ดิจิตอลซิกแนลโปรเซสซิ่ง

DIGITAL SIGNAL PROCESSING (DSP)

โดย

นายณัฐวุฒิ	ระวีวงค์อโนทัย	35104142
นางสาวธารทิพย์	ชุติกุลรังษี	35104187
นายธีระ	จิระรัตนโพธิ์ชัย	35104193

อาจารย์ที่ปรึกษา

อ.เทอดศักดิ์ ลิ่วหาทอง

วัน เดือน ปี 1 ส.ค. 2540
เลขทะเบียน 037098
เลขเรียกหนังสือ T 58141 ก86ป1 ค1

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037098

ดิจิตอลซิกแนลโปรเซสซิ่ง

ณัฐวุฒิ ระวีวงค์อโนทัย
ธารทิพย์ ชุติกุลรังษี
ธีระ จิระรัตนโพธิ์ชัย
อ.เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษา
ปีการศึกษา 2538

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาการใช้งานการ์ดอินเทอร์เฟส DSK (DSK Starter Kit) และการเขียนโปรแกรม เพื่อนำมาประยุกต์ใช้งานด้านระบบการประมวลผลเชิงตัวเลขหรือดิจิตอลซิกแนลโปรเซสซิ่ง โดยปริญญานิพนธ์ฉบับนี้ได้นำเอา TMS 320C50 มาเป็นหน่วยประมวลผลรวม ใช้ในการคำนวณ FFT (FAST FOURIER TRANSFORM) เพื่อนำผลที่ได้มาแสดงผลในรูปสเปกโตรแกรม (SPECTROGRAM) บนเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIGITAL SIGNAL PROCESSING

Natawut Raveevonganotai
Tharntip Chutikulrungsee
Teera Jirarattanapochai
Terdsak Leewhathong Advisor
1996

Abstract

This thesis is the study to use DSK interface card (DSP Starter Kit) and write program for applying in digital signal processing system. In this thesis , TMS320C50 is used as co-processing unit for calculating spectrogram (FFT - Fast Fourier Transform with Hanning Window) and display this result onto computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	TMS320C50	2
	2.1 จุดเด่นของ TMS320C50	2
	2.2 ตำแหน่งขาและหน้าที่การทำงานของ TMS320C50	6
	2.3 การจัดหน่วยความจำของ TMS320C50	12
	2.3.1 หน่วยความจำข้อมูล	13
	2.3.2 หน่วยความจำโปรแกรม	16
	2.4 โหมดการอ้างถึงหน่วยความจำ	17
	2.5 อุปกรณ์เสริม	18
	2.5.1 อินเทอร์รัพท์	19
	2.5.2 พอร์ตอนุกรม	20
	2.5.3 ไทม์เมอร์	21
บทที่ 3	วงจรอินเทอร์เฟสสัญญาณอนาลอก TLC32040	23
	3.1 ลักษณะสำคัญของ TLC32040	23
	3.2 ฟังก์ชันบล็อกไดอะแกรม	23
	3.3 ตำแหน่งขาและหน้าที่การทำงานของแต่ละขา	24
	3.4 การทำงานของ TLC32040	27
	3.5 การอินเทอร์เฟสระหว่าง TMS320C50 กับ TLC32040	32
บทที่ 4	โครงสร้างของ TMS 320C50 DSP Starter Kit	34
	4.1 ลักษณะโดยทั่วไปของบอร์ด	34
	4.2 การสร้างโปรแกรมเพื่อใช้กับ DSK	36
บทที่ 5	การเชื่อมต่อแบบขนาน	37
	5.1 โครงสร้าง หน้าที่ และ การเชื่อมต่อแบบขนาน	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

บทที่ 6	สเปกโตรแกรม	44
	6.1 การแปลงฟาสท์ฟูริเยร์	44
	6.2 สเปกโตรแกรม	47
	6.3 โปรแกรมภาษาแอสเซมบี้	50
	6.4 โปรแกรมภาษาซี	60
บทที่ 7	ผลการทดลองและสรุปผลการทดลอง	66

ภาคผนวก ก. รูปวงจรแสดงส่วนต่าง ๆ บนบอร์ด DSK และ Schematic Diagrams

ภาคผนวก ข. คู่มือ IC 74LS373 และ Schematic Diagrams

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 บล็อกไดอะแกรมของ TMS320C50 4
รูปที่ 2.2 บล็อกไดอะแกรมส่วนหน่วยประมวลผลกลาง 5
รูปที่ 2.3 การเข้าถึงหน่วยความจำของ TMS320C50 13
รูปที่ 2.4 แสดงการต่อพอร์ตอนุกรม 20
รูปที่ 2.5 บล็อกไดอะแกรมการทำงานของพอร์ตอนุกรม 21
รูปที่ 2.6 บล็อกไดอะแกรมของไทม์เมอร์ 22
รูปที่ 3.1 ฟังก์ชั่นบล็อกไดอะแกรม 23
รูปที่ 3.2 อินเทอร์นอลไทม์มิ่งคอนฟิกกูเรชั่น 28
รูปที่ 3.3 แสดงรูปแบบบิตใน AIC DR หรือ DX เวิร์ด 29
รูปที่ 3.4 แสดงรูปแบบการฟอร์แมตใน AIC DR หรือ DX เวิร์ด 29
รูปที่ 3.5 แสดงการเชื่อมต่อระหว่าง TLC32040 กับ TMS320C50 32
รูปที่ 3.6 แสดงเวลาในการรับส่งข้อมูลระหว่าง TLC32040 กับ TMS320C50 32
รูปที่ 4.1 บล็อกไดอะแกรมของ TMS320C5x DSK 34
รูปที่ 4.2 หน่วยความจำภายใน TMS320C50DSK 35
รูปที่ 4.3 การต่อระหว่าง DSK และพีซีโดยผ่านทาง RS232 35
รูปที่ 4.4 แสดงขั้นตอนการสร้างโปรแกรม 36
รูปที่ 5.1 แผนภาพการอินเทอร์เฟสแบบขนาน 37
รูปที่ 6.1 แสดงหน่วยผีเสื้อของการคำนวณตามขั้นตอนวิธีการลดทอนทางเวลา 46
รูปที่ 6.2 แสดงวิธีการของ FFT แบบลดทอนทางเวลาสำหรับข้อมูลขนาด 8 บิต 46
รูปที่ 6.3 การคำนวณหาสเปกโตรแกรมของสัญญาณ 48
รูปที่ 6.4 การแสดงผลสเปกโตรแกรม 49
รูปที่ 6.5 โฟล์ชาร์ตแสดงการหา FFT 59
รูปที่ 7.1 รูปสัญญาณ Sine Wave 1KHz ขนาด 130mV 66
รูปที่ 7.2 รูปสเปกตรัมสัญญาณ Sine Wave 1KHz ขนาด 130mV 66
รูปที่ 7.3 รูปสเปกโตรแกรมสัญญาณ Sine Wave 1KHz ขนาด 130mV 67
รูปที่ 7.4 รูปสัญญาณ Triangle Wave 1KHz ขนาด 130mV 68
รูปที่ 7.5 รูปสเปกตรัมสัญญาณ Triangle Wave 1KHz ขนาด 130mV 68

เอกสารนี้ [illegible] ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 7.6 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 1KHz ขนาด 130mV 69

รูปที่ 7.7 รูปสัญญาณ Square Wave 1KHz ขนาด 130mV 70

รูปที่ 7.8 รูปสเปกตรัมสัญญาณ Square Wave 1KHz ขนาด 130mV 70

รูปที่ 7.9 รูปสัญญาณสเปกโตรแกรม Square Wave 1KHz ขนาด 130mV 71

รูปที่ 7.10 รูปสัญญาณ Sine Wave 6KHz ขนาด 130mV 72

รูปที่ 7:11 รูปสเปกตรัมสัญญาณ Sine Wave 6KHz ขนาด 130mV 72

รูปที่ 7.12 รูปสเปกโตรแกรมสัญญาณ Sine Wave 6KHz ขนาด 130mV 73

รูปที่ 7.13 รูปสัญญาณ Triangle Wave 6KHz ขนาด 130mV 74

รูปที่ 7.14 รูปสเปกตรัมสัญญาณ Triangle Wave 6KHz ขนาด 130mV 74

รูปที่ 7.15 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 6KHz ขนาด 130mV 75

รูปที่ 7.16 รูปสัญญาณ Square Wave 6KHz ขนาด 130mV 76

รูปที่ 7.17 รูปสเปกตรัมสัญญาณ Square Wave 6KHz ขนาด 130mV 76

รูปที่ 7.18 รูปสเปกโตรแกรมสัญญาณ Square Wave 6KHz ขนาด 130mV 77

รูปที่ 7.19 รูปสัญญาณ Sine Wave 1KHz ขนาด 2V 78

รูปที่ 7.20 รูปสเปกตรัมสัญญาณ Sine Wave 1KHz ขนาด 2V 78

รูปที่ 7.21 รูปสเปกโตรแกรมสัญญาณ Sine Wave 1KHz ขนาด 2V 79

รูปที่ 7.22 รูปสัญญาณ Triangle Wave 1KHz ขนาด 2V 80

รูปที่ 7.23 รูปสเปกตรัมสัญญาณ Triangle Wave 1KHz ขนาด 2V 80

รูปที่ 7.24 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 1KHz ขนาด 2V 81

รูปที่ 7.25 รูปสัญญาณ Square Wave 1KHz ขนาด 2V 82

รูปที่ 7.26 รูปสเปกตรัมสัญญาณ Square Wave 1KHz ขนาด 2V 82

รูปที่ 7.27 รูปสเปกโตรแกรมสัญญาณ Square Wave 1KHz ขนาด 2V 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตำแหน่งขา และหน้าที่การทำงานของ TMS320C50 6

ตารางที่ 2.2 การเซ็ตบิตรีจิสเตอร์ควบคุมหน่วยความจำเพื่อใช้หน่วยความจำข้อมูล 14

ตารางที่ 2.3 ตำแหน่งแอดเดรสข้อมูลเพจ 0 15

ตารางที่ 2.4 การกำหนดค่าสำหรับใช้หน่วยความจำโปรแกรม 16

ตารางที่ 2.5 ตำแหน่งของอินเทอร์รัพท์เวกเตอร์ 17

ตารางที่ 2.6 ขาสัญญาณของพอร์ตอนุกรม 20

ตารางที่ 2.7 รีจิสเตอร์พอร์ตอนุกรม 20

ตารางที่ 2.8 รีจิสเตอร์ควบคุมไทม์เมอร์ (TCR) 22

ตารางที่ 3.1 แสดงตำแหน่งขา และหน้าที่การทำงานแต่ละขา 24

ตารางที่ 3.2 แสดงการกำหนดค่าใน D1 และ D0 ในการทำงานโหมดต่าง ๆ 29

ตารางที่ 3.3 แสดงรูปแบบการสื่อสารแบบอนุกรมรองของ DX 30

ตารางที่ 3.4 เงื่อนไข และการตอบสนองของ AIC 31

ตารางที่ 5.1 การเชื่อมต่อกับเครื่องพิมพ์โดยใช้สายเคเบิลแบบเซนโทรนิกส์ 39

ตารางที่ 5.2 รีจิสเตอร์ข้อมูล 41

ตารางที่ 5.3 รีจิสเตอร์สถานะ 41

ตารางที่ 5.4 รีจิสเตอร์ควบคุม 42

ตารางที่ 6.1 แสดงการสลับตำแหน่งข้อมูลเข้าด้วยการผันบิตกลับ 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ดิจิตอลซิกแนลโปรเซสซิ่ง ก็คือการเปลี่ยนรูปแบบของสัญญาณ โดยใช้การคำนวณทางคณิตศาสตร์ด้วยดิจิตอลซิกแนลโปรเซสเซอร์ ซึ่งจะเห็นได้ว่าเราสามารถจะนำสัญญาณนั้นใส่ลงในสมการคณิตศาสตร์ แล้วได้ผลลัพธ์ออกมาได้โดยตรง

ดิจิตอลซิกแนลโปรเซสซิ่งสามารถนำไปประยุกต์ใช้งานได้อย่างมากมาย อาทิเช่น การทำอิมเมจโปรเซสซิ่ง (Image Processing) ,การทำฟิลเตอร์ และ การคำนวณการแปลงฟาสท์ฟูริเยร์ (Fast Fourier Transform : FFT) เพื่อนำเอาไปใช้ดูผลตอบสนองของสัญญาณ ในที่นี้ จะขอกล่าวถึงการประยุกต์ใช้งานดิจิตอลโปรเซสซิ่งในการคำนวณ FFT แล้วนำผลลัพธ์ที่ได้มาแสดงผลในรูปสเปกโตรแกรม สเปกโตรแกรม คือ การนำเอาค่าสเปกตรัมของสัญญาณที่คำนวณได้จาก FFT มาแสดงในแกนความถี่ และ แกนเวลา พร้อม ๆ กัน โดยให้แกนนอนเป็นแกนเวลา แกนตั้งเป็นแกนความถี่ และค่าขนาดแสดงโดยใช้สีต่าง ๆ

ที่ใช้การคำนวณแบบ FFT เพื่อการประมวลผลแบบเวลาจริง (Real Time) นั่นเอง การประมวลผลแบบเวลาจริง คือ เมื่อเราป้อนสัญญาณอินพุตเข้าไปก็จะได้สัญญาณเอาท์พุตออกมาเลย โดยมีการหน่วงเวลา (Delay) น้อยมาก ซึ่งจะมีประโยชน์ในการใช้งานมากกว่าการประมวลผลแบบไม่ใช่เวลาจริง

ในปริญญานิพนธ์นี้ เราใช้การ์ด DSK (DSP Starter Kit) ในการคำนวณ FFT แล้วนำผลที่ได้มาแสดงผลในรูปสเปกโตรแกรม ซึ่งในการ์ด DSK ประกอบไปด้วย TMS320C50 และ TLC32040 ตัว TMS320C50 เป็นตัวประมวลผลที่ถูกออกแบบมาโดยเฉพาะให้สามารถทำงานได้อย่างประสิทธิภาพโดยการประมวลผลแบบขนาน และใช้ชุดคำสั่งที่สร้างขึ้นมาเฉพาะ ทั้งนี้ก็เพื่อให้สามารถประมวลผลแบบเวลาจริงได้ และตัว TLC32040 เป็นตัวอินเทอร์เฟสอนาลอก โดยจะทำการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลเพื่อประมวลผลใน TMS320C50 และรับผลจาก TMS320C50 มาแปลงจากสัญญาณดิจิตอลเป็นสัญญาณอนาลอกอีกครั้งหนึ่งเพื่อแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

TMS320C50

TMS320C50 เป็นโปรเซสเซอร์ที่ทำงานด้าน DSP (Digital Signal Processing) ซึ่งตระกูล TMS320C5X จะมีอยู่หลายเบอร์ เช่น TMS320C50 TMS320C51 และ TMS320C53 ซึ่งใช้ซีพียู ของ TMS320C25 ร่วมสถาปัตยกรรมและปรับปรุงให้ดีขึ้น ทั้งได้ความเร็วและความสะดวกอื่น ๆ

2.1 จุดเด่นของTMS320C50

1. มีแรมบนบอร์ด 10K เวิร์ด
2. ทำงานได้รวดเร็ว คือ 30-50 ns ต่อหนึ่งคำสั่ง
3. ใช้งานแทน TMS320C1X และ TMS320C2X ได้
4. มีแรม สำหรับ โปรแกรม/ข้อมูล ขนาด 9K x 16 บิต
5. มีรอมขนาด 2K x 16 บิต สำหรับการบูต
6. สามารถต่อหน่วยความจำข้างนอกได้ถึง 224K x 16 บิต ซึ่งประกอบด้วย 64K หน่วยความจำสำหรับเก็บโปรแกรม 64K หน่วยความจำเก็บข้อมูล 64K สำหรับ I/O และอื่น ๆ อีก 64K
7. มี ALU (Arithmetic Logic Unit) , ACC (Accumulator) และ ACCB (Accumulator Buffer) ขนาด 32 บิต
8. มี PLU (Parallel Logic Unit) ขนาด 16 บิต
9. มีคำสั่งในการคูณ 16 บิต ที่ทำงานใน 1 ไซเคิล
10. มีรีจีสเตอร์ถึง 8 ตัว ในการคำนวนและเก็บค่า
11. มีสแต็ก (Stack)
12. มีคำสั่งสำหรับการเลื่อนบิต (Shift) ตั้งแต่ 0-16 บิต
13. มีการอ้างแอดเดรสแบบเซอร์คูลาร์ (Circular) โดยการใช้เซอร์คูลาร์บัฟเฟอร์ (Circular Buffer)
14. มีคำสั่งสำหรับการทำซ้ำ ซึ่งเฉพาะสำหรับการเคลื่อนย้ายบล็อกภายในคำสั่งเดียว
15. มีคำสั่งสำหรับเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำข้อมูลและหน่วยความจำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. มีพอร์ตอนุกรมที่รับส่งแบบฟูลดูเพล็ก (Full Duplex) สำหรับ TMS320C50 กับอุปกรณ์อื่น ๆ
17. มี Time-division Multiple (TDM) ของพอร์ตอนุกรม
18. สามารถกำหนดสัญญาณนาฬิกา โดยใช้ไทม์เมอร์(Timer , เคาท์เตอร์ (Counter)
19. มีซอฟแวร์ได้ทั้ง หยุด (Stop), เริ่ม (Start) และรีเซ็ต (Reset)
20. มี พอร์ต I/O ได้ถึง 64K และ มี 16 ตำแหน่ง สำหรับการเข้าถึงหน่วยความจำ
21. ทำงานแบบไปป์ไลน์ (Pipeline)
22. สามารถผลิตสัญญาณนาฬิกา โดยการหารสัญญาณนาฬิการับมาได้
23. ใช้เทคโนโลยีของซีมอส (CMOS) และใช้ไฟเลี้ยงเพียง 5V
24. ไอซีมีขนาด 132 ขา

สถาปัตยกรรมของ TMS320C50 สร้างขึ้นมาเพื่อความเร็วในการทำงานด้าน DSP และเพื่อให้การทำงานของบัสไม่ขึ้นต่อกัน จึงแยกเป็นบัสโปรแกรม (Program Bus) และบัสข้อมูล (Data Bus) ออกจากกัน โดยบัสโปรแกรมจะเป็นทางเข้าของรหัสคำสั่ง และโอเปอร์แลนด์ของคำสั่ง ส่วนบัสข้อมูลจะเชื่อมต่อโดยตรงกับวงจรการทำงานประมวลผล เช่น ALU และ AR0-AR7 ซึ่งโครงสร้างการคำนวนทางคณิตศาสตร์นี้ ยึดหลักการให้ทำงานด้วยประสิทธิภาพ เช่น การเลื่อนบิต การคูณ และคำสั่งทางลอจิก

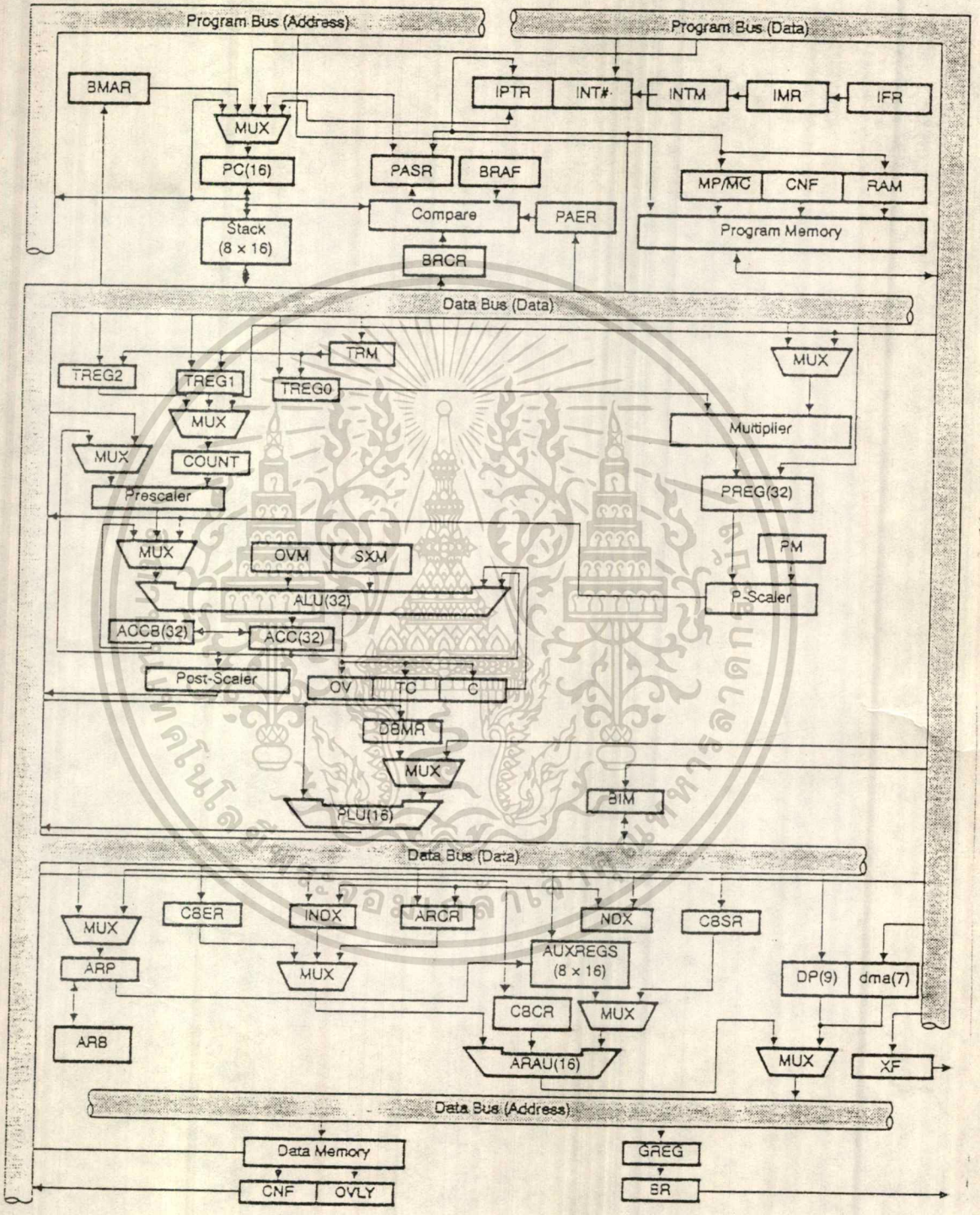
นอกจากนี้ TMS320C50 ยังมีหน่วยประมวลผลกลาง (Central Arithmetic Logic Unit หรือ CALU) ไว้ทำงานด้านการคำนวณโดยเฉพาะ ซึ่งจะประกอบด้วยรีจิสเตอร์ต่าง ๆ คือ รีจิสเตอร์สำหรับการเลื่อน (Scaling Shifter), วงจรคูณขนาด 16 x 16 บิต (16 x 16-bit Parallel Multiplier), หน่วยประมวลผลทางคณิตศาสตร์ขนาด 32 บิต (32-bit Arithmetic Logic Unit หรือ ALU) ,แอคคิวมูเลเตอร์และแอคคิวมูเลเตอร์บัฟเฟอร์ขนาด 32 บิตที่ช่วยให้การคำนวณทำได้อย่างรวดเร็ว รูปที่ 2.2 แสดงบล็อกไดอะแกรมส่วนประกอบของ CALU ซึ่งขั้นตอนโดยทั่วไปในการทำงานคำสั่งที่เกี่ยวกับการคำนวณ คือ

1. ข้อมูลถูกนำมาจากหน่วยความจำมาไว้บนบัสข้อมูล
2. ข้อมูลผ่านไปยังรีจิสเตอร์สำหรับการเลื่อน และ ALU เพื่อใช้คำนวณตามคำสั่ง
3. ผลการคำนวณที่ได้จะเก็บไว้ที่แอคคิวมูเลเตอร์

ข้อมูลที่อินพุตที่ใช้ในการคำนวณในคำสั่งเกี่ยวกับการคำนวณจะเป็นข้อมูลในแอคคิวมูเลเตอร์ และอินพุตอีกตัวอาจจะมาจากรีจิสเตอร์ที่เก็บผลลัพธ์การคูณ (PREG) ,รีจิสเตอร์สำหรับการเลื่อน ,แอคคิวมูเลเตอร์บัฟเฟอร์ หรืออาจจะถูกโหลดจากหน่วยความจำข้อมูล หรืออาจจะเป็นข้อมูลในแอคคิวมูเลเตอร์เองก็ได้

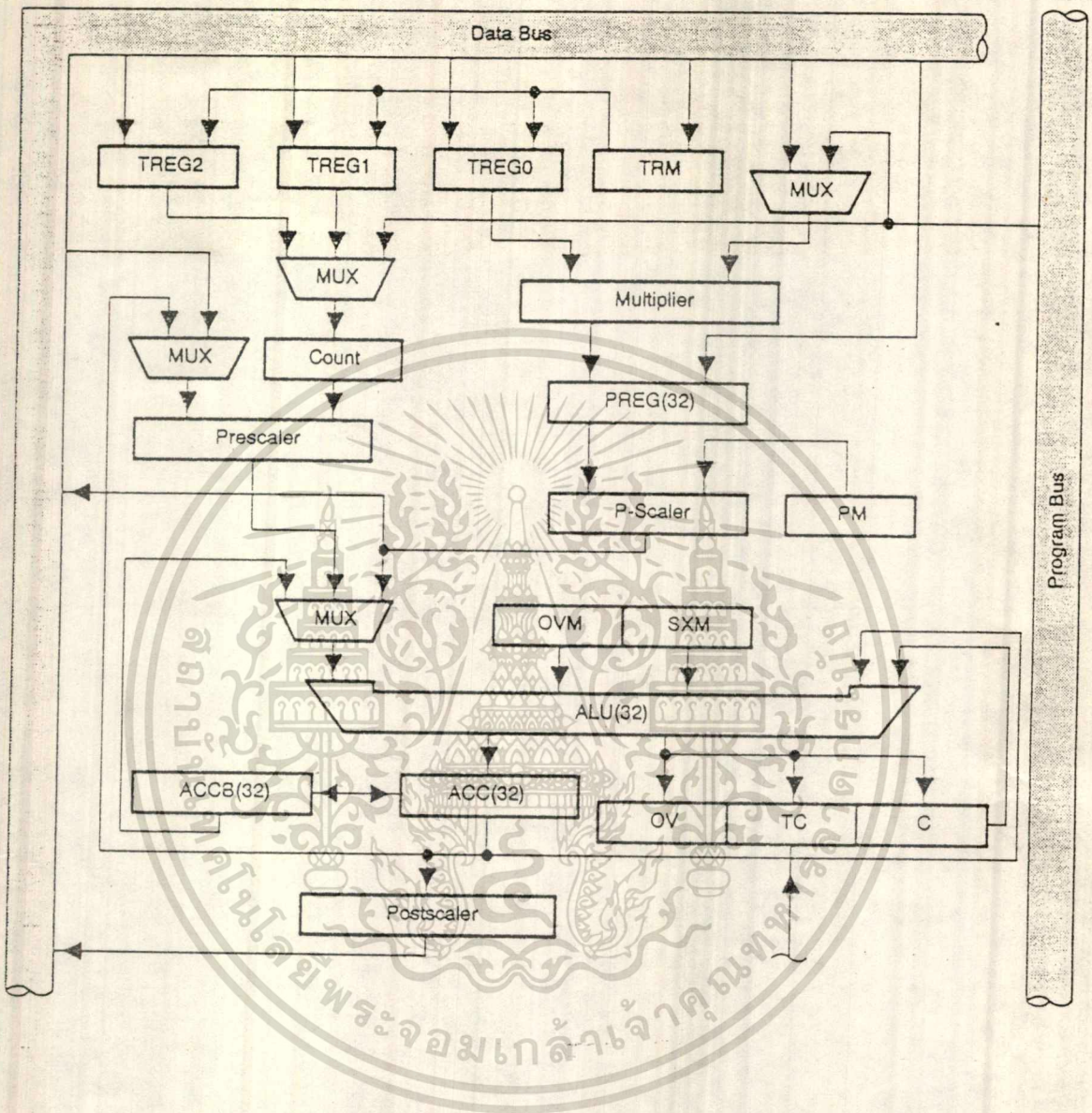
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดอะแกรมของ TMS320C50 แสดงดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมของ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 บล็อกไดอะแกรมส่วนหน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ตำแหน่งขา และหน้าที่การทำงานของ TMS320C50

ตำแหน่งขา และหน้าที่การทำงานของ TMS320C50 แสดงในตารางที่ 2.1

ตารางที่ 2.1 ตำแหน่งขา และหน้าที่การทำงานของ TMS320C50

สัญญาณ	ขา	สถานะ	การทำงาน
			กลุ่ม Adress และ Data Bus
A15 (MSB)	77	I/O/Z	เป็นบัสแบบขนาน (Parallel Address Bus) ใช้สำหรับ
A14	76		ชี้ตำแหน่งของหน่วยความจำข้อมูลและหน่วยความจำ
A13	75		โปรแกรม หรือ I/O ภายนอก เมื่ออยู่ในโหมดโฮลด์
A12	74		(Hold Mode) จะเป็นอิมพีแดนซ์สูง (High Impedance)
A11	73		สัญญาณเหล่านี้ใช้เป็นอินพุต สำหรับ DMA ภายนอก
A10	72		ของแรมภายใน (On-chip Single Access RAM) ซึ่งจะ
A9	64		เป็นอินพุต เมื่อ $\overline{HOLDA}$ และ $\overline{BR}$ ถูกขับ (Drive)
A8	63		ให้เป็นสถานะต่ำ
A7	62		
A6	61		
A5	60		
A4	59		
A3	58		
A2	57		
A1	56		
A0 (LSB)	55		
D15 (MSB)	6	I/O/Z	เป็นบัสข้อมูลแบบขนาน (Parallel Data Bus) ใช้ส่ง
D14	7		ผ่านข้อมูลระหว่างซีพียูหลัก (Core CPU) กับหน่วย
D13	8		ความจำข้อมูลหรือโปรแกรมภายนอก หรืออุปกรณ์ I/O
D12	9		เมื่อไม่มีเอาท์พุตสัญญาณเหล่านี้จะเป็นอิมพีแดนซ์สูง
D11	10		หรือเมื่อขา $\overline{RS}$ หรือ $\overline{HOLD}$ อยู่ในสถานะต่ำและขา
D10	11		$\overline{OFF}$ เป็นสถานะต่ำ นอกจากนี้ยังใช้สำหรับ DMA
D9	12		ภายนอกของแรม (Single Access RAM)
D8	13		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
D7 D6 D5 D4 D3 D2 D1 D0	23 24 25 26 27 28 29 30		
			กลุ่มสัญญาณควบคุมหน่วยความจำ
$\overline{DS}$ $\overline{PS}$ $\overline{IS}$	89 91 90	O/Z	เป็นสัญญาณเลือกหน่วยความจำข้อมูลหรือโปรแกรม และ I/O ปกติจะอยู่ในสถานะสูง แต่เมื่อเป็นสถานะต่ำจะเป็นการติดต่อกับภายนอก เมื่อขา $\overline{OFF}$ อยู่ในสถานะต่ำจะอยู่ในสถานะอิมพีแดนซ์สูง
READY	128	I	สัญญาณข้อมูลพร้อม (Data Ready Input) ใช้แสดงเมื่ออุปกรณ์ภายนอกส่งข้อมูลเรียบร้อยแล้ว และเมื่อยังทำงานไม่เสร็จ (READY = 0) จะต้องมีการรอ 1 ไซเคิล และเช็คขา READY อีกครั้ง ในสภาวะปกติขา READY จะทำงานหลังจากที่มีสัญญาณ $\overline{BR}$
$R / \overline{W}$	92	I/O/Z	สัญญาณอ่านหรือเขียน (Read/Write Signal) เป็นสัญญาณควบคุมการอ่านและเขียนข้อมูล จะเป็นสถานะอิมพีแดนซ์สูงเมื่ออยู่ในโหมดโฮลด์ และถูกใช้ใน DMA ของแรมภายนอกเมื่อ $\overline{HOLDA}$ และ $\overline{IAQ}$ อยู่ในสถานะต่ำ สัญญาณนี้ใช้แสดงทิศทางของบัสข้อมูลสำหรับ DMA อ่าน (เมื่ออยู่ในสถานะสูง) และเขียน (เมื่ออยู่ในสถานะต่ำ)
$\overline{STRB}$	93	I/O/Z	สัญญาณสโตรบ (Strobe Signal) ปกติมักอยู่ในสถานะสูง และจะเป็นสถานะต่ำเมื่อแสดงค่าของบัสภายนอกเป็นอิมพีแดนซ์สูง เมื่ออยู่ในโฮล์ดโหมด เมื่อ $\overline{HOLDA}$ และ $\overline{IAQ}$ สัญญาณนี้จะใช้เลือกการเข้าถึงหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
$\overline{RD}$	82	O/Z	สัญญาณเลือกอ่าน (Read Select) ขานี้จะต่อโดยตรงกับ OE เมื่อมีการอ่านของอุปกรณ์ภายนอก สัญญาณนี้จะใช้ในการอ่านค่าหน่วยความจำโปรแกรม หรือข้อมูล และ I/O ภายนอกทั้งหมดเป็นอิมพีแดนซ์สูงเมื่ออยู่ในโหมดโฮล์ด
$\overline{WE}$	83	O/Z	สัญญาณเขียน (Write Enable) จะใช้สำหรับการเขียนค่าในหน่วยความจำโปรแกรม หรือข้อมูล และ I/O ภายนอกทั้งหมดเป็นอิมพีแดนซ์สูงเมื่ออยู่ในโหมดโฮล์ด
			กลุ่มสัญญาณ Multiprocessing
$\overline{HOLD}$	129	I	สัญญาณโฮล์ด (Hold Input) เป็นสัญญาณที่ใช้เพื่อแสดงว่ากำลังมีการติดต่อกับบัสตำแหน่งข้อมูลและบัสควบคุมเมื่อถูกตอบรับ (Acknowledge) โดย TMS320C5X เหล่านี้จะอยู่ในสภาวะอิมพีแดนซ์สูง
$\overline{HOLDA}$	108	O/Z	สัญญาณตอบรับสัญญาณโฮล์ด (Hold Acknowledge Signal) ใช้แสดงว่าวงจรอยู่ในสถานะโฮล์ดของบัสตำแหน่ง บัสข้อมูล และบัสควบคุม อยู่ในสภาวะอิมพีแดนซ์สูง
$\overline{BR}$	94	I/O/Z	สัญญาณขอการใช้บัส (Bus Request Signal) แสดงเมื่อมีการติดต่อกับหน่วยความจำข้อมูล สัญญาณจากขานี้ใช้กับหน่วยความจำข้อมูลที่ว่างได้ถึง 32k Words เมื่อขา $\overline{HOLDA}$ อยู่ในสถานะต่ำสัญญาณจากขานี้ใช้กับ DMA ของแรมภายนอก $\overline{BR}$ จะเป็นสถานะต่ำเมื่อมีการติดต่อกับแรมภายนอก
$\overline{IAQ}$	1	O/Z	สัญญาณรับคำสั่ง (Instruction Acquisition Signal) จะแสดงค่าสถานะต่ำเมื่อมีเข้าถึงคำสั่งที่อยู่บนแอดเดรสบัส และใช้กับ DMA ของแรมภายนอกเมื่อ $\overline{HOLDA}$ อยู่ในสถานะต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
$\overline{BIO}$	130	I	สัญญาณควบคุมบรานช์ (Branch Control Input) ถ้าเป็นสถานะต่ำจะเป็นการให้ทำคำสั่งที่เป็นเงื่อนไข สัญญาณนี้จะทำงานเมื่อมีการนำคำสั่ง (Fetch) ที่เป็นเงื่อนไข
XF	109	O/Z	สัญญาณติดต่อภายนอก (External Flag Output) ถูกเซตให้เป็นสถานะสูงหรือสถานะต่ำโดยคำสั่งพิเศษ หรือโดยโหลดค่าในเสตทรีจิสเตอร์ (State Register (ST1)) เมื่อรีเซตขานี้จะเป็นสถานะสูง
$\overline{IACK}$	112	O/Z	สัญญาณตอบรับอินเทอร์รัพท์ (Interrupt Acknowledge Signal) แสดงค่าเมื่อมีการรับค่าอินเทอร์รัพท์และโปรแกรมเคาท์เตอร์จะนำค่าอินเทอร์รัพท์เวกเตอร์ (Interrupt Vector) ซึ่งกำหนดโดย A15-A0
			Initialization,Interrupt และ Reset Operations
INT4 INT3 INT2 INT1	41 40 39 38	I	สัญญาณอินเทอร์รัพท์จากผู้ใช้ภายนอก (External User Interrupt Input) ซึ่งกำหนดโดยรีจิสเตอร์ควบคุมอินเทอร์รัพท์ (Interrupt Mask Register) และอินเทอร์รัพท์โหมดบิต (Interrupt Mode Bit) โดยสามารถรีเซตผ่านรีจิสเตอร์บอกอินเทอร์รัพท์ (Interrupt Flag Register)
$\overline{NMI}$	42	I	สัญญาณนอน-มาร์กเอเบิ้ลอินเทอร์รัพท์ (Non Maskable Interrupt) เป็นอินเทอร์รัพท์ภายนอก ไม่สามารถควบคุมโดย INTM หรือ IMR เมื่อ $\overline{MNI}$ ทำงานก็จะมีการทำงานอินเทอร์รัพท์
$MC / \overline{MP}$	5	I	ขาเลือกโหมดไมโครโปรเซสเซอร์หรือไมโครคอมพิวเตอร์(Microprocessor/Microcomputer Mode Select Pin) ถ้ามีค่าเป็นสถานะต่ำ (Microcomputer Mode) จะทำให้โปรแกรมรอมภายในส่งไปยังหน่วยความจำโปรแกรม (Program Memory Space) ในโหมดไมโครโปรเซสเซอร์ (Microprocessor Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
			สัญญาณ Oscillator/Timer และ CLKIN1/2
CLKOUT1	110	O/Z	สัญญาณนาฬิกาส่งออก (Master Clock Output Signal หรือ CLKIN2 Frequency) มีค่าไซเคิลเท่ากับอัตราแมชชีนไซเคิล (Machine-Cycle) ของซีพียู
CLKMD1 CLKMD2	71 103	I	CLKMD1 CLKMD2 0 0 เป็นสัญญาณนาฬิกาภายนอก เป็นสัญญาณนาฬิกาเข้าจากขา X2/CLKIN ทำให้ออสซิลเลเตอร์ภายใน และ PLL Disable 0 1 สำหรับตรวจสอบ 1 0 เป็นสัญญาณนาฬิกาอินพุต (Input Clock) สำหรับ CLKIN2 ทำให้ออสซิลเลเตอร์ภายในหยุดทำงาน และ PLL ทำงานแทน 1 1 เป็นสัญญาณนาฬิกาอินพุตสำหรับขา X2/CLKIN1 ทำให้ออสซิลเลเตอร์ภายในทำงาน และ PLL-ภายในไม่ทำงาน
X2/CLKIN	122	O	ขาอินพุตสำหรับออสซิลเลเตอร์ภายใน (Input Pin to Internal Oscillator from Crystal) ถ้าออสซิลเลเตอร์ภายในไม่ถูกใช้สัญญาณนาฬิกาจะเป็นอินพุตสำหรับอุปกรณ์บนขานี้ แมชชีนไซเคิลภายในเป็นครึ่งหนึ่งของอัตรา CLK
X1			เป็นขาเอาท์พุตของออสซิลเลเตอร์ภายในสำหรับคริสตัล ถ้าไม่ใช้ออสซิลเลเตอร์ภายในจะไม่มีการต่อกับขานี้
CLKIN2			เป็นอินพุตสำหรับสัญญาณนาฬิกาสำหรับขับอัตราแมชชีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
TOUT			เอาท์พุตไทม์เมอร์ (Timer Output) ขานี้ให้สัญญาณพัลส์เมื่อไทม์เมอร์ภายใน (On-chip Timer) นับถึง 0 ความกว้างพัลส์เท่ากับ CLKOUT 1 ไซเคิล
			สัญญาณ Serial Port
CLKR CLKRT	46 126	I I	เป็นขาที่รับสัญญาณนาฬิกาจากข้างนอกเพื่อกำหนดให้การรับข้อมูล (DR/TDR) เข้าไปเก็บไว้ที่ RSR (Serial Port Reccive Shift Register) แต่ถ้าขานี้ไม่ใช้สามารถที่จะใช้เป็นขาอินพุตของ IN0 ของ SPC/TSPC รีจิสเตอร์
CLKX TCLKX	124 123	I/O/Z I/O/Z	เป็นขาที่แสดงสัญญาณนาฬิกาจากข้างนอกเพื่อกำหนดให้ DR/TDR ส่งข้อมูลไปที่ DX/TDX CLKX จะเป็นอินพุตถ้า MCM บิตที่การควบคุมพอร์ตอนุกรม (Serial Port Control) มีค่าเป็น 0 และอาจจะขับความถี่เป็น $\frac{1}{4}$ CLKOUT1 เมื่อ MCM เป็น 1 ถ้าขานี้ไม่ใช้สามารถที่จะทำเป็นอินพุตของบิต IN1 ของ SPC/TSPC รีจิสเตอร์
DR TDR	43 44	I I	เป็นขาเพื่อรับสัญญาณข้อมูลซึ่งเมื่อรับมาแล้วจะเก็บไว้ที่ RSR (Serial Port Receive Shift Register)
DX TDX	106 107	O/Z	เป็นขาเพื่อส่งสัญญาณข้อมูลซึ่งข้อมูลจะส่งจาก XSR (Serial Port Transmit Shift Register)
FSR TFSR/TADD	45 125	I I/O/Z	สัญญาณการพร้อมของเฟรม(Frame Synchronization) สำหรับรับสัญญาณอินพุต TFSR จะเป็นได้ทั้งอินพุตหรือเอาท์พุต (TADD) เมื่อพอร์ตอนุกรมอยู่ในโหมด TDM
FSX TFSX/TFRM	104 105	I/O/Z I/O/Z	สัญญาณการพร้อมของเฟรม (Frame Synchronization) สำหรับการส่งสัญญาณ ขานี้จะเลือกได้โดยทางซอฟแวร์ จะเป็นเอาท์พุตเมื่อ TSM ถูกเซตให้เป็น 1
TDI	67	I	เป็นสัญญาณนาฬิกาเพื่อเลือกรีจิสเตอร์ในขอบขาขึ้นของ TCK
TDO	100	O/Z	เป็นการทดสอบเอาท์พุตข้อมูลในขอบขาลงของ TCK

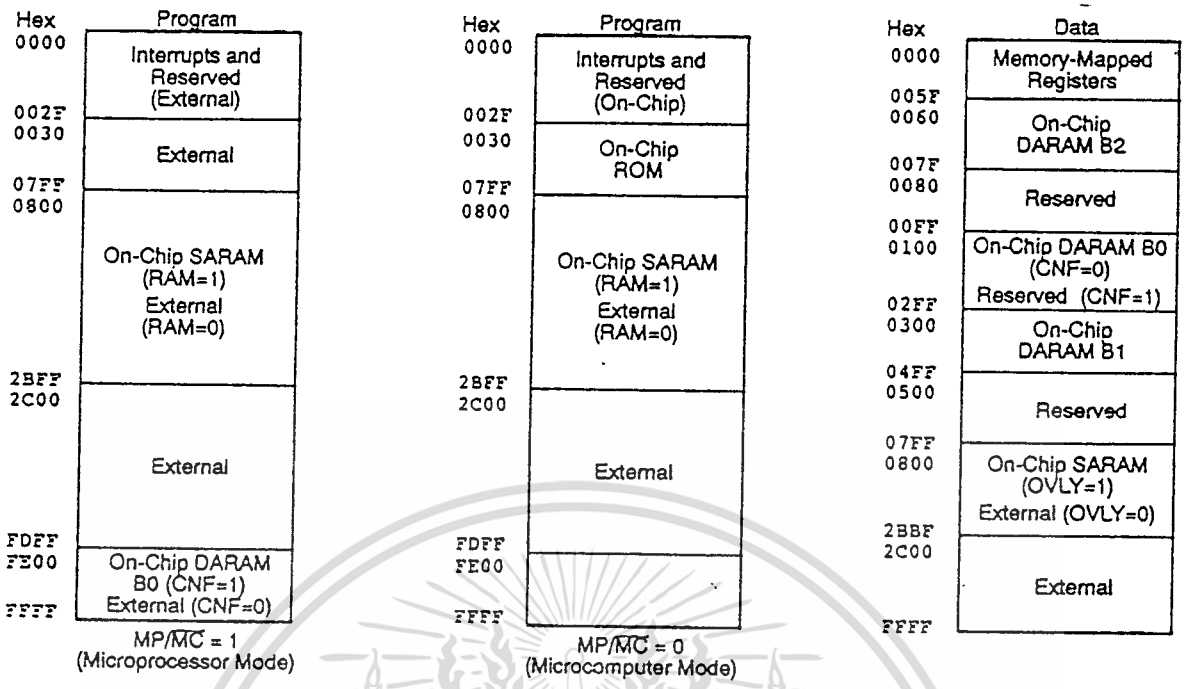
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	ขา	สถานะ	การทำงาน
TMS	31	I	เป็นการเลือกโหมดทดสอบ JTAG และเป็นสัญญาณนาฬิกาอินพุตที่ทดสอบพอร์ตตรวจสอบการเข้าถึง (Test Access Port (TAP)) จะทำงานที่ขอบขาขึ้นของ TCK
TCK	34	I	สัญญาณนาฬิกาตรวจสอบ JTAG (JTAG Test Clock) เป็นสัญญาณนาฬิกาแบบฟรี-รันนิ่ง (Free-running) ซึ่งมีค่าดิวตี้ไซเคิล (Duty Cycle 50%) การเปลี่ยน TAP (Test Access Port) ซึ่งเป็นอินพุตสัญญาณนาฬิกาจะเข้าไปควบคุม TAP รีจิสเตอร์คำสั่ง (Instruction Registe) หรือเลือกการทดลองรีจิสเตอร์ข้อมูล (Data Register) ที่ขอบขาขึ้นของ TCK ในการเปลี่ยน TAP จะเป็นสัญญาณเอาท์พุตจะปรากฏที่ขอบขาลงของ TCK
$\overline{TRST}$	2	I	ทดสอบการรีเซตจะเป็นสถานะสูง
EMU0 $EMU1/\overline{OFF}$	118 119	I/O/Z	ขาอีมูเลเตอร์ 1 หรือขาหยุดการทำงานของเอาท์พุต (emulator pin 1 / disable all output) จะทำงานที่สถานะต่ำ

2.3 การจัดหน่วยความจำของ TMS320C50

TMS320C50 จะมีการแบ่งหน่วยความจำเป็นส่วนของหน่วยความจำโปรแกรม และหน่วยความจำข้อมูล ตามรูป 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การเข้าถึงหน่วยความจำ ของ TMS320C50

2.3.1 หน่วยความจำข้อมูล (Data Memory)

ในการออกแบบของ TMS320C5X จะใช้สถาปัตยกรรมแบบฮาร์วาร์ด (Harward) กล่าวคือ มีการแยกหน่วยความจำข้อมูลและหน่วยความจำโปรแกรมออกจากกัน ซึ่งทำให้แยกส่วนอย่างชัดเจนซึ่งมีผลดีต่อความเร็ว

หน่วยความจำข้อมูลของ TMS320C5X สามารถขยายได้ถึง 64Kx16 บิต ซึ่ง TMS320C51 มีแรมบนชิพถึง 9K ซึ่งแรมจะเป็นแบบ SARAM (Single Access RAM) และ DARAM (Dual Access RAM) ขนาด 1056 เวิร์ด บนชิพ ซึ่งมีข้อดี คือ

1. มีการปฏิบัติการได้เร็วไม่มีการรอ (Wait State)
2. ทำงานโดยวิธีการไปป์ไลน์ ทำให้ทำงานได้รวดเร็ว
3. ช่วยลดค่าใช้จ่ายเรื่องการต่อหน่วยความจำภายนอก
4. ช่วยประหยัดไฟ ซึ่งถ้าให้การต่อหน่วยความจำภายนอกจะเปลืองกว่า

ในการกำหนดการใช้หน่วยความจำข้อมูลจะต้องมีการเซ็ตค่าของบิตของรีจิสเตอร์ควบคุมหน่วยความจำ (Control State Register) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การเซตบิตรีจิสเตอร์ควบคุมหน่วยความจำเพื่อใช้หน่วยความจำข้อมูล

CNF	OVLY	DRAM B0	DRAM B1	DRAM B2	SARAM	Off-Chip
0	0	100h-2FFh	300h-4FFh	60h-7Fh		800h-FFFFh
0	1	100h-2FFh	300h-4FFh	60h-7Fh	800h-2BFFh	2C00h-FFFFh
1	0	-	300h-4FFh	60h-7Fh		800h-FFFFh
1	1	-	300h-4FFh	60h-7Fh	800h-2BFFh	2C00h-FFFFh

เนื่องจากการติดต่อกับรีจีสเตอร์ต่าง ๆ และการส่งค่าต่าง ๆ จะทำบนเพจ 0 ซึ่งจะเป็นการส่งแบบรีจิสเตอร์ (Register Memory Map) ดังตารางที่ 2.3

ตารางที่ 2.3 ตำแหน่งแอดเดรสข้อมูลเพจ 0

Name	Address		Description
	Dec	Hex	
Core Processor Memory-Mapped Registers			
——	0–3	0–3	Reserved
IMR	4	4	Interrupt Mask Register
GREG	5	5	Global Memory Allocation Register
IFR	6	6	Interrupt Flag Register
PMST	7	7	Processor Mode Status Register
RPTC	8	8	Repeat Counter Register
BRCR	9	9	Block Repeat Counter Register
PASR	10	A	Block Repeat Program Address Start Register
PAER	11	B	Block Repeat Program Address End Register
TREG0	12	C	Temporary Register Used for Multiplicand
TREG1	13	D	Temporary Register Used for Dynamic Shift Count (5 bits only)
TREG2	14	E	Temporary Register Used as Bit Pointer In Dynamic Bit Test (4 bits only)
DBMR	15	F	Dynamic Bit Manipulation Register
AR0	16	10	Auxiliary Register Zero
AR1	17	11	Auxiliary Register One
AR2	18	12	Auxiliary Register Two
AR3	19	13	Auxiliary Register Three
AR4	20	14	Auxiliary Register Four
AR5	21	15	Auxiliary Register Five
AR6	22	16	Auxiliary Register Six
AR7	23	17	Auxiliary Register Seven
INDX	24	18	Index Register
ARCR	25	19	Auxiliary Register Compare Register
CBSR1	26	1A	Circular Buffer 1 Start Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตำแหน่งแอดเดรสข้อมูลเพจ 0 (ต่อ)

Name	Address		Description
	Dec	Hex	
CBER1	27	1B	Circular Buffer 1 End Register
CBSR2	28	1C	Circular Buffer 2 Start Register
CBER2	29	1D	Circular Buffer 2 End Register
CBCR	30	1E	Circular Buffer Control Register
BMAR	31	1F	Block Move Address Register
			Peripheral Memory-Mapped Registers
DRR	32	20	Data Receive Register
DXR	33	21	Data Transmit Register
SPC	34	22	Serial Port Control Register
——	35	23	Reserved
TIM	36	24	Timer Register
PRD	37	25	Period Register
TCR	38	26	Timer Control Register
—	39	27	Reserved
PDWSR	40	28	Program/Data S/W Wait-State Register
IOWSR	41	29	I/O Port S/W Wait-State Register
CWSR	42	2A	Control S/W Wait-State Register
—	43–47	2B–2F	Reserved for Test/Emulation
TRCV	48	30	TDM Data Receive Register
TDXR	49	31	TDM Data Transmit Register
TSPC	50	32	TDM Serial Port Control Register
TCSR	51	33	TDM Channel Select Register
TRTA	52	34	Receive/Transmit Address Register
TRAD	53	35	Received Address Register
—	54–79	36–4F	Reserved
			Memory-Mapped I/O Ports
PA0	80	50	I/O Port 80
PA1	81	51	I/O Port 81
PA2	82	52	I/O Port 82
PA3	83	53	I/O Port 83
PA4	84	54	I/O Port 84
PA5	85	55	I/O Port 85
PA6	86	56	I/O Port 86
PA7	87	57	I/O Port 87
PA8	88	58	I/O Port 88
PA9	89	59	I/O Port 89
PA10	90	5A	I/O Port 90
PA11	91	5B	I/O Port 91
PA12	92	5C	I/O Port 92
PA13	93	5D	I/O Port 93
PA14	94	5E	I/O Port 94
PA15	95	5F	I/O Port 95
B2	96–127	60–7F	Scratch Pad RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 หน่วยความจำโปรแกรม (Program Memory)

ในการใช้งานของหน่วยความจำโปรแกรมซึ่งสามารถขยายได้ถึง 64K TMS320C50 มีรอม, SARAM และ DARAM ซึ่งมีความเร็วสูงโดยไม่มีสภาวะการรอ

ในการทำงานของ TMS320C50 สามารถใช้ร่วมกับรอมภายในขนาด 4K สามารถโปรแกรมได้จากโรงงานทำให้มีความเร็วในการทำงานเต็มที่ ในการเลือกใช้จะต้องกำหนดที่ขาของ MP/MC หากขาดังกล่าวเป็นสถานะสูง ตำแหน่ง 4K แรกจะเป็นหน่วยความจำภายนอกชิพ แต่ถ้าเป็นสถานะต่ำ ตำแหน่ง 4K เวิร์ดแรกจะเป็นรอมภายในชิพ ดังรูปที่ 2.3

การกำหนดสถานะ (Status) ของค่าต่าง ๆ จะกำหนดตามตารางที่ 2.4

ตารางที่ 2.4 การกำหนดค่าสำหรับใช้หน่วยความจำโปรแกรม

CNF	RAM	$MP/\overline{MC}$	ROM	SARAM	DRAM B0	Off-Chip
0	0	0	0000-07FF			0800-FFFF
0	0	1				0000-FFFF
0	1	0	0000-07FF	2000-23FF		2C00-FFFF
0	1	1		2000-23FF		0000-FFFF
						2C00-FFFF
1	0	0	0000-07FF		FE00-FFFF	0800-FDFF
1	0	1			FE00-FFFF	0000-FDFF
1	1	0	0000-07FF	2000-23FF	FE00-FFFF	2C00-FDFF
				2000-23FF	FE00-FFFF	0000-07FF
						2C00-FDFF

การใช้งานของการอินเตอร์รัพท์จะต้องกำหนดเวกเตอร์ของแอดเดรสในหน่วยความจำโปรแกรม ซึ่งอินเตอร์รัพท์เวกเตอร์จะแสดงดังตารางที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 ตำแหน่งของอินเทอร์รัพท์เวกเตอร์

Name	Location		Priority	Function
	Dec	Hex		
$\overline{RS}$	0	0	1 (highest)	External reset signal
$\overline{NMI}$	36	24	2	Nonmaskable interrupt
$\overline{INT1}$	2	2	3	External user interrupt #1
$\overline{INT2}$	4	4	4	External user interrupt #2
$\overline{INT3}$	6	6	5	External user interrupt #3
TINT	8	8	6	Internal timer interrupt
RINT	10	A	7	Serial port receive interrupt
XINT	12	C	8	Serial port transmit interrupt
TRNT	14	E	9	TDM port receive interrupt
TXNT	16	10	10	TDM port transmit interrupt
$\overline{INT4}$	18	12	11	External user interrupt #4
——	20–33	14–21	N/A	Reserved
TRAP	34	22	N/A	Trap instruction vector
——	38–39	26–27	N/A	Reserved
——	40–63	28–3F	N/A	Software interrupts

2.4 โหมดการอ้างถึงหน่วยความจำ (Memory Addressing Mode)

การอ้างแอดเดรสสามารถอ้างได้ 64k สำหรับหน่วยความจำโปรแกรม และ 96k สำหรับหน่วยความจำข้อมูล TMS320C50 มีวิธีการอ้างแอดเดรส ได้ถึง 8 วิธีดังนี้

1. วิธีเข้าถึงโดยตรง (Direct Address Bus)โดยการใช้แอดเดรสบัสโดยตรง ซึ่งต้องมี รีจิสเตอร์ชี้เพจของข้อมูล (Data Page Pointer (DP)) บอกด้วยว่าเป็นเพจใด ซึ่ง DP สามารถอ้างได้ 512 เพจ แต่ละเพจเท่ากับ 128 เวิร์ด ตัวอย่างเช่น ADD 01 0h หมายถึง เอาข้อมูลที่ชี้ที่ตำแหน่ง 010h (ซึ่งมี DP ชี้บอกว่าอยู่เพจใด) นำมาบวกกับ ACC

2. วิธีเข้าถึงหน่วยความจำ (Memory Map) จะเหมือนกับวิธีแรกแต่จะใช้เฉพาะเพจ 0 เท่านั้น ซึ่งเพจ 0 จะเป็นรีจิสเตอร์ที่ใช้เป็นรีจิสเตอร์ที่เป็นหน่วยความจำ (Memory Map Register) เช่น LAMM PMST โดยที่ PMST เป็นรีจิสเตอร์

3. วิธีเข้าถึงโดยใช้ออกซิลลารีรีจิสเตอร์ (Auxiliary register) โดยการอ้างแอดเดรสที่บรรจุใน AR0-AR7 เช่น ADD * ซึ่ง ARP จะเป็นตัวบอกว่าเป็น AR ตัวที่เท่าใด ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARP	4	4
AR4	0302h	0302h

หน่วยความจำข้อมูล

302h	2h	2h
ACC	2h	4h

จะเห็นว่าแอดเดรสที่ได้อยู่ใน AR4 (0302h)

4. วิธีใช้รีจิสเตอร์คำสั่ง (Instruction Register) ซึ่งเป็นการเอาแอดเดรสโดยตรงมาจากคำสั่งเลย เช่น

ADD # 0FFh เอาข้อมูลที่แอดเดรส FFh มาบวกกับ ACC

โดยวิธีนี้จะอ้างได้แบบสั้น (Short Immediate) คือ ค่าแอดเดรส จะเป็น 00-FF

5. โดยการใช้ PC (Program Counter) จะเหมือนการใช้รีจิสเตอร์คำสั่ง แต่สามารถอ้างแบบยาว (Long Immediate (0000-FFFF)) เช่น

ADD # 0FFFh

6. โดยเข้าถึงรีจิสเตอร์โดยตรง (Register Access) เป็นการใช้รีจิสเตอร์ที่ทำงานโดยเฉพาะ เช่น TREG0, TREG1,TREG2, ARCR, DBMR ตัวอย่าง คำสั่ง

APL * ,AR7

7. โดยใช้คำสั่งตัวที่ 2 เป็นตัวชี้แอดเดรส เช่น คำสั่ง BLDD # 02345h, 012h หมายถึง เอาข้อมูลที่ชี้ที่แอดเดรส 02345h ไปเก็บที่แอดเดรส 012h (คิดเป็นเพจ)

8. โดยอ้างถึงหน่วยความจำเป็นบล็อก (Block Memory Address Register) ซึ่ง BMAR จะเป็นตัวชี้ข้อมูล ตัวอย่างเช่น

BLDD BMAR, 012h

2.5 อุปกรณ์เสริม (Peripherals)

TMS320C50 มีการอินเตอร์เฟสกับภายนอกได้ถึง 7 อุปกรณ์ ซึ่งประกอบด้วย

1. พอร์ตอนุกรม (Serial Port)
2. พอร์ตอนุกรมแบบแบ่งเวลา (TDM Serial Port)
3. ไทม์เมอร์ (Timer)
4. รีจิสเตอร์ใช้โปรแกรมสภาวะการรอ (Software-programable Wait State)
5. พอร์ตติดต่อภายนอก (I/O Port)
6. รีจิสเตอร์หารสัญญาณนาฬิกา (Divide-by-one Clock)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. XF และ BIO

ซึ่งทั้งหมดจะควบคุมผ่านรีจีสเตอร์ควบคุมโดยใช้วิธีการอ้างหน่วยความจำโดยการส่งลงที่แอดเดรสของหน่วยความจำโปรแกรมและข้อมูล

2.5.1 อินเทอร์รัพท์ (Interrupt)

TMS320C50 มีอินเทอร์รัพท์ทั้งหมด 16 อินเทอร์รัพท์ (INT16-INT1) แต่จะไม่ใช้พร้อมกันทั้งหมด (ปกติใช้ 9)

การรีเซต (Reset)

จะเป็นอินเทอร์รัพท์แบบนอน-มาร์กเอเบิ้ลจากภายนอก (Nonmaskable External Interrupt) ซึ่งจะเกิดขึ้นเมื่อมีการรีเซต ดังนี้

1. CNF = 0 หมายถึง การใช้หน่วยความจำข้อมูลบนชิพ
2. PC = 0000H
3. INTM = 1 และ IFR = 0 ทำให้อินเทอร์รัพท์ทุกตัวไม่ทำงาน
4. บิตสถานะ (Status Bit) จะเป็นดังนี้

   0 → OV 1 → XF 1 → SXM 0 → PM 1 → HM

   0 → BRAI 0 → TRM 0 → NDX 0 → CENM1 0 → CENM2

   0 → IPTR 0 → OVLY 0 → AVIS 0 → RAM 0 → BIG

   0 → CNF 1 → INTM MP/MC (pin) → PMST (MP/MC) 1 → C
5. GREG = 000000
6. RPTC = 00
7. IACK จะแสดงสภาวะการเก็บการรีเซต

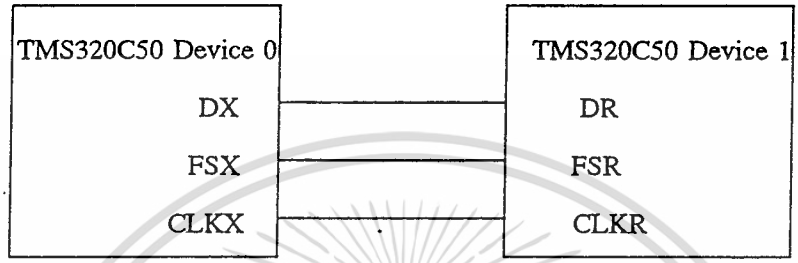
ในการควบคุมการอินเทอร์รัพท์จะมี IFR (Interrupt Flag Register) แสดงสภาวะอินเทอร์รัพท์ และ IMR (Interrupt Mask Register) แสดงการมาร์กการอินเทอร์รัพท์

ในการกำหนดให้ใช้อินเทอร์รัพท์จะต้องทำการเคลียร์ที่ INTM บิต ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
037098

2.5.2 พอร์ตอนุกรม (Serial Port)

พอร์ตอนุกรมของ TMS320C50 จะเป็นแบบฟูลดูเพล็ก คือ สามารถรับและส่งภายในเวลาเดียวกัน ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกอื่น ๆ ได้ ไม่ว่าจะเป็น CODEC A/D หรือระบบอื่น ๆ ในการต่อขาต่าง ๆ จะเป็นตามรูปที่ 2.4



รูปที่ 2.4 แสดงการต่อพอร์ตอนุกรมกับพอร์ตภายนอก

ตารางที่ 2.6 ขาสัญญาณของพอร์ตอนุกรม

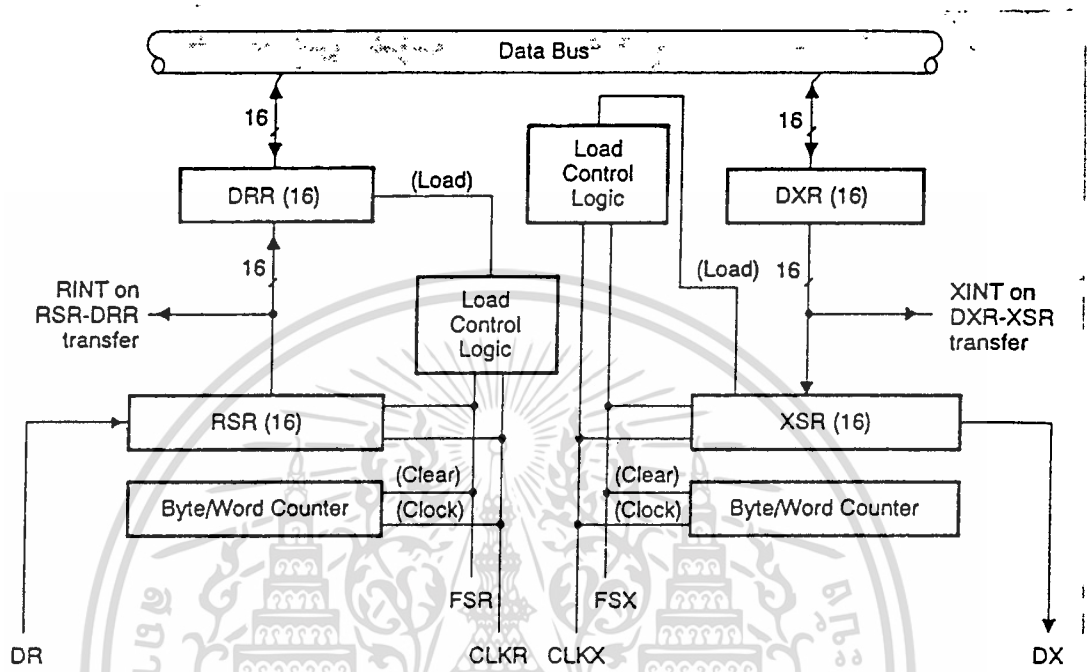
ขาสัญญาณ	ความหมาย
CLKX	สัญญาณคล็อกที่ใช้ในการส่ง
CLKR	สัญญาณคล็อกที่ใช้ในการรับ
DX	สัญญาณข้อมูลอนุกรมในการส่ง
DR	สัญญาณข้อมูลอนุกรมในการรับ
FSX	สัญญาณซิงโครไนซ์เฟรมการส่ง
FSR	สัญญาณซิงโครไนซ์เฟรมการรับ

ในการควบคุมพอร์ตอนุกรมจะมีรีจิสเตอร์ที่จำเป็นตามตารางที่ 2.7

ตารางที่ 2.7 รีจิสเตอร์พอร์ตอนุกรม

รีจิสเตอร์	ความหมาย
SPC	รีจิสเตอร์ควบคุมพอร์ตอนุกรม
DXR	รีจิสเตอร์ส่งข้อมูล
DRR	รีจิสเตอร์รับข้อมูล
XSR	รีจิสเตอร์เลื่อนการส่ง
RSR	รีจิสเตอร์เลื่อนการรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



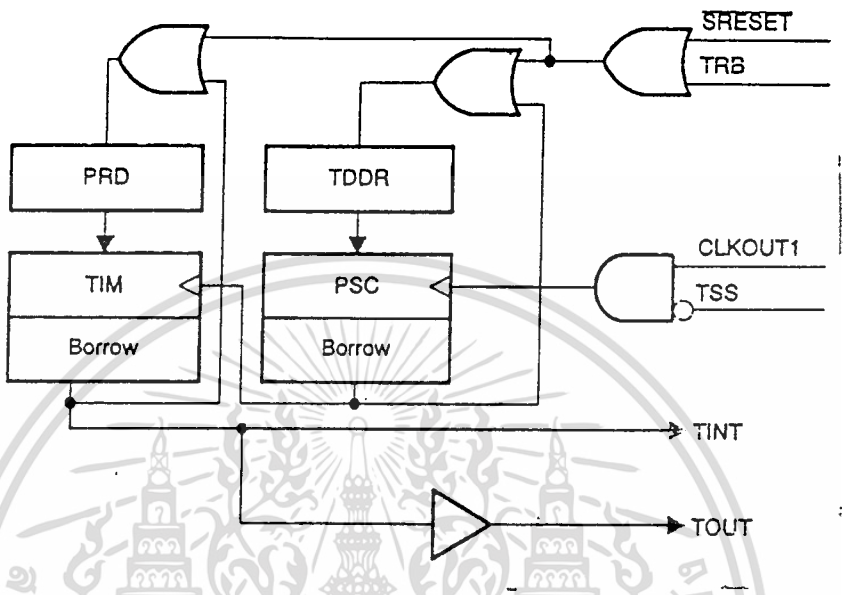
รูปที่ 2.5 บล็อกไดอะแกรมการทำงานของพอร์ตอนุกรม

ในการส่งข้อมูลทำได้โดยการเขียนข้อมูลลงใน DXR (Data Transmit Register) โดย XSR (Transmit Shift Register) จะเป็นตัวเลื่อนบิตตามสัญญาณนาฬิกาจนครบไปตามขา DX ส่วนการรับข้อมูลจะรับมาทางขา DR เข้ามาที่ RSR (Receive Shift Register) เมื่อครบแล้วจะทำการอินเทอร์รัพท์บอกให้ทราบ แล้วสามารถรับข้อมูลโดยการอ่านข้อมูลจาก DRR (Data Receive Register)

2.5.3 ไทม์เมอร์ (Timer)

ในการใช้ไทม์เมอร์ของ TMS320C50 ซึ่งจะสามารถตั้งคาบเวลาของสัญญาณนาฬิกาได้ โดยสัญญาณจะออกที่ขา Tout ของ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 บล็อกไดอะแกรมของไทม์เมอร์

ซึ่งการคำนวนคาบเวลา จะเป็นตามสูตร

$$T_{INT} = \frac{1}{t_c \times (TDDR + 1) \times (PRD + 1)} \qquad (2.1)$$

ซึ่ง t_c เป็นคาบเวลาของสัญญาณนาฬิกาที่จ่ายให้ที่ขา CLKOUT1

TDDR (Timer Divide Down Ratio)

PRD (Period Register) เป็นรีจีสเตอร์ที่สามารถใส่ค่าคาบเวลาได้

ในการทำงานจะต้องมีการใส่ค่าต่าง ๆ ลงบน TCR (Timer Control Register) ดังตารางที่ 2.8

ตารางที่ 2.8 รีจิสเตอร์ควบคุมไทม์เมอร์ (TCR)

15-12	11	10	9-6	5	4	3-0
Resersed	SOFT	FREE	PSC	TRB	TSS	TDDR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

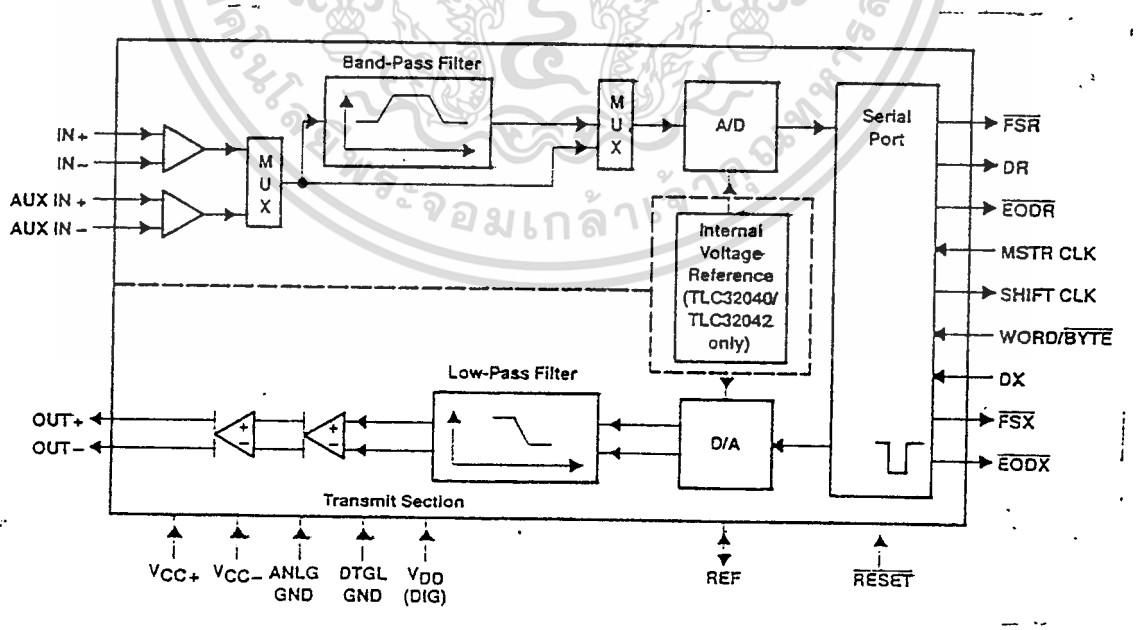
วงจรอินเทอร์เฟสสัญญาณอนาลอก TLC32040

TLC 320C40 เป็น ชิพซีมอส (CMOS Chip) ที่ใช้เชื่อมต่อกับ TMS320C50 ในการทำงาน DSP

3.1 ลักษณะสำคัญของ TLC320C40 มีดังนี้

- ใช้เทคโนโลยีการผลิต Advanced LinCMOS™ Silicon-Gate Process
- ความละเอียดของ ADC และ DAC เป็น 14 บิต
- สามารถเปลี่ยนอัตราแซมปลิ้งของ ADC และ DAC ได้ถึง 19,200 ครั้ง/วินาที
- มี Switched-Capacitor Antialiasing Input Filter และ Output-Reconstruction Filter
- มีพอร์ตอนุกรมสำหรับติดต่อโดยตรงกับ TMS3211, TMS320C17, TMS320C20 และ TMS320C25 DSP
- สามารถปรับอัตราการแปลงของ ADC และ DAC ทั้งแบบซิงโครนัส หรืออะซิงโครนัส โดยใช้ซอฟท์แวร์ควบคุม

3.2 ฟังก์ชันบล็อกไดอะแกรม (Function block Diagram)



รูปที่ 3.1 ฟังก์ชันบล็อกไดอะแกรมของ TLC32040

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ตำแหน่งขาและหน้าที่การทำงานของแต่ละขา

ตารางที่ 3.1 แสดงตำแหน่งขาและหน้าที่การทำงานของแต่ละขา

ชื่อ	หมายเลข	I/O	คำอธิบาย
ANLG GND	17,18		กราวด์อนาลอก (แยกกับกราวด์ดิจิตอล)
AUX IN+	24	I	นอน-อินเวอร์ติงออกซิลลารีอินพุต(Noninverting Auxilary Input)
AUX IN-	23	I	อินเวอร์ติงออกซิลลารีอินพุต (Inverting Auxilary Input)
DGTL GND	9		กราวด์ดิจิตอล
DR	5	O	ใช้สำหรับส่งเอาท์พุต ADC จาก AIC (Analog Interface Circuit) ไปยัง TMS320C50 ผ่านทางพอร์ตอนุกรม ซึ่งการส่งผ่านพอร์ตอนุกรมจะต้องซิงค์กับ Shift Clock
DX	12	I	ใช้รับอินพุต DAC หรือคำสั่งการควบคุมจาก TMS320C50 ซึ่งการส่งผ่านพอร์ตอนุกรมจะต้องซิงค์กับ Shift Clock
$\overline{EODR}$	3	O	สัญญาณหยุดรับข้อมูล (End of Data Receive) : ในการติดต่อผ่านพอร์ตอนุกรมในโหมดเวิร์ด (Word Mode) สัญญาณ /EODR จะอยู่ในสถานะต่ำทันทีเมื่อทั้ง 16 บิตของเอาท์พุต A/D ได้ถูกส่งจาก AIC ไปยัง TMS320C50 ซึ่งสามารถใช้สัญญาณนี้ในการอินเทอร์รัพท์ไมโครโปรเซสเซอร์ให้ทราบว่าสิ้นสุดการติดต่อแล้ว หรือใช้สโตรบและให้รีจิสเตอร์เลื่อนข้อมูลออก (Enable External Serial-to-parallel Shift Register) ก็ได้ แต่ถ้าเป็นโหมดไบต์ (Byte Mode) สัญญาณ $\overline{EODR}$ จะอยู่ในสถานะต่ำ หลังจากไบต์แรกได้ส่งไปยัง TMS320C50 แล้วและยังคงรักษาสถานะต่ำจนกระทั่งไบต์ที่สองได้ส่งไป ทั้งนี้ก็เพื่อให้รู้ว่าไบต์แรกหรือไบต์ที่สองออกไป
$\overline{EODX}$	11	O	สัญญาณหยุดส่งข้อมูล (End of Data Transmit) ก็คล้ายกับ $\overline{EODR}$ ซึ่งจะบอกให้ทราบว่าการติดต่อจาก TMS320C50 ไปยัง AIC นั้นเสร็จแล้วทั้งในโหมดเวิร์ด และโหมดไบต์ ก็คล้ายกับ $\overline{EODR}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ	หมายเลข	I/O	คำอธิบาย
$\overline{FSR}$	4	O	สัญญาณซิงค์การรับ (Frame Sync Receive): ในการติดต่อทางพอร์ตอนุกรม $\overline{FSR}$ จะอยู่สถานะต่ำตลอดการส่งจาก AIC ไปยัง TMS320C50 ซึ่งบิตแรกที่จะส่งจะต้องพร้อมอยู่ที่ขา DR ก่อน $\overline{FSR}$ จะอยู่ในสถานะต่ำ
$\overline{FSX}$	14	O	สัญญาณซิงค์การส่ง (Frame Sync Transmit): เมื่อสัญญาณนี้อยู่ในสถานะต่ำ พอร์ตอนุกรม TMS320C50 จะส่งบิตไปยัง AIC โดยมาที่ขา DX ในการติดต่ออนุกรมทุกโหมด $\overline{FSR}$ จะสถานะต่ำตลอดการส่ง
IN+	26	I	นอน-อินเวอร์ติง อินพุต (Noninverting Input)
IN-	25	I	อินเวอร์ติง อินพุต (Inverting Input)
MSTR CLK	6	I	สัญญาณนาฬิกามาสเตอร์ (Master Clock) จะใช้ในการควบคุมทุกส่วนภายใน AIC ไม่ว่าจะเป็นสัญญาณนาฬิกาเลื่อน (Shift Clock), สัญญาณนาฬิกาควบคุมฟิลเตอร์ (Switched-Capacitor Filter Clock), A/D และ D/A ไทม์มิ่ง
OUT+	22	O	นอน-อินเวอร์ติง เอาท์พุต (Noninverting Output)
OUT-	21	O	อินเวอร์ติง เอาท์พุต (Inverting Output)
REF	8	I/O	สำหรับ TLC32040 และ TLC32042 แรงดันอ้างอิงภายในจะถูกส่งออกมาที่ขานี้ แต่ถ้าเป็น TLC32040 และ TLC32042 แรงดันอ้างอิงจากภายนอกจะถูกต่อเข้าที่ขานี้
$\overline{RESET}$	2	I	รีเซตจะทำการตั้งค่า TA, TA′, TB, RA, RA′, RB และรีจิสเตอร์ควบคุม (Control Register) ให้เป็นค่าดีฟอลท์ (Default) ซึ่งจะบอกภายหลัง
SHIFT CLK	10	O	สัญญาณนาฬิกาเลื่อนเกิดจากการหารความถี่ของสัญญาณนาฬิกามาสเตอร์ด้วย 4 ซึ่งสัญญาณนี้จะใช้ในการติดต่อทางพอร์ตอนุกรม
VDD	7		ไฟเลี้ยงวงจรดิจิตอล (Digital Supply Voltage) 5V±5%
Vcc+	20		ไฟเลี้ยงวงจรอนาลอกด้านบวก (Positive Analog Supply Voltage) 5V±5%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ	หมายเลข	I/O	คำอธิบาย
Vcc -	19		ไฟเลี้ยงวงจรอนาลอกด้านลบ (Negative Analog Supply Voltage) -5V±5%
WORD/ $\overline{BYTE}$	13	O	ขานี้จะทำงานร่วมกับรีจิสเตอร์ควบคุมเพื่อใช้ในการเลือกโหมดการติดต่ออนุกรม ซึ่งมี 4 แบบดังนี้ <u>การติดต่อแบบ อะซิงโครนัส</u> <u>ในโหมดไบต์</u> (WORD/$\overline{BYTE}$ = สถานะต่ำ) พอร์ตอนุกรมจะติดต่อโดยตรงกับ TMS320C50 และจะติดต่อทีละ 8 บิต 2 ครั้ง ซึ่งมีขั้นตอนการทำงานดังนี้ 1. $\overline{FSX}$ หรือ $\overline{FSR}$ อยู่ในสถานะต่ำ 2. 8 บิตแรกถูกส่งออกไป หรือรับเข้ามา 3. $\overline{EODX}$ หรือ $\overline{EODR}$ อยู่ในสถานะต่ำ 4. $\overline{FSX}$ หรือ $\overline{FSR}$ อยู่ในสถานะสูงประมาณ 4 สัญญาณนาฬิกาเลื่อน แล้ว อยู่ในสถานะต่ำ 5. 8 บิตต่อมา(ของไบต์ที่สอง)ถูกส่งหรือรับเข้ามา 6. $\overline{EODX}$ หรือ $\overline{EODR}$ อยู่ในสถานะสูง 7. $\overline{FSX}$ หรือ $\overline{FSR}$ อยู่ในสถานะสูง <u>ในโหมดเวิร์ด</u> พอร์ตอนุกรมจะต่อตรงกับพอร์ตอนุกรมของ TMS320C50 และมีการติดต่อครั้งเดียว 16 บิต ซึ่งมีขั้นตอนดังนี้ 1. $\overline{FSX}$ หรือ $\overline{FSR}$ อยู่ในสถานะต่ำ 2. 16 บิตถูกส่งหรือรับเข้ามา 3. $\overline{FSX}$ หรือ $\overline{FSR}$ อยู่ในสถานะสูง 4.$\overline{EODX}$ หรือ $\overline{EODR}$ อยู่ในสถานะต่ำ <u>การติดต่อแบบซิงโครนัส</u> ในกรณีนี้ แบนด์พาส ฟิลเตอร์ (bandpass filter) และ อัตราการแปลง A/D (A/D conversion timing) จะถูกกำหนดจาก TX เคาน์เตอร์ A, TX เคาน์เตอร์ B และ TA, TA′,และ TB แทนส่วนในการติดต่อทั้งในแบบโหมดเวิร์ดและโหมดไบต์นั้นมีขั้นตอนเหมือนกับการติดต่อแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของ TLC32040

อินพุตทางอนาลอก (Analog Input)

สำหรับอนาลอกอินพุตจะมี 2 กลุ่มคือ IN+, IN- และ AUX IN+, AUX IN- ซึ่งสามารถเลือกใช้กลุ่มใดกลุ่มหนึ่งโดยจะใช้ในแบบ ดิฟเฟอเรนเชียล (Differential) หรือซิงเกิล-เอนด์ (Single-ended) และค่าเกนสำหรับอินพุต IN+, IN-, AUX IN+ และ AUX IN- สามารถจะใช้โปรแกรมตั้งค่าได้ (มี 3 ค่าคือ 1, 2, หรือ 4) การเลือกใช้กลุ่มอินพุตใดจะเลือกโดยใช้ซอฟท์แวร์ควบคุม

A/D Bandpass Filter, A/D Bandpass Filter Clocking และ A/D Conversion Timing

สำหรับ A/D แบนด์พาส ฟิลเตอร์ (A/D Bandpass Filter) เราสามารถที่จะเลือกใช้หรือไม่ใช้ก็ได้โดยใช้ซอฟท์แวร์ควบคุม ความถี่ของสัญญาณนาฬิกาควบคุมฟิลเตอร์ (Filter Clock) จะเป็นตัวกำหนดทรานเฟอร์ฟังก์ชัน(Transfer Function) ของฟิลเตอร์โดยจะคิดอัตราส่วนจากความถี่สัญญาณนาฬิกาควบคุมฟิลเตอร์ 28kHz ที่ความถี่ต่ำที่เริ่มมีลักษณะเป็นความถี่สูงผ่านจะมีความถี่เป็น 300 Hz อัตราการแปลง D/A ก็หาได้จากความถี่ที่หาร 228 kHz ด้วย RX เคาน์เตอร์ B

เอาท์พุตทางอนาลอก (Analog Output)

อนาลอกเอาท์พุตจะมีเพาเวอร์แอมปลิไฟล์ (Power Amplifier) มีเอาท์พุตทั้งแบบนอน-อินเวอร์ติง (Noninverting) และแบบอินเวอร์ติง (Inverting) เนื่องจากมีแอมปลิไฟล์ทำให้เอาท์พุต สามารถขับ ทรานฟอร์เมอร์ชนิดไฮ-บริด (Transformer Hybrid) หรือโหลดอิมพีแดนซ์ต่ำได้โดยใช้ทั้งแบบดิฟเฟอเรนเชียล หรือซิงเกิล-เอนด์

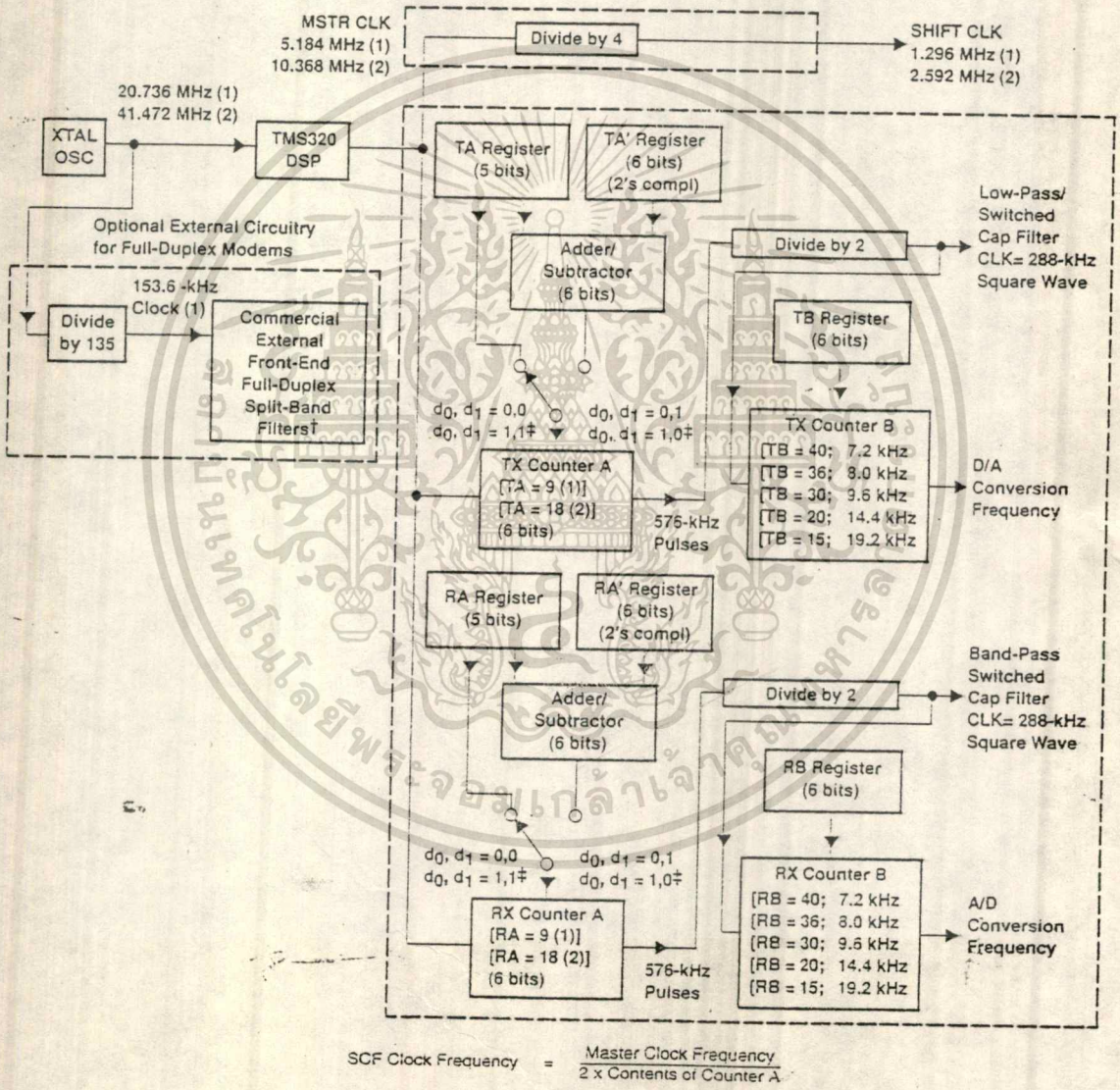
วงจรกรองความถี่ต่ำของ D/A ,วงจรกรองความถี่แบบแบนด์พาสของ D/A ,สัญญาณนาฬิกาควบคุมวงจรกรองความถี่ต่ำ และอัตราการแปลงสัญญาณดิจิตอลเป็นอนาลอก (D/A Lowpass Filter, D/A Lowpass Filter Clocking และ D/A Conversion Timing)

เช่นเดียวกับ A/D ฟิลเตอร์โดยทรานเฟอร์ฟังก์ชันของฟิลเตอร์ถูกกำหนดจากอัตราส่วนกับความถี่ 228 kHz และอัตราการแปลง D/A ก็หาได้จากความถี่ 228kHz หารด้วย TX เคาน์เตอร์ B

การต่อกลับ (Loopback)

การต่อกลับจะให้ผู้ใช้ตรวจสอบวงจรโดย OUT+ และ OUT- จะต่อเข้าภายในกับ IN+ และ IN- ดังนั้นบิต DAC(D15-D2) จะถูกส่งไปยังขา DX และเปรียบเทียบกับบิต ADC ที่รับมาจากขา DR ซึ่งโดยปกติจะต้องมีค่าเท่ากัน (ในทางปฏิบัติอาจไม่เท่ากันก็ได้) ในการตรวจสอบ ถ้าใช้ขา IN+ และ IN- สัญญาณภายนอกที่ต่อกับ IN+ , IN- จะไม่มีผลแต่ถ้าใช้ AUX IN+, AUX IN-สัญญาณภายนอกจะถูกรวมกับ OUT+ และ OUT- สำหรับการควบคุมการต่อกลับจะทำโดยการตั้งค่าในรีจิสเตอร์ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 อินเทอแนลไทม์มิ่งคอนฟิกกูเรชั่น (Internal Timing Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของบิตใน AIC DR หรือ DX Word

A/D or D/A MSB

1st bit sent ↓ 1st bit sent of 2nd byte ↓ A/D or D/A LSB ↓

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

รูปที่ 3.3 แสดงรูปแบบบิตใน AIC DR หรือ DX Word

รูปแบบของบิตใน AIC DX (AIC DX Data Word Format Section)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

รูปที่ 3.4 แสดงรูปแบบการ formatใน AIC DX Data Word

ในรูปแบบการส่งครั้งแรก (Primary DX Serial Communication Protocal (D0-D15)) นั้น เราจะใช้ D1 และ D0 เป็นตัวเลือก ส่วน D2-D15 จะถูกส่งไปยังรีจิสเตอร์ควบคุมการแปลง D/A (D/A Converter Register) ซึ่งจะแบ่งการทำงานออกเป็นกรณีต่าง ๆ ได้ดังนี้คือ

ตารางที่ 3.2 แสดงการกำหนดค่าใน D1 และ D0 ในการทำงานในโหมดต่างๆของ AIC

D1	D0	การทำงาน
0	0	จะทำการโหลดค่าในรีจิสเตอร์ TA และ RA ไปยัง TX และ RX เคาน์เตอร์ A ตามลำดับ และทำการโหลด TX และ RX เคาน์เตอร์ B ด้วยค่าในรีจิสเตอร์ TB และ RB
0	1	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่า TA+TA′ และ RA+RA′ แต่จะทำการโหลด TX และ RX เคาน์เตอร์ B ด้วยค่าในรีจิสเตอร์ TB และ RB
1	0	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่า TA-TA′ และ RA-RA′ แต่จะทำการโหลด TX และ RX เคาน์เตอร์ B ด้วยค่าในรีจิสเตอร์ TB และ RB
1	1	TX และ RX เคาน์เตอร์ A จะถูกโหลดด้วยค่าในรีจิสเตอร์ TA และ RA ส่วน TX และ RX เคาน์เตอร์ B จะถูกโหลดด้วยค่าในรีจิสเตอร์ TB และ RB หลังจากนั้นก็จะรอเวลาประมาณ 4 สัญญาณนาฬิกาเลื่อนก็จะทำการส่งครั้งที่สอง (Secondary Communication) เพื่อไปโปรแกรม AIC ให้ทำงานตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของบิตในการส่งครั้งที่สอง (Secondary DX Serial Communication protocol)

ตารางที่ 3-3 แสดง Secondary DX Serial Communication Protocol

XX ←to TA register→ X X ←to RA register→ 0 0	D13 และ D6 เป็น MSR (เป็นบวก)
X ←to TA′register→ X ← to RA′register → 0 1	D14 และ D7 เป็นบิตแบบ 2's
X ←to TB register→ X ← to RB register → 1 0	D14 และ D7 เป็น MSB (เป็นบวก)
X X X X X X X X D7 D6 D5 D4 D3 D2 1 1 Control ← Register →	D2 = 0/1 ลด/เพิ่ม Bandpass Filter D3 = 0/1 Disable/Enable ฟังก์ชัน Loopback D4 = 0/1 Disable/Enable ขา AUX IN+ และ ขา AUX IN- D5 = 0/1 บิตกำหนดการรับส่งแบบ Asynchronous / Synchronous D6, D7 เป็นบิตควบคุม Gain

ฟังก์ชันรีเซต (Reset)

ฟังก์ชันรีเซตจะเป็นการตั้งค่าอัตราการเปลี่ยนแปลง A/D และ D/A ให้เป็น 8 kHz ถ้าใช้สัญญาณนาฬิกามาสเตอร์ 5.184 Mhz โดยค่าในรีจิสเตอร์จะถูกตั้งค่าเริ่มต้นเป็น

TA = RA = 9 , TA′= RA′= 1 , TB = RB = 24 และบิตในรีจิสเตอร์ควบคุมเป็น D7=1, D6=1, D5=1, D4=0, D3=0, D2=1

ข้อบังคับในการกำหนดค่าในรีจิสเตอร์ของ AIC

1. รีจิสเตอร์ TA ในโหมดเวิร์ด ต้องมีค่า ≥ 4
2. รีจิสเตอร์ TA ในโหมดไบต์ ต้องมีค่า ≥ 5
3. รีจิสเตอร์ TA′สามารถเป็นได้ทั้ง ค่าบวก, ลบ หรือ ศูนย์
4. รีจิสเตอร์ RA ในโหมดเวิร์ด ต้องมีค่า ≥ 4
5. รีจิสเตอร์ RA ในโหมดไบต์ ต้องมีค่า ≥ 5
6. รีจิสเตอร์ RA′สามารถมีค่าเป็นได้ทั้ง ค่าบวก, ลบ หรือ ศูนย์
7. (TA±TA′) จะต้องมีค่า > 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. (RA±RA′) จะต้องมีค่า > 1

9. TB จะต้องมีค่ามากกว่า 1

ถ้าไม่เป็นไปตามเงื่อนไขทั้ง 9 ข้อที่กล่าวมา AIC จะทำงานตามเงื่อนไขดังแสดงในตารางที่ 3.4

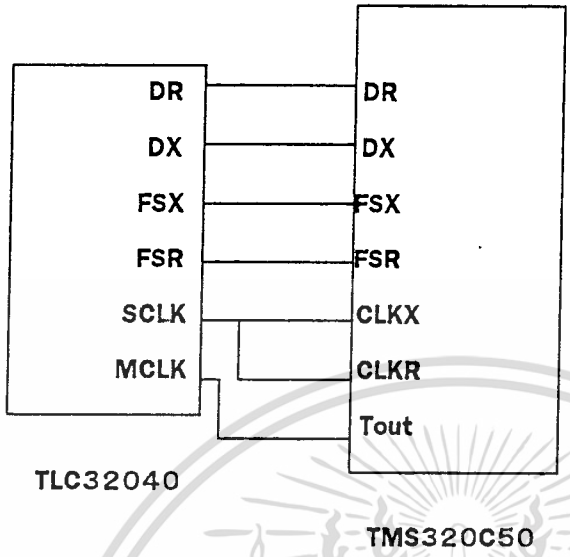
ตารางที่ 3.4

เงื่อนไข	การตอบสนองของ AIC
1. ถ้า TA+TA′= 0 หรือ 1 หรือ TA-TA′= 0 หรือ 1	จะทำการโหลดค่าในรีจิสเตอร์ TA ลงใน TX เคาน์เตอร์ A แทน
2. ถ้า TA+TA′ < 0	MODULO 64 จะถูกใช้เพื่อให้แน่ใจว่า ค่าที่เป็นบวกถูกโหลดไปไว้ที่ TX เคาน์เตอร์ A แทน เช่น TA+TA′+ 40h โหลดไปไว้ที่ TX เคาน์เตอร์ A
3. ถ้า RA+RA′ = 0 หรือ 1 RA-RA′ = 0 หรือ 1	จะทำการโหลดค่าในรีจิสเตอร์ RA ลงใน RX เคาน์เตอร์ A
4. ถ้า RA+RA′ = 0 หรือ 1	MODULO 64 ถูกใช้เหมือนข้อ 2 เพียงแต่โหลดไปที่ RX เคาน์เตอร์ A
5. ถ้า TA = 0 หรือ 1 หรือ RA = 0 หรือ 1	AIC หยุดทำงาน
6. ถ้า TA < 4 ในโหมดเวิร์ด TA < 5 ในโหมดไบต์ RA < 4 ในโหมดเวิร์ด RA < 5 ในโหมดไบต์	การติดต่อทางพอร์ตอนุกรมของ AIC ไม่ทำงาน
7. ถ้า TB = 0 หรือ 1	โหลดค่า TB ใหม่ ด้วยค่า 24h
8. ถ้า RB = 0 หรือ 1	โหลดค่า RB ใหม่ด้วยค่า 24h
9. ถ้า AIC และ DSP ไม่สามารถติดต่อกันได้	จะเก็บค่า DAC Output ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การอินเทอร์เฟสระหว่าง TMS320C50 และ TLC32040

การใช้อินเทอร์รัพท์ของพอร์ตอนุกรม



รูปที่ 3.5 แสดงการเชื่อมต่อระหว่าง TLC32040 กับ TMS320C50

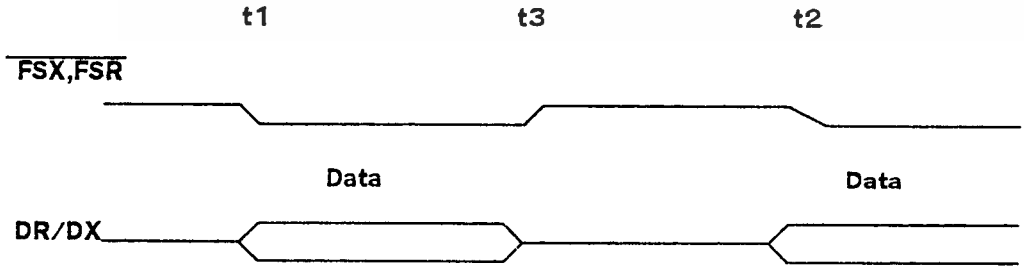
TMS320C50

การส่ง โดยการเขียน DXR จะเป็นการขออินเทอร์รัพต์ของ XINT (โดยไม่มีการมาร์ค)

การรับ เมื่อ RSR ---> DRR เต็มจะเกิดอินเทอร์รัพต์ของ RINT (โดยไม่มีการมาร์ค)

TLC32040

เนื่องจากการส่งระหว่าง TLC32040 กับ TMS32050 เป็นแบบซิงโครนัส TLC32040 จะส่งและรับทุก ๆ เฟรมซิงโครนัส (Frame Synchronous)



รูปที่3.6 แสดงช่วงเวลาในการส่งและรับข้อมูลระหว่าง TLC32040 กับ TMS320C50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากความถี่แซมปลิ้งสูงสุดเท่ากับ 19.2 Khz เพราะฉะนั้นข้อมูลต้องมาก่อน (ภายใน t2 - t1) t2-t1 ต้องไม่น้อยกว่า 52.08 ไมโครวินาที ข้อมูลที่มาจาก TLC32040 แล้วทำการประมวลผลทางคณิตศาสตร์แล้วส่งกลับมาที่ TLC32040 อีกครั้ง ต้องใช้เวลาน้อยกว่า 1/Fs จึงจะสามารถได้สัญญาณที่มีความถี่เข้าเท่ากับความถี่ออก เนื่องจาก t3-t1 ช่องของข้อมูล (16บิต) ใช้เวลาประมาณ (1/Shift Clock) x 16 ประมาณเท่ากับ 6.17 ไมโครวินาที เพราะฉะนั้น 6.17 ไมโครวินาที เป็นเวลาที่ใช้ในการส่ง 1 ข้อมูล จะต้องใช้เวลาเหลือประมวลผล (52.08-6.17) เท่ากับ 45.91 ไมโครวินาที

หรือประมาณ (45.91ไมโครวินาที)/(50นาโนวินาทีต่อ 1 คำสั่ง) เท่ากับ 918 คำสั่ง

สรุป สามารถเขียนโปรแกรมถึงกว่า 900 คำสั่ง เพื่อประมวลผลข้อมูล 1 ข้อมูลก่อนที่จะเกิดการอินเทอร์รัพท์อีกครั้ง แต่เราใช้ได้ถึงเพราะ Fs มักจะน้อยกว่า 19.2 Khz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

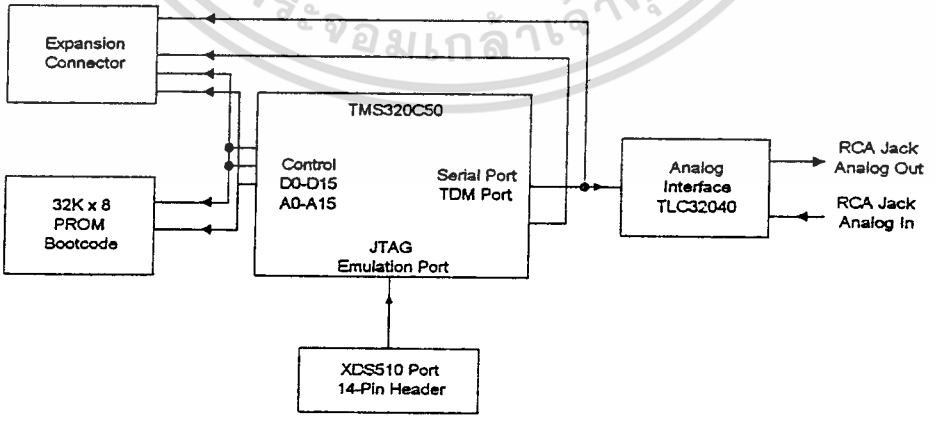
โครงสร้างของ TMS320C50 DSP Starter Kit

4.1 ลักษณะโดยทั่วไปของบอร์ด

- ใช้โปรเซสเซอร์เบอร์ TMS320C50 ซึ่งใช้งาน DSP โดยเฉพาะ
- คำสั่งใช้เวลาประมาณ 50 ns (Instruction Cycle Time)
- PROM 32 กิโลไบท์
- สามารถใช้ในช่วงความถี่เสียง โดยผ่านทาง TLC32040
- ใช้มาตรฐาน RCA คอนเนกเตอร์ สำหรับ อนาลอกอินพุท และเอาท์พุท เพื่อให้สามารถต่อเข้ากับไมโครโฟน และลำโพงได้โดยตรง
- ในการติดต่อกับการควบคุม มีคอนเนกเตอร์ XDS510 ติดต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม
- สามารถขยาย I/O บัสได้เพื่อสำหรับการออกแบบภายนอก

รูปที่ 4.1 แสดงบล็อกไดอะแกรมของบอร์ดซึ่งประกอบไปด้วย โฮสอินเทอร์เฟส, อนาลอกอินเทอร์เฟส และอีมูเลชั่นพอร์ต ทำให้สามารถติดต่อกับพีซีได้โดยผ่านทาง RS232 นอกจากนี้ยังมี PROM ขนาด 32 กิโลไบต์ ที่ใช้เก็บเคอร์แนลโปรแกรมไว้สำหรับการบูต

ในบอร์ดยังมี TLC32040 วงจรอินเทอร์เฟสสัญญาณอนาลอก (Analog Interface Circuit : AIC) ที่มี RCA คอนเน็กเตอร์ 2 ตัวสำหรับอินพุต และเอาท์พุต



รูปที่ 4.1 บล็อกไดอะแกรมของ TMS320C5x DSK

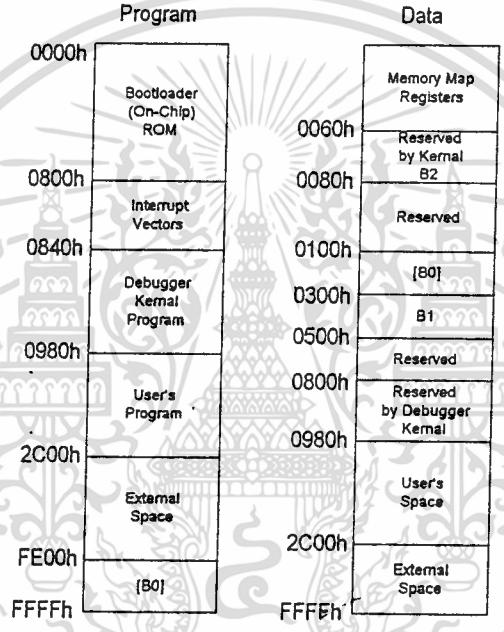
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำใน TMS320C50 DSK นั้น จะแบ่งออกเป็นหน่วยความจำสำหรับโปรแกรม และ หน่วยความจำสำหรับข้อมูลซึ่งในแต่ละแอดเดรสก็จะถูกใช้งานต่าง ๆ ตามแสดงในรูปที่ 4.2 และผู้ใช้สามารถใช้หน่วยความจำภายในได้ขนาด 10 K โดยไม่จำเป็นต้องต่อภายนอก

TLC32040 บนบอร์ดจะมีลักษณะดังนี้

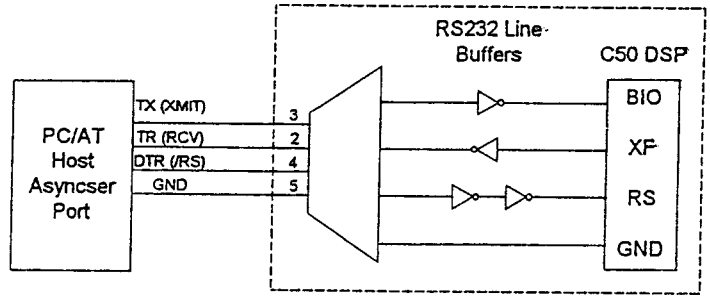
- ใช้ในการแปลง A/D และ D/A ขนาด 14 บิต
- สามารถเปลี่ยนอัตราการแซมปลิ้งทั้งของ A/D, D/A และความถี่ฟิลเตอร์ได้

การติดต่อระหว่าง AIC และ TMS320C50 จะผ่านทางพอร์ตอนุกรมภายในบอร์ด



รูปที่ 4.2 หน่วยความจำภายใน TMS320C50 DSK

การต่อใช้งาน TMS320C50 DSK กับ PC จะใช้ขา XF และ BIO\ ที่ต่อผ่านพอร์ตอนุกรม RS232 ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 การต่อระหว่าง DSK และ PC โดยผ่านทาง RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

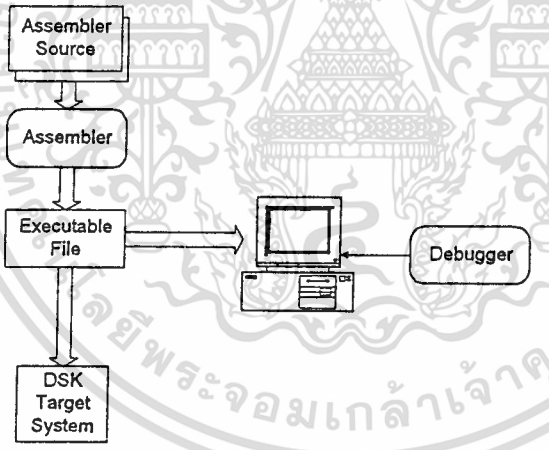
นอกจากนี้ DSK ยังมีแอสเซมเบอร์ และดีบักเกอร์ของมันเอง ซึ่งจะมีประโยชน์มากเนื่องจากภาษาแอสเซมบี้ที่ใช้ใน DSK นั้นจะสนับสนุนการทำงานด้านประมวลผลสัญญาณ (Signal Processing) และดีบักเกอร์ก็มีความสามารถที่จะทำงานในแบบทีละขั้น (Single-step) และเบรคพอยท์เพื่อช่วยในการแก้ไขโปรแกรม

4.2 การสร้างโปรแกรมเพื่อใช้กับ DSK

1. ทำการสร้างไฟล์หลัก (Source File) สำหรับโปรแกรม เช่น EXAM.ASM
2. แปลงไฟล์หลัก โดยใช้ DSK แอสเซมเบอร์ ซึ่งมีคำสั่งดังนี้

   DSK5A EXAM.ASM

3. หากต้องการตรวจแก้ไขโปรแกรมโดยใช้ดีบักเกอร์ก็ทำได้โดยใช้โปรแกรม DSK5D.EXE ซึ่งได้ให้มากับบอร์ดแล้ว



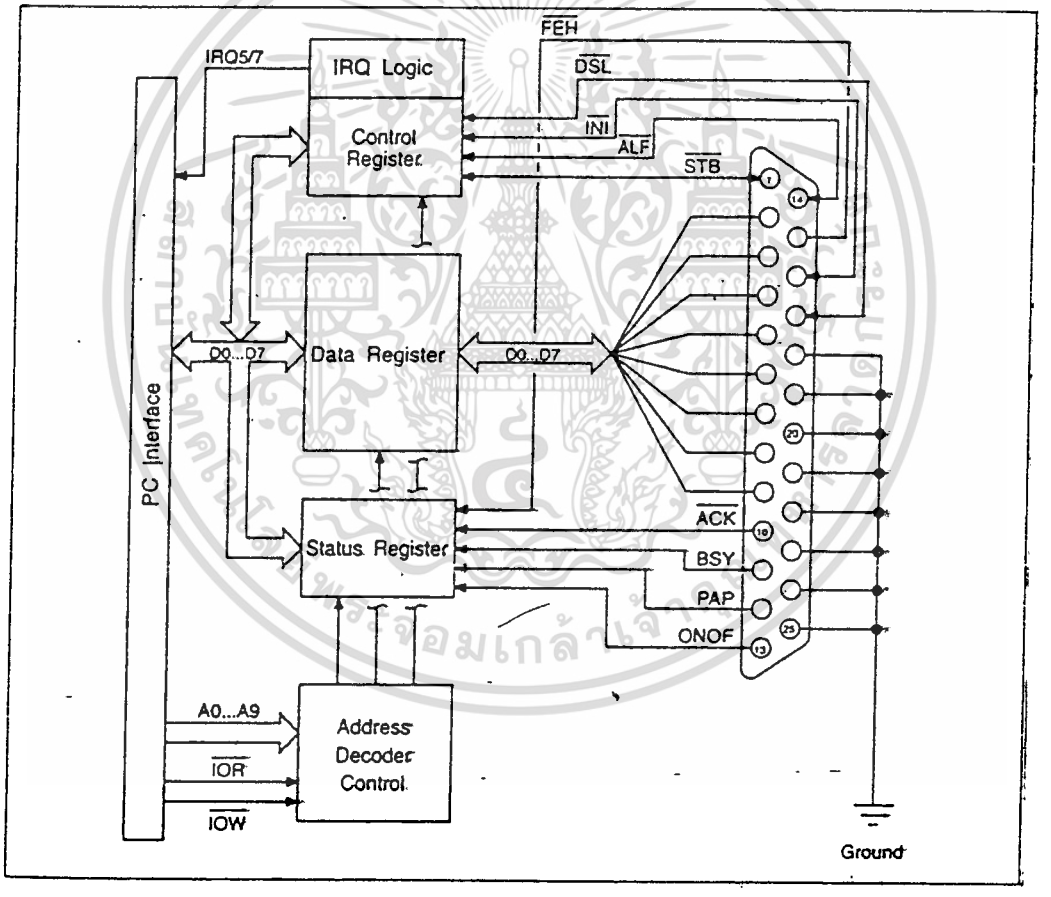
รูปที่ 4.4 แสดงขั้นตอนการสร้างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การเชื่อมต่อแบบขนาน

5.1 โครงสร้าง หน้าที่ และการเชื่อมต่อแบบขนาน

โดยปกติคอมพิวเตอร์ใช้การอินเทอร์เฟสแบบขนาน (Parallel Interface) เพื่อส่งข้อมูลออกไปเครื่องพิมพ์ การอินเทอร์เฟสแบบขนาน มีโครงสร้างอย่างง่ายดังต่อไปนี้



รูปที่ 5.1 แผนภาพแสดงการอินเทอร์เฟสแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับระบบบัสของคอมพิวเตอร์นั้นเป็นการอินเทอร์เฟสของพีซี (PC) ทั่วไป ซึ่งการ์ดที่ใช้ในการเชื่อมต่อนั้นมีดาต้าบัส (Databus) ขนาด 8 บิตเท่านั้น แต่ก็สามารถใช้งานได้ในคอมพิวเตอร์แบบ ISA หรือแบบ EISA ที่ขนาดความกว้างเท่ากับ 8 บิต

การอินเทอร์เฟสเป็นการส่งข้อมูลที่มีขนาด 8 บิตออกไปยังเครื่องพิมพ์และรีจิสเตอร์ภายในที่ใช้ในการอินเทอร์เฟสมีขนาด 8 บิต สัญญาณที่ใช้ในการอินเทอร์เฟส คือ IOR และ IOW โดยถ้าสัญญาณใดสัญญาณหนึ่งได้รับการติดต่อและมีการเลือกรีจิสเตอร์ ตัวถอดรหัสตำแหน่งจะถอดรหัสตำแหน่งโดยใช้บิตที่ A0 - A9 ซึ่งมีส่วนควบคุมทำการควบคุมการรับค่าเข้าและการส่งค่าออกของรีจิสเตอร์

การอินเทอร์เฟสแบบขนาน ประกอบด้วยส่วนต่าง ๆ ดังต่อไปนี้

1. รีจิสเตอร์ข้อมูล (Data Register)
2. รีจิสเตอร์สถานะ (Status Register)
3. รีจิสเตอร์ควบคุม (Control Register)
4. ส่วนควบคุมการถอดรหัสตำแหน่ง (Addres Decoder Register)
5. ลอจิก IRQ (IRQ Logic)

รีจิสเตอร์ข้อมูล (Data Register) มีขนาด 8 บิต ซึ่งซีพียูสามารถทำการอ่านหรือเขียนได้ เรียกว่าเป็นการติดต่อแบบสองทิศทาง (Bidirectional) โดยซีพียูจะทำการส่งผ่านข้อมูลที่มีขนาดหลายไบต์จากรีจิสเตอร์ข้อมูลไปยังเครื่องพิมพ์ (Printer) ถ้าโปรเซสเซอร์อ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลแล้วจะเกิดการ OR กันระหว่างค่าของไบต์ที่ถูกเขียนครั้งสุดท้ายกับค่าของสัญญาณที่เข้ามาอินเทอร์เฟสที่ขา 2-9 ซึ่งค่าที่ตอบกลับมาคือค่าที่ได้จากการ OR แล้ว โดยปกติเครื่องพิมพ์จะไม่ทำงานถ้าสายที่ 2-9 อยู่ในสภาวะสูงจนกระทั่งได้รับคำสั่ง IN ก็จะให้ค่าที่ถูกเขียนครั้งสุดท้ายกลับมา ในสถานการณ์การเปลี่ยนแปลงที่กล่าวมานี้สายที่เชื่อมต่อกับอุปกรณ์ภายนอกจะอยู่ในสภาวะสูง การแลกเปลี่ยนข้อมูลที่เกิดขึ้นนั้นใช้การอินเทอร์เฟสแบบขนาน โดยถ้าเราสามารถเชื่อมต่อกับเครื่องพิมพ์ได้ก็จะสามารถโหลดค่าข้อมูลจากรีจิสเตอร์ข้อมูลได้

รีจิสแตอร์สถานะ (Status Register) จะแสดงค่าสถานะปัจจุบันขณะเชื่อมต่อกับเครื่องพิมพ์ ซึ่งซีพียูสามารถอ่านค่าจากรีจิสเตอร์สถานะได้เพียงอย่างเดียวไม่สามารถที่จะเขียนค่าลงในรีจิสเตอร์สถานะได้ เรียกว่าเป็นการติดต่อแบบทิศทางเดียว (Unidirectional)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่5.1 การเชื่อมต่อกับเครื่องพิมพ์ใช้สายเคเบิลแบบเซนทรอนิกส์(Centronic Register)

แบบ 25 ขา	แบบ 36 ขา	ชื่อสัญญาณ	ความหมาย
1	1	/STR	สภาวะต่ำ มีการส่งข้อมูลไปยังเครื่องพิมพ์
2	2	DO	บิตข้อมูลที่ 0
3	3	D1	บิตข้อมูลที่ 1
4	4	D2	บิตข้อมูลที่ 2
5	5	D3	บิตข้อมูลที่ 3
6	6	D4	บิตข้อมูลที่ 4
7	7	D5	บิตข้อมูลที่ 5
8	8	D6	บิตข้อมูลที่ 6
9	9	D7	บิตข้อมูลที่ 7
10	10	/ACK	สภาวะต่ำ แสดงว่า เครื่องพิมพ์ได้รับหนึ่งตัวอักษร เรียบร้อยแล้วและสามารถที่จะรับอักษรตัวต่อไป
11	11	BSY	สภาวะสูง แสดงว่า -เครื่องพิมพ์ได้รับตัวอักษรเรียบร้อยแล้ว -บัฟเฟอร์เครื่องพิมพ์เต็ม -เครื่องพิมพ์อยู่ในระหว่างพิมพ์ข้อมูล -เครื่องพิมพ์อยู่ในระหว่างสถานะไม่ทำงาน -เครื่องพิมพ์เกิดความผิดพลาด (Error)
12	12	PAP	สภาวะสูง แสดงว่า ขาดกระดาษ
13	13	OFON	สภาวะสูง แสดงว่า เครื่องพิมพ์อยู่ในระหว่างสถานะทำงาน (Online State)
14	14	/ALF	สภาวะต่ำ แสดงว่า เครื่องพิมพ์ป้อนกระดาษอัตโนมัติ(Auto Line Feed)
15	32	/FEH	สภาวะต่ำ แสดงว่า -เครื่องพิมพ์อยู่ในระหว่างไม่ทำงาน -เครื่องพิมพ์เกิดความผิดพลาด
16	31	INI	สภาวะต่ำ แสดงว่า เครื่องพิมพ์อยู่ในสภาวะเริ่มพิมพ์
17	36	/DSL	สภาวะต่ำ แสดงว่า เป็นการเลือกเครื่องพิมพ์
18-25	19-30,33	กราวด์	

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ควบคุม (Control Register) หน้าที่ควบคุมการเชื่อมต่อเครื่องพิมพ์, สัญญาณควบคุม และการอินเทอร์เฟสซึ่งซีพียูสามารถที่จะอ่านค่าหรือเขียนค่าในรีจิสเตอร์ควบคุมได้ รีจิสเตอร์ควบคุมจะควบคุมการส่งข้อมูลไปยังเครื่องพิมพ์

ลอจิก IRQ (IRQ Logic) จะแสดงความพร้อมในการที่จะยอมรับตัวอักษรตัวต่อไปเมื่อเกิดการอินเทอร์เฟสขึ้น

เครื่องพิมพ์ส่วนใหญ่จะต่อสัญญาณ DSL ลงกราวด์ ซึ่งเป็นการเลือกเครื่องพิมพ์ในระบบเก่า ซีพียูสามารถใช้สัญญาณเฉพาะที่จะเลือกหรือไม่เลือกเครื่องพิมพ์ เราสามารถส่งข้อมูลขนาด 8 บิตที่ขา D0-D7 ไปยังเครื่องพิมพ์ได้ และสามารถที่จะรับข้อมูลขนาด 8 บิตที่ขา D0-D7 ได้ด้วย สายสัญญาณ STR, ALF และ INI ของเครื่องพิมพ์ถูกควบคุมโดยซีพียูและการอินเทอร์เฟส การที่สัญญาณ STR เปลี่ยนสภาวะจากสูงเป็นต่ำนั้น เครื่องพิมพ์จะรับข้อมูลจากขา D0-D7 ขณะที่มีการเปลี่ยนข้อมูล ส่วนความหมายของสัญญาณอื่น ๆ สามารถดูได้จากที่กล่าวมาข้างต้น

การอินเทอร์เฟสแบบขนาน มีรีจิสเตอร์ 3 ตัวที่สามารถส่งข้อมูลควบคุมเครื่องพิมพ์ในการอินเทอร์เฟสได้ ตำแหน่งพื้นฐานของรีจิสเตอร์สำหรับการอินเทอร์เฟสทั้งหมดนั้น (LPT1-LPT4) ถูกเก็บไว้ในพื้นที่ส่วนข้อมูลของไบออส (Bios)

รีจิสเตอร์ข้อมูลจะมีค่าออฟเซท (OFFSET) เท่ากับ 00H (ดังตารางที่ 5.2) ส่วนรีจิสเตอร์สถานะมีค่าออฟเซทเท่ากับ 01H (ดังตารางที่ 5.3) และรีจิสเตอร์ควบคุมมีค่าออฟเซทเท่ากับ 02H (ดังตารางที่ 5.4) ค่าตำแหน่งพื้นฐานของ LPT 1 เท่ากับ 378H ส่วนค่าตำแหน่งพื้นฐานของ LPT2 มีค่าเท่ากับ 278H ส่วนการอินเทอร์เฟสแบบขนานของการ์ดอะแดปเตอร์ (Adapter) ของ เฮอร์คิวส์ลิส (Hercules) หรือโมโนโครม (Monochrome) จะมีค่าตำแหน่งพื้นฐานเริ่มต้นที่ 3BCH

ในขณะที่เครื่องเริ่มทำการบูต (Booting) ไบออสทำการตรวจสอบตำแหน่งอินพุตและเอาท์พุตตามลำดับ ดังนี้ คือ จะตรวจสอบที่ตำแหน่ง 3BCH, 378H ,278H และ 2BCH แต่บางรุ่นของไบออสจะมีความแตกต่างออกไป คือ จะสนับสนุนการอินเทอร์เฟสแบบขนานเพียงสองเท่านั้น ซึ่งคอนฟิกกูเรชั่นจะทำการตรวจสอบโดยอินเทอร์รัพท์ไบออส INT 11H ซึ่งจะคืนค่าติดตั้งการอินเทอร์เฟสเพียงสองเท่านั้น แม้ว่าจะมีการติดตั้งถึงสี่ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 รีจิสเตอร์ข้อมูล (ออฟเซท 00H)

บิตที่	7	6	5	4	3	2	1	0
ชื่อสัญญาณ	D7	D6	D5	D4	D3	D2	D1	D0
หมายเลขขาสัญญาณ	9	8	7	6	5	4	3	2

รีจิสเตอร์ข้อมูลมีความกว้างขนาด 8 บิต และมีการจัดส่งข้อมูลอย่างเหมาะสม เราสามารถทำการอ่านและเขียนรีจิสเตอร์นี้ อย่างง่าย ๆ โดยการใช้คำสั่งอิน (IN) หรือเอาท์ (OUT)

ตารางที่ 5.3 รีจิสเตอร์สถานะ (ออฟเซท 01H)

บิตที่	7	6	5	4	3	2	1	0
ชื่อสัญญาณ	/BSY	/ACK	PAP	OFON	/FEH	X	X	X
หมายเลขขาสัญญาณ	11	10	12	13	15	-	-	-

/BSY : Busy

1 = เครื่องพิมพ์ว่าง 0 = เครื่องพิมพ์ไม่ว่าง , ออฟไลน์ (OFF - LINE) หรือเกิดข้อผิดพลาด

/ACK : Acknowledge

1 = อยู่ในกระบวนการถ่ายข้อมูล 0 = ข้อมูลส่งไปยังเครื่องพิมพ์เรียบร้อย

PAP : Paper

1 = กระดาษหมด 0 = กระดาษมีอยู่

OFON : Off-Line /On-Line

/FEH : เครื่องพิมพ์ผิดพลาด

1 = ไม่ผิดพลาด 0 = เครื่องพิมพ์ผิดพลาด

X : ไม่ใช้ (โดยทั่วไปจะเท่ากับ 1)

รีจิสเตอร์สถานะ สามารถอ่านค่าได้อย่างเดียว และจะบอกค่าข่าวสารการควบคุมจากเครื่องพิมพ์ เมื่อทำการเคลียร์บิต /BSY เครื่องพิมพ์จะอยู่ในสถานะ /BSY ซึ่งแสดงว่าบัฟเฟอร์ของเครื่องพิมพ์เต็ม ตัวอักษรกำลังถูกทำการเคลื่อนย้าย หรือเครื่องพิมพ์กำลังมีการเริ่มต้น การอินเทอร์เฟสจะไม่มีการส่งตัวอักษรใด ๆ จนกว่า บิต /BSY จะถูกเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต /ACK และ บิต STR ของรีจิสเตอร์ควบคุม มีไว้สำหรับทำให้การส่งข้อมูลถูกต้องถ้าการอินเทอร์เฟสมีการส่งข้อมูลมายังเครื่องพิมพ์แล้ว ขาสัญญาณ /ACK ในรีจิสเตอร์สถานะเป็นตัวบอกว่าได้รับข้อมูลเรียบร้อยแล้ว

ถ้าเครื่องเครื่องพิมพ์ไม่มีกระดาษ บิต PAP ถูกเซตเป็น 1 และในเวลาเดียวกันบิต OFON ถูกเคลียร์ และบิต /FEH ถูกเซตเพื่อบ่งชี้ว่าเกิดผิดพลาดขึ้น

ตารางที่ 5.4 รีจิสเตอร์ควบคุม (ออฟเซต 02h)

บิตที่	7	6	5	4	3	2	1	0
ชื่อสัญญาณ	X	X	X	IRQ	DSL	/INI	ALF	STR
หมายเลขขาสัญญาณ	-	-	-	-	17	16	14	1

X : ไม่ใช้ (โดยทั่วไปให้เท่ากับ 1)

IRQ : Hardware Interrupt Request

1 = อีนาเบิ้ล (Enabled) 0 = ดิสเอเบิ้ล (Disabled)

DSL : การเลือกเครื่องพิมพ์

1 = เครื่องพิมพ์ถูกเลือก 0 = เครื่องพิมพ์ไม่ถูกเลือก

/INI : การอินนิเชียลเครื่องพิมพ์ (Initial)

1 = เครื่องพิมพ์ปฏิบัติการธรรมดา 0 = ทำการอินนิเชียล

ALF : ป้อนกระดาษอัตโนมัติ

1 = ทำการป้อนกระดาษอัตโนมัติโดยเครื่องพิมพ์ 0 = ทำการป้อนกระดาษโดยผู้ให้

STR : สโตรป (Strobe)

1 = ส่งข้อมูลไปยังเครื่องพิมพ์ 0 = ไม่มีการส่งข้อมูล

รีจิสเตอร์ควบคุม จะทำการควบคุมเครื่องพิมพ์และควบคุมการเกิดฮาร์ดแวร์อินเทอร์รัพท์โดยการอินเทอร์เฟสแบบขนาน ถ้าบิต IRQ ถูกเซตก็จะเกิดการอินเทอร์เฟสทางฮาร์ดแวร์อินเทอร์รัพท์โดย IRQS (LPT2) หรือ IRQ7 (LPT1) ติดต่อไปยัง INT 01H และ INT OFH ตามลำดับ ถ้าสัญญาณ /ACK เกิดการเปลี่ยนแปลงระดับสูงเป็นระดับต่ำ ดังนั้นจะเกิดการอินเทอร์รัพท์ขึ้นจริงเมื่อเครื่องพิมพ์ได้รับตัวอักษร การจัดการควบคู่กันไปเช่นนี้สามารถที่จะทำการส่งตัวอักษรไปยังเครื่องพิมพ์ที่รอรับข้อมูลอยู่เว้นแต่ไบออสของพีซี โดยทั่วไปไม่ได้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์รัพท์นี้ไปขับข้อมูลให้ส่งออกไป แต่โดยทั่วไปจะใช้บิต /ACK ในการตรวจสอบการรับรู้โดยเครื่องพิมพ์

บิต DSL มีไว้สำหรับใช้เลือกว่าจะใช้เครื่องพิมพ์หรือไม่ ส่วนบิต /INI ใช้สำหรับอินนิเชียลเครื่องพิมพ์ และถ้าทำการเซตบิต ALF เครื่องพิมพ์จะทำการป้อนกระดาษอัตโนมัติหลังจากมีการพิมพ์แต่ละบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การแปลงฟาสท์ฟูริเยร์ (Fast Fourier Transform)

โดยทั่วไปการแปลงฟูริเยร์ (Discrete Fourier Transform หรือ DFT) เป็นการคำนวณที่ใช้เวลาค่อนข้างมาก เช่น การคำนวณ DFT ของข้อมูล N ลำดับจะต้องมีการคำนวณจำนวนเชิงซ้อนถึง N*N ครั้งและมีการบวกจำนวนเชิงซ้อนอีก N(N-1) ครั้ง จะเห็นได้ว่าความเร็วและจำนวนครั้งในการคำนวณเป็นสิ่งสำคัญที่จะทำให้ผลการคำนวณ DFT ช้าหรือเร็ว

ลำดับขั้นตอนในการคำนวณ DFT ให้เร็วขึ้น เรียกว่า ฟาสท์ฟูริเยร์ (Fast Fourier Transform หรือ FFT) วิธีการนี้จะทำให้การคำนวณ DFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N\log_2 N$ ครั้งเท่านั้นซึ่งจะทำให้การคำนวณ DFT เร็วขึ้นมากและจะทำให้มีลักษณะเป็นเรียลไทม์มากขึ้น

การแปลง FFT แบ่งได้เป็น 2 ชนิดใหญ่ ๆ คือ ชนิดลดทอนทางเวลา (Decimation In Time หรือ DIT) และชนิดลดทอนทางความถี่ (Decimation In Frequency หรือ DIF) สำหรับในส่วนนี้จะแสดงเฉพาะชนิดลดทอนทางเวลา ซึ่งเกี่ยวข้องกับโครงงานนี้เท่านั้น

6.1 ขั้นตอนวิธีลดทอนทางเวลา (Decimation In Time หรือ DIT)

วิธีนี้เป็นการจัดแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา X(n) ขนาด N จุด ออกเป็น 2 ลำดับสัญญาณขนาด N/2 จุดเท่ากัน คือ ลำดับคู่และลำดับคี่ โดยที่ลำดับคู่เกิดจากการเอาลำดับในตำแหน่งคู่มาเรียงกัน ที่เหลือเป็นลำดับคี่ ดังนั้นจะได้

$$\begin{aligned} x_e(m) &= x(2n) \quad ; \; m = 0,1,2,\ldots,(N/2)-1 \\ x_0(m) &= x(2n+1) \quad ; \; m = 0,1,2,\ldots,(N/2)-1 \end{aligned} \qquad (6.1.1)$$

ถ้าให้ W_N เท่ากับ exp(-j2(/N) จะทำให้การคำนวณ DFT ของลำดับ x(n) ที่ยาว N จุด สามารถเขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N-1} x_e(m)(W_N)^{km} + \sum_{m=0}^{N-1} x_o(m)(W_N)^{km} \qquad (6..1.2)$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

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

ซึ่ง $W_{N/2}$ เป็นค่า W ของลำดับความยาว N/2 จุดและ สมการ(6.1.2) สามารถเขียนใหม่ได้เป็น

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

$$X(k) = X_o(k) + (W_N)^k X_o(k) \qquad (6.1.3)$$

โดยที่ $X_e(k)$ และ $X_o(k)$ แทนผลการแปลง DFT ขนาด N/2 จุด ของลำดับ $X_e(m)$ และ $X_o(m)$ ตามลำดับ จากวิธีการนี้จะเห็นได้ว่าการคำนวณ DFT ขนาด N จุด สามารถแบ่งย่อยเป็นการคำนวณ DFT ขนาด N/2 จุด สองอันดับได้ โดยหลักการเดียวกันนี้ ถ้าแบ่งลำดับ $X_e(m)$ และ $X_o(m)$ ออกเป็นลำดับคู่ลำดับคี่ จนเหลือลำดับขนาด 2 จุด จะทำให้การคำนวณ DFT ขนาด N จุด สามารถทำได้โดยการแปลง DFT ขนาด 2 จุดจำนวน N/2 ภาค ดังแสดงในรูปที่ 6.2

การนำผลการแปลง DFT ขนาด 2 จุดจำนวน N/2 ภาคมารวมกันเพื่อให้เป็นการคำนวณ DFT ขนาด N จุด จะต้องมีหลักเกณฑ์ที่ถูกต้องด้วย เพราะจาก (6.1.3) ทั้ง $X_1(k)$ และ $X_2(k)$ เป็น DFT ขนาด N/2 จุดที่นิยาม เฉพาะช่วง $0 \le k \le N/2$ เท่านั้น ดังนั้นเมื่อนิยามในช่วง $k > N/2$ ด้วย จะได้

$$\begin{aligned} X(k) &= X_e(k) + (W_N)^k X_o(k) && ; 0 < k < N/2-1 \\ &= X_e(k-N/2) + (W_N)^k X_o(k-N/2) && ; N/2 < k < N-1 \end{aligned} \qquad (6.1.4)$$

เทอม $(W_N)^k$ เรียกว่า ตัวประกอบหมุน (Twiddle Factor) ซึ่งใช้ร่วมกันกับ DFT ขนาด 2 จุด หรือขนาด N/2 จุด ในการนำมาประกอบเป็น DFT ขนาด N จุดได้เหมือนเดิม

และจากความสัมพันธ์ $(W_N)^{k-N/2} = -(W_N)^k$ จะได้

$$\begin{aligned} X(k) &= X_1(k) + (W_N)^k X^2(k) && ; 0 < k < N/2-1 \\ &= X_1(k-N/2) + (W_N)^{k-N/2} X^2(k-N/2) && ; N/2 < k < N-1 \end{aligned} \qquad (6.1.5)$$

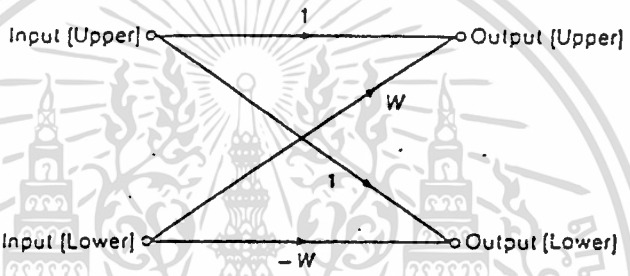
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากสมการนี้สามารถนำไปใช้สร้างหน่วยคำนวณที่ เรียกว่า หน่วยผีเสื้อ (Butterfly Unit) โดยมีข้อมูลเข้า คือ A และ B และข้อมูลออก คือ X และ Y เป็น

$$X = A + (W_N)^k B$$

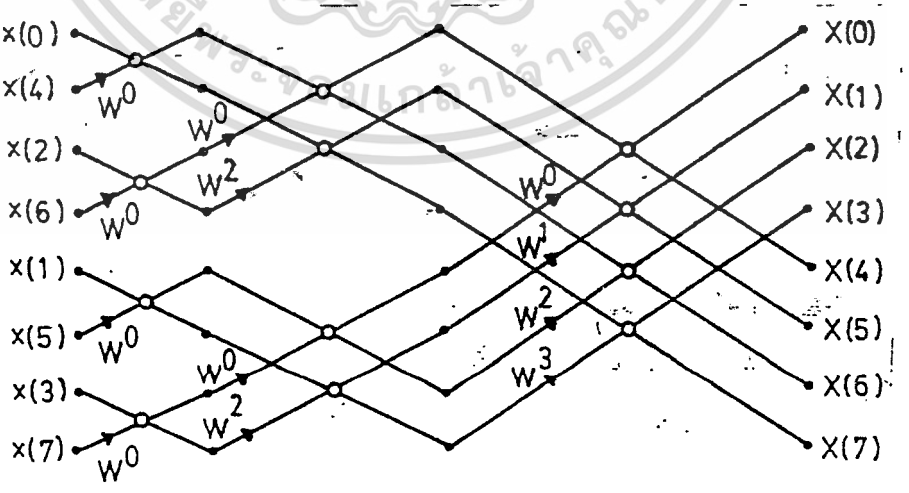
$$Y = A + (W_N)^k B \qquad (6.1.6)$$

ซึ่งสามารถเขียนอธิบายแทนด้วยโฟล์ชาร์ต ดังแสดงในรูปที่ 6.1



รูปที่ 6.1 แสดงหน่วยผีเสื้อของการคำนวณตามขั้นตอนวิธีลดทอนทางเวลา

สำหรับตัวอย่างการคำนวณ DFT โดยวิธี FFT แสดงได้ดังรูปที่ 6.2



รูปที่ 6.2 แสดงวิธีการของ FFT แบบลดทอนทางเวลา (DIT) สำหรับข้อมูลขนาด 8 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตเห็นได้ว่า เมื่อมีการแบ่งข้อมูลเข้าเป็นลำดับคู่และคี่ไปเรื่อย ๆ จนได้ข้อมูล 2 จุด จะทำให้ลำดับข้อมูลเข้ามีการเปลี่ยนแปลงไป ซึ่งการเปลี่ยนแปลงนี้เป็นการเปลี่ยนอย่างมีหลักเกณฑ์ ที่เรียกว่า การผันบิตกลับ (Bit Reverse) และสามารถแสดงได้ดังตารางที่ 6.1

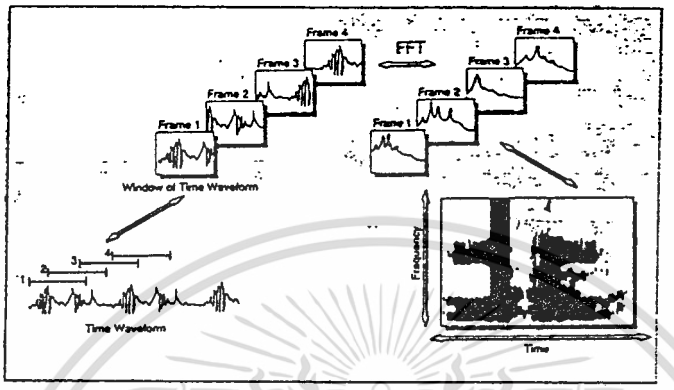
ตารางที่ 6.1 แสดงการสลับตำแหน่งข้อมูลเข้าด้วยการผันบิตกลับ

Index	Bit Pattern	Bit-Reversed Pattern	Bit-Reversed Index
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

6.2 สเปกโตรแกรม (Spectrogram)

สเปกโตรแกรมเป็นเครื่องมือที่ใช้วิเคราะห์สัญญาณอะนาลอกให้อยู่ในรูปเวลา-ความถี่ (Time-Frequency Domain) พร้อมกัน ซึ่งสามารถประยุกต์ใช้กับงานทั่วไป เช่น การประมวลผลทางเสียง (Speech Processing), Sonar และ Acoustics รูปที่ 6.3 แสดงวิธีการหาสเปกโตรแกรมของสัญญาณที่อินพุต

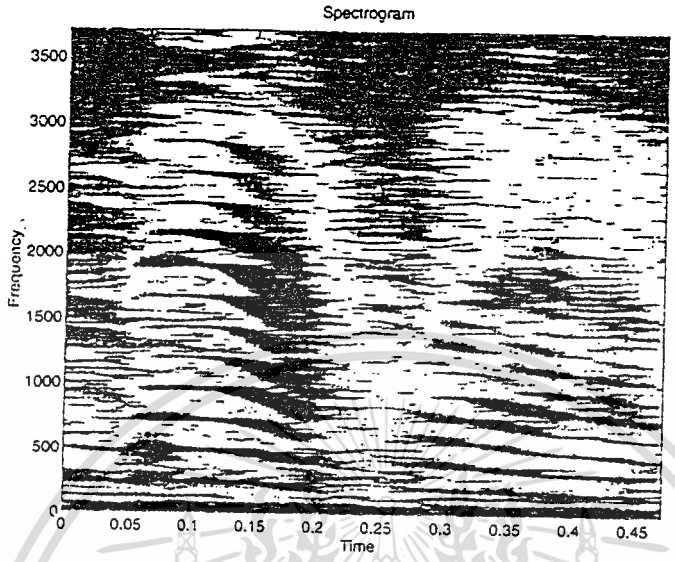
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 การคำนวณหาสเปกโตรแกรมของสัญญาณ

โดยสัญญาณอินพุตที่แซมปลิ้งเข้ามาจะถูกแบ่งออกเป็นส่วน ๆ ที่ซ้อนกันโดยใช้หน้าต่าง (Window) ครอบ ในที่นี้ใช้หน้าต่างแฮนนิ่ง (Hanning Window) แล้วคำนวณการแปลงฟูริเยร์ (Discrete-Time Fourier Transform) ของแต่ละส่วนโดยใช้ FFT เพื่อหารายละเอียดเกี่ยวกับความถี่ของสัญญาณนั้นในแต่ละหน้าต่าง จากนั้นนำผลที่คำนวณได้มาแสดงในรูปกราฟบนแกนเวลา-ความถี่ ดังรูปที่ 6.4 ซึ่งข้อมูลที่นำมาแสดงผลนั้นจะใช้เพียงครึ่งหนึ่งของจำนวนจุดทั้งหมดที่ใช้ในการคำนวณ FFT แต่ก็มีข้อจำกัดของวิธีนี้คือ ความกว้างของหน้าต่าง เพราะถ้าหน้าต่างมีความกว้างมากจะทำให้สเปกตรัมในแกนความถี่ (Frequency Domain) มีความละเอียดมากแต่ขณะเดียวกันก็ให้ความละเอียดต่ำในแกนเวลา (Time Domain) และเพื่อไม่ให้มีการสูญเสียข้อมูล หน้าต่างจะต้องมีการซ้อนกัน โดยถ้าต้องการความถูกต้องมากต้องให้มีการซ้อนกันมากขึ้น แต่ในทางปฏิบัติความกว้างของหน้าต่างที่ซ้อนกันจะเป็นครึ่งหนึ่งของความกว้างหน้าต่างเพื่อให้สามารถคำนวณ FFT ได้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 การแสดงผลสเปกโตรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 โปรแกรมภาษาแอสเซมบี้

```
;-------------------------------------------------------------------------
; This is the source code for the DSK spectrogram analyzer.
; Assemble using the DSK5A assembler and execute the DSK
; loader to load and run the application on your PC
; Assemble using the DSK5A assembler and execute the DSK
; loader to load and run the application on your PC
;-------------------------------------------------------------------------
                .mmregs
FFT_S           .set        1024
FFT_S-1         .set        1023
FFT_S/2         .set        512
(FFT_S/2)-1     .set        511
;-------------------------------------------------------;
TA              .set        6       ;
RA              .set        6       ;
Tap             .set        1       ;
Rap             .set        1       ;
TB              .set        18      ;
RB              .set        18
AIC_CMD         .set        080h    ;      GAIN
                .ds     0f00h
TEMP            .word       0       ;location of TEMPorary storage
ACCU_lo         .word       0       ;
ACCU_hi         .word       0       ;
STAT1           .word       0       ;STAT1 storage
AR4_temp        .word       _B_base
O_temp          .word       0
B_temp          .word       _B_base
;-------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                .ps     080ah
        B       RINT        ;0A; Serial port receive interrupt RINT
        B       XINT        ;0C; Serial port transmit interrupt XINT
*********************************************************************
* TMS32C05X INITIALIZATION                                          *
* This routine initializes the C5x registers, internal RAM and      *
* external RAM from xxxx to FFFF                                    *
*********************************************************************
                .ds 01000h
                .listoff
                .include "twidle.asm"  ;NOTE: Internal RAM block B2 is used
                .liston
                .ps     0a00h
                .entry
start   setc    INTM                ; Disable interrupts
        ldp     #0                  ; Set data page pointer
        splk    #830h,PMST          ; 9K on-chip RAM as Data, No ROM
        lacl    #0                  ; Set Wait State Control Regester
        samm    CWSR                ; for 0 waits in pgm & data memory
        samm    PDWSR               ;
*
* initalize and reset serial port
*
        splk    #20h,TCR
        splk    #1,PRD
        mar     *,AR0
        lacl    #08h            ; set FSM bit for FSX/FSR per frame
        samm    spc             ; Configure for 16 bit mode with
        lacl    #0C8h           ; external CLKX, reset tx and rx
        samm    spc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lamm  drr             ; clear first int
lacc  #0080h
sach  dxr             ; clear first int
sacl  GREG            ; Pulse AIC reset by setting it low
lar   AR0,#0FFFFFh
rpt   #10000          ; and taking it high after 1000 cycles
lacc  *,0,AR0         ; (.5ms at 50ns)
sach  GREG
setc  SXM
setc  OVM

lacc  #_B_base        ; initial circular register
samm  CBSR1           ; AR6 and AR4
samm  CBSR2           ;
lacc  #_B_end         ;
samm  CBER1
samm  CBER2
splk  #04eh,CBCR      ; disable AR4, enable
;---------------------------------------------------------
call  AIC_SET         ; DO NOT CHANGE DP WITHOUT RESTORING IT!
lacl  #010h           ; RINT
samm  IMR             ;
ldp   #TEMP

lar   AR6,#_B_base          ;AR6=ISR BUFF data pointer
lar   AR7,#FFT_S-1          ;AR7=ISR loop counter
lacl  #_B_base              ;AR4 point to INPUT buffer
sacl  AR4_temp
clrc  INTM                  ;enable RINT
b     MOVE_IO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FFT:      lar    AR0,#FFT_S/2          ;
          mar    *,AR0                 ;start FFT with AR0=FFTSize
new_stg:  lar    AR1,#_D_base          ;AR1 is the TOP BFLY address
          lar    AR2,#_D_base          ;AR2 is the BOT BFLY address
          lar    AR3,#_T_base+1        ;AR3 is the TWiddle pointer
          lar    AR4,#FFT_S/2          ;AR4 counts DFT blocks
          b      n_DFT2,*,AR1          ;
DFT:      mar    *BR0+,AR5             ;complete circular buffer for TW's
          lar    AR5,#1                ;set up DFT loop with *BR0+/BANZ
          mar    *BR0+,AR1             ;using 1 cuts *BR0+ loop in half!
          ;--------------------------------------
          ; AR1=Top AR2=Bottom AR3=Twiddle
          ;--------------------------------------
BFLY:     lacc   * ,14,AR2             ;(imag1+imag2)/4
          add    * ,14,AR1             ;
          sach   *+,1 ,AR2             ;store TOP imag
          sub    * ,15                 ;(imag1-imag2)/2
          sach   *+,1 ,AR1             ;store BOT imag
          lacc   * ,14,AR2             ;(real1+real2)/4
          add    * ,14,AR1             ;
          sach   *+,1 ,AR2             ;store TOP real
          sub    * ,15                 ;(real1-real2)/2
          sach   * ,1,AR5              ;store BOT real
          banz   OK,*BR0+,AR3          ;If at DFT end quit early
          ;----------------------
          mar    *+,AR2                ;clean up TW base (xxx0000+1)
          mar    *+                    ;modify BOTom DATA pointer
          mar    *0+                   ;
          mar    *0+,AR1               ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
n_DFT2: mar    *0+                ;modify the TOP pointer
        mar    *0+,AR4            ;
        banz   DFT,*0-,AR3        ;dec DFT block count AR4 by OFFset
        mar    *,AR0              ;
        mar    *BR0+              ;
        banz   new_stg,*          ;if OFFset was 1, now cleared
        b      endFFT             ;
        ;-------------------------
OK      lt     *-,AR2             ;TREG=TWR    *NOTE* Twiddles are Q15
        mpy    *-                 ;PREG=REAL*TWR
        ltp    *+,AR3             ;TREG=IMAG    ACCU=REAL*TWR
        mpy    *                  ;PREG=IMAG*TWI          AR2=R AR3=I
        lts    *+,AR2             ;TREG=TWI     ACCU=REAL*TWR-IMAG*TWI
        mpy    *                  ;PREG=REAL*TWI
        sach   *-,1               ;
        ltp    * ,AR3             ;TREG=IMAG    ACCU=REAL*TWI
        mpy    *BR0+,AR2          ;PREG=IMAG*TWR
        apac                      ;ACCU=IMAG*TWR+REAL*TWI
        sach   *+,1               ;
        b      BFLY,*+,AR1        ;
        ;-------------------------------------------------------
endFFT: mar    *,AR2              ;Transform REAL & IMAG to log magnitude
        lar    AR2,#_D_base       ;AR3=FFT data pointer
        lar    AR3,#FFT_S-1       ;AR5=FFT loop counter

more_MAG  sqra   *+               ;PREG=IMAG^2
          ltp    *                ;TREG=REAL    ACCU=IMAG^2
          mpy    *,AR1            ;PREG=REAL^2
          apac                    ;ACCU=REAL^2+IMAG^2
          lar    AR1,#31          ;NORMalize the accumulator
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        rpt     #30             ;use for other types of conversion
        norm    *-              ;
        nop
        nop
        mar     *,AR2
        bcnd    sig_NZ,NEQ      ;if zero must return 0
        lar     AR1,#0                  ;
sig_NZ  sach    *,2             ;clear explicit 1.0 from mantissa
        lacl    *               ;load into accumulator and
        and     #0FF80h         ;; clear LSB's for AIC
        sar     AR1,*           ;append the exponent (AR5)
        add     *,16            ;
        xor     #020h,15        ;change to 2's compliment
        rpt     #3              ;jam result to top of ACCU
        sfl                     ;
        sach    *+,7,AR3        ;
        banz    more_MAG,*-,AR2 ;keep going until all done
        ;-------------------------------------------------
BITREV: lar     AR0,#FFT_S      ;Now perform Output bit reversal
        lar     AR1,#_D_base    ;by moving the magnitude, which
        lar     AR2,#_D_base+1  ;is in the REAL slots, into the
        lar     AR3,#FFT_S-1    ;IMAG slots of the FFT data array
more_BR: lacc   *+              ;load the magnitude
        mar     *+,AR1          ;
        sacl    *BR0+,0,AR3     ;move it to an open IMAG slot
        banz    more_BR,*-,AR2  ;more data to move?
        ;-------------------------------------------------
MOVE_IO: mar    *,AR7                   ;wait until buffer is full
        banz    MOVE_IO,*,AR2           ;(AR7 is decremented by ISR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lar     AR1,#_O_base            ;AR1=OUTPUT data pointer
        lar     AR2,#_T_base+1          ;AR2=WIN data pointer
        lar     AR3,#_D_base            ;AR3=FFT data pointer
        lar     AR4,AR4_temp            ;AR4=FFT input pointer
        lar     AR5,#(FFT_S/2)-1        ;AR5=FFT loop counter
        lacc    #0CEh                   ;enable circular AR4
        samm    CBCR
        lacc    #_O_base
        sacl    O_temp
        lar     AR7,#(FFT_S/2)-1        ;AR7=ISR BUFF loop counter
        ;-------------------------
        lar     AR0,#FFT_S/2            ; Use twiddle table for raised
more_IO: lacc   #04000h,1               ; cosine window
        add     *BR0+,0,AR4             ;
        sfr                             ;
        sacl    TEMP                    ;
        lt      TEMP                    ;TREG=WIN
        mpy     *+,AR3                  ;PREG=IN*WIN
        lacl    *,AR1                   ;ACCU=magnitude (put in Buffer)
        sacl    *+,0,AR3
        sach    *+                      ;IMAG=0
        pac                             ;
        sach    *+,1,AR5                ;REAL=IN (windowed buffer)
        clrc    INTM                    ;1st BUFF posn clr so enable INT's
        banz    more_IO,*-,AR2          ;
        lar     AR5,#(FFT_S/2)-1        ;AR5=FFT loop counter
        sar     AR4,AR4_temp            ;save AR4 for used next cycle
more_IO2 lacc   #04000h,1               ; cosine window
        add     *BR0-,0,AR4             ;
        sfr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

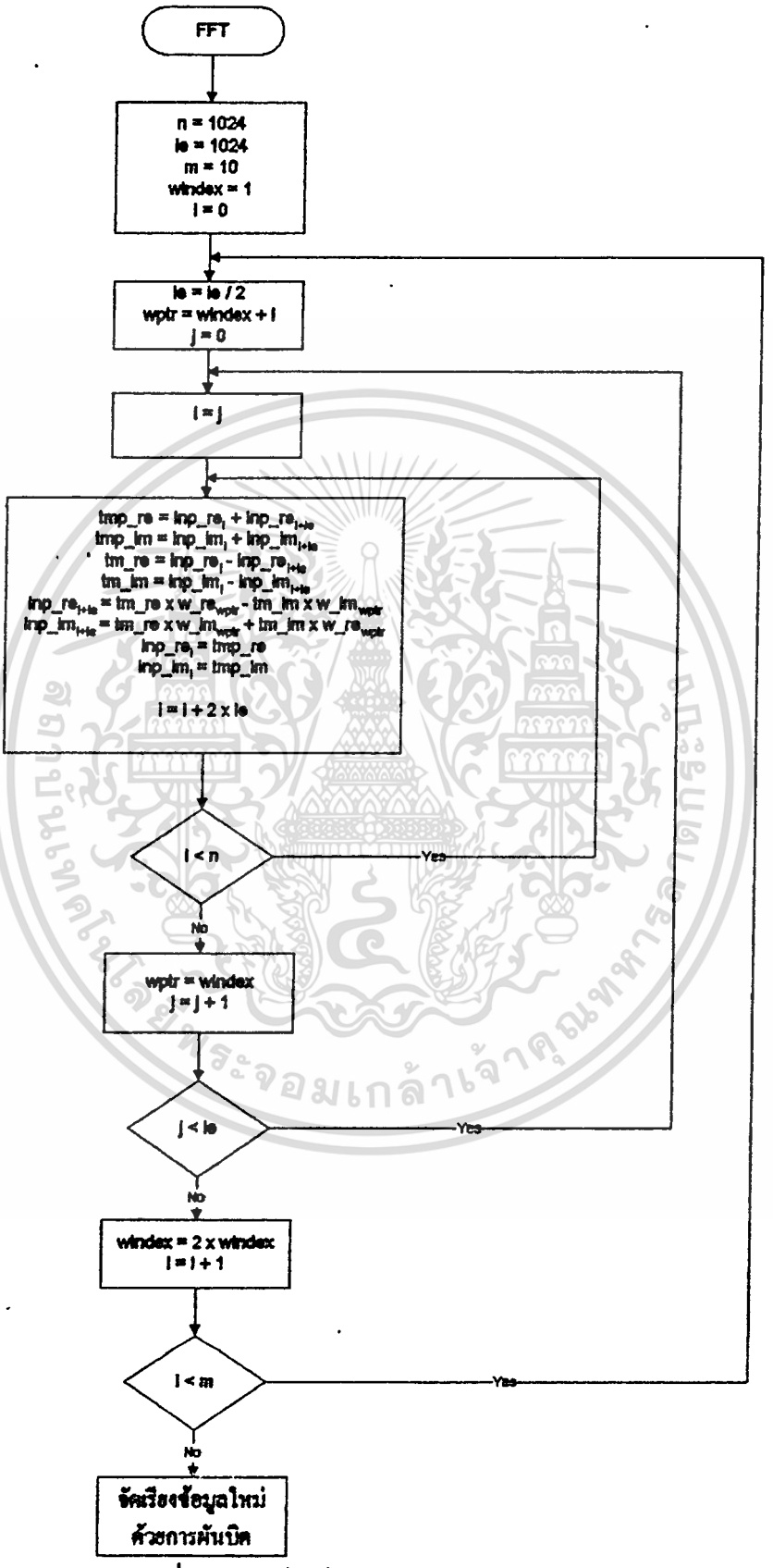
```
        sacl    TEMP            ;
        lt      TEMP            ;TREG=IN
        mpy     *+,AR3          ;PREG=IN*WIN
        lacl    *,AR1           ;ACCU=magnitude (put in Buffer)
        sacl    *+,0,AR3
        sach    *+              ;IMAG=0
        pac                     ;
        sach    *+,1,AR5        ;REAL=IN (windowed buffer)
        banz    more_IO2,*-,AR2 ;
        lacc    #04eh                   ;disable circular AR4
        samm    CBCR
        b       FFT             ;
;---------------------------------------------------------
XINT    rete
;---------------------------------------------------------
RINT:   mar     *,AR7                   ;;Recover the ARP from ARB (MON26 only)
        banz    more_buf,*-,AR6         ;AR6 = current buffer position EINT
        lar     AR7,#0                  ;if buffer is full RET w/o
        reti                    ;
more_buf
        lamm    DRR             ;
        sacl    *+                      ;store data from DRR
        sar     AR6,B_temp              ;exchange AR6 from _B_base to _O_base
        lar     AR6,O_temp
        out     *+,PA1                  ;send data to PC
        sar     AR6,O_temp              ;restore AR6 from _O_base to _B_base
        lar     AR6,B_temp
        rete                    ;
*************************************************************************

AIC_SET: lacl   #020h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        samm   IMR              ;XMIT interrupt
        lacc   #AIC_CMD,2       ;
        add    #03h             ;
        call   AIC_2nd          ;
      ;-----------------------
        lacc   #TB,9            ;
        add    #RB,2            ;
        add    #02h             ;
        call   AIC_2nd          ;
      ;-----------------------
        lacc   #TA,9            ;
        add    #RA,2            ;
        call   AIC_2nd          ;
      ;-----------------------
        ret                     ;
;-------------------------------------------------------
AIC_2nd:                        ;
        sach   DXR              ;
        clrc   INTM             ;;;
        idle                    ;
        add    #6,15            ;0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
        sach   DXR              ;
        idle                    ;ACCU_hi requests 2nd XMIT
        samm   DXR              ;
        idle                    ;ACCU_lo sets up registers
        lacl   #0               ;
        samm   DXR              ;make sure the word got sent
        idle                ;
        setc   INTM
        ret                 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 โฟล์ชาร์ตแสดงการหา FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 โปรแกรมภาษาซี

```
/* This is spectrogram */
#define PORT 1
void sport(),port_intit(),chack_stat(),wait();
int i,n,d,s,p,N,rd;
char A;
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <graphic.h>
#define PI 3.14159
int n,N,lll,num,my,x;
int D=1024;
int huge DetectVGA256(void);
float y[1024],z[1024];
void spectrogram();
void menu();
void initialize_graphics_mode(void);
void drawblock1(int xs,int ys,int xl,int yl,int dimen);
void rdata();
void plotfft();
void plota();
main()
{
    clrscr();
    plotfft();
    for(;;)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          setfillstyle(1,WHITE);
          bar(14,25,(getmaxx()-8),(getmaxy()-13));
          plota();
        }
       menu();
  }
  void rdata(void)
   {
          for(n=0;n<1024;n++)
         {
           A = inp(0x379);
           while ((A & 0x80) == 0x00)
           A = inp(0x379);
           A = inp(0x378);
           y[n]=  A ;
         }
  }
  void menu()
   {
    int d;
    clrscr();
    initialize_graphics_mode();
    drawblock1(10,5,getmaxx(),getmaxy()-15,0);
    setcolor(RED);
    outtextxy(240,100,"(F1) : Plot Frenquency Response ");
    outtextxy(240,130,"(F2) : Exit ");
    outtextxy(9,23+1*(getmaxy()-20)/5," ");
    outtextxy(9,23+2*(getmaxy()-20)/5," ");
    for(;;)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {   d=bioskey(0);
        switch(d)
        {
          case 0x3b00: plotfft();break;
          case 0x3c00: exit(1);break;
          case 0x3d00: exit(1);break;
          case 0x3e00: exit(1);break;
        }
    }
}
void initialize_graphics_mode(void)
 {
   int gdriver =DETECT,gmode,errorcode;
   initgraph(&gdriver,&gmode,"c:\\tc\\graphic");
   errorcode = graphresult();
   if (errorcode != grOk)
    {
      printf(" Graphics error : %s\n",grapherrormsg(errorcode));
      printf(" Press any key halt : ");
      getch();
      exit(1);
    }
}
void drawblock1(int xs,int ys,int xl,int yl,int dimen)
{
   setviewport(0,0,getmaxx(),getmaxy(),1);
   setfillstyle(1,WHITE);
   bar(xs,ys,xl,yl);
   setcolor(BLACK);
   rectangle(xs+2,ys+2,xl-2,yl-2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    line(xs+3,ys+3+dimen,xl-3,ys+3+dimen);
    setcolor(DARKGRAY);
    rectangle(xs+3,ys+3,xl-3,yl-3);
    line(xs+2,ys+2+dimen,xl-2,ys+2+dimen);
  }
void plotfft()
 {
  int d,tx,ty;
  float my,nmax;
  initialize_graphics_mode();
  drawblock1(10,22,getmaxx()-5,getmaxy()-10,0);
  setcolor(LIGHTRED);
    outtextxy(270,1," SPECTROGRAM ");
  setcolor(CYAN);
    tx =14;
    ty =5+(getmaxy()-20);
  }
void plota()
  {int tx,ty,d;
   float my,nmax,f;
   setlinestyle(SOLID_LINE,1,1);
     for(x=1;x<=(getmaxx()-24);x=x+3)
      {
        rdata();
        tx=14+x-1;
        ty=6+(getmaxy()-19);
        for(d=1;d<(getmaxy()-37);d=d+2)
       {
        z[d] = (y[d]+y[d+1])/2;
        f=((z[d]*10)/128.00)+12;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setfillstyle(1,f);
            bar(tx,ty,(tx+3),(ty-3));
             tx=tx;
             ty=ty-3;
            d=d++;
     } }
    }
     void port_intit(int port,unsigned code)
    {
         union REGS r;
         r.x.dx =port;
         r.h.ah =0;
         r.h.al =code;
         int86(0x14,&r,&r);
    }
  void sport(int port,char c)
  {
         union REGS r;
         r.x.dx =port;
         r.h.al =c;
         r.h.ah =1;
         int86(0x14,&r,&r);
         if(r.h.ah & 128)
          {
            printf("send error\n");
            exit(1);
          }
  }
  check_stat(int port)
  {
```

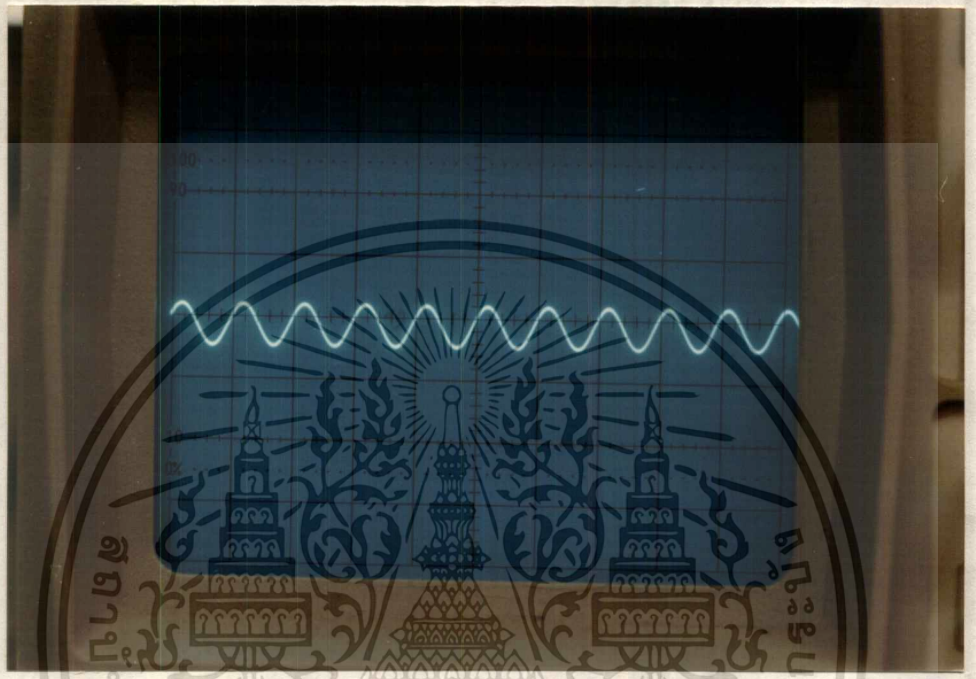
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    union REGS r;
    r.x.dx =port;
    r.h.ah =3;
    int86(0x14,&r,&r);
    return r.x.ax;
}
rport(int port)
{
   union REGS r;
       while(!(check_stat(PORT)&256))
           if(kbhit())
              {
              getch();
              exit(1);
              }
        r.x.dx =port;
        r.h.ah =2;
        int86(0x14,&r,&r);
        if(r.h.ah & 128)
        printf("read error\n");
        return r.h.al;
}
wait(int port)
{
     if(rport(port) != "0x01d")
     {
       printf("communication error\n");
       exit(1);
     }
}
```

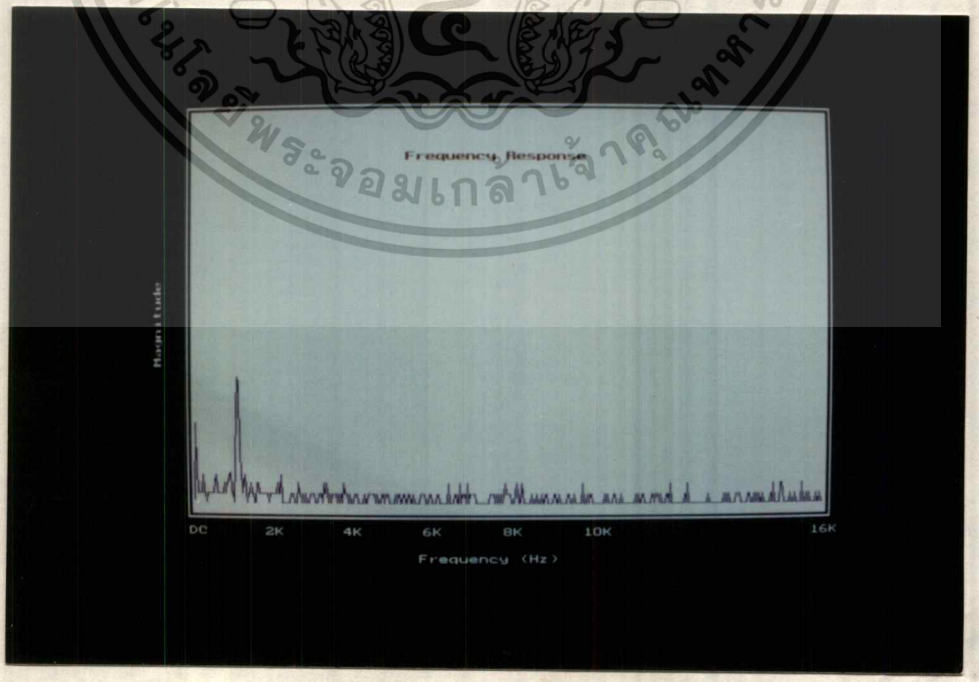
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง และ สรุปผลการทดลอง

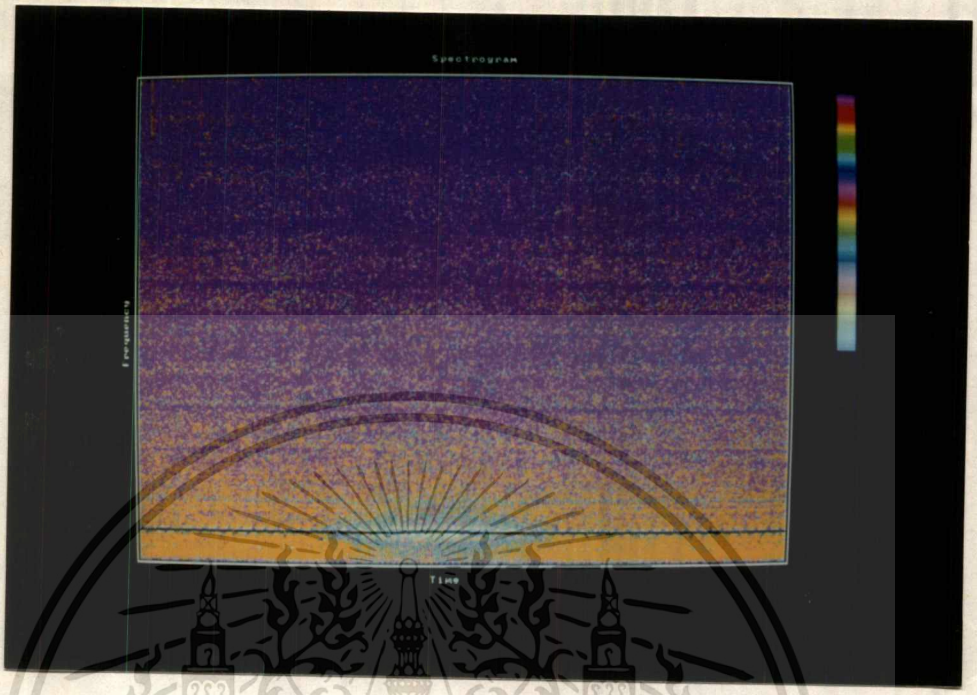


รูปที่ 7.1 รูปสัญญาณ SineWave 1KHz ขนาด 130mV



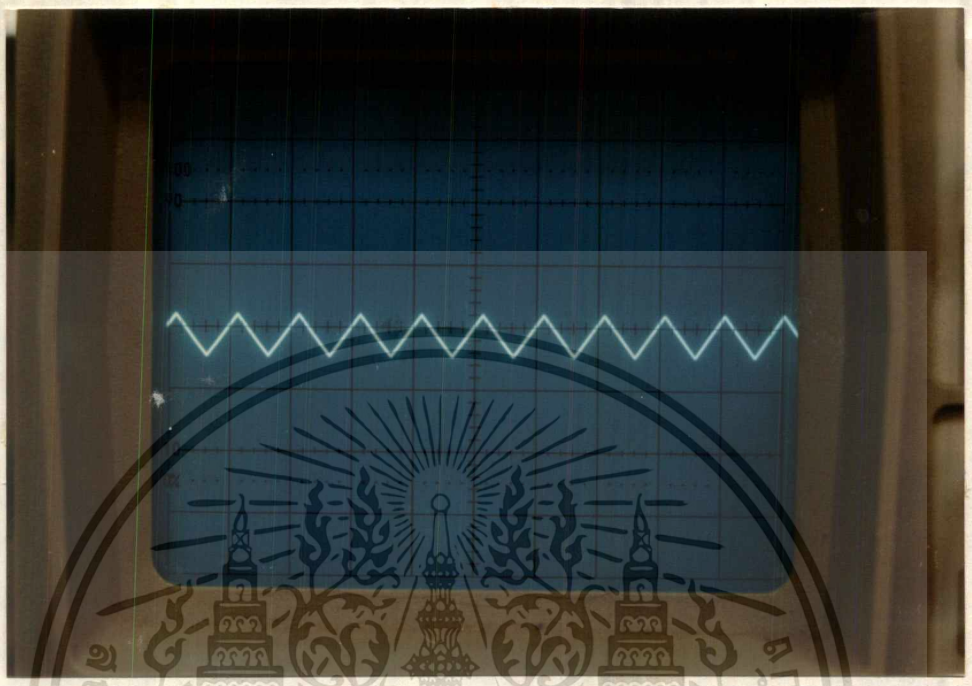
รูปที่ 7.2 รูปสเปกตรัมสัญญาณ Sine Wave 1KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.3 รูปสเปกโตรแกรมสัญญาณ Sine Wave 1KHz ขนาด 130 mV

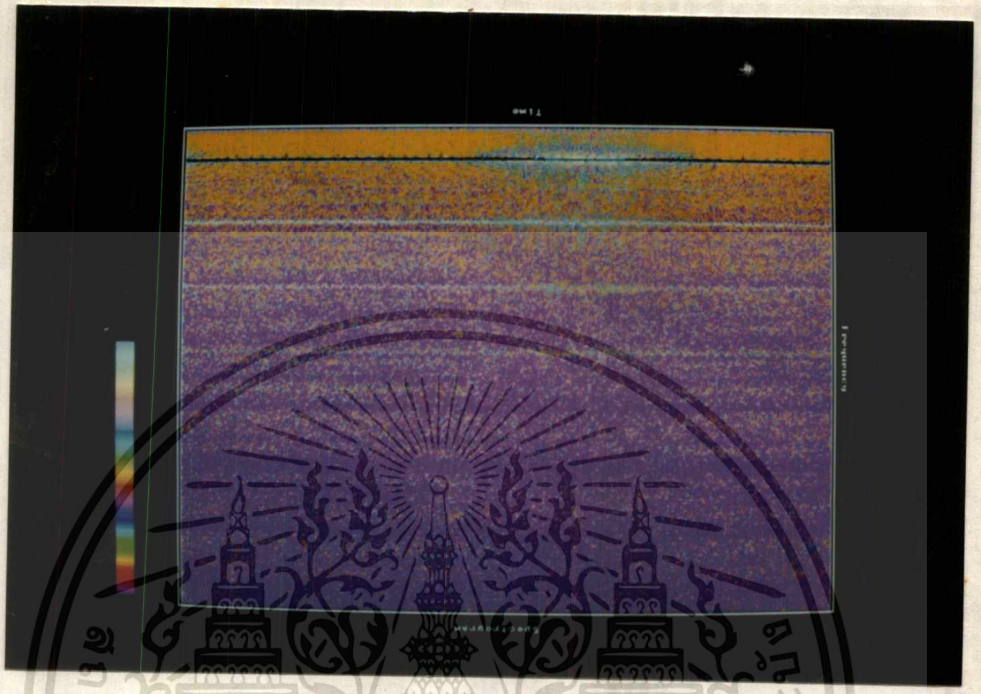
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 รูปสัญญาณ Triangle Wave 1KHz ขนาด 130mV

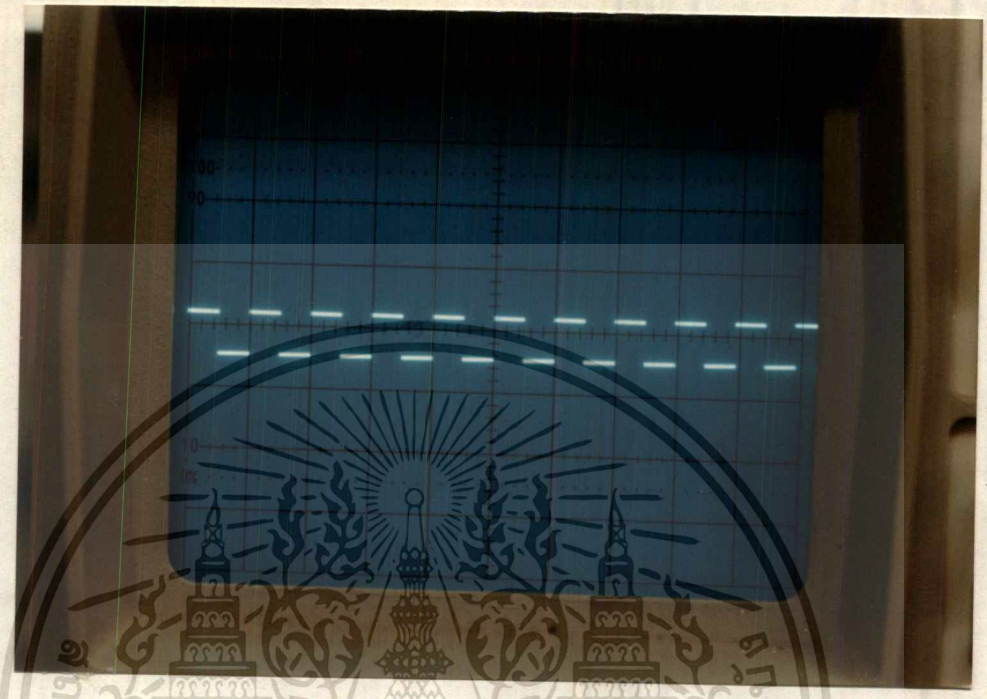
รูปที่ 7.5 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 1KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

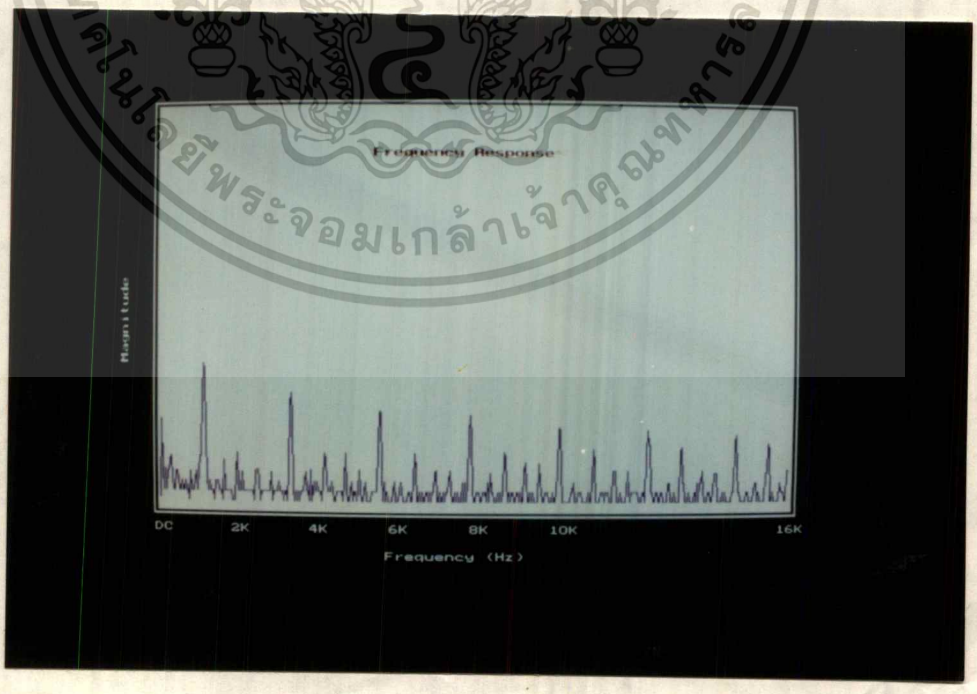


รูปที่ 7.6 สเปกโตรแกรมสัญญาณ Triangle Wave 1KHz ขนาด 130 mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

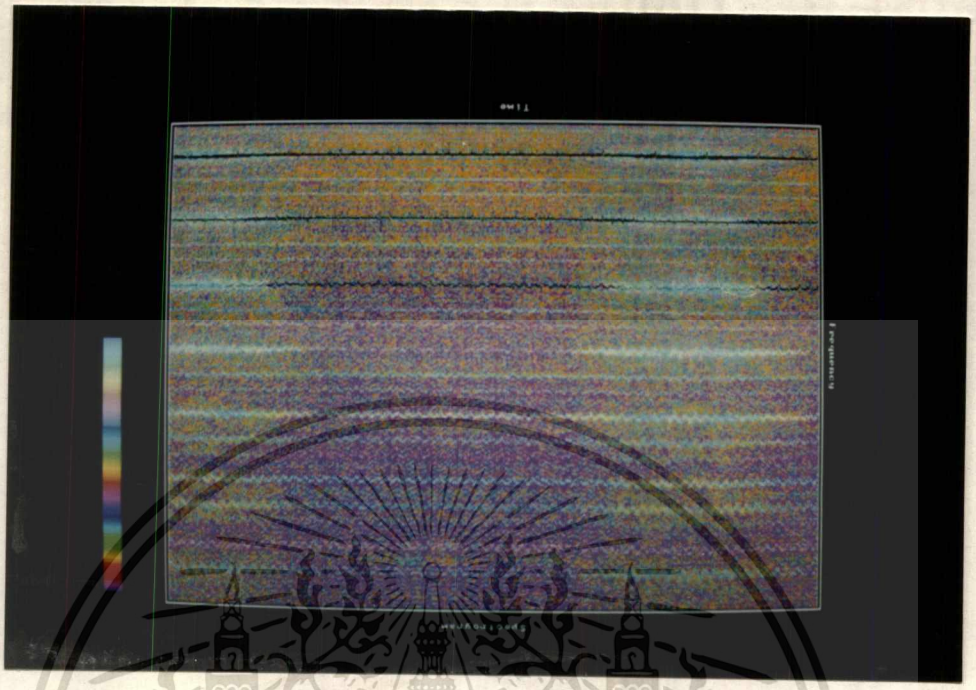


รูปที่ 7.7 รูปสัญญาณ Square Wave 1KHz ขนาด 130mV



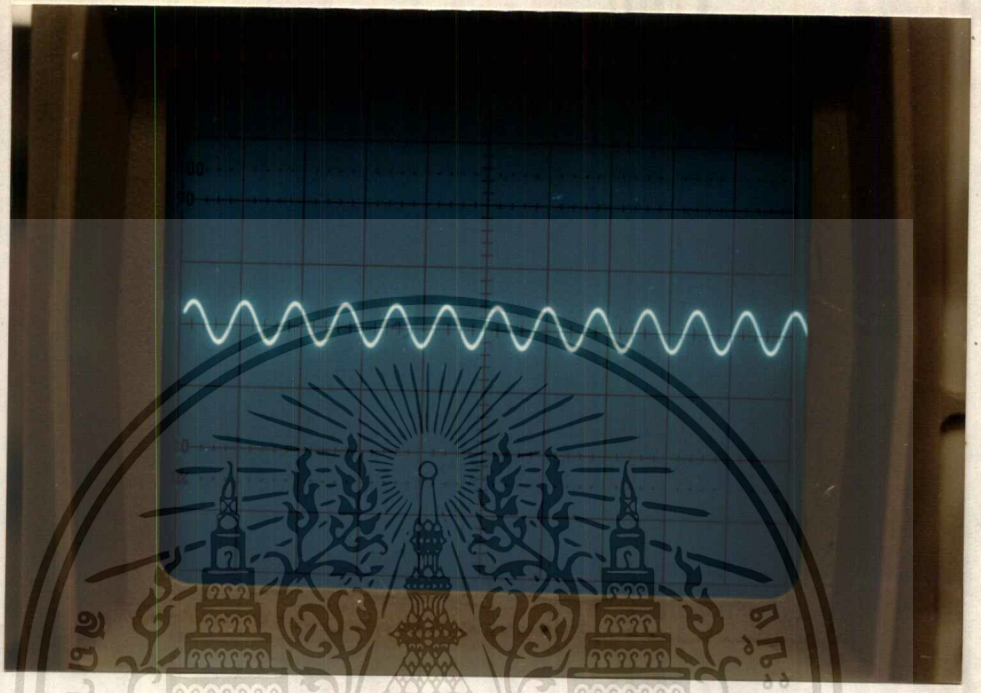
รูปที่ 7.8 สเปกตรัมสัญญาณ Square Wave 1KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

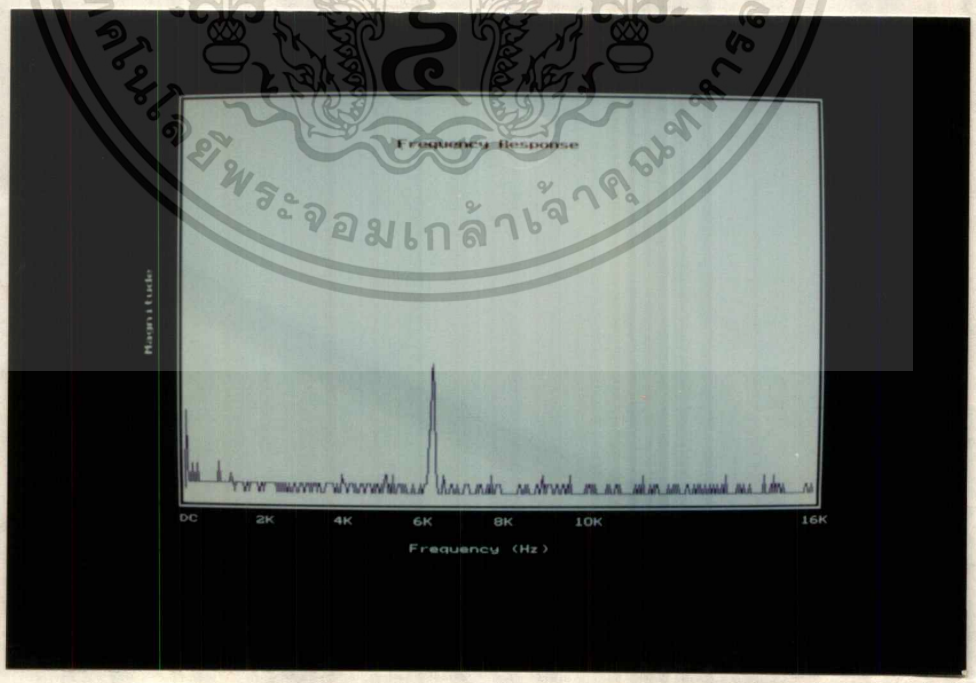


รูปที่ 7.9 รูปสัญญาณสเปกโตรแกรม Square Wave 1 KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

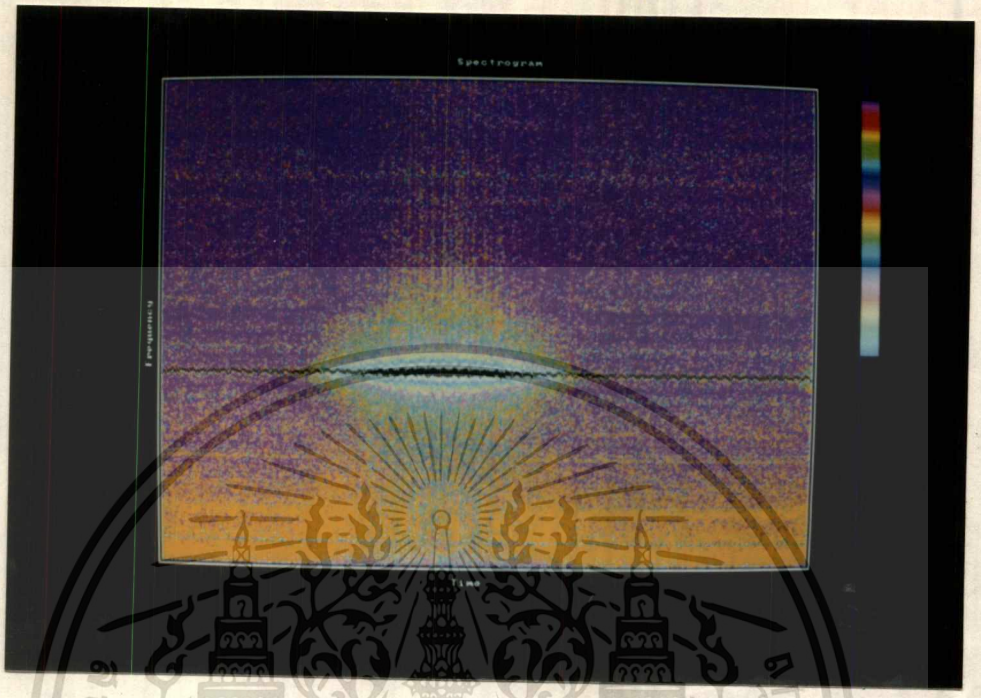


รูปที่ 7.10 รูปสัญญาณ Sine Wave 6KHz ขนาด 130mV



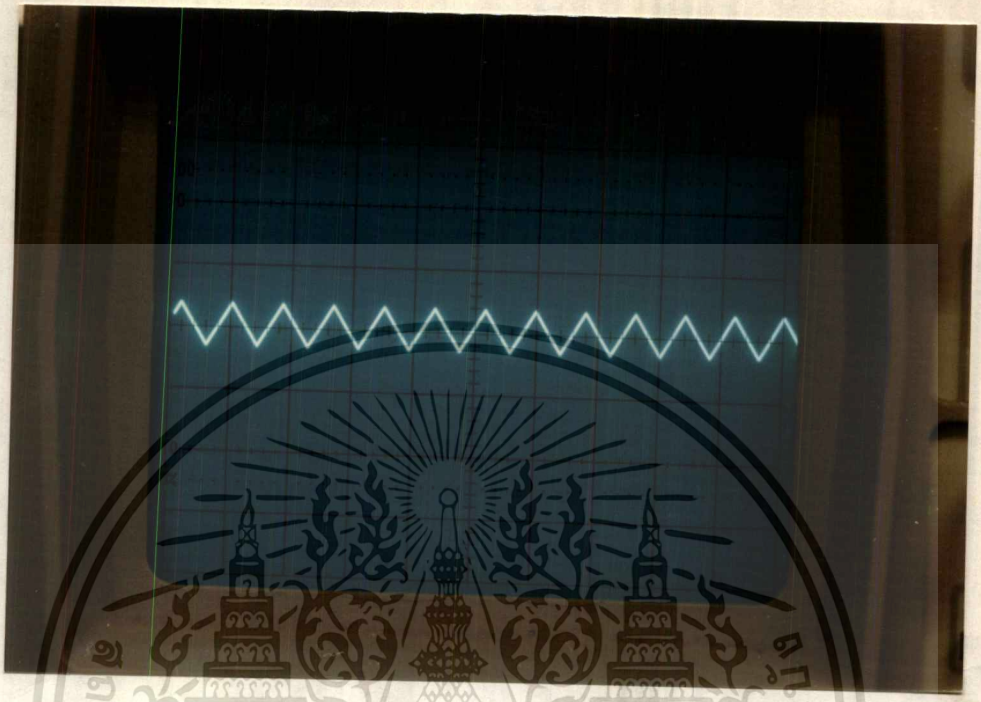
รูปที่ 7.11 รูปสเปกตรัมสัญญาณ Sine Wave 6KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

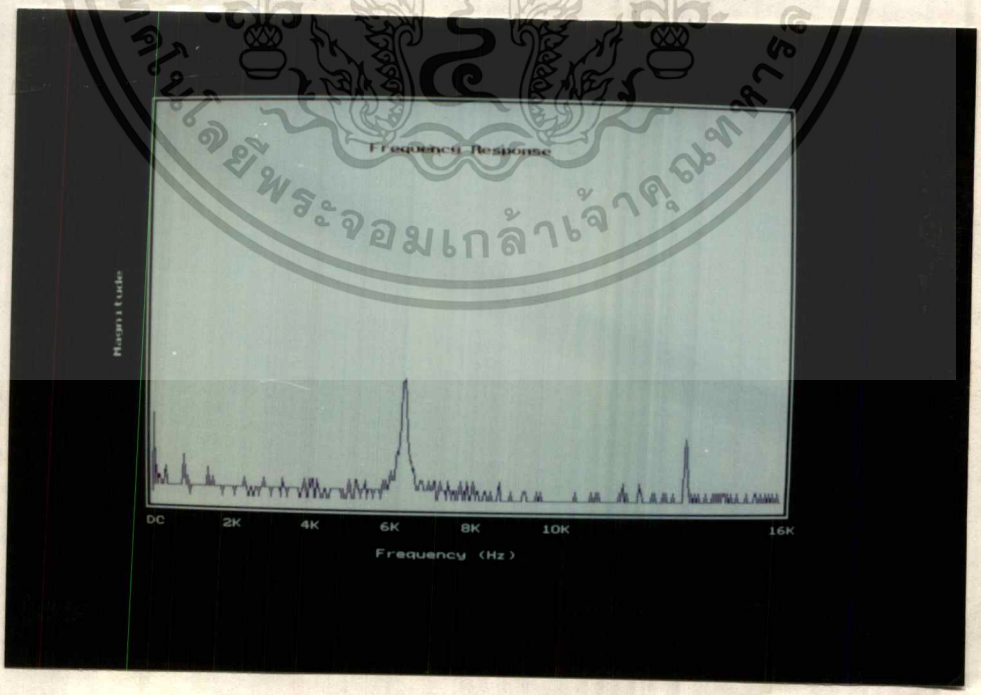


รูปที่ 7.12 รูปสเปคโตรแกรมสัญญาณ Sine Wave 6KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

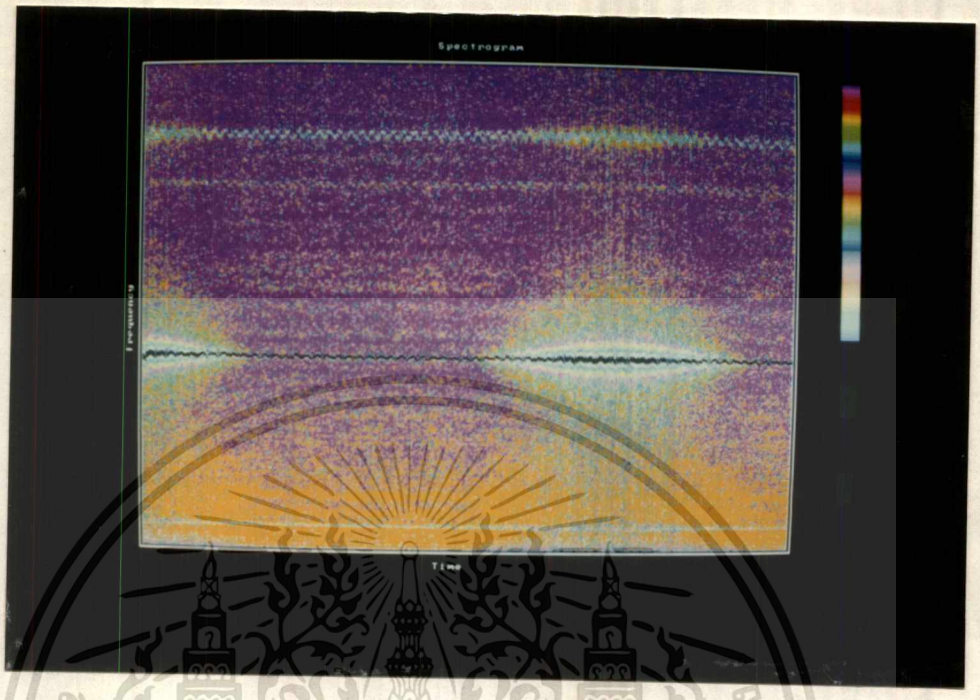


รูปที่ 7.13 รูปสัญญาณ Triangle Wave 6KHz ขนาด 130mV



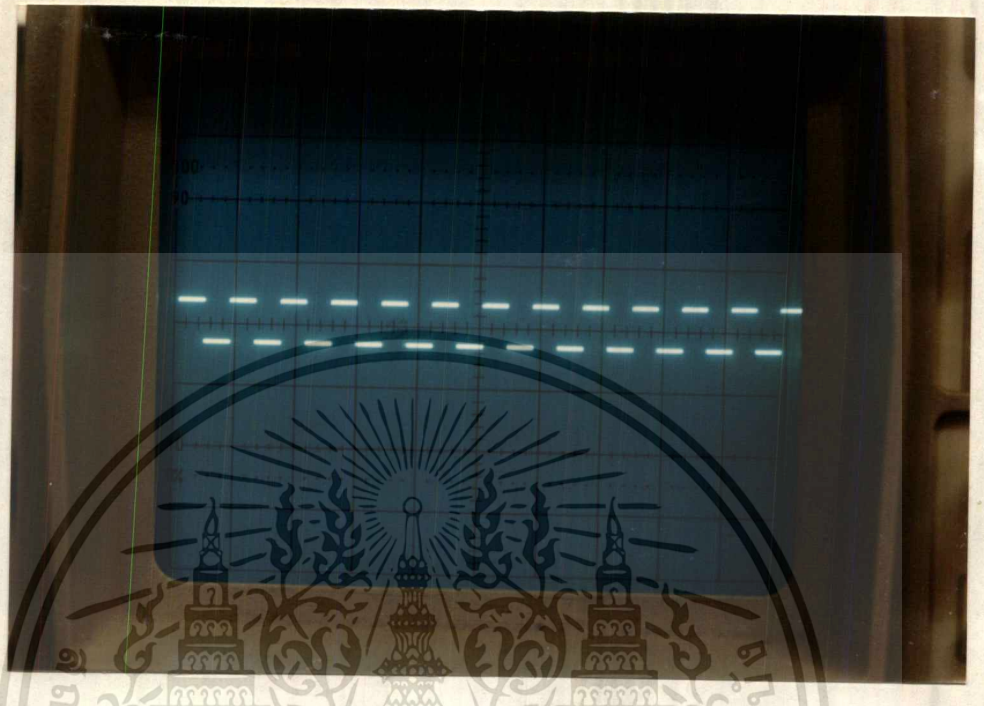
รูปที่ 7.14 รูปสเปกตรัมสัญญาณ Triangle Wave 6KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

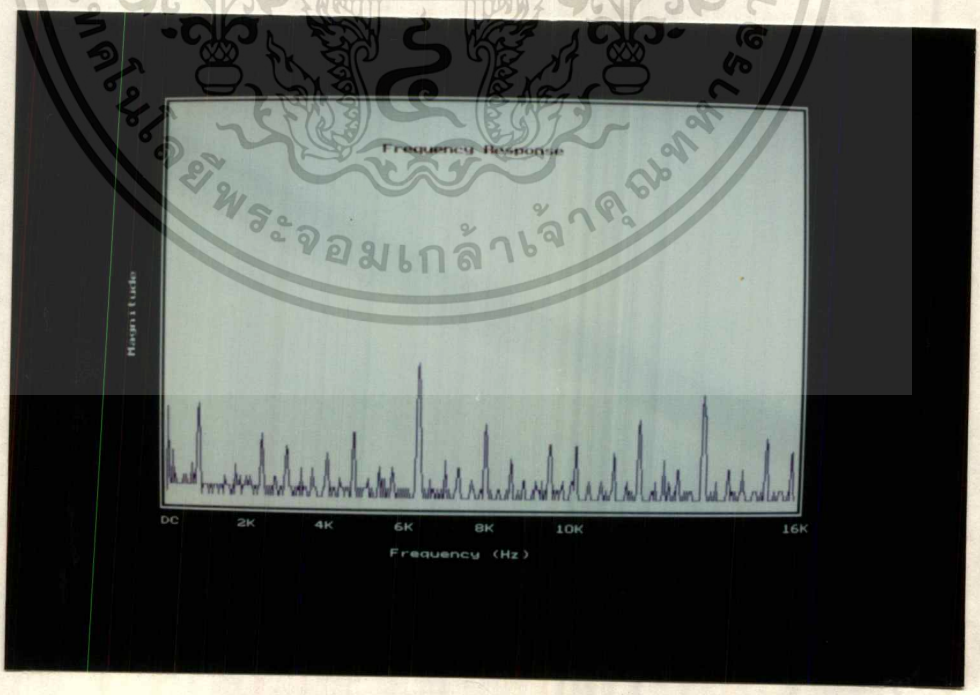


รูปที่ 7.15 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 6KHz ขนาด 130 mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

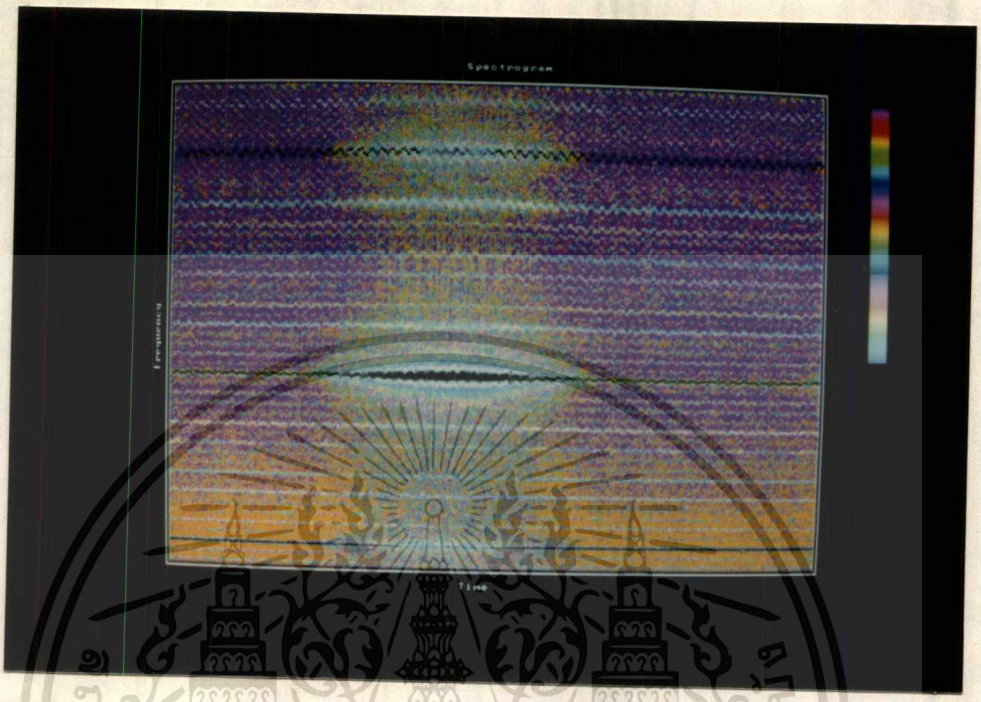


รูปที่ 7.16 รูปสัญญาณ Square Wave 6KHz ขนาด 130mV



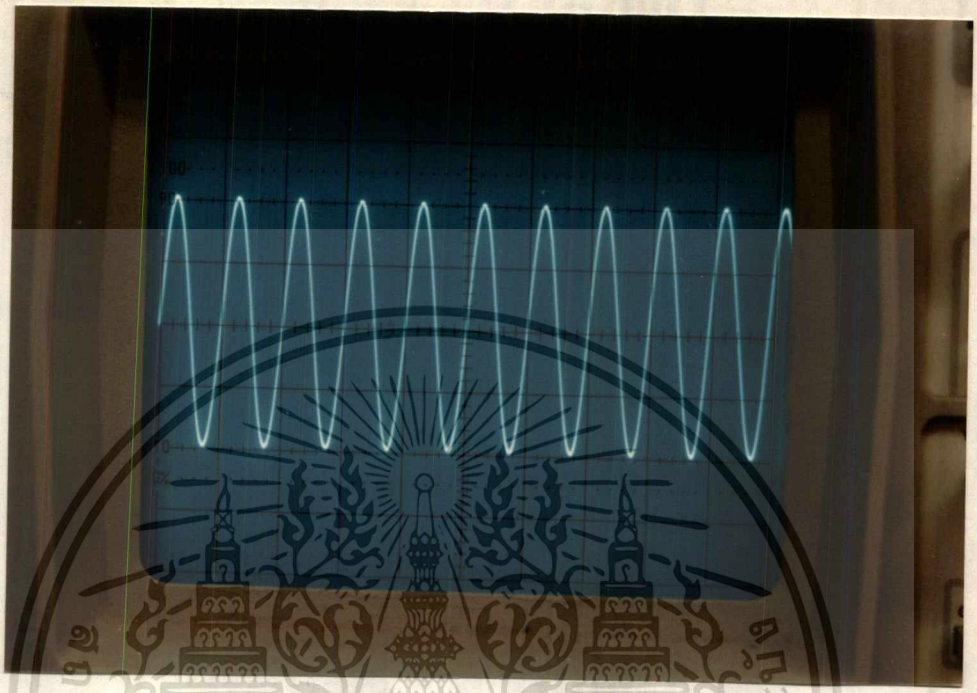
รูปที่ 7.17 รูปสเปกตรัมสัญญาณ Square Wave 6KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

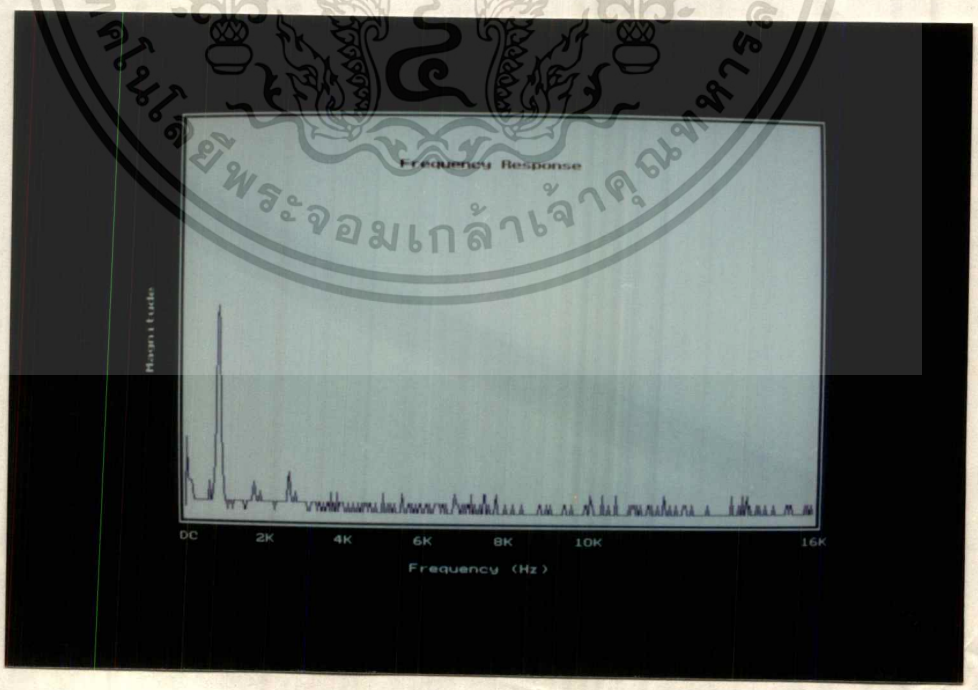


รูปที่ 7.18 รูปสเปกโตรแกรมสัญญาณ Square Wave 6KHz ขนาด 130mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

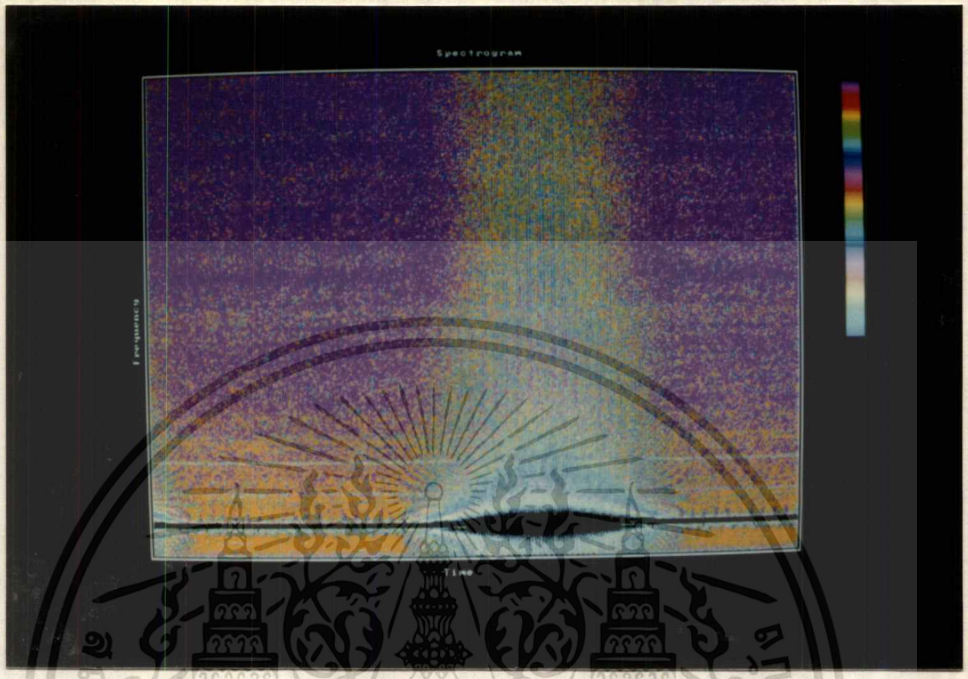


รูปที่ 7.19 รูปสัญญาณ Sine Wave 1KHz ขนาด 2V



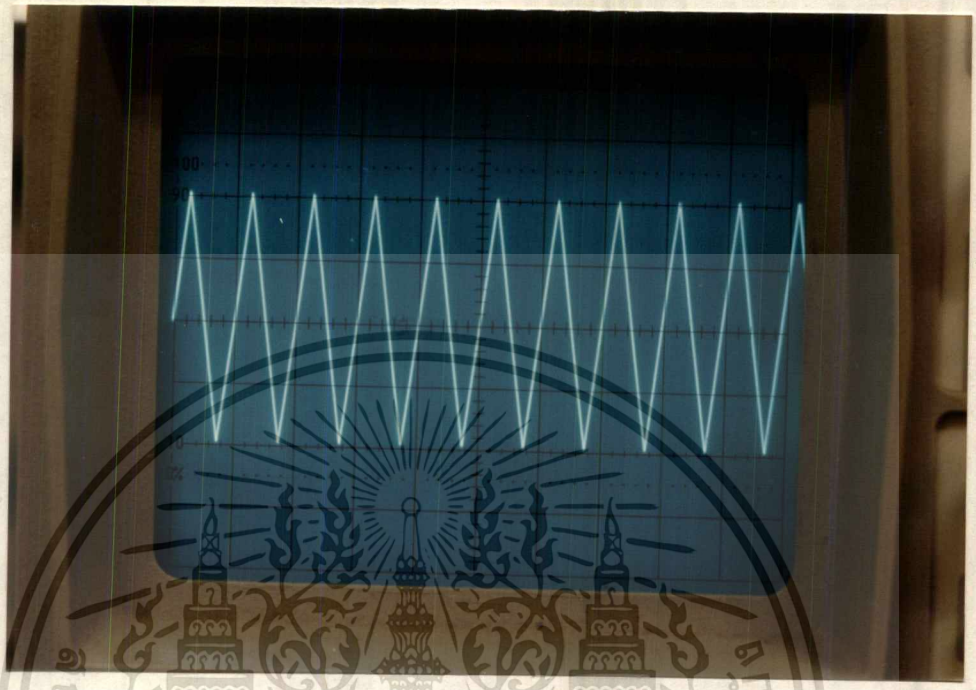
รูปที่ 7.20 รูปสเปกตรัมสัญญาณ Sine Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

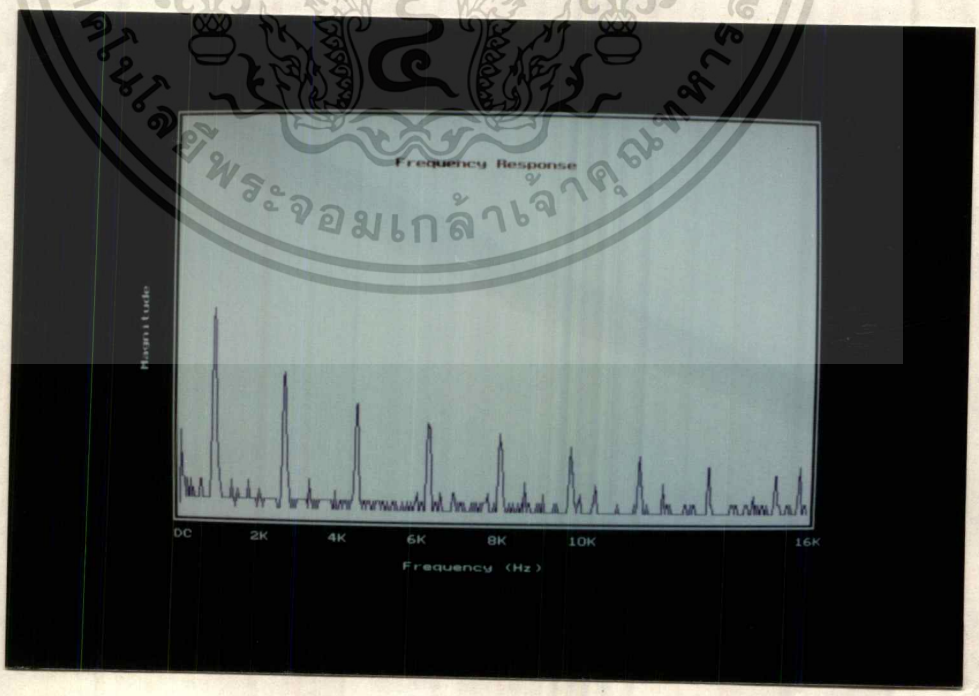


รูปที่ 7.21 รูปสเปกโตรแกรมสัญญาณ Sine Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

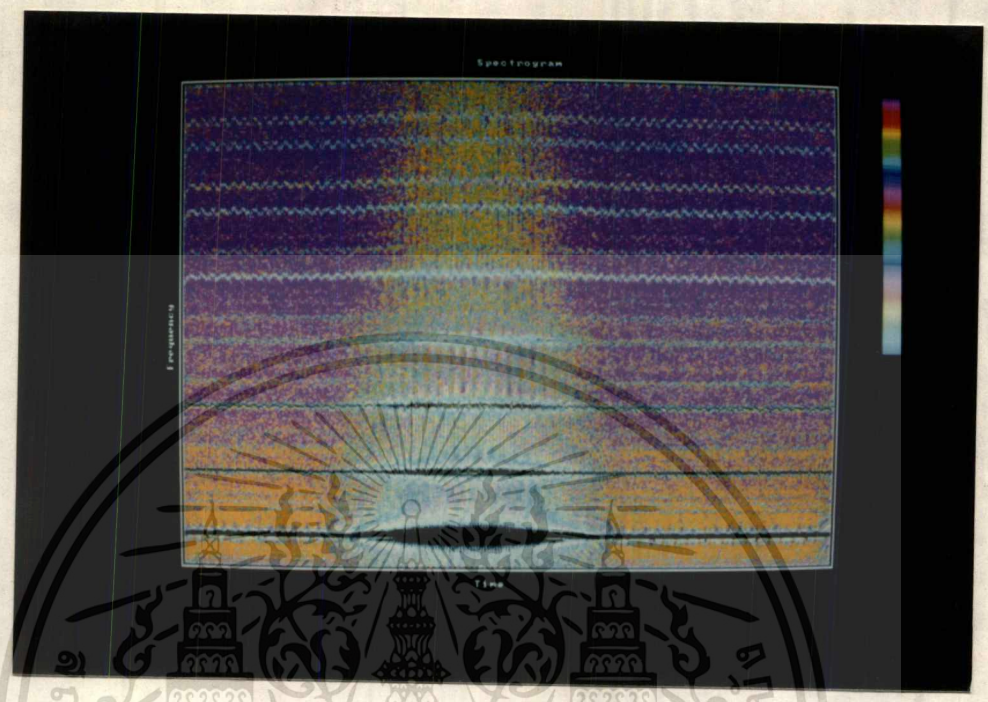


รูปที่ 7.22 รูปสัญญาณ Triangle Wave 1KHz ขนาด 2V



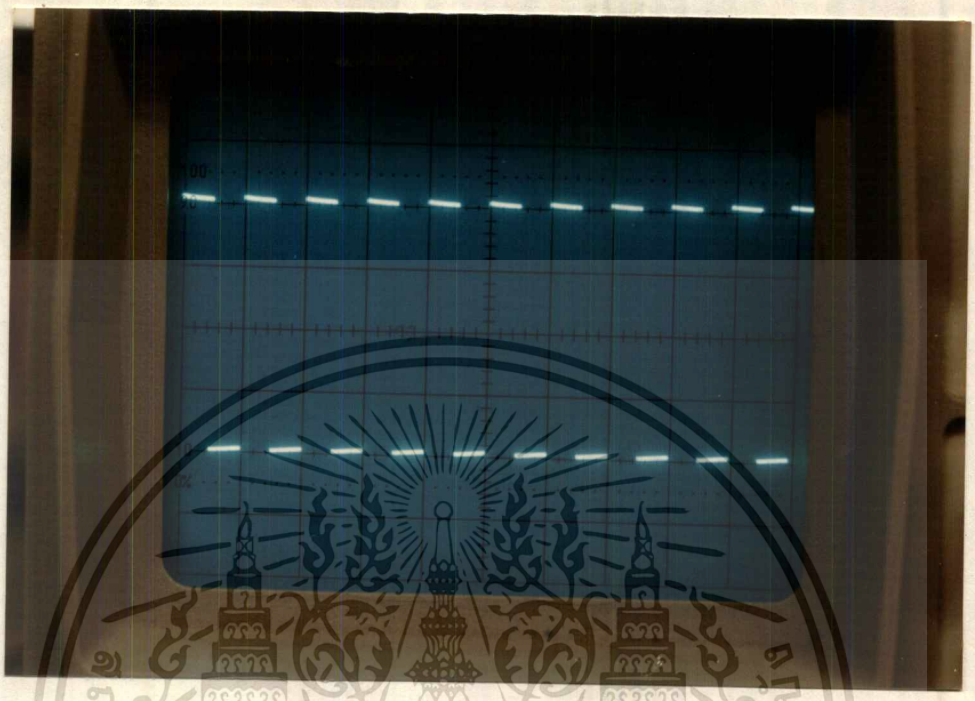
รูปที่ 7.23 รูปสเปกตรัมสัญญาณ Triangle Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

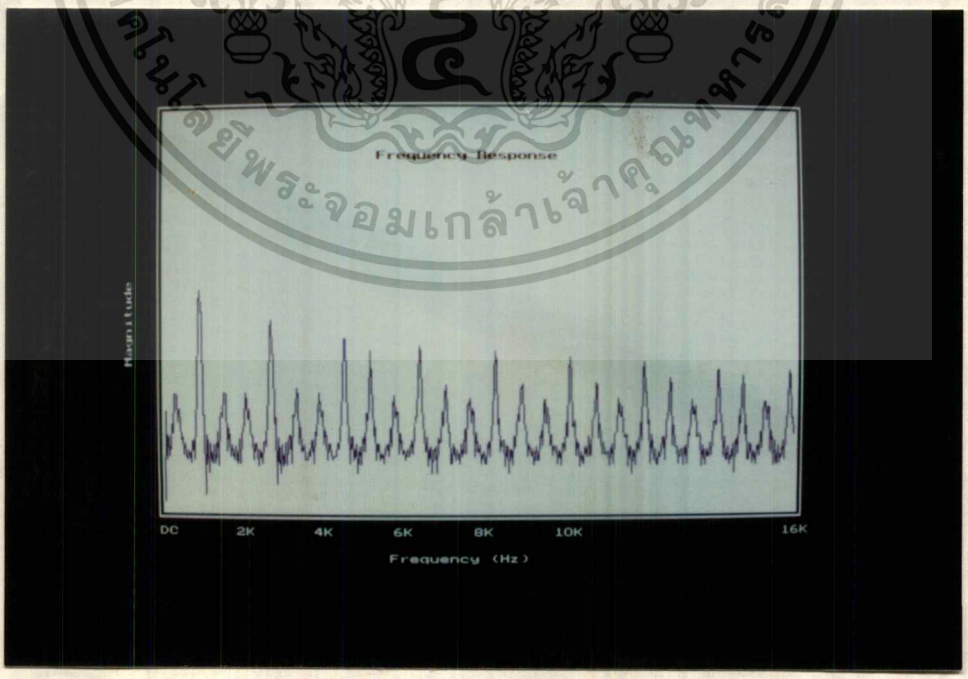


รูปที่ 7.24 รูปสเปกโตรแกรมสัญญาณ Triangle Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

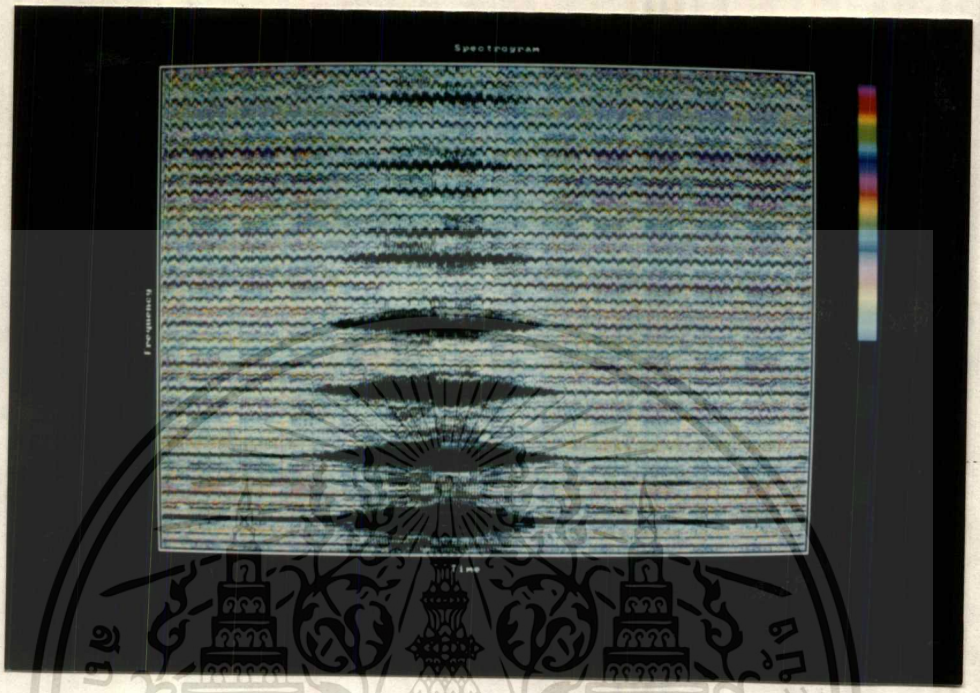


รูปที่ 7.25 รูปสัญญาณ Square Wave 1KHz ขนาด 2V



รูปที่ 7.26 รูปสเปกตรัมสัญญาณ Square Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.27 รูปสเปกโตรแกรมสัญญาณ Square Wave 1KHz ขนาด 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลองเราได้ทำการทดลองพล็อตสเปกตรัม และสเปกโตรแกรมโดยมีการเปลี่ยนค่าความถี่ และเปลี่ยนค่าแรงดันดังนี้

1. ใช้ค่าความถี่ 1 KHz ที่แรงดันขนาด 130 mV
2. ใช้ค่าความถี่ 6 KHz ที่แรงดันขนาด 130 mV
3. ใช้ค่าความถี่ 1 KHz ที่แรงดันขนาด 2 V

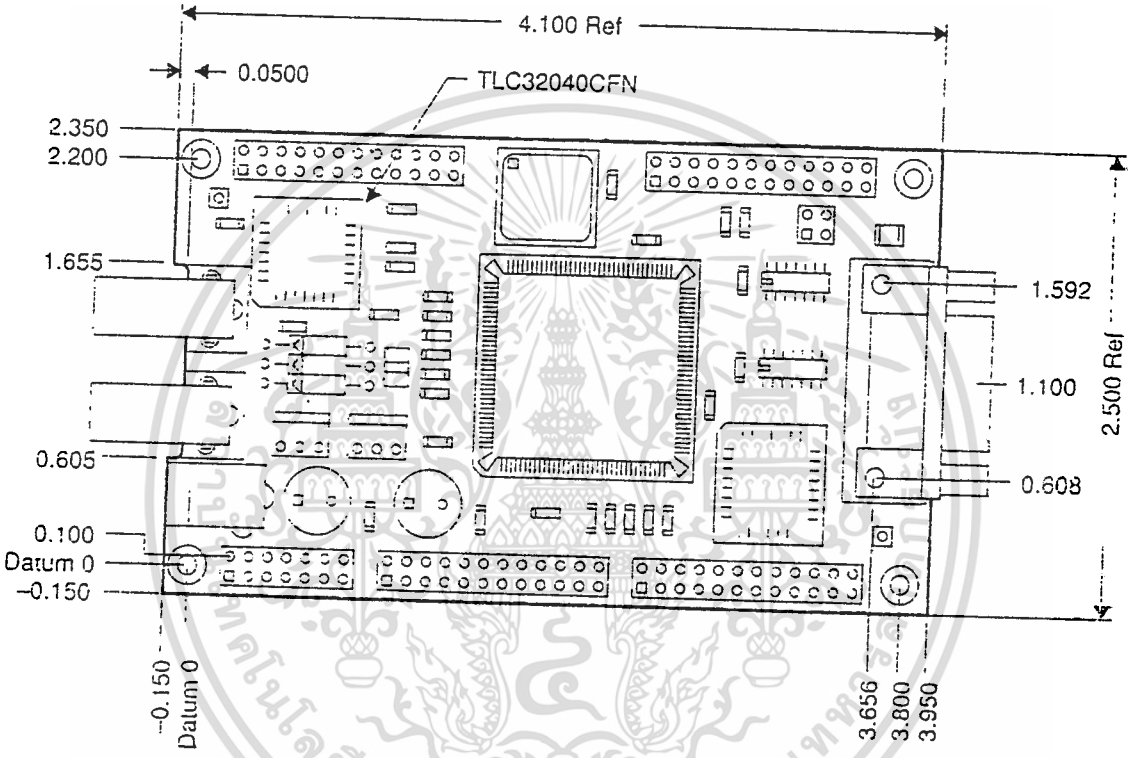
ซึ่งเราได้ทำการเปรียบเทียบกัน 3 สัญญาณ คือสัญญาณ Sine Wave, สัญญาณ Triangle Wave, และสัญญาณ Square Wave โดยผลการทดลองที่ออกมาได้ใกล้เคียงกับความเป็นจริงมาก แต่ก็ยังมีข้อผิดพลาดอีกเล็กน้อยในการนำผลออกแสดงทางจอคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

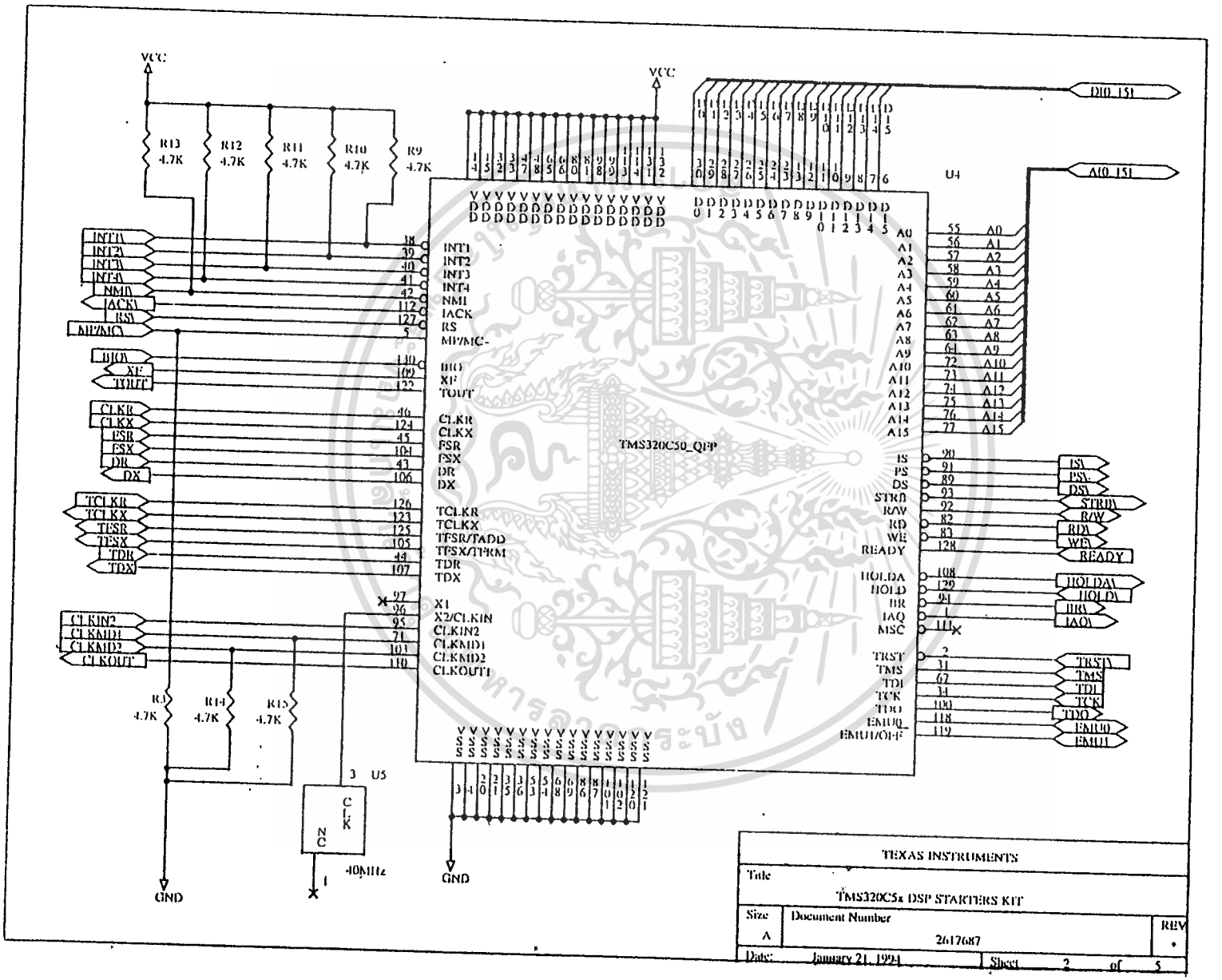
ภาคผนวก ก

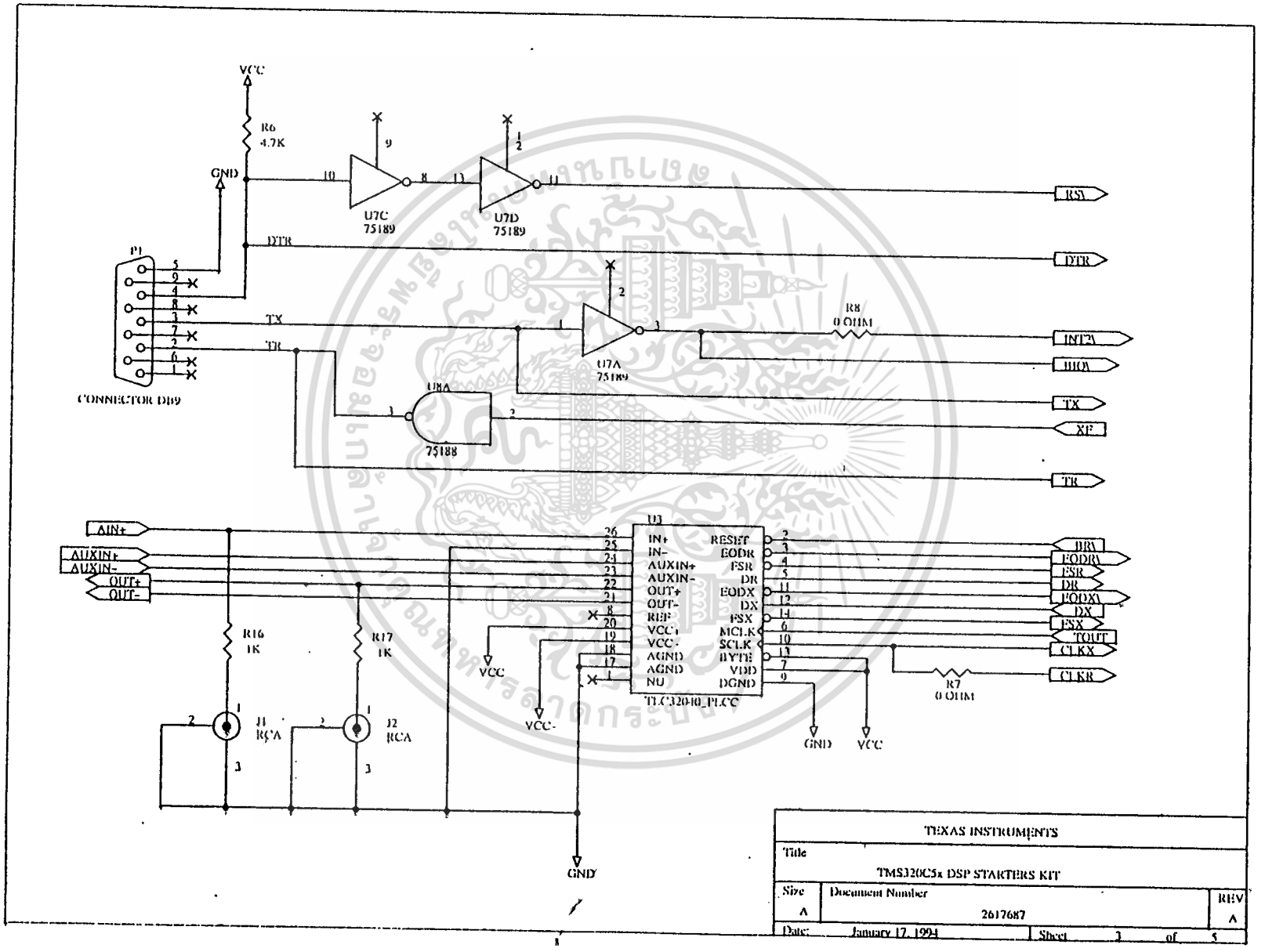
รูปวงจรแสดงส่วนต่าง ๆ บนบอร์ด DSK และ Schematic Diagrams

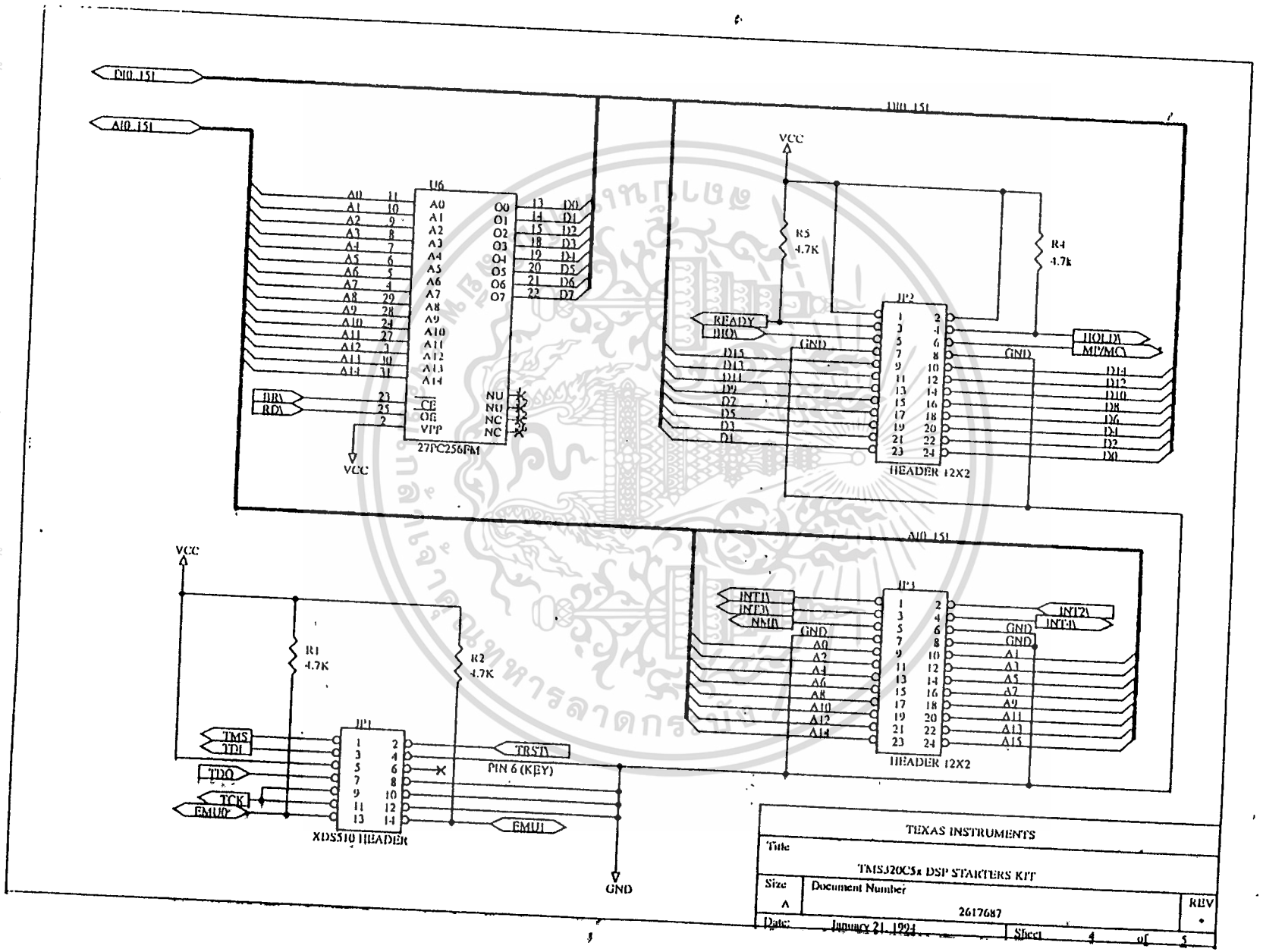


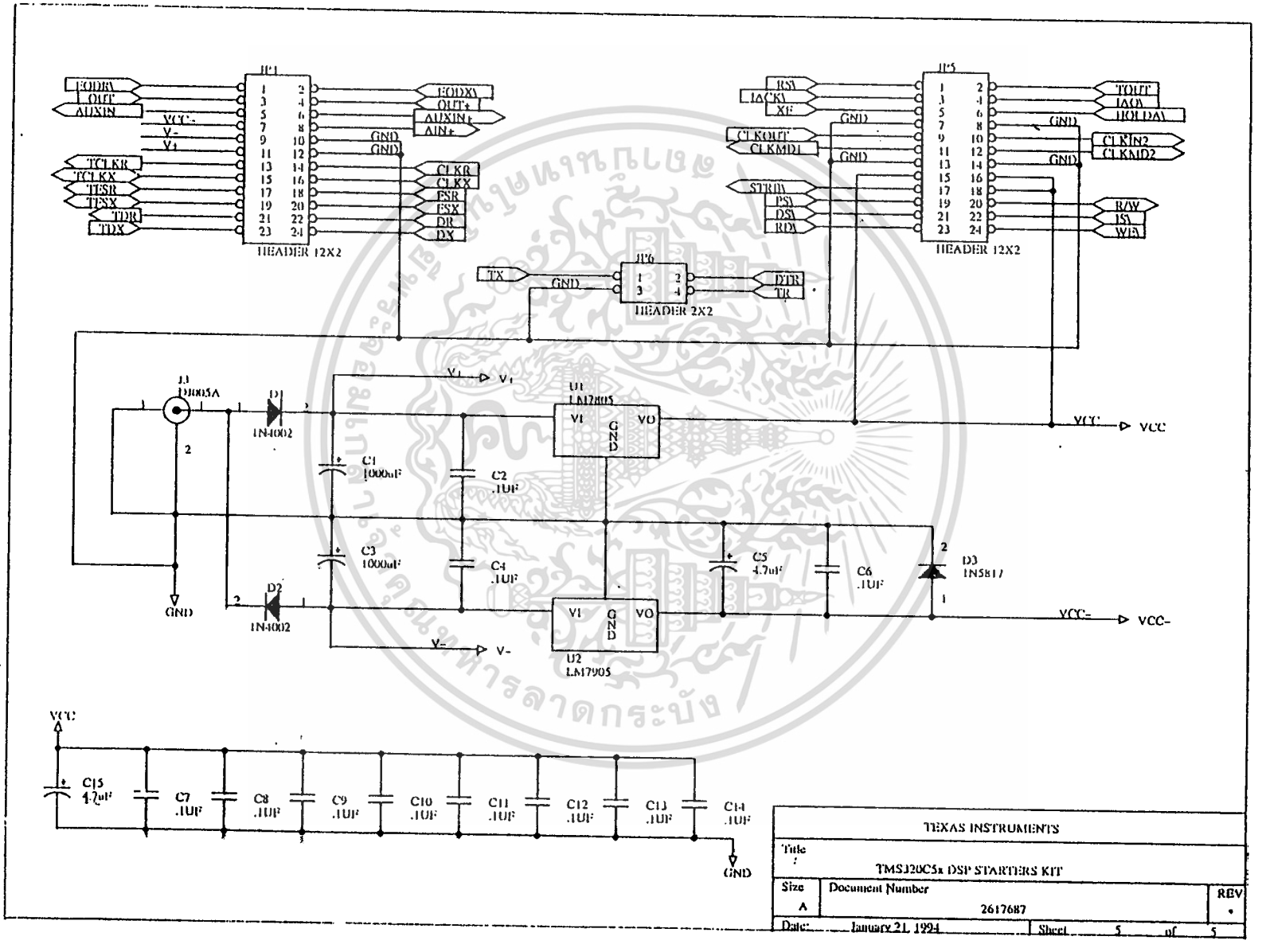
Note: Dimensions are in inches.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
R13 4.7K
R12 4.7K
R11 4.7K
R10 4.7K
R9 4.7K
U4
TMS320C50_QFP
INT1
INT2
INT3
INT4
NMI
IACK
RS
MP/MC-
BIO
XF
TOUT
CLKR
CLKX
FSR
FSX
DR
DX
TCLKR
TCLKX
TFSR/TADD
TFSX/TRM
TDR
TDX
X1
X2/CLKIN
CLKIN2
CLKMD1
CLKMD2
CLKOUT
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
IS
PS
DS
STRB
R/W
RD
WE
READY
HOLDA
HOLD
BR
IAQ
MSC
TRST
TMS
TDI
TCK
TDO
EMU0
EMU1/OFF
D[0..15]
A[0..15]
IS\
PS\
DS\
STRB\
R/W
RD\
WE\
READY
HOLDA\
HOLD\
BR\
IAQ\
TRST\
TMS
TDI
TCK
TDO
EMU0
EMU1
R3 4.7K
R14 4.7K
R15 4.7K
U5
NC
CLK
40MHz
GND
TEXAS INSTRUMENTS
TMS320C5x DSP STARTERS KIT
Size A
Document Number 2617687
REV *
Date: January 21, 1994
Sheet 2 of 5


VCC
R6
4.7K
GND
U7C
75189
U7D
75189
P1
CONNECTOR DB9
DTR
TX
TR
U8A
75188
U7A
75189
R8
0 OHM
RSV
DTR
INTV
BIO
TX
XF
TR
AIN+
AUXIN+
AUXIN-
OUT+
OUT-
U3
IN+
IN-
AUXIN+
AUXIN-
OUT+
OUT-
REF
VCC+
VCC-
AGND
AGND
NU
RESET
EODR
FSR
DR
EODX
DX
FSX
MCLK
SCLK
BYTE
VDD
DGND
TLC32040_PLCC
BR\
EODR\
FSR
DR
EODX\
DX
FSX
TOUT
CLKX
CLKR
R16
1K
R17
1K
J1
RCA
J2
RCA
R7
0 OHM
VCC
VCC-
GND
GND
TEXAS INSTRUMENTS
Title
TMS320C5x DSP STARTERS KIT
Size
A
Document Number
2617687
REV
A
Date: January 17, 1994
Sheet 3 of 5


D[0..15]
A[0..15]
U6
27PC256FM
HEADER 12X2
JP2
R5
4.7K
R4
4.7k
VCC
GND
READY
HOLD\
MP/MC\
JP3
INT1\
INT2\
NMI\
R1
4.7K
R2
4.7K
JP1
TMS
TDI
TDO
TCK
EMU0\
TRST\
PIN 6 (KEY)
EMU1\
XDS510 HEADER
TEXAS INSTRUMENTS
TMS320C5x DSP STARTERS KIT
Size
A
Document Number
2617687
REV
Date: January 21, 1994
Sheet 4 of 5


JP1
HEADER 12X2
JP5
HEADER 12X2
JP6
HEADER 2X2
J1
DB005A
D1
1N4002
D2
1N4002
D3
1N5817
U1
LM7805
U2
LM7905
C1 1000uF
C2 .1UF
C3 1000uF
C4 .1UF
C5 4.7uF
C6 .1UF
C15 4.7uF
C7 .1UF
C8 .1UF
C9 .1UF
C10 .1UF
C11 .1UF
C12 .1UF
C13 .1UF
C14 .1UF
GND
VCC
V-
TEXAS INSTRUMENTS
TMSJ20C5x DSP STARTERS KIT
Size A
Document Number 2617687
REV *
Date: January 21, 1994
Sheet 5 of 5

ภาคผนวก ข.

คู่มือ IC 74LS373 และ Schematic Diagrams

54373/74373 Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF
T.I	SN54S373	J ①									SN54LS373	J ①													
	SN74S373	J ①N①									SN74LS373	J ①N①													
FAIRCHILD																									
MOTOROLA																									
N. S. C.																									
PHILIPS																									
SIGNETICS																									
SIEMENS																									
FUJITSU																									
HITACHI																									
MITSUBISHI																									
NEC																									
TOSHIBA																									

Electrical Characteristics SN54LS373/SN74LS373

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

recommend operating conditions

		SN54LS373			SN74LS373			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				−1			−2.6	mA
High-level output voltage, V_{OH}				5.5			5.5	V
Pulse width, t_w	Clock enable high	15			15			ns
	Clock enable high	15			15			
Setup time, t_{SU}	TUP	0↓			0↓			ns
Hold time, t_{hold}		10↓			10↓			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

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

PARAMETER		TEST CONDITIONS‡		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max, I_{OH} = MAX		2.4	3.1		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max	I_{OL} = 24mA		0.35	0.5	V
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 2.7V				20	μA
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 0.4V				−20	μA
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				−0.4	mA
I_{OS}	Short-circuit output current §	V_{CC} = MAX		−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX, Output control at 4.5V	'LS373		24	40	mA

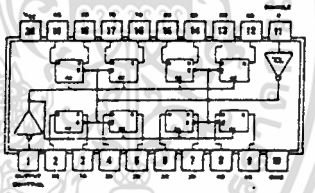
switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}			C_L = 45pF, R_L = 667Ω, See Notes 2 and 3				MHz
t_{PLH}	Data	Any Q			12	18	ns
t_{PHL}					12	18	
t_{PLH}	Clock or enable	Any Q			20	30	ns
t_{PHL}					18	30	
t_{PZH}	Output Control	Any Q			15	28	ns
t_{PZL}					25	36	
t_{PHZ}	Output Control	Any Q	C_L = 5pF, R_L = 667Ω, See Note 3		12	20	ns
t_{PLZ}					15	25	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
§ Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
3. See load circuits and waveforms on page 3-11.
f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
t_{PZH} = output enable time to high level
t_{PZL} = output enable time to low level

Pin Assignments (Top View)



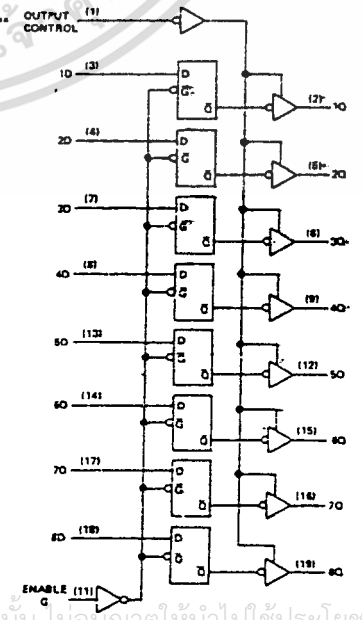
SN54LS373 (J) SN74LS373 (J, N)
SN54S373 (J) SN74S373 (J, N)

'LS373, 'S373 FUNCTION TABLE

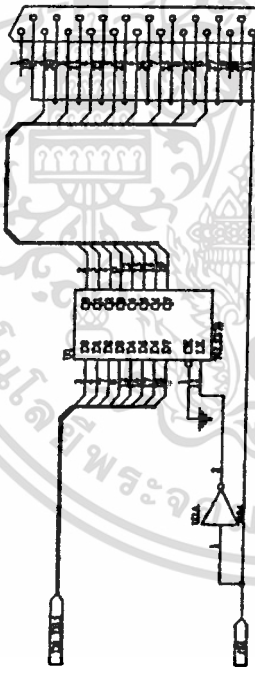
OUTPUT CONTROL	ENABLE G	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

'LS373, 'S373 TRANSPARENT LATCHES

'LS373, 'S373 TRANSPARENT LATCHES



ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย
เอกสารนี้ใช้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ห้ามดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานโครงงานฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือเป็นอย่างดีจาก อาจารย์เทอดศักดิ์ ลิ่วหาทอง ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน และคณาจารย์ท่านอื่น ๆ ที่กรุณาให้ยืมเอกสารตำราต่าง ๆ ที่เกี่ยวข้องกับการทำโครงงาน รวมทั้งการอนุญาตให้ใช้เครื่องมืออุปกรณ์การทดลองต่าง ๆ ห้องทดลอง รวมถึงการให้บริการของห้องคอมพิวเตอร์ของภาคด้วย พร้อมทั้งกรุณาให้คำแนะนำอันมีค่าและเป็นประโยชน์อย่างมาก อีกทั้งยังควบคุมดูแลตลอดระยะเวลาที่ได้ทำโครงงาน ทางกลุ่มนักศึกษาที่จัดทำโครงงานจึงขอขอบพระคุณคณะอาจารย์ทุกท่านเป็นอย่างสูง ที่ให้ความช่วยเหลือและสนับสนุนการทำโครงงานนี้ และท้ายนี้ขอขอบคุณเพื่อน ๆ ที่ได้ช่วยเหลือสนับสนุน และเป็นกำลังใจให้โครงงานนี้สำเร็จลุล่วงด้วยดีไว้ ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1.ดร.ไพรัช ธัชยพงษ์, "การประมวลผลสัญญาณดิจิตอล",ศูนย์เทคโนโลยิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์แลเทคโนโลยีแห่งชาติ กระทรวงวิทยาศาสตร์เทคโนโลยี และสิ่งแวดล้อม , 2535

2. Roman Kuc, "Introduction to Digital Signal Processing",McGraw-Hill, Singapore, 1982

3. Texas Instruments, " TMS320C5x User's Guide ",1993

4. Texas Instruments, " TMS320C5x DSP Starter Kit User's Guide ", 1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้