



การแสดงผลสเปกตรัมของเสียงโดยใช้บอร์ด DSP
A RECEIVER VOICE SPECTRAL DISPLAY BASED-DSP



โดย

นาย ชาติ ไชยันคร รหัส 37013046

นาย ดำเวง ชื่นชม รหัส 37013047

อาจารย์ที่ปรึกษา

อ. เกวียงไกร วงศ์โรจนภรณ์

วัน เดือน ปี..... 18 ค.ค ๒5๕๐

เลขทะเบียน..... 03๗๑๒๐

เลขเรียกหนังสือ..... T.3๑๐๑๑ ส ๖4๒๑

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

037320

ปริญญาโทบริหารการศึกษา 2539 ภาคเรียนที่ 2

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การแสดงผลสเปกตรัมของเสียงโดยใช้บอร์ด DSP

A RECEIVER VOICE SPECTRAL DISPLAY BASED-DSP

ผู้จัดทำ

1. นาย สาทิต ไชตยันดร 37013046

2. นาย สำเริง ชื่นชม 37013047


.....อาจารย์ที่ปรึกษา
(อาจารย์ เกียรติกร วงศ์โรจนารณ์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลสเปกตรัมของเสียงโดยใช้บอร์ด DSP
A RECEIVER VOICE SPECTRAL DISPLAY BASED-DSP

โดย นาย สาธิต โชตยันดร 37013046
นาย สำเริง ชื่นรม 37013047

อาจารย์ที่ปรึกษา

อ. เกียรติกร วงศ์โรจนกรณ์

บทคัดย่อ

ปริญญานิพนธ์นี้ ได้ทำการประยุกต์ใช้ บอร์ด DSP (Digital Signal Processing) เพื่อทำการแสดงผลสเปกตรัมของเสียง ซึ่งบอร์ด DSP นี้ จะใช้ไมโครโปรเซสเซอร์เบอร์ TMS 320C26 ในการทำการแปลงฟูเรียร์อย่างรวดเร็ว (Fast Fourier Transform) และวิเคราะห์สเปกตรัมของเสียง โดยการสั่งงานด้วยโปรแกรมคอมพิวเตอร์ และส่งข้อมูลกลับมาแสดงผลที่หน้าจอของคอมพิวเตอร์ ซึ่งจะใช้ RS 232C ในการเชื่อมต่อระหว่างบอร์ด DSP กับ คอมพิวเตอร์

ABSTRACT

This thesis use Based-DSP (Based-Digital Signal Processing) as a receiver voice spectral display , that TMS 320C26 was used as a microprocessor on DSP board. It will be computes FFT (Fast Fourier Transform) and analyses voice spectral. This function works by computer programming and shows result at monitor which uses RS 232C link to communicate.

สารบัญ

	หน้า
บทที่ 1 บทนำ.....	1
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 การสื่อสารภายในคอมพิวเตอร์.....	3
2.2 TMS 320C26.....	20
2.3 ฟูเรียส(Fourier).....	33
2.4 ทฤษฎีของการแปลงสัญญาณ.....	46
บทที่ 3 แนวทางการออกแบบและการสร้าง.....	56
3.1 TMS 320C26 DSP Starter Kit.....	56
3.2 ขั้นตอนและแผนผังการทำ FFT.....	60
บทที่ 4 การทดลองและผลการทดลอง.....	63
4.1 การทดลองที่ 1.....	63
4.2 การทดลองที่ 2.....	64
4.3 การทดลองที่ 3.....	64
4.4 การทดลองที่ 4.....	64
4.5 การทดลองที่ 5.....	64
4.6 ผลการทดลองของทั้ง 5 การทดลอง.....	64
บทที่ 5 บทวิจารณ์และบทสรุป.....	70
ภาคผนวก	
ก. แสดงวงจรในส่วนต่างๆ	
ข. รายละเอียดของอุปกรณ์ที่สำคัญ	
ค. แสดง List Programs	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

สารบัญรูป

รูปที่		หน้า
1.1	แสดงการประมวลผลเฉพาะอย่าง	1
1.2	แสดงการใช้ไมโครโปรเซสเซอร์ประมวลผล	1
1.3	แสดงส่วนประกอบของ DSP	2
.....		
2.1.1	การติดต่อสื่อสารข้อมูลในระบบคอมพิวเตอร์	3
2.1.2	แสดงการชิงโครโมรีระหว่างข้อมูลกับสัญญาณนาฬิกา	4
2.1.3	การส่งข้อมูลแบบขนาน	5
2.1.4	การส่งข้อมูลแบบอนุกรม	6
2.1.5	รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม	7
2.1.6	ฟอร์มแมตการสื่อสารแบบอะซิงโครนัส	7
2.1.7	รูปแบบของข้อมูลหนึ่งตัวอักษร	10
2.1.8	การใช้ RS 232C เชื่อมต่ออุปกรณ์	12
2.1.9	ย่านของแรงดันไฟฟ้าใช้ในสัญญาณ RS 232C	13
2.1.10	การกำหนดของข้อต่อ RS 232C	13
2.1.11	แสดงปัญหาความผิดพลาดทางสัญญาณไฟฟ้า	16
2.1.12	ผลของระดับสัญญาณกราวด์ที่แตกต่างกัน	17
2.1.13	RS 422 วงจรการเชื่อมต่อสัญญาณดิจิทัลแบบบาลานซ์	17
2.1.14	RS 423 วงจรการเชื่อมต่อสัญญาณดิจิทัลแบบไม่บาลานซ์	18
2.1.15	การถ่ายข้อมูลของ RS 232C ระหว่างไมโครคอมพิวเตอร์อย่างง่าย	19
2.1.16	การต่อไมโครคอมพิวเตอร์ผ่าน RS 232C แบบมี Hand Shake	19
2.2.1	โครงสร้างของ TMS 320C26	21
2.2.2	การจัดหน่วยความจำของ TMS 320C2x	23
2.2.3	โครงสร้างของ ARAU	24
2.2.4	Methods of Instruction Operand Addressing	27
2.2.5	Indirect Auxiliary Register Addressing Example	27
2.3.1	แถบความถี่ไม่ต่อเนื่อง	34
2.3.2	รูปคลื่นแบบสมมาตร	34
2.3.3	รูปคลื่นแบบไม่สมมาตร	35
2.3.4	รูปคลื่นสามเหลี่ยม	36

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.3.5 รูปคลื่นฟันปลา	36
2.3.6 Signal Flow Graph $N=8$	45
2.4.1 แสดงระบบที่มีการประมวลผลข้อมูลดิจิทัล	47
2.4.2 แสดงการสุ่มสัญญาณ	48
2.4.3 แสดงทรานสเฟอร์ฟังก์ชันของการควอนไทซ์ 4 บิต	49
2.4.4 แสดงความผิดพลาดของการควอนไทซ์	49
2.4.5 แสดงวงจรพื้นฐานของการแซมเปิลแอนด์โฮลด์	50
2.4.6 แสดงวงจรแซมเปิลแอนด์โฮลด์ที่ใกล้เคียงกับวงจรที่ใช้จริง	50
2.4.7 แสดงการตกของแรงดันที่โฮลด์ไว้	51
2.4.8 แสดงการเปลี่ยนสัญญาณจากวงจรแซมเปิลแอนด์โฮลด์ในทางอุดมคติ	51
2.4.9 แสดงทรานสเฟอร์ฟังก์ชันของ ดีเอซี ขนาด 3 บิต	52
2.4.10 แสดงบล็อกไดอะแกรมของ ดีเอซี	52
2.4.11 แสดงวงจร ดีเอซี ขนาด 4 บิต แบบ R-2R ladder	53
2.4.12 แสดงการแบ่งแรงดันจากแรงดันอ้างอิง	53
2.4.13 แสดงบล็อกไดอะแกรมของตัวแปรแบบซัคเซสซีฟ	54
2.4.14 แสดงไทม์มิงไดอะแกรมของ เอดีซี แบบ SAR	54
.....	
3.1 หน้าตาและขนาดของบอร์ด DSK	56
3.2 วงจรสมบูร์นของบอร์ด DSK	57
3.3 การต่อใช้งานบอร์ด DSK ร่วมกับออสซิลโลสโคป	58
3.4 ขั้นตอนการพัฒนาโปรแกรม	59
3.5 An In-place DIT FFT With In-Order Inputs but Bit-Reversed Output	61
.....	
4.1 แสดงการต่ออุปกรณ์ที่ใช้ในการทดลองทั้ง 5 การทดลอง	63
4.2 รูปแสดงผลสเปกตรัมของสัญญาณ Sine Wave ที่ความถี่ 500 Hz	64
4.3 รูปแสดงผลสเปกตรัมของสัญญาณ Sine Wave ที่ความถี่ 1 kHz	65
4.4 รูปแสดงผลสเปกตรัมของสัญญาณ Sine Wave ที่ความถี่ 10 kHz	65
4.5 รูปแสดงผลสเปกตรัมของสัญญาณ Sine Wave ที่ความถี่ 20 kHz	66
4.6 รูปแสดงผลสเปกตรัมของสัญญาณ Square Wave ที่ความถี่ 1 kHz	66
4.7 รูปแสดงผลสเปกตรัมของสัญญาณ Square Wave ที่ความถี่ 10 kHz	67

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.8	รูปแสดงผลสเปกตรัมของสัญญาณ Sawtooth Wave ที่ความถี่ 1 kHz	67
4.9	รูปแสดงผลสเปกตรัมของสัญญาณ Sawtooth Wave ที่ความถี่ 10 kHz	68
4.10	รูปแสดงผลสเปกตรัมของสัญญาณ Continuous signal (Input Music)	68
4.11	รูปแสดงผลสเปกตรัมของสัญญาณ Single Pulse Signal (Speech)	69



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1 ก.	มาตรฐานของการใช้แรงดันไฟฟ้า	12
2.1 ข.	คุณลักษณะโดยย่อของสัญญาณ RS 232C	16
2.2 ก.	Interrupt Locations และ Priorities	26
2.2 ข.	การทำงานและลักษณะการอ้างตำแหน่งของ TMS 320C26	28
2.2 ค.	ชุดคำสั่งของ TMS 320C26	29-32
2.3 ก.	เปรียบเทียบจำนวนการคูณค่าเชิงซ้อนระหว่าง DFT กับ FFT	41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

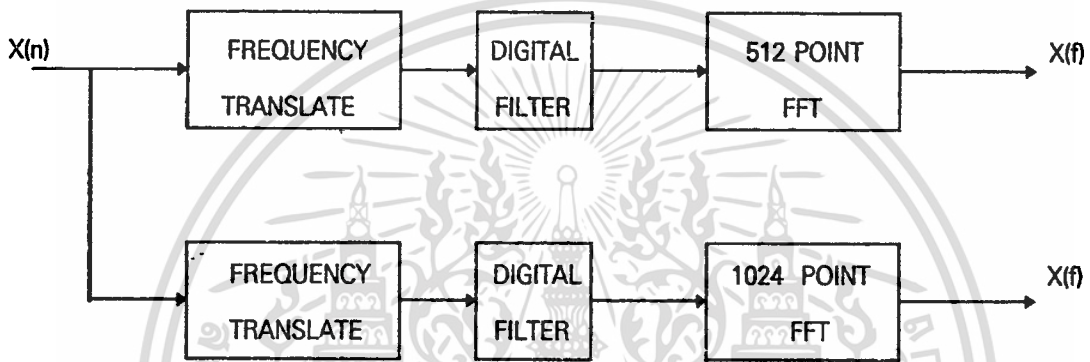
บทที่ 1

บทนำ

ดิจิทัลซิกแนลโปรเซสซิง ก็คือการเปลี่ยนรูปสัญญาณ โดยใช้การคำนวณทางคณิตศาสตร์ ด้วยดิจิทัลซิกแนลโปรเซสเซอร์ ซึ่งจะเห็นได้ว่าเราสามารถจะนำสัญญาณนั้นใส่ลงในสมการคณิตศาสตร์ แล้วได้ผลลัพธ์ออกมาโดยตรงเลยทีเดียว

ระบบดิจิทัลซิกแนลโปรเซสซิง จะมีอยู่ 2 รูปแบบ ใหญ่ๆ ดังนี้ คือ

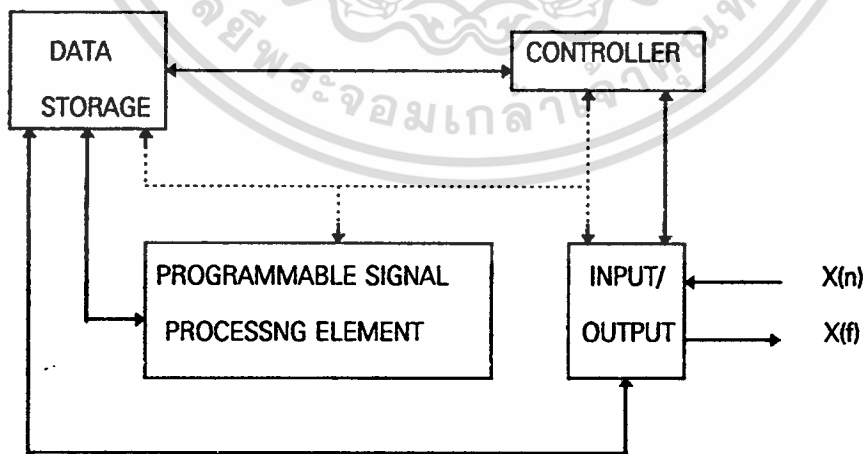
1.1 ระบบที่ออกแบบมาเพื่อให้ทำงานอย่างหนึ่งโดยเฉพาะ แสดงดังรูปที่ 1.1



รูปที่ 1.1 แสดงการประมวลผลเฉพาะอย่าง

1.2 ระบบที่ใช้ไมโครโปรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่ไว้ ดังรูปที่

1.2



รูปที่ 1.2 แสดงการใช้ไมโครโปรเซสเซอร์ประมวลผล

ความก้าวหน้าทางอิเล็กทรอนิกส์ครั้งใหญ่เกิดขึ้น เมื่อได้มีการพัฒนาสร้างดิจิตอลซิกแนลโปรเซสเซอร์ความเร็วสูง (High - Speed Signal Processor) ซึ่งในปัจจุบันนี้ใช้กันอย่างแพร่หลายและไม่แพงมากนัก

ดิจิตอลซิกแนลโปรเซสเซอร์ จำเป็นต้องมีความเร็วที่สูง และถูกออกแบบมาโดยเฉพาะให้สามารถทำงานได้อย่างมีประสิทธิภาพ ทั้งนี้ก็เพื่อการประมวลผลแบบเวลาจริง (Real time) นั่นเอง การประมวลผลแบบเวลาจริงคือ เมื่อเราใส่สัญญาณอินพุตเข้าไป ก็จะได้สัญญาณเอาต์พุตออกมาเลย โดยการมีกรหน่วงเวลาเกิดขึ้นน้อยมาก

ด้วยความสามารถของดิจิตอลซิกแนลโปรเซสซิ่ง จึงได้นำไปใช้งานในรูปแบบต่างๆ ซึ่งให้ผลลัพธ์ที่ดีกว่าใช้วงจรรอนาล็อก การออกแบบวงจรทางอนาล็อกนั้นก็ยังน่าเบื่อทั้งวงจรหนึ่งๆ จะมีความสามารถทำงานได้อย่างเดียวเท่านั้น แต่ถ้าต้องการให้ทำงานอื่นๆ ก็จะต้องเปลี่ยนอุปกรณ์และจะต้องเดินสายกันใหม่ นอกจากนี้แล้วการทดลองวงจรรอนาล็อกอาจจะเป็นปัญหาได้เนื่องจาก อิมพีแดนซ์ของเครื่องมือวัดที่ต่อเข้าไปในวงจร ปัญหาพื้นฐานเหล่านี้ของวงจรรอนาล็อก สามารถแก้ไขได้โดยการใช้ ดิจิตอลซิกแนลโปรเซสซิ่ง ข้อได้เปรียบอีกประการของการใช้ ดิจิตอลซิกแนลโปรเซสซิ่ง แทนวงจรรอนาล็อก คือ การเปลี่ยนแปลงคุณสมบัติของการประมวลผล สามารถทำได้โดยง่ายซึ่งจะทำให้เราสามารถทดลองในทุกทางที่เป็นไปได้ เพื่อเลือกเอาผลที่ดีที่สุดที่เราพอใจ

โดยทั่วไปแล้ว ดิจิตอลซิกแนลโปรเซสซิ่ง จะประกอบด้วยส่วนต่างๆดังรูปที่ 1.3



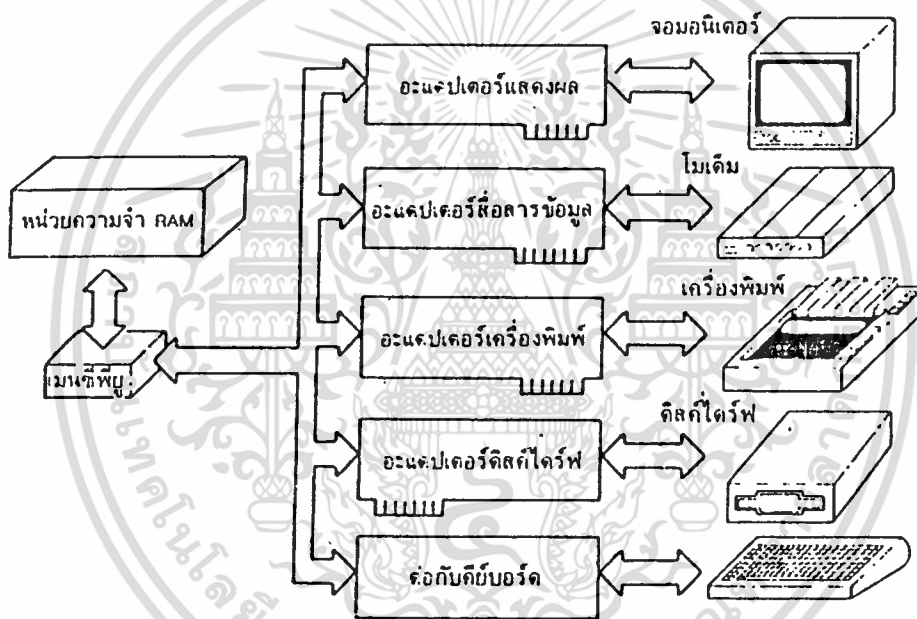
รูปที่ 1.3 แสดงส่วนประกอบของ DSP

สำหรับ โครงการนี้เราได้นำบอร์ด DSP สำเร็จรูปของบริษัท เท็กซัส อินสตรูเมนต์ TMS3200051 โดยที่บอร์ดรุ่นนี้จะใช้ ไมโครโปรเซสเซอร์เบอร์ TMS 320C26 เราจะนำบอร์ด DSP รุ่นนี้มาประยุกต์ใช้งานทำให้เครื่องคอมพิวเตอร์กลายเป็นเครื่องแสดงผลสเปกตรัม ในช่วงความถี่เสียง คือประมาณ 0-20 kHz นั่นเอง

บทที่ 2 ทฤษฎีและหลักการ

2.1 การสื่อสารภายในระบบคอมพิวเตอร์

การสื่อสารระหว่างคีย์บอร์ดกับเครื่อง อาจจะเป็นได้ทั้งแบบอนุกรมหรือแบบขนาน ตัวอย่างเครื่อง แอปเปิลทู ส่วนตัวคีย์บอร์ดติดต่อกับตัวเมนบอร์ดแบบขนาน ในแผงคีย์บอร์ดมีส่วนตัวควบคุมวงจรรหัสคีย์บอร์ด ตัวหนึ่งแปลงสัญญาณการกดคีย์ออกมาเป็นรหัส ASCII ทันทีเมื่อเรากดแป้นพิมพ์ รหัส ASCII ที่ออกมา 8 บิต จะถูกส่งไปยังเมนบอร์ดพร้อมกับสัญญาณพิเศษที่เรียกว่า *สโตรป* เพื่อเป็นการบอกให้รู้ว่า “มีข้อมูลว่าง มาให้แล้ว รับเอาไว้ด้วย” ฮาร์ดแวร์ในเมนบอร์ดจะคอยดูสัญญาณสโตรปอยู่ตลอดเวลาเพื่อคอยรับเอาข้อมูล เก็บไว้ก่อน เอาไว้ให้ซีพียูมาคอยเก็บเอาไปใช้ทีหลัง

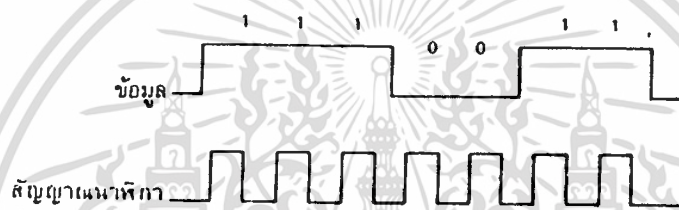


รูปที่ 2.1.1 การติดต่อสื่อสารข้อมูลในระบบคอมพิวเตอร์

สำหรับไมโครคอมพิวเตอร์ IBM PC ไม่ใช่นับว่าเป็นพิมพ์มีวงจรควบคุมคีย์บอร์ดคอยตรวจสอบการกดแป้นพิมพ์เหมือนกันเมื่อมีการกดคีย์ ส่วนของวงจรควบคุมคีย์บอร์ดตัวนี้ก็ส่งสัญญาณต่อไปยังเมนซีพียู แต่ส่งไปแบบอนุกรม แล้วก็ไม่ได้เข้ารหัส ASCII ส่งไป แต่เป็นรหัสประจำแป้นพิมพ์เท่านั้นเองว่าคีย์ไหนถูกกดไว้ให้เมนซีพียูไปแปลงเอาเอง มาถึงปัญหาว่าเมนซีพียูจะรู้ได้อย่างไรว่าแป้นพิมพ์ถูกกดและข้อมูลที่ส่งมาก็เป็นพัลส์ พัลส์ต่อเนื่องกันแบบอนุกรม ซีพียูจะรู้ได้อย่างไรว่าตัวไหนควรจะ เป็น “1” หรือควรจะ เป็น “0” ในสัญญาณที่ส่งมาจากคีย์บอร์ดของไอบีเอ็มจะมีสัญญาณจากอินเทอร์รัพต์มาด้วยเส้นหนึ่ง เป็นการขัดจังหวะการทำงานของซีพียู สัญญาณอันนี้จะทำให้ซีพียูกระโดดไปทำงาน ในรูทีนอันหนึ่งใน BIOS (โปรแกรมจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุท - เอาต์พุทพื้นฐานที่มากับเครื่อง อยู่ในรูปของ ROM หรือ PROM) ซึ่งทำหน้าที่รับข้อมูลจากคีย์บอร์ด โดยเฉพาะรับเสร็จก็เอาไปเก็บไว้ก่อนยังไม่เอาไปใช้งานทันทีรอให้โปรแกรมที่ทำงานจริง ๆ มาเรียกไปใช้งานอีกที (โดยที่ผู้ใช้เครื่องไม่รู้ตัว) คราวนี้ เมนซีพียูจะรู้ได้อย่างไรว่าคีย์บอร์ดส่งอะไรมา ในเมื่อสัญญาณที่ส่งมามีแต่พัลส์ที่สูง 5 โวลต์กับ 0 โวลต์ ต้องหาทางแยกให้ได้ว่าตรงไหนควรจะเป็นบิตที่ 1 หรือบิตที่ 2 หรือ 3 ... และจำเป็นจะต้องมีการชิงโครโนซ์หรือบอกกันให้รู้แน่ๆว่าตรงไหนคือข้อมูลคีย์บอร์ดก็จะมีสัญญาณอีกตัวหนึ่ง ที่เรียกว่าสัญญาณนาฬิกา สัญญาณนาฬิกาจะเป็นตัวบอกให้ตัวเปลี่ยนข้อมูลจากอนุกรมเป็นขนานบนซีพียู บอร์ดได้รู้ว่าตรงนั้นคือบิตที่เท่าไรของข้อมูล ถ้าเลือกเอาบิตตามใจแล้วละก็เป็นยุ่งแน่ ผู้ส่งกับผู้รับก็จะตีความ ไม่เหมือนกันลองดูรูปที่ 2.1.2 ถ้าหากไม่มีสัญญาณนาฬิกากำกับมาด้วยก็คงไม่รู้ว่าเป็นบิตไหนเป็น “0” หรือ “1” แน่



รูปที่ 2.1.2 แสดงการชิงโครโนซ์ระหว่างข้อมูลกับสัญญาณนาฬิกา

สำหรับคีย์บอร์ด เราก็ได้เห็นตัวอย่างของการสื่อสารทั้งแบบขนานและอนุกรมแล้ว ในเครื่องพิมพ์ก็ทำนองเดียวกัน เมนซีพียูจะติดต่อกับเครื่องพิมพ์ทางช่องสื่อสารอาจจะแบบอนุกรมและแบบขนาน ส่วนมากจะเป็นแบบขนาน ที่แตกต่างออกไปก็คือ ความเร็วในการพิมพ์ ยังห่างไกลกับความเร็วของซีพียูมากนัก ฉะนั้นจำเป็นจะต้องมีการโต้ตอบระหว่างเครื่องพิมพ์ การที่จำเป็นจะต้องมีการโต้ตอบกันก็เพื่อป้องกันไม่ให้สัญญาณหายไปไหน ซึ่งเรียกว่า Hand-shake. หรือการจับมือกันโต้ตอบกัน

จากการพิจารณาดูการทำงานกว้าง ๆ ของการติดต่อระหว่างซีพียูกับคีย์บอร์ดและซีพียูกับเครื่องพิมพ์ก็จะเห็นภาพของการสื่อสารทั้งสองแบบ คือแบบอนุกรมและแบบขนาน นอกเหนือจากการติดต่อภายในของเครื่องคอมพิวเตอร์แล้วบางครั้งไมโครคอมพิวเตอร์จำเป็นจะต้องติดต่อกับคอมพิวเตอร์ตัวอื่น เพื่อเพิ่มความสามารถของมันเองให้สูงขึ้นเป็นต้นว่า ไมโครคอมพิวเตอร์หลายๆตัว อาจจะสื่อสารเข้าหากันเป็นเน็ตเวิร์คเพื่อประโยชน์ในการใช้อุปกรณ์ที่จำเป็นต่างๆร่วมกัน และที่สำคัญก็คือ เอาไว้ใช้ข้อมูลที่เก็บไว้ในที่ต่างๆร่วมกันได้ คอมพิวเตอร์มีข้อจำกัดของมันเองหลายประการ เป็นต้นว่า ขนาดของความจำจำกัด ความเร็วใน

การทำงานจำกัด และอื่นๆ ข้อสำคัญก็คือ การติดต่อสื่อสารภายนอกโดยเฉพาะคอมพิวเตอร์เครื่องใหญ่ที่มีขนาดความจำสูงจึงจะเป็นการเพิ่มประสิทธิภาพการทำงานให้สูงขึ้น

วิธีการถ่ายโอนข้อมูล

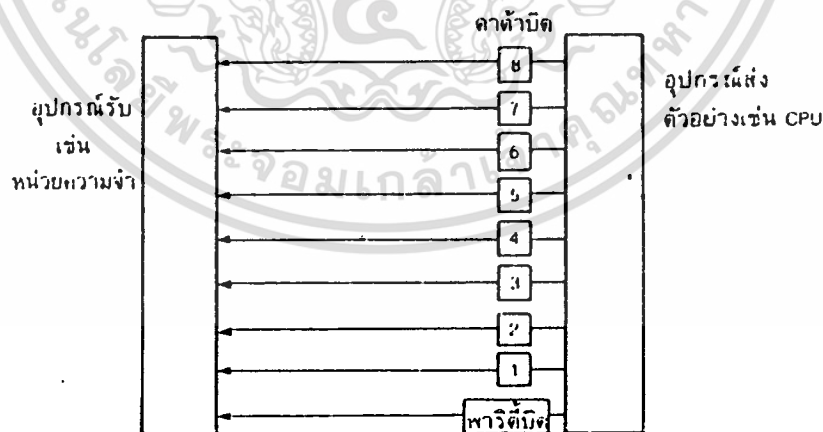
ก่อนที่จะกล่าวถึงการสื่อสารระหว่างเครื่อง เรามามองดูก่อนว่าสัญญาณที่ส่งออกมาจากเครื่อง และรับเข้าไปในเครื่องเป็นอย่างไร

1. การถ่ายโอนข้อมูลแบบขนาน

ลักษณะของการส่งข้อมูลขนาน ทำได้โดยการส่งข้อมูลออกมาทีละ 1 ไบต์ คือ 8 บิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ตัวกลางระหว่าง 2 เครื่องจะต้องมีช่องทางให้ข้อมูลเดินอย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสวิ่งมากกว่าจะเป็นตัวกลางชนิดอื่น เนื่องจากมีสัญญาณสูญหายไปกับความต้านทานของสาย ระยะทางระหว่างสองเครื่อง ไม่ควรจะเกิน 100 ฟุต ปัญหาที่เกิดขึ้นเนื่องจากระยะทางของสายมากกว่านี้ก็คือ ระดับของกราวด์ในทางไฟฟ้าที่จุดรับผิดไปจากจุดส่ง ทำให้เกิดการผิดพลาดในการรับสัญญาณลอจิกทางฝ่ายรับ

นอกจากสายที่เป็นทางเดินของข้อมูลแล้วอาจจะมีทางเดินของสัญญาณควบคุมอื่นๆ อีก เป็นต้นว่า บิตที่บอกพริตตี้ของสัญญาณ เพื่อเป็นการตรวจสอบความผิดพลาดของการรับสัญญาณที่ปลายทาง หรือที่สายควบคุมการโต้ตอบ หรือ แชนเช็กกิ้งดังที่ได้กล่าวมาแล้ว

จะเห็นว่าการส่งแบบขนานส่วนมากจะทำในระยะใกล้ๆ เนื่องจากจะต้องมีช่องทางเดินของสัญญาณมากกว่า 8 สาย และเป็นอุปกรณ์ที่ติดต่อกันเป็นแบบขนานกับคอมพิวเตอร์ก็จะเห็นว่าได้แก่เครื่องพิมพ์ดังกล่าวมาแล้ว

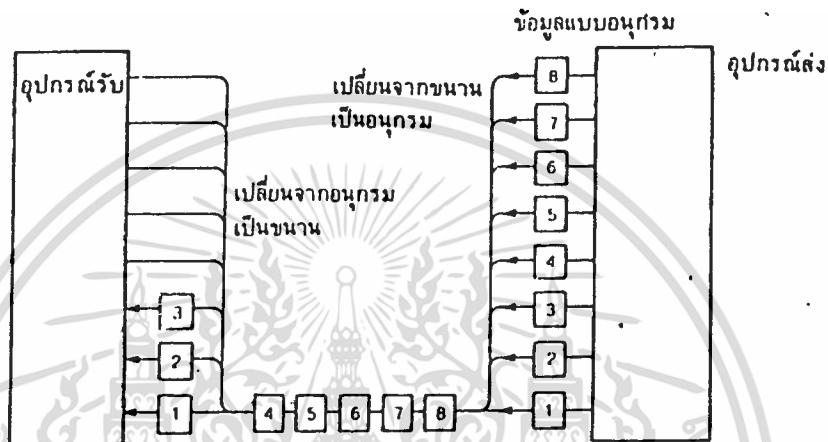


รูปที่ 2.1.3 การส่งข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การถ่ายโอนข้อมูลแบบอนุกรม

ในการถ่ายโอนข้อมูลแบบอนุกรม ข้อมูลถูกส่งออกมาทีละบิต ระหว่างจุดส่งและจุดรับจะเห็นว่าการส่งข้อมูลแบบนี้จะช้ากว่าแบบขนานที่กล่าวมาแล้วแน่นอน แล้วทำไมจะต้องมีการส่งแบบนี้ละ คำตอบก็คือตัวกลางการสื่อสารต้องการเพียงช่องเดียวหรือสายเพียงคู่เดียว ค่าใช้จ่ายในการสื่อกลางจะต้องถูกกว่าแบบขนานอย่างแน่นอน สำหรับการส่งระยะทางไกลๆ โดยเฉพาะเมื่อเรามีระบบการสื่อสารทางโทรศัพท์ไว้ใช้งานอยู่แล้วย่อมจะเป็นการประหยัดกว่าที่จะทำการติดต่อสื่อสารทีละ 8 ช่อง เพื่อการถ่ายโอนข้อมูล แบบขนานอย่างแน่นอน



รูปที่ 2.1.4 การส่งข้อมูลแบบอนุกรม

จากรูปที่ 2.1.4 แสดงให้เห็นการส่งข้อมูลแบบอนุกรมข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นอนุกรมเสียก่อนแล้วค่อยทยอยส่งออกไปยังจุดรับ ณ.ที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาทีละบิตให้เป็นสัญญาณแบบขนานซึ่งลงตัวพอดีนั่นคือ บิต 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่จะทำการแปลงสัญญาณอนุกรมทีละบิตให้ลงพอดีนั่นจำเป็นจะต้องมีกลไกที่เหมาะสม เพื่อป้องกันความผิดพลาดในการรับ กลไกที่วางนี้แบ่งเป็น 2 แบบ คือ

1. การสื่อสารแบบซิงโครนัส
2. การสื่อสารแบบอะซิงโครนัส

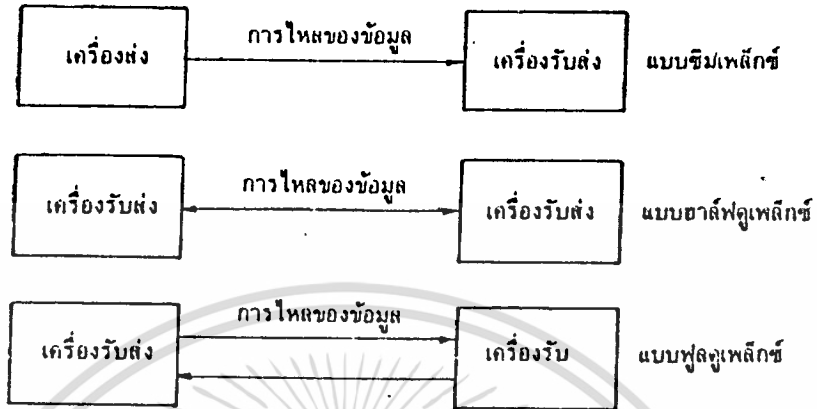
รูปแบบของการสื่อสารแบบอนุกรม

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบ ตามรูปที่ 2.1.5

1. **แบบซิมเพล็กซ์ (Simplex)** ข้อมูลส่งได้ในทางเดียวเท่านั้น บางครั้งก็เรียกว่า การส่งทิศทางเดียว (Undirection data bus)
2. **แบบฮาล์ฟดูเพล็กซ์ (Half duplex)** ข้อมูลสามารถส่งได้ทั้งสองสถานี แต่จะต้องผลัดกันส่งผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้
3. **แบบฟูลดูเพล็กซ์ (Full duplex)** ทั้งสองสถานีสามารถรับ และส่งได้ในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งแบบพหุคู่เพลิงท์และฮาล์ฟคู่เพลิงท์ ไม่ขึ้นอยู่กับจำนวนของสายในการติดต่อบางครั้งคำว่า ทูไวร์ (two wire) หรือ สองเส้น และ โฟร์ไวร์ (four wire) หรือ 4 เส้น ใช้ในการบรรยายถึงลักษณะการสื่อสารข้อมูลซึ่งอาจจะทำให้เข้าใจ และฮาล์ฟคู่เพลิงท์ สายโทรศัพท์ทั่วไปเป็นแบบ 2 เส้น ส่วนในสายที่เป็นแบบเช่า (Lease line) นั้นส่วนมากจะเป็น 4 เส้น



รูปที่ 2.1.5 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม

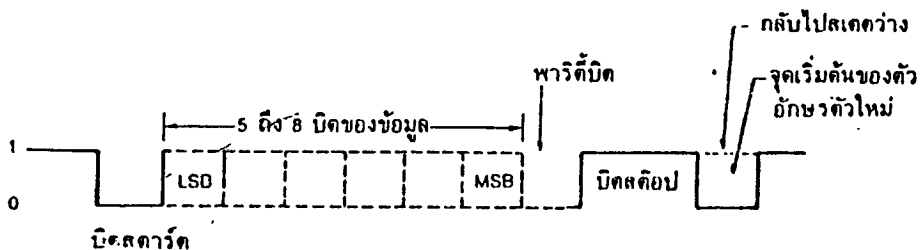
ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรม หน่วยวัดเป็นบิตต่อวินาที (bps) เป็นหน่วยที่บรรยายถึงการเปลี่ยนแปลงของสัญญาณใน 1 วินาที เรียกว่า บอดเรต (baud rate) หรืออัตราบอด หลายคนยังเข้าใจสับสนระหว่างอัตราบอดและอัตราบิต (bit rate) การเปลี่ยนแปลงของสัญญาณ 1 ครั้ง อาจจะแสดงถึงการส่งข้อมูลแบบอนุกรมมากกว่า 1 บิต ถ้าเขียนในรูปของสมการคณิตศาสตร์เราก็จะได้

$$\text{อัตราบิต (bit rate)} = \text{อัตราบอด (baud rate)} \times (\text{บิตใน 1 บอด})$$

การสื่อสารแบบอะซิงโครนัส

การส่งแบบอะซิงโครนัสนี้ พัฒนามาจากการส่งโทรพิมพ์ในสมัยก่อน ลักษณะของสัญญาณแสดงไว้ในรูปที่ 2.1.6 เพื่อกลไกในการรับส่งอย่างถูกต้อง สัญญาณอะซิงโครนัสจะประกอบด้วยบิตเริ่มต้นหรือบิตสตาร์ท (Start bit) และบิตสิ้นสุดหรือบิตสตอป (Stop bit)



รูปที่ 2.1.6 ฟอแมตการสื่อสารแบบอะซิงโครนัส

ขณะที่สถานะของการส่งเป็นแบบว่าง (Idle) คือ ยังไม่มีสัญญาณส่งออกจะมีแรงดันหรือกระแส (สัญญาณ) ตลอดเวลา เพื่อความแน่ใจว่าฝ่ายรับยังติดต่อกับฝ่ายส่ง เมื่อเริ่มจะส่งข้อมูลสัญญาณอะซิงโครนัสจะเป็น "0" หนึ่งช่วงของสัญญาณนาฬิกาบิตนี้เรียกว่าสตาร์ทบิต ตามหลังของสตาร์ทบิตก็จะเป็นข้อมูลสำหรับ 1 ตัวอักษร ซึ่งอาจจะมีขนาดตั้งแต่ 5 บิต จนถึง 8 บิต โดยบิตที่มีค่าน้อยที่สุด (LSB) จะถูกส่งออกมาก่อนไล่ไปจนถึงบิตที่มีค่ามากที่สุด (MSB) การเข้ารหัสอักขระนี้ ส่วนมากจะนิยมใช้รหัส ASCII แรกเริ่มทีเดียวในงานของโทรพิมพ์ เขาใช้รหัส (Baudot) ซึ่งใช้ 5 บิตในการส่งแทนตัวอักขระ 1 ตัว ตามหลังข้อมูลก็จะเป็นพาริตีบิต ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้ พาริตีบิตทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับพาริตีบิตอาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) หมายความว่า ถ้าหากเป็นพาริตีคู่ จำนวนบิตที่เป็น 1 ในช่วงของบิตข้อมูลกับบิตพาริตีรวมแล้วจะต้องเป็นจำนวนคู่ ผู้ส่งจะต้องทำหน้าที่ตรวจสอบข้อมูลแล้วใส่พาริตีบิตเอง ฝ่ายรับเมื่อได้รับแล้วก็ต้องตรวจสอบดูว่าเป็นจริงตามสถานการณ์ที่ตั้งเอาไว้หรือไม่ หากมีความผิดพลาดเกิดขึ้นก็หมายความว่า สัญญาณที่รับนั้นผิดพลาดไปจากที่สถานีส่งส่งออกมา ทั้งนี้ทั้งนั้นจะต้องเป็นจำนวนคี่เท่านั้น คือผิดไป 1 บิต 3 บิต หรือ 5 บิต พร้อมกันจึงจะตรวจสอบได้ว่าผิด มองเห็นง่าย ๆ ว่าถ้าผิดเป็นจำนวนคู่ ผลรวมของจำนวนหนึ่งก็ยังคงอยู่ดี ทั้งนี้ทั้งนั้นไม่ได้หมายความว่าพาริตีคี่ (Odd parity) จะตรวจสอบความผิดพลาดเป็นจำนวนคี่ความจริงแล้วตรวจสอบความผิดพลาดได้เหมือนกับพาริตีคู่ (Even parity) แต่แทนที่จะตรวจสอบดูว่าสัญญาณที่รับเข้ามามีจำนวนคู่ ก็ตรวจสอบดูว่ามีจำนวนคี่หรือเปล่าอย่างไรก็ตามโอกาสที่จะผิดพลาด 2 บิตพร้อมกันมีน้อยมาก

ย้อนกลับมาดูสัญญาณอะซิงโครนัสใหม่ หลังจากบิตพาริตีแล้ว ก็จะมีสตอปบิตหนึ่งความกว้างของสตอปบิตอาจจะเป็น 1, 1.5, หรือ 2 พัลส์ของสัญญาณนาฬิกา แล้วแต่ผู้รับส่งจะตกลงใช้กันเองการเริ่มใช้พอร์ตอนุกรม (ทางออกอนุกรม) จึงจำเป็นจะต้องตั้งค่าต่างๆ สำหรับเป็นการส่งแบบอนุกรมอันได้แก่

1. ความเร็วในการส่ง
2. ความยาวรหัส 1 อักขระ
3. บิตตรวจสอบ
4. จำนวนบิตสตอป

ในการส่งโทรพิมพ์หรือโทรเลขเมื่อก่อนนี้ใช้ความเร็วแค่ 70 บอด และ 110 บอด สำหรับคอมพิวเตอร์ความเร็วในการส่งมีให้เลือกตั้งแต่ 110, 200, 300, 1200, 2400, 4800, 9600 บอด และสูงไปกว่านั้น เนื่องจากมี IC หลายเบอร์ทำหน้าที่รับส่งแบบอะซิงโครนัสให้ใช้การส่งแบบอนุกรม จึงสะดวกสบายสำหรับคนออกแบบพอร์ตอนุกรม

จะเห็นว่ากลไกในการซิงโครนัสของการสื่อสารอะซิงโครนัส มีลักษณะเป็นไปทีละตัวอักขระจำนวน พัลส์ของสัญญาณที่ส่งออกมายังมีส่วนใช้ในการควบคุมการส่งอยู่ อันได้แก่ บิตสตาร์ท บิตสตอป และบิตพาริตี ทำให้ความเร็วในการส่งอักขระต่อวินาทีน้อยลงไป การส่งสัญญาณด้วยความเร็ว 300 บอด สำหรับการเข้ารหัส 7 บิต ไม่ได้หมายความว่าส่งได้ 300 ฮาด้วย 7 อักขระต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลแบบอะซิงโครนัส ที่มีการแมตช์ความเร็ว

เราได้กล่าวถึงการถ่ายโอนข้อมูลจากแบบอนุกรมจากอุปกรณ์เครื่องหนึ่งไปยังอีกเครื่องหนึ่งซึ่งอาจจะ เป็นคอมพิวเตอร์หรืออุปกรณ์สื่อสารชนิดอื่น โดยสมมติฐานว่าความเร็วในการเปลี่ยนสัญญาณจากขนาน เป็นอนุกรมได้เร็วพอ และฝ่ายรับเปลี่ยนจากอนุกรมเป็นขนาน แล้วนำไปแสดงบนจอพิมพ์ออกที่เครื่องพิมพ์ หรือเก็บไว้ในดิสก์ทันทีได้ทันเวลาด้วยความเร็วในแต่ละการทำงานเท่ากันทั้งฝ่ายรับและ ฝ่ายส่งไม่มีการหน่วง เวลาหรือการอินเตอร์รัพระหว่างกลางอย่างไรก็ตาม การสมมุติฐานนี้ย่อมไม่เป็นความจริงฝ่ายส่งทำหน้าที่ส่ง อย่างเดียวแต่ฝ่ายรับอาจต้องทำหน้าที่หลายอย่าง เช่น รับ แสดงผล เก็บ พิมพ์ เป็นต้น ความเร็วของฝ่ายรับ หากไม่เพียงพอที่จะทำหลายอย่างให้ทันฝ่ายส่ง (แน่นอเนย่มขึ้นอยู่กับความเร็วในการส่ง) ก็จำเป็นจะต้องมี กลไกในการควบคุมการถ่ายโอน เทคนิคในการควบคุมความเร็วในการส่งมีอยู่หลายรูปแบบ ซึ่งอาจจะแบ่ง ออกได้เป็น 2 ลักษณะคือข้อมูลบอกการออน - ออฟการทำงาน (on-off data flow toggle) และหาที่เก็บข้อมูล ชั่วคราวหรือสร้างบัฟเฟอร์ (Temporary data storage mechanism)

การควบคุมการส่งเมื่อความเร็วในการทำงานของฝ่ายรับและฝ่ายส่งไม่เท่ากัน

เนื่องจากการใช้ภาษาในระดับสูงเขียนเป็นโปรแกรมสำหรับการควบคุมการทำงานของการทำงานของการถ่ายโอนข้อมูลแบบอนุกรม อาจจะใช้เวลามากกว่าที่จะรับข้อมูลเข้ามาได้ทันกับสถานีส่งข้อมูลมาจำเป็นต้องมีวิธีการควบคุมไม่ให้เกิดการสูญหายของข้อมูลที่สถานีส่ง ส่งมาวิธีการดังกล่าวมีอยู่หลายวิธีคือ

การมีบัฟเฟอร์ในการสื่อสารข้อมูล

บัฟเฟอร์สำหรับการสื่อสารก็คือหน่วยความจำในคอมพิวเตอร์ ซึ่งแยกออกมาจากหน่วยความจำหลัก สำหรับเก็บข้อมูลในการติดต่อชั่วคราว บัฟเฟอร์สำหรับการสื่อสารนี้ส่วนมากใช้สำหรับฝ่ายรับเท่านั้น เนื่องจากฝ่ายรับจำเป็นจะต้องตามฝ่ายส่งให้ทันถ้าหากฝ่ายรับใช้ภาษาแอสเซมบลีควบคุมมีความเร็วพอ อาจไม่จำเป็นต้องใช้บัฟเฟอร์สำหรับการสื่อสารเนื่องจากภาษาแอสเซมบลีมีความเร็วสูง

ข้อมูลที่จัดส่งให้คอมพิวเตอร์ที่เป็นฝ่ายรับ ส่วนมากจะอ่านมาจากแท้มที่บันทึกไว้ในดิสก์ หากพิจารณา ระหว่างการส่งข้อมูลออก ข้อมูลที่อ่านมาจากดิสก์จะมีลักษณะเป็นกลุ่มได้รับการนำมาสู่บัฟเฟอร์การอ่าน แต่ละกลุ่มดำเนินไปจนกระทั่งบัฟเฟอร์เต็ม การอ่านจะหยุดลงจนกระทั่งบัฟเฟอร์ถูกส่งออกไปหมดในลักษณะ ของเข้าก่อนออกก่อน ข้อมูลก็จะถูกอ่านออกมาใส่ในบัฟเฟอร์ส่งอีกครั้ง โดยปกติบัฟเฟอร์ส่งจะมีขนาด 255 ตัวอักษรหรือประมาณ 3 บรรทัดของ 80 อักขร

บัฟเฟอร์รับของฝ่ายรับมีผลกระทบต่อกรรับ - ส่งข้อมูลมากกว่าบัฟเฟอร์ส่ง บัฟเฟอร์รับทำหน้าที่เช่นเดียวกับบัฟเฟอร์ส่ง แต่ทิศทางการไหลของข้อมูลอยู่ในทางตรงกันข้าม ฝ่ายรับรับข้อมูลเข้ามาเก็บไว้ใน บัฟเฟอร์รับจนกว่าโปรแกรมควบคุมการสื่อสารจะนำข้อมูลออกไปจากบัฟเฟอร์รับ เพื่อไปแสดงหรือพิมพ์ หรือ เก็บไว้ในแท้มก็แล้วแต่ ขอเพิ่มเติมอีกนิดว่าในระบบควบคุมการทำงานของไมโครคอมพิวเตอร์อย่างเช่น IBM PC มีกลไกบัฟเฟอร์รับส่งนี้ไว้รออยู่ในโปรแกรมในระดับสูง จึงเพียงแต่ทำหน้าที่ดึงเอาข้อมูลจากบัฟเฟอร์นี้ไปใช้ เราจะเห็นได้ชัดจากความจำเป็นในการใช้บัฟเฟอร์นี้เมื่อความเร็วในการส่งสูงเกินกว่า 600 บอด ภาษาในระดับสูงเช่น ภาษาเบสิกไม่สามารถที่จะรับข้อมูลจากพอร์ตอนุกรมได้ทันแน่ ๆ ระบบควบคุมการทำงาน จึงถูกออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบมาเพื่อการสื่อสารข้อมูลโดยการใช้อินเตอร์พอร์ทเข้าช่วย เมื่อมีข้อมูลมาที่พอร์ตอนุกรมเมื่อระบบควบคุมอินเตอร์พอร์ทการทำงานเพื่อดึงข้อมูลไปใส่ในบัฟเฟอร์รับทันที เพื่อไม่ให้ข้อมูลที่รับหายไปก่อน เมื่อมีตัวใหม่ส่งมาที่พอร์ตอนุกรมหน้าที่ของโปรแกรมควบคุมการรับส่งก็คือการอ่านข้อมูลจากบัฟเฟอร์รับไปใช้เมื่อถูกอ่านจากบัฟเฟอร์รับไปแล้วตัวอ่านที่ออกไปก็จะหายไปจากบัฟเฟอร์ ลองนึกภาพดูจะเห็นว่าฝ่ายหนึ่งคือระบบควบคุมการทำงาน (OS) รับข้อมูลจากพอร์ตอนุกรมใส่บัฟเฟอร์อีกฝ่ายหนึ่งคือโปรแกรมควบคุมการรับส่งดึงข้อมูลออกจากบัฟเฟอร์เปรียบเสมือนคนหนึ่งตักน้ำใส่ตุ่มอีกคนหนึ่งตักออกจากตุ่ม ถ้าฝ่ายที่ตักออกมีความเร็วมากกว่าตุ่มก็จะมีโอกาสแห้ง ในทางตรงกันข้ามถ้าฝ่ายตักออกช้ากว่าฝ่ายตักเข้าโอกาสที่จะล้นตุ่มก็ย่อมจะมี ในทางสื่อสารเรียกว่าบัฟเฟอร์รับโอเวอร์โฟลว์ (receive buffer overflow) การไหลล้นดังกล่าวทำให้ข้อมูลที่รับหายไป

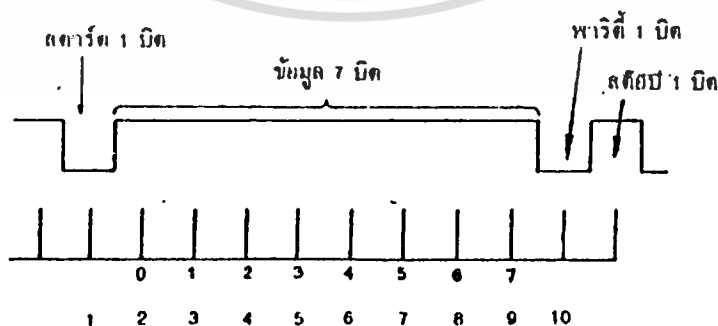
โปรแกรมที่เขียนด้วยภาษาแอสเซมบลีสามารถทำงานได้เร็ว และอาจไม่จำเป็นต้องใช้บัฟเฟอร์สำหรับการรับส่งเลยก็ได้ แต่ถ้าหากเขียนด้วยภาษาเบสิกจำเป็นจะต้องมีบัฟเฟอร์อย่างน้อย 1024 ไบต์ สำหรับการสื่อสารที่ความเร็วไม่เกิน 600 บอด

ในเครื่อง IBM PC เราสามารถกำหนดบัฟเฟอร์สำหรับการสื่อสารได้เมื่อเราเรียกใช้ BASIC หรือ BASICA เช่น A> BASIC\VC:4096

เป็นการกำหนดบัฟเฟอร์การสื่อสารด้วยขนาด 4 กิโลไบต์ ถ้าหากเราไม่กำหนดค่าบัฟเฟอร์สื่อสารเอาไว้ DOS จะตั้งค่าให้เท่ากับ 256 ไบต์ โดยการใส่ /C: ทำให้เราสามารถบัฟเฟอร์ได้จาก 0 ถึง 32,767 ไบต์ อย่างไรก็ตามภาษาเบสิกใช้หน่วยความจำได้ 64 กิโลไบต์ หากเรากำหนดค่าบัฟเฟอร์สื่อสารเอาไว้มากย่อมจะมีหน่วยความจำเหลือสำหรับใช้อย่างอื่นน้อยลง เมื่อเข้าสู่ภาษาเบสิกแล้วแรกเริ่มภาษาเบสิกจะบอกจำนวนหน่วยความจำเหลือใช้งานจริง ๆ ในขณะที่โหลดโปรแกรมภาษาเบสิกเข้ามาขนาดของโปรแกรม ย่อมจะเข้าไปอาศัยที่ในหน่วยความจำที่เหลือทำให้หน่วยความจำที่จะใช้งานสำหรับเก็บตัวแปรต่าง ๆ น้อยลงไปอีก เราสามารถใช้คำสั่งโดยตรงสำหรับการตรวจสอบขนาดของหน่วยความจำที่ว่างโดยการพิมพ์คำสั่ง PRINT FRE(O)

การควบคุมโดยใช้ XON/XOFF

ถึงแม้ว่าเราจะมีบัฟเฟอร์สำหรับการสื่อสารแล้วก็ตาม ในบางครั้งการถ่ายโอนข้อมูลด้วยความเร็วสูง และด้วยขนาดของแฟ้มที่จะทำการถ่ายโอนมีขนาดใหญ่กว่าบัฟเฟอร์สื่อสาร



รูปที่ 2.1.7 รูปแบบของข้อมูลหนึ่งตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โอกาสที่ข้อมูลจะหายไปมีอยู่มากลองมาคำนวณดูง่าย ๆ สมมติว่าเราใช้ความเร็วในการถ่ายโอนข้อมูล 9600 บิตต่อวินาที สตอปบิตเป็น 1 บิต ข้อมูล 7 บิต และพาริตีเป็นคู่ ใน 1 ตัวอักษรจะต้องใช้ 11 บิต

เพราะฉะนั้นฝ่ายรับจะต้องอ่านข้อมูลจากพอร์ตนุกรมทุก 11/9600 ได้ผลประมาณ (ความเร็วนาฬิกาของ IBM PC = 4.77 MHz หรือประมาณ 0.2 ไมโครวินาทีต่อหนึ่งพัลส์ $1000/0.2=5000$) ในเมื่อบัพเฟอร์ไม่เพียงพอเราก็จำเป็นต้องควบคุมการรับส่ง โดยการบอกให้ฝ่ายส่งหยุดส่งชั่วคราว (XOFF) นกว่าฝ่ายรับจะจัดการเอาข้อมูลออกจากบัพเฟอร์สื่อสารหมดเสียก่อน จึงบอกให้ฝ่ายส่งจัดการส่งต่อไป (XON) ในรหัส ASCII XON มีค่าเท่ากับ 17 XOFF มีค่าเท่ากับ 19 เป็นหน้าที่ของนักเขียนโปรแกรมที่จะต้องจัดการส่ง XOFF ออกไปให้ฝ่ายส่งได้รู้ก่อนที่บัพเฟอร์สื่อสารจะเต็มเสียก่อน

คอมพิวเตอร์เมนเฟรมส่วนมากจะมีระบบ XON/XOFF ให้สำหรับการเชื่อมต่อทางด้านความเร็ว (Speed Matching) แต่โปรแกรมสื่อสารที่มีขายสำหรับ IBM PC ไม่มี XON/XOFF ทุกตัวโดยมากโปรแกรมที่เขียนโดยภาษาแอสเซมบลีจะมี XON/XOFF ให้ แต่โปรแกรมที่เขียนโดยภาษาเบสิกจะมีเพียงบางโปรแกรมเท่านั้นโปรแกรมที่เขียนโดยภาษาแอสเซมบลีมีความเร็วพอที่จะคอยตรวจสอบดู XOFF ที่ส่งมาจากฝ่ายรับแต่ถ้าเป็นภาษาเบสิกความเร็วอาจจะไม่เพียงพอต่อการตรวจสอบ XOFF ที่ฝ่ายส่ง

ฮาร์ดแวร์ที่เกี่ยวข้องกับการสื่อสาร

พอร์ต RS232C

โดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรม เรียกชื่อกันว่า RS232C อยู่ในตัวเองอยู่แล้วหลายเครื่องไม่มีมากับเครื่อง อย่างเช่น IBM PC จำเป็นจะต้องมีการิตที่เรียกว่าอะซิงโครนัสแอดปเตอร (Asynchronous Communication Adapter) มาเสียบใส่

พอร์ต RS 232C ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS 232C ก็เนื่องจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของอเมริกาหรือ EIA ได้กำหนดมาตรฐานของอุปกรณ์การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS 232C ความจริงมาตรฐานการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมใช้กันมากที่สุดสำหรับไมโครคอมพิวเตอร์ก็คือ RS 232C

หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสก็คือ
รับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. ตัดสตอปบิตและพาริตีบิตออก
4. ส่งสัญญาณให้ซีพียูรู้ว่ารับสัญญาณได้แล้ว

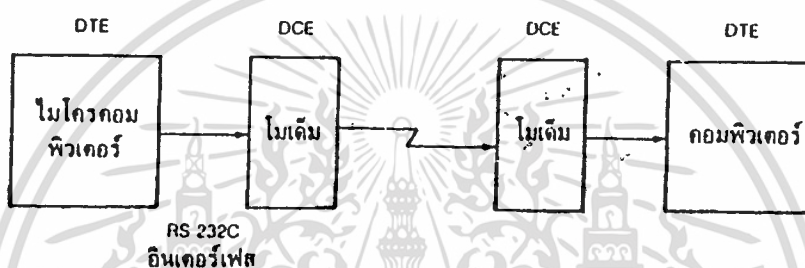
ส่งสัญญาณ

1. เปลี่ยนสัญญาณแบบขนานจากซีพียู ต่อกและทยอยส่งออกเป็นแบบอนุกรม
2. เพิ่มสตอปบิตและพาริตีบิต
3. เพิ่มสัญญาณควบคุมโมเด็มที่เชื่อมต่อ (ถ้ามี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน RS 232C

มาตรฐาน RS 232C ได้จัดพิมพ์ขึ้นเมื่อ ปี ค.ศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งสหรัฐอเมริกา RS ย่อมาจากคำว่า Recommended Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขฉบับสุดท้ายของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐานนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment :DTE) กับอุปกรณ์สื่อสารข้อมูล (Communication Equipment :DCE) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE ก็หมายถึงตัวไมโครคอมพิวเตอร์และ DCE ก็หมายถึงโมเด็ม อุปกรณ์อื่นๆ เช่นเครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะใช้ได้ทั้ง DTE และ DCE ขึ้นอยู่กับบริษัทผู้ผลิต ข้อแตกต่างของ DTE และ DCE จะเห็นได้จาก รูปที่ 2.1.8 จากรูปนี้เราจะเห็นได้ว่า RS 232C มีส่วนสำคัญอย่างมากต่อวงจรสำหรับการสื่อสารข้อมูลระหว่างไมโครคอมพิวเตอร์



รูปที่ 2.1.8 การใช้ RS 232C เชื่อมต่ออุปกรณ์

ความจริงอีกประการหนึ่งของ RS 232C ก็คือความเร็วและระยะทางของการเชื่อมต่อ RS 232C สามารถเชื่อมต่อการถ่ายโอนข้อมูล ได้จาก 0-20,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับการสื่อสารไมโครคอมพิวเตอร์กับอุปกรณ์รอบนอก

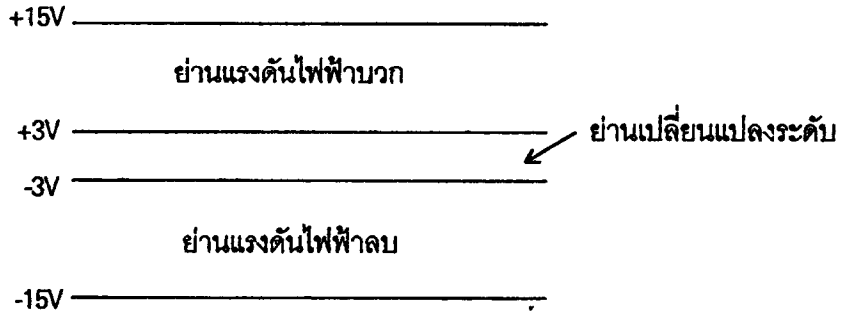
ลักษณะของสัญญาณ RS 232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้อง และอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นจะต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้ มาตรฐาน RS 232C กำหนดเดียนของแรงดันไฟฟ้าในสัญญาณเพื่อสนองจุดประสงค์ข้างบน ดังแสดงในตารางที่ 2.1 ก และรูปที่ 2.1.9

ตารางที่ 2.1 ก มาตรฐานของการใช้แรงดันไฟฟ้า

แรงดันไฟฟ้า	สถานะภาพลอจิก	สถานะภาพของสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	สเปซ	อน
ลบ	2	มาร์ค	ออฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

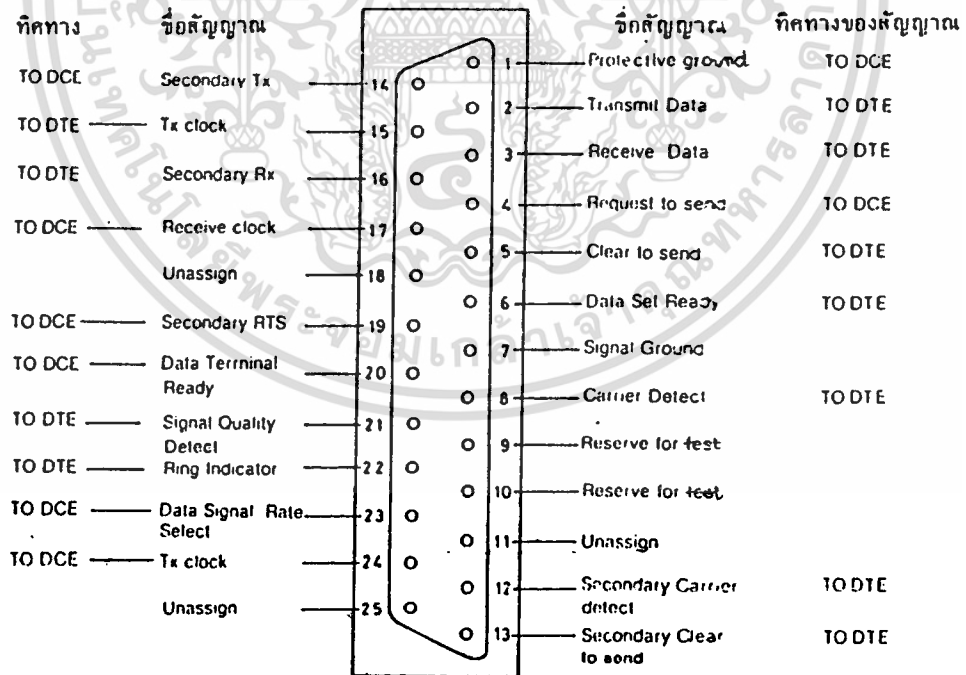


รูปที่ 2.1.9 ย่านของแรงดันไฟฟ้าที่ไม่ใช้สัญญาณ RS 232C

สำหรับไมโครคอมพิวเตอร์บางเครื่อง ใช้แต่สัญญาณลอจิกออกมาเป็นสัญญาณของ RS 232C เลย อย่างเช่น อะซิงโครนัสอะแคบเตอร์ของ IBM PC ในกรณีเช่นนี้ระยะทางของสายที่เชื่อมต่ออาจจะไปได้สั้นกว่า 50 ฟุต ดังที่กล่าวเอาไว้ เนื่องจากระดับของกราวนด์เปลี่ยนแปลงไป อันเนื่องจากการสูญเสียไปในความต้านทานของสาย ผู้ที่เคยใช้ IBM PC อาจจะเคยประสบกับปัญหานี้มาแล้วว่าทำไมต่อสัญญาณ RS 232C เกินกว่า 10 ฟุต แล้วใช้งานไม่ได้ แต่อย่างไรก็ตาม RS 232C ของ IBM PC ยังมีโอกาสให้เลือกใช้ 20 มิลลิแอมแปร์ กระแสสวนกลับแทนแรงดันไฟฟ้า

การกำหนดจุดต่อของ RS 232C

ในทางฟิสิกส์แล้ว มาตรฐาน ของ RS 232C กำหนดข้อต่อแบบ DB-25 แต่ละขาของข้อต่อกำหนดไว้ ดังรูปที่ 2.1.10 อย่างไรก็ตามผู้ผลิตไมโครคอมพิวเตอร์อาจจะใช้ข้อต่อที่นอกเหนือไปจาก DB-25 ยกตัวอย่าง เช่น Fujitsu F-8 ,IBM AT, IBM Jr เป็นต้น



DTE = Data terminal Equipment
DCE = Data,Communication Equipment (Modem)

รูปที่ 2.1.10 การกำหนดของข้อต่อ RS 232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเมียของข้อต่อควรจะไม่เต็ม ขณะที่ตัวผู้ควรจะไม่เต็ม Asynchronous communication adapter หรือที่ตัวไมโครคอมพิวเตอร์เอง อย่างไรก็ตามผู้ผลิตหลายรายไม่ได้ทำตามกฎเกณฑ์ที่ว่านี้

สัญญาณต่างๆ ถูกมอบหมายให้ทำหน้าที่ดังนี้

Transmit Data (TD ขาที่ 2)

เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อเข้าโดยตรงกับ ไมโครคอมพิวเตอร์ตัวอื่น หรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออกสถานะภาพของลอจิกที่ขานี้จะมีค่าเท่ากับ "1" หรือเทียบเท่ากับสตอปบิต

Receive Data (RD ขาที่ 3)

เป็นทางของสัญญาณเข้าไปยัง DTE หรือ ไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามานาขานี้จะมี สถานะภาพทางลอจิกเป็น "1"

Request To Send (RTS ขาที่ 4)

ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทาง ขาที่ 2 สัญญาณนี้ใช้ควบคู่กับ CTS หรือ Clear to send อุปกรณ์รับหากได้รับสัญญาณ RTS จะตรวจสอบตัวเอง ว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะรับก็ส่งสัญญาณออกไปที่สาย CTS

Clear To Send (CTS ขาที่ 5)

ดังอธิบายไว้ใน RTS เมื่อสัญญาณนี้อยู่ในสถานะลอฟ (Negative voltage หรือลอจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่าพร้อมที่จะรับข้อมูลแล้ว

Data Set Ready (DSR ขาที่ 6)

เมื่อสัญญาณสายนี้อยู่ในสถานะฮอน (หรือลอจิก "0") เป็นการบอกไมโครคอมพิวเตอร์หรือฝ่ายส่งว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว โมเด็มที่มีการหมุนหมายเลขอัตโนมัติจะ ส่งสัญญาณนี้ไปบอกให้คอมพิวเตอร์รู้ว่าต่อโทรศัพท์ได้สำเร็จแล้ว

Signal Ground (SG ขาที่ 7)

SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสายของสัญญาณ จะมีแรงดันเป็น "0" เมื่อเทียบกับ สัญญาณตัวอื่น

Carrier Detect (CD ขาที่ 8)

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะฮอน (ลอจิก "0") เมื่อพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมาก จะไม่รายงานสถานะภาพของตัวเอง (CD, DSR และ CTS) ให้คอมพิวเตอร์รู้ หากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR

Ring Indicator (RI ขาที่ 22)

สัญญาณนี้ใช้ในโมเด็มที่เป็นระบบตอบอัตโนมัติ (Auto-answer) สัญญาณนี้จะฮอนเมื่อมีสัญญาณ กระดิ่งมา และออฟระหว่างเสียงดังของกระดิ่ง

ท่านผู้อ่าน อาจจะสับสนระหว่างสถานะภาพของลอจิกกับสถานะภาพของสัญญาณ โดยปกติเราจะ ค้นเคยอยู่กับความรู้สึกที่ว่า เมื่อแรงดันเป็นบวก หรือสัญญาณฮอน ลอจิกน่าจะเป็น "1" สำหรับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆ ที่กล่าวมานี้จะมีลักษณะตรงกันข้าม ทำไมเรากำหนดกฎเกณฑ์ออกมาอย่างนี้ ก็เพราะว่าแต่เดิมนั้น การติดต่อกันทางโทรเลขการทำงานของสัญญาณจะต้องครบวงจรทั้งฝ่ายส่งและฝ่ายรับ เมื่อเป็นลอจิก "0" หรือขณะที่ไม่มีอะไรส่งควรจะมีสัญญาณทางไฟฟ้าครบวงจรอยู่ตลอดเวลา จะได้ว่าวงจรไม่ขาดระหว่างทาง ตรงไหนควรจะรู้ว่าวงจรครบอยู่ตลอดเวลา ก็โดยการให้ค่าแรงดันที่ฝ่ายส่ง ดังนั้นจึงถือกันว่าสัญญาณไฟบวก ให้เป็นลอจิก "0"

มาตรฐาน RS 232C

ได้กล่าวถึงมาตรฐานตามสมาคมผู้ผลิตอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกาหรือ RS 232C ไปแล้ว สหประชาชาติและกลุ่มของ CCITT (Committee Consytitative International Telephone and Telegraph) ได้ออกมาตรฐานมาเหมือนกัน และก็หลายฉบับตั้งแต่การประชุมครั้งที่สองที่กรุงนิวยอร์กปี ค.ศ. 1960 ออกมาเป็นสมุดสีส้ม (คงจะไล่ทำกันจนหมดสีที่จะทำปกนั้นแหละจึงเป็นมาตรฐานถาวร) ได้มาตรฐานออกมา 3 รูปแบบคือ

V.24 บรรยายถึงการเชื่อมต่อของอุปกรณ์รับส่งข้อมูล (DTE) กับอุปกรณ์รับส่งข้อมูลปลายทาง

V.28 บรรยายถึงลักษณะทางไฟฟ้าสำหรับการใช้ Unbalance Double Current Interchange Circuit

จะเป็นการบังเอิญหรือโชคของผู้ใช้ก็แล้วแต่ V.24 และ RS 232C มีลักษณะคล้ายคลึงกันจนสามารถใช้ด้วยกันได้ ท่านสามารถจะหารายละเอียดเปรียบเทียบได้จากหนังสือชื่อ Technical Aspect of Data Communication โดย John E. MeNamara จัดพิมพ์โดยบริษัท DEC ปี ค.ศ. 1977 หมายเลขหนังสือ ISBN 0-932376-01-0 หรือจะเขียนจดหมายไปขอรายละเอียดจากสหประชาชาติหรือสมาคมผู้ผลิตอุตสาหกรรมของอเมริกาก็ตามแต่ท่านจะสะดวก

RS 422 และ RS 433

จุดอ่อนของ EIA RS 232C พอดีสรุปได้ 3 ประการ

1. ใช้ระดับแรงดันไฟเลี้ยง -15 โวลต์ นอกเหนือจาก -5 โวลต์ที่ใช้ในลอจิก

2. ค่าตัวเก็บประจุของตัวอุปกรณ์รับสัญญาณ RS 232C รวมทั้งตัวเก็บประจุสแตย์ (Stay capacitance) ในสายจะต้องมีมากกว่า 2500 pF สายที่รวมกันหลายๆสายส่วนมากจะมีตัวเก็บประจุสแตย์ประมาณ 40-50 pF ต่อ 1 ฟุต ดังนั้นสายนี้จะต่อได้ยาวสุด 50 ฟุต ก่อนที่ค่าตัวเก็บประจุสแตย์จะมากกว่าที่กำหนดนี้ ช่วงเวลาการเปลี่ยนแปลงของระดับของสัญญาณจะมากกว่า 4 เปอร์เซ็นต์ ตามที่ยอมให้ได้ในมาตรฐาน RS 232C เมื่อเป็นเช่นนี้จะทำให้ฝ่ายรับตีความหมายของสัญญาณผิดไปจากความเป็นจริง มาร์กบิต (MARK bit) จะยาวกว่า สเปซบิต (SPACE bit) หรือ สเปซบิตยาวกว่ามาร์กบิต ขึ้นอยู่กับวงจรการตรวจสอบความผิดเพี้ยนแบบนี้ เรียกว่า " Bias distortion " รูปที่ 2.1.11 แสดงถึงการตีความข้อมูลที่มีผิดของฝ่ายรับ อันเนื่องมาจากไบอัสดิสทอร์ชัน และสัญญาณนาฬิกาเร็วกว่าปกติ

3. ปัญหาที่ 3 เป็นปัญหาทางสัญญาณไฟฟ้า ที่ EIA ไม่ได้แก้เอาไว้ สำหรับวงจรที่ใช้ IC ก็คือปัญหาเรื่องกราวนด์ที่แตกต่างกัน ตามมาตรฐาน EIA สัญญาณที่ส่งออกเทียบกับกราวนด์ของเครื่องส่งเท่านั้น ถ้าหากเครื่องรับกับเครื่องส่งมีระดับแรงดันกราวนด์ สมมติว่า 2 โวลต์กระแสที่ไหลในเส้นที่เป็นกราวนด์ (ขา 7) ก็จะทำให้เกิดขึ้น

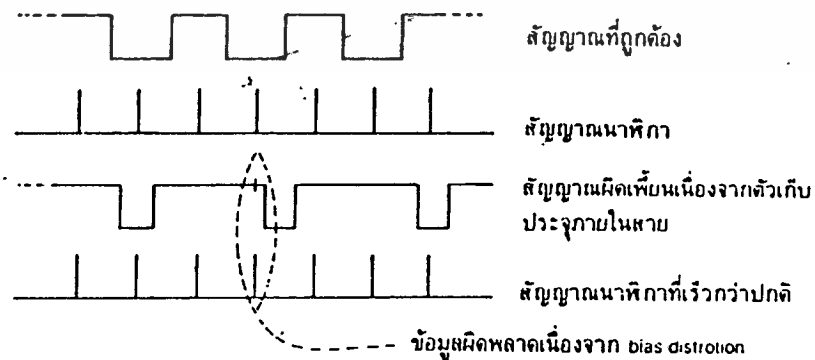
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ข คุณสมบัติโดยย่อของสัญญาณ RS 232C

Driver output logic levels with 3k to 7k load	$15V > V_h > 5V$ $-5 > V_l > -15$ โวลต์
Driver output voltage when open circuit	$V_o < 25$ โวลต์
Driver output impedance with Power off	$R_o > 300$ Ohms
Output short circuit current	$I_o < 0.5$ A
Driver slew rate	$dv/dt < 30$ V/μs
Receiver input impedance	$7k > R_{in} > 3k$
Receiver input voltage	+15 compatible with driver
Receiver output with open circuit input	MARK
Receiver output with +3V input	SPACE
Receiver output with -3V input	MARK
+15	LOGIC 0 = SPACE =
+5	CONTROL ON
+5	Noise Margin
+3	Transition Region
+3	
-3	Noise Margin
-3	

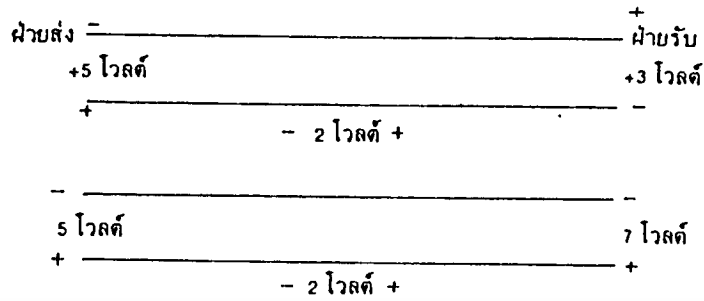
สมมติความต้านทานของสายเป็นศูนย์ ความต่างศักย์ที่เกิดจากกระแสเท่าเดิม ระดับของสัญญาณที่ฝ่ายส่งและฝ่ายรับมองเห็นก็จะแตกต่างกัน 2 โวลต์ ฝ่ายส่งใส่แรงดันเข้าไป 5 โวลต์ ฝ่ายรับจะมองเห็นแค่ 3 โวลต์เท่านั้น ในทางกลับกันถ้าฝ่ายส่งใส่แรงดัน -5 โวลต์ ฝ่ายรับจะมองเห็นเป็น -7 โวลต์ ความต่างศักย์ของกราวนด์จะคงที่ 2 โวลต์ ไม่ว่าจะฝ่ายส่งจะใส่แรงดันเข้าไปเท่าไรก็ตาม ผลของกราวนด์ที่แตกต่างกันนี้อาจจะเกิดมาจากสถานีรับและสถานีส่งมีระบบไฟฟ้าที่มีกราวนด์แตกต่างกันก็ได้



รูปที่ 2.1.11 แสดงปัญหาความผิดพลาดทางสัญญาณไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากตระหนักถึงปัญหาเหล่านี้ EIA ได้ออกมาตรฐานมาใหม่ 2 มาตรฐานคือ RS 422 และ RS 423

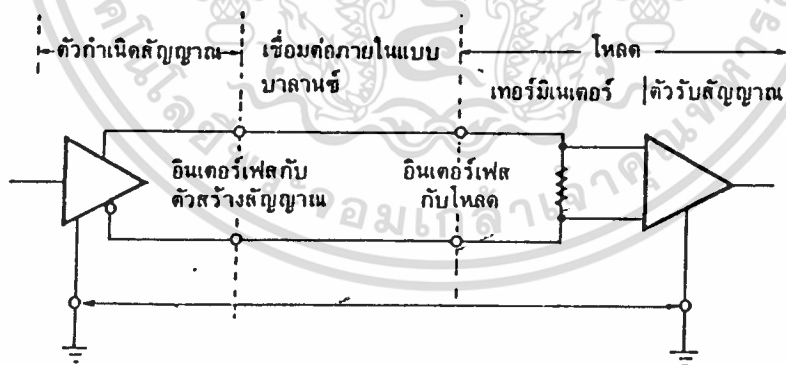


รูปที่ 2.1.12 ผลของระดับสัญญาณกราวนด์ที่แตกต่างกัน

ใน RS 422 แก้ปัญหา RS 232C โดยการส่งสัญญาณแบบแรงดันบาลานซ์ (Balance Voltage) โดยกฎเกณฑ์สรุปไว้ในรูปที่ 2.1.13

RS 423 เป็นการแก้ปัญหาอีกวิธีหนึ่ง โดยการส่งสัญญาณแบบแรงดันไม่บาลานซ์ (Unbalance Voltage) กฎเกณฑ์สรุปเอาไว้ในรูปที่ 2.1.14

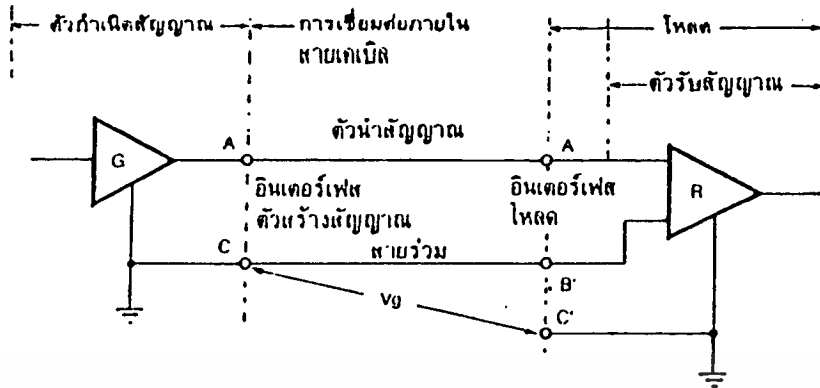
ในเครื่อง IBM PC หากใช้อะแดปเตอร์สื่อสารข้อมูลแบบอะซิงโครนัส มีหนทางให้เลือกในการเชื่อมต่ออุปกรณ์อื่นอีก 2 แบบ คือ แบบแรงดันตามกฎเกณฑ์ของ RS 232C (แต่เป็นแรงดันแค่ 0 ถึง 5 โวลต์) และแบบกระแสสวนรอบ เพื่อขยายระยะทางของสายให้มากขึ้น



- RI = ความต้านทานของสายนำสัญญาณ
- Vg = ความต่างศักรากับกราวนด์
- A, B = จุดเชื่อมต่อกับตัวกำเนิดสัญญาณ
- A', B' = จุดเชื่อมต่อโหลด
- C = กราวนด์ของวงจรสร้างสัญญาณส่งออก
- C' = กราวนด์ของวงจรโหลด

รูปที่ 2.1.13 RS 422 วงจรการเชื่อมต่อสัญญาณดิจิทัลแบบบาลานซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- A, C = อินเทอร์เฟซกับตัวกำเนิดสัญญาณ
 A', B' = อินเทอร์เฟซกับโหนด
 C' = ความต้านของวงจรโหนด
 C = ความต้านของวงจรตัวรับสัญญาณ
 V_g = ศักลต่างระหว่างความต้านของโหนดและตัวรับสัญญาณ

รูปที่ 2.1.14 RS 422 วงจรการเชื่อมต่อสัญญาณดิจิทัลแบบไม่บาลานซ์

การเชื่อมต่อคอมพิวเตอร์กับคอมพิวเตอร์โดยตรง

ในบางครั้งท่านอาจต้องการย้ายข้อมูล จากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่ง แน่แน่นอนถ้าหากว่าไมโครคอมพิวเตอร์ที่ใช้มีดิสเก็ตต์และเป็นชนิดเดียวกัน ท่านก็เพียงแต่ถือปี่ดิสเก็ตต์แล้วก็นำไปใช้อีกเครื่องหนึ่ง ถ้าหากเป็นคนละยี่ห้อ คนละ OS วิธีดังกล่าวจะใช้ไม่ได้ผลเพราะการบันทึกลงแผ่นดิสก์ของคอมพิวเตอร์แต่ละยี่ห้อหรือ OS แต่ละชนิดส่วนมากจะใช้ด้วยกันไม่ได้ คืออีกเครื่องหนึ่งจะอ่านไม่ออก ขอยกตัวอย่างเช่น ถ้าท่านสร้างแฟ้มโดย CP/M86 ของ IBM PC ท่านจะให้ PC DOS บนเครื่องเดียวกันอ่านไม่ได้ ท่านอาจจะซื้อ IBM PC มาใหม่แล้วข้อมูลเก่าของท่านที่เคยใช้กับเครื่องเก่าจะอย่างไรก็ดี ขึ้นอยู่กับข้อมูลเข้าไปใหม่ตายแน่ๆ ถ้าหากข้อมูลเป็นร้อยๆ แฟ้ม

หนทางที่จะแก้ปัญหาข้างบน ก็การต่อเครื่องคอมพิวเตอร์เข้าด้วยกันโดยใช้ RS 232C แล้วถ่ายโอนข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง วิธีการต่อแบบนี้เรียกว่าใช้ Null Modem คือไม่ใช้โมเด็มนั่นเอง (ในกรณีนี้ระยะทางไม่เกิน 50 ฟุต)

วิธีการต่อ RS 232C เข้าระหว่างเครื่องคอมพิวเตอร์โดยตรงมีหลายวิธี ตามแต่ขบวนการที่จะใช้ ถ้าไม่ต้องการให้มีการตรวจสอบสัญญาณกันก็ต่อ RD เข้า TD ของอีกเครื่องหนึ่ง สายกราวด์ต่อถึงกันดังรูปที่ 2.1.15 ก็สามารรถใช้งานถ่ายโอนข้อมูลได้แล้ว

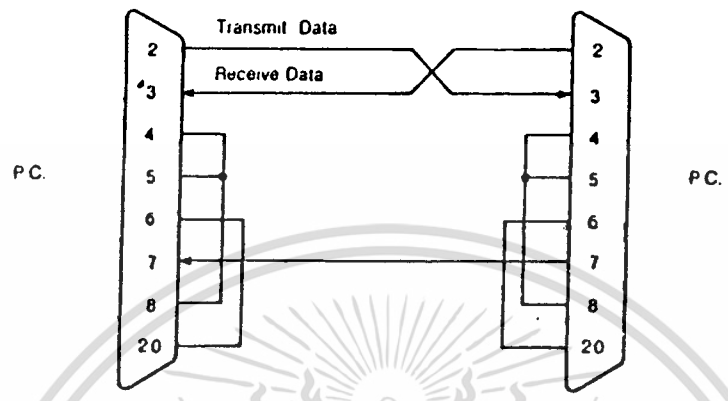
ปกติ OS ที่ให้บริการเกี่ยวกับพอร์ต RS 232C สัญญาณ RTS หรือ Request to send ออกมาที่ขา 4 ก่อน เมื่อ CS หรือ Clear to send ที่ขา 5 เป็นลอจิก "1" (หรือโฟลต) จึงจะเริ่มทำการส่งข้อมูลที่คอมพิวเตอร์บอกให้ส่งออกไปที่ขา 2 ในกรณีที่เป็นการต่อแบบง่าย ๆ ในรูปที่ 2.1.15 จึงถือว่าเป็นการหลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

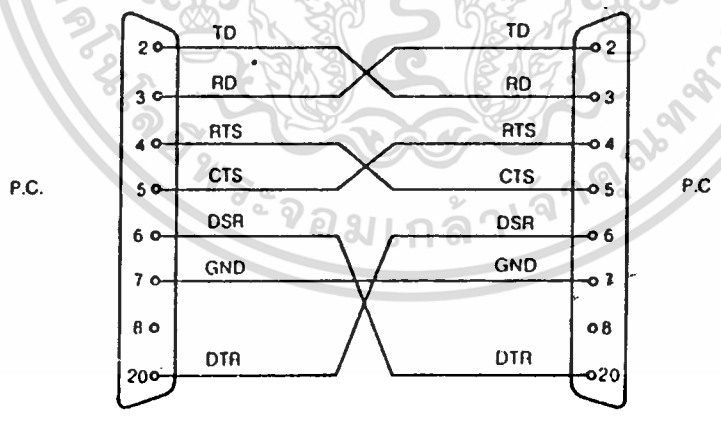


คอมพิวเตอร์ โดยการเอาขา 4 (RTS) ต่อเข้ากับขา 20 (DTR) เพื่อให้คอมพิวเตอร์ส่งข้อมูลได้ทันทีโดยไม่ต้องการ
 ความเรียบร้อยของฝ่ายรับ สำหรับขา 6 (Data Set Ready) ต่อเข้ากับขาที่ 20 (Data terminal ready) ก็ทำนอง
 เดียวกัน โดยปกติคอมพิวเตอร์จะถามอุปกรณ์ที่มาต่อพ่วงกับ RS 232C ว่าพร้อมที่จะส่งหรือยัง โดยส่ง
 สัญญาณถามที่ขา 20 และรอคำตอบที่ขา 6 นี่ก็เป็นการหลอกคอมพิวเตอร์เหมือนกัน คือถามที่ขา 20 ก็ได้รับ
 คำตอบกลับมาจากขา 6 ทันที



รูปที่ 2.1.15 การถ่าย RS 232C ระหว่างไมโครคอมพิวเตอร์อย่างง่าย

ในการต่อแบบนี้ฝ่ายรับจะต้องรอรับอยู่ก่อนแล้ว ก่อนที่ฝ่ายส่งจะเป็นผู้ส่งไม่เช่นนั้นข้อมูลที่ส่งออกมา
 หายแน่ๆ เพราะฝ่ายส่งไม่ได้ตรวจสอบความเรียบร้อยของฝ่ายรับก่อน
 เราอาจจะต่อสายให้มีการตรวจสอบสัญญาณโต้ตอบ (Hand Shake) ที่ดีกว่านี้ในรูปที่ 2.1.16



รูปที่ 2.1.16 การต่อไมโครคอมพิวเตอร์ผ่าน RS 232C แบบมี Hand Shake

ในกรณีเช่นนี้ จะมีการโต้ตอบที่ดีขึ้น เมื่อฝ่ายรับยังไม่พร้อมที่จะรับก็จะยังไม่มีสัญญาณ RTS ออกมา
 (พอร์ตอนุกรมยังไม่เปิด (open) "COM 1" ในภาษาเบสิกยังไม่ถูก เอกซ์คิวต์) ฝ่ายส่งซึ่งถือเอา RTS ของฝ่าย
 รับเป็น CTS ก็จะไม่ส่งข้อมูล

2.2 TMS 320C26

TMS 320C26 เป็นการปรับปรุงขึ้นจากรุ่นเดิมคือ TMS 32020 จึงทำให้ขีดความสามารถของ TMS 320C26 นี้เพิ่มขึ้นอีกมากมายคือ

- คำสั่งแต่ละคำสั่งที่ทำงานจะใช้เพียง 100 nS และคำสั่งส่วนใหญ่จะใช้เพียงหนึ่งไซเคิลเท่านั้น ดังนั้น TMS 320C26 จึงมีความเร็วในการทำงานได้สูงกว่า 10 ล้านคำสั่งในเวลา 1 วินาที
- มีคำสั่งที่จะโอนย้ายข้อมูล ระหว่างหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลได้
- ในการทำงานแต่ละคำสั่ง TMS 320C26 จะทำงานสามจังหวะแบบ pipeline คือ fetch, decode, execute โดยการทำงานนี้จะไม่เห็นได้โดยผู้ใช้งาน เพราะเป็นการทำงานที่เกิดขึ้นพร้อมกันเพื่อเพิ่มความเร็ว
- ได้เพิ่มจำนวนหน่วยความจำขึ้นอีก โดยมี RAM ภายในถึง 544 word (1 word มี 16 bit) หน่วยความจำภายนอกที่ใช้สามารถต่อได้ถึง 64K word ซึ่งมีจำนวนสูงมาก จึงเป็น DSP ที่มีการต่อหน่วยความจำภายนอกได้สูง
- มี ROM ให้ใช้ตามผู้สั่งทำอยู่บนชิพอีก 4K word การทำงานของชิพนี้ให้ข้อเด่นอีกข้อคือ สามารถโหลดโปรแกรมจากหน่วยความจำภายนอกเข้าไปเก็บใน RAM ภายใน เพื่อให้การทำงานทำได้เร็วยิ่งขึ้น
- ขีดความสามารถที่เพิ่มเติมมาอีกประการหนึ่งคือ การมี hardware timer และขีดความสามารถในการโอนย้ายข้อมูลเป็น block

Diagram ของ TMS 320C26 แสดงได้ดังรูปที่ 2.2.1

สถาปัตยกรรมของ TMS 320C26 ถูกสร้างขึ้นมาเพื่อความเร็วในการทำงานและโครงสร้างที่ทำงานได้ ด้วยขีดความสามารถที่สูงขึ้น และเพื่อให้การทำงานของบัสไม่ชนต่อกัน จึงแยกบัสเป็นบัสของโปรแกรมจะเป็นทางเข้าออกรหัสคำสั่ง และ operand ของคำสั่ง ส่วนบัสข้อมูลจะเชื่อมต่อโดยตรงกับวงจรการทำงานการประมวลผล เช่น CALU (Central Arithmetic Logic Unit) การทำโครงสร้างการคำนวณทางคณิตศาสตร์นี้ ยึดหลักการให้ทำงานด้วยประสิทธิภาพ เช่น การเลื่อนบิต (shift) การคูณหารหรือคำสั่งทางลอจิก

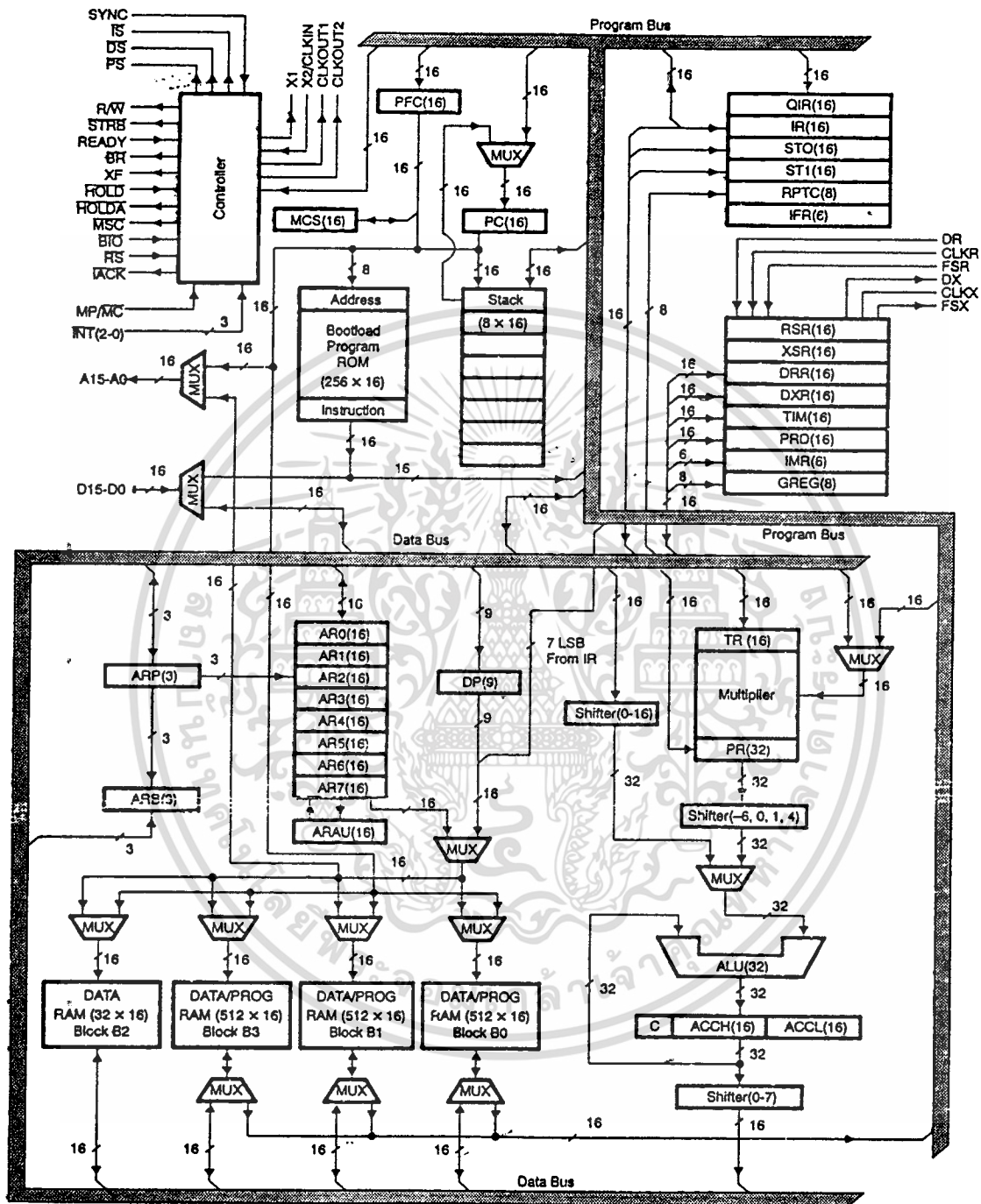
2.2.1 การจัดหน่วยความจำ

TMS 320C26 มีการแบ่งหน่วยความจำเป็นส่วน Program และ data ตามรูปที่ 2.2.2

2.2.1.1 data memory

Block B0 256 word (1 word มี 16 bit) B1 256 word B2 32 word เป็นหน่วยความจำภายใน (On-chip Ram) ซึ่งสามารถทำงานได้ด้วยความเร็วเต็มที่ โดยสามารถโปรแกรมให้เป็น Data Memory หรือ Program Memory ได้โดยคำสั่ง CNFD หรือ CNFP รูปที่ 2.2.2 a แสดง Memory Map หลังจากใช้คำสั่ง CNFD ส่วนรูปที่ 2.2.2 b แสดง Memory Map หลังจากใช้คำสั่ง CNFP ส่วนบล็อก B1 และ B2 จะเป็น Data Memory เสมอ

TMS 320C26 สามารถอ้างอิงหน่วยความจำข้อมูลได้ทั้งหมด 64 Kwords หากเราใช้หน่วยความจำข้อมูลบน chip ด้วย ตำแหน่งของมันจะถูก Map ลงในตำแหน่งที่ต่ำกว่า 1 Kwords ของ Data memory space และหน่วยความจำข้อมูลสามารถขยายเพิ่มขึ้นโดยตรงจนมีขนาด 64 Kwords ขณะที่ยังคงทำงานด้วยความเร็วเต็มที่



NOTE: Shaded areas indicate a bus.

รูปที่ 2.2.1 โครงสร้างของ TMS 320C26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 program_memory

โปรแกรม RAM และ ROM บนชิปหรือหน่วยความจำภายนอกความเร็วสูงสามารถถูกใช้ด้วยความเร็วเต็มที่โดยไม่มีสภาวะรอ (wait state) หรือมีจะนั้นขา READY ก็สามารติดต่อกับ TMS 320C26 ให้ช้าลง หากใช้หน่วยความจำภายนอกที่ทำงานช้า TMS 320C26 สามารถมีหน่วยความจำโปรแกรมได้ทั้งหมด 64 Kwords โดย RAM ภายใน (block Bo) สามารถกำหนดให้เป็นหน่วยความจำโปรแกรมได้ โดยใช้ software ซึ่งการ execute จาก block BO สามารถเริ่มได้หลังจาก Memory space ถูกกำหนด

หาก TMS 320C26 ถูกใช้ร่วมกับ on-chip program ROM ขนาด 4 Kwords ซึ่งสามารถโปรแกรมจากโรงงาน On-chip ROM ทำให้การ execute โปรแกรมได้ด้วยความเร็วเต็มที่ การกำหนดตำแหน่ง 4Kwords แรกว่าจะเป็นหน่วยความจำภายนอก chip หรือบน chip สามารถเลือกได้โดยกำหนดจากขา $\overline{MP} / \overline{MC}$ (Microprocessor/ Microcomputer) หากขาดังกล่าวเป็น High ตำแหน่ง 4Kwords แรกจะเป็นหน่วยความจำนอกชิป ถ้ามีสถานะเป็น Low ตำแหน่ง 4Kwords แรกเป็น On-chip ROM

มีการใช้แอสแตริสค์ระหว่างโปรแกรมข้อมูล และ I/O ทำให้เกิดความคล่องตัวในการทำงานหน่วยความจำภายในของ DSP นี้แบ่งเป็น register สำหรับ peripheral register 6ตัว ซึ่งประกอบด้วย DRR และ DXR (serial port register) , TIM (timer register) , PRD(period register) , IMP (interrupt mask register) และ GREG (global memory allocation register)

ภายใน TMS 320C26 ยังมี register file 8 ตัว คือ ARO-AR7 ไว้สำหรับการทำการอ้าง address โดยอัตโนมัติหรือเป็นที่เก็บข้อมูลชั่วคราว register ทั้ง 8 ตัว นี้สามารถเรียกใช้โดยตรงจากคำสั่งหรือจากการอ้างอิงเพียง 3 bit จาก ARP (Auxiliary Register Pointer) ARP นี้รับข้อมูลได้โดยตรงจากหน่วยความจำข้อมูลหรือมาจาก operand ของคำสั่ง

register file นี้ต่อโดยตรงเข้ากับ ARAU ดังแสดงในรูปที่ 2.2.3 ARAU สามารถทำการเพิ่มค่าหรือลดค่าเพื่อทำการ auto index ซึ่งทำให้การอ้าง address เพื่อนำข้อมูลจากตารางทำได้ง่ายและรวดเร็ว ดังนั้น ARAU จึงมีหน้าที่สำคัญสำหรับการสนับสนุน การอ้าง address ให้ได้ขีดความสามารถสูงขึ้น โครงสร้างของ ARAU แสดงได้ดังรูปที่ 2.2.3

นอกจากนี้ ARAU ยังสามารถเป็นหน่วยคำนวณทางคณิตศาสตร์ได้ด้วย โดยสามารถทำการคำนวณตัวเลขขนาด 16 bit ในขณะที่ CALU คำนวณตัวเลขขนาด 32 bit

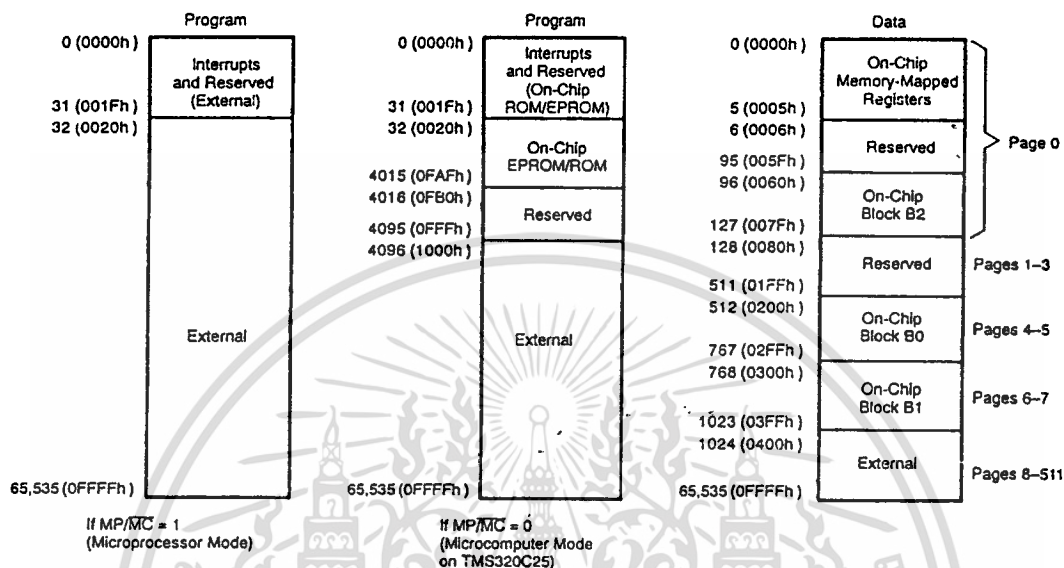
2.2.2 หน่วยคำนวณทางคณิตศาสตร์และลอจิกกลาง CALU

ประกอบด้วยวงจรถ่ายโอน bit ขนาด 16 bit วงจรคูณเลขขนาด 16x16 bit และ ALU ขนาด 32 bit โดยมี accumulator รับข้อมูลขนาด 32 bit วงจรเลื่อน bit จะทำการเลื่อน bit ไปทางซ้ายได้ตั้งแต่ 0-16 bit โดยโปรแกรมได้จากคำสั่งเมื่อเลื่อน bit แล้ว bit ทางขวามือจะได้รับการเติม 0 ให้

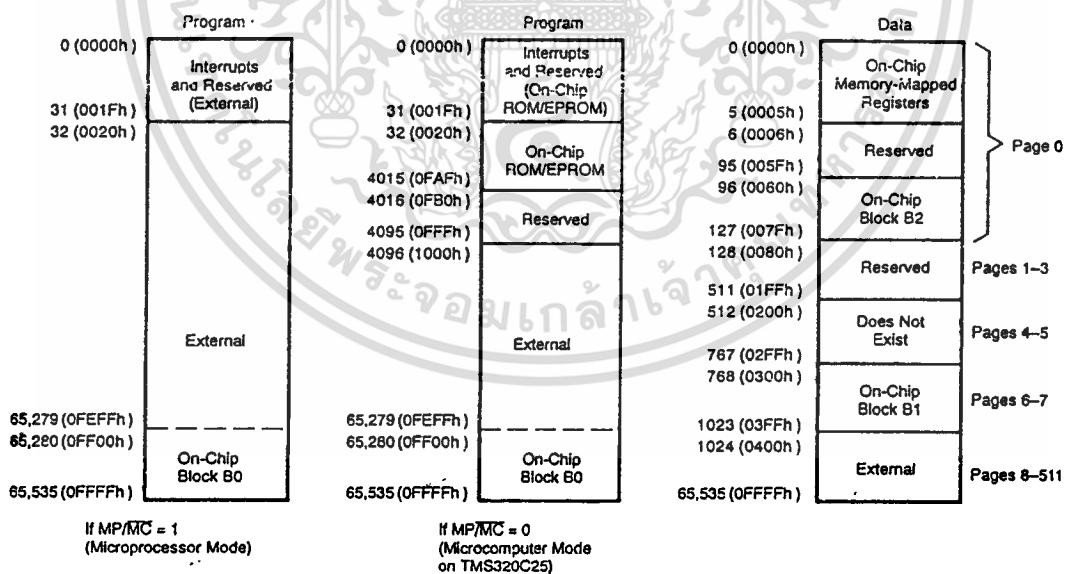
ALU ขนาด 32 bit จะทำงานตามคำสั่งทางคณิตศาสตร์และลอจิก สังเกตว่าการทำงานของ ALU นี้จะมีวงจรผ่านค่า เช่น ผ่านค่าการ shift bit แล้วมาทำใน ALU (ดูไดอะแกรมรูปที่ 2.2.3) ซึ่งทำให้การทำงานแต่ละคำสั่งทำงานได้เร็วมาก โดยเฉพาะคำสั่งการคูณบวกสำหรับการประมาณผลสัญญาณดังที่กล่าวมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(a) Memory Maps After a CNFD Instruction



(b) Memory Maps After a CNFP Instruction



รูปที่ 2.2.2 การจัดหน่วยความจำของ TMS 320C2X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การควบคุมการทำงาน

TMS 320C26 มีส่วนการควบคุมการทำงานสำคัญภายใน ซึ่งได้แก่วงจรตั้งเวลา วงจรนับให้เกิดการทำงาน maskable interrupt จากภายนอก 3 ตัว และ interrupt ภายในซึ่งเกิดจาก port อนุกรมหรือวงจรตั้งเวลา

วงจรตั้งเวลาประกอบด้วย register วงจรนับขนาด 16 bit ซึ่งเป็นวงจรนับลง (down counter) การทำงานจะทำการนับสัญญาณนาฬิกาจาก CLKOUT1 และเมื่อวงจรรับเวลาลดค่า register ลงมาเหลือ 0 ก็จะทำให้เกิดการ interrupt (TINT) และจากนั้นวงจรตั้งเวลาก็จะโหลดค่าจาก PRD เข้ายัง register วงจรนับใหม่เป็น cycle ต่อไป ดังนั้นช่วงเวลาที่เกิด interrupt แต่ละครั้งจึงเท่ากับ $(PRD+1) \times CLKOUT$ ซึ่งทำให้เกิดประโยชน์ในการกำหนดการ synchronize กับอุปกรณ์ภายนอก

ค่าของ register RPTC (repeat counter) เป็นตัวกำหนดจำนวนวงรอบของการทำงานของคำสั่ง ซึ่งนำค่ามาจากหน่วยความจำข้อมูลหรือจากคำสั่งโดยตรง การกำหนด RPTC จะทำให้คำสั่งได้รับการทำซ้ำเป็นจำนวนครั้งเท่ากับค่าที่โหลดไว้ ซึ่งจะทำให้การซ้ำได้ถึง 256 ครั้ง

การ interrupt CPU จากภายนอกจะมีขา interrupt 3 ขา คือ INTO ถึง INT2

2.2.5 พอร์ตอนุกรม

พอร์ตอนุกรมเป็นพอร์ตที่ใช้เชื่อมโยงกับโลกภายนอกโดยการรับส่งข้อมูลแบบอนุกรม การเชื่อมโยงข้อมูลนี้อาจจะเชื่อมต่อกับตัวอุปกรณ์ A/D และ D/A โดยวงจรเชื่อมต่อภายนอกจะต้องเป็น hardware ที่รับสัญญาณนี้ได้ ซึ่งภายใน chip มี register 2 ตัวที่ทำหน้าที่เก็บข้อมูล โดย register ตัวหนึ่งเป็น register ส่งข้อมูลอีกตัวหนึ่งเป็น register รับข้อมูล การรับข้อมูลจะรับข้อมูลนี้ได้รับการสร้างสัญญาณซิงค์จากภายใน ซึ่งสามารถรับส่งสัญญาณข้อมูลได้สูงถึง 5 MHz

ข้อปรับปรุงพอร์ตอนุกรมของ TMS 320C26 ได้แก่การสร้าง buffer ของตัวรับและตัวส่งข้อมูล และ CLKR_CLK คือ สัญญาณนาฬิกาสำหรับการรับส่งข้อมูล สามารถใช้สัญญาณนาฬิกาได้ตั้งแต่ 0 Hz เป็นต้นไป นอกจากนี้ยังเพิ่ม FSM-frame sync mode เพื่อให้ใช้กับระบบ โทรศัพท์แบบ PCM ตามมาตรฐาน AT&T T-1 และ CCITT G.711/712 จึงทำให้การประยุกต์ TMS 320C26 ในระบบโทรศัพท์ทำได้ง่าย

2.2.6 การเชื่อมต่อกับอินพุต-เอาต์พุต

TMS 320C26 มีอินพุต - เอาต์พุตพอร์ตอย่างละ 16 พอร์ต แต่ละพอร์ตเป็นพอร์ตแบบขนาน 16 บิต โดยเชื่อมต่อกับบัสข้อมูล คำสั่งที่เกี่ยวข้องกับ IN และ OUT จะใช้เวลา 2 cycle แต่ถ้าให้ทำงานแบบซ้ำโดยกำหนดค่าเข้าไปในรีจิสเตอร์ทำซ้ำแล้ว คำสั่ง IN หรือ OUT นี้จะให้เวลาได้เพียงไรเคลเดียว การเชื่อมโยงทาง I/O จะแยกออกจากระบบเพราะไมโครโปรเซสเซอร์ DSP นี้มีช่องทางสำหรับโปรแกรมข้อมูลและ I/O นอกจากนี้ยังสามารถมอง I/O เหมือนเป็นหน่วยความจำได้

บนตัวไอซีจะประกอบด้วย DO-D15 เป็น data bus A0-A15 เป็น address bus และยังประกอบด้วยขา ไอซี 3 ขา เพื่อเลือกว่าเป็นโปรแกรม, ข้อมูลหรืออินพุตเอาต์พุต (DS, PS และ IS) การกำหนดทิศทางจะใช้ RAW และมีสัญญาณ STRB เป็นตัวควบคุมการรับส่งข้อมูล

TMS 320C26 มีขา HOLD เพื่อใช้ทำ DMA โดยเมื่อขา this active จะทำให้สถานะของ address bus data bus และสัญญาณควบคุม อยู่ในสถานะ high impedance เพื่อให้อุปกรณ์ภายนอกติดต่อกับ ROM และ RAM ต่อไป

ตารางที่ 2.2 ก Interrupt Locations และ Priorities

INTERRUPT NAME	MEMORY LOCATION	PRIORITY	FUNCTION
\overline{RS}	0	1(highest)	External reset signal
$\overline{INT0}$	2	2	External user interrupt #0
$\overline{INT1}$	4	3	External user interrupt #1
$\overline{INT2}$	6	4	External user interrupt #2
	8 - 23		Reserved locations
TINT	24	5	Internal timer interrupt
RINT	26	6	Serialport receive interrupt
XINT	28	7(lowest)	Serialport transmit interrupt
TRAP	30	N/A	TRAP instruction address

2.2.7 INTERRUPT

การจัดการ interrupt ของ TMS 320C26 แสดงได้ดังตารางที่ 2.2 ก ซึ่งเป็นการ interrupt ทาง hardware ทั้งหมดยกเว้น TRAP เป็นการ interrupt ทาง software แต่ละ interrupt address จะใช้ เนื้อที่ 2 address

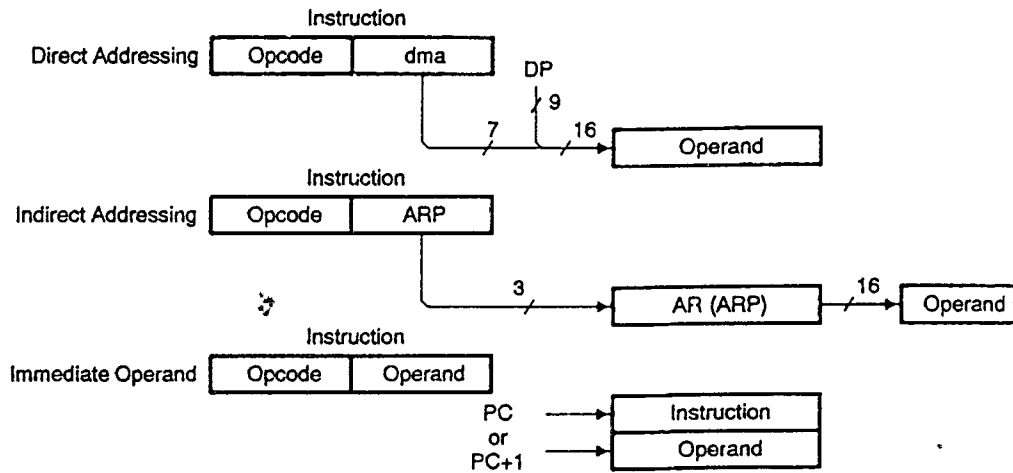
2.2.8 โหมดการอ้างแอสแตรของ TMS 320C26

คำสั่งส่วนใหญ่ของ TMS 320C26 เป็นคำสั่งที่ใช้รหัส 16 bit เพื่อว่าการ fetch และ execute จะกระทำได้เพียง cycle address 8 bit การอ้าง address จะสามารถอ้างได้ 3 mode ดังรูปที่ 2.2.4 คือแบบอ้าง address โดยตรง (direct) การอ้าง address โดยอ้อม (indirect) และแบบ immediate การอ้าง address โดยตรง และโดยอ้อมเป็นการต้องการติดต่อกับหน่วยความจำ ส่วนการอ้างอิง address แบบ immediate เป็นการนำเอา operand มาใช้โดยตรง

การอ้าง address โดยตรงจะใช้คำสั่ง 16 bit โดยแยกออกเป็น opcode 9 bit และอีก 7 bit เป็น operand ในการทำงานจะใช้ 7 bit ร่วมกับ 9 bit จาก page pointer (DP) เพื่อรวมให้เป็น 16 bit สำหรับอ้างหน่วยความจำ 64 Kwords ดังนั้นการจัดหน่วยความจำ จึงแบ่งเป็น page 128 word และมีจำนวน 512 page

การอ้าง address แบบทางอ้อมจะอ้างผ่าน register AR0 - AR7 register เหล่านี้เป็น register ตัวนับ loop หรือสำหรับเก็บข้อมูลชั่วคราวก็ได้ รูปที่ 2.2.5 เป็นการชี้ register AR3 เป็นตัวอ้าง address ไปยังหน่วยความจำ การกำหนด register ทำได้โดยการใช้ register ชั่ว (Auxiliary Register Pointer) ARP ขนาด 3 bit ซึ่งมีค่าได้จาก 0-7 เพื่อแทน AR0 - AR7 ค่าใน ARP นี้สามารถ load เข้าโดยตรงจากข้อมูลในหน่วยความจำ หรือจากคำสั่งที่มีการกำหนดค่าแบบ immediate operand และเราสามารถ load ข้อมูลจาก register ชั่วเหล่านี้กลับเข้าหน่วยความจำได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

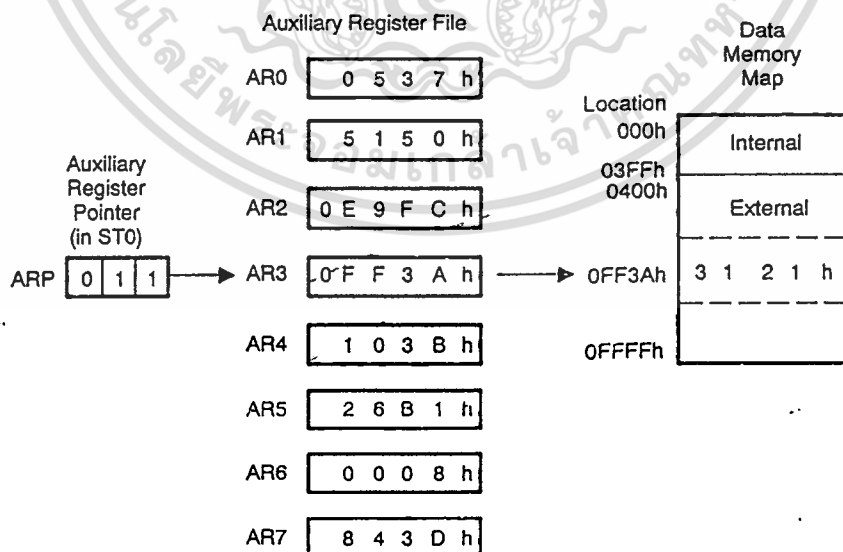


รูปที่ 2.2.4 Methods of Instruction Operand Addressing

จากตารางที่ 2.2 ข จะเห็นว่ามีการอ้างแอดเดรสแบบทางอ้อม 7 แบบ คือ

- การ index ด้วยการเพิ่มค่า
- การ index ด้วยการลดค่า
- การ index โดยการบวกข้อมูลของ AR0
- การ index โดยการลบข้อมูลของ AR0
- การ index โดยการบวกข้อมูลของ AR0 สำหรับการทำให้ bit-reversing ใน algorithm FFT
- การ index โดยการลบข้อมูลของ AR0 สำหรับการทำให้ bit-reversing ใน algorithm FFT
- ไม่มีการindex

การอ้าง address แบบ immediate จะมีส่วนของ operand ประกอบอยู่ในคำสั่ง โดยค่าตัวเลขนี้จะเป็นไปได้ 2 ขนาด คือ แบบ 8 bit หรือแบบ 13 bit และสำหรับคำสั่งแบบที่มีขนาด 2 word จะมีขนาดของตัวเลข operand ที่ใช้งานขนาด 16 bit



รูปที่ 2.2.5 Indirect Auxiliary Register Addressing Example

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข

โหมดแอดเดรส	ลักษณะการอ้าง	การทำงาน
OP A	แอดเดรสโดยตรง	
OP. (NARP)	แอดเดรสโดยทางอ้อม	ไม่มีการเปลี่ยนค่าใน AR
OP.+ (NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR เดิมเพิ่มขึ้น 1
OP. - (NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR เดิมลดค่าลง 1
OP.0 + (NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR_0 บวกกับค่าของ AR เดิม
OP.0 - (NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR_0 ลบกับค่าของ AR เดิม
OP.BRO+(NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR_0 บวกกับค่าของ AR เดิม โดยมีการให้บิตทดกระจายไว้
OP.BRO- (NARP)	แอดเดรสโดยทางอ้อม	ค่าของ AR_0 ลบจากค่าของ AR เดิม โดยมีการให้บิตทดกระจายไว้

2.2.8 ชุดคำสั่งของ TMS320C26

สำหรับ TMS320C26 นี้ได้ปรับปรุงโครงสร้างทางสถาปัตยกรรมเพิ่มขึ้นอีก ดังนั้นจึงมีชุดคำสั่งที่เพิ่มขึ้นอีกหลายคำสั่งและบางคำสั่งก็มีความสามารถสูงขึ้น สัญลักษณ์ที่ใช้แทนคำสั่งมีดังนี้

ACC - accumulator

ARB - register ช่วยที่ buffer

ARn - register ช่วย (auxiliary register) ตัวที่ n

ARP - ตัวที่ register ช่วย (auxiliary register pointer)

BIO - Branch control input

C - bit ตัวทศ

CM - 2 bit ที่ใช้สำหรับการกำหนดโหมดการเปรียบเทียบ (compare mode)

CNF - bit ที่ใช้สำหรับการควบคุม RAM บนชิพ (configuration control bit)

dma - data address

DP - pointer บอก page ข้อมูล (data page pointer)

FO - bit บอกสถานะ format (format status bit)

FOM - bit บอก mode frame synchronize

HM - bit บอก mode การ hold

INTM - bit flag mode interrupt

>nn - บอกว่า nn คือ เลขฐาน 16

OV - flag overflow

OVM - bit mode overflow

P - register ผลคูณ

PA - address port โดยใช้ PA0 หมายถึง port0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PC - program counter
 PM - 2bit ที่บอก P register ให้เอาที่หลุดเลื่อนบิต
 pma - program counter
 Prog - register ผลคูณ
 RPTC - counter นับทำซ้ำ
 STn - registerสถานะ (ST0 หรือ ST1)
 SXM - bit บอก mode sign extension
 T - register ตัวครว
 TC - bit ทดสอบการควบคุม (test control register)
 TOS - จุดสูงสุดของ stack (top of stack)
 Treg - register ตัวครว
 TXM - bit บอกmode ส่งข้อมูลอนุกรม
 Usg - คำที่ไม่คิดเครื่องหมาย
 XF - bit บอกสถานะของขา XF

ตารางที่ 2.2 ค เป็นชุดคำสั่งของ TMS320C26 ซึ่งได้ให้รายละเอียดเป็นกลุ่มคำสั่งไว้

ตารางที่ 2.2 ค ชุดคำสั่งของ TMS320C26

CONTROL INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE
			15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
BIT†	Test bit	1	1 0 0 1 ←8→ M ←D→
BITT†	Test bit specified by T register	1	0 1 0 1 0 1 1 1 M ←D→
CONF0#	Configure all blocks as Data	1	1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 0
CONF1#	Configure block B0 as program	1	1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1
CONF2#	Configure blocks B0 and B1 as program	1	1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0
CONF3#	Configure blocks B0, B1 and B3 as program	1	1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 1
DINT	Disable interrupt	1	1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1
EINT	Enable interrupt	1	1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0
IDLE†	Idle until interrupt	1	1 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1
LST	Load status register ST0	1	0 1 0 1 0 0 0 0 M ←D→
LST1†	Load status register ST1	1	0 1 0 1 0 0 0 1 M ←D→
NOP	No operation	1	0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0
POP	Pop top of stack to low accumulator	1	1 1 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1
POPD†	Pop top of stack to data memory	1	0 1 1 1 1 0 1 0 M ←D→
PSHD†	Push data memory value onto stack	1	0 1 0 1 0 1 0 0 M ←D→
PUSH	Push low accumulator onto stack	1	1 1 0 0 1 1 1 0 0 0 0 1 1 1 0 0
RC	Reset carry bit	1	1 1 0 0 1 1 1 0 0 0 1 1 0 0 0 0
RHM	Reset hold mode	1	1 1 0 0 1 1 1 0 0 0 1 1 1 0 0 0
ROVM	Reset overflow mode	1	1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 0
RPT†	Repeat instruction as specified by data memory value	1	0 1 0 0 1 0 1 1 M ←D→
RPTK†	Repeat instruction as specified by immediate value	1	1 1 0 0 1 0 1 1 ←K→
RSXM†	Reset sign-extension mode	1	1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 0
RTC	Reset test/control flag	1	1 1 0 0 1 1 1 0 0 0 1 1 0 0 1 0
SC	Set carry bit	1	1 1 0 0 1 1 1 0 0 0 1 1 0 0 0 1
SHM	Set hold mode	1	1 1 0 0 1 1 1 0 0 0 1 1 1 0 0 1
SOVM	Set overflow mode	1	1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1
SST	Store status register ST0	1	0 1 1 1 1 0 0 0 M ←D→
SST1†	Store status register ST1	1	0 1 1 1 1 0 0 1 M ←D→
SSXM†	Set sign-extension mode	1	1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1
STC	Set test/control flag	1	1 1 0 0 1 1 1 0 0 0 1 1 0 0 1 1
TRAP†	Software interrupt	;	1 1 0 0 1 1 1 0 0 0 0 1 1 1 1 0

† These instructions are not included in the TMS32010 instruction set.

เอกสาร* These instructions replace CNFD and CNFP in the TMS320C25 instruction set. นั่น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ค ชุดคำสั่งของ TMS320C26 (ต่อ)

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE																
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ABS	Absolute value of accumulator	1	1	1	0	0	1	1	1	0	0	0	0	1	1	0	1	1	
ADD	Add to accumulator with shift	1	0	0	0	0	←S→	M	←D→										
ADDC	Add to accumulator with carry	1	0	1	0	0	0	0	1	1	M	←D→							
ADDH	Add to high accumulator	1	0	1	0	0	1	0	0	0	M	←D→							
ADDK	Add to accumulator short immediate	1	1	1	0	0	1	1	0	0	←K→	←D→							
ADDS	Add to low accumulator with sign extension suppressed	1	0	1	0	0	1	0	0	1	M	←D→							
ADDT [†]	Add to accumulator with shift specified by T register	1	0	1	0	0	1	0	1	0	M	←D→							
ADLK [†]	Add to accumulator long immediate with shift	2	1	1	0	1	←S→	0	0	0	0	0	0	1	0				
AND	AND with accumulator	1	0	1	0	0	1	1	1	0	M	←D→							
ANDK [†]	AND immediate with accumulator with shift	2	1	1	0	1	←S→	0	0	0	0	0	1	0	0				
CMPL [†]	Complement accumulator	1	1	1	0	0	1	1	1	0	0	←D→	0	1	0	0	1	1	1
LAC	Load accumulator with shift	1	0	0	1	0	←S→	M	←D→										
LACK	Load accumulator immediate short	1	1	1	0	0	1	0	1	0	←K→	←D→							
LACT [†]	Load accumulator with shift specified by T register	1	0	1	0	0	0	0	1	0	M	←D→							
LALK [†]	Load accumulator long immediate with shift	2	1	1	0	1	←S→	0	0	0	0	0	0	0	1				
NEG [†]	Negate accumulator	1	1	1	0	0	1	1	0	0	0	←D→	0	1	0	0	0	1	1
NORM [†]	Normalize contents of accumulator	1	1	1	0	0	1	1	1	0	1	←D→							
OR	OR with accumulator	1	0	1	0	0	1	1	0	1	M	←D→							
ORK [†]	OR immediate with accumulator with shift	2	1	1	0	1	←S→	0	0	0	0	1	0	1	0				
ROL	Rotate accumulator left	1	1	1	0	0	1	1	1	0	0	←D→	0	1	1	0	1	0	1
ROR	Rotate accumulator right	1	1	1	0	0	1	1	1	0	0	←D→	0	1	1	0	1	0	1
SACH	Store high accumulator with shift	1	0	1	1	0	1	←X→	M	←D→									
SACL	Store low accumulator with shift	1	0	1	1	0	0	←X→	M	←D→									
SBLK [†]	Subtract from accumulator long immediate with shift	2	1	1	0	1	←S→	0	0	0	0	0	0	1	1				
SFL [†]	Shift accumulator left	1	1	1	0	0	1	1	1	0	0	←D→	0	0	1	1	0	0	0
SFR [†]	Shift accumulator right	1	1	1	0	0	1	1	1	0	0	←D→	0	0	1	1	0	0	1
SUB	Subtract from accumulator with shift	1	0	0	0	1	←S→	M	←D→										
SUBB	Subtract from accumulator with borrow	1	0	1	0	0	1	1	1	1	M	←D→							
SUBC	Conditional subtract	1	0	1	0	0	0	1	1	1	M	←D→							
SUBH	Subtract from high accumulator	1	0	1	0	0	0	1	0	0	M	←D→							
SUBK	Subtract from accumulator short immediate	1	1	1	0	0	1	1	0	1	←K→	←D→							
SUBS	Subtract from low accumulator with sign extension suppressed	1	0	1	0	0	0	1	0	1	M	←D→							
SUBT [†]	Subtract from accumulator with shift specified by T register	1	0	1	0	0	0	1	1	0	M	←D→							
XOR	Exclusive-OR with accumulator	1	0	1	0	0	1	1	0	0	M	←D→							
XORK [†]	Exclusive OR immediate with accumulator with shift	2	1	1	0	1	←S→	0	0	0	0	0	1	1	0				
ZAC	Zero accumulator	1	1	1	0	0	1	0	1	0	0	←D→	0	0	0	0	0	0	0
ZALH	Zero low accumulator and load high accumulator	1	0	1	0	0	0	0	0	0	M	←D→							
ZALR	Zero low accumulator and load high accumulator with rounding	1	0	1	1	1	1	0	1	1	M	←D→							
ZALS	Zero accumulator and load low accumulator with sign extension suppressed	1	0	1	0	0	0	0	1	M	←D→								

[†] These instructions are not included in the TMS32010 instruction set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ค ชุดคำสั่งของ TMS320C26 (ต่อ)

AUXILIARY REGISTERS AND DATA PAGE POINTER INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE
			15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ADRK	Add to auxiliary register short immediate	1	0 1 1 1 1 1 1 0 ←K→
CMPR [†]	Compare auxiliary register with auxiliary register ARO	1	1 1 0 0 1 1 1 0 0 1 0 1 0 0 ←CM→
LAR	Load auxiliary register	1	0 0 1 1 0 ←R→ M ←D→
LARK	Load auxiliary register short immediate	1	1 1 0 0 0 ←R→ ←K→
LARP	Load auxiliary register pointer	1	0 1 0 1 0 1 0 1 1 0 0 0 1 ←R→
LDP	Load data memory page pointer	1	0 1 0 1 0 0 1 0 M ←D→
LDPK	Load data memory page pointer immediate	1	1 1 0 0 1 0 0 ←DP→
LRLK [†]	Load auxiliary register long immediate	2	1 1 0 1 0 ←R→ 0 0 0 0 0 0 0 0
MAR	Modify auxiliary register	1	0 1 0 1 0 1 0 1 M ←D→
SAR	Store auxiliary register	1	0 1 1 1 0 ←R→ M ←D→
SBRK	Subtract from auxiliary register short immediate	1	0 1 1 1 1 1 1 1 ←K→
T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE
			15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
APAC	Add P register to accumulator	1	1 1 0 0 1 1 1 0 0 0 1 0 1 0 1
LPH [†]	Load high P register	1	0 1 0 1 0 0 1 1 M ←D→
LT	Load T register	1	0 0 1 1 1 1 0 0 M ←D→
LTA	Load T register and accumulate previous product	1	0 0 1 1 1 1 0 1 M ←D→
LTD	Load T register, accumulate previous product, and move data	1	0 0 1 1 1 1 1 1 M ←D→
LTP [†]	Load T register and store P register in accumulator	1	0 0 1 1 1 1 1 0 M ←D→
LTS [†]	Load T register and subtract previous product	1	0 1 0 1 1 0 1 1 M ←D→
MAC [†]	Multiply and accumulate	2	0 1 0 1 1 1 0 : M ←D→
MACD [†]	Multiply and accumulate with data move	2	0 1 0 1 1 1 0 0 M ←D→
MPY	Multiply (with T register, store product in P register)	1	0 0 1 1 1 0 0 0 M ←D→
MPYA	Multiply and accumulate previous product	1	0 0 1 1 1 0 1 0 M ←D→
MPYK	Multiply immediate	1	1 0 1 ←K→
MPYS	Multiply and subtract previous product	1	0 0 1 1 1 0 1 1 M ←D→
MPYU	Multiply unsigned	1	1 1 0 0 1 1 1 1 M ←D→
PAC	Load accumulator with P register	1	1 1 0 0 1 1 1 0 0 0 1 0 1 0 0
SPAC	Subtract P register from accumulator	1	1 1 0 0 1 1 1 0 0 0 1 1 0
SPH	Store high P register	1	0 1 1 1 1 1 0 1 M ←D→
SPL	Store low P register	1	0 1 1 1 1 1 0 0 M ←D→
SPM [†]	Set P register output shift mode	1	1 1 0 0 1 1 1 0 0 0 0 1 0 ←PM→
SQRA [†]	Square and accumulate	1	0 0 1 1 1 0 0 1 M ←D→
SQRS [†]	Square and subtract previous product	1	0 1 0 1 1 0 1 0 M ←D→

[†]These instructions are not included in the TMS32010 instruction set.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ค ชุดคำสั่งของ TMS320C26 (ต่อ)

BRANCH/CALL INSTRUCTIONS																		
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE															
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally,	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	
BACC [†]	Branch to address specified by accumulator	1	1	1	0	0	1	1	1	0	0	0	1	0	0	1	0	1
BANZ	Branch on auxiliary register not zero	2	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
BBNZ [†]	Branch if TC bit = 0	2	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
BBZ [†]	Branch if TC bit = 0	2	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1
BC	Branch on carry	2	0	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1
BGEZ	Branch if accumulator ≥ 0	2	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1
BGZ	Branch if accumulator > 0	2	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1
BIOZ	Branch on I/O status = 0	2	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
BLEZ	Branch if accumulator ≤ 0	2	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1
BLZ	Branch if accumulator < 0	2	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
BNC	Branch on no carry	2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
BNV [†]	Branch if no overflow	2	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
BNZ	Branch if accumulator = 0	2	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1
BV	Branch on overflow	2	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
BZ	Branch if accumulator = 0	2	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1
CALA	Call subroutine indirect	1	1	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0
CALL	Call subroutine	2	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
RET	Return from subroutine	1	1	1	0	0	1	1	1	0	0	0	1	0	0	1	1	0

I/O AND DATA MEMORY OPERATIONS																			
MNEMONIC	DESCRIPTION	NO. WORDS	INSTRUCTION BIT CODE																
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
BLKD [†]	Block move from data memory to data memory	2	1	1	1	1	1	1	0	1	M	←D	←D	←D	←D	←D	←D	←D	←D
BLKP [†]	Block move from program memory to data memory	2	1	1	1	1	1	1	0	0	M	←D	←D	←D	←D	←D	←D	←D	←D
DMOV	Data move in data memory	1	0	1	0	1	0	1	1	0	M	←D	←D	←D	←D	←D	←D	←D	←D
FORT [†]	Format serial port registers	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	1	FO	
IN	Input data from port	1	1	0	0	0	←PA	←M	←D	←D	←D	←D	←D	←D	←D	←D	←D	←D	
OUT	Output data to port	1	1	1	1	0	←PA	←M	←D	←D	←D	←D	←D	←D	←D	←D	←D	←D	
RFSM	Reset serial port frame synchronization mode	1	1	1	0	0	1	1	1	0	0	0	1	1	0	1	1	0	
RTXM [†]	Reset serial port transmit mode	1	1	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	
RXF [†]	Reset external flag	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0	
SFSM	Set serial port frame synchronization mode	1	1	1	0	0	1	1	1	0	0	0	1	1	0	1	1	1	
STXM [†]	Set serial port transmit mode	1	1	1	0	0	1	1	1	0	0	0	1	0	0	0	0	1	
SXF [†]	Set external flag	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	1	
TBLR	Table read	1	0	1	0	1	1	0	0	0	M	←D	←D	←D	←D	←D	←D	←D	
TBLW	Table write	1	0	1	0	1	1	0	0	1	M	←D	←D	←D	←D	←D	←D	←D	

[†] These instructions are not included in the TMS32010 instruction set.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ฟูเรียส(Fourier)

หัวข้อนี้เราจะศึกษาถึงการวิเคราะห์สัญญาณด้วยอนุกรมฟูเรียสที่เราสามารถนำมาใช้แสดงสัญญาณเป็นคาบ โดยจะให้อยู่ในรูปการรวมเอาสัญญาณรูปคลื่นไซน์จำนวนอนันต์เข้าด้วยกัน ต่อจากนั้นก็จะมีการพัฒนาการแปลงฟูเรียสที่ทำงานในลักษณะเดียวกัน แต่ใช้วิเคราะห์สัญญาณที่ไม่เป็นคาบผลลัพธ์จากการวิเคราะห์จะให้สเปกตรัม (Spectrum) หรือแถบความถี่ของสัญญาณที่ถูกวิเคราะห์ทำให้รู้ว่าสัญญาณนี้ประกอบด้วยความถี่อะไรบ้าง

2.3.1 อนุกรมฟูเรียส

ให้สัญญาณ $f(t)$ เป็นสัญญาณที่ซ้ำ ๆ กันเป็นคาบทุก ๆ T วินาที ซึ่งสามารถเขียนฟังก์ชันของสัญญาณ $f(t)$ แทนได้ด้วยอนุกรมฟูเรียส กล่าวคือ

$$f(t) = a_0 + \sum_{n=1}^N a_n \cos n\omega t + \sum_{n=1}^N b_n \sin n\omega t \quad (2.3-1)$$

เมื่อ a_n, b_n เป็นสัมประสิทธิ์ที่สามารถคำนวณได้จากสมการข้างล่างนี้

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos n\omega t dt \quad (2.3-2)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin n\omega t dt \quad (2.3-3)$$

เมื่อ T เป็นคาบเวลา และ $\omega = 2\pi f$ เทอม dc คือ a_0 ซึ่งให้ค่าเฉลี่ยของสัญญาณ $f(t)$ ในช่วงคาบ T คำนวณได้ดังนี้

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt \quad (2.3-4)$$

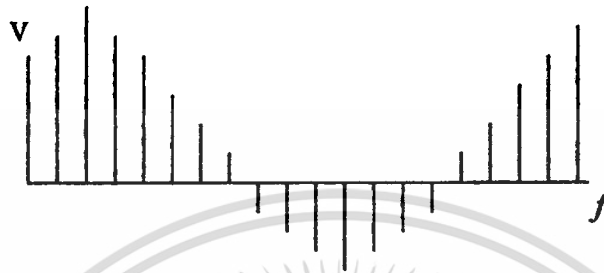
ข้อสังเกต

1. ถ้า $f(t) = f(-t)$ ฟังก์ชัน $f(t)$ เป็นฟังก์ชันคู่ จะเกิดความสมมาตรรอบจุดกำเนิด และจะให้เพียงเทอมของ COSINE ในสมการ (2.3-1)
2. ถ้า $f(t) = -f(-t)$ ฟังก์ชัน $f(t)$ เป็นฟังก์ชันคี่ จะปรากฏเพียงเทอมของ SINE เท่านั้น ในสมการ (2.3-1)
3. ถ้า $f(t+T/2) = f(t)$ ในสมการ (2.3-1) จะปรากฏเพียงเทอมฮาร์โมนิคคู่ (EVEN HARMONICS)
4. ถ้า $f(t+T/2) = -f(t)$ ในสมการ (2.3-1) จะปรากฏเพียงเทอมฮาร์โมนิคคี่ (ODD HARMONICS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 แถบความถี่ที่ไม่ต่อเนื่อง (DISCRETE SPECTRUM)

อนุกรมฟูเรียสที่ใช้แทนฟังก์ชันในแกนเวลา $f(t)$ โดยเราจะแสดงให้เห็นองค์ประกอบทางความถี่ของสัญญาณนั้น องค์ประกอบความถี่เหล่านี้รวมกันเป็นแถบความถี่ที่ไม่ต่อเนื่อง โดยขนาดของแต่ละความถี่ถูกกำหนดโดยค่าสัมประสิทธิ์ a_n และ b_n ทุกความถี่จะเป็นฮาร์มอนิกของความถี่พื้นฐานที่เท่ากับ $1/T$ และช่วงกว้างของความถี่ต่าง ๆ นี้จะเรียกว่าแบนด์วิดท์ของสัญญาณ



รูปที่ 2.3.1 แถบความถี่ที่ไม่ต่อเนื่อง

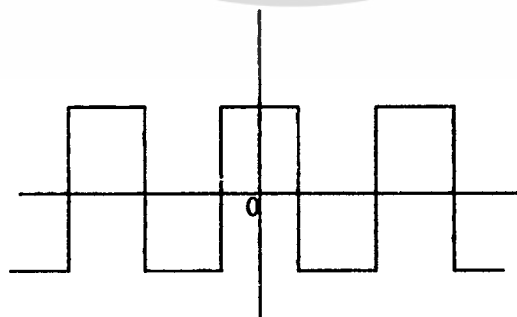
2.3.3 ตัวอย่างชนิดของอนุกรม (TYPICAL SERIES)

ถึงแม้แถบความถี่อาจจะประกอบด้วยความถี่ต่าง ๆ จำนวนอนันต์ก็ตาม ขนาดของความถี่จะลดลงเรื่อยๆ เมื่อความถี่สูงขึ้น (ค่าเพิ่มขึ้น) สำหรับในทางปฏิบัติแล้วเราพิจารณาเพียงความถี่จำนวนจำกัดที่พอเพียงกับการติดต่อสื่อสาร

ข้อควรคำนึงในการประมวลผลสัญญาณเชิงเลข โดยเฉพาะในด้านการสื่อสารนั้น จุดสำคัญของการติดต่อสื่อสาร คือ การประหยัดในการใช้แถบความถี่ (band width) ในระบบการติดต่อสื่อสาร ถ้าหากเรารู้ถึงแถบของความถี่จะช่วยให้ระบบการส่งและการรับสัญญาณทำได้อย่างมีประสิทธิภาพและช่วยให้ประหยัดด้วย สำหรับลักษณะรูปคลื่นสัญญาณที่ปรากฏในระบบการประมวลผลสัญญาณเชิงเลขมีหลายรูปแบบ ในที่นี้จะแสดงอนุกรมสัญญาณเฉพาะที่ใช้กันมาก

องค์ประกอบของฟูเรียสสำหรับรูปคลื่นตัวอย่างชนิดต่าง ๆ พอจะยกตัวอย่างได้ดังนี้

1. รูปคลื่นสี่เหลี่ยมสมมาตร (SYMMETRICAL SQUARE WAVE)



รูปที่ 2.3.2 รูปคลื่นแบบสมมาตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก $f(t)$ มีลักษณะสมมาตรทางแนวนอน ค่าเฉลี่ยของพื้นที่จึงเป็นศูนย์ ทำให้เทอม dc มีค่าเท่ากับศูนย์ ($a_0=0$) และโดยที่ $f(t) = f(-t)$ จึงทำให้สัมประณาดังกล่าวจะมีเฉพาะเทอมของ COSINE เท่านั้น กล่าวคือ $b_n=0$ ส่วนค่า a_n คำนวณได้ดังนี้

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos n\omega t dt$$

$$f(t) = -1 \quad \text{จาก } -T/2 \text{ ถึง } -T/4$$

$$f(t) = 1 \quad \text{จาก } -T/4 \text{ ถึง } T/4$$

$$f(t) = -1 \quad \text{จาก } T/4 \text{ ถึง } T/2$$

$$\begin{aligned} a_n &= \frac{2}{T} \left\{ \int_{-T/2}^{-T/4} (-1) \cos n\omega t dt + \int_{-T/4}^{T/4} (1) \cos n\omega t dt + \int_{T/4}^{T/2} (-1) \cos n\omega t dt \right\} \\ &= \frac{2}{T} \left\{ -\frac{1}{n\omega} \int_{-T/2}^{-T/4} \cos n\omega t dt + \frac{1}{n\omega} \int_{-T/4}^{T/4} \cos n\omega t dt - \frac{1}{n\omega} \int_{T/4}^{T/2} \cos n\omega t dt \right\} \\ &= \frac{2}{n\omega T} \left\{ -[\sin n\omega t]_{-T/2}^{-T/4} + [\sin n\omega t]_{-T/4}^{T/4} - [\sin n\omega t]_{T/4}^{T/2} \right\} \\ &= \frac{2}{n\omega T} \left\{ \sin \frac{n\omega T}{4} - \sin \frac{n\omega T}{2} + \sin \frac{n\omega T}{4} + \sin \frac{n\omega T}{4} - \sin \frac{n\omega T}{2} + \sin \frac{n\omega T}{4} \right\} \\ &= \frac{2}{n\omega T} \left\{ 4 \sin \frac{n\omega T}{4} - 2 \sin \frac{n\omega T}{2} \right\} \end{aligned}$$

เนื่องจาก $\omega T = (2\pi f)T = 2\pi$ ดังนั้นจะได้

$$a_n = \frac{1}{n\pi} \left\{ 4 \sin \frac{nT}{2} - 2 \sin n\pi \right\}$$

แต่ $\sin n\pi = 0$ เมื่อ $n = 0, 1, 2, \dots, \infty$

$$\text{ดังนั้น } a_n = \frac{4}{n\pi} \sin \frac{n\pi}{2}$$

$$a_1 = \frac{4}{\pi} \sin \frac{\pi}{2} = \frac{4}{\pi}$$

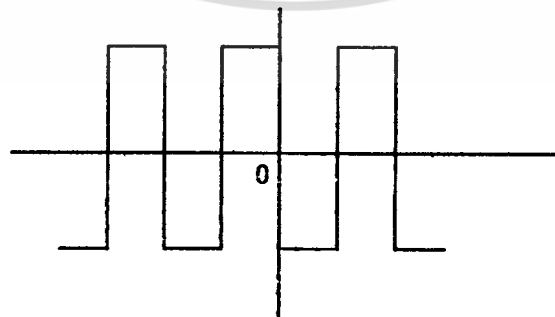
$$a_2 = \frac{2}{\pi} \sin \pi = 0$$

$$a_3 = \frac{4}{3\pi} \sin \frac{3\pi}{2} = -\frac{4}{3\pi}$$

$$a_4 = \frac{1}{\pi} \sin 2\pi = 0$$

$$f(t) = \frac{4}{\pi} \left(\cos \omega t - \frac{1}{3} \cos 3\omega t + \cos 5\omega t \dots \right)$$

2. รูปคลื่นสี่เหลี่ยมแบบไม่สมมาตร (ASYMMETRICAL SQUARE WAVE)



รูปที่ 2.3.2 รูปคลื่นแบบไม่สมมาตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน $f(t)$ เป็นรูปคลื่นที่สมมาตรทางแนวนอน ดังนั้นค่าเฉลี่ยได้พื้นที่จึงเป็นศูนย์ทำให้เทอม $a_0=0$ และเนื่องจาก $f(t) = f(-t)$ จะได้เทอม $a_n=0$ เหลือเฉพาะเทอม b_n ที่ต้องคำนวณ

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin n\omega t dt$$

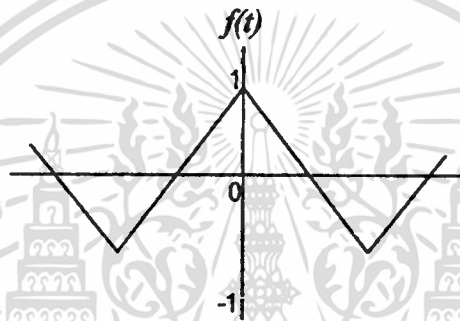
$$f(t) = -1 \quad \text{จาก } -T/2 \text{ ถึง } 0$$

$$= 1 \quad \text{จาก } 0 \text{ ถึง } T/2$$

จะได้

$$f(t) = \frac{4}{\pi} \left[\sin \omega t + \frac{1}{3} \sin 3\omega t + \frac{1}{5} \sin 5\omega t + \dots \right]$$

3. รูปคลื่นสามเหลี่ยม (TRIANGULAR WAVE)

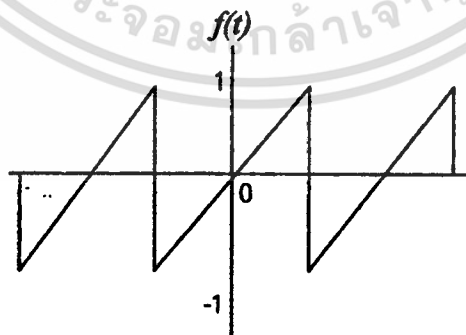


รูปที่ 2.3.4 รูปคลื่นสามเหลี่ยม

จากรูปข้างบน จะได้ว่า

$$f(t) = \frac{8}{\pi^2} \left(\cos \omega t + \frac{1}{9} \cos 3\omega t + \frac{1}{25} \cos 5\omega t + \dots \right)$$

4. รูปคลื่นฟันปลา (SAWTOOTH WAVE)



รูปที่ 2.3.5 รูปคลื่นฟันปลา

จากรูปข้างบน จะได้ว่า

$$f(t) = \frac{2}{\pi} \left(\sin \omega t - \frac{1}{2} \sin 2\omega t + \frac{1}{3} \sin 3\omega t + \dots \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 รูปคอมเพล็กซ์ (COMPLEX FORM)

ฟังก์ชัน $f(t)$ นี้สามารถเขียนแทนได้ด้วยปริมาตรคอมเพล็กซ์ อีกวิธีหนึ่งโดยจากเทอมของสัญญาณรูปคอมเพล็กซ์ ซึ่งจะเห็นว่า

$$\cos n\omega t = \frac{e^{jn\omega t} + e^{-jn\omega t}}{2}$$

$$\sin n\omega t = \frac{e^{jn\omega t} - e^{-jn\omega t}}{2j}$$

ทำการแทนค่าของ $\cos n\omega t$ และ $\sin n\omega t$ ลงในอนุกรมของฟูเรียร์ ก็จะได้

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \left(\frac{e^{jn\omega t} + e^{-jn\omega t}}{2} \right) + \sum_{n=1}^{\infty} b_n \left(\frac{e^{jn\omega t} - e^{-jn\omega t}}{2j} \right)$$

$$= a_0 + \sum_{n=1}^{\infty} \left\{ \frac{(a_n - jb_n)e^{jn\omega t}}{2} + \frac{(a_n + jb_n)e^{-jn\omega t}}{2} \right\}$$

กำหนดให้

$$c_n = \frac{1}{2}(a_n - jb_n)$$

$$c_{-n} = \frac{1}{2}(a_n + jb_n)$$

$$c_0 = a_0 \frac{1}{T}$$

โดย c_{-n} เป็น Complex conjugate ของ c_n ดังนั้นการคำนวณหาค่า c_n ทำได้โดย

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) [\cos n\omega t - j \sin n\omega t] dt$$

$$c_{-n} = \frac{1}{T} \int_{-T/2}^{T/2} f(t) [\cos n\omega t + j \sin n\omega t] dt$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cdot e^{-jn\omega t} dt$$

$$c_{-n} = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cdot e^{jn\omega t} dt$$

ในที่สุดจะได้

$$f(t) = c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega t} + \sum_{n=-\infty}^{-1} c_n e^{jn\omega t}$$

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega t}$$

2.3.5 การแปลงฟูเรียร์ FOURIER TRANSFORM)

เทคนิคของอนุกรมฟูเรียร์สามารถปรับปรุงมาใช้กับรูปคลื่นที่ไม่เป็นคาบอย่างเช่น **ลูกพัลส์** ใดๆ ๆ

(simple pulse) โดยการให้ค่า T เป็นค่าอนันต์ (∞)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติว่า $f(t)$ เดิมเป็นสัญญาณที่เป็นคาบจะได้ว่า

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega t}$$

$$\text{เมื่อ } c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \cdot e^{-jn\omega t} dt$$

จากเงื่อนไขที่ว่าสัญญาณนั้นเป็นพัลส์เดี่ยว ๆ ทำให้

$$T \rightarrow \infty ; \quad \omega = \frac{2\pi}{T} \rightarrow d\omega$$

$$\text{ดังนั้น } \frac{1}{T} = \frac{\omega}{2\pi} \rightarrow \frac{d\omega}{2\pi}$$

ยิ่งไปกว่านั้น ฮาร์โมนิกที่ n ของอนุกรมฟูเรียร์ $n\omega$ จะกลายเป็น $nd\omega$ ซึ่งถือว่ามีค่าเป็น ω นั้นเอง ในที่สุดเครื่องหมายซิกมา (Sigma) จะกลายเป็นเครื่องหมายอินทิกรัล (Integral) ดังนั้น

$$c_n = \frac{d\omega}{2\pi} \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$$

$$f(t) = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \left[\int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \right] e^{j\omega t}$$

เทอมที่อยู่ในเครื่องหมายก้ามปู เป็นเทอมของความถี่เพียงอย่างเดียวซึ่งให้เป็น $g(\omega)$ โดย

$$g(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt$$

ซึ่ง $g(\omega)$ เรียกว่าการแปลงฟูเรียร์ของฟังก์ชัน $f(t)$ ดังนั้นสมการของ $f(t)$ จะกลายเป็น

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g(\omega) \cdot e^{j\omega t} d\omega$$

2.3.5.1 ความต่อเนื่องของสเปกตรัม (CONTINUOUS SPECTRUM)

สัญญาณที่มีลักษณะเป็นพัลส์เดี่ยว ๆ สามารถเขียนแทนด้วยผลรวมของความถี่ต่าง ๆ จำนวนอนันต์ ทั้งนี้เนื่องจาก $g(\omega)$ มี ω เป็นเทอมของความถี่ จากการอินทิเกรตนี้เองทำให้ได้ว่าสเปกตรัมที่ได้มีความต่อเนื่องตลอด ซึ่งตรงกันข้ามกับกรณีของรูปคลื่นที่เป็นคาบที่ให้สเปกตรัมไม่ต่อเนื่อง โดยความจริงแล้วทุกความถี่จะอยู่ใกล้กันมาก ทั้งนี้เพราะช่องว่างระหว่างความถี่นั้น คือ $1/T$ จะมีค่าเป็นศูนย์เมื่อ T เป็น ∞

โดยทั่วไปแล้ว $g(\omega)$ เป็นเทอมคอมเพล็กซ์ โดยที่ขนาดและเฟสสามารถนำมาพล็อตเป็นแถบความถี่ของสัญญาณ $f(t)$ ขนาด $|g(\omega)|$ จะแปรไปตามคาบและ $|g(\omega)|$ เป็นพื้นที่ใต้กราฟภายในช่วง $d\omega$ ถูกเรียกว่า สเปกตรัลเดนซิตี (SPECTRAL DENSITY) แต่เนื่องจาก $d\omega/2\pi$ มีค่าเกือบเป็นหนึ่ง ดังนั้นพื้นที่ของ $|g(\omega)|d\omega/2\pi$ จึงมีค่าเพียงขนาดของ $|g(\omega)|$ ซึ่งเป็นแอมพลิจูดเดนซิตี (AMPLITUDE DENSITY)

2.3.5.2 การแปลงฟูเรียร์ของสัญญาณไม่ต่อเนื่อง (Discrete Fourier transform)

จากความเจริญก้าวหน้าทางเทคโนโลยี ทำให้ปัจจุบันมีเครื่องคอมพิวเตอร์ใช้กันอย่างกว้างขวาง ด้วยเหตุนี้เองการคำนวณการแปลงฟูเรียร์ด้วยการอินทิเกรต จึงไม่นิยมกันเพราะยุ่งยากและเสียเวลาทำให้มีการคิด

โปรแกรมการแปลงฟูเรียร์ขึ้น ดังนั้นสัญญาณต่อเนื่องที่จะนำแปลงด้วยฟูเรียร์จึงต้องมีการสุ่ม (SAMPLING) ให้เป็นสัญญาณที่ไม่ต่อเนื่องเพื่อนำเอาขนาดของแอมป์เปิดไปคำนวณนี้

สูตรในการแปลงฟูเรียร์ของสัญญาณต่อเนื่องคือ

$$G(\omega) = \int_{-\infty}^{\infty} g(t) \cdot e^{-j2\pi f t} dt$$

$$g(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j2\pi f t} dt$$

เมื่อแปลงให้เป็นการแปลงฟูเรียร์ของสัญญาณที่ไม่ต่อเนื่องจะได้

$$G(u) = \frac{1}{N} \sum_{x=0}^{N-1} g(x) e^{-\frac{j2\pi ux}{N}} \quad ; u = 0, 1, 2, \dots, N-1$$

เมื่อ N เป็นจำนวนแอมป์เปิด ในการแปลงกลับฟูเรียร์ของสัญญาณที่ไม่ต่อเนื่อง คือ

$$g(x) = \sum_{u=0}^{N-1} G(u) e^{-\frac{j2\pi ux}{N}} \quad ; x = 0, 1, 2, \dots, N-1$$

ตัวอย่าง จากสัญญาณไม่ต่อเนื่องในรูปข้างบนจะให้ $N = 4$ โดย $g(x) = \{2, 3, 4, 4\}$ สเปกตรัมของสัญญาณดังกล่าวคำนวณได้จากสูตรการแปลงฟูเรียร์ของสัญญาณที่ไม่ต่อเนื่องคือ

$$G(u) = \frac{1}{4} \sum_{x=0}^3 g(x) e^{-\frac{j2\pi ux}{4}}$$

$$G(u) = \frac{1}{4} \sum_{x=0}^3 g(x) = \frac{1}{4} [g(0) + g(1) + g(2) + g(3)]$$

$$= \frac{1}{4} [2 + 3 + 4 + 4]$$

$$= 3.25$$

สูตรการแปลงฟูเรียร์ของสัญญาณต่อเนื่อง

$$H(f) = \int_{-\infty}^{\infty} h(t) \cdot e^{-j2\pi f t} dt$$

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(f) \cdot e^{j\omega t} d\omega$$

สูตรการแปลงฟูเรียร์ของสัญญาณที่ไม่ต่อเนื่อง

$$H(k) = \frac{1}{N} \sum_{m=0}^{N-1} h(m) e^{-\frac{j2\pi km}{N}} \quad ; k = 0, 1, 2, \dots, N-1$$

$$h(m) = \sum_{k=0}^{N-1} H(k) e^{-\frac{j2\pi km}{N}} \quad ; m = 0, 1, 2, \dots, N-1$$

จาก sequence ของสัญญาณ $h(m) = \{2, 3, 4, 4\}$ ให้หา spectrum ของสัญญาณนั้นคือ $N=4$

$$H(k) = \frac{1}{4} \sum_{m=0}^3 h(m) e^{-\frac{j2\pi km}{4}} = \frac{1}{4} \sum_{m=0}^3 h(m) e^{-\frac{j\pi km}{2}}$$

$$H(0) = \frac{1}{4} \sum_{m=0}^3 h(m)$$

$$= \frac{1}{4} [h(0) + h(1) + h(2) + h(3)]$$

$$= \frac{1}{4} [2 + 3 + 4 + 4] = 3.25$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 H(1) &= \frac{1}{4} \sum_{m=0}^3 h(m) e^{-\frac{j\pi km}{2}} \\
 &= \frac{1}{4} [h(0)e^{-j0} + h(1)e^{-\frac{j\pi}{2}} + h(2)e^{-j\pi} + h(3)e^{-\frac{j3\pi}{2}}] \\
 &= \frac{1}{4} [2(1) + 3(-j) + 4(-1) + 4(j)] \\
 &= \frac{1}{4} [2 - 3j - 4 + 4j] \\
 &= \frac{1}{4} [-2 + j] \\
 H(2) &= \frac{1}{4} \sum_{m=0}^3 h(m) e^{-j\pi x} \\
 &= \frac{1}{4} [h(0)e^{-j0} + h(1)e^{-j\pi} + h(2)e^{-j2\pi} + h(3)e^{-j3\pi}] \\
 &= \frac{1}{4} [2(1) + 3(-1) + 4(1) + 4(-1)] \\
 &= \frac{1}{4} [2 - 3 + 4 - 4] \\
 &= \frac{1}{4} [2 - 3 + 4 - 4] \\
 &= -0.25 \\
 H(3) &= \frac{1}{4} \sum_{m=0}^3 h(m) e^{-\frac{j3\pi m}{2}} \\
 &= \frac{1}{4} [h(0)e^{-j0} + h(1)e^{-\frac{j3\pi}{2}} + h(2)e^{-j3\pi} + h(3)e^{-\frac{j9\pi}{2}}] \\
 &= \frac{1}{4} [2(1) + 3(j) + 4(-1) + 4(-j)] \\
 &= \frac{1}{4} [2 + 3j - 4 - 4j] \\
 &= -\frac{1}{4} [2 + j]
 \end{aligned}$$

เพราะฉะนั้น สเปกตรัมคือ

$$\begin{aligned}
 |H(0)| &= 3.25 \\
 |H(1)| &= \frac{1}{4} \sqrt{(-2)^2 + (1)^2} = \frac{\sqrt{5}}{4} \\
 |H(2)| &= |-0.25| = 0.25 \\
 |H(3)| &= \frac{1}{4} \sqrt{(2)^2 + (1)^2} = \frac{\sqrt{5}}{4}
 \end{aligned}$$

2.3.5.3 การแปลงฟูเรียร์อย่างรวดเร็ว (Fast Fourier Transform)

ขั้นตอนวิธี หรือลำดับการในการคำนวณฟูเรียร์ให้เร็วมีชื่อเรียกว่า การแปลงฟูเรียร์อย่างรวดเร็ว คิดค้นโดย คูลีย์ (J.W. Cooley) กับ ทูคีย์ (J.W. Tukey) ซึ่งได้เสนอไว้ในปี ค.ศ. 1965 หลังจากนั้นทำให้เกิดการพัฒนาวิธีหลายวิธีแต่ในวิธีของคูลีย์และ ทูคีย์ ช่วยให้การคำนวณเชิงซ้อนเพียง $N \log_2 N$ หรือลดจำนวนครั้งในการคูณตัวเลขลดลงไปถึง $N/\log_2 N$ เท่า ผลดีอีกประการหนึ่งก็คือทำให้การสร้างวงจรเฉพาะเพื่อการคำนวณ DFT ทำได้ง่าย และคำนวณได้เร็วขึ้น

จากหัวข้อที่ผ่านมาแสดงถึงการแปลงฟูเรียร์สำหรับสัญญาณไม่ต่อเนื่อง จะสังเกตได้ว่าจะมีส่วนที่เราต้องคำนวณซ้ำ ๆ คือ การคูณค่าจำนวนเชิงซ้อน ซึ่งมักเป็นเลขทศนิยมความละเอียด 1-2 เท่า กันมาก ทำให้เสียเวลาในการคำนวณ ดังในตารางแสดงให้เห็นถึงจำนวนที่ลดลงเมื่อมีการใช้การแปลงฟูเรียร์อย่างรวดเร็ว เช่น N ขนาด 1024 จุด เมื่อใช้ DFT จะต้องคูณจำนวนเชิงซ้อน 1048576 แต่เมื่อใช้ FFT จะเหลือเพียง 10240 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหัวข้อที่แล้วเราได้สูตรการแปลงฟูเรียร์ คือ

$$H(k) = \frac{1}{N} \sum_{m=0}^{N-1} h(m)W^{km} \quad ; k = 0, 1, 2, \dots, N-1$$

โดยที่ $W = e^{-\frac{j2\pi}{N}} \Rightarrow j = \sqrt{-1}$

ตารางที่ 2.3 ก เปรียบเทียบจำนวนการคูณค่าเชิงซ้อนระหว่าง DFT กับ FFT

Length of Transform (N)	DFT Operations (N^2)	FFT Operations $N \log_2(N)$
8	64	24
16	256	64
32	1024	160
64	4096	384
128	16384	896
256	65536	1024
512	262144	4096
1024	1048576	10240
2048	4194304	22528

คุณสมบัติและเทคนิคช่วยแปลงฟูเรียร์อย่างเร็วมีดังนี้

1. คุณสมบัติของการแปลงฟูเรียร์ (Motivation to search for an algorithm)

$$H\left(\frac{N}{2} + l\right) = \overline{H\left(\frac{N}{2} - l\right)} \quad ; l = 1, 2, \dots, N/2 - 1$$

โดยที่ $\overline{H(k)}$ เป็น Complex Conjugate ของ $H(k)$

2. เทคนิคสำหรับการลดการคำนวณ (Key to Developing the algorithm)

$$m = m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_12^1 + m_02^0$$

โดยที่ $m_v = 0$ หรือ 1 เมื่อ $v = 0, 1, \dots, n-1$ โดย $n = \log_2 N$

ทำนองเดียวกัน

$$k = k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \dots + k_12^1 + k_02^0$$

โดยที่ $k_v = 0$ หรือ 1 เมื่อ $v = 0, 1, \dots, n-1$ โดย $n = \log_2 N$

ถ้าเราให้ $\overline{h}(m)$ เป็น sequence ของข้อมูลที่ตัวแปรเป็นเลขฐานสองที่ใช้แทน $h(m)$ ซึ่งเป็นเลขฐานสิบเราจะได้ว่า

$$\begin{aligned} h(m) &= \overline{h}(m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_12^1 + m_02^0) \\ &= \overline{h}(m_{n-1}, m_{n-2}, \dots, m_1, m_0) \end{aligned} \quad (2.3-5)$$

จากสมการที่ (2.3-5) จะได้ว่า

$$\sum_{m=0}^{N-1} h(m)W^{km} = \sum_{m_0=0}^1 \sum_{m_1=0}^1 \dots \sum_{m_{n-1}=0}^1 \bar{h}(m_{n-1}, m_{n-2}, \dots, m_0) W^{k[m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_0 2^0]} \quad (2.3-6)$$

ตัวอย่างในกรณีนี้ที่ $N=4$ หรือ $n=\log_2 4=2$ ดังนั้นสมการที่ (2) จะเป็น

$$\begin{aligned} \sum_{m_0} \sum_{m_1} \bar{h}(m_1, m_0) W^{k(2m_1+m_0)} &= \sum_{m_0} \{ \bar{h}(0, m_0) W^{km_0} + \bar{h}(1, m_0) W^{k(2+m_0)} \} \\ &= \sum_{m_0=0}^1 \bar{h}(0, m_0) W^{km_0} + \sum_{m_0=0}^1 \bar{h}(1, m_0) W^{k(2+m_0)} \\ &= \bar{h}(0,0) + \bar{h}(0,1)W^k + \bar{h}(1,0)W^{2k} + \bar{h}(1,1)W^{3k} \end{aligned} \quad (2.3-7)$$

ดังนั้นสมการที่ (2.3-7) จะได้ว่า

$$\sum_{m_0} \sum_{m_1} \bar{h}(m_1, m_0) W^{k(2m_1+m_0)} = \sum_{m=0}^3 h(m) W^{km} \quad \text{นั่นเอง}$$

ตัวอย่างการนำไปใช้งาน เราเอากรณีนี้ที่ $N=8$ เป็นพื้นฐานเบื้องต้นในการอธิบายขบวนการของ FFT

นั่นคือ

$$H(k) = \frac{1}{8} \sum_{m=0}^7 h(m) W^{km} \quad \text{โดย } k = 0, 1, \dots, 7 \quad (2.3-8)$$

เมื่อ $W = e^{-\frac{j2\pi}{8}} = e^{-\frac{j\pi}{4}}$ และเราให้ m เขียนอยู่ในรูป binary คือ

$$m = m_2 2^2 + m_1 2^1 + m_0 2^0 \quad (2.3-9)$$

จากสมการ (2.3-8) เอา 8 คูณตลอดจะได้

$$8H(k) = \sum_{m=0}^7 h(m) W^{km} \quad (2.3-10)$$

เมื่อแปลงเทอมทางขวามือของสมการที่ (6) ให้อยู่ในรูปของเลขฐานสอง จะได้ว่า

$$\begin{aligned} 8H(k) &= \sum_{m_0=0}^1 \sum_{m_1=0}^1 \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) W^{k[4m_2+2m_1+m_0]} \\ &= \sum_{m_0=0}^1 \sum_{m_1=0}^1 \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) W^{k4m_2} W^{2km_1} W^{km_0} \end{aligned} \quad (2.3-11)$$

จากสมการที่ (2.3-11) เราให้ Summation ในสุดเป็น M_2 นั่นคือ

$$M_2 = \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) W^{4km_2}$$

แต่เนื่องจาก $W^4 = -1$ และแทน k ด้วยเลขฐานสองจะได้

$$M_2 = \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) (-1)^{m_2[4k_2+2k_1+k_0]}$$

แต่เนื่องจาก $(-1)^{m_2[4k_2+2k_1+k_0]} = 1$, M_2 สามารถเขียนใหม่เป็น

$$M_2 = \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) (-1)^{m_2[4k_2+2k_1+k_0]} \quad (2.3-12)$$

Summation ของสมการที่ (2.3-12) เป็นการแทนค่า m_2 ด้วย 0 และ 1 ดังนั้นสมการที่ (2.3-12) จะมีตัว

แปรที่ไม่ทราบค่าคือ k_0 , m_1 และ m_0 ดังนั้น M_2 เขียนใหม่ได้เป็น

$$M_2 = \sum_{m_2=0}^1 \bar{h}(m_2, m_1, m_0) (-1)^{m_2[4k_2+2k_1+k_0]} = \bar{h}(k_0, m_1, m_0)$$

(2.3-13)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าสมการ (2.3-13) ลงในสมการ (2.3-11)

$$8H(k) = \sum_{m_0=0}^1 \sum_{m_1=0}^1 \bar{h}_1(k_0, m_1, m_0) W^{2km_1} W^{km_0} \quad (2.3-14)$$

Summation ชุดในของสมการที่ (2.3-14) เราสมมุติให้เป็น M_1 นั่นคือ

$$\begin{aligned} M_1 &= \sum_{m_1=0}^1 \bar{h}_1(k_0, m_1, m_0) W^{2km_1} \\ &= \sum_{m_1=0}^1 \bar{h}_1(k_0, m_1, m_0) (-j)^{(2k_1+k_0)m_1} \end{aligned} \quad (2.3-15)$$

โดยที่จากสมการที่ (2.3-15) นั้น $W^2 = e^{-\frac{j2\pi 2}{4}} = -j$ และเทอม $(-j)^{4k_2m_1} = 1$

ดังนั้น M_1 เขียนได้ใหม่เป็น $M_1 = \sum_{m_1=0}^1 \bar{h}_1(k_0, m_1, m_0) (-j)^{(2k_1+k_0)m_1} \quad (2.3-16)$

จากสมการที่ (2.3-16) เมื่อทำการแปรค่า M_1 เป็น 0 และ 1 ดังนั้นเทอม M_1 จะเป็นตัวแปรของ k_0, k_1 และ m_0 นั่นคือสมการ (2.3-16) จึงกลายเป็น

$$M_1 = \bar{h}_2(k_0, k_1, m_0) \quad (2.3-17)$$

แทน (2.3-17) ลงใน (2.3-13) จะได้

$$8H(k) = \sum_{m_0=0}^1 \bar{h}_2(k_0, k_1, m_0) W^{m_0 k_2}$$

ในทำนองเดียวกับ M_2 และ M_1 เมื่อให้ M_0 เป็นผลการ summation ของค่า M_0 นั่นคือ

$$\begin{aligned} M_0 &= \sum_{m_0=0}^1 \bar{h}_2(k_0, k_1, m_0) W^{m_0 [4k_2+2k_1+k_0]} \\ &= \sum_{m_0=0}^1 \bar{h}_2(k_0, k_1, m_0) \left(\frac{1-j}{\sqrt{2}}\right)^{[4k_2+2k_1+k_0]m_0} \end{aligned} \quad (2.3-18)$$

หลังจากทำการแปรค่า M_0 ไปแล้วจะพบว่า M_0 เป็นฟังก์ชันของ k_0, k_1 และ k_2 ซึ่งเราจะแทน

ด้วย

$$M_0 = \bar{h}_3(k_0, k_1, k_2) \quad (2.3-19)$$

ดังนั้นจากสมการที่ (2.3-11) เราจะได้คำตอบว่า

$$8H(k) = 8\bar{H}(k_2, k_1, k_0) = \bar{h}_3(k_0, k_1, k_2) \quad (2.3-20)$$

จากสัมประสิทธิ์ $(-1), (-j)$ และ $(1-j)/\sqrt{2}$ ค่าเหล่านี้จะเป็นรากของ $e^{-j2\pi}$ นั่นเอง กล่าวคือ

$(-1), (-j)$ และ $(1-j)/\sqrt{2}$ เป็นรากที่ 2, 4 และที่ 8 ของ unity ซึ่งเราจะใช้สัญลักษณ์แทนด้วย

$$A_2 = e^{-\frac{j2\pi}{2^r}} \quad \text{เมื่อ } r = 1, 2, \dots, \log_2 N \quad (2.3-21)$$

โดย A_2 จะมีคุณสมบัติคือ (i) $A_2 = W^{\frac{N}{2^r}}$; $W = e^{-\frac{j2\pi}{N}}$ (2.3-22 a)

(ii) $(A_2)^{\lambda+i} = -(A_2)^i$ เมื่อ $r = 1, 2, \dots, \log_2 N; i = 0, 1, \dots, 2^{r-1}$
 $\lambda = 2^{r-1}$ (2.3-22 b)

(iii) $(A_N)^{\frac{N}{2}} = -1$ (2.3-22 c)

จากสมการที่ (2.3-13) เราเขียนอยู่ในเทอม \bar{h}_1 จะได้ว่า

$$\bar{h}_1(k_0, m_1, m_0) = \sum_{m_2} \bar{h}(m_2, m_1, m_0) A_2^{k_0}$$

นั่นคือ

$$\bar{h}_1(k_0, m_1, m_0) = \bar{h}(0, m_1, m_0) + \bar{h}(1, m_1, m_0) A_2^{k_0} \quad (2.3-23)$$

สมการที่ (2.3-22) นี้ จะมีค่าไม่เป็น 0 ก็เป็น 1 จะได้ว่าค่า แต่ละค่าให้ 4 สมการ คือ

case ที่ 1 $k_0=0$

$$\begin{aligned} \bar{h}_1(0,0,0) &= \bar{h}(0,0,0) + \bar{h}(1,0,0) \Rightarrow h_1(0) = h(0) + h(4) \\ \bar{h}_1(0,0,1) &= \bar{h}(0,0,1) + \bar{h}(1,0,1) \Rightarrow h_1(1) = h(1) + h(5) \\ \bar{h}_1(0,1,0) &= \bar{h}(0,1,0) + \bar{h}(1,1,0) \Rightarrow h_1(2) = h(2) + h(6) \\ \bar{h}_1(0,1,1) &= \bar{h}(0,1,1) + \bar{h}(1,1,1) \Rightarrow h_1(3) = h(3) + h(7) \end{aligned} \quad (2.3-24 a)$$

case ที่ 2 $k_0=1$

$$\begin{aligned} \bar{h}_1(1,0,0) &= \bar{h}(0,0,0) + A_2 \bar{h}(1,0,0) \Rightarrow h_1(4) = h(0) - h(4) \\ \bar{h}_1(1,0,1) &= \bar{h}(0,0,1) + A_2 \bar{h}(1,0,1) \Rightarrow h_1(5) = h(1) - h(5) \\ \bar{h}_1(1,1,0) &= \bar{h}(0,1,0) + A_2 \bar{h}(1,1,0) \Rightarrow h_1(6) = h(2) - h(6) \\ \bar{h}_1(1,1,1) &= \bar{h}(0,1,1) + A_2 \bar{h}(1,1,1) \Rightarrow h_1(7) = h(3) - h(7) \end{aligned} \quad (2.3-24 b)$$

สมการที่ (2.3-16) เมื่อแทน $(-j)$ ด้วยเทอม A_4 จะได้ว่า

$$\bar{h}_2(k_0, m_1, m_0) = \bar{h}_1(k_0, 0, m_0) + \bar{h}_1(k_0, 1, m_0) A_4^{(2k_1 - k_0)}$$

จากการ fixed ค่า k_0, k_1 แล้วทำการแปร m_0 ไปแต่ละครั้งจะได้ 2 สมการดังนี้ คือ

case 1 $(k_1, k_0) = (0, 0)$ (2.3-25 a)

$$\begin{aligned} \bar{h}_2(0,0,0) &= \bar{h}_1(0,0,0) + \bar{h}_1(0,1,0) \Rightarrow h_2(0) = h_1(0) + h_1(2) \\ \bar{h}_2(0,0,1) &= \bar{h}_1(0,0,1) + \bar{h}_1(0,1,1) \Rightarrow h_2(1) = h_1(1) + h_1(3) \end{aligned}$$

case 2 $(k_1, k_0) = (0, 1)$ (2.3-25 b)

$$\begin{aligned} \bar{h}_2(1,0,0) &= \bar{h}_1(1,0,0) + A_4 \bar{h}_1(1,1,0) \Rightarrow h_2(4) = h_1(4) + A_4 h_1(6) \\ \bar{h}_2(1,0,1) &= \bar{h}_1(1,0,1) + A_4 \bar{h}_1(1,1,1) \Rightarrow h_2(5) = h_1(5) + A_4 h_1(7) \end{aligned}$$

case 3 $(k_1, k_0) = (1, 0)$ (2.3-25 c)

$$\begin{aligned} \bar{h}_2(0,1,0) &= \bar{h}_1(0,0,0) + A_4^2 \bar{h}_1(0,1,0) \Rightarrow h_2(2) = h_1(0) - h_1(2) \\ \bar{h}_2(0,1,1) &= \bar{h}_1(0,0,1) + A_4^2 \bar{h}_1(0,1,1) \Rightarrow h_2(3) = h_1(1) - h_1(3) \end{aligned}$$

case 4 $(k_1, k_0) = (1, 1)$ (2.3-25 d)

$$\begin{aligned} \bar{h}_2(1,1,0) &= \bar{h}_1(1,0,0) + A_4^2 \bar{h}_1(1,1,0) \Rightarrow h_2(6) = h_1(4) - A_4 h_1(6) \\ \bar{h}_2(1,1,1) &= \bar{h}_1(1,0,1) + A_4^2 \bar{h}_1(1,1,1) \Rightarrow h_2(7) = h_1(5) - A_4 h_1(7) \end{aligned}$$

สมการที่ (2.3-25 a) ถึง (2.3-25 d) เป็นการทำให้ iteration ครั้งที่ 2 ดังรูปที่ 2.3.6 โดยการทำให้ iteration ครั้งที่ 2 นี้คือ ในสมการที่ (2.3-21) นั่นเอง

ในที่สุดสมการที่ (2.3-19) เราเขียนค่าสัมประสิทธิ์ในเทอมของ A_8 จะได้ว่า

$$\bar{h}_3(k_0, k_1, k_2) = \bar{h}_2(k_0, k_1, 0) + \bar{h}_2(k_0, k_1, 1) A_8^{[4k_2 + 2k_1 + k_0]}$$

ซึ่งจะได้ว่า

case 1 (k_2, k_1, k_0) = (0, 0, 0) (2.3-26 a)

$$\bar{h}_3(0,0,0) = \bar{h}_2(0,0,0) + \bar{h}_2(0,0,1) \Rightarrow h_3(0) = h_2(0) + h_2(1)$$

case 2 (k_2, k_1, k_0) = (0, 0, 1) (2.3-26 b)

$$\bar{h}_3(1,0,0) = \bar{h}_2(1,0,0) + A_8 \bar{h}_2(1,0,1) \Rightarrow h_3(4) = h_2(4) + A_8 h_2(5)$$

case 3 (k_2, k_1, k_0) = (0, 1, 0) (2.3-26 c)

$$\bar{h}_3(0,1,0) = \bar{h}_2(0,1,0) + A_8^2 \bar{h}_2(0,1,1) \Rightarrow h_3(2) = h_2(2) + A_8^2 h_2(3)$$

case 4 (k_2, k_1, k_0) = (0, 1, 1) (2.3-26 d)

$$\bar{h}_3(1,1,0) = \bar{h}_2(1,1,0) + A_8^3 \bar{h}_2(1,1,1) \Rightarrow h_3(6) = h_2(6) + A_8^3 h_2(7)$$

case 5 (k_2, k_1, k_0) = (1, 0, 0) (2.3-26 e)

$$\bar{h}_3(0,0,1) = \bar{h}_2(0,0,0) + A_8^4 \bar{h}_2(0,0,1) \Rightarrow h_3(1) = h_2(0) - A_8^4 h_2(1)$$

สมการที่เหลือเราใช้คุณสมบัติของ $H\left(\frac{N}{2} + l\right) = \overline{H\left(\frac{N}{2} - l\right)}$ นั้นเอง จากกลุ่มสมการที่ (23) นี้เป็น
การทำ iteration ครั้งที่ 3 นั่นคือ $r = 3$ ในสมการที่ (17) นั้นเอง

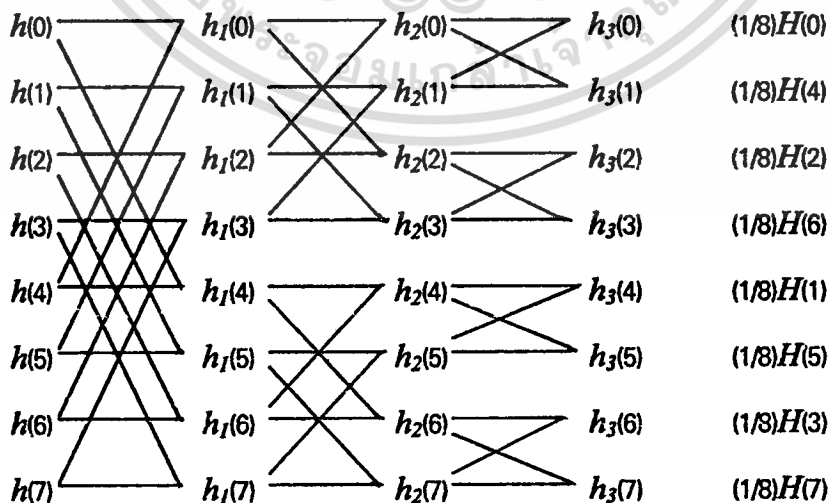
ในที่สุดจะได้ว่า

$$\begin{aligned} h_3(0) &= 8H(0) \\ h_3(4) &= 8H(1) \\ h_3(2) &= 8H(2) \\ h_3(6) &= 8H(3) \\ h_3(1) &= 8H(4) \end{aligned}$$

ส่วน $H(5)$, $H(6)$ และ $H(7)$ ได้จากคุณสมบัติทาง Complex Conjugate หรือถ้าคำนวณจะพบว่า

$$\begin{aligned} 8H(5) &= h_3(5) = h_2(4) - A_8 h_2(5) \\ 8H(6) &= h_3(3) = h_2(2) - A_8^2 h_2(3) \\ 8H(7) &= h_3(7) = h_2(6) - A_8^3 h_2(7) \end{aligned}$$

จากสมการที่ (20), (21) และ (23) จะได้ว่า Signal flow graph $N=8$ คือ



รูปที่ 2.3.6 Signal flow graph $N=8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า $\{h(n)\} = \{1, 2, 1, 1, 3, 2, 1, 2\}$ ให้หา $H(k)$

$h(0) = 1$	$h_1(0) = 4$	$h_2(0) = 13$	$\rightarrow 1/8 \rightarrow$	$H(0)$
$h(1) = 2$	$h_1(1) = 4$	$h_2(1) = -1$	$\rightarrow 1/8 \rightarrow$	$H(4)$
$h(2) = 1$	$h_1(2) = 2$	$h_2(2) = 2-j$	$\rightarrow 1/8 \rightarrow$	$H(2)$
$h(3) = 1$	$h_1(3) = 3$	$h_2(3) = 2+j$	$\rightarrow 1/8 \rightarrow$	$H(6)$
$h(4) = 3$	$h_1(4) = -2$	$h_2(4) = -1.293+j0.707$	$\rightarrow 1/8 \rightarrow$	$H(1)$
$h(5) = 2$	$h_1(5) = 0$	$h_2(5) = -2.707-j0.707$	$\rightarrow 1/8 \rightarrow$	$H(5)$
$h(6) = 1$	$h_1(6) = 0$	$h_2(6) = -2.707+jj0.707$	$\rightarrow 1/8 \rightarrow$	$H(3)$
$h(7) = 2$	$h_1(7) = -1$	$h_2(7) = -1.293-j0.707$	$\rightarrow 1/8 \rightarrow$	$H(7)$

2.3.5 เติมนั้น (DECIMATION)

เติมนั้น คือ การลดทอนจำนวนของสัญญาณลง ตัวอย่างเช่น สัญญาณอินพุตถูกสุ่มที่ความถี่ 8000 Hz เพราะฉะนั้นใน 1 วินาที จะมีพัลส์ที่ถูกสุ่มจำนวน 8000 ค่าเราสามารถลดจำนวนสัญญาณลงได้โดยไม่ทำให้สัญญาณสูญหายไป จากทฤษฎีการสุ่มสัญญาณจะต้องมีจำนวนมากกว่าขนาดความถี่เป็นจำนวนสองเท่า แต่เราสามารถลดสัญญาณลงมาได้อีก เช่นจากสัญญาณอินพุตขนาด 8000 Hz เราทำการเติมนั้นลง 2 : 1 เอาต์พุตใหม่ขนาด 4000 Hz เพื่อมิให้สัญญาณสูญหายไป เมื่อต้องการคืนรูปสัญญาณจำเป็นต้องอาศัยขั้นตอนวิธีอินเตอร์โพลเลชัน

2.3.6 อินเตอร์โพลเลชัน (INTERPOLATION)

สืบเนื่องจากเมื่อมีการเติมนั้นสัญญาณ เมื่อต้องการคืนรูปสัญญาณจำเป็นต้องอาศัยขั้นตอนวิธีอินเตอร์โพลเลชัน การอินเตอร์โพลเลชันจะต้องใช้ให้ถูกกับการเติมนั้น ตัวอย่างเช่น เติมนั้นขนาด 2:1 จำเป็นต้องอินเตอร์โพลเลชัน 1:2 เหมือนเดิม สำหรับขั้นตอนการอินเตอร์โพลเลชันมีหลายวิธี เช่น ใช้ IIR FILTER หรือ ฟิวเรียสทรานส์ฟอร์มก็ได้ ในที่นี้จะแสดงเฉพาะวิธี ฟิวเรียส

สำหรับขั้นตอนการอินเตอร์โพลเลชันขนาด 1:2 ด้วยฟิวเรียส มีดังนี้

1. แปลงฟิวเรียสทรานส์ฟอร์มตามขนาดกำลัง 2 ของสัญญาณ
2. เติมค่า 0 ลงในโดเมนความถี่ที่ทำการแปลงฟิวเรียสจำนวน $N-1$ ตัว โดยใส่ระหว่างครึ่งบวกและครึ่งลบของสเปกตรัม และทำการหารค่ากึ่งกลางระหว่างสเปกตรัมด้วย 2 ตามทฤษฎีของโควิท

2.4 ทฤษฎีของการแปลงสัญญาณ (Data Acquisition and Conversion)

รูปแบบของสัญญาณทางไฟฟ้า ที่เราคุ้นเคยกันแต่เดิมนั้น โดยมากมักจะอยู่ในรูปของสัญญาณขนาดออก แต่ก่อนนั้นการที่จะนำเอาสัญญาณทางไฟฟ้ามาประมวลผลเพื่อให้เกิดรูปแบบที่ต้องการนั้นจะกระทำได้โดยใช้อุปกรณ์ทางอนาลอกนั่นเอง

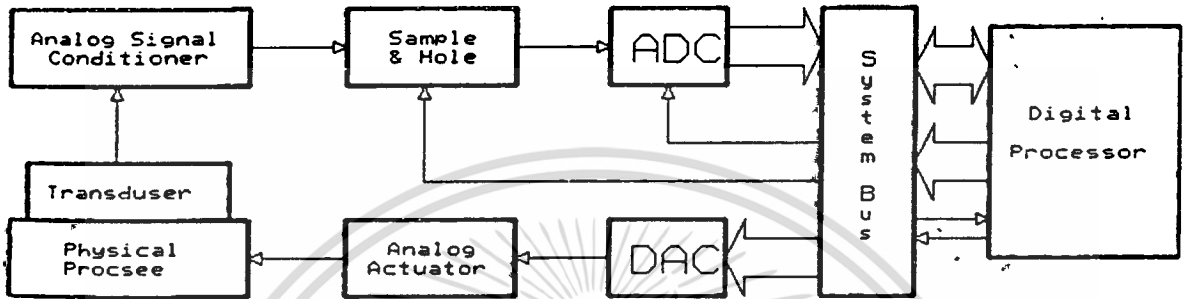
แต่ปัจจุบันนี้เทคโนโลยีทางด้านดิจิทัลได้ก้าวหน้าไปมากทำให้การประมวลผลสัญญาณทางดิจิทัลสามารถทำได้อย่างรวดเร็ว มีประสิทธิภาพและความเชื่อถือได้ ดีกว่าอดีตมากมายนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการเปลี่ยนรูปแบบของสัญญาณ (Conversion) จึงได้มีความจำเป็นขึ้น จากสัญญาณอนาลอกที่มีอยู่แล้วได้เปลี่ยนให้เป็นสัญญาณทางดิจิทัล โดยอุปกรณ์อนาลอกทูดิจิทัลคอนเวอร์เตอร์ และจะถูกประมวลผลโดยตัวประมวลผลสัญญาณดิจิทัล เช่น คอมพิวเตอร์ เป็นต้น จากนั้นผลลัพธ์ที่ได้จะถูกนำมาแสดงผลโดยตรงเลย หรืออาจถูกเปลี่ยนกลับให้อยู่ในรูปของสัญญาณอนาลอกที่ใช้งานได้ การที่จะเปลี่ยนสัญญาณดิจิทัลกลับเป็นสัญญาณอนาลอกนั้นสามารถทำได้โดยใช้อนาลอกทูดิจิทัลคอนเวอร์เตอร์

สำหรับระบบที่มีการประมวลผลข้อมูลทางดิจิทัลแสดงดังรูปที่ 2.4.1



รูปที่ 2.4.1 แสดงระบบที่มีการประมวลผลข้อมูลทางดิจิทัล

จากรูปข้างต้นสามารถอธิบายได้คือ การเปลี่ยนแปลงทางกายภาพในลักษณะใด ๆ ก็ตาม เช่น อุณหภูมิ ความดัน ความเร็ว จะถูกเปลี่ยนให้มาเป็นสัญญาณไฟฟ้าแบบอนาลอก โดยทรานสดิวเซอร์ (Transducer) ที่มีรูปแบบเหมาะสมกับลักษณะทางกายภาพนั้น ๆ จากนั้นสัญญาณทางไฟฟ้าก็จะถูกปรับให้อยู่ในรูปแบบ และขนาดที่เหมาะสมก่อน โดยวงจรต่าง ๆ เช่น วงจรขยาย หรือวงจรกรองสัญญาณเป็นต้น

วงจรแอมพลิฟายเออร์จะส่งขนาดของสัญญาณอนาลอกออกมา แล้วก็จะทำการไฮลด์สัญญาณนั้นไว้ชั่วขณะ เพื่อให้ไม่จำเป็นต้องใช้ ADC ที่มีความเร็วมากนัก เมื่อสัญญาณถูกแปลงเป็นดิจิทัลโดย ADC แล้วข้อมูลดิจิทัลจะถูกส่งต่อไปยังบัสของระบบ จากนั้นตัวโปรเซสเซอร์จะทำการประมวลผลข้อมูลแล้วเปลี่ยนข้อมูลผลลัพธ์กลับมาเพื่อควบคุมกิจกรรมทางกายภาพของระบบ โดยผ่านตัวกระทำทางกล (Analog Actuator)

2.4.1 ทฤษฎีของการสุ่มข้อมูล (Sampling)

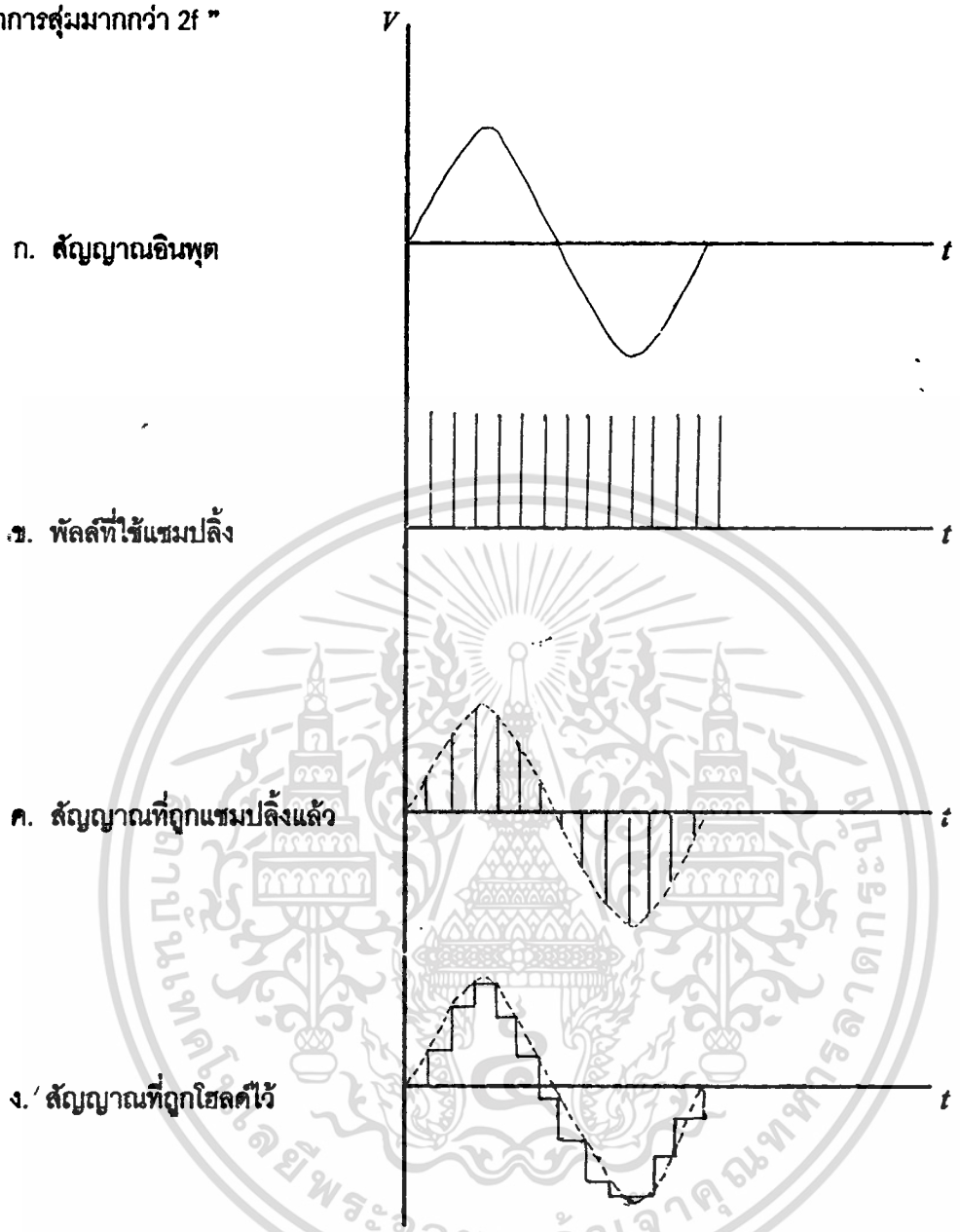
ในการสุ่มข้อมูลนั้นสัญญาณอนาลอกจะถูกสุ่มเป็นระยะ ๆ คงที่ กลุ่มของสัญญาณที่สุ่มจะแทนด้วยความสูง ซึ่งเกิดจากการตัดต่อสัญญาณอนาลอกด้วยระยะเวลาอันสั้น ผลของการสุ่มด้วยความเร็วจะเหมือนกับ การคูณขบวนสัญญาณพัลส์ กับสัญญาณอนาลอก ซึ่งจะได้สัญญาณที่มอดูเลต (Modulate) ระหว่างขบวนพัลส์กับสัญญาณอนาลอก ดังแสดงในรูปที่ 2.4.2 และถ้าหากเรามีการไฮลด์สัญญาณที่สุ่มได้เอาไว้ เรา ก็จะได้สัญญาณดังรูปที่ 2.4.2 ง

อัตราการสุ่มสัญญาณ หรือ ความถี่ของการสุ่มสัญญาณควรมีค่าเท่าใด ที่ข้อมูลที่สุ่มได้นี้จะเป็นตัวแทนที่ดีของสัญญาณต่อเนื่องนั้น คำตอบคือขึ้นอยู่กับความถี่ของสัญญาณอนาลอก และ จากทฤษฎีของการสุ่ม (Nyquist Theorem) กล่าวไว้ว่า “ ถ้าสัญญาณต่อเนื่องที่มีความถี่ฮาร์โมนิก (Harmonic Frequency) ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิน f แล้ว สัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาเช่นเดิมโดยไม่สูญเสียรายละเอียด หรือผิดเพี้ยนไป ถ้าอัตราสุ่มมากกว่า $2f$ "

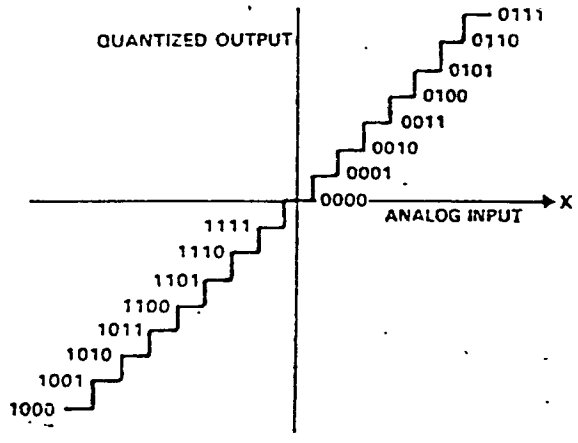


รูปที่ 2.4.2 แสดงการสุ่มสัญญาณ

2.4.2. ทฤษฎีการควันไทซ์ (Quantizing Theorem)

การควันไทซ์ เป็นขบวนการที่แปลงสัญญาณอนาลอก ให้เป็นข้อมูลทางดิจิทัลที่เป็นสัดส่วนกับสัญญาณอนาลอก เช่นอยู่ในรูปของรหัสไบนารี เป็นต้น หากเรานำเอาทั้งหมดที่ได้จากการควันไทซ์และขนาดของสัญญาณอนาลอก มาเขียนเป็นกราฟ ก็จะได้กราฟควันไทซ์ทราสเฟอร์ฟังก์ชัน (Quantize Transfer Function) ดังแสดงในรูปที่ 2.4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.3 แสดงทรานสเฟอร์ฟังก์ชันของการควันไทซ์ 4 บิต

จุดสำคัญของการควันไทซ์นั้นได้แก่ รีโซลูชัน (Resolution) ของตัวควันไทซ์ ซึ่งจะสามารถกำหนดได้จากจำนวนบิตของเอาต์พุตหรือรหัสดิจิทัล หรือจากกราฟรูปที่ 2.4.3 ก็คือส่วนกลับของความกว้างของขั้นบันไดทางแกนนอนนั่นเอง

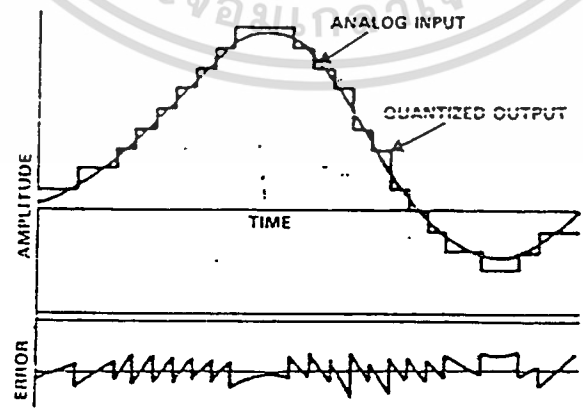
ค่าความกว้างของขั้นบันไดนี้จะสามารถคำนวณได้จาก

$$Q = FRS/2^n$$

โดยที่

- Q : ค่าความกว้างของขั้นบันไดทางแกนนอน
- FSR : ช่วงเต็มสเกลของสัญญาณอนาล็อก (Full Scale Range)
- n : จำนวนบิตของรหัสดิจิทัล

จากสมการข้างต้นจะเห็นว่าเมื่อจำนวนบิตมากขึ้น ขนาดของ Q ก็จะลดลงและถ้าเรานำสัญญาณอนาล็อกใดๆ มาทำการควันไทซ์ จะเห็นได้ว่าเมื่อนำเอาผลที่ได้จากการควันไทซ์มาเปรียบเทียบกับสัญญาณอนาล็อกนั้นแล้ว ก็จะพบว่ามีความผิดพลาดเกิดขึ้น ซึ่งเราจะเรียกความผิดพลาดนี้ว่า ความผิดพลาดควันไทซ์ (Quantizing Error) ดังแสดงในรูปที่ 2.4.4



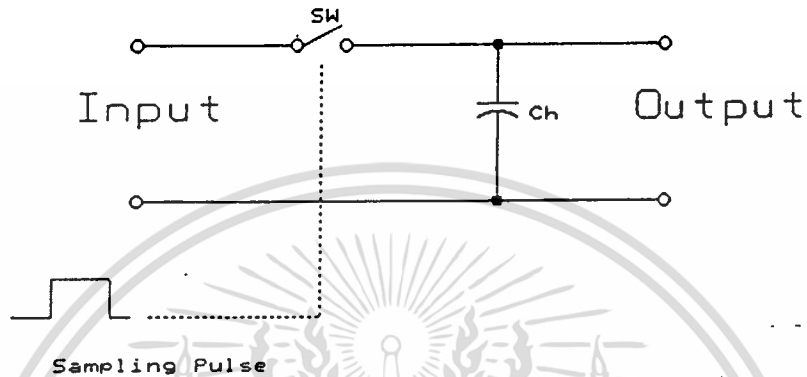
รูปที่ 2.4.4 แสดงความผิดพลาดควันไทซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดนี้เป็นธรรมชาติของการคว้นไตร์ ซึ่งจะทำให้การแก้ไขไม่ได้แต่เราจะลดค่าความผิดพลาดนี้ได้ก็โดยการเพิ่มจำนวนบิตของการคว้นไตร์ ค่าความผิดพลาดนี้เองที่จะเป็นตัวกำหนดความแม่นยำในการใช้งาน DSP

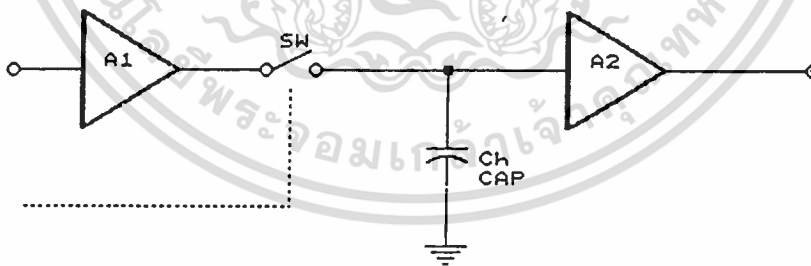
2.4.3 วงจรแซมเปิลแอนด์โฮลด์

วงจรแซมเปิลแอนด์โฮลด์ โดยพื้นฐานแล้วจะเป็นวงจรหรือ อุปกรณ์เก็บแรงดัน (Voltage Memory) ซึ่งใช้อุปกรณ์ที่สำคัญคือ ตัวเก็บประจุ ดังในรูปที่ 2.4.5



รูปที่ 2.4.5 แสดงวงจรพื้นฐานของการแซมเปิลแอนด์โฮลด์

อิเล็กทรอนิกส์สวิตช์จะต่อสัญญาณแรงดันเข้ากับตัวเก็บประจุ ซึ่งสวิตช์จะถูกควบคุมจากแซมปลิงพัลส์ ช่วงเวลาการตัดสวิตช์ และเวลาในการประจุแรงดันถึงค่าที่สุ่มมานั้นจะเรียกว่า อเพอร์เชอร์ไทม์ (Aperture time) ของแซมปลิงแอนด์โฮลด์



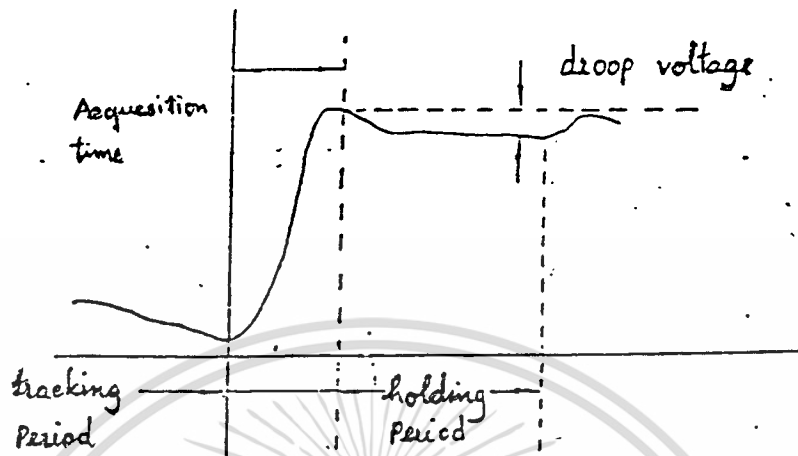
รูปที่ 2.4.6 แสดงวงจรแซมปลิงแอนด์โฮลด์ที่ใกล้เคียงกับวงจรที่ใช้งานจริง

จากรูปที่ 2.4.6 โดยการเพิ่มเต็มบัฟเฟอร์แอมพลิฟายเออร์ (Buffer Amplifier) เข้าทางส่วนอินพุต และเอาท์พุทของวงจรมี แอมพลิฟายเออร์ทางด้านอินพุตจะช่วยเก็บประจุได้เร็วขึ้น ส่วนแอมพลิฟายเออร์ทางด้านเอาท์พุท จะช่วยให้เอาท์พุทมีอิมพีแดนซ์ต่ำสามารถขับเอ็ดซีได้ง่าย ส่วนสำคัญที่ต้องพิจารณาคือ จะ

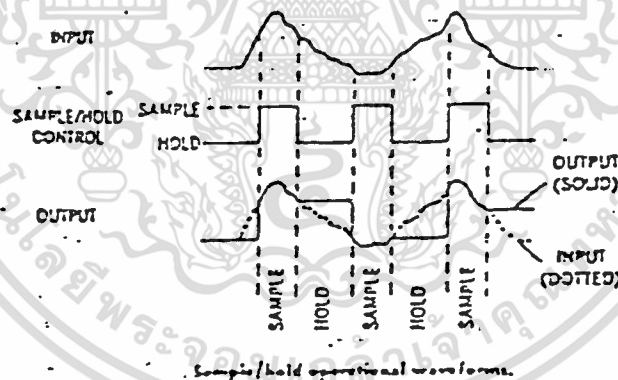
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องใช้แอมพลิฟายเออร์ที่กินกระแสต่ำ เพื่อให้ดึงกระแสจากตัวเก็บประจุในช่วงโวลต์สัญญาณน้อยที่สุด มิฉะนั้นจะเกิดการตก (Drop) แก่แรงดันที่ โวลต์ ดังแสดงในรูปที่ 2.4.7

ส่วนรูปที่ 2.4.8 แสดงรูปคลื่นสัญญาณที่เกิดจากวงจรแซมเปิลแอนด์โฮลด์ในทางอุดมคติ



รูปที่ 2.4.7 แสดงการตกของแรงดันที่โวลต์ได้

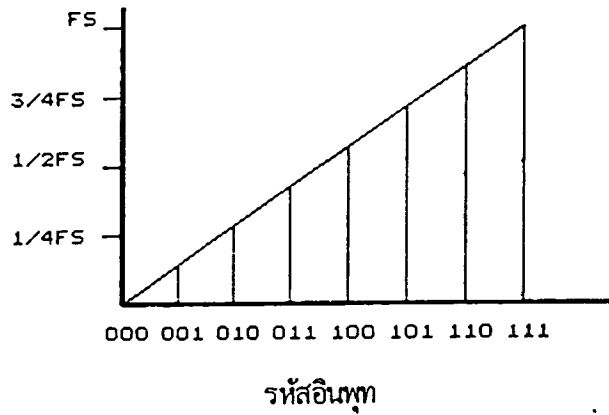


รูปที่ 2.4.8 แสดงเปลี่ยนสัญญาณจากวงจรแซมเปิลแอนด์โฮลด์ในทางอุดมคติ

2.4.4 วงจรเปลี่ยนสัญญาณดิจิตอลให้เป็นอนาลอก (DAC)

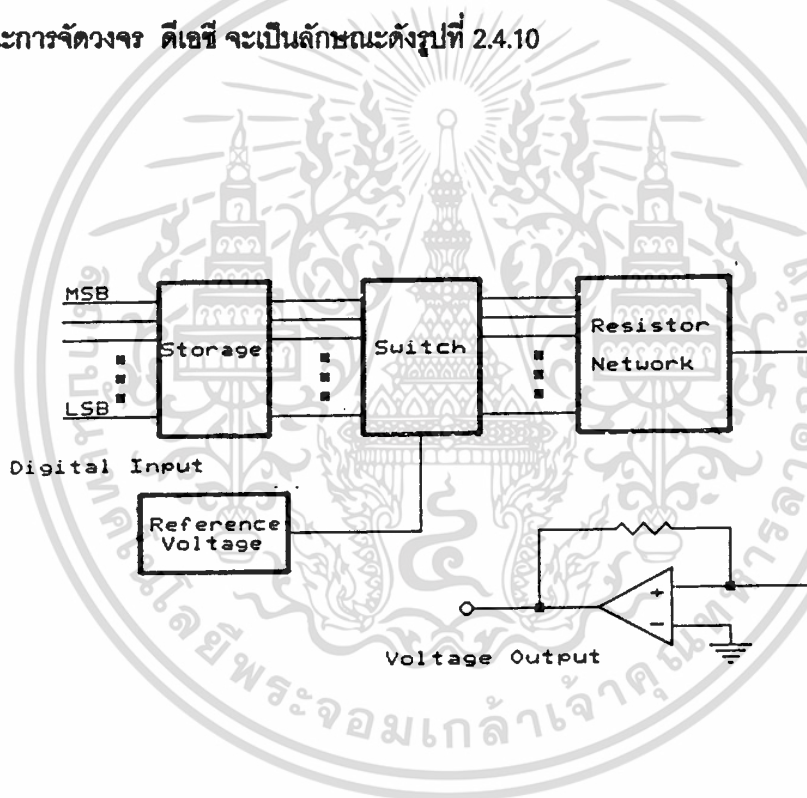
ดีเอซี เป็นอุปกรณ์สำคัญอย่างหนึ่ง ที่ทำให้นำเอาคอมพิวเตอร์ไปใช้ร่วมกับอุปกรณ์หรือวงจรรอนาลอกอื่น ๆ รูปที่ 2.4.9 จะแสดงทรานสเฟอร์ฟังก์ชันของ ดีเอซี ขนาด 3 บิต จะเห็นได้ว่ารหัสดิจิตอลอินพุต 1 เวกต์ จะเปลี่ยนเป็นแรงดันอนาลอก 1 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.9 แสดงทรานสเฟอร์ฟังก์ชันของดีเอซี ขนาด 3 บิต

ลักษณะการจัดวงจร ดีเอซี จะเป็นลักษณะดังรูปที่ 2.4.10

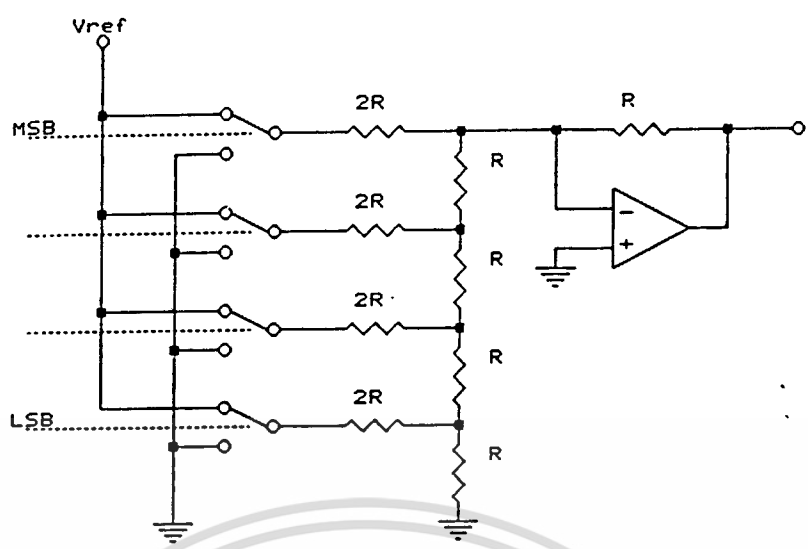


รูปที่ 2.4.10 แสดงบล็อกไดอะแกรมของดีเอซี

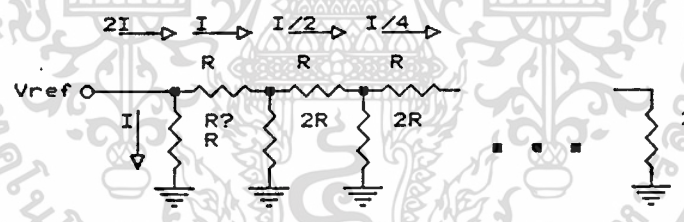
วงจรดีเอซี ที่ใช้กันมีหลายแบบแต่ในที่นี้จะขอกล่าวถึง วงจรดีเอซีที่เป็นแบบตัวต้านทานชั้นบันได (R-2R ladder) ซึ่งเป็นแบบที่นิยมใช้กันมาก ลักษณะของวงจรแสดงดังรูปที่ 2.4.11

ในวงจรนี้สวิตช์จะตัดต่อให้แรงดันอ้างอิง ต่อเข้ากับวงจรชั้นบันได จะเห็นได้ว่าสวิตช์อินพุต จะต่อที่ รีซิสเตอร์ (2R) เมื่อมองเข้าไปจะเห็นคู่ของรีซิสเตอร์ระหว่างจุดต่อ R-2R ที่ติดกันกระแสจะถูกบั่นทอนไปในอัตรา 2/1 ซึ่งสอดคล้องกับรหัสไบนารี ดังรูปที่ 2.4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.11 แสดงวงจรดีเอซี ขนาด 4 บิต แบบ R-2R ladder



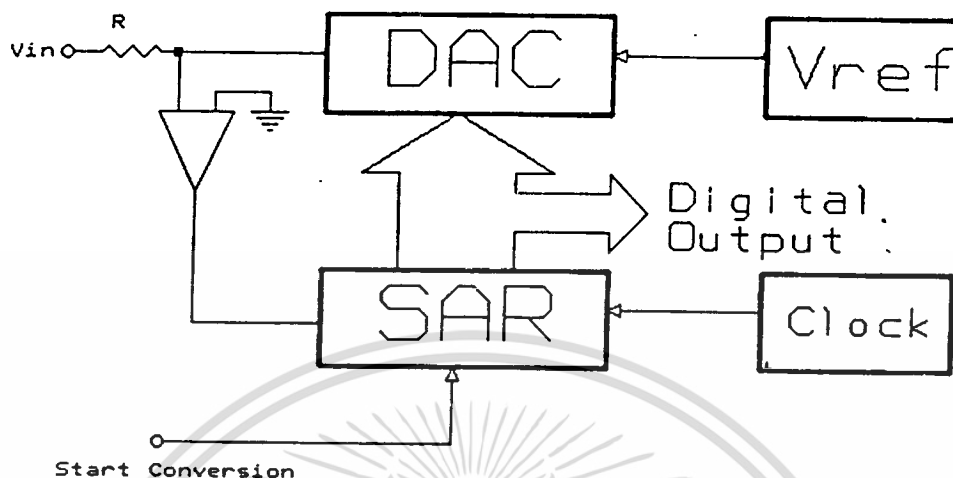
รูปที่ 2.4.12 แสดงการแบ่งแรงดันจากแรงดันอ้างอิง

2.4.5 วงจรเปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัล

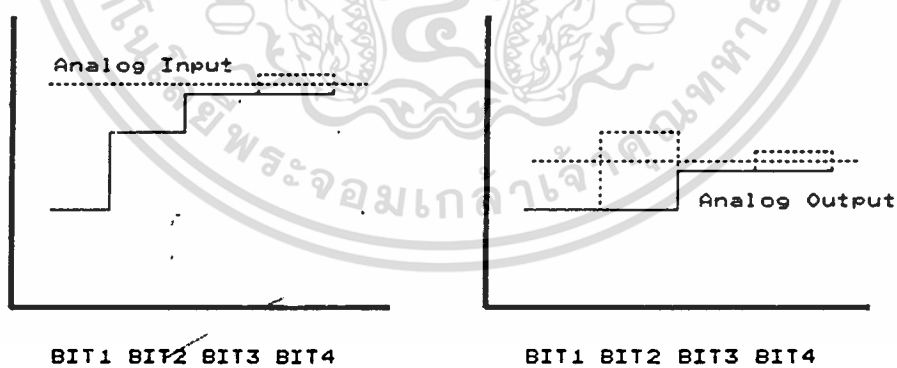
วงจรเอดีซีส่วนใหญ่ จะอยู่ในรูปของไอซี และแบบที่นิยมใช้คือ แบบการประมาณซัคเซสซีฟ (Successive Approximatin) ซึ่งนิยมใช้ในงานที่ต้องการความเร็วสูงและปานกลาง วงจรแบบนี้คล้ายกับเคาท์เตอร์ที่ทำงานในลักษณะป้อนกลับ

รูปที่ 2.4.13 แสดงฟังก์ชันต่าง ๆ ในเอดีซี คอมพาราเตอร์จะคอยเปรียบเทียบเอาท์พุทจากเอดีซี กับสัญญาณอนาลอกอินพุท เอาท์พุทที่ได้จะควบคุมรีจิสเตอร์ของตัวประมาณแบบซัคเซสซีฟ การทำงานของ SAR จะเป็นดังนี้ (ดูรูปที่ 2.4.14)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.13 แสดงบล็อกไดอะแกรม ของตัวแปลงแบบซัคเซสซีฟ



รูปที่ 2.4.14 แสดงไทม์มิงไดอะแกรมของเอดีซีแบบ SAR

เมื่อสัญญาณนาฬิกาเข้ามา 1 ลูก ก็จะทำให้บิตสูงสุดเป็น "1" ส่วนบิตอื่นๆ ยังคง เป็น "0" ดีเอซี จะเปลี่ยนเอาท์พุทของ SAR ให้เป็นสัญญาณอนาลอกเปรียบเทียบกับสัญญาณอนาลอกอินพุท ถ้าผลการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบที่ตัวเปรียบเทียบที่ตัวเปรียบเทียบ ว่าน้อยกว่าอินพุตก็คงบิทนั้นเป็น “1” แต่ถ้ามากกว่า ก็จะทำให้บิทเป็น “0” จากนั้นจะทำการทดสอบบิทถัดไป โดยทำให้บิทนั้นเป็น “1” หากผลรวมของสองบิทมีค่ามากกว่าก็ทำให้บิทนั้นเป็น “0” แต่ถ้าน้อยกว่าก็คงค่า “1” เอาไว้ แล้วทำการทดสอบบิทถัดไปด้วยวิธีดังกล่าวจนครบทุกบิท หรือจนกว่าเอาท์พุทจะต่างจาก ค่าแรงดันอินพุท ไม่เกิน “1” LSB

ข้อจำกัดประการหนึ่งสำหรับการทำงาน คืออนาล็อกอินพุทจะต้องคงที่ ในช่วงเวลาที่ทำการเปลี่ยนสัญญาณ โดยเปลี่ยนแปลงได้ไม่เกิน 1/2 LSB ในช่วงสุดท้ายของการเปลี่ยนสัญญาณดิจิทัล เอาท์พุทจะออกมาขนานกันทุกบิท



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

แนวทางการออกแบบและการสร้าง

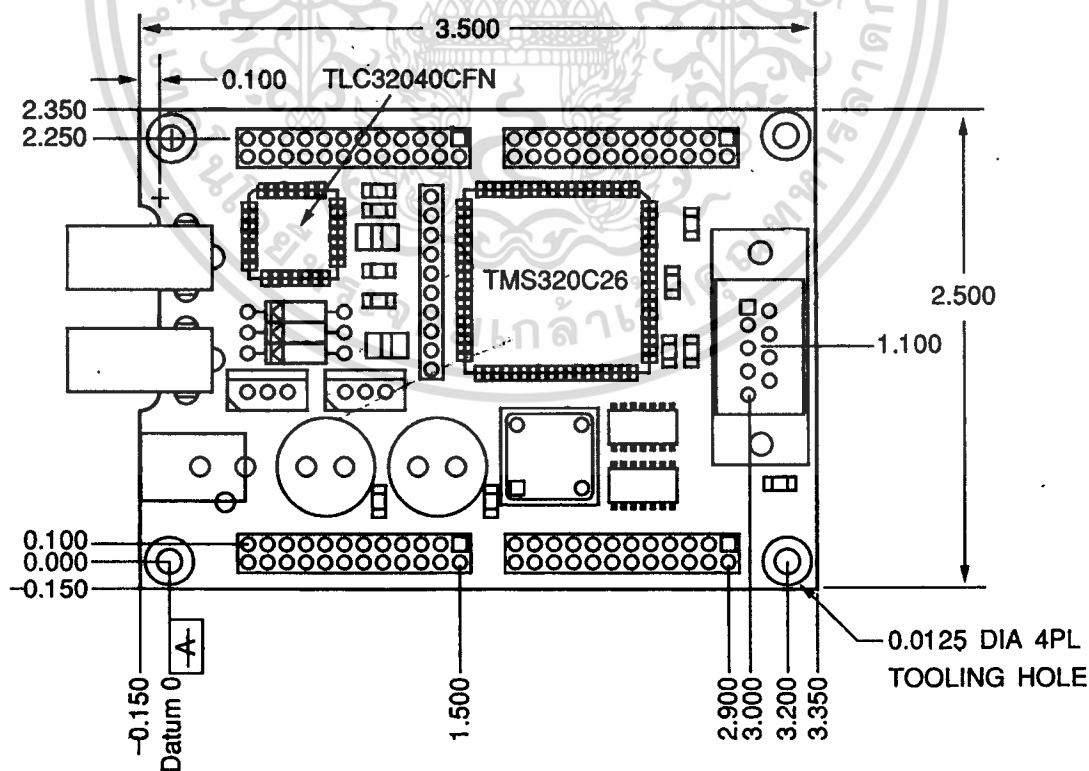
3.1 TMS 320C26 DSP Starter Kit

ในปัจจุบันที่อุปกรณ์อิเล็กทรอนิกส์สมัยใหม่ มักมีอุปกรณ์ Digital Signal Processing (DSP) เป็นส่วนประกอบที่สำคัญ เช่น การ์ด FAX/MODEM ของเครื่องคอมพิวเตอร์ หรือโมเด็มความเร็วสูงรุ่นใหม่ ที่เมื่อเปิดดูภายในก็มีแคชิพ DSP และอุปกรณ์เชื่อมต่อกับสัญญาณโทรศัพท์เท่านั้น นอกจากนี้แล้วยังพบอีกมากในการ์ดเสียง, การ์ดวิดีโอคอมพิวเตอร์/ดีคอปเปอร์สชั่น, ดิจิตอลคอนโทรลเลอร์, ดิจิตอลออสซิลโลสโคป, สเปคตรัมอะนาลิเซอร์, อุปกรณ์ codec (coder-decoder) ของโทรศัพท์ภาพหรือระบบการประชุมผ่านจอภาพ, เครื่องรับสัญญาณดาวเทียม DTH, โทรศัพท์มือถือแบบดิจิตอลทั้ง PCN1800 และ GSM900 ฯลฯ

ในปัจจุบันได้มีผู้นำเข้า TMS 320C26 DSP Starter Kit (เรียกย่อว่า DSK) ของเท็กซัส อินสตรูเมนต์เข้ามาจำหน่าย เพื่อให้ นักศึกษา นักออกแบบ ได้ทดลองศึกษาใช้งาน เป็นการพัฒนางาน DSP ของบ้านเรา

3.1.1 ระบบที่ต้องการ

ระบบเครื่องคอมพิวเตอร์ที่จะใช้กับชุด DSK นี้จะต้องมีคุณสมบัติดังนี้ เป็นเครื่อง IBM PC/AT หรือเทียบเท่า 100% หน่วยความจำ RAM น้อยที่สุด 640 กิโลไบต์ มีฟลอปปีดิสก์ขนาด 5.25 นิ้ว และฮาร์ดดิสก์ตัว DSP บอร์ด ต้องการไฟเลี้ยง 9 VAC (กระแสขั้นต่ำ 250 มิลลิแอมป์) พร้อมแจ๊คต่อขนาด 3.1 มิลลิเมตรและสายเชื่อมโยงกับคอมพิวเตอร์ผ่านทาง RS232C



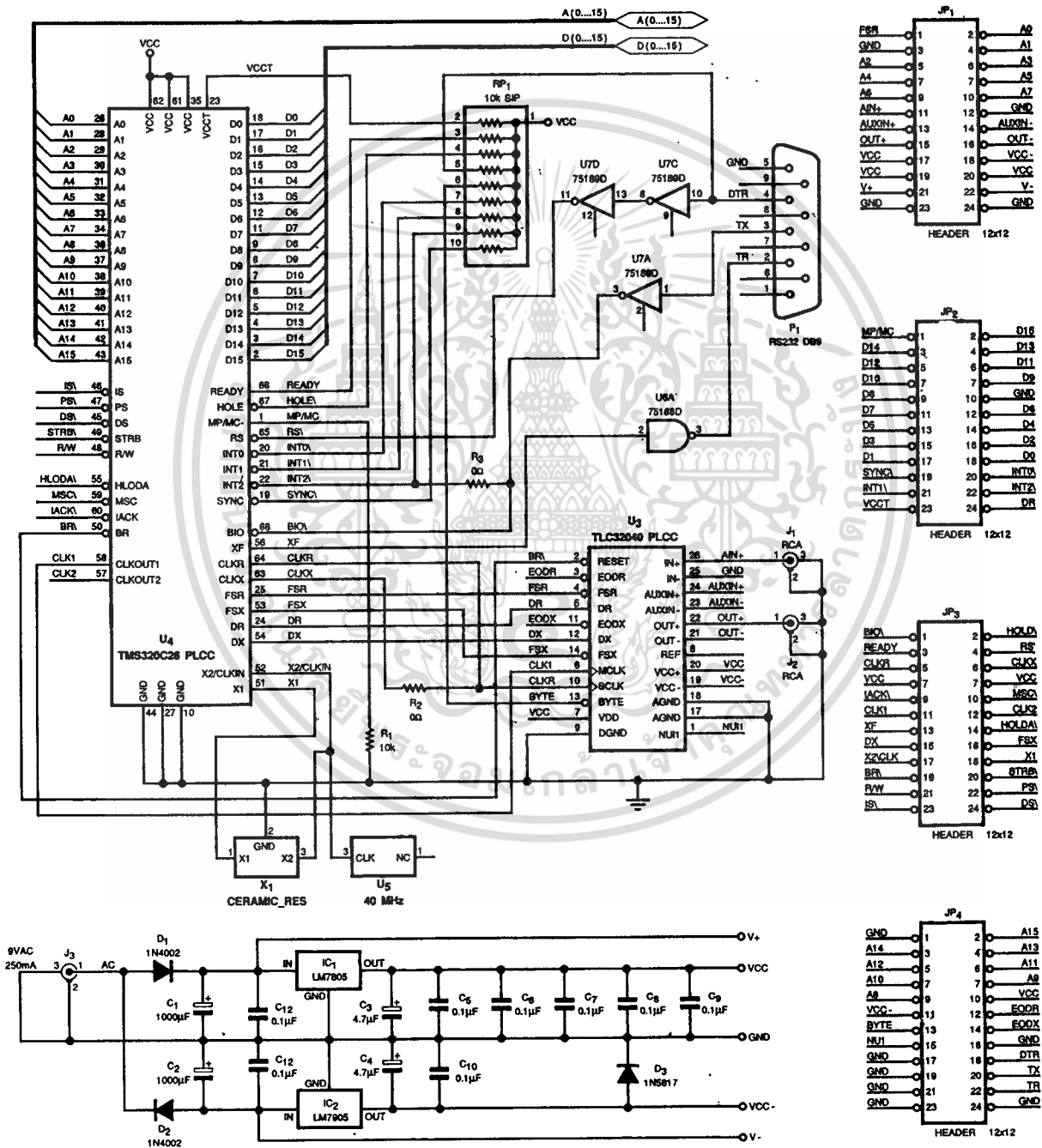
รูปที่ 3.1 หน้าตาและขนาดของบอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 รายละเอียดของบอร์ด DSK

รูปร่างภายนอกของบอร์ด DSK ดังแสดงในรูปที่ 3.1 อุปกรณ์โดยส่วนใหญ่เป็นแบบติดตั้งบนผิวหน้า
 ลงแผ่นวงจรพิมพ์ จะมีเพียงตัวเก็บประจุค่าสูงและไอซีเรกูเลเตอร์ที่เป็นแบบธรรมดา บนบอร์ด DSK จะมีชิพ
 ที่เป็นหัวใจหลักๆ อยู่ 2 อย่างคือ TMS320C26 ซึ่งเป็น DSPโปรเซสเซอร์ และชิพแปลงสัญญาณอะนาลอกเป็น
 ดิจิตอล และแปลงจากดิจิตอลเป็นอะนาลอก เบอร์ TLC32040CFN

ขนาดของบอร์ดอยู่ที่ 2.5 x 3.5 นิ้ว มีคอนเน็คเตอร์ชนิด D ตัวเมีย เพื่อใช้ในการเชื่อมต่อกับ
 คอมพิวเตอร์ผ่านทางบอร์ดอนุกรม RS-232C 1 ชุด มีแจ๊ค RCA สำหรับรับและส่งสัญญาณอะนาลอก



รูปที่ 3.2 วงจรสมมุติของบอร์ด DSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 วงจรสมบูรณ

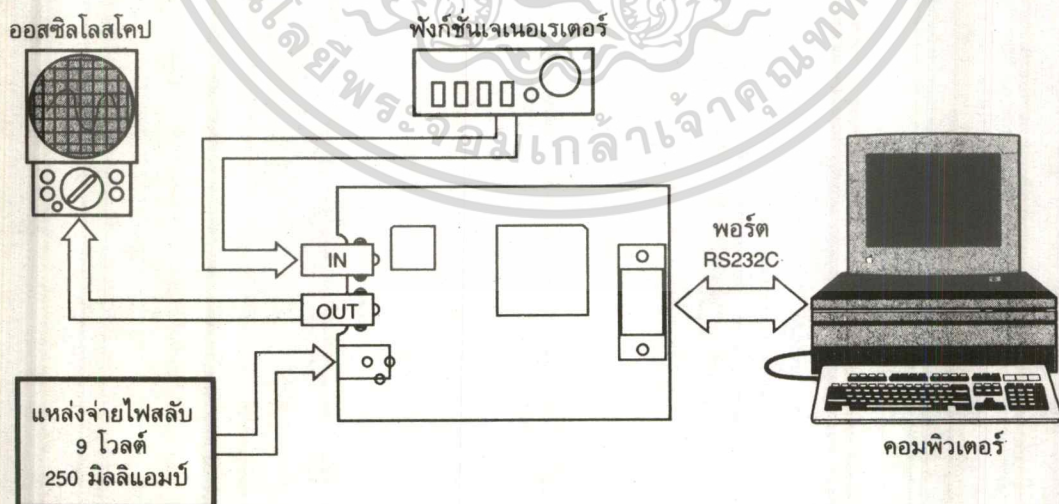
บอร์ด DSK มีวงจรสมบูรณแสดงดังรูปที่ 3.2 ในส่วนบนเป็นวงจรของการเชื่อมต่อระหว่างชิพ DSP กับอุปกรณ์ต่อรวมอื่นๆ ในส่วนล่างเป็นภาคจ่ายไฟของบอร์ด DSK หัวใจสำคัญของบอร์ดอยู่ที่ U_4 ชิพ DSP เบอร์ TMS320C26 เป็นไมโครโปรเซสเซอร์ตัวหนึ่งที่ใช้งานด้านการประมวลผลสัญญาณดิจิทัลโดยเฉพาะ ทำหน้าที่ประมวลผลข้อมูลทั้งหมดบนบอร์ด DSK นี้ โดยต้องทำงานร่วมกับซอฟต์แวร์ ซึ่งจะกล่าวถึงภายหลัง

U_4 ทำงานได้โดยอาศัยสัญญาณนาฬิกา 40 เมกกะเฮิรตซ์ U_5 โดยมี X_1 เซรามิกเรโซเนเตอร์ ช่วยควบคุมความถี่ของสัญญาณนาฬิกาให้มีความเที่ยงตรงอีกชั้นหนึ่ง

U_3 ชิพ ADC/DAC เบอร์ TLC32040 ของเท็กซัสอินสตรูเมนต์ ทำหน้าที่รับสัญญาณอะนาลอกทางอินพุตที่จ่ายเข้ามาทาง J_1 แล้วแปลงเป็นข้อมูลทางดิจิทัลส่งไปประมวลผลต่อที่ U_4 และเมื่อต้องการส่งสัญญาณออก U_4 จะส่งข้อมูลดิจิทัลมายัง U_3 เพื่อทำการแปลงเป็นสัญญาณอะนาลอกจ่ายออกไปทาง J_2 ความละเอียดของการแปลงสัญญาณดิจิทัล จะมีขนาด 14 บิต สามารถกำหนดอัตราแซมปลิงได้สูงสุด 19,200 แซมปลิง ใน 1 วินาที

สำหรับการติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม RS-232C นั้น จะเป็นหน้าที่ของ U_{6A} เบอร์ 75188D และ U_{7A} , U_{7C} , U_{7D} เบอร์ 75189D โดย U_6 ทำหน้าที่ส่งข้อมูลจากบอร์ด DSK ออกไปทาง P_1 คอนเน็คเตอร์ชนิด D 9 ขา ที่ต่อเข้ากับพอร์ตอนุกรม RS-232C ส่วน U_7 ทำหน้าที่เป็นบัฟเฟอร์กำหนดข้อมูลจากคอมพิวเตอร์

ส่วนภาคจ่ายไฟบอร์ด DSK ต้องการไฟสลับ 9 โวลต์ จากภายนอกจ่ายเข้ามาทาง J_3 , D_1 , D_7 , C_1 และ C_2 ทำหน้าที่เป็นวงจรทวิคูณแรงดันแล้วแบ่งแรงดันเป็นแรงดันบวกและแรงดันลบจ่ายให้แก่ไอซีเรกูเลเตอร์ U_1 และ U_2 ได้เป็นแรงดันไฟบวกและแรงดันไฟลบ 5 โวลต์ไปเลี้ยงบอร์ด DSK



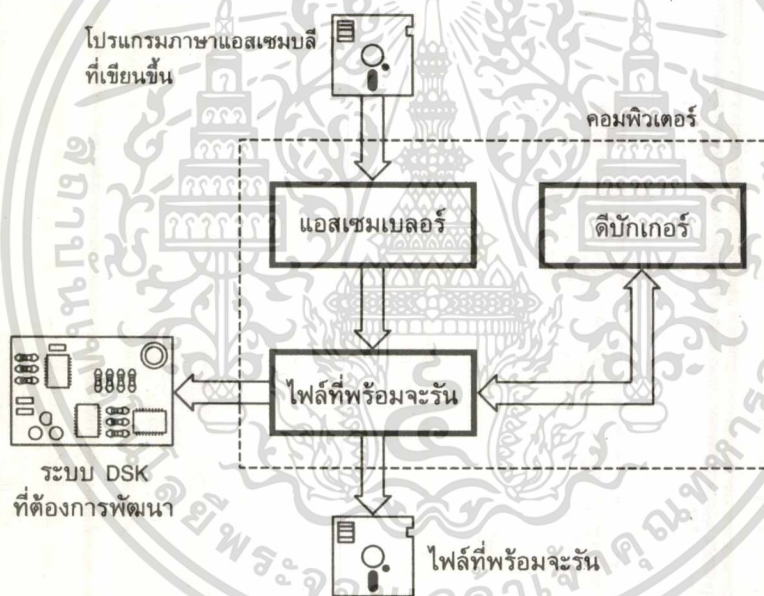
รูปที่ 3.3 การต่อใช้งานบอร์ด DSK ร่วมกับออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 เกี่ยวกับชิพ TMS320C26

TMS320C26 เป็นหัวใจหลักของบอร์ด DSK เป็นผลงานของเท็กซัสอินสตรูเมนต์ เป็นไมโครโปรเซสเซอร์ตัวหนึ่งที่ถูกออกแบบมาให้ใช้งานด้าน DSP โดยเฉพาะ มีคาบเวลาของไซเคิลของคำสั่งเป็น 100 นาโนวินาที ในการทำงานคำสั่งแต่ละตัวจะใช้เวลา 1 ไซเคิล ทำให้สามารถประมวลผลได้ถึง 10 ล้านคำสั่งต่อวินาที จึงส่งผลให้ชิพนี้สามารถทำการประมวลผลแบบเวลาจริง (real time) ได้ทันที

ในตัว TMS320C26 มีหน่วยความจำความเร็วสูงขนาด 1,568 เวิร์ด โดยที่แต่ละเวิร์ดจะมีขนาด 16 บิต และมีรวมภายในขนาด 256 เวิร์ด สามารถอ้างอิงหน่วยความจำภายนอกได้ถึง 128 กิโลเวิร์ด แบ่งเป็นหน่วยความจำข้อมูลและหน่วยความจำโปรแกรมอย่างละ 64 กิโลเวิร์ด ALU (Arithmetic Logic Unit) และแอมพลิฟายเออร์ ขนาด 32 บิตที่สามารถคูณเลขขนาด 32 บิต ได้ภายใน 1 แมกซ์ไซเคิล ทั้งนี้เนื่องจากตัว ALU มีวงจรคูณแบบ 16×16 บิตฮาร์ดแวร์ นอกจากนี้ยังมีตัวตั้งเวลาและส่วนกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ ส่วนของการอินเตอร์เฟสเป็นแบบขนาน 16 บิต แบ่งเป็นอินพุต 16 แชนแนล เอาต์พุต 16 แชนแนล, วงจรติดต่อหน่วยความจำร่วม (global data memory interface)



รูปที่ 3.4 ขั้นตอนการพัฒนาโปรแกรม

ตัว TMS320C26 สามารถต่อใช้งานเพียงตัวเดียวหรือจะต่อใช้งานขนานกันหลายตัว ที่เรียกว่าเป็นระบบ multi-processor system ก็ได้

ในการใช้งานกับหน่วยความจำภายนอกไม่ว่าจะเป็นอีพ롬หรือรอมตัวชิพจะทำงานได้เข้ามา จึงได้เน้นให้ใช้หน่วยความจำแบบภายในชิพเป็นที่เก็บโปรแกรมแทนในการใช้งานให้ดาวน์โหลดโปรแกรมจากหน่วยความจำภายนอกมาเก็บในชิพแล้วจึงรันโปรแกรมทำงาน โดย TMS320C26 ได้เพิ่มคำสั่ง "block transfer" ที่

ใช้ในการโหลดข้อมูลจากหน่วยความจำภายนอกมาที่ละบล็อค แล้วนำมาเก็บไว้ในแรมของชิพเอง ด้วยวิธีการนี้จะช่วยให้การทำงานเร็วมากยิ่งขึ้น

การอ้างแอดเดรสของ TMS320C26 มีด้วยกัน 3 โหมด คือ โดยตรง (direct), ทางอ้อม (indirect) และทันทีทันใด (immediate) นอกจากนี้ยังมีจีสเตอร์เสริมอีก 8 ตัว

3.1.5 การต่อใช้งานบอร์ด DSK

การต่อใช้งานบอร์ด DSK ทำได้ไม่ยาก อุปกรณ์ที่ต้องใช้คือ สายสัญญาณที่ใช้ต่อระหว่างบอร์ดกับคอมพิวเตอร์ อาจจะใช้สายโมเด็มมาตรฐานก็ได้ โดยมีเงื่อนไขว่า ปลายด้านที่ต่อกับบอร์ด DSK ต้องเป็นคอนเน็กเตอร์ชนิด D 9 ขาตัวผู้ การต่อใช้งานแสดงได้ดังรูปที่ 3.3 สายสัญญาณที่ใช้มีอยู่ด้วยกัน 4 เส้น คือ Tx, Rx, DTR และ GND ผู้ใช้งานอาจต่อเองได้

3.1.6 ขั้นตอนการพัฒนาโปรแกรม

ขั้นตอนต่างๆ แสดงไว้ในรูปที่ 3.4 จุดประสงค์ของการพัฒนา คือ ต้องการสร้างซอฟต์แวร์ที่สามารถใช้งานร่วมกับบอร์ด DSK เป้าหมายได้โดยตรง ในชุด DSK จะมาพร้อมกับโปรแกรมแอสเซมเบลอร์และดีบั๊กเกอร์ที่ใช้ในการเขียน แก้ไขทดสอบและปรับปรุงซอฟต์แวร์ให้ได้ผลตามที่ต้องการ

ตัวแอสเซมเบลอร์

แอสเซมเบลอร์มีหน้าที่ในการแปลไฟล์ข้อมูลภาษาแอสเซมบลีให้เป็นไฟล์ภาษาเครื่อง หรือที่มีชื่ออีกอย่างว่า object code ที่ทำให้ชิพสามารถนำข้อมูลรหัสไปปฏิบัติงานได้โดยตรง ตัวแอสเซมเบลอร์ของ DSK เป็นแบบที่สามารถกำหนด "directive" ก่อนที่จะทำการแอสเซมบลีได้ ทำให้สามารถกำหนดตำแหน่งแอดเดรสของโปรแกรมเป็นแบบสัมพัทธ์ (คือเป็นแอดเดรสค่าลอยๆ ที่สามารถกำหนดตำแหน่งที่แน่นอนได้ในภายหลัง) ทำให้ไม่จำเป็นต้องมีขั้นตอนของการ linker

ดีบั๊กเกอร์

ดีบั๊กเกอร์มีหน้าที่ช่วยในการหาตำแหน่งและแก้ไขข้อผิดพลาด (error หรือ bug) ต่างๆ ที่อาจเกิดขึ้นในตัวโปรแกรม การทำงานของโปรแกรมจะแบ่งการทำงานในส่วนต่างๆ แยกเป็นคนละหน้าต่างที่ง่ายต่อการใช้งาน ตัวดีบั๊กเกอร์จะมีหน้าที่พื้นฐานที่สำคัญ ได้แก่ single step execution, breakpoint setting, และ runtime execution halt เป็นต้น

3.2 ขั้นตอนและแผนผังการทำ FFT (Fast Fourier Transform Execution Flow Diagram)

การใช้ FFT ในตัวอย่างที่ประยุกต์มานี้ คือการใช้วิธี Complex radix-2 decimation ในรูปของความถี่ โดยใช้ FFT ที่เลือกใช้กรณีนี้ เพราะว่าต้องทำการลดขนาดของบล็อคแต่ละบล็อกลงเมื่อทำได้ตามเป้าหมายที่วางเอาไว้ แต่ละสเตจของ FFT จะถูกลดลงครึ่งหนึ่ง การลดลงทีละครึ่งหนึ่งนี้จะเริ่มจากทางซ้ายและจะเลื่อนไปทางขวา ซึ่งจะเป็นการง่ายเมื่อใช้ TMS320C26 ใน Addressing Modes รายละเอียดเพิ่มเติมจะอยู่ในคู่มือ User's Guide และในตัวอย่าง Bit Reverse Addressing trick ซึ่งจะอยู่ในภาคผนวก

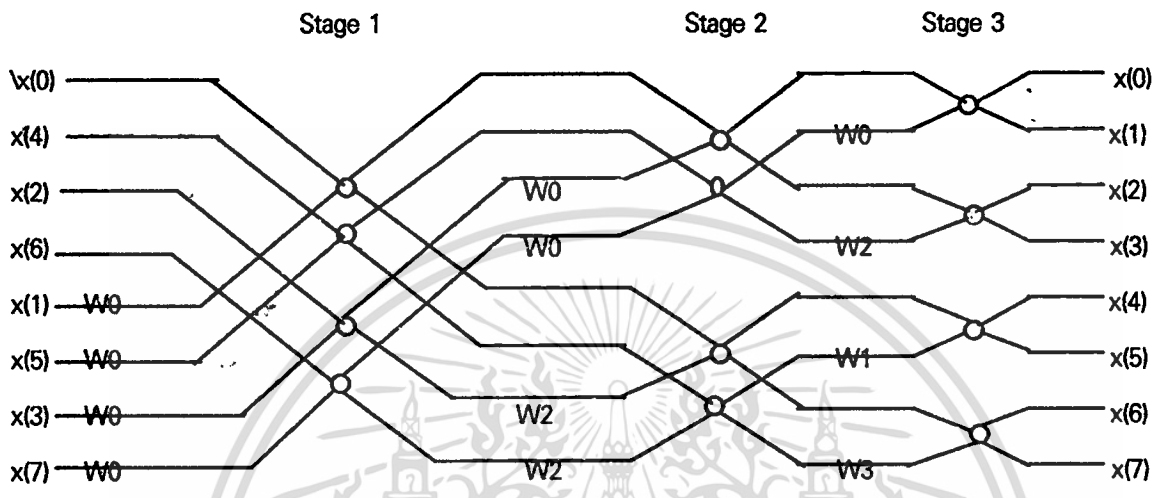
เป็นการง่ายและรวดเร็วในการตรวจสอบ ขั้นตอนการทำงาน (Flow Diagram) ของ FFT โดยการนำเอาขนาดของ DFT/2 ก็จะเป็นจำนวนค่า ARO และทำการจนกว่า ARO จะมีค่าเป็น 1 จำนวนครั้งที่ใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาร ก็คือจำนวนสเตจที่ใช้ทำ FFT และ ถ้า AR0 มีค่าเท่ากับ 0 ก็จะเป็นการบอกให้ทราบว่าจะหยุดลำดับการทำงานแล้ว

จุดสำคัญ ถ้าคุณต้องการที่จะเปลี่ยนขนาดของ FFT จะเริ่มต้นโดยการเปลี่ยนค่าขนาดของ FFT ถ้าขนาดใหม่นี้ น้อยกว่าค่าสูงสุดที่จะเป็นไปได้ โดยขนาดของ Twiddle array, ซึ่งจะเป็นเหมือนตารางที่สามารถนำกลับมาใช้ได้อีก และถ้าทำเป็นภาษาขั้นสูงไปแล้ว ในการโปรแกรมในขณะนี้ ซึ่งจะยากที่จะนำกลับมาใช้และแก้ไขอีกในแต่ละครั้งที่ FFT ต้องการจะใช้



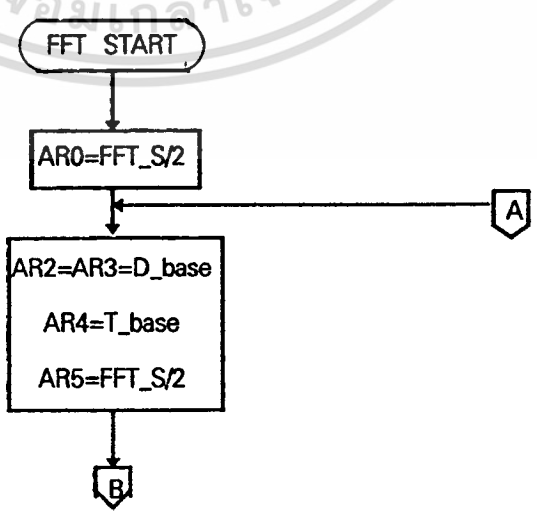
Legend for twiddle factor : $W_9 = W_8^0 W_1 = W_8^1 W_2 = W_8^2 W_3 = W_8^3$

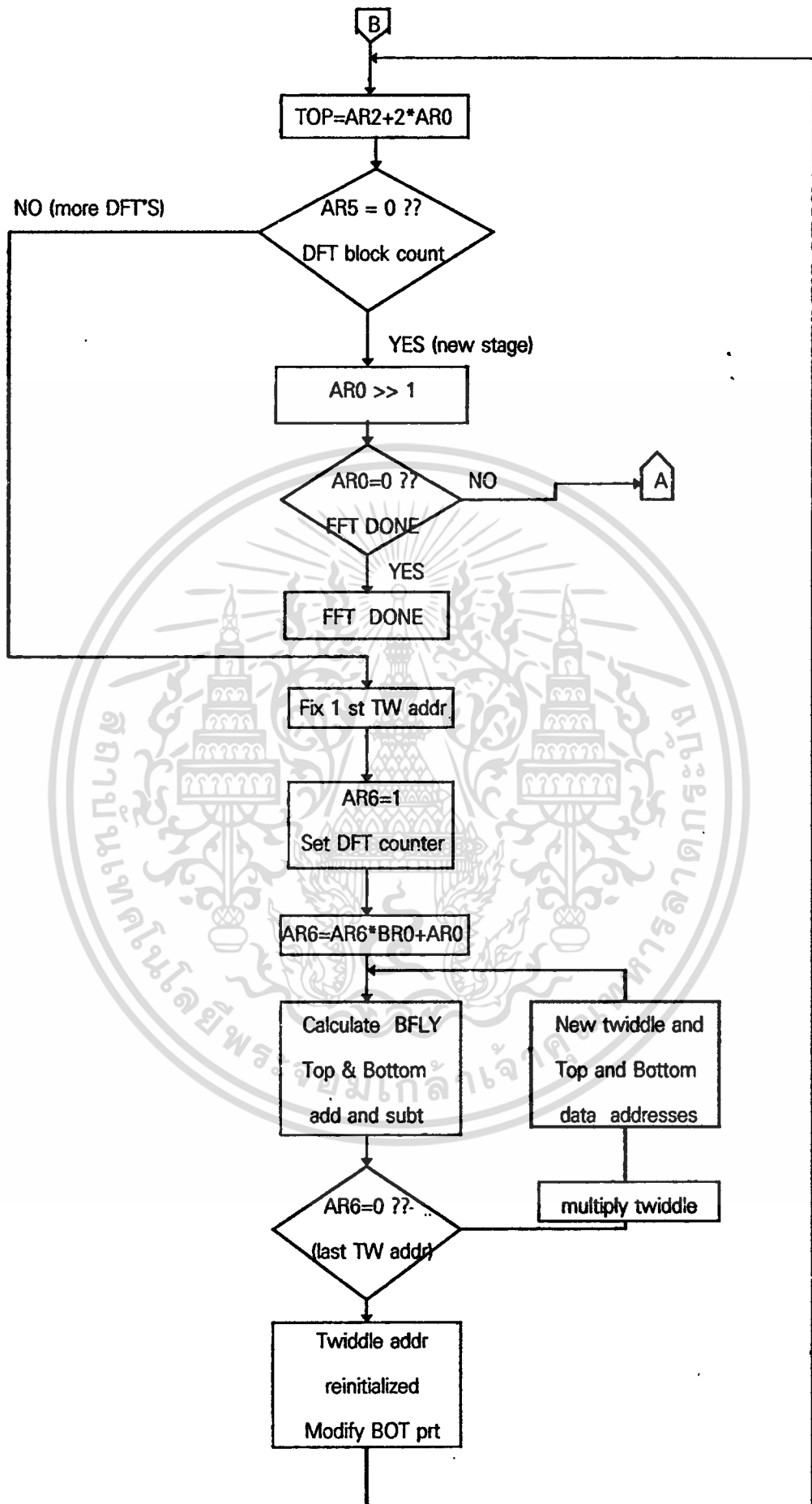
รูปที่ 3.5 An In-Place DIT FFT With In-Order Inputs but Bit- Reversed Outputs

ขั้นตอนที่ 1 : โหลดขนาดของ FFT ไป Run ใน AR0 (ค่ามากที่สุด ใน DSK คือ 256)
(ขั้นตอนนี้เป็นขั้นตอนที่ผู้ใช้ควรจะทำ)

ขั้นตอนที่ 2 : โหลด AR2, AR4 และ AR5 ไปไว้ที่ Base Address ของข้อมูลอินพุตและเอาต์พุต ขบวนการข้อมูล Twiddle Array และขนาดของ FFT ที่หารสองแล้วตามลำดับ

Flow Diagram of FFT





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองที่ 1

ทำการทดลองการแสดงผลสเปคตรัมของเสียงโดยใช้คอมพิวเตอร์ โดยมีขั้นตอนการทำการทดลองดังต่อไปนี้ (แล้วนำผลที่ได้มาเปรียบเทียบกับทฤษฎี เพื่อสรุปและวิจารณ์การทดลอง, ผลการทดลอง ต่อไป)

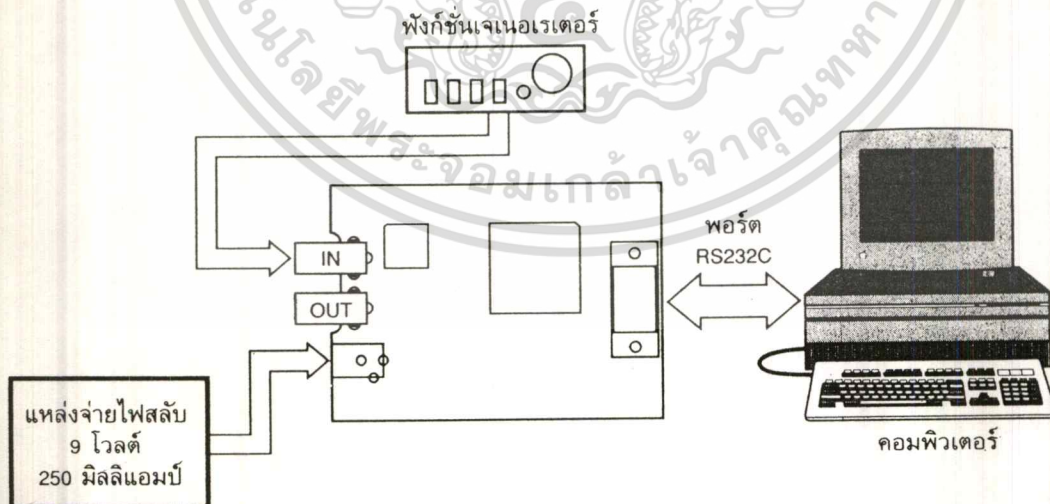
4.1.1 อุปกรณ์การทดลอง

ทำการจัดเตรียมอุปกรณ์ในการทำการทดลอง มีรายการดังต่อไปนี้

1. ชุดบอร์ด TMS 320C26 DSP Starter Kit
2. สายต่อแบบ RS 232C (ใช้ต่อกับพอร์ตอนุกรมของคอมพิวเตอร์)
3. เครื่องคอมพิวเตอร์ (ที่มีหน่วยความจำ RAM อย่างน้อยที่สุด 640 กิโลไบต์)
4. แผ่นโปรแกรมที่ต้องการ RUN (ทางผู้จัดทำได้ทำการพัฒนาขึ้นมาใช้โดยเฉพาะ)
5. Function Generater (ใช้กำเนิดสัญญาณรูปแบบต่างๆในการทดลอง)

4.1.2 ลำดับขั้นตอนการทดลอง

- 4.1.2.1 ต่อดวงจรและอุปกรณ์ต่างๆ เข้าด้วยกันดังรูปที่ 4.1
- 4.1.2.2 ป้อนสัญญาณรูป Sine Wave ความถี่ 500 Hz
- 4.1.2.3 ทำการ Load File ลงบอร์ด DSP โดยใช้ข้อมูลในไฟล์ JACKY.ASM
- 4.1.2.4 ทำการ Run โปรแกรมที่ได้โหลดลงบอร์ด DSP ไว้ในข้อ 3.
- 4.1.2.5 สังเกตผลที่ได้แสดงออกทางหน้าจอคอมพิวเตอร์ และบันทึกผลการทดลอง
- 4.1.2.6 ทำการเปลี่ยนความถี่เป็น 1 kHz, 10 kHz, และ 20 kHz สังเกตและบันทึกผลการทดลอง



รูปที่ 4.1 แสดงการต่ออุปกรณ์ที่ใช้ในการทดลองทั้ง 5 การทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองที่ 2

ทำการทดลองเหมือนกับการทดลองที่ 1 แต่เปลี่ยนสัญญาณที่ป้อนให้บอร์ด DSP เป็นสัญญาณ Square Wave ความถี่ 1 kHz, 10 kHz สังเกตและบันทึกผลการทดลอง

4.3 การทดลองที่ 3

ทำการทดลองเหมือนกับการทดลองที่ 1 แต่เปลี่ยนสัญญาณที่ป้อนให้บอร์ด DSP เป็นสัญญาณ Sawtooth Wave ความถี่ 1 kHz, 10 kHz สังเกตและบันทึกผลการทดลอง

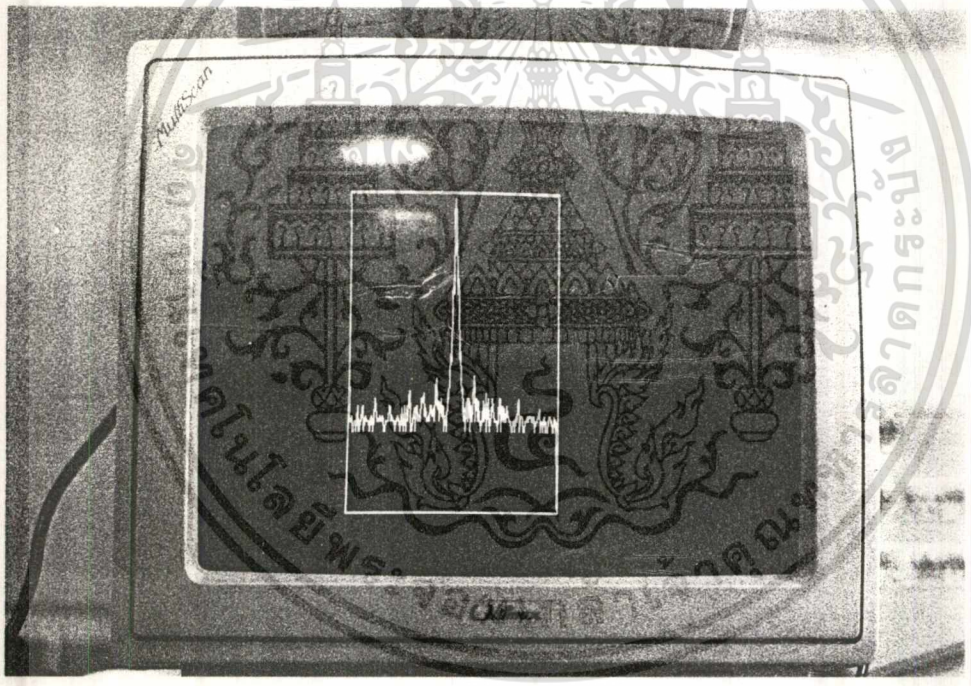
4.4 การทดลองที่ 4

ทำการทดลองเหมือนกับการทดลองที่ 1 แต่เปลี่ยนสัญญาณที่ป้อนให้บอร์ด DSP เป็นสัญญาณ Continuous Signal (Input Music) สังเกตและบันทึกผลการทดลอง

4.5 การทดลองที่ 5

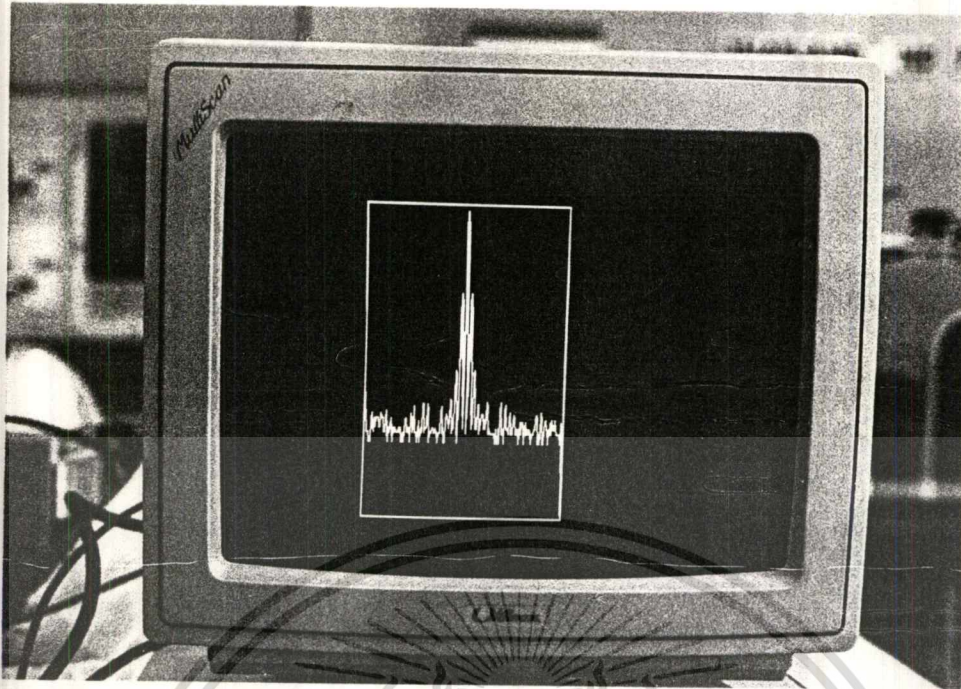
ทำการทดลองเหมือนกับการทดลองที่ 1 แต่เปลี่ยนสัญญาณที่ป้อนให้บอร์ด DSP เป็นสัญญาณ Single Pulse Signal (Speech) สังเกตและบันทึกผลการทดลอง

4.6 ผลการทดลองของทั้ง 5 การทดลอง

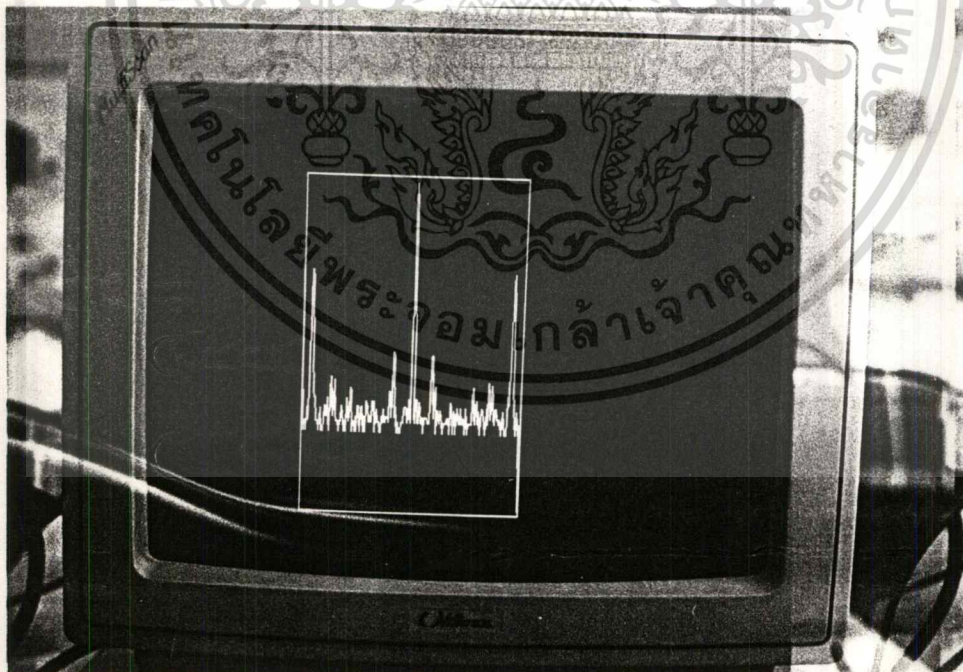


รูปที่ 4.2 รูปแสดงผลสเปกตรัมของสัญญาณ Sine Wave ที่ความถี่ 500Hz ในการทดลองที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

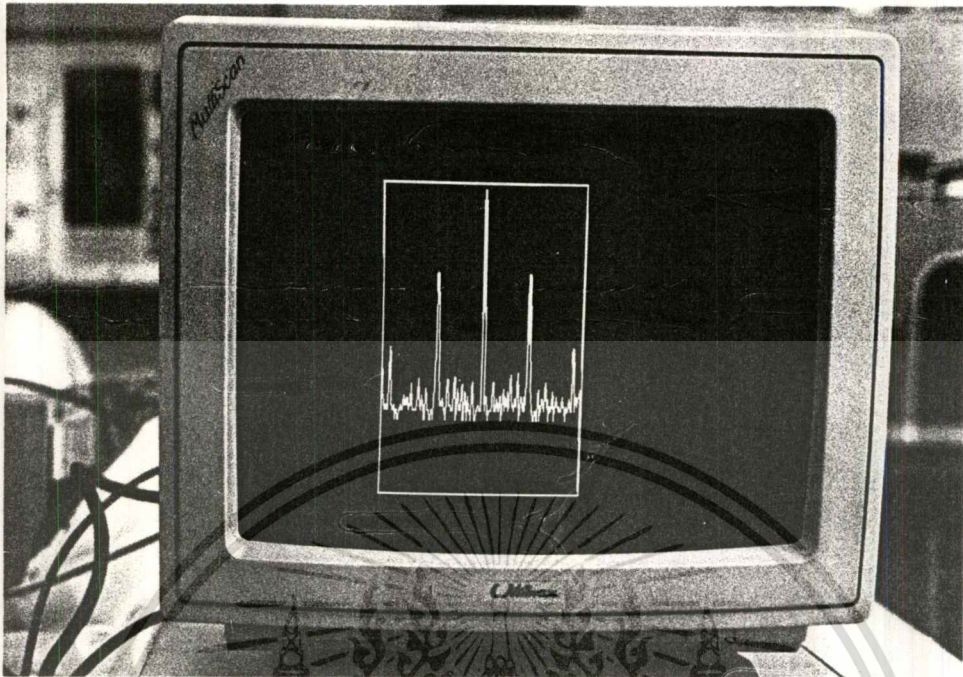


รูปที่ 4.3 รูปแสดงผลเปคตรัมของสัญญาณ Sine Wave ที่ความถี่ 1kHz ในการทดลองที่ 1

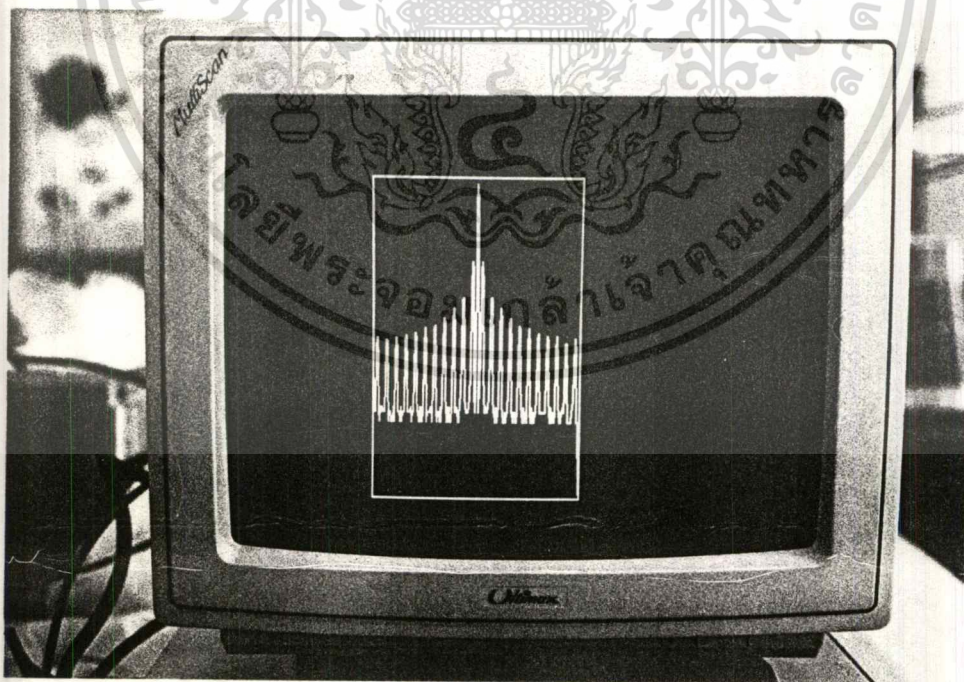


รูปที่ 4.4 รูปแสดงผลเปคตรัมของสัญญาณ Sine Wave ที่ความถี่ 10kHz ในการทดลองที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

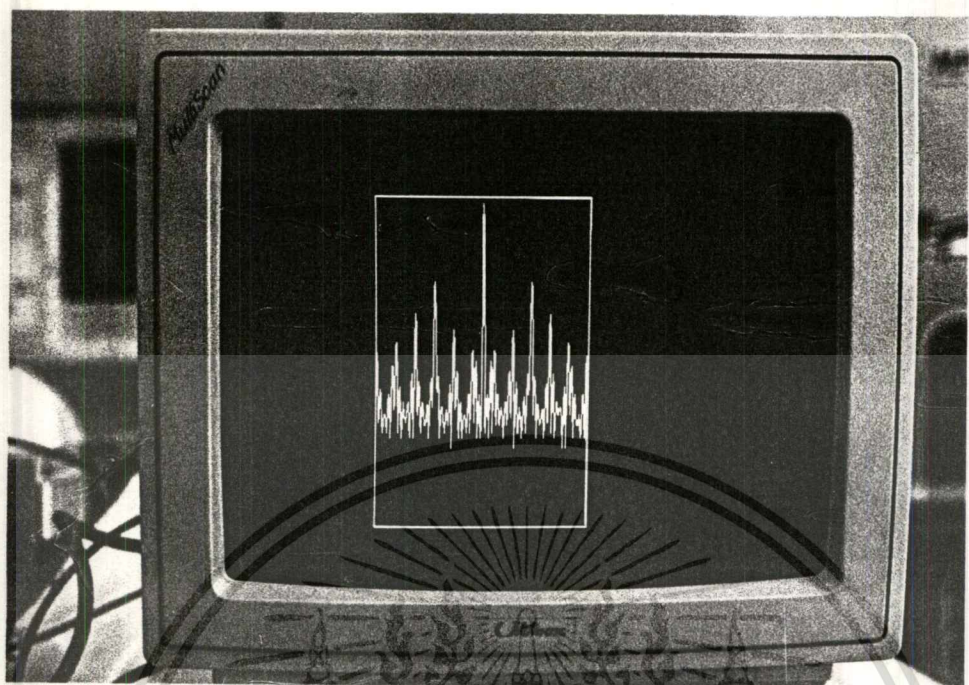


รูปที่ 4.5 รูปแสดงผลเบคตรัมของสัญญาณ Sine Wave ที่ความถี่ 20kHz ในการทดลองที่ 1

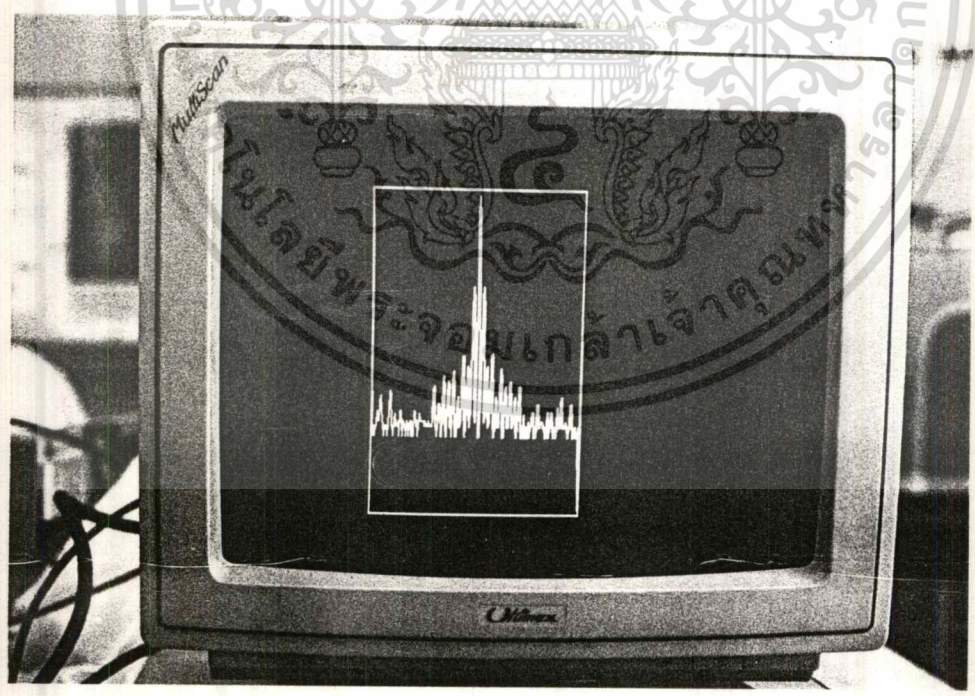


รูปที่ 4.6 รูปแสดงผลเบคตรัมของสัญญาณ Square Wave ที่ความถี่ 1kHz ในการทดลองที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

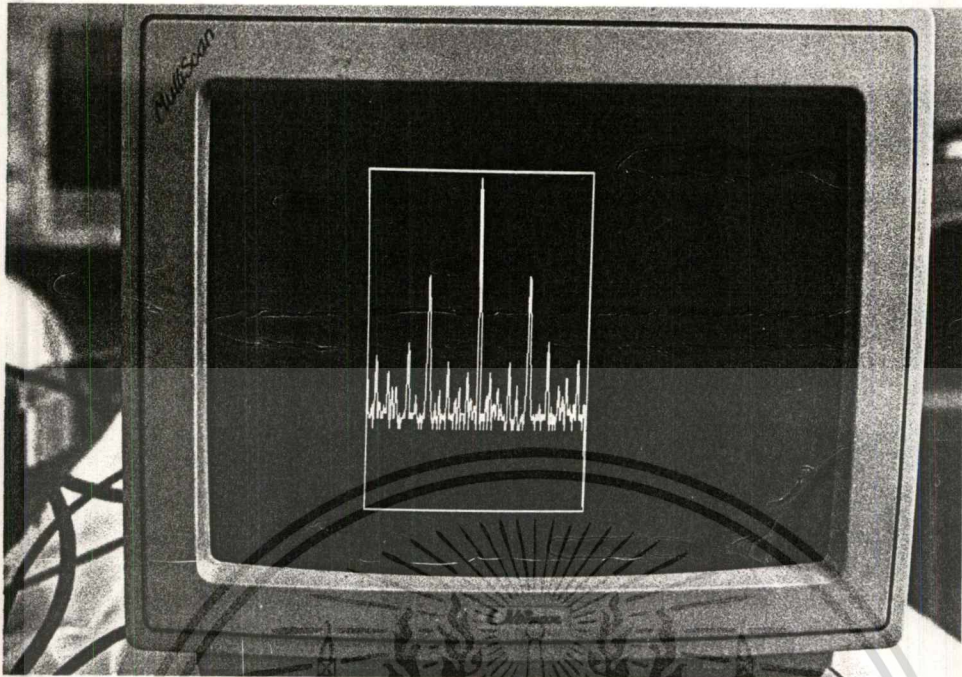


รูปที่ 4.7 รูปแสดงผลเปคตรัมของสัญญาณ Square Wave ที่ความถี่ 10kHz ในการทดลองที่ 2

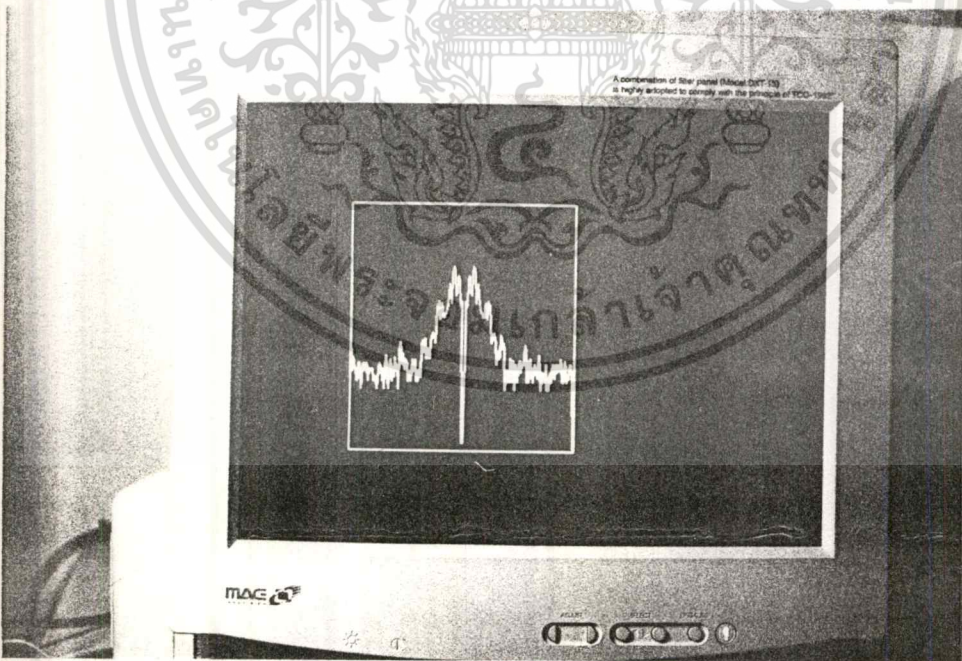


รูปที่ 4.8 รูปแสดงผลเปคตรัมของสัญญาณ Sawtooth Wave ที่ความถี่ 1kHz ในการทดลองที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

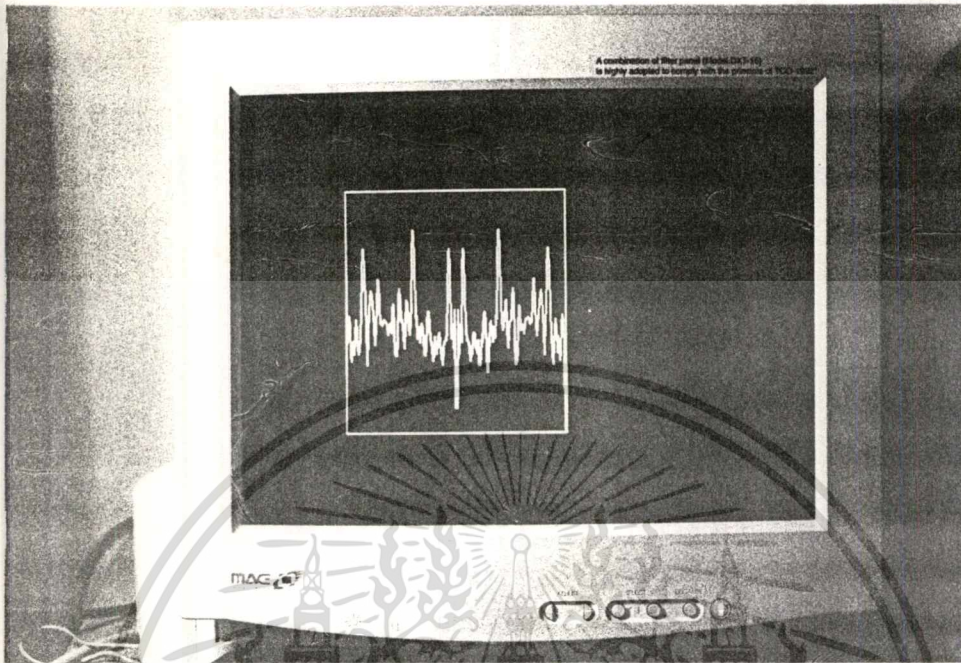


รูปที่ 4.9 รูปแสดงผลเปคตรัมของสัญญาณ Sawtooth Wave ที่ความถี่ 10kHz ในการทดลองที่ 3



รูปที่ 4.10 รูปแสดงผลเปคตรัมของสัญญาณ Continuous Signal (Input Music) ในการทดลองที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 รูปแสดงผลแปดครึ่งของสัญญาณ Single Pulse Signal (Speech) ในการทดลองที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และบทสรุป

จากรายละเอียดและผลของการทดลองที่ได้กล่าวมาแล้วข้างต้น เป็นการนำเอาหลักการและทฤษฎีการประมวลทางสัญญาณดิจิทัล โดยใช้บอร์ด DSP มาใช้ประมวลผลของสัญญาณเสียงเพื่อแสดงสเปกตรัมของสัญญาณเสียงออกทางหน้าจอคอมพิวเตอร์ โดยการศึกษาทดลองนี้จะใช้หลักการของ FFT มาทำการวิเคราะห์ค่าแอมพลิจูด ซึ่งจะทำให้เกิดความคล่องตัวและง่ายต่อการสร้างโปรแกรมขึ้นมาใช้งาน และผลที่ได้จากการทดลองก็เป็นที่น่าพอใจอยู่ในระดับๆ หนึ่ง เพราะเมื่อนำมาเปรียบเทียบกับทฤษฎีและเครื่องมือชนิดอื่นก็มีค่าใกล้เคียงกัน

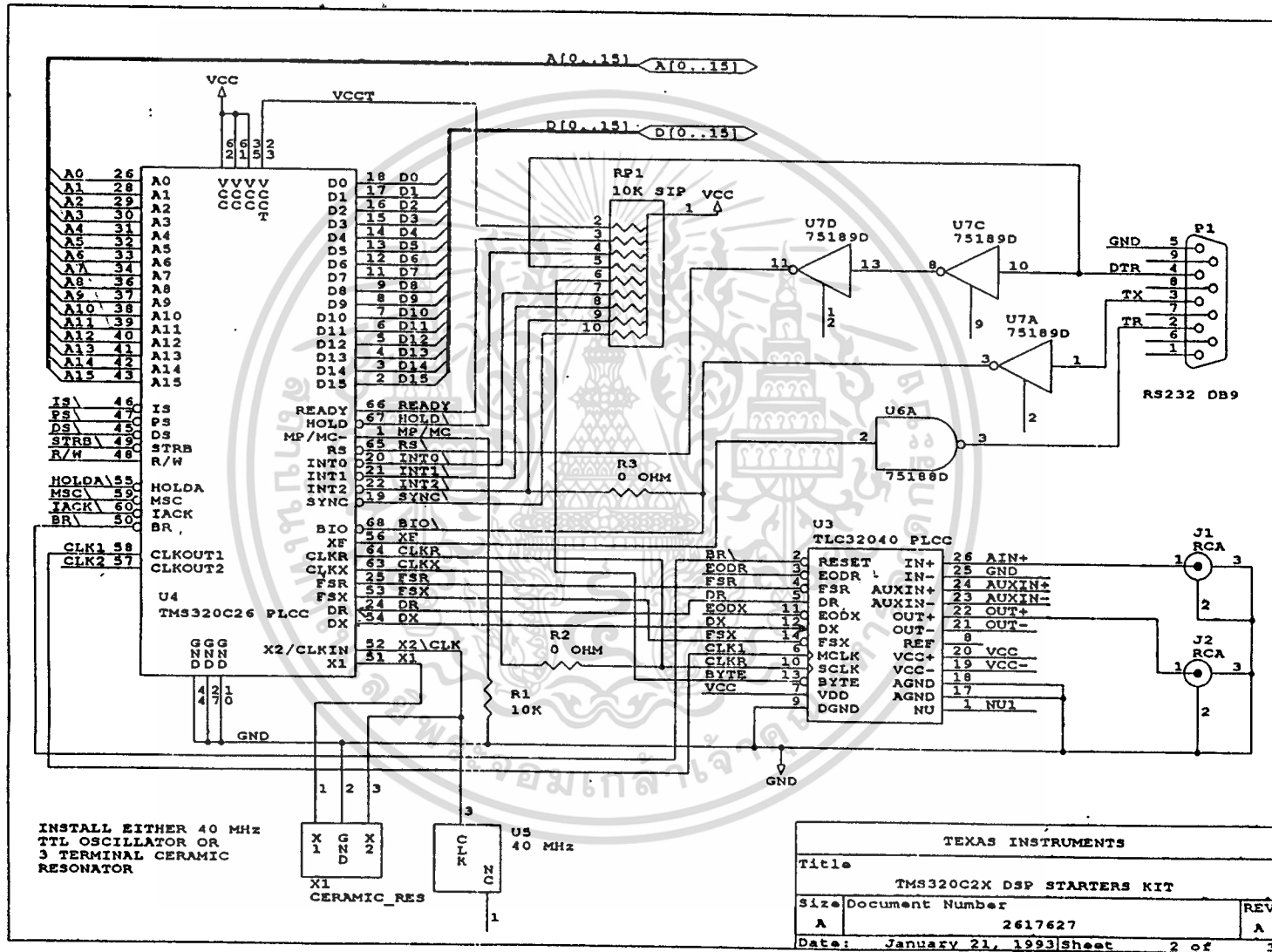
อย่างไรก็ตามโครงงานนี้ ยังสามารถแก้ไขการแสดงผลทางหน้าจอคอมพิวเตอร์ให้มีประสิทธิภาพให้ดีขึ้น และสวยงามขึ้น หรือจะทำการขยายย่านความถี่ของการวิเคราะห์สเปกตรัมให้กว้างขึ้น และให้มีการทำงานได้รวดเร็วยิ่งขึ้น โดยการพัฒนาทางด้านฮาร์ดแวร์และซอฟต์แวร์ขึ้นไปอีกขั้นหนึ่ง ซึ่งทางด้านฮาร์ดแวร์อาจจะใช้ไมโครโปรเซสเซอร์เบอร์อื่นๆ ที่มีความเร็วในการประมวลผลสูงขึ้น การติดต่อกับคอมพิวเตอร์อาจจะใช้การติดต่อและส่งข้อมูลทางพอร์ตนาน ซึ่งถ้าฮาร์ดแวร์เปลี่ยนแปลงไป ทางด้านซอฟต์แวร์ก็จะต้องพัฒนาตามไปด้วย เพื่อให้ได้ผลตามที่เราต้องการ

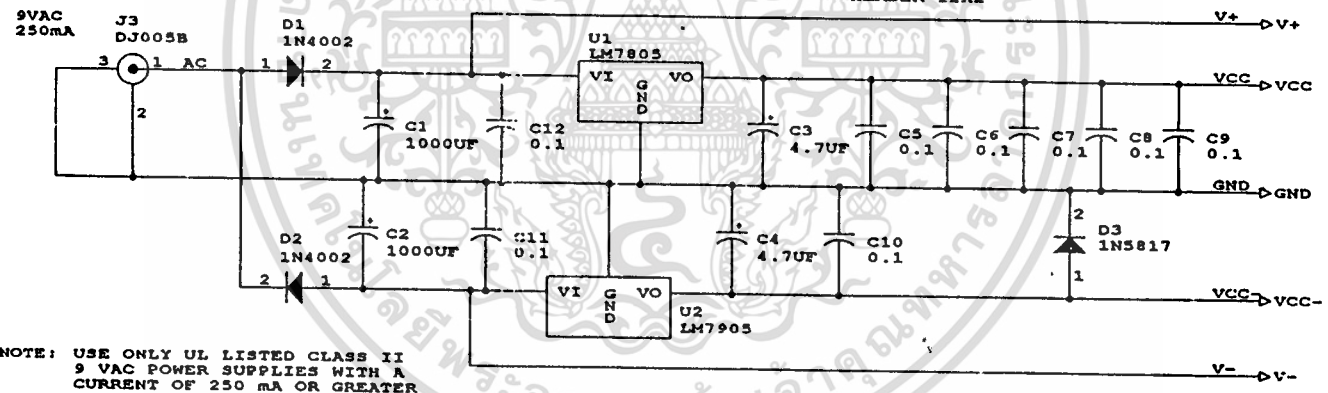
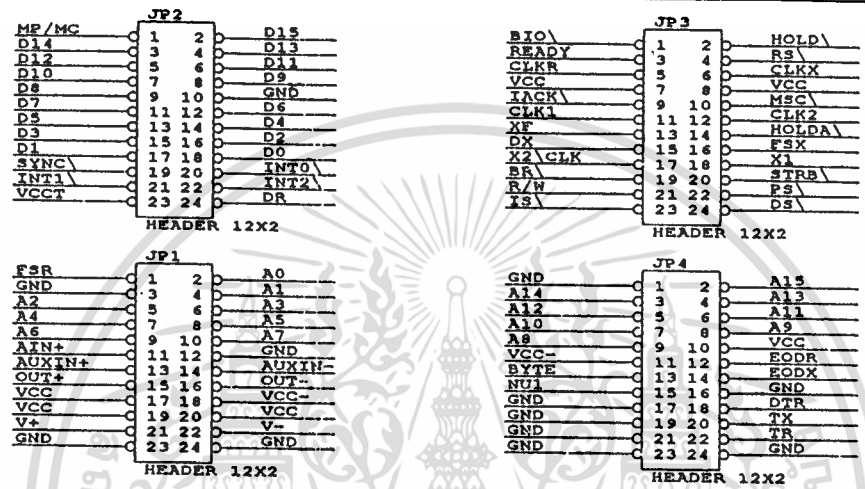
ปัญหาที่ประสบในการทำโครงงานนี้ ก็คือในการเขียนโปรแกรมในการสั่งงานให้ไมโครโปรเซสเซอร์ทำงาน, ความเร็วในการทำงาน และการแสดงผลที่ยังไม่มีประสิทธิภาพและสวยงามพอ ซึ่งผู้จัดทำเชื่อว่าเมื่อมีการพัฒนาแก้ไขข้อบกพร่องเหล่านี้แล้วนั้น จะทำให้โครงงานนี้สมบูรณ์แบบยิ่งขึ้น ทำให้นักศึกษาและพร้อมที่จะนำไปใช้งานจริงได้ดียิ่งขึ้น



ภาคผนวก ก.
แสดงวงจรในส่วนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

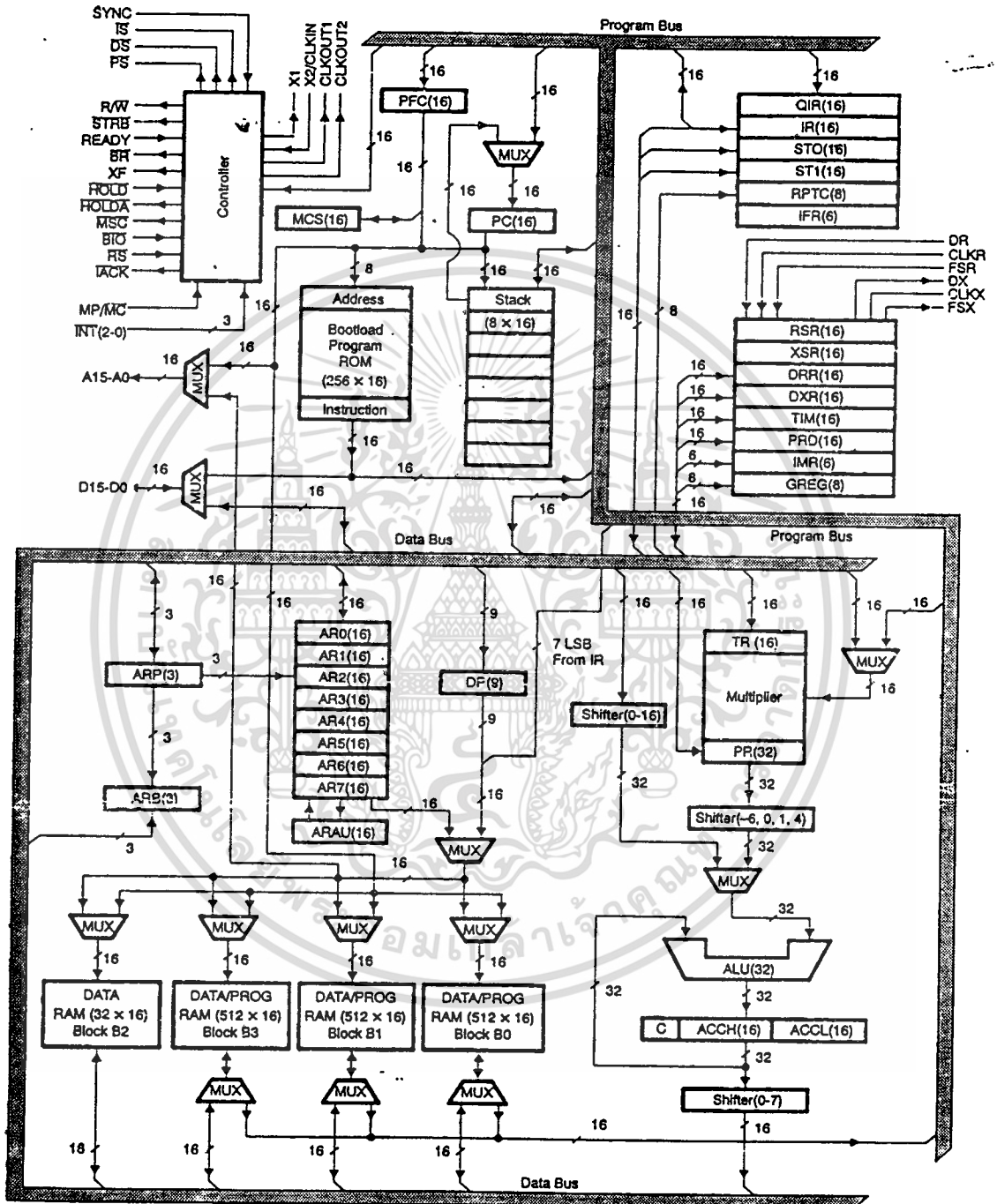




NOTE: USE ONLY UL LISTED CLASS II 9 VAC POWER SUPPLIES WITH A CURRENT OF 250 mA OR GREATER

TEXAS INSTRUMENTS		
Title		
TMS320C2X DSP STARTERS KIT		
Size	Document Number	REV
A	2617627	A
Date:	January 21, 1993	sheet 3 of 3

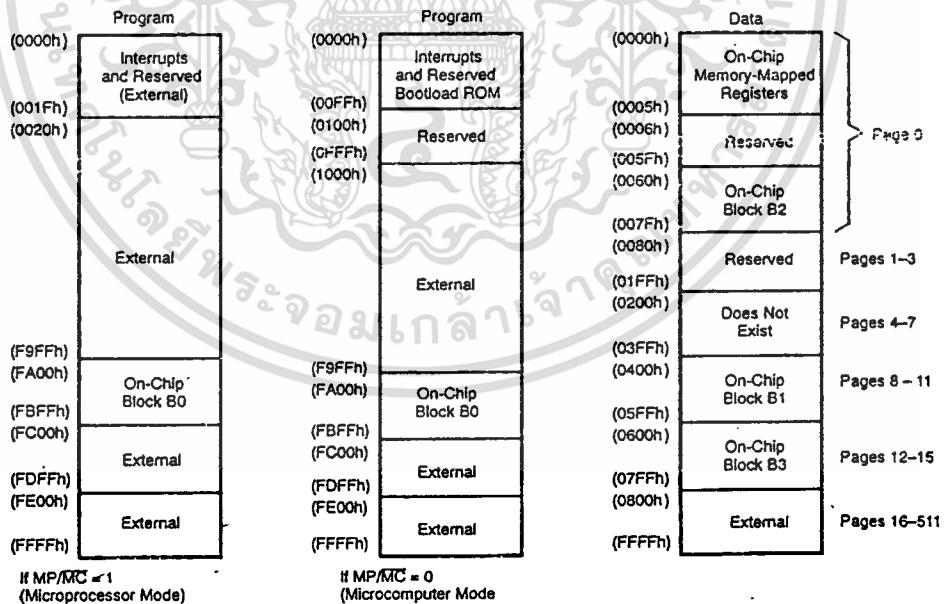
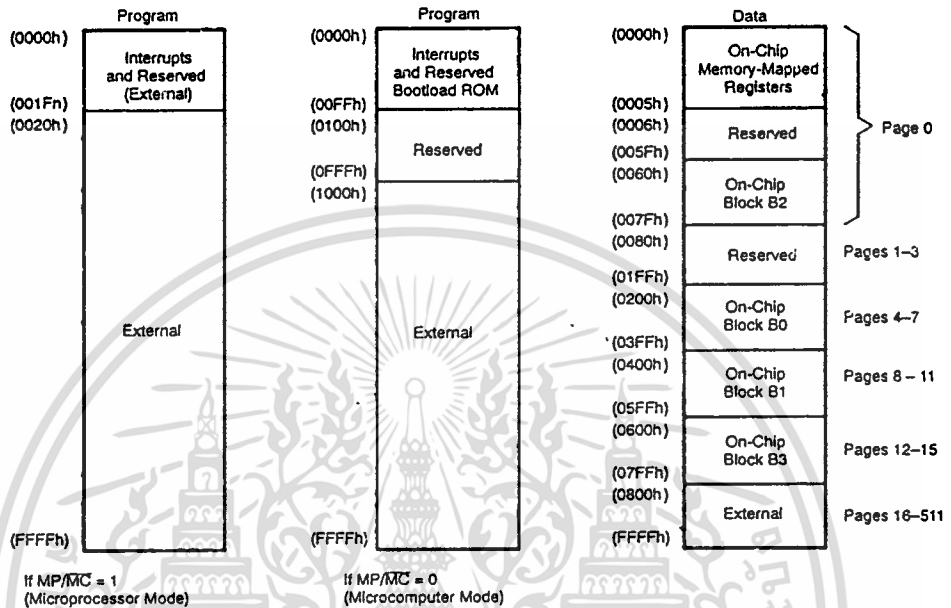
TMS320C26 Block Diagram



NOTE: Shaded areas indicate a bus.

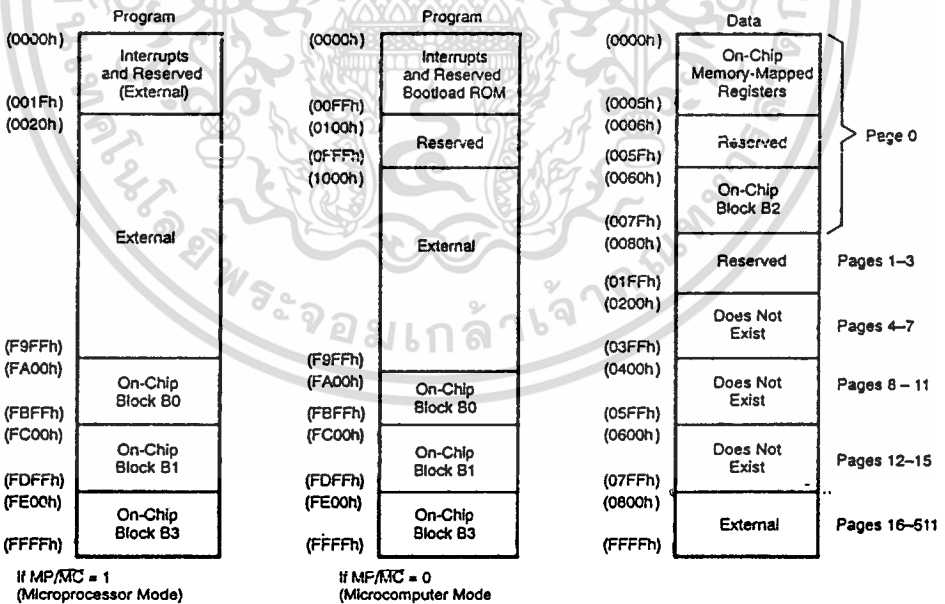
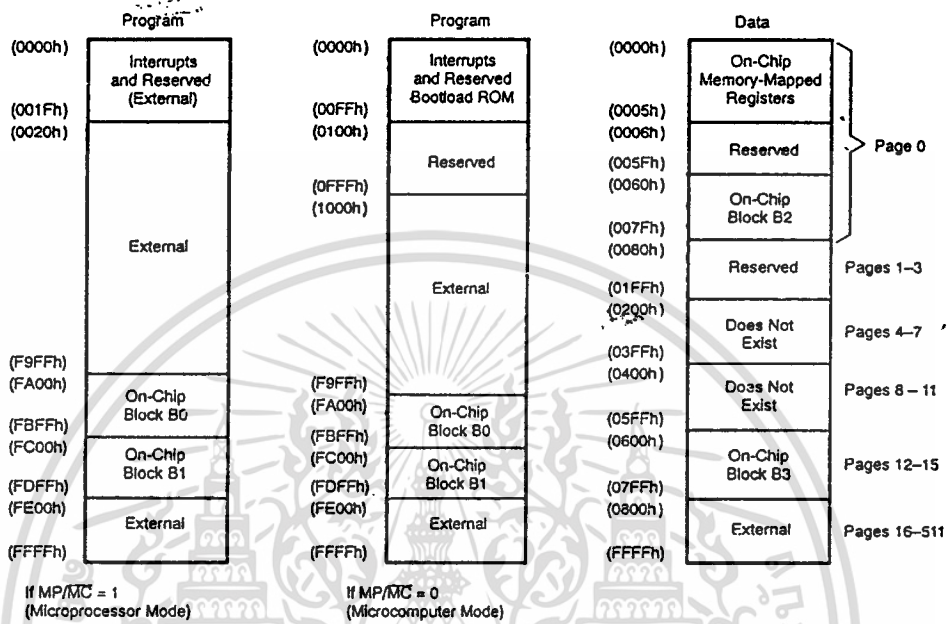
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26 Memory Maps



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26 Memory Maps (continued)





ภาคผนวก ข.
รายละเอียดของอุปกรณ์ที่สำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFNL DIGITAL SIGNAL PROCESSOR

SPRS013 – JUNE 1989 – REVISED AUGUST 1992



Caution. This device contains circuits to protect its inputs and outputs against damage due to high static voltages or electrostatic fields. These circuits have been qualified to protect this device against electrostatic discharges (ESD) of up to 2 kV according to MIL-STD-883C, Method 3015; however, it is advised that precautions be taken to avoid application of any voltage higher than maximum rated voltages to these high-impedance circuits. During storage or handling, the device leads should be shorted together or the device should be placed in conductive foam. In a circuit, unused inputs should always be connected to an appropriate logic voltage level, preferably either V_{CC} or ground. Specific guidelines for handling devices of this type are contained in the publication "Guidelines for Handling Electrostatic-Discharge Sensitive (ESDS) Devices and Assemblies" available from Texas Instruments.

absolute maximum ratings over specified temperature range (unless otherwise noted) †

Supply voltage range, V_{CC} ‡	–0.3 V to 7 V
Input voltage range	–0.3 V to 7 V
Output voltage range	–0.3 V to 7 V
Continuous power dissipation	1.5 W
Operating free-air temperature range	0°C to 70°C
Storage temperature range	–55°C to 150°C

† Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ All voltage values are with respect to V_{SS} .

recommended operating conditions

		MIN	NOM	MAX	UNIT	
V_{CC}	Supply voltage	4.75	5	5.25	V	
V_{SS}	Supply voltage		0		V	
V_{IH}	High-level input voltage	All inputs except CLKIN/CLKX/CLKR / INT (0 - 2)		2.35	$V_{CC} - 0.3$	V
		INT (0 - 2)		2.5	$V_{CC} - 0.3$	V
		CLKIN/CLKX/CLKR		3.5	$V_{CC} - 0.3$	V
V_{IL}	Low-level input voltage	All inputs except CLKIN		–0.3	0.8	V
		CLKIN		–0.3	0.8	V
I_{OH}	High-level output current			300	μ A	
I_{OL}	Low-level output current			2	mA	
T_A	Operating free-air temperature	0		70	°C	

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

PARAMETER		TEST CONDITIONS	MIN	YP †	MAX	UNIT
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN}, I_{OH} = \text{MAX}$	2.4	3		V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN}, I_{OL} = \text{MAX}$		0.3	0.6	V
I_Z	Three-state current	$V_{CC} = \text{MAX}$	–20		20	μ A
I_I	Input current	$V_I = V_{SS} \text{ to } V_{CC}$	–10		10	μ A
I_{CC}	Supply current	Normal		110	220	mA
		Idle/HOLD		70	100	
C_I	Input capacitance	$T_A = 0^\circ\text{C}, V_{CC} = \text{MAX}, f_x = \text{MAX}$		15		pF
C_O	Output capacitance			15		pF

† All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013 – JUNE 1989 – REVISED AUGUST 1992

CLOCK CHARACTERISTICS AND TIMING

The TMS320C26 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 2). The frequency of CLKOUT1 is one-fourth the crystal fundamental frequency. The crystal should be either fundamental or overtone mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF. Note that overtone crystals require an additional tuned LC circuit (see the application report, *Hardware Interfacing to the TMS320C25*).

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_x Input clock frequency	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$	6.7	40.96		MHz
f_{sx} Serial port frequency	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$	0*	5.120		kHz
C1, C2	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		10		pF

* The serial port is tested at a minimum frequency of 1.25MHz. However, the serial port is fully static and will properly function down to $f_{sx} = 0\text{Hz}$

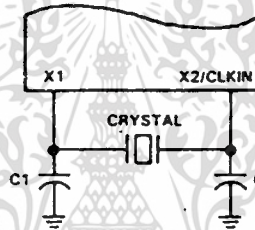


FIGURE 2. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the following table.

switching characteristics over recommended operating conditions (see Note 3)

PARAMETER	MIN	TYP	MAX	UNIT
$t_{c(C)}$ CLKOUT1/CLKOUT2 cycle time	97.7		597	ns
$t_{d(CIH-C)}$ CLKIN high to CLKOUT1/CLKOUT2/STRB high/low	5		30	ns
$t_{f(C)}$ CLKOUT1/CLKOUT2/STRB fall time			5	ns
$t_{r(C)}$ CLKOUT1/CLKOUT2/STRB rise time			5	ns
$t_w(CL)$ CLKOUT1/CLKOUT2 low pulse duration	$2Q - 8$	$2Q$	$2Q + 8$	ns
$t_w(CH)$ CLKOUT1/CLKOUT2 high pulse duration	$2Q - 8$	$2Q$	$2Q + 8$	ns
$t_{d(C1-C2)}$ CLKOUT1 high to CLKOUT2 low, CLKOUT2 high to CLKOUT1 high, etc.	$Q - 5$	Q	$Q + 5$	ns

NOTE 3: $Q = 1/4t_{c(C)}$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A - JUNE 1989 - REVISED AUGUST 1992

timing requirements over recommended operating conditions (see Note 3)

	MIN	NOM	MAX	UNIT
$t_{c(CI)}$ CLKIN cycle time	24.4		150	ns
$t_f(CI)$ CLKIN fall time			5*	ns
$t_r(CI)$ CLKIN rise time			5*	ns
$t_w(CIL)$ CLKIN low pulse duration, $t_c(CI) = 50$ ns (see Note 4)	20			ns
$t_w(CIH)$ CLKIN high pulse duration, $t_c(CI) = 50$ ns (see Note 4)	20			ns
$t_{su}(S)$ SYNC setup time before CLKIN low	5	Q-8		ns
$t_h(S)$ SYNC hold time from CLKIN low	8			ns

*Value not tested

NOTES: 3. $Q = 1/4 t_{c(CI)}$.

4. CLKIN duty cycle $(t_r(CI) + t_w(CIH)) / t_c(CI)$ must be within 40-60%.

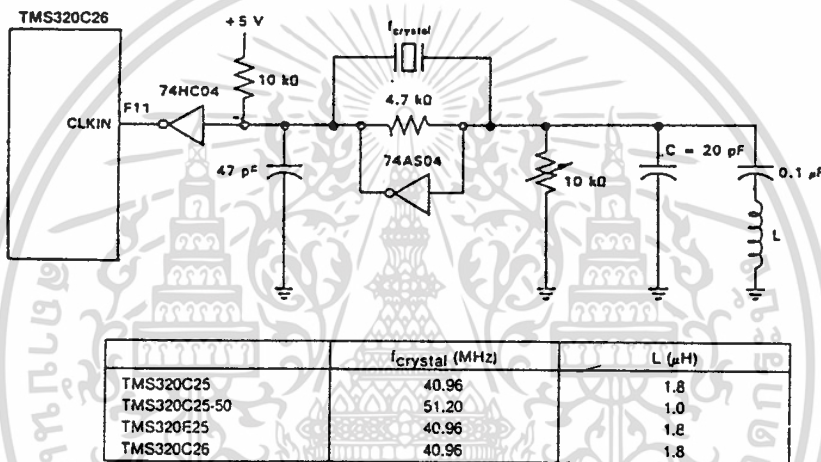


FIGURE 3. EXTERNAL CLOCK OPTION

Shown above is a crystal oscillator circuit suitable for providing the input clock signal to the TMS320C25, TMS320C26, TMS320E25, and TMS320C25-50. Please refer to *Hardware Interfacing to the TMS320C25* (document number SPRA014A) for details on circuit operation.

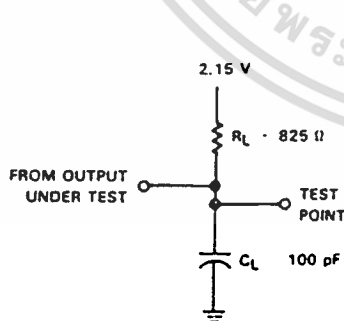


FIGURE 4. TEST LOAD CIRCUIT

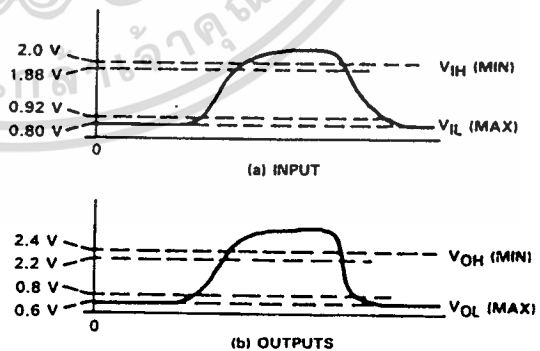


FIGURE 5. VOLTAGE REFERENCE LEVELS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFNL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions (see Note 3)

PARAMETER	MIN	TYP	MAX	UNIT
$t_{d(C1 S)}$ \overline{STRB} from CLKOUT1 (if \overline{STRB} is present)	0 - 6	Q	Q - 6	ns
$t_{d(C2 S)}$ CLKOUT2 to \overline{STRB} (if \overline{STRB} is present)	- 6	0*	6	ns
$t_{su(A)}$ Address setup time before \overline{STRB} low (see Note 5)	Q - 12			ns
$t_h(A)$ Address hold time after \overline{STRB} high (see Note 5)	Q - 8			ns
$t_w(SL)$ \overline{STRB} low pulse duration (no wait states, see Note 6)	2Q - 5		2Q + 5	ns
$t_w(SH)$ \overline{STRB} high pulse duration (between consecutive cycles, see Note 6)	2Q - 5		2Q + 5	ns
$t_{su(D)W}$ Data write setup time before \overline{STRB} high (no wait states)	2Q - 20			ns
$t_h(D)W$ Data write hold time from \overline{STRB} high	Q - 10	Q		ns
$t_{en(D)}$ Data bus starts being driven after \overline{STRB} low (write cycle)	0*			ns
$t_{dis(D)}$ Data bus three-state after \overline{STRB} high (write cycle)		Q	Q + 15*	ns
$t_{d(MSC)}$ \overline{MSC} valid from CLKOUT1	- 12	0	12	ns

*Value derived from characterization data and not tested

NOTES 3. $Q = 1/4 t_{Cl(C)}$

5. A15-A0, \overline{PS} , \overline{DS} , \overline{IS} , $\overline{R\overline{W}}$, and \overline{BR} timings are all included in timings referenced as "address."

6. Delays between CLKOUT1, CLKOUT2 edges and \overline{STRB} edges track each other, resulting in $t_w(SL)$ and $t_w(SH)$ being 2Q with no wait states.

timing requirements over recommended operating conditions (see Note 3)

	MIN	NOM	MAX	UNIT
$t_a(A)$ Read data access time from address time (read cycle, see Notes 5 and 7)			3Q - 35	ns
$t_{su(DIR)}$ Data read setup time before \overline{STRB} high	23			ns
$t_h(DIR)$ Data read hold time from \overline{STRB} high	0			ns
$t_{d(SL-R)}$ \overline{READY} valid after \overline{STRB} low (no wait states)			Q - 20	ns
$t_{d(C2H-R)}$ \overline{READY} valid after CLKOUT2 high			Q - 20	ns
$t_h(SL-R)$ \overline{READY} hold time after \overline{STRB} low (no wait states)	Q - 3			ns
$t_h(C2H-R)$ \overline{READY} hold after CLKOUT2 high	Q - 3			ns
$t_{d(M-R)}$ \overline{READY} valid after \overline{MSC} valid			2Q - 25	ns
$t_h(M-R)$ \overline{READY} hold time after \overline{MSC} valid	0			ns

NOTES: 3. $Q = 1/4 t_{Cl(C)}$

5. A15-A0, \overline{PS} , \overline{DS} , \overline{IS} , $\overline{R\overline{W}}$, and \overline{BR} timings are all included in timings referenced as "address."

7. Read data access time is defined as $t_a(A) = t_{su(A)} + t_w(SL) - t_{su(DIR)}$.

\overline{RS} , \overline{INT} , \overline{BIO} , and XF TIMING

switching characteristics over recommended operating conditions (see Note 3)

PARAMETER	MIN	TYP	MAX	UNIT
$t_{d(RS)}$ CLKOUT1 low to reset state entered			22*	ns
$t_{d(IACK)}$ CLKOUT1 to \overline{IACK} valid	- 6	0	12	ns
$t_{d(XF)}$ XF valid before falling edge of \overline{STRB}	Q - 15			ns

*Value derived from characterization data and not tested

NOTES: 3. $Q = 1/4 t_{Cl(C)}$

8. \overline{RS} , \overline{INT} , and \overline{BIO} are asynchronous inputs and can occur at any time during a clock cycle. However, if the specified setup time is met, the exact sequence shown in the timing diagrams will occur.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFNL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

timing requirements over recommended operating conditions (see Note 3)

	MIN	NOM	MAX	UNIT
$t_{su(IN)}$ $\overline{INT}/\overline{BIO}/\overline{RS}$ setup before CLKOUT1 high	32			ns
$t_h(IN)$ $\overline{INT}/\overline{BIO}/\overline{RS}$ hold after CLKOUT1 high	0			ns
$t_f(IN)$ $\overline{INT}/\overline{BIO}$ fall time			8*	ns
$t_w(IN)$ $\overline{INT}/\overline{BIO}$ low pulse duration	$t_c(C)$			ns
$t_w(RS)$ \overline{RS} low pulse duration	$3t_c(C)$			ns

*Value not tested

NOTES: 3. $Q = 1/4t_c(C)$.

8. \overline{RS} , \overline{INT} , and \overline{BIO} are asynchronous inputs and can occur at any time during a clock cycle. However, if the specified setup time is met, the exact sequence shown in the timing diagrams will occur.

HOLD TIMING

switching characteristics over recommended operating conditions (see Note 3)

PARAMETER	MIN	TYP	MAX	UNIT
$t_d(C1L-A)$ \overline{HOLDA} low after CLKOUT1 low	0		10	ns
$t_{dis}(A-L-A)$ \overline{HOLDA} low to address three-state		0*		ns
$t_{dis}(C1L-A)$ Address three-state after CLKOUT1 low (HOLD mode, see Note 9)			20*	ns
$t_d(HH-AH)$ \overline{HOLD} high to \overline{HOLDA} high			25	ns
$t_{en}(A-C1L)$ Address driven before CLKOUT1 low (HOLD mode, see Note 9)			8*	ns

*Value derived from characterization data and not tested

NOTES: 3. $Q = 1/4t_c(C)$.

9. A15-A0, \overline{PS} , \overline{DS} , \overline{IS} , \overline{STRB} , and R/\overline{W} timings are all included in timings referenced as "address."

timing requirements over recommended operating conditions (see Note 3)

	MIN	NOM	MAX	UNIT
$t_d(C2H-H)$ \overline{HOLD} valid after CLKOUT2 high			$Q - 24$	ns

NOTE 3: $Q = 1/4t_c(C)$.

SERIAL PORT TIMING

switching characteristics over recommended operating conditions (see Note 3)

PARAMETER	MIN	TYP	MAX	UNIT
$t_d(CH-DX)$ DX valid after CLKX rising edge (see Note 10)			75	ns
$t_{df}(FL-DX)$ DX valid after FSX falling edge (TXM = 0, see Note 10)			40	ns
$t_d(CH-FS)$ FSX valid after CLKX rising edge (TXM = 1)			40	ns

NOTES: 3. $Q = 1/4t_c(C)$.

10. The last occurrence of FSX falling and CLKX rising.

timing requirements over recommended operating conditions (see Note 3)

	MIN	NOM	MAX	UNIT
$t_c(SCK)$ Serial port clock (CLKX/CLKR) cycle time**	200			ns
$t_f(SCK)$ Serial port clock (CLKX/CLKR) fall time			25*	ns
$t_r(SCK)$ Serial port clock (CLKX/CLKR) rise time			25*	ns
$t_w(SCK)$ Serial port clock (CLKX/CLKR) low pulse duration (see Note 11)	80			ns
$t_w(SCK)$ Serial port clock (CLKX/CLKR) high pulse duration (see Note 11)	80			ns
$t_{su}(FS)$ FSX/FSR setup time before CLKX/CLKR falling edge (TXM = 0)	18			ns
$t_h(FS)$ FSX/FSR hold time after CLKX/CLKR falling edge (TXM = 0)	20			ns
$t_{su}(DR)$ DR setup time before CLKR falling edge	10			ns
$t_h(DR)$ DR hold time after CLKR falling edge	20			ns

*Value not tested

**The serial port is tested at a minimum frequency of 1.25MHz. However, the serial port is fully static and will properly function down to $f_{sx} = 0$ Hz

NOTES: 3. $Q = 1/4t_c(C)$.

11. The duty cycle of the serial port clock must be within 40-60%.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

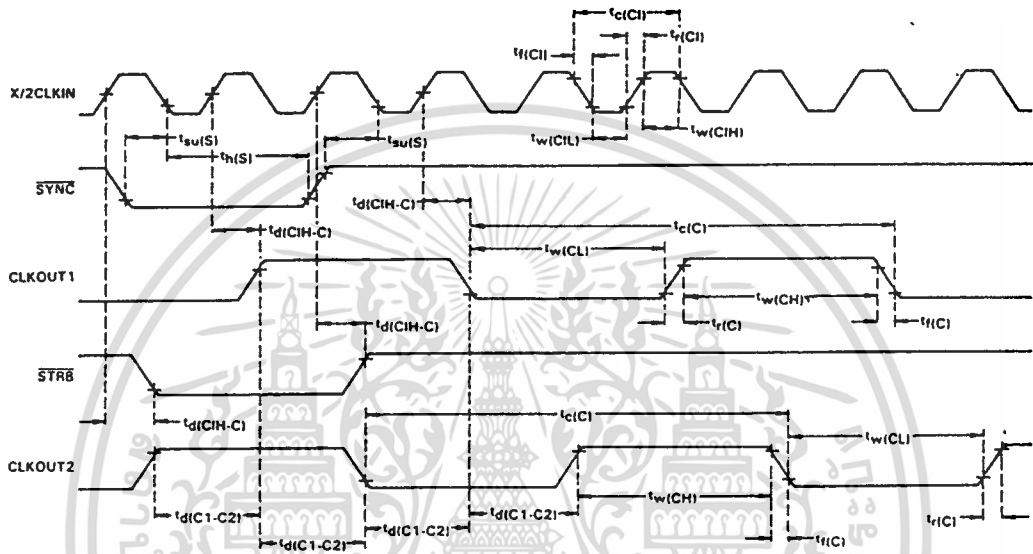
TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

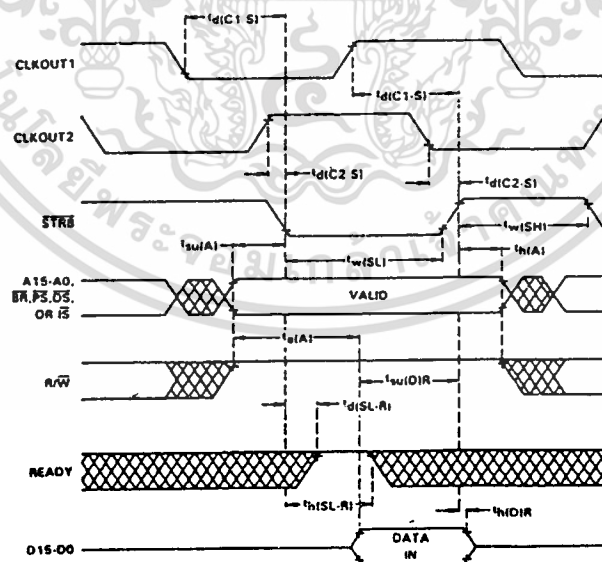
TIMING DIAGRAMS

Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing



memory read timing

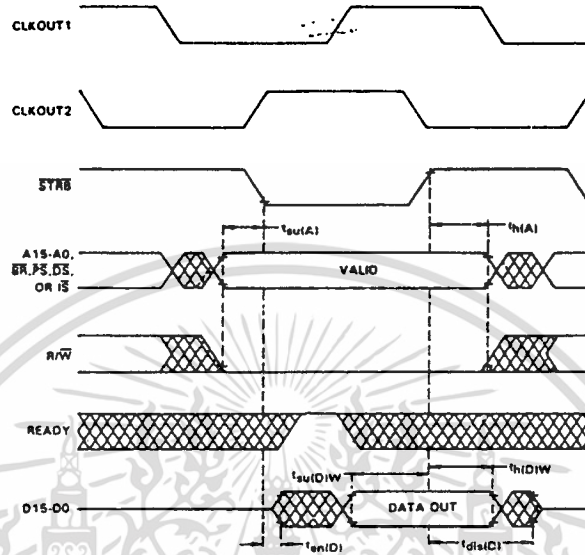


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

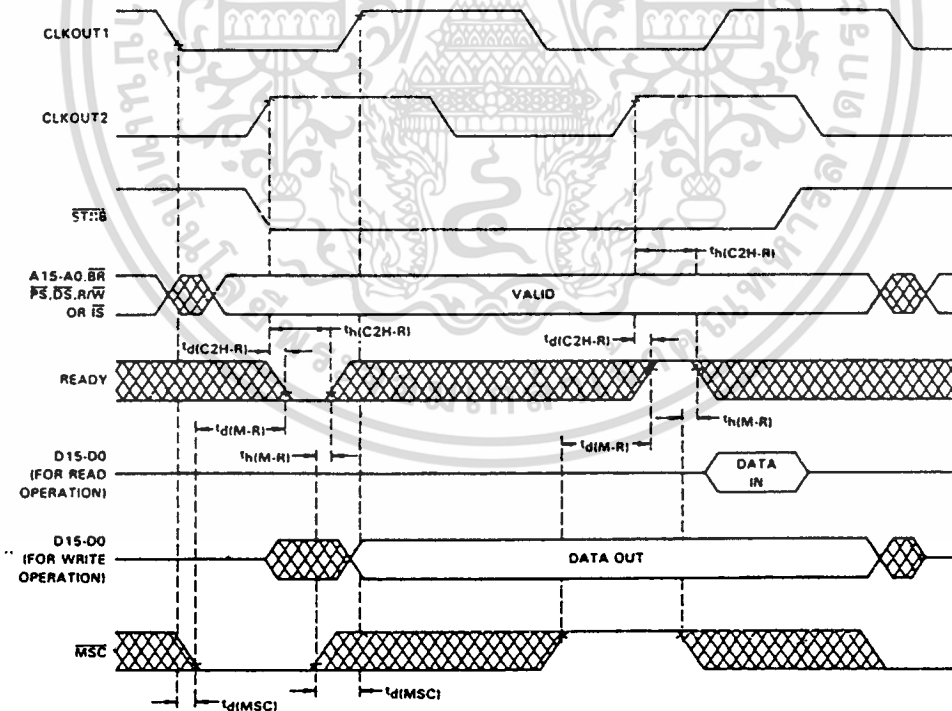
TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

memory write timing



one wait-state memory access timing

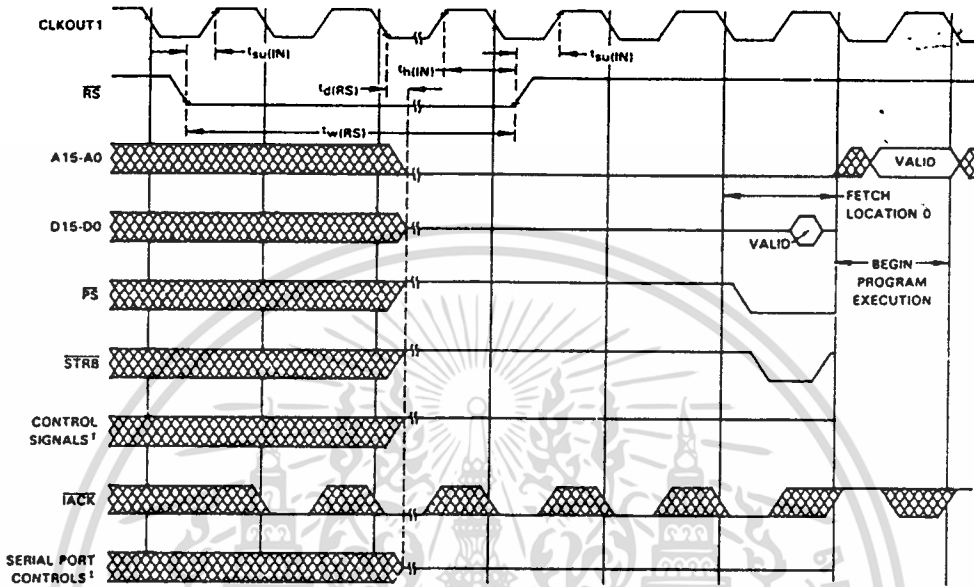


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A - JUNE 1989 - REVISED AUGUST 1992

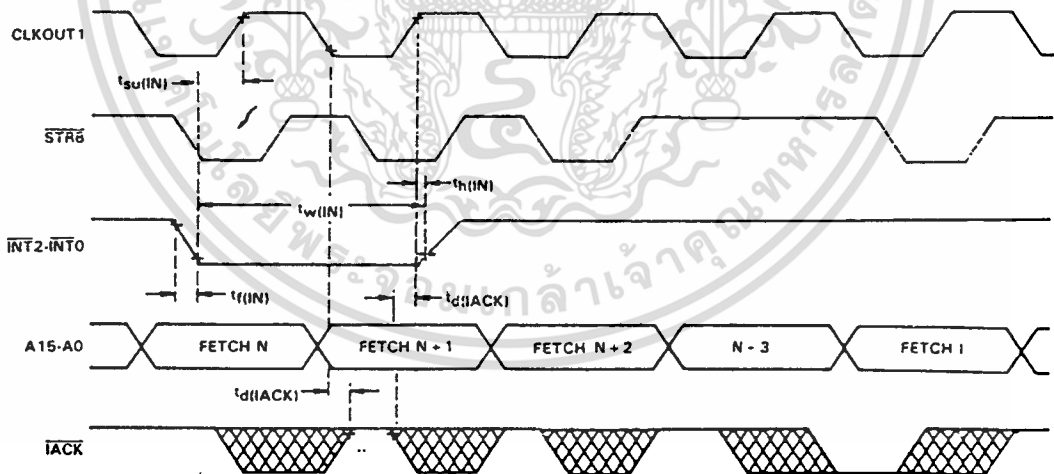
reset timing



[†]Control signals are DS, IS, R/W, and XF.

[‡]Serial port controls are DX and FSX.

interrupt timing

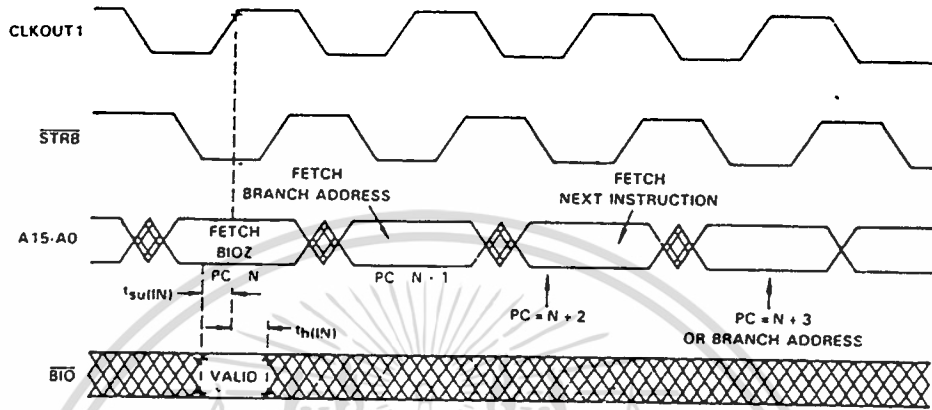


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

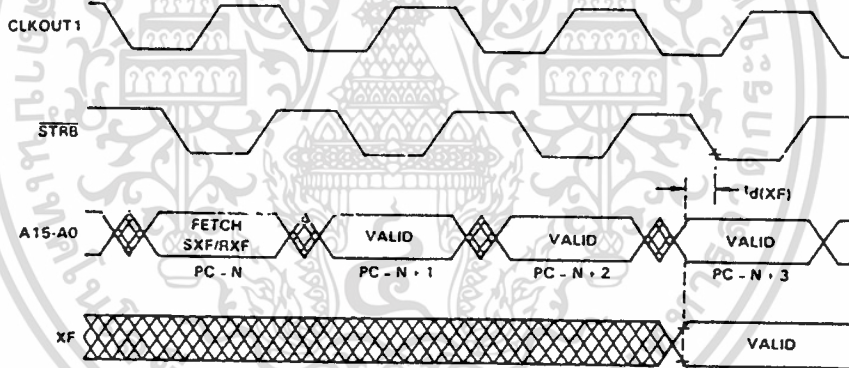
TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

BIO timing



external flag timing

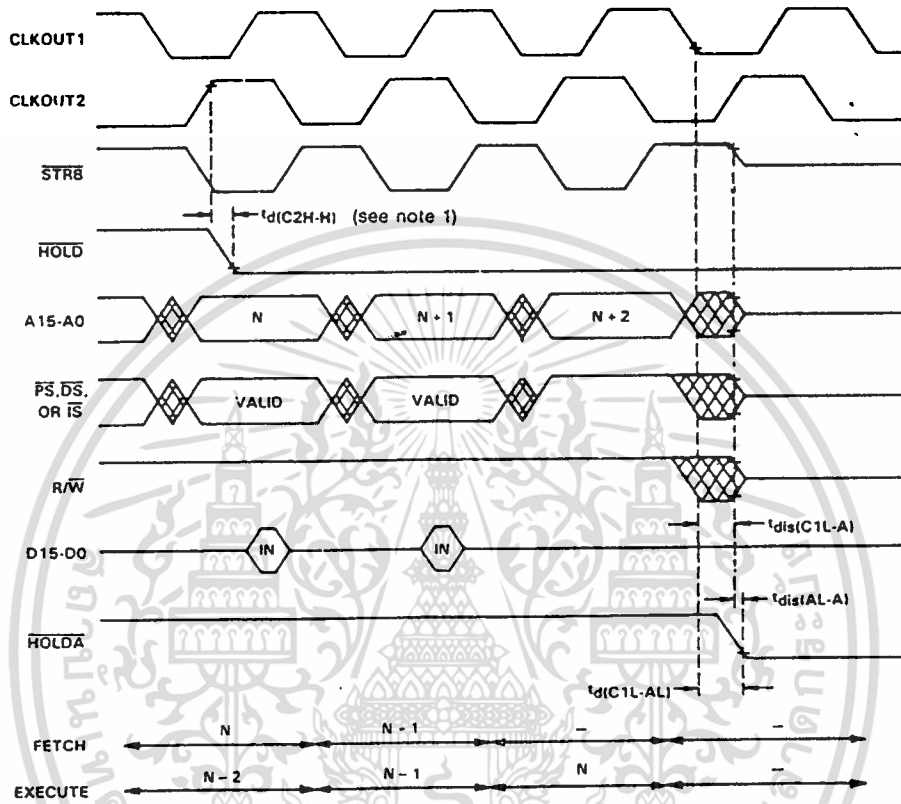


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

HOLD timing (part A)



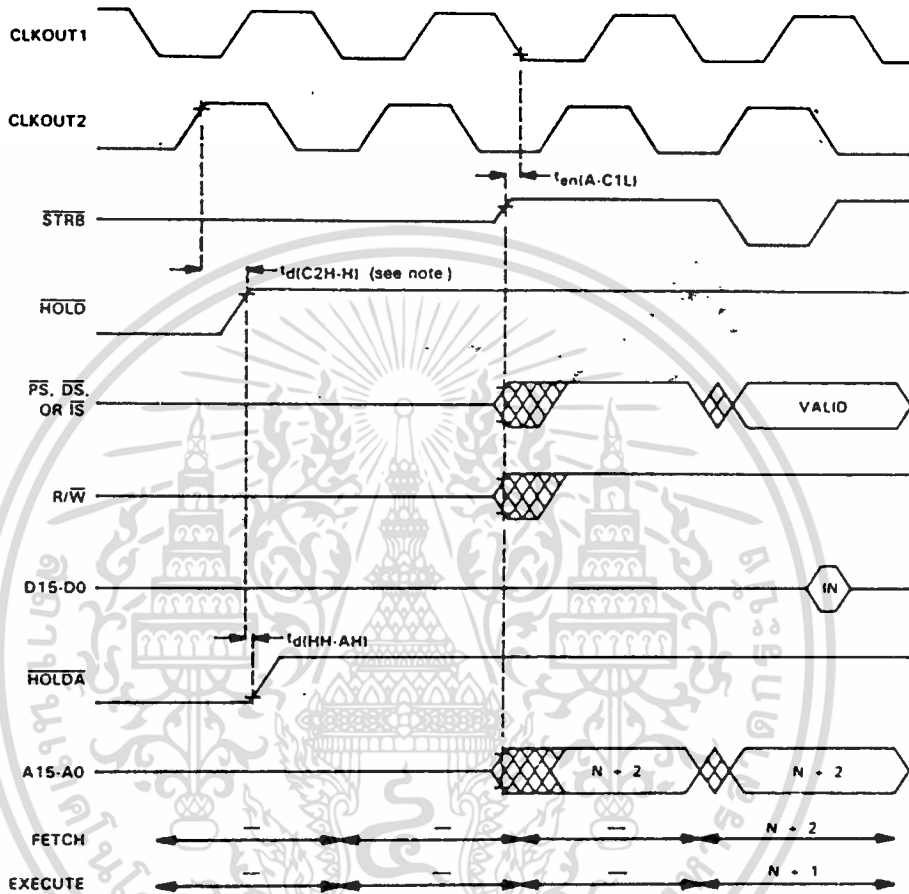
Note: \overline{HOLD} is an asynchronous input and can occur at any time during a clock cycle. If the specified timing is met, the exact sequence shown will occur; otherwise a delay of one CLKOUT2 cycle will occur.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

HOLD timing (part B)



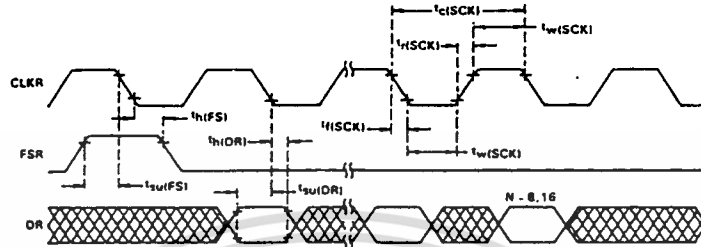
Note: HOLD is an asynchronous input and can occur at any time during a clock cycle. If the specified timing is met, the exact sequence shown will occur; otherwise a delay of one CLKOUT2 cycle will occur.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

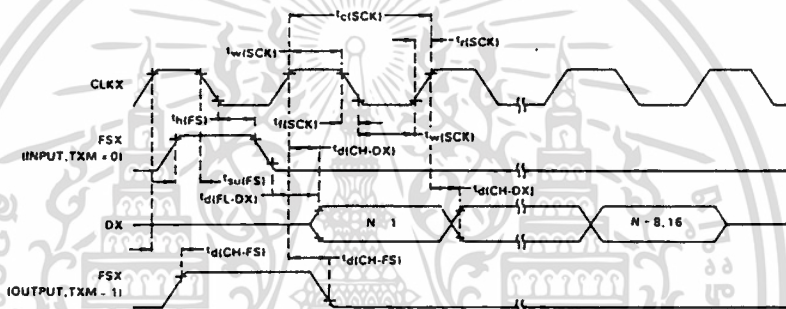
TMS320C26BFNL DIGITAL SIGNAL PROCESSOR

SPRS013A – JUNE 1989 – REVISED AUGUST 1992

serial port receive timing



serial port transmit timing



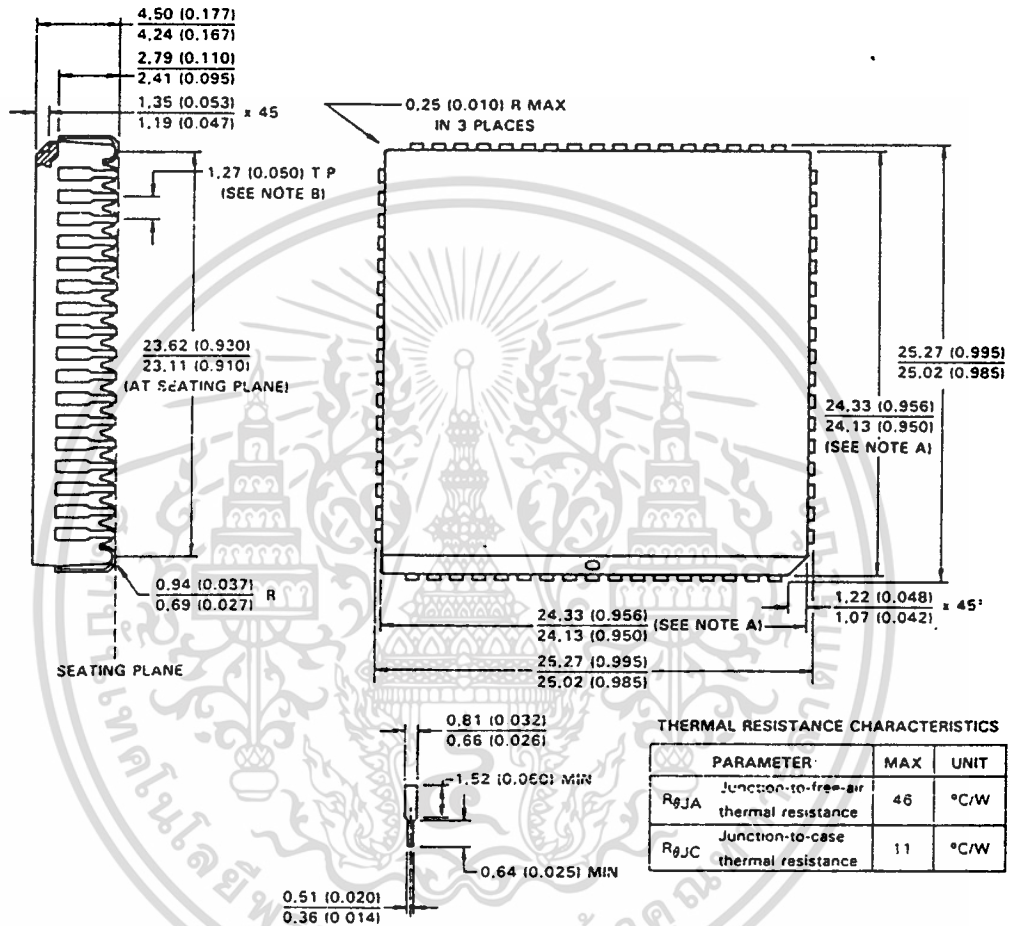
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C26BFNL DIGITAL SIGNAL PROCESSOR

SPRS013A - JUNE 1989 - REVISED AUGUST 1992

MECHANICAL DATA

68-pin plastic leaded chip carrier package

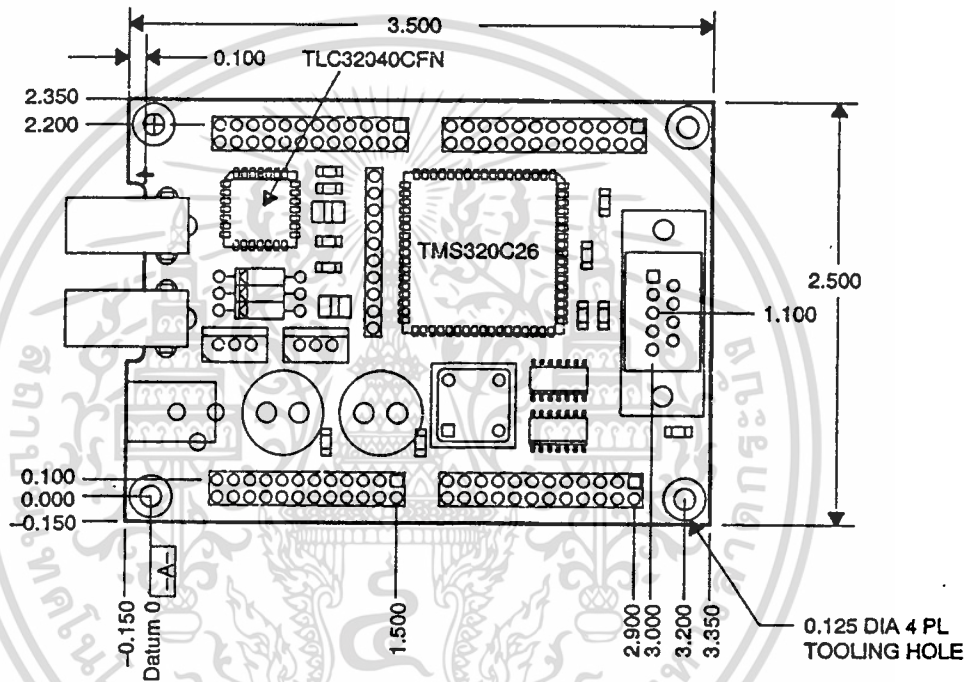


NOTES A Centerline of center pin each side is within 0.10 (0.004) of package centerline as determined by this dimension.
B Location of each pin is within 0.127 (0.005) of true position with respect to center pin on each side

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHECALLY IN INCHES.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320C2x DSP Starter Kit



Note:

Dimensions are in inches.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The seal of Rajabhat Buriram University is a circular emblem. It features a central sunburst with rays emanating from a central point. Below the sunburst are two traditional Thai stupas (chedis) flanking a central decorative element. The entire emblem is surrounded by a circular border containing Thai text. The text at the top of the border reads "มหาวิทยาลัยราชภัฏบุรีรัมย์" (Mahavithayalai Rajabhat Buriram) and the text at the bottom reads "พระจอมเกล้าเจ้าคุณทหารลาดกระบัง" (Phra Chomklao Jaokunthar Ladkrabang).

ภาคผนวก ค.

แสดง List Programs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----;
;          JACKY.ASM          ;
;-----;

YES      .set      1
NO       .set      0
FFT_S    .set      256
FFT_S-1  .set      255
FFT_S/2  .set      128
(FFT_S/2)-1 .set    127
;-----;
AIC_1    .set      0x0C18
AIC_2    .set      0x0205
AIC_3    .set      0x264e
AIC_CMD  .set      0x0003
;-----;
BCMD     .set      0xFA10
BXMIT    .set      0xFA12
BXMIT16  .set      0xFA14
BRCV     .set      0xFA16
BRCV16   .set      0xFA18
BCXMIT   .set      0xFA1A
;-----;
STAT1    .set      0x72
ACCU_lo  .set      0x78
ACCU_hi  .set      0x79
REAL     .set      0x7a
IMAG     .set      0x7b
TEMPX    .set      0x7c
AUX0     .set      0x7d
AUX1     .set      0x7e
;-----;
; 2. SECONDARY
;-----;
        .include  "mmregs.asm"
        .ps      0xfa00
;B      start
;B      start
;B      start
;B      start
;B      start
;B      start
        .ps      0fa0ah
        B        RINT
        eint
        ret
; Begin TRAP/DSKD Kernal
;-----;
; APPLICATION CODE
;-----;
        .ps      0xFB00
        .entry
;-----;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

start:  sxf
        ssxm
        sovm
        ldpk      0
        fort      0
        rtxm
        sfsm
        lack      0x80
        sach      DXR
        sacl      GREG
        lrlk      ARO,0xFFFF
        rptk      255
        lac       *,0,ARO
        conf      1

```

```

;-----
AIC_RS  lack      0x20
        sacl      IMR
        idle
        lalk      AIC_1
        call      AIC_2nd
        lalk      AIC_2
        call      AIC_2nd
        lalk      AIC_3
        call      AIC_2nd
        lalk      AIC_CMD
        call      AIC_2nd
;-----

```

```

        lark      AR7,0
        lack      0x10
        sacl      IMR
;-----

```

```

        lark      AR7,0
FFT:    lrlk      ARO,FFT_S/2
        larp      ARO
new_stg lrlk      AR1,_D_base
        lrlk      AR2,_D_base
        lrlk      AR3,_T_base+1
        lrlk      AR4,FFT_S/2
        b         n_DFT2,*,AR1
DFT:    mar       *BR0+,AR5
        lark      AR5,1
        mar       *BR0+,AR1

```

```

;-----
; AR1=Top AR2=Bottom AR3=Twiddle
;-----
BFLY:  lac      *,14,AR2
      add      *,14,AR1
      sach    *+,1,AR2
      sub      *,15
      sach    *+,1,AR1
      lac      *,14,AR2
      add      *,14,AR1
      sach    *+,1,AR2
      sub      *,15
      sach    *,1,AR5
      banz    OK,*BR0+,AR3
;-----
      mar      *+,AR2
      mar      *+
      mar      *0+
      mar      *0+,AR1
n_DFT2: mar      *0+
      mar      *0+,AR4
      banz    DFT,*0-,AR3
      larp    ARO
      mar      *BR0+
      banz    new_stg,*
      b       endFFT
;-----
OK     lt       *-,AR2
      mpy     *-
      ltp     *+,AR3
      mpy     *
      lts     *+,AR2
      mpy     *
      sach    *-,1,AR2
      ltp     *,AR3
      mpy     *BR0+,AR2
      apac
      sach    *+,1,AR2
      b       BFLY,*+,AR1
;-----

endFFT: larp    AR2
      lrlk   AR2,_D_base
      lrlk   AR3,FFT_S-1
      lrlk   ARO,FFT_S

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; WINDOW:
;-----

```

```

;mar      *BR0+
more_MAG
mar      *BR0-
lac      *BR0+,15
subh     *BR0+
add      *BR0-,15
sach     IMAG
mar      *+
mar      *BR0-
lac      *BR0+,15
subh     *BR0+
add      *BR0-,15,AR1
sach     REAL
sqra     IMAG
ltp      REAL
mpy      REAL
apac
addk     0x1
lark     AR1,22
rptk     31
norm     *-
larp     AR2
mar      *BR0-
sach     *,2
zals     *
sar      AR1,*
addh     *
rptk     10
sf1
; sach   *
; addh   *
sach     *
lac      *
andk     0xfffc,0
sac1     *BR0+
mar      *-
mar      *BR0+,AR3
banz     more_MAG,*-,AR2
;-----

```

```

BITREV: 1r1k  ARO,FFT_S
        1r1k  AR1,_D_base
        1r1k  AR2,_D_base+1
        1r1k  AR3,FFT_S-1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

more_BR lac    *+
        mar    *+,AR1
        sac1   *BR0+,0,AR3
        banz   more_BR,*-,AR2
        ;-----

```

```

MOVE_IO larp   AR7
        banz   MOVE_IO,*-,AR2
        ;-----

```

```

        lrlk   AR3,_D_base
        lrlk   AR4,_B_base
        lrlk   AR5,FFT_S-1
        lrlk   AR6,_B_base
        lrlk   AR7,FFT_S-1
        ;-----

```

```

        dint
        zac
        call   DAT2HOST
more_IO larp   AR2
        lar    AR2,*-,AR3
        lac    *,0,AR4
        sac1   *+,0,AR3

        rptk   7
        sfr    1
        ork    1
        call   DAT2HOST
        larp   AR3

        zac
        sach   *+,0
        sar    AR2,*+,AR5

        banz   more_IO,*-,AR4
        eint
        b      FFT
        ;-----

```

```

RINT:   sst1   STAT1
        larp   AR7
        banz   more_buf,*-,AR6
        lark   AR7,0
        lst1   STAT1
        ret

```

```

more_buf sac1   ACCU_lo
        sach   ACCU_hi
        zalh   *
        sach   DXR

```

```

;-----
lac      DRR
bit      TEMPX,15
bbz      NO_NVRT
neg
NO_NVRT
sac1     *+
lac      TEMPX
xork     1
sac1     TEMPX
zalh     ACCU_hi
adds     ACCU_lo
lst1     STAT1
eint
ret
*****
AIC_2nd  adlk     6,15
         idle
         sach     DXR
         idle
         sac1     DXR
         idle
         sac1     DXR,2
         idle
         eint
         ret
*****
DAT2HOST sac1     ACCU_lo
         sach     ACCU_hi
         sar      ARO,AUX0
         sar      AR1,AUX1
         call     BXMIT
         zals     ACCU_lo
         addh     ACCU_hi
         lar      ARO,AUX0
         lar      AR1,AUX1
         ret
;=====

.listoff
.ds      0x400
.include "dsk_twid.asm"
.liston
.end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

ปริญญาบัตรฉบับนี้ สำเร็จลงได้ด้วยความช่วยเหลือเป็นอย่างดีจาก ท่านอาจารย์ เกรียงไกร วงศ์โรจนภรณ์ ซึ่งเป็นอาจารย์ที่ปรึกษาตลอดการทำปริญญาบัตรนี้ ได้กรุณาให้ยืมชุดบอร์ด DSP และ เอกสารที่เกี่ยวข้องกับการทำปริญญาบัตร และคุณอาจารย์ท่านอื่นๆ ที่ได้ให้คำแนะนำอันมีค่าและประโยชน์ อย่างมาก ผู้จัดทำจึงขอขอบพระคุณคุณอาจารย์ทุกท่านที่ได้กล่าวนามมาข้างต้น และบุคคลอื่นๆ รวมทั้ง เพื่อนๆของผู้จัดทำเอง ที่ได้สนับสนุนการทำปริญญาบัตรนี้ ซึ่งไม่สามารถกล่าวรายนามได้ครบถ้วนทุกท่าน เป็นอย่างสูงไว้ ณ. ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ยืน ภู่วรวรรณ และ ไพศาล สงวนหมู่ “การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เน็ตเวิร์ค” กรุงเทพฯ : สำนักพิมพ์ ซีเอ็ดยูเคชั่น, 2528.
2. ชูชัย ธนสารตั้งเจริญ และ ทินกร ดุ๊ก “การสื่อสารข้อมูล” กรุงเทพฯ : สำนักพิมพ์ฟิสิกส์เซ็นเตอร์.
3. ธานินทร์ ภาวศาตนวนศ์ และ ทินกร ดุ๊ก “การอินเทอร์เฟส IBM/PC” กรุงเทพฯ : สำนักพิมพ์ฟิสิกส์เซ็นเตอร์.
4. Texas Instrument, “First Generation TMS32010 User’s Guide”,1988.
5. Texas Instrument, “Digital Signal Processing Products and Applications”,1988.
6. International Business Machines Corporation, “IBM Technical Reference”,1983.
7. Texas Instrument, “TMS320C26 User’s Guide”,1993.
8. Texas Instrument, “TMS320C26 DSP Starter Kit User’s Guide”,1993.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้