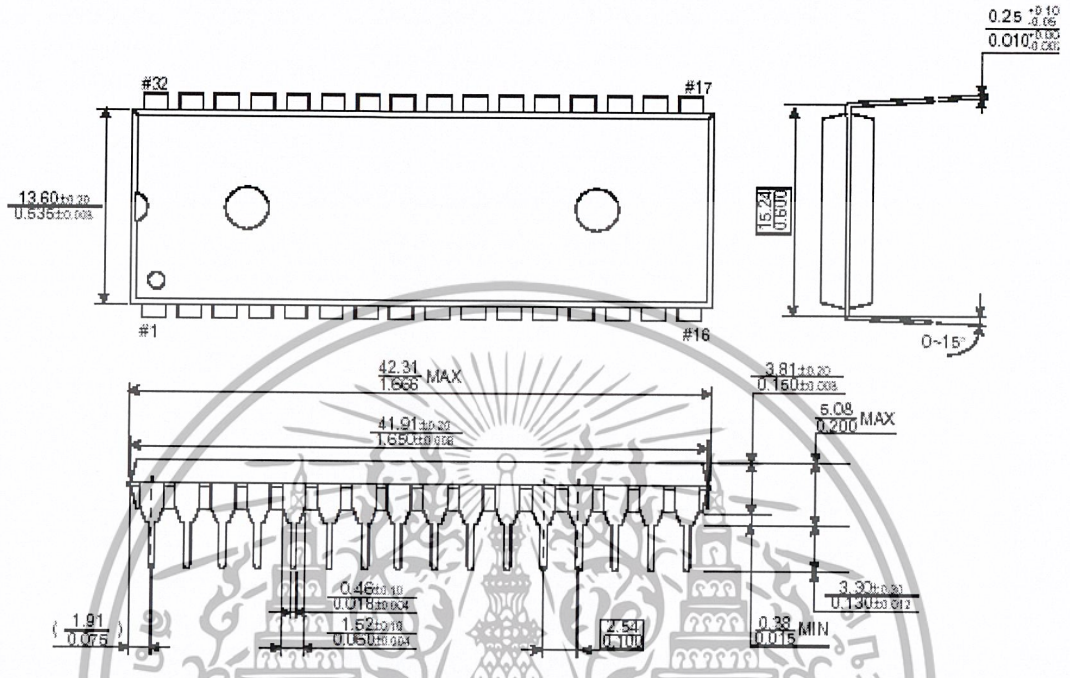


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

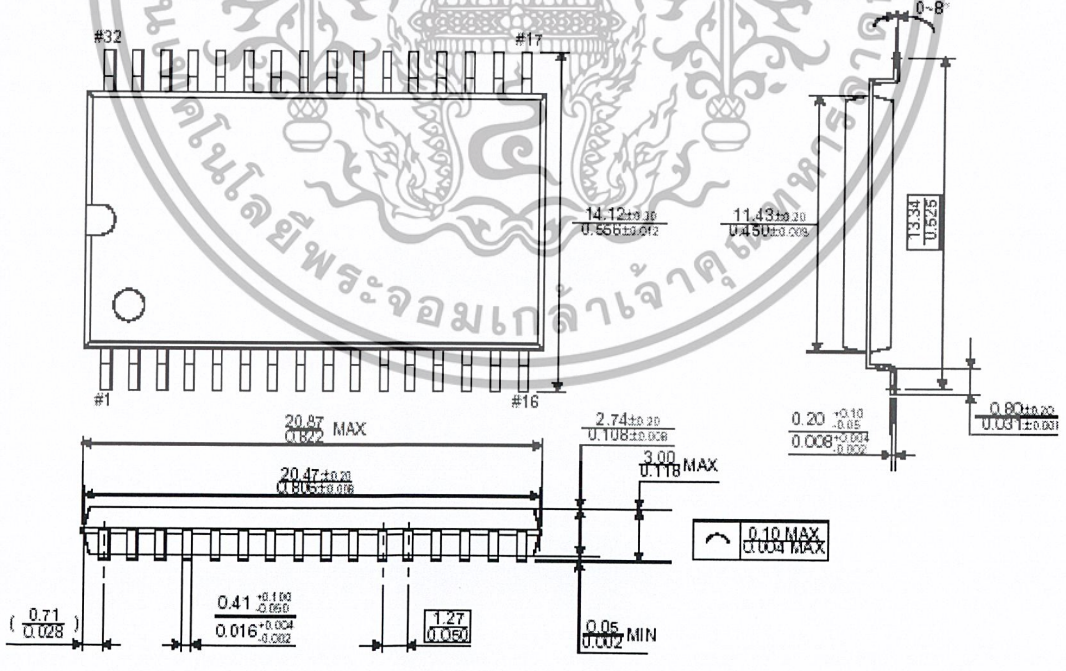
PACKAGE DIMENSIONS

32 DUAL INLINE PACKAGE (600mil)

Units: millimeters(inches)



32 PLASTIC SMALL OUTLINE PACKAGE (525mil)

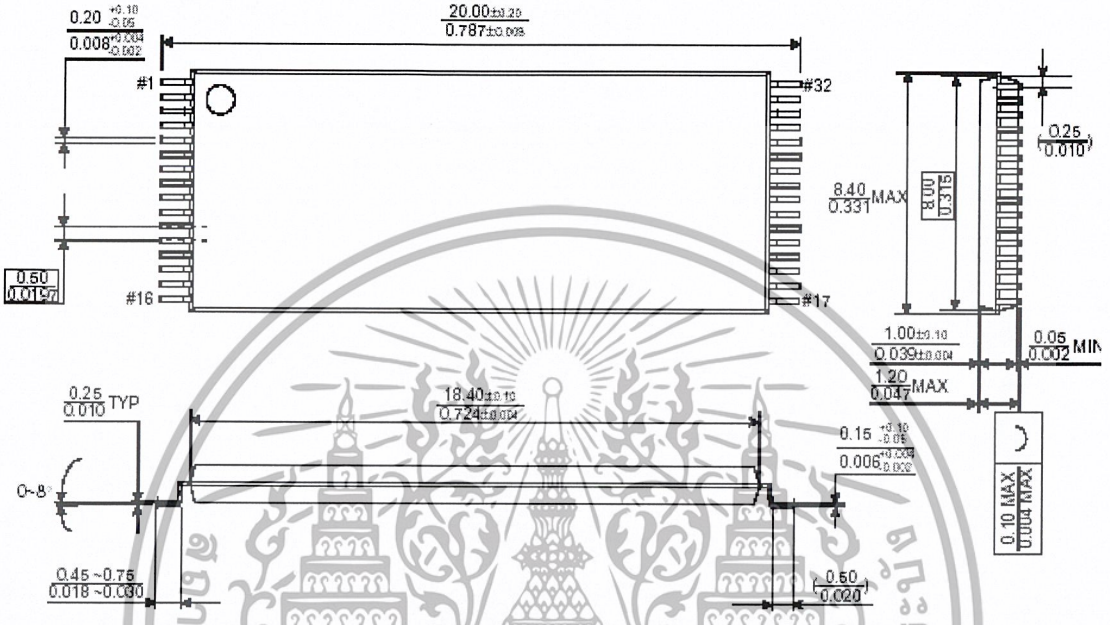


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

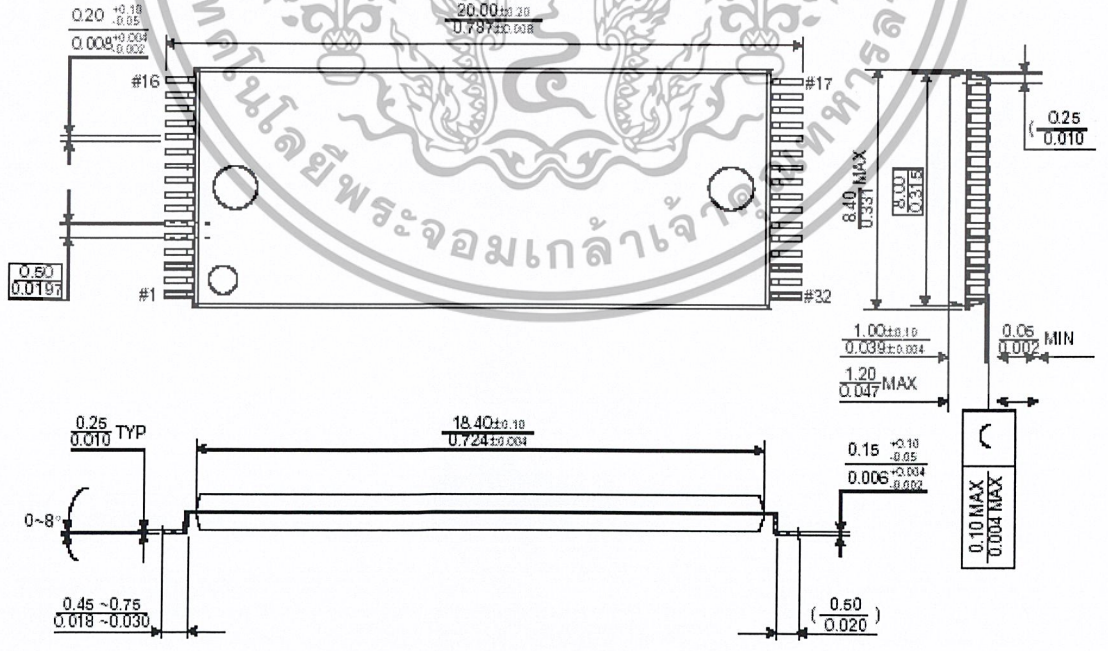
PACKAGE DIMENSIONS

Units: millimeters(inches)

32 PIN THIN SMALL OUTLINE PACKAGE TYPE I (0820F)



32 THIN SMALL OUTLINE PACKAGE TYPE I (0820R)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

อรรถสิทธิ์ ล่าสกุล.**Digital Signal Processing,**พิมพ์ครั้งที่1,กรุงเทพฯ:แผนกตำรา คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีเจ้าคุณทหารลาดกระบัง.2543

Rulph Chassaing.**Digital Signal Processing,**University of Massachusetts,Darmouth, A Wiley-Interscience Publication.1999

TEXAS Instruments.**TMS320C3X DSP STARTER KIT Use's Guide,**Texas Instruments Incorperate.1996



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นายจักรกฤษณ์ หมายมั่น
วัน เดือน ปี เกิด	6 เมษายน 2524
สถานที่เกิด	จังหวัดอ่างทอง
ภูมิลำเนา	64 หมู่ 3 ตำบล บ้านรี อำเภอเมือง จังหวัดอ่างทอง
ที่อยู่ปัจจุบัน	จังหวัดอ่างทอง 64 หมู่ 3 ตำบล บ้านรี อำเภอเมือง

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดรุ้ง วิบูลย์วิทยาคาร จังหวัดอ่างทอง
มัธยมศึกษาตอนต้น	โรงเรียนอ่างทองปัทมโรจน์วิทยาคม จังหวัดอ่างทอง
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคอ่างทอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคอ่างทอง
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผลงานที่ได้รับ -

คติพจน์ วันนี้วันไหนก็เต็มใจทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นายณัฐวัฒน์ ทองนิรันดร์
วัน เดือน ปี เกิด	23 มิถุนายน 2523
สถานที่เกิด	จังหวัดนนทบุรี
ภูมิลำเนา	58/1 หมู่ 2 ตำบล พิมลราช อำเภอ บางบัวทอง จังหวัดนนทบุรี
ที่อยู่ปัจจุบัน	จังหวัดนนทบุรี 58/1 หมู่ 2 ตำบล พิมลราช อำเภอบางบัวทอง
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดละหาร จังหวัดนนทบุรี
มัธยมศึกษาตอนต้น	โรงเรียนบางบัวทอง จังหวัดนนทบุรี
มัธยมศึกษาตอนปลาย	โรงเรียนบางบัวทอง (1ปี) จังหวัดนนทบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตนนทบุรี (4ปี)
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	จงทำด้วยตัวเอง อย่ารอโชคชะตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นายพิทยา ถวิลไพร
วัน เดือน ปี เกิด	23 กรกฎาคม 2523
สถานที่เกิด	จังหวัดชลบุรี
	38/1 หมู่ 7 ตำบล หนองตำลึง อำเภอ พานทอง
	จังหวัดชลบุรี
ที่อยู่ปัจจุบัน	จังหวัดชลบุรี
	23/5 หมู่ 5 ตำบล สำนักบก อำเภอเมือง
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านเขาแรต จังหวัดชลบุรี
มัธยมศึกษาตอนต้น	โรงเรียนพนมไพรวิทยาคาร จังหวัดชลบุรี
มัธยมศึกษาตอนปลาย	โรงเรียนพนมไพรวิทยาคาร จังหวัดชลบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคชลบุรี
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
คติพจน์	สู้ไปเรื่อยๆ เหนื่อยก็พัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้